

Overview

Bidirectional Features help describes Microsoft Visual Basic 5.0 standard features to create Windows applications with full bidirectional language functionality. These features are available only when Microsoft Visual Basic 5.0 is installed in a bidirectional 32-bit Microsoft Windows environment. Arabic Microsoft Windows 95 is an example of such an environment. Other bidirectional 32-bit Microsoft Windows environments are available.

Unless otherwise noted, all topics in Bidirectional Features Help assume Microsoft Visual Basic 5.0 has been installed in a bidirectional 32-bit Microsoft Windows environment.

‘Bidirectional’ is a generic term used to describe software products that support Arabic and other languages that are written right-to-left. More specifically, bidirectional refers to the product ability to manipulate and display text for both left-to-right and right-to-left languages. For example, to display a sentence containing words written in both English and Arabic requires bidirectional capability.

Terminology

‘Bidirectional’ is a generic term used to describe software products that support Arabic and other languages that are written right-to-left. More specifically, bidirectional refers to the product ability to manipulate and display text for both left-to-right and right-to-left languages. For example, to display a sentence containing words written in both English and Arabic requires bidirectional capability.

RTL is frequently used as an abbreviation for right-to-left. Similarly, LTR is frequently used as an abbreviation for left-to-right.

Bidirectional-specific Features

Although the Microsoft Visual Basic 5.0 user interface (menus, dialog boxes, and Help) is in English, you will find the convenience and ease of use of Microsoft Visual Basic 5.0 bidirectional features are indispensable for your bidirectional programming needs:

- **Create bidirectional applications quickly and easily** Many bidirectional features appear as new properties in the Properties window for easy program development.
- **Combine languages** Mix right-to-left and left-to-right language text (for example, Arabic and English) in your code as you build applications for a bidirectional 32-bit Microsoft Windows environment.
- **Create right-to-left visual features** Add right-to-left visual features to forms, menus, and more than 15 custom controls such as grids, list boxes, and combo boxes.
- **Database support** Develop database solutions with support for Arabic and other right-to-left language sort orders.
- **Single solution** There is no need to install a separate bidirectional version of Visual Basic to create bidirectional applications.

Compatibility

Project files that contain bidirectional feature references can be directly opened and edited in any environment that supports Microsoft Visual Basic 5.0. However, the file will run successfully only in a bidirectional 32-bit Microsoft Windows environment.

Project files that contain no bidirectional feature references can be opened, edited and run in any environment that supports Microsoft Visual Basic 5.0. Microsoft Visual Basic 5.0 provides full forward compatibility for project files saved in earlier versions of Microsoft Visual Basic, including bidirectional versions such as Arabic Add-On Pack for Microsoft Visual Basic 4.0.

Setting Up

Bidirectional features are available only when Microsoft Visual Basic 5.0 is installed in a bidirectional 32-bit Microsoft Windows environment. Arabic Microsoft Windows 95 is an example of such an environment. Other bidirectional 32-bit Microsoft Windows environments are available.

All other setup steps are the same as the standard Microsoft Visual Basic 5.0 setup procedures.

Keyboard Language

Keyboard language, as used in this supplement, refers to both the character set (Middle East or Latin) and the character entry direction appropriate for each character set (right-to-left, or *RTL*, for Middle East; left-to-right, or *LTR*, for Latin). Latin usually indicates U.S. English, but more accurately refers to English and other European languages. U.S. English is the default language using Latin characters. Middle East languages include Arabic and other languages written RTL.

To change the keyboard language with a mouse in a bidirectional Microsoft Windows 95 environment, use the Windows 95 taskbar to choose the desired language. For example, in Arabic Windows 95, the Language button at the rightmost end of the taskbar (next to the time display) is marked “Ar” for Arabic and “En” for English.

To change keyboard languages from the keyboard in either operating environment

Press ALT+RIGHT SHIFT or ALT+LEFT SHIFT.

Note

When there are only two keyboard languages installed, for example Arabic and English, the keyboard combinations work identically, switching between the two languages. However, when there are more than two languages available, pressing ALT+RIGHT SHIFT repeatedly rotates through the languages in one direction while pressing ALT+LEFT SHIFT rotates through the list in the opposite direction.

Reading Order

Text entered in Visual Basic has either an RTL (Middle East) or an LTR (Latin) *reading order*. Reading order describes the order in which words are displayed in mixed text. It is instructive to note that reading order pertains to the word order, not the order of the entered characters. When Arabic is the keyboard language, new characters will always flow in a right-to-left manner. Conversely, when Latin is the keyboard language, characters flow in a left-to-right manner.

Reading order becomes important for mixed text situations where text strings contain words in both Middle East and Latin languages. This concept is illustrated in the following examples using Arabic and English text:

- The first two lines demonstrate that no matter which reading order is chosen, text in a pure Arabic or Latin run appears according to the expected behavior of each language.
- The third line shows that in an LTR reading order, Arabic text added to an existing Latin run flows toward the right of the Latin run. An RTL reading order displays the added Arabic text to the left of the Latin run.
- The fourth line shows that in an LTR reading order, Latin text added to an existing Arabic run flows toward the right of the Arabic run. An RTL reading order arranges the added Latin text to the left of the Arabic run.

<u>LTR Reading Order</u>	<u>RTL Reading Order</u>
Latin	Latin
عربي	عربي
Latinعربي	عربي Latin
عربيLatin	Latinعربي

Cursor Characteristics

Cursor movement and text selection can conform to either a *logical* or *visual* set of rules. Mixed text editing in Visual Basic only supports logical selection and movement.

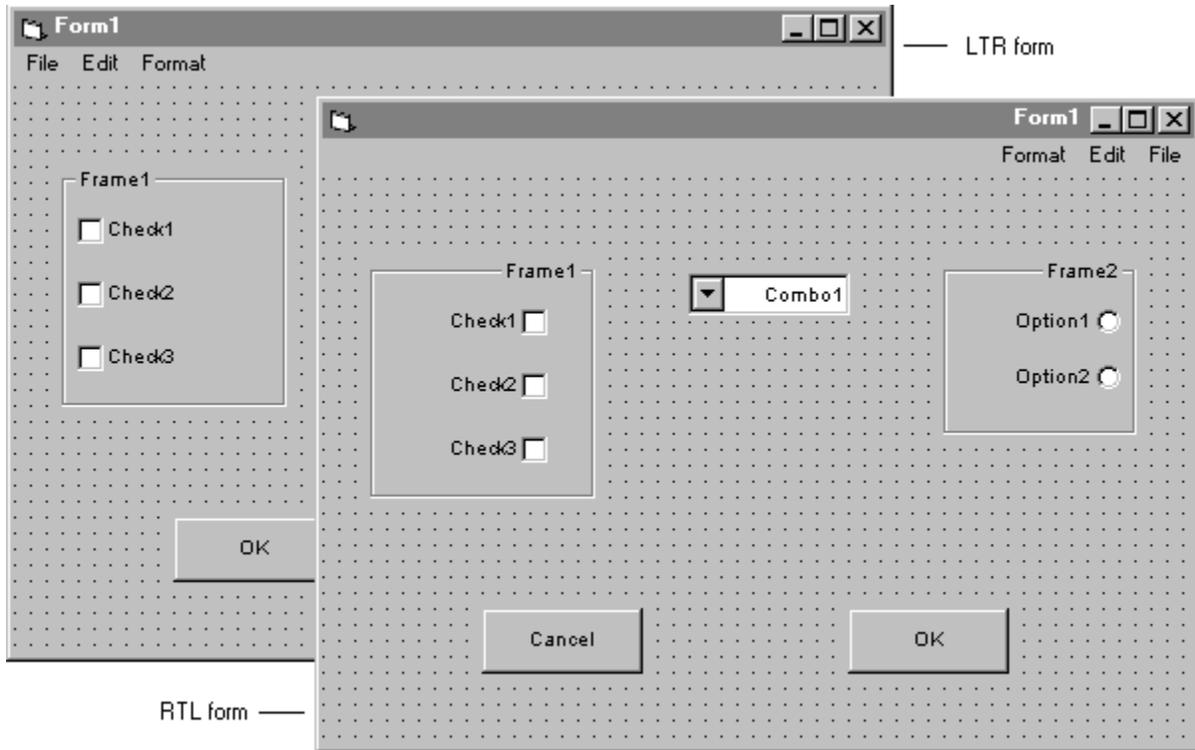
Logical movement and selection follows the reading order of mixed Latin and Middle East text. For example, this causes the insertion point (or cursor) to move to the right in English text and to the left in Arabic text.

Direction

Forms and their associated objects can be displayed in layouts familiar to the Middle East or Latin user. The object's *direction* is RTL for Middle East appearance and LTR for Latin.

When you start Visual Basic, the default form and control direction is LTR. Subsequently, any form within the project will have an initial LTR direction. You can easily change any individual form's direction using the RightToLeft property. If you change a form's direction, the direction of subsequent new controls will be the same as the new form direction.

The following illustration shows examples of LTR and RTL forms...



RightToLeft Property

The majority of bidirectional features in Visual Basic 5.0 involve added ability to create forms, menus, and controls that are oriented in a right-to-left format, as well as in a left-to-right format.

The RightToLeft property has been added to forms, controls, and other Visual Basic objects to provide an easy mechanism for creating objects with Middle East characteristics such as:

- RTL visual direction
- RTL reading order
- Right-aligned text
- Mixed text data entry

Note: The value of a form's RightToLeft property determines the initial RightToLeft value of any objects that are placed on it. Therefore, it is recommended that you create a form with the direction you want before adding controls. You can change the direction of a form after controls are placed on it but you will have to manually reset each existing control's bidirectional properties.

Alignment Property

Visual Basic uses the Alignment property to re-orient controls so they appear in a proper RTL format. For example, if the Alignment property for a check box is set to Right Justify, the check box portion of the control will be placed to the right of the label portion of the control.

Note

New objects placed on an RTL form will appear with their RightToLeft property set to True and Alignment property (if applicable) set to Right Justify and will display in a correct RTL direction.

Form Object

Forms:

- Control the layout of menus.
- Determine the form caption reading order.
- Establish the default RightToLeft property value for newly added controls that have a RightToLeft property.
- Determine printing behavior.

Note

Bidirectional form behavior applies to all forms: MDI (multiple document interface) forms, MDI child forms, or non-MDI forms.

Forms are RTL or LTR depending on the setting for their RightToLeft property. A form with a RightToLeft property set to False behaves exactly as it's described in Visual Basic 5.0 documentation when not running in a bidirectional 32-bit Microsoft Windows environment (this is the default). If the RightToLeft property is set to True, several actions occur:

- Menus and their associated command names (or items) are right-aligned with RTL reading order. They appear at the right edge of the form in an overall RTL layout.
- Form captions have an RTL reading order and are right-aligned in the title bar.
- MDI forms have their scroll bars changed to appear in an RTL layout. The value of the ScrollBars property affects which scroll bars change:

ScrollBars property setting	Behavior
1	Vertical scroll bars are placed on the left
2	Horizontal scroll bars are placed with the scroll box to the right.
3	Both scroll bars are positioned as described for settings 1 and 2.

- Any text that the Print method prints to the form will begin at the right boundary of the form and have RTL reading order.

Menus

The Menu Editor provides full Middle East, Latin, and mixed text editing capabilities that allow you to create applications with a menu structure your users are comfortable with.

The Menu Editor displays items LTR, with LTR reading order, left alignment, and with menu indentation indicators on the left. When you are designing RTL applications, at design time you will notice the menu headings are also left-aligned in the menu bar.

However, at run time, the reading order and layout of menu items is determined by the `RightToLeft` property setting for the parent form. Therefore, a RTL form (whose `RightToLeft` property is set to `True`) will correctly align menus and menu items to the right with RTL reading order.

Printing

Visual Basic provides an RTL printing capability for these three components:

- Form object
- Printer object
- PictureBox control

When the RightToLeft property is set to True for forms, picture boxes, or the Printer object, the upper-right corner is treated as the beginning of the print area, and the Print method will then print text in an anticipated RTL reading order starting at the right margin. Tab and Space print commands will advance from right-to-left across the print area. Carriage Return will move to the right edge of the print line.

However, the underlying XY coordinate system retains the upper-left corner point as the origin reference point. Subsequently, all CurrentX and CurrentY references are determined by using the upper-left corner as the point of origin.

Specific printing behaviors for forms and picture boxes are covered in their respective topics.

Printer Object

The Printer object has configurable reading order and alignment of text printed by its Print method.

When the RightToLeft property setting is set to True:

- Reading order will be right-to-left.
- Printing begins at the right printer margin.

When the RightToLeft property setting is set to False:

- Reading order will be left-to-right.
- Printing begins at the left printer margin.

CheckBox Control

Bidirectional Features The CheckBox control has configurable caption text alignment and reading order, and automatic adjustment of check box position for new controls.

At design time, CheckBox controls are initially created with the appearance and behavior determined by the RightToLeft property setting of their parent form. For example, if the RightToLeft property of the form is True, the two CheckBox properties that affect bidirectional behavior are set as follows:

- RightToLeft=True
- Alignment=1-Right Justify

Caption text reading order will be RTL, caption text will be right aligned, and the check box will be positioned to the right of the caption.

After a CheckBox control is placed on a form, caption and check box behavior are independently controlled by setting the RightToLeft and Alignment properties, respectively.

Developer's Tip

The RightToLeft property can be set in design time or run time. The Alignment property can be set only in design time, following the same behavior the property displays in a non-bidirectional Windows environment.

ComboBox Control

Bidirectional Features The ComboBox control has configurable alignment and reading order of list box items, automatic adjustment of ComboBox appearance, and supports bidirectional language editing in the edit box.

The state of the RightToLeft property determines reading order, combo box appearance, and text alignment.

When RightToLeft is set to True:

- Reading order of list box items is RTL and items are right aligned.
- Reading order of edit box text is RTL and text is right aligned.
- If the property Style=Dropdown Combo or Dropdown List, the drop down arrow is at the left of the combo box.

When RightToLeft is set to False, reading order is LTR, text is left aligned and the drop down arrow appears at the left of the combo box.

Developer's Tip

The RightToLeft property can be set only in design time. It cannot be set in run time. If a procedure attempts to set the RightToLeft property during run time, a warning message box will be displayed.

Command Button Control

Bidirectional Features The Command button control has configurable caption reading order and provides bidirectional language support for the command button caption.

The default reading order is inherited from the parent form RightToLeft property setting; however you can change the direction with the Command button's own RightToLeft property.

Developer's Tip

The RightToLeft property can be changed in design time or run time.

Data Control

Bi-directional Features The Data control has configurable navigation button behavior and caption text layout.

When the `RightToLeft` property is set to `True`:

- The order of the navigation buttons is inverted to flow RTL. For example, the ‘Move to first row’, ‘Move to previous row’, ‘Move to next row’, and ‘Move to last row’ triangle buttons flow from right to left across the navigation bar.
- Reading order for `Caption` property text displayed on the navigation bar is RTL.
- `Caption` property text is right aligned in the caption region of the navigation bar.

When the `RightToLeft` property is set to `False`, navigation buttons flow in the same LTR order as occurs for Data controls displayed in a non-bidirectional Windows environment (this is the default).

DBCombo Control

Bi-directional Features The DBCombo control has configurable item name alignment and reading order, automatic adjustment of DBCombo appearance, and supports bidirectional language editing in the edit box.

The DBCombo control's RightToLeft property provides the same bidirectional features as the basic Combo control. The state of the RightToLeft property determines reading order, combo box appearance, and text alignment.

When the RightToLeft property is set to True:

- Reading order of list box items is RTL and items are right aligned.
- Reading order of edit box text is RTL and text is right aligned.
- If the property Style=Dropdown Combo or Dropdown List, the drop down arrow is at the left of the combo box.

When the RightToLeft property is set to False, reading order is LTR, text and items are left aligned and the drop down arrow appears at the right of the combo box.

Developer's Tip

The RightToLeft property can be set only in design time. It cannot be set in run time. If a procedure attempts to set the RightToLeft property during run time, a warning message box will be displayed.

DBList Control

Bi-directional Features The DBList control has configurable alignment and reading order of list box items, and automatic positioning of the vertical scroll bar.

When the RightToLeft property is set to True:

- Reading order of DBList box items is RTL
- List box items are right aligned.
- If the list area is smaller than the number of items to be displayed, a vertical scroll bar appears automatically at the left of the list box.

When the RightToLeft property is set to False, item reading order is LTR, items are left aligned, and the vertical scroll bar is positioned at the right of the list box.

Developer's Tip

The RightToLeft property can be set only in design time. It cannot be set in run time. If a procedure attempts to set the RightToLeft property during run time, a warning message box will be displayed.

Frame Control

Bi-directional Features The Frame control tool has configurable caption text reading order and automatic caption positioning.

When the RightToLeft property is set to True:

- Caption reading order is set to RTL
- Caption is placed at the upper right corner of the frame.

If the RightToLeft property is False, the caption reading order is LTR and the caption is placed at the upper left corner of the frame.

Note

The Frame border is sized to fit caption text length, so caption text alignment is irrelevant.

Grid Control

Bi-directional Features The Grid control has a configurable cell matrix with several associated behaviors that provide an appearance and functionality familiar to Middle East or Latin users.

When the RightToLeft property is set to True:

- Grid columns begin at the right boundary of the grid.
- Fixed columns are located on the right side of the grid.
- LeftCol identifies the rightmost visible column (the first column beyond the leftmost fixed column).
- SelStartCol and SelStartRow together specify the cell in the upper-right corner of a selected range, while SelEndCol and SelEndRow specify the cell in the lower-left corner of a selected range.
- If the ScrollBars property is set to 3, a vertical scroll bar is placed on the left of the grid and a horizontal scroll bar with the scroll box on the right is placed at the bottom. A ScrollBars property setting of 1 places only the horizontal scroll bar, while a setting of 2 places only the vertical scroll bar.
- Cell values (grid Text property) have RTL reading order.
- Data is automatically right aligned in each cell by setting the FixedAlignment and ColAlignment properties to 1-Right-aligned. Subsequently, the developer can independently configure cell data alignment by programmatically adjusting the FixedAlignment and ColAlignment properties at run time.
- Cells filled with the Clip property fill from right to left beginning at the upper rightmost selected cell.

When the RightToLeft property is set to False, the control behavior is the same as described in the standard Grid control documentation. Data is automatically left aligned in each cell by setting the FixedAlignment and ColAlignment properties to 0-Left-aligned. Subsequently, the developer can independently configure cell data alignment by programmatically adjusting the FixedAlignment and ColAlignment properties at run time.

Developer's Tip

The FixedAlignment, ColAlignment, and Clip properties can be set only in run time. They are not available in design time.

Note

The RightToLeft property is always False by default, regardless of the RightToLeft property value of its parent form.

HScrollBar Control

Bi-directional Features The HScrollBar control has configurable positioning.

When the RightToLeft property is set to True, the scroll box is placed:

- At the right of the bar when the scroll box position value equals the Min property value (Min position)
- At the left of the bar when the scroll box position value equals the Max property value (Max position)

When the RightToLeft property is set to False, the control behavior is the same as described in the standard HScrollBar control documentation.

Developer's Tip

The same behavior can be achieved with a standard HScrollBar by setting Max less than Min, and inverting reported scroll bar positions within HScrollBar event handling procedures.

Label Control

Bi-directional Features The Label control has configurable label text alignment and reading order.

Label controls are initially created with a layout determined by the default RightToLeft property setting of their parent form.

When the RightToLeft property is set to True:

- Caption text reading order is RTL
- The Alignment property is set to Right Justify to right align the caption text.

When the RightToLeft property is set to False:

- Caption text reading order is LTR
- The Alignment property is set to Left Justify to left align the caption text.

Subsequently, caption reading order and alignment behavior are independently controlled by setting RightToLeft and Alignment. The Alignment property can be set only in design time, following the same behavior the property displays in a non-bidirectional Windows environment.

Developer's Tip

Whenever wordwrap occurs (AutoSize and WordWrap properties of the Label control are both set True, or AutoSize property is False), alignment of wrapped lines is determined by the Alignment property.

ListBox Control

Bi-directional Features The List box control has configurable alignment and reading order of list box items and automatic positioning of scroll bars.

When the RightToLeft property is set to True:

- Reading order of list box items is RTL and items are right aligned.
- If the list box scrolls vertically (Columns property=0) and the list area is smaller than the number of items to be displayed, a drop-down arrow and vertical scrollbar appear at the left of the list box.
- If the list box scrolls horizontally (Columns property>0), list box items are arranged in snaking columns, filling the first column on the right, followed by subsequent columns on the left. If the list area is smaller than the number of items to be displayed, a horizontal scrollbar appears at the bottom of the list box and initial scroll box position is at the right.

When the RightToLeft property is set to False, the control behavior is the same as described in the standard ListBox control documentation.

Developer's Tip

The RightToLeft property can be set only in design time. It cannot be set in run time. If a procedure attempts to set the RightToLeft property during run time, a warning message box will be displayed.

MSFlexGrid Control

Bi-directional Features The MSFlexGrid control has a configurable cell matrix with several associated behaviors that provide an appearance and functionality familiar to Middle East users.

When the RightToLeft property is set to True:

- Grid columns begin at the right boundary of the grid. Columns and rows are numbered from zero, beginning at the right for columns and at the top for rows.
- Fixed columns are located on the right side of the grid.
- Cell values have RTL reading order.
- ColAlignment property is set to 1-Right Aligned to right align data in each cell. Subsequently, cell data alignment can be adjusted programmatically at run time.
- LeftCol identifies the rightmost visible column (the first column beyond the leftmost fixed column).
- If the ScrollBars property is set to 3, a vertical scroll bar is placed on the left of the grid and a horizontal scroll bar with the scroll box on the right is placed at the bottom. A ScrollBars property setting of 1 places only the horizontal scroll bar, while a setting of 2 places only the vertical scroll bar.
- Cells filled with the Clip property fill from right to left beginning at the upper rightmost selected cell.

When the RightToLeft property is set to False, the control behavior is the same as described in the standard MSFlexGrid control documentation. ColAlignment property is set to 0-Left Aligned to left align data in each cell. Subsequently, cell data alignment can be adjusted programmatically at run time.

Developer's Tip

ColAlignment and Clip properties can be set only in run time. They are not available in design time.

OptionButton Control

Bi-directional Features The OptionButton control has configurable caption text alignment and reading order, and automatic adjustment of option button position for new controls.

At design time, OptionButton controls are initially created with their appearance and behavior determined by the RightToLeft property setting of the parent form.

When the RightToLeft property is set to True:

- Caption reading order is RTL and caption text is right-aligned.
- The Alignment property is set to Right Justify to place the option button to the right of the caption.

Subsequently, caption and option button behavior are independently controlled by setting the RightToLeft and Alignment properties.

Developer's Tip

The RightToLeft property can be set in design time or run time. The Alignment property can be set only in design time.

PictureBox Control

Bi-directional Features The PictureBox control has configurable printed text alignment and reading order.

The reading order and alignment of text displayed by the Print method of the PictureBox control is determined by the RightToLeft property setting.

If RightToLeft property is set to True:

- Text reading order is RTL.
- Text begins at the right boundary of the picture box.

When RightToLeft is set to False, the opposite effects occur.

See also [Printing](#) for further details on RTL printing displayed by the Print method.

TextBox Control

Bi-directional Features The TextBox control has configurable reading order and alignment for displayed text and automatic adjustment of horizontal and vertical scroll bars, including scroll box positioning.

TextBox controls are initially created with a layout determined by the default RightToLeft property setting of their parent form.

If the RightToLeft property is set to True:

- Text reading order is RTL.
- The Alignment property is set to Right Justify to right align text.

The Alignment property also determines scroll bar placement and behavior for multi-line text boxes (when the MultiLine property is set to True).

- A vertical scroll bar is placed on the left of the text box if the ScrollBars property is set to 2 or 3
- A horizontal scroll bar (with the scroll box on the right) is placed at the bottom if the ScrollBars property is set to 1 or 3.

Note

If the MultiLine property setting of a TextBox control is False, the Alignment property is ignored (this is standard Visual Basic behavior). In this circumstance, Visual Basic internally forces text alignment to 1-Right Justify if the RightToLeft property is set to True.

Whenever wordwrap occurs, alignment of wrapped lines is determined by the Alignment property. Text can be Middle East, Latin, or mixed.

If the RightToLeft property is set to False, the opposite effects occur as described in standard TextBox control documentation.

After a TextBox control is placed on a form, text reading order and alignment behavior are independently controlled by setting the RightToLeft and Alignment properties.

Developer's Tip

The RightToLeft property and Alignment property can be set only in design time. They cannot be set in run time. If a procedure attempts to set either of these properties during run time, a warning message box will be displayed.

Creating Middle East Database Applications

Middle East sort order support is included in the Jet database engine to enable searching, sorting, and display of Middle East and Latin text in a widely accepted collating sequence. Separate sort orders are provided for Arabic and other right-to-left Middle East languages.

Each Middle East sort order accommodates both Latin and the appropriate Middle East language text. For example, the Arabic sort order describes the collating sequence for Latin (primarily English and French) and Arabic text.

Refer to the discussion of the `CreateDatabase` method in online Help for information about how to create a database with Middle East collating order. Databases with Middle East sort orders are supported only when Visual Basic 5.0 runs in a bidirectional 32-bit Windows environment.

Visual Basic 5.0 running in a non-bidirectional 32-bit Windows environment cannot open Visual Basic 5.0 databases created with a Middle East sort order. The database must be compacted with General sort order in a bidirectional 32-bit Windows environment before it can be opened. This can be done programmatically using the `CompactDatabase` method or interactively using the Compact Database command of Microsoft Access 97 or later.

Hijri Calendar Support

Microsoft Visual Basic 5.0 and Microsoft Access for Windows 95-Arabic Edition (Arabic Access) or later contain support for both the Hijri and Gregorian calendars. The foundation for this capability is a new Calendar property in VBA and extensions to existing date-related VBA functions.

The following paragraphs describe how to perform calendar-related functions using Visual Basic 5.0. Unlike Arabic Access, which has user interface and other extensions beyond VBA to automate creation and management of databases based on either calendar, Visual Basic requires application developers to programmatically control the calendar associated with the entry, calculation, and display of date information.

- The Gregorian calendar is the default calendar for all date-related information and VBA functions in Arabic Visual Basic. To see or change the calendar setting to be used by VBA date-related functions, use the new VBA Calendar property. The syntax for Calendar property usage is:

```
[VBA.]Calendar [=value]
```

Items in brackets are optional. 'value' is one of the following settings or values.

Setting	Value	Description
vbCalGreg	0	(Default) Gregorian calendar used for VBA date-related calculations and formatting.
vbCalHijri	1	Hijri calendar used for VBA date-related calculations and formatting.

- When the calendar is Gregorian, the language of Gregorian month names is determined by the Date tab in the Regional Settings Control Panel of Microsoft Windows 95. When the calendar is Hijri, the language of Hijri month names is always Arabic. Hijri month names are always returned in full month name format.
- The behavior of all date-related functions in VBA is determined by the Calendar property setting. Functions affected by the Calendar property setting include CDate, Date, DateAdd, DateDiff, DatePart, DateSerial, DateValue, Day, Format, Month, Weekday and Year.
- The Visual Basic data type Date does not change when the calendar changes. Values of type Date are stored in a format that is not calendar-specific. This not only allows date information to be readily formatted for a particular calendar but also provides a convenient way to convert formatted date values from one calendar to another.

The following example creates a function to convert date strings between the supported calendar systems. The example code that follows uses the function to convert a Gregorian date string to the equivalent Hijri string in Long Date format.

```
Function ConvertDateString ( _  
    ByRef StringIn As String, _  
    ByRef OldCalendar As Integer, _  
    ByVal NewCalendar As Integer, _  
    ByRef NewFormat As String) As String  
  
    Dim SavedCal As Integer  
    Dim d As Date  
    Dim s As String  
  
    '// Save VBA Calendar setting to restore when finished  
    SavedCal = Calendar
```

```

// Convert date to new calendar and format
Calendar = OldCalendar      ' Change to StringIn calendar
d = CDate (StringIn)       ' Convert from String to Date
Calendar = NewCalendar     ' Change to calendar of new string
s = CStr (d)               ' Convert to short format String
ConvertDateString = Format _ ' Reformat
    (s, NewFormat)

// Restore VBA Calendar setting
Calendar = SavedCal
End Function

```

Call ConvertDateString function from a procedure to perform conversion...

```

Dim GregorianDate As String
Dim HijriDate As String
Dim HijriFormat As String

GregorianDate = "12/31/93" ' Gregorian string to convert
HijriFormat = "Long Date" ' Format for Hijri date

// Convert to Hijri date 7/8/1414 and return in Long Date format
HijriDate = ConvertDateString (_
    GregorianDate, _
    vbCalGreg, _
    vbCalHijri, _
    HijriFormat)

```

- To ensure Hijri dates strings are evaluated correctly in VBA statements, use the CDate function to evaluate the string (e.g., CDate("1/25/14")) instead of declaring the string as a constant (e.g., #1/25/14#). CDate will always return a value based on the active database calendar, whereas date constants are always evaluated using the Gregorian calendar.
- For reliable behavior, dates should generally be entered and displayed in an unambiguous format. For example, dates entered in short date format may be misinterpreted if the year or the day of the month are 12 or less (e.g. 3/11/10). For reliable behavior, use long date format whenever possible. You can alter the way a date is displayed in long format (e.g. exclude the day of the month) by changing the format in the Windows 95 Regional Settings Control Panel.
- If an existing database or file with date information is opened and the wrong calendar setting has been selected, Visual Basic may report a variety of errors or other unexpected behaviors. For example, if Calendar = vbCalHijri, Visual Basic may report an error when a statement makes reference to Gregorian dates that are invalid as Hijri dates (e.g., 'February 17, 1996' because no Hijri month has the name February).

Creating ActiveX Controls

Microsoft Visual Basic 5.0 includes features to create ActiveX controls that you can use in your applications. If you create an ActiveX control for your Middle East application, you may want to use the RightToLeft ambient property to improve its usability. Refer to the discussion of Ambient Property and Ambient Object in online Help for details about ambient properties.

An ActiveX control may inspect ambient properties provided by the control container to suggest behavior to controls. As an example, BackColor is one of the standard ambient properties; by providing this property, the container is suggesting what the well behaved control should set its back color property to.

The RightToLeft ambient property is also a standard ambient property defined in the ActiveX Controls Standard. All of the standard ambient properties are contained in the Ambient object provided by Visual Basic. The RightToLeft ambient property is used to communicate the RightToLeft setting of the control container to the ActiveX control. The default value for the ambient RightToLeft property is False.

The Form object is a typical container for ActiveX controls you create and provides ambient RightToLeft property information. If the form's standard RightToLeft property setting is True, then the RightToLeft ambient property will also be set to True.

By inspecting the RightToLeft ambient setting, an ActiveX control that is being newly created on a form can adapt its visual features and behavior to be consistent with the form on which it resides. For example, an ActiveX control that has a RightToLeft property can set the initial value of this property based on the ambient RightToLeft property setting of the form.

Creating Distribution Disk Sets

A key aspect of Visual Basic to developers of Middle East applications is its ability to seamlessly create run time versions of applications that include bidirectional feature support. However, there are a few points to keep in mind when creating distribution disks:

- Most bidirectional features of Middle East applications installed from distribution disks will run successfully only in a bidirectional 32-bit Microsoft Windows environment.
- The user interface for the Setup program included on distribution disks is English.

Accessing Bidirectional Features of Windows API's

Bidirectional features of Windows APIs are generally implemented as additional arguments and constants for standard APIs. The collection of API text files viewable with the API Text Viewer, particularly Win32api.txt, include references to arguments and constants for bidirectional features. Proper syntax is displayed in these files that you can copy and paste into your code.

These arguments and constants can be included in Windows API calls to any version of 32-bit Microsoft Windows. However, the API calls will only perform the requested bidirectional function if the operating environment is a bidirectional version of 32-bit Microsoft Windows, such as Arabic Microsoft Windows 95.

Creating Help Files

The Help Compiler Workshop included with Microsoft Visual Basic 5.0 can be used to create online Help files that display bidirectional text. Following are additional tips for creating Help files.

- The created files will display bidirectional text correctly only when run in a bidirectional 32-bit Microsoft Windows environment.
- To build an online Help file that displays bidirectional text, you must first create an .RTF format file containing bidirectional text and formatting attributes such as right-to-left paragraphs. Certain bidirectional-enabled word processors will automatically add this information to .RTF format files they create. Microsoft Word for Windows 95-Arabic Edition or later and other Middle East versions of Microsoft Word for Windows 95 are examples of such products.
- If the online Help file you create will be run on a platform other than a bidirectional version of 32-bit Microsoft Windows, be sure to specify the appropriate LCID language option in the Help project file. For example, if the file will be run on the U.S. English version of Windows 95, choose English as the language of the Help file.
- For each .RTF file compiled with Help Compiler Workshop, all Latin text in right-to-left paragraphs will have the same appearance regardless of the attributes that may have been assigned during file editing. This limitation causes the affected Latin text to be assigned the attributes (for example, font and font size) of the Default Latin font. This limitation does not exist for left-to-right paragraphs.
- Right-to-left tables, a standard feature of word processors such as Arabic Microsoft Word, are not a supported feature of Help Compiler Workshop. Help Compiler Workshop will compile a right-to-left table as if it is a left-to-right table.

Object Summary

Object	Bi-directional properties	State	Behavior	Set at design time	Set at run time
Form	RightToLeft	True	<ul style="list-style-type: none"> ▪ Menus and command names have an RTL reading order and are right aligned ▪ Form captions have an RTL reading order and are right aligned in the title bar. ▪ Scroll bars in MIDI forms appear in an RTL layout. If the Scrollbars property is set to: (1) a vertical scroll is placed on the left, (2) a horizontal scroll bar is placed with the scroll box on the right, (3) both 1 and 2 apply. ▪ Text that the Print method prints to the form begins at the form's right boundary with RTL reading order. 	X	X
Printer	RightToLeft	True	<ul style="list-style-type: none"> ▪ Printing begins at the right printer margin with RTL reading order. 	X	X

Control Summary

Control	Bidirectional properties	State	Behavior	Set at design time	Set at run time
CheckBox 	RightToLeft	True	<ul style="list-style-type: none"> ▪ Caption text reading order is RTL. ▪ Caption text is right aligned. 	X	X
	Alignment	Right Justify	<ul style="list-style-type: none"> ▪ Check box is positioned to right of caption. 	X	
ComboBox 	RightToLeft	True	<ul style="list-style-type: none"> ▪ Reading order of list box items and edit box text is RTL. ▪ List box items and edit box text are right aligned. ▪ If property Style is set to Dropdown Combo or Dropdown List, drop-down arrows are on left of combo box. 	X	
Command Button 	RightToLeft	True	<ul style="list-style-type: none"> ▪ Caption reading order is RTL. 	X	X
Data	RightToLeft	True	<ul style="list-style-type: none"> ▪ Order of navigation buttons 	X	X



flows RTL.

- Caption text reading order is RTL.
- Caption text is right aligned.

DBCombo



RightToLeft True

- Reading order of list box items is RTL and items are right aligned.
- Reading order of edit box text is RTL and text is right aligned.
- If the property Style=Dropdown Combo or Dropdown List, drop-down arrow is at left of combo box.

X

DBList



RightToLeft True

- Reading order of list box items is RTL.
- List box items right aligned.
- Vertical scroll bar appears at the left of the list box.

X

Frame

RightToLeft True

- Caption reading order is RTL.
- Caption is positioned at the upper-right corner of the frame.

X

X

Grid



RightToLeft True

- Grid columns begin at the right boundary of the grid.
- Fixed columns are located on the right side of the grid.
- LeftCol identifies the rightmost visible column.
- SelStartCol and SelStartRow together specify the cell in the upper-right corner of a selected range.
- SelEndCol and SelEndRow specify the cell in the lower-left corner of a selected range.
- If Scrollbars property is 3, a vertical scroll bar is placed on the left of the grid and a horizontal scroll bar (with scroll box on the right) is placed at the grid bottom.
- If Scrollbars property is 1, only a horizontal scroll bar is placed; if 2 only a vertical scroll bar is placed.
- Cell values (grid Text property) have RTL reading order.
- Data is automatically right aligned in each cell by FixedAlignment and

X

X

			ColAlignment properties to 1-Right aligned.		
			<ul style="list-style-type: none"> Cells filled with the Clip property fill from right to left, beginning at the upper rightmost selected cell. 		
	RightToLeft	True	<ul style="list-style-type: none"> Scroll box is placed at the right of the bar when the scroll box position value equals the Min property value. Scroll box is placed at the left of the bar when the scroll box position value equals the Max property value. 	X	X
	RightToLeft	True	<ul style="list-style-type: none"> Caption text reading order is RTL. 	X	X
	Alignment	Right Justify	<ul style="list-style-type: none"> Alignment property is set to Right Justify to right align caption text. 	X	
	RightToLeft	True	<ul style="list-style-type: none"> Reading order of list box items is RTL and items are right aligned. If the list box scrolls vertically and the list area is smaller than the number of items to be displayed, a drop-down arrow and vertical scrollbar appear at the left of the list box. If the list box scrolls horizontally, list box items are arranged in snaking columns. If the list area is smaller than the number of items to be displayed, a horizontal scrollbar appears at the bottom of the list box and scroll box position is at the right. 	X	
	RightToLeft	True	<ul style="list-style-type: none"> Grid columns begin at the right boundary of the grid. Fixed columns are located on the right side of the grid. Cell values have RTL reading order. ColAlignment property is set to 1-Right Aligned to right align data in each cell. LeftCol identifies the rightmost visible column (the first column beyond the leftmost fixed column). 	X	X

- If the ScrollBars property is set to 3, a vertical scroll bar is placed on the left of the grid and a horizontal scroll bar with the scroll box on the right is placed at the bottom. A ScrollBars property setting of 1 places only the horizontal scroll bar, while a setting of 2 places only the vertical scroll bar.
- Cells filled with the Clip property fill from right to left beginning at the upper rightmost selected cell.

Option Button 	RightToLeft	True	<ul style="list-style-type: none"> ▪ Caption reading order is RTL. ▪ Caption text is right aligned. 	X	X
	Alignment	Right Justify	<ul style="list-style-type: none"> ▪ Alignment property is set to Right Justify to place option button to the right of the caption. 	X	
PictureBox 	RightToLeft	True	<ul style="list-style-type: none"> ▪ Text reading order is RTL. ▪ Text begins at the right boundary of the picture box. 	X	X
TextBox 	RightToLeft	True	<ul style="list-style-type: none"> ▪ Text reading order is RTL. 	X	
	Alignment	Right Justify	<ul style="list-style-type: none"> ▪ Alignment property is set to Right Justify to right align text. ▪ If the MultiLine property is set to True, the Alignment property also determines scroll bar placement. If the Scrollbars property is set to: (2 or 3) a vertical scroll bar is placed on the left of the text box; (1 or 3) a horizontal scroll bar, with scroll box on the right, is placed on the bottom. 	X	

