Training cards implementation:

The three main components in the training card implementation are 1) the activex controls in the html files that issue a WM_TCARD message, 2) the WithEvents object that raises a TCard event when the WM_TCARD message is received and 3) the event handler which calls some user defined code when a TCard event is raised.

Step one in the implementation is to declare the constant short that will be recognized as a training card windows message.

```
Private Const WM_TCARD As Short = &H52
```

Step two: Declare the WinHelp function from the HTML Help API

```
Private Declare Function WinHelp Lib "user32" Alias "WinHelpA" (ByVal
hWnd As Integer, ByVal lpHelpFile As String, ByVal wCommand As Integer,
ByVal dwData As Integer) As Integer


Private Declare Function HTMLHelp Lib "hhctrl.ocx" Alias "HtmlHelpA" _
    (ByVal hWnd As IntPtr, _
     ByVal lpHelpFile As String, _
     ByVal wCommand As Long, _
     ByVal dwdata As Long) _
    As IntPtr
```

Step three in the implementation is to get the training cards started by displaying an initial card with activex control buttons added to it. So by virtue of a button or menu command call the HTML help function. See Figure 1.

Figure 1

In this example, when a user clicks on a training card type such as Basin Design or Border Design the showInitialCard subroutine is called. This brings up the initial training card for the Design Seminar.

```
'Show the first card in a design seminar sequence.
    Overloads Sub showInitialCard(ByVal hwndCaller As Form)
        Dim hwnd As IntPtr
        hwnd = HTMLHelp(hwndCaller.Handle,
"..\..\BorderDesign\DesignSeminar.chm::\TrainingCard1.htm",
HH_DISPLAY_TOPIC, 0)
    End Sub
```

To call the HTMLHelp function you must pass it three parameters. The first parameter is the handle (*hwnd*) of the window calling HtmlHelp(). The second parameter is the path to the .chm file that contains the training card you wish to display and a reference to that

training card. The third parameter is an HTML HELP API command. In this example the HH_DISPLAY_TOPIC command is used which enables you to open a compiled help (.chm) file in a specific help window and display a specific topic within the help file. In this example the .chm file is named DesignSeminar and we tell it to display the training card TrainingCard1.htm.

Once the initial card is displayed it should have some mechanism for calling other training cards. In this example we use previous and next buttons so the user can cycle through a series of training cards. See Figure 2.

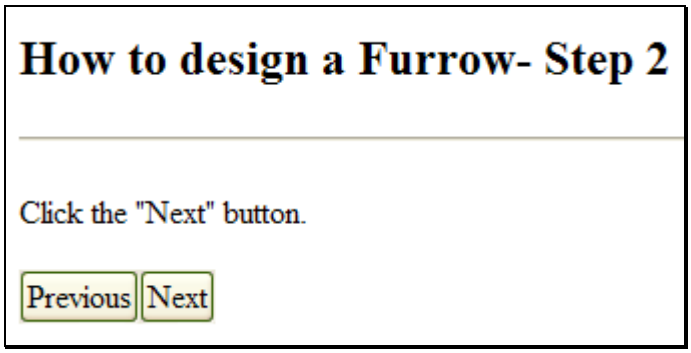

Figure 2

Each button is an activex control embedded in an html file. The code for the Previous button looks like the following.

```
<OBJECT[1] id=hhctrl type="application/x-oleobject"
      classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11"
      codebase="hhctrl.ocx#Version=5,2,3735,0"
      width=100
      height=100>
   <PARAM[2] name="Command" value="TCard">
   <PARAM name="Button" value="Text:Previous">
   <PARAM name="Item1" value="050200">
</OBJECT>
```

The Command “name” value in the first PARAM tag is the type of message the button will send when clicked. The Button “name” value is the text that appears on the surface of the button. The Item “name” value is the integer value that will be transmitted as the

[1] The Object element tag supplies the browser with information to load and render data types that are not natively supported by the browser. If the browser must load some external program (a Java applet, a plugin, or some other helper) the information about the content that is to be rendered is contained by the OBJECT element. Its attributes and optionally associated PARAM elements nested inside of it.

[2] The PARAM element may be nested within an APPLET or OBJECT element to pass parameters to the Java apple or object as it is being loaded. Parameters provide ways for HTML authors to adjust settings of an applet or object without having to recode either one. A parameter typically passes a name/value pair which is assigned to the NAME and VALUE attributes. You can have more than one PARAM element per applet or object.

WM_TCARD message. This integer is interpreted by a subroutine which then decides which training card to display.

To handle the WM_TCARD event, in the application, declare a class to raise an event when a training card (WM_TCARD) message is issued. In this example we created the TEVENTClass. Within this class is the event generated when the WM_TCARD event is found, TCardEvent, and a method to raise the TCARD event, CauseTCardEvent. Finally we have a WithEvents object, TEvent that is used to raise the TCardEvent.

```
Private Class TEventClass
        ' Event generated when a WM_TCARD is found
        Public Event TCardEvent(ByVal tCard As Int32)

        ' Method to raise event
        Sub CauseTCardEvent(ByVal tCard As Int32)
            RaiseEvent TCardEvent(tCard) ' Raise TCard event.
        End Sub
    End Class


    WithEvents TEvent As New TEventClass ' Object to raise TCardEvent
```

Also create an event handler to handle the TCardEvent.

```
  Private Sub TCardEvent_Handler(ByVal tCard As Int32) Handles
TEvent.TCardEvent
        Dim hwnd As IntPtr
        Me.verifyDefaultValues(tCard, Me)
    End Sub
```

In the main application, override the WndProc to catch Windows messages and raise the TCardEvent when a WM_TCARD message is received.

```
'*********************************************************************
***
    ' WndProc() - override of WndProc for intercepting Windows messages
    '
    Protected Overrides Sub WndProc(ByRef m As Message)
        '
        ‘Intercept all Windows messages prior to system handling;
        ‘look for training card messages
        '
        Select Case (m.Msg)
            Case WM_TCARD ' Training card message

                ' Get Training Card ID from WPARAM
                Dim tCard As Int32 = m.WParam.ToInt32

                ' Don't handle here; send it on as an event
                TEvent.CauseTCardEvent(tCard)
        End Select
        '
```

```
        ' Call base class method to continue Windows message processing
        '
        MyBase.WndProc(m)
    End Sub
```

When the WM_TCARD event is recognized the TCardEvent is raised. The event handler is then called. In this example, the event handler calls a function named "verifyDefaultValues." This subroutine takes the tcard 32 bit integer as a parameter and interprets it to decide which training card to display. Once interpreted, again the HTML Help method is called to access the .chm file and display a particular training card encased inside.

```
'This method will ensure the user enters the correct values for the
selected project.
    Overrides Sub verifyDefaultValues(ByVal tCard As Int32, ByVal
hwndCaller As Form)
        Dim hwnd As IntPtr
        Select Case (tCard)

        Case 50109  'Card(01) Next(09)
                hwnd = HTMLHelp(hwndCaller.Handle,
"..\..\FurrowDesign\FurrowDesign.chm::\TrainingCard2.htm",
HH_DISPLAY_TOPIC, 0)
            Case 50200 ' Card(02); Previous(00)
                hwnd = HTMLHelp(hwndCaller.Handle,
"..\..\FurrowDesign\FurrowDesign.chm::\TrainingCard1.htm",
HH_DISPLAY_TOPIC, 0)
            Case 50209  'Card(02); Next(09)
                hwnd = HTMLHelp(hwndCaller.Handle,
"..\..\FurrowDesign\FurrowDesign.chm::\TrainingCard3.htm",
HH_DISPLAY_TOPIC, 0)
            Case 50300 ' Card(03); Previous(00)
Etc…
```

Each training card is organized in the following manner.

```
tCard value encoding:  ddccbb  dd = deck; cc = card; bb = button
```

Thus in TrainingCard1.htm (50109) is the next button of the first card in the fifth deck. 9 is used to designate the next button and 0 for the previous button. Therefore the html code for the TrainingCard1 next button would contain the following in the PARAM tag.

<PARAM name="Item1" value="050109">

We use the next button case to point to the card that is NEXT in the sequence of pages, so in this case TrainingCard2.htm.

```
Case 50109  'Card(01) Next(09)
                hwnd = HTMLHelp(hwndCaller.Handle,
"..\..\FurrowDesign\FurrowDesign.chm::\TrainingCard2.htm",
HH_DISPLAY_TOPIC, 0)
```

In this example, if the Next button is clicked on TrainingCard1, 50109 is passed as the tCard parameter and TrainingCard2 will be displayed. If the Previous button is clicked on TrainingCard2, 50200 is passed as the tCard parameter and TrainingCard1 will be displayed. However 50200 is the SECOND card in the deck “02” and the button is a PREVIOUS, “00.”

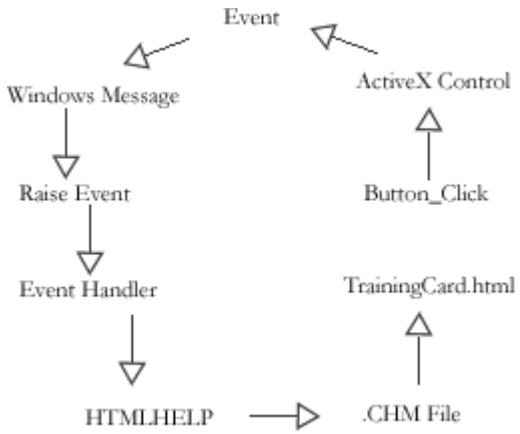

Figure 3