New Blog System

Posted At : April 27, 2007 8:52 PM | Posted By : Jon Hartmann
Related Categories: Experiments

Given my frustrations with the BlogCFC setup, I'm endeavoring to create a system that will be a little easier to use to maintain multiple blogs. I'm building this system inside Mach II and will use Reactor for database interactions so I can get a handle on these systems. I've outlined some constraints and put together a kind of time-line.

Constraints

New blog system must use the blog database structure from BlogCFC as much as possible. Although I hate it, its a practical decision; I can maintain my blogs with BlogCFC while getting a test database and its easier for others to migrate later.
The new blog system must be able to more flexibly support multiple blogs, keeping file duplication between blogs to an absolute minimum.
Blogs must be "skinnable", having each blog page pass through a single display page which can be selected dynamically. This allows end users to better control the appearance of their blog from a single page.
Blogs should be Object-Orient as much as possible (make use of beans, etc).
The new system must have better integration of a WYSIWYG editor for ease of use.

Benchmarks

1. Create a blog object that can be loaded with data and create a "skinning" process for that object.
2. Create the process to load the blog object from the database.
3. Build a new admin interface for the blog.
    - WYSIWYG
    - "Static" page creation
4. Setup caching of blog builds to increase performance.
5. Optimize the blog for integration with Mach-II (automatic bean creation and such).
6. Implementation of Ajax and Javascript elements.
    - "In-line" comment and post edits
    - "Real-time Previews" - Use of Javascript and Ajax to display post previews as you create the post.
7. Widgets - The ability to place small interfaces where you choose on your blog along with a system to easily plug in new widgets
    - Link Managers
    - Image galleries
    - Message Board Support
    - Multimedia Posts (video, etc.)

Technologies

- Framework: Mach-II - I've picked Mach-II as my framework because it has some very nice features that handle the automatic creation of Beans and then can pass them off to other event handlers, and I think that this is a very nice feature when looking at an OO design. In addition, I'm hoping that I can find some synergy with the Record Objects that Reactor Creates.
- Database Interactions: Reactor - Really, its the only database tool of its kind that I've come across so far. It also has several advantages over traditional DAO objects I'm used to working with: its Record Objects are very similar to Beans, and it allows the creation of new requests without affecting old queries. The first part is useful if I can connect record objects to my Mach-II framework as Beans, and the second is useful for implementing a system for Widgets that allows for plug-in-play like interfaces.
- Javascript/Ajax Framework: Prototype - I'm just in love with Prototype, and between it and the libraries based around it (such as Scriptaculous) I've got about everything I need. I might even try getting into the Spry framework from Adobe Labs, as they really seem to be getting a lot of things right in that system.