Subject: Re: Development Diary (markcarranza)
Posted by markcarranza on Tue, 01 Jan 2013 01:54:08 GMT
View Forum Message <> Reply to Message

A key realization for coding with Radicore: a task file has code like
$table_id = 'dict_column'; // table name
or
$outer_table = 'dict_table' // name of outer table
$inner_table = 'dict_column'; // name of inner table
but these variables aren't exactly names of tables.
They're class names of subclasses of the business object, Default_Table, or further subclasses, found in /[subsystem]/classes/[name of table class].class.inc
class dict_column extends Default_Table
// or
class dict_column_s01 extends dict_column
I could change the default transaction pattern code to read like:
$table_id = '#tablename#'; // name of table subclass
or
$outer_table = '#outer_table#' // name of outer table task subclass
$inner_table = '#inner_table#'; // name of inner table task subclass

I've started my own naming convention using a 'task abbreviation' like so,

task file = [table]([transaction_pattern])[task abbreviation].php
screen file = [table].[transaction_pattern].[task abbreviation].screen.inc
report file = [table].[transaction_pattern].[task abbreviation].report.inc

So I'm contemplating simply generating with new tasks, task-specific business classes,

class file = [table]_[task abbreviation].class.inc
require_once '#tablename#.class.inc';
class #tablename#_#task_abbrev# extends #tablename#and including all(?) foundation '_cm_' method stubs with terse comments and/or a comment link to documentation.

Generating a task file as,

$outer_table = '#outer_table#_#task_abbrev#' // name of outer table-task subclass
$inner_table = '#inner_table#_#task_abbrev#'; // name of inner table-task subclass

For basics, no changes to the subclass, but having an empty dbObject pattern tied to the task would, I think, more than balance the bother of extra files that basically do nothing. Especially for newbies to Radicore. Maybe it could be turned off for experts...