

Integrating Synchronous Tools with WBT Manager™

Integrity eLearning

June 2003

DISCLAIMERS	1
INTRODUCTION	2
SUMMARY	2
LEVELS OF INTEGRATION.....	2
ADDING LINKS TO WBT MANAGER MENUS	3
<i>Adding links to the Student menu.....</i>	<i>3</i>
<i>Adding links to a Course menu</i>	<i>4</i>
<i>Adding links to a Certificate Program menu.....</i>	<i>4</i>
<i>General link information.....</i>	<i>5</i>
SIMPLE INTEGRATION TECHNIQUES.....	6
SIMPLE LAUNCH.....	6
SINGLE LOGIN	7
MORE SOPHISTICATED INTEGRATION	11
JUMP INTO CLASS	11
LAUNCH CLASS AS LESSON	21
AUTOMATIC REGISTRATION AND ENROLLMENT.....	22
APPENDIX A – INTEGRATION GATEWAY SCRIPTS	24
WBT MANAGER SERVER SCRIPTS	24
<i>ll_golearnlinc.asp</i>	<i>24</i>
<i>ll_config.inc</i>	<i>29</i>
<i>ll_msg.inc.....</i>	<i>30</i>
LEARNLINC SERVER SCRIPTS	30
<i>wbtm_llremote.asp.....</i>	<i>30</i>
<i>wbtm_config.inc.....</i>	<i>39</i>
<i>wbtm_msg.inc.....</i>	<i>41</i>
<i>wbtm_listids.asp.....</i>	<i>42</i>

Disclaimers

1. Every attempt has been made to insure the accuracy of the information included in this document. However, no guaranty of accuracy or suitability is expressed or implied. Users must verify the suitability and stability of any integration solutions based on this information through testing in their specific environment.
2. Unless otherwise noted in the WBT Manager documentation. Internal structures and behaviors of WBT Manager components are subject to change without notice in future WBT Manager upgrades, patches, and releases. Interfaces based on these internal structures may require modifications to operate successfully with future versions of WBT Manager.

Introduction

Summary

Synchronous communication tools allow students to communicate with each other and/or with an instructor in real-time. Typical tools support audio and video conferencing, slide-show presentations, application sharing, and chat rooms.

WBT Manager does not include synchronous tools in its standard configuration. We at Integrity eLearning decided to focus on providing an affordable, scalable LMS solution primarily for self-paced web-based training.

We did recognize early on that many of our customers could benefit from synchronous collaboration tools. There are provisions in WBT Manager included specifically to allow customers to integrate their WBT Manager installations with the collaboration tools that they feel meet their unique requirements.

This document will demonstrate the methods used to integrate WBT Manager with a synchronous tool developed by one of our business partners. The tool (LearnLinc, from EDT Learning Inc) supports on-line, real-time, instructor-based training. For more information about LearnLinc contact us at Integrity eLearning or contact EDT Learning directly:

EDT Learning
www.edtlearning.com

Although the code samples presented in this article are specific to LearnLinc the techniques will be applicable to integrating any 3rd-party tool with WBT Manager.

Levels of integration

This paper will address integration levels of varying usefulness and complexity.

Simple integration techniques

Simple Launch – User is taken to the login page of LearnLinc and logs in using the appropriate id and password.

Single Login – User is taken to their “home” page in the LearnLinc directly from a link in WBT Manager.

More sophisticated integration techniques

Jump into Class – User is taken directly into a synchronous session from a link in WBT Manager.

Launch Class as Lesson – User is taken directly into a synchronous session by selecting a lesson from a WBT Manager course menu.

Automatic Registration – WBT Manager user is automatically registered in LearnLinc the first time that they attempt to enter it (eliminates duplicate data entry).

Automatic Enrollment – WBT Manager user is automatically enrolled in a class when they launch it as a lesson the first time (eliminates duplicate data entry).

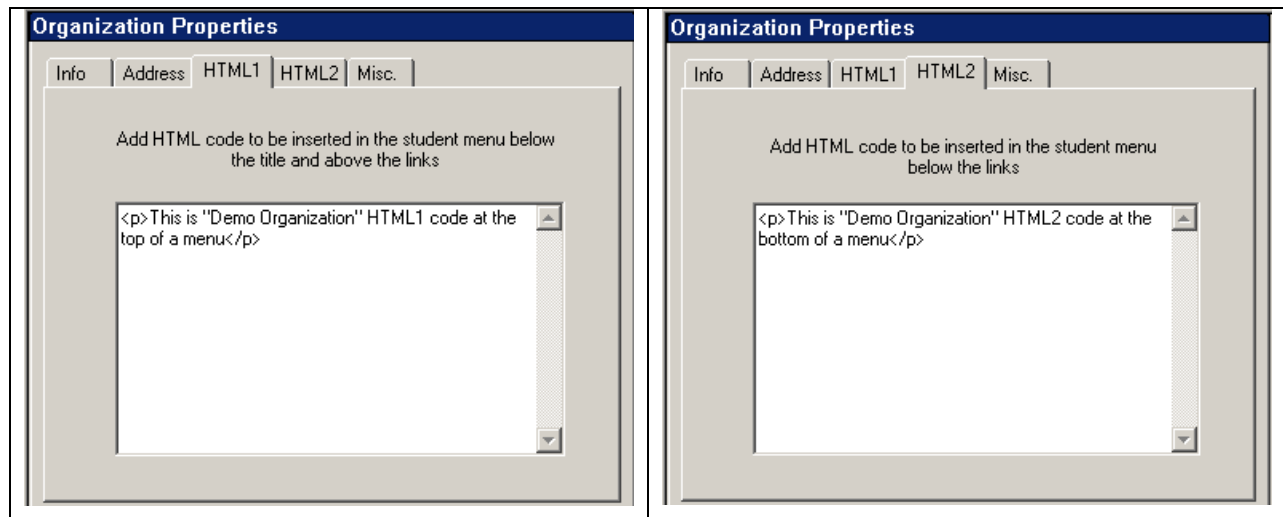
Adding Links to WBT Manager Menus

Many of the techniques demonstrated are made possible by the ability provided in WBT Manager to add custom HTML code (including links) to the standard menus without modifying the menu programs. Code can be added through the WBT Manager administrative functions to:

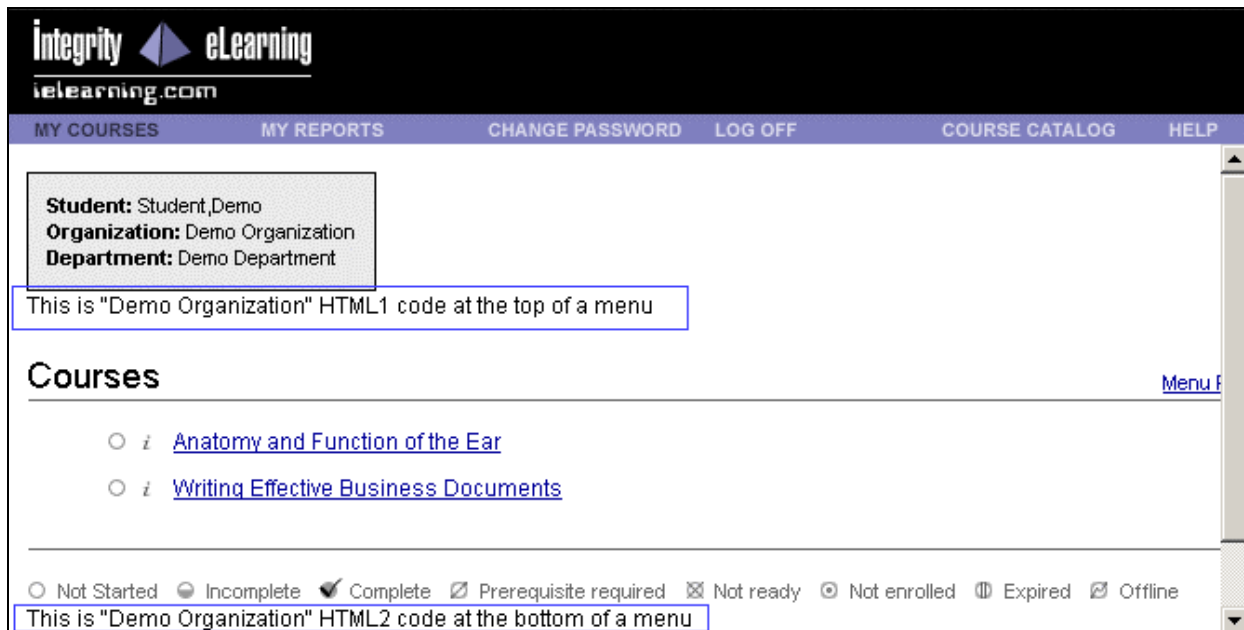
- The Student Menu
- A Course's Lesson menu
- A Certificate Program's menu

Adding links to the Student menu

HTML code can be added to student menus for specific organizations and departments using the administrative organization and department edit functions. There are 2 defined locations where code will appear, the top of the menu, and the bottom of the menu. The data fields are named HTML1 and HTML2. Here is an example of setting HTML1 and HTML2 at the organization level.



Here is what a student in the “Demo” organization would see on their menu (custom code outlined in blue):



Adding links to a Course menu

As with the organization and department, course definitions include HTML1 and HTML2 as well. See the WBT Manager User Guide for more information.

Adding links to a Certificate Program menu

As with the organization, department, and course, Certificate Program definitions include HTML1 and HTML2 as well. See the WBT Manager User Guide for more information.

General link information

The code that generates the WBT Manager menus in the student interface automatically embeds the HTML code from the appropriate HTML1 or HTML2 data field at run time. In order to support links to external applications, several placeholder strings may be used in the HTML to represent user data at run-time. These placeholders will be replaced by the associated data when the menu is generated.

Placeholder string	Run-time value
%nm%	Student's display name, HTML Encoded for proper display on the menu page.
%enm%	Student's display name, URL Encoded for use in the HREF of a link.
%lid%	Student's login id, HTML Encoded for proper display on the menu page.
%elid%	Student's login id, URL Encoded for use in the HREF of a link.
%pwd%	Student's WBT Manager password, URL Encoded for use in the HREF of a link. <i>Note: if this placeholder is used in a link then the student's password will be visible if the source code is viewed. It may be more desirable to have the link jump to a script file that can make use of the student's session variables to get the password.</i>

Simple Integration Techniques

Simple Launch

“Simple Launch” means just that. By selecting a link on a menu the user is taken to the login page of LearnLinc.

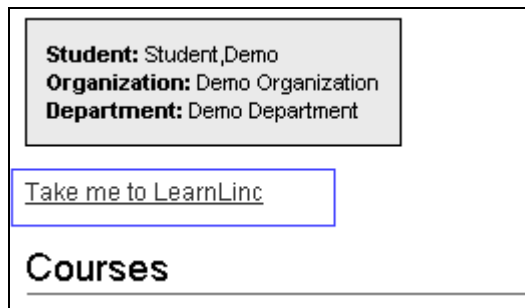
Once in LearnLinc, the user can log on using their LearnLinc id and password whereupon they will be taken to their LearnLinc home page.

Lets add a link to the student’s menu that launches LearnLinc.

Modify HTML1 for the student’s organization to add the following (make href point to your LearnLinc installation):

```
<br>
<p>
<a href="http://yourserver/learnlinc/default.asp" target="_blank">
  Take me to LearnLinc
</a>
</p>
```

This will result in a student menu that looks like:

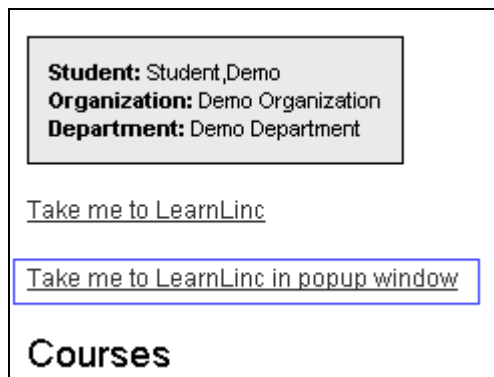


Clicking the link will bring up LearnLinc in a new browser window.

You can also use a WBT Manager script function to bring LearnLinc up in a popup browser window (no toolbar). Add the following to HTML1:

```
<p>
<a
  href="#"
  onclick="OpenPopupWindow(
    'http://yourserver/learnlinc/default.asp',
    'l', false, 800, 600, true, 0, 0, true, false, false, null )"
>
  Take me to LearnLinc in popup window
</a>
</p>
```

This results in:



Clicking the new link brings LearnLinc up in a browser window that has no toolbar or menu.

Single Login

A “Simple Launch” link is very easy to create but that approach requires the student to log in twice. Once to WBT Manager, and again to LearnLinc. It would be much nicer if the student could just jump into their LearnLinc home page from WBT Manager. We will call this a “Single Login” approach.

In order to use a single log in, the student’s LearnLinc login information must be included in the link placed on the student’s menu. Other requirements are:

- Student must already be registered in LearnLinc
- Student’s LearnLinc ID and Password must match their WBT Manager ID and Password.

We will discuss options for dealing with these requirements later in this article.

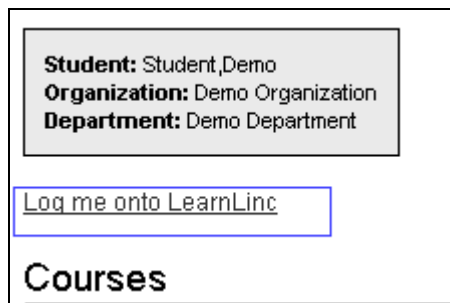
LearnLinc allows a user to jump directly to their home page by passing their login id and password to the login page on the URL or with a form POST (you should be able to do something similar with almost any 3rd party web-based application). The format to do this is:

```
http://yourserver/learnlinc/default.asp?loginname=<userid>&password=<userpwd>
```

Put the following code into HTML1

```
<br>
<p>
<a
  href="http://yourserver/learnlinc/default.asp?loginname=%elid%&password=%pwd%" target="_blank"
>
  Log me onto LearnLinc
</a>
</p>
```

This results in:



Clicking the link will open LearnLinc in a new browser window and the user will be logged on and redirected to their home page.

There is a risk to the above method. Place your mouse over the link. What do you see in the browser's status bar? Do you see the user's password?

If that is not satisfactory for your installation you can hide the user information a bit by using the OpenPopupWindow command. Replace the HTML1 code with:

```
<br>
<p>
<a
  href="#"
  onclick="OpenPopupWindow(
    'http://yourserver/learnlinc/default.asp?loginname=%elid%&password=%pwd%',
    'll', false, 800, 600, true, 0, 0, true, false, false, false, null )"
  >
  Log me onto LearnLinc
</a>
</p>
```

Rolling the mouse over the link will no longer show the user's password

However, the password can still be seen with "View Source" as seen below:

```
<br><p><a href="#" onclick="OpenPopupWindow(
'http://yourserver/learnlinc/default.asp?loginname=demo&password=demo', 'll', false, 800, 600, true, 0, 0, true, false, false,
false, null )">Log me onto LearnLinc</a></p>
```

A better way to launch LearnLinc is by using an intermediary ASP script to collect the login information and then POST the data to LearnLinc. The sample interface script file **ll_logmeon1.asp** does this. The HTML1 code will change to:

```
<br>
<p>
<a
  href="#"
  onclick="OpenPopupWindow( 'll_logmeon1.asp', 'll', false,
    800, 600, true, 0, 0, true, false, false, false, null )"
  >
  Log me onto LearnLinc
</a>
</p>
```

Now clicking on the link will cause a self-submitting form to be created that immediately submits itself to the LearnLinc login page. The password can be seen by “View Source” only if the submission of the form is blocked (by disabling JavaScript for instance).

Here are the contents of **ll_logmeon1.asp**
(would be loaded to the WBT Manager ASP directory):

```
<%@ LANGUAGE="VBSCRIPT"%>
<%
Option Explicit

'WBT Manager to LearnLinc interface script
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'WBT Manager licensees are granted the right to copy and modify this
'script as desired for use in integrating 3rd party applications
'with WBT Manager.
'
' 03142003 ms - created as part of "Integrating Synchronous Tools" white-paper.

'include information about how to find learnlinc
%>
<!--#include file="ll_config.inc"-->
<%

Dim strLoginID
Dim strPassword
Dim strLLPath

'get the information needed for LearnLinc login
strLoginID = Trim( Session("strWBTLgonID") )
strPassword = Trim(Session("strWBTLgonPW"))

'is the student logged on
If Len( strLoginID ) = 0 Then
    Response.Write "You must be logged onto WBT Manager to use this Function"
Else

    'create a page with a self-submitting form to jump into LearnLinc

    'submit URL
    strLLPath = LL_HOME & "default.asp"

%>

<html>
<head>
<title>--&gt;&gt;LearnLinc</title>
</head>

<body onload="document.launch_form.submit();">
    <form method="post" name="launch_form" id="launch_form" action="<%=Trim(strLLPath)%>">
        <input type="hidden" name="loginname" value="<%=strLoginID%>">
        <input type="hidden" name="password" value="<%=strPassword%>">
        <center>
            <input type="submit" border="0"
                value="Go"
                id="go" name="go">
        </center>
    </form>
</body>
</html>

<%
End If
%>
```



Here are the contents of **ll_config.inc**
(would be loaded to the WBT Manager ASP directory):

```
<%  
'WBT Manager user-customizable include file  
,  
,  
'Copyright © 2003 Integrity eLearning. All Rights Reserved.  
,  
'This include file contains parameters used to control  
'scripts that integrate WBT Manager with LearnLinc(TM)  
,  
  
'path to the LearnLinc server  
const LL_HOME = "http://yourserver/learnlinc/"  
  
%>
```

More sophisticated integration

The previous techniques can usually be applied to any 3rd party web-based application. For further integration there has to be a mechanism for performing some of the application's functions outside of its user interface.

LearnLinc provides a set of API tools to perform application functions. One of the tools included is a script called **IIRemoteJoin.asp** that performs many functions (including automatically creating student accounts, instructor accounts, and classes “on the fly” as well as logging the student on and starting a synchronous session). This script is very complex. For this article we will create simpler, single-purpose scripts based on the functions found in **IIRemoteJoin.asp**.

If you are integrating a tool other than LearnLinc you will need to consult the appropriate documentation and/or consult with the tool vendor to determine what you will be able to do and how to do it.

Jump into Class

The “single login” method does eliminate some steps for the user but it would be nice if they could jump from a link in WBT Manager directly into a synchronous session or discussion group associated with a particular course.

In this section we will discuss how to jump directly into a LearnLinc “Class” from a link embedded on a WBT Manager course menu.

Requirements:

- Student must be enrolled in LearnLinc with the same login id and password as in WBT Manager.
- The target course and class must already exist in LearnLinc.
- The Student must be assigned to the target course and class in LearnLinc

We will address easing those requirements (automating more of the administrative functions later).

The LearnLinc API supports joining an existing class session on LearnLinc without navigating through the LearnLinc campus. In order to do this the student's LearnLinc login information and information identifying the class must be passed to a script on the LearnLinc server.

We will add a link to a WBT Manager course's menu that jumps into a meeting class set up for that course in LearnLinc.

First, a course named “Integrating with LearnLinc”, containing a single lesson named “Joining a Class” is created in WBT Manager (for this example the lesson will not need a valid URL). The demo student is assigned to the WBT Manager course.

Next, a course with the identical name containing a class called “Course Meeting Room” (with floor-type “meeting”) are set up in LearnLinc and the demo student is assigned to the LearnLinc course.

After that, the LearnLinc identifiers for the course and the meeting class must be determined. One way to do this is to open the LearnLinc database with a DBMS tool and look in the **courses** and **classes** tables to find the internal numbers. Another is to go to the course details page in LearnLinc and “View Source”.

Look for an HTML **<input>** tag with the name=”c”. For example:

```
<input type="hidden" name="c" value="7">
```

The course Identifier is “7”.

Now click the “Classes” link then the “Settings” link for the class you are interested in. View source on that page. Again look for an **<input>** tag named “c” (you should also find a tag named “co” containing the course id). For example:

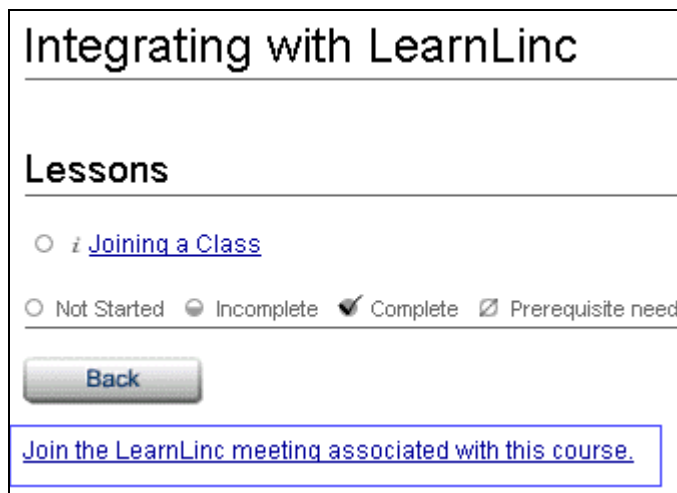
```
<input type="hidden" name="c" value="34">  
<input type="hidden" name="co" value="7">
```

The class identifier is “34”.

Finally, a link is added to the WBT Manager course menu. Add the following to the HTML2 field associated with the “Integrating with LearnLinc” course in WBT Manager.

```
<p>  
<a href="ll_gomeeting.asp?llcourse=7&llclass=34" target="_blank">  
  Join the LearnLinc meeting associated with this course.  
</a>  
</p>
```

This results in a link on the course menu:



When the link is clicked a self-submitting form is generated that transfers control (along with the user’s login information) to an integration script on the LearnLinc server. That script generates output to the user’s browser that causes it to launch the LearnLinc classroom software and join the class.

Here are the contents of **ll_gomeeting.asp** and the updated contents of **ll_config.inc** (would be loaded to the ASP directory on the WBT Manager site):

ll_gomeeting.asp:

```
<%@ LANGUAGE="VBSCRIPT"%>
<%
Option Explicit

'WBT Manager to LearnLinc interface script
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'WBT Manager licensees are granted the right to copy and modify this
'script as desired for use in integrating 3rd party applications
'with WBT Manager.
'
' 03182003 ms - created as part of "Integrating Synchronous Tools" white-paper.

'include information about how to find learnlinc
%>
<!--#include file="ll_config.inc"-->
<%

'this script demonstrates how to jump directly into a synchronous meeting
'from a link embedded in WBT Manager
'
'student must already be registered in LearnLinc
'course and LearnLinc class for the meeting must already exist
'student must already be enrolled in the meeting class

'URL parameters (may be GET or POST)
'
'      llcourse: numeric           Internal LearnLinc identifier for course
'
'      llclass:  numeric           required
'                                 Internal LearnLinc identifier for class

Dim nClassID
Dim nCourseID
Dim strLoginID
Dim strPassword
Dim strLLPath

'get the information needed for LearnLinc login
strLoginID = Trim( Session("strWBTLgonID") )
strPassword = Trim( Session("strWBTLgonPW") )

If UCase(Trim(Request.ServerVariables("REQUEST_METHOD"))) = "GET" Then
    nCourseID = Trim(Request.QueryString("llcourse"))
    nClassID = Trim(Request.QueryString("llclass"))
Else
    nCourseID = Trim(Request.Form("llcourse"))
    nClassID = Trim(Request.Form("llclass"))
End If
```

ll_gomeeting.asp (continued)

```

'is the student logged on
If Len( strLoginID ) = 0 Then

    Response.Write "You must be logged onto WBT Manager to use this Function"

'was a course id specified
Elseif Not IsNumeric( nCourseID ) Then

    Response.Write "Invalid LearnLinc course id"

'was a class id specified
Elseif Not IsNumeric( nClassID ) Then

    Response.Write "Invalid LearnLinc class id"

Else

    'create a page with a self-submitting form to jump into LearnLinc

    'submit URL
    strLLPath = LL_WBTMAN & "wbtm_join1.asp?nocache=" & Server.URLEncode(Now())
%>

<html>
<head>
<title>--&gt;&gt;LearnLinc</title>
</head>

<body onload="document.launch_form.submit();">
    <form method="post" name="launch_form" id="launch_form" action="<%=Trim(strLLPath)%>">
        <input type="hidden" name="loginname" value="<%=strLoginID%>">
        <input type="hidden" name="password" value="<%=strPassword%>">
        <input type="hidden" name="classid" value="<%=nClassID%>">
        <input type="hidden" name="courseid" value="<%=nCourseID%>">
        <center>
            <input type="submit" border="0"
                value="Go"
                id="go" name="go">
        </center>
    </form>
    <br>
    <br>
    <center>
        <a href="javascript:window.close()" target="_self">Close</a>
    </center>
</body>
</html>

<%
End If
%>

```


wbtm_join1.asp (continued)

```

'URL parameters (must be POST parameters)
'
'      loginname:      string
'                               required
'                               WBT Manager/LearnLinc user login id
'
'      password:      string
'                               required
'                               WBT Manager/LearnLinc user password
'
'      courseid:      numeric
'                               required
'                               Internal LearnLinc identifier for course
'
'      classid:      numeric
'                               required
'                               Internal LearnLinc identifier for class
'
'include information about the calling WBT Manager site
%>
<!--#include file="wbtm_config.inc"-->
<%

const E_RECORD_NOT_FOUND = -104

Dim nCourseID
Dim nClassID
Dim strLoginID
Dim nUserID
Dim strPassword
Dim strEncPW
Dim bOk
Dim objLL
Dim nReturn
Dim strError
Dim strCourseName
Dim dummy
Dim strLRNFileData

bOk = True

'get the information needed for LearnLinc login
'NOTE: FORM POST was required to hide the student's
'      login information. If your site requires additional
'      protection for this information then an encryption
'      mechanism should be used for this information
'

strLoginID = Trim( Request.Form("loginname") )
strPassword = Trim( Request.Form("password") )
nCourseID = Trim( Request.Form("courseid") )
nClassID = Trim( Request.Form("classid") )

'Form must be submitted by a script in the WBT Manager ASP directory
' if browsers are encountered that do not support http_referer then
' a different security mechanism should be used such as passing a key
' field in the form.
If Left( LCase( Trim( Request.ServerVariables("HTTP_REFERER") ) ), Len( WBTMAN_ASP ) ) <> WBTMAN_ASP Then

    bOk = False
    Response.Write "Must be called from WBT Manager."
    Response.End

End If

'get the LearnLinc database object (will be created automatically by server)
Set objLL = Session("lldbo")

```

wbtm_join1.asp (continued)

```

'check for missing parameters
If bOk Then

    If      Not IsNumeric( nCourseID ) Or _
           Not IsNumeric( nClassID ) Or _
           Len( strLoginID ) < 1 Or _
           Len( strPassword ) < 1 Then

        bOk = False
        Response.Write "Missing required parameters. Cannot join class."
        Response.End

    End If

End If

*****
' Uncomment this next section to require that
'     passwords match between WBT Manager and LearnLinc
'     otherwise LearnLinc password is ignored
'
'check to see if the student logon information matches an existing LearnLinc user
'If bOk Then
'
'     nReturn = objLL.Login( strLoginID, strPassword, nUserID )
'     If nReturn <> 0 Then
'
'         bOk = False
'         Response.Write "Invalid user id or password (" & nReturn & "). Unable to connect to LearnLinc."
'         Response.End
'
'     End If
'
'End If
*****

'get the user's encrypted password for use in the LRN file
'also checks to see if the user exists in LearnLinc
If bOk Then

    nReturn = objLL.LookupPassword( strLoginID, strEncPW, dummy, nUserID )

    If nReturn <> 0 Then

        bOk = False
        Response.Write "Invalid user id:" & nReturn & "-" & objLL.GetErrorString( nReturn )
        Response.End

    End If

End If

'check to see if a valid course is specified

```

wbtm_join1.asp (continued)

```

If bOk Then
    nReturn = objLL.GetCourse( nCourseID, strCourseName, dummy, _
                                dummy, dummy, dummy, _
                                dummy, dummy )

    If nReturn = E_RECORD_NOT_FOUND Then
        bOk = False
        Response.Write "The specified course does not exist in LearnLinc."
        Response.End

    ElseIf nReturn <> 0 Then
        bOk = False
        Response.Write "Error retrieving course:" & nReturn & "-" & objLL.GetErrorString( nReturn )
        Response.End

    End If
End If

'check that the student is enrolled in the course
If bOk Then
    nReturn = objLL.GetIsRegistered( nUserID, nCourseID )

    If nReturn = 0 Then
        bOk = False
        Response.Write "You are not enrolled in "" & strCourseName & "" in LearnLinc."
        Response.End

    ElseIf nReturn <> 1 Then
        bOk = False
        Response.Write "Error checking course enrollment:" & nReturn & "-" & _
            objLL.GetErrorString( nReturn )
        Response.End

    End If
End If

'check to see if a valid class is specified
If bOk Then
    nReturn = objLL.GetClass( nClassID, dummy, dummy, _
                                dummy, dummy, dummy, _
                                dummy, dummy, dummy, _
                                dummy, dummy, dummy )

    If nReturn = E_RECORD_NOT_FOUND Then
        bOk = False
        Response.Write "The specified class does not exist in LearnLinc."
        Response.End

    ElseIf nReturn <> 0 Then
        bOk = False
        Response.Write "Error retrieving class:" & nReturn & "-" & objLL.GetErrorString( nReturn )
        Response.End

    End If
End If

```

```
'Join the class
```

wbtm_join1.asp (continued)

```
If bOk Then
    nReturn = objLL.CreateLRNFile( Application("server"), CInt(Application("port")), _
                                nUserID, CStr(strEncPW), CInt(nClassID), _
                                CInt(0), "", CInt(0), "", strLRNFileData )
    'Make sure that create worked
    If nReturn <> 0 Then
        Response.Write "Creation of LRN file failed.<BR>"
        Response.Write "Here is the Error: " & nReturn & "-" & objLL.GetErrorString( nReturn ) & "<BR>"
        Response.Write "Server: " & Application("server") & "<BR>"
        Response.Write "port: " & Application("port") & "<BR>"
        Response.Write "UserID: " & nUserID & "<BR>"
        Response.Write "UserPassword: " & strEncPW & "<BR>"
        Response.Write "ClassID: " & nClassID & "<BR>"
        Response.Write "Here is the Data: " & strLRNFileData & "<BR>"
    Else
        Response.ContentType ="application/x-learnlinc"
        Response.Write strLRNFileData
    End If
Else
    Session.Abandon 'failed to join class
End If

'release the LearnLinc database object
Set objLL = Nothing
Response.End

%>
```

wbtm_config.inc:

```
<%
'WBT Manager user-customizable include file
'
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'This include file contains parameters used to control
'scripts that integrate WBT Manager with LearnLinc(TM)
'
'path to the LearnLinc server
const WBTMAN_HOME = "http://yourserver/wbtman/"
const WBTMAN_ASP = "http://yourserver/wbtman/asp/"

%>
```

How the **wbtm_join1** script works:

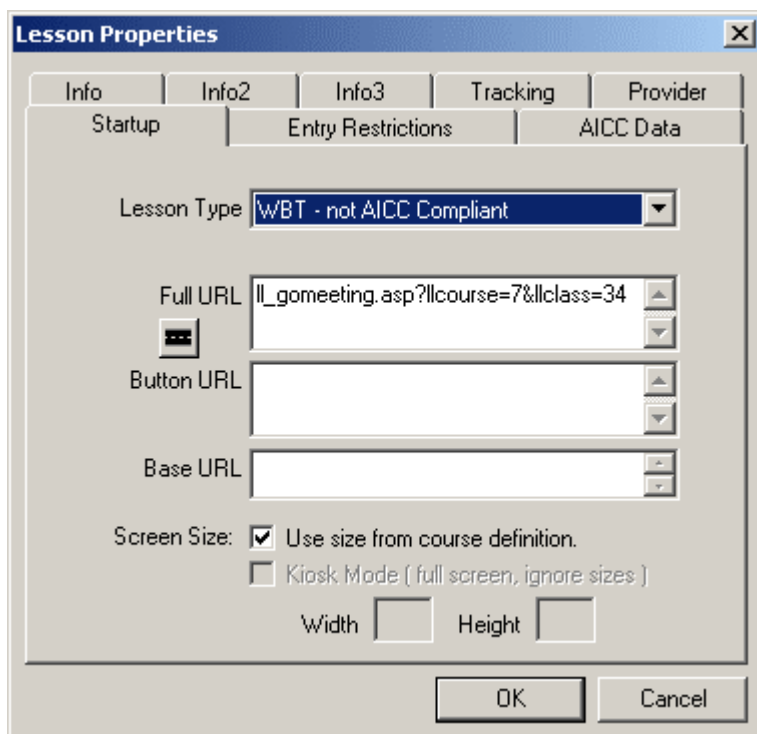
1. Data is collected from the posted form (form method=POST is required).
2. The submission is tested to make sure that it came from a link on the WBT Manager site.
3. Parameters are checked.
4. The LearnLinc database is checked to make sure that the user exists.
5. The LearnLinc database is checked to make sure that the course exists.
6. The LearnLinc database is checked to make sure that the user is enrolled in the course.
7. The LearnLinc database is checked to make sure that the class exists.
8. The LearnLinc classroom software is launched.

Launch Class as Lesson

The above approach provides a link on the course menu to allow a student to jump into a meeting. Another approach is to define a link to a class within the course structure so that an administrator may enter status later (or the lesson could be set to complete the first time a student enters it).

The procedure is nearly identical with that for “Jump into Class” above. The only difference is that now a lesson is defined in WBT Manager to contain the link (instead of placing the link in the HTML fields for the course).

Modify the startup tab of the “Joining a class” lesson within “Integrating with LearnLinc” as shown below:



The screenshot shows the "Lesson Properties" dialog box with the "Startup" tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with tabs for "Info", "Info2", "Info3", "Tracking", and "Provider". Under the "Startup" tab, there are sub-tabs for "Entry Restrictions" and "AICC Data". The "Lesson Type" is set to "WBT - not AICC Compliant". The "Full URL" field contains the text "ll_gomeeting.asp?llcourse=7&llclass=34". The "Button URL" and "Base URL" fields are empty. The "Screen Size" section has a checked checkbox for "Use size from course definition." and an unchecked checkbox for "Kiosk Mode (full screen, ignore sizes)". Below these are empty input fields for "Width" and "Height". At the bottom of the dialog are "OK" and "Cancel" buttons.

Now when a student selects the “Joining a class” lesson from the WBT Manager course menu they will be taken directly into the meeting.

Automatic Registration and Enrollment

One of the drawbacks of the previous approaches is that the student must already be registered in LearnLinc and enrolled in the LearnLinc course to be able to jump into a class. Since the student is already registered and enrolled in the WBT Manager database, this involves duplicate effort with the associated possibility of error.

A better way is to have the interface automatically register and enroll students when they attempt to log onto LearnLinc or enter a class for the first time.

Caution: As with previously described approaches, this approach requires that the login id in WBT Manager match the LearnLinc login name. If student "John Smith" is added to the LearnLinc database with the name "jsmith", and his brother "James Smith" is added to WBT Manager as "jsmith", these interface scripts will treat John and James as the same student. The easiest approach to preventing this problem may be to require that all users log onto WBT Manager first and jump to LearnLinc from there using one of scripts described below.

An "integration gateway kit" has been developed for WBT Manager clients who desire to use LearnLinc. The kit contains the following scripts along with documentation in their installation and use (source code is in appendix A).

File	Purpose
ll_golearnlinc.asp	Provides a "gateway" for WBT Manager to perform administrative functions in LearnLinc and to jump into a LearnLinc session. This is the script that will be called (with various parameters) by links within WBT Manager.
ll_config.inc	Contains parameters that control how the WBT Manager side of the integration gateway operates.
ll_msg.inc	Contains text messages displayed by the WBT Manager side of the integration gateway.
wbtm_llremote.asp	Provides access to LearnLinc functions to ll_golearnlinc.asp through HTTP POST requests. Uses LearnLinc functions exposed by the LearnLinc API.
wbtm_config.inc	Contains parameters that control how the LearnLinc side of the integration gateway operates.
wbtm_msg.inc	Contains text messages displayed by the LearnLinc side of the integration gateway.
wbtm_listids.asp	Allows administrators to list the internal LearnLinc identifiers for courses and classes so that they can be included in WBT Manager links and lesson definitions.

The principal scripts in the integration kit are **ll_golearnlinc.asp** and **wbtlm_llremote.asp**. There several significant differences between these 2 scripts and the scripts presented earlier.

1. The user's LearnLinc password can be ignored in case the user's password is changed on one system but not the other (this can be disabled).
2. The user's password is never included where it can be seen using "View Source".
3. If the user's login name does not exist in LearnLinc then a record is added for the student automatically (this can be disabled).
4. If a LearnLinc class id is included as a parameter then the scripts will attempt to join the student to the class. Otherwise the student will be take to their LearnLinc home page.
5. If an attempt is made to join a class for a course that the student is not assigned to in LearnLinc then the student can be added to that course automatically (this can be disabled).

How the scripts work:

1. **ll_golearnlinc.asp** is called from a link in WBT Manager.
2. **ll_golearnlinc.asp** sends a POST request to **wbtlm_remote.asp** to see if the student already exists. This post is sent from the WBT Manager server to the LearnLinc server directly, not through the user's browser.
3. If the student does not exist (**wbtlm_remote.asp** returns 0), and the site is configured to automatically add students to LearnLinc, student information is collected from the WBT Manager database. Another post to **wbtlm_remote.asp** sends the data to LearnLinc where a new student record is created.
4. **ll_golearnlinc.asp** builds a self-submitting form to launch LearnLinc (by posting to **wbtlm_remote.asp**).
5. **wbtlm_remote.asp** logs the user onto LearnLinc.
6. If a LearnLinc class id was supplied to **ll_golearnlinc.asp** then **wbtlm_remote.asp** checks to see if the user is enrolled in the course containing that class. If not, and if the site is configured to allow auto-enrollment, the user is enrolled in the course. The user's browser is then directed to join the class.

Appendix A – Integration Gateway Scripts

WBT Manager server scripts

These scripts must be loaded into the **ASP** directory of the WBT Manager installation.

ll_golearnlinc.asp

```
<%@ LANGUAGE="VBSCRIPT"%>
<%
Option Explicit

'WBT Manager to LearnLinc interface script
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'WBT Manager licensees are granted the right to copy and modify this
'script as desired for use in integrating 3rd party applications
'with WBT Manager.
'
*****
'Parameters:
'
'       name:    llclass
'       type:    numeric, optional
'       use:     identifies the class that the user wishes to join.
'                internal LearnLinc course id number
'                if not specified then user is taken to their learnlinc home page
'
*****
'Change History:
'
' 03182003 ms - created as part of "Integrating Synchronous Tools" white-paper.

'include configuration information and string table
%>
<!--#include file="ll_config.inc"-->
<!--#include file="ll_msg.inc"-->
<!--#include file="adovbs.inc"-->
<%
Const WBTM_C_INSTRUCTILT = 4096 ' from v 1.61 wbtconst.inc

Dim strLoginID
Dim strPassword
Dim strLLPath
Dim bContinue
Dim nStatus

Dim strCode 'authentication code to send in place of user password.
            'may also contain error message if a function fails.

Dim xmlHTTP
Dim strProgUtil

'name of wbtmanager util object for later
strProgUtil = "wbtman4d.webutil2"

bContinue = True

'get the information needed for LearnLinc login
strLoginID = Trim( Session("strWBTLgonID") )
strPassword = Trim( Session("strWBTLgonPW") ) 'optional depending on configuration settings

'is the student logged on
```

ll_golearnlinc.asp (continued)

```

If Len( strLoginID ) = 0 Then
    Response.Write LL_MSG_NOTLOGGEDON
    Response.End
    bContinue = False
End If

'create an instance of the XMLHTTP object
Set xmlHTTP = Server.CreateObject("MSXML2.ServerXMLHTTP")

'see if the student is in the LearnLinc database
If bContinue Then

    nStatus = CheckLLUser( xmlHTTP, strLoginID, strPassword, strCode )

    If nStatus = "0" Then

        'ok to add automatically?
        If LL_ADD_USERS Then

            'try to add
            If Not AddLLUser( strProgUtil, xmlHTTP, strLoginID, strPassword, strCode ) Then
                Response.Write LL_MSG_ERRORADDINGUSER & " " & strCode
                Response.End
                bContinue = False
            End If

        Else

            'manual registration is required
            Response.Write LL_MSG_LLREGREQ
            Response.End
            bContinue = False

        End If

    ElseIf nStatus <> "1" Then

        'error in check user function
        Response.Write LL_MSG_ERRORCHECKINGUSER & " " & nStatus & ":" & strCode
        Response.End
        bContinue = False

    End If

End If

Set xmlHTTP = Nothing

If bContinue Then

    'create a page with a self-submitting form to jump into LearnLinc

    'submit URL
    strLLPath = LL_WBTMAN & "wbtm_llremote.asp?nocache=" & server.URLEncode(Now())

    Response.Write _
        "<html>" & vbCrLf & _
        "<head>" & vbCrLf & _
        "<title>" & Server.HTMLEncode( LL_MSG_STARTINGTITLE ) & "</title>" & vbCrLf

```

ll_golearnlinc.asp (continued)

```

Response.Write _
    "<script language=""javascript"">" & vbCrLf & _
    "function closeme()" & vbCrLf & _
    "{" & vbCrLf & _
    "//wake up opener window" & vbCrLf & _
    "if( window.opener != null ) window.opener.focus();" & vbCrLf & _
    "//log out of LearnLinc and close window" & vbCrLf & _
    "top.document.location="" & strLLPath & "&function=logout"";" & vbCrLf & _
    "}" & vbCrLf & _
    "</script>" & vbCrLf

Response.Write _
    "</head>" & vbCrLf & _
    "<body onload=""document.launch_form.submit();"">" & vbCrLf

Response.Write _
    "<form method=""post"" name=""launch_form"" & _
    "id=""launch_form"" action="" & Server.URLEncode(strLLPath) & """">" & vbCrLf & _
    "<input type=""hidden"" name=""function"" value=""go"">" & vbCrLf & _
    "<input type=""hidden"" name=""loginname"" value="" & _
    Server.URLEncode( strLoginID ) & """">" & vbCrLf & _
    "<input type=""hidden"" name=""code"" value="" & _
    Server.URLEncode( strCode ) & """">" & vbCrLf & _
    "<input type=""hidden"" name=""classid"" value="" & _
    Server.URLEncode( Trim(Request("lclass"))) & """">" & vbCrLf & _
    "</form>" & vbCrLf

Response.Write "<p>" & LL_MSG_LLSTARTING & "</p>" & vbCrLf
Response.Write "<p>" & LL_MSG_LLSTARTING2 & "</p>" & vbCrLf
Response.Write "<center><p>" & LL_MSG_LLSTARTING3 & "</p></center>" & vbCrLf

Response.Write _
    "</body>" & vbCrLf & _
    "</html>" & vbCrLf

Response.End

End If

'check LearnLinc user
' returns "0" if user not found
' returns "1" if user found. Returns authentication code in strCode.
' any other return value indicates error.
Function CheckLLUser( ByRef xmlHTTP, ByVal strLoginID, ByVal strPassword, ByRef strCode )

    Dim strData
    Dim strResult
    Dim strTargetURL

    CheckLLUser = "0"

    strTargetURL = LL_WBTMAN & "wbtm_llremote.asp?nocache=" & Server.URLEncode(Now())

    strData = _
        "function=checkuser" & _
        "&keystring=" & Server.URLEncode(LL_IDENT_KEY) & _
        "&loginname=" & Server.URLEncode(strLoginID)
    If LL_USE_PW Then
        strData = strData & "&password=" & Server.URLEncode(strPassword)
    End If

    'send the request
    on error resume next
    xmlHTTP.open "POST", strTargetURL, False
    xmlHTTP.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
    xmlHTTP.send strData

    'check for runtime errors

```

ll_golearnlinc.asp (continued)

```

If Err.number <> 0 Then
    CheckLLUser = "-1"
    strCode = Err.number & ":" & Err.description

Elseif xmlHTTP.status <> "200" Then
    CheckLLUser = "-1"
    strCode = xmlHTTP.status & ":" & xmlHTTP.statusText

Else 'no local errors
    'get results
    strResult = xmlHTTP.responseText
    Call ParseResults( strResult, CheckLLUser, strCode )

End If

End Function

Function AddLLUser( ByVal strUtilProgID, ByRef xmlHTTP, ByVal strLoginID, ByVal strPassword, ByRef strCode )

    Dim strData
    Dim strResult
    Dim strTargetURL
    Dim nStatus
    Dim objUtil
    Dim dbConn
    Dim rsData
    Dim strSQL
    Dim strName
    Dim strEmail
    Dim nLLAuthLevel
    Dim vTmp

    AddLLUser = False

    'first get user information to send to LearnLinc
    'need loginid, password, display name, email
    Set objUtil = Server.CreateObject( strUtilProgID )
    Set dbConn = objUtil.OpenDBConnection()

    'is user an administrator
    vTmp = Trim(UCCase(Session("strWBTUserType")))
    If vTmp = "A" Then
        strSQL = "select name, email, perm1 from admin where admin_id=" & _
                CStr(Session("nWBTUserID"))
    Else
        strSQL = "select name, email from students where stud_id=" & _
                CStr(Session("nWBTUserID"))
    End If

    Set rsData = dbConn.Execute( strSQL, adCmdText )
    If rsData.EOF Then
        'something seriously wrong here, clean up and bail out
        Response.Write "error getting your record."
        rsData.close
        Set rsData = Nothing
        dbConn.Close
        Set dbConn = Nothing
        Set objUtil = Nothing
        Response.End
        Exit Function
    Else

```

ll_golearnlinc.asp (continued)

```

strName = Trim(rsData("name"))
If IsNull(strName) Then strName = "missing"
strEmail = Trim(rsData("email"))
If IsNull(strEmail) Then strEmail = ""

'is this an instructor
nLLAuthLevel = 7 'default to student
If vTmp = "A" Then
    vTmp = rsData("perm1")
    If IsNull(vTmp) Then vTmp = 0
    vTmp = CLng(vTmp)
    If (vTmp And WBTM_C_INSTRUCTILT) <> 0 Then
        'add as a learnlinc instructor
        nLLAuthLevel = 511
    End If
End If

End If

End If

rsData.Close
Set rsData = Nothing
dbConn.Close
Set dbConn = Nothing
Set objUtil = Nothing

'Send a request to add the user
strTargetURL = LL_WBTMAN & "wbtm_llremote.asp?nocache=" & Server.URLEncode(Now())

strData = _
    "function=adduser" & _
    "&keystring=" & Server.URLEncode(LL_IDENT_KEY) & _
    "&username=" & Server.URLEncode(strName) & _
    "&email=" & Server.URLEncode(strEmail) & _
    "&authlevel=" & CStr(nLLAuthLevel) & _
    "&loginname=" & Server.URLEncode(strLoginID)

If LL_USE_PW Then
    strData = strData & "&password=" & Server.URLEncode(strPassword)
Else
    strData = strData & "&password=dummpw"
End If

'send the request
on error resume next
xmlHTTP.open "POST", strTargetURL, false
xmlHTTP.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
xmlHTTP.send strData

'check for runtime errors
If Err.number <> 0 Then

    strCode = Err.number & ":" & Err.description

Elseif xmlHTTP.status <> "200" Then

    strCode = xmlHTTP.status & ":" & xmlHTTP.statusText

Else 'no local errors

    'get results
    strResult = xmlHTTP.responseText
    Call ParseResults( strResult, nStatus, strCode )

    If nStatus = "0" Then
        AddLLUser = True
    End If

End If

End If

```

ll_golearnlinc.asp (continued)

```
End Function

'split result code from message in response from wbtm_llremote.asp
Sub ParseResults( ByVal result, ByRef resultcode, ByRef resultmsg )

    dim aresults

    aresults = split( result, vbcrf )
    resultcode = aresults(0)
    resultmsg = aresults(1)

End Sub

%>
```

ll_config.inc

```
<%
'WBT Manager user-customizable include file
'
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'This include file contains parameters used to control
'scripts that integrate WBT Manager with LearnLinc(TM)
'

'path to the LearnLinc server
const LL_HOME = "http://<yourserver>/learnlinc/"

'path to the WBT Manager interface script files on the LearnLinc server
const LL_WBTMAN = "http://<yourserver>/learnlinc/wbtmanager/"

'flag to indicate whether to send WBT Manager user password to LearnLinc
const LL_USE_PW = False

'flag to indicate whether users should be automatically added to LearnLinc
const LL_ADD_USERS = True

'key string to send with background posts between WBT Manager and Learnlinc
'for simple authentication. Must match value in wbtm_config.inc on the
'LearnLinc server.
const LL_IDENT_KEY = "tis me!"

%>
```

ll_msg.inc

```
<%
'WBT Manager user-customizable include file
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'This include file contains message strings used in the
'scripts that integrate WBT Manager with LearnLinc(TM)
'

const LL_MSG_NOTLOGGEDON = "You must be logged onto WBT Manager to use this Function"
const LL_MSG_ERRORCHECKINGUSER = "Error authenticating user in LearnLinc:"
const LL_MSG_ERRORADDINGUSER = "Error adding user to LearnLinc:"
const LL_MSG_LLREGREQ = "You must be registered in LearnLinc before you can use this Function"
const LL_MSG_STARTINGTITLE = "-->LearnLinc"
const LL_MSG_LLSTARTING = "LearnLinc will now start..."
const LL_MSG_LLSTARTING2 = "After you exit your LearnLinc session click the ""close"" link to close this window."
const LL_MSG_LLSTARTING3 = "<a href=""javascript:closeme();"">close</a>"
%>
```

LearnLinc server scripts

These scripts must be loaded into **llweb\wbmanager** (may need to be created) on the LearnLinc server.

wbtm_llremote.asp

```
<%@ LANGUAGE="VBSCRIPT"%>
<%
Option Explicit

'WBT Manager to LearnLinc interface script
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'WBT Manager licensees are granted the right to copy and modify this
'script as desired for use in integrating 3rd party applications
'with WBT Manager.
'
'Note: This script is based the code from the LearnLinc API file llRemoteJoin.asp.
'This script provides only a subset of the functionality of llRemoteJoin.asp.
'
' 03182003 ms - created as part of "Integrating Synchronous Tools" white-paper.

'this script demonstrates how:
'   verify that a user exists in LearnLinc
'   add a user to LearnLinc
'
'*****
'USAGE:
'
'return value:
'   For all functions except "go" this script returns a text/plain message
'   containing 2 lines of text
'   The first line contains a numeric code specific to the function called
'   The second line contains a string message specific to the function called
'
'Global parameters (always required, must be POST parameters)
'   function:      string
'                 required
'                 identifies the function requested by the caller
```

Copyright © 2003, Integrity eLearning. All rights reserved.

wbtl_llremote.asp (continued)

*Function-specific parameters (must be POST parameters)

Function: checkuser:

keystring: string
required
string identifying the caller. used for additional security in
functions called by back-end server scripts.
must match LL_IDENT_KEY value in wbtm_config.inc

loginname: string
required
WBT Manager/LearnLinc user login id

password: string
optional (required if LL_USE_PW is true in wbtm_config.inc)
WBT Manager/LearnLinc user password

return values:

User not found: "0"
<LL_MSG_USERNOTFOUND>

User found but:
pw wrong "-1"
<LL_MSG_INVALIDPW>
only returned if LL_USE_PW is true

User found: "1"
<temporary authentication key string>
(used so password need not be included in "go" function)

Function: adduser:

keystring: string
required
string identifying the caller. used for additional security in
functions called by back-end server scripts.
must match LL_IDENT_KEY value in wbtm_config.inc

loginname: string
required
WBT Manager/LearnLinc user login id

password: string
required
WBT Manager/LearnLinc user password

username: string
required
WBT Manager/LearnLinc user display name

email: string
optional
WBT Manager/LearnLinc user email address

authlevel: numeric
required
LearnLinc user authorization level

return values:

User added: "0"
<temporary authentication key string>
(used so password need not be included in "go" function)

User already exists: "1"
<temporary authentication key string>
(used so password need not be included in "go" function)

error adding user: "-1"
<LL_MSG_ERRORADDING>:<error code and message>

wbtm_llremote.asp (continued)

```
'abandon the local session
If strFunction <> "go" Then Session.Abandon
Response.End

***** end of main script *****

'authenticate the specified user against the LearnLinc database
'return a short-lived login code if user is valid
Sub CheckUser()

    Dim objLL
    Dim strLoginID
    Dim strPassword
    Dim dummy
    Dim nReturn

    'get the information needed for LearnLinc login
    strLoginID = Trim( Request.Form("loginname") )
    strPassword = Trim( Request.Form("password") )

    'check parameters
    If Len(strLoginID) < 1 Or _
       ( LL_USE_PW And Len(strPassword) < 1 ) Then

        Response.Write "-1" & vbCrLf & LL_MSG_MISSINGPARAMS
        Exit Sub

    End If

    'get the LearnLinc database object (will be created automatically by server)
    Set objLL = Session("lldbo")

    'use LookupPassword to see if the user exists
    nReturn = objLL.LookupPassword( strLoginID, dummy, dummy, dummy )
    If nReturn <> 0 Then
        Response.Write "0" & vbCrLf & LL_MSG_USERNOTFOUND
    End If

    'now check password
    If nReturn = 0 And LL_USE_PW Then

        'have to use the login function to validate the user's password
        nReturn = objLL.Login( strLoginID, strPassword, dummy )
        If nReturn <> 0 Then
            Response.Write nReturn & vbCrLf & LL_MSG_INVALIDPW
        End If

    End If

    'if user exists and pw is ok then return keystring
    ' MakeKeyString function is in wbtm_config.inc so that clients can customize it
    If nReturn = 0 Then
        Response.Write "1" & vbCrLf & MakeKeyString()
    End If

    'release the learnlinc object
    Set objLL = Nothing

End Sub
```

wbtm_llremote.asp (continued)

```

Sub AddUser()

    Dim objLL
    Dim strLoginID
    Dim strPassword
    Dim strName
    Dim strEmail
    Dim nAuthLevel
    Dim dummy
    Dim nReturn

    'get the information needed for LearnLinc login
    strLoginID = Trim( Request.Form("loginname") )
    strPassword = Trim( Request.Form("password") )
    strName = Trim( Request.Form("username") )
    strEmail = Trim( Request.Form("email") )
    nAuthLevel = Trim( Request.Form("authlevel") )

    'check parameters
    If Len(strLoginID) < 1 Or _
        Len(strPassword) < 1 Or _
        Len(strName) < 1 Or _
        Not IsNumeric(nAuthLevel) Then

        Response.Write "-1" & vbCrLf & LL_MSG_MISSINGPARAMS
        Exit Sub

    End If
    nAuthLevel = CLng(nAuthLevel)

    'get the LearnLinc database object (will be created automatically by server)
    Set objLL = Session("lldbo")

    'use LookupPassword to see if the user already exists
    nReturn = objLL.LookupPassword( strLoginID, dummy, dummy, dummy )
    If nReturn = 0 Then
        nReturn = 1 ' don't add a new student but see if this one is valid.

        If LL_USE_PW Then

            'have to use the login function to validate the user's password
            nReturn = objLL.Login( strLoginID, strPassword, dummy )
            If nReturn <> 0 Then
                Response.Write nReturn & vbCrLf & LL_MSG_INVALIDPW
                nReturn = -1
            End If

        End If

    Else
        nReturn = 0 'try to add student
    End If

    'now add the student
    If nReturn = 0 Then

        nReturn = objLL.CreateUser( strLoginID, strName, strEmail, _
            "", "", LL_MSG_AUTOADDCOMMENT, _
            strPassword, nAuthLevel, "", dummy )

    End If

```

wbtm_llremote.asp (continued)

```

'if user exists or was added then return keystring
' MakeKeyString function is in wbtm_config.inc so that clients can customize it
If nReturn = 0 Or nReturn = 1 Then
    Response.Write CStr(nReturn) & vbCrLf & MakeKeyString()
Else
    Response.Write CStr(nReturn) & vbCrLf & _
        LL_MSG_ERRORADDING & ":" & objLL.GetErrorString( nReturn )
End If

'release the learnlinc object
Set objLL = Nothing

End Sub

'go to student's home page or join a class
Sub Go()

    Dim nUserID
    Dim strEncPW
    Dim nClassID

    nClassID = Trim(Request.Form("classid"))
    If IsNumeric( nClassID ) Then
        If Login( nUserID, strEncPW ) Then
            Call Join( nUserID, strEncPW, CLng( nClassID ) )
        End If
    Else
        'log on and jump to student's home page
        If Login( nUserID, strEncPW ) Then
            Response.Redirect "../cframe.asp"
        End If
    End If

End Sub

End Sub

'log the user onto LearnLinc
'sets user id and encrypted password values if successful
Function Login( ByRef nUserID, ByRef strEncPW )

    Dim strLoginID
    Dim strKey
    Dim objLL
    Dim dummy
    Dim nReturn

    'get the information needed for LearnLinc login
    strLoginID = Trim( Request.Form("loginname") )
    strKey = Trim( Request.Form("code") )

    'check parameters
    If Len(strLoginID) < 1 Or _
        Len(strKey) < 1 Then

        Response.Write LL_MSG_MISSINGPARAMS
        nReturn = -1

    End If

    'check for valid keystring
    ' note: error message is intentionally vague
    ' CheckKeyString function is in wbtm_config.inc so that clients can customize it
    If Not CheckKeyString( strKey ) Then

        Response.Write LL_MSG_MISSINGPARAMS
        nReturn = -1

    End If

```

wbtm_llremote.asp (continued)

```
    If nReturn = 0 Then

        'get the LearnLinc database object (will be created automatically by server)
        Set objLL = Session("lldbo")

        'password is not provided
        'get the user's encrypted password for use in LoginEncrypted
        nReturn = objLL.LookupPassword( strLoginID, strEncPW, dummy, nUserID )
        If nReturn <> 0 Then
            Response.Write LL_MSG_USERNOTFOUND & ": " & nReturn & "-" & objLL.GetErrorString( nReturn
)
        End If

        'log the user on
        If nReturn = 0 Then

            nReturn = objLL.LoginEncrypted( strLoginID, strEncPW, nUserID )
            If nReturn <> 0 Then
                Response.Write LL_MSG_LOGINERROR & ": " & nReturn & "-" & _
                    objLL.GetErrorString( nReturn )
            End If

        End If

    End If

    'set session variables (from LearnLinc login page default.asp)
    If nReturn = 0 Then

        Session("TestClient") = TRUE
        Session("LoggedIn") = TRUE
        Session("Name") = nUserID
        Session("LoginName") = strLoginID
        Login = True

    Else

        Login = False
        Session.Abandon 'failed to log on

    End If

    'release the LearnLinc database object
    Set objLL = Nothing

End Function

'join a learnlink class
```

wbtm_llremote.asp (continued)

```

Sub Join( ByVal nUserID, ByVal strEncPW, ByVal nClassID )

    Dim nReturn
    Dim objLL
    Dim nCourseID
    Dim blsInstructor
    Dim strLRNFileData
    Dim dummy

    'user must be logged on
    If Not Session("LoggedIn") Then
        Response.Write LL_MSG_NOTLOGGEDON
        Exit Sub
    End If

    'get the LearnLinc database object (will be created automatically by server)
    Set objLL = Session("lldbo")

    'Is user enrolled in course containing class

    'get the course id
    nReturn = objLL.GetClass( nClassID, dummy, dummy, dummy, dummy, _
        dummy, dummy, dummy, dummy, dummy, _
        nCourseID )

    If nReturn <> 0 Then
        Response.Write LL_MSG_ERRORGETCLASS & ": " & nReturn & "-" & objLL.GetErrorString( nReturn )
    End If

    'is the student in that course
    If nReturn = 0 Then

        nReturn = objLL.GetIsRegistered( nUserID, nCourseID )

        If nReturn = 0 Then

            'enroll them?
            If LL_ENROLL_USERS Then

                blsInstructor = False
                nReturn = objLL.RegisterUser( nUserID, nCourseID, blsInstructor )
                If nReturn <> 0 Then
                    Response.Write LL_MSG_ERRORENROLLING & ":" & _
                        nReturn & "-" & objLL.GetErrorString( nReturn )
                End If

            Else

                nReturn = -1
                Response.Write LL_MSG_NOTENROLLED

            End If

        ElseIf nReturn = 1 Then

            nReturn = 0 'user is enrolled, continue launch

        Else

            Response.Write LL_MSG_ERRORCHECKENROLL & ":" & _
                nReturn & "-" & objLL.GetErrorString( nReturn )

        End If

    End If

    'launch class

```

wbtm_llremote.asp (continued)

```

    If nReturn = 0 Then
        nReturn = objLL.CreateLRNFile( Application("server"), CInt(Application("port")), _
            nUserID, CStr(strEncPW), CInt(nClassID), _
            CInt(0), "", CInt(0), "", strLRNFileData )
        'Make sure that create worked
        If nReturn <> 0 Then
            Response.Write "Creation of LRN file failed.<BR>"
            Response.Write "Here is the Error: " & nReturn & "-" & objLL.GetErrorString( nReturn ) & "<BR>"
            Response.Write "Server: " & Application("server") & "<BR>"
            Response.Write "port: " & Application("port") & "<BR>"
            Response.Write "UserID: " & nUserID & "<BR>"
            Response.Write "UserPassword: " & strEncPW & "<BR>"
            Response.Write "ClassID: " & nClassID & "<BR>"
            Response.Write "Here is the Data: " & strLRNFileData & "<BR>"

            Session.Abandon
        End If
    Else
        'clear text/plain content type an send LRN data.
        Response.Clear
        Response.ContentType ="application/x-learnlinc"
        Response.Write strLRNFileData
    End If
Else
    Session.Abandon 'failed to join
End If

'release the learnlinc object
Set objLL = Nothing

End Sub

'log the user out of LearnLinc
' and close the browser window
Sub LogoutAndClose()

    'remove session variables
    Session.Abandon

    'output HTML page with window.close command
    Response.Clear

    Response.Write _
        "<html><head></head><body onload=""window.close()""></body></html>"

End Sub
%>

```

wbtm_config.inc

```
<%
'WBT Manager user-customizable include file
'
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'This include file contains parameters used to control
'scripts that integrate WBT Manager with LearnLinc(TM)
'

'path to the WBT Manager application
const WBTMAN_HOME = "http://<yourserver>/wbtmanager/"

'Flag to indicate that LearnLinc should validate against user's learnlinc password.
'If this flag is set to True then the user's password in LearnLinc must always
'match their password in WBT Manager. If set to False then the login process on
'LearnLinc will ignore passwords.
const LL_USE_PW = False

'flag to indicate whether users should be automatically added to LearnLinc
const LL_ADD_USERS = True

'flag to indicate whether users should be automatically enrolled in courses
const LL_ENROLL_USERS = True

'key string that will be sent with background posts between WBT Manager and Learnlinc
'for simple authentication. Must match value in ll_config.inc on the
'WBT Manager server.
const LL_IDENT_KEY = "'tis me!"

*****
'keysting code below is used so that user's passwords need not be included
'in the self-submitting form that takes them to LearnLinc.
'In default version the keysting is a date and is good for 10 seconds
' Functions MakeKeyString and CheckKeyString are required.
*****
const LL_KEY_GOOD_FOR = 10

'array that controls the order in which the date digits are placed in the key string
Dim aKeyScramble(14)

aKeyScramble(1) = 14
aKeyScramble(2) = 2
aKeyScramble(3) = 12
aKeyScramble(4) = 6
aKeyScramble(5) = 3
aKeyScramble(6) = 10
aKeyScramble(7) = 8
aKeyScramble(8) = 4
aKeyScramble(9) = 5
aKeyScramble(10) = 13
aKeyScramble(11) = 7
aKeyScramble(12) = 11
aKeyScramble(13) = 9
aKeyScramble(14) = 1

'do not rename this function or alter its parameter list
Function MakeKeyString()

    'default keysting is the current date/time
    'scrambled up.
    MakeKeyString = ScrambleKey( Now() )

End Function
```

wbtm_config.inc (continued)

```

'do not rename this function or alter its parameter list
Function CheckKeyString( ByVal strKey )

    Dim dtKey

    CheckKeyString = False
    strKey = UnscrambleKey( strKey )

    If IsDate(strKey) Then

        dtKey = CDate(strKey)

        'time in key must be within LL_KEY_GOOD_FOR seconds of now
        If dtKey <= Now() And _
            DateAdd("s", LL_KEY_GOOD_FOR, dtKey) >= Now() Then
            CheckKeyString = True
        End If

    End If

End Function

'obfuscate the key date a bit
'NOTE: this is not secure encryption, just makes it a little more difficult to
'guess what the keystring should be.
Function ScrambleKey( dtKey )

    Dim strKey
    Dim yr, mon, dy, hr, min, sec
    Dim i

    'input must be a date
    If Not IsDate( dtKey ) Then dtKey = Now()
    dtKey = CDate(dtKey)

    'change the date into a string with no punctuation
    yr = CStr(Year(dtKey))
    mon = Right( "0" & CStr(Month(dtKey)), 2 )
    dy = Right( "0" & CStr(Day(dtKey)), 2 )
    hr = Right( "0" & CStr(Hour(dtKey)), 2 )
    min = Right( "0" & CStr(Minute(dtKey)), 2 )
    sec = Right( "0" & CStr(Second(dtKey)), 2 )

    strKey = yr & mon & dy & hr & min & sec

    'scramble the characters
    ScrambleKey = ""
    for i = 1 to 14
        ScrambleKey = ScrambleKey & _
            ( Mid( strKey, aKeyScramble(i), 1 ) )
    Next

End Function

```

wbtm_config.inc (continued)

```
'de-obfuscate the key date
'NOTE: this is not secure encryption
Function UnscrambleKey( ByVal strKey )

    Dim aKey(14)
    Dim i

    aKey(0)=""

    'unscramble the key string
    For i = 1 To 14
        aKey( aKeyScramble(i) ) = _
            ( Mid( strKey, i, 1 ) )
    Next
    strKey = ""
    For i = 1 To 14
        strKey = strKey & aKey(i)
    Next

    UnscrambleKey = Mid(strKey, 1,4) & "-" & _
        Mid(strKey, 5, 2) & "-" & _
        Mid(strKey, 7, 2) & "-" & _
        Mid(strKey, 9, 2) & "-" & _
        Mid(strKey, 11, 2) & "-" & _
        Mid(strKey, 13, 2)

End Function
%>
```

wbtm_msg.inc

```
<%
WBT Manager user-customizable include file
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'This include file contains message strings used in the
'scripts that integrate WBT Manager with LearnLinc(TM)
'

const LL_MSG_INVALIDFUNCTION = "Invalid function"
const LL_MSG_USERNOTFOUND = "User not found in LearnLinc Database"
const LL_MSG_INVALIDPW = "Invalid password"
const LL_MSG_LOGINERROR = "Error logging onto LearnLinc"
const LL_MSG_NOTLOGGEDON = "Not logged on"
const LL_MSG_MISSINGPARAMS = "Missing required parameters"
const LL_MSG_ERRORADDING = "Error adding user"
const LL_MSG_AUTOADDCOMMENT = "Registered automatically from WBT Manager"
const LL_MSG_ERRORGETCLASS = "Error getting class info"
const LL_MSG_ERRORCHECKENROLL = "Error checking course enrollment"
const LL_MSG_ERRORENROLLING = "Error enrolling in course"
const LL_MSG_NOTENROLLED = "You are not enrolled in that course"

%>
```

wbtm_listids.asp

```
<%@ LANGUAGE="VBSCRIPT"%>
<%
Option Explicit

'WBT Manager to LearnLinc interface script
'
'Copyright © 2003 Integrity eLearning. All Rights Reserved.
'
'WBT Manager licensees are granted the right to copy and modify this
'script as desired for use in integrating 3rd party applications
'with WBT Manager.
'
'Note: This script is based the code from the LearnLinc API file IIRemoteJoin.asp.
'This script provides only a subset of the functionality of IIRemoteJoin.asp.
'
' 03182003 ms - created as part of "Integrating Synchronous Tools" white-paper.

'this script displays the internal ids for courses and classes in the LearnLinc
'database

'URL parameters
'
'      courseid: numeric string
'                optional
'                LearnLinc course id
'                If this is present then script will display classes in the course.
'
Dim nCourseID
Dim objLL

nCourseID = Trim( Request("courseid") )

'get the LearnLinc database object (will be created automatically by server)
Set objLL = Session("lldbo")

Response.Write "<html><head></head><body>"

If IsNumeric( nCourseID ) Then
    Call ListClasses( objLL, nCourseID )
Else
    Call ListCourses( objLL )
End If

Response.Write "</body></html>"

'release the LearnLinc database object
Set objLL = Nothing
Response.End
```

wbtm_listids.asp (continued)

```

Sub ListCourses( ByRef objLL )

    Dim strCourses
    Dim nResult
    Dim aCourses
    Dim i

    'get a list of courses from the LearnLinc object
    nResult = objLL.GetCourses( "%", strCourses )
    If nResult <> 0 Then
        Response.Write nResult & "." & objLL.GetErrorString( nResult )
    Else

        aCourses = Split( strCourses, vbLF )

        'output list of courses to select from
        Response.Write "<form action=""wbtm_listids.asp"">" & vbCrLf
        Response.Write "<select name=""courseid"">" & vbCrLf

        For i = 1 To UBound( aCourses ) Step 2
            Response.Write "<option value="" & aCourses(i-1) & """">" & _
                Server.URLEncode( aCourses(i) ) & " (ID:" & aCourses(i-1) & ")</option>" & vbCrLf
        Next

        Response.Write "</select>" & vbCrLf
        Response.Write "<input type=""submit"">" & vbCrLf
        Response.Write "</form>" & vbCrLf

    End If

End Sub

Sub ListClasses( ByRef objLL, ByVal nCourseID )

    Dim strClasses
    Dim nResult
    Dim aClasses
    Dim i
    Dim strCourseName
    Dim dummy

    'get name of course
    nResult = objLL.GetCourse( nCourseID, strCourseName, _
        dummy, dummy, dummy, dummy, dummy, dummy )
    nResult = objLL.GetClasses( nCourseID, strClasses )
    If nResult <> 0 Then
        Response.Write nResult & "." & objLL.GetErrorString( nResult )
    Else
        Response.Write "<h3>" & Server.URLEncode( strCourseName ) & "</h3>" & vbCrLf
    End If

    'get a list of classes from the LearnLinc object

```

wbtm_listids.asp (continued)

```
nResult = objLL.GetClasses( nCourseID, strClasses )
If nResult <> 0 Then
    Response.Write nResult & ":" & objLL.GetErrorString( nResult )
Else
    aClasses = Split( strClasses, vbLF )

    'output list of classes
    Response.Write "<table border=""0"" cellpadding=""5"">" & vbCrLf
    For i = 1 To UBound( aClasses ) Step 2
        Response.Write "<tr><td>" & _
            Server.HtmlEncode( aClasses(i) ) & "</td><td>" & _
            aClasses(i-1) & "</td></tr>" & vbCrLf
    Next

    Response.Write "</table>" & vbCrLf
    Response.Write "<p><a href=""wbtm_listids.asp"">Go back</a></p>" & vbCrLf

End If

End Sub

%>
```