

PRIORITY:	Normal
DATE:	March 22, 2001
TITLE:	Using the Adaptive™ Alpha® Marquee ActiveX® control in Microsoft® Word software.
ECO REFERENCE:	N/A
PRODUCT(S) AFFECTED:	All Alpha® electronic signs, except those with Smart Alec® or Gateway firmware EPROMS.
SUMMARY:	This document shows how you can use the Alpha® Marquee ActiveX® control, which has been developed for use with Alpha® Sign Communications Protocol, to program messages and communicate with Alpha® signs.

Introduction

The Adaptive™ Alpha® Marquee ActiveX® control is a development tool that increases the messaging capability of Alpha® electronic signs. This tool enables sign communication through software applications that support ActiveX®, such as Microsoft® Word.

The Adaptive™ Alpha® Marquee ActiveX® control also includes Message Manager, an accessory application that saves different types of messages in XML format and automates several messaging functions.

This TechMemo provides an overview of the steps required to insert the Adaptive™ Alpha® Marquee ActiveX® control into Word.

To create message instructions and schedule and trigger the display of messages on Alpha® signs you will need to create scripts in Visual Basic for Applications (sometimes shortened to “VBA”). Previous programming experience with VBA would be helpful, to reduce the slope of the message-scripting learning curve.

- ActiveX® controls allow programmers to encapsulate program functions (somewhat the same way that plug-ins or macros do), inside an ActiveX® “container”.
- With Word as the ActiveX® container, the Alpha® Marquee ActiveX® control permits communication with Alpha® signs and enables programmers familiar with Visual Basic for Applications (the programming environment for Word) to script messaging instructions for these signs, accessing data from Word documents.
- The Message Manager utility is provided with the control to simplify layout, setup and updating of messages. A single message file can hold many defined messages.
- Along with the creation and display of messages, the Alpha® Marquee ActiveX® control allows you to automate the presentation of real-time production statistics, broadcast events or alarm conditions, and control text attributes (such as color, font, and size). The Alpha® directly displays information as it is updated in your existing Word document.

© Copyright 2001 Adaptive Micro Systems, Inc. All rights reserved.

Adaptive Micro Systems • 7840 North 86th Street • Milwaukee, WI 53224 USA • 414-357-2020 • 414-357-2029 (fax) • www.adaptivedisplays.com

The following are trademarks of Adaptive Micro Systems: Adaptive, Alpha, AlphaNet plus, AlphaEclipse, AlphaPremiere, AlphaTicker, AlphaVision, AlphaVision InfoTracker, Automode, BetaBrite, BetaBrite Director, BetaBrite Messaging Software, Big Dot, PPD, Smart Alec, Solar, TimeNet

The distinctive trade dress of this product is a trademark claimed by Adaptive Micro Systems, Inc.

Due to continuing product innovation, specifications in this manual are subject to change without notice.

Adding the control to an Alpha[®] sign network

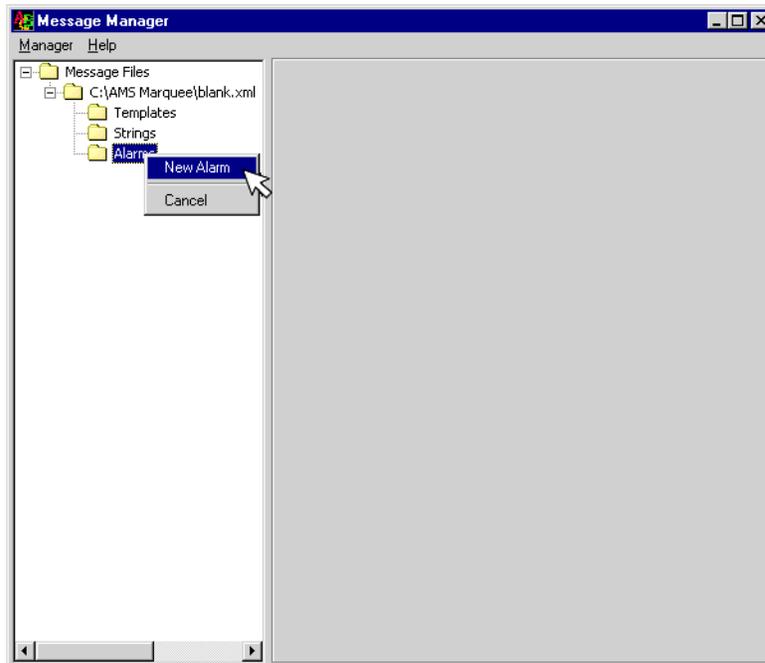
If you already have one or more Alpha[®] signs connected to your computer and want to improve your messaging capability with the Alpha[®] Marquee ActiveX[®] control, you must have the following:

- Microsoft[®] Internet Explorer version 5.0 (or newer) installed on your computer. Internet Explorer does not have to be set up as the default browser but the Alpha[®] Marquee ActiveX[®] control uses several libraries that are installed with Internet Explorer. The Alpha[®] Marquee ActiveX[®] CD contains all of the required Internet Explorer files. They can be copied to your hard drive during installation.
- Install the CD. See **Alpha[®] Marquee ActiveX[®] Installation Instructions** (9709-2050).
- The **ActiveX[®] Developer's Reference** (9709-2054), the complete guide to programming with the Alpha[®] Marquee ActiveX[®] control.
- A serial or a LAN (TCP/IP) connection to an Adaptive electronic sign with Alpha[®] protocol firmware. If your Alpha[®] sign hasn't been connected, refer to the Installation manual that came with the sign(s), the **Network Configurations** manual (9708-8046) and/or **Networking Alpha[®] signs on a TCP/IP network** (9708-8093).

Setting up a message folder

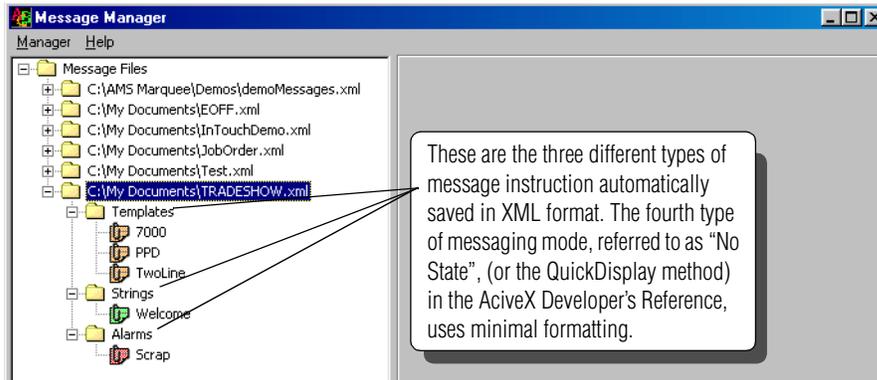
The Message Manager utility automates the task of message creation and storage. The Message Files folder at the top of the directory path is empty when the Message Manager utility is first installed on your computer.

1. Right-click Message Files (or use the *Manager* pull-down in the menu bar) to create, name and save a new Message File folder. The new file folder always includes sub-folders for three different types of message instruction—Templates, Strings, and Alarms.
2. Right-click any sub-folder and select *New* to open an active window to create and save a new message. (Refer to the **ActiveX[®] Developer's Reference** for complete and detailed information on programming messages, and the different types of message instruction.)



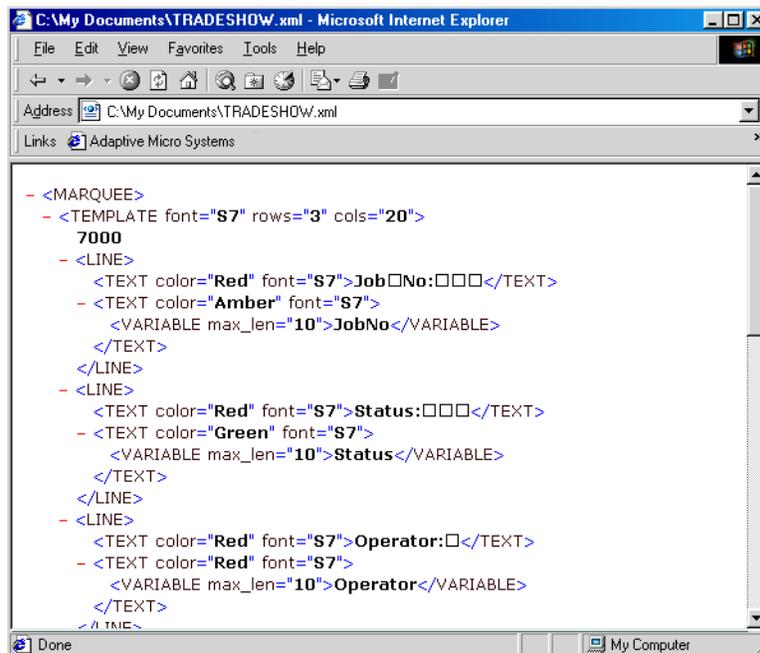
3. The folder that is highlighted in the following screen view, "Tradeshaw.XML", contains a sample folder with Template, String and Alarm message files created for three different types of signs (7000, PPD and two-line). All files are automatically saved in XML format.

- To see how messages in the sample folder have been integrated into a Word document, turn to "ActiveX®/Word messaging examples" on page 6. A single Message File folder can contain many individual messages, for multiple signs. Multiple ActiveX® controls may be used to direct messages to individual signs in a sign network.



- You can use Internet Explorer to view any of the created XML files, and also to check for errors if you suspect a file has been corrupted or damaged.
- The sample XML code only shows formatting parameters for the three message files in the templates folder of the Tradeshow.XML folder. The full XML file contains data corresponding to all of the message instructions (Templates, Strings and Alarms) that can be created in Message Manager..

NOTE: Internet Explorer 5.0 has a built-in XML parser. Go to the File pull-down menu, select *Properties* and click the *Analyze* button to see if a message has an XML error.



- A fourth category of message formatting through which the ActiveX® control can function is the "No State", or *QuickDisplay* mode. *QuickDisplay* does not use XML formatting created through Message Manager. Refer to the **ALPHA Marquee Control ActiveX® Developer's Reference** for more information.

NOTE: You can automate *QuickDisplay* messages by creating macros. See "Example 2—using a Word macro" on page 9. Messages created in *QuickDisplay* mode script all messaging text and basic formatting parameters in a single string. Since they are not saved as XML files, *QuickDisplay* messages will not appear in any *Message File* folder in Message Manager.

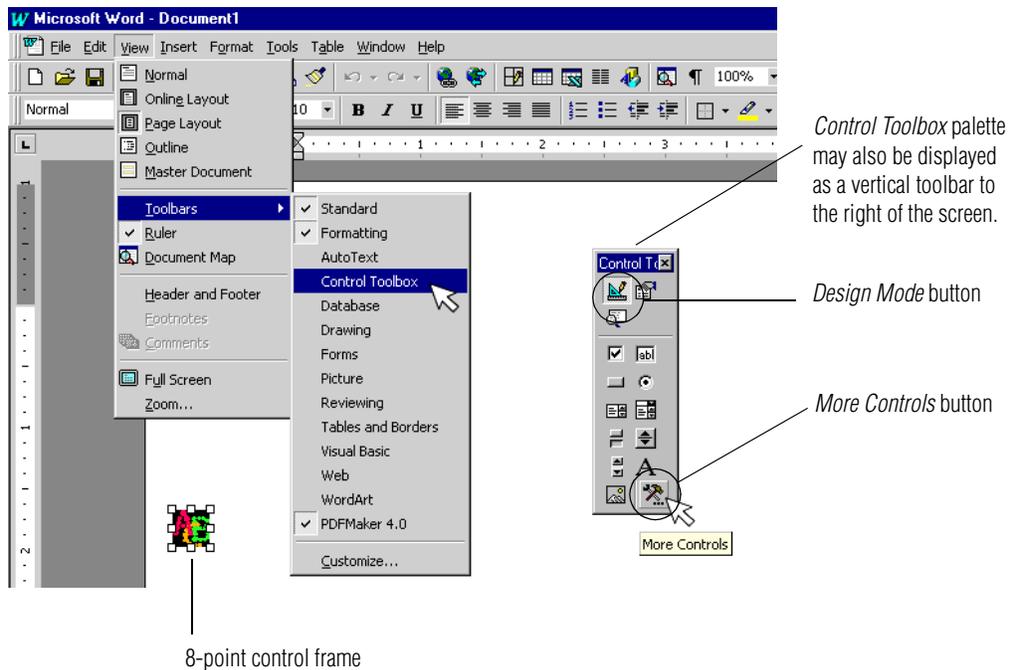
Inserting the Alpha[®] Marquee ActiveX[®] control into Word

Once a message file has been created you may insert the Alpha[®] Marquee ActiveX[®] control into Word by “dropping”, or physically placing an HTML tag in a Word doc file. (The Alpha[®] Marquee ActiveX[®] control must first have been installed on your computer, from the CD supplied with the Alpha[®] Marquee ActiveX[®] messaging software kit.)

You can use the *Control Toolbox* toolbar within Word to insert the HTML tag that references the Alpha[®] Marquee ActiveX[®] control:

1. If the toolbar isn't visible, you can display it by clicking *View — Toolbars — Control Toolbox* from the top pull-down menu, as shown in the screen view, below.
2. Select the control by left-clicking the *More Controls* button on the toolbar and highlighting *AMSMarqueeControl.Marquee* from the extensive list of available controls.
3. Load/install the control within Word by double-clicking your cursor anywhere in the document.
4. An 8-point control frame will appear on the document. (A sample control frame is shown in the lower left corner of the document, below.)
5. If the control frame shows a transparent object, you can make the HTML tag visible by slightly tugging a corner of the frame with the left mouse button.
6. The control has been embedded in the document.

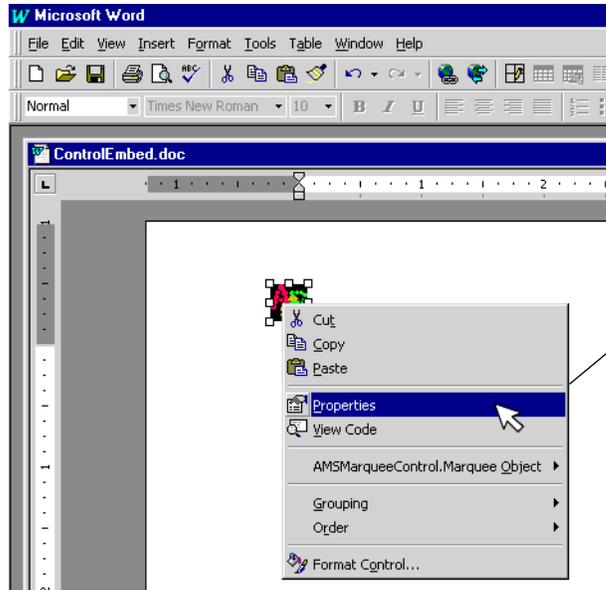
NOTE: You may insert more than one control into a document. You may choose or you may need to insert more than one control into a document in order to communicate with more than one Alpha[®] sign.



Setting up the Alpha[®] Marquee ActiveX[®] control

After the control has been inserted in the document configure communication properties:

1. Right-click on the 8-point ActiveX[®] HTML tag that was just embedded.
2. Select *Properties*. Configure values in the dialogue box that pops up to match your settings.



Right-click on the embedded ActiveX[®] control.

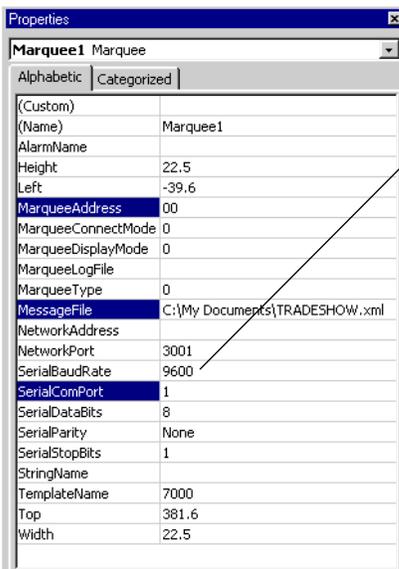
Select *Properties*.

NOTE: The ActiveX[®] HTML tag is only visible:

- when Word is in *Design Mode* (an option in the *Control Toolbox* palette/toolbar shown on page 4.) or
- when you open a Word doc file that has an embedded ActiveX[®] control with *Macros Disabled*.

3. If you will be communicating with more than one sign, multiple controls allow you to set individual address locations for each one. (See "Note", below.)

Serial configuration requires that you set an address, identify the message directory path/ filename, and select a com port.

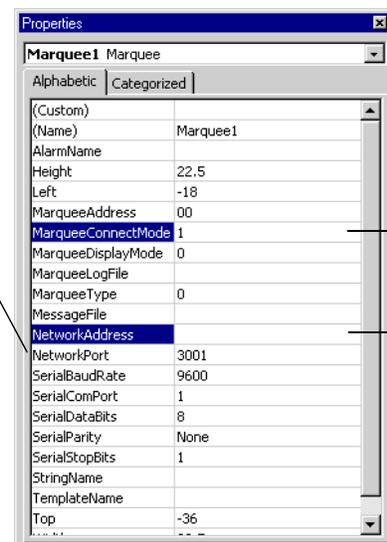


Default communications settings:

- SERIAL configuration should be left at default 9600/8/No (BaudRate/DataBits/Parity)
- TCP/IP configuration requires Network Port set to default 3001.

NOTE: If you have more than one sign, repeat the previous step and insert other controls into the Word document. Configure *Properties* to set up each new sign and a corresponding control, at a specific address (serial or TCP/IP).

A TCP/IP connection requires you to reset the MarqueeConnectMode from default "0" to "1". You must type in a Network Address.



ActiveX®/Word messaging examples

Once the ActiveX® control has been fully configured you are ready to create, control and update message displays. The application examples that follow have been scripted into a single Word document, "ControlEmbed.doc":

- Opening the document with Macros Enabled (below) allows your connected signs to access data in the document, if you set up controls to initialize as soon as the document opens. If your messages have been programmed to display data that is actively changing in the document, you will be able to observe text, values, or variables changing on your displays, as they update in Word.

ControlEmbed.doc

File Edit View Insert Format Tools Table Window Help

Normal Arial 12 B I U

Word ActiveX Example 1

Job Number: 123-45

Status: Active

Operator: Gino

Update Data

Values in the selected areas of the document are displayed on the connected ALPHA electronic sign.

If you open an existing Word document with "macros enabled" you will not see the control frame containing an HTML tag for any ActiveX® control(s).

You must select *Design Mode* from the *Controls* Toolbar/ Palette to make the ActiveX® controls visible.

Opening Word with *Macros Enabled*

If messaging data or formatting does not need to be changed, opening any Word file with "Macros Enabled" permits immediate data transfer between the file and any connected signs. The "7000 Series" template in our sample document displays on the sign just as you see it in the file. Keystroking changes into the Word text boxes changes the display.

Message Manager allows a programmer to select a specific sub-set of the total message display area, and use only that small area to display real-time updates of document data:

Job No: 123-45

Status: Active

Operator: Gino

ALPHA

The Word template (above and at right) corresponds exactly to the display on the 7000 sign.

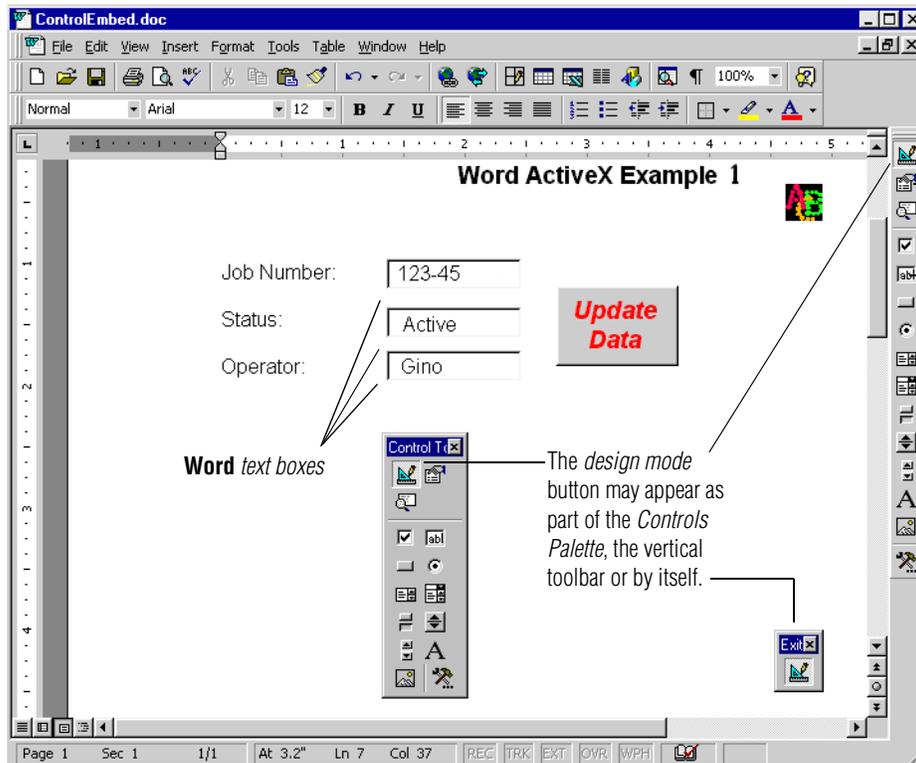
See "Example 1—programming a Template" on page 8

Opening Word in *Design Mode*, or with Macros Disabled

If message formatting requires changing, or if you must import other ActiveX® controls into your document to communicate with new signs, select *Design Mode* from the Controls Palette or open your Word document with Macros Disabled.

When you are programming VBA scripts, Word can pull data from any part of the document. For example, in the message on the 7000 series sign (preceding illustration), the values for “job number/status/operator” are updated through *text boxes* that have been inserted into the document. Any Word document element can be

“Example 1—programming a Template” on page 8 for an example that uses the Message Manager template shown in this Word document.



— Toggling to *design mode* activates the *view code* button.

— Clicking *view code* takes you directly to the programming environment for Word! Visual Basic for Applications, allowing you to see everything that's been programmed into the document and make any needed changes.

— You can insert *text boxes*, *list boxes* or other Word elements to expedite your message scripting.

Example 1—programming a Template

1. Set up and create a new template file in Message Manager by right-clicking on the *Template* folder and selecting New Template. A default Template window opens up.
2. Make editing changes from the default values.
3. Click the horizontal Save bar in the bottom of the window to save your new file in XML format.

In the Template file that has already been set up below all of the values designated by an “x” in the Template update to a specific area of the sign. This corresponds to data that changes in the *text boxes* (or other elements) of the Word document.

Right-clicking the folders and selecting *New* allows you to create, edit and save new *String* and *Alarm* (XML) files. These are the same steps that are used in this example to create a new *Template* file.

The Row/Column layout defined in Message Manager can not be larger than the number of rows and characters that can be displayed on the sign. Example shown is for a 3-line display.

Clicking the horizontal bar at the bottom part of the window saves your file in XML format.

The Visual Basic code corresponding to the template, above, is included below (along with scripting that is used in Example 2, on the next page).

```

WordActiveX - ThisDocument [Code]
Document
Open
Public Flag As Integer ' Declare Flag as a global variable
Private Sub CommandButton1_Click() 'Runs when the "Send" button is clicked
    ThisDocument.Marquee1.QuickDisplay (TextBox4.Text) 'Uses control's QuickDisplay method
    'to send a message without variables
    'to a sign.
    Flag = 1 'Sets the flag indicating that QuickDisplay was used.
End Sub

Private Sub CommandButton2_Click() 'Runs when the "Update Data" button is clicked
    Call Update(Flag) ' Runs when the "Update Variables" button is pressed
End Sub

Private Sub Document_Open()

    Flag = 0 ' Sets the flag to 0 when the document is initially opened

    Marquee1.Initialize ' Initialize the Control
    Marquee1.TemplateName = "7000" ' Specify template to use
    Marquee1.ChangeMode ("0") ' Set the "template" mode

    JobNo = TextBox1.Text
    Call Marquee1.UpdateVariable("JobNo", JobNo) ' Update variable "JobNo" in the template
    'with JobNo from this script

    Status = TextBox2.Text
    Call Marquee1.UpdateVariable("Status", Status)
    Operator = TextBox3.Text
    Call Marquee1.UpdateVariable("Operator", Operator)

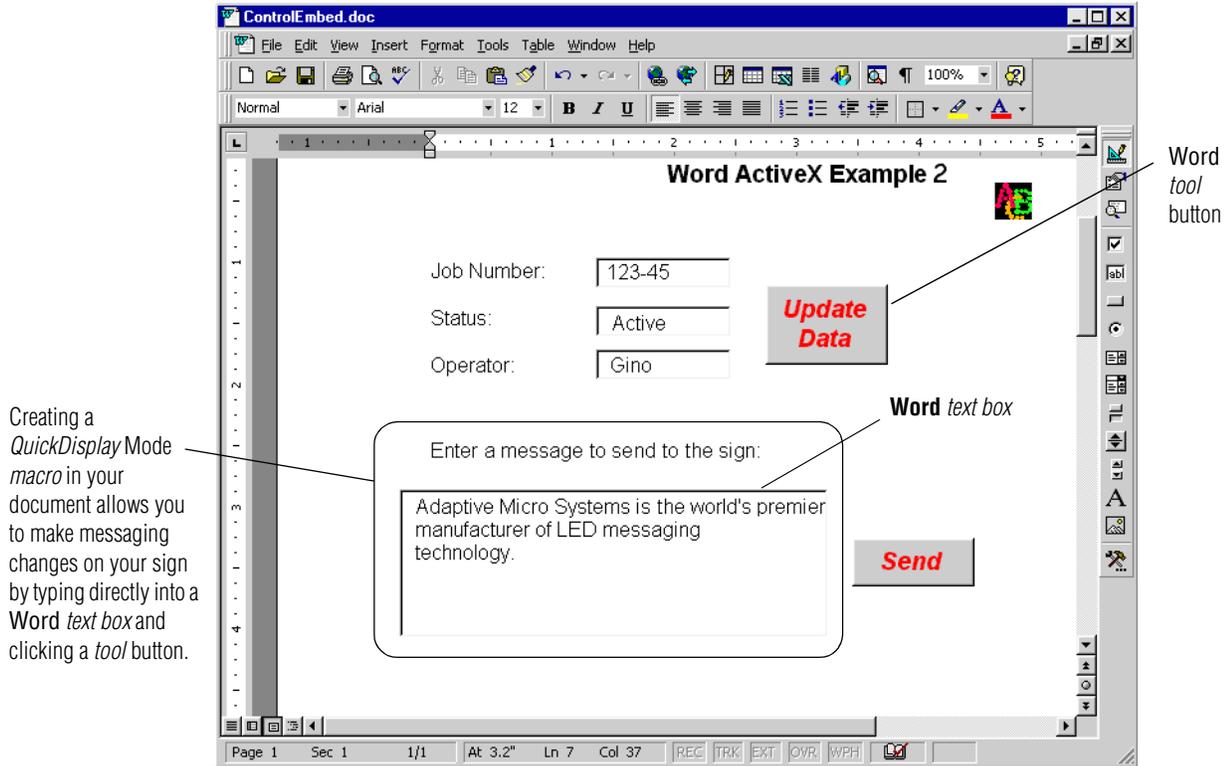
End Sub
    
```

NOTE:
Clicking the "View Code" button in the Word Control Palette brings up the programming environment for the Word file, Visual Basic for Applications.

Example 2—using a Word macro

You can send a message to a display directly (without creating a template, string, or alarm) by using a macro that triggers the No State *QuickDisplay* ActiveX[®] mode. The message below has been scripted that way. *QuickDisplay* mode messages don't require XML formatting. In the sample message the content of the display is all text. The ActiveX[®] interface allows text characters and minimal display information to be directly transmitted to the sign.

NOTE: This messaging method does not allow you to use variables or trigger alarms.



The *Update Data* button (shown above) activates the *Module1* macro in Visual Basic for Applications, shown below. Clicking the *Send* button (above) activates the script at the top of the window in the *View Code* screen, at the bottom of the preceding page.

NOTE: It is not necessary to declare variables in the VBA script having the same names as message templates. It was done in this case to simplify the example.

Refer to the **ActiveX[®] Developer's Reference** for information on strings and alarms.

```
WordActiveX - Module1 [Code]
(General) Update
Sub Update(Flag As Integer) ' Runs when called by "Update Data" button click
    If Flag = 1 Then ' Check to see if Flag was set (i.e. QuickDisplay was used)
        ThisDocument.Marquee1.ChangeMode ("0") ' Reset the mode to "template", makes the
            ' display blank for a second
        Flag = 0 ' reset the Flag
    End If

    'Update variables
    JobNo = ThisDocument.TextBox1.Text
    Call ThisDocument.Marquee1.UpdateVariable("JobNo", JobNo)
    Status = ThisDocument.TextBox2.Text
    Call ThisDocument.Marquee1.UpdateVariable("Status", Status)
    Operator = ThisDocument.TextBox3.Text
    Call ThisDocument.Marquee1.UpdateVariable("Operator", Operator)

End Sub
```