

Using the Alpha Marquee ActiveX control in Excel

The Alpha Marquee ActiveX control can be used in any software package capable of using ActiveX technology. These programs are commonly referred to as ActiveX Containers. Most Human Machine Interface (HMI) packages are such containers. Microsoft's Word, Excel, and Access are also designed to accept ActiveX. This document will explain how to apply our ActiveX control in the Excel environment.

I. *Setting up a Message File*

The first step in communicating with an Alpha sign is to set up a message file that will contain Strings, Templates, and Alarms that are to be displayed. The message file can be created in one of two ways. The easiest approach is to use the Message Manager that comes with the AMS Marquee ActiveX (See Figure 1.)

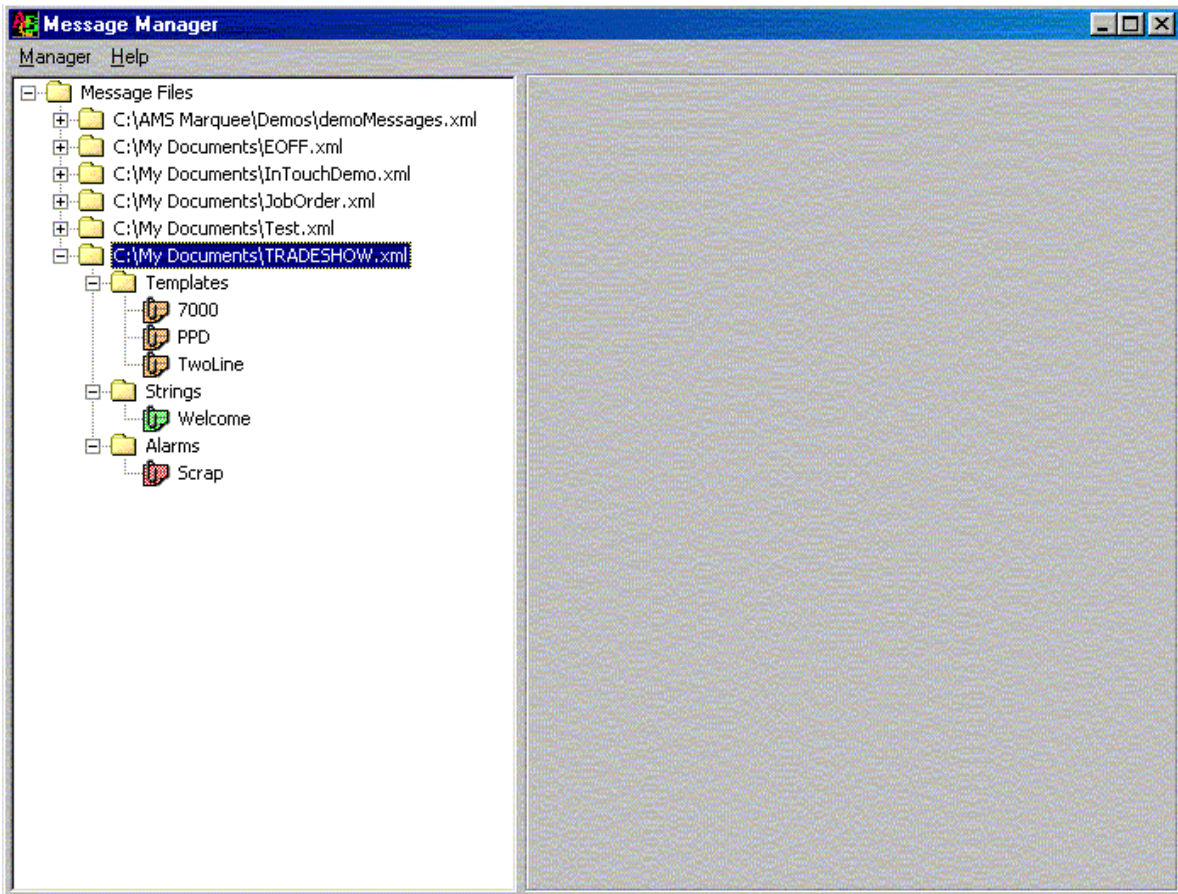


Figure 1.

Create a new message file by right clicking on the "Message Files" folder displayed and choosing "New Message File", or using the Manager option on the menu bar. Once the new file is saved, it will appear in the Message Manager. You will note that the file created is in the .XML format. The second option for creating the message file deals with writing XML code directly. Most text editors can modify this code. Internet Explorer can be used to view XML files (See Figure 2).

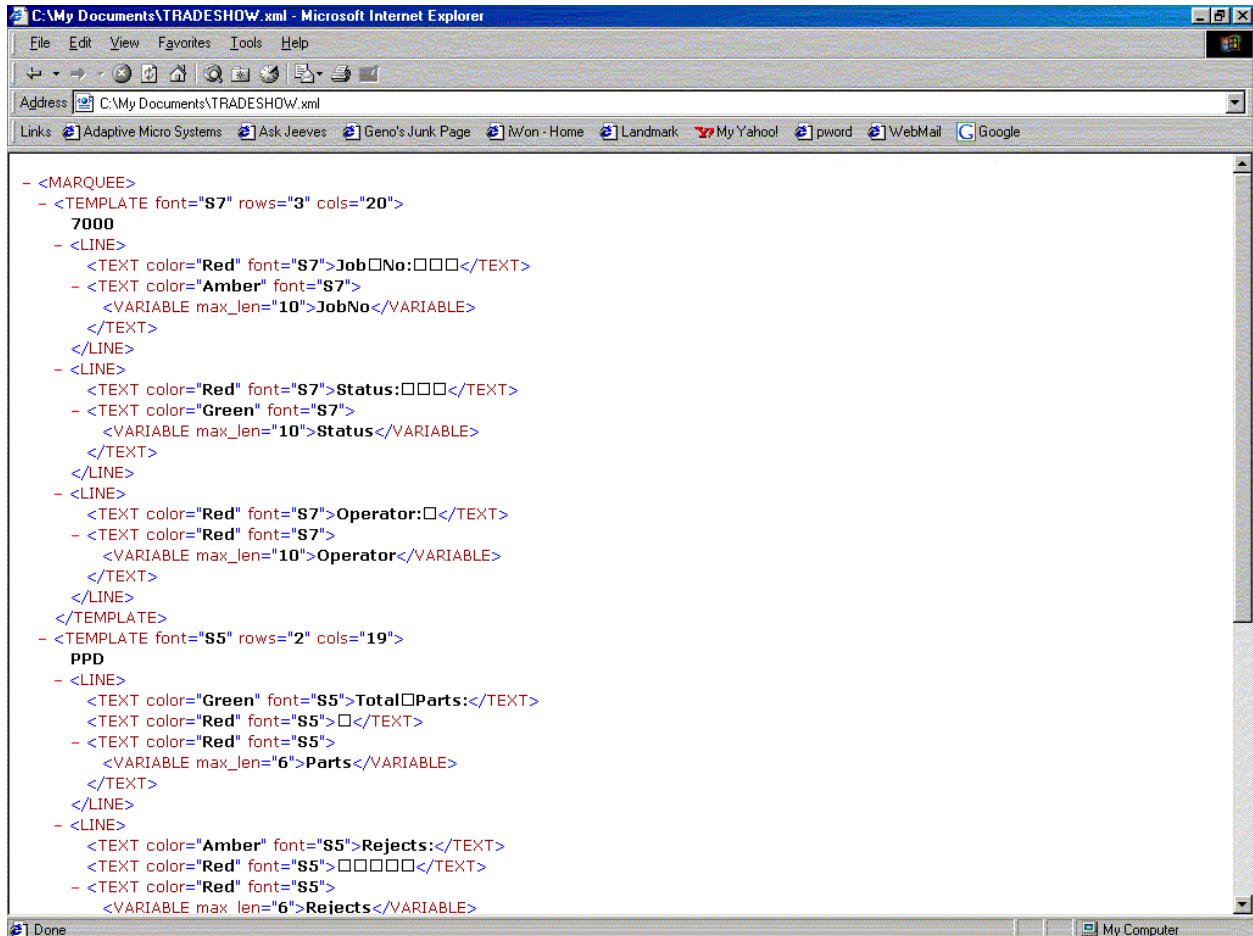


Figure 2.

Clicking on the plus sign next to the new folder will reveal three additional folders: Templates, Strings, and Alarms (See Figure 1). Depending on the display needs, create these items following the Alpha™ Marquee Control ActiveX Developer’s Reference. Please make sure to document the names of the templates, strings, alarms, and variables used. These will be referred to from Excel.

II. Inserting the AMS Marquee Control into Excel

Once the message file has been set up, the control can now be dropped (inserted) into an Excel spreadsheet. Please note that the following procedure assumes that the AMS Marquee ActiveX control has been properly installed on the computer.

The easiest way to insert any ActiveX control into a spreadsheet is from the “Control Toolbox” toolbar. If this toolbar is not displayed, activate it by clicking on View – Toolbars – Control Toolbox on the Excel menu bar. Insert the control by clicking the “More Controls” button on the toolbar (See Figure 3) and choosing the AMSMarqueeControl.Marquee option from available controls.

Select an area anywhere in the sheet you would like to drop the control. Your control is now embedded in the sheet. Following is an explanation of the set up and the VBA scripting required to use the control.

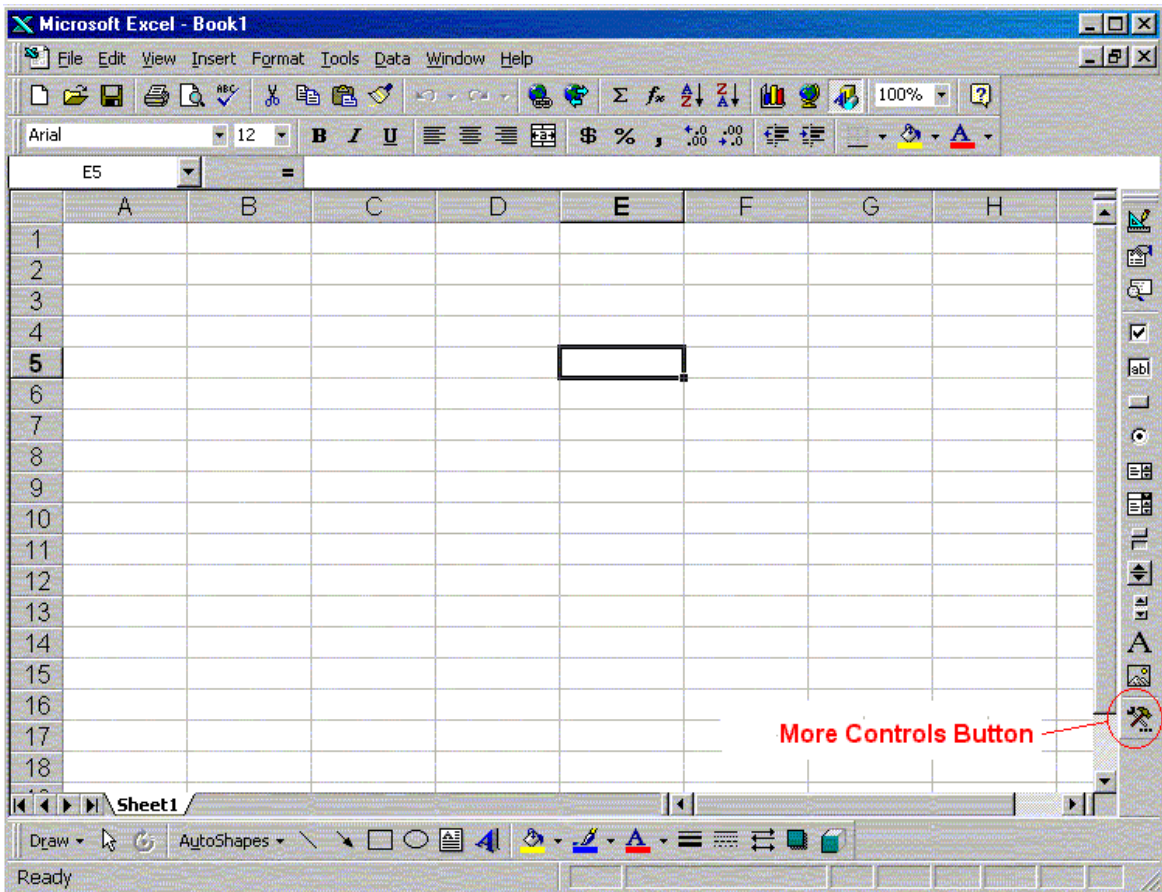


Figure 3

III. Control Setup

The first thing to do after the control has been embedded in the worksheet is to configure the communications. Right click on the control and choose AMSMarqueeControl.Marquee_Object and click on the "Properties" option (the only available option). Use the dialog box that appears to choose the message file created in step I above, select the correct marquee address, and choose the appropriate COM port (See Figure 4).

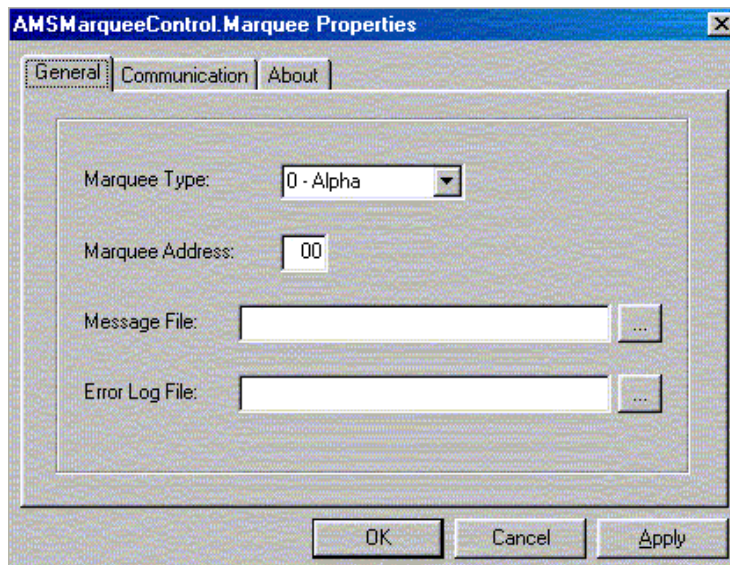


Figure 4.

Please note that the control will be invisible, if Excel is not in the Design Mode.

The example used here will simply update three variables declared in a template. The control will look at the values in the cells A7, B7, and C7 (see Figure 5) and update the variables on the display. Please note that the example below is intended for at least two-line displays (e.g. 4000, 7000 series) with 20 characters minimum.

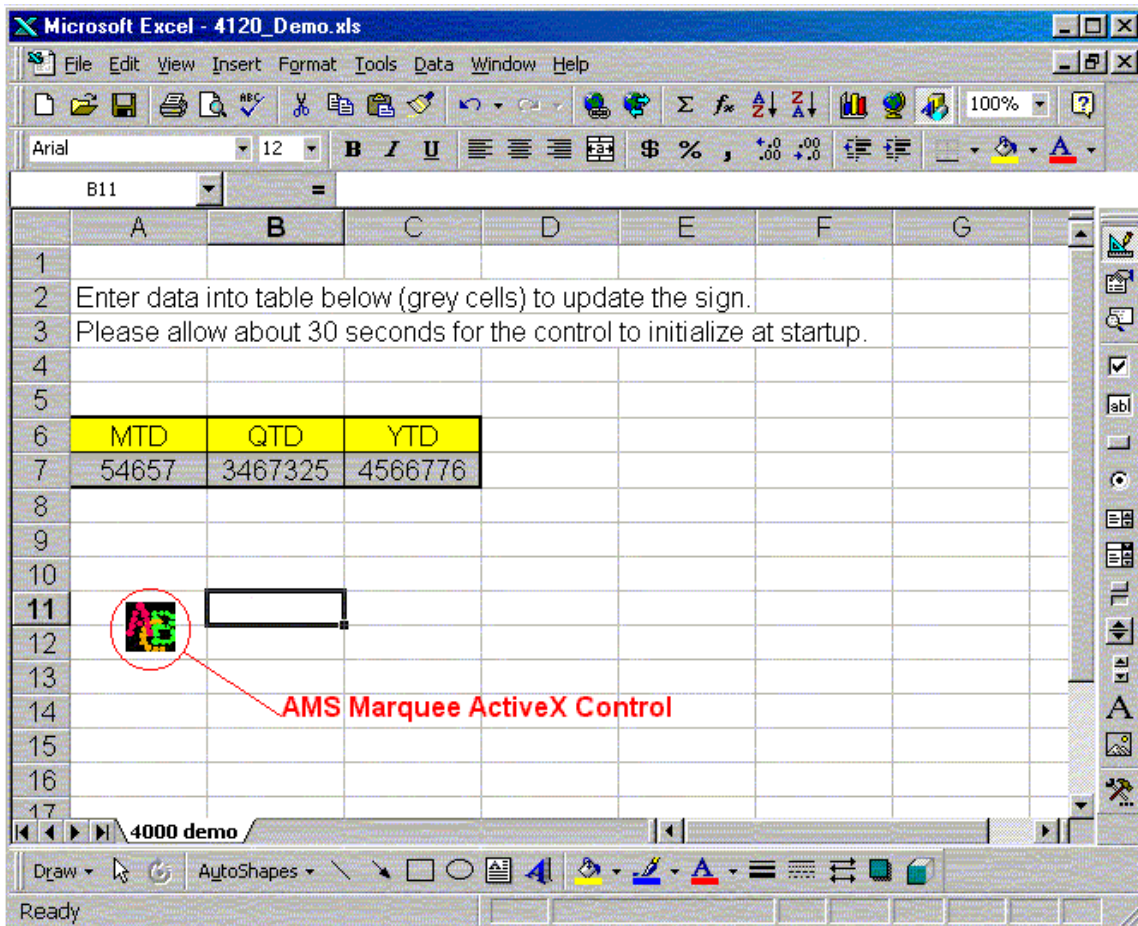


Figure 5.

Figure 6 illustrates what the template looks like. Please note that the number of characters available on the display will determine the setup of the template and the length of the message that can be sent to it. Sending a string longer than allowed by the template setup will truncate the message. For example, in our setup, variable "MTD" is limited to five characters while "YTD" will allow a maximum of seven.

When the information is updated on the sign will depend greatly on the application. For this example, it was assumed that the sign would be updated when the workbook is open. Once the workbook is open, the marquee will then update whenever there is a change on the particular worksheet.

Figure 7 shows the script required to update the sign on opening the workbook. The code to update the sign after a change is illustrated on Figure 8.

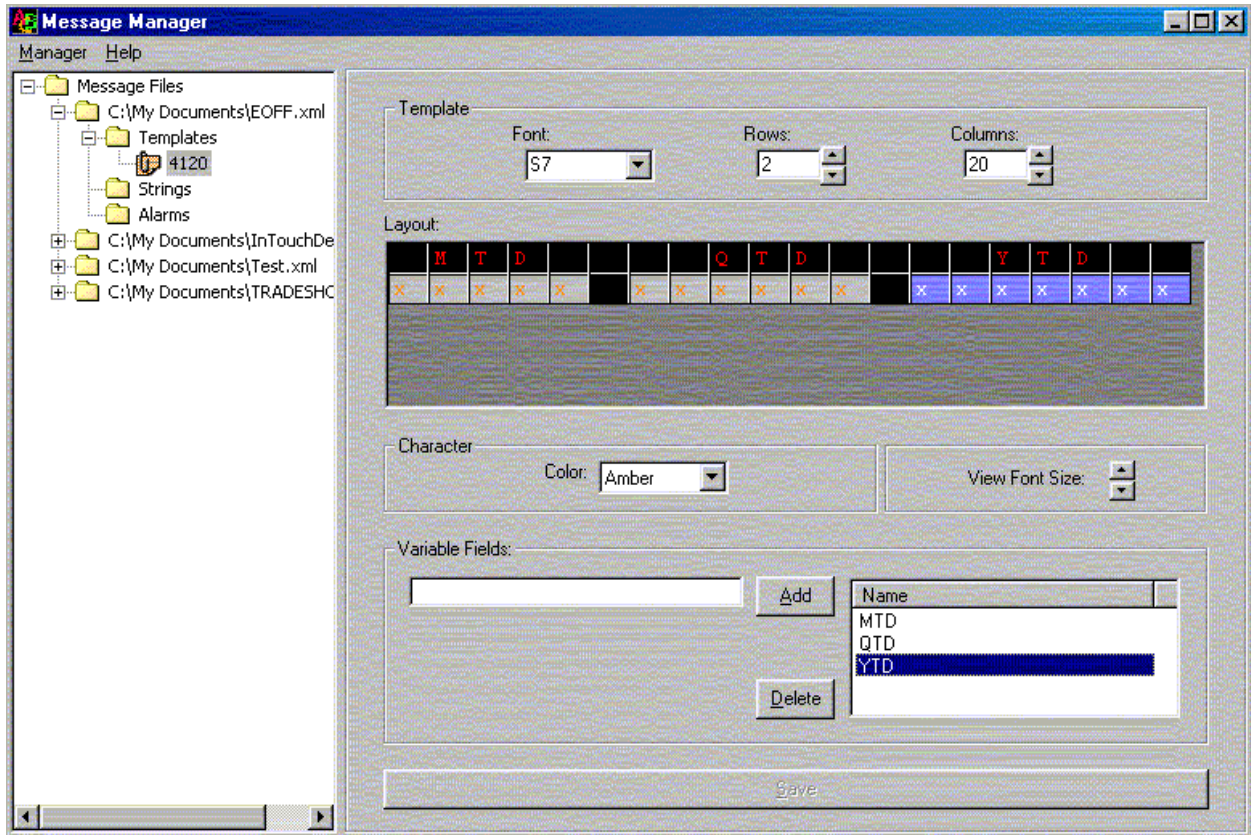


Figure 6.

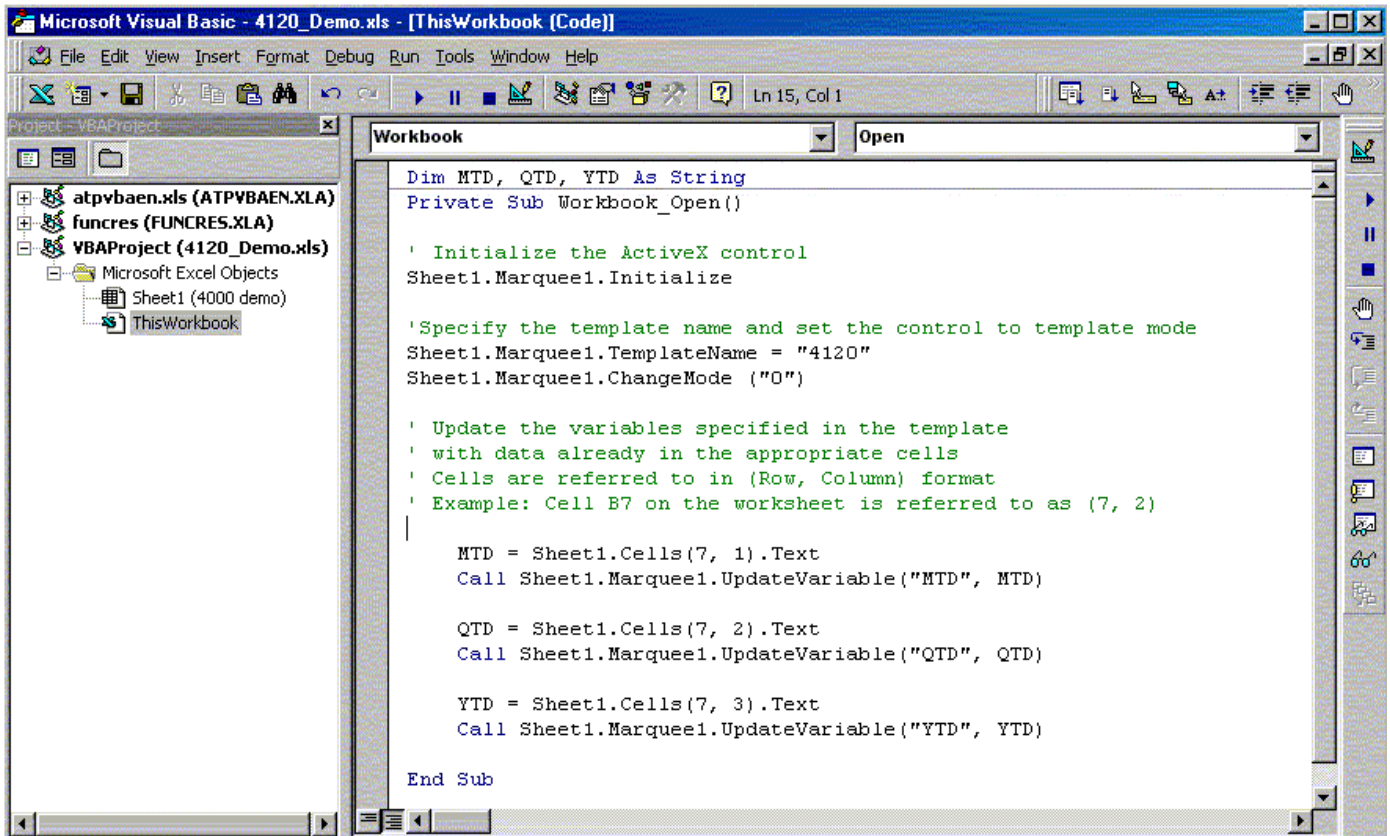


Figure 7.

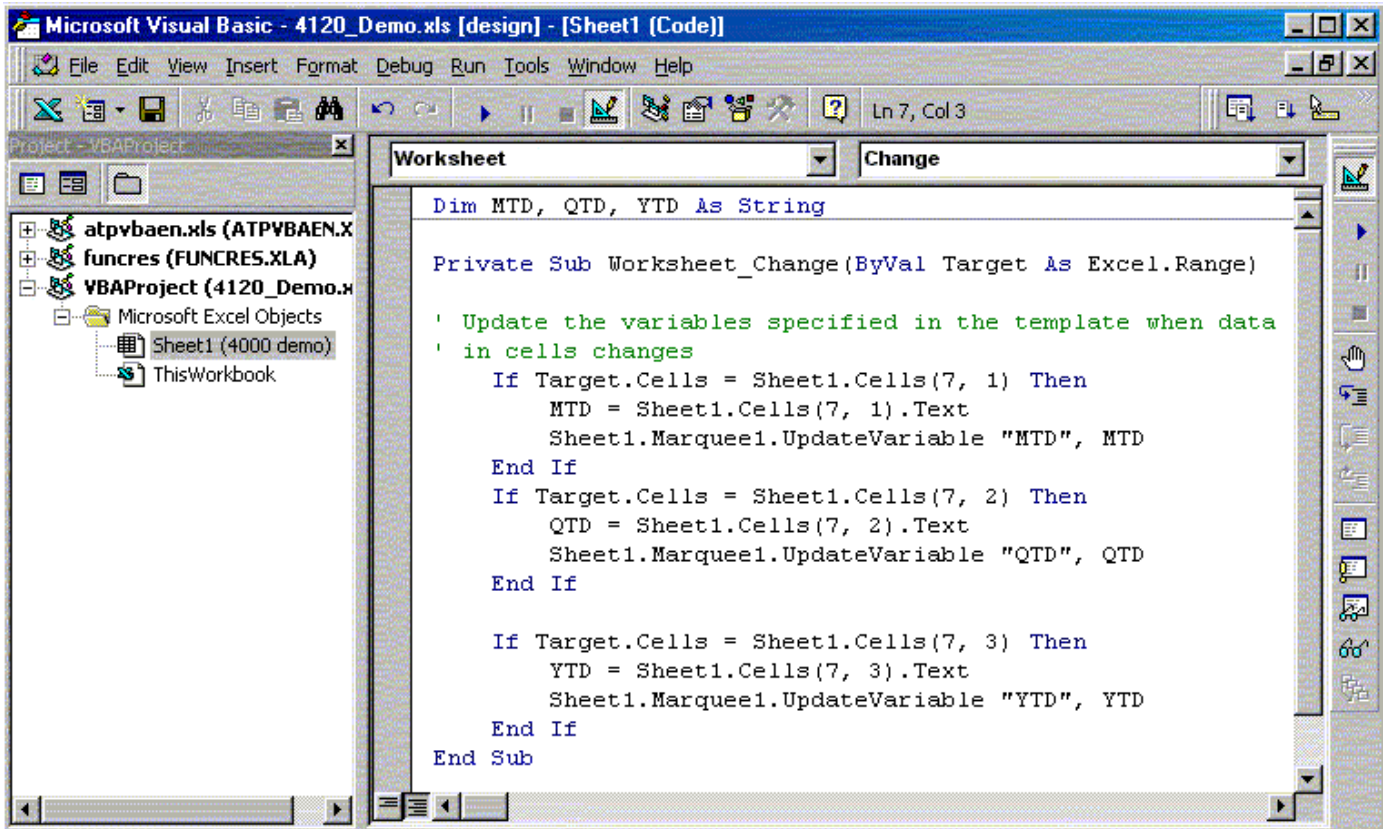


Figure 8.

Please note that it is not necessary to declare variables in the VBA script with same names as the message template. It was done in this case to simplify the example.

While this example covered only template manipulation, triggering strings and alarms is accomplished in much the same manner. Please refer to the ActiveX Developer's Reference for additional methods and properties.