

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

General Principals

When any user is on the system the workspace is live.

Each ASPX file opens a unique Namespace/Class when initiated and the function Page_Load is called.

```
:Class TCC_Login : Page
:Using System.Web.UI, System.web.dll
:Using System.Web.UI.HtmlControls
:Using System.Web.UI.WebControls
:Using System.Data, system.data.dll
:Using System
:Using System.Type
:Using System.Globalization

:Access public

    ▽ Page_Load;system
    A test function
      :Access Public
      :Signature Page_Load Object obj, EventArgs ev
ETC.....
```

As part of the :Class statement all .NET namespaces

At this point it is easy to get confused between .NET namespaces and APL namespaces. These are 2 different things, thought up in two different worlds by people who (I guess) have never met.

Each APL function that needs to be called by ASP.NET requires an :Access Public statement and a :Signature statement

Global variables can be created in the workspace and used by any function called by the application.

Any files created by a process will exist for ASP.NET calls to the workspace.

Session variables can be used and assigned by both the ASP.NET program and the APL workspace or any other active .NET process.

**Techniques used in the LASER system:
APL driving ASP.NET**

version 1.0

Initiating the Workspace.

When a user first calls the LOGIN.ASPX page the LASER system will check if the workspace is live i.e. one or more other users are currently attached.

If not initiate the following

Tie all required files:-

Create Global variable:

Tie values for each file tied to the workspace

Directory Variable showing file director path

Matrix of users currently attached to the workspace

Information in the variable includes:-

User ID,

Security level,

The IP address

A welcome message

Last activity of the user

Create session variables :-

User ID

Status

System

Security Level

This must be done within a :Hold [CS" to prevent 2 or more users attempting to attach at the same time.

Once the workspace is initiated these items are used to manage the user session and insure that no-one circumvents the security of the system. I.e Information held in the Session Variables to be consistent with the information in the Matrix of users.

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Function calls from ASP.NET:

```
<%@ Page Language="Dyalog" - Language used (VB)
      src="Dyalog/Training.dws" - The APL workspace
      Inherits="Login" %> - The Namespace
<%@ Import Namespace="System.IO" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
```

In the APL workspace the Page_Load function will be called:-

▽ Page_Load;system

⌈ test function

:Access Public

:Signature Page_Load Object obj, EventArgs ev

:If 0=⊖NC'#.dir' ⌈ Check if variable is available

Initiate system

:EndIf

#.utilityfns←⊖FREAD#.ftie,11 ⌈ Get standard utility function

⊖FX`#.utilityfns ⌈ Define function into namespace

Session[⊖'UserID']←⊖0

Session[⊖'Login']←⊖0

Session[⊖'Back']←⊖''

Session[⊖'Status']←⊖'OK'

⌈ Identify version of the system you are currently running in from the Directory Path

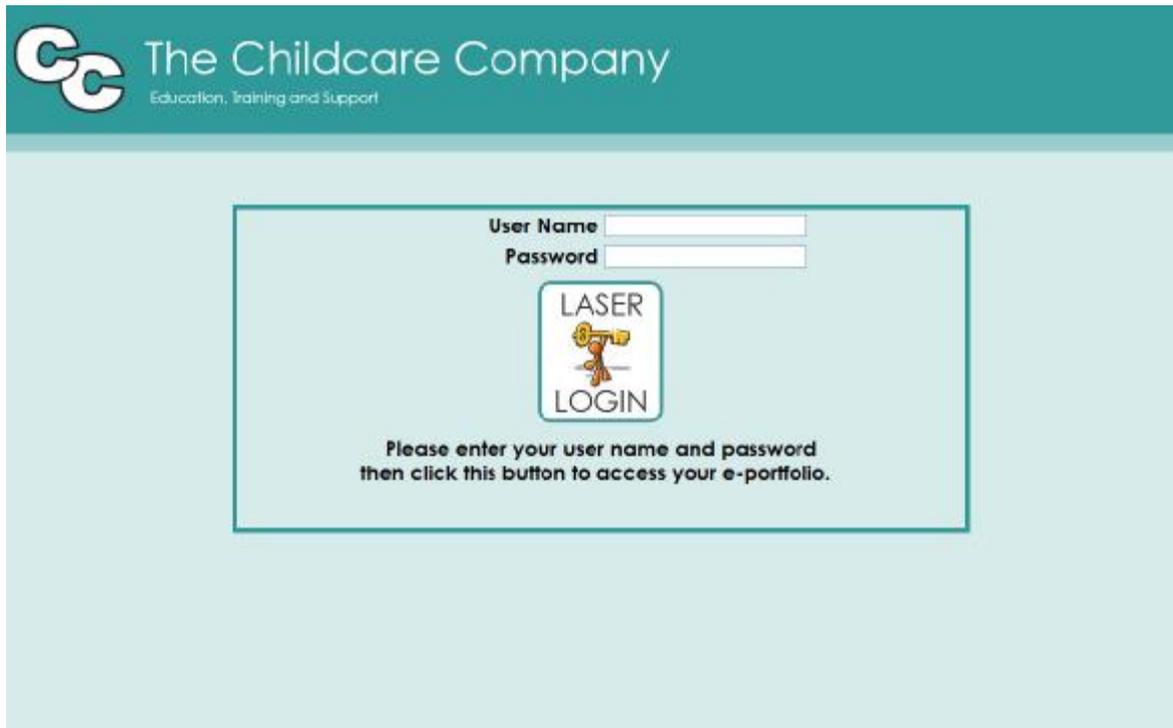
system←1↑(∨/'live' 'test' 'demo' ⅉ(⊖'#.dir'))' 'Test System' 'Demo System'

Session[⊖'System']←system

▽

Techniques used in the LASER system: APL driving ASP.NET

version 1.0



The screenshot shows a web page for 'The Childcare Company' with the tagline 'Education, Training and Support'. The main content is a login form titled 'LASER LOGIN'. It features two input fields: 'User Name' and 'Password'. Below the fields is a button with a key icon and the text 'LASER LOGIN'. A message below the button reads: 'Please enter your user name and password then click this button to access your e-portfolio.'

```
<table>
  <tr bordercolor="#D7EBEB">
    <td align=right valign="middle" >User Name</td>
    <td align=left>
      <asp:TextBox id="Username" runat="server" Wrap="false"
        Width="200px" Font-Names="Century Gothic" />
    </td>
  </tr>
  <tr bordercolor="#D7EBEB">
    <td align=right valign="middle">Password </td>
    <td align=left>
      <asp:TextBox ID="Password" runat="server" TextMode="Password"
        Wrap="false" Width="200px" /> </td>
  </tr>
  <tr bordercolor="#D7EBEB">
    <td valign="middle" colspan="2" >
      <asp:ImageButton ID="Login" runat="server"
        OnClick="Login_Click"
        ImageUrl="~/images/laserlogin.jpg"
        ImageAlign="Middle" />
    </td>
  </tr>
  <tr>
    <td bordercolor="#D7EBEB" style="height: 21px" colspan="2" >
      Please enter your user name and password <br />
      then click this button to access your e-portfolio.
    </td>
  </tr>
</table>
```

MORE later...

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

In APL the clicking of the Login Button executes the function Login_Click

All APL object parameters can be accessed and set by the APL function from simple things like writing text to the page:-

```
Object.Text ← 'Text Message'
```

To (as we will see later) make something Visible or Invisible on the screen.

```
Object.Visible ← 1 ⌘ Make Visible
```

```
Object.Visible ← 0 ⌘ Make Invisible
```

```
▽ Login_Click;text;UserID;coursecode
:Access public
:Signature Login_Click Object, ImageClickEventArgs
FUNPACK #.ftie,12 ⌘ Get Login Functions
User_Error.Text←''
Pass_Error.Text←''

:If 0≠pΔRB Username.Text
:Trap 0
  Session[<'TCC_User']←CheckUser(Username.Text Password.Text)
  UserID↔Session[<'UserID']
  :If 0≠Session[<'UserID']
    Session[<'Login']←1
    coursecode↔#.online_users[#.online_users[;2]⌈UserID;1]
    Response.Redirect(<'MainMenu.aspx')
  :EndIf
```

MORE later....

In the function CheckUser if the password is incorrect:

```
TB_Pass_Error.Text←'Invalid Password'
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Continuation of ASP.NET program....

```
<tr bordercolor="#D7EBEB">
  <td valign="middle" align=center style="width: 320px">
    <asp:TextBox
      ID="User_Error"
      runat="server"
      ReadOnly="True"
      BackColor="#D7EBEB"
      Width="300"
      BorderStyle="None"
      BorderWidth="0px"
      Height="30px" Font-Bold="True"
      Font-Size="Large"
      ForeColor="Red">
    </asp:TextBox>
  </td>
  <td valign="middle" align=center style="width: 320px" >
    <asp:TextBox
      ID="Pass_Error"
      runat="server"
      ReadOnly="True"
      BackColor="#D7EBEB"
      Width="300"
      BorderStyle="None"
      BorderWidth="0px"
      Height="30px"
      Font-Bold="True"
      Font-Size="Large"
      ForeColor="Red">
    </asp:TextBox></td>
</tr>
</table>
```

Continuation of APL program....

```
    User_Error.Text←'ERROR: '⎕EXCEPTION.Message
:EndTrap
:Else
    User_Error.Text←'Please enter user name.', Username.Text
:EndIf
▽
:EndClass
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Actions from Buttons



Text fields are populated by the Page_Load function and any button attributes can also be set. For example if new mail is available then the Inbox Icon should spin. To achieve this re-assign the image button attributes ImageUrl with a motion GIF as opposed to a static GIF.

In ASP.NET

```
<td><asp:ImageButton ID="inbox"
runat="server"
OnClick="inbox_Click"
ImageUrl="~/images/inbox_man_90.gif"
/></td>
```

In APL (Page_Load Function)

```
:If NewMail
    inbox.ImageUrl←'~/images/InBox_Motion.gif'
:EndIf
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

When the button is clicked you can:-

Execute an APL Function	<code>OnClick="inbox_Click"</code>
Start another ASP.NET Program	<code>PostBackUrl="~/New_Operation.aspx?ur=764766"</code>
Start A local Java Scrip funtion	<code>OnClick=</code> <code>"javascript:JS_OpenLesson('help/laserhelp.pdf')"</code>

You need to use Javascript if you wish to open the new window, e.g in the case above you want to open the Help Document but not close the current ASP.NET page.

Please also note that the helpful and friendly user messages are set from the Page_Load function:-

Inside Page_Load APL Function:

```
Greeting.Text←User_Name  
Assessor.Text←'Your Assessor is: ',Assessor_name
```

If the button executes an APL function that function must be defined in the class and must have Public Access and the correct Signature:

```
▽ inbox_Click;text;UserID  
:Access public  
:Signature inbox_Click Object, ImageClickEventArgs  
Response.Redirect(<'Mailbox.aspx')
```

▽

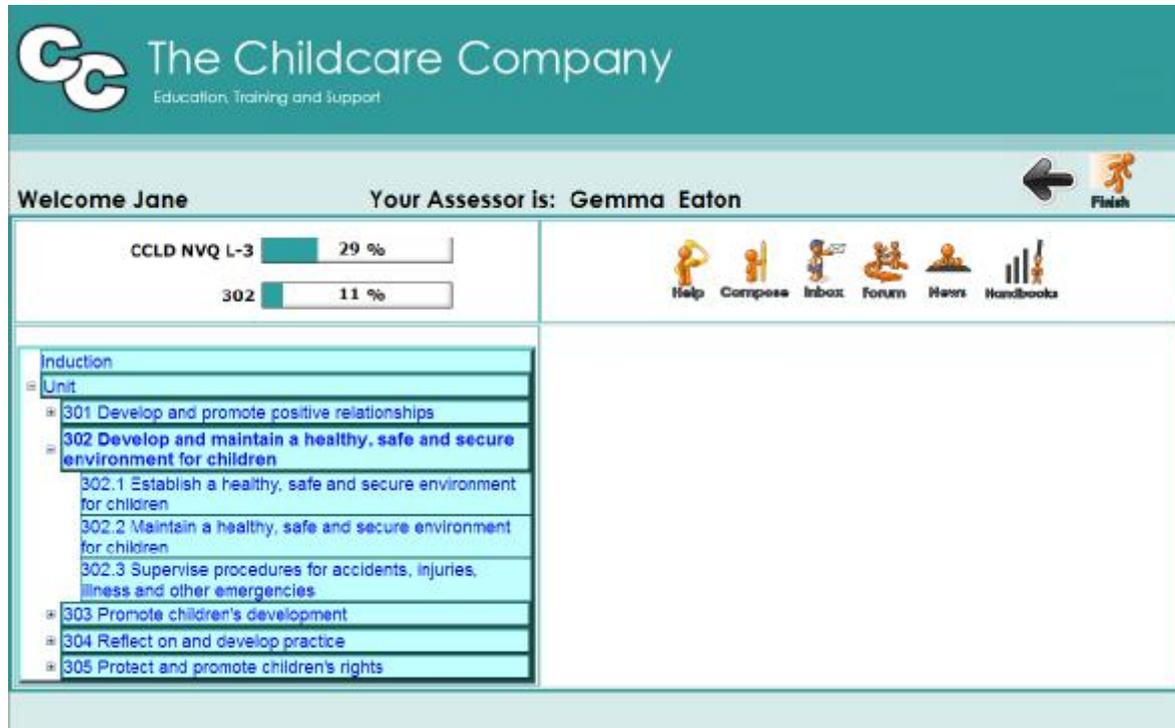
Please note that when the button is not an image the arguments of the signature statement must be as follows:

```
▽ Click_SAVE args;sender;e  
:Access public  
:Signature Click_SAVE Object sender, EventArgs e  
sender e←args
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Using Flash Objects



An Flash Plugin program was created to show the percentage image on the left under the greeting. The program as designed to take in 2 parameters, a title and a percentage value.

In this example the flash program is called progressBar.swf and has 2 input variables:

passedPercent = A number between 0 and 100
passedText = A text string.

ASP.NET can have an embedded Flash program, but the trick is to get APL to pass the text string and the percentage value

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

In ASP.NET:-

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/s
wflash.cab#version=7,0,19,0"
style="width: 380px; height: 50px">
  <param name="movie" value=
    <% Response.Write(Session("BAR1")) %> />
  <param name="quality" value="high" />
  <embed src=<% Response.Write(Session("BAR2")) %>
Quality="High"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash"
width="380" height="50"></embed>
</object>
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/s
wflash.cab#version=7,0,19,0"
style="width: 380px; height: 50px">
  <param name="movie"
value=<% Response.Write(Session("BAR2")) %> />
  <param name="quality" value="high" />
  <embed src=<% Response.Write(Session("BAR2")) %>
Quality="High"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash" width="380" height="50">
  </embed>
</object>
```

The Session variables are set from with inside the APL Page_Load Function

In APL:-

```
flashcall1←'progressBar.swf?passedPercent=',⊕GetTotalPercent UserID
flashcall1,←'&passedText=', GetCourseName UserID
```

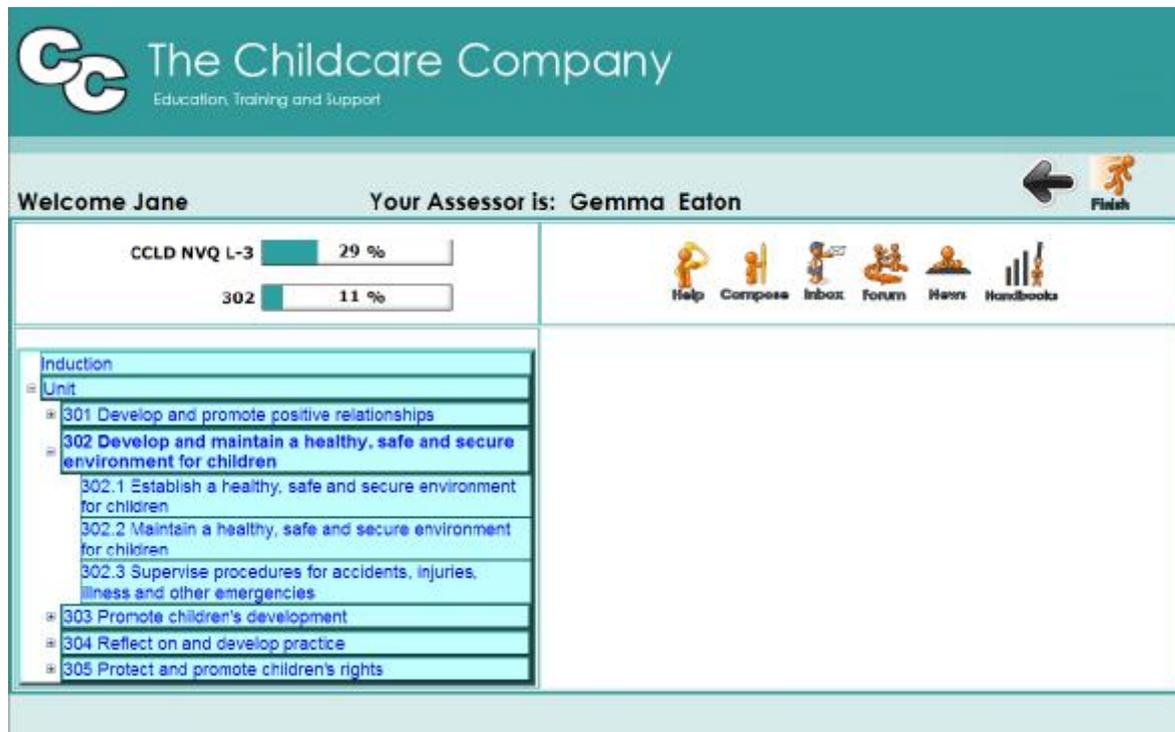
```
Session['BAR1']← flashcall1
```

```
Session['BAR2']← flashcall2  Ⓐ calculated in a similar way as flashcall1
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Table Views



The screenshot shows the user interface of 'The Childcare Company' web application. At the top, the logo and name 'The Childcare Company' are displayed, along with the tagline 'Education, Training and Support'. Below this, a navigation bar includes 'Welcome Jane', 'Your Assessor is: Gemma Eaton', and a 'Finish' button. The main content area is divided into two columns. The left column contains progress bars for 'CCLD NVQ L-3' (29%) and '302' (11%). The right column features a set of icons for 'Help', 'Compose', 'Inbox', 'Forum', 'News', and 'Handbooks'. Below these, a treeview is expanded to show a list of units under the 'Induction' category. The units listed are: 'Unit', '301 Develop and promote positive relationships', '302 Develop and maintain a healthy, safe and secure environment for children' (with sub-items 302.1, 302.2, and 302.3), '303 Promote children's development', '304 Reflect on and develop practice', and '305 Protect and promote children's rights'.

Treewiews have Nodes and Leaves where the nodes down from a defined number of levels can expand and contract, revealing leaves at any level.

In ASP.NET

```
<asp:TreeView Id="MyTree"
SkinId="Explorer"
PathSeparator = "|"
OnTreeNodePopulate="PopulateNode"
OnSelectedNodeChanged="MyTree_Select"
OnTreeNodeCollapsed="MyTree_Collapse"
OnTreeNodeExpanded="MyTree_Expand"
ExpandDepth="1"
runat="server"
BorderColor="#80FFFF"
BorderStyle="Outset"
Font-Size="Large"
Width="520px"
NodeWrap="True">

<Nodes>
```

See Later.....

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

The above statement contains the attribute:-

```
OnTreeNodePopulate="PopulateNode"
```

This indicates that the .NET program will call a function PopulateNode in the APL class and this program will populate the tree structure.

```
▽ PopulateNode(source e) ; local vars
:Signature PopulateNode Object,TreeNodeEventArgs
:Access public

A Some code to define data
A Where data has 2 columns,
    The Text to me displayed
    The Value to be returned

:For line :In data
    newNode←NEW TreeNode(line)
    :If level=1
        newNode.PopulateOnDemand←1
        newNode.SelectAction←TreeNodeSelectAction.Expand
    :EndIf
    node.ChildNodes.Add newNode
:EndFor
▽
```

The three stributes:-

```
OnSelectedNodeChanged="MyTree_Select" A Select
OnTreeNodeCollapsed="MyTree_Collapse" A Callapse
OnTreeNodeExpanded="MyTree_Expand" A Expand
```

Indicate which APL functions are executed

```
▽ MyTree_Select(source e);Locals
:Signature MyTree_Select Object,EventArgs
:Access public
A Make visible what needs to be visible and invisible what must
be invisible
    lesson.Visible←1
    attach.Visible←1
    PCItemsGrid.Visible←1
    induction.Visible←0
    Induction.Visible←0

    item←MyTree.SelectedNode.Value A Get the node value
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

In MyTree_Collapse and MyTree_Expand no code is required to expand or collapse the tree, as this is all handled by the ASP.NET and **PopulateNode**. However some objects may need to be made invisible or visible.

```
▽ MyTree_Collapse(source e);item;user
:Signature MyTree_Collapse Object,TreeNodeEventArgs
:Access public
lesson.Visible←0
attach.Visible←0
induction.Visible←0
... ETC
```

Back in ASP.NET just to finish off the NODE definition:

```
<Nodes>
  <asp:TreeNode Text="Induction"
    PopulateOnDemand="True"
    Value="Induction"
    SelectAction="Expand"/>

  <asp:TreeNode Text="Unit"
    PopulateOnDemand="True"
    Value="Unit"
    SelectAction="Expand"/>
</Nodes>
<HoverNodeStyle Font-Bold="True" />
<NodeStyle BackColor="#C0FFFF"
  BorderColor="#339999"
  BorderStyle="Outset"
  BorderWidth="3px"
  Width="500px" />
<LeafNodeStyle BackColor="#C0FFFF"
  BorderColor="#339999"
  BorderStyle="Ridge"
  BorderWidth="1px"
  Font-Size="Large" />
<ParentNodeStyle BackColor="#C0FFFF" />
</asp:TreeView>
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

View Grid

Performance Criteria/Knowledge Specification	Work Note	Rate
302.3.1 Make sure that accidents, injuries, signs of illness and other emergencies are promptly identified		
302.3.2 Follow the correct procedures to deal with accidents, injuries, signs of illness and other emergencies calmly and safely	4	
302.3.3 Make sure that yourself and others are not put at unnecessary risk		
302.3.4 Provide comfort and reassurance to those involved	3	
302.3.5 Make sure that first aid and medication are provided, according to the correct procedures		
302.3.6 Follow the correct procedures for recording and reporting accidents, injuries, signs of illness and other emergencies		
K3P189 Regulations covering manual handling and the risks associated with lifting and carrying children		
K3H190 The basic stages of child development and the implications		

In the example of the learner view once a leaf has been selected the GridView and some extra buttons on the right of the screen are made visible.

You will see above that these are made visible by the APL function `MyTree_Select`. This function goes on to populate the gridview with the data for the correct selected item, in this case a Work Element in the training course.

The GridView is defined in the ASP.NET file.

The Gridview itself contains buttons that will open new ASP.NET program, these buttons can have static GIFs or Motion GIFs depending if information for that object has changed. Spinning icons indicate some action is needed.

There are two invisible column in the GridView called UserID and PCKspec. These are invisible to the user of the system but are required by the system. See below:-

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

The ASP.NET code:-

```
<asp:GridView id="PCItemsGrid"
    BorderColor="black"
    BorderWidth="1"
    CellPadding="3"
    AutoGenerateColumns="false"
    runat="server" BackColor="#C0FFFF">

    <HeaderStyle BackColor="#00AAAA">
    </HeaderStyle>

    <Columns>
        <asp:BoundField
            HeaderText="User ID"
            DataField="UserID"
            Visible=false/>
        <asp:BoundField
            HeaderText="PCKspec code"
            DataField="PCKcode"
            Visible=false/>
        <asp:BoundField
            HeaderText="Performance Criteria/Knowledge..."
            DataField="PCKspec"/>

        <asp:HyperLinkField
            HeaderText="Work"
            DataTextField="Upload"
            DataNavigateUrlFields="UserID, PCKcode"

            DataNavigateUrlFormatString="Viewcomments.aspx?ur={0}&it={1}"/>

        <asp:HyperLinkField
            HeaderText="Note"
            DataTextField="Comments"
            DataNavigateUrlFields="UserID, PCKcode"

            DataNavigateUrlFormatString="Viewcomments.aspx?ur={0}&it={1}"/>

        <asp:BoundField
            HeaderText="Rate"
            DataField="Score"/>

    </Columns>

</asp:GridView>
```

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

The Grid is populated by the APL function as follows:-

```
PCdata←GetPCs user item
:With DT←NEW DataTable
  Columns.Add NEW DataColumn,←'UserID'(System.Type.GetType<'System.Int32'>)
  Columns.Add NEW DataColumn,←'PCKcode'(System.Type.GetType<'System.String'>)
  Columns.Add NEW DataColumn,←'PCKspec'(System.Type.GetType<'System.String'>)
  Columns.Add NEW DataColumn,←'Upload'(System.Type.GetType<'System.String'>)
  Columns.Add NEW DataColumn,←'Comments'(System.Type.GetType<'System.String'>)
  Columns.Add NEW DataColumn,←'Score'(System.Type.GetType<'System.String'>)
  :For I :In 1tpPCdata
    DR←NewRow
    PCKcode←1>PCdata[I;]
    data←PCdata[I;1 2]
    upload←'<img src=',(SELECT I)> 'empty.gif' 'paperclip.gif' 'paperclip_motion.gif'),' border='0''>'
    comment←'<img src=',(SELECT I)> 'empty.gif' 'comment.gif' 'comment_motion.gif'),' border='0''>'
    score←,'BI2' FMT PCdata[I;10]
    DR.ItemArray←user PCKcode data upload comment score
    Rows.Add DR
  :EndFor
:EndWith
DV←NEW DataView DT
PCItemsGrid.DataSource←DV
PCItemsGrid.DataBind
A JD set Backcolor on Cells in GridView
topscore←(4=PCdata[;10])/1tpPCdata
:For I :In 1ptopscore
  (PCItemsGrid.Rows[-1+I>topscore].Cells).BackColor←Color.LightSeaGreen
:EndFor
```

REMEMBER: make DR DV and DT local to this function.

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

The ASP.NET contains Hyperlink definitions for the icons, whether spinning, static or blank:

```
<asp:HyperLinkField
  HeaderText="Work"
  DataTextField="Upload"
  DataNavigateUrlFields="UserID, PCKcode"
  DataNavigateUrlFormatString="Viewcomments.aspx?ur={0}&it={1}"/>
```

If the user clicks the icon an call will be generated using the information from the 2 invisible columns as parameters. For example if one row has a UserID of 999003 and a PCKcode values of 302.2.2 then the following call will be generated:-

```
Viewcomments.aspx?ur=999003&it=302.2.2
```

The ASP.NET program Viewcomments.aspx will be called and the correct APL Page_Load function will be executed.

The APL function can read the parameter information by using the following command:-

```
params←'&',⌘Request.QueryString
params←1↓(params←'&')←params
ur user←⌘VFI(1+params←'ur')→params
item←(1+params←'it')→params
```

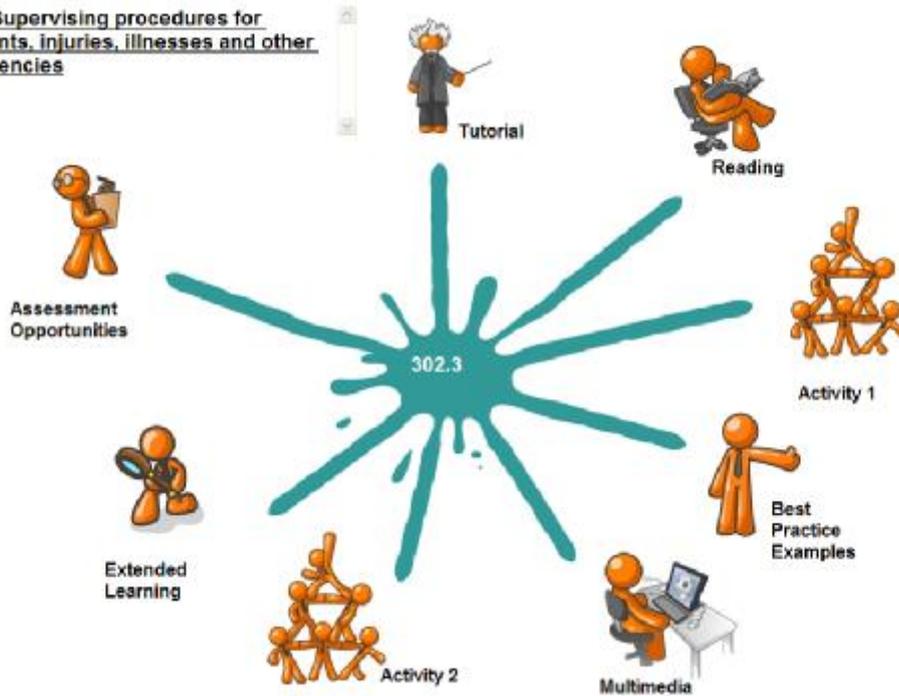
You can of course interrogate this however you wish but Request.QueryString will get you the parameters.

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Template Pages

302.3 Supervising procedures for accidents, injuries, illnesses and other emergencies



There are 11 template from which all the lesson pages can be generated.

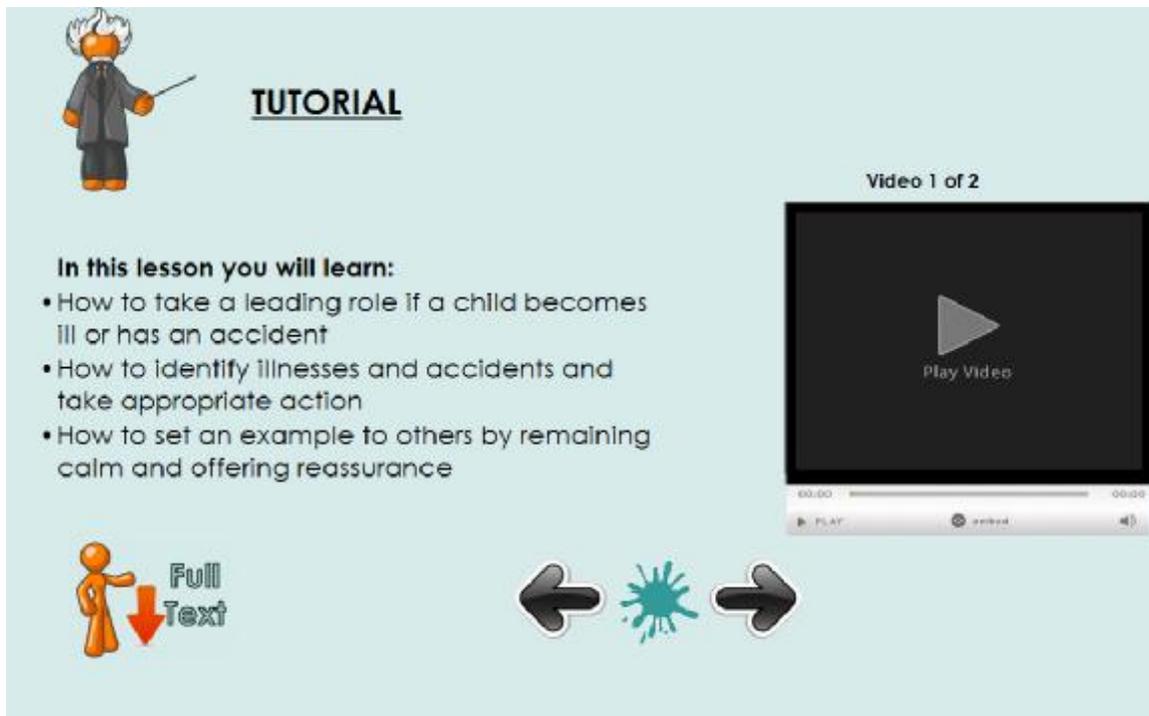
This is a component file containing one component for each lesson page, it contains details about each item, what it look like, what will happen if selected etc.

In the above example each orange man is defined by this information (as is the spider splat in the middle) and each orange man will take you to another lesson page.

Techniques used in the LASER system: APL driving ASP.NET

version 1.0

Selects, Links and Video Plugins



This is an example of Template 1

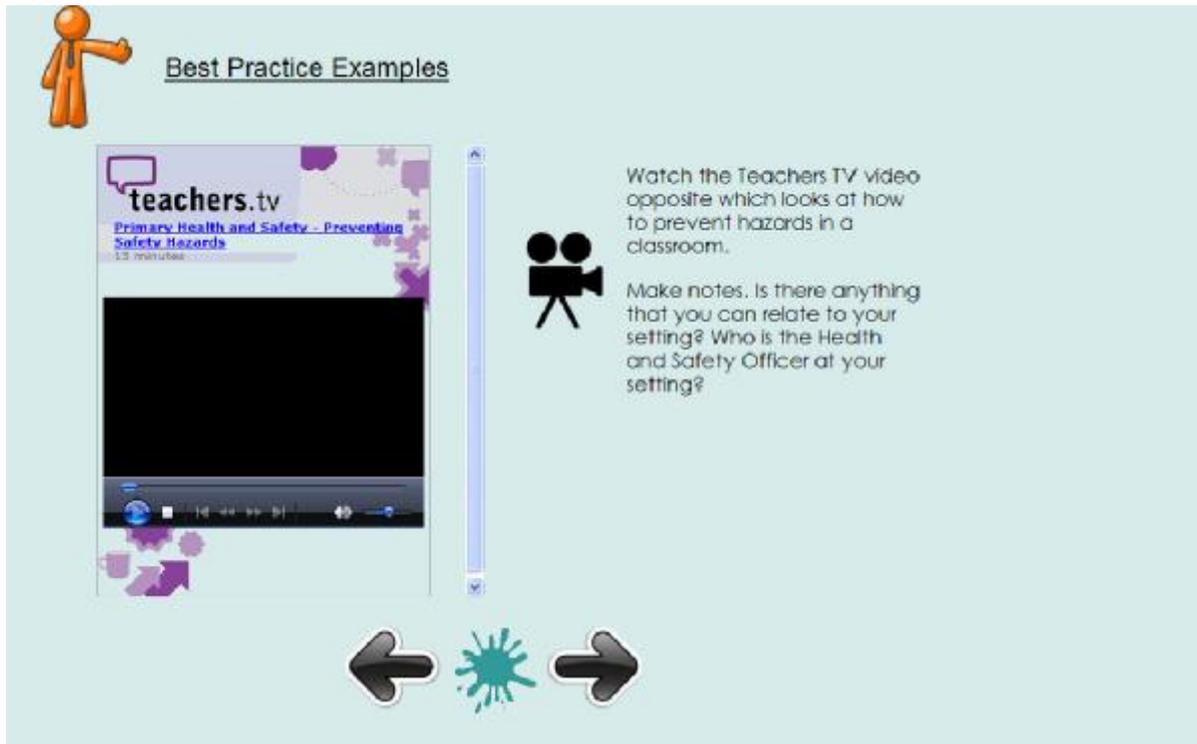
In this case Einstein Man is defined in the template itself, but this is not always the case.

Item 0	The Title of the page (Item 0 is always the title)
Item 1	The filename for the video to be displayed
Item 2	The filename for the full text version of the video
Item 3	Text for under the video (Not used here)
Item 4	1 st bullet point
Item 5	2 nd Bullet point . . . Etc

Additional information held in each lesson page indicates the lesson to the right, the left and the home (or splat). Therefore all the information to generate this page is contained in a standard structure in the lesson file.

Techniques used in the LASER system: APL driving ASP.NET

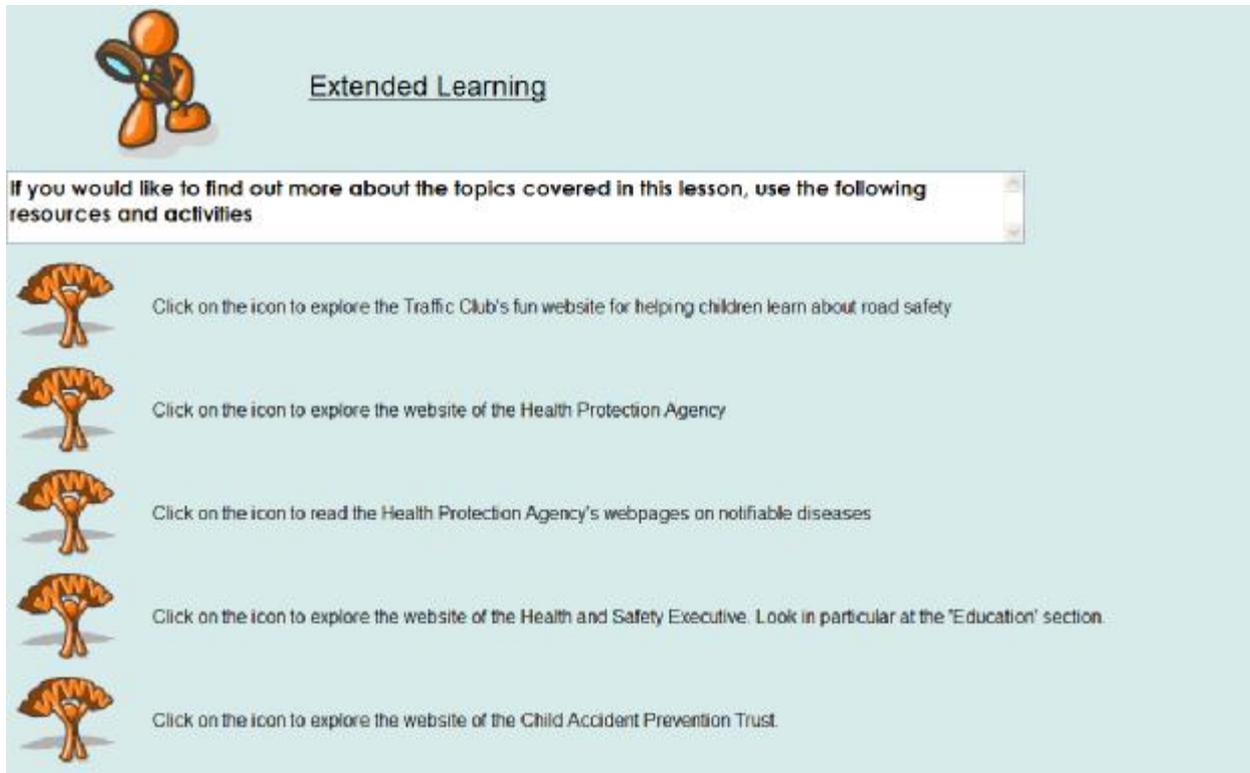
version 1.0



In this example (Template 4) item 2 contains embedded HTML to display and link the page to a resource owned by an associate's company. Therefore we can show video under licence.

Techniques used in the LASER system: APL driving ASP.NET

version 1.0



Extended Learning

If you would like to find out more about the topics covered in this lesson, use the following resources and activities

-  Click on the icon to explore the Traffic Club's fun website for helping children learn about road safety
-  Click on the icon to explore the website of the Health Protection Agency
-  Click on the icon to read the Health Protection Agency's webpages on notifiable diseases
-  Click on the icon to explore the website of the Health and Safety Executive. Look in particular at the 'Education' section.
-  Click on the icon to explore the website of the Child Accident Prevention Trust.

Here we have provided links to many other websites, that when clicked will open these sites in a new window (using Javascript as mentioned before).

**Techniques used in the LASER system:
APL driving ASP.NET**

version 1.0

APL continues to be a key part of our plans:

- We were able to develop quickly
- We will be able to adapt fast to customer needs
- Data is being collected to provide detail statistical information about the use of the system.
- We achieved a user friendly look
- We have beaten all our expectation and income targets and are now expanding our staff to cope with the number of learners we have taken on.

The LASER program was created by The Childcare Company (Old Windsor) Ltd

By

Chris "Ziggi" Paul The Childcare Company Ltd Tithe Barn Tithe Court Langley Berkshire ENGLAND SL3 8AS Ziggi@thechildcarecompany.com	Chris Hogan – 4Xtra
--	---------------------