DoclibraryPHP: Proposed File Link Schema

25 June 2012

Background

In order to reduce un-necessary duplication of file content, a linking schema needs to be devised allowing users to upload a file once and then link to it from other directories. Throughout this document, the file being *linked to* will be referred to as the 'parent item'.

Security Considerations

In order to mitigate possible security risks, cross subsite linking should not be supported. When external links are implemented, a specific user role should be created so that administrators can control which users can link to external resources.

Link Schema

In order to reduce query times, a dedicated links table should be created, ideally with the following shcema

Field	*Type*	*Index*	*Notes*
Id	Int	PRIMARY	
LinkRef	Int	INDEX	DocID of the Link created in the *Files* table.
LinkSource	Int	INDEX	DocID of the Parent Item

Link Identification

It would be redundant to add an additional field to the Files table in order to identify links. Instead the MIMEType field can be used as these already need to be checked for each file when displaying a directory listing/search results etc.

Index filetypes already provides an index of all MIME-Types in the Files table, so a global search for Links would not require a full search of the Files database.

In order to identify an internal link, the MIME Type LINK/LINK should be used. All relevant scripts will then be updated to cross-reference the Links table and then retrieve relevant information from the parent item.

External Links should carry the MIME-Type LINK/EXTLINK. Further cross-referencing will be necessary in order to retrieve the URL to link to, but phase 1 will not include support for this function.

When Phase 2 is implemented, alongside external links the MIME type LINK/CALLINK will be implemented, allowing users to link directly to a calendar entry.

File Deletion

Measures will need to be taken to prevent deletion of a Parent Item whenever any page that allows file deletion is presented. This includes Directory listings and search results.

The proposed mechanism is as follows;

For each file, retrieve a count of any Linked files and store in a hidden field. If the count is greater than 0 then the user should be notified that "there are x links to this file" and asked if they wish to continue. If so, then any child links should also be deleted.

Additional consideration needs to be paid to the directory deletion functionality. Currently, individual files are not examined so a mechanism needs to be devised to prevent the creation of orphan links. Simply deleting the Parent item and any related links is not considered a suitable long-term mechanism, but may be acceptable if the user is warned that this is a possibility when submitting the deletion request.

Link Creation

A new page will need to be created in order to allow users to create links. This should be implemented with the final phase in mind. Creation should be via an icon in the parent directory (much like the 'New Text File' function) with the following work-flow proposed

- User clicks 'New Link' icon
- User selects link type from a drop-down menu (for phase 1 this will be hidden and a choice of Internal assumed)
- Hidden indicator set (for benefit of Post Processing)
- Relevant fields displayed

Internal Link

User presented with a Pop-Up box allowing them to select the file to link to. This should function in the same manner as the Directory change mechanism for file/directory editing.

External Link

User should be presented with a field to enter a valid URL in. Basic validation (i.e. check for http/https)

Calendar Link

A new page will need to be created, allowing the user to browse calendar entries and select the relevant entry.