Subject: Re: Documentation - Broken links
Posted by jjtoranzo2004 on Wed, 07 Nov 2012 15:30:53 GMT
View Forum Message <> Reply to Message

Tony, I found the following Uniface broken links:

Uniface
=======
http://www.marston-home.demon.co.uk/Tony/uniface/3tiers2mode lsXMLstreams.html
(should point to http://www.tonymarston.net/uniface/3tiers2modelsXMLstreams.h tml)
linked from page:
http://www.tonymarston.net/php-mysql/standards.html

Line 289
The leaders of this doomed project had a team of 6 developers working for 6 months building and testing their development infrastructure, but at the end of the day it proved to be totally worthless. When I left the client's site and returned to my head office I decided to build my own version of a 3-tier infrastructure with a separate model for the application layer. What those supposed 'experts' could not accomplish in 6 months I managed to complete on my own in 2 weeks. The results can be reviewed in <a href="
http://www.marston-home.demon.co.uk/Tony/uniface/3tiers2mode lsXMLstreams.html">3 Tiers, 2 Models, and XML Streams. I have subsequently used these ideas to convert the 150+ forms in my demonstration UNIFACE system, which contains a Menu and Security system and a sample application, so that it is totally 3-tier. It just goes to show what you can achieve if your approach is sound to begin with.

---
http://www.marston-home.demon.co.uk/Tony/uniface/componentte mplates.html
(should point to http://www.tonymarston.net/uniface/componenttemplates.html)
linked from page:
http://www.tonymarston.net/php-mysql/object-relational-mappe rs-are-evil.html

Line 229
The UNIFACE language had a feature called <a href="
http://www.marston-home.demon.co.uk/Tony/uniface/componentte mplates.html">component templates which made it easy to generate new components from a pre-defined template or pattern. It worked on the premise that each component can be categorised by structure, behaviour and content, with the template built around a particular structure (screen layout) and behaviour (code). A functioning component could be created by combining a template with a particular entity (database table). As I had found this feature a great aid to rapid application development I wanted something similar in my PHP framework, but as it did not exist I had to invent it. I devised a catalog of transaction patterns where the structure is provided by an XSL stylesheet, and the behaviour is provided by a page controller. By combining a pattern with a database table class (the content) the result is a functioning transaction.

---
http://www.marston-home.demon.co.uk/Tony/uniface/largevssmal l.html
(should point to http://www.tonymarston.net/uniface/largevssmall.html)
linked from pages:
http://www.tonymarston.net/php-mysql/role-based-access-contr ol.html
http://www.tonymarston.net/php-mysql/model-view-controller.h tml
http://www.tonymarston.net/php-mysql/stateless-protocol.html

---
http://www.marston-home.demon.co.uk/Tony/uniface/menuguide/i ndex.html
(should point to http://www.tonymarston.net/uniface/menuguide/index.html)
linked from page:
http://www.tonymarston.net/php-mysql/role-based-access-contr ol.html

http://www.marston-home.demon.co.uk/Tony/uniface/uniface(bb).html
(should point to http://www.tonymarston.net/uniface/uniface(bb).html)
linked from page:
http://www.tonymarston.net/php-mysql/role-based-access-contr ol.html

Line 60
In the 1990s I rewrote this <a href="
http://www.marston-home.demon.co.uk/Tony/uniface/menuguide/i ndex.html">Menu and Security system in a little-known 4th generation language called UNIFACE.

---
http://www.marston-home.demon.co.uk/Tony/uniface/standards.h tml
(should point to http://www.tonymarston.net/uniface/standards.html)
linked from page:
http://www.tonymarston.net/php-mysql/standards.html

Line 24
(adapted from an article on my UNIFACE page)

Thanks.