
WBT Manager™

Installation and Customization Manual

By Integrity eLearning



This manual was produced using *Doc-To-Help*[®], by WexTech Systems, Inc.

Microsoft[®], Windows[®], Windows NT[®], ActiveX[®], Visual Basic[®] and Visual Studio[®] are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Crystal Reports[™] is either a registered trademark or trademark of Seagate Software, Inc. Seagate[®] is a registered trademark of both Seagate Technology and Seagate Software.

Oracle[®] is a registered trademark and Oracle8[™] is a trademark or registered trademark of Oracle Corporation.

Netscape and the Netscape N and Ship's Wheel logos are registered trademarks of Netscape Communications Corporation in the U.S. and other countries.

Netscape Navigator and Netscape Communicator are also trademarks of Netscape Communications Corporation and may be registered outside the U.S.

Contents

Installing WBT Manager™	1-1
How WBT Manager works	1-1
Performing a New Installation	1-2
Before You Begin	1-2
Installing the Web Server Module	1-4
Upgrading WBT Manager from version 1.3	1-6
Before you Begin	1-6
The Upgrade Process	1-7
Sample Upgrade Task Plan (1.3-1.51).....	1-8
Upgrading WBT Manager from version 1.50	1-13
Before you Begin	1-13
The Upgrade Process	1-13
Sample Upgrade Task Plan (1.50-1.51)	1-14
Using a different DBMS	1-19
About Shared Status	1-19
Customizing the WBT Manager™ Web Interface	2-1
Overview	2-1
Proprietary Requirements	2-1
Customizable files	2-1
<root>/settings.asp	2-2
<root>/default.asp*, <root>/asp/set1.asp*	2-5
<root>/html/ banner.htm.....	2-5
<root>/html/home.htm.....	2-5
<root>/html/ie.css*, <root>/html/ns.css*	2-6
<root>/asp/logoff.inc*	2-6
<root>/asp/amenu1.inc,.....	2-6
<root>/asp/amenu2.inc.....	2-6
Button and Background Image Files	2-6
A Simple Customization Case Study	2-8
Menu customization	2-9
Standard Menu Files	2-9
To Create a Custom Menu.....	2-9
How the menus work.....	2-10
The WBT Manager utility object	2-10
Session variables of interest.....	2-22
Direct lesson launch using austart.asp	2-23
Using austart.asp.....	2-23
WBT Manager Logo requirements	2-25
General Requirements	2-25
Logo Usage	2-25

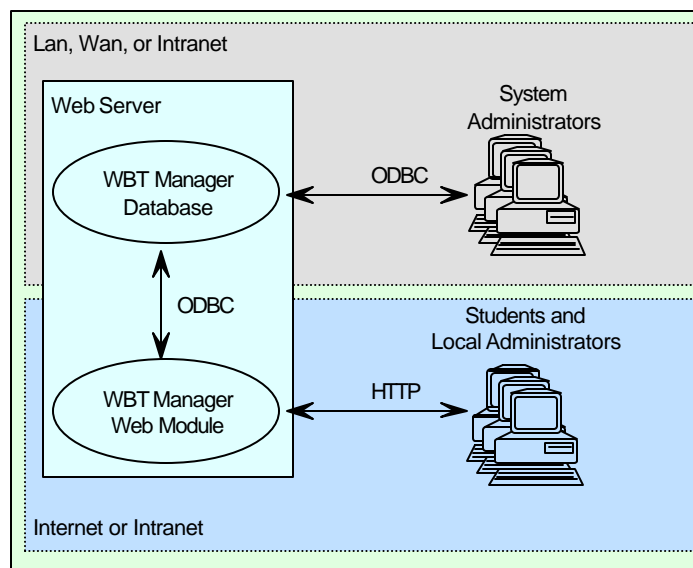
Installing WBT Manager™

Installation of WBT Manager™ consists of loading the Web Server Module on the designated system and loading the System Administrator Module on as many client systems as required. There are several required steps to performing a successful installation. A basic understanding of how WBT Manager components are connected together is essential.

How WBT Manager works

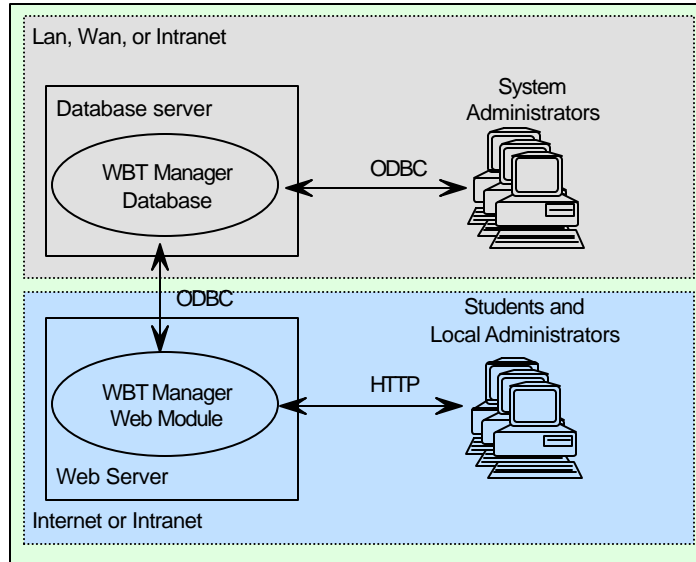
WBT Manager consists of two modules. The Web Server module is an Active Server Pages (ASP) application that resides on the web server. This module provides browser-based interfaces for both students and local administrators. The System Administrator Module resides on a network workstation and provides the System Administrator interface. See the pertinent sections of the WBT Manager User's Guide for descriptions of these interfaces.

Both modules communicate with the WBT Manager database through ODBC connections as shown here.



{bmc blkdia1.gif}

Fig 1 – WBT Manager block diagram showing default installation. MS Access database resides physically on the web server



{bmc blkdia2.gif}

Fig. 2 – WBT Manager block diagram showing installation with a DBMS on a separate server.

The ODBC connections are critical to the operation of WBT Manager. The DBMS and ODBC drivers must support ADO recordsets from the Web Server Module as well as MFC (Microsoft Foundation Classes) recordsets from the System Administrator Module. In general, this requires a Level 2 ODBC driver. It is also desirable that the DBMS support transactions. The MS Access database and ODBC drivers shipped with WBT Manager meet these requirements. Other compatible DBMS/driver combinations are listed in "Using a different DBMS" below or on the WBT Manager support web site.

The System Administrator Module updates the database directly through its database connection. User interactions with the graphical interface are translated into SQL statements that retrieve data from or update tables in the database.

The student and local administrator interfaces have no direct connection with the database. The Web Server Module generates forms and links on the HTML pages that are sent to the user's web browser. Information is sent back to the Web Server Module in HTTP POST requests. The Web Server Module translates the posted data into SQL statements to retrieve data from or update tables in the database.

Performing a New Installation

Before You Begin

Before beginning to install WBT Manager, you will need to make some preliminary decisions and ensure that your system is correctly set up for the installation. You should also familiarize yourself with the recommended sequence for installing the various components.

Preliminary decisions

1. Select a database engine.

WBT Manager ships with a pre-configured Microsoft Access database. For most production uses you will want to use a server-based DBMS.

WBT Manager is fully supported under Oracle and Microsoft SQLServer.

2. Select a location on the web server for the web application files.
This location should be on an NTFS volume so that file permissions may be set.
3. Identify one or more systems to receive the System Administrator module.
These systems must be capable of making an ODBC connection to the database.
4. Select a DisplayName format for student and administrator names.
You may set the system to use <firstname> <middlename> <lastname> <suffix> or <lastname> <suffix>,<firstname> <middlename>. You can change this format later, but doing so will require a conversion run through the database.
5. Decide whether to enable **lesson status sharing**.
See "About Shared Status" on page 19 for a description of this feature and its pros and cons. Lesson status sharing may be enabled at any time but could require a lengthy conversion if there is student results data in the statistics tables.

Note: Once you have enabled lesson status sharing, you cannot disable it without completely rebuilding the database. This will cause the loss of all existing data. Make sure that you understand the implications of this option before enabling it.

6. Decide if you want to install the **External Script Interface Kit**.
This kit allows you to construct store-front interfaces and/or self-registration forms. This component can be installed later if desired.
7. Decide if you want to install the **Student Import Toolkit**.
This kit allows bulk import of student records. This component can be installed later. It is installed on s System Administrator's workstation.
8. Decide if you want to install the **Batch Maintenance Module** on the web server.
This module performs periodic purges of database tables and can send email notifications for certain events.

Pre-installation steps

Before beginning the installation process make sure that:

1. The web server software is up to date.
WBT Manager requires Microsoft Data Access Components version 2.5 or higher, along with IIS4 or higher. A copy of the MDAC installation set is on the CD in the utilities/database/mda2 directory. You can also download the latest version from **www.microsoft.com/data**.
2. The IIS, ASP, and the ADO database objects work properly.
There is a set of test scripts in the **utilities\trouble_shooting\server_check** directory on the CD

that may help. Use the **checkado.asp** script to verify that the ADO components are working.

3. The selected DBMS is installed and working.

MS Access will be installed and configured by the setup programs. Instructions for setting up MS SQLServer and Oracle may found in **/documentation/oraclesetup.html** and **/documentation/sqlserversetup.html**.

Installation sequence

The recommended installation sequence is:

1. Install the web server application on the web server.
2. Install the External Script Interface Kit on the web server (if desired).
3. Install the Batch Maintenance Module on the web server (if desired).
4. Install the System Administrator module on one or more systems.
5. Install the Student Import toolkit on one or more administrator systems (if desired).
6. Install the WBT Manager license key
7. Set the name format.
8. Convert to shared lesson status (if desired).

You can launch steps 1 through 5 from the main setup program (wbtman_setup.exe) on the CD. You can also launch them directly from their respective directories on the CD image.

Installing the Web Server Module

The help file **Web_Server_Application\readme2.hlp** on the WBT Manager distribution CD contains a step by step installation procedure to walk you through the installation. You should use this help file during the installation process. What follows is a general description of the installation procedure.

IIS must be installed and operating correctly before proceeding with the installation. ADO, ASP, and ASP session support must be available.

If you are going to be using a DBMS other than the MS Access database included with WBT Manager, you should set it up before starting the installation. See "Using a different DBMS" below.

Installing the External Script Interface kit

The External Script Interface kit is a set of scripts designed to allow custom forms and/or external web applications to add students and assign students to courses. It consists of a set of ASP scripts and sample forms that can be used as a starting point. Documentation for this kit is in **documentation\ExtScriptInt.html** on the CD.

To install the kit, select it from the main Setup Menu or directly execute **utilities\script_interface_kit\setup.exe**. The setup program will prompt you for the location of the WBT Manager Web-Server Application. Enter the path to that directory as you did when installing the Web-Server Application. The setup program will copy a number of scripts to the **/asp**, **/html**, and **/msgtemplates** directories.

Note: Installing this kit allows external systems to add students and assign courses without being validated against the WBT Manager administrator access tables. There are customizable files that you can use to help protect against unauthorized use. See **ExtScriptInt.html** for information.

Installing the Batch Maintenance Module

The batch maintenance module is a program that runs as a scheduled job on the web-server. It has the ability to purge various data tables by date and to send email notifications for various events. See the help file

\utilities\maintenance_module\wbmmnt.hlp for more details.

To install the module, select it from the main Setup Menu or directly execute **\utilities\maintenance_module\setup.exe**. Select a directory to install the programs and support files into.

See the help file for information about setting parameters and scheduling runs.

Installing the System Administrator Module

The System Administrator Module installation is fully automated. Run the setup program **Administrator Module\setup.exe** to begin the installation.

The installation program will first check that you have Microsoft Data Access Components version 2.5 or higher installed. If not, you will be prompted to install the latest version from the Microsoft data access site. A copy of the MDAC installation program for MDAC 2.5 can be found in the **\utilities\database\mda2** directory on the installation CD.

The installation program will copy WBT Manager files to the selected destination directory and set up a shortcut on the program menu. Setup will not overwrite newer versions of these files. Files installed or updated in the windows or windows\system directories are:

- Crystal reports files placed in the windows\crystal directory:
u2dmapi.dll, u2ddisk.dll, u2dpost.dll, u2fsepv.dll, u2fdif.dll, u2fhtml.dll, u2fodbc.dll, u2frec.dll, u2frtf.dll, u2fcr.dll, u2ftext.dll, u2fwks.dll, u2fwordw.dll, u2fxls.dll, crxlat32.dll, u2lsamp1.dll, p2ixbse.dll, p2lodbc.dll, u252000.dll, u25dts.dll, u25store.dll, u25total.dll, u2dnotes.dll, u2dvim.dll, u2frdef.dll, u2l2000.dll, u2lbar.dll, u2lcom.dll, u2ldts.dll, u2lexch.dll, u2lfinra.dll
- Crystal reports files placed in the windows\system directory:
crpe32.dll, implode.dll, p2sodbc.dll, crpaig32.dll, msvcr40.dll, msvcr20.dll, msvcr.dll, crpe32.dep, ctl3d32.dll

Installing the Student Import Toolkit

The Student Import Toolkit is an ActiveX component that allows programs to access student data. Sample programs written in Visual Basic are also included. The sample programs allow bulk import of student records and status values.

To install the kit, select it from the main Setup Menu or directly execute **\utilities\student_import_kit\setup.exe**.

Installing the license key

The license key is installed by starting the System Administrator Module for the first time. An invalid license key error will be displayed, followed by the License Key Entry dialog. Enter the license key exactly as it appears on your license key memo. Once the key has been installed, you will be prompted for the administrator ID and password. The default ID for new installations is **“admin”**. The default password is also **“admin”**.

Setting the display name format

The student and administrator display name format defaults to <first> <middle> <last> <suffix>. If you would like to use <last> <suffix>, <first> <middle>, you will need to run the site configuration tool. Documentation on using this tool is in **\utilities\site_configuration\site_configuration_utility.html**. It can also be found in the "Utilities" chapter of the main manual.

1. Launch the utility from the main Setup Menu, or directly execute **\utilities\site_configuration\wbtmconfig.exe**.

Notes: If you are using Oracle you will need to set up a DSN to the WBT Manager database using the **Microsoft ODBC for Oracle** driver from Microsoft Data Access Components version 2.5. The Oracle ODBC driver does not support the recordset types needed by the Visual Basic components.

If you did not install the Student Import Toolkit on the system from which you will be running the site configuration utility, you will need to install the Visual Basic 6 runtime and support files. To do this, execute **utilities\site_configuration\setup.exe** and accept the defaults.

Converting to shared lesson status

2. Select the desired name format from the first tab in the form. You can also modify the captions and the local administrator access for the user defined text fields in the student record.

Shared lesson status causes a student's status for a lesson to be shared between courses containing the same lesson if more than one such course is assigned to the student.

Warning: Once shared status is enabled for a site it cannot be disabled without recreating the database. Read "About Shared Status" on page 19 carefully before running the conversion utility.

To convert the database to shared lesson status, execute **utilities\conversion_tools\shared_status\sharestats.exe**. This program will make changes to the standard database structure.

Note: If you are using Oracle, you will need to set up a DSN to the WBT Manager database using the **Microsoft ODBC for Oracle** driver from Microsoft Data Access Components version 2.5. The Oracle ODBC driver does not support the recordset types needed by the Visual Basic components.

Upgrading WBT Manager from version 1.3

Before you Begin

Before beginning to upgrade to WBT Manager 1.51, you will need to make some preliminary decisions.

Version 1.51 of WBT Manager has made significant changes to the database structure that was used in version 1.3. In order to upgrade from 1.3, you will need to modify the structure of your database. SQL command scripts have been supplied for this purpose.

Preliminary decisions

1. Select a DisplayName format for student and administrator names.

You may set the system to use <firstname> <middlename> <lastname> <suffix> or <lastname> <suffix>, <firstname> <middlename>. You can change this format later, but doing so will require a conversion run through the database.

2. Decide whether to enable "lesson status sharing".

See "About Shared Status" on page 19 for a description of this feature and its pros and cons. Lesson status sharing may be enabled at any

time but could require a lengthy conversion if there is student results data in the statistics tables.

Note: Once you have enabled lesson status sharing, you cannot disable it without completely rebuilding the database. This will cause the loss of all existing data. Make sure that you understand the implications of this option before enabling it.

3. Decide if you want to install the **External Script Interface Kit**.
This kit allows you to construct storefront interfaces and/or self-registration forms. This component can be installed later if desired. It is installed on the web server.
4. Decide if you want to install the **Student Import Toolkit**.
This kit allows bulk import of student records. This component can be installed later. It is installed on a System Administrator's workstation.
5. Decide if you want to install the **Batch Maintenance Module** on the web server.
This module performs periodic purges of database tables and can send email notifications for certain events. This component can be installed later if desired. It is installed on the web server.
6. Decide on an approach to upgrading the web-server application. There are two possibilities:
 - Install WBT Manager 1.51 on top of the existing installation directory.
The advantage of this approach is that it does not require a great deal of file copying. The disadvantage is that it destroys the previous version of WBT Manager.
Note that if you use this approach you must *not* uninstall version 1.3 using the add/remove programs applet in the control panel. Doing so would remove part of the new installation.
 - Copy the contents of the existing WBT Manager directory to a new directory. Then install WBT Manager 1.51 to the new directory and change the mapping in IIS Administrator to point to the new directory (recommended).
The advantage of this approach is that it maintains the previous version. The disadvantage is that it requires more file copying. The sample upgrade task plan below assumes this approach.

The Upgrade Process

The fact that you are upgrading rather than performing a new installation indicates that you have data in your database that you wish to preserve. The process described here provides precautions that are important in preserving that data.

Creating an upgrade task plan

Upgrading is a complex process. To avoid the risk of losing data, you should develop a detailed task plan before beginning to make the upgrade. The task plan should contain provisions for:

- backing up the database and the WBT Manager virtual directory
- restoring to the previous state in case the upgrade must be aborted.

A sample task plan is provided in the "Sample Upgrade Task Plan (1.3-1.51)" section on page 8. You will need to modify this plan to take into account your specific circumstances.

Performing trial runs

Before beginning the actual upgrade, you should perform one or more trial runs of the installation and conversion process using the following steps.

1. Create a test WBT Manager site identical to the production installation (preferably on a separate test server). To do so:
 - a) Install WBT Manager Web-Server Application version 1.3.
 - b) Copy any changed files from the production site.
 - c) Make a copy of the production database.
 - d) For MS Access: Copy wbtman.mdb from the production site database directory to the test site database directory when no users are logged in.
 - e) For MS SQLServer: Use the DBMS tools to perform a full backup of the production database and restore that backup to a new database. Do not use the Export and Import wizards to copy the database. These wizards do not transfer the database indices and constraints.
 - f) For Oracle: Use the DBMS tools to perform a full backup of the production database and restore that backup to a new database.
 - g) Link the copy to the test site by creating a DSN to the new database and modifying the settings.asp file (line 2) to use that DSN.
 - h) Verify that the test site functions exactly as the production site before proceeding.
2. Execute the conversion task plan against the test site.

You will probably want to keep track of the elapsed time for the complete conversion. This information will help you in scheduling downtime for the production site.
3. Modify the task plan as necessary. Then repeat steps 1 and 2 until the conversion proceeds with no problems.
4. Verify the behavior of the test site.
5. Identify any new graphics that need to be customized. There are several new buttons in version 1.5.
6. Identify any customizations to the standard files (crsmenu.asp, menu.asp, clsmenu.asp, nav.asp, set1.asp, and default.asp) that may need to be re-done in the new version.

Sample Upgrade Task Plan (1.3-1.51)

Note: This is a generic task plan. You should modify it to match your specific circumstances.

Preliminary tasks

1. Determine the current database schema version.

The upgrade scripts supplied with WBT Manager 1.51 are designed to upgrade the database from version 1.02 or 1.03 to 1.07. You can view the current database schema version by selecting "About WBT Manager..." from the system administrator help menu. Look for "DB

Version". If it is not 1.02 or 1.03, contact your vendor for customer support.

- Version _____

2. Schedule site downtime.

Use the elapsed times from your trial run conversions to schedule site downtime. Note that the MDAC 2.5 upgrade may require that the entire server be re-booted.

- Site will be down starting _____ until approximately _____

3. Identify new directory.

- Install new version to _____.

4. Verify disk space

- Database server:

Size of full database backup file: _____

Available disk space

(need at least 2 times backup file size) _____

- Web server:

Size of WBT Manager virtual directory _____

Available disk space in target directory

(need enough to duplicate original +20MB) _____

Conversion tasks

1. Re-verify disk space

- Database server ok? _____
- Web server ok? _____

2. Disable production site.

- <specify steps here>

3. Upgrade MDAC on web-server if needed.

4. Back up database.

- <specify steps here>

5. Copy production directory to new directory.

6. Run database upgrade scripts.

- a) Execute **runssql.exe** from the **\utilities\database** directory on the WBT Manager CD.

- b) Click the [...] button next to **Data Source Name** and select the DSN which connects to the production database.

If you are prompted for a Login ID and Password, use the database account that was originally used to set up the WBT Manager database.

- c) Make sure that the **Pause on SQL Error** and **Remove trailing semi-colons before executing** check boxes are checked and that **Pause between statements** is un-checked.

- d) Execute each of the files in the appropriate table below, in the order shown. To execute a script file:
1. Click the [...] button next to **SQL Script File**.
 2. Browse down to the **utilities\database\scripts** directory on the WBT Manager CD.
 3. Select the file and click **Open**.
 4. Click the **Execute** button.

Scripts for MS Access

Script filename	Notes
access_102_to_103.sql	skip if database version is 1.03 already
access_103_to_104.sql	
access_104_to_105.sql	
access_105_to_106.sql	
access_106_to_107.sql	

Scripts for MS SQLServer

Script filename	Notes
sqlserv_102_to_103.sql	skip if database version is 1.03 already
sqlserv_103_to_104.sql	
sqlserv_104_to_105.sql	
sqlserv_105_to_106.sql	
sqlserv_106_to_107.sql	
grantdb.sql	modify the Login ID wbtmanuser in this file to match the Login ID used by the web-server application on line 2 of settings.asp

Scripts for Oracle

Script filename	Notes
oracle_102_to_103.sql	skip if database version is 1.03 already
oracle_103_to_104.sql	
oracle_104_to_105.sql	
oracle_105_to_106.sql	
oracle_106_to_107.sql	
grantdb.sql	modify the Login ID wbtmanuser in this file to match the Login ID used by the web-server application on line 2 of settings.asp
orasyn.sql	

7. Install Version 1.51 of the Web-Server Application

Install the new version of WBT Manager on top of the copy of the production version in the new directory. See "Installing the Web Server Module" on page 4 for specific details.

8. Install External Script Interface Kit (if desired).
See "Installing the External Script Interface kit" on page 4 for details.
9. Install batch maintenance module (if desired).
See "Installing the Batch Maintenance Module" on page 5 for details.
10. Move style-sheet settings.

In WBT Manager 1.51, style settings are located in the HTML directory in standard style-sheet (.css) files. If you have customized the styles in **pgstyle.asp**, you will need to copy the changes to the new style-sheets. There are two style-sheet files, **ie.css** for Microsoft Internet Explorer and **ns.css** for NetScape. Replace the contents of each of these files with the text between the <STYLE> and </STYLE> tags in **pgstyle.asp**. Then rename or delete **pgstyle.asp**.

11. Re-make any custom changes to menus.

If you have made any changes to **menu.asp**, **crsmenu.asp**, or **clsmenu.asp**, you will need to make the equivalent changes in the new versions of those files.

12. Add missing custom graphics.

The following graphics are new with WBT Manager 1.50 or 1.51. You may want to create custom versions of them to match your site.

Filename(s)	description
crsstatus.gif*	Recalc Status button on course menu
minus.gif* plus.gif*	Expand/collapse symbols from tree-style course menu
tab0.gif – tab9.gif*	Tabs from tabbed-style course menu
tcomp.gif* tincomp.gif* tnotstart.gif* tdesc.gif*	status and description symbols from tabbed-style course menu
edit.gif**	“Edit” button. Replaces “Drop” button on student course enrollment form. “Drop” button is now on student course assignment form.
boffline.gif** offline.gif**	“Go Off-Line” button and offline status indicator. Used with optional Offline Student Module.

* added in 1.50

** added in 1.51

13. Install version 1.51 System Administrator module.

See "Installing the System Administrator Module" on page 5 for more details. Do not install to the same directory as WBT Manager 1.3. If you do so, uninstalling WBT Manager 1.3 will disable WBT Manager 1.51.

14. Install the version 1.51 Student Import Toolkit (if desired).

See "Installing the Student Import Toolkit" on page 5 for more details. Do not install to the same directory as WBT Manager 1.3. If you do so, uninstalling the version 1.3 toolkit will disable the version 1.51 toolkit.

15. Install your new license key.

Because of the changes in database structure, a new key is required for WBT Manager 1.5 to prevent older versions of the System Administrator or Web Application from damaging the data in the database. After the new System Administrator Module is installed, start it and enter the new license key when prompted. If you have not received a new key, contact your vendor.

16. Run the name conversion program (optional).

The student and administrator name fields have been split into first, middle, last, and suffix parts. If you would like to populate the new fields from the existing display name field, then execute **utilities\conversion_tools\name_conversion\cvtname.exe**. See the documentation file **cvtname.html** in the same directory for more details.

If you do not run the name conversion at this time the system will still operate. However, the following issues will occur:

- The Student Import Toolkit will be unable to match students by name.
- Any time an administrator attempts to edit a student or another administrator, he/she will have to fill in the individual name component fields.

The conversion program can also be run at a later date.

17. Set the name format as desired.

See "Setting the display name format" on page 5 for more details. Note that if you did not run the name conversion program as described in the previous step, the Site Configuration Tool will not be able to reset the display names of existing students to the new format. Newly entered and edited names will conform to the new format.

18. Run the shared status conversion (if desired).

See "Converting to shared lesson status" on page 6 for more details.

Warning: Once shared status is enabled for a site, it cannot be disabled without recreating the database. Read "About Shared Status" on page 19 carefully before running the conversion utility.

19. Verify operation of site.

Verify to your satisfaction that the new version of the site is operating as desired. Do not continue with the upgrade until you are satisfied.

- <specify your own test steps and approval criteria here>

20. Perform a full backup of the current database.

- <specify your own backup steps here>

21. Modify IIS configuration to point the old virtual directory name at the new installation directory.

22. Release the site to users.

- <specify your own steps here>

23. Start status rebuild program (optional).

WBT Manager 1.5 now tracks status at the Instructional Block level as well as at the lesson level. If your courseware makes use of Instructional Blocks, you may want to build out the block statuses based on the current lesson statuses.

This can be a very time-consuming process. In one case an 8,000-student database took 4 hours to rebuild.

If you do not choose to perform this step, the block statuses will gradually be built as students take lessons in the courses.

To rebuild the statuses, execute

utilities\conversion_tools\status_rebuild\rebstats.exe. See the documentation file **rebstats.html** in the same directory for more details.

1. Restore the database from the full backup made in step 4 of **conversion tasks**

- <specify your own restore steps here>

2. Release original site to users.

- <specify your own steps here>

Aborting a failed conversion

Upgrading WBT Manager from version 1.50

Before you Begin

Version 1.51 of WBT Manager has made moderate changes to the database structure that was used in version 1.50. In order to upgrade from 1.50, you will need to modify the structure of your database. SQL command scripts have been supplied for this purpose.

The Upgrade Process

The fact that you are upgrading rather than performing a new installation indicates that you have data in your database that you wish to preserve. The process described here provides precautions that are important in preserving that data.

Creating an upgrade task plan

Upgrading from version 1.50 to 1.51 is a fairly simple process but, like any operation involving live data, you will want to guard against unforeseen problems. To avoid the risk of losing data, you should develop a detailed task plan before beginning to make the upgrade. The task plan should contain provisions for:

- backing up the database and the WBT Manager virtual directory
- restoring to the previous state in case the upgrade must be aborted.

A sample task plan is provided in the “Sample Upgrade Task Plan (1.50-1.51)” section on page 14. You will need to modify this plan to take into account your specific circumstances.

Performing trial runs

Before beginning the actual upgrade, you should perform one or more trial runs of the installation and conversion process using the following steps.

1. Create a test WBT Manager site identical to the production installation (preferably on a separate test server). To do so:

- a) Install WBT Manager Web-Server Application version 1.50.
 - b) Copy any changed files from the production site.
 - c) Make a copy of the production database.
 - d) For MS Access: Copy wbtman.mdb from the production site database directory to the test site database directory when no users are logged in.
 - e) For MS SQLServer: Use the DBMS tools to perform a full backup of the production database and restore that backup to a new database. Do not use the Export and Import wizards to copy the database. These wizards do not transfer the database indices and constraints.
 - f) For Oracle: Use the DBMS tools to perform a full backup of the production database and restore that backup to a new database.
 - g) Link the copy to the test site by creating a DSN to the new database and modifying the settings.asp file (line 2) to use that DSN.
 - h) Verify that the test site functions exactly as the production site before proceeding.
2. Execute the conversion task plan against the test site.
 You will probably want to keep track of the elapsed time for the complete conversion. This information will help you in scheduling downtime for the production site.
 3. Modify the task plan as necessary. Then repeat steps 1 and 2 until the conversion proceeds with no problems.
 4. Verify the behavior of the test site.
 5. Identify any new graphics that need to be customized. There are several new buttons in version 1.51.
 6. Identify any customizations to the standard files (crsmenu.asp, menu.asp, clsmenu.asp, nav.asp, set1.asp, and default.asp) that may need to be re-done in the new version.

Sample Upgrade Task Plan (1.50-1.51)

Preliminary tasks

Note: This is a generic task plan. You should modify it to match your specific circumstances.

1. Determine the current database schema version.
2. The upgrade script supplied with WBT Manager 1.51 is designed to upgrade the database from version 1.06 to 1.07. You can view the current database schema version by selecting "About WBT Manager..." from the system administrator help menu. Look for "DB Version". If it is not 1.06, contact your vendor for customer support.
 - Version _____
3. Schedule site downtime.

Use the elapsed times from your trial run conversions to schedule site downtime. Note that the MDAC 2.5 upgrade may require that the entire server be re-booted.

- Site will be down starting _____ until approximately _____
5. Identify new directory.
 - Install new version to _____.
 6. Verify disk space
 - Database server:
Size of full database backup file: _____
Available disk space
(need at least 2 times backup file size) _____
 - Web server:
Size of WBT Manager virtual directory _____
Available disk space in target directory
(need enough to duplicate original +20MB) _____

Conversion tasks

1. Re-verify disk space
 - Database server ok? _____
 - Web server ok? _____
2. Disable production site.
 - <specify steps here>
3. Upgrade MDAC on web-server if needed.
4. Back up database.
 - <specify steps here>
5. Back up WBT Manager virtual directory.
 - <specify steps here>
6. Back up version 1.50 DLLs
 - Copy the following files from the WINNT\SYSTEM32 directory on the web server to a backup location.
 - wbtman1a.dll
wbtman1b.dll
wbtman1c.dll
wbtman1d.dll
wbtman1f.dll
wbtman1g.dll (if present)
wkuptbl.dll
7. Run database upgrade scripts.
 - a) Execute **runssql.exe** from the **utilities\database** directory on the WBT Manager CD.

- b) Click the [...] button next to **Data Source Name** and select the DSN which connects to the production database.
- If you are prompted for a Login ID and Password, use the database account that was originally used to set up the WBT Manager database.
- c) Make sure that the **Pause on SQL Error** and **Remove trailing semi-colons before executing** check boxes are checked and that **Pause between statements** is un-checked.
- d) Execute each of the files in the appropriate table below, in the order shown. To execute a script file:
1. Click the [...] button next to **SQL Script File**.
 2. Browse down to the **utilities\database\scripts** directory on the WBT Manager CD.
 3. Select the file and click **Open**.
 4. Click the **Execute** button.

Scripts for MS Access

Script filename	Notes
access_106_to_107.sql	
initdb.sql	Not necessary if the LMSAPI update was applied to the existing 1.50 installation. If needed, uncheck Pause on SQL Error before executing.

Scripts for MS SQLServer

Script filename	Notes
sqlserv_106_to_107.sql	
grantdb.sql	modify the Login ID wbtmanuser in this file to match the Login ID used by the web-server application on line 2 of settings.asp
initdb.sql	Not necessary if the LMSAPI update was applied to the existing 1.50 installation. If needed, uncheck Pause on SQL Error before executing.

Scripts for Oracle

Script filename	Notes
oracle_106_to_107.sql	
grantdb.sql	modify the Login ID wbtmanuser in this file to match the Login ID used by the web-server application on line 2 of settings.asp
orasyn.sql	
initdb.sql	Not necessary if the LMSAPI update was applied to the existing 1.50 installation. If needed, uncheck Pause on SQL Error before executing.

8. Install Version 1.51 of the Web-Server Application

Install the new version of WBT Manager on top of the copy of the production version. You can launch the file installation program from **\web_server_application\setup.exe**.

This installation will require that the web-server be re-booted unless you stop the **IISADMIN** service before running setup.

9. If you had to enter a date conversion function's definition in **settings.asp** to support your DBMS, edit **settings.asp** to make the function use the date format "YYYY-MM-DD HH:MM:SS" instead of "MM/DD/YYYY HH:MM:SS". See **<root>settings.asp** in "Cutomizing the WBT Manager Web Interface" for more details.

10. Install the new version of the External Script Interface Kit (if desired).

See "Installing the External Script Interface kit" on page 4 for details.

11. Install the new version of the batch maintenance module (if desired).

See "Installing the Batch Maintenance Module" on page 5 for details.

12. Re-make any custom changes to menus.

If you have made any changes to **menu.asp**, **crsmenu.asp**, or **clsmenu.asp**, you will need to make the equivalent changes in the new versions of those files.

13. Add missing custom graphics.

The following graphics are new with WBT Manager 1.51. You may want to create custom versions of them to match your site.

Filename(s)	description
edit.gif	"Edit" button. Replaces "Drop" button on student course enrollment form. "Drop" button is now on student course assignment form.
boffline.gif offline.gif	"Go Off-Line" button and offline status indicator. Used with optional Offline Student Module.

14. Install version 1.51 System Administrator module.

See "Installing the System Administrator Module" on page 5 for more details. Install on the existing WBT Manager 1.50 directory

15. Install the version 1.51 Student Import Toolkit (if desired).

See "Installing the Student Import Toolkit" on page 5 for more details. You will have to uninstall version 1.50 first.

16. Verify operation of site.

Verify to your satisfaction that the new version of the site is operating as desired. Do not continue with the upgrade until you are satisfied.

- <specify your own test steps and approval criteria here>

17. Perform a full backup of the current database.

- <specify your own backup steps here>

18. Release the site to users.

Aborting a failed conversion

- <specify your own steps here>
1. Restore the database from the full backup made in step 4 of **conversion tasks**.
 - <specify your own restore steps here>
 2. Restore the WBT Manager virtual directory from the backup made in step 5 of **conversion tasks**.
 - <specify your own restore steps here>
 3. Restore the version 1.50 DLLs
 - a) Stop the **IISADMIN** service.
 - b) Unregister the version 1.51 DLLs.
 1. Launch a console prompt and change directories to `\WINNT\SYSTEM32`.
 2. Use **regsvr32.exe** to unregister the WBT Manager DLLs by typing the following at the command prompt:

```
regsvr32 -u wbtman1a.dll
regsvr32 -u wbtman1b.dll
regsvr32 -u wbtman1c.dll
regsvr32 -u wbtman1d.dll
regsvr32 -u wbtman1f.dll
regsvr32 -u wbtman1g.dll
regsvr32 -u wlkuptbl.dll
```
 - c) Delete the WBT Manager DLLs.
 - d) Copy the version 1.50 DLLs from the backup directory where they were placed in step 6 of **conversion tasks**.
 - e) Re-register the WBT Manager 1.50 DLLs.
 1. Launch a console prompt and change directories to `\WINNT\SYSTEM32`.
 2. Use **regsvr32.exe** to register the WBT Manager DLLs by typing the following at the command prompt:

```
regsvr32 wbtman1a.dll
regsvr32 wbtman1b.dll
regsvr32 wbtman1c.dll
regsvr32 wbtman1d.dll
regsvr32 wbtman1f.dll
regsvr32 wbtman1g.dll (if present)
regsvr32 wlkuptbl.dll
```
 - f) Restart the web services.
 4. Verify operation of the restored site.
 5. Release restored site to users.
 - <specify your own steps here>

***Preparing a database
for use with WBT
Manager***

Using a different DBMS

WBT Manager has been successfully tested with and is fully supported under Microsoft SQLServer 7 and Oracle 8i. Detailed instructions for installation using these databases can be found in \documentation\sqlserversetup.html and \documentation\oraclesetup.html on the WBT Manager CD.

Contact Integrity eLearning customer support if you have questions about using other database engines.

In general, the steps to prepare a database for use with WBT Manager are:

1. Create the database
2. Create a user account for WBT Manager
3. Create an ODBC Data Source for the database
4. Create and initialize WBT Manager tables.
5. Assign access to all tables to the user

The Microsoft Access database that is shipped with WBT Manager is already configured as installed and requires no modification.

About Shared Status

The ability to re-use the same lesson in multiple courses has a major drawback. In the default WBT Manager installation, the student's status for a lesson is not propagated to all instances of the lesson in the assigned curriculum. This means that if two courses containing the same lesson are assigned to a given student, the student must complete the lesson twice in order to receive credit for both courses.

Version 1.5 of WBT Manager offers a Shared Status Conversion that will cause WBT Manager to use the same status entry for a lesson in all of the courses that contain it. After this conversion is run, students will not have to repeat lessons that occur in more than one of their courses.

Status sharing has some drawbacks, however. System performance may be degraded by the need to compute status for more than one course each time a lesson is completed. Also, custom reports may be more difficult to create and the results may be misleading. For example, accumulating hours of instruction by course may give false values when a lesson taken once is included in more than one course.

It is important to weigh the advantages and disadvantages of shared status carefully before you make the decision to convert. Once your database is converted, it cannot be converted back to "unshared" status tracking.

Customizing the WBT Manager™ Web Interface

Overview

One of the design goals for WBT Manager™ was to allow clients to produce a training web site consistent with their own appearance guidelines. There are various levels of customization possible in the WBT Manager Web-Server Application. These range from simple graphics swaps to complete menu reprogramming. This section will first look at the files normally available for customization. Then it will discuss the course menu development API.

Note: In the code samples in this chapter, alternate lines of code are shaded. When editing the actual code, do not place any carriage returns within a code line. Ignore any word wrapping that has been necessary to fit the lines on this page.

Proprietary Requirements

We at Integrity eLearning want you to be able to customize the web module as you see fit. However, for brand name and copyright reasons, we have the following requirements:

- If you customize the home page and frameset to the extent that you remove the graphics containing the WBT Manager name, you must include a "Powered By WBT Manager" logo in a prominent location on the home page. See "WBT Manager Logo requirements" on page 2-24 for more information.
- You may not remove or conceal the Integrity eLearning copyright notice on the logout page. You may add your own notice by including it in **logout.inc** as described below.

Customizable files

Many files are currently available for user customization, as explained in the following sections. The installation program for future releases will not overwrite these files. However, you should make a backup of all customized files in case it is necessary to reinstall WBT Manager from scratch. Any other user modifications to the WBT Manager Web Server Module files will be

overwritten by future installations. Files marked with an asterisk (*) are newly designated as customizable.

Some changes to customizable files will not take effect unless the web module's application settings are reloaded. The simplest way to do this is to re-start the web server. However, doing so can become very awkward when performing repeated changes to refine the look of the site. The script **reload.asp** can reload the application settings without re-starting the server. It is located in **\utilities\trouble_shooting\server_check**. Copy it to the root of the WBT Manager directory. You can then call the script in your browser to reload the application settings.

The individual customizable files are discussed below. In the file names, **<root>** indicates the installation location.

<root>/settings.asp

This file contains the Data Source Name, User ID and Password to be used to connect to the WBT Manager database. It also controls the frame-set sizes, page background colors or graphics, and various other application settings. You can edit this file with a text editor if these values need to be changed.

Note: The web server will typically need to be restarted for any changes to take effect.

The following is the standard text of **settings.asp**.

Note: Alternate lines of code are shaded. When editing the actual code, do not place any carriage returns within a code line. Ignore any word wrapping that has been necessary to fit the lines on this page.

```

-----start of settings.asp-----
<%
'dsn=WBT Manager Database;uid=wbtmanuser;pwd=wbtmanuser
'
'
'#FFFFFF,#FFFFFF,#FFFFFF,#FFFFFF,nav_bg.jpg,#FFFFFF
'25,160,168,123,,,,
'
'
'
'line 2 = ODBC connection string
'line 3 = list of student id's for which AICC interface calls
         are to be logged. separate ids with leading and
         trailing commas ( ",jsmith,htruman," etc).
'line 4 = list of lesson id's for which AICC interface calls
         are to be logged. separate ids with leading and
         trailing commas ( ",1001,1005," etc).
'NOTE: log files will need to be purged manually and could get
       very large. use only for debugging.
'line 5 = list of background images or colors for pages
         (leading # indicates RGB color value). default
         page,class menu,course menu,reports,left navbar,
         course catalog
'line 6 = misc application setting separated by ", "s
'   position 1 = Navigation button height for button placement
'   position 2 = Navigation frame width for MSIE
'   position 3 = Navigation Frame width for other browsers
'   position 4 = Top Banner height
'   position 5 = Flag to turn off student account maintenance
                 function. Set to 1 to disable maintenance.
                 NOTE:class certifications will not expire
                 if this is set.
'   position 6 = Student menu ASP page for custom student menu.
                 Page must reside in ASP directory
'   position 7 = Certification Class menu ASP page for custom
                 menu. Page must reside in ASP directory
'   position 8 = Flag to turn off URLEncoding of AICC_URL and
                 AICC_SID parameters for lesson launch (1=off).
'line 7 = function used in sql statements for converting a
         date string in yyyy-mm-dd hh:mm:ss format to the
         DBMS date format
'         provided by default for ACCESS, SQL Server, or Oracle.
         Use %%Date%% for the date string. For example the
         oracle function is:
'         TO_DATE('%%DATE%%','YYYY-MM-DD HH24:MI:SS'),
         SQL Server is CONVERT(datetime,'%%DATE%%'),
         ACCESS is #%%DATE%%#
'line 8 = redirection delay times (in seconds)
'   position 1 = delay on austart page (lesson launch) after
                 session is created. Defaults to 2
%>
<html><head><title>Error 403.2</title>

<meta name="robots" content="noindex">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-
8859-1"></head>

<body>

<h2>HTTP Error 403</h2>

<p><strong>403.2 Forbidden: Read Access Forbidden</strong></p>

<p>This error can be caused if there is no default page available
and directory browsing has not been enabled for the directory, or
if you are trying to display an HTML page that resides in a
directory marked for Execute or Script permissions only.</p>

<p>Please contact the Web server's administrator if the problem
persists.</p>

```

```
</body></html>
-----end of settings.asp-----
```

Note: When making modifications to the file, **do not** remove the leading single quote ('). Otherwise WBT Manager will be unable to use the information.

- **Line 2.** This line contains the information that WBT Manager needs to connect to the database. If you change the User ID or Password in the database, you must also change it here before WBT Manager can connect again.
- **Line 3.** This line is used for testing AICC compliant lessons. For testing purposes, add a student ID to a comma-separated list (with leading and trailing commas) on this line. When the test student takes an AICC-compliant lesson, WBT Manager will log the data passed by the lesson to the AICC Interface. The log file **aicclog.txt** will be created in the database subdirectory of the WBT Manager installation. You should remove the student's ID from this line when testing is complete. Log files must be maintained manually.
- **Line 4.** This line performs the same function as line 3 except that it uses Lesson ID instead of Student ID. Lesson ID numbers can be found by viewing the source to a course menu containing the lesson. The lesson ID is passed as parameter **lsnid** on the link to the lesson.
- **Line 5.** This line contains a list of the images or colors to use as backgrounds to the WBT Manager pages. It is a comma-separated list containing either image files residing in the images directory of the WBT Manager installation or color codes preceded by a "#". Colors and images may be mixed in a single installation, but not on a single page.
 - **Entry 1** – Default page background used on all pages except those listed below.
 - **Entry 2** – Certification Class Menu page background.
 - **Entry 3** – Course Menu page background used on course menus and lesson launch pages.
 - **Entry 4** – Report page background used on standard web reports.
 - **Entry 5** – Navigation bar background.
 - **Entry 6** – Course Catalog page background.
- **Line 6.** This line contains comma-separated list of miscellaneous WBT Manager settings.
 - **Entry 1** – Navigation button height to be used for determining navigation button positioning. If you replace the navigation button graphics you may need to modify this value to adjust their position on the screen.
 - **Entry 2** – Navigation frame width for Microsoft Internet Explorer browsers. If you replace the navigation button graphics you may want to adjust the width of the navigation frame.

- **Entry 3** – Navigation frame width for other browsers. Some browsers, notably Netscape, cannot position graphics at the edge of the browser window. Allow extra frame width here.
- **Entry 4** – Frame height for the top banner.
- **Entry 5** – Flag to disable student account maintenance. If you do not use Certification Classes in your installation, you may want to disable the certificate expiration script that runs when a student logs on. Set this value to "1" to disable it.
- **Entry 6** – Custom student menu script. If you customize the student menu as described later, you will put the name of the new menu script here. This script must reside in the **asp** directory.
- **Entry 7** – Custom certification class menu script. If you customize the certification class menu as described later you will put the name of the new menu script here. This script must reside in the **asp** directory.
- **Entry 8** – Flag to turn off URL Encoding of AICC_URL and AICC_SID parameters for lesson launch. A "1" here will send the two AICC parameters in unencoded form.

Note: The flag to turn off URL encoding should be used *only* if your content authoring tool cannot process the URL-encoded strings. Even in that case, its use is not recommended, because it may break other courseware that does comply with the AICC standard. It would be better to contact the authoring tool vendor and get them to implement the interface correctly.

- **Line 7.** This line is used to specify a DBMS-specific function used to convert a date from string format to the DBMS internal format. The function described here will be embedded in any SQL Statements that require a date as a parameter. This line is not necessary for MS Access, MS SQL Server, or Oracle.
- **Line 8.** This line controls the time delay before redirection on pages that automatically redirect to another page.
 - **Entry 1** – Delay in the austart page after the lesson session is created until the lesson URL is launched. Defaults to 2 seconds.

**<root>/default.asp*,
<root>/asp/set1.asp***

These files contain the WBT Manager top/bottom and left/right framesets respectively. You can easily change the orientation and size of each frame or add additional frames. You should not, however alter the names or parent-child relationships of the standard frames. The navigation menu update function depends on the parent of **right** having a child called **left** which contains the navigation bar.

**<root>/html/
banner.htm**

This file contains the top banner for WBT Manager. To adjust the frame height to match your desired banner, either:

- Use entry 4 on line 6 of **settings.asp** to specify the frame height.
- Customize the framesets directly in **default.asp** and **set1.asp**.

<root>/html/home.htm

This file contains the WBT Manager home page.

<root>/html/ie.css*,
<root>/html/ns.css*

These pages contain the web page style-sheets for Microsoft Internet Explorer and Netscape Communicator. Styles can be edited with a text editor or html style-sheet editor.

Note: In prior versions of WBT Manager, this information was held in <root>pgstyle.asp.

<root>/asp/logoff.inc*

The contents of this file will be embedded on the logoff page between the "logged out" message and the WBT Manager copyright notice. You can place any valid HTML code in this file.

Note: If you use code to automatically redirect the user to a different page, you must include a delay of at least 4 seconds so that the copyright notice is displayed.

<root>/asp/amenu1.inc,
<root>/asp/amenu2.inc

These files will be included at the top and bottom (amenu1 at the top, amenu2 at the bottom) of the Local Administrator menu. Include any HTML code you desire to have embedded in the administrator menu. The behavior is similar to that of the HTML1 and HTML2 fields in the System Administrator module for Organizations, Departments, Courses, and Certification Classes.

Button and Background Image Files

The following graphics, located in the **images** directory, can be replaced with custom graphics. New graphics for version 1.51 are listed in the appropriate tables, where they are marked with an asterisk. They are also called out in a separate table at the end of this section.

Navigation bar buttons

(In the filenames, _p indicates the normal button state, _g indicates the mouseover state, _d indicates the disabled state)

Filename	Button Name/Description
admlog_p.gif admlog_g.gif	Administrator Logon
stulog_p.gif stulog_g.gif	Student Logon
cpw_p.gif cpw_g.gif cpw_d.gif	Change Password
cat_p.gif cat_g.gif cat_d.gif	Course Catalog
help_p.gif help_g.gif	Help
loff_p.gif loff_g.gif	Log Off
rep_p.gif rep_g.gif rep_d.gif	Reports
rmnu_p.gif	Menu

rmnu_g.gif rmnu_d.gif	
whome_p.gif whome_g.gif	WBT Manager Home

Other Buttons

Filename	Button Name/Description
close.gif	Close popup window
descr.gif tdescr.gif (tabbed-style menu only)	Show description
enrollnow.gif	Enroll Now button on catalog for self-enrollable courses
logon.gif	Logon button on logon screen
start.gif	Start button on reports menu
save.gif	Save button on data entry screens
update.gif	Update button on data entry screens
clear.gif	Clear button on data entry screens
reset.gif	Reset button on data entry screens
delete.gif	Delete button on data entry screens
new.gif	New button on selection screens
go.gif	Go button on selection screens
enroll.gif	Enroll button on course and certification class assignment screens
drop.gif	Drop button on course and certification class assignment screens
postchg.gif	Post Changes button on manual result entry screens
resetstat.gif	Reset Stats button on course menus
crsstatus.gif	Recalc Status button on course menu
edit.gif*	“Edit” button on course enrollment form. Replaces “drop” button with has been moved to the course assignment information form.
boffline.gif*	“Go Off-Line” button for the course menu. Valid for systems on which the optional “Offline Student Module” is installed.

Status Graphics

(Filenames beginning with "t" denote graphics for tabbed-style menu only)

Filename	Description
notstart.gif tnotstart.gif	Lesson not yet started
incomp.gif	Lesson incomplete

tincomp.gif	
comp.gif tcomp.gif	Lesson complete
offline.gif*	Indicates that a course is currently in “offline” mode.

Standard Page Backgrounds

Filename	Description
pgbkg.gif	Default page background
crsbkg.gif	Course Menu background
clsbkg.gif	Certification Class menu background
repbkg.gif	Report background
navbkg.gif	Navigation Bar background
catbkg.gif	Course Catalog background

Course Menu Graphics

Filename	Description
indent.gif	Used to create indents in course menu displays. Default is 10x10 transparent gif.
minus.gif plus.gif	Expand/collapse symbols from tree-style course menu

New Graphics for Version 1.51

Filename	Description
edit.gif	“Edit” button on course enrollment form. Replaces “drop” button with has been moved to the course assignment information form.
boffline.gif	“Go Off-Line” button for the course menu. Valid for systems on which the optional “Offline Student Module” is installed.
offline.gif	Indicates that a course is currently in “offline” mode.

A Simple Customization Case Study

A simple customization consists of changing the button graphics, home page, banner, and text styles for the WBT Manager web-server application.

For a detailed review of the changes in appearance made between WBT Manager 1.31 and WBT Manager 1.31a see

\documentation\white_papers\customization\simple.html on the WBT Manager CD. All of the customizations in this example were implemented using customary web design tools, with no programming whatsoever.

As we document other customization examples they will be made available on our web site at www.ielearning.com/wbt/support.

Menu customization

Some clients may want to customize the behavior of the Student, Certification Class, or Course menus. We have provided a set of menu construction methods in a utility object that can be created within an ASP page. We have also opened up the source code for the standard menus so that they can be used as an example.

Standard Menu Files

The following files can be used as a starting point:

- **menu.asp** - contains the code to generate a student's menu.
- **clsmenu.asp** - contains the code to generate a certification class menu.
- **crsmenu.asp, crsmenu1.asp, crsmenu2.asp** - contains code to generate the default course menu.
- **crstreemenu.asp, crstreemenu1.asp, crstreemenu2.asp** - contain code to generate a menu that uses DHTML to create an expanding and collapsing tree menu.
- **crstabmenu.asp, crstabmenu1.asp, crstabmenu2.asp** - contain code to generate a menu that uses DHTML to simulate a tabbed dialog box containing lessons.

To Create a Custom Menu

The recommended approach for creating custom menus is:

1. Copy the appropriate file or files to different filenames in the asp directory.
2. Modify the file(s) as desired. *This requires programming expertise.*
3. Point the associated WBT Manager functions at the new file(s), as explained in the following sections.

Incorporating custom student and certification class menus

To connect WBT Manager to custom student and class menus, put the names of the new script files into the **settings.asp** file. Then reload the web-server application settings. This process is described in "Customizable files" on page 2-1. Note that there can be only one custom student menu and one custom class menu in place at one time.

Incorporating custom course menus

Setting up WBT Manager to use a new course menu is a little more complicated. You may have a nearly unlimited number of custom course menus available at one time.

Custom course menus are installed in the **mnustyle** table in the WBT Manager database. This table contains the list of **Menu Styles** that can be selected from on the **Startup** tab in a course's property sheet in the System Administrator module. Selecting a menu style associates the course with a particular menu generation script.

You can add or modify entries in the **mnustyle** table using the **Site Configuration Utility**. The utility is found in the **utilities\site_configuration** directory on the WBT Manager CD. See the documentation file **site_configuration_utility.html** in that directory for instructions.

The **mnustyle** table contains eleven columns, of which only three are fixed in definition in regard to custom course menus. The fields of interest are described below.

mnustyle fields used for custom course menus

Field name	Acceptable values	Description
StyleID	integer	The index value of the menu used in the course table to reference the style. Note: Values from 0-999 are reserved for use by Integrity eLearning. You may use any other values. The Site Configuration Utility will assign this value automatically
StyleName	character string, maximum of 255 characters	The name to display on the Startup tab of the course property sheet
GeneratorURL	character string, maximum of 255 characters	The URL to the script which is to generate the course's menu. Normally this will be a script in the asp directory. Any parameters must be URL Encoded.

To add the menu style, launch the site configuration utility and log in as a system-level administrator. Click the **Menu Style** tab and then the **Add** button. Enter values for **StyleName** and **GeneratorURL**. You may also enter values for the other menustyle fields if your menu will be using them.

The information you insert into the **mnustyle** table can be retrieved by the **GetMenuStyle** API function described later.

How the menus work

This section contains a simplified explanation of how the menu code operates. Actual programming details are beyond the scope of this manual. You will need to examine the code for the standard menus and proceed from there.

In general, the menu code:

1. Creates an instance of the WBT Manager utility object.
2. Verifies that the student is logged on.
3. Extracts any desired parameters from the request object.
4. Connects to the database.
5. Retrieves any custom HTML code from the database. The System Administrator will have entered any desired code as HTML1 and HTML2 on the Organization, Department, Course, and/or Certification Class property sheets.
6. Requests a list of the items for the menu. The utility object provides member functions to retrieve lists of assigned classes and courses as well as lists of lessons in a course and the student's current status in each lesson.
7. Uses the data in the list of items to create links on the page as desired.
8. Closes the database connection, releases objects and closes the page.

The WBT Manager utility object

To facilitate the creation and customization of menus, the WBT Manager web-server application includes a utility object that performs common operations associated with generating WBT Manager menus. Using this object will insulate you from the details of the WBT Manager database.

The object is instantiated using the **CreateObject** method of the ASP Server object. The following is the VBScript code to create an instance of the object and assign it to a variable named **objutils**

```
Set objUtils = Server.CreateObject("wbtman2d.webutl2")
```

wbtman2d.webutl2 is the **program id (progid)** of the utility object used by IIS to load the appropriate DLL and create the object. This progid will typically change between major releases of WBT Manager to allow for parallel operation of different versions. Because of this, it is a good idea to create the object only once per menu page and to do so in a well defined location (such as at the top of the page). This will make it easier to locate and change for new releases.

The utility object provides the following methods:

- **CheckLoginStatus(ByRef objStringTable As Object, ByVal strType)**

CheckLoginStatus is called to verify that the user is logged onto WBT Manager. If the user is not logged on, this function will output a page informing the user of the situation and including a link to the WBT Manager login page.

- Return value:
 - **True** if the user is logged onto WBT Manager
 - **False** if the user is not logged on.
- Parameters:
 - **objStringTable** - A reference to the global string table containing all of WBT Manager's displayed text. This object can be referenced from any ASP page through the global object name **StringItems**
 - **strType** - The user type. "S" means student and "A" means administrator.
- Usage:

```
If Not objUtils.CheckLoginStatus(StringItems,"S") Then  
    Response.End'stop right here.  
End If
```

- **StartNewPage(ByVal strLocIDIn, ByVal strTargetIn, ByVal bForceFrame)**

StartNewPage generates the beginning of a standard WBT Manager web page. It outputs the HTML code to link to the style sheets and the JavaScript functions to update the navigation bar (if necessary). Outputs the <HEAD> tag but not the </HEAD> tag.

- Return value: NONE
- Parameters:
 - **strLocIDIn** - A string value which identifies the current page to the navigation bar. Typically contains the name of the ASP script. The navigation bar script uses this value to decide which buttons (if any) to display and which to make active.

- **strTargetIn** - A string value to be used in the BASE TARGET tag. If an empty string is passed, the value "right" is used.
- **bForceFrame** - A Boolean value which controls whether the WBT Manager frameset is forced for the page. If set to "true," the page will automatically reload the frameset if launched stand-alone.

- Usage:

```
objUtils.StartNewPage "crsmenu", "", False
objUtils.StartNewPage "menu", "", True
```

- **OpenBrowserWindow()** and **OpenBrowserWindow2()**

OpenBrowserWindow and **OpenBrowserWindow2** output JavaScript code used to open popup windows. The code generated by **OpenBrowserWindow2()** also tracks the child windows to keep them on top. The actual Javascript code included by **OpenBrowserWindow** is in **openc.js** in the **jar** directory. The code for **OpenBrowserWindow2** is in **openc3.js** and **vbchkwin.vbs**. The function contained in these files is called **opencentered**.

- Return value: NONE
- Parameters: NONE
- Usage:

```
objUtils.OpenBrowserWindow
objUtils.OpenBrowserWindow2
```

- Usage of JavaScript opencentered function:

```
opencentered(url,nm,nx,ny,nsc)
```

url – the URL to the page to be opened in a new window.

nm – the internal window name for the new window (no more than 6 characters).

nx – the desired width of the new window.

ny – the desired height of the new window.

nsc – scroll bar control. If "0" then new window will have scrollbars.

- Example:

```
<A CLASS="clsCrsmenuTxt "
  HREF="javascript:opencentered('austart.asp?crsid=1044&ls
  nid=9', 'lesson', 640, 480,0)" target="_self">Adding
  Departments on the Web</A><BR>
```

- **OpenDBConnection()**

OpenDBConnection creates an ADO connection object and connects to the WBT Manager database. The connection parameters

are retrieved from Application variables set by the initialization functions in **global.asa**.

- Return value:
An ADO connection object connected to the WBT Manager database. You should close this object and set its variable to nothing when you are done making database access calls.
- Parameters: NONE
- Usage:

```
Set dbConn=objUtils.OpenDBConnection()
```

- **GetMenuStyle(ByRef dbConn, ByVal nStyle, ByRef strErrMsg)**

GetMenuStyle retrieves data from the **mnustyle** table for the specified menu style. The menu style id for a course can be found in the **MenuStyle** property of the course data objects returned by the **GetStudentCourses** and **GetClassCourses** functions described below.

- Return value:
An object with the properties listed below. If a menu style does not exist in the table for the specified value of **nStyle**, the object will still be returned but the **StyleID** property will be set to 0. If an error occurs, the return value will be "Nothing".

In the following list only StyleID, MenuName, and GeneratorURL have a hard and fast definition. The other fields may be used as desired by custom menus.

- **StyleID** – The internal identifier for the menu style. 32bit integer value. Values 0-999 are reserved for use by Integrity eLearning.
- **MenuName** – The name of the menu style as displayed in the system administrator module on the course startup property page. 1-255 characters in length.
- **GeneratorURL** – The URL specifying the script to be called to create the course menu for this style. Typically specifies an ASP script in the ASP directory.
- **Background** – String value 0 to 255 characters long. Standard usage is to contain a URL to an image file for use as the menu background.
- **DescriptionGraphic** – String value 0 to 255 characters long. Standard usage is to contain a URL to an image file for use as "description" icon.
- **NotStartedGraphic** – String value 0 to 255 characters long. Standard usage is to contain a URL to an image file for use as the "not started" indicator for the menu.
- **IncompleteGraphic** – String value 0 to 255 characters long. Standard usage is to contain a URL to an image file to use as the "incomplete" indicator for the menu.

- **CompleteGraphic** – String value 0 to 255 characters long. Standard usage is to contain a URL to an image file to use as the "completed" indicator for the menu.
- **StyleSheetURL** – String value 0 to 255 characters long. Standard usage is to contain a URL to the HTML style sheet for the menu.
- **Graphical** – String value 0 to 1 character long. Standard usage is as an indicator that the menu style supports the use of graphical links to course components.
- **HardCoded** – String value 0 to 1 character long. Standard usage is as an indicator that the menu style is for a "hard-coded" menu course where the menu directly calls the course's lessons without using the WBT Manager lesson launch mechanism.
- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nStyle** – The integer value which identifies the menu desired menu style.
 - **strErrMsg** - A text message returned if an unexpected error occurs.
- Usage:

```
Set = objStyle = GetMenuStyle (dbConn, 1000, strErrMsg)
```

- **GetCourseHTMLBlocks(ByRef dbConn, ByVal nCourseID, ByRef strTopHTML, ByRef strBotHTML) and GetClassHTMLBlocks(ByRef dbConn, ByVal nClassID, ByRef strTopHTML, ByRef strBotHTML)**
GetCourseHTMLBlocks and **GetClassHTMLBlocks** retrieve custom HTML code to be embedded in the course or certification class menus. The retrieved data comes from the **HTML1** and **HTML2** properties in the system administrator modules course or class properties pages.
 - Return value: NONE
 - Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nCourseID, nClassID**– The internal ID number of the course or class for which data is to be returned. This value will normally be passed as an HTML parameter to the menu script.
 - **strTopHTML** – String variable to receive the contents of the HTML1 property associated with the course or class.
 - **strBotHTML** – String variable to receive the contents of the HTML2 property associated with the course or class.

- Usage:

```
objUtils.GetCourseHTMLBlocks
dbConn,nCourseID,strTopHTML,strBotHTML
objUtils.GetClassHTMLBlocks
dbConn,nClassID,strTopHTML,strBotHTML
```

- **GetOrgHTMLBlocks(ByRef dbConn, ByRef strTopHTML, ByRef strBotHTML) and GetDeptHTMLBlocks(ByRef dbConn, ByRef strTopHTML, ByRef strBotHTML)**

GetOrgHTMLBlocks and **GetDeptHTMLBlocks** retrieve custom HTML code to be embedded in the student menu for specific organizations and departments. The retrieved data comes from the **HTML1** and **HTML2** properties in the system administrator module organization or department properties pages. The organization or department is specified by session variables set during the student login process.

- Return value: NONE
- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **strTopHTML** – String variable to receive the contents of the HTML1 property associated with the organization or department.
 - **strBotHTML** – String variable to receive the contents of the HTML2 property associated with the organization or department.
- Usage:

```
objUtils.GetOrgHTMLBlocks dbConn, strTopHTML, strBotHTML
objUtils.GetDeptHTMLBlocks dbConn, strTopHTML,
strBotHTML
```

- **GetStudentClasses(ByRef dbConn, ByVal nStudentID, ByRef nBounds)**

GetStudentClasses returns an array of the Certification Classes in which a student is enrolled. Each entry in the array contains an object whose properties provide enough information to create a link to the Certification Class menu for that class.

- Return value:

A zero-based array with an upper bound of nBounds. Each element in the array contains an object with the following properties:

 - **ID** – The internal WBT Manager ID for the certification class.
 - **ClassName** – The display name for the class.
 - **Status** – The student's current status in the class. "C" for completed status, "I" for incomplete status.
- Parameters:

- **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
- **nStudentID** – The internal WBT Manager ID for the student whose class assignments are desired. This value will typically be drawn from a Session variable (see "Session variables of interest" on page 2-22).
- **nBounds** – A variable to receive the upper bound of the array returned by **GetStudentClasses**. If there are no classes assigned to the student, this variable will contain "-1."
- Usage:

```

aClass = objUtils.GetStudentClasses(dbConn, nStudentID,
nClass)
If nClass >= 0 Then
  For i = 0 To nClass
    .
    .
    .
  Next
End If

```

- **GetStudentCourses(ByRef dbConn, ByVal nStudentID, ByRef nBounds)**

GetStudentCourses returns an array of the courses in which a student is enrolled. Each entry in the array contains an object whose properties provide enough information to create a link to the course menu for that course.

- Return value:

A zero-based array with an upper bound of **nBounds**. Each element in the array contains an object with the following properties:

- **ID** – The internal WBT Manager ID for the course.
- **CourseName** – The display name of the course.
- **Status** – The student's current status in the course. "C" = completed, "I" = incomplete, "N" = not started.
- **VName** – The vendor specific name for the course. Corresponds to the "Provider's course name" field in the course property pages within the WBT Manager administrator module.
- **MenuGen** – The script to be used to generate the course's menu.
- **MenuStyle** – The internal WBT Manager ID for the menu style associated with the course.
- **X** – The desired width of the browser window to contain the course.
- **Y** – The desired height of the browser window to contain the course.

- **FullScr** – Boolean value indicating whether the course should be displayed in full-screen mode. "True" if full-screen mode is desired, otherwise "False."
- **Shadow** - Boolean value indicating whether the course is a "shadow" course containing no online content. "True" if course is a shadow course, otherwise "False."
- **HardMenu** - Boolean value indicating whether the course has a hard-coded menu and will not use the WBT Manager menu generation mechanism. "True" if course has a menu, otherwise "False." If "True" then **MenuURL** should contain the path to the menu page.
- **GraphicMenu** - Boolean value indicating whether the course menu will display graphical links to the lessons in the course. "True" if menu is graphical, otherwise "False." Paths to the graphic files will be found in the **ImageURL** property of objects returned by the **BuildMenuList** function.
- **MenuURL** – The URL to a hard-coded menu for the course.
- **ImageURL** – The URL to an image to be used in place of a text link to launch the course from the student menu.
- **IsExpired** – Boolean value indicating whether the student's enrollment in the course has expired based on the "cut-off date" set when they were enrolled.
- **IsReady** – Boolean value indicating whether the student is allowed in the course yet based on the "start-after" date set when they were enrolled.
- **AllowReset** – Boolean value indicating whether the student should be allowed to "reset stats" for the course.
- **AllowOffline** – Boolean value indicating whether the student may take the course in offline mode.
- **IsOffline** – Boolean value indicating whether the course is currently being taken in "offline mode" by the student.
- **AllowSelfEnroll** – Boolean value indicating whether the student is allowed to "self-enroll" in the course through the course catalog.
- **AllowDrop** – Boolean value indicating whether the student is allowed to "self-drop" the course.
- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nStudentID** – The internal WBT Manager ID for the student whose course assignments are desired. This value will typically be drawn from a Session variable (see Session variables of interest).
 - **nBounds** – A variable to receive the upper bound of the array returned by **GetStudentCourses**. If there are no courses assigned to the student then this variable will contain "-1."

- Usage:

```

aCrss = objUtils.GetStudentCourses(dbConn, nStudentID,
nCrss)
If nCrss >= 0 Then
  For i = 0 To nCrss
    .
    .
    .
  Next
End If

```

- **GetClassCourses(ByRef dbConn, ByVal nClassID, ByVal nStudentID, ByRef nBounds)**

GetClassCourses returns an array of the courses in a certification class. Each entry in the array contains an object whose properties provide enough information to create a link to the course menu for that course.

- Return value:

A zero-based array with an upper bound of **nBounds**. Each element in the array contains an object identical to the objects returned by **GetStudentCourses**.

- Parameters:

- **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
- **nClassID** – The internal WBT Manager ID for the certification class whose courses are desired.
- **nStudentID** – The internal WBT Manager ID for the student whose course assignments are desired. This value will typically be drawn from a Session variable (see Session variables of interest).
- **nBounds** – A variable to receive the upper bound of the array returned by **GetStudentCourses**. If there are no courses in the class then this variable will contain "-1."

- Usage:

```

aCrss = objUtils.GetClassCourses( dbConn, nClassID,
nStudentID, nCrss )
If nCrss >= 0 Then
  For i = 0 To nCrss
    .
    .
    .
  Next
End If

```

- **GetCourseInfo(ByRef dbConn, ByVal nCourseID, ByVal nStudentID, ByVal nOptions)**

GetCourseInfo returns the same data about an individual course that **GetStudentCourses** and **GetClassCourses** return for a list of courses.

Return value:

An object identical to one of the elements in the array returned by **GetStudentCourses**. If the course is not found then the **ID** property of the object will be set to 0.

- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nCourseID** – The internal WBT Manager ID for the course whose data is desired.
 - **nStudentID** – The internal WBT Manager ID for the student whose course assignments are desired. This value will typically be drawn from a Session variable (see Session variables of interest).
 - **nOptions** – reserved for future use. Should always be set to 0.
- Usage:

```
Set objCrse = objUtils.GetCourseInfo( dbConn, nCourseID, nStudentID, 0 )
```

- **IsCourseExpired(ByRef dbConn, ByVal nCourseID, ByVal nStudentID, ByRef strStatus)**

IsCourseExpired tests to see if a student's enrollment in a course has expired based on the course cutoff date set when the student was enrolled in the course.

- Return value:
 - "True: if the student is no longer allowed to take the course because the current date is past the cutoff date for the course enrollment. Otherwise "False."
- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nCourseID** – The internal WBT Manager ID for the course that is being checked.
 - **nStudentID** – The internal WBT Manager ID for the student whose course expiration is being tested.
 - **strStatus** – A variable to receive the current status of the course (if any). "C" = completed, "I" = in progress, "" = not started or student not enrolled.
- Usage:

```
bExpired = objUtils.IsCourseExpired( dbConn, nCourseID, nStudentID, strStatus)
```

- **IsCourseReady(ByRef dbConn, ByVal nCourseID, ByVal nStudentID, ByRef dtStartAfter)**

IsCourseReady tests to see if a course is available for a student to enter based on the course "start-after" date set when the student was enrolled in the course.

- Return value:
"True" if the student is allowed in the course. Otherwise "False."
- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nCourseID** – The internal WBT Manager ID for the course that is being checked.
 - **nStudentID** – The internal WBT Manager ID for the student whose course availability is being tested.
 - **dtStartAfter** – A variable to receive the actual "start-after-date" if the course is available to the student.
- Usage:

```
bStartOK = objUtils.IsCourseReady(dbConn, nCourseID, nStudentID, dtStartAfter)
```

- **BuildMenuList(ByVal nCourseID, ByRef nBounds, ByVal nStudentID)**

BuildMenuList returns an array of the instructional blocks and lessons in a course, where each entry in the array is an object whose properties contain the information required to launch the lesson.

- Return value:
A zero-based array with an upper bound of **nBounds**. Each element in the array contains an object representing either a lesson or an instructional block. Each object has the following properties:
 - **ID** – The internal WBT Manager ID for the lesson or block.
 - **EntryType** – Flag indicating whether the object represents a lesson or a block. "l" = lesson, "b" = block.
 - **EntryName** – The display name for the lesson or block.
 - **VName** – The vendor specific name for the lesson (lessons only, not valid for blocks). Corresponds to the "Provider's block name" or "Provider's lesson name" property in the WBT Manager system administrator module.
 - **Indent** – The indentation level of the lesson or block. This integer field indicates the hierarchical structure of the course.
 - **Status** – The student's current status in the lesson or block. For blocks this may contain "C" (completed), "I" (incomplete),

or "N" (not started). For lessons it may contain "C" (completed), "P" (passed), "F" (failed), "I" (incomplete), or "N" (not started).

- **sDate** – The date that the status was last changed.
- **URL** – The URL which will launch the lesson (lessons only, not valid for blocks).
- **ImageURL** – The URL to a graphic to be used in place of a text link on the course menu to launch the lesson (lessons only, not valid for blocks).
- **X** – The course's desired browser window width (lessons only, not valid for blocks).
- **Y** – The course's desired browser window height (lessons only, not valid for blocks).
- **FullScr** – A flag indicating that the course is to run in full-screen mode (lessons only, not valid for blocks). "1" = fullscreen.
- **Itfc** – The data communication mechanism used by the lesson (lessons only, not valid for blocks). "0" = AICC WBT Interface, "1" = No data communication interface.
- Parameters:
 - **dbConn** – An ADO connection object connected to the WBT Manager database (as returned by **OpenDBConnection**).
 - **nCourseID** – The internal WBT Manager ID for the course.
 - **nStudentID** – The internal WBT Manager ID for the student.
- Usage:

```
aLsns=objUtils.BuildMenuList(nCourseID,nLsns,nStudentID)
If nLsns >= 0 Then
  For i = 0 To nLsns
    .
    .
    .
  Next
End If
```

- **EndPage()**

EndPage closes the HTML page by outputting the </BODY> and </HTML> tags, along with any standard WBT Manager closing text.

- Return value: NONE
- Parameters: NONE
- Usage:

```
objUtils.EndPage
```

- **UserLogon(ByRef objStringTable As Object, ByVal strUserID, ByVal strPassword, ByVal strType, ByRef strMsg)**

UserLogOn validates a user against the WBT Manager database and creates the user's session variables if successful. It can be used to create a login mechanism external to the standard WBT Manager login page (see **slogin.asp** for an example).

- Return Value:
"0" if successful, "1" if the specified user id and password are not valid, "99" if an unexpected error occurs
- Parameters:
 - **objStringTable** - A reference to the global string table containing all of WBT Manager's displayed text. This object can be referenced from any ASP page through the global object name **StringItems**
 - **strUserID** - The user's login id.
 - **strPassword** - The user's login password.
 - **strType** - The user type. "S" means student and "A" means administrator.
 - **strMsg** - A variable to contain an error message if an unexpected error occurs.
- Usage:

```
nResult = objUtils.UserLogon( StringItems, strLogID, strPW, strType, strTemp )
```

Note: In order to create a valid WBT Manager session, this function must be called from a script contained in the WBT Manager virtual directory on the web server. The ASP session cookie generated during this process must be returned to the browser.

Session variables of interest

Within the WBT Manager Web-server application, a user's logged in state is preserved in ASP Session variables. The following variables may be of interest when creating custom menus.

- **Session("strWBTBrowser")** - "IE" if the user's browser is identified as Microsoft Internet Explorer. Otherwise "Netscape"
- **Session("strWBTBrowserVer")** - The browser version as a string.
- **Session("nWBTBrowserMVer")** - The browser major version number.
- **Session("strWBTLogonID")** - The user's logon ID.
- **Session("strWBTLogonPW")** - The user's login password.
- **Session("strWBTLogonName")** - The user's display name.
- **Session("strWBTUserType")** - The user type. "S" = student, "A" = administrator.
- **Session("nWBTUserID")** - The WBT Manager internal ID for the user.

- **Session("nWBSTudCompID")** – The internal ID of the student’s organization.
- **Session("nWBSTudDeptID")** – the internal ID of the student’s department.

Direct lesson launch using astart.asp

We have often been asked if there is any way to bypass the WBT Manager course menu systems and directly launch a lesson. For instance, a client may want to build a non-web front-end program or create a static course menu to reduce the overhead on the web-server. We have added this capability with WBT Manager 1.51.

NOTE: Replacing the standard WBT Manager web interface does not exempt you from the “Powered by WBT Manager” logo requirements mentioned below. Both the WBT Manager logo and the WBT Manager copyright notice must be included in your user interface unless you have made contractual arrangements with Integrity eLearning to the contrary.

A lesson may be launched by a direct call to **astart.asp** under the following conditions.

- Student must be logged on
The student must have logged onto WBT Manager using either the standard WBT Manager login screen or through a specialized login script such as **slogin.asp**.
The browser or browser-substitute application must support the ASP session cookie.
- Student must be enrolled in the course

Using astart.asp

astart.asp is the WBT Manager launch script for AICC compliant lessons. In normal use it is called from a course menu in a popup window. The script creates a new AICC session for the lesson and then redirects to the lesson launch URL using a piece of Javascript code (found in **jar/astart.js**).

Here is an example of what the result of **astart.asp** might be.

```

<!doctype html public "-//IETF//DTD HTML//EN">
<HTML><HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=iso8859-1">
<link rel="stylesheet" href="/wbtman_dev/html/ie.css">
<BASE TARGET="_self">
<SCRIPT LANGUAGE="Javascript"
SRC="/wbtman_dev/jar/updmenu.js"></SCRIPT>
<script
language="javascript">reloadnav("http://it_fserv1/wbtman_dev/asp/na
v1.asp","austart");</script>
<TITLE>Session created</TITLE>
<SCRIPT LANGUAGE="Javascript">
var
lsnurl=" ../courses/ie_samples/flash/flash5/lmsapi/lmsapi.html?AICC_
SID=demo267&AICC_URL=http%3A%2F%2Fit%5Ffserv1%2Fwbtman%5Fdev%2Fasp%
2Faicc%5Fint%2Easp";
</script>
<SCRIPT LANGUAGE="Javascript"
SRC="/wbtman_dev/jar/austart.js"></SCRIPT>
</HEAD><BODY
onload=" javascript:austart(lsnurl,'demo267',2);"BGCOLOR="#FFFFFF">
A session has been created for this lesson.
Click <A
HREF=" ../courses/ie_samples/flash/flash5/lmsapi/lmsapi.html?AICC_SI
D=demo267&AICC_URL=http%3A%2F%2Fit%5Ffserv1%2Fwbtman%5Fdev%2Fasp%2F
aicc%5Fint%2Easp">here</A> if the lesson does not start
automatically.
<p class="clsSmInst"><B></B></p>
</BODY>
</HTML>

```

Note that the launch url is embedded in the page in 2 places.

```
lsnurl=" ../courses/ie_samples/flash/flash5/lmsapi/lmsapi.html?AICC_SID=d
emo267&AICC_URL=http%3A%2F%2Fit%5Ffserv1%2Fwbtman%5Fdev%2Fasp%2Faicc%5Fint%2Easp";
```

```
<A HREF=
" ../courses/ie_samples/flash/flash5/lmsapi/lmsapi.html?AICC_SID=demo26
7&AICC_URL=http%3A%2F%2Fit%5Ffserv1%2Fwbtman%5Fdev%2Fasp%2Faicc%5Fint%2Easp">
```

Direct lesson launch

In order to directly launch the lesson you will need to call **austart.asp** with parameters that identify the course, lesson, and CMI mode (the student is identified by a session variable). **austart.asp** accepts the following parameters for this purpose:

- crsvname** – The “provider’s course name” for the course. This value can be found on the Provider tab in the course property pages within the WBT Manager system administrator program.
- lsnvname** - The “provider’s lesson name” for the lesson. This value can be found on the Provider tab in the lesson property pages within the WBT Manager system administrator program.
- cmode** – The desired lesson mode for the launch. Allowed values are: “n” for “normal”, “r” for “review”, and “b” for “browse”. If this parameter is not specified then **austart.asp** may respond with a form requesting that the student select a mode.

When called with these parameters austart.asp will perform some validation checking and then create a session for the lesson and return the launch page. Errors may be returned under the following conditions:

Student is not logged in

Specified course does not exist

Specified lesson does not exist

Specified lesson is not in the specified course

Student is not enrolled in the course or enrollment is not active (expired or not reached the start-after date).

Here is an example URL that would launch the first lesson in the **Using WBT Manager** course that is provided with WBT Manager.

<http://www.ielearning.com/wbroot/asp/austart.asp?cmode=n&crsvname=integ-wbtm001&lsnvname=integ-wbtm001-sysintro>

WBT Manager Logo requirements

We designed the WBT Manager web-server application to allow our clients to customize the look as much as desired. However, in order to protect our trademark and copyright, you are required by our standard license agreement to comply with the following.

Unless you have a written contractual agreement with Integrity eLearning to the contrary:

- You may not remove or obscure the Integrity eLearning copyright notice on the logout page. If you customize the logout page in such a way that the user is automatically redirected to another page when logging out of WBT Manager, you must include the WBT Manager copyright notice on the target page or delay the redirection on logout so that the Integrity eLearning copyright notice is visible for at least 4 seconds.
- If you completely remove the standard WBT Manager logo and name from the home page, you must include one of the standard "Powered by WBT Manager" logo graphics on the redesigned home page. See below for usage details.
- If you use an alternative login mechanism that bypasses the WBT Manager home page completely, you must include one of the standard "Powered by WBT Manager" logo graphics on the web page containing the link used to jump into the WBT Manager web-server application. See below for usage details.

General Requirements

Logo Usage

Standard "powered by" logo graphics are available in the **utilities\powered_by_logos** directory on the WBT Manager CD or from customer support.

The following excerpt from Appendix B of the WBT Manager license contract provides guidelines for the use of these graphics.

APPENDIX "B" - Logo Usage Guidelines



General Guidelines

- The "Powered by WBT Manager" trademark can only be used by licensed parties.

- The trademark must never be altered and must be reproduced from the supplied digital file.
- The trademark may not be used in connection with the display, advertising or promotion of products that do not contain WBT Manager modules.

Location Guidelines

- The "Powered by WBT Manager" logo must appear on the splash screen, logon screen or within the software product.

Size Guidelines

- The logo must maintain its original size as supplied in the digital file.

Color Guidelines

- Either the color or black and white version of the "Powered by WBT Manager" logo is acceptable on screen.

Background

- The logo can be placed on any background as long as the logo is clearly visible.
- The logo may NOT be reversed to white. Use the black and white version supplied.