
WBT Manager™ 1.51 Changes

This document is a supplement for the WBT Manager™ version 1.50 user manual and system help files. It describes the new features and behavior of version 1.51 compared to version 1.50.

New features for version 1.51

Many of the changes for version 1.51 are “under the hood” so to speak and are not readily apparent to the user. We have added a number of new features though.

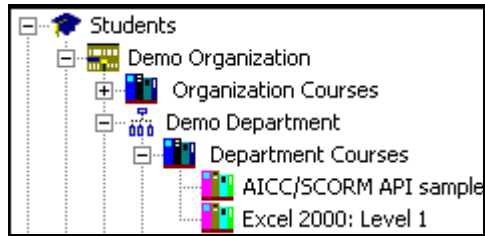
| | |
|--|----------|
| WBT Manager™ 1.51 Changes | 1 |
| New features for version 1.51 | 1 |
| Department course assignment | 1 |
| Auto enrollment of transferred students | 4 |
| Organizations can be created through web interface | 4 |
| Web module can edit course cut-off dates | 4 |
| Students can drop a course themselves | 5 |
| Lesson entry-count restrictions | 6 |
| HACP 3 support | 7 |
| LMS API support | 7 |
| Data collection changes | 10 |

Department course assignment

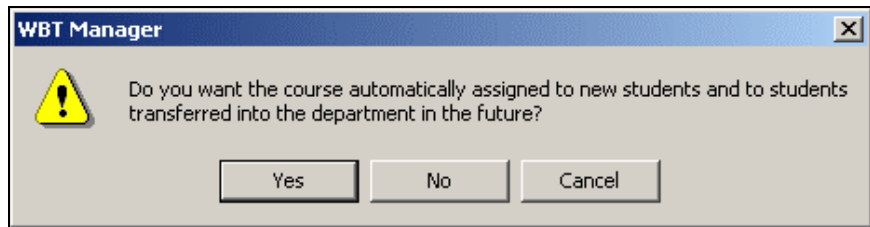
A feature was added in WBT Manager 1.50 that allowed a course to be assigned to an Organization so that only Local Administrators in that Organization could use the web assignment functions to enroll students in the course. The feature also allowed for automatically enrolling new students in the Organizations courses when they were added to the WBT Manager database.

In WBT Manager 1.51 that feature has been extended to the Department level. Courses can now be restricted for web enrollment to specific departments and automatically assigned to new students. The control mechanism is very similar to the existing mechanism for Organizations.

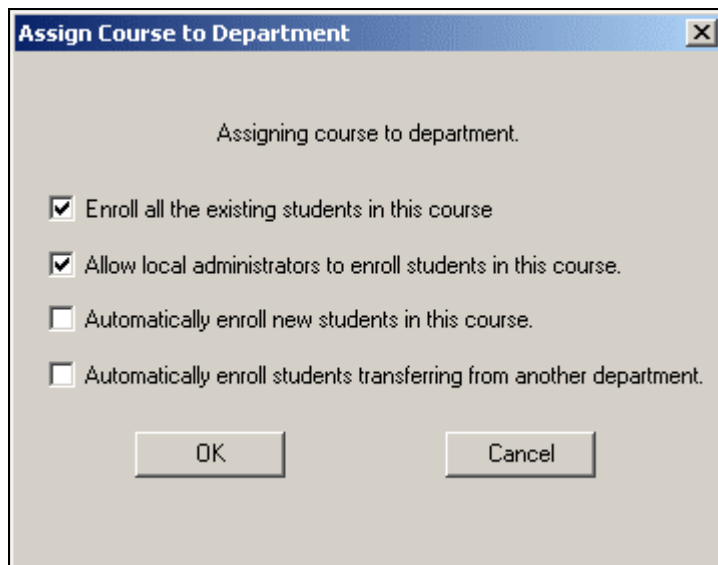
The Navigation Tree now contains a **Department Courses** node beneath each Department.



Dropping a course on the **Department Courses** node (or right-clicking and selecting **Add Course**) will result in the system asking if the user wants to enable auto-enrollment (this can be modified later).



Dropping a course directly on the **Department** node results in the system presenting the user with an options dialog. The selected auto-enrollment options may be changed later.



To change the auto-enrollment options for a department's course. Check or uncheck the boxes in the **Department Courses** properties (select one of the courses beneath **Department Courses** to see the properties).

Department Courses

Department Course

Name: Excel 2000: Level 1

Description: Training on Excel 2000: Level 1 provided through 15 self-study tutorials with software simulation.

Automatically enroll new students in this course.

Automatically enroll students transferred from another department.

This course can be assigned by local administrators of the department.

The course “Access” properties tab now shows both Organizations and Departments that are allowed to enroll students in the course. An asterisk (*) indicates a department entry. Select an entry and click the **Remove** button to remove the course from the Department (or right-click on the course beneath the **Department Courses** node and select **Remove Course**).

Course Properties

Info | Provider | Startup | Offline | HTML1

HTML2 | **Access** | Billing | AICC Modes

Allow all local administrators to enroll students in this course

Allow only local administrators from the organizations and departments listed below to enroll students in this course [departments are preceded by a "*"].

test organization

*Demo Department

Remove

Add organizations and departments by dropping this course on them in the navigation tree.

Auto enrollment of transferred students

As mentioned in **Department Course Assignment**, Organizations and Departments can be assigned courses in which new students are automatically enrolled. The option has also been added to automatically enroll students in Department or Organization courses when they are transferred from another Department.

This behavior is controlled from the **Department Courses** property page, as is the auto-enrollment of new students (see **Department Course Assignment**).

Organizations can be created through web interface

When a system-level administrator logs onto the web interface they will see a new option on the menu to allow them to **Add/Edit Organization**. This option allows Organizations to be added through the web interface. The function behaves almost identically to the **Add/Edit Department** function.

NOTE: A system administrator can create a new organization through the web interface at any time. In order to administer that organization the administrator must log off and then back on as the new Organization's administrator.

Web module can edit course cut-off dates

The **Drop** button on the **Assign/Drop Courses** page in the Local Administrator interface has been changed to an **Edit** button.

Assign/Drop Courses

Student: Student,Demo

Enroll Student in Course

Use one of the options below to select a course to enroll the student in.

Browse courses by category >>

Search Course Name

Enter a word or words to search course titles for.

Match: All words Any word

Edit Enrollment Information

To drop a student from a course or to edit their start/end dates for the course, select the course from the list below and press the "Edit" button.

"Tree" Menu

Clicking the **Edit** button brings up **the Student Enrollment Form**.

The screenshot shows a form titled "Student Enrollment Form" with the text "Editing: Student, Demo" below it. The form contains the following fields and controls:

- Course: AICC/SCORM API sample
- Start After Date: Mo: [dropdown menu] Dy: [text box] Yr: [text box]
- Cut Off Date: Mo: [dropdown menu] Dy: [text box] Yr: [text box]
- Billing Mode: Not Billed
- Billing Code: [text box]
- Billing Rate: 0.00

At the bottom of the form, there are three buttons: "update" (with an upward arrow icon), "reset" (with a square icon), and "drop" (with a downward arrow icon).

The **Start After** and **Cut Off After** dates for a student's course assignment can be modified from here (**Cut Off Date** will be unavailable for edit if the course definition has a **Disable-After** time period set on the course **Billing** properties tab).

The student can be dropped from the course by clicking the **Drop** button.

Students can drop a course themselves

If a course is set to allow "self-drop" then a **Drop** button will appear on the course menu when the student selects the course. Clicking the **Drop** button will cause the student's statistics to go to history and remove the student from the course. This behavior is controlled by a checkbox on the course "Info" property tab.

Course Properties

HTML2 Access Billing AICC Modes

Info Provider Startup Offline HTML1

Course Name: Excel 2000: Level 1

Description: Training on Excel 2000: Level 1 provided through 15 self-study tutorials with software simulation.

Set status "completed" on first entry
 This is an offline "shadow" course
 Allow students to "self-enroll" in this course
 Allow students to drop course
 Allow students to reset course statistics.

Checking the “Allow students to drop course” will enable the **Drop** button on the course menu (custom menus will have to be programmed to support this).

Lesson entry-count restrictions

Some of our clients expressed a desire to only allow a student one attempt at a scored lesson. This capability has been added in WBT Manager 1.51. It is controlled from the lesson “Startup” property tab.

Lesson Properties

Info Provider Startup AICC Data

Lesson Type: WBT - AICC Compliant

Full URL: ../courses/elementk/74200/742002a.html

Button URL:

Base URL: ../courses/elementk/74200/742002a.ht

Allow student to select "test only" mode (if available).
 Restrict number of scored attempts to:
 Restrict number of lesson entries to:

OK Cancel

Putting an integer value in the “Restrict number of scored attempts to:” box will cause the WBT Manager lesson launch mechanism to return an error to the student if they already have that many scores on record (based on the **NUMSC** column in the **STATS** table).

Putting an integer value in the “Restrict number of lesson entries to:” box will cause the WBT Manager lesson launch mechanism to return an error to the student if they already have entered the lesson that many times (based on the **NUMIN** column in the **STATS** table).

NOTE: Once a student has exceeded the allowed number of attempts for either scores or lesson entries then their stats for the course will have to be reset (by **Reset Stats** on the course menu or by dropping and re-enrolling them in the course). There is no way in the current version to “release” the number of tries allowed on an individual student basis (the number allowed could be increased for everyone though).

HACP 3 support

WBT Manager 1.51 now supports the changes from the AICC CMI001 specification version 3.02.

LMS API support

In order to make it easier for content providers to build browser-based content the AICC and the ADL (Advanced Distributed Learning Initiative) cooperated in defining a set of JavaScript function calls to perform communication between a lesson and LMS. These function calls form the basis for the SCORM Runtime Environment and are included in the AICC version 3 specification.

For a complete description of the functions and their behavior see the AICC CMI specification CMI001 available from www.aicc.org or the SCORM 1.0 specification available from www.adlnet.org.

WBT Manager 1.51 supports the LMS API for SCORM version 1.0 (AICC version 3.02) through the use of a Java Applet embedded in the WBT Manager course menu window. On lesson launch the applet's functions are made available to the lesson as a JavaScript object residing in the "opener" window of the lesson.

The following data elements are supported:

LMS to Lesson

- cmi.core.student_id
- cmi.core.student_name
- cmi.core.lesson_location
- cmi.core.credit
- cmi.core.lesson_status
- cmi.core.entry

cmi.core.score.raw
cmi.core.score.min
cmi.core.score.max
cmi.core.lesson_mode
cmi.core.total_time
cmi.suspend_data
cmi.launch_data
cmi.student_data.attempt_number
cmi.student_data.mastery_score

Lesson to LMS

cmi.core.lesson_location
cmi.core.lesson_status
cmi.core.exit
cmi.core.score.raw
cmi.core.score.min
cmi.core.score.max
cmi.core.session_time
cmi.suspend_data

See the AICC or SCORM specifications for descriptions of the usage of these data elements.

Restrictions with the current API version

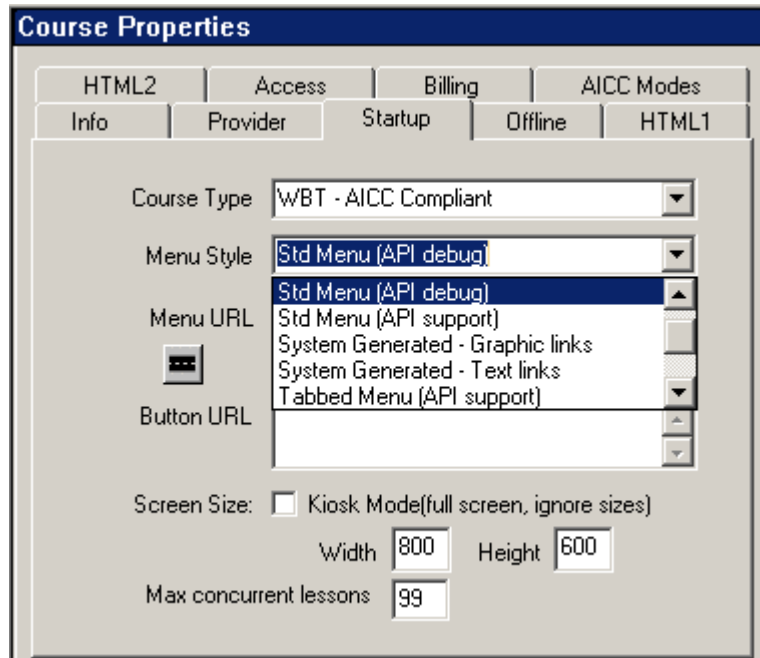
The WBT Manager API Adapter has been tested with MSIE 4+, and Netscape 4.5+ and is not supported with older browser versions.

Netscape 6 is not currently supported. A Netscape 6 compatible version will be posted to our support site as soon as it is available.

Under MSIE the lesson content must reside in the same domain as the WBT Manager web-server application or a security error will occur. This is intentional behavior on the part of the browser to prevent pages on one web site from affecting the contents of pages from another web site. We are investigating mechanisms to get around this difficulty.

Using the API

The WBT Manager API Adapter is embedded in the WBT Manager course menu pages. Because there are many pieces of courseware already written to the AICC HACP specification, WBT Manager allows you to control whether API support is provided for a course by selecting a **Menu Style** on the course's **Startup** property tab. The menu styles that support the API are:



- **Std Menu (API support)**
- **Tree Menu (API support)**
- **Tabbed Menu (API support)**
- **Std Menu (API debug)**

The first 3 new menu styles are identical in appearance to the original WBT Manager menu styles. The difference is that they embed the WBT Manager LMSAPI Adapter applet in a hidden frame. **Std Menu (API debug)** is different in that the frame containing the API applet is not hidden and you can see the LMSAPI function activity on the applet's surface. The API menu styles use the same menu generation scripts as the standard menus but parameters are added to turn API support and debugging support on. HACP is still supported in these menu styles so you can mix the two interface mechanisms in the same course.

Usage hints

- You must call LMSInitialize once and only once per lesson.
- You must call LMSFinish once before exiting.
- You can call LMSGetLastError after a function call to see if it completed successfully.
- If you set any values to be sent back to WBT Manager you should always set **cmi.core.lesson_status** and **cmi.core.score.raw**. If you do not then WBT Manager will assume that you intended them to be empty and update the student's record accordingly. The prior value will not be used (the assumption is that status and score belong to the current session). There is enough vagueness in the specification that other LMSs may use the previous value for the new session and record unexpected results. In order to have consistent results you should always set data elements to the values you want and not depend on default behavior of the LMS.

Sample API course

A simple HTML course has been provided to help you get started. This sample consists of 2 lessons.

- Lesson 1 has a single page with a set of links that allow you to send a specific status along with a random time to WBT Manager.
- Lesson 2 is an HTML form with links to call various API functions and receive results.

The course files are automatically installed with the WBT Manager web application but the course definition has to be imported into the WBT Manager database before you can run the samples:

- Use the course import utility to import **lmsapisamp.crs** into WBT Manager. This file will be found in the **courses/lmsapisamp** directory beneath the WBT Manager virtual directory.
- Set the menu style for the course to one of the API support styles.

Data collection changes

The lesson data collection capability of WBT Manager has been enhanced for version 1.51

Interactions data

WBT Manager now supports collection of AICC “interactions” data. This data is typically used to identify the student’s actual responses to individual questions in a quiz or test contained within a lesson. It may also be used to record information about a student’s performance in other lesson activities. Collection of interaction data is controlled at the lesson level through a check-box on the lesson’s “AICC Data” properties tab.

The screenshot shows a dialog box titled "Lesson Properties" with a close button (X) in the top right corner. The dialog has four tabs: "Info", "Provider", "Startup", and "AICC Data", with "AICC Data" currently selected. The "AICC Data" tab contains the following fields and controls:

- "Mastery Score": A text input field.
- "AU Password": A text input field.
- "Additional Parameters": A text input field with up and down arrow buttons on the right side.
- "Core Vendor Data": A text input field with up and down arrow buttons on the right side.
- "Track interaction data": A checkbox that is currently unchecked.

At the bottom of the dialog are "OK" and "Cancel" buttons.

Checking the **Track interaction data** checkbox will cause WBT Manager's AICC interface to include the value **c:\temp\cmiinter.txt** in the **interactions_file** data element of the response to a **GetParam** request.

See "WBT Manager and the AICC Specification" in the WBT Manager user guide or the AICC CMI001 specification for more details about AICC requests.

If a lesson subsequently sends a **PutInteraction** request then the data sent will be stored in a database table named **IACT**. The **IACT** table contains the following data elements:

| Column Name | Usage |
|-----------------|--|
| DTLOGGED | The date and time that the interaction data was received. |
| COURSE_ID | The WBT Manager internal ID (links to the COURSES table) for the course that the student was in when the interaction data was received. |
| LESSON_ID | The WBT Manager internal ID (links to the LESSONS table) for the lesson that the student was in when the interaction data was received. |
| STUDENT_ID | The WBT Manager internal ID (links to the STUDENTS table) for the student that the interaction data belongs to. |
| INSTNC | Reserved for future use. |
| IA_STUDENT_ID | The actual value for student_id sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_COURSE_ID | The actual value for course_id sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_LESSON_ID | The actual value for lesson_id sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_DATE | The actual value for date sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_TIME | The actual value for time sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_ID | The actual value for interaction_id sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_OBJECTIVE_ID | The actual value for objective_id sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_TYPE | The actual value for type_interaction sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_CRESP | The actual value for correct_response sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_VRESP | The actual value for response_value sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |

| | |
|------------|---|
| IA_SRESP | The actual value for student_response sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_RESULT | The actual value for result sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_WEIGHT | The actual value for weighting sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_LATENCY | The actual value for latency sent by the lesson in the interaction data. See the AICC CMI001 specification for an explanation of the usage of this data element. |
| IA_XTRA | The AICC specification allows for non-standard data elements to be included in a PutInteraction request. Any such data elements will be stored in this column as name/value pairs. |

Interaction data collection is currently supported only for the AICC HACP interface. Support for the LMSAPI will be added with the SCORM 1.1 upgrade. Check the WBT Manager support “patches” page periodically for the availability of this and other upgrades (www.ielearning.com/wbt/support).

IMPORTANT NOTE: WBT Manager can only collect interaction data that is sent to it. The individual lessons must be programmed to accumulate and send this data in a PutInteraction request.

High Score

Because a student can usually re-enter a lesson after they have received a score for it there is a possibility of the student re-taking a test and receiving a lower score than they originally earned.

WBT Manager 1.51 will now keep track of the student’s highest score for the lesson. New columns have been added to the **STATS** and **LSNHIST** tables.

| Column Name | Usage |
|--------------------|---|
| FHISCORE | The student’s highest score for the lesson. |
| HSCORDT1 | The date the student first achieved the high score. |
| HSCORDT2 | The date the student last matched the high score. |

Best status

For the same reason that a student’s score might be lower than their best, a student’s status for a lesson might drop. WBT Manager now tracks the student’s “best” status. A status is considered to have “improved” if:

- Old status was “not attempted” (or null), or “browsed”, and new status is neither.

- Old status is “incomplete” or “failed” and new status is “passed” or “completed”.

New columns have been added to the **STATS** and **LSNHIST** tables.

| Column Name | Usage |
|--------------------|--|
| HSTATUS | The student’s current “best” status for the lesson. |
| HSTATDT1 | The date the student first achieved the best status. |
| HSTATDT2 | The date the student last matched the best status. |