



Deploying content in the **Lotus Learning Management System**

Level: Intermediate
Works with: LMS
Updated: 03-Mar-2003

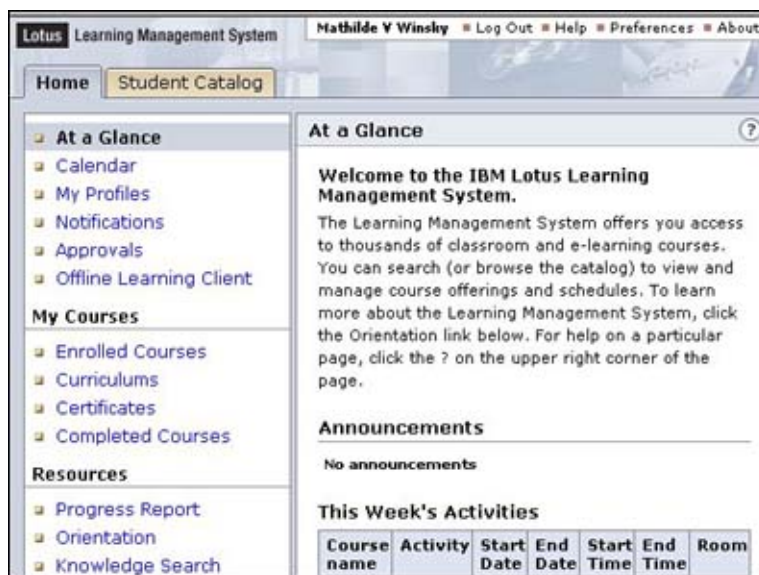
by
Kerry
Woodward

The IBM Lotus Learning Management System (LMS) provides a framework for delivering many types of training. This article explains how course content fits into this framework, regardless of where that content comes from. The information is intended for course developers and course administrators, whether they are new to LMS or have used LearningSpace, a previous online learning product developed by Lotus.

What is content?

Content is the collection of words, pictures, videos, sounds, questions, and other interactions that make up a student's learning experience. Content is organized into courses that are delivered and managed in LMS. Most courses contain several learning activities, targeted to specific training objectives. For example, a course might have a pre-test to assess a student's current skills, a choice of topics geared to students' pre-test results, and a post-test to judge the course's effectiveness.

Students launch LMS from their Web browsers, and then click links to launch a course and an activity.



Each activity may contain several screens of content with Next and Previous buttons to help students navigate through the screens. Each screen has several files associated with it to display all the text, graphics, and other multimedia content needed for the learning activity.



LMS distinguishes between course content (what students see on the screen) and course structure (the organization that determines how students move through the course). The distinction will be important later in this article when you learn how courses are stored and updated in the system.

LMS handles content in a variety of file formats:

- Web-based content is designed to be displayed in a Web browser. It is usually delivered as HTML and multimedia files, but may be delivered in a proprietary format that requires a browser plug-in. Web-based content needs to be stored on a Web server running HTTP.
- File-based content usually runs from a local machine. The files may be accessed from CDs, file servers, or local client machines. LMS launches file-based content from a Web browser.
- Virtual Classroom content is delivered in the Virtual Classroom, a venue for students and an instructor to do coursework simultaneously at a pre-arranged time. Virtual Classroom sessions are run in a Web browser and the files are stored on a LearningSpace - Virtual Classroom (LVC) server.

This article focuses on deploying Web-based content.

Where does content come from?

Content is built by course developers, people who have the knowledge and skills to create an online learning experience that is both effective and fun. As individuals or teams, course developers must be adept in all aspects of online education—as experts in a certain subject area, as Web developers experienced with course development software, as Web designers who can create appealing screens, and as instructional designers who understand online teaching strategies. When course developers design a content screen, they consider the visual and auditory impressions, as well as the student's own interactions with the screen, to create a complete learning experience. Each screen must also blend with the context and design of other screens and contribute to the instructional goals for the course.

Many courses are developed and created by companies that sell online courses for profit or as an additional service to their customers. Such companies include IBM, NETg, SmartForce, and SkillSoft, but there are many other vendors in the business of developing and selling courses. These training vendors generally use their own software tools to develop courses. Courses may also be developed by consultants hired by an organization to create a particular course, or they may be employees who have the expertise to create specialized training for their work groups. These individuals may use off-the-shelf online development tools such as the Macromedia eLearning Studio (Authorware, Flash, and Dreamweaver), or they can use the authoring tool built into LMS.

Regardless of how and where courses originate, they can all be deployed in LMS, which is designed to be flexible enough to handle any course that adheres to the industry's two online learning standards—AICC and SCORM.

These standards emerged in the last decade to ensure a uniform method of Web-based communication between content and learning management systems. To read more about these standards, see the [AICC specification](#) and the [ADL Sharable Content Object Reference Model \(SCORM\) specification](#).

Servers that interact with content

The following servers are involved in the management and delivery of course content within LMS.

LMS Server

The LMS Server is the core of the LMS installation. There can be only one LMS Server per installation, but each LMS Server can be hosted on a server cluster. The LMS Server provides the main user interface and has its own relational database to store information about users, courses, other servers in the system, and system settings.

Delivery Server

The Delivery Server is part of the LMS installation. The primary function of each Delivery Server is to sequence, launch, and track content, and to provide a frame for content within the larger LMS interface. There can be more than one Delivery Server, and each one can be hosted on a server cluster. The Delivery Server has its own relational database that stores information about the LMS Server and temporary tracking information.

Each Delivery server:

- Launches course content in a student's browser window.
- Interprets sequencing instructions built into the content that navigate a student to the appropriate location in a course.
- Performs interim progress tracking and submits it to the LMS Server for permanent storage.

Content Server

This server is a separate Web (HTTP) server. It is not part of the LMS installation. The Content Server stores course content files, including HTML, image, and multimedia files. Each Delivery Server can use one or more content servers—if more than one Content Server is used, the content is stored on each of them.

Course deployment

Before you can offer a course in the LMS Student Catalog, you must import and register it in the system. This process distributes the course content and structure so that the system can launch activities and track student progress correctly. A course that you receive from a course developer goes through the following phases before it can be distributed to students:

1. Packaging

The LMS Server imports any course that is compressed into a course package. This package contains one or more XML manifest files that define the course structure, a list of the files it uses, and any additional data associated with the course. The package also contains either the content files needed for running the course or the URLs that define the location of content files. If your course is not already delivered in a course package, you need to import your course into the LMS authoring tool and then package it from there.

Each course package processed through the authoring tool contains a unique package ID and a version number. This allows courses to be updated correctly in the system later. Packages created elsewhere may or may not have these identifiers. If the package ID and version number are missing, a package can still be imported, but the course won't recognize subsequent imports of the package as updates, so the package won't be updated automatically.

2. Importing

You initiate the importing process using one of the two client-side utilities provided with LMS—the authoring tool or the LMS command line import utility (CLIMP). This process sends the course package to an FTP server, where it is imported by the LMS Server, which extracts the structure and content files.

The LMS Server creates a new directory for each imported package and extracts the contents into that directory. The name of the directory is generated randomly. By default, the directory is created in a top-level directory called "packages" in the Learning Management System Web application directory, but you can specify a different directory when you install LMS.

3. Registering

A course shown in the Student Catalog is an offering. Each offering is based on the content and structure of a course master. The package that you imported appears in the LMS Course Masters Catalog as an unregistered master. Register the master to make it available to anyone who wants to use it as a template

for their course offerings.

To register a master, you need to be logged in to the LMS interface with permission to create course masters. This permission, "Manage Masters Catalog," is assigned to the administrator role by default.

4. Offering

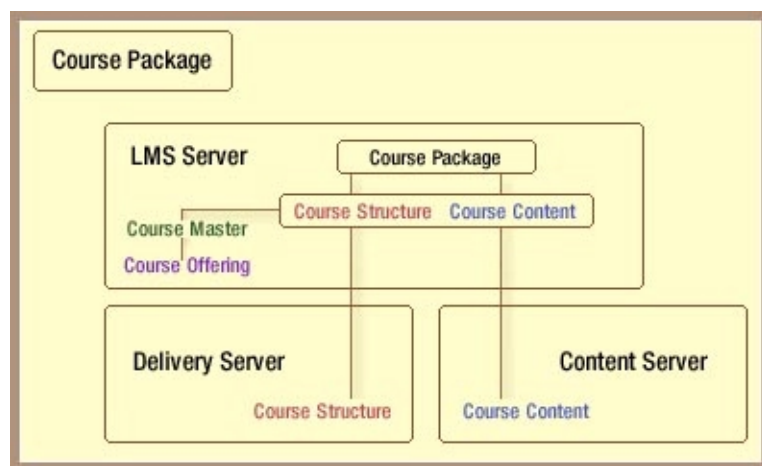
Each offering is unique with its own Delivery and Content Servers, availability schedule, instructor, location, enrollment list, and so on. After you create an offering and make it available, it appears in the Student Catalog for students to preview, enroll in, and complete.

You need to be logged in to the LMS interface with permission to create course offerings. This permission, "Manage Offerings Catalog," is assigned to the administrator role by default.

Where course information is stored after deployment

When you save a course offering, course information is transmitted to the servers that will be involved when students launch a course. The LMS Server sends the structure of the course to the Delivery Server associated with the offering. The LMS Server then transmits the content to the Content Server teamed with the Delivery Server. If the Delivery Server is teamed with a cluster of Content Servers, the content is sent to each of them. When a student launches a course, the Delivery Server gets the content from the Content Server, so it can be displayed in the student's browser window.

The following illustration shows where course information is stored when you save a course offering. Most information is stored in the relational databases associated with the LMS Server and the Delivery Server, rather than the servers' local file systems. The exceptions are course content and packages, which are stored as local files.



How installation settings affect course deployment

Much of the infrastructure for course deployment is set up when the LMS and Delivery Servers are installed. If you are not the person who installed the software, you may find it helpful to see the installation settings that affect how and where courses are deployed.

Content Management Settings

This screen allows you to specify Content Management settings on the LMS Server. The "Path to Imported Course Packages" is the path on the LMS Server where imported course packages are stored. You have to create the directory for the course packages by hand. By default, the IBM HTTP server expects this directory to reside in the directory under ...Program Files\IBMHTTPServer\htdocs appropriate to the installed language (en_us, for example). If a non-absolute path is entered, it is assumed to be relative to the Web application directory. If you use a cluster of LMS Servers, you should specify a location on a file server that all the LMS Servers can access. The FTP user name/password combination is an account that the LMS Server uses to access any Content Servers that have content sent via FTP. Any Content Server (which you define in a later screen) that uses FTP must create an account with the specified FTP user name/password.

Content Management Settings
Path to Imported Course Packages <input type="text"/>
FTP User Name for Deploying Content <input type="text"/>
FTP Password for Deploying Content <input type="password"/>
Notification E-mail Format <input type="text" value="Plain"/>

Web Server Settings

This screen defines two important pieces of information for the Delivery Server. The URL of Deployed Content is where the Delivery Server looks for content when it launches a course (for example, <http://contentserver.acme.com/content>).

You need to create the directory in which to place deployed content, as the application does not do this for you. By default, the application looks for a subdirectory in the HTTP server's htdocs subdirectory for the installed language (for example, in a subdirectory of IBMHTTPServer\htdocs\en_US).

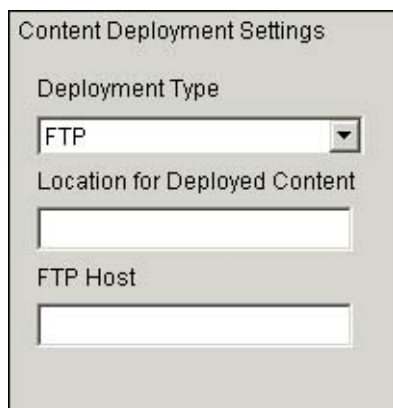
If Content Servers are clustered, this is the single URL that is used to access the cluster. The Tracking URL is the URL to the tracking mechanism on the Delivery Server. This is pre-populated with the Base URL specified for the Delivery Server during installation and generally does not need to be changed.

Web Server Settings
URL of Deployed Content (Example: http://contentserver.acme.com/content) <input type="text"/>
Tracking URL <input type="text"/>

Content Deployment Settings

This screen allows you to add one or more Content Servers to the system. For each server you add, you specify a Deployment Type to determine how the LMS Server sends content to the Content Server when a course offering is saved. Content can be sent through FTP if the Content Server is set up for FTP or through the local file system.

If you choose FTP, the LMS Server sends content to the FTP host you specify, in the location you specify (if any), using the FTP user name and password you defined on the Content Management Settings screen.



The image shows a 'Content Deployment Settings' dialog box. It has a title bar at the top. Below the title bar, there is a label 'Deployment Type' followed by a dropdown menu showing 'FTP'. Below that is a label 'Location for Deployed Content' followed by an empty text input field. At the bottom is a label 'FTP Host' followed by another empty text input field.

When you choose File System, the LMS Server sends content to the absolute file location you specify. Note that the specified location must be accessible to the LMS Server.



The image shows a 'Content Deployment Settings' dialog box. It has a title bar at the top. Below the title bar, there is a label 'Deployment Type' followed by a dropdown menu showing 'File System'. Below that is a label 'Location for Deployed Content' followed by an empty text input field.

Managing changes

Over time, courses need to be updated to incorporate new material, to remove obsolete information, or to improve the learning experience. Structural changes affect a course's outline—the components, properties, and sequencing that make up the flow of a course. Content changes affect the text, images, or media files that students see when they take a course.

Deploying updated courses

The steps for distributing changes to a course are similar to those used to deploy the course originally. However, you have the option of deciding how far to disseminate the changes. You can either update a master without changing existing offerings, or you can distribute changes to all active offerings created from the original master. Email notifications are sent to administrators when the update has been accepted or rejected.

1. Packaging

If you used the LMS authoring tool to create the original course package, the first step is to repackage the course with the changes you made. The package ID remains the same, but the version number increments by one. The course content and structure version numbers are also updated if they have changed since the last time the course was packaged. The content version number is checked only when content is stored in the package. If all the content is external to the package, the content version number is ignored.

2. Importing

Use the authoring tool or the command-line import utility to import the updated package into the system. The LMS Server checks the package ID and version number to determine if the package is an update to a previously-deployed course. If no package ID was specified, the server generates one. If the updated course package contains only a structural update, the structure is imported to the LMS Server, but the content is not.

If you have received an updated course from a courseware vendor and it doesn't have a package ID or its package ID isn't the same as the previous course, you can't update the corresponding master directly. Import the package and follow the instructions (described later in this procedure) to use the Masters Catalog option "Update a master using an imported master."

3. Updating the master and its offerings

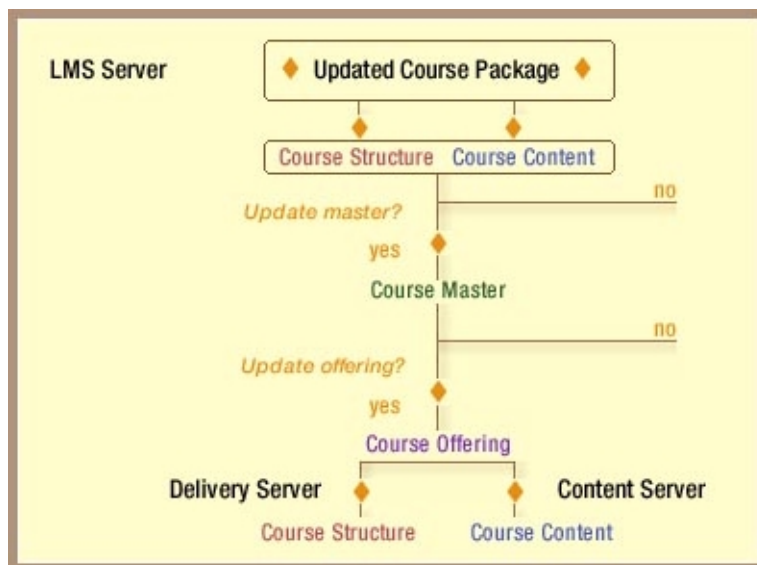
When LMS finds a match between the course ID of the re-imported updated course package and the original master, it flags the master in the catalog as a master with changes pending. You now have choices about how to handle the updated course:

- Do not update the master. Do this when you don't want the original master or its current or future offerings to be affected by the update. Reject the update by selecting Discard update. This action deletes the master from the list of masters with pending updates.
- Update the master, but don't update any offerings. Use the Update Master button in the Course Catalog to initiate the process, but do not select Update all offerings. Any new offerings created from the master will reflect the update, but you won't change or disrupt any active course offerings.
- Update the master and all its offerings. Use the Update Master button in the Course Catalog to initiate the process and select Update all offerings. This updates the master and all existing and future course offerings created from it. This choice changes all active course offerings created from the master. Avoid confusing students by making the course unavailable until the content has been successfully updated.

You can also update a master by using the content and structure of a newly imported package. Select "Update a master using an imported master" and select the new unregistered master to use as the new format.

Course update workflow

The illustration below summarizes how updates flow through the system. If an offerings manager chooses to update all offerings, existing offerings created from this master are changed, and structure and content changes are sent to the Delivery and Content Servers.



Would you like to learn more?

Deploying content in the Lotus Learning Management System represents just one piece of a multifaceted management system. If you'd like an "under-the-hood" perspective, see the *LDD Today* article "[Understanding the architecture of the Lotus Learning Management System.](#)"

ABOUT THE AUTHOR

Kerry Woodward is a technical writer in the IBM Lotus Software Division. She has worked on documentation for Notes and Domino, LearningSpace, and the Learning Management System during her 12 years with Lotus.