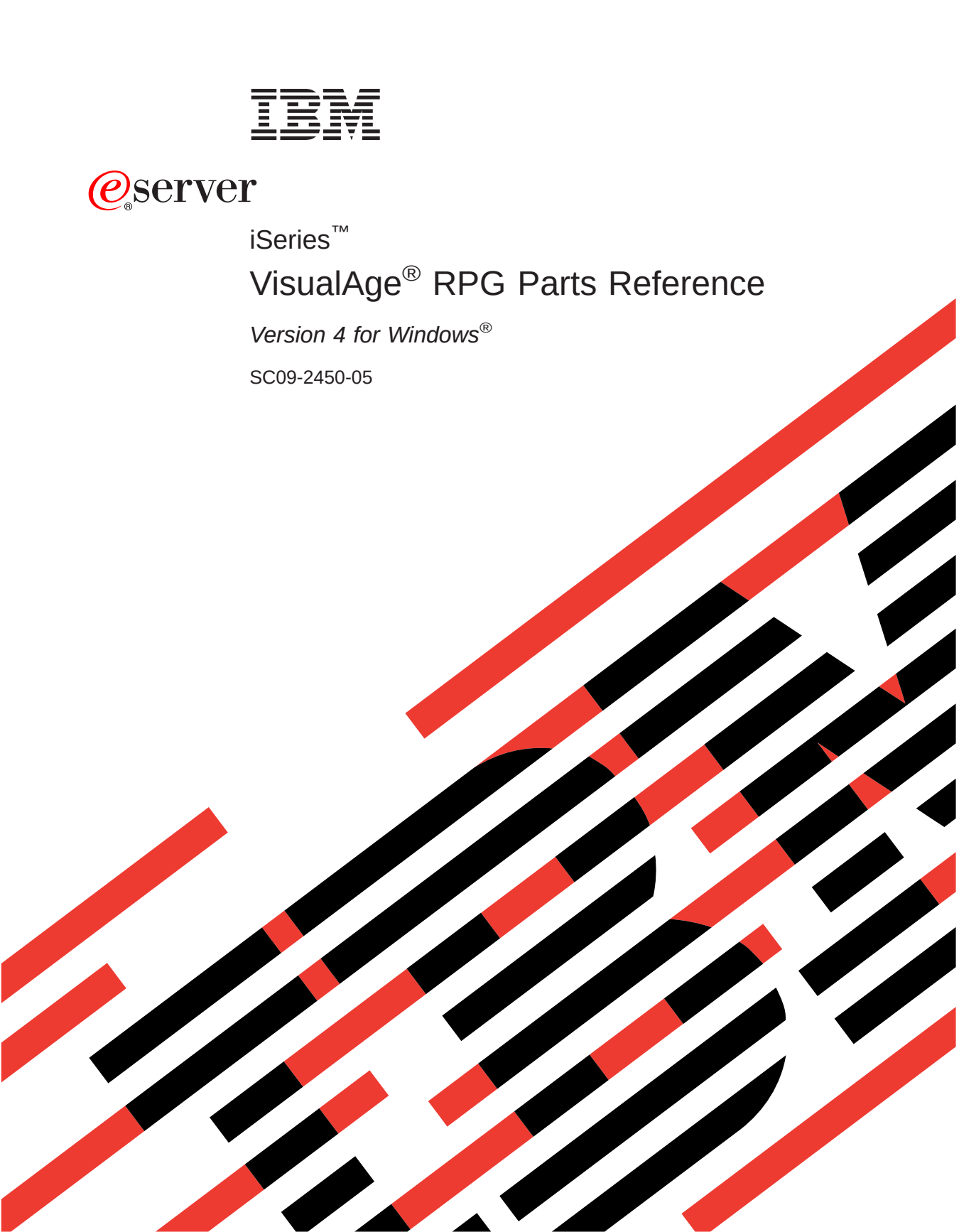

iSeries™

VisualAge® RPG Parts Reference

Version 4 for Windows®

SC09-2450-05

iSeries™

VisualAge® RPG Parts Reference

Version 4 for Windows®

SC09-2450-05

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 547.

Sixth Edition (May 2002)

This edition applies to Version 4 of IBM WebSphere Development Studio Client for iSeries and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC09-2450-04 .

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. You can send your comments to:

IBM Canada Ltd. Laboratory
Information Development
B3/KB7/8200/MKM
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

You can also send your comments electronically to IBM. See "How to Send Your Comments" on page x for a description of the methods.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About This Manual

This book provides reference information on VisualAge® RPG **parts** which you use to create your **graphical user interface** (GUI) application; the **attributes** and **events** associated with each part; and the **event attributes** that you can query in your VARPG application. It is intended for anyone who is programming applications using VisualAge RPG.

Before using this book, you should read *Getting Started with WebSphere Development Studio Client for iSeries*.

Highlighting Conventions Used in this Book

Throughout this book, all attribute, event, and event attribute names appear in **boldface** type, as do the first appearance of a term, other emphasized terms, and any references to text that appears on the VisualAge RPG interface. Variables appear in *italics*. VisualAge RPG operation codes appear in ALL CAPS.

Prerequisite and Related Information

Use the iSeries Information Center as your starting point for looking up iSeries and AS/400e technical information. You can access the Information Center in two ways:

- From the following Web site:

  `http://www.ibm.com/eserver/iseries/infocenter`
- From CD-ROMs that ship with your OS/400 order:

  iSeries Information Center, SK3T-4091-00. This package also includes the PDF versions of iSeries manuals, *iSeries Information Center: Supplemental Manuals*, SK3T-4092-00, which replaces the Softcopy Library CD-ROM.

The iSeries Information Center contains advisors and important topics such as CL commands, system application programming interfaces (APIs), logical partitions, clustering, Java™, TCP/IP, Web serving, and secured networks. It also includes links to related IBM® Redbooks and Internet links to other IBM Web sites such as the Technical Studio and the IBM home page.

The VisualAge RPG Library

The VisualAge RPG library contains the following publications:

Programming with VisualAge RPG

This book contains specific information about creating applications with VisualAge RPG. It describes the steps you have to follow at every stage of the application development cycle, from design to packaging and distribution. Programming examples are included to clarify the concepts and the process of developing VisualAge RPG applications.

VisualAge RPG Parts Reference

This book provides information on the VisualAge RPG **parts**, **part attributes**, **part events**, and **event attributes**. It is a reference for anyone who is developing applications using VisualAge RPG.

VisualAge RPG Language Reference

This book provides information about the RPG IV language as implemented using the VisualAge RPG compiler. It contains:

- Language fundamentals such as the character set, symbolic names and reserved words, compiler directives, and indicators
- Data types and data formats
- Error and exception handling
- Specifications
- Built-in functions, expressions, and operation codes.

For an overview of the entire product, see *Getting Started with WebSphere Development Studio Client for iSeries*.

For a list of **related publications**, see the Bibliography at the end of this book.

You can also find the most current information about IBM WebSphere Development Studio Client for iSeries on the following online source:

The WebSphere Development Studio Client Home Page
`ibm.com/software/ad/wdsc/`

How to Send Your Comments

Your feedback is important in helping us to provide the highest quality information possible. IBM welcomes any comments about this book or any other iSeries documentation.

- If you prefer to send comments by mail, use the following address:

  IBM Canada Ltd. Laboratory
  Information Development
  B3/KB7/8200/MKM
  8200 Warden Avenue
  Markham, Ontario, Canada L6G 1C7

- If you prefer to send comments electronically, use one of these e-mail addresses:
  - Comments on books:

    torrcf@ca.ibm.com

    IBMLink: to toribm(torrcf)
  - Comments on the iSeries Information Center:

    RCHINFOC@us.ibm.com

Be sure to include the following:

- The name of the book
- The publication number of the book
- The page number or topic to which your comment applies.

Accessing Online Information

VisualAge RPG contains a variety of online books and online help. You can access the help while you are using the product, and can view the books either while you are using the product, or independently.

Using Online Books

To view an online book, either:

- Select the name of the book from the **Help** pull-down menu of the VisualAge RPG GUI Designer or the editor window.
- Access the books from the **Start** menu. Select **Programs → IBM WebSphere Development Studio Client for iSeries**. Then select **Documentation**.

Publications in PDF Format

VisualAge RPG publications are available in Portable Document Format (PDF) from the iSeries Information Center at URL `http://www.ibm.com/eserver/iseries/infocenter` .

Note: You need the Adobe Acrobat Reader, Version 3.01 or later for Windows, to view the PDF format of our publications on the workstation. If your location does not have the reader, you can download a copy from the Adobe Systems Web site (http://www.adobe.com).

The following VisualAge RPG publications are available in PDF format:

- *Programming with VisualAge RPG*
- *VisualAge RPG Parts Reference*
- *VisualAge RPG Language Reference*

For information on the product, see *Getting Started with WebSphere Development Studio Client for iSeries*, SC09-2625-06.

Using Online Help

Online help is available for all areas of VisualAge RPG. To get help for a particular window, dialog box, or properties notebook, select the **Help** push button (when available).

Note: To view help that is in HTML format, your workstation must have a frames-capable Web browser, such as Netscape Navigator 4.04 or higher, or Microsoft® Internet Explorer 4.01 or higher. (Recommended browser is Netscape Navigator 4.6 or Internet Explorer 5.0)

Using context-sensitive help

To receive context-sensitive help at any time, press F1. The help that appears is specific to the area of the interface that has input focus. Input focus can be on menu items, windows, dialog boxes, and properties notebooks, or on specific parts of these.

For context-sensitive help on dialog boxes, click on the question mark (when available) in the top right-hand corner of the window. A question mark will appear beside the mouse arrow. Click on a word or field and help information on that specific field will be displayed.

Using hypertext

Some help windows contain words, phrases, or graphics that are highlighted. These are hypertext links that take you from one topic to another. To display help that is specific to a highlighted topic, click on it. When you follow a hypertext link, a **Synchronize** button may appear in the upper right-hand corner of the help topic. (You may need to page up to see the button.) If you click the Synchronize button, the list of topics in the left-hand frame is refreshed to show you how the current topic fits into the overall table of contents.

Using the help table of contents

When the Help window is displayed, press the **Synchronize** button (when available) to display the *Components* section in the left-hand frame. Click the plus + and minus − symbols to expand and collapse the section for the desired component. To view a topic, click on it.

Using the search facility

The help system uses an advanced, full-text search engine, which returns ″hits″ based on your search request. To search the current information set, click the Search tab, at the bottom left-hand corner of the navigation (table of contents) frame. Enter the search string and press the **GO** button.

For tips on refining your searches, see the *Searching online information* link.

Using language-sensitive help

To receive language sensitive help, press F1 in an edit window. If the cursor is on an operation code, you receive help for that operation code; otherwise, you receive help for the current specification.

What's New This Release

New part attributes and events include:

- **Canvas** part now has the **VKeyPress** event. You only need to code one action subroutine for all parts that can respond to this event.
- **ActiveX**: **ReturnVal** returns the value of a method call; **OCXPropIdx** sets or retrieves the index for an array property. The array can be string or numeric.
- **Container**: **DeleteRcd** can be set to delete one or all records.
- ***component**: **HostName** returns the workstation host name in the form of host_name IP_address.
- **Menu Item**:**Visible** attribute added. More shortcut keys have been added; short-cut key text is now displayed automatically next to the menu-item label at run time.

This publication includes information from the previous releases' Readme and other technical corrections. Previous Readme content includes descriptions for the following new VisualAge RPG part attributes:

- Subfile part attributes

  EditColumn, **EditIndex**, and **EditText** return the column number, column row, or text being edited.

  MapViewCol and **ViewColumn** can be used for changing the order in which Subfile data columns are displayed.

  DColFRVCol and **VColFRDCol** return the data column and view column, respectively.

  SortAsc and **SortDesc** can be used to sort data in a column.
- **PrintAsIs** for the Window and Image parts prints images and maintains their aspect ratio.
- **DropValue** is used to change or retain a label after a drop operation.
- **OnTop** changes the specified Notebook Page into the current page.
- **Graph** part: now has the **Popup** event.

Changes are noted by a vertical bar (|).

Chapter 1. Parts

Parts are the building blocks you use when creating a graphical user interface (GUI) application. They act as templates that you build on, enabling you to create your layout quickly and easily. Each part has attributes and events associated with it.

This chapter describes the VisualAge® RPG parts. Each description contains the following:

- A general description of the part
- A list of the attributes that apply to the part
- A list of the events that apply to the part
- Any restrictions that apply

ActiveX

*** Restriction:** This part is unsupported in Java® applications.

Use the ActiveX™ part to add ActiveX control objects to your project. Your applications can then access their attributes and monitor for events. (ActiveX controls are developed and provided by third party vendors.)

You must be familiar with the ActiveX controls you are adding. The VARPG GUI Designer cannot control the functions provided by these parts.

Note: VARPG only works with ActiveX controls that have interfaces written in C++. Check with your ActiveX control provider to make sure that VARPG will work with the ActiveX control you want to use.

Part Attributes

Activate	AddEvent	Bottom	DeActivate
HasPrpPage	Height	Left	Method
OCXProp	OCXPropIdx	OCXValue	ParentName
PartName	PartType	Refresh	ReturnVal
RmvEvent	ShowProp	Top	UserData
Visible	Width		

Applicable Events

Create	Destroy	OCXEvent

Animation Control

In Windows® applications, the animation control part plays video files with the AVI extension. This part differs from the media part in that the video is actually played independently of the program logic. One typical use of this part is to display an AVI file that shows some progress, such as a file being moved from one folder to another.

The animation control part plays video files with no sound. The AVI file cannot be in compressed format, unless it was compressed with the Running-Length Encoded (RLE) method.

In Java applications, the animation control part is used to play back an animated GIF file sequence using the **NbrOfImage** attribute.

Part Attributes

FileName	FrameRate	Handle*	Left
Mode	NbrOfImage	ParentName	PartName
PartType	Top	UserData	Visible

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy

Calendar

The calendar part represents a monthly calendar. By clicking on one of the month arrows, the user can navigate the calendar by going to the next or previous month.

You also have complete program control over the calendar such as going to a specific date, determining which date the user has selected, and adding short text comments to individual days in the calendar.

Part Attributes

Border	Bottom	ClearAll	ClearDate
ClearMonth	ClearYear	Color	ColorArea
ColorMix	Date	DateIdx	DateText
DateUnder	Day	DayIdx	DayLen
DayNumPos	DayNumRect	DayStart	Enabled
FontArea	FontBold	FontItalic	FontName
FontSize	FontStrike*	FontUnder*	FrmtString
Handle*	Height	HRule	Left
Month	MonthArrow	MonthIdx	MonthLen
OutlineRcl	ParentName	PartName	PartType
Refresh	ShowRects	ShowText	TipText
Top	UserData	Visible	VRule
WeekDay	WeekDayIdx	Width	Year
YearIdx	YearLen		

*** Note:** See the attribute description for restrictions.

Applicable Events

Click	Create	Destroy	DblClick
MouseDown	MouseEnter	MouseExit	MouseMove
MouseUp	MthChange	YearChange	

Canvas

Use the canvas part with a window or a notebook page if you want to place more than one part on your window or notebook page. You can point and click various parts onto the canvas, position them, and organize them to produce a graphical user interface.

The canvas part occupies the client area of either a window or a notebook page. If there is no canvas in your window or notebook page, then you can put only one part on the client area, unless the parts are extensions to the window, such as menu bars and message subfiles.

More often than not, you will be creating windows and notebook pages with more than one part on them. In that case, you should use the notebook page with canvas part and the window with canvas part. They save you a step because they already contain the canvas part.

At build time, you can also include a bitmap image as the canvas background by specifying the **FileName** attribute. This image can be tiled by setting the **Tile** attribute. For Java applications, you can include GIF or JPG images as the background.

Notes:

1. The canvas part, the window (without canvas) part, and the notebook page (without canvas) part are located on the parts catalog, not the parts palette. If you want to use them frequently, you can add them to the parts palette.
2. If parts located on a canvas part have the default font setting specified (*Default System Font*), they will inherit the font definition specified for the canvas part. To apply a certain font to the majority of parts on a canvas, specify that font for the canvas part rather than for each individual part.

For related information, see:

- "Window" on page 50
- "Window with Canvas" on page 51
- "Notebook Page" on page 32
- "Notebook Page with Canvas" on page 33.

Part Attributes

BackColor	BackMix	Bottom*	Enabled
FileName	FontBold	FontItalic	FontName
FontSize	FontStrike*	FontUnder*	Handle*
Height	Left	ParentName	PartName
PartType	ReadOnly	Refresh	Tile
Top*	UserData	Width	

*** Note:** See the attribute description for restrictions.

Applicable Events

Click	Create	DblClick	Destroy
MouseDown	MouseMove	MouseUp	Popup
VKeyPress			

Check Box

Use the check box part if you want the user to choose between two clearly distinguishable states; for example, on and off.

A label associated with the check box describes what its setting is when it is selected.

Typically, you use a group of check boxes to provide a list of options. The user can select one or more check boxes, or not select any. The options are not mutually exclusive; therefore, selecting one check box has no effect on others on the window. If you want the user to be able to select only one option from two or more, use radio buttons instead. See "Radio Button" on page 40 for more information.

To set the state of a check box, the user can either click on it with the mouse, press the space bar on the keyboard when the check box is in focus, or (if you have assigned one) press the mnemonic key. For information on assigning mnemonic keys, see *Programming with VisualAge RPG*, SC09-2449-05.

Part Attributes

AddLink*	AllowLink*	BackColor	BackMix
Bottom	Checked	Enabled	Focus
FontBold	FontItalic	FontName	FontSize
FontStrike*	FontUnder*	ForeColor	ForeMix
Handle*	Height	Highlight*	Label
Left	ParentName	PartName	PartType
Refresh	RemoveLink*	ShowTips	TipText
Top	UserData	Visible	Width

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	MouseEnter	MouseExit
MouseMove	Popup	Select	

Combination Box

A combination box provides the user with the option of entering specific information, or selecting from a list of commonly used choices.

It consists of an entry field with an attached list box that presents a list of values which the user can scroll through. If the user selects one of these values, it will appear in the entry field and replace any existing text. Alternatively, the user can type a value, that does not have to match any of the listed ones, directly into the entry field.

Combination boxes come in different styles. You can select the style you want from the part's properties notebook

For related information, see:

- "List Box" on page 24
- "Entry Field" on page 14

Part Attributes

AddItemEnd	AutoScroll*	BackColor	BackMix
Bottom	Case*	Count	DelimChar
DeSelect*	DragEnable*	DropEnable*	Enabled
FieldExit	FirstSel	Focus	FontBold
FontItalic	FontName	FontSize	FontStrike*
FontUnder*	ForeColor	ForeMix	GetItem
Handle*	Height	Index	InsertItem*
ItemKey	Left	ParentName	PartName
PartType	ReadOnly	Refresh	RemoveItem
Search*	SearchType*	Selected	SelectItem
Sequence*	SetItem	SetTop*	Showtips
SizeToFit*	Text	TipText	Top
UseDelim	UserData	Visible	Width

* **Note:** See the attribute description for restrictions.

Applicable Events

Change	Create	Destroy	Drop*
DropDown*	Enter	GotFocus	KeyPress
LostFocus	MouseEnter	MouseExit	MouseMove
Popup	Select	VKeyPress	

* **Note:** See the event description for restrictions.

*Component

The *component part allows programmers to access and use component- and system-wide attributes.

A *component part is the ″part representation″ of the component. One *component part is created for each component automatically; it is invisible and not on the palette.

Part Attributes

Active*	Alarm	AppData	Button
ClipBoard	CurrentDir	Dialog	DIRName*
DlgOwner	DoEvents*	DspHeight	DspWidth
FileName	HelpWindow	HostName*	LookNFeel*
MsgData	MsgFile*	MsgID	MsgText
Name	OS	Parent	PartCount
PartList	Platform	PlugCmd*	PlugDLL*
PlugID*	PlugRC*	PlugResult*	Printer*
SelPrinter*	ShData	ShDataLen	ShDataName
ShDataPos	ShowMsgID	SwitchTo*	WrkStnName*

*** Note:** See the attribute description for restrictions.

Applicable Events

There are no events associated with this part.

Component Reference

The component reference part enables one VARPG component to communicate with another. You use the component reference part to affect a part on the other component. The component being referenced must be running in the same application as the component reference part.

The component reference part also monitors a specified event in the other component. When the monitored event occurs, a **Notify** event is signaled by the component reference part.

Part Attributes

AddSrcEvt	AttrValue	Bottom	CompName
Left	NotSrcEvt	NotSrcPart	NotSrcWin
ParentName	PartName	PartType	RefAttr
RefParent	RefPart	RmvSrcEvt	UserData
Visible			

Applicable Events

Create	Destroy	Notify

Container

Use the container part to store related records. The records can be shown in an icon view, tree view, text tree view, or details view.

Part Attributes

AddRcd	Arrange	BackColor	BackMix
BlankChar	Bottom	ChildCount	ChildList
Collapsed	ColNumber	Count	DeleteRcd
DeSelect	EditItem	Enabled	ExtSelect*
FirstSel	Focus	FontBold	FontItalic
FontName	FontSize	FontStrike*	FontUnder*
ForeColor	ForeMix	GetNewID	GetRcdFld
GetRcdIcon	GetRcdText	Handle*	Height
InUse*	Label	Left	MiniIcon
ParentId	ParentList	ParentName	PartName
PartType	ReadOnly	RecordID	Refresh
RemoveRcd	Selected	SelectRcd	SetRcdFld
SetRcdIcon	SetRcdText	SetTop*	ShowTips
SortAsc	SortDesc	TipText	Top
UserData	View*	Visible	VisTitle
VisTitlSep	Width		

*** Note:** See the attribute description for restrictions.

Applicable Events

Click	Collapsed	ColSelect	Create
DblClick	Destroy	Enter	Expanded
GotFocus	KeyPress	LostFocus	MouseDown
MouseEnter	MouseExit	MouseMove	MouseUp
Popup	Select	VKeyPress	

DDE Client

*** Restriction:** This part is unsupported in Java applications.

Use the DDE client part to exchange data with other applications, such as spreadsheet applications, that support the dynamic data exchange (DDE) protocol.

The exchange is called a **DDE conversation**. The application that initiates the conversation is the **client**, and the application that responds is the **server**. To determine if an application supports DDE, refer to the documentation that came with it.

The DDE client part supports both **cold-link** and **hot-link conversations**. A cold-link conversation consists of a client program making explicit requests to the server program. A hot-link conversation consists of a server program automatically updating the client program when its data changes.

You can configure cold-link or hot-link conversations from the DDE client part's properties notebook and in your program logic.

Part Attributes

AppName	Bottom	DDEAddLink	DDEMode
DDERmvLink	Execute	Format	Item
Left	ParentName	PartName	PartType
Poke	Request	TimeOut	Top
Topic	UserData	Visible	

Applicable Events

Create	Data	Destroy	ExecuteAck
PokeAck	Terminate	TimeOut	

Entry Field

Use the entry field part if you want the user to input something that you cannot predict the value of. An entry field is an area into which the user can type or place text. Its boundaries are usually indicated. The user can scroll through the text in the entry field if more information is available than is currently visible.

You can configure the entry field part to accept character, numeric, or double-byte character set (DBCS) data.

You can also make the entry field read-only, so that it contains information that cannot be directly altered by the user.

You can point and click on an entry field part in the parts palette and then click it onto the subfile part to create a subfile entry field.

Part Attributes

AddLink*	Alignment	AllowLink*	AutoScroll*
AutoSelect	BackColor	BackMix	Bottom
CapsLock	Copy	CsrAtEnd	Cut
DataType	Delete	DragEnable*	DropEnable*
Enabled	FieldExit	Focus	FontBold
FontItalic	FontName	FontSize	FontStrike*
FontUnder*	ForeColor	ForeMix	Handle*
Height	InsertMode*	Left	Masked
ParentName	PartName	PartType	Paste
ReadOnly	Refresh	RemoveLink*	ShowTips
Text	TextEnd	TextLength	TextSelect
TextStart	TipText	Top	UserData
Visible	Width		

* **Note:** See the attribute description for restrictions.

Applicable Events

Change	Click	Create	DblClick
Destroy	Drop	GotFocus	KeyPress
Link*	LostFocus	MouseDown	MouseEnter
MouseExit	MouseMove	MouseUp	Popup
VKeyPress			

*** Note:** See the event description for restrictions.

Graph

The graph part allows you to create and design a graph in your project. At runtime, you can send data to the graph and change graph attributes and the graph type. The graph part supports Pie, Line, Bar, and Line and Bar graph types.

The Bar and Line graph types support the **ToolTip** text control. When enabled, moving the mouse over a data point displays the tooltip text control. To use this support in your program logic, set the value for the point into the **TipText** attribute and set the **ShowTips** attribute on for the window that contains the graph part.

Part Attributes

AutoInc	BarLabel	Bottom	Color
ColorArea	ColorMix	DataGroup	DataPoint
DataValue	Enabled	FillStyle*	FontArea
FontBold	FontItalic	FontName	FontSize
FontStrike*	FontUnder*	GnEqGrpCol	GnEqPntCol
GraphType	GroupLabel	GrphHiLite	Handle*
Height	HitItem*	HlitPoints*	LabelPlace
Left	LegendType	ParentName	PartName
PartType	Refresh	StartNew	TipText
Title	TitlePlace	Top	UnderPoint*
UseData	UserData	Visible	Width
XAxisLabel	YAxisLabel	YInc	

* **Note:** See the attribute description for restrictions.

Applicable Events

Click	Create	DblClick	Destroy
MouseDown	MouseEnter	MouseExit	MouseMove
MouseUp	Popup		

Graphic Push Button

Use graphic push buttons to provide convenient access to frequently used actions.

A graphic push button provides the same function as a push button. It indicates an action that will be initiated when the user selects it, but instead of displaying a label to describe its function, it displays an image. The **FileName** attribute specifies the name of the image to use.

Valid **Windows** image formats include:

- Windows and OS/2® Bitmaps (BMP, VGA, BGA, RLE, DIB, RL4, RL8)
- Icon (ICO)
- Microsoft/Aldus Tagged Image File Format (TIF, TIFF)
- CompuServe Graphics Interchange Format (GIF)
- ZSoft PC Paintbrush Image File Format (PCX)
- Truevision Targa/Vista Bitmap (TGA, VST, AFI)
- Amiga Interleaved Bitmap Format (IFF, ILBM)
- X Windows Bitmap (XBM)
- IBM Printer Page Segment (PSE, PSEG, PSEG38PP, PSEG3820)
- Joint Photographic Experts Group format (JPG, JPEG)

Note: This product's support for the JPG/JPEG format is based in part on the work of the Independent JPEG Group.

Valid **Java** image formats include:

- CompuServe Graphics Interchange Format (GIF)
- Joint Photographic Experts Group format (JPG, JPEG)

For related information, see "Push Button" on page 39.

Part Attributes

Bottom	Enabled	FileName	Focus
Handle*	Height	HelpEnable	HighLight
Left	ParentName	PartName	PartType
Refresh	ShowTips	TipText	Top
UserData	Validate	Visible	Width

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	GotFocus	LostFocus
MouseEnter	MouseExit	MouseMove	Popup
Press			

Group Box

Use a group box to visually distinguish a group of parts in a window.

A group box is a rectangular box that is drawn around a group of parts to indicate that they are related. It is generally advisable to label a group box. If a label is not needed, you can use an outline box.

Part Attributes

Bottom	Enabled	FontBold	FontItalic
FontName	FontSize	FontStrike*	FontUnder*
ForeColor	ForeMix	Handle*	Height
Label	Left	ParentName	PartName
PartType	Refresh	Top	UserData
Visible	Width		

Note: See the attribute description for restrictions.

Applicable Events

Create	Destroy

Horizontal Scroll Bar

Use the horizontal scroll bar part to allow users to scroll through a pane of information from left-to-right, or right-to-left. The information can be a list of files, records in a database, columns in a document, and so on. You can use the **Range** attribute to represent the total number of objects to be scrolled through and the **PageSize** attribute to determine the number of objects that can be displayed on a page.

Part Attributes

Bottom	Enabled	Focus	Handle*
Height	Left	NextLine	NextPage
PageSize	ParentName	PartName	PartType
Position	PrevLine	PrevPage	Range
Top	UserData	Visible	Width

* **Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	Scroll

Image

Use the image part to display a picture. The **FileName** attribute specifies the name of the image to use.

Valid **Windows** image formats include:

- Windows and OS/2 Bitmaps (BMP, VGA, BGA, RLE, DIB, RL4, RL8)
- Icon (ICO)
- Microsoft/Aldus Tagged Image File Format (TIF, TIFF)
- CompuServe Graphics Interchange Format (GIF)
- ZSoft PC Paintbrush Image File Format (PCX)
- Truevision Targa/Vista Bitmap (TGA, VST, AFI)
- Amiga Interleaved Bitmap Format (IFF, ILBM)
- X Windows Bitmap (XBM)
- IBM Printer Page Segment (PSE, PSEG, PSEG38PP, PSEG3820)
- Joint Photographic Experts Group format (JPG, JPEG)

Note: This product's support for the JPG/JPEG format is based in part on the work of the Independent JPEG Group.

Valid **Java** image formats include:

- CompuServe Graphics Interchange Format (GIF)
- Joint Photographic Experts Group format (JPG, JPEG)

These files reside on the programmable workstation (PWS), not on the host. You should store system-specific bitmap and icon files in the appropriate runtime directory (RT_JAVA, or RT_WIN32) so that the packaging utility includes them when you package your application.

Note: The image part can only be dropped on a notebook page with canvas or window with canvas.

Part Attributes

AddLink*	AllowLink*	BackColor	BackMix
Border	Bottom	Enabled	FileName
Handle*	Height	Left	Magnify
Panel	ParentName	PartName	PartType
Print	PrintAsIs	Refresh	RemoveLink*
ShowTips	TipText	Top	UserData
Visible	Width		

*** Note:** See the attribute description for restrictions.

Applicable Events

Click	Create	DblClick	Destroy
Link*	MouseEnter	MouseExit	MouseMove

*** Note:** See the event description for restrictions.

Java Bean

*** Restriction:** This part is unsupported in Windows applications.

Use the Java Bean part to add JavaBeans® to your project. You can use JavaBeans by calling Java methods, directly. For more information on calling Java methods, see *Programming with VisualAge RPG*, SC09-2449-05.

To develop applications that use the Java Bean part, you must have Sun Microsystem's Java 2 Software Development Kit (J2SDK), Standard Edition, version 1.2 or higher, installed on your workstation. If you do not have the J2SDK, you can download it from Sun Microsystems at the following URL:

```
http://java.sun.com/products/
```

After installing the J2SDK, set the PATH environment variable to point to the location of both the Java compiler and the **jvm.dll**, which is part of the J2SDK and JRE (Java Runtime Environment). For example, if your home directory for the J2SDK is *x:\jdk1.2*, add the following path statements:

```
x:\jdk1.2\bin
```

```
x:\jdk1.2\jre\bin\classic
```

Part Attributes

AddEvent	Bottom	Enabled	Height
Left	ParentName	PartName	PartType
RmvEvent	ShowProp	Top	UserData
Visible	Width		

Applicable Events

Create	Destroy	BeanEvent

List Box

Use the list box part to provide the user with a list of items from which one or more items can be selected. A list box consists of read-only items. An item in a list box is a string of characters.

Horizontal and vertical scroll bars allow the user to view sections of the list that are not currently displayed. You can configure the list box so that the user can select either just one item or multiple items. You can use the **Search**, **SearchType**, and **Case** attributes to easily search for a particular item in the list.

Part Attributes

AddItemEnd	AddLink*	AllowLink*	BackColor
BackMix	Bottom	Case*	Count
DelimChar	DeSelect	DragEnable*	DropEnable*
Enabled	ExtSelect*	FirstSel	Focus
FontBold	FontItalic	FontName	FontSize
FontStrike*	FontUnder*	ForeColor	ForeMix
GetItem	Handle*	Height	Index
InsertItem*	ItemKey	Left	MultSelect
NbrOfSel	ParentName	PartName	PartType
Refresh	RemoveItem	RemoveLink*	Search*
SearchType*	Selected	SelectItem	SelectList
Sequence*	SetItem	SetTop	ShowTips
SizeToFit	TipText	Top	UseDelim
UserData	Visible	Width	

* **Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	Drop*	Enter*
GotFocus	KeyPress	LostFocus	MouseEnter
MouseExit	MouseMove	Popup	Select
VKeyPress			

* **Note:** See the event description for restrictions.

Media

Use the media part to play or record audio information or to play video files.

The media part gives your programs the ability to process wave (.WAV), MIDI (.MID), and QuickTime Movie (.MOV) files. If you want to use audio files, the computer must be equipped with a sound card capable of processing these files. To record a sound file, you will need a microphone or some other supported input device for the sound card. MIDI files cannot be recorded with the media part.

Java applications require the Java Media Framework (JMF) API. The media part only supports the playback of audio and video files in the Java environment.

The video file formats that can be processed are: MPEG (*.mpg) files, QUICKTIME Movie (*.mov) files, *.dat files, Microsoft® Video for Windows *.avi files are supported for Windows. To play these video files, the computer must have the appropriate drivers.

Part Attributes

AddLink*	AllowLink*	AudioMode	Bass*
Bottom	FileName	Handle*	InPlace
Left	Length	Panel	ParentName
PartName	PartType	Position	RemoveLink*
Top	Treble*	UserData	Visible
Volume			

*** Note:** See the attribute description for restrictions.

Applicable Events

Complete	Create	Destroy	Link*

*** Note:** See the event description for restrictions.

Media Panel

Note: This part is not supported in Java applications.

Use the media panel part to provide convenient access to frequently used actions.

You can also use it to give the user control over other parts without your having to write any program logic. For example, you can use it to create push buttons or slider controls that control the volume or mode of a media part.

In the properties notebook for the media panel part, you can determine:

- Which buttons, from a defined set of buttons, the media panel will contain
- Whether or not the position and volume slider controls will be visible

Note: The media panel part can only be dropped on a notebook page with canvas or window with canvas.

Part Attributes

AddLink	AllowLink	BackColor	BackMix
Bottom	Enabled	Handle	Height
Left	PanelItem	PanelMode	ParentName
PartName	PartType	Position	RemoveLink
Top	UserData	Visible	Volume
Width			

Applicable Events

Change	Create	Destroy	Link
MouseEnter	MouseExit	MouseMove	Popup
Press			

Menu Bar

Use the menu bar part to give users access to pull-down menus. You can add submenu parts and menu item parts to the menu bar.

A menu bar appears near the top of the window frame, just below the title bar. When the user selects a menu item from it, a pull-down menu appears, showing the items on that menu. Selecting a menu item immediately initiates the action it describes.

Note: You can manipulate this part's properties, events, and so on, only from its pop-up menu in the project tree view.

For related information, see:

- "Menu Item" on page 28
- "Submenu" on page 47
- "Pop-up Menu" on page 37

Part Attributes

PartType	PartName	ParentName	UserData

Applicable Events

Create	Destroy

Menu Item

Use menu items to construct pull-down or pop-up menus.

A menu item describes an action that is initiated when the user selects that item.

To construct a menu:
1. Drop a submenu part onto a menu bar or pop-up menu.
2. Drop menu items onto the submenu.

Note: You can manipulate this part's properties, events, and so on, only from its pop-up menu in the project tree view.

For related information, see:

- "Menu Bar" on page 27
- "Pop-up Menu" on page 37
- "Submenu" on page 47

Part Attributes

Checked	Enabled	FileName	Label
ParentName	PartName	PartType	UserData
Visible			

Applicable Events

Create	Destroy	MenuSelect

Message Subfile

Use the message subfile part to display predefined messages or to display text that you supply in your program logic: for example, error or status information.

This part is always positioned at the bottom of the window frame and runs the width of the window. You cannot resize its width; you can, however, resize its height so that it shows more messages. At run time, users can use scroll bars to view all of the messages.

Part Attributes

AddMsgID	AddMsgText	Count	DragEnable*
DropEnable*	Enabled	FirstSel	FontBold
FontItalic	FontName	FontSize	FontStrike*
FontUnder*	ForeColor	ForeMix	GetItem
Handle*	Height	Index	MsgSubText
NbrOfSel	ParentName	PartName	PartType
RemoveMsg	Selected	ShowTips	TipText
UserData	Visible		

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	Drop	Enter
MouseEnter	MouseExit	MouseMove	Popup
Select			

Multiline Edit

Use the multiline edit part if you want the user to be able to type in several lines of text.

The multiline edit part has defined boundaries. Sometimes not all of its text is visible. The user can scroll up, down, left, or right to view text that is currently not visible.

Part Attributes

AddLineEnd	AddOffset	BackColor	BackMix
Bottom	CanUndo	CharOffset	Copy
CsrLine	CsrPos	Cut	Delete
DragEnable*	DropEnable*	Enabled	Focus
FontBold	FontItalic	FontName	FontSize
FontStrike*	FontUnder*	ForeColor	ForeMix
Handle*	Height	InsertLine	InsertText
Left	LineNumber	LineText	NbrOfLines
ParentName	PartName	PartType	Paste
ReadOnly	Refresh	ShowTips	Text
TextEnd	TextLength	TextSelect	TextStart
TextString	TipText	Top	TopLine
Undo	UserData	Visible	Width
WordWrap			

*** Note:** See the attribute decription for restrictions.

Applicable Events

Change	Click	Create	DblClick
Destroy	Drop	GotFocus	KeyPress
LostFocus	MouseDown	MouseEnter	MouseExit
MouseMove	MouseUp	Popup	VKeyPress

Notebook

Use the notebook part to present data that can be logically grouped by topic: for example, customer information divided into categories such as Name, Shipping Address, Orders, and Credits.

A notebook part is a graphical representation of a bound notebook. (In Windows applications, this is known as a tab control.) You can add pages to the notebook, and you can group the pages into sections separated by tabbed dividers. If the notebook page has a canvas, you can add more than one part to it. If it does not have a canvas, you can add only one part to it.

The user can turn the pages of the notebook to move from one page to the next, or go straight to a section by clicking on its divider tab.

You can add notebook pages by:

- Using the properties notebook for the notebook part
- Pointing-and-clicking (or dragging-and-dropping) a properties tab or notebook page with canvas onto the notebook part

For related information, see:

- "Notebook Page" on page 32
- "Notebook Page with Canvas" on page 33

Part Attributes

BackColor	BackMix	Bottom	Count
Enabled	Focus	FontBold	FontItalic
FontName	FontSize	FontStrike*	FontUnder*
ForeColor	ForeMix	Handle*	Height
Left	PageNumber	ParentName	PartName
PartType	Refresh	ShowTabs*	Top
UserData	Visible	Width	

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy

Notebook Page

Use the notebook page part to add pages to a notebook.

You can add only one part to a notebook page; that part will be automatically sized to fit the entire page. If you want to add more than one part on a page, you must point-and-click a canvas part onto the notebook page. Alternatively, you can use the notebook page with canvas part to save a step.

Note: You can manipulate this part's properties, events, and so on, only from its pop-up menu in the project tree view.

The user can press the left and right arrow keys to move from one page to the next.

For related information, see

- "Notebook" on page 31
- "Notebook Page with Canvas" on page 33

Part Attributes

Enabled	OnTop	ParentName	PartName
PartType	Refresh	TabImage	TabLabel
UserData	Visible		

Applicable Events

Create	Destroy	PageSelect	SelPending*

* **Note:** See the event description for restrictions.

Notebook Page with Canvas

Use the notebook page with canvas to add pages to a notebook part.

The canvas part occupies the client area of a notebook page part. By adding parts to the canvas part, you can create a graphical user interface.

If you want to add only one part to the page, you can use the notebook page part instead of the notebook page with canvas part. Because the notebook page part does not have a canvas on it, the part you add will be sized automatically.

For related information, see:

- "Notebook" on page 31
- "Notebook Page" on page 32

Part Attributes

Enabled	OnTop	ParentName	PartName
PartType	Refresh	TabImage	TabLabel
UserData	Visible		

Applicable Events

Create	Destroy	PageSelect	SelPending

ODBC/JDBC Interface

The ODBC/JDBC Interface part provides the ability to process database files that support the Windows ODBC API or Sun Microsystem's JDBC API. Examples of these database file types include Foxpro, Access, and Paradox.

To develop applications that can use the ODBC/JDBC Interface part, you must be familiar with SQL and have either the Windows ODBC SDK or Sun Microsystem's Java 2 Software Development Kit (J2SDK), Standard Edition, installed on your workstation.

If you do not have the ODBC SDK, you can download it from Microsoft at the following URL:

`http://www.microsoft.com/odbc/download.htm`

The JDBC support is part of the Java™ 2 Software Development Kit (J2SDK) Version 1.2 for Windows. If you do not have the J2SDK, you can download it from Sun Microsystems at the following URL:

`http://java.sun.com/products/`

Applications that access and manipulate data in a JDBC database require the appropriate JDBC 2.0 compliance driver. You can find JDBC driver and other information at the following URL:

`http://java.sun.com/products/jdbc/`

Note: JDBC is not supported in applets.

An ODBC or JDBC database consists of one or more tables. Data is stored in a table as a series of rows. Each row, or record, contains a number of columns with data. Your program can submit SQL statements along with ODBC/JDBC Interface part attributes to manipulate rows, or to move data between program fields and table columns.

Before you can process an existing database, your VARPG program must first connect to the database and indicate which table to reference. To manipulate the rows in a table, your program must create a record set that identifies the records to be returned and maintained by the ODBC/JDBC Interface part. To access the data in a row, you must bind each column used in the table row

with a program field in your program. In Java applications, pointers are not supported. A column is bound to a part; only the static text and entry field parts can be used for binding.

If you are creating a Java application that uses the ODBC/JDBC interface part, end users running your application must install the *varpgjdb.jar* file on their workstation and add its location to their classpath statement. The packaging utility does inlcude this JAR file. The JAR file is located in the *WDSC\java* subdirectory.

Part Attributes

AllowChg*	BindPart	Bottom	BufferDec*
BufferLen*	BufferPtr*	BufferType*	CharData
Column	ColumnDec	ColumnLen	ColumnName
Columns	ColumnType	Connect	Connected
ConnectStr	CurrentRow	DeleteRow	ExecuteSQL
Fetch	FetchNext	FetchPrior	GetTables
Handle*	Height	InsertRow	IsData
Left	ParentName	PartName	PartType
Refresh	Rows*	SQLError	SQLMsgBox
SQLQuery	Top	UnBind	UpdateRow
UserData	Visible	Width	

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy

Outline Box

Use an outline box around a group of parts to indicate that they are related.

An outline box is a rectangular, unlabeled box. If you need a label on the box, use the group box part instead.

For related information, see "Group Box" on page 19.

Part Attributes

Bottom	Handle*	Height	Left
ParentName	PartName	PartType	Refresh
Top	UserData	Visible	Width

Note: See the attribute description for restrictions.

Applicable Events

Create	Destroy

Pop-up Menu

Use the pop-up menu part to display a number of choices that pertain to a particular part on your interface. You can add menu item parts and submenu parts to the pop-up menu part.

The menu is called a "pop-up" because it appears when the user presses the appropriate key or mouse button.

Note: You can manipulate this part's properties, events, and so on, only from its pop-up menu in the project tree view.

For related information, see:

- "Menu Bar" on page 27
- "Menu Item" on page 28
- "Submenu" on page 47

Part Attributes

Handle*	InvName	InvPName	ParentName
PartName	PartType	UserData	Visible*
X	Y		

*** Note:** See the attribute description for restrictions.

Applicable Events

There are no events for this part.

Progress Bar

Use the progress bar part to indicate graphically the progress of a process, such as copying files, loading a database, and so on.

For example, to show the progress of copying 100 files, you could set the **PBRange** attribute to 100 and the **PBStepSize** attribute to 10. Your code could then monitor the copyfile process and move the progress bar indicator forward in steps for every ten files copied.

In Java applications, if the progress bar's width is smaller than its height, the progress bar will have a vertical orientation.

Part Attributes

Bottom	Handle*	Height	Left
ParentName	PartName	PartType	PBRange
PBSetPos	PBStep	PBStepSize	Top
UserData	Visible	Width	

* **Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy

Push Button

Use push buttons to provide convenient access to frequently used actions.

Each push button part controls a specific action. When the user clicks on a push button, its action is initiated immediately. The text label on the push button describes its action.

Compare with "Graphic Push Button" on page 17.

Part Attributes

BackColor	BackMix	Border*	Bottom
Enabled	Focus	FontBold	FontItalic
FontName	FontSize	FontStrike*	FontUnder*
ForeColor	ForeMix	Handle*	Height
HelpEnable	HighLight	Label	Left
ParentName	PartName	PartType	Refresh
ShowTips	TipText	Top	UserData
Validate	Visible	Width	

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	GotFocus	LostFocus
MouseEnter	MouseExit	MouseMove	Popup
Press			

Radio Button

Use radio buttons if you want the user to select only one of a group of related but mutually exclusive choices. When the user makes a selection, the previously selected choice in the group is deselected.

A radio button appears as a raised circular button that is labeled with text beside it. When selected, the circular button displays a dot.

Do not use radio buttons if you want the user to be able to select more than one choice at a time. In that case, see "Check Box" on page 7.

Part Attributes

BackColor	BackMix	Bottom	Checked
Enabled	Focus	FontBold	FontItalic
FontName	FontSize	FontStrike*	FontUnder*
ForeColor	ForeMix	Handle*	Height
HighLight*	Label	Left	ParentName
PartName	PartType	Refresh	SelectIdx
ShowTips	TipText	Top	UserData
Visible	Width		

* **Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	Enter	MouseEnter
MouseExit	MouseMove	Popup	Select

Slider

Use the slider part if you want the user to be able to display, set, or modify a value by moving a slider arm along a slider shaft.

Sliders are typically used for values that have familiar increments, such as seconds or degrees, or to show the percentage of a task that has been completed.

By default, a slider is placed horizontally in the center of a box with the slider shaft on the left side. A scale can be displayed to show the units of measure for the shaft.

Use the properties notebook for the slider part to:

- Set the range of values that a slider can return
- Position the slider vertically or horizontally in a window
- Provide a scale to indicate the units of measure represented by the slider

Part Attributes

AddLink*	AllowLink*	BackColor	BackMix
Bottom	Enabled	Focus	FontBold
FontItalic	FontName	FontSize	FontStrike*
FontUnder*	ForeColor	ForeMix	Handle*
Height	Left	Maximum	Minimum
ParentName	PartName	PartType	Refresh
RemoveLink*	ShowTips	TickLabel	TickNumber
TipText	Top	UserData	Value
Visible	Width		

*** Note:** See the attribute description for restrictions.

Applicable Events

Change	Create	Destroy	GotFocus
Link*	LostFocus	MouseEnter	MouseExit
MouseMove	Popup		

*** Note:** See the event description for restrictions.

Spin Button

Use the spin button part to display, in sequence, a group of related but mutually exclusive choices that have a logical consecutive order; for example, months of the year. The choices are displayed as though they were arranged in a ring. The user can move (or "spin") through the choices by pressing the up arrow to go to the next higher value, or the down arrow to go to the next lower one. Alternatively, one of the choices can be typed directly into the entry field for the spin button.

Part Attributes

AddItemEnd	Alignment*	BackColor	BackMix
Bottom	Enabled	Focus	FontBold
FontItalic	FontName	FontSize	FontStrike*
FontUnder*	ForeColor	ForeMix	Handle*
Height	Left	Maximum	Minimum
ParentName	PartName	PartType	ReadOnly
Refresh	RemoveItem	ShowTips	Text
TipText	Top	UserData	Value
Visible	Width		

* **Note:** See the attribute description for restrictions.

Applicable Events

Change	Create	Destroy	GotFocus
Link*	LostFocus	MouseEnter	MouseExit
MouseMove	Popup	SpinDown	SpinEnd
SpinUp			

* **Note:** See the event description for restrictions.

Static Text

Use the static text part as a label for other parts, such as a prompt for an entry field part. Static text parts do not accept end user input. In Java applications, static text can be displayed only on a single line.

Part Attributes

Alignment	BackColor	BackMix	Bottom
DataType	DragEnable*	DropEnable*	DropValue*
Enabled	FontBold	FontItalic	FontName
FontSize	FontStrike*	FontUnder*	ForeColor
ForeMix	Handle*	Height	Label
Left	ParentName	PartName	PartType
Refresh	ShowTips	TipText	Top
UserData	Visible	Width	

*** Note:** See the attribute description for restrictions.

Applicable Events

Click	Create	DblClick	Destroy
Drop	Link*	MouseDown	MouseEnter
MouseExit	MouseMove	MouseUp	Popup

*** Note:** See the event description for restrictions.

Status Bar

Use the status bar part to provide additional information about a process or action for your window. You can create up to five panes for the status bar. The status bar part provides more flexibility than the **StatusBar** attribute for the window part.

By default, a status bar is created at the bottom of the window. However, you can use the properties notebook to reposition it to the top. You can also set the border style, number of panes, and text alignment.

Part Attributes

Handle*	ParentName	PartName	PartType
SBIndex	SBLabel	SBPanes	UserData
Visible			

*** Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy

Subfile

Use the subfile part to display a list of records, each consisting of one or more fields.

The subfile part has similar function to an iSeries™ subfile. The user can scroll horizontally or vertically through the list using the subfile's scroll bars.

To create a subfile entry field, point-and-click on a field from the Define Reference Fields window or the parts palette and click it onto the subfile part. You can also add fields using the properties notebook.

Note: The subfile part can only be point-and-clicked onto a notebook page with canvas or window with canvas.

Part Attributes

AddItemEnd	AllowEdit	AutoSelect	BackColor
BackMix	Bottom	ButtonIdx	Buttons
ButtonTip	CapsLock	CellBGClr	CellBGMix
CellFGClr	CellFGMix	ColBGClr	ColBGMix
ColFGClr	ColFGMix	ColNumber	ColWidth
Count	DColFRVCol	DeSelect	EditColumn
EditIndex	EditText	EnableBtn	Enabled
ExtSelect*	FirstSel	Focus	FontArea
FontBold	FontItalic	FontName	FontSize
FontStrike*	FontUnder*	ForeColor	ForeMix
Handle*	HdgBGClr	HdgBGMix	HdgFGClr
HdgFGMix	HdgIdx	HdgText	Height
Hidden	HRule	Index	ItemCount
Left	MapViewCol	MultSelect	NbrOfSel
OpenEdit	PageSize	ParentName	PartName
PartType	RemoveItem	RowBGClr	RowBGMix
RowFGClr	RowFGMix	Scale	Selected
SelectItem	SelectList	SetTop	SflNxtChg
ShowTips	SizeToFit	SortAsc	SortDesc
StartAt	TipText	Top	TopRecord
UserData	VColFRDCol	ViewColumn	Visible
VRule	Width		

*** Note:** See the attribute description for restrictions.

Applicable Events

Change	ColSelect	Create	Destroy
Enter	FirstRec	GotFocus	KeyPress
LastRec	LostFocus	MouseEnter	MouseExit
MouseMove	NextRec	PageDown	PageEnd
PageTop	PageUp	Popup	PrevRec
Select	VKeyPress		

Submenu

Use a submenu to:
- Start a new cascaded menu from a menu item on an existing menu.
- Start a pull-down menu from a menu item on the menu bar.

After creating a submenu, you can add menu items to it by pointing-and-clicking (or dragging-and-dropping) the menu item part onto the submenu part in the tree view only.

Note: You can manipulate this part's properties, events, and so on, only from its pop-up menu in the project tree view.

For related information, see "Menu Item" on page 28.

Part Attributes

ParentName	PartName	PartType	UserData

Applicable Events

Create	Destroy

Timer

Use the timer part if your program must perform certain operations at preset time intervals. For example, you can use it to close a window, or perhaps end an application, after a certain period of inactivity.

A timer part counts units of time and tracks the preset time interval between two events, triggering the second event once the interval has passed.

When you create a timer part in the GUI builder, the part is represented as an icon on the design window. However, in the properties notebook for a timer part, you can specify that you do not want the icon displayed while the program is executing.

Note: Do not use the timer part when precise timing is required. Due to other programs running on your system, the **Tick** event may not necessarily occur at the exact interval you specify.

Part Attributes

AddLink*	AllowLink*	Bottom	Interval
Left	Multiplier	ParentName	PartName
PartType	RemoveLink*	TimerMode	TimerTicks
Top	UserData	Visible	

* **Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	Link*	Tick

* **Note:** See the event description for restrictions.

Vertical Scroll Bar

Use the vertical scroll bar part to allow users to scroll through a pane of information vertically. The information can be a list of files, records in a database, columns in a document, and so on. You can use the **Range** attribute to represent the total number of objects to be scrolled through and the **PageSize** attribute to determine the number of objects that can be displayed on a page.

Part Attributes

Bottom	Enabled	Focus	Handle*
Height	Left	NextLine	NextPage
PageSize	ParentName	PartName	PartType
Position	PrevLine	PrevPage	Range
Top	UserData	Visible	Width

* **Note:** See the attribute description for restrictions.

Applicable Events

Create	Destroy	Scroll

Window

Windows are the user's primary means of interacting with your program. Your application must contain at least one window.

You can add only one part to the client area of a window, except for parts that are extensions to the window frame, such as menu bars, pop-up menus and message subfiles. The part you add is automatically sized to fit the client area.

If you want a window to contain more than one part, you must add a canvas part to it. Or, use the window with canvas part to save a step.

Note: The window part is located in the **Frames** section of the parts catalog, not on the parts palette.

For related information, see:

- "Canvas" on page 5
- "Window with Canvas" on page 51

Part Attributes

Bottom	Center	Enabled	FileName*
Focus*	FontBold*	FontItalic*	FontName*
FontSize*	FontStrike*	FontUnder*	Handle*
Height	IconHandle*	Label	Left
MouseIcon*	MouseShape*	ParentName	PartName
PartType	PBRange	PBSetPos	PBStep
PBStepSize	Print	PrintAsIs	ProgresBar
Refresh	SBLabel	SBPosition	SBStyle
ShowTips	StatusBar	Top	UserData
Visible	Width	WindowMode*	

* **Note:** See the attribute description for restrictions.

Applicable Events

Activate	Close	Create	DeActivate
Destroy	LClickTray	Moved	RClickTray
ReSize	ShutDown		

Window with Canvas

Windows are the end user's primary means of interacting with your program. The canvas, on a window with canvas part, allows you to add many parts to the window.

You can point and click various parts onto the canvas portion, position them, and organize them to produce a graphical user interface. You can also add parts that are extensions of the window's frame, such as menu bars, pop-up menus and message subfiles.

If you need to put only one part on the client area of the window, you do not need the window with canvas part: you should use the window part instead (found in the **Frames** section of the parts catalog). Without a canvas, the part you add will be automatically sized to fit the client area.

For related information, see:

- "Canvas" on page 5
- "Window" on page 50

Part Attributes

Bottom	Center	Enabled	FileName*
Focus*	FontBold*	FontItalic*	FontName*
FontSize*	FontStrike*	FontUnder*	Handle*
Height	IconHandle*	Label	Left
MouseIcon*	MouseShape*	ParentName	PartName
PartType	PBRange	PBSetPos	PBStep
PBStepSize	Print	PrintAsIs	ProgresBar
Refresh	SBLabel	SBPosition	SBStyle
ShowTips	StatusBar	Top	UserData
Visible	Width	WindowMode*	

* **Note:** See the attribute description for restrictions.

Applicable Events

Activate	Close	Create	DeActivate
Destroy	LClickTray	Moved	RClickTray
ReSize	ShutDown		

Chapter 2. Part Attributes

When creating your graphical user interface (GUI) using VisualAge RPG, you can set or change most of a part's attributes by opening the **properties notebook** for that part.

You can also set most part attributes at run time in your VisualAge RPG application program by using built-in functions or operation codes. You do this by coding the **%setatr** built-in function on the free-form EVAL operation code or by using the SETATR operation code. To retrieve the values of part attributes, you code the **%getatr** built-in function on the free-form EVAL operation code or use the GETATR operation code.

This section describes the part attributes. Each description contains the following:

- A general description of the attribute.
- The data type of the attribute.
- A list of the parts the attribute applies to.
- A table indicating whether you can set or get the attribute by using the built-in functions or operation codes. This table also indicates whether you can set an attribute using a part's properties notebook.
- The allowed values for the attribute.
- Examples of the free-form and operation-code syntax. For syntax details, see the *VisualAge RPG Language Reference* manual.

  Note: Some examples used in this chapter use uppercase text only. However, the logic can be entered in either upper- or lowercase.
- Any applicable restrictions.

Activate

Set this attribute to 1 to activate an ActiveX object.

Type Numeric

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'AX1': 'Activate') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    1             'ACTIVATE'
 *
```

Active

*** Restriction:** This attribute is unsupported in Java applications.

Determines whether another instance of the component is running.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

The active value can be one of the following:

0 The current instance is the only instance of the component

1 Another instance of the component is running

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Act=%getatr('*Component':'*Component':'Active')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'Active'      Act
 *
```

AddEvent

Adds the event name to the event filter list. You can use the special value *ALL to have the ActiveX or Java Bean part route all available events stored in the list to the VARPG program. By default, the ActiveX or Java Bean part does not route any events.

To build a new, partial event list after issuing an AddEvent(*ALL), empty the list first with the **RmvEvent** attribute.

Type String

Applicable Parts

ActiveX Java Bean

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'AddEvent')='EVT2'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    'EVT2'        'ADDEVENT'
 *
```

AddItemEnd

Adds a new item to the end of the list.

For a subfile part, this attribute applies to a subfile entry field that has values defined as its validation type. Use this attribute to add values to the subfile entry field's validation list. First, set the **ColNumber** attribute to identify which column to work with.

Type String

Applicable Parts

Combination Box List Box Spin Button Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The item to add to the list, or the subfile entry field's validation value

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'LB1':'AddItemEnd')=newitem
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    'NEWITEM'     'ADDITEMEND'
 *
```

AddLineEnd

Adds a new line after the last line in a multiline edit.

Type String

Applies to

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The new line of text

Free form example

Get the text value from an entry field, and add it to the end of a multiline edit.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'MLE1':'AddLineEnd')=newtext
 *
```

Fixed form example

Add a line to the end of a multiline edit.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    'NEWTEXT'     'ADDLINEEND'
 *
```

AddLink

*** Restriction:** This attribute is unsupported in Java applications.

Links a target part to this source part. If the **AllowLink** attribute is also set, the target part will generate a **Link** event when this part changes.

For a list of part types that may be linked, refer to Table 1.

Table 1. Allowed Source and Target Parts

When	Source Part	Linked Target Part Response
Change in mode or position	Media	Media Panel responds to Media part mode change.
Selection	Check Box	Timer part is enabled or disabled.
Value changed	Entry Field	Media: file name is set. Entry field: text value is updated. Image: file name is set. Slider: numeric value is updated. Spin Button: numeric value is updated. Static Text: text value is updated.
File name changes	Image	Static Text: text value is updated.
Selection	List Box	Media: file name is set. Entry field: text value is updated. Image: file name is set. Static Text: text value is updated.
Control change	Media Panel	Media part responds to Media Panel part mode change.
Arm change	Slider	Entry field: text value is updated. Slider: numeric value is updated. Static Text: text value is updated.
Tick	Timer	Static Text: text value is updated. Timer count is set to zero.

Type String

Applicable Parts

CheckBox	Entry Field	Image	List Box
Media	Media Panel	Slider	Timer

Operation Codes

	Operational at Run time
SETATR*	yes
GETATR	no

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Link** tab and filling in the **Target Part name** and **Target Parent Part name** fields.

Allowed Value

The name of the part to link to.

The part name must be in the format:

```
'window|part'
```

where *window* is the window on which the part is placed, and *part* is the actual part name.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'CB1':'AddLink')='WIN1|TIMER1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    'WIN1|TIMER1' 'ADDLINK'
 *
```

AddMsgID

Adds a message to the message subfile part.

Messages are added after any existing messages in the message subfile part.

Type Numeric

Applicable Parts

Message Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The *message identifier* (msgid), which is a number up to four digits long.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MSG1': 'AddMsgID') = 1234
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MSG1'        SETATR    1234          'ADDMSGID'
 *
```

AddMsgText

Adds a message to the message subfile part. The message is added in the form of a string.

Messages are added after any existing messages in the message subfile part.

Type String

Applicable Parts

Message Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The message text string

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'MSG1':'AddMsgText')='Help!'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MSG1'        SETATR    'Help!'       'ADDMSGTEXT'
 *
```

AddOffset

Inserts the specified text into the multiline edit at the character offset specified by the **CharOffset** attribute.

Type String

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The string to insert in the multiline edit

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'CharOffset') = 10
C                   EVAL      %setatr('win01': 'MLE1': 'AddOffset') = 'XXX'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    10            'CHAROFFSET'
C     'MLE1'        SETATR    'XXX'         'ADDOFFSET'
 *
```

AddRcd

Adds a new record to a container.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The value passed for this attribute must be a string that consists of the following parameters, each separated by a space:

Record Structure

ID A numeric value used to identify the record. Declare this value to contain the largest number of records you expect the container to hold.

Note: You can use the **GetNewID** attribute to have VisualAge RPG generate a unique ID for you.

Text The text that will appear next to the record icon when the container is in icon view or tree view.

FileName
The filename for the icon.

ParentID
The identifier of the parent record for this record. If the record has no parent, this parameter is set to 0.

field_data
This value is optional. If it is specified, data will be placed into corresponding fields in the record. Data must be provided for each field in the record. The number of fields for each record of the container part in the GUI designer is already defined; therefore, you should specify only that number.

Note: If the data for a field is not yet available, use the underscore (_) character as a place holder for the field. If a field contains imbedded blanks, they should be

replaced with the underscore character before you add the record. The imbedded blanks will appear in the field data of the added record.

Field data can be updated after the record has been added by using the **SetRcdFld** attribute.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
DName++++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++
D NewRecord
D   RecordID              1      6
D   FileName              8     64
D   Parent                1      6    INZ(0)
D   Text                  8     20    INZ('Next record')
 *
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   Eval      RecordID=%getatr('win01': 'CN1': 'GetNewID')
C                   Eval      FileName = 'D:\VRPG\NEW.ICO'
C                   Eval      %setatr('win01': 'CN1': 'AddRcd') = NewRecord
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    NewRecord     'ADDRCD'
 *
```

AddSrcEvt

Adds a new source event to the current list of events being monitored by the component reference part. This attribute allows one component reference part to monitor multiple events. To remove source events from the list, use the **RmvSrcEvt** attribute.

To identify the source event being added, specify the component, window, part, and event names altogether. The format of the string is:

```
'COMPONENTNAME | WINDOWNAME | PARTNAME | EVENTNAME'
```

As the compiler does no validation checking, you must identify each source event accurately.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
DName++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++
D Event           S             100
*
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Event='compb|win02|ef02|change'
C                   EVAL      %setatr('win01': 'CRP1':
C                             'AddSrcEvt')=Event
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
DName++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++
D Event           S             100
*
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Event='compb|win02|ef02|change'
C     'CRP1'        SETATR    Event         'ADDSRCEVT'
 *
```

Alarm

Causes the speaker to beep.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The value for this attribute must be set to 1.

Free form example

Set the speaker to beep:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'Alarm')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    1             'ALARM'
 *
```

Alignment

This attribute determines how text and labels are aligned in parts.

Type Numeric

Applicable Parts

Entry Field* Spin Button* Static Text

*** Restrictions:** Once this attribute is set, it cannot be changed at run time.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

Alignment can be one of the following:

1 Text is left-justified

2 Text is right-justified

3 Text is centered

Free form example

If the text in a part is left-justified, make it right-justified:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      align = %getatr('win01': 'ST1': 'Alignment')
 *
C     align         IFEQ      1
C                   EVAL      %setatr('win01': 'ST1': 'Alignment') = 2
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'ST1': 'Alignment') = 1
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         GETATR    'ALIGNMENT'   ALIGN             1 0
C     ALIGN         IFEQ      1
C     'ST1'         SETATR    2             'ALIGNMENT'
 *
C                   ELSE
C     'ST1'         SETATR    1             'ALIGNMENT'
 *
 *
C                   ENDIF
 *
```

AllowChg

*** Restriction:** This attribute is unsupported in Java applications.

Identifies the column data as an unsupported VARPG data type. No updates are performed on this column data.

When you perform a **FetchNext**, **FetchPrior**, **InsertRow**, or **UpdateRow**, data is moved to or from the program field buffer to the corresponding table column. If the column has an unsupported VARPG data type, set the **AllowChg** attribute to 0. The default value is 1. (See "BufferType" on page 98 for a list of supported data types.)

Set the Column attribute to the desired column before setting the **AllowChg** value.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

AllowChg can be one of the following:

0 Unsupported VARPG data type.

1 Supported VARPG data type. This is the default.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'ODBC1': 'AllowChg') = 0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    0             'AllowChg'
 *
```

AllowEdit

Determines whether or not subfile entry fields can be edited. Edit status can be specified at the subfile, record, column, or cell level.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

AllowEdit can be any of the following:

1 Prevents editing of the entire subfile.

2 Prevents editing of the record specified by the **Index** attribute.

3 Prevents editing of the column specified by the **ColNumber** attribute.

4 Prevents editing of the cell specified by the **ColNumber** and **Index** attributes.

11 Allows editing of the entire subfile.

12 Allows editing of the record specified by the **Index** attribute.

13 Allows editing of the column specified by the **ColNumber** attribute.

14 Allows editing of the cell specified by the **ColNumber** and **Index** attributes.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFL1': 'Index') = 5
C                   EVAL      %setatr('win01': 'SFL1': 'AllowEdit') = 12
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    5             'INDEX'
C     'SFL1'        SETATR    12            'ALLOWEDIT'
 *
```

AllowLink

*** Restriction:** This attribute is unsupported in Java applications.

This attribute determines if the target part specified by the **AddLink** or **DDEAddLink** attribute will generate a **Link** event when the source part changes.

Type Numeric

Applicable Parts

CheckBox	Entry Field	Image	List Box
Media	Media Panel	Slider	Timer

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Link** tab and selecting **Enable notify target**.

Allowed Value

AllowLink can be one of the following:

0 **Link** event is not signaled

1 **Link** event is signaled

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      allow = %getatr('win01': 'EF1': 'AllowLink')
 *
C     allow         IFEQ      1
C                   EVAL      %setatr('win01': 'EF1': 'AllowLink') = 0
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'EF1': 'AllowLink') = 1
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF01'        GETATR    'ALLOWLINK'   ALLOW             1 0
 *
C     ALLOW         IFEQ      1
C     'EF01'        SETATR    0             'ALLOWLINK'
 *
C                   ELSE
C     'EF01'        SETATR    1             'ALLOWLINK'
 *
C                   ENDIF
 *
```

AppData

Contains global data that any other component can reference. This attribute provides an alternative to passing parameters that must otherwise be declared.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'AppData')
C                              = NData
C                   EVAL      TmpFld=%getatr('*Component':'*Component':
C                             'AppData')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    NData       'APPDATA'
C     '*COMPONENT'  GETATR    'APPDATA'     TmpFld
 *
```

AppName

*** Restriction:** This attribute is unsupported in Java applications.

This is the name of the server application that the DDE client part will connect to in a DDE conversation. The server name is determined by the server application. Refer to the application documentation for the **AppName** to use for that application.

Setting the **AppName** does not begin the DDE conversation. For information on beginning a DDE conversation, refer to "DDEMode" on page 176.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The name of the server application

If the server program is a VisualAge RPG application, this attribute will be the name of the application including its extension. For example, if the VisualAge RPG application was called *server*, you would set the **AppName** attribute as follows:

```
EVAL    %setatr('WIN':'DDE1':'AppName') = 'SERVER.EXE'
```

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'AppName') = 'APP01'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'APP01'       'APPNAME'
 *
```

Arrange

Arranges items in a container in icon view, such that the icons are lined up in rows.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The value for this attribute must be 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   Eval      %setatr('window': 'CN1': 'Arrange') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'ARRANGE'
 *
```

AttrValue

The value of the attribute being referenced by the component reference part. First, set the component reference part to identify the component, window, and part name for the attribute.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'COMP1': 'AttrValue')='Test'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'COMP1'       SETATR    'Test'        'ATTRVALUE'
 *
```

AudioMode

Sets the operating mode for the part.

Type Numeric

Applicable Parts

Media

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

AudioMode can be one of the following:

1 Pause

2 Play

3 Record

4 Stop

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'AUDIO01': 'AudioMode') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AUDIO01'     SETATR    1             'AUDIOMODE'
 *
```

AutoInc

This attribute causes the current **DataPoint** value to be incremented whenever the **DataValue** attribute is set.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

AutoInc returns one of the following:

0 The **DataPoint** is not incremented automatically

1 The **DataPoint** is incremented automatically

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      auto = %getatr('win01': 'GP1': 'AutoInc')
 *
C     auto          IFEQ      1
C                   EVAL      %setatr('win01': 'GP1': 'AutoInc') = 0
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'GP1': 'AutoInc') = 1
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GP01'        GETATR    'AUTOINC'     Auto              1 0
 *
C     AUTO          IFEQ      1
C     'GP01'        SETATR    0             'AUTOINC'
 *
C                   ELSE
C     'GP01'        SETATR    1             'AUTOINC'
 *
C                   ENDIF
 *
```

AutoScroll

*** Restriction:** This attribute is unsupported in Java applications.

Allows a user to scroll data in an entry field when more text can be typed or when more information is available than can be displayed in the entry field.

Type Numeric

Applicable Parts

Combination Box Entry Field

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

AutoScroll can be one of the following:

0 Autoscroll is not allowed

1 Autoscroll is allowed

Free form example

Toggle the **AutoScroll** attribute of an entry field.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'EF1': 'AutoScroll')
 *
C     rc            IFEQ      1
C                   EVAL      %setatr('win01': 'EF1': 'AutoScroll') = 0
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'EF1': 'AutoScroll') = 1
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'AUTOSCROLL'  rc                1 0
 *
C     rc            IFEQ      1
C     'EF1'         SETATR    0             'AUTOSCROLL'
 *
C                   ELSE
C     'EF1'         SETATR    1             'AUTOSCROLL'
C                   ENDIF
 *
```

AutoSelect

This attribute indicates whether the text in a field is highlighted when the entry field gets focus or a subfile field is opened for editing. For a subfile part, first set the **ColNumber** attribute to identify which column to work with.

Type Numeric

Applicable Parts

Entry Field Subfile

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

AutoSelect can be one of the following values:

0 The entry field value is not highlighted

1 The entry field value is highlighted

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      highlight=%getatr('win01':'EF1':'AutoSelect')
 *
C     highlight     IFEQ      0
C                   EVAL      %setatr('win01':'EF1':'AutoSelect')=1
C                   ELSE
C                   EVAL      %setatr('win01':'EF1':'AutoSelect')=0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'AUTOSELECT'  highlight         1 0
 *
C     'EF1'         SETATR    1             'AUTOSELECT'
 *
```

BackColor

A numeric value that determines the background color of a part.

If the background color attribute had been set using the **BackMix** attribute, the color index that most closely represents the background color mix value is returned.

Type Numeric

Applicable Parts

Canvas	Check Box	Combination Box	Container
Entry Field	Image	List Box	Media Panel
Multiline Edit	Notebook	Push Button	Radio Button
Slider	Spin Button	Static Text	Subfile

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The background color must be one of the following:

-1 Default(not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Note: The default color (-1) cannot be used for SETATR.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      color = %getatr('win01': 'EF1': 'BackColor')
 *
C     color         IFEQ      *Red
C                   EVAL      %setatr('win01': 'EF1': 'BackColor') = *Blue
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    *BLUE         'BACKCOLOR'
 *
```

BackMix

Queries or sets the background color mix.

Type String

Applicable Parts

Canvas	Check Box	Combination Box	Container
Entry Field	Image	List Box	Media Panel
Multiline Edit	Notebook	Push Button	Radio Button
Slider	Spin Button	Static Text	Subfile

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The background color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      color=%getatr('win01':'EF1':'BackMix')
 *
C     color         IFNE      '10:23:200'
C                   EVAL      %setatr('win01':'EF1':'BackMix')='10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    '10:23:200'   'BACKMIX'
 *
```

BarLabel

Indicates the label of the current data item (as specified by **DataPoint** and **DataGroup**).

Type String

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GPH1': 'BarLabel') = 'BAR01'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GPH1'        SETATR    'BAR01'       'BARLABEL'
 *
```

Bass

Note: This attribute applies to Windows only.

Sets the bass level for the media part.

Type Numeric

Applicable Parts

Media

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Bass can be a value from 0 to 100.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'AUD1': 'Bass') = 37
 *
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AUD1'        SETATR    37            'BASS'
 *
```

BindPart

Restriction: This attribute applies only to Java applications.

Identifies the part to be bound or that is bound to the data column. In the Java environment, binding is supported for Entry Field and Static Text parts, only.

Type String

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The format of the parent/part name to be bound is:

```
'FRAMEWINDOW|PARTNAME'
```

where FrameWindow is the name of the window that owns the PartName.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'WIN1': 'BindPart')
C                                 = 'WIN1|ENT2'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    'WIN1|ENT2'   'BINDPART'
 *
```

BlankChar

Specifies the character that is to represent the blank character in a string.

The container part uses a blank as the delimiter for elements in a record. If the element you are adding contains blanks, such as the icon text, set the **BlankChar** attribute to some non-blank character.

The default blank character is the underscore (_).

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'CTR1':'BlankChar')='&'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CTR1'        SETATR    '&'           'BLANKCHAR'
 *
```

Border

Determines if the part has a border.

Type Numeric

Applicable Parts

Calendar	Image	Push Button*

*** Restriction:** Can set and get this attribute for the Push Button part only in Java applications.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

Border can be one of the following:

0 Part has no border

1 Part has a border

Free form example

Toggle the **Border** attribute of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      brdval = %getatr('win01': 'IMG1': 'Border')
C     brdval        IFEQ      0
C                   EVAL      %setatr('win01': 'IMG1': 'Border') = 1
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'IMG1': 'Border') = 0
C                   ENDIF
 *
```

Fixed form example

Toggle the current border setting for a push button:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'IMG1'        GETATR    'BORDER'      brdval            1 0
C     brdval        IFEQ      1
C     'IMG1'        SETATR    0             'BORDER'
 *
C                   ELSE
C     'IMG1'        SETATR    1             'BORDER'
C                   ENDIF
```

Bottom

Specifies the bottom screen coordinate, in pixels, of the part or the object.

Note: The origin is the upper left corner, and begins with zero.

Type Numeric

Applicable Parts

ActiveX	Calendar	Canvas*	Check Box
Combination Box	Component Reference	Container	DDE Client
Entry Field	Graph	Graphic Push Button	Group Box
Horizontal Scroll Bar	Image	Java Bean	List Box
Media	Media Panel	Multiline Edit	Notebook
ODBC/JDBC Interface	Outline Box	Progress Bar	Push Button
Radio Button	Slider	Spin Button	Static Text
Subfile	Timer	Vertical Scroll Bar	Window

* **Restriction:** Can only get this attribute at run time for the Canvas part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The number of pixels

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      bot = %getatr('win01': 'Image1': 'Bottom')
C     bot           SUB       100           newbot
C                   EVAL      %setatr('win01': 'Image1': 'Bottom') =  newbot
 *
```

Fixed form example

Move an image 200 pixels in its window.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'IMAGE1'      GETATR    'BOTTOM'      OldBot            4 0
C     OldBot        ADD       200           NewBot            4 0
C     'IMAGE'       SETATR    NewBot        'BOTTOM'
 *
```

BufferDec

*** Restriction:** This attribute is unsupported in Java applications.

Specifies the number of decimal places for the buffer column.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'BufferDec')=4
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    4             'BufferDec'
 *
```

BufferLen

*** Restriction:** This attribute is unsupported in Java applications.

Specifies the length of the program field available for the column data. The ODBC/JDBC Interface part needs to know the length of the data to set and get into the program field when moving data from and to a column.

Before using this attribute, set the **Column** attribute to the desired column.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'BufferLen')=20
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    20            'BufferLen'
 *
```

BufferPtr

*** Restriction:** This attribute is unsupported in Java applications.

Specifies the pointer value to the program field. The ODBC/JDBC Interface part uses this value to move data from the database to the program field.

Before using this attribute, set the **Column** attribute to the desired column.

To get the address of a program field, use the %ADDR built-in as follows:

```
D FieldPtr        S               *   INZ(%Addr(Field))
```

Note: This attribute is not applicable in the Java environment.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Note: For the SETATR operation, the BufferPtr attribute can be coded **only** in **fixed-form** syntax.

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    FieldPtr      'BufferPtr'
 *
```

BufferType

* **Restriction:** This attribute is unsupported in Java applications.

Indicates the data type of the VisualAge RPG program field referenced by the **BufferPtr** attribute. The ODBC/JDBC Interface part uses the **BufferType** attribute to perform the correct data translation when moving data between the program field and table column. It is important to set this attribute correctly, as there is no checking for proper field types.

Set the **Column** attribute before using the **BufferType** attribute. If the program field is associated with a part on the interface, you can use the **DataType** attribute to get the buffer type.

If a column contains a data type that is not supported by the ODBC/JDBC Interface part, set the **AllowChg** attribute to 0 for that column. The ODBC/JDBC Interface part will not move data between any program field and the column. The data remains unchanged.

Use the following chart to set the VARPG data type for the corresponding, supported SQL data type. Specify the **BufferLen** and **BufferDec** attributes only as listed in the chart.

SQL Data Type	VARPG Data Type	Specify Program Field Length (use BufferLen)	Specify Decimal Places for Buffer Column (use BufferDec)
Character	CHAR	X	
Decimal	Zoned	X	
Integer	Zoned	X	
Small Integer	Zoned	X	
Double	8F		
Double	Zoned	X	X
Float	4F		
Float	Zoned	X	X
Real	4F		
Real	Zoned	X	X

For character, decimal, integer, or small integer data types, specify only the **BufferLen** attribute.

Note that Double, Float, and Real data types can be defined, in VARPG, as either Float(F) or Zoned. If you define these as Zoned, the VARPG run time will only use the number of decimal places specified by the **BufferDec** attribute when moving data from the column. This can result in a loss of

precision if the data source has more decimal places than is specified by the **BufferDec** attribute. If you define these fields as Float(F), do **NOT** specify the **BufferLen** or **BufferDec** attribute.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

BufferType can be one of the following:

0 Indicates numeric data.

Supported SQL data types include:
- Decimal
- Integer
- Small integer
- Float
- Real
- Double

1 Indicates character data.

Supported SQL data types include:
- Character
- Variable-length character
- Date
- Time
- Timestamp

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'BufferType')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'BUFFERTYPE'
 *
```

Button

Queries whether the OK or CANCEL button has been pressed or if the file dialog has been closed.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

Button status can be one of the following:

0 The file dialog has not been closed

1 The OK button has been pressed

2 The CANCEL button has been pressed

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      But=%getatr('*Component':'*Component':'Button')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'Button'      But
 *
```

ButtonIdx

Sets and gets the index for the navigation buttons. The index must be set before activating the **EnableBtn** attribute.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The index can be set to:

1 The pagetop button

2 The pageup button

3 The prevrec button

4 The nextrec button

5 The pagedown button

6 The pageend button

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFI1': 'ButtonIdx') = 4
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFI1'        SETATR    4             'BUTTONIDX'
 *
```

Buttons

Expands or collapses the navigation bar.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Buttons can be one of the following:

0 Do not show the buttons

1 Show the buttons

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      btnval = %getatr('win01': 'SF1': 'Buttons')
C     btnval        IFEQ      0
C                   EVAL      %setatr('win01': 'SF1': 'Buttons') = 1
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'SF1': 'Buttons') = 0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'BUTTONS'      btnval            1 0
C     btnval        IFEQ      1
C     'SF1'         SETATR    0             'BUTTONS'
 *
C                   ELSE
C     'SF1'         SETATR    1             'BUTTONS'
C                   ENDIF
```

ButtonTip

Sets the tooltip text for the corresponding navigation button set by **ButtonIdx**.

The text can be set as a string or as a message (*MSGnnnn).

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The tool tip string

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFI1': 'ButtonTip') = string
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFI1'       SETATR     STRING        'BUTTONTIP'
 *
```

CanUndo

Indicates if undoing the last action is possible. If 1 is returned, the last action can be undone. A 0 indicates it cannot.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Chng=%getatr('win01':'ML1':'CanUndo')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ML1'         GETATR    'CANUNDO'     CHNG
 *
```

CapsLock

Determines if the text being typed in an entry field or a subfile entry field is to be folded to upper case.

Note: To get or set **CapsLock** in a subfile entry field, the **ColNumber** attribute has to be set to a correct column number.

Type Numeric

Applicable Parts

Entry Field Subfile

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

CapsLock can be one of the following:

0 Text is not folded to upper case

1 Text is folded to upper case

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'EF1': 'CapsLock') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    1             'CAPSLOCK'
 *
```

Case

*** Restriction:** This attribute is unsupported in Java applications.

Determines whether the search comparison is case sensitive, or not.

Type Numeric

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Case can be one of the following:

0 The comparison is not case sensitive.

1 The comparison is case sensitive.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CB1': 'Case') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    1             'CASE'
 *
```

CellBGClr

A numeric value that determines the background color of a subfile cell. You must use the **Index** and **ColNumber** attributes to identify the cell that will have its color set.

If the background color has been set using the **CellBGMix** attribute, the background color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default(not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CBcolor = %getatr('win01': 'SFL1': 'CellBGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'ColNumber') = 1
C                   EVAL      %setatr('win01': 'SFL1': 'Index') = 4
C                   EVAL      %setatr('win01': 'SFL1': 'CellBGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'CELLBGCLR'   'CBcolor'         2 0
 *
C     'SFL1'        SETATR    1             'COLNUMBER'
C     'SFL1'        SETATR    4             'INDEX'
C     'SFL1'        SETATR    5             'CELLBGCLR'
 *
```

CellBGMix

Queries or sets the background color mix of a subfile cell. You must use the **Index** and **ColNumber** attributes to identify the cell that will have its color mix queried or set.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The background color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CBclrMx = %getatr('win01': 'SFL1': 'CellBGMix')
 *
C     CBclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01': 'SFL1': 'CellBGMix') = '10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'CELLBGMIX'
 *
```

CellFGClr

A numeric value that determines the foreground color of a subfile cell. You must use the **Index** and **ColNumber** attributes to identify the cell that will have its color set.

If the foreground color has been set using the **CellFGMix** attribute, the foreground color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default(not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CFcolor = %getatr('win01': 'SFL1': 'CellFGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'ColNumber') = 1
C                   EVAL      %setatr('win01': 'SFL1': 'Index') = 4
C                   EVAL      %setatr('win01': 'SFL1': 'CellFGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'CELLFGCLR'   'CFcolor'         2 0
 *
C     'SFL1'        SETATR    1             'COLNUMBER'
C     'SFL1'        SETATR    4             'INDEX'
C     'SFL1'        SETATR    5             'CELLFGCLR'
 *
```

CellFGMix

Queries or sets the foreground color mix of a subfile cell. You must use the **Index** and **ColNumber** attributes to identify the cell that will have its color mix queried or set.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The foreground color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CFclrMx=%getatr('win01':'SFL1':'CellFGMix')
 *
C     CFclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01':'SFL1':'CellFGMix')='10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'CELLFGMIX'
 *
```

Center

When set to 1, the window will be centered when first displayed. The user can later move this window as desired.

If after centering the window the top left corner's x or y coordinate is negative, the x y coordinates will be set to (0,0) to ensure that the window's title bar is always visible.

Type Numeric

Applicable Parts

Window	Window with Canvas

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	no

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The attribute can only be set to 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('FW1': 'FW1': 'Center') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'FW1'         SETATR    1             'CENTER'
 *
```

CharData

Returns the character text for a specific column in a returned record set. Set the **Column** attribute before using this attribute.

Type String

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ColData=%getatr('win01':'ODBC1':'CharData')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'CHARDATA'    COLDATA
 *
```

CharOffset

Determines the character offset value in a multiline edit where text will be inserted when the **AddOffset** attribute is set.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The character offset

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'CharOffset') = 10
 *
C                   EVAL      value = %getatr('win01': 'MLE1': 'CharOffset')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    10            'CHAROFFSET'
C     'MLE1'        SETATR    'text'        'ADDOFFSET'
 *
```

Checked

Determines if the part is checked.

Type Numeric

Applicable Parts

Check Box Menu Item Radio Button

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. For the Check Box and Radio Button parts, do this by selecting the **Selected** style. For Menu Item parts, do this by selecting the **Checked** style.

Allowed Value

The **Checked** state of the part can be one of the following:

0 The part is not checked

1 The part is checked

Free form example

Set the checked state of a check box the same as a radio button:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ischecked = %getatr('win01': 'RB1': 'Checked')
 *
C                   EVAL      %setatr('win01': 'CB1': 'Checked') = ischecked
 *
```

Fixed form example

Toggle the checked state of a check box:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         GETATR    'CHECKED'     ischecked
 *
C     ischecked     IFEQ      1
C     'CB1'         SETATR    0             'CHECKED'
 *
```

```
C                   ELSE
C     'CB1'         SETATR    1             'CHECKED'
C                   ENDIF
 *
```

ChildCount

Returns the number of direct children of a parent specified by **RecordId**.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      children = %getatr('win01': 'CTR1': 'ChildCount')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CTR1'        GETATR    'CHILDCOUNT'  children
 *
```

ChildList

Returns a blank delimited list of the direct children for a record contained in the container part. Specify which record's child list is desired using the **RecordId** attribute.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      list = %getatr('win01': 'CTR1': 'ChildList')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CTR1'        GETATR    'CHILDLIST'   list
 *
```

ClearAll

When set, **ClearAll** removes all user text from the calendar.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The only valid value of **ClearAll** is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'ClearAll') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1             'CLEARALL'
 *
```

ClearDate

When set, **ClearDate** removes any user text set for the date indicated by **DayIdx/MonthIdx/YearIdx**.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The only valid value of **ClearDate** is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'ClearDate') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1             'CLEARDATE'
 *
```

ClearMonth

When set, **ClearMonth** removes any user text set for the month indicated by **MonthIdx**/**YearIdx**.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The only valid value of **ClearMonth** is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'ClearMonth') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1             'CLEARMONTH'
 *
```

ClearYear

When set, **ClearYear** removes any user text set for the year indicated by **YearIdx**.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The only valid value of **ClearYear** is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'ClearYear') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1             'CLEARYEAR'
 *
```

ClipBoard

Copies text from or to the clipboard.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

Get the clipboard contents:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'ClipBoard')=text
 *
C                   EVAL      text=%getatr('*Component':'*Component':'ClipBoard')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    text          'CLIPBOARD'
 *
C     '*COMPONENT'  GETATR    'CLIPBOARD'   text
 *
```

ColBGClr

A numeric value that determines the background color of a subfile column. You must use the **ColNumber** attribute to identify the column that will have its color set.

If the background color has been set using the **ColBGMix** attribute, the background color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default (not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CBcolor = %getatr('win01': 'SFL1': 'ColBGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'ColNumber') = 1
C                   EVAL      %setatr('win01': 'SFL1': 'ColBGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'COLBGCLR'   'CBcolor'         2 0
 *
C     'SFL1'        SETATR    1             'COLNUMBER'
C     'SFL1'        SETATR    5             'COLBGCLR'
 *
```

ColBGMix

Queries or sets the background color mix of a subfile column. You must use the **ColNumber** attribute to identify the column that will have its color mix queried or set.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The background color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CBclrMx = %getatr('win01': 'SFL1': 'ColBGMix')
 *
C     CBclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01': 'SFL1': 'ColBGMix') = '10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'COLBGMIX'
 *
```

ColFGClr

A numeric value that determines the foreground color of a subfile column. You must use the **ColNumber** attribute to identify the column that will have its color set.

If the foreground color has been set using the **ColFGMix** attribute, the foreground color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default (not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CFcolor = %getatr('win01': 'SFL1': 'ColFGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'ColNumber') = 1
C                   EVAL      %setatr('win01': 'SFL1': 'ColFGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'COLFGCLR'   'CFcolor'         2 0
 *
C     'SFL1'        SETATR    1             'COLNUMBER'
C     'SFL1'        SETATR    5             'COLFGCLR'
 *
```

ColFGMix

Queries or sets the foreground color mix of a subfile column. You must use the **ColNumber** attribute to identify the column that will have its color mix queried or set.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The foreground color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CFclrMx = %getatr('win01': 'SFL1': 'ColFGMix')
 *
C     CFclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01': 'SFL1': 'ColFGMix') = '10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'COLFGMIX'
 *
```

Collapsed

Applies to a container in the tree view. This attribute can be used to expand or collapse items in a record specified by the **RecordID** attribute.

Specifying a record ID of zero reverses the state of all records in the tree view.

When you expand a specific record in a Java application, the record's children and parent records become visible. When you collapse a record, only the children are hidden; parent records remain visible.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Collapsed can be one of the following:

1 Collapse the record

0 Expand the record

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
 * Collapse the record
C                   Eval      %setatr('window': 'CN1': 'Collapsed') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'COLLAPSED'
 *
```

ColNumber

Establishes the field column number to be used when setting other subfile attributes. This attribute is used with the **Index** attribute to indicate which field in a subfile record is being referenced. It is also used with the **ViewColumn** and **MapViewCol** attributes to change the order in which subfile data columns are displayed.

Type Numeric

Applicable Parts

Container Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The field column number

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SF1': 'Index') = 5
C                   EVAL      %setatr('win01': 'SF1': 'ColNumber') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    5             'INDEX'
C     'SF1'         SETATR    2             'COLNUMBER'
 *
```

Color

Sets and gets the color of the areas indicated by the value of the **ColorArea** attribute.

Type Numeric

Applicable Parts

Calendar Graph

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The color must be one of the following:

-1 Default(not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Note: The default color (-1) cannot be used for SETATR.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      color = %getatr('win01': 'GP1': 'Color')
 *
C     color         IFEQ      *Red
C                   EVAL      %setatr('win01': 'GP1': 'Color') = *Blue
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GP1'         SETATR    *BLUE         'COLOR'
 *
```

ColorArea

Indicates the area to which the **Color** and **ColorMix** attributes should apply when get or set.

Type Numeric

Applicable Parts

Calendar Graph

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

* **Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

For the **graph** part, **ColorArea** can be one of the following:

1 Title color

2 Background color

3 Axis label color

4 Axis line color

5 Data point label color

6 Highlight color

7 Graph background color

8 Border color

9 Bar/slice/line color

For the **calendar** part, **ColorArea** can be one of the following:

1 Background

2 Grid

3 Calendar interior

4 Month arrows

5 Month arrow outlines

6 Month/Year outline

7 Month/Year background

8 Month/Year foreground

9 Day names background

10 Day names foreground

11 Selected date outline

12 Today outline

13 Default color for user outlines

14 Default color for user text

15 Border

16 Date numbers

17 Date number outlines

18 User outline (run time only)

19 User text (run time only)

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      colarea = %getatr('win01': 'GP1': 'ColorArea')
 *
C     colarea       IFEQ      3
C                   EVAL      %setatr('win01': 'GP1': 'ColorArea') = 4
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GP1'         SETATR    4             'COLORAREA'
 *
```

ColorMix

Sets and gets the color mix as indicated by the value of the **ColorArea** attribute.

Type String

Applicable Parts

Calendar Graph

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      color=%getatr('win01':'GP1':'ColorMix')
 *
C     color         IFNE      '10:23:200'
C                   EVAL      %setatr('win01':'GP1':'ColorMix')='0:0:255'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GP1'         SETATR    '10:23:200'   'COLORMIX'
 *
```

Column

Establishes which column in a table is being referenced by other attributes, such as, **BufferLen**, **BufferPtr**, and **BufferType**.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

From 1 to the number of columns in the table.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'Column')=12
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    12            'COLUMN'
 *
```

ColumnDec

Returns the number of decimal positions defined for a column. Set the **Column** attribute before using this attribute.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Dec=%getatr('win01':'ODBC1':'ColumnDec')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'COLUMNDEC'   DEC
 *
```

ColumnLen

Returns the defined length of a specific column in a table. Set the **Column** attribute before using this attribute.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Len=%getatr('win01':'ODBC1':'ColumnLen')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'COLUMNLEN'   LEN
 *
```

ColumnName

Returns the name of a specific column in a record set. Set the **Column** attribute before using this attribute. A successful SQL query using the **SQLQuery** and **ExecuteSQL** attributes makes the column names available.

Type String

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CName=%getatr('win01':'ODBC1':'ColumnName')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'COLUMNNAME'  CNAME
 *
```

Columns

Returns the number of columns in the result set. A successful SQL query using the **SQLQuery** and **ExecuteSQL** attributes makes the number of columns available.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Cols=%getatr('win01':'ODBC1':'Columns')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'COLUMNS'     COLS
 *
```

ColumnType

Returns the data type of a column. Set the **Column** and **ExecuteSQL** attributes before using this attribute.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

The value returned can be one of the following:

1 Variable-length character

2 Long variable-length character

3 Character

4 Numeric

5 Decimal

6 Integer

7 Small integer

8 Timestamp

0 For the following **JDBC data types**: Date, Time, Big integer, Bit, Double, Float, Real, Tiny integer, Binary, Long variable-length binary, Variable-length binary, Other, Null

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CType=%getatr('win01':'ODBC1':'ColumnType')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'COLUMNTYPE'  CTYPE
 *
```

ColWidth

Determines the width of a subfile column in pixels.

The **ColNumber** attribute must be set to indicate which column is affected.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFI1': 'ColWidth') = 50
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFI1'        SETATR    50            'COLWIDTH'
 *
```

CompName

The name of the component to be referenced.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   Eval      %setatr('win1': 'comp1': 'CompName')='Palette'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'COMP1'       SETATR    'Palette'     'COMPNAME'
 *
```

Connect

Set this attribute to 1 to connect the ODBC/JDBC Interface part with a database. Before using this attribute, set the **ConnectStr** attribute to determine which data source to connect to.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'Connect')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'CONNECT'
 *
```

Connected

Determines if a connection attempt to a table was successful or not. If this attribute returns a 1, the connection was successful. Any other value indicates that the connection failed.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Status=%getatr('win01':'ODBC1':'Connected')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'CONNECTED'   STATUS
 *
```

ConnectStr

Provides the connection string to pass to the ODBC manager to make a connection with a data source. Set this attribute before using the Connect attribute. If you set the connection string to *BLANKS, the *Select Data Source* dialog appears. You can select the table to connect to from this dialog.

Refer to the ODBC API documentation for the syntax of a connection string.

Type String

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'ConnectStr')=Str
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    STR           'CONNECTSTR'
 *
```

Copy

Copies the selected text to the clipboard. The selected text will not be deleted.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'Copy') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'          SETATR   1             'copy'
 *
```

Count

Returns the number of items in the list portion of a part.

The **Count** attribute is updated when an item is added to or removed from the list.

Type Numeric

Applicable Parts

Combination Box	Container	List Box	Message Subfile
Notebook	Subfile		

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'LB1': 'Count')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'COUNT'       count             2 0
 *
```

CsrAtEnd

Set this attribute to 1 to position the cursor after the last character in an entry field, when the entry field gets focus.

Type Numeric

Applicable Parts

Entry Field

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'EN1': 'CsrAtEnd') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EN1'         SETATR    1             'CSRATEND'
 *
```

CsrLine

Indicates the line number on which the cursor is positioned.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The line number

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'CsrLine') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    2             'CSRLINE'
 *
```

CsrPos

Indicates the cursor position relative to the first character of the multiline edit field.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The cursor position

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'MLE1':'CsrPos')=4
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    4             'CSRPOS'
 *
```

CurrentDir

Returns the current working directory.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Dir = %getatr('*component': '*component':
C                             'CurrentDir)
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'CurrentDir'  Dir
 *
```

CurrentRow

Returns the current row number of the result set.

Note: In Java applications, this attribute requires the appropriate JDBC 2.0 compliance driver.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      CRow=%getatr('win01':'ODBC1':'CurrentRow')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'CURRENTROW'  CROW
 *
```

Cut

Deletes the selected text and copies it to the clipboard.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'Cut') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    1             'cut'
 *
```

DataGroup

Used in conjunction with **DataPoint** to indicate a specific data item, or on its own when referring to a group of points as a whole.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value must be greater than 0.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GR1': 'DataGroup') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GR1'         SETATR    1             'DATAGROUP'
 *
```

DataPoint

Used in conjunction with **DataGroup** to indicate a specific data item.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value must be greater than 0.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GR1': 'DataPoint') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GR1'         SETATR    1             'DATAPOINT'
 *
```

DataType

Determines the type of data allowed in this field.

Type Numeric

Applicable Parts

Entry Field Static Text

Operation Codes

	Operational at Run Time
SETATR*	no
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

DataType can be one of the following:

0 Zoned Numeric - This part supports only numeric characters. It also supports a hyphen (-) as a leading character to indicate a negative number.

1 Character - This part supports only alphanumeric characters.

2 DBCS Only - This part supports only DBCS characters.

3 DBCS Either - This part supports either SBCS or DBCS characters, but not both, depending on the first character entered.

4 DBCS Mixed - This part supports both SBCS and DBCS characters.

Free form example

Get the character type for an entry field:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      type = %getatr('win01': 'EF1': 'DataType')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'DATATYPE'    type              1 0
 *
```

DataValue

Specifies the value assigned to a particular data item

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GR1': 'DataValue') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GR1'         SETATR    1             'DATAVALUE'
 *
```

Date

Indicates the currently selected date. The date is always a string in the form YYYYMMDD and is synchronized with the **Year**/**Month**/**Day** attributes.

Type String

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value
The date in the form YYYYMMDD

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'Date') = '19980221'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    '19980221'    'DATE'
 *
```

DateIdx

Allows the user to set data on a date not displayed on the current calendar page. The date is always a string in the form YYYYMMDD and matches **YearIdx/MonthIdx/DayIdx**.

Type String

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The index value

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DateIdx') =
C                             = '19980221'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    '19980221'    'DATEIDX'
 *
```

DateText

The text that is displayed inside a date square for the date indicated by **DayIdx/MonthIdx/YearIdx**.

Type String

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DateText') = 'text'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    'text'        'DATETEXT'
 *
```

DateUnder

Returns the date under the mouse cursor as a string YYYYMMDD. If the mouse is not over a date, **DateUnder** returns an empty string.

Type String

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      yyyymmdd = %getatr('win01': 'CAL1':
C                             'DateUnder')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'DATEUNDER'   yyyymmdd
 *
```

Day

Indicates the currently selected day. When set, the selected date in the calendar part is updated to reflect the changes.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The day of the month

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'Day') = 21
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    21            'DAY'
 *
```

DayIdx

Allows the user to set data on a date not displayed on the current calendar page.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The index value

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DayIdx') = 21
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    21            'DAYIDX'
 *
```

DayLen

Indicates the format for displaying the day name.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

DayLen can be one of the following:

1 Full name

2 Abbreviated three-character name

3 Abbreviated, two-character prefix (for example, SU, MO, TU...)

4 Abbreviated, single-character prefix (for example, S, M, T...)

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DayLen') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'DAYLEN'      daylen
 *
```

DayNumPos

Indicates the position of the numbers of the days of the month.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

DayNumPos can be one of the following:

1 Top left

2 Top right

3 Bottom left

4 Bottom right

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DayNumPos') = 3
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'DAYNUMPOS'   daynumpos
 *
```

DayNumRect

Indicates whether or not the days of the month will be outlined.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

DayNumRect can be one of the following:

0 The numbers will not be outlined

1 The numbers will be outlined

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isout = %getatr('win01': 'CAL1': 'DayNumRect')
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DayNumRect') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'DAYNUMRECT'  isout
 *
```

DayStart

Determines which day is the first day of the week.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

DayStart can be one of the following:

0 Sunday

1 Monday

2 Tuesday

3 Wednesday

4 Thursday

5 Friday

6 Saturday

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'DayStart') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    2             'DAYSTART'
 *
```

DColFRVCol

Returns the data column that corresponds to the column specified by the **ViewColumn** attribute.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'SF1':'ViewColumn') = 4
C                   EVAL      DataCol4 = %getatr('win01':'SF1':'DColFRVCol')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    4             'VIEWCOLUMN'
C     'SF1'         GETATR    'DCOLFRVCOL'  DATACOL4
 *
```

If your subfile has the field ***CustNo*** placed in column 1, but you change the view so that ***CustNo*** now appears in column 4, the GETATR operation returns **1** to indicate where the data column is actually located in the subfile.

DDEAddLink

Setting this attribute establishes a DDE hot-link conversation between the server application defined by the **AppName** and **Topic** attributes, and the part indicated by the *part* parameter.

When a hot-link conversation is established, the server application will automatically signal a **Data** event to the DDE client part when the item specified by the **Item** attribute of the DDE client part changes. The DDE client part will then update the linked part with the new data if the **AllowLink** attribute is set.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	no

* **Note:** Can be set using the properties notebook in the GUI designer. Do this by selecting the **Add** push button and entering values in the **Add DDE Link** dialog box.

Allowed Value

The *part* parameter must be coded as follows:

```
'window|part'
```

Free form example

Set up a hot-link conversation between an entry field and a cell in a spreadsheet. When the cell changes, the contents of the entry field will change.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'DDE1':'AppName')='APP01'
C                   EVAL      %setatr('win01':'DDE1':'Topic')='APP01.WG2'
C                   EVAL      %setatr('win01':'DDE1':'Item')='A1'
C                   EVAL      %setatr('win01':'DDE1':'DDEAddLink')='win1|ef1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'win1|ef1'    'DDEADDLINK'
 *
```

DDEMode

Determines the current mode of a DDE client part.

Type Numeric

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The part mode represented by one of the following values:

1 *Initiate* - Initiates a DDE conversation. Before the conversation can be initiated, the **AppName** and **Topic** attributes must have been set.

When the DDE conversation has been successfully initiated, a cold-link conversation is established between the server application and the DDE client part. A cold-link conversation means the DDE client part must set the **Request** attribute to get data from the server application, and set the **Poke** attribute to send data to the server application.

2 *Terminate* - Terminates the DDE conversation. When this attribute is set, a terminate request is sent to the server application. When the server application has processed the terminate request, it will respond by signaling a **Terminate** event to your program. To process this event, you must code a terminate action subroutine in your program.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'DDEMode') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    1             'DDEMODE'
 *
```

DDERmvLink

Removes a link from this DDE client part. The **Item** attribute is used to determine which link to remove.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The *part* parameter must coded as follows:

```
'window|part'
```

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'DDE1':'DDERmvLink')='win1|ef1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'win1|ef1'    'DDERMVLINK'
 *
```

DeActivate

Set this attribute to 1 to deactivate an ActiveX object.

Type Numeric

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'AX1': 'DeActivate') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    1             'DEACTIVATE'
 *
```

Delete

Deletes the selected text.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'Delete') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    1             'delete'
 *
```

DeleteRcd

Deletes a specific record, or all records in a container part.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

DeleteRcd can be one of the following:

0 Deletes all records.

n Deletes the record specified by the **RecordID** attribute.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CT1': 'DeleteRcd') = 0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CT1'         SETATR    0             'DELETERCD'
 *
```

DeleteRow

Determines the row number to delete in the result set. You do not need to fetch a row first to delete it. This attribute will cause any row to be deleted.

To delete a row that has just been fetched, use the **CurrentRow** attribute to get its row number.

Note: In Java applications, this attribute requires the appropriate JDBC 2.0 compliance driver.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'DeleteRow')=4
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    4             'DELETEROW'
 *
```

DelimChar

Specifies the delimiter character that separates multiple items in a data string. The **DelimChar** and **UseDelim** attributes are used together.

For example, you can use a semicolon (;) to delimit the items in a data string as follows- abcd;efgh;ijkl. Specifying the **AddItemEnd** attribute will produce a vertical item list from this string. This allows adding multiple items with a single operation.

Type String

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'CB1':'DelimChar') = ';'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    ';'           'DELIMCHAR'
 *
```

DeSelect

Deselects items in the list portion of a part.

In Windows applications, **DeSelect** has no effect when:

- The container is set for extended or multiple selection and is in the tree view.
- The subfile is set for single selection.

  In these cases, if an item is selected, it remains selected.

In Java applications, the container can be set for extended or multiple selection in the tree view.

Type Numeric

Applicable Parts

Combination Box* Container List Box Subfile

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

Specify the number of the item to deselect. To deselect all items in the list, set the value to zero.

Free form example

Deselect the fifth item in the list.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'DeSelect') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    5             'DESELECT'
 *
```

Dialog

Allows the user to choose an **Open** or **Save As** dialog.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Dialog can be one of the following:

1 Open

2 SaveAs

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('*component': '*component':
C                             'Dialog')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'DIALOG'      rc
 *
```

DIRName

*** Restriction:** This attribute is unsupported in Java applications.

Returns the name of the directory where the component resides.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

text string

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      dir = %getatr('*component': '*component':
C                             'DIRName')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'DIRNAME'     dir
 *
```

DlgOwner

Indicates which window owns a dialog. If it is not specified, the ownership defaults to the desktop.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value of should take the form:

```
WindowName   WindowName
```

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'DlgOwner')=
C                             ='win1 win1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    'win1 win1'   'DLGOWNER'
 *
```

DoEvents

*** Restriction:** This attribute is unsupported in Java applications.

Set this attribute to 1 to release control from the currently running action subroutine so that other events can trigger associated action subroutines to run. This is especially useful when an action subroutine is inside a long running loop.

Type Numeric

Applicable Parts

*component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'DoEvents')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    1             'DOEVENTS'
 *
```

DragEnable

*** Restriction:** This attribute is unsupported in Java applications.

When this attribute is enabled, the user can move the text or label of a part by placing the pointer over the part, and pressing and holding mouse button 2.

Type Numeric

Applicable Parts

Combination Box	Entry Field	List Box	Message Subfile
Multiline Edit	Static Text		

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

One of the following:

0 The part text or label cannot be dragged

1 The part text or label can be dragged

Note: This attribute can only be set to 1.

Free form example

Allow **DragEnable** for a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'EF1': 'DragEnable') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    1             'DRAGENABLE'
 *
```

DropEnable

*** Restriction:** This attribute is unsupported in Java applications.

Determines whether the part will respond to **Drop** events.

Type Numeric

Applicable Parts

Combination Box	Entry Field	List Box	Message Subfile
Multiline Edit	Static Text		

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

One of the following:

0 Part does not respond to **Drop** events

1 Part responds to **Drop** events

Note: The default setting for this attribute is 0, or off. Once it is set to 1 in the builder, it cannot be reset.

Free form example

Drop enable a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'EF1': 'DropEnable') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    1             'DROPENABLE'
 *
```

DropValue

*** Restriction:** This attribute is unsupported in Java applications.

Determines whether the text or label can be changed by a drop operation. Set the **DropEnable** attribute so the part can respond to **Drop** events.

For example, you can code a ″Send Mail″ label that users can drag-and-drop e-mail addresses onto to initiate a send mail dialog. Set the DropValue attribute to 0 so that the ″Send Mail″ label remains unchanged after the Drop operation.

Type Numeric

Applicable Parts

Static Text

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

One of the following:

0 The part text or label cannot be changed

1 The part text or label can be changed

Note: Set the **DropEnable** attribute to 1 to allow Drop events.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'ST1': 'DropValue') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         SETATR    1             'DROPVALUE'
 *
```

DspHeight

Returns the display height, in pixels.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

Get the display height:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      DH=%getatr('*Component':'*Component':'DspHeight')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'DSPHEIGHT'   DH
 *
```

DspWidth

Returns the display width, in pixels.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

Get the display width:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Dwd=%getatr('*Component':'*Component':
C                             'DspWidth')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'DSPWIDTH'    Dwd
 *
```

EditColumn

Returns the column number of the cell that is opened for editing.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      edcol = %getatr('win01': 'SF1': 'EditColumn')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'EDITCOLUMN'  EDCOL
 *
```

EditIndex

Returns the row number of the cell that is opened for editing.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      edrow = %getatr('win01': 'SF1': 'EditIndex')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'EDITINDEX'   EDROW
 *
```

EditItem

In Windows applications, setting this attribute to 1 opens an edit field on the record text for a container in the icon or tree view. In Java applications, this attribute applies only to the tree view. You must set the **ColNumber** attribute before using this attribute.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'EditItem')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'EDITITEM'
 *
```

EditText

Returns the edited text of the cell currently being edited. Returns the empty string if no cell is opened for editing.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      celtext = %getatr('win01': 'SF1': 'EditText')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'EDITTEXT'    celtext         100
 *
```

EnableBtn

Enables or disables the navigation buttons. **ButtonIdx** must be set before using the **EnableBtn** attribute.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value can be set to:

0 Disable the button

1 Enable the button

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFI1': 'EnableBtn') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFI1'        SETATR    1            'ENABLEBTN'
 *
```

Enabled

Determines whether the part will respond to user interface events.

Note: If this attribute is specified for parts on a canvas, the attribute value set for the canvas part takes precedence.

Type Numeric

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Graphic Push Button
Group Box	Horizontal Scroll Bar	Image	Java Bean
List Box	Media Panel	Menu Item	Message Subfile
Multiline Edit	Notebook	Notebook Page	Notebook Page with Canvas
Push Button	Radio Button	Slider	Spin Button
Subfile	Vertical Scroll Bar	Window	

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

One of the following:

0 The part cannot respond to events

1 The part can respond to events

Free form example

Toggle the enabled state of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'EF1': 'Enabled')
 *
C     rc            IFEQ      0
C                   EVAL      %setatr('win01': 'EF1': 'Enabled') = 1
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'EF1': 'Enabled') = 0
C                   ENDIF
 *
```

Fixed form example

Toggle the enabled state of an entry field:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'ENABLED'     rc                1 0
C     rc            IFEQ      1
C     'EF1'         SETATR    0             'ENABLED'
 *
C                   ELSE
C     'EF1'         SETATR    1             'ENABLED'
C                   ENDIF
 *
```

Execute

Setting this attribute posts an *execute* command to the server application in a DDE conversation. The structure of the command string is determined by the server application. Refer to the server application documentation for supported commands and command structure.

A DDE conversation must have been successfully started before you set this attribute.

When the server application has processed the *execute* command, it will generate an **ExecuteAck** event. Check the **%DDEAck** event attribute in your ExecuteAck event action subroutine to determine if the execute was successful.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'Execute') = 'FILEOPEN'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'FILEOPEN'    'EXECUTE'
 *
```

ExecuteSQL

Setting this attribute to 1 executes the SQL statement submitted by the last **SQLQuery** attribute.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'ExecuteSQL')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'EXECUTESQL'
 *
```

ExtSelect

Determines whether the part allows extended selection. Extended selection allows the user to select an item and then select additional items. The user can mark multiple entries by dragging the mouse cursor across them.

The **MultSelect** attribute has no effect on the **ExtSelect** attribute. The **ExtSelect** attribute overrides the settings for either **MultSelect** or single selection.

Type Numeric

Applicable Parts

Container (1.) List Box(2.) Subfile (2.)

Restrictions:

1. In Windows applications, can only set this attribute at run time for the Container part.
2. In Java applications, can only get this attribute at run time for the List Box and Subfile parts.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

One of the following:

0 Extended selection is not allowed

1 Extended selection is allowed

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'LB1': 'ExtSelect')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'EXTSELECT'   rc                1 0
 *
```

Fetch

Setting this attribute to 1 returns a specific row from the attached data source. The data returned depends on the last type of fetch request made. For a **GetTables** attribute request, **Fetch** returns a list of tables for the database. For an **ExecuteSQL** request, **Fetch** returns a row of data from the table.

Note: In Java applications, this attribute requires the appropriate JDBC 2.0 compliance driver.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'Fetch')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'FETCH'
 *
```

FetchNext

Setting this attribute to 1 fetches the next row of data from the result set.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'FetchNext')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'FETCHNEXT'
 *
```

FetchPrior

Setting this attribute to 1 fetches the previous row of data from the result set.

Note: In Java applications, this attribute requires the appropriate JDBC 2.0 compliance driver.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'FetchPrior')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'FETCHPRIOR'
 *
```

FieldExit

Causes the cursor to tab to the next tabbable entry field when the Enter key is pressed.

Type Numeric

Applicable Parts

Combination Box Entry Field

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

One of the following:

0 The cursor will not tab to the next applicable entry field

1 The cursor will tab to the next applicable entry field

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'EF1': 'FieldExit') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    1             'FIELDEXIT'
 *
```

FileName

Specifies the file to be associated with the part.

For a **media** part, this can be a valid wave (.wav) or MIDI (.mid) file, or a valid video file with the .mov or .mpg extension.

In Windows applications using an **animation control** part, the file can be a video file with the .avi extension. In Java applications that use the animation control part with the **NbrOfImage** attribute, the file can be a GIF image.

For a canvas, graphic push button, image, or menu item part, the file must be a valid image file. For a list of valid image formats, see "Image" on page 21.

For a **menu item** part, if **FileName** refers to a bitmap, the menu item is displayed as the bitmap. If **FileName** refers to an icon file, the icon is displayed next to the menu item text. In Java applications, if **FileName** refers to a GIF file, the GIF image is displayed next to the menu item text.

In Java applications, **FileName** refers to a GIF file that is displayed as an icon in the **Window** part.

Changing the **FileName** attribute for a media part does not cause that file to be processed. To process the file, the **AudioMode** attribute must be set.

Changing the **FileName** attribute for a canvas, graphic push button, or an image part will cause the picture on the part to change immediately.

Setting the **FileName** attribute for *component will cause the dialog chosen with the **Dialog** attribute to be displayed.

The filename can be specified using either the filename with complete path information, or just the filename. If only the filename is specified, the file should be put in the runtime subdirectory: the RT_JAVA subdirectory for a Java application, or the RT_WIN32 subdirectory for a Windows application. When just the filename is specified, the files are found only if the RT_JAVA, or RT_WIN32 subdirectory is the current directory.

See *Programming with VisualAge RPG* for more on filename considerations.

Type String

Applicable Parts

Animation Control	Canvas	*Component	Graphic Push Button
Image	Media	Menu Item	Window
Window with Canvas			

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

*** Restriction:** In Java applications, this attribute is settable only.

Allowed Value

The name of the workstation file to associate with this part

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'audio1':'FileName')='BARK.WAV'
 *
```

Fixed form example

Save the current filename for an image part, and set it to a new value:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'IMAGE01'     GETATR    'FILENAME'    oldfile          64
C     'IMAGE01'     SETATR    newfile       'FILENAME'
 *
```

FillStyle

*** Restriction:** This attribute is unsupported in Java applications.

Describes how an area will be filled

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

FillStyle can be one of the following:

1 Solid

2 Hollow

3 Low density filled

4-9 Increasing degrees of density filled

10 High density filled

11 Vertically striped

12 Horizontally striped

13 Forward diagonally striped

14 Very forward diagonally striped

15 Backward diagonally striped

16 Very backward diagonally striped

17 Halftone filled

18 Hatched

19 Diagonally hatched

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      style = %getatr('win01': 'GP1': 'FillStyle')
 *
C     style         IFEQ      3
C                   EVAL      %setatr('win01': 'GP1': 'FillStyle') = 12
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GP1'         SETATR    12           'FILLSTYLE'
 *
```

FirstSel

Returns the index value of the first selected item in the list portion of a part. Index values begin with 1. If no list item is selected, a value of zero is returned.

Type Numeric

Applicable Parts

Combination Box Container List Box Message Subfile Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ix = %getatr('win01': 'LB1': 'FirstSel')
 *
```

Fixed form example

Check if an item has been selected in a list box, and display a message if it has not:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'FIRSTSEL'    IX               3 0
C     IX            IFEQ      *ZERO
C     'Select error'DSPLY     STYLE         rc
C                   ENDIF
 *
```

Focus

Determines if the part currently has focus. When a part has focus, the input cursor is moved to it and user input is directed to it. For example, before the user can type into an entry field, the entry field must have focus.

When a part receives focus, it receives a **GotFocus** event.

Type Numeric

Applicable Parts

Check Box	Combination Box	Container	Entry Field
Graphic Push Button	Horizontal Scroll Bar	List Box	Multiline Edit
Notebook	Push Button	Radio Button	Slider
Spin Button	Subfile	Vertical Scroll Bar	Window*

*** Restriction:** Can only set this attribute at run time for the Window part in Java applications.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

One of the following:

0 The part does not have input focus

1 The part has input focus

Note: This attribute can only be set to 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'EF1': 'Focus') = 1
 *
C                   EVAL      rc = %getatr('win01': 'EF1': 'Focus')
 *
```

Fixed form example

Toggle focus for a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'FOCUS'       HasFocus          1 0
 *
C     'EF1'         SETATR    1             'FOCUS'
 *
```

FontArea

Indicates the area in the calendar, graph or subfile part to which the other font-related attributes should apply.

Type Numeric

Applicable Parts

Calendar Graph Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

For the calendar part, the value can be one of the following:

1 Month/Year

2 Day names

3 Default for user text

4 Day numbers

5 User text

For a **graph** part, the value can be one of the following:

1 Title font

2 Axis label font

3 Data point label font

4 Legend font

For a **subfile** part, the value can be one of the following:

1 Heading

2 Record

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Farea = %getatr('win01': 'GRA1': 'FontArea')
 *
C                   EVAL      %setatr('win01': 'GRA1': 'FontArea') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        GETATR    'FONTAREA'     Farea            1 0
 *
```

FontBold

Determines whether the label or text is displayed in **bold** font.

Type Numeric

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Group Box
List Box	Message Subfile	Multiline Edit	Notebook
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Window*	

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Note: In the case of the Calendar and Graph parts, the actual font area affected is determined by the value of the **FontArea** attribute.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Font** tab and selecting the **Change font** push button.

Allowed Value

FontBold can be one of the following:

0 Font is not displayed in bold

1 Font is displayed in bold

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      bold = %getatr('win01': 'ST1': 'FontBold')
 *
C     bold          IFEQ      0
C                   EVAL      %setatr('win01': 'ST1': 'FontBold') = 1
C                   ELSE
C                   EVAL      %setatr('win01': 'ST1': 'FontBold') = 0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         GETATR    'FONTBOLD'    bold              1 0
 *
C     'ST1'         SETATR    1             'FONTBOLD'
 *
```

FontItalic

Determines if the label or text is shown in *Italic* font.

Type Numeric

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Group Box
List Box	Message Subfile	Multiline Edit	Notebook
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Window*	

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Note: In the case of the Calendar and Graph parts, the actual font area affected is determined by the value of the **FontArea** attribute.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Font** tab and selecting the **Change font** push button.

Allowed Value

FontItalic can be one of the following:

0 Font is not displayed in italics

1 Font is displayed in italics

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      italic = %getatr('win01': 'ST1': 'FontItalic')
 *
C     italic        IFEQ      0
C                   EVAL      %setatr('win01': 'ST1': 'FontItalic') = 1
C                   ELSE
C                   EVAL      %setatr('win01': 'ST1': 'FontItalic') = 0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         GETATR    'FONTITALIC'  italic            1 0
 *
C     'ST1'         SETATR    1             'FONTITALIC'
 *
```

FontName

Specifies the font to be used for the text or label that applies to this part.

Type String

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Group Box
List Box	Message Subfile	Multiline Edit	Notebook
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Window*	

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Note: In the case of the Calendar and Graph parts, the actual font area affected is determined by the value of the **FontArea** attribute.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Font** tab and selecting the **Change font** push button.

Allowed Value

The font name

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldfont = %getatr('win01': 'ST1': 'FontName')
 *
C                   EVAL      %setatr('win01': 'ST1': 'FontName') = 'COURIER'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
```

```
C     'ST1'         GETATR    'FONTNAME'    oldfont          64
 *
C     'ST1'         SETATR    'COURIER'     'FONTNAME'
 *
```

FontSize

Specifies the size of the font to be used with this part. The **FontSize** attribute is expressed in points.

Type Numeric

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Group Box
List Box	Message Subfile	Multiline Edit	Notebook
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Window*	

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Note: In the case of the Calendar and Graph parts, the actual font area affected is determined by the value of the **FontArea** attribute.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Font** tab and selecting the **Change font** push button.

Allowed Value

The font size specified in points

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldsize = %getatr('win01': 'ST1': 'FontSize')
 *
C                   EVAL      %setatr('win01': 'ST1': 'FontSize') = 10
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         GETATR    'FONTSIZE'    oldsize           2 0
 *
C     'ST1'         SETATR    10            'FONTSIZE'
 *
```

FontStrike

*** Restriction:** This attribute is unsupported in Java applications.

Determines if the text or label is displayed in Strikeout font.

Type Numeric

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Group Box
List Box	Message Subfile	Multiline Edit	Notebook
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Window	

Note: In the case of the Calendar and Graph parts, the actual font area affected is determined by the value of the **FontArea** attribute.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Font** tab and selecting the **Change font** push button.

Allowed Value

FontStrike can be one of the following:

0 Font is not displayed in strikeout

1 Font is displayed in strikeout

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      strikeout=%getatr('win01':'ST1':'FontStrike')
 *
C     strikeout     IFEQ      0
C                   EVAL      %setatr('win01':'ST1':'FontStrike')=1
C                   ELSE
C                   EVAL      %setatr('win01':'ST1':'FontStrike')=0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         GETATR    'FONTSTRIKE'  strikeout         1 0
 *
C     'ST1'         SETATR    1             'FONTSTRIKE'
 *
```

FontUnder

*** Restriction:** This attribute is unsupported in Java applications.

Determines if the text or label is Underlined.

Type Numeric

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Group Box
List Box	Message Subfile	Multiline Edit	Notebook
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Window	

Note: In the case of the Calendar and Graph parts, the actual font area affected is determined by the value of the **FontArea** attribute.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **Font** tab and selecting the **Change font** push button.

Allowed Value

FontUnder can be one of the following:

0 Text is not underlined

1 Text is underlined

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      underline=%getatr('win01':'ST1':'FontUnder')
 *
C     underline     IFEQ      0
C                   EVAL      %setatr('win01':'ST1':'FontUnder')=1
C                   ELSE
C                   EVAL      %setatr('win01':'ST1':'FontUnder')=0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ST1'         GETATR    'FONTUNDER'   underline         1 0
 *
C     'ST1'         SETATR    1             'FONTUNDER'
 *
```

ForeColor

A numeric value that determines the foreground color of a part.

If the foreground color had been set using the **ForeMix** attribute, the foreground color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Check Box	Combination Box	Container	Entry Field
Group Box	List Box	Message Subfile	Multiline Edit
Notebook	Push Button	Radio Button	Slider
Spin Button	Static Text	Subfile	

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The color must be one of the following:

-1 Default (not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Note: The default color (-1) cannot be used for SETATR.

Free form example

Reverse image a field:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      fcolor = %getatr('win01': 'EF1': 'ForeColor')
C                   EVAL      bcolor = %getatr('win01': 'EF1': 'BackColor')
 *
C                   EVAL      %setatr('win01': 'EF1': 'ForeColor') = 2
C                   EVAL      %setatr('win01': 'EF1': 'BackColor') = 15
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'FORECOLOR'   fcolor            2 0
 *
C     'EF1'         SETATR    2             'FORECOLOR'
 *
```

ForeMix

Queries or sets the foreground color mix.

Type String

Applicable Parts

Check Box	Combination Box	Container	Entry Field
Group Box	List Box	Message Subfile	Multiline Edit
Notebook	Push Button	Radio Button	Slider
Spin Button	Static Text	Subfile	

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

* **Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The foreground color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      colormix = %getatr('win01': 'EF1': 'ForeMix')
 *
C                   EVAL      %setatr('win01': 'EF1': 'ForeMix') = '10:23:200'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    '10:23:200'   'FOREMIX'
 *
```

Format

Determines the format of the data being referenced by the **Item** attribute and sent to the server application in a DDE conversation. The value of this attribute is determined by the server application. Refer to the server application documentation for valid format types. The default value is *cf_text*. This attribute should be set before you use the **Poke** attribute.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The format of the data referenced and sent to the server application

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'Format') = 'bitmap'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'bitmap'      'Format'
 *
```

FrameRate

Determines the time, in milliseconds, to pause between frames in an animated GIF sequence.

Type Numeric

Applicable Parts

Animation Control

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The number of milliseconds to pause between images.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'FrameRate')=1000
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ANI'         SETATR    1000          'FRAMERATE'
 *
```

FrmtString

Indicates how the month and year are displayed together.

Wherever MMONTH or YYEAR is found in the format string it will be replaced by the current month or year respectively. For example, the format string ″MMONTH - YYEAR″ for July 1997 becomes ″July - 1997″.

Type String

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'FrmtString') =
C                             'MMONTH/YYEAR'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    'MMONTH/YYEAR''FRMTSTRING'
 *
```

GetItem

Returns an item from a list. The item to be returned is determined by first setting the **Index** attribute.

Type String

Applicable Parts

Combination Box List Box Message Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      x = %getatr('win01': 'LB1': 'GetItem')
 *
```

Fixed form example

Save an item in a list box, then change it:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    3             'INDEX'
C     'LB1'         GETATR    'GETITEM'     saveitem         64
C     'LB1'         SETATR    'New string'  'SetItem'
 *
```

GetNewID

Returns a new, unique ID for a container record. Use this attribute to guarantee that each container record ID is unique. The record ID returned is a numeric value.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      id = %getatr('win01': 'CN1': 'GetNewID')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         GETATR    'GetNewID'    id
 *
```

GetRcdFld

Returns the contents of a field in a container record. The container's **RecordID** attribute must be set to indicate which container record is to be used. Also, the **ColNumber** attribute must be set to indicate from which field the data is to be retrieved.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'RecordID')=ID
C                   EVAL      %setatr('win01': 'CN1': 'ColNumber')=1
C                   EVAL      data = %getatr('win01': 'CN1': 'GetRcdFld')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    ID            'RECORDID'
C     'CN1'         SETATR    1             'COLNUMBER'
C     'CN1'         GETATR    'GETRCDFLD'   data
 *
```

GetRcdIcon

Returns the filename for the icon used to represent this record in the tree and icon view. The **RecordID** attribute must be set to identify which record is being referenced.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      file = %getatr('win01': 'CN1': 'GetRcdIcon')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         GETATR    'GETRCDICON'  file
 *
```

GetRcdText

Returns the icon text for a record in the container. The **RecordID** attribute must be set to identify which record is being referenced.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      text = %getatr('win01': 'CN1': 'GetRcdText')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         GETATR    'GETRCDTEXT'  text
 *
```

GetTables

Setting this attribute to 1 retrieves the list of tables stored in a data source. The ODBC/JDBC Interface part returns the list as a data set. This attribute is equivalent to the SQL Tables ODBC API. You can use the **FetchNext** attribute to get the table records from the returned data set.

For ODBC, each record in the list corresponds to a table and contains the following information:

Column	Column Name	Data Type	Comments
1	Table qualifier	Character	
2	Table owner	Character	The table owner
3	Table name	Character	The name of the table
4	Table type	Character	A string representing the table type
5	Remarks	Character	A description of the table

Refer to the ODBC API documentation for details on the column descriptions.

For JDBC, the following information is returned:

Column	Column Name	Data Type	Comments
1	TABLE_CAT	Character	Table catalog (may be null)
2	TABLE_SCHEM	Character	Table schema (may be null)
3	TABLE_NAME	Character	The name of the table
4	TABLE_TYPE	Character	A string representing the table type. Typical values include TABLE, VIEW, SYSTEM TABLE, ALIAS, and so on.
5	REMARKS	Character	A description of the table

Refer to the JDBC API documentation for details on the column descriptions.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'GetTables')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'GETTABLES'
 *
```

GetTables Example

This example shows how to get a list of tables from a connected data source and list them in the listbox TABLES:

```
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
C     'ODBC1'       Setatr    1             'GetTables'
C     'ODBC1'       Setatr    1             'Fetch'
C     'ODBC1'       Getatr    'IsData'      Temp              1 0
 *
C                   If        Temp = 1
C     'Tables'      Setatr    0             'RemoveItem'
 *
C                   DoW       Temp = 1
C     'ODBC1'       Setatr    3             'Column'
C     'ODBC1'       Getatr    'CharData'    TableName        20
C     'Tables'      Setatr    TableName     'AddItemEnd'
C     'ODBC1'       Setatr    1             'FetchNext'
C     'ODBC1'       Getatr    'IsData'      Temp
C                   EndDo
 *
C                   EndIf
```

GnEqGrpCol

Generates a random set of colors such that all datapoints in the same group are the same color.

This attribute is useful when showing a bar graph.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The value can only be set to 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'GR1':'GnEqGrpCol')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GR1'         SETATR    1             'GNEQGRPCOL'
 *
```

GnEqPntCol

Generates a random set of colors such that all values at a given datapoint are the same color.

This is useful when showing a pie chart.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The value can only be set to 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'GR1':'GnEqPntCol')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GR1'         SETATR    1             'GNEQPNTCOL'
 *
```

GraphType

Indicates which type of graph is used.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

GraphType can be one of the following:

1 Line graph

2 Bar graph

3 Bar and line graph

4 Pie chart

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      GRtype = %getatr('win01': 'GRT1': 'GraphType')
 *
C                   EVAL      %setatr('win01': 'GRT1': 'GraphType') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRT1'        GETATR    'GRAPHTYPE'   'GRtype'          1 0
 *
C     'GRT1'        SETATR    2             'GRAPHTYPE'
 *
```

GrphHiLite

Determines whether or not an item is highlighted.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

GrphHiLite can be one of the following:

0 The item is not highlighted

1 The item is highlighted

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      lite = %getatr('win01': 'GR1': 'GrphHiLite')
 *
C     lite          IFEQ      0
C                   EVAL      %setatr('win01': 'GR1': 'GrphHiLite') = 1
C                   ELSE
C                   EVAL      %setatr('win01': 'GR1': 'GrphHiLite') = 0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GR1'         GETATR    'GRPHHILITE'  lite              1 0
 *
C     'GR1'         SETATR    1             'GRPHHILITE'
 *
```

GroupLabel

Indicates the label of the current group. The group labels are used as a heading for each group of points if the graph is in pie chart mode and is used in the legend if the legend is set to display group labels.

Type String

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'graph1':'GroupLabel')='July'
 *
```

Fixed form example

Save the current label for a graph part, and set it to a new value:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRAPH01'     GETATR    'GROUPLABEL'  oldlabel
C     'GRAPH01'     SETATR    'July'        'GROUPLABEL'
 *
```

Handle

*** Restriction:** This attribute is unsupported in Java applications.

Returns the window handle of the part. If you are familiar with Windows APIs, you can use this handle to invoke API functions. For ODBC processing, returns the environment handle.

For example, when the ODBC/JDBC Interface part connects to a database, an environment handle is created and returned to the part. The part uses this handle to communicate with the ODBC API. You can use the **Handle** attribute to retrieve the handle variable in your program and invoke ODBC API functions. The handle variable is a 32-bit value. You should code the variable to receive the handle as an integer data type with a length of 10 and 0 decimals:

```
DName++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++
D hWnd            S             10I 0
```

Type Numeric

Applicable Parts

All parts **except** the following: ActiveX, *component, Component Reference, Menu Bar, Menu Item, Notebook Page, Notebook Page with Canvas, Submenu, Timer.

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

Returns the part's handle.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      hWnd = %getatr('win01': 'FW1': 'Handle')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'FW1'         GETATR    'HANDLE'      hWnd
 *
```

HasPrpPage

Returns a value of 1 if the ActiveX object has its own property-page dialog. Returns a 0 if it does not have one.

Type Numeric

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      PPG=%getatr('win01':'AX1':'HasPrpPage')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         GETATR    'HASPRPPAGE'  PPG
 *
```

HdgBGClr

A numeric value that determines the background color of a subfile heading.

If the background color has been set using the **HdgBGMix** attribute, the background color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      HBcolor = %getatr('win01': 'SFL1': 'HdgBGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'HdgBGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'HDGBGCLR'    HBcolor           2 0
 *
C     'SFL1'        SETATR    5             'HDGBGCLR'
 *
```

HdgBGMix

Queries or sets the background color mix of a subfile heading.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The background color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      HBclrMx=%getatr('win01':'SFL1':'HdgBGMix')
 *
C     HBclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01':'SFL1':'HdgBGMix')='10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'HDGBGMIX'
 *
```

HdgFGClr

A numeric value that determines the foreground color of a subfile heading.

If the foreground color has been set using the **HdgFGMix** attribute, the foreground color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      HFcolor = %getatr('win01': 'SFL1': 'HdgFGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'HdgFGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'HDGFGCLR'    HFcolor           2 0
 *
C     'SFL1'        SETATR    5             'HDGFGCLR'
 *
```

HdgFGMix

Queries or sets the foreground color mix of a subfile heading.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The foreground color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      HFclrMx = %getatr('win01': 'SFL1': 'HdgFGMix')
 *
C     HFclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01': 'SFL1': 'HdgFGMix')
C                              = '10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'HDGFGMIX'
 *
```

HdgIdx

Identifies the row in a multiline column heading. For example, a value of 1 specifies the first row. Use this attribute together with the **ColNumber** attribute to retrieve or change the text specified in the **HdgText** attribute.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

HdgIdx can be from 1 to 3.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'SF1':'HdgIdx')=3
C                   EVAL      %setatr('win01':'SF1':'ColNumber')=2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    3             'HDGIDX'
C     'SF1'         SETATR    2             'COLNUMBER'
 *
```

HdgText

The text in the column heading. This text string can contain up to three lines of text. It can be a message ID from the message file, or substitution label. To change this text at run time, set the **ColNumber** and **HdgIdx** attributes, first.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'SF1':'HdgText')='HEAD1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    'HEAD1'       'HDGTEXT'
 *
```

Height

Specifies the height of the part in pixels.

The **Height** attribute cannot exceed the maximum height determined by the display resolution used. For example, you cannot create a part 600 pixels high when working with a 480 pixel height display.

Type Numeric

Applicable Parts

ActiveX	Calendar	Canvas	Check Box
Combination Box	Container	Entry Field	Graph
Graphic Push Button	Group Box	Horizontal Scroll Bar	Image
Java Bean	List Box	Media Panel	Message Subfile
Multiline Edit	Notebook	ODBC/JDBC Interface	Outline Box
Progress Bar	Push Button	Radio Button	Slider
Spin Button	Static Text	Subfile	Vertical Scroll Bar
Window			

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The height of the part in pixels

Free form example

Change the height of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      howtall = %getatr('win01': 'PB1': 'Height')
C     howtall       add       150           howtall
C                   EVAL      %setatr('win01': 'PB1': 'Height') = howtall
 *
```

Fixed form example

Make a pushbutton 100 pixels shorter:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'PB1'         GETATR    'HEIGHT'      OldHeight         2 0
C     OldHeight     SUB       100           NewHeight         2 0
C     'PB1'         SETATR    NewHeight     'HEIGHT'
 *
```

HelpEnable

Displays Help information when the user presses a push button or graphic push button. This is most useful when you wish to display general Help for a window. Normally, to display Help for a part, the user presses the F1 key when the part is in focus.

Note: The user must give the push button or graphic push button part focus before help can be displayed.

Type Numeric

Applicable Parts

Graphic Push Button Push Button

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

* **Note:** Can be set using the properties notebook in the GUI designer. Do this by selecting the **Display help** style.

Allowed Value

HelpEnable can be one of the following:

0 Help is not enabled for this part

1 Help is enabled for this part

Free form example

Toggle the Help-enabled state of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'PB1': 'HelpEnable')
 *
C     rc            IFEQ      0
C                   EVAL      %setatr('win01': 'PB1': 'HelpEnable') = 1
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'PB1': 'HelpEnable') = 0
C                   ENDIF
 *
```

Fixed form example

Toggle the Help-enabled status of an entry field:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'PB1'         GETATR    'HELPENABLE'  HasHelp           1 0
C     HasHelp       IFEQ      1
C     'PB1'         SETATR    0             'HELPENABLE'
 *
C                   ELSE
C     'PB1'         SETATR    1             'HELPENABLE'
C                   ENDIF
 *
```

HelpWindow

Causes a specific help panel window to be displayed. The helpid value is the help ID that is specified in the help file. If a help ID value of 0 is used, a Windows style ″help topics″ dialog will be displayed.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The helpid

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'HelpWindow')=
C                             12
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    12            'HELPWINDOW'
 *
```

Hidden

Determines if a column in a subfile is visible or not. The **ColNumber** attribute must be set to indicate which column is affected.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Hidden can be one of the following:

0 Column is not hidden

1 Column is hidden

Free form example

Toggle the Hidden state of a column

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'SF1': 'Hidden')
 *
C     rc            IFEQ      0
C                   EVAL      %setatr('win01': 'SF1': 'Hidden') = 1
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'SF1': 'Hidden') = 0
C                   ENDIF
 *
```

Fixed form example

Toggle the Hidden state of an attribute

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'HIDDEN'      IsHidden          1 0
C     IsHidden      IFEQ      1
C     'SF1'         SETATR    0             'HIDDEN'
 *
C                   ELSE
C     'SF1'         SETATR    1             'HIDDEN'
C                   ENDIF
 *
```

HighLight

Determines whether the specified part is highlighted. A highlighted part has the same appearance as a button that has mouse button 1 pressed while the mouse pointer is over the button control.

Type Numeric

Applicable Parts

Check Box*	Graphic Push Button	Push Button	Radio Button*

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

HighLight can be one of the following:

0 Part is not highlighted

1 Part is highlighted

Free form example

Toggle the HighLight state of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'PB1': 'HighLight')
 *
C     rc            IFEQ      0
C                   EVAL      %setatr('win01': 'PB1': 'HighLight') = 1
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'PB1': 'HighLight') = 0
C                   ENDIF
 *
```

Fixed form example

Toggle the HighLight state of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'PB1'         GETATR    'HIGHLIGHT'   IsHighlighted     1 0
C     IsHilighted   IFEQ      1 0
C     'PB1'         SETATR    0             'HIGHLIGHT'
 *
C                   ELSE
C     'PB1'         SETATR    1 0           'HIGHLIGHT'
C                   ENDIF
 *
```

HitItem

After **UnderPoint** is set, **HitItem** will indicate what aspect of the graph was under the specified point.

If a bar or a slice is hit **DataGroup** and **DataPoint** are set.

If a graph is in pie chart mode and a group label is hit, **DataGroup** is set.

Note: This attribute is not applicable for Java applications.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

One of the following:

0 Nothing was hit

1 A bar was hit

2 A slice was hit

3 The legend was hit

4 The title was hit

5 The X-Axis label was hit

6 The Y-Axis label was hit

7 A group label was hit

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      itemhit = %getatr('win01': 'GRA1': 'HitItem')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        GETATR    'HITITEM'     itemhit           1 0
 *
```

HlitPoints

***Restriction:** This attribute is unsupported in Java applications.

If **HlitPoints** is set then all of the data items in the group indicated by **DataGroup** will be highlighted. This is only relevant if the graph is in bar graph mode.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The allowed values are:

0 The current data group is not highlighted

Any non-zero value
The current data group is highlighted

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GP1': 'HlitPoints') = 20
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GP1'         SETATR    20            'HLITPOINTS'
 *
```

HostName

*** Restriction:** This attribute is unsupported in Java applications.

Returns the host name of the workstation running the VisualAge RPG application in the form of *host_name ip_address*. For example:

```
is267sys 9.21.230.171
```

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      HostIs=%getatr('*component':'*component':'HostName')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'HOSTNAME'    HOSTIS
 *
```

HRule

Displays horizontal rules.

Type Numeric

Applicable Parts

Calendar Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

HRule can be one of the following:

0 Turns rule off

1 Turns rule on

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Hzrule = %getatr('win01': 'SF1': 'HRule')
 *
C     Hzrule        IFEQ      0
C                   EVAL      %setatr('win01': 'SF1': 'HRule') = 1
C                   ELSE
C                   EVAL      %setatr('win01': 'SF1': 'HRule') = 0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'HRULE'       Hzrule            1 0
 *
C     'SF1'         SETATR    1             'HRULE'
 *
```

IconHandle

*** Restriction:** This attribute is unsupported in Java applications.

Returns the handle of the window icon.

Type Numeric

Applicable Parts

Window	Window with Canvas

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ichW=%getatr('win01':'FW1':'IconHandle')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'FW1'         GETATR    'ICONHANDLE'  ICHW
 *
```

Index

Establishes the index value for the list portion of a part. This value determines which item is affected when using particular attributes. For example, the **Index** attribute must be set for an item before the **GetItem** attribute can be used to retrieve it.

Type Numeric

Applicable Parts

Combination Box List Box Message Subfile Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The index value

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'Index') = 4
 *
C                   EVAL      index = %getatr('win01': 'LB1': 'Index')
 *
```

Fixed form example

Save the current index value for a list box, then change it:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'INDEX'       OldIndex          2 0
C     'LB1'         SETATR    NewIndex      'INDEX'           2 0
 *
```

InPlace

Note: This attribute applies to Windows only.

Use **InPlace** to designate where the video file should play. If **InPlace** is set to 1, the video will play in the media part's position. If it is set to 0, the file will play in a separate window.

Type Numeric

Applicable Parts

Media

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

InPlace can be one of the following:

0 The file will open in a separate window

1 The file will open in the position of the media part

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'AUD1': 'InPlace') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AUD1'        SETATR    1             'INPLACE'
 *
```

InsertItem

*** Restriction:** This attribute is unsupported in Java applications.

Inserts an item in the list portion of a part. The current setting of the **Sequence** attribute for the part determines where the item is inserted in the list.

If the **Index** method is used to insert items in the list, then the item at the index value and all items following it are moved down one position in the list.

Type String

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The item to insert in the list

Free form example

Insert an item in a list box:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'Index') = 3
C                   EVAL      %setatr('win01': 'LB1': 'InsertItem') = 'new'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    3             'INDEX'
C     'LB1'         SETATR    'new'         'InsertItem'
 *
```

InsertLine

Inserts the specified string in the multiline edit at the position specified by the current **LineNumber** attribute. All lines following the inserted line are shifted down one line.

Type String

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The string to add to the multiline edit

Free form example

Insert text at offset 10 in a multiline edit:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'LineNumber') = 10
C                   EVAL      %setatr('win01': 'MLE1': 'InsertLine') = 'text'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    10            'LINENUMBER'
C     'MLE1'        SETATR    'text'        'INSERTLINE'
 *
```

InsertMode

*** Restriction:** This attribute is unsupported in Java applications.

Determines if the entry field portion of the part is in insert mode. When insert mode is on, typing characters in a field will cause characters to the right of the cursor to shift. When it is off, typing characters will cause overtyping of existing characters.

Type Numeric

Applicable Parts

Entry Field

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

InsertMode can be one of the following:

0 The entry field is not in insert mode

1 The entry field is in insert mode

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      mode = %getatr('win01': 'EF1': 'InsertMode')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'INSERTMODE'  InsertMode        1 0
 *
```

InsertRow

Setting this attribute to 1 adds a new row to the table. Before the new row is added, data is first moved from the program fields that are bound to the table columns.

Note: In Java applications, this attribute requires the appropriate JDBC 2.0 compliance driver.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'InsertRow')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    1             'INSERTROW'
 *
```

InsertText

Inserts specified text at the current cursor position of a multiline edit.

Type String

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The text string to be inserted

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'InsertText') = 'New'
 *
```

Fixed form example

Add text to a multiline edit at the current cursor position:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    'New'         'InsertText'
 *
```

Interval

Determines the interval value for a timer part. This attribute is used with the **Multiplier** attribute to determine the number of milliseconds that will elapse before the timer generates a **Tick** event.

Type Numeric

Applicable Parts

Timer

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The interval value in milliseconds

The maximum interval value is 65 535.

The minimum interval value is 100.

Free form example

Change the interval value for a timer part to one second (1000 milliseconds):

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldint=%getatr('win01':'TIMER01':'Interval')
 *
C                   EVAL      %setatr('win01':'TIMER01':'Interval')=1000
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'TIMER01'     GETATR    'INTERVAL'    OldInt            6 0
 *
C     'TIMER01'     SETATR    1000          'INTERVAL'
 *
```

InUse

*** Restriction:** This attribute is unsupported in Java applications.

Queries whether an object in the container currently has in-use emphasis. An object with in-use emphasis is inside an open design window, and is shown with a shaded box behind its icon, like this:

The **RecordID** attribute must be set to indicate which record is being referenced.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

InUse can be one of the following:

0 Not in use

1 In use

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'InUse') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'INUSE'
 *
```

InvName

*** Restriction:** Applies only to Java applications.

Specifies the name of the part that invokes, or anchors the pop-up menu. This attribute is used together with the **InvPname**, **X**, and **Y** attributes.

Type String

Applicable Parts

Pop-up Menu

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'POP1':'InvName')='INVN1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'POP1'        SETATR    'INVN1'       'INVNAME'
 *
```

InvPName

*** Restriction:** Applies only to Java applications.

Specifies the parent name of the part that invokes, or anchors the pop-up menu. This attribute is used together with the **InvName**, **X**, and **Y** attributes.

Type String

Applicable Parts

Pop-up Menu

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'POP1':'InvPName')='INVPN1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'POP1'        SETATR    'INVPN1'      'INVPNAME'
 *
```

IsData

Indicates if a **Fetch**, **FetchNext**, or **FetchPrior** operation returned any data.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

0 Indicates no data was returned.

1 Indicates data was returned.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Flag=%getatr('win01':'ODBC1':'IsData')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'ISDATA'      FLAG
 *
```

Item

A DDE item is the specific unit of data that is referenced in a DDE conversation. An example of a DDE item would be a cell in a spreadsheet application. The format of a DDE item is determined by the server application. Refer to the server application documentation for this format. For the spreadsheet example, this could be 'A1'.

The following table summarizes which parts can be an item in a DDE conversation, and which attribute is referenced during a **Poke** or **Request** event. If any other part type is referenced, the operation is ignored.

Part Type	Poke	Request
Media	Filename	Filename
Check Box	Label	Label
Entry Field	Text	Text
Image	Filename	Filename
List Box	none	Selected item
Slider	Value	Value
Static Text	Label	Label
Timer	Count	Count

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The unit of data being referenced in a DDE conversation

If the server program is a VisualAge RPG component, the **Item** is the name of the part being referenced. It is specified as follows:

```
'window|part'
```

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'Item') = 'A1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'A1'          'ITEM'
 *
```

ItemCount

Returns the number of items in the list of values of a subfile column. First set the **ColNumber** attribute to identify which column to work with. The subfile column must have Values as its validation type.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ItCnt = %getatr('win01': 'SF1': 'ItemCount')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'ITEMCOUNT'   ITCNT
 *
```

ItemKey

Returns the key value of an item in the list. The **Index** attribute must be set first to indicate which item is being referenced.

Note: If a key is not defined **ItemKey** will return a null string.

Type String

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Note: **ItemKey** for combination box can be set using the properties notebook.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      keyitem = %getatr('win01': 'LB1': 'ItemKey')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'ITEMKEY'     keyitem
 *
```

Label

Text string that is associated with a part, such as the text that would appear on a push button.

The **label** attribute can be in one of the following forms:

- A string as typed
- A substitution label
- A message ID from the message file

If the title of a container column has blanks, the container does not display correctly. To correct this, replace all blanks in the title by the underscore character (_).

Windows still display the title bar even though the Title Bar has been disabled in the properties notebook. When the setting for Title Bar is not selected, the user can still move the window by using the mouse on the title bar area, but has to close the window using the keyboard.

Type String

Applicable Parts

Check Box	Container	Group Box	Menu Item
Push Button	Radio Button	Static Text	Window

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by using the **General** tab and specifying a value in the **Label** or **Title** fields.

Allowed Value

The part label

Free form example

Change the label on a push button:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      checked = %getatr('win01': 'CB1': 'Checked')
 *
C     checked       IFEQ      0
C                   EVAL      %setatr('win01': 'PB1': 'Label') = 'No'
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'PB1': 'Label') = 'Yes'
C                   ENDIF
 *
```

Fixed form example

Change text on a push button:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'PB1'         GETATR    'LABEL'       OldText           2 0
C     'PB1'         SETATR    'Cancel'      'LABEL'
 *
```

LabelPlace

Indicates where on the graph the label should appear for the indicated data item.

This attribute does not apply when the graph is in pie chart mode.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

LabelPlace can be one of the following:

0 The data item should not have a label

1 The label should be displayed at the X-Axis

2 The label should be displayed above the bar if the value of the item is positive, or below the bar if the item's value is negative

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('Win1': 'Gra1': 'LabelPlace') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'Gra1'        SETATR    2             'LABELPLACE'
 *
```

Left

Specifies the coordinate, in pixels, from the left edge of the object or screen that contains this part.

Type Numeric

Applicable Parts

ActiveX	Animation Control	Calendar	Canvas
Check Box	Combination Box	Component Reference	Container
DDE Client	Entry Field	Graph	Graphic Push Button
Group Box	Horizontal Scroll Bar	Image	Java Bean
List Box	Media	Media Panel	Multiline Edit
Notebook	ODBC/JDBC Interface	Outline Box	Progress Bar
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Timer	Vertical Scroll Bar
Window			

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The number of pixels

Free form example

Move a part to the left in its window:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      leftside = %getatr('win01': 'image1': 'Left')
C                   EVAL      %setatr('window': 'part': 'Left') =
                              leftside - 120
 *
```

Fixed form example

Move a push button 200 pixels to the right:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'PB01'        GETATR    'LEFT'        OldLeft           2 0
C     OldLeft       ADD       200           NewLeft           2 0
C     'PB01'        SETATR    NewLeft       'LEFT'
 *
```

LegendType

Indicates the type of legend used for the graph part.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value must be one of the following:

0 No legend

1 Points only

2 Groups only

3 All items

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('Win1': 'Gra': 'LegendType') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'Gra1'        SETATR    2             'LegendType'
 *
```

Length

Note: This attribute applies to Windows only.

Specifies the length of the file in milliseconds.

Type Numeric

Applicable Parts

Media

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Len = %getatr('win01': 'AUD01': 'Length')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AUD01'       GETATR    'LENGTH'      Len               6 0
 *
```

LineNumber

Determines at which line the **InsertLine** attribute will insert a new line in a multiline edit. This value must be greater than zero and less than or equal to the current **NbrOfLines** attribute. If it is greater than the current **NbrOfLines** attribute, the new line will be inserted after the last line.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The line number

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldline = %getatr('win01': 'MLE01': 'LineNumber')
C                   EVAL      %setatr('win01': 'MLE01': 'LineNumber') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE01'       GETATR    'LINENUMBER'  OldLine           2 0
C     'MLE01'       SETATR    5             'LINENUMBER'
 *
```

LineText

Retrieves the line of text of a multiline edit line specified by **LineNumber**.

Type String

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

The text of the line

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      lines = %getatr('win01': 'MLE1': 'NbrOfLines')
C                   Do        lines                       ix
C                   EVAL      %setatr('win01': 'MLE1': 'LineNumber') = ix
C                   EVAL      nextline= %getatr('win01': 'MLE1': 'LineText')
C                   ...
C                   EndDo
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'LINETEXT'    line
 *
```

LookNFeel

*** Restriction:** A **Java-only** attribute.

Allows your Java applications to use Swing's Pluggable Look and Feel (PLAF) mechanism to control how your GUI ″looks and feels″.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

LookNFeel can be one of the following:

Look **String Value**

Metal javax.swing.plaf.metal.MetalLookAndFeel

Motif com.sun.java.swing.plaf.motif.MotifLookAndFeel

Windows
com.sun.java.swing.plaf.windows.WindowsLookAndFeel

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   eval      %setatr('*component':'*component':
C                             'LookNFeel')=
C                             'com.sun.java.swing.plaf.motif.' +
C                              'MotifLookAndFeel'
 *
```

Magnify

Specifies the value, in percentage, of the magnification slider on the image part controlling the scale of the displayed image. A zero value will make the image scale to fit the image part.

Type Numeric

Applicable Parts

Image

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The magnification slider value can be one of the following:

0 Scale to fit

25 - 200
The magnification value in percentage

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win1': 'Img1': 'Magnify') = 50
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'IMG1'        SETATR    50            'MAGNIFY'
 *
```

MapViewCol

Set this attribute to 1 to display a subfile data column (specified by the **ColNumber** attribute) in the order indicated by the **ViewColumn** attribute. The default is to display data columns in the same order as that listed in the Field List page of the Subfile properties notebook.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

1 Displays a subfile data column in the order indicated by the ViewColumn attribute.

Free form example

Display data column 2 (ColNumber) as viewable column 4 (ViewColumn).

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'SF1':'ColNumber') = 2
C                   EVAL      %setatr('win01':'SF1':'ViewColumn') = 4
C                   EVAL      %setatr('win01':'SF1':'MapViewCol') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    2             'COLNUMBER'
C     'SF1'         SETATR    4             'VIEWCOLUMN'
C     'SF1'         SETATR    1             'MAPVIEWCOL'
 *
```

Masked

When this attribute is set, typing into the entry field will display asterisk (*) characters rather than the characters being typed. This attribute is most often used when the user will be typing in sensitive data, such as a password or account number.

Type Numeric

Applicable Parts

Entry Field

Operation Codes

	Operational at Run Time
SETATR*	no
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The field's password protected state can be one of the following:

0 The field is not masked

1 The field is masked

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isMasked = %getatr('win01': 'EF1': 'Masked')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'MASKED'      isMasked          1 0
 *
```

Maximum

Determines the maximum value that can be returned from a slider or a spin button.

This attribute is allowed only for numeric spin buttons.

The maximum value is 65 535.

Type Numeric

Applicable Parts

Slider Spin Button

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Maximum value for the part

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldmax=%getatr('window':'SLIDER01':'Maximum')
 *
C                   EVAL      %setatr('window':'SLIDER01':'Maximum')=100
 *
```

Fixed form example

Change the maximum value for a slider:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SLIDER01'    GETATR    'MAXIMUM'     OldMax            3 0
C     'SLIDER01'    SETATR    100           'MAXIMUM'
 *
```

Method

Specifies the name of the ActiveX method to invoke, along with any values to be passed to the method. You control ActiveX parts by setting their properties and methods.

When some ActiveX controls are activated, such as Netscape 4.x, an application-modal dialog may be displayed and may be hidden under the GUI Designer or a VARPG application. Be aware of this, since the application or GUI Designer will appear to be locked up.

The syntax for the string value is:

```
method_name, value1, value2, ...
```

Restriction: The method and property parameters are restricted to numeric (that is, integer and float) and string (char*) data types. Data types such as Font, Color, UserDefined, pointer (PTR) are not supported.

Type String

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'Method')='PN1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    'PN1'         'METHOD'
 *
```

Method Example

In the following example, the DisplayText method of an ActiveX part is invoked and passed a single value. If the method expects a numeric value, pass the character equivalent.

```
D dspText          S           Inz('DisplayText,New Text')
 *
C     'MSCAL'   Setatr   dspText    'Method'
```

MiniIcon

This attribute determines the presentation of the icons in the container part. When the attribute is set, the icons appear in the container part as mini icons. If the attribute is not set, the icons appear as regular icons.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

MiniIcon can be one of the following:

0 Icons are displayed at normal size

1 Icons are displayed as mini icons

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'Cont1': 'Miniicon') = 0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'Cont1'       SETATR    0             'MINIICON'
 *
```

Minimum

Determines the minimum value that can be returned from a slider or a spin button.

This attribute is allowed only for numeric spin buttons.

The minimum value is -32 767.

Type Numeric

Applicable Parts

Slider Spin Button

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Minimum value for the part

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldmin = %getatr('window': 'SLIDER01': 'Minimum')
 *
C                   EVAL      %setatr('window': 'part': 'Minimum') = 0
 *
```

Fixed form example

Change the minimum value for a slider:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SLIDER01'    GETATR    'MINIMUM'     OldMin            3 0
C     'SLIDER01'    SETATR    0             'MINIMUM'
 *
```

Mode

Determines whether or not to play the video or GIF file.

Type Numeric

Applicable Parts

Animation Control

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

Mode can be one of the following:

1 Start playing the file

2 Stop playing the file

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'WIN1': 'Mode') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    1             'MODE'
 *
```

Month

Indicates the currently selected month. When set, the selected date in the calendar part will be updated to reflect the changes.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value
The month

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'Month') = 02
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    02            'Month'
 *
```

MonthArrow

Indicates whether or not month (< or >) or year (<< or >>) arrows are available so that users can scroll backwards or forwards through the months or years. If there are no arrows, users will have no way to change the currently viewed month or year.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

MonthArrow can be one of the following:

0 The arrows will not be shown

1 The arrows will be shown

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isshown = %getatr('win01': 'CAL1': 'MonthArrow')
 *
C                   EVAL      %setatr('win01': 'CAL1': 'MonthArrow') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'MONTHARROW'  isshown
 *
```

MonthIdx

Allows the user to set data on a date not displayed on the current calendar page.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The index value

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'MonthIdx') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    2             'MonthIdx'
 *
```

MonthLen

Indicates the format for displaying the name of the month.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

MonthLen can be one of the following:

1 Full name

2 Abbreviated three-character name

3 Two digits

4 Two digits padded with a zero for months with only one digit

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'MonthLen') = 3
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'MONTHLEN'    MonthLen          1 0
 *
```

MouseIcon

*** Restriction:** This attribute is unsupported in Java applications.

Specifies the name of the icon file to use for the mouse pointer's shape. Set the **MouseShape** value to 99 before using this attribute.

The file can be a cursor (.CUR) or icon (.ICO or .ANI) file.

Type String

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'MouseIcon')=Myicon
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    MYICON        'MOUSEICON'
 *
```

MouseShape

Changes the shape of the mouse pointer when the pointer is positioned on the window frame of the parent or its children.

Type Numeric

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

MouseShape can be one of the following:

0 Default

1 Arrow icon

2 Hourglass icon

3 Stop or Illegal icon

4 Arrow with hourglass (application starting)

5 Cross hair

6 I-beam

7 Up arrow

8 Northeast/Southwest arrows

9 North/South arrows

10 Northwest/Southeast arrows

11 East/West arrows

12 All sides arrows (North/South/East/West)

13 Pointing hand cursor

99 User-defined icon as specified by the **MouseIcon** attribute. (This value only applies to Windows applications.)

The **MouseShape** attribute is not persistent; it applies to the window frame only. If you change the mouse pointer shape from the default, you must reset this attribute to **0** to use the default pointer shape.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('Win1': 'Win1': 'MouseShape') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'Win1'        SETATR    2             'MOUSESHAPE'
 *
```

MsgData

Specifies the substitution text to use for variables appearing in message text. For a message that has several variables, delimit each substitution text string by a space. Use the underscore (_) character to include blanks that are part of the substitution text. For example, to substitute the text strings ″A B C″ and ″CustNo″ for message variables %1 and %2, respectively, specify the message data as follows:

```
'A_B_C CustNo'
```

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The substitution string for the message variable

Free form example

Get message text for message 0001 and use CustNo for substitution data:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'MsgID')=0001
C                   EVAL      %setatr('*Component':'*Component':'MsgData')=CustNo
C                   EVAL      text = %getatr('*Component':'*Component':'MsgText')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    0001          'MSGID'
C     '*COMPONENT'  SETATR    CustNo        'MSGDATA'
C     '*COMPONENT'  GETATR    'MSGTEXT'     text
 *
```

MsgFile

*** Restriction:** This attribute is unsupported in Java applications.

Specifies the default message file to use when a message identifier is not found in the component's message file.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The full name of the default message file, *filename.txm*. The current path is used for the file search.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'MsgFile')
C                             = Fn
C                   EVAL      Mf = %getatr('*Component':'*Component':
C                             'MsgFile')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    Fn            'MSGFILE'
 *
C     '*COMPONENT'  GETATR    'MSGFILE'     Mf
 *
```

MsgID

Specifies the message number in the component message file.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The message number

Free form example

Get message text for message 0001 and use CusNo for substitution data:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'MsgID')=0001
C                   EVAL      %setatr('*Component':'*Component':'MsgData')=CusNo
C                   EVAL      text=%getatr('*Component':'*Component':'MsgText')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    0001          'MSGID'
C     '*COMPONENT'  SETATR    CusNo         'MSGDATA'
C     '*COMPONENT'  GETATR    'MSGTEXT'     text
 *
```

MsgSubText

Defines the message substitution text for a message subfile part. This attribute should be set before you add a message to a message subfile part using the **AddMsgID** attribute or the **AddMsgText** attribute. The message substitution text is referenced when a message has substitution variables in the message text. Message substitution variables are identified by a '%' symbol followed by a numeric value. Each message substitution variable is replaced by the corresponding blank delimited word in the **MsgSubText** attribute before the message is added to the message subfile part. The message substitution string will remain in effect until another **MsgSubText** attribute is set.

Type String

Applicable Parts

Message Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The message substitution text

Free form example

If message 1000 consisted of:

```
Press %1 to continue, %2 to cancel
```

then the message would be displayed as follows:

```
Press Yes to continue, No to cancel
```

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'MSG1':'MsgSubText')='Yes No'
C                   EVAL      %setatr('win01':'MSG1':'AddMsgID')=1000
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MSG1'        SETATR    'Yes No'      'MSGSUBTEXT'
C     'MSG1'        SETATR    1000          'ADDMSGID'
 *
```

MsgText

Returns the message text from the component message file for the specified message identifier.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

Get the message text for message 0001 :

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*Component':'*Component':'MsgID')=0001
C                   EVAL      text=%getatr('*Component':'*Component':'MsgText')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    0001          'MSGID'
C     '*COMPONENT'  GETATR    'MSGTEXT'     text
 *
```

Multiplier

Determines the number of timer interval values that must elapse before the timer causes a **Tick** event.

The maximum multiplier value is 65 535.

The minimum multiplier value is 1.

Type Numeric

Applicable Parts

Timer

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	no

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value
The multiplier value

Free form example

If the **Interval** attribute is set for 1 000 milliseconds, the timer will generate a **Tick** event every three seconds:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'TIMER01': 'Multiplier') = 3
 *
```

Fixed form example

Change the **Multiplier** attribute of a timer:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'TIMER01'     SETATR    3             'MULTIPLIER'
 *
```

MultSelect

Determines if more than one item may be selected in the list portion of a part. When this attribute is **not** set, selecting an item will deselect any currently selected item.

This attribute must be set for the list box to allow extended selection as provided by the **ExtSelect** attribute.

The **MultSelect** attribute has no effect on the **ExtSelect** attribute. The **ExtSelect** attribute overrides the settings for either **MultSelect** or single selection.

Type Numeric

Applicable Parts

List Box Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

The multiple selection status can be one of the following:

0 Only one item may be selected

1 Multiple items may be selected

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'LB1': 'MultSelect')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'MULTSELECT'  rc                1 0
 *
```

Name

Returns the name of the component.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

Get the name of the component:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      appname=%getatr('*Component':'*Component':'Name')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'NAME'        appname
 *
```

NbrOfImage

*** Restriction:** A **Java-only** attribute.

Sets the number of images to play in a GIF animation sequence. Use this attribute together with the **FileName** and **Mode** attributes.

For example, to play back a base file *File.GIF* as a sequence of 10 images, set **FileName** to File.GIF, **NbrOfImage** to 10, and **Mode** to 1. Alternatively, you could play back several different GIF images, like File1.GIF, File2.GIF, ... File10.gif.

Type Numeric

Applicable Parts

Animation Control

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The number of images to play back.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'NbrOfImage')=10
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    1             'NBROFIMAGE'
 *
```

NbrOfLines

Returns the number of lines in the multiline edit. If the multiline edit is empty, a value of zero is returned.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      NumL = %getatr('win01': 'MLE1': 'NbrOfLines')
 *
```

Fixed form example

Get the number of lines in a multiline edit:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'NBROFLINES'  NumL              2 0
 *
```

NbrOfSel

Returns the number of items selected in the list portion of a part.

If no items are currently selected, a value of zero is returned.

Type Numeric

Applicable Parts

List Box Message Subfile Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

Get the number of selected items in a list box:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      num=%getatr('win01':'LB1':'NbrOfSel')
 *
C     num           IFEQ      *ZERO
C                   EVAL      %setatr('win01':'MSG1':'AddMsgText')='None'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'NBROFSEL'    NbrSelected       2 0
 *
```

NextLine

Setting this attribute to 1 moves the scroll bar slider to a position that represents the next line in a list of items. The slider moves relative to the **Range** established for the scroll bar part.

For example, you may have a subfile part and want to show where a user is within a set of records when a command or cursor movement key is pressed. The scroll bar part attached to a subfile only shows where the user is based on the number of records currently in the subfile. By capturing the **VKeyPress** event, you can set the appropriate slider attribute. When the slider moves to its new position, a **Scroll** event is signalled and the **%Position** event attribute is updated.

The **PageSize** attribute determines the visible page size for the items displayed. The **Range** attribute determines the upper limit for the number of items to be scrolled through.

Type Numeric

Applicable Parts

Horizontal Scroll Bar

Vertical Scroll Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'VSC1':'NextLine')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'VSC1'        SETATR    1             'NEXTLINE'
 *
```

NextPage

Setting this attribute to 1 moves the scroll bar slider to a position that represents the next page in a list of items. The slider moves relative to the **Range** established for the scroll bar part.

For example, you may have a subfile part and want to show where a user is within a set of records when a command or cursor movement key is pressed. The scroll bar part attached to a subfile only shows where the user is based on the number of records currently in the subfile. By capturing the **VKeyPress** event, you can set the appropriate slider attribute. When the slider moves to its new position, a **Scroll** event is signalled and the **%Position** event attribute is updated.

The **PageSize** attribute determines the visible page size for the items displayed. The **Range** attribute determines the upper limit for the number of items to be scrolled through.

Type Numeric

Applicable Parts

Horizontal Scroll Bar

Vertical Scroll Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'VSC1':'NextPage')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'VSC1'        SETATR    1             'NEXTPAGE'
 *
```

NotSrcEvt

Use this attribute, in conjunction with the **NotSrcPart** and the **NotSrcWin** attributes, to have the component reference part monitor the occurrence of a specific event on a specific part in a specific window in another component. This attribute is used to identify the desired event. Use **NotSrcPart** and **NotSrcWin** to specify the particular part and window you wish to monitor for that event's occurrence.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      event = %getatr('win01': 'CR1': 'NotSrcEvt')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CR1'         GETATR    'NOTSRCEVT'   event
 *
```

NotSrcPart

Use this attribute, in conjunction with the **NotSrcEvt** and the **NotSrcWin** attributes, to have the component reference part monitor the occurrence of a specific event on a specific part in a specific window in another component. This attribute is used to identify the desired part. Use **NotSrcEvt** and **NotSrcWin** to specify the particular event you wish to monitor for that part and the particular window that part is found in.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      PartN = %getatr('win1': 'CR1': 'NotSrcPart')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CR1'         GETATR    'NOTSRCPART'  PartN
 *
```

NotSrcWin

Use this attribute, in conjunction with the **NotSrcEvt** and the **NotSrcPart** attributes, to have the component reference part monitor the occurrence of a specific event on a specific part in a specific window in another component. This attribute is used to identify the desired window. Use **NotSrcPart** and **NotSrcEvt** to specify the particular part and event you wish to monitor in that window.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      WinNam = %getatr('win1': 'CR1': 'NotSrcWin')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CR1'         GETATR    'NotSrcWin'   WinNam
 *
```

OCXProp

Determines the name of the property. This name is used to determine the property value with the **OCXValue** attribute.

Type String

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'OCXProp')='PN1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    'PN1'         'OCXPROP'
 *
```

OCXPropIdx

Sets or retrieves the index for an ActiveX array property. You can use this attribute together with the **OCXProp** (property name) and **OCXValue** (property value) attributes to manipulate array properties.

To set an element of a multi-dimensional array property, pass the index value for each element in a string as 'n1 n2 n3', where n1 is the index for dimension 1, and so on.

Type String

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'OCXPropIdx')='0'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    '0'           'OCXPROPIDX'
 *
```

OCXValue

Determines the value of the property as provided by the **OCXProp** attribute.

Type String

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'OCXValue')='VL1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    'VL1'         'OCXVALUE'
 *
```

OnTop

Sets the specified notebook page to be the current page, or determines if the notebook page is the current page.

Type Numeric

Applicable Parts

Notebook Page	Notebook Page with Canvas

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

1 Sets the specified notebook page to be the current page.

Returned Value

1 This notebook page is the current page.

0 This notebook page is **not** the current page.

Free form example

Set notebook page **NP4** to be the current page.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'NP4': 'OnTop') = 1
C                   EVAL      Curr = %getatr('win01': 'NP4': 'OnTop')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'NP4'         SETATR    1             'ONTOP'
C     'NP4'         GETATR    'ONTOP'       CURR
 *
```

OpenEdit

Determines if an entry field in a subfile record can be edited. Before setting this attribute, you must set the **ColNumber** and **Index** attributes.

The value returned by the **OpenEdit** attribute indicates whether any field in the subfile is currently open for editing. Therefore, it is not necessary to set the **ColNumber** and **Index** attributes before getting this attribute.

When the **OpenEdit** attribute is set to zero, any field that is currently open for editing is closed. The **ColNumber** and **Index** attributes are ignored.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

OpenEdit can be one of the following:

0 Field is not editable

1 Field is editable

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isopen = %getatr('win01': 'SF1': 'OpenEdit')
 *
C                   EVAL      %setatr('win01': 'SF1': 'OpenEdit') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'OPENEDIT'    isopen
 *
```

OS

Returns the name of the operating system.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

WINDOWS: Windows operating system

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ostring=%getatr('*Component':'*Component':'Os')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'OS'          ostring
 *
```

OutlineRcl

Indicates whether or not there is a rectangle outlining the **YearIdx/MonthIdx/DayIdx date**.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

OutlineRcl can be one of the following:

0 There is not a rectangle

1 There is a rectangle

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'OutlineRcl') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1             'OUTLINERCL'
 *
```

PageNumber

Determines which page number is currently selected.

Type Numeric

Applicable Parts

Notebook

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The currently selected page number

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'NB1': 'PageNumber') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'NB1'         SETATR    1             'PAGENUMBER'
 *
```

PageSize

For a subfile part, indicates how many rows of data are visible. This is useful if you want to do page-at-a-time processing.

For a horizontal or vertical scroll bar part, determines the visible page size. The **PageSize** value cannot exceed the **Range** value.

Type Numeric

Applicable Parts

Horizontal Scroll Bar	Vertical Scroll Bar	Subfile*

*** Restriction:** Can only get this attribute at run time for the Subfile part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

For a horizontal or vertical scroll bar part, the **PageSize** value can be between 1 and 32,767. It cannot exceed the **Range** value.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      PS = %getatr('win01':'SFL1':'PageSize')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'PAGESIZE'    PS
 *
```

Panel

Note: This attribute applies to Windows only.

For the image part, **panel** determines if the part has a magnification panel attached. The magnification panel contains two controls that allow manipulation of the image:

Scale to fit checkbox
When this checkbox is not checked, the image is scaled to fit the image part. When it is checked, the magnification slider is enabled.

Magnification slider
When enabled, this slider will adjust the magnification of the image between 25% and 200% of its original size.

For the media part, **panel** is used to show a media panel for playing files.

Type Numeric

Applicable Parts

Image Media

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

One of the following:

0 The magnification panel is not attached, or the play panel is hidden.

1 The magnification panel is attached, or the play panel is shown.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldval = %getatr('window': 'MMP1': 'Panel')
 *
C                   EVAL      %setatr('window': 'MMP1': 'Panel') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MMP1'        GETATR    'PANEL'       OldVal            1 0
C     'MMP1'        SETATR    1             'PANEL'
 *
```

PanelItem

Returns a value indicating which button on a media panel was last changed. A button can be changed either by the user clicking on it with the mouse, or by setting the **PanelMode** attribute. You could check this attribute after the user had clicked one of the media panel buttons. This would cause a **Press** event. You could then use the **PanelItem** attribute in your press action subroutine to determine what function to perform.

Type Numeric

Applicable Parts

Media Panel

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

The status of the last media-panel button pressed:

0 No button pressed

1 Pause

2 Play

3 Record

4 Stop

5 Next track

6 Previous track

7 Next index

8 Previous index

9 Volume

10 Position

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      button = %getatr('win01': 'MMP1': 'PanelItem')
 *
C     button        CASEQ     2             PLAY
C     button        CASEQ     3             RECORD
C                    ...
C                   ENDCS
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MMP1'        GETATR    'PANELITEM'   button            2 0
 *
C     button        CASEQ     2             PLAY
C     button        CASEQ     3             RECORD
C                    ...
C                   ENDCS
 *
```

PanelMode

Sets the operating mode for the media panel part.

Type Numeric

Applicable Parts

Media Panel

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The operating mode for a media panel part can be:

1 Pause

2 Play

3 Record

4 Stop

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MMP1': 'PanelMode') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MMP1'        SETATR    1             'PANELMODE'
 *
```

Parent

Set the parent part whose list of children is to be retrieved by the **PartList** attribute. The list of immediate children will be returned. To get the list of frame windows in a component, set the **Parent** attribute with the reserved name ***component**.

Note: This attribute must be set before **PartCount** or **PartList** can be used.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The format of the parent part name is:

```
PARENTNAME|PARTNAME
```

ParentName is the name of the window part that owns the **PartName**. If you want to get a list of parts in a window, use the canvas name as the part name.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'Parent')
C                             = 'WIN1|CANVS1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    'WIN1|CANVS1' 'PARENT'
 *
```

If you want to get a canvas name from a parent window part, you need to specify the window name twice. The following code example shows how to do this. (The *WinName* variable contains the Window name of the parent window.)

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
 * Set the Window as the parent; enter the window name twice.
C                   EVAL      %setatr('*component':'*component':'Parent')
C                             = WinName + '|' + WinName
 *
 * Get PartList which now has the canvas name and assign it to the Canvas variable.
C                   EVAL      Canvas=%getatr('*component':'*component':'PartList')
```

See the **PartList** description for information on the values returned by this attribute.

ParentId

Returns the record ID of the parent record of a record in a container. Before using this attribute, set the **RecordID** attribute to the id of the record whose parent's id you want to determine.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

The following example assumes that a record has been selected. The record ID of the parent record of the selected record will be determined.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
CSRN01Factor1+++++++Opcode(E)+Extended-factor2++++++++++++++++++++++++++----
 * Set the ID of the selected record
C                   EVAL      %setatr('Main':'CT1':'RecordID')=
C                             %getatr('Main':'CT1':'FirstSel')
 * Get the ID of the parent record of the selected record
C                   EVAL      Parent=%getatr('Main':'CT1':'ParentId')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CT1'         GETATR    'FIRSTSEL'    ID               4 0
C     'CT1'         SETATR    ID            'RECORDID'
C     'CT1'         GETATR    'PARENTID'    PARENT           4 0
 *
```

ParentList

Returns the blank, delimited list of parent ids for the current record. Before using this attribute, set the **RecordID** attribute to the id of the current record.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

The following example assumes that a record has been selected. The record ID of the parent record will be determined.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
CSRN01Factor1+++++++Opcode(E)+Extended-factor2++++++++++++++++++++++++++----
 * Set the ID of the selected record
C                   EVAL      %setatr('Main':'CT1':'RecordID')=
C                             %getatr('Main':'CT1':'FirstSel')
 * Get the list of parent IDs for the selected record
C                   EVAL      ParentL=%getatr('Main':'CT1':'ParentList')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CT1'         GETATR    'FIRSTSEL'    ID               4 0
C     'CT1'         SETATR    ID            'RECORDID'
C     'CT1'         GETATR    'PARENTLIST'  PARENTL          4 0
 *
```

ParentName

Returns the name of the part that contains the part referenced by the *part* parameter. For parts that are on a window, the name of the window is returned. When you are using this attribute for a window part, the window part name is returned.

Type String

Applicable Parts

All parts except the *component part

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      PName = %getatr('win01': 'EF1': 'ParentName')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'PARENTNAME'  PName             1 0
 *
```

PartCount

Returns the number of children for the parent part set by attribute **Parent**.

Type Numeric

Applicable Parts

*Component Java Bean

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

The number of children for the specified parent part.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Numchild=%getatr('*component':'*component':
C                             'PartCount')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'PARTCOUNT'   Numchild
 *
```

PartList

Returns a list of children parts for the parent part set by the **Parent** attribute.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

The format of the string is a continuous string of part names. Each part name occupies a fixed string length of 10 characters. If a part name portion is less than 10 characters, the name is right-padded with blanks.

```
FRAMEbbbbbCANVASbbbbENTRYFIELD
```

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      list = %getatr('*component': '*component':
C                             'PartList')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'PARTLIST'    list
 *
```

PartName

Returns the name of the part. This is the name that was assigned to the part when it was first created using the notebook setting.

The length of the name value is 10 characters. This value will be returned left-justified in the result field. If the result field is shorter than 10 characters, the name value will be truncated. If the result field is greater than 10 characters, the field will be padded with blanks.

Type String

Applicable Parts

All parts except the *component part

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ctlname = %getatr('win01': 'MLE1': 'PartName')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'PARTNAME'    ctlname          10
 *
```

PartType

Returns the type of the part. The value returned for a VARPG part is prefixed with the characters FVDES. For example, the value returned for an entry field part is *FVDESEntryField*.

Note: For a component reference part, the value returned is prefixed with the characters FVDESV.

Type String

Applicable Parts

All parts except the *component part

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Type = %getatr('win01': 'MLE1': 'PartType')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'PARTTYPE'    Type            50
 *
```

Paste

Pastes the text from the clipboard at the cursor position. If text is selected, the text from the clipboard will replace the selected text.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'Paste') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    1             'paste'
 *
```

PBRange

Determines the upper range of the progress bar. The default value is 100. This value can represent the upper limit of the process your program is monitoring. For example, if your program is uploading 1,000 files, you may want to set the upper range to 1,000 for this process.

Type Numeric

Applicable Parts

Progress Bar Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
 * Set the range for a progress bar attached to a window
C                   EVAL      %setatr('win01':'WIN1':'PBRange')=1000
 * Set the range for a progress bar part
C                   EVAL      %setatr('win01':'PBar1':'PBRange')=1000
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
 * Set the range for a progress bar attached to a window
C     'WIN1'        SETATR    1000          'PBRANGE'
 * Set the range for a progress bar part
C     'PBAR1'       SETATR    1000          'PBRANGE'
```

PBSetPos

Moves the progress bar indicator to the specified absolute position. This value depends on how the **PBRange** attribute value was set.

Type Numeric

Applicable Parts

Progress Bar Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'PBSetPos')=50
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    50            'PBSETPOS'
 *
```

PBStep

Set this attribute to 1 to move the progress bar indicator by the number of step units defined in the **PBStepSize** value.

Type Numeric

Applicable Parts

Progress Bar Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'PBStep')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    1             'PBSTEP'
 *
```

PBStepSize

Determines how far the progress bar indicator moves when the **PBStep** attribute is set on. The default value is 10.

Type Numeric

Applicable Parts

Progress Bar Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'PBStepSize')=10
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    10            'PBSTEPSIZE'
 *
```

Platform

Returns the platform name that the application is running on.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

One of the following:

WINDOWS
Windows

JAVA Java

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      app1=%getatr('*Component':'*Component':'Platform')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  GETATR    'PLATFORM'    APP1
 *
```

PlugCmd

*** Restrictions:**

1. This attribute is unsupported in Java applications.
2. This attribute applies only to plugins written in VisualAge RPG.

Passes a command to the Plugin component of the GUI Designer. The **PlugDLL** and **PlugID** attributes must be set before using **PlugCmd**.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The plugin command

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component': '*component': 'PlugCmd')
C                             = cmdstring
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    cmdstring     'PLUGCMD'
 *
```

PlugDLL

*** Restrictions:**

1. This attribute is unsupported in Java applications.
2. This attribute applies only to plugins written in VisualAge RPG.

Sets the the plugin DLL utility name. The **PlugDLL** and **PlugID** attributes must be set before using **PlugCmd**.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The plug-in DLL utility name

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component': '*component': 'PlugDLL')
C                             = PlugDLL
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    DLLname       'PLUGDLL'
 *
```

PlugID

*** Restrictions:**

1. This attribute is unsupported in Java applications.
2. This attribute applies only to plugins written in VisualAge RPG.

Sets the ID of the GUI designer that is passed to the plugin. The **PlugDLL** and **PlugID** attributes must be set before using **PlugCmd**.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The ID of the GUI designer

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL     %setatr('*component': '*component': 'PlugID')
C                            = idname
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    idname        'PLUGID'
 *
```

PlugRC

*** Restrictions:**

1. This attribute is unsupported in Java applications.
2. This attribute applies only to plugins written in VisualAge RPG.

Gets the return code from the last plugin command.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      code = %getatr('*component': '*component':
C                             'PlugRC')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'PLUGRC'      code              2 0
 *
```

PlugResult

*** Restrictions:**

1. This attribute is unsupported in Java applications.
2. This attribute applies only to plugins written in VisualAge RPG.

Gets the result string from the last plugin command.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Result = %getatr('*component': '*component':
C                             'PlugResult')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'PLUGRESULT'  Result
 *
```

Poke

Sends data to the server application. To poke data to a server application, a DDE conversation must have been successfully started for the DDE part. Also, the **DDEItem** attribute must be set to indicate which server item will receive the poked data.

When the server application processes the poke request, it will respond with a **PokeAck** event. You should check the **%PokeAck** attribute in your PokeAck action subroutine to determine if the poke completed successfully or not.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'DDEItem') = 'A1'
C                   EVAL      %setatr('win01': 'DDE1': 'Poke') = '456.78'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    '123'         'POKE'
 *
```

Position

The value returned for this attribute depends on the type of the part.

For a media part, this attribute will return the current position in the audio file. This value will be in milliseconds. If the audio file has not yet been processed, a value of zero will be returned.

For a media panel part, this attribute will return the position of the position slider. This value will be between 0 and 100.

For a horizontal or vertical scroll bar part, this attribute will return the position of the scroll box based on the **Range** attribute value.

Type Numeric

Applicable Parts

Horizontal Scroll Bar | Media | Media Panel | Vertical Scroll Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

For a media panel part, the **Position** value, between 0 and 100, represents a percentage. For a horizontal or vertical scroll bar part, this is an absolute value based on the **Range** value.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldpos = %getatr('win01': 'MMP1': 'Position')
 *
C                   EVAL      %setatr('win01': 'MMP1': 'Position') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MMP1'        GETATR    'POSITION'    OldPos            3 0
 *
C     'MMP1'        SETATR    5             'POSITION'
 *
```

PrevLine

Setting this attribute to 1 moves the scroll bar slider to a position that represents the previous line in a list of items. The slider moves relative to the **Range** established for the scroll bar part.

For example, you may have a subfile part and want to show where a user is within a set of records when a command or cursor movement key is pressed. The scroll bar part attached to a subfile only shows where the user is based on the number of records currently in the subfile. By capturing the **VKeyPress** event, you can set the appropriate slider attribute. When the slider moves to its new position, a **Scroll** event is signalled and the **%Position** event attribute is updated.

The **PageSize** attribute determines the visible page size for the items displayed. The **Range** attribute determines the upper limit for the number of items to be scrolled through.

Type Numeric

Applicable Parts

Horizontal Scroll Bar
Vertical Scroll Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'VSC1':'PrevLine')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'VSC1'        SETATR    1             'PREVLINE'
 *
```

PrevPage

Setting this attribute to 1 moves the scroll bar slider to a position that represents the previous page in a list of items. The slider moves relative to the **Range** established for the scroll bar part.

For example, you may have a subfile part and want to show where a user is within a set of records when a command or cursor movement key is pressed. The scroll bar part attached to a subfile only shows where the user is based on the number of records currently in the subfile. By capturing the **VKeyPress** event, you can set the appropriate slider attribute. When the slider moves to its new position, a **Scroll** event is signalled and the **%Position** event attribute is updated.

The **PageSize** attribute determines the visible page size for the items displayed. The **Range** attribute determines the upper limit for the number of items to be scrolled through.

Type Numeric

Applicable Parts

Horizontal Scroll Bar

Vertical Scroll Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'VSC1':'PrevPage')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'VSC1'        SETATR    1             'PREVPAGE'
 *
```

Print

Allows the window's client area or the image to be printed. The string value represents the print job's name and title. Setting this attribute displays the Windows print dialog so that the user can choose the printer.

Type String

Applicable Parts

Image	Window	Window with Canvas

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ima2':'Print')='Image2'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN01'       SETATR    'IMAGE2'      'PRINT'
 *
```

PrintAsIs

Prints the image and maintains its aspect ratio. The string value represents the print job's name and title. Setting this attribute displays the Windows print dialog so that the user can choose the printer.

Type String

Applicable Parts

Image	Window	Window with Canvas

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ima2':'PrintAsIs')='Image2'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN01'       SETATR    'IMAGE2'      'PRINTASIS'
 *
```

Printer

*** Restriction:** This attribute is unsupported in Java applications.

Returns the printer name selected in the printer dialog. The Windows system printer dialog is displayed by setting the **SelPrinter** attribute.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      prtname = %getatr('*component': '*component':
C                             'Printer')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'PRINTER'     prtname
 *
```

ProgresBar

Adds or removes a simple progress bar to or from a window as an extension. If there is a status window present, the progress bar will be moved to the opposite location on the window. Use the progress bar part if you need a more varied progress bar.

Type Numeric

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

ProgresBar can be one of the following:

1 Create progress bar

0 Remove progress bar

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'ProgresBar')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    1          'PROGRESBAR'
 *
```

Range

Determines the upper range limit for the items to be scrolled through. This value can represent the number of items displayed in a list. For example, if your program is displaying a list of 100 files, you may want to set the upper range to 100 for this purpose.

The **PageSize** attribute determines the visible page size for the items displayed.

Type Numeric

Applicable Parts

Horizontal Scroll Bar
Vertical Scroll Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Range can be between 1 and 32,767.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'HScroll1':'Range')=100
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'HSCROLL1'    SETATR    100           'RANGE'
 *
```

ReadOnly

Determines if the entry field portion of a part is read only. When set to read only, typing into the entry field is not allowed.

When this attribute is applied to the canvas part, all entry fields on the canvas will inherit the **ReadOnly** attribute setting.

Type Numeric

Applicable Parts

Canvas	Combination Box	Container	Entry Field
Multiline Edit	Spin Button		

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer, except for a Container part. For a Combination Box part, do this by selecting a **Drop-down list** type for the Combination Box style. For other parts, do this by selecting the **Read only** style.

Allowed Value

The **ReadOnly** status of field can be:

0 The entry field is not read only

1 The entry field is read only

Free form example

Toggle the read-only status of an entry field:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'EF1': 'ReadOnly')
 *
C     rc            IFEQ      1
C                   EVAL      %setatr('win01': 'EF1': 'ReadOnly') = 0
C                   ELSE
C                   EVAL      %setatr('win01': 'EF1': 'ReadOnly') = 1
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'READONLY'    rc                1 0
 *
C     rc            IFEQ      1
C     'EF1'         SETATR    0             'READONLY'
C                   ELSE
C     'EF1'         SETATR    1             'READONLY'
C                   ENDIF
 *
```

RecordID

Establishes the identification of a container record that is referenced by other attributes. For example, before using the **GetRcdFld** attribute, you must first set the **RecordID** attribute.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The ID of the container record

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'RecordID') = id
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    id            'RECORDID'
 *
```

RefAttr

The attribute of the part being referenced in a component reference part.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'COMP1': 'RefAttr') = 'Text'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'COMP1'       SETATR    'Text'        'REFATTR'
 *
```

RefParent

The name of the parent window of the target part.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'COMP1': 'RefParent')
C                             = 'WIN01'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'COMP1'       SETATR    'WIN01'       'REFPARENT'
 *
```

RefPart

The name of the part in the target component being referenced.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('window': 'COMP1': 'RefPart') = 'EF01'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'COMP1'       SETATR    'EF01'        'REFPART'
 *
```

Refresh

Causes the part to be repainted or redrawn.

Type Numeric

Applicable Parts

ActiveX	Calendar	Canvas	Check Box
Combination Box	Container	Entry Field	Graph
Graphic Push Button	Group Box	Image	List Box
Multiline Edit	Notebook	Notebook Page	ODBC/JDBC Interface
Outline Box	Push Button	Radio Button	Slider
Spin Button	Static Text	Window	

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The **Refresh** attribute can only be set to 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'Refresh') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    1             'REFRESH'
 *
```

RemoveItem

Removes an item from a list. The *index* parameter determines which item to remove. To remove all items from a list, specify 0 as the index value.

For a subfile part, this attribute applies to a subfile entry field that has values defined as its validation type. Use this attribute to remove values from the subfile entry field's validation list. First, set the **ColNumber** attribute to identify which column to work with.

For a spin button part, you can only remove all items; that is, the index value must be set to 0 (remove all). Removing a specific item from a spin button is not allowed.

Type Numeric

Applicable Parts

Combination Box List Box Spin Button Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The index value of the item to remove

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'RemoveItem') = 3
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    3             'REMOVEITEM'
 *
```

RemoveLink

*** Restriction:** This attribute is unsupported in Java applications.

Removes the link between this part and the part indicated by the *part* parameter. If the part is not currently linked, no action is taken.

Type String

Applicable Parts

CheckBox	Entry Field	Image	List Box
Media	Media Panel	Slider	Timer

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The name of the part to be removed

It must be in the format:

```
'window|part'
```

where *window* is the window on which the part is placed, and *part* is the part name.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'EF1':'RemoveLink')='WIN1|EF1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         SETATR    'WIN1|EF1'    'REMOVELINK'
 *
```

RemoveMsg

Removes a message from the message subfile part. The message to be removed is determined by the *index* parameter. To remove all messages, specify 0 as the index value.

Type Numeric

Applicable Parts

Message Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The index number of the message to be removed

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MSG1': 'RemoveMsg') = 3
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MSG1'        SETATR    3             'REMOVEMSG'
 *
```

RemoveRcd

Removes a record from a container. To remove all records, specify a value of 0.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The ID of the record to be removed

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'RemoveRcd') = id
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    id            'REMOVERCD'
 *
```

Request

Requests the current value of an item in a server application. Before the request is made, the **AppName** and **Topic** attributes must have been set and the DDE conversation successfully started.

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'Request') = 'A1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'A1'          'REQUEST'
 *
```

ReturnVal

Returns the value of a method call.

Type String

Applicable Parts

ActiveX

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ocxlist':'method')='ReturnPtr'
C                   EVAL      ptrstr=%getatr('win01':'ocxlist':'ReturnVal')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'OCXLIST'     SETATR    'METHOD'      'RETURNPTR'
C     'OCXLIST'     GETATR    'RETURNVAL'   'PTRSTR'
 *
```

RmvSrcEvt

Removes a source event from the list of events being monitored by the component reference part. This attribute, together with **AddSrcEvt**, allows one component reference part to monitor multiple events. To add new source events, use the **AddSrcEvt** attribute.

To identify the source event being removed, specify the component, window, part, and event names altogether. The format of the string is:

```
'COMPONENTNAME | WINDOWNAME | PARTNAME | EVENTNAME'
```

As the compiler does no validation checking, you must identify each source event accurately.

Type String

Applicable Parts

Component
Reference

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
DName++++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++
D Event           S             100
*
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Event='compb|win02|ef02|change'
C                   EVAL      %setatr('win01': 'CRP1':
C                             'RmvSrcEvt')=Event
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
DName++++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++
D Event           S             100
*
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Event='compb|win02|ef02|change'
C     'CRP1'        SETATR    Event         'RMVSRCEVT'
 *
```

RmvEvent

Removes an event name from the event filter list. You can use the special value *ALL to have the ActiveX or Java Bean part stop routing any events stored in the list to the VARPG program.

Type String

Applicable Parts

ActiveX Java Bean

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'RmvEvent')='EVT2'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    'EVT2'        'RMVEVENT'
 *
```

RowBGClr

A numeric value that determines the background color of a subfile row. You must use the **Index** attribute to identify the row that will have its color set.

If the background color has been set using the **RowBGMix** attribute, the background color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default (not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      RBcolor = %getatr('win01': 'SFL1': 'RowBGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'Index') = 1
C                   EVAL      %setatr('win01': 'SFL1': 'RowBGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'ROWBGCLR'   'RBcolor'         2 0
 *
C     'SFL1'        SETATR    1             'INDEX'
C     'SFL1'        SETATR    5             'ROWBGCLR'
 *
```

RowBGMix

Queries or sets the background color mix of a subfile row. You must use the **Index** attribute to identify the row that will have its color mix queried or set.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The background color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      RBclrMx = %getatr('win01': 'SFL1': 'RowBGMix')
 *
C     RBclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01': 'SFL1': 'RowBGMix') = '10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'ROWBGMIX'
 *
```

RowFGClr

A numeric value that determines the foreground color of a subfile row. You must use the **Index** attribute to identify the row that will have its color set.

If the foreground color has been set using the **RowFGMix** attribute, the foreground color returned is the one that most closely matches the mix color.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The color must be one of the following:

-1 Default (not applicable for Java applications)

0 White

1 Black

2 Blue

3 Red

4 Pink

5 Green

6 Cyan

7 Yellow

8 DarkGray

9 DarkBlue

10 DarkRed

11 DarkPink

12 DarkGreen

13 DarkCyan

14 Brown

15 PaleGray

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      RFcolor = %getatr('win01': 'SFL1': 'RowFGClr')
 *
C                   EVAL      %setatr('win01': 'SFL1': 'Index') = 1
C                   EVAL      %setatr('win01': 'SFL1': 'RowFGClr') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        GETATR    'ROWFGCLR'   'RFcolor'         2 0
 *
C     'SFL1'        SETATR    1             'INDEX'
C     'SFL1'        SETATR    5             'ROWFGCLR'
 *
```

RowFGMix

Queries or sets the foreground color mix of a subfile row. You must use the **Index** attribute to identify the row that will have its color mix queried or set.

Type String

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The foreground color mix must be in the following format:

```
'red:green:blue'
```

where red, green, and blue are values between 0 and 255, and represent the intensity of the respective color.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      RFclrMx = %getatr('win01': 'SFL1': 'RowFGMix')
 *
C     RFclrMx       IFNE      '10:23:200'
C                   EVAL      %setatr('win01': 'SFL1': 'RowFGMix') = '10:23:200'
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+... .5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    '10:23:200'   'ROWFGMIX'
 *
```

Rows

*** Restriction:** This attribute is unsupported in Java applications.

Returns the number of rows in a data source.

Note: Use this attribute only if your ODBC driver is at Version 3, or later. Earlier drivers will return unpredictable results.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Nrws=%getatr('win01':'ODBC1':'Rows')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'ROWS'      NRWS
 *
```

SBIndex

Establishes the index value for a status bar pane. This value determines which pane is the active pane. If there is a mismatch between the number of panes and the **SBIndex** value, this value defaults to 1.

Type Numeric

Applicable Parts

Status Bar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The **SBIndex** value can be from 1 to 5. The default is 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'SBIndex')=3
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    3             'SBINDEX'
 *
```

SBLabel

The text string that appears on a status bar or a status bar pane. For a status bar part, the **SBIndex** value determines which pane receives the string. The text string can also be a message ID from the message file.

Type String

Applicable Parts

Status Bar Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'SBLabel')='Wait...'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    'Wait...'     'SBLABEL'
 *
```

SBPanes

Retrieves the number of active panes for a status bar part. Use the **SBIndex** attribute to set which pane is active.

Type Numeric

Applicable Parts

Status Bar

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      NPanes=%getatr('win01':'WIN1':'SBPanes')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        GETATR    'SBPANES'     NPANES
 *
```

SBPosition

Detemines the position of a status bar. The default position is on the bottom of the window. If a progress bar is on the bottom, it is moved to the top.

Type Numeric

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

SBPosition can be one of the following:

1 Place status bar on the top

2 Place status bar on the bottom

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'SBPosition')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    1             'SBPOSITION'
 *
```

SBStyle

Detemines how the status bar is drawn. It can be depressed (the default), raised, or flat.

Type Numeric

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

SBStyle can be one of the following:

1 Depressed (the default)

2 Raised

3 Flat

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'SBStyle')=2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    2             'SBSTYLE'
 *
```

Scale

Allows the programmer to adjust the width of a column by a percentage of the width of the character ″W″.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFI1': 'Scale') = 8
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFI1'        SETATR    8             'SCALE'
 *
```

* **Restriction:** This attribute is unsupported in Java applications.

Searches for the specified string in the list. The search begins with the item specified by the Index attribute. The Index value is returned if the item is found, or 0 if the item was not found.

You can control how the search is performed by using the **Case** and **SearchType** attributes.

Type String

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CB1': 'Search') = 'Str1'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    'STR1'        'SEARCH'
 *
```

SearchType

*** Restriction:** This attribute is unsupported in Java applications.

Determines how the comparison is done for the search string when using the **Search** attribute.

Type Numeric

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

SearchType can be one of the following:

1 Compare the item prefix.

2 Compare the entire item.

3 Exact match (default)

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CB1': 'SearchType') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    1             'SEARCHTYPE'
 *
```

Selected

Indicates whether a specific item in the list portion of a part is selected. The item to be checked is specified by first setting the **Index** or **RecordID** attribute for the applicable part.

Type Numeric

Applicable Parts

Combination Box Container List Box Message Subfile Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

The **Selected** status can be:

0 The item is not selected

1 The item is selected

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Sel = %getatr('win01': 'LB1': 'Selected')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'SELECTED'    Sel               1 0
 *
```

SelectIdx

Returns an index value that indicates the selected radio button's position in the Tabs and Groups window. Radio buttons are created in logical groups, and selecting one button will cause all other buttons to be deselected. If no radio button has been selected, a value of 0 or a number greater than the number of radio buttons in the group is returned.

You can use the name of any radio button within the group as the part name.

Type Numeric

Applicable Parts

Radio Button

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      sel = %getatr('win01': 'RB1': 'SelectIdx')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'RB1'         GETATR    'SELECTIDX'   sel               4 0
 *
```

SelectItem

Selects an item in the list portion of a part. The item to select is specified by the *index* parameter. To select all items in the list, set the **Index** attribute to zero. However, for a list box part, **SelectItem** has no effect when the **Index** attribute is zero.

If the list only allows single selection, then selecting an item will deselect any currently selected items.

Note: Selecting an item will cause a **Select** event to be generated.

Type Numeric

Applicable Parts

Combination Box List Box Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The index value of the item

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'SelectItem') = 3
 *
```

Fixed form example

Select all entries in a list box if requested:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB01'        GETATR    'SELECTED'    IsChecked         1 0
C     IsChecked     IFEQ      1
C     'LB1'         SETATR    0             'SELECTITEM'
C                   ENDIF
 *
```

SelectList

Returns a list of indices of selected records in a List Box or Subfile part. The list is returned as a character string consisting of a series of 6 position values. Each value represents the index of the selected record.

Type String

Applicable Parts

List Box Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      List=%getatr('Win01':'LB1':'SelectList')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'SELECTLIST'  LIST
 *
```

SelectList Example

The following code segment shows one way to process a series of selected records in a list box using the **SelectList** attribute. This example assumes that a maximum of 10 records may be selected.

```
 *
D                 DS
D theList                        60A
D T                               6  0  Dim(10) Overlay(theList)
 *
C      'ListBox' Getatr 'NbrOfSel' N    2 0
 *
C            If  N > 0
C      'ListBox' Getatr 'SelectList' theList
 *
C           Do  N      I       2 0
C      'ListBox' Setatr T(I)   'Index'
C      'ListBox' Getatr 'GetItem'   theItem 60
 * ....
 * Process the item
 * ....
C           EndDo
 *
C           EndIf
```

SelectRcd

Selects a record in a container.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The record ID of the record to be selected

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'SelectRcd') = id
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    id            'SELECTRCD'
 *
```

SelPrinter

*** Restriction:** This attribute is unsupported in Java applications.

Causes the Windows system printer selection dialog to be shown.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component': '*component':
C                             'SelPrinter') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    1             'SELPRINTER'
 *
```

Sequence

*** Restriction:** This attribute is unsupported in Java applications.

Determines the order in which items are added to the list portion of a part when the **InsertItem** or **SetItem** attribute is set.

Changing the **Sequence** attribute will not affect the current order of the list. The order is changed when the next **InsertItem** or **SetItem** attribute is set.

Type Numeric

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The sequence for items added to a list can be:

1 Items are inserted in ascending order

2 Items are inserted in descending order

3 Items are inserted at the position specified by the **Index** attribute

4 Items are inserted at the end of the list

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'Sequence') = 1
 *
C                   EVAL      rc = %getatr('win01': 'LB1': 'Sequence')
 *
```

Fixed form example

Change the sequence of a list box:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'SEQUENCE'    sequence          1 0
 *
C     sequence      IFEQ      1
C     'LB1'         SETATR    2             'SEQUENCE'
 *
C                   ELSE
C     sequence      IFEQ      2
C     'LB1'         SETATR    3             'SEQUENCE'
 *
C                   ELSE
C     'LB1'         SETATR    2             'SEQUENCE'
C                   ENDIF
 *
C                   ENDIF
 *
```

SetItem

Replaces an existing item in the list. The item to be replaced is determined by the current **Index** attribute value for the part.

To insert a new item in a list, use the **InsertItem** attribute.

If the **Sequence** attribute specifies a sort order (that is, ascending or descending), the position of the updated item may be different than that indicated by the **Index** attribute.

Type String

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The new item text

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'Index') = 5
C                   EVAL      %setatr('win01': 'LB1': 'SetItem') = 'new'
 *
```

Fixed form example

Change the current value of a list box item:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         SETATR    5             'INDEX'
C     'LB1'         SETATR    'New'         'SetItem'
 *
```

SetRcdFld

Sets the data for a field in a container record. Before this attribute can be used, the **RecordID** attribute must be set to indicate the container record and the **ColNumber** attribute must be set to indicate which field number is to be updated in that record.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The data to set in the container field

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'SetRcdFld') = 'Test'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    'Test'        'SetRcdFld'
 *
```

SetRcdIcon

Sets the filename for the icon used to represent this record in the tree view or icon view. In Java applications, this attribute applies to all views. The **RecordID** attribute must be set to indicate which record is being used.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The icon file name

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'CN1':'SetRcdIcon')
C                             ='C:\TEST.ICO'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    'C:\TEST.ICO' 'SetRcdIcon'
 *
```

SetRcdText

Sets the text to be shown with the icon when the container is displayed in the tree view. The **RecordID** attribute must be set to indicate which record is being used.

Type String

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

Record text

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'SetRcdText') = text
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    text          'SETRCDTEXT'
 *
```

SetTop

Sets the list within the part such that the specified item is scrolled to the top of the list.

If the list box has no scroll bars (that is, all items in the list are currently being displayed), the list is not scrolled.

Type Numeric

Applicable Parts

Combination Box* Container List Box Subfile

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The index of the item to be moved to the top of the list

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'LB1': 'SetTop') = 5
 *
```

Fixed form example

Scroll the first selected entry to the top of the list:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'LB1'         GETATR    'FIRSTSEL'    index             2 0
 *
C     index         IFGT      *ZERO
C     'LB1'         SETATR    index         'SETTOP'
C                   ENDIF
 *
```

SflNxtChg

Indicates whether or not a subfile record has changed. You must set the **Index** attribute to indicate which record is being affected.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

Set to 1 to indicate the record has changed.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('window': 'SFX': 'Index') = 12
C                   EVAL      %setatr('window': 'SFX': 'SflNxtChg') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFX'         SETATR    12            'INDEX'
C     'SFX'         SETATR    1             'SFLNXTCHG'
 *
```

ShData

Specifies the user data to be read out from or written into the shared data space established.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'ShData')='DTA'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    'DTA'         'SHDATA'
 *
```

ShDataLen

Specifies the size of the data to use from the shared data. The default is 1024 bytes.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'ShDataLen')=100
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    100         'SHDATALEN'
 *
```

ShDataName

Specifies the name of the global data that can be shared across applications. Each shared data size is fixed at 1024 bytes.

When you set **ShDataName**, the system checks if the named, shared data space exists. If it does not, it will be created.

Note: Only one instance of named, shared memory can be used. For inter process communication, you can use shared memory only in an application, not in a component. For components, inter process communication is done best using parameters.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'ShDataName')='DTAN'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    'DTAN'        'SHDATANAME'
 *
```

ShDataPos

Specifies the position within the shared data to be referenced. The default starting position is 1. Together, the **ShDataLen** and **ShDataPos** value cannot exceed 1024.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'ShDataPos')=100
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  SETATR    100           'SHDATAPOS'
 *
```

ShowMsgID

Enables or disables the display of the message IDs (MSGnnnn) in the message box when the DSPLY operation code is specified.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

ShowMsgID can be one of the following:

0 Do not display message IDs.

1 Display message IDs. The default.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('*component':'*component':'ShowMsgID')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*COMPONENT'  SETATR    1             'SHOWMSGID'
 *
```

ShowProp

Set this attribute to 1 to have the part show the object's property-page dialog.

Type Numeric

Applicable Parts

ActiveX Java Bean

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'AX1':'ShowProp=') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AX1'         SETATR    1             'SHOWPROP'
 *
```

ShowRects

Indicates whether or not user-defined highlighted rectangles will be shown.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

ShowRects can be one of the following:

0 The rectangles will not be shown

1 The rectangles will be shown

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isshown = %getatr('win01': 'CAL1': 'ShowRects')
 *
C                   EVAL      %setatr('win01': 'CAL1': 'ShowRects') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'       GETATR    'SHOWRECTS'    isshown           1 0
 *
```

ShowTabs

*** Restriction:** This attribute is unsupported in Java applications.

Setting this attribute to 0 hides all tabs in a notebook part.

Type Numeric

Applicable Parts

Notebook

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'WIN1': 'ShowTabs') = 0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    0             'SHOWTABS'
 *
```

ShowText

Indicates whether or not user defined text will be shown.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

ShowText can be one of the following:

0 The text will not be shown

1 The text will be shown

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isshown = %getatr('win01': 'CAL1': 'ShowText')
 *
C                   EVAL      %setatr('win01': 'CAL1': 'ShowText') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'SHOWTEXT'    shown             1 0
 *
```

ShowTips

Determines whether or not the tooltip should be displayed for a specific part, or for all parts in the window. If you set **ShowTips** for a window, this value takes precedence over any value set for a specific part in the window. For example, if you disable showing **TipText** for the window, this setting applies to all parts in the window, as well.

Type Numeric

Applicable Parts

Check Box	Combination Box	Container	Entry Field
Graphic Push Button	Image	List Box	Message Subfile
Multiline Edit	Push Button	Radio Button	Slider
Spin Button	Static Text	Subfile	Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

ShowTips can be one of the following:

0 No tool tip will be displayed for a specific part, or for all parts in the window.

1 A tool tip will be displayed for those parts in the window that have **TipText** defined.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isshown = %getatr('win01': 'Win01': 'ShowTips')
 *
C                   EVAL      %setatr('win01': 'Win01': 'ShowTips') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'Win01'       GETATR    'SHOWTIPS'    shown             1 0
 *
```

SizeToFit

Resizes the part automatically so that the last item in the list is completely visible.

Type Numeric

Applicable Parts

Combination Box* List Box Subfile

*** Restriction:** This attribute is unsupported at run time in Java applications for this part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

SizeToFit can be one of the following:

0 Do not resize the part.

1 Resize the part automatically.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CB1': 'SizeToFit') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    1             'SIZETOFIT'
 *
```

SortAsc

Sorts the records in a container or subfile in ascending sequence. Columns with text data are sorted alphabetically.

Type Numeric

Applicable Parts

Container Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The column number to be sorted.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CTR1': 'SortAsc') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CTR1'        SETATR    2             'SORTASC'
 *
```

SortDesc

Sorts the records in a container or subfile in descending sequence. Columns with text data are sorted alphabetically.

Type Numeric

Applicable Parts

Container Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The column number to be sorted.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CTR1': 'SortDesc') = 2
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CTR1'        SETATR    2             'SORTDESC'
 *
```

SQLError

Indicates if the previous operation was successful, or not. If a value of 0 is returned, the previous operation was successful. Any other returned value indicates that an error occurred.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Err=%getatr('win01':'ODBC1':'SQLError')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       GETATR    'SQLERROR'    ERR
 *
```

SQLMsgBox

Suppresses or displays the ODBC message box dialog when errors occur processing a table.

If you set this attribute to 0 to suppress the message box, you can use the **SQLError** attribute to determine if an error occurred. You can then use the **Handle** attribute and call the ODBC API to return additional information.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

SQLMsgBox can be one of the following:

0 Do not display the message box.

1 Display the message box. This is the default.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'SQLMsgBox')=0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    0             'SQLMSGBOX'
 *
```

SQLQuery

Defines the SQL statement to run when the **ExecuteSQL** attribute is set to 1.

Type String

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Query = 'Select * From "Customers"'
C                   EVAL      %setatr('win01':'ODBC1':'SQLQuery')=Query
C                   EVAL      %setatr('win01':'ODBC1':'ExecuteSQL')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
DName++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++++++
 *
D Query           C                   'Select * From "Customers"'
 *
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC'        Setatr    Query         'SQLQuery'
C     'ODBC'        Setatr    1             'ExecuteSQL'
 *
```

StartAt

Forces the READC operation to start at the subfile record specified. This attribute is useful in programs that use READC iteratively to check for and return changed records in a subfile. See "StartAt Example" on page 434.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The record to start at

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'SFL1': 'StartAt')=4
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFL1'        SETATR    4             'STARTAT'
 *
```

StartAt Example

The following code segment uses the READC operation iteratively to check for, return, and process all changed records in a subfile. The OPT field is monitored for correct input. If the OPT value is incorrect, the program uses the **SflNxtChg** attribute to flag the incorrect record as changed. The **StartAt** attribute causes the next READC operation to begin at the record after the one in error.

By default, READC begins the search for changed records at the first subfile record. When a changed record is found and returned to the program, its status is reset to *unchanged*. In this example, the **StartAt** attribute also prevents the READC operation from looping at the record flagged by **SflNxtChg**.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 * Assume the subfile has a field called OPT which must be a value
 * greater than 0
 *
C                   ReadC     SFL1                                    99
C                   Dow       NOT *IN99
 *
C                   If        Opt > 0
 *
 * Process the record
C                   Else
 *
 * The record is invalid, so set SFLNXTCHG so we read it in the next iteration.
C     'SFL1'        Getatr    'Index'       X
C     'SFL1'        Setatr    X             'Index'
C     'SFL1'        Setatr    1             'SFLNXTCHG'
 *
 * Set STARTAT so we begin reading after the record in error.
C                   Eval      X = X + 1
C     'SFL1'        Setatr    X             'StartAt'
C                   ReadC     SFL1                                    99
C                   EndDo
 *
```

StartNew

When this attribute is set to 1 any existing data is put aside and maintained to continue drawing the graph. New data that is set will not be used in the graph until the **UseData** attribute is set.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GRA1': 'StartNew') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA'         SETATR    1             'STARTNEW'
 *
```

StatusBar

Adds or removes a simple, one-pane status bar to or from a window. Its default position is on the bottom of the window. Use the status bar part if you need a varied status bar with multiple panes and action information.

Type Numeric

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

StatusBar can be one of the following:

1 Create status bar

0 Remove status bar

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'WIN1':'StatusBar')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        SETATR    1           'STATUSBAR'
 *
```

SwitchTo

*** Restriction:** This attribute is unsupported in Java applications.

When this attribute is set, starting a second instance of a program will give the current instance of this program focus. The **SwitchTo** attribute accepts as its value, the handle of the window in the program that is to get focus. If the window is iconized, it is restored.

The handle is a 32-bit value and must be declared on a D specification as type I (unsigned) with a length of 10 and 0 decimals.

Type Numeric

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      hWnd=%getatr('Win1':'Win1':'Handle')
C                   EVAL      %setatr('*component':'*component':'SwitchTo')=hWnd
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN1'        GETATR    'HANDLE'      HWND
C     '*COMPONENT'  SETATR    HWND          'SWITCHTO'
 *
```

TabImage

Specifies the filename of the image to be displayed on a Notebook Page tab, next to the label. See the **FileName** attribute for a list of supported image types. If *BLANK or no filename is specified, the image is removed.

Type String

Applicable Parts

Notebook Page	Notebook Page with Canvas

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'NP1':'TabImage') = 'N1.ICO'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'NP1'         SETATR    'N1.ICO'      'TABIMAGE'
 *
```

TabLabel

Defines the label that appears on a notebook page tab. The **PageNumber** attribute must first be set to indicate which page is affected.

Type String

Applicable Parts

Notebook Page

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	Yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by specifying a value in the **Tab text** field.

Allowed Value

The text that appears on a notebook page tab

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'NBP1': 'TabLabel')
C                             = 'Section A'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'NBP1'        SETATR    'Section A'   'TABLABEL'
 *
```

Text

Sets or returns the text value in the entry field portion of a part.

Type String

Applicable Parts

Combination Box Entry Field Multiline Edit Spin Button

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer, except for a Multiline Edit or Spin Button part. For a Combination Box part, do this by using the **Data** tab and entering values. For an Entry Field part, do this by using the **General** tab and entering a value in the **Text** field.

Allowed Value

The value of the entry field

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldstring = %getatr('win01': 'EF1': 'Text')
 *
C                   EVAL      %setatr('win01': 'EF1': 'Text') = 'Newstring'
 *
```

Fixed form example

Save the value of an entry field, then change it:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'TEXT'        oldstring        64
C     'EF1'         SETATR    'Newstring'   'TEXT'
 *
```

TextEnd

Determines the character offset of the last selected character. If no text has been selected, a value of 0 is returned. Text is selected by positioning the mouse pointer over the text, then pressing the mouse button and moving the mouse pointer across the text.

This attribute can be used with **TextStart** to select a certain portion of text in the part, and to determine the length of the selected text.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'TextEnd') = value
 *
C                   EVAL      value = %getatr('win01': 'MLE1': 'TextEnd')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'TEXTEND'     endoffset         3 0
 *
```

TextLength

Returns the length of the text in a part.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      length = %getatr('win01': 'EF1': 'TextLength')
 *
```

Fixed form example

Get the length of the text in an entry field:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'TEXTLENGTH'  length            3 0
 *
```

TextSelect

Returns the selected text from the part. If no text has been selected, a null string is returned.

To select text in the field, set the **TextStart** and **TextEnd** attributes.

Type String

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      value = %getatr('win01': 'MLE1': 'TextSelect')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'TEXTSELECT'  seltext         100
 *
```

TextStart

Determines the first character of the text field that has been selected. If no text has been selected, a value of 0 is returned.

This attribute can be used with **TextEnd** to select a certain portion of text in the part.

Type Numeric

Applicable Parts

Entry Field Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'TextStart') = value
 *
C                   EVAL      value = %getatr('win01': 'MLE1': 'TextStart')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'TEXTSTART'   endoffset         3 0
 *
```

TextString

Returns all of the text in the multiline edit part and converts all carriage return line feed characters to blanks.

Type String

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      ATxt=%getatr('Win01':'ML1':'TextString')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ML1'         GETATR    'TEXTSTRING'  ATXT
 *
```

TickLabel

Assigns a string to a tick mark on a slider part. The **TickNumber** attribute determines which **TickLabel** is being referenced.

Type String

Applicable Parts

Slider

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by selecting the **Add** push button and entering values in the **Add Tick text** dialog box.

Allowed Value

The label assigned to the slider tick

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   Z-ADD     0             percent
 *
C                   DO        5             ticknbr
C                   EVAL      %setatr('win01':'SLIDER1':'TickNumber')=ticknbr
C                   EVAL      %setatr('win01':'SLIDER1':'TickLabel')=percent
C                   ADD       25            percent
C                   ENDDO
 *
```

Fixed form example

Change the text for a slider tick:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   Z-ADD     *ZERO         value             2 0
C     1             DO        11            tick              2 0
C     'SLIDER1'     SETATR    tick          'TICKNUMBER'
C     'SLIDER1'     SETATR    value         'TICKLABEL'
C                   ADD       10            value
C                   ENDDO
 *
```

TickNumber

Specifies at which slider tick the next **TickLabel** text will be placed.

The number will be a value ranging from 1 to the number of ticks specified for the slider in the notebook settings.

Type Numeric

Applicable Parts

Slider

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The tick number

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      x = %getatr('win01': 'SLIDER1': 'TickNumber')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SLIDER1'     GETATR    'TICKNUMBER'  OldTick           3 0
 *
C     'SLIDER1'     SETATR    NewTick       'TICKNUMBER'
 *
```

Tile

When set to 1, the image on the canvas will be tiled.

Type Numeric

Applicable Parts

Canvas

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

Tile can be one of the following:

0 The image is not tiled

1 The image is tiled

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAN': 'Tile') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAN1'        SETATR    1             'TILE'
 *
```

TimeOut

This value determines how long a DDE client part will wait to establish a DDE conversation after setting the **DDEMode** to *Initiate*. This value is expressed in seconds.

If the DDE conversation is not completed in this time, a **Timeout** event is signaled to your program.

Type Numeric

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'TimeOut') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    5             'TIMEOUT'
 *
```

TimerMode

Determines the current operating mode for a timer part. When the mode is set to *Start*, the timer will begin generating **Tick** events. Setting the mode to *Stop* ceases generating **Tick** events. When the **TimerMode** is set to *Pause*, the timer will cease generating **Tick** events but will continue to update the **TimerTicks** attribute.

Type Numeric

Applicable Parts

Timer

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

TimerMode can be one of the following:

1 Start

2 Stop

3 Pause

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     PB1           BEGACT    PRESS
C                   EVAL      %setatr('win01': 'TIMER01': 'TimerMode') = 1
C                   ...
C                   ENDACT
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'TIMER1'      SETATR    1             'TIMERMODE'
 *
```

TimerTicks

Determines the current tick value of a timer. The **TimerTicks** value is the number of times that the timer has generated a **Tick** event.

Note that the **Interval** and **Multiplier** attributes also determine the number of tick events. The **TimerTicks** attribute is determined as follows:

```
TimerTicks = (Elapsed time in milliseconds) / (Interval * Multiplier)
```

Type Numeric

Applicable Parts

Timer

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The current timer tick count

Free form example

Change the timer tick value:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      count=%getatr('win01':'TIMER01':'TimerTicks')
 *
C     count         IFGT      1000
C                   EVAL      %setatr('win01':'TIMER01':'TimerTicks')=0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'TIMER01'     GETATR    'TIMERTICKS'  count             6 0
 *
C     count         IFGT      1000
C     'TIMER01'     SETATR    0             'TIMERTICKS'
C                   ENDIF
 *
```

TipText

Specifies the text displayed in the tool tip control.

For a graph part, associate **TipText** with a data point. In a bar type, for example, the tooltip text will be displayed next to the bar respresenting the data point.

For a calendar part, associate **TipText** with the **DateIdx** attribute. The tooltip will be assocated with that date.

Type String

Applicable Parts

Calendar	Check Box	Combination Box	Container
Entry Field	Graph	Graphic Push Button	Image
List Box	Message Subfile	Multiline Edit	Push Button
Radio Button	Slider	Spin Button	Static Text
Subfile			

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	no

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

Up to 80 characters of text or a message identifier (*MSGnnn)

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'TipText')='*msg0002'
 */xmp>
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    'Hello'       'TipText'
 *
```

Title

Determines the graph's title text.

Type String

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      value = %getatr('win01': 'GRA1': 'Title')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        GETATR    'TITLE'       seltitle          1 0
 *
```

TitlePlace

Indicates where the title should be located within the graph.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

TitlePlace can be one of the following:

0 No title

1 Above the graph

2 Below the graph

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GRA1': 'TitlePlace') = 3
 *
C                   EVAL      where = %getatr('win01': 'GRA1': 'TitlePlace')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        GETATR    'TITLEPLACE'  where             1 0
 *
```

Top

Specifies the top screen coordinate, in pixels, of the part or the object.

Note: The origin is the upper left corner, and begins with zero.

Type Numeric

Applicable Parts

ActiveX	Animation Control	Calendar	Canvas*
Check Box	Combination Box	Container	DDE Client
Entry Field	Graph	Graphic Push Button	Group Box
Horizontal Scroll Bar	Image	Java Bean	List Box
Media	Media Panel	Multiline Edit	Notebook
ODBC/JDBC Interface	Outline Box	Progress Bar	Push Button
Radio Button	Slider	Spin Button	Static Text
Subfile	Timer	Vertical Scroll Bar	Window

* **Restriction:** Can only get this attribute at run time for the Canvas part.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

* **Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The number of pixels

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      otop = %getatr('win01': 'Image1': 'Top')
C     otop          SUB       200           newtop
C                   EVAL      %setatr('win01': 'Image1': 'Top') =  newtop
 *
```

Fixed form example

Move an image 200 pixels in its window.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'IMAGE1'      GETATR    'TOP'         OldTop            4 0
C     OldTop        SUB       200           NewTop            4 0
C     'IMAGE'       SETATR    NewTop        'TOP'
 *
```

Topic

The **Topic** attribute is set to inform the server application in a DDE conversation of the subject of the conversation. Before the conversation can begin, the **AppName** attribute must also be set.

A DDE topic is a set of data that the server application can process. In the case of a spreadsheet application, this would be a spreadsheet file.

If the server program is a VisualAge RPG component, the syntax of this attribute is:

```
'appname|appname'
```

where *appname* is the name of the VisualAge RPG component. For example, if the server program was a VisualAge RPG component named *server*, this attribute would be set as follows:

```
EVAL    %setatr('WIN1':'DDE1':'Topic') = 'SERVER|SERVER'
```

Type String

Applicable Parts

DDE Client

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'DDE1': 'AppName') = 'APP01'
C                   EVAL      %setatr('win01': 'DDE1': 'Topic') = 'APP01.WG1'
C                   EVAL      %setatr('win01': 'DDE1': 'DDEMode') = 1
C                   ...
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'DDE1'        SETATR    'APP01.WG1'   'TOPIC'
 *
```

TopLine

Indicates the line number to be the top of a multiline edit field. If **TopLine** is set to zero, the multiline edit will be displayed on the last page.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The line number

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'MLE1': 'TopLine') = 0
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        SETATR    0             'TOPLINE'
 *
```

TopRecord

Sets the list within the subfile part such that the specified index item is scrolled to the top of the list or gets (returns) the index of the top visible row.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      itemtop = %getatr('win01': 'SFI1': 'TopRecord')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SFI1'        GETATR    'TOPRECORD'   itemtop
 *
```

Treble

Note: This attribute applies to Windows only.

The **Treble** attribute sets the treble level for the audio file. The level can be set from 0 to 100.

Type Numeric

Applicable Parts

Media

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value must be from 0 to 100

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'AUD1': 'Treble') = 42
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'AUD1'        SETATR    42            'TREBLE'
 *
```

UnBind

*** Restriction:** This attribute applies only to Java applications.

Set this attribute to 1 to unbind previously bound parts.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'OD1':'UnBind')=1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'OD1'         SETATR    1             'UNBIND'
 *
```

UnderPoint

*** Restriction:** This attribute is unsupported in Java applications.

Set this attribute to the string representation of a postition relative to the graph part's top-left corner, for example ″25 10″. Then use the values of the **HitItem**, **DataPoint**, and **DataGroup** attributes to determine what part of the graph is located under the point given.

Type String

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Allowed Value

The string in the form ″x y″

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      xystring = %getatr('win01': 'GPH1': 'UnderPoint')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GPH1'        GETATR    'UNDERPOINT'  xystring
 *
```

Undo

Set this attribute to 1 to undo the last change that has been made.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'ML1': 'Undo') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ML1'         SETATR    1             'UNDO'
 *
```

UpdateRow

Determines the row number to update in the result set. You do not need to fetch a row first to update it. This attribute will cause any row to be updated.

To update a row that has just been fetched, use the **CurrentRow** attribute to get its row number.

Note: In Java applications, this attribute requires the appropriate JDBC 2.0 compliance driver.

Type Numeric

Applicable Parts

ODBC/JDBC
Interface

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'ODBC1':'UpdateRow')=4
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'ODBC1'       SETATR    4             'UPDATEROW'
 *
```

UseData

Set this attribute to 1 to start using the data that has been set for the graph since the last time **StartNew** was set.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

The allowed value is 1

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GRA1': 'UseData') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        SETATR    1             'USEDATA'
 *
```

UseDelim

Set this attribute to 1 to identify data that has a delimiter character associated with it. Use the **DelimChar** attribute to specify the delimiter character, if any.

Type Numeric

Applicable Parts

Combination Box List Box

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CB1': 'UseDelim') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CB1'         SETATR    1             'USEDELIM'
 *
```

UserData

Allows any user-defined string to be associated with a part. There is no length restriction on the string.

Type String

Applicable Parts

All parts except the *component part

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The data to associate with the part

Free form example

Save the value of an entry field, then change it:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldstring = %getatr('win01': 'EF1': 'UserData')
 *
C                   EVAL      %setatr('win01': 'EF1': 'UserData') = 'Newstring'
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'USERDATA'    oldstring        64
C     'EF1'         SETATR    'Newstring'   'USERDATA'
 *
```

Validate

When this attribute is set, pressing the button it applies to will cause field validation to occur for those parts that have field validation settings and are in the same window as the button. Field validation includes such things as range- and values-checking.

If any errors are found during field validation, a message is displayed and the **Press** event is not signaled.

If there are no errors, or if there are no fields with field validation, the press event action subroutine is invoked.

Note: If no push button or graphic push button on the window has the **Validate** attribute set, field validation will not occur.

Type Numeric

Applicable Parts

Graphic Push Button
Push Button

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer. Do this by selecting the **Perform field validation** style.

Allowed Value

Validate can be one of the following:

0 Do not perform entry validation

1 Perform entry validation

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      validate = %getatr('win01': 'EF01': 'Validate')
 *
C                   EVAL      %setatr('win01': 'EF01': 'Validate') = 1
 *
```

Fixed form example

Save the value of an entry field, then change it:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'VALIDATE'    validate          1 0
C     'EF1'         SETATR    1             'VALIDATE'
 *
```

Value

Returns the value of a slider part. The value will be equal to or greater than the minimum value assigned to the slider, and less than or equal to the maximum value.

For a spin button, the current contents of the entry field are returned.

Type Numeric

Applicable Parts

Slider Spin Button

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The value of the slider or entry field

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      oldvalue = %getatr('win01': 'SLIDER1': 'Value')
 *
C                   EVAL      %setatr('win01': 'SLIDER1': 'Value') = 5
 *
```

Fixed form example

Save the value of an entry field, then change it:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SLIDER1'     GETATR    'VALUE'       OldValue          3 0
C     'SLIDER1'     SETATR    5             'VALUE'
 *
```

VColFRDCol

Returns the view column for the corresponding data column specified by the **ColNumber** attribute.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'SF1':'ColNumber') = 4
C                   EVAL      ViewCol4 = %getatr('win01':'SF1':'VColFRDCol')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    4             'COLNUMBER'
C     'SF1'         GETATR    'VCOLFRDCOL'  VIEWCOL4
 *
```

View

Determines how the records in a container are displayed. A container can display records in an icon, tree, details, or text tree view.

Type Numeric

Applicable Parts

Container

*** Restriction:** In Java applications, can only get this attribute at run time for this part.

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The container view can be one of the following:

1 Icon view

2 Tree view

3 Details view

4 Text tree view

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'View') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'VIEW'
 *
```

ViewColumn

Indicates the order in which a subfile data column is displayed. This attribute is used with the **ColNumber** (the data column) and **MapViewCol** attributes to change the order in which data columns are displayed. The default is to display data columns in the same order as that listed in the Field List page of the Subfile properties notebook.

Type Numeric

Applicable Parts

Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

Display data column 2 (ColNumber) as viewable column 4 (ViewColumn). For example, if data columns are defined as follows:

```
Data Column
1        2      3         4
Custno   Name   Address   City
```

The columns view after this operation becomes:

```
View after:
1        2        3      4
Custno   Address  City   Name
```

Note that all other columns are shifted leftwards in the view.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'SF1':'ColNumber') = 2
C                   EVAL      %setatr('win01':'SF1':'ViewColumn') = 4
C                   EVAL      %setatr('win01':'SF1':'MapViewCol') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         SETATR    2             'COLNUMBER'
C     'SF1'         SETATR    4             'VIEWCOLUMN'
C     'SF1'         SETATR    1             'MAPVIEWCOL'
 *
```

Visible

Determines if the part is visible or not.

Type Numeric

Applicable Parts

ActiveX	Animation Control	Calendar	Check Box
Combination Box	Component Reference	Container	DDE Client
Entry Field	Graph	Graphic Push Button	Group Box
Horizontal Scroll Bar	Image	Java Bean	List Box
Media	Media Panel	Menu Item	Message Subfile
Multiline Edit	Notebook	Notebook Page	ODBC/JDBC Interface
Outline Box	Pop-up Menu*	Progress Bar	Push Button
Radio Button	Slider	Spin Button	Static Text
Status Bar	Subfile	Timer	Vertical Scroll Bar
Window			

***Restriction:** For Windows applications, a query (GETATR) returns a value of 0 for this attribute. For Java applications, this attribute returns the state of the pop-up menu.

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

* **Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

Visible can be one of the following:

0 The part is not visible

1 The part is visible

Note: For pop-up menus in Java applications, you must also set the **InvPname** (invoker's parent name) and **InvName** (invoker's name) attributes, and the x-y coordinates of the pop-up in the invoker's coordinate space.

At least one menu item must be visible on a menu bar part. If you use bit map images to represent menu items, then all menu items must be bit map images. Otherwise, refresh problems occur when the **Visible** attribute is reset.

Free form example

Toggle the visible state of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      isVisible = %getatr('win01': 'EF1': 'Visible')
 *
C     isVisible     IFEQ      1
C                   EVAL      %setatr('win01': 'EF1': 'Visible') = 0
 *
C                   ELSE
C                   EVAL      %setatr('win01': 'EF1': 'Visible') = 1
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'VISIBLE'     IsVisible         1 0
C     IsVisible     IFEQ      1
C     'EF1'         SETATR    0             'VISIBLE'
 *
C                   ELSE
C     'EF1'         SETATR    1             'VISIBLE'
C                   ENDIF
 *
```

VisTitle

Determines if the container title is visible or not.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

One of the following:

0 Title is not visible

1 Title is visible

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'VisTitle) = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'VisTitle'
 *
```

VisTitlSep

Determines if the container title separator is visible or not.

Type Numeric

Applicable Parts

Container

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Allowed Value

One of the folllowing:

0 Title separator is not visible

1 Title separator is visible

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CN1': 'VisTitlSep) = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CN1'         SETATR    1             'VisTitlSep'
 *
```

Volume

Note: This attribute applies to Windows only.

Sets the volume for the media part and the system's waveout and synthesizer.

For a media panel, it determines the setting of the volume slider. This value will range from 0 to 100.

Type Numeric

Applicable Parts

Media Media Panel

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The volume setting

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      curvol = %getatr('win01': 'MMP1': 'Volume')
 *
C                   EVAL      %setatr('win01': 'MMP1': 'Volume') = 5
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MMP1'        GETATR    'VOLUME'      curvol            3 0
 *
C     'MMP1'        SETATR    5             'VOLUME'
 *
```

VRule

Adds a vertical rule between each column of a subfile data or calendar row. There must be at least one row of data.

Type Numeric

Applicable Parts

Calendar Subfile

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

VRule can be one of the following:

0 Turns rule off

1 Turns rule on

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      Vtrule = %getatr('win01': 'SF1': 'VRule')
 *
C     Vtrule        IFEQ      0
C                   EVAL      %setatr('win01': 'SF1': 'VRule') = 1
C                   ELSE
C                   EVAL      %setatr('win01': 'SF1': 'VRule') = 0
C                   ENDIF
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'SF1'         GETATR    'VRULE'       Vtrule            1 0
 *
C     'SF1'         SETATR    1             'Vtrule'
 *
```

WeekDay

Indicates the selected day of the week.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

One of the following:

0 Sunday

1 Monday

2 Tuesday

3 Wednesday

4 Thursday

5 Friday

6 Saturday

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      index = %getatr('win01': 'CAL1': 'WeekDay')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'WEEKDAY'     02
 *
```

WeekDayIdx

Returns a day of the week that corresponds to the **YearIdx**, **MonthIdx** and **DayIdx** settings.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Returned Value

One of the following:

0 Sunday

1 Monday

2 Tuesday

3 Wednesday

4 Thursday

5 Friday

6 Saturday

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      index = %getatr('win01': 'CAL1': 'WeekDayIdx')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'WEEKDAYIDX'  INDEX
 *
```

Width

Specifies the width of the part in pixels. Changing the width of a window generates a **ReSize** event.

Note: The **Width** attribute cannot exceed the maximum width determined by the display resolution used. For example, you cannot create a part 800 pixels wide when working with a 640 pixel width display.

Type Numeric

Applicable Parts

ActiveX	Calendar	Canvas	Check Box
Combination Box	Container	Entry Field	Graph
Graphic Push Button	Group Box	Horizontal Scroll Bar	Image
Java Bean	List Box	Media Panel	Multiline Edit
Notebook	ODBC/JDBC Interface	Outline Box	Progress Bar
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile	Vertical Scroll Bar	Window

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

The width of the part in pixels

Free form example

Increase the width of a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      width = %getatr('win01': 'EF1': 'Width')
 *
C                   EVAL      %setatr('win01': 'EF1': 'Width') =
                              width + 100
 *
```

Fixed form example

Widen a push button by 200 pixels:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'EF1'         GETATR    'WIDTH'       OldWidth          2 0
C     OldTop        ADD       200           NewWidth          2 0
C     'EF1'         SETATR    NewWidth      'WIDTH'
 *
```

WindowMode

*** Restriction:** This attribute is unsupported in Java applications.

Determines how a window is displayed.

Type Numeric

Applicable Parts

Window

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

WindowMode can be one of the following values:

1 **Maximized** - Displays the window in its largest size possible

2 **Minimized** - Removes the window from the workplace and represents it as an icon on the workplace

3 **Restored** - Returns the window to the size it was before it was minimized or maximized

Note: When a window is restored from a maximized state, the WindowMode attribute may indicate that the window is still in maximized state even though it is not.

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'win01': 'WindowMode') = 1
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'WIN01'       SETATR    1             'WINDOWMODE'
 *
```

WordWrap

Determines if text entered in a multiline edit part should wrap on a word boundary. If **WordWrap** is disabled, the text will continue on the same line.

Note: This attribute does not function if the horizontal scroll style is set.

Type Numeric

Applicable Parts

Multiline Edit

Operation Codes

	Operational at Run Time
SETATR*	no
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

WordWrap can be one of the following:

0 Word wrap disabled

1 Word wrap enabled

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      rc = %getatr('win01': 'MLE1': 'WordWrap')
 *
```

Fixed form example

Toggle the wordwrap status of a multiline edit:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'MLE1'        GETATR    'WORDWRAP'    break             1 0
 *
```

WrkStnName

*** Restriction:** This attribute is unsupported in Java applications.

Returns the name of the workstation.

Type String

Applicable Parts

*Component

Operation Codes

	Operational at Run Time
SETATR	no
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      WRKN=%getatr('*component':'*component':'WrkStnName')
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     '*component'  GETATR    'WRKSTNNAME'  WRKN
 *
```

*** Restriction:** Applies only to Java applications.

Specifies the x coordinate, in pixels, of the pop-up menu in the invoker's coordinate space. This attribute is used together with the **InvName**, **InvPname**, and **Y** attributes.

Type Numeric

Applicable Parts

Pop-up Menu

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'POP1':'X')=35
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'POP1'        SETATR    35            'X'
 *
```

XAxisLabel

The text used as a label for the X-axis.

Type String

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GRA1': 'XAxisLabel') = label
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA'         SETATR    'label'       'XAxisLabel'
 *
```

*** Restriction:** Applies only to Java applications.

Specifies the y coordinate, in pixels, of the pop-up menu in the invoker's coordinate space. This attribute is used together with the **InvName**, **InvPname**, and **X** attributes.

Type Numeric

Applicable Parts

Pop-up Menu

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	no

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01':'POP1':'Y')=45
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'POP1'        SETATR    45            'Y'
 *
```

YAxisLabel

The text used as a label for the Y-axis.

Type String

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GRA1': 'YAxisLabel') = label
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        SETATR    'label'       'YAxisLable'
 *
```

Year

Indicates the currently selected year of the calendar. When set, the selected date of the calendar part will be updated to reflect the changes.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The year

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'Year') = 1998
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1998          'YEAR'
 *
```

YearIdx

Allows the user to set data on a date not displayed on the current calendar page.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Allowed Value

The index value

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'YearIdx') = 1998
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        SETATR    1998          'YEARIDX'
 *
```

YearLen

Indicates the format for displaying the year.

Type Numeric

Applicable Parts

Calendar

Operation Codes

	Operational at Run Time
SETATR*	yes
GETATR	yes

*** Note:** Can be set using the properties notebook in the GUI designer.

Allowed Value

YearLen can be one of the following:

1 Full four-digit year

2 Two digits

Note: Windows displays the two digits as the two digits padded option. No distinction is made.

3 Two digits padded with a zero for years with only one digit

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'CAL1': 'YearLen') = 3
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'CAL1'        GETATR    'YEARLEN'     YearLen           1 0
 *
```

Yinc

Sets the number of equally-spaced tick marks to use on the Y-axis. The y-axis will be divided into an equal number of increments based on the minimum and maximum values allowed.

Type Numeric

Applicable Parts

Graph

Operation Codes

	Operational at Run Time
SETATR	yes
GETATR	yes

Free form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C                   EVAL      %setatr('win01': 'GRA1': 'YInc') = 20
 *
```

Fixed form example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq----
 *
C     'GRA1'        SETATR    20            'YInc'
 *
```

Chapter 3. Part Events

When a part recognizes an event (for example, when the user selects a push button), it invokes the **action subroutine** that you defined within specific BEGACT and ENDACT operation codes in your program. If no action subroutine has been defined for an event, the event is ignored. The Action Subroutines window displays which action subroutines are invoked by which part events.

In some cases, the event will also make available **event attributes**, which contain relevant data. These event attributes are accessed in your program by referencing them.

This chapter describes the part events. Each description contains the following:

- A list of parts the event applies to
- A general description of the event
- A list of the event attributes which apply to the event (if applicable)
- An example (if applicable)

Activate

This event occurs when a window becomes the active window. A window becomes active when the user clicks on it with the mouse pointer, or selects a part on it.

Applicable Parts

Window

Event Attributes

%EventName %Part %Window

BeanEvent

*** Restriction:** This event is unsupported in Windows applications.

This event occurs whenever a Java Bean event occurs.

Applicable Parts

Java Bean

Event Attributes

%EventName %Part %Window

Change

This event occurs when the contents or the value of a part has changed. For each part type, this event occurs when the following change is made:

Combination Box
The user types in the entry field area or selects an item from the drop-down list.

Entry Field
The user types in the entry field.

Multiline Edit
The user types in the multiline edit part.

Media Panel
The user changes the position of a slider on the panel.

Slider The user changes the slider position.

Spin Button
The user selects one of the spin buttons, or types a value in the entry field.

Subfile
The user changes data in a subfile cell and leaves the cell.

Note that the value of a part can also be changed as the result of setting certain attributes, for example, the part is set up as a target part and is responding to a **Link** event.

Applicable Parts

Combination Box Entry Field Media Panel Multiline Edit
Slider Spin Button Subfile

Event Attributes

%ColNumber %EventName %Index %Part
%Window

Click

This event occurs when the user positions the mouse pointer over the part and then presses and releases one of the mouse buttons.

The part must be enabled to receive this event.

Note: The Click event is not signaled if a Container part is in Details view.

Applicable Parts

Calendar	Canvas	Container	Entry Field
Graph	Image	Multiline Edit	Static Text

Event Attributes

%Button	%EventName	%MouseX	%MouseY
%Part	%Window		

Close

This event occurs only when a window is closed using the system menu. The CLSWIN operation code has no effect.

Applicable Parts

Window

Event Attributes

%EventName	%Part	%Window

Collapsed

This event occurs when a user collapses a node in a container in the tree view. You can use this event to change the icon for a record to show its state.

Applicable Parts

Container

Event Attributes

%EventName	%Part	%Window

ColSelect

This event occurs when a user selects a column heading by clicking on it. A container part must be in the Details view. The %ColNumber event attribute returns the column number of the selected column.

Applicable Parts

Container	Subfile

Event Attributes

%ColNumber	%EventName	%Part	%Window

Complete

This event occurs when the media part has processed an audio file. The **%Complete** event attribute indicates whether or not the operation was successful.

Applicable Parts

Media

Event Attributes

%Complete	%EventName	%Part	%Window

Create

This event occurs when a part is first created.

For a window part, the **Create** event occurs only after all parts on that window have been created and their **Create** events have occurred.

It is possible to set part attributes at this time. For example, you may want to change the default value of an entry field before the field appears on the window.

Applicable Parts

ActiveX	Animation Control	Calendar	Canvas
Check Box	Combination Box	Component Reference	Container
DDE Client	Entry Field	Graph	Graphic Push Button

Group Box	Horizontal Scroll Bar	Image	Java Bean
List Box	Media	Media Panel	Menu Bar
Menu Item	Message Subfile	Multiline Edit	Notebook
Notebook Page	ODBC/JDBC Interface	OutlineBox	Progress Bar
Push Button	Radio Button	Slider	Spin Button
StaticText	Status Bar	Subfile	Submenu
Timer	Vertical Scroll Bar	Window	

Event Attributes

%EventName	%Part	%Window

Data

This event occurs when data has become available from the server application in a DDE conversation. It is generated as the result of a *Request* command, or if the data has changed or the DDE client part is set up in a hot-link conversation with another part via the **AllowLink** attribute.

Applicable Parts

DDE Client

Event Attributes

%Data	%EventName	%Item	%Part
%Window			

DblClick

This event occurs when the user presses and releases the mouse button twice in quick succession.

Applicable Parts

Calendar	Canvas	Container	Entry Field
Graph	Image	Multiline Edit	Static Text

Event Attributes

%Button	%EventName	%MouseX	%MouseY
%Part	%Window		

DeActivate

This event occurs when the window is no longer active. A window becomes inactive when another part is made active.

This event occurs when a CLSWIN operation code is executed.

Applicable Parts

Window

Event Attributes

%EventName	%Part	%Window

Destroy

This event occurs when the window that contains the part is closed, either by the user selecting the **Close** menu option from the system menu on the window, or by using the CLSWIN operation code.

This event will also occur for each part in your program when the program is terminating.

Applicable Parts

ActiveX	Animation Control	Calendar	Canvas
Check Box	Combination Box	Component Reference	Container
DDE Client	Entry Field	Graph	Graphic Push Button
Group Box	Horizontal Scroll Bar	Image	Java Bean
List Box	Media	Media Panel	Menu Bar
Menu Item	Message Subfile	Multiline Edit	Notebook
Notebook Page	ODBC/JDBC Interface	OutlineBox	Progress Bar
Push Button	Radio Button	Slider	Spin Button
StaticText	Status Bar	Subfile	Submenu
Timer	Vertical Scroll Bar	Window	

Event Attributes

%EventName	%Part	%Window

Drop

This event occurs when an object has been dragged and then dropped onto this part. The **DropEnable** attribute must be set for this part in order for it to respond to this event. You can query the value associated with the dropped part by checking the **%Data** event attribute.

Applicable Parts

Combination Box*	Entry Field	List Box*	Message Subfile
Multiline Edit	Static Text		

*** Restriction:** This event is unsupported at run time in Java applications for this part.

Event Attributes

%Data	%EventName	%Part	%Window

DropDown

*** Restriction:** This event is unsupported in Java applications.

For a combination box, this event occurs when the following occurs:

- The user clicks on the down arrow to the right of the box.

Applicable Parts

Combination Box

Event Attributes

%EventName	%Part	%Window

Enter

For a combination box, container, list box, message subfile, or subfile, this event occurs when either of the following occurs:

- The user double-clicks on the part.
- The user presses the Enter key when the part has focus and an item has been selected.

For a radio button, this event occurs when you double click on the part.

Applicable Parts

Combination Box	Container	List Box*	Message Subfile
Radio Button	Subfile		

*** Restriction:** This event is unsupported at run time in Java applications for this part.

Event Attributes

%EventName	%Part	%Window

ExecuteAck

This event occurs when the server application has processed an execute command. The **%Complete** event attribute indicates if the execute was successful or not.

Applicable Parts

DDE Client

Event Attributes

%Complete	%Data	%EventName	%Part
%Window			

Expanded

This event occurs when a user expands a node in a container in the tree view. You can use this event to change the icon for a record to show its state.

Applicable Parts

Container

Event Attributes

%EventName	%Part	%Window

FirstRec

This event occurs when the 'first record' button is selected on the navigator bar.

Applicable Parts

Subfile

Event Attributes

%EventName	%Part	%Window

GotFocus

This event occurs when the part receives input focus. This occurs when the user moves the cursor to an enabled part, or if the focus is set to this part using the **Focus** attribute. Note that the part must be enabled to receive this event.

Applicable Parts

Combination Box	Container	Entry Field	Graphic Push Button
List Box	Multiline Edit	Push Button	Slider
Spin Button	Subfile		

Event Attributes

%EventName	%Part	%Window

KeyPress

This event occurs when a key is pressed and the part has input focus. The part must be enabled to receive this event.

Note: The **ReadOnly** attribute must not be set.

This event occurs only for the character keys. To handle events generated by noncharacter keys, such as Page Up or Page Down, refer to the **VKeyPress** event.

Applicable Parts

Combination Box	Container	Entry Field	List Box
Multiline Edit	Subfile		

Event Attributes

%Alt	%Character	%Control	%EventName
%Part	%Shift	%Window	

LastRec

This event occurs when the 'last record' button is selected on the navigator bar.

Applicable Parts

Subfile

Event Attributes

%EventName	%Part	%Window

LClickTray

This event occurs when a user left clicks the application's system tray icon. The application must be registered as a system tray application using the Windows Shell_NotifyIconA API.

Applicable Parts

Window	Window with Canvas

Event Attributes

%EventName	%Part	%Window

Link

*** Restriction:** This event is unsupported in Java applications.

This event occurs if the part has been linked to a source part, using the **AddLink** attribute, and the source part has changed. The **AllowLink** attribute must be set in order for the source part to signal this event.

Applicable Parts

Entry Field	Image	Media	Media Panel
Slider	Spin Button	Static Text	Timer

Event Attributes

%EventName	%Part	%Window

LostFocus

This event occurs when the part loses input focus.

Applicable Parts

Combination Box	Container	Entry Field	Graphic Push Button
List Box	Multiline Edit	Push Button	Slider
Spin Button	Subfile		

Event Attributes

%EventName	%Part	%Window

MenuSelect

This event occurs when the user has selected a menu item by clicking the mouse or by using the accelerator key.

Note that this event does not mean that a menu item has been checked. To determine whether a menu item is checked or not, query the **Checked** attribute for the item.

Applicable Parts

Menu Item

Event Attributes

%EventName	%Part	%Window

MouseDown

This event occurs when a mouse button is pressed.

Applicable Parts

Calendar	Canvas	Container	Entry Field
Graph	Multiline Edit	Static Text	

Event Attributes

%Button	%EventName	%MouseX	%MouseY
%Part	%Window		

MouseEnter

This event occurs when a mouse pointer first moves into a part.

Applicable Parts

Calendar	Check Box	Combination Box	Container
Entry Field	Graph	Graphic Push Button	Image
List Box	Media Panel	Message Subfile	Multiline Edit
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile		

Event Attributes

%EventName	%Part	%Window

MouseExit

This event occurs when a mouse pointer first moves out of a part.

Applicable Parts

Calendar	Check Box	Combination Box	Container
Entry Field	Graph	Graphic Push Button	Image
List Box	Media Panel	Message Subfile	Multiline Edit
Push Button	Radio Button	Slider	Spin Button
Static Text	Subfile		

Event Attributes

%EventName	%Part	%Window

MouseMove

This event occurs when the mouse pointer is moved over a part. You could use this event to create an information area to inform the user of the function of the part.

Applicable Parts

Calendar	Canvas	Check Box	Combination Box
Container	Entry Field	Graph	Graphic Push Button
Image	List Box	Media Panel	Message Subfile
Multiline Edit	Push Button	Radio Button	Slider
Spin Button	Static Text	Subfile	

Event Attributes

%EventName	%MouseX	%MouseY	%Part
%Window.			

Example

Create an information area by changing the **Label** attribute of a static text part when the mouse pointer is moved over a part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
 *
C     PB1           BEGACT    MOUSEMOVE
 *
C     ST1           SETATR    'Ends program'LABEL
 *
C                   ENDACT
 *
```

MouseUp

This event occurs when a mouse button is released.

Applicable Parts

Calendar	Canvas	Container	Entry Field
Graph	Multiline Edit	Static Text	

Event Attributes

%Button	%EventName	%MouseX	%MouseY
%Part	%Window		

Moved

This event occurs whenever a user moves a window.

Applicable Parts

Window

Event Attributes

%EventName %Part %Window

MthChange

This event occurs when a user selects a new month by clicking one of the month arrows.

Applicable Parts

Calendar

Event Attributes

%EventName %Part %Window

NextRec

This event occurs when the 'next record' button is selected on the navigator bar.

Applicable Parts

Subfile

Event Attributes

%EventName %Part %Window

Notify

In the properties notebook for the component reference part, you can specify an event associated with a particular part in a specified window. When an event whose name matches the **NotSrcEvt** attribute setting occurs at the specified part in the specified window, a **Notify** event is routed to the component reference part.

For a Notify event to be signaled, an action subroutine must exist for the event being monitored. The subroutine does not need to contain any code. If there is no action subroutine, the Notify event does not occur.

Applicable Parts

Component
Reference

Event Attributes

%EventName	%Part	%SrcCompName	%SrcEvtName
%SrcPartName	%SrcWinName	Window	

OCXEvent

*** Restriction:** This event is unsupported in Java applications.

This event occurs whenever an ActiveX control event occurs.

Applicable Parts

ActiveX

Event Attributes

%EventName	%Part	%RealName	%Window

PageDown

This event occurs when the user presses the PageDown key, clicks on the empty area towards the bottom of the vertical scroll bar, or selects the PageDown button navigation bar.

Applicable Parts

Subfile

Event Attributes

%EventName	%Part	%Window

PageEnd

This event occurs when the user presses the PageDown key or the Down-Arrow key, or if the user clicks on the empty area towards the bottom of the vertical scroll bar or drags the slider down the scroll bar to the bottom of the subfile list when the last record in a subfile is displayed. In this case the **PageDown** event will not be sent, as the list is already at the bottom. A **PageEnd** event is sent instead.

Applicable Parts

Subfile

Event Attributes

%EventName %Part %Window

PageSelect

This event occurs when the user selects a notebook page in a new notebook section by selecting its tab.

The **%Page** event attribute will return the page number of the page selected.

Applicable Parts

Notebook Page

Event Attributes

%EventName %Page %Part %Window

PageTop

This event occurs when the user presses the PageUp key or the Up-Arrow key, or if the user clicks on the empty area towards the top of the vertical scroll bar or drags the slider up the scroll bar to the top of the subfile list. In this case the **PageUp** event will not be sent.

Applicable Parts

Subfile

Event Attributes

%EventName %Part %Window

PageUp

This event occurs when the user presses the PageUp key, if the user clicks the empty area towards the top of the vertical scroll bar or selects the PageUp navigation button.

Applicable Parts

Subfile

Event Attributes

%EventName %Part %Window

PokeAck

This event occurs when the server application in a DDE conversation has processed a poke command. The **%Complete** event attribute indicates if the poke was successful or not.

Applicable Parts

DDE Client

Event Attributes

%Complete %Data %EventName %Item
%Part %Window

Example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
 *
C     DDE1          BEGACT    POKEACK
 *
C     %COMPLETE     IFNE      *ZERO
C     'Poke failed' DSPLY     style         X
 *
C                   ENDIF
 *
C                   ENDACT
 *
```

Popup

This event occurs when the mouse pointer is positioned over a part and the right mouse button is pressed.

Applicable Parts

Canvas	Check Box	Combination Box	Container
Entry Field	Graphic Push Button	List Box	Media Panel
Message Subfile	Multiline Edit	Push Button	Radio Button
Slider	Spin Button	Static Text	Subfile

Event Attributes

%ColNumber	%EventName	%Index	%MouseX
%MouseY	%Part	%Window	

Press

This event occurs when the user clicks on the part with the mouse, or presses the Enter key when the part has input focus.

Applicable Parts

Graphic Push Button	Media Panel	Push Button

Event Attributes

%EventName	%Part	%Window

PrevRec

This event occurs when the 'previous record' button is selected on the navigator bar.

Applicable Parts

Subfile

Event Attributes

%EventName	%Part	%Window

RClickTray

This event occurs when a user right clicks the application's system tray icon. The application must be registered as a system tray application using the Windows Shell_NotifyIconA API.

Applicable Parts

Window	Window with Canvas

Event Attributes

%EventName	%Part	%Window

ReSize

This event occurs when the part has been resized. For example, it can occur if the user resizes a window using the resize border, or changes the size of the part by setting the **Height** or **Width** attribute.

You can use this event to relocate or move parts on a window after the user resizes the window.

During the **ReSize** event, the **Height** and **Width** attributes contain the current size of the window, while the **%NewHeight** and **%NewWidth** event attributes contain the size of the window when the **ReSize** event is complete.

Applicable Parts

Window

Event Attributes

%EventName	%NewHeight	%NewMode*	%NewWidth
%Part	%Window		

*** Note:** See the attribute description for restrictions.

Scroll

This event occurs when a user presses the scroll arrow on a horizontal or vertical scroll bar part. The **%Position** event attribute contains the position of the scroll box when the user stops scrolling.

Applicable Parts

Horizontal Scroll Bar	Vertical Scroll Bar

Event Attributes

%EventName	%Part	%Position	%Window

Select

This event occurs when the user selects a part by clicking on it with the mouse or by using the arrow keys. For a combination box, list box, message subfile, or subfile, this event occurs when an item is selected in the list.

Applicable Parts

Check Box	Combination Box	Container	List Box
Message Subfile	Radio Button	Subfile	

Event Attributes

%ColNumber	%EventName	%Index	%Part
%Window			

SelPending

*** Restriction:** This event is supported in Windows applications only.

This event occurs when the user selects a notebook page in a new notebook section by selecting its tab, and before the page is displayed.

Applicable Parts

Notebook Page

Event Attributes

%EventName	%Part	%Window

ShutDown

This event occurs when the GUI builder ends and the window handle has been specified on the NotifyOnClose plugin command. You can use this event to monitor any plugins running from the GUI builder and have them closed, as well.

Applicable Parts

Window

Event Attributes

%EventName %Part %Window

SpinDown

This event occurs when the user selects the down arrow button on a spin button part.

Applicable Parts

Spin Button

Event Attributes

%EventName %Part %Window

SpinEnd

This event occurs when the user releases the up arrow or down arrow button on a spin button part.

Applicable Parts

Spin Button

Event Attributes

%EventName %Part %Window

SpinUp

This event occurs when the user selects the up arrow button on a spin button part.

Applicable Parts

Spin Button

Event Attributes

%EventName %Part %Window

Terminate

This event occurs when the server in a DDE conversation has requested that the conversation be terminated, or when the DDE client part has its **DDEMode** attribute set to *Terminate*.

Applicable Parts

DDE Client

Event Attributes

%EventName %Part %Window

Example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
 *
C     PB1           BEGACT    TERMINATE
 *
C     'Terminating' DSPLY     STYLE         X
 *
C                   ENDACT
 *
```

Tick

This event occurs each time a timer's interval value has elapsed. The **Multiplier** attribute also determines when the **Tick** event will be generated.

Applicable Parts

Timer

Event Attributes

%EventName %Part %Window

Example

Update a static text part with the current tick value of a timer part:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
 *
C     TIMER1        BEGACT    TICK
 *
C     'TIMER1'      GETATR    TIMERTICKS    timerticks        6 0
C     ST1           SETATR    timerticks    'LABEL'
 *
C                   ENDACT
 *
```

Timeout

This event occurs when a DDE conversation has not been established within the time defined by the **TimeOut** attribute.

Applicable Parts

DDE Client

Event Attributes

%EventName	%Part	%Window

Example

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
 *
C     DDE1           BEGACT    TIMEOUT
 *
C     'DDE Failed!' DSPLY     STYLE         X
C                   ENDACT
 *
```

VKeyPress

This event occurs when a command key has been pressed. Examples of command keys are Page Up, Pause, and Ctrl. Refer to the **%Character** event attribute for a description of the value returned. If you associate this event with the canvas part, you only need to code one action subroutine for all parts that can respond to this event.

Applicable Parts

Canvas	Combination Box	Container	Entry Field
List Box	Multiline Edit	Subfile	

Event Attributes

%Alt	%Character	%Control	%EventName
%Part	%Shift	%Window	

Example

Clear an entry field if F5 is pressed when the entry field has input focus:

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7....+.
CSRN01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq
 *
C     EF1            BEGACT    VKEYPRESS
 *
```

```
C     %character    IFEQ      '29'
C     'EF1'         SETATR    *BLANKS       'TEXT'
C                   ENDIF
 *
C                   ENDACT
 *
```

YearChange

This event occurs when a user selects a new year by clicking one of the year arrows, or by clicking on a month arrow so that it causes the year to change.

Applicable Parts

Calendar

Event Attributes

%EventName %Part %Window

Chapter 4. Event Attributes

Event attributes contain data relevant to the event being handled. They cannot be changed by your application. They must be defined on the *Definition* specification with the correct type and length. The following table lists the event attributes with their corresponding data types and lengths:

Event attribute	Data type	Length
%Alt	Numeric	1
%Button	Numeric	1
%Character	String	1[1]
%ColNumber	Numeric	*
%Complete	Numeric	1
%Control	Numeric	1
%Data	String	*
%EventName	String	10
%Index	Numeric	*
%Item	String	*
%MouseX	Numeric	4
%MouseY	Numeric	4
%NewHeight	Numeric	4
%NewMode	Numeric	1
%NewWidth	Numeric	4
%Page	Numeric	3
%Part	String	10
%Position	Numeric	4
%RealName	String	10
%Shift	Numeric	1
%SrcCompName	String	10
%SrcEvtName	String	10
%SrcPartName	String	10
%SrcWinName	String	10
%Window	String	10

*Define length as required.

[1]For **VKeyPress**, the length is 2.

Example:

```
DName++++++++++++ETDsFrom+++To/L+++IDc.Keywords+++++++++++++++++++++++++++++
 *
 * Declare the %Button event attribute
D %Button        S               1P 0
 *
 * Declare the %Data event attriubte
D %Data          S              32A
```

This chapter describes the event attributes. Each description contains the following:

- The data type of the event attribute
- A general description of the event attribute
- A list of the events the attribute applies to

%Alt

Indicates if the Alt key was pressed when the event occurred:

0 Alt key was not pressed

1 Alt key was pressed

Type Numeric

Applicable Events

KeyPress VKeyPress

%Button

Indicates which mouse button was pressed during the event:

1 Button 1

2 Button 2

3 Button 3

4 Button 1 and Button 2

Type Numeric

Applicable Events

Click DblClick MouseDown MouseUp

%Character

The character code generated by the event.

During the **VKeyPress** event, the **%Character** event attribute will be returned as a value that represents the noncharacter key pressed. The values are as follows:

00 – Esc	13 – End	25 – Break	37 – F14
01 – Tab →|	14 – Page Up	26 – F2	38 – F15
02 – Tab |←	15 – Page Down	27 – F3	39 – F16
03 – Space	16 – ← (Left)	28 – F4	40 – F17
04 – ← Backspace	17 – → (Right)	29 – F5	41 – F18
05 – Enter (Numeric keypad)	18 – ↑ (Up)	30 – F6	42 – F19
06 – Enter	19 – ↓ (Down)	31 – F7	43 – F20
07 – Shift	20 – Caps Lock	32 – F8	44 – F21
08 – Ctrl	21 – Num Lock	33 – F9	45 – F22
09 – Alt	22 – Scroll Lock	34 – F11	46 – F23
10 – Insert	23 – Pause	35 – F12	47 – F24
11 – Delete	24 – SysRq	36 – F13	56 – Alt
12 – Home			

Type String

Applicable Events

Keypress	VKeyPress

%ColNumber

Returns the column number for a container or subfile part.

Type Numeric

Applicable Events

Change	ColSelect	Popup	Select

%Complete

Indicates whether the previous operation completed successfully or not:

0 Not successful

1 Successful

Type Numeric

Applicable Events

Complete ExecuteAck PokeAck

%Control

Indicates if the Ctrl key was pressed when the event occurred:

0 Ctrl key was not pressed.

1 Ctrl key was pressed.

Type Numeric

Applicable Events

KeyPress VKeyPress

%Data

Contains the data sent by a server application in response to a *Request* command. The content and structure of this data is determined by the server application.

This event attribute is also available with the **PokeAck** event. It will contain the original poked data.

Type String

Applicable Events

Data Drop ExecuteAck PokeAck

%EventName

Returns the name of the event.

Type String

Applicable Events

All events

%Index

Returns the index number of the record that is under the mouse pointer when a **Popup** or **Select** event occurs. The record does not have to be in focus.

Type Numeric

Applicable Events

Change Popup Select

%Item

Contains the name of the item that is currently involved in a DDE conversation.

Type String

Applicable Events

Data PokeAck

%MouseX

The current location of the mouse pointer's *X* coordinate within the part when the event occurred. It is expressed in pixels. It will be a value between zero and the width of the part. A value of zero represents the left edge of the part.

Note: For the **MouseMove** event, this coordinate is relative to the desktop.

Type Numeric

Applicable Events

Click DblClick MouseDown MouseMove
MouseUp Popup

%MouseY

The current location of the mouse pointer's *Y* coordinate within the part when the event occurred. It is expressed in pixels. It will be a value between zero and the height of the part. A value of zero represents the bottom edge of the part.

Note: For the **MouseMove** event, this coordinate is relative to the desktop.

Type Numeric

Applicable Events

Click DblClick MouseDown MouseMove
MouseUp Popup

%NewHeight

The height of the part after it has been resized.

Note: The **Height** attribute of the part is not changed until the action subroutine for the **ReSize** event has completed.

Type Numeric

Applicable Events

ReSize

%NewMode

*** Restriction:** This attribute is unsupported in Java applications.

Indicates which mode the window is about to be set to:

1 Maximized

2 Minimized

3 Restored

Type Numeric

Applicable Events

ReSize

%NewWidth

The width of the part after it has been resized.

Note: The **Width** attribute of the part is not changed until the action subroutine for the **ReSize** event has completed.

Type Numeric

Applicable Events

ReSize

%Page

Returns the page number of the selected page when the **PageSelect** event occurs.

Type Numeric

Applicable Events

PageSelect

%Part

Contains the name of the part that generates the event.

Type String

Applicable Events

All events

%Position

Returns the current position of the scroll box when the user stops scrolling.

Type Numeric

Applicable Events

Scroll

%RealName

Returns the actual name of the ActiveX event that occurs.

Type String

Applicable Events

OCXEvent

%Shift

Indicates if the Shift key was pressed when the event occurred:

0 Shift key was not pressed

1 Shift key was pressed

Type Numeric

Applicable Events

KeyPress VKeyPress

%SrcCompName

When a **Notify** event is sent to the component reference part, this event attribute returns the name of the component for the notification event. If one component reference part is used to monitor multiple events, this event attribute can determine the exact source for the event that triggers the Notify event by using the **%SrcCompName** attribute in combination with the **%SrcWinName**, **%SrcPartName**, and **%SrcEvtName** attributes.

Type String

Applicable Events

Notify

%SrcEvtName

When a **Notify** event is sent to the component reference part, this event attribute returns the name of the event that was set in the **NotSrcEvt** attribute, along with a particular part and window, to trigger a **Notify** event to the component reference part.

Type String

Applicable Events

Notify

%SrcPartName

When a **Notify** event is sent to the component reference part, this event attribute returns the name of the part that was set in the **NotSrcPart** attribute, along with a particular event and window, to trigger a **Notify** event to the component reference part.

Type String

Applicable Events

Notify

%SrcWinName

When a **Notify** event is sent to the component reference part, this event attribute returns the name of the window that was set in the **NotSrcWin** attribute, along with a particular part and event, to trigger a **Notify** event to the component reference part.

Type String

Applicable Events

Notify

%Window

Indicates the name of the window containing the part that generated the event.

Type String

Applicable Events

All events

Glossary

This glossary includes terms and definitions from:

- The *American National Dictionary for Information Systems* ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York, 10018. Definitions are defined by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (|) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition indicating that the final agreement has not yet been reached among participating National Bodies of SC1.
- *IBM Dictionary of Computing* , New York: McGraw-Hill, 1994.
- *Object-Oriented Interface Design IBM Common User Interface Guidelines*, SC34-4399-00, Carmel, IN: Que Corporation, 1992.

A

action. (1) Synonym for *action subroutine*. (2) An executable program or command file used to manipulate a project's parts or participate in a build.

action subroutine. Logic that you write to respond to a specific event.

active window. The window with which a user is currently interacting. This is the window that receives keyboard input.

activeX part. A part that adds ActiveX control objects to the project. VARPG applications can then access their attributes and monitor for events.

anchor. Any part that you use as a reference point for aligning, sizing, and spacing other parts.

animation control part. A part that allows the playback of video files, with the AVI extension, in Windows, or the playback of animated GIF sequences in Java applications.

API. Application programming interface.

applet. A program that is written in Java and runs inside of a Java-compatible browser or AppletViewer.

application. A collection of software components used to perform specific user tasks on a computer.

application programming interface (API). A functional interface supplied by the operating system or a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

ASCII (American National Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

B

BMP. The file extension of a bitmap file.

build. The process by which the various pieces of source code that make up components of a VARPG application are compiled and linked to produce an executable version of the application.

button. (1) A mechanism on a pointing device, such as a mouse, used to request or start an action. (2) A graphical mechanism in a window that, when selected, results in an action. An example of a button is an OK push button that, when selected, initiates an action.

C

calendar part. A part that adds a calendar that can be modified by the user to include text, color and other attributes.

canvas part. A part onto which you can point and click various other parts, position them, and organize them to produce a graphical user interface. A canvas part occupies the client area of either a window part or a notebook page part. See also *notebook page with canvas part* and *window with canvas part*.

check box part. A square box with associated text that represents a choice. When a user selects a choice, an indicator appears in the check box to indicate that the choice is selected. The user can clear the check box by selecting the choice again. In VisualAge RPG, you point and click on a check box part in the parts palette or parts catalog and click it onto a design window.

click. To press and release a mouse button without moving the pointer off of the choice or object. See also *double-click*.

client. (1) A system that is dependent on a server to provide it with data. (2) The PWS on which the VARPG applications run. See also *DDE client*.

client area. The portion of the window that is the user's workspace, where a user types information and selects choices from selection fields. In primary windows, the area where an application programmer presents the objects that a user works on.

client/server. The model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server. See also *client*, *server*, *DDE client*, *DDE server*.

clipboard. An area of storage provided by the system to hold data temporarily. Data in the clipboard is available to other applications.

cold-link conversation. In DDE, an explicit request made from a client program to a server program. The server program responds to the request. Contrast with *hot-link conversation*.

color palette. A set of colors that can be used to change the color of any part in your application's GUI.

combination box. A control that combines the functions of an entry field and a list box. A combination box contains a list of objects that a user can scroll through and select from to complete the entry field. Alternatively, a user can type text directly into the entry field. In VisualAge RPG, you can point and click on a combination box part in the parts palette or parts catalog and click it onto a design window.

Common User Access architecture (CUA architecture). Guidelines for the dialog between a human and a workstation or terminal.

compile. To translate a source program into an executable program (an object program).

component. A functional grouping of related files within a project. A component is created when the NOMAIN and EXE keywords are not present on the control specifications.

component reference part. A part that enables one component to communicate with another component in a VARPG application.

***component part.** A part that is the "part representation" of the component. One *component part is created for each component automatically, and it is invisible.

CONFIG.SYS. The configuration file, located in the root directory of the boot drive, for the DOS, OS/2, or Windows operating systems. It contains information required to install and run hardware and software.

configuration. The manner in which the hardware and software of an information processing system are organized and interconnected (T).

container part. A part that stores related records and displays them in a details, icon, or tree view.

CUA architecture. Common User Access architecture.

cursor. The visible indication of the position where user interaction with the keyboard will appear.

D

database. (1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (T) (2) All the data files stored in the system.

data object. An object that conveys information, such as text, graphics, audio, or video.

DBCS. Double-byte character set.

DDE. Dynamic data exchange.

DDE client. An application that initiates a DDE conversation. Contrast with *DDE server*. See also *DDE client part*, *DDE conversation*.

DDE client part. A part used to exchange data with other applications, such as spreadsheet applications, that support the dynamic data exchange (DDE) protocol.

DDE conversation. The exchange of data between a DDE client and a DDE server. See also *cold-link conversation* and *hot-link conversation*.

DDE server. An application that provides data to another DDE-enabled application. Contrast with *DDE client*. See also *DDE conversation*.

default. A value that is automatically supplied or assumed by the system or program when no value is specified by the user. The default value can be assigned to a push button or graphic push button.

default action. An action that will be performed when some action is taken, such a pressing the Enter key.

dereferencing. The action of removing the association between a part and an iSeries database field.

design window. The window in the GUI designer on which parts are placed to create a user interface.

details view. A standard contents view in which a small icon is combined with text to provide descriptive information about an object.

dimmed. Pertaining to the reduced contrast indicating that a part can not be selected or directly manipulated by the user.

direct editing. The use of techniques that allow a user to work with an object by dragging it with a mouse or interacting with its pop-up menu.

DLL. Dynamic link library.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, displaying, and printing of DBCS characters requires hardware and programs that support DBCS. Four double-byte character sets are supported by the system: Japanese, Korean, Simplified Chinese, and Traditional Chinese. Contrast with *single-byte character set (SBCS)*.

double-click. To quickly press a mouse button twice.

drag. To use a mouse to move or to copy an object. For example, a user can drag a window border to make it larger by holding a button while moving the mouse. See also *drag and drop*.

drag and drop. To directly manipulate an object by moving it and placing it somewhere else using a mouse.

drop-down combination box. A variation of a combination box in which a list box is hidden until a user takes explicit acts to make it visible.

drop-down list. A single selection field in which only the current choice is visible. Other choices are hidden until the user explicitly acts to display the list box that contains the other choices.

dynamic data exchange (DDE). The exchange of data between programs or between a program and a datafile object. Any change made to information in one program or session is applied to the identical data created by the other program. See also *DDE conversation*, *DDE client*, *DDE server*.

Dynamic link library (DLL). A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set of 256 8-bit characters.

emphasis. Highlighting, color change, or other visible indication of conditions relative to an object or choice that affects a user's ability to interact with that object or choice. Emphasis can also give a user additional information about the state of a choice or an object.

entry field part. An area on a display where a user can enter information, unless the field is read-only. The boundaries of an entry field are usually indicated. In VisualAge RPG, you point and click on an entry field part in the parts palette or parts catalog and click it onto a design window.

error logging. Keeps track of errors in an **error log**. The editor takes you to the place in the source where the error occurred.

event. A signal generated as a result of a change to the state of a part. For example, pressing a button generates a *Press* event.

exception. (1) In programming languages, an abnormal situation that may arise during execution, that may cause a deviation from the normal execution sequence, and for which facilities exist in a programming language to define, raise, recognize, ignore, and handle it. (I) (2) In VisualAge RPG, an event or situation that prevents, or could prevent, an action requested by a user from being completed in a manner that the user would expect. Exceptions occur when a product is unable to interpret a user's input.

EXE. The extension of an executable file.

EXE module. An EXE module consists of a main procedure and subprocedures. It is created when the EXE keyword is present on the control specification. All subroutines (BEGSR) must be local to a procedure. The EXE must contain a procedure whose name matches the name of the source file. This will be the main entry point for the EXE, that is, the main procedure.

export. A function that converts an internal file to some standard file format for use outside of an application. Contrast with *import*.

F

field. (1) An identifiable area in a window, such as an entry field where a user types text. (2) A group of related bytes, such as a name or amount, that is treated as a unit in a record.

file. A collection of related data that is stored and retrieved by an assigned name. A file can include information that starts a program (program-file object), contains text or graphics (data-file object), or processes a series of commands (batch file).

focus. Synonym for *input focus*.

font palette. A set of fonts that can be used to change the font of a part in your application's GUI.

G

graph part. A part that allows the user to add a graph to the GUI. The graph styles available are line, bar, line and bar, or pie chart.

graphical user interface (GUI). A type of user interface that takes advantage of high-resolution graphics. A graphical user interface includes a combination of graphics, the object-action paradigm, the use of pointing devices, menu bars and other menus, overlapping windows, and icons.

graphic push button part. A push button, labeled with a graphic, that represents an action that will be initiated when a user selects it. Contrast with *push button part*.

group box part. A rectangular frame around a group of controls to indicate that they are related and to provide an optional label for the group. In VisualAge RPG, you point and click on a group box part in the parts palette or parts catalog and click it onto a design window.

group marker. A mark that identifies a part as being the first one in a group. When a user moves the cursor through a group of parts and reaches the last part, the cursor returns to the first part in the group.

GUI designer. A suite of tools used to create interfaces by dragging and dropping parts from the parts palette to the design window.

H

hide button. A button on a title bar that a user clicks on to remove a window from the workplace without closing the window. When the window is hidden, the state of the window, as represented in the window list, changes. Contrast with *maximize button* and *minimize button*.

horizontal scroll bar part. A part that adds a horizontal scroll bar to a window. This part allows users to scroll through a pane of information, from left-to-right or right-to-left.

hot-link conversation. In DDE, an automatic update of a client program by a server program when data changes on the server. Contrast with *cold-link conversation*.

I

ICO. The file extension of an icon file.

icon. A graphical representation of an object, consisting of an image, image background, and a label.

icon view. A standard contents view in which each object contained in a container is displayed as an icon.

image part. A part used to display a picture, from a BMP or ICO file, on a window.

import. A function that converts display file objects to the appropriate VARPG part. Contrast with *export*.

inactive window. A window that can not receive keyboard input at a given moment.

index. The identifier of an entry in VARPG parts such as list boxes or combination boxes.

information area. A part of a window in which information about the object or choice that the cursor is on is displayed. The information area can also contain a message about the normal completion of a process. See also *status bar*.

Information Presentation Facility (IPF). A tool used to create online help on a programmable workstation.

Information Presentation Facility (IPF) file. A file in which the application's help source is stored.

INI. The file extension for a file in the OS/2 or Windows operating system containing application-specific information that needs to be preserved from one call of an application to another.

input focus. The area of a window where user interaction is possible from either the keyboard or the mouse.

input/output (I/O). Data provided to the computer or data resulting from computer processing.

IPF. Information Presentation Facility

item. In dynamic data exchange, a unit of data. For example, the top left cell position in a spreadsheet is row 1, column 1. This cell position may be referred to as item R1C1.

J

JAR files (.jar). In Java, abbreviation for Java ARchive. A file format that is used for aggregating many files into one.

Java. An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

java bean part. A part that allows VARPG applications to access Sun Microsystem's JavaBeans.

JavaBeans. In Java, a portable, platform-independent reusable component model.

Java Database Connectivity (JDBC). An industry standard for database-independent connectivity between Java and a wide range of databases. The JDBC provides a call-level application programming interface (API) for SQL-based database access.

Java 2 Software Development Kit (J2SDK). Software that Sun Microsystems distributes for Java developers. This software includes the Java interpreter, Java classes, and Java development tools. The development tools include a compiler, debugger, dissassembler, AppletViewer, stub file generator, and documentation generator.

Java Native Interface (JNI). A programming interface that allows Java code that runs inside of a Java Virtual Machine (JVM) to interoperate with functions that are written in other programming languages.

Java Runtime Environment (JRE). A subset of the Java Developer Kit for end users and developers who want to redistribute the JRE. The JRE consists of the Java Virtual Machine, the Java Core Classes, and supporting files.

Java Virtual Machine (JVM). The part of the Java Runtime Environment (JRE) that is responsible for interpreting Java bytecodes.

L

link event. An event that a target part receives whenever the state of a source part changes.

list box part. A control that contains scrollable choices that a user can select. In VisualAge RPG, you can point and click on a list box part in the parts palette or parts catalog and click it onto a design window.

M

main procedure. A main procedure is a subprocedure that can be specified as the program entry procedure and receives control when it is first called. A main procedure is only produced when creating an EXE. See *EXE module*

main source section. In a VARPG program, the main source section contains all the global dedfinitions for a module. For a component, this section also includes the action and user subroutines.

main window. See *primary window*.

manipulation button. See *mouse button 2*.

maximize button. A button on the rightmost part of a title bar that a user clicks on to enlarge the window to its largest possible size. Contrast with *minimize button*, *hide button*.

media panel part. A part used to give the user control over other parts. For example, a media panel part can be used to control the volume of a media part.

media part. A part that gives a program the ability to process sound files and video files.

menu. A list of choices that can be applied to an object. A menu can contain choices that are not available for selection in certain contexts. Those choices are dimmed.

menu bar part. The area near the top of a window, below the title bar and above the rest of the window, that contains choices that provide access to other menus. In VisualAge RPG, you can point and click on a menu bar part in the parts palette or parts catalog and click it onto a design window.

menu item part. A part that is a graphical or textual item on a menu. A user selects a menu item to work with an object in some way.

message. (1) Information not requested by a user but displayed by a product in response to an unexpected event or when something undesirable could occur. (2) A communication sent from a person or program to another person or program.

message file. A file containing application messages. The file is created from the message source file during the build process. See also *build*.

message subfile part. A part that can display predefined messages or text supplied in program logic.

migrate. (1) To move to a changed operating environment, usually to a new release or version of a system. (2) To move data from one hierarchy of storage to another.

MID. The file extension of a MIDI file.

MIDI file. Musical Instrument Digital Interface file.

minimize button. A button, located next to the rightmost button in a title bar, that reduces the window to its smallest possible size. Contrast with *maximize button* and *hide button*.

mnemonic. A single character, within the text of a choice, identified by an underscore beneath the character. See also *mnemonic selection*.

mnemonic selection. A selection technique whereby a user selects a choice by typing the mnemonic for that choice.

mouse. A device with one or more push buttons used to position a pointer on the display without using the keyboard. Used to select a choice or function to be performed or to perform operations on the display, such as dragging or drawing lines from one position to another.

mouse button. A mechanism on a mouse used to select choices, initiate actions, or manipulate objects with the pointer. See also *mouse button 1* and *mouse button 2*.

mouse button 1. By default, the left button on a mouse used for selection.

mouse button 2. By default, the right button on a mouse used for manipulation.

mouse pointer. Synonym for *cursor*.

multiline edit (MLE) part. A part representing an entry field that allows the user to enter multiple lines of text.

N

navigation panel. A group of buttons that can be used to control the visible selection of records in a subfile.

NOMAIN module. A module that contains only subprocedures. There are no action or standalone user subroutines in it. A NOMAIN module is created when the NOMAIN keyword is present on the control specification.

notebook part. A graphical representation of a notebook. You can add notebook pages to the notebook part and then group the pages into sections separated by tabbed dividers. In Windows, a notebook is sometimes referred to as a Windows tab control. See also *notebook page part*, *notebook page with canvas part*.

notebook page part. A part used to add pages to a notebook part. See also *notebook*.

notebook page with canvas part. A combination of a notebook page part and a canvas page part. See also *notebook*, *canvas part*.

O

object. (1) A named storage space that consists of a set of characteristics that describe itself and, in some situations, data. An object is anything that exists in and occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, libraries, and folders. (2) A visual component of a user interface that a user can work with to perform a task. An object can appear as text or an icon.

object-action paradigm. A pattern for interaction in which a user selects an object and then selects an action to apply to that object.

object-oriented programming. A method for structuring programs as hierarchically organized classes describing the data and operations of objects that may interact with other objects. (T)

object program. A target program suitable for execution. An object program may or may not require linking. (T)

odbc/jdbc part. A part that allows VAPRG applications to access and process database files that support the Windows ODBC API or Sun Microsystem's JDBC API.

operating system. A collection of system programs that control the overall operation of a computer system.

outline box part. A part that is a rectangular box positioned around a group of parts to indicate that all the parts are related.

P

package. A function used to collect all the parts of a VARPG application together for distribution.

parts. Objects that make up the GUI of a VARPG application.

parts catalog. A storage space for all of the parts used to create graphical user interfaces for VARPG applications.

parts palette. A collection of parts that are most appropriate for building the current graphical user interface for an application. When you finish one GUI, you can wipe the palette clean and add parts from the parts catalog that you require for the next application.

plugin. A function created by the user or an outside vendor that can be used in VARPG programs.

point and click. (1) A selection method which is used to copy a part from the parts palette or catalog to the GUI design window, the icon view, or the tree view. (2) To place a part in any of the desired views, point to and click on the part, then move the cursor to the chosen window and point the cursor and click where you want the part to appear. In the icon and tree views, the part will be placed on the parent part, and you will then have to move it where you would like it to appear in the design window.

pop-up menu. A menu that, when requested, appears next to the object with which it is associated. It contains choices appropriate for the object in its current context.

pop-up menu part. A part that, when added to an object on your interface, appears next to the object with which it is associated when requested. You can point and click on a pop-up menu part in the parts palette or parts catalog and click it onto a design window.

pop-up window. A movable window, fixed in size, in which a user provides information required by an application so that it can continue to process a user request. Synonymous with *secondary window*.

primary window. The window in which the main interaction between the user and the application takes place. Synonymous with *main window*.

procedure. A procedure is any piece of code that can be called with the CALLP operation code.

procedure interface definition. A procedure interface definition is a repetition of the prototype information within the definition of a procedure. It is used to declare the entry parameters for the procedure and to ensure that the internal definition of the procedure is consistent with the external definition (the prototype)

programmable workstation (PWS). A workstation that has some degree of processing capability and that allows a user to change its functions.

progress bar part. A part that can be used to indicate graphically the progress of a process, such as copying files, loading a database, and so on.

progress indicator. One or more controls used to inform a user about the progress of a process.

project. The complete set of data and actions needed to build a single target, such as dynamic link library (DLL) or an executable file (EXE).

prompt. (1) A visual or audible message sent by a program to request the user's response. (T) (2) A displayed symbol or message that requests input from the user or gives operational information. The user must respond to the prompt in order to proceed.

properties notebook. A graphical representation that resembles a bound notebook containing pages separated into sections by tabbed divider pages. Select the tabs of a notebook to move from one section to another.

prototype. A prototype is a definition of the call interface. It includes information such as: whether the call is bound (procedure) or dynamic (program); the external name; the number and nature of the parameters; which parameters must be passed; the data type of any return value (for a procedure)

pull-down menu. A menu that extends from a selected choice on a menu bar or from a system-menu symbol. The choices in a pull-down menu are related to one another in some manner.

push button part. A button labeled with text that represents an action that starts when a user selects the push button. You can point and click on a push button part in the parts palette or parts catalog and click it onto a design window. See also *graphic push button part*.

PWS. Programmable workstation.

R

radio button part. A circle with text beside it. Radio buttons are combined to show a user a fixed set of choices from which only one can be selected. The circle is partially filled when a choice is selected. You can point and click on a radio button part in the parts palette or parts catalog and click it onto a design window.

reference field. An iSeries database field from which an entry field part can inherit its characteristics.

restore button. A button that appears in the rightmost corner of the title bar after a window has been maximized. When the restore button is selected, the window returns to the size and position it was in before it was maximized. See also *maximize button*.

S

SBCS. Single-byte character set.

scroll bar. A part that shows a user that more information is available in a particular direction and can be moved into view by using a mouse or the page keys.

secondary window. A window that contains information that is dependent on information in a primary window, and is used to supplement the interaction in the primary window. See also *primary window*. Synonym for *pop-up window*.

secure sockets layer (SSL). A popular security scheme that was developed by Netscape Communications Corp. and RSA Data Security, Inc. SSL allows the client to authenticate the server and all data and requests to be encrypted. The URL of a secure server that is protected by SSL begins with https rather than http.

selection border. The visual border that appears around a VARPG part or a custom-made part, allowing the part to be moved with the mouse or keyboard.

selection button. See *mouse button 1*.

server. A system in a network that handles the requests of another system, called a client.

server alias. A name you define that can be used instead of the server name.

shared component. A component that can be accessed by more than one project.

single-byte character set (SBCS). A character set in which each character is represented by a one-byte code. Contrast with *double-byte character set (DBCS)*.

sizing border. The border or frame around a part (or set of parts) that you select to resize the part (or set of parts) using the mouse or the keyboard.

slider part. A visual component of a user interface that represents a quantity and its relationship to the range of possible values for that quantity. A user can also change the value of the quantity. You can point and click on a slider part in the parts palette or parts catalog and click it onto a design window.

slider arm. The visual indicator in the slider that a user can move to change the numerical value.

source directory. The directory in which all source files for a VARPG application are stored.

source part. A part that can notify target parts whenever the state of the source part changes. A source part can have multiple targets.

spin button part. A type of entry field that shows a ring of related but mutually exclusive choices through which a user can scroll and select one choice. A user can also type a valid choice in the entry field. You can point and click on a spin button part in the parts palette or parts catalog and click it onto a design window.

SSL. Secure sockets layer.

static text part. A part used as a label for other parts, such as a prompt for an entry field part.

status bar. A part of a window that displays information indicating the state of the current view or object. See also *information area*.

status bar part. A part on a window that can display additional information about a process or action for the window.

subfile field. A field used to define fields in a subfile part. See also *subfile part*.

subfile part. A part used to display a list of records, each consisting of a number of fields. This part is similar to an iSeries subfile. See also *subfile field*.

submenu. A menu that appears from, and contains choices related to, a cascading choice in another menu. Submenus are used to reduce the length of a pull-down menu or a pop-up menu. See also *submenu part*.

submenu part. A part used to start a submenu from a menu item or existing menu, or to start a pull-down menu from a menu item on a menu bar. See also *submenu* and *menu item part*.

subprocedure. A subprocedure is a procedure specified after the main source section. It must have a corresponding prototype in the definition specifications of the main source section

syntax checking. Verifies that the syntax of each line is correct while you are editing the source. By doing so, it can avoid compile errors. You can set this option on or off. You can view only certain specification types, such as C specs, or a line with a specific string.

T

tab stop. An attribute used to set a tab stop for a part so that users can focus on it when they use the Tab key to move through the interface.

target part. A part that receives a link event from a source part whenever the state of the source part changes.

target directory. The directory in which the compiled VARPG application is stored after a build. Contrast with *target folder*.

target folder. The object in which the icon representing a VARPG application is placed.

target program. The object to be built by the project, such as a dynamic link library (DLL).

thread. The smallest unit of operation to be performed within a process.

timer part. A part used to track the interval of time between two events and trigger the second event when the interval has passed.

title bar. The area at the top of each window that contains the system-menu symbol.

token highlighting. Enhances the readability of the code. You can configure highlighting of different language constructs with different colors or fonts to identify the program structures. You can turn token highlighting on or off.

tool bar. A menu that contains one or more graphical choices representing actions a user can perform using a mouse.

topic. In dynamic data exchange (DDE), the set of data that is the subject of a DDE conversation.

tree view. A way of displaying the contents of an object in a hierarchical fashion.

U

user-defined part. A part, consisting of one or more parts you have customized, that you save to the parts palette or parts catalog for reuse. When in the palette or catalog, you can point and click this part onto the design window as you would any other VARPG part.

utility DLL. See *NOMAIN module*

V

vertical scroll bar part. A part that adds a vertical scroll bar to a window. This part allows users to scroll through a pane of information vertically.

W

WAV. The file extension of a wave file.

wave file. A file used for audio sounds on a waveform device.

window part. An area with visible boundaries that represents a view of an object or with which a user conducts a dialog with a computer system. You can point and click on a window part from the parts palette or parts catalog and click it onto the project window.

window with canvas part. A combination of the window part and the canvas part. See also *window part* and *canvas part*.

work area. An area used to organize objects according to a user's tasks. When a user closes a work area, all windows opened from objects contained in the work area are removed from the workplace.

workplace. An area that fills the entire display and holds all of the objects that make up the user interface.

workstation. A device that allows a user to do work. See also *programmable workstation*.

Bibliography

For additional information about topics related to WebSphere Development Studio Client, refer to the following IBM publications:

WebSphere Development Studio Client manuals:

- *Getting Started with WebSphere Development Studio Client for iSeries*, SC09-2625-06, provides information about WebSphere Development Studio Client for iSeries, giving an overview of the various components, how they work together, and the business advantages of using them.

VisualAge RPG manuals:

- *Programming with VisualAge RPG*, SC09-2449-05, contains specific information about creating applications with VisualAge RPG. It describes the steps you have to follow at every stage of the application development cycle, from design to packaging and distribution. Programming examples are included to clarify the concepts and the process of developing VARPG applications.
- *VisualAge RPG Parts Reference*, SC09-2450-05, provides a description of each VARPG part, part attribute, part event, part attribute, and event attribute. It is a reference for anyone who is developing applications using VisualAge RPG.
- *VisualAge RPG Language Reference*, SC09-2451-04, provides reference information about the VARPG language and compiler.
- *Java for RPG Programmers* introduces you to the Java language (and RPG IV) by comparing it to the RPG language. It is a good first step in your Java journey. It also includes an interactive CD tutorial on Java and VisualAge for Java, by MINDQ.
- *Experience RPG IV Tutorial* is an interactive CD tutorial that teaches you RPG IV and ILE, in a fun and step-by-step approach. The book is a handbook with questions and exercises to help you get hands-on experience with this exciting new version of RPG.
- Another non-IBM book of interest to VisualAge RPG users is *VisualAge for RPG by Example*.

If you have internet access, you can obtain current iSeries and AS/400e information and publications from the following Web site:

`http://www.ibm.com/eserver/iseries/infocenter`

For the PDF version of iSeries publications, refer to the CD ROM *iSeries Information Center: Supplemental Manuals*, SK3T-4092.

Application Development Manager manuals:

- *ADTS/400: Application Development Manager Introduction and Planning Guide*, GC09-1807-00, describes the basic concepts and the planning needed to make effective use of the Application Development Manager feature.
- *ADTS: Application Development Manager User's Guide*, SC09-2133-02, describes how to create and manage projects defined to the Application Development Manager feature.
- *ADTS/400: Application Development Manager Self-Study Guide*, SC09-2138-00, provides practical hands-on experience using the Application Development Manager feature. The guide illustrates how to use the Application Development Manager feature by leading you through a series of step-by-step exercises.
- *ADTS/400: Application Development Manager API Reference*, SC09-2180-00, describes how application programmers can write their own interface to the Application Development Manager feature.

Information Presentation Facility manual:

- *Information Presentation Facility Programming Guide* G25H-7110, describes the elements that make up the Information Presentation Facility (IPF). IPF is a tool that supports the design and development of online documents and online help facilities.

SQL manuals:

- *IBM SQL Reference Version 2* SC26-8416, Volume 2, compares the facilities of
  - DB2
  - SQL/DS™
  - DB2/400™
  - DB2/6000™
  - IBM SQL
  - ISO-ANSI (SQL92E)
  - X/Open™ (XPG4-SQL).
- *DB2 Universal Database Administration Guide* S10J-8157, provides information necessary to use and administer the DB2 product.
- *DB2 Universal Database Embedded SQL Programming Guide* S10J-8158, describes how to design and code application programs that access DB2 Client/Server family servers (such as DB2 or DB2/400). It presents detailed information on the use of Structured Query Language (SQL), and API calls in applications.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Lab Director
IBM Canada Ltd. Laboratory
B3/KB7/8200/MKM
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This publication is intended to help you to create and manage VisualAge RPG applications and user interfaces on the workstation, in a client/server environment. This publication documents General-Use Programming Interface and Associated Guidance Information provided by IBM WebSphere Development Studio Client for iSeries.

Trademarks and Service Marks

The following terms are trademarks or registered trademarks of the International Business Machines Corporation in the United States or other countries or both:

AS/400e	Common User Access	CUA
DATABASE 2	DB2	DB2 Connect

DB2 Universal Database	IBM	iSeries
OS/400	SQL/DS	VisualAge
WebSphere	400	

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

ActiveX, Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation in the United States, or other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Printed in U.S.A.

SC09-2450-05