

Moodle Auto-Linking Feature

Jim Williamson
UCLA OID Teaching Enhancement Center
July 2007

The following essay illustrates some of the rewards and complexities using an open source software system like Moodle. While the benefits of OSS include access and manipulation of the underlying source code, simply having those freedoms does not lessen the burden of running and maintaining the system. What follows is an example that begins with trying to understand a seemingly trivial part of Moodle functionality, and ends in further unexplored technical and policy-related issues. Ultimately, the example reiterates the idea that understanding complex systems is not easy, and resolving even apparently simple issues sometimes requires great expenditures of time, energy, and perseverance with unforeseen ramifications.

The Moodle course management system has a feature called “Resource Name Auto-Linking” which as its name indicates, allows Moodle to automatically recognize and create links between site resources. That is, if a user creates a web page titled “Moodle Feature Set” and then in another Moodle web page writes a sentence reading, “There are many options in the Moodle feature set,” the system will automatically highlight the common words with a surrounding light-gray box and create a hyperlink between the two documents (see below). Some users find this auto-linkage between documents a valuable feature, but other users might prefer not to have it.

Introduction to Moodle

- Moodle Feature Set
- Moodle editing tools
- Moving Moodle items
- Notes on Using Moodle
- Elements of the UCLA Default Template
- Hidden items

Unfortunately, the feature is not sophisticated and can only be turned on or off by the System Administrator. The feature has no parameters or settings, and the toggle affects all users of the system: if it is turned off, it is turned off for everyone, and turning it on makes it available to everyone. When auto-linking is turned on, end users can still explicitly “unlink” items using a button in the toolbar, but they must do this manually for each link they do not want.

Filters ?

Name	Hide/Show	Up/Down	Settings
TeX Notation		↓	Settings
Database Auto-linking		↑ ↓	
Glossary Auto-linking		↑ ↓	
Resource Names Auto-linking		↑ ↓	
Wiki Page Auto-linking		↑ ↓	
Activity Names Auto-linking		↑ ↓	
Email Protection		↑ ↓	
Multi-Language Content		↑ ↓	
Tidy		↑ ↓	
Algebra Notation		↑ ↓	
Multimedia Plugins		↑	Settings
Word Censorship			Settings

Note: No Settings

Through accident and necessity, I have recently learned more about this feature than I ever wanted to. I had been developing a training course on a Moodle installation where the feature was turned off, and when I moved the course to the UCLA campus production system (PROD), suddenly my course pages had numerous gray boxes and items linked to each other. I was annoyed because I had not asked the system to behave this way, nor did I want it to.

Perplexed as to the cause of the behavior and since I am one of the project team members, I began researching where this feature had come from. On the Moodle discussion forums I found the answer, and identified the culprit as the aforementioned filter called “Activity Names Auto-linking.” In an effort to better understand the feature, I logged onto the campus development server (DEV), created some tests, but found to my chagrin that turning the setting on or off did not give me the results I expected. Worse, I was getting different results between the DEV and PROD servers even when using the identical settings!

Something very strange was happening:

- the feature was activated on both machines;
- it was the same course and course content;
- it was the same user account;
- it was the same user browser and operating system;
- I logged off, cleared the browser cache, and restarted the browser before switching between environments.

Despite all this, Moodle was behaving differently in the DEV and PROD environments. Unfortunately, it was a mystery that at that moment I did not have time to research further.

A week later, the auto-linking feature came up in discussion amongst the project support people, who told us faculty were complaining. When a System Operations Subgroup meeting ended early, I explained the odd behavior to the Sys Ops team and, using a projector and laptops, four of us team members spent an hour experimenting with the DEV and PROD systems, changing settings, trying different browsers and operating systems, refreshing the browsers, looking at HTML and PHP source code, etc., but at the end of the hour the issue remained a mystery: we could not get consistent behavior across the DEV and PROD environments using the exact same settings.

Shortly thereafter, on a day off, I spent numerous hours performing multiple tests between five different instances of Moodle using multiple servers, operating systems and browsers.¹ My tests involved creating various items and the corresponding text phrases that initiate an auto-link, then checking the items and the links in all my test environments. This meant moving from computer to computer, visiting and revisiting various URLs in multiple browsers, repeating the experiments with multiple sets of variables, and recording the results in a spreadsheet.

Only at the end of this testing did I conclude with a fair degree of certainty that the odd behavior is caused by a server-side caching mechanism.² In an effort to increase server efficiency, the server cache stores rarely-changing documents and only releases new, correct versions of Moodle pages after about a half-dozen page refreshes; in our Sys Ops group tests we had simply not refreshed our browsers enough times. To complicate debugging matters, the server-side caching mechanism appears to have some built-in logic and intelligence, so Moodle does not treat every function in the system the same way; some functions, like forum posts and chat session data, are excluded from the caching and therefore refreshed immediately, presumably because of the immediate and changing nature of these communication tools.

Additionally, I learned the Moodle administration lists multiple modules creating auto-links: Resource Names Auto-linking, Glossary Auto-linking, Database Auto-linking, Wiki Page Auto-linking, and Activity Names Auto-linking.³

One might reasonably ask why the Moodle developers only made this setting at the system-wide level and not at the course-level so professors could turn it on or off as desired; in fact, there is a bug filed with Moodle requesting this change.⁴ While in hindsight it is easy to say that the system-level-only approach is not the best option, understanding why this feature of Moodle was developed this way is not easy. After extensive experience with open source projects like Plone, Sakai, and Moodle, oddities like this no longer surprise me. Since the Moodle system is driven by contributions from the community, there are dozens of layers and forces at play.

Here is my speculation on how auto-linking became part of the Moodle code base (even though I have conducted some preliminary research, I have not yet discovered the actual reason): A developer learned how to create the auto-links and offered the idea to the Moodle community,

¹ Three different versions of Moodle (versions 1.7, 1.7.2, and 1.8), four different versions of PHP (the programming language Moodle is written in: versions 4.3.9, 4.4.4, 5.1.2, 5.1.6), three different operating systems (Windows XP and Vista, and Mac OS X), and four different web browsers: Firefox and IE (Windows) and Firefox and Safari (Mac).

² This conclusion is based on observation and may not prove to be entirely accurate.

³ None of these modules offer additional administrative settings, but in the Glossary the professor has the option of turning of auto-linking at the course level.

⁴ Link to the Moodle bug database: <http://tracker.moodle.org/browse/MDL-7336>

probably announcing it in a discussion forum posting. Those who saw the forum post and liked the auto-linking idea, replied with, “Yes, this sounds like a great feature – let’s add it,” and so the Moodle core team added it into the code base. It would not surprise me if the people who determine what features are included in the Moodle core do not have the resources to exhaustively review and understand the implications of every community contribution. Part of a community-based open source project is trusting the user community, and if the community appeared to accept and like the auto-links, a decision was made to include it. Unfortunately, developers are notorious for not fully appreciating the end users’ perspectives, so it might be the case that the original developer never thought someone would not want it, or perhaps he did, but realized writing the additional code to also create a course-wide setting was too difficult or too time-consuming to address. Or perhaps someone decided there are already too many settings per each course, and adding more simply adds confusion and frustration to the end user. Perhaps the general benefit outweighed the particular inconvenience: it might have been reasoned that since there is a way to explicitly unlink items, Moodle didn’t need to add yet another course setting. But I can’t really say, and each unanswered question remains a research project in of itself.

From this episode, though, we can make the following comments:

1. Complaints about the auto-linking feature should be considered minor (because it does not involve data loss or security breaches), but because the function is prevalent, a determination should be made. Because auto-linking affects every user on the system, this episode reveals a policy issue as to whether auto-linking should remain activated. Some users may react fiercely to it.
2. Although it is a setting that end-users cannot change, they can explicitly unlink items using a button on the toolbar. Unfortunately, this imposes a burden on users who do not want auto-linking and they may not view it as merely an inconvenience.
3. While this is a seemingly trivial problem, a tier-1 Help Desk staff or student staff would not have been able to resolve the issue because they do not have the system-wide administrative access to do the necessary testing. We only gained an understanding of the issue after extensive research through the lowest system level access. For practical and security reasons, this low-level system-wide access is restricted to few people, which creates a cascading series of dilemmas:
 - a. The frontline Help Desk staff could not have easily understood the issue because they do not have the necessary system access, but we only learned this after a great deal of work. How many more settings or features in Moodle exist which require this amount of work to gain understanding?
 - b. Although full system access necessarily resides with the few, the few don’t have the time needed to devote to very problem, especially seemingly surface issues. How can we best utilize the limited time of the most experienced people?
 - c. The auto-linking behavior is not a bug, but a characteristic of the system. Given our resource constraints, how do we gracefully evaluate and address the continual flow of end-user functionality requests?

Obviously, more and deeper experience with Moodle will lessen the impact of this early illustrative case, but it points to how difficult seemingly trivial issues can be.

4. A completely satisfactory resolution of this issue will require more work, such as research into whether Moodle plans to offer a course setting for turning auto-linking on or off. If not, does UCLA deem this important enough to devote the programming resources necessary to develop a custom, flexible way to resolve the issue? If so, even if we offer our customization back to the Moodle community, there is no guarantee Moodle will embrace it or deem it as important as we do. In this case, UCLA will need to reconcile and incorporate our changes into each new upgrade of Moodle, and hope our code interacts seamlessly with the new Moodle release. If not, we then need to revisit our work and bring it into compliance with the new Moodle code. In balance, it is doubtful UCLA would devote too many resources for this specific functionality, but this general case study will certainly apply in other instances.

In the larger scheme of things, the auto-linking feature is a minor issue determined by a single setting inside of a system with hundreds of settings at multiple levels, but it affects all users on the system and took a concerted effort to gain an understanding of where the auto-linking highlighting came from, why we could not seem to turn it off, and why the DEV and PROD servers were acting differently. Between the many people involved, a dozen or more hours were spent coming to a resolution. Was it worth the time spent tracking down the odd behavior? In this case, I'd say yes because we have learned a bit; with the discovery of the server caching we now have an area to focus on in the future if similar odd behavior appears; and we have this case study to illustrate some of the complexities we're facing rolling out such a complicated system to thousands of users.

Postscript

The ability to control this in multiple contexts (i.e., Allow auto-linking in Discussion forums, but turn it off in Quizzes) was introduced in Moodle 2.0.