Subject: Re: Is Radicore better than Ruby On Rails?
Posted by martin on Wed, 21 Jun 2006 09:11:41 GMT
View Forum Message <> Reply to Message

Hi Aviator,

It would be easier to discuss if you provided some information about what you are dissappointed about in the Radicore framework or more specifically the demo application. Also, what is it that you think is Rubbish in the demo application?

Personally, I see the following Radicore features as superior to other frameworks when it comes to building an administrative web application to be used on a daily basis behind the firewall in a company. The features also applies when building, for example, a backend application that handles all the contents of a public web application:

User Transaction Templates: One of the major features of this framework is the user transaction/web page templates that the programmer can choose from instead of starting from scratch for each page. The developer chooses a user transaction that he or she whishes to implement, such as view employee data, and then finds a suitable template for that user transaction. The framework comes with 25+ templates that can be mixed together in an endless number of combinations to create the desired web page flow...

Web Page Flow Context Management: One of the major hurdles when developing web applications is how to manage the context when navigating between web pages. This is completely automated in Radicore and transparent to the developer. If a developer for example wants to list employees and then view one or more of those, appropriate transaction/web page templates can be combined to achieve that without any session data programming.

Role-based Access Control (RBAC): Radicore comes with a fully integrated role-based access control system. Access to any task (i.e. menu, page, operation) in the system can be controlled dynamically by assigning Task Access rights and Field Access rights to roles. A user is then assigned a role that control the access to the application for that user and menus will also contain only the accessible menu items...

Dynamic Menu System: The menu system and the navigation system in Radicore are build on top of the RBAC module. When a user logs in the menu will automatically be build to show only those menu items that the user has access to depending on his or her role.

RBAC Management: A web application to administer the access control comes with the framework. You can also integrate the pages of this application with your own security management application as all components build with Radicore are re-usable.

Audit Logging Management: Audit logging is becoming more and more important these days with the Sarbanes-Oxly act etc. Radicore is build with this in mind from the ground up. All data in the system are by default auditable and the framework comes with an Audit Log Management application where audit logs can be inspected.

Internationalization (I18N): When a new web application is developed with Radicore it is automatically supporting internationalization. All text in the new application is contained in property files and all standard texts for buttons and messages are contained in property files. The UI is generated via an XSLT process and the standard stylesheets for different web page templates are unaware of any specific entity data. All application specific texts are fed into the XSLT process at runtime. So it is quite easy to support I18N.

Integrated Help System: Every task in the system is described in the database and context sensitive help links are
available in the UI on each page to take you to the help page
for current task.

Print Preview: Each web page in a Radicore application can be converted into a print preview page by one simple click on a link at the top of the screen. No extra programming.

Validation: Primary validation is build in to the framework and it will be executed automatically by looking at the field specifications to determine if an error message need to be displayed. Secondary validation is also supported via custom programming to for example check two fields against each other.

Themes/Styles: The applications build with Radicore can be made to look just the way you want them too by modifying the CSS files that comes with the framework.

Code Generator: To develop with Radicore is already extremely fast, but to speed it up even more it comes with a code generator in combination with a data dictionary. With the code generator you connect to the database and read all metadata for tables you which to create web pages for. After that you work in steps to configure field specifications, setup table relationships etc and in the final step you generate code, that is ready to run.

I think these are some really nice features I have not come across in any other framework.