


## ORM in Quantum Star SE Evolved

I've gotten (over my stint offline) a few enquiries about the Data Access model used in Quantum Star SE. Not everyone has come across the concept before, so I decided to take a stab at explaining it, and why I implemented it in the fashion used by QSE (or more accurately its backend, Partholan). This is just an overview - no real details. You can explore the concept better using Google and other sources.

ORM: Object Relational Mapping. ORM is (in this context) a method for mapping Objects to Database Rows. The simplest example is a `db_user` database table with three fields: `user_name`, `user_id` and `email_address`. At this point we have data fields and data - but how do we represent/use these in a PHP application?

The simplest approach is a direct SQL call, using the PHP functions to query the database and loading the resulting data into an array. The problem is that an array is still a variable - you need to somehow inject it into functions to be operated on, then you need more functions to change, update or delete the data - not to mention create new row records. At this point the inevitable conclusion is hard coded SQL, based on data arrays which are given a GLOBAL scope for use in functions (just might be all functions allow them as parameters). This is the de-facto standard practice in the majority of PHP applications. I'm not saying it's a bad practice as such - it's certainly fast and inherently insecure only for using GLOBALS - but it's not re-useable nor easily maintained. Why? Because SQL queries may be replicated across many files - a bug in one necessitates fixes to many. There are other maintenance pitfalls with the approach - but that's not the topic of this entry!

However if you progress to using Object Oriented Programming - you learn a valuable lesson. GLOBALS are a necessary evil, but an evil nonetheless... It's at this point developers might start thinking in new directions. If we have user data, perhaps we should have User objects? Suddenly the OOP approach clicks home... 

. On a side note - I'm not a

OOP "Purist". If you are procedural minded, feel free to stay with that approach - OOP is however a very powerful approach to programming, and not half as complex as people think.

It's here that ORM comes into play. If you intend populating an object with data from a database you have several options:

1. Build SQL queries into object methods (internal handling)
2. Pass data arrays into the object at construction or with a separate `import()` style method (external handling)
3. Give the object responsibility for managing data (hands-off handling)

The first two, again, are common. They are most intuitive when you have limited exposure to OOP (and even then it often just makes sense). The third has a bad reputation for being needlessly complex and over designed - untrue if all you look at are Propel style libraries in the context of simple web applications. Now all are just as valid as the other - but the third neatly sidesteps using manually coded SQL. It also "boils down" the functionality into a standard, predictable set of methods. A classic example common to nearly all ORM solutions.

We have a user (#3) for whom we need to reset an email address. Using the ORM model one could (in their PHP action) code this using:

```
$user = new User();  
$user->getByPk(3);  
$user->setEmailAddress('newaddress@example.com');  
$user->save();
```

Our User object has suddenly been transformed into a powerful unit for managing data. No SQL, no acrobatics, no blocks of code calling database functions (or database abstraction layers), no error management - nothing but 4 lines of code that do it all for you.

Now if that sounds magically fantastic I suggest reading up on ORM elsewhere (sorry, but I'm a truly bad technical writer and I'll just ramble across the topic without digging into minutiae). All this does come with a cost - which differs depending on the ORM approach taken. Since we rely on methods generated standardised SQL in place of custom optimised SQL, there is a performance hit. No JOINS, UNIONS, etc have that effect. So does not being able to limit the range of data from a row that is requested - this all can be solved by the way, but my home brewed ORM was built for simplicity, not speed. See ORM solutions like Propel for something far more complete (and therefore complex).

I think the above serves as a very non-detailed intro to the workings of ORM - if not a full technical debate over its merits. Now on to the implementation of the version in QSE.

To implement ORM you need a few building blocks as basics:

1. Database Table details and metainformation
2. An SQL Generator (and optionally an Optimiser - never added one)
3. A database abstraction layer (to enable rapid switching between database types)

Many fast and easy implementations of a dead simple ORM class like ActiveRecord omit all but number 2 - which is usually built in. This is a direction that personally I view as dangerous. One ActiveRecord implementation I saw did the following (most follow it closely):

1. Request Table details from MySQL
2. Generate a domain class from this metainfo specific to the Table structure
3. Instantiate the generated class to an object

That makes sense (it works for Ruby on Rails afterall




) but it imposes a cost - all that extra querying and class

generation is done at runtime, for every request a server handles. Now personally I prefer to take the pre-generate option. If my database tables are going to remain static (I don't see many PHP applications changing table structure every few minutes) I will pre-generate all those domain objects and store them for use in the application before its released/published as Transfer Objects. Why not skip all three of those costly steps?

This still does not tell anyone where the SQL comes from. Since I aimed for simplicity, I designed a small class which generates SQL depending on the structure of the table - most of the SQL follow the simplest route for CRUD operations (Create, Read, Update, Delete). This Data Access object in combination with a Transfer Object enables a very simple ORM layer without the frills and thrills of something like Propel, but without the performance hit of RoR's ActiveRecord. Is it feature full? No. Is it simple? Yes. Objectives met.

The last piece of the puzzle is using ADOdb-Lite as a database abstraction layer in place of where PHP's native library functions (mysql\_query() and such) would be. ADOdb-Lite simplifies other aspects of SQL generation a great deal - making even the all powerful Data Access class incredibly simple.


Basic operation: The user instantiates the selected Transfer Object (e.g. User). This object includes a set of CRUD methods which delegate SQL query operations to the Data Access object. The DAO generates the required SQL, and executes via the ADOdb-Lite database abstraction library. The sequence of method calls allows the return of any resulting data, usually a boolean false on failure, or some other value (an array for row data for example) back to the parent Transfer Object which stores such data internally. The stored data is accessible using a set of getters/setters which are based in individual names (e.g. setUsername() maps to the user\_name field).

There ends my overview. Maybe it sheds a little light of what I was up to in QS, then again maybe not. If anyone needs further clarification, well, you all know my email address from the forums. 


Posted by Pádraic Brady at 04:22

Wednesday, March 29. 2006

## Template-Lite...here we go again...

Just to note [Panama Jack released Template-Lite](#) to Sourceforge. Template-Lite does for Smarty (oops, am I breaching Trademark by mentioning it? 

) what ADOdb-Lite did for ADOdb. That is, faster executions and a smaller memory footprint. For those intent on creating small lean applications expected to see some stress on shared hosts than PJ offers another helping hand.

Of course that's not the only condition for using Template-Lite... Let's not sideline libraries into niches. I look forward to reading the code and seeing if there's a `assign_by_ref()` object method (hint..) which is useful for passing large arrays and such into a template engine rather than by value. If not, well, I am free to edit it 

There was also another ADOdb-Lite release - see [the blog entry](#).

Posted by Pádraic Brady in PHP General at 06:47

Thursday, March 23. 2006

## ADODB-Lite PerfMon: Basics Completed

Back on the quest to add the ADODB Performance Management features to ADODB-Lite...

Spent some more time on the job this evening. I added a few adjustments and the Log fetching now works for both MySQL4 and MySQL5. I also removed a few optional (IMO) methods from the methods in tohtml.inc.php - namely some date-time functions which were of limited use in any case. The impact to the usual stats is negligible.

The features now work as in ADODB for MySQL4, MySQL5 and PostgreSQL. However I still have a bit of tidying up to do. Primarily I need to vut all the duplicate code from the individual driver modules and push them up into a common parent. I've started on this and it looks workable - if with a little acrobatics. I have also pushed the perfmon specific `$_logsql` property and `LogSQL()` method into the common module parent - and out of `adodb.inc.php`. Finally the `NewPerfMonitor()` function in `adodb.inc.php` is now a simple three liner to return a reference to the connection object it is passed - this is purely for backwards compatibility. The function is not required to use perfmon anymore but ADODB users will still have it in their code.

Its coming together quite nicely at this point. There is still a lot of non-standard HTML hanging around, including naughty font tag usage. Not sure whether to just update to some XHTML 1.0 standard and leave it, or set up a small collection of template files (in native PHP) to allow those template be editable in the future. It just makes things look messy within the module classes. I have also excised all the gymnastics surrounding `$_logsql` and `LogSQL()` calls (they are used basically as alternatives and are just plain confusing) and made `LogSQL()` the standard method of switching sql logging on and off to avoid logging queries sourced in the perfmon modules.

Once I get this all sorted out I'll add further driver modules for those DBMS's supported at present in ADODB-Lite already available in ADODB. It's a small extra that I may as well do rather than take half measures. In all there's maybe another few hours work tops. Hopefully I can make a release at the weekend for anyone interested in testing and offering some feedback for improvements.

My two current tasks:

- Moving everything to using a common module parent
- Doing something about the messy page creation via `tohtml.inc.php` and the PHP/HTML mixing


Posted by Pádraic Brady in PHP General at 12:03

## Upcoming - Elder Scrolls IV: Oblivion

There are a few constants in my life online.


- 1) I ramble about topics at random
- 2) I only use an IM if bored crazy
- 3) I vanish for weeks when a decent RPG or strategy game is released

No. 3 is going to happen this coming Friday (in Europe) when Elder Scrolls IV: Oblivion hits shop shelves across Europe. Much in the vein of addictive games like Civilisation IV and Rome: Total War, there will be a few weeks where countless people will drop offline, lose contact with close friends and relatives, and lock doors in case the significant other in their lives tries to interrupt them.

The problem with games like the Elder Scrolls series is the replay value and depth of gameplay. Morrowind has been a constant presence on my hard drive, alongside Rome TW, since 2002. Even playing last time in January there was still lots to do and explore. It was wearing thin I'll admit - only so much questing you can do over a 3-4 year period even with some quality mods added - but the game is so huge that new encounters are inevitable if you go looking hard enough 

Rome TW has something similar. No matter how many times you play, there are always moments of satisfaction to be found. The best one yet I've had was a huge battle with my fellow Brutii outnumbered and eventually routed by a horde of Gauls. Rallying up my troops over the course of 20 minutes, I sent them all over the battle map avoiding enemy formations until some semblance of order was restored. Atop a deserted hill, I sent out small units of cavalry to keep the Gauls off balance. That done I used expended Archers as shock troops (excellent method of stalling an enemy formation's charge) before pounding them from the sides with heavy infantry. After almost 30 minutes of near routs, after the near defeat of my army, I'm victorious. If you haven't played a TW game before - you don't deserve to live. Even when it seems all is lost some creative thinking can switch the course of events.

Anyways, Oblivion is looking like a fantastic game. If the world is as detailed as Fry Cry's it will be a sight to behold - and who didn't sit for a while with their mouths open after seeing that island paradise the first time? Combat and Magic have been improved - and the addition of physics will add a certain feeling of satisfaction when I get to smashing foes around the place with fireballs. NPC's have seen improvement in daily activity schedules, lip synching, real voice conversation, etc. I doubt the hyped realism is as evident as publicised - and early reviews seem to bear out the level of repetition I expected. More important Magic, Archery and Stealth have all been revamped and balance out melee combat nicely by all accounts. There really is nothing more irritating than being forced into melee combat because magic is too weak or requires range to work well.

A few well hyped quirks are interesting in themselves. You can now buy (more likely steal ) horses. And yes, in case you've hidden under a rock for the last year, you can ride them. The world is now something along the lines of 16 square miles of forest, plains, mountains and urban areas. Towns are better guarded - town watch are no longer random patrolling loners, they congregate in groups and will hunt down criminals, i.e. your character, pretty efficiently. I could continue with a long list.

Suffice it to say Oblivion is looking to be a long way ahead of Morrowind. They've kept all the best points of Morrowind, and revamped the more repetitive/boring bits where players had little interaction before - such as the Melee/Magic use.

Since this is one of those games I'm a huge fan of I'm going to spend a little extra to supplement my PC's capacity and buy a Creative XFire Sound Card at the weekend. My aging Audigy 2 is probably starting to use a fair number of my processor cycles on more recent games, and this will offer better sound (esp. for Headphone use when emulating 5.1 surround) and ensure the processor (which I don't want to replace for at least another 6 months) is as free as possible to handle the game. Another piece of good news is that NVidia are releasing an updated ForceWare driver specifically addressing performance in Oblivion later today which will hopefully keep my 6800 GT up to scratch.

Another 3 days until I get around to scorching Skeletons with a maniacal grin plastered across my face...

Posted by PÃ;draic Brady in PC Gaming at 07:19

Wednesday, March 22, 2006

### **I'm not sure what I should entitle this...**

What's worse than a Monday at work? A Tuesday at work with nothing to do, and no real activity to be found worthwhile. So being bored out of my skull, I've caught up on my reading list. After months of tracking Ruby progress I've come to the conclusion - perhaps belatedly - that although its a very cool general purpose language it's not really all that comparable to PHP.

But seriously, besides Ruby on Rails (duplicate a dozen times in PHP by now with limited success), ActiveRecord (duplicated...yada, yada, yada - I dislike metaprogramming of that nature), and a nifty pure OOP syntax it doesn't really offer anything truly awe inspiring. If you're a purist, with strict OOP-Forever ideals go have fun - but PHP just works better in the real world whether its familiarity, wasting effort on another language to duplicate what PHP already does well, hosting preferences, or some other reason why PHP is preferable. Maybe in a few years with some maturity under its belt, and more widespread adoption...

Important to note I don't bash it's non-web usage. It's a very nice general purpose language off the web app area. I have

Anyways, I ramble...as usual.

Just saw [lamsure's small mention of my doodling in adding Performance Management to ADOdb-Lite](#).

So, spurred on by news at Maugrims Blog, I'm becoming excited at the prospect of using adodb-lite for a few projects (Next forum possibly included). Some of the phrasing he's using in his latest posts has me nervous around just how drop-in it will be (shades of prior discussions about adodb-lite come to mind), but I'm more than happy to wait and try it out once he's had a chance to hack on it.

Being drop-in, as in "strictly no changes" is not possible. One would need to reference the module to load in an ADONewConnection() parameter. Apart from that though its intended that's as far as the changes go interface wise. There are a few areas where maintaining backwards compatibility with prior use have been added. I've retained a tiny version of the adodb\_perf class for the static adodb\_perf::table() calls. ADOPerfMonitor() function will probably stay - although all it will do is return a reference to the ADONewConnection object which would include loaded modules, and therefore perfmon. This will keep prior \$perf->UI() etc. use intact. Not sure there's too many (if any) other BC issues - could be wrong, I haven't focused on it entirely yet.

I suppose one changeable area is deciding which modules to load for any given request. Since modules are loaded via a ADONewConnection() parameter it need to be setup before creating a database connection - not later on when perfmon is needed. This introduces some need to be able to set up a module list to load, before creating a connection, specific to a request or PHP file. I think that's the biggest block to using ADOdb as a "drop in" replacement for ADOdb - but then again it's a large reason why ADOdb-Lite works so well. It only loads what you tell it to. I've never seen it as a problem before because an OOP method for creating connections is easily adapted - might be more difficult to perform selective module loading in a procedural app where the connection code is static - not capable of being dynamically changed by individual requests.

On to other things in this general purpose post -

Sourceforge updated its Software Map design moving to its new layout for these pages. They've added some very nifty filter functionality which beats the prior incarnation hands down. It made finding an open source

stock portfolio manager simple at least - even found one in Ruby which looks very cool if needing a more dynamic method of defining a portfolio than editing a text pf file - still it's new and it's beta and it works. Might try it out for a few days - my needs are pretty simple. Thumbs up to the improvement.


Back on something else I decided to have a go at installing Blacknova Traders. People keep wondering whether I'll ever actually host any form of game on my server - so while QS is in development I'm thinking of offering a few open source space strats in rotation. Maybe a BNT game for a month or so, then AATraders, then on to Quasar. I won't be hosting Solar Empire or QS2 - I can use each for so many XSS and CSRF exploits its almost frightening. The install went well, though I couldn't locate the default password for my account. Using the password retrieval function it ended up being a partial string of my email address...hmmm. Maybe I missed something in the configuration to create a unique random password - never actually edited the config files other than db\_config.php. I'll put the game live in a few days perhaps and do a little exploring - been three years since I last played BNT with Genesis.

Whoa - this is getting long. Boredom breeds verbosity... Until next ramble!

Posted by PÃ;draic Brady at 07:06

### **What happened to the Google ads?**

Folk probably (or not) remember the day when there were two squat text blocks from AdSense on every page hosted by quantum-star.com . I removed them back in October I think when redesigning the public web offering to get rid of that phpNuke horror. Just to warn people who have a habit of stating an ad-free website is cool - it's not so cool when you are needlessly forgoing a little income to pay for the hosting.

So the ads will be making a return. As before they'll all be fairly small Google AdSense types - all text and no images. You don't have to click them, you can probably I assume block them, and feel free to do so 

. I'll just work towards getting at least one cheque in a year to pay for a few month's hosting. The ads will not however appear on the blog - it's my personal space, so not interested in giving space to a bunch of ads here.

In anticipation of ruining everyone's browsing with ads ;), I checked my own account for the first time since May 2005. Really, before now I saw no point. I was a bit surprised to see the number \$80.36 appear. It was \$35 or something last Spring before I took those months off for study. That covers (or will) about 4 months hosting costs - or at least compensate me for 4 prepaid months.

Anyway, enough of my rambling...

Posted by PÃ;draic Brady in Irishisms at 02:09

Tuesday, March 21. 2006


## More on ADOdb-Lite and Perfmon...

Mondays, there is nothing worse than a Monday back at work after a busy tiring weekend.

In between St. Paddy's Day celebrations, the Rugby celebrations (Triple Crown for us!), and Sunday's cleanup of my Study (the small dark place packed with three PCs and boxes of books), I managed to make some additional headway on adding Performance Management to ADOdb-Lite.

The main change since my last post is that I've switched to using the Module system in ADOdb-Lite. I thought the classes were fairly non-dependent (almost exclusive) for perfmon but turns out that's not the case when using ADOdb-Lite. The perfmon module requires a selection of methods not supplied by default in ADOdb-Lite - nor in current modules, e.g. MetaType() in ADOResultSet, ServerInfo() in ADOConnection, so it's back to a module structure. Which is fine if a bit different - it means the perfmon methods are now attached to the ADOConnection object (which is extended from by modules). Not sure yet how this impacts BC usage of NewPerfMonitor() since it makes it defunct (the connection will load the relevant module as an ADOConnection child). That is, there is no separate adodb\_perf object as with ADOdb.


Other noted issues still remain (even after downloading the latest version):

- There's no mysqli support for perfmon in ADOdb
- htmlspecialchars() still par for the course
- Still the mysqli "native" dataProvider value
- Lots of confusing setFetchMode() calls and use of global \$ADODB vars. All this controls fetch modes. It's just mind bogglingly complex. I'm just cutting out these and waiting for a use case where its lack breaks something - It's better to live in ignorance than be driven insane 

After the above I have a polling monitor running for PostgreSQL 8.1 and MySQL5 - haven't tested earlier versions. Pulling out the Log records for SQL queries however has proven more difficult.

Other issues:

- Mysql code (MySQLi is a mirror image), one query in particular, will not work for mysqli with MySQL 5. Needs to be fixed and a check added if this is not a problem using 4.1+ - possibly a check using the MySQL version() function. This causes the SuspiciousSQL() method to fail - same happens for ExpensiveSQL() etc.
- Still lots of messy bits from mixing PHP and HTML.
- The modules specific to each driver contain a lot of duplicated code at the moment. Would be nice to parcel this separately so as to provide default fallback functions. Not sure how to go about this within the constraints of the module system other than to add a parent class somewhere to extend the ADOConnection object, before the specific perfmon class child classes.
- No support for PDO (par for the course in ADOdb-Lite anyway). This is needed since I work on a project using PDO exclusively (internal business app).
- Table switching (setting a custom table name) requires a static adodb\_perf class - just to hold the static adodb\_perf::table() method. I've stuck this into the adodb-perf.inc.php file with the adodb\_\*() functions used in ADOdb. Providing it for BC and nothing more - might even remove it if it's a minor upgrade issue. How many such calls could be used in an app?

Where things stand a lot of progress was made - despite the whining 

. Mainly the last few things are personal takes, and areas which ADOdb lacks to start with. Perfmon in ADOdb is currently useless when using mysqli or PDO. My guess is the code would work just fine with mysql but I'm not interested in leaving it there and honestly haven't bothered testing it on MySQL4 yet.

Posted by Pádraic Brady in PHP General at 11:10

Monday, March 20, 2006

## Spammers and Bots

For the last week the server has been under siege by the spammers. Before I activated the automatic reject setting, my email was flooded with 1500 or so moderation requests. Chances are anyone trying to comment over the weekend or earlier met the emergency comment shutdown mode in Serendipity for those occasions when the spammers are in overdrive. To ease off the restrictions I've bumped up the required delay between comments, added some additional search terms, and removed the emergency shutdown option - it's just preventing comments from legitimate users at this stage.

Sort of while doing this I also took a look at another highlighted issue - bots. The number and frequency of bots crawling the blog and forums has increased a great deal since the start of the year. Normally I don't mind search bots, but a few of these are hitting again, and again, and again. Now checking in is normal - but what's all the multiple daily hits about? How many times is the blog being downloaded a day at the expense of sucking up my personal paid for bandwidth?


Posted by Pádraic Brady at 22:56

Sunday, March 19. 2006

## Performance Monitoring for ADOdb-Lite?

A while back I said I was thinking vaguely about adding performance monitoring into ADOdb-Lite. The news is that I started doing so today. The reasoning is fairly obvious - ADOdb's perfmon library is a very useful tool even if it is underused, and I want to use it. Since nagging someone else about a feature I want is pointless (its open source!), I'm adding it myself - and hopefully well enough to see it in a release one day.

Day 1 was a two hour session taken up primarily in reading and understanding the ADOdb source code - man, it's like wading through a bomb site at times. adodb.inc.php is huge! Anyways, I noted a few things and, just to make it clear, I was reading a July 2005 version (so anything noted by me could well have been changed since). Anyways, main points on Perfmon in ADOdb (as of July '05):


- Perfmon is not workable with mysqli driver; there's a small possible bug where the driver class supplies the \$dataProvider property with the value "native". Perfmon deliberately blocks this value - and the mysql driver uses the "mysql" value. So at the expense of some unknown issue and assuming what works for mysql works for mysqli, changing this to "mysqli" from "native" works fine.
- ADOdb output is escaped via htmlspecialchars(). Same as for ADOdb-Lite by default. Personally I prefer htmlentities() with ENT\_QUOTES. Setting the character encoding is usually advisable to prevent charenc misinterpretations. Nag, nag, nag... 
- Output and PHP is mixed up. Never liked that, never will.
- Code uses the long names for superglobals - I don't use old PHP versions, and I have the PHP6 CVS installed, so I'm changing that one for better or worse...

So after the reading I looked at what I needed to do. Given Perfmon is fairly standalone the simplest path is just to dump the code into ADOdb-Lite with minimal changes as follows:

- add a new property, \$\_logsql, to ADOConnection
- add a new method, LogSQL(), to ADOConnection
- stick the driver specific subclasses of the main perfmon class into the driver directory for each DBMS
- add a NewPerfMonitor() function to adodb.inc.php

Done, dust, test and finish... The approach isn't part of the module system - like DataDictionary perfmon is called outside the connection object so that route seems to be of little real value. NewPerfMonitor() does all the legwork, and its use is pre-existing already if you use ADODB in code, so BC and all that prevailing...

I did a lot of the above in the second hour, tidied up and modified the source where necessary to make it ADOdb-Lite compatible, and ran a few tests. It still won't work, but that's for the moment because the \$\_logsql property acts as a switch which is, well, switched, on every query (to avoid logging the sql query FOR logging). It's one of those confusing "track the value" puzzles I'm not up to getting my brain around at 11pm after having a few drinks earlier (rugby match - Ireland won over at Twickenham beating England for the Triple Crown - now if only Wales won...).

Tomorrow I'll see about finishing it off, maybe tidying up output, and sticking a version online somewhere for someone else to look at and tell me what a bad lazy programmer I am... 

. Anyways, perfmon for ADOdb-Lite, soon to be checked

off my todo list.


Posted by Pdraic Brady in PHP General at 13:16

Friday, March 17, 2006

## Text/Browser Games on Wikipedia - proposed articles

I remember a few months back doing an idle tour of Wikipedia for text and browser based games. The content was slim, sometimes non-existent, and often disputed. A few months later and little has changed. A few people, Moocats among them, has added bits and pieces which are growing slowly.

Particularly evident is that few open source games in the strain of using server-side source code are evident. I think LOGD perhaps is the only such with an article present - seems a bit thin on the ground really. Related to BNT/AAT the closest seems to be the TradeWars article.

Anyways, I'm thinking of throwing my hat in the ring and attempting to write an objective article or two to fill in the obvious gaps. Not sure how that'll be received but the hope is to shed some light on these subjects and flesh out any pre-existing articles. If Moocats is reading (and I know you do on occasion 

) maybe you can contact me and we'll collaborate on this - I'd prefer to do so rather than edge a few edits here and there to avoid over-writing your contribution (which first caught my attention). Anyone else interested feel free to chip in - no forum but comments are there for now. My email address is .

The problems as I see them in the articles to date are fairly simple. The information given, though informative, is too small. Browser Games or Text Games are a broad subject with a lot of details regarding implementations, requirements, etc. If these were addressed the articles would be pretty good, and open up the field to subsequent discussions on additional contributions. As it stands though its a summary without depth.

I'll wait a while and see what response this blog gets, I know its not exactly viewable by everyone in browser based game related hobbies/jobs so I added to the current discussion on Wikipedia also.

Links:

[http://en.wikipedia.org/wiki/Browser\\_game](http://en.wikipedia.org/wiki/Browser_game)

Possible articles:

Browser based games

Text based games

PHP games - links over to Browser based games and Text based games as related

Posted by PÃdraic Brady at 07:31

## Official Website goes live this weekend!

I finally did the unthinkable and proofread the entire text being used in the website for QSE - the one currently online for people to stare at during testing: <http://www.quantum-star.com/devsite2/>. I'll upload the new correctly spelled and gramtically corrected version at the weekend and place at the URI root.


This is still (and always will be) open country. If you believe more information should be added then drop us a note on the forums. At present it basically a set of minimal pages designed to sketch out what the project is all about, how to contact us, how to help us out, etc. Items like a FAQ, manual etc. will get added eventually.

Of particular relevance are the Bug Reporting Guidelines which have proven highly successful so far. I know those following Generations are waiting for another release, but hopefully Ikoda will be available at some point in the near future to make a release. QSE hasn't seen any major bugs - but then it's not even playable yet, nor close to for the moment...

Posted by Pdraic Brady in Quantum Star SE at 02:03


## QS User Suggestions

Since I won't be online tomorrow... Hope everyone has a good St. Patrick's Day, and for those with the day off from work, enjoy the parades!

I won't go into huge detail on current QS progress. I've not had much chance to do anything on the project this week, so at the weekend I'll spend a few hours on it. I have most of the game support ready to go, so all I really want to do for next release is have a location page finished, and fleet navigation enabled. Maybe an Admin Panel section for universe generation. That would make the first release where you can explore the universe and actually do something 


I also need to get that translation code pulled together and tested - see if this system is workable or an unrealistic pipedream. No detailed planning for this one so its off to unit testing to build it from the ground up.

Back to topic.

I made a new topic on the QS forums regarding what players/users want in QS. It's one of those sticky threads that are meant as permanent fixtures. So far there's been a good amount of contributions, and some ideas being swirled around. Many at the moment have been covered by QSE planning - its sad to see the most mentioned are truly bad areas of QS2 in terms of gameplay. Even QSE in its basic form would be a significant improvement over QS2 I think... Anyway, one of those perpetual followers of all things QS (and BNT/AAT more likely 

), Kitty, made some good points. One thing she noted was planets and how they're done in SE style games. QS2 did a little tinkering to limit planet counts, but it was never ideal (and it's buggy in the current QSG legacy release unfortunately). BNT of course uses the star type to determine number of planets allowable - most logical system.

The QSE aim, and I've thought about a bit more, should attempt to force a diminishing return in benefits from creating more and more planets in the same star system. Rather than having an upper limit of 5 planets where those planets will all be identical in type, instead do something more along the lines of reducing the size/limits of each extra planet created (noting there would still be an upper limit of about 4-5 planets).

So a 5 planet limit would give rise to a normal "large" planet, a medium, and three smalls - or maybe just use a fixed scale where the limits for subsequent planets are always 15-20% less than the last preceding planet. This would encourage players to move outwards since a star system would only support one large planet - the rest being smaller. This of course has a neat side effect of placing a lot of value on the number of systems controlled by a player - something completely missing from QS2. What's a strategy game without some need for grabbing territory? 

A few other posters have noted a need for more ships (and more variety within them), along with similar suggestions. The point to make is that Ships technically are nothing more than a virtual container of attributes in QS Evolved. They basically act as a collection of "Components" each of which confers an Attribute value upon the ship. So, for example, you could have a ship comprised of Cargo Bays +1000, Mining Laser +5, and Thrusters +10. Result is a ship with attributes of:

Cargo Bays: 1000  
Mining Rate: 5  
Manouver: 10

These attributes would be used within certain calculations and process effecting that ship, as well as specifying any ship restrictions. Since they're all component based, and attributes are simple serialised arrays, attributes can be added to the current list with ease - another factor to consider in creating future modifications and plugins.

Posted by PÃ;draic Brady in Quantum Star SE at 01:49

Saturday, March 11. 2006

## The next major QS piece of work

Well, after my outage for a few days I'm moving on to do some work on a new section for QS - namely those translations - its time I had a final answer for all those people who keep dropping by requesting info on, or even offering, translation services. The general idea is to pre-parse (read: code generation) as many language specific sets of templates as possible. As translation on the fly makes me a bit nervous (doing it every request seems something one should be able to avoid - well to some degree), so the aim is to borrow a little functionality from ezComponents, store translations within XML files, and for every release (or by command of the local Developer permissable user) generate translated versions of the English templates in one go, the aim of course being to cache them for later use.

That will still leave the database specific text for translation, but most of this are limited to some English descriptions of the content so on-the-fly translating for these is less painful - hopefully. Anyways the current task is the template translations only, that's where the larger performance hit is since it effects everything - not just a limited array of values.

Using ezComponents will allow a shortcut in terms of coding my own system, and in either case its performance is negligible - it's done once (usually before a release) not on every page request. If I can get it finished I'll add it to the next release on March 20-25.


Posted by Pádraic Brady at 07:38

Friday, March 10. 2006

## On Subversion and why my phone company sucks

The migration to Subversion is coming. I'm going to export/import my local repository which has been witnessing most of my work on QSE for the last month. I'm up to revision 80 (made a small front controller patch yesterday - sad I know, but this is what happens when you're sick) and I want the entire history (not just the files) imported to svn. When I do it is another question - my internet connection - lowly 56k piece of  $\text{Ä,Ä£\$}\text{Ä,Ä£}$  - is back up and operating but the thought of transferring something of 1MB or higher is depressing. Dial up was never designed for speed, especially when the physical line is connected across a few dozen line splitters to the local exchange.

In a fit of monopolistic distaste the Irish phone company (Eircom - not actually Irish owned anymore) has been criticised by the OECD (again - its an annual event) for having abnormally high charges. Apparently Eircom has also been reported for false advertising (again - another annual event) and chances are they'll get a wrist slapping and a few thousand Euros in fines (again, blah, blah, blah).

I love monopolies. Honest. Why look for improvement when your local monopoly will spin tall tales to ease the rumblings of the local natives over OECD reports and failures to fix your bloody phone line in less than a week and a half? 

Anyways, the word on the streets of Ireland is that we pay around 10 Euros more than the UK on an average phone bill. That's more like 25 for mine. For what I don't know. It's not like they're improving the phone lines except around large towns and the usual Dublin/Cork/Galway trio - probably under coercion by the government. I haven't seen any improvement - lots of degradation - but definitely no improvements. Still its fodder for Irish blogs such as the dedicated (if slightly maniac) [ComWreck](#) which still does a fair job of pointing out the fault lies as much with toothless regulators and ill-advised government ministers who may or may not be slightly incompetent.

Take Eircom's answer to a damaged line a week back (personal experience follows...). They fixed it - all they needed was a penknife, black insulating tape, and some careful handy work to twist the cut wires back together. Dandy - it's fixed, Sir. Thank you for using Eircom. What's that, Sir? The phone cable is sitting on your hedge, Sir? Why, so it is, Sir! Thank you for using... Eh, I don't know about that, Sir. 'Tis certainly odd there being all them pole's for holdin' lines, Sir. Ain't no lines on 'em, ya see!

Black tape? Penknives? Wow, the engineering and electrical skills in Ireland must be experiencing a boom. Sons of  $\text{Ä,Ä£}\text{Ä,Ä£}\text{Ä,Ä£}\text{Ä,Ä£}$ , I could have done that myself in 5 minutes. I wouldn't have needed a penknife either. Bleh! Now I can have fun waiting for a rainy day, the return of the static, the probability of a slower internet connection due to interference...another phone call (from my mobile) telling the operator (not their fault) that the company they work for better fix my line properly or I'm going to take myself down to the Small Claims Court with my 50 Euros and make life a little more inconvenient for them.

Adding insult to injury the line has been sitting off the friggin' telephone posts and sitting on my border hedge for over a year. Which is why it got cut - its a local law that hedges bordering a rural road be trimmed during February/March! BAHHHH.

I think I left the sedatives over on the coffee table...excuse me a moment...


Posted by PÄ;draic Brady at 04:41


Thursday, March 9, 2006

## He lives!

Wow, it's the 9th of March. Here I sit after spending 4 days being subjected to the horror Man calls the flu. Yes, 4 days of tiredness, getting sick, hacking like I just chain smoked my way through a few packs of fags, etc. Horrible.

And no, I still don't feel great.

Anyways. 9 days to St. Paddy's day. Maybe this year will break the world of calling it St. Patty's Day. I mean folk do realise Patty is a girl's name right? No offence to the females, but someone needs to keep up with Irish name conventions... 

And wouldn't you know I'd miss at least one PHP somethign happening online? I missed the release on March 4th of the Zend Framework - hurrah, it really does exist! 

More interesting for me, John Lim has posted an ActiveRecord implementation using ADOdb. I'm still fiddling with the internals of Partholan's DataAccess Object using ADOdb-Lite from time to time. Maybe I can get some extra inspiration...

What else is weird and wonderful in PHP for the last week I missed? Iamsure's blog is apparently MIA, people are still testing QS 0.19pa, the Zend Framework may or may not be GPL compatible (a lot of debate about the Zend Framework License), my blog has been targeted by the spam bots more than is usual. The Spam is odd. Generally I get an average 10 pieces a day, over the last 5 days I've received an average of 225 junk emails a day. They are nearly all comment emails from this blog - so it appears my blog is now being targeted by the evil spammers. It's probably time to move from moderation to auto-delete settings. None is getting through - its the usual poker/drugs crap spammers love to send to Serendipity blogs...

More amusing is the newest IE7 site - <http://www.ie7.com>. The folk over at Microsoft should have seen this one coming. I wonder who the FF fan is?

Posted by Pádraic Brady in Irishisms at 23:57

Friday, March 3. 2006


### **FUD: Fear, uncertainty, doubt...**

<http://slashdot.org/article.pl?sid=06/03/02/0018201>

Microsoft will introduce a search engine better than Google in six months in the United States and Britain followed by Europe, its European president said on Wednesday.

"What we're saying is that in six months' time we'll be more relevant in the U.S. market place than Google," said Neil Holloway, Microsoft president for Europe, Middle East and Africa.


"The quality of our search and the relevance of our search from a solution perspective to the consumer will be more relevant," he told the Reuters Global Technology, Media and Telecoms Summit.

Its kind of weird, when I first hit the net I was using Yahoo! all the time. Then people started moaning about Google and how great it was. I remember wondering if the the cheap little page was a joke until I got hooked 

In essence I've never used MSN except by default (that annoying habit of IE to redirect when unable to find a URL or on a misspelling).

I also spotted the cheap shot in the original article:

"Should we add a Google-like search engine but twice as good hard-core into Windows? Guess what. If we did that, I don't think a company called Google would be very happy," he said.

Google wouldn't be happy? heh 

The EU would probably have apoplexy and sue them a second time under the competition regulations. It's bad enough MS has been taking beating after beating for consistently appealing decisions and sending out spotty illegible documentation. Hard coding MSN Search into Vista would be incredibly stupid. They already got a little shock after Hotmail started its past "You're not using Internet Explorer!" warnings when using Firefox and Opera, they really want another one from European users?

Posted by Pádraic Brady at 07:30

### **QS Evolved Braindump**

Its been a long time since I've bothered with an over of the Quantum Star SE universe. I doubt even long time QS players even remember any reference to it. But here goes a quick synopsis.


Quantum Star SE is set in a future timeline, roughly 250 years in the future. It's a galaxy where various alien species are recovering from an attack, 250 years ago, by a mysterious alien species commonly referred to as the "Grey Ones" who launched a campaign of genocide against all sentient species. Humanity barely survived the assault rebuilding their civilisation along conservative lines under the control of the Sol Authority. Over time they've grown a bit of a superiority streak (along with the rest of the surviving Galaxy) because they survived the rash of attacks and engineered plagues designed to eradicate them.

Along with the Humans, several other races have reached a stage of long distance space exploration. They're all pretty high on the idea that their civilisations are the superior ones - everyone else are just non-entities. As Humans start the

first round of colonisation, the other species quickly follow suit. The result of course is chaos.

Now being a story based fellow, a lot of game features will act to support that general worldview. There will be Races available in QSE, though probably optional since traditional QS was a purely Human affair. Still, it will be a large set of features that are well supported. And of course adding new Races will be as simple as adding new Resources - just add a database entry and have fun.

This also allows a little random streak in the game. Who are these Grey Ones? Have they left odd stuff around the Galaxy waiting for lucky players to find - you get the idea. A little surprise and luck never hurt a game.

Since we have races, it also opens the possibility of Factions, something I wrote about before on the Solar Empire forums that was in planning. Factions and Races would form the basis of a matrix measure of faction alignment. You can be aligned by race, but also by faction. This would also hinge a few feature like alignment specific facilities and services. Afterall you can't expect to access a Blackmarket unless the Pirates think you're not one of those pansy Sol Authority sidekicks 

Both Races and Factions would eventually allow some minor bonus attributes. These would get added (or deducted) from your ship attributes. Their overall effect would probably be limited to 5% of the maximum attribute ranking possible, valuable when starting a game, but much less so as the game progresses (higher your attribute grows, the less a bonus contributes overall).

If you follow that trail of thought it suggests enabling professions as a Faction property. And also suggests a Faction alignment modifier based on in-game actions. For example you could start as a Level 1 Pirate, run around blackmarkets and drop raiding player ships on the side. After a while you could go legit, buy off your bounty, start trading and wind up a Level 1 Pirate (0), Level 4 Merchant (+60). In English, a neutral aligned Pirate, and a positive aligned Merchant. Of course because Factions oppose this could translate to alignment Pirate -60, Merchant +60 - i.e. the Pirates have decided they don't like you since you're a Merchant, their traditional prey.

Maybe this little braindump will offer a preview of things to come and give people an insight to how I approach QSE's game design.

As a sort of other braindump, I recently wrote out an introduction piece for one of the possible new Races. I'll only be adding 2 or 3 for development work but here's the intro for the insectoid Kalykk:

Listen, Sons and Daughters of Kalykk. It is I, your Emperor who speaks this day. Far from here our enemies lay in wait. The Humans expand outward from their blue-green world daring to set themselves against us. They are dangerous these Humans, fanatics who talk of wisdom and justice as they slaughter our brethren in their thousands!

And yet I have stayed the hand of true justice. For in my wisdom I have seen these pink-skinned monsters fail and fall back a defeated and lost race in due course. It is inevitable. We will have our vistory over these jumped up mammals! Why they only have two legs, the pitiful creatures!

But the Humans are but one such species! Remember the Dark Days! When the Grey Ones came to us masquerading as allies and friends. "Let us share our technology," they said. "Let us be peaceful friends," they said. Fools! We should have destroyed them and they diskships, and buried their foul flesh in lava holes to rot. But we did not, for we were young and sought peace - the arrogance of youthful idealism.

We are no longer weak! From the survivors of the Grey One's final betrayal we have become a superior species, capable of feats beyond the imagination of the pink-skins! We will confound them as they seek to steal our rightful territory, terrify them in battle, and set them fleeing back to their ugly world with their furry mammal tails between their

legs!

But let us not forget the others! The Grey Ones out there may be found. Or they may return, to destroy us all out of fear and ignorance! But that would be their last mistake. We will kill them all, my Sons and Daughters, and I will be remembered forever in awe.

To you, noble warriors of Kalykk, I grant our finest ships. Take them with my blessing and set out into the Darkvoid to oppose those who dare challenge our supremacy. Do not falter or hesitate in your task. Destroy them wherever you find them. Subjugate their colonies, terrify their precious leaders!


But remember! You will not stray from your duty to me, your glorious Emperor. Obey my will in all things and you will be rewarded greatly. Disobey me, and your carapace will adorn the walls of the Imperial Nest for all my people to witness your shame!

Leave me! Go out into the Darkvoid and spread my will!

Posted by Pádraic Brady at 06:11


Thursday, March 2. 2006

## Subversion on Sourceforge

It's good news for open source developer's seeking Subversion access. It is now available from [Sourceforge](#) for open source projects. I've been using SVN on my personal PC at home while developing QS Evolved so I'm at revision 68 according to my SVN repository logs. I'll have to export my repository and migrate it to Sourceforge at the weekend if possible and get this development effort more public accessibility. It may finally prove the elusive QSE is actually progressing 


Posted by Pádraic Brady at 11:47

## ADOdb vs ADOdb-Lite? It's not a war...


Making a short (yeah, like that'll ever happen 

) commentary on iamsure's post about ADOdb over on

<http://www.kabal-invasion.com/bugjuice/>. Iamsure is a developer on the [Blacknova Traders](#) space strategy game.

First up ADOdb is a fantastic library. But it's not perfect. I'll be murdered for saying that perhaps - but it's not. It has problems - now these are problems unlikely to be noticed in the right circumstances. An opcode cache running on a dedicated server can hide a few warts 

In short - it uses a lot of memory, its ungainly, its slow (usual limits apply - see second paragraph), it had a stupid security flaw until v4.71 and it just feels plain bloated. As a player of TL said after the recent switch to ADOdb-Lite: "Hey, did you upgrade the server or summit?" Under stress ADOdb-Lite performs significantly better. It still performs better than ADOdb even with an opcode cache installed.

Its a matter of personal circumstances. ADOdb is great - but it sucks a little too much RAM for something hosted on a shared host or a cheap dedicated server seeing high usage. ADOdb-Lite is Lite. It doesn't suck a lot of RAM, and its faster. So nah nah nah.. 

Feature wise the news is hardly grim. ADOdb-Lite supports bind variables and has done for a long time - it's the basis of Partholans Data Access layer where every SQL query is bound. It supports transactions via an optional module - i.e. you need only include it when required. In fact it supports a bit with these modules. And more can be added if someone feels the need. That includes the PEAR and DATE modules to name a few popular ones. It also supports ADOdb's XMLSchema which is used in QS-Evolved's installer, and also in our UserModules.

It does not (to date) support resultset cache-to-file or performance monitoring. The former I don't really mind (I could work around it) and I'm looking into whether a perfmon module is possible in any case - hopefully I can figure that out in the next few weeks.

The latter is difficult to assess. On the one hand many DBMS's have some form of intelligent memory cache, on the other it may be badly configured or disabled - typically by setting a low or zero cache size. In those

cases a file cache may boost performance, assuming file access is faster than database access (varies). In any case ADOdb does not offer an "intelligent" cache. Its not a simple decision to make anymore. MySQL 4.0 is standard fare these days.

An interesting mention was PostgreSQL string escaping. I reported this months ago incidentally to John Lim - even blogged it for what it was worth. Nothing was ever done about it despite multiple mentions, and I was migrating to ADOdb-Lite in any case so I dropped it. The fix was in fact to uncomment the qstr() function in the postgres64.inc.php driver. Stupidly easy - never done.

It took a much more visible public security notice (<http://nvd.nist.gov/nvd.cfm?cvename=CVE-2006-0410>, and also a Gentoo Linux security mention) to get it finally resolved. Bad publicity has that effect. I can't say I was entirely happy using a library claiming to escape data when it still did nothing of the sort months after its reported as a security flaw in a less public manner.

ADOdb-Lite had this fixed within a short time of my reporting it. The security conscious at work...

Posted by Pádraic Brady in PHP General at 01:39

## No mbstrings attached: PHP UTF-8

It's interesting to see where setting Community Aims will get you. QS-E has a few so-called Community Aims, these are basically goals we must attain to support some of the most requested features in Quantum Star SE Evolved. A close second to the ideal of being able to transparently add new content to the game was the idea of allowing a full translation feature and native support for all multi-byte character encodings - We have gotten translation offers before for Hindi, Romanian, Russian, and more.

Recently given my prior blog post on a translation system I was looking at the system as a test implementation, and the broader area of I18N (Internationalisation). Further to my earlier post of today (Man, I write too much to this blog!) I found out from a [Planet-PHP.net](#) blog entry that [Harry Fuecks recently released a godsend](#). He's written and released the [0.1 version of a PHP UTF-8 library](#) which does NOT rely upon having the mbstring library available. This is fantastic news! Using this library one can actually operate on UTF-8 multi-byte characters safely - no corruption of multi-byte strings!

I modified a test included in the library download. It offers the following results - with my name finally represented accurately!

Operating on a 20 character string of both single and multi-byte characters...

The following represents String operations utilising a default PHP installation without the mbstring library.

String is: IÃfÃ±tÃfÃ«rnÃfÃçtiÃfÃ´nÃfÃ lizÃfÃ!tiÃfÃ,n

This is a UTF-8 string - standard PHP manipulations follow (no mbstring enabled):

Num chars: 27 (it's actually 20 - PHP cannot count characters since strlen() actually counts bytes)

Uppercase: IÃfÃ±TÃfÃ«RNÃfÃçTIÃfÃ´NÃfÃ LIZÃfÃ!TIÃfÃ,N (php ignored all the multi-byte characters)

Reversed:

nÃfÃ´iÃfÃ½iÃfÃ½zÃfÃ½lÃfÃ½iÃfÃ½tÃfÃ½nÃfÃ½rÃfÃ½tÃfÃ½!ÃfÃ½LÃfÃ½I (php corrupted the multi-byte characters, turning each into twin corrupted byte chars)

The following represents String operations utilising a default PHP installation without the mbstring library. The test does however utilise a PHP implementation of UTF-8 safe String functions.

String is: IÃfÃ±tÃfÃ«rnÃfÃçtiÃfÃ´nÃfÃ lizÃfÃ!tiÃfÃ,n

It's well formed UTF-8

Num chars: 20

Uppercase: IÃfÃ±TÃfÃ«RNÃfÃçTIÃfÃ´NÃfÃ,-LIZÃfÃ TIÃfÃËœN

Reversed: nÃfÃ´,itÃfÃ!zilÃfÃ nÃfÃ´itÃfÃçnrÃfÃ«tÃfÃ±I

My Name:

Maugrim's real name: PÃ¡draic

Reversed (Default PHP): ciardÃ´iÃfÃ½P


Reversed (with UTF-8 Lib): ciardÃ¡P

Posted by PÃ¡draic Brady in PHP General at 12:46

## QS Evolved: The Ship Model

I've gotten a few questions (again) about needing to add more ships to QS Evolved over what is available from Solar Empire and QS2. I've blogged about this before so here's the short version.

QS Evolved uses a hierarchical system. At the top are Fleets. Fleets are a collection of Ships. Here's the kicker, Ships are a collection of Ship Components which infer attributes and value upon the Ship. I know it's not instantly intuitive but ships are not a fixed entity. There are fixed Ships you can buy - however these are standard offerings only - even they can be modified over time. The big clue is Ship Production. If you want a ship you have two options. Buy it or make it. If you make it you can within reason design your ship from the Hull up, add components to it, and manufacture it. In the end what the ship is capable of is your choice - you designed it!

The other thing about ships is they'll carry a few new limitations. Some weapons may require a finite supply of ammunition. Some may depend on your ship having sufficient power to fire them. Similar restrictions impact engines and shields. The idea behind all this is to remove the stupidly obvious scenario in QS2/SE that a game with unlimited resources will tend to identical fleet strengths - i.e. a stalemate, and usually one that emerges within days - not weeks. It also adds more strategic flavour to the game - if the ship components are up to you, then you can make a lot of choices to favour your tactics. You could outfit a ship with nothing but big powerful engines - let a Warship try catching you then 


The attribute model of ships also enables ships to continually improve their performance - again within reason, and bounded by the capabilities of available components. Rather than having a ton of ships with limited abilities, you'll have a smaller (25-50 range max) more easily managed set of ships which you could spend weeks constantly improving in small and large ways.

Absolutely key to everything is a simple assumption:

"Ships are a valuable asset. Losing a ship imposes a cost in time and cash."

In Solar Empire and QS2 ships are NOT a valuable asset. You can lose 150 ships, and rebuild your fleet from scratch in a few days if the game settings are liberal (which they nearly always are). This is a big let down - ships are not worth upgrading unless they have a real value, enough that losing even a handful is a painful experience. The more value driven they are - the more incentive to upgrade, improve and increase their inherent value more. The more these occurs the more time and cash it takes to replace lost ships.

Like I said before, I am not cloning Solar Empire. This is a new Ship Model where losing a ship is a big deal - not an inconvenience solved by visiting a shipyard at Earth. A well upgraded ship with top-notch components could have cost you 5 days of mined resources. Even if you dragged all your other ships up to a superminer configuration it could still take 3 days of mining to even come close to replacing it - and that's without the time needed to build it, and maybe collect any rare resources required for the construction of its rarer components.

So keep this in mind. When QSE is playable you will be able to manage up to 50 ships. Personally I'd prefer less - but lets say 50. To get truly powerful ships you'll need to build them on planets. Doing this takes time, and a lot of cash. Losing a ship will be a big deal - it could take days to replace if of a high level. Ships are therefore a valuable asset - sacrificing them is still a strategic option, but a costly one if done for something stupid 

. Ship management will require actual thought - those of a strategic mind would do well to plan carefully.

Posted by PÃ;draic Brady in Quantum Star SE at 06:11


## PHP Applications using UTF-8 - should we believe them?

It's not everyday you see both these terms side by side. PHP has poor UTF-8 support. Well, actually its non-existent once you ignore the optional libraries that many shared hosts probably won't compile with PHP. Now many will know all

about setting up HTML pages for UTF-8, for many its the latest programming mantra. Unfortunately just setting a content-type is not enough. This will work completely for English webpages - and only English webpages using unaccented roman characters [a-z0-9].

So what is this mysterious character encoding and why is it so very important to get it right?

The problem is the 16 ISO-8859's (ISO-8859-16 is a necessity for the EU Euro symbol I think), ASCII, and dozens of other language and custom character encodings to support native character sets. In short its a case of too many cooks. To try and unify all these variants and offer a single standard supporting all character sets (and therefore all languages) we have Unicode, and by extension UTF-8 encoding of the Unicode character set. With UTF-8, a UTF-8 encoded file can represent anything from standard ASCII (read English) to Hindi and more. For a multi-lingual web application this is the best thing since fire was invented - or would be if PHP supported UTF-8 and PHP developers figured out that string functions in PHP at present are doomed to failure with UTF-8.

Consider the example of a Sourceforge page - [https://sourceforge.net/users/maugrim\\_t\\_r/](https://sourceforge.net/users/maugrim_t_r/) . Yes, that's my profile 

However there's a small problem. The page insists it is encoded in UTF-8 (view the source). That's not true. You see, a simple European accented character in my first name is not displayed - its replaced with a ?. Why? Because its not UTF-8 encoded. Ha! So much for internationalisation on Sourceforge. It claims to be UTF-8 - but its obviously not UTF-8. The truth is that only parts (if any) are actually UTF-8. The rest is most likely ISO-8859-1 - i.e. ASCII English.

One could blame me - I'm a bad person who uses a name containing a Gaelic accented character (equivalent to the french a-acute character). Why not blame the French why we're at it, or the rest of those pesky non-English speaking people?

Its because of this that many self-proclaimed web apps purporting to support internationalisation or localisation (we use the abbreviation I18N for the latter) are being unrealistic. Many will support I18N for English, and a subset of European languages where accented characters can be given an English character variant, or maybe if they use a template system with UTF-8 encoded templates that require no manipulation - i.e. no database storage or use of PHP string functions. Some will require the mbstring library be available for PHP.

There are still problems. Take a simple example. I create a form accepting a username of no more than 32 characters. So someone from India drops by and enters a Hindi username - 32 Hindi characters. Now the form is submitted. Everything is great, right? We just check for 32 characters. But wait - there is no character length function in PHP! And no, strlen() is not that - not at all.

strlen() does NOT count characters - it counts BYTES. Never forget that distinction lest it burn you. Now UTF-8 is a multi-byte encoding. Keeping in mind UTF-8 incorporates single-byte ASCII we can say the string "Reaper" has 6 bytes, equalling 6 characters. This is what most of us already do - its the evil assumption. But what about Hindi? Hindi uses multi-byte characters.


A 32 character Hindi string will actually contain a lot more than 32 bytes - some languages may take up to 32 \* 6bytes, i.e. 192 bytes to represent a 32 character string! Result - our filtering logic fails miserably, a victim of our false assumption that the whole world speaks English and writes with Roman characters.

There are other deeply set practices that fail to account for UTF-8. While Perl supports UTF-8 in RegExp, PHP's PCRE does not. How many of us use regular expressions in form validation? Probably the same number who use strlen() to count characters... There are other examples. Basically you could list all the string functions.

There is a solution however - we can compile (or edit php.ini for dynamic inclusion) the mbstring library for PHP. However this has one flaw - it's not a default library which means many shared hosts will not actually support it. So much for PHP's UTF-8 support. The other solution is to wait for PHP6 which will have native UTF-8 support. But then what

about all those hosts who'll persist in using PHP4 because PHP5 is not backwards compatible?

In short, unless you're a native English speaker - people using PHP and outputting UTF-8 html while claiming to support I18N are (in the majority) completely ignorant. Afterall for an English speaker there is zero difference between using ISO-8859-1, ASCII and UTF-8. Its the rest of the world that must jump through hoops. Even worse - many of the UTF-8 HTML out there is not even encoded in UTF-8. Often its just a typical ISO-8859 document (written on WindowsXP in almost all cases) with a UTF-8 content-type charset - this largely explains why name is commonly corrupted with question marks and boxes on many English sites claiming (falsely) to be UTF-8 encoded.

My name is not P?draic or Pdraic - it's PÃ¡draic! 

Posted by PÃ¡draic Brady at 02:16