Subject: Internationalisation suggestion
Posted by bonzo_bcn on Mon, 30 Jun 2008 19:46:47 GMT
View Forum Message <> Reply to Message

What do you think about this suggestion?

In the column dictionary I would suggest adding a one to many relationship with a new table that had two fields: language_id and title_for_the_column_in_this_language.

When importing a table you could create a record with a default title (as you do now) for the default language.

Then, when you need to add a description for a new language you should come to the column dictionary and add the record to this new table.

I also think that the titles shouldn't be hardcoded in the .screen.inc files, couldn't the framework retrieve the titles from the dictionary? This way if you change a title for a table you shouldn't go file by file updating it.

Subject: Re: Internationalisation suggestion
Posted by AJM on Mon, 30 Jun 2008 20:03:44 GMT
View Forum Message <> Reply to Message

I do not use the data dictionary for internationalisation, instead I use a series of text files, one per language, in the 'text' subdirectory. I deliberately chose this method as a single text file can contain multiple entries whereas with the database option I would would require a separate read for each entry, and this would be much slower.

It is also easier to provide translations as all you have to do is give the translator a text file which he can modify instead of getting him to change the contents of the database.

It is possible to have translations for any piece of text - screen titles, column names, menu buttons, navigation buttons as well as error messages.

As my current method works very well I see no reason to change it.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Mon, 30 Jun 2008 20:14:26 GMT
View Forum Message <> Reply to Message

It's not about if it works or it doesn't I guess it's about improving radicore

By using the data dictionary for labels you simplify the internationalisation, don't have to deal with

files etc. Is much easier for an inexperienced user.
It's consistent as metadata is stored in the dictionary, not in a file. You can visually see tha labels without having to go to the files etc.
You could also have dictionary tables for the other things that are not related to tables.

Also having the .screen.inc files retrieving the labels from the dictionary would be great, I'm sure there is a way of caching the titles the first time you need them so there is no need to retrieve them every time.

Subject: Re: Internationalisation suggestion
Posted by AJM on Mon, 30 Jun 2008 20:37:14 GMT
View Forum Message <> Reply to Message

I do not consider you suggestion to be an improvement over the existing method, it is just different. I have worked with other systems in the past that used non-database files for all foreign language text, so I am not alone in thinking that is has merits.

Unless you can present a compelling reason why I should switch to a different method I see no reason to change anything.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 08:55:57 GMT
View Forum Message <> Reply to Message

AJM wrote on Mon, 30 June 2008 16:37I do not consider you suggestion to be an improvement over the existing method, it is just different. I have worked with other systems in the past that used non-database files for all foreign language text, so I am not alone in thinking that is has merits.

Unless you can present a compelling reason why I should switch to a different method I see no reason to change anything.

I've been working also with a big ERP (kind of SAP) and they had it done this way. There was a cache system where all the labels where stored the first time they where retrieved.

In this scenario: you have you application finished, full of transactions etc, now you client asks you to change the title of a column, as it is right now you should have to deal with a lot of files and I'm sure you will forget some. With my suggestion you just go to the dictionary, change the label and that's it.

As for other labels fr buttons etc create a table with "label_id, language_id, text" then a global function can be used wherever you want to retrieve a label for a given language.

You also avoid the hardcoding of titles in the screen.inc files.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 09:09:37 GMT
View Forum Message <> Reply to Message

OTOH I believee radicore is about making web application easier, I believe that after creating tables etc, you should only need a web browser and an editor to create/maintain a web application.
I don't want to bother you more, but I belive my suggestion is easier to use, more flexible and it separates more the presentation layer from the data layer.

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 09:29:29 GMT
View Forum Message <> Reply to Message

Quote:In this scenario: you have you application finished, full of transactions etc, now you client asks you to change the title of a column, as it is right now you should have to deal with a lot of files and I'm sure you will forget some. With my suggestion you just go to the dictionary, change the label and that's it.
If you have a separate entry in your database for each language then changing the label text for a column will still require a separate update for each language.

In my design each subsystem has a single text file for each language containing the text for everything - task titles, column labels, menu buttons, navigation buttons and error messages - so to change the label for a column still requires one update per language. There is no saving with your method.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 09:36:24 GMT
View Forum Message <> Reply to Message

AJM wrote on Tue, 01 July 2008 05:29Quote:In this scenario: you have you application finished, full of transactions etc, now you client asks you to change the title of a column, as it is right now you should have to deal with a lot of files and I'm sure you will forget some. With my suggestion you just go to the dictionary, change the label and that's it.
If you have a separate entry in your database for each language then changing the label text for a column will still require a separate update for each language.

In my design each subsystem has a single text file for each language containing the text for everything - task titles, column labels, menu buttons, navigation buttons and error messages - so to change the label for a column still requires one update per language. There is no saving with your method.

I found myself wanting to change a label in a list1 transaction and I had to oppen the screen.inc file in order to change it, what if I had hundreds of screen.inc files? Should I go one by one changing the label?

In my option I should go to the column dictionary and update the label for each language, but only one update per language.

With my suggestion internationalization is straigh out of the box.

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 09:37:57 GMT
View Forum Message <> Reply to Message

Quote:I belive my suggestion is easier to use, more flexible and it separates more the presentation layer from the data layer.
I do not see that it is easier to use as it holds column labels in the database completely separate from other text.

I do not see how it is "more flexible", just "different".

As for "separating the presentation layer from the data layer" I see that you do not have a clear understanding of what this means. It is far more complicated than dealing with label text.

Let me reiterate my point of view. My current design holds foreign language translations for ALL text in a single place, one file per subsystem per language. All text, whatever it is, is extracted using the same API. How could that be "improved" by holding some of the text in the database?

My current method works very well, so I see no reason to change it.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 09:41:12 GMT
View Forum Message <> Reply to Message

Just answer this question:

I found myself wanting to change a label in a list1 transaction and I had to oppen the screen.inc file in order to change it, what if I had hundreds of screen.inc files? Should I go one by one changing the label?

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 09:49:39 GMT
View Forum Message <> Reply to Message

AJM wrote on Tue, 01 July 2008 05:37Quote:I belive my suggestion is easier to use, more flexible and it separates more the presentation layer from the data layer.
I do not see that it is easier to use as it holds column labels in the database completely separate from other text.

So why don't store all the other text in the database too?

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 09:50:19 GMT
View Forum Message <> Reply to Message

Although each *.screen.inc file contains a label for each column it is still possible to have an entry in the language_text.inc file which changes 'label-A' to 'label-B'. Thus you can make one change in your language_text.inc file which overrides every entry in multiple *.screen.inc files.

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 09:54:18 GMT
View Forum Message <> Reply to Message

Quote:So why don't store all the other text in the database too?
Beacuse this would require a separate I/O for each piece of text. With my implementation a single I/O loads ALL the text for a entire subsystem into memory. Each call to getLanguageText() then references memory without any additional I/O.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 09:56:55 GMT
View Forum Message <> Reply to Message

AJM wrote on Tue, 01 July 2008 05:54Quote:So why don't store all the other text in the database too?
Beacuse this would require a separate I/O for each piece of text. With my implementation a single I/O loads ALL the text for a entire subsystem into memory. Each call to getLanguageText() then references memory without any additional I/O.
The software I used to work with used a cache system to solve this.
I won't continue arguing (but I still belive my suggestion is better)

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 10:05:01 GMT
View Forum Message <> Reply to Message

Quote:I still belive my suggestion is better
It is different, but not better. In fact it is worse because of the following:

text is split between database and non-database files - all my text is in a single file, one file per language.
it would require a separate I/O operation to retrieve each piece of text - my method has a single

I/O operation to retrieve ALL text.
getting somebody to translate the text into a different language would require creating a new database record for each piece of text - my method has all the text for a single language in a single file.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 10:49:46 GMT
View Forum Message <> Reply to Message

AJM wrote on Tue, 01 July 2008 06:05Quote:I still belive my suggestion is better

text is split between database and non-database files - all my text is in a single file, one file per language.
Wrong. You could have everything in the database. Radicore programmers wouldn't have to deal with files. All metadata would be stored in the database.
it would require a separate I/O operation to retrieve each piece of text - my method has a single I/O operation to retrieve ALL text.
Solved with a cache system.
getting somebody to translate the text into a different language would require creating a new database record for each piece of text - my method has all the text for a single language in a single file.
Exactly. But an export and import procedure wouldn't be difficult to implement. Updates of any label would require just an update in the database

Your method: User has to deal with files. Updates introduce a ftp download, file editing and upload just to change a simple label or text. My method doesn't.

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 11:32:19 GMT
View Forum Message <> Reply to Message

Your method just affects the changing of column labels but has nothing to do with screen titles, menu buttons, navigation buttons and error messages. My method covers everything.

Besides, changing column labels is not something you do everyday. They are normally set once and remain for eternity.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 11:45:45 GMT
View Forum Message <> Reply to Message

AJM wrote on Tue, 01 July 2008 07:32Your method just affects the changing of column labels but has nothing to do with screen titles, menu buttons, navigation buttons and error messages. My method covers everything.

Besides, changing column labels is not something you do everyday. They are normally set once and remain for eternity.

No, you could use the same idea for everything. In fact, the software I used had every single string of text of the application in the database, so nothing was hard coded anywhere.
If you wnated to change any string you got into the dictionary and changed it.
You could have a table for error messages, a table for titles, for menu buttons etc. or everything in a single table.

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 11:58:08 GMT
View Forum Message <> Reply to Message

I could change my method, but I'm not going to. The cost of changing would not be outweighed by any benefits, so your suggestion is a non-starter.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 12:01:19 GMT
View Forum Message <> Reply to Message

Time will tell..

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 12:44:31 GMT
View Forum Message <> Reply to Message

Time will tell nothing. The only way to tell if your suggestion has any advantages would be to implement it and make comparisons with the current method which I am not going to do for the reasons I have already stated.

Subject: Re: Internationalisation suggestion
Posted by bonzo_bcn on Tue, 01 Jul 2008 12:47:00 GMT
View Forum Message <> Reply to Message

AJM wrote on Tue, 01 July 2008 08:44Time will tell nothing. The only way to tell if your suggestion has any advantages would be to implement it and make comparisons with the current method which I am not going to do for the reasons I have already stated.
Sorry for trying to help.

Subject: Re: Internationalisation suggestion
Posted by AJM on Tue, 01 Jul 2008 13:49:59 GMT
View Forum Message <> Reply to Message

Although I welcome suggestions I am only interested in methods which are better, not just different.