Deploying PHP applications using Virtuoso as Application Server

Table of Contents

What

OpenLink Virtuoso has the capability to act as a host for PHP applications. Where you're currently used to seeing a LAMP (Linux, Apache, MySQL, PHP) stack, Virtuoso can do the job of both Apache as web-server and MySQL as database - and lots more.

Virtuoso brings a myriad of extra features, from not just a fast relational database store (although it does that too), to RDF triple-store (actually a quad-store; each triple is associated with a graph), to semantic-web engine to application-development platform and beyond.

Why

Existing applications such as PHPBB and WordPress are large pools of personal and social data - a "data space" - yet the means of querying them are mostly limited to the domain and application themselves. For example, within a phpBB3 instance, you can type a word into the search box, but the results are constrained to coming from only that instance, notably the format in which it chooses to display them and the actions it permits you to take based on the results, and (in this case) there is no URL with which to identify the query itself (so no bookmarks, no sharing with others). More advanced queries require digging around the underlying tables with SQL, if you have access to the backend DB.

One could summarize the possible query-methods:

- application-provided search boxes
- site-wide statistical free-text (google)
- custom reports requiring SQL/developer abilities

None of these methods allow one to execute a search on a WordPress blog for "red, by which I mean the square in Moscow not the colour". As we shall see, semantic-web technologies will address this.

The good news is that the underlying data is much richer, and may be exposed in such a way for users to deploy their own query-tools against it by opening it up in the form of RDF.

A note on RDF

RDF is a framework based on the Entity-Attribute-Value data-model; it can be thought of as a set of statements expressed in triples subject, predicate and object where the subject entity and predicate must be identified using URIs, and the value object may be either a URI or a literal value.

As a separate consideration, RDF may be serialized in several formats; there is RDF/XML which is orientated towards machine-reading, but there are also Turtle and N-3 syntaxes more readily accessible to humans. We regard the representation as of secondary importance compared to the fact it is statements being represented in the form of triples.

A set of predicates combine into vocabularies known as "ontologies"; there are several well-known ontologies such as FOAF and Dublin Core which, amongst many other things, allows you to make a statement such as

```
prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://www.openlinksw.com/> dc:subject "database" .
```

Entities, described by RDF, group into "graphs" - a set of nodes linked one to another by predicates. This is an improvement on Web-1.0 and 2.0, where all that could be said for a link is that one page links to another, optionally with some significance attached to the freeform keywords used for the link. With RDF, all links are precisely meaningful: you know what it means to be a foaf:nickName, as well as the fact that a particular entity has a given nickname, for example.

Further, there is a language for querying RDF data, SPARQL, where the queries may be routed over HTTP as a transport (by the data-provider making a /sparql endpoint visible). This gives every SPARQL query a fixed URL for later reference.

Querying extends to browsing, made possible with generic RDF-browser utilities such as OpenLink's ODE and Zitgist, or Tabulator, etc.

When URIs and URLs collide: Linked Data

A URL is the location of a document on the Web; its purpose is to express location only.

A URI is a unique identifier for an entity; there is implicit binding between an entity and its metadata through a single identifier. In programming terms it may be regarded as similar to a "variable name" or a symbol. As such there is no obvious need to use the HTTP protocol, ie to name everything starting http://.... but it helps if you do.

An RDF triple- or quad-store contains URIs for subjects and predicates; it may be queried purely as though it were an abstract object-database with no significance attached to the "names" of the subjects retrieved.

However, when HTTP URIs are used as names for entities, they become real - they work - they can be dereferenced. You can publish RDF data on the web simply, precisely because it takes the form of URIs using the HTTP protocol. From an alternative perspective, the entire WWW becomes an abstract database free to be queried. It allows data published by different organizations to be federated, a process known as Linked Data.

As a bonus, a web-server can be configured to perform content-negotiation based on what kind of a response a client requests; the same URL may return text/html (better, application/xhtml+xml) for a web-browser, but a representation in application/rdf+xml for an RDF browser requesting it.

Coming full-circle, to ease the production of data for existing webmasters, RDF may be embedded in (X)HTML documents using either RDFa, GRDDL or eRDF markup; all these and custom microformats may be translated into RDF using an appropriate agent (such as OpenLink's Sponger, which can be seen in action at URIBurner).

Data Example

It is possible and desirable to define a mapping between relational and RDF-based databases.

In the most obvious form, for a given table, each field becomes a predicate and each row an entity.

Taking as an example the case of PHPBB3's well-known `phpbb_users` table,

user_id	user_type	group_id	user_permissions	user_perm_from	user_ip	user_regdate	username	username_clean	user_password	user_passchg	user_pass_convert	user_email
1	2	1	null	0	null	1222764432	Anonymous	anonymous	null	0	0	<none>

one can obviously make a set of statements about this user,

```
<http://localhost/rdf/phpBB/user_1>

<http://localhost/rdf/phpBB/has_email> "root@localhost" ;

<http://localhost/rdf/phpBB/has_name> "Anonymous"  .
```

etc.

Obviously this doesn't gain us much. However, you could further extend this to other well-known ontologies; for example, formulate a mapping from PHPBB's discussion forums to sioc:Forum entities, from posts within those forums to sioc:Post entities, etc; users can become entities of type sioc:User, with data filled-out using the FOAF ontology to express their properties (name, nickname, email address, homepage, etc). Then anyone using an RDF browser will experience the well-known sense of hierarchy that SIOC brings, that "this forum" *contains* "this thread" which *contains* "this comment" - because the SIOC ontology brings that nesting structure by defining what `/contains/' means.

This adds value to the existing data as well as transforming it. (As an aside, consider that the same hierarchy suffices to describe Amazon's website: a section (music/dvd/books/electronics) might correspond to a sioc:Forum, with a product for sale being a sioc:Post and a review being a sioc:Comment within it. This enables you to browse RDF by predicate, impossible with current conventional search-engines.)

OpenLink Virtuoso has a system known as RDF Views, whereby this translation from relational to RDF data is made dynamic; the triples are not realised but rather calculated from the relational data on demand. This way you never have out-of-date data whilst waiting for another import or sync.

You can write the scripts to implement these RDF Views by hand, but Virtuoso has automated the whole process, both in general with transforming relational data from tables to RDF, and specifically for the cases of PHPBB, Wordpress, MediaWiki and Drupal.

How to

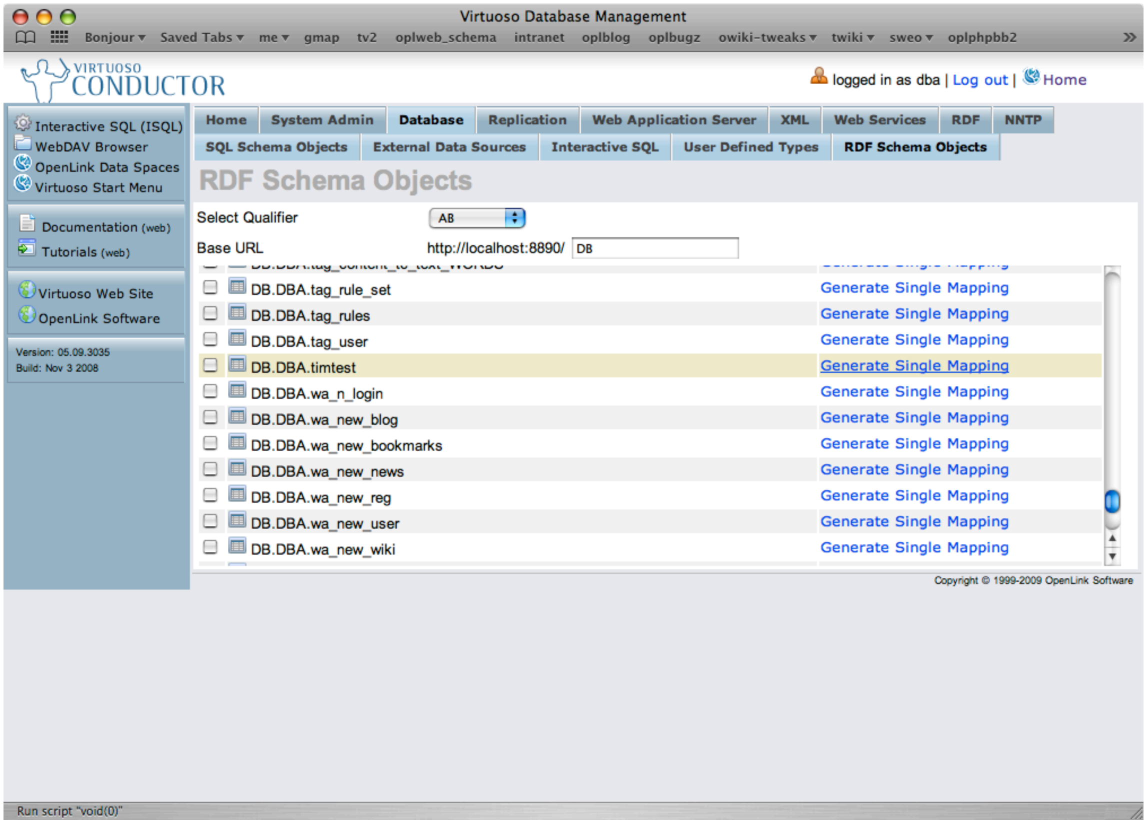
Using the Conductor

Virtuoso has a generalized mechanism for automatically generating a canned RDF view from a given table, available from the Conductor administrative interface.

First we create a simple test table:

```
SQL> create table timtest (id integer primary key, str varchar,
dt datetime);
Done. -- 36 msec.
SQL> insert into timtest values (1, 'hello world', now());

Done. -- 2 msec.
SQL> insert into timtest values (2, 'this is a test', now());

Done. -- 2 msec.
SQL> grant all on timtest to public;

Done. -- 2 msec.
```

Point your browser at http://localhost:8890/conductor/ and log in; navigate your way to Database / RDF Schema Objects. You will see a list of existing tables in the given schema/qualifier, and an action `generate single mapping' beside:



RDF Schema Objects

Select Qualifier: AB

Base URL: http://localhost:8890/ DB

DB.DBA.tag_rule_set	Generate Single Mapping
DB.DBA.tag_rules	Generate Single Mapping
DB.DBA.tag_user	Generate Single Mapping
DB.DBA.timtest	Generate Single Mapping
DB.DBA.wa_n_login	Generate Single Mapping
DB.DBA.wa_new_blog	Generate Single Mapping
DB.DBA.wa_new_bookmarks	Generate Single Mapping
DB.DBA.wa_new_news	Generate Single Mapping
DB.DBA.wa_new_reg	Generate Single Mapping
DB.DBA.wa_new_user	Generate Single Mapping
DB.DBA.wa_new_wiki	Generate Single Mapping

To test the results, we first observe the line in the RDF Views script where it declares the graph IRI (a container for triples) it's going to create:

```
   create virtrdf:DB as graph iri
("http://^{URIQADefaultHost}^/db") option (exclusive)
```

Next, point your browser at http://localhost:8890/sparql/ , ie the /sparql query endpoint in the Virtuoso instance. Set the default graph to `http://localhost:8890/db` and enter the query

```
SELECT * WHERE {?s ?p ?o} limit 100
```

Execute it, and you should be presented with an HTML table showing the RDF data mapping of your SQL table:

s	p	o
http://gentoo:8890/DB/timtest/id/1#this	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://localhost:8890/DB#timtest
http://gentoo:8890/DB/timtest/id/2#this	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://localhost:8890/DB#timtest
http://gentoo:8890/DB/timtest/id/1#this	http://localhost:8890/DB#id	1
http://gentoo:8890/DB/timtest/id/2#this	http://localhost:8890/DB#id	2
http://gentoo:8890/DB/timtest/id/1#this	http://localhost:8890/DB#str	hello world
http://gentoo:8890/DB/timtest/id/2#this	http://localhost:8890/DB#str	this is a test
http://gentoo:8890/DB/timtest/id/1#this	http://localhost:8890/DB#dt	2009-07-20T16:41:59.000000
http://gentoo:8890/DB/timtest/id/2#this	http://localhost:8890/DB#dt	2009-07-20T16:42:13.000000

As this table demonstrates, the RDF Views generator has decided:

- to use /db as a graph IRI to identify this transformation project
- to use /DB/timtest/id/ as a base for entity URIs
- to use /DB/#predicate as a base for predicate URIs

Naturally you can use the auto-generated RDF View script as a starting point for your own scripts, adding value through references to existing ontologies and using stored functions/procedures to manipulate the values into a suitable string form for use in the URIs. There is more information about this to be found in the documentation.

Existing Applications

In this case we follow the process of creating RDF data from an empty Virtuoso instance, using PHPBB3 as an example.

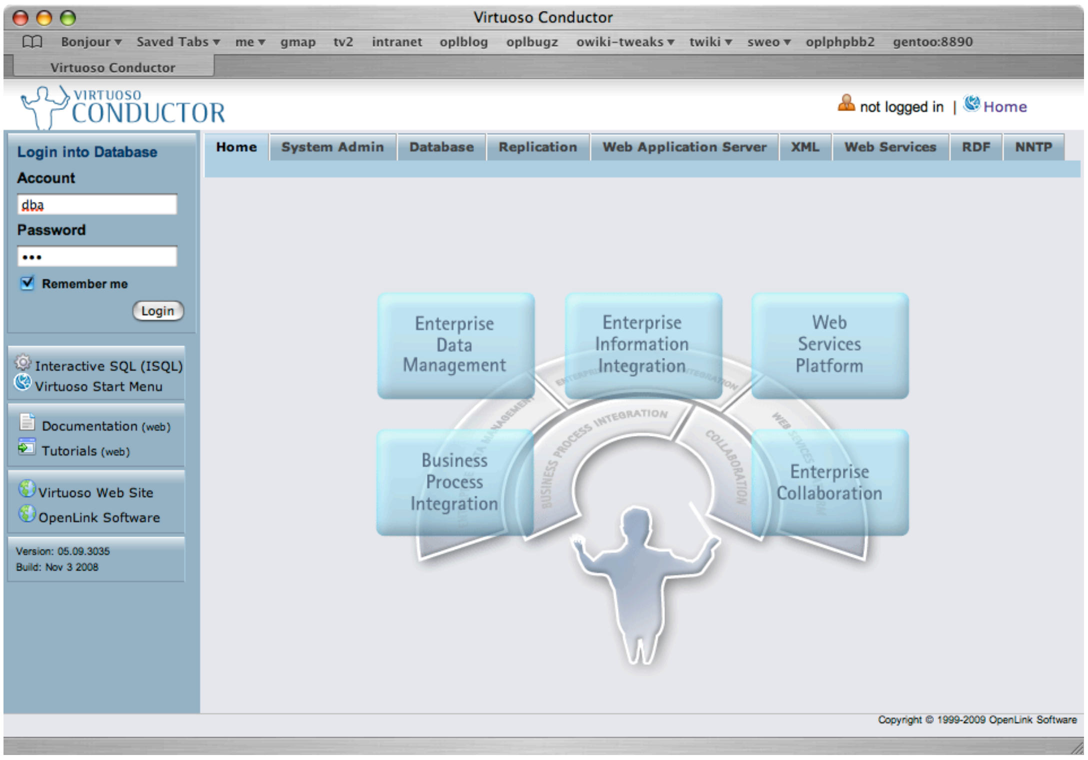
Download, install and start Virtuoso:

```
zsh, gentoo 11:30AM virtuoso/ % ./bin/virtuoso-start.sh
Starting Virtuoso instance in [database]
```

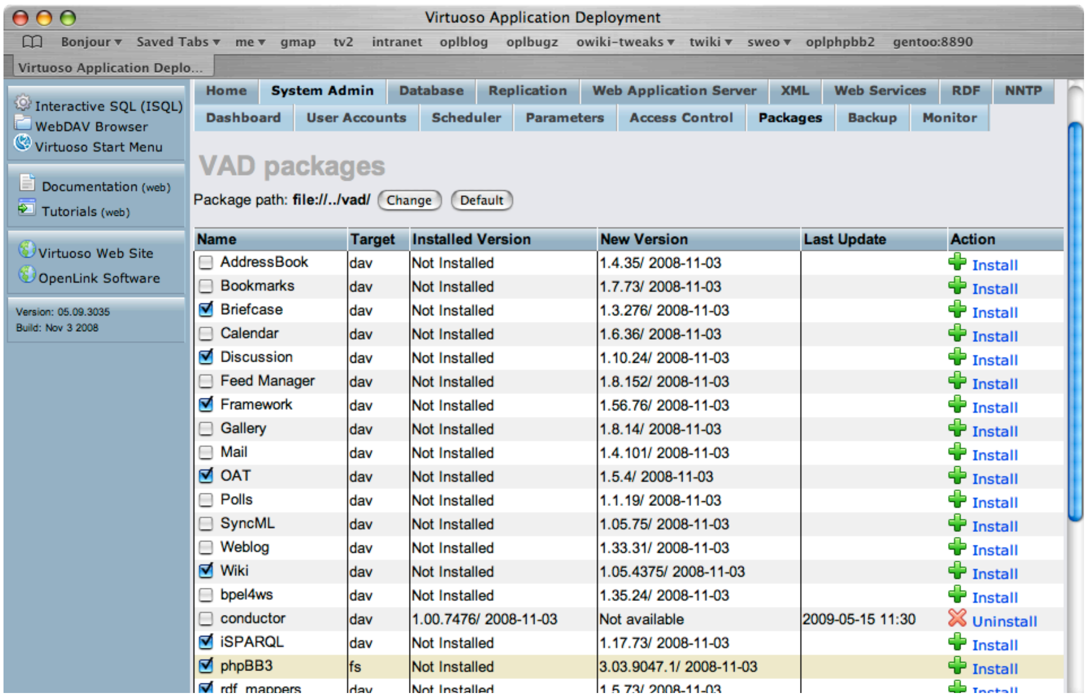
By default this will give you an instance listening on port 1111 for SQL / ODBC connections and port 8890 for HTTP.

Installing ODS

Point your browser at http://hostname:8890/conductor/ and log in to the conductor. Choose the System Admin tab and Packages there.


Virtuoso Conductor
Bonjour Saved Tabs me gmap tv2 intranet oplblog oplbugz owiki-tweaks twiki sweo oplphpbb2 gentoo:8890
VIRTUOSO CONDUCTOR
not logged in | Home
Home System Admin Database Replication Web Application Server XML Web Services RDF NNTP
Login into Database
Account
dba
Password
Remember me
Login
Interactive SQL (ISQL)
Virtuoso Start Menu
Documentation (web)
Tutorials (web)
Virtuoso Web Site
OpenLink Software
Version: 05.09.3035
Build: Nov 3 2008
Enterprise Data Management
Enterprise Information Integration
Web Services Platform
Business Process Integration
Enterprise Collaboration
Copyright © 1999-2009 OpenLink Software

Select a handful of ODS packages, such as Discussion, Framework and Wiki and definitely PHPBB3, iSPARQL and rdf_mappers and press `install'.


Virtuoso Application Deployment
Bonjour Saved Tabs me gmap tv2 intranet oplblog oplbugz owiki-tweaks twiki sweo oplphpbb2 gentoo:8890
Home System Admin Database Replication Web Application Server XML Web Services RDF NNTP
Dashboard User Accounts Scheduler Parameters Access Control Packages Backup Monitor
Interactive SQL (ISQL)
WebDAV Browser
Virtuoso Start Menu
Documentation (web)
Tutorials (web)
Virtuoso Web Site
OpenLink Software
Version: 05.09.3035
Build: Nov 3 2008

VAD packages

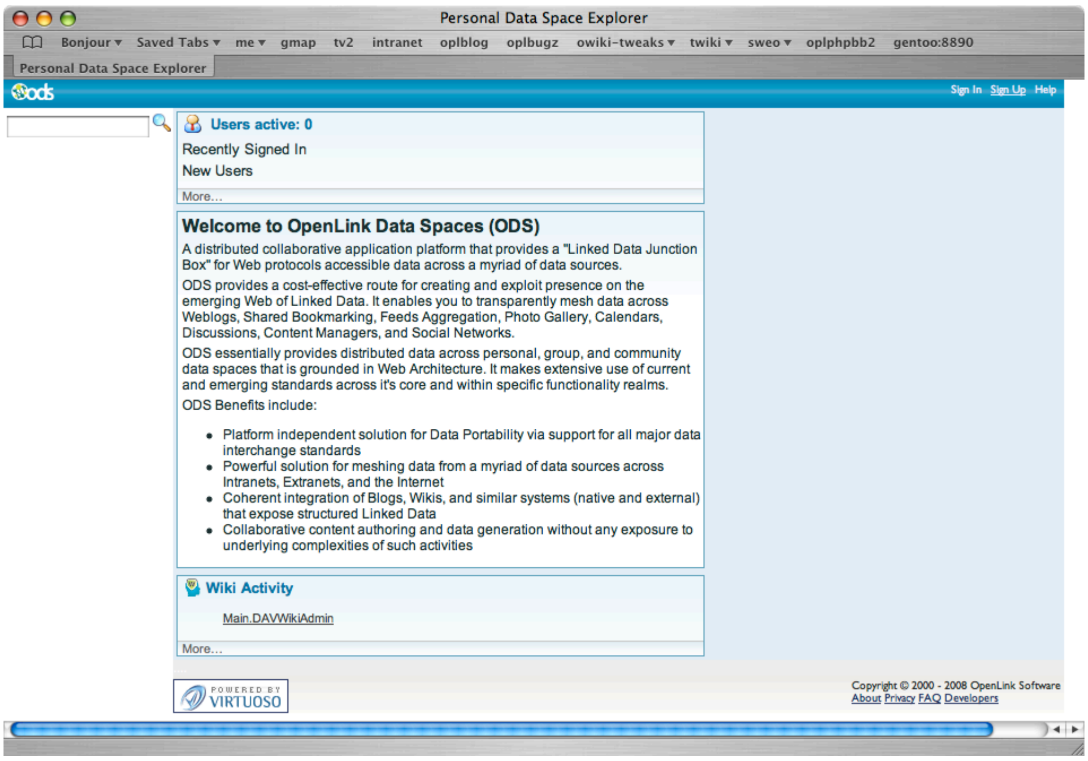
Package path: file://../vad/ Change Default

Name	Target	Installed Version	New Version	Last Update	Action
☐ AddressBook	dav	Not Installed	1.4.35/ 2008-11-03		Install
☐ Bookmarks	dav	Not Installed	1.7.73/ 2008-11-03		Install
✓ Briefcase	dav	Not Installed	1.3.276/ 2008-11-03		Install
☐ Calendar	dav	Not Installed	1.6.36/ 2008-11-03		Install
✓ Discussion	dav	Not Installed	1.10.24/ 2008-11-03		Install
☐ Feed Manager	dav	Not Installed	1.8.152/ 2008-11-03		Install
✓ Framework	dav	Not Installed	1.56.76/ 2008-11-03		Install
☐ Gallery	dav	Not Installed	1.8.14/ 2008-11-03		Install
☐ Mail	dav	Not Installed	1.4.101/ 2008-11-03		Install
✓ OAT	dav	Not Installed	1.5.4/ 2008-11-03		Install
☐ Polls	dav	Not Installed	1.1.19/ 2008-11-03		Install
☐ SyncML	dav	Not Installed	1.05.75/ 2008-11-03		Install
☐ Weblog	dav	Not Installed	1.33.31/ 2008-11-03		Install
✓ Wiki	dav	Not Installed	1.05.4375/ 2008-11-03		Install
☐ bpel4ws	dav	Not Installed	1.35.24/ 2008-11-03		Install
☐ conductor	dav	1.00.7476/ 2008-11-03	Not available	2009-05-15 11:30	Uninstall
✓ iSPARQL	dav	Not Installed	1.17.73/ 2008-11-03		Install
✓ phpBB3	fs	Not Installed	3.03.9047.1/ 2008-11-03		Install
✓ rdf_mappers	dav	Not Installed	1.5.73/ 2008-11-03		Install

Setting up ODS & PHPBB3

Background: when new ODS (WebDAV) users are created, a trigger automatically adds them as PHPBB3 users. This is the preferred way to register PHPBB3 users.

Point your browser at http://hostname:8890/ods/ and click sign-up. Account-creation is simply a case of filling in the form; you will be presented with a view of your account profile when complete.



Personal Data Space Explorer
Bonjour ▾ Saved Tabs ▾ me ▾ gmap tv2 intranet oplblog oplbugz owiki-tweaks ▾ twiki ▾ sweo ▾ oplphpbb2 gentoo:8890
Personal Data Space Explorer
Sign In Sign Up Help

Users active: 0

Recently Signed In
New Users
More...

Welcome to OpenLink Data Spaces (ODS)

A distributed collaborative application platform that provides a "Linked Data Junction Box" for Web protocols accessible data across a myriad of data sources.

ODS provides a cost-effective route for creating and exploit presence on the emerging Web of Linked Data. It enables you to transparently mesh data across Weblogs, Shared Bookmarking, Feeds Aggregation, Photo Gallery, Calendars, Discussions, Content Managers, and Social Networks.

ODS essentially provides distributed data across personal, group, and community data spaces that is grounded in Web Architecture. It makes extensive use of current and emerging standards across it's core and within specific functionality realms.

ODS Benefits include:

- Platform independent solution for Data Portability via support for all major data interchange standards
- Powerful solution for meshing data from a myriad of data sources across Intranets, Extranets, and the Internet
- Coherent integration of Blogs, Wikis, and similar systems (native and external) that expose structured Linked Data
- Collaborative content authoring and data generation without any exposure to underlying complexities of such activities

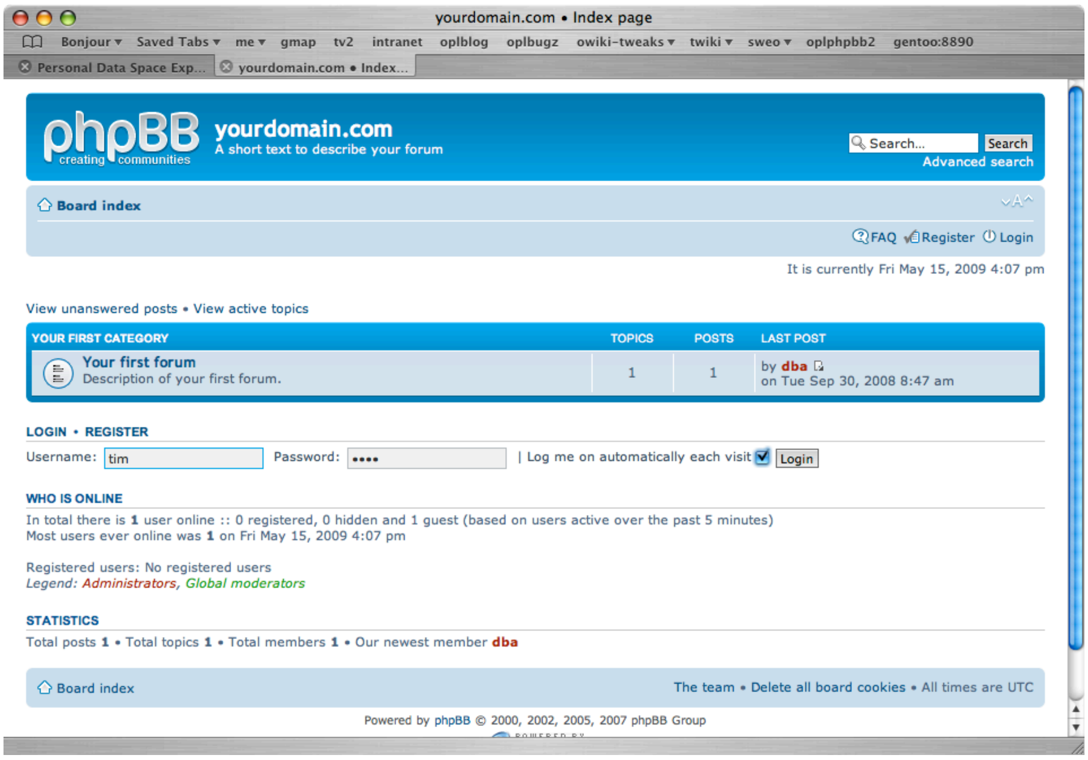
Wiki Activity

Main.DAVWikiAdmin
More...

POWERED BY VIRTUOSO

Copyright © 2000 - 2008 OpenLink Software
About Privacy FAQ Developers

Point your browser to http://hostname:8890/phpBB3/ - this installation is the result of installing the VAD package earlier. Log in as the user you created above and peruse the default post etc.



yourdomain.com • Index page
Bonjour ▾ Saved Tabs ▾ me ▾ gmap tv2 intranet oplblog oplbugz owiki-tweaks ▾ twiki ▾ sweo ▾ oplphpbb2 gentoo:8890
Personal Data Space Exp... yourdomain.com • Index...
phpBB creating communities
Search... Search
Advanced search

yourdomain.com
A short text to describe your forum

Board index

FAQ Register Login

It is currently Fri May 15, 2009 4:07 pm

View unanswered posts • View active topics

YOUR FIRST CATEGORY	TOPICS	POSTS	LAST POST
Your first forum Description of your first forum.	1	1	by **dba** on Tue Sep 30, 2008 8:47 am

LOGIN • REGISTER

Username: tim Password: •••• | Log me on automatically each visit ☑ Login

WHO IS ONLINE

In total there is **1** user online :: 0 registered, 0 hidden and 1 guest (based on users active over the past 5 minutes)
Most users ever online was **1** on Fri May 15, 2009 4:07 pm

Registered users: No registered users
Legend: Administrators, Global moderators

STATISTICS

Total posts **1** • Total topics **1** • Total members **1** • Our newest member **dba**

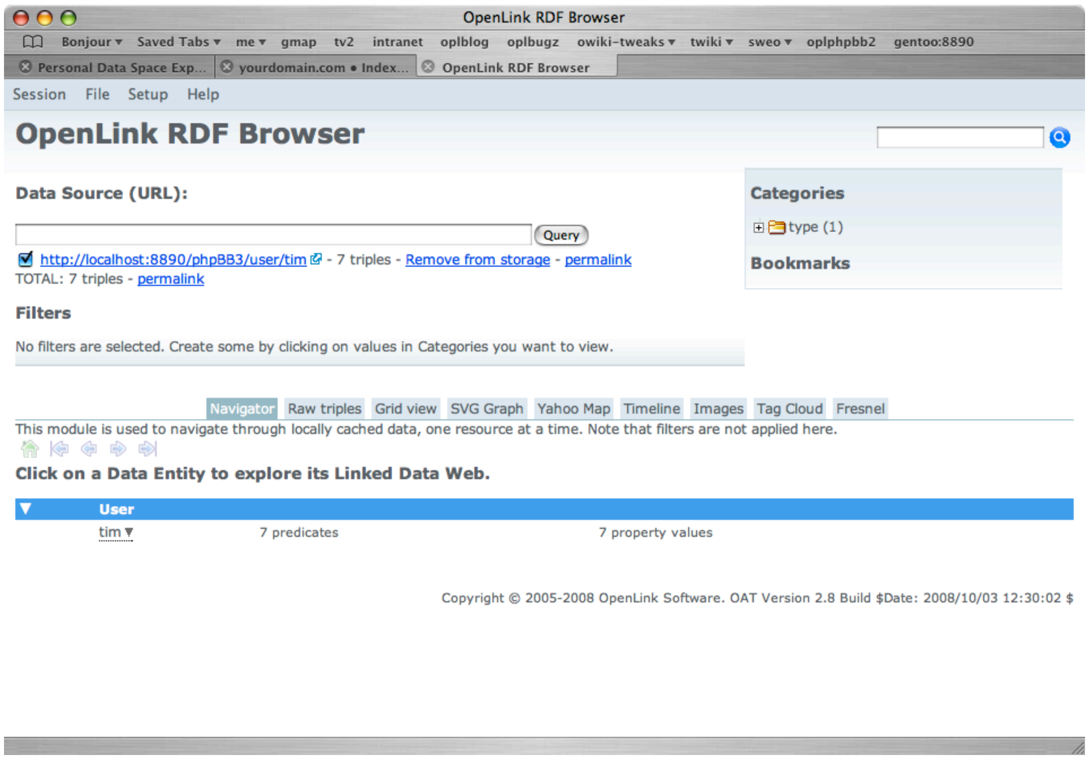
Board index The team • Delete all board cookies • All times are UTC

Powered by phpBB © 2000, 2002, 2005, 2007 phpBB Group

By extension, the system `dba' user has been made administrator of this phpBB instance.

As an interface straight into the RDF data, point your browser at http://hostname:8890/phpBB3/user/someuserid . This will present you with an instance of the OpenLink RDF Browser, with http://hostname:8890/phpBB3/user/someuserid as the source

URI from which to start browsing.



OpenLink RDF Browser

Data Source (URL):

http://localhost:8890/phpBB3/user/tim - 7 triples - Remove from storage - permalink
TOTAL: 7 triples - permalink

Filters

No filters are selected. Create some by clicking on values in Categories you want to view.

This module is used to navigate through locally cached data, one resource at a time. Note that filters are not applied here.

Click on a Data Entity to explore its Linked Data Web.

Copyright © 2005-2008 OpenLink Software. OAT Version 2.8 Build $Date: 2008/10/03 12:30:02 $

This is implemented using the same sort of content-negotiation as mentioned previously. If you experiment with `curl(1)` on the commandline, you'll see that the URL http://localhost:8890/phpBB3/user/tim gives an HTTP 303 `see other' response:

```
zsh, gentoo  5:14PM virtuoso/ % curl -I
'http://localhost:8890/phpBB3/user/tim'
HTTP/1.1 303 See Other
Server: Virtuoso/05.09.3035 (Linux) i686-generic-linux-glibc26-32
VDB
Connection: close
Content-Type: text/html; charset=UTF-8
Date: Fri, 15 May 2009 17:14:06 GMT
Accept-Ranges: bytes
Location: http://localhost:8890/rdfbrowser/index.html?
uri=http%3A//localhost:8890/phpBB3/user/tim#this
Content-Length: 0
```

However, treated as a URI, by requesting RDF data, one sees:

```
zsh, gentoo  5:15PM virtuoso/ % curl -H 'Accept:
application/rdf+xml' \
-I 'http://localhost:8890/phpBB3/user/tim'
HTTP/1.1 200 OK
Server: Virtuoso/05.09.3035 (Linux) i686-generic-linux-glibc26-32
VDB
Connection: Keep-Alive
Date: Fri, 15 May 2009 17:15:14 GMT
Accept-Ranges: bytes
Content-Type: application/rdf+xml; charset=UTF-8
Content-Length: 1457
```

http://localhost:8890/phpBB3/user/tim

Click around! Browse the data - check the SVG views and `raw triples' etc. Explore the links from statements about a given phpBB3 users to forums and articles posted by them.

For background reading: see this video of PHP hosting on Windows Vista.

Other PHP Applications

The above process can be repeated for MediaWiki, Drupal and WordPress; OpenLink has a VAD package for each of these applications and existing documentation pages with the same workflow as the above phpBB3 example:

Application	DataSpace	Documentation	Characteristic data URI
phpBB3	Bulletin-board, forums	phpBB	http://localhost:8890/phpBB3/user/me
Drupal	Content-management system	Drupal	http://localhost:8890/drupal/user/me
WordPress	Blog and CMS	WordPress	http://localhost:8890/wordpress/user/me
MediaWiki	Wiki documentation server	MediaWiki	http://localhost:8890/MediaWiki/user/me

Browsing

Each of these examples shows how to get "my URI" out of these things and play with an RDF mapping straight away. Additionally you can use a standalone RDF browser, such as the OpenLink Data Explorer, Tabulator or Disco to browse the data.

Developers: building Virtuoso Open-Source with PHP support

Here we walk through the process of hosting PHP in Virtuoso, building from compiling sources to extensions, writing a simple web-application and viewing data.

At this point it would be wise to double-check the top-level README.php5 file in the open-source distribution, as it may contain more up-to-date instructions.

There are three stages involved in using a custom build of PHP with Virtuoso;

- build PHP
- add any required PHP modules
- configure and build Virtuoso to reference the PHP you built.

Building PHP

In the event that you have to recompile the PHP hosting module yourself for use with an existing application, the following options are a *minimum* starting point:

```
./configure \
                --prefix=/usr/local/php5 \
                --enable-maintainer-zts \
                --with-tsrm-pthreads \
                --enable-embed=shared \
                --disable-static \
                --with-config-file-path=. \
                --disable-cgi \
                --disable-cli \
                --disable-ipv6 \
                --with-zlib \
                --with-iodbc=/usr
                .....
```

The ZTS and TSRM pthread options are particularly important; without them, Virtuoso will have serious threading incompatibilities. Note that many or most Linux distributions do not use these options, as they tend to package PHP for use as `mod_php` with the Apache web-server.

For reference, OpenLink uses the following configure options:

```
zsh/scr, gentoo  2:17PM php-5.2.10/ % ./configure \
                --prefix=/usr/local/php5 \
                --enable-maintainer-zts \
                --enable-embed=shared \
                --with-config-file-path=. \
                --with-tsrm-pthreads \
                --disable-static \
                --disable-cgi \
                --disable-ipv6 \
                --without-mysql \
                --without-pear \
                --enable-bcmath=shared \
                --enable-calendar \
                --enable-dbase=shared \
                --enable-dba=shared \
                --enable-dom=shared \
                --enable-exif=shared \
                --enable-ftp=shared \
                --enable-gd-native-ttf \
                --enable-mbstring=shared \
                --enable-pdo \
                --enable-shmop=shared \
                --enable-soap=shared \
                --enable-sockets=shared \
                --enable-sysvmsg=shared \
                --enable-sysvsem=shared \
                --enable-sysvshm=shared \
                --enable-wddx=shared \
                --enable-xmlreader=shared \
                --enable-xmlwriter=shared \
                --with-bz2=shared \
                --with-curl=shared \
                --with-gd=shared \
                --with-iodbc=/usr/local/iODBC \
                --with-ldap=shared \
                --with-mime-magic=shared \
                --with-openssl=shared \
                --with-pdo-odbc="generic,/usr/local/iODBC,iodbc,\
                                -L/usr/local/iODBC/lib,-
I/usr/local/iODBC/include" \
                --with-sqlite=shared \
                --with-xmlrpc=shared \
                --with-xsl=shared \
                --with-xsl=shared \
                --with-zlib \
                && nice make
...
```

You should pick and choose the options starting from the minimal requirements and whatever modules your application requires and/or are likely to be useful from the above. Note that many of these options require external libraries to build (eg openldap, bzip2, curl, libxml2, etc).

Some time later, the build completes and we install PHP:

```
...
Build complete.
Don't forget to run 'make test'.

zsh/scr, gentoo 12:56PM php-5.2.10/ % sudo make install
Password:
Installing PHP SAPI module:       embed
Installing build environment:     /usr/local/php5/lib/php/build/
Installing header files:          /usr/local/php5/include/php/
Installing helper programs:       /usr/local/php5/bin/
  program: phpize
  program: php-config
Installing man pages:             /usr/local/php5/man/man1/
  page: phpize.1
  page: php-config.1
```

Building Virtuoso with PHP

Now proceed to build Virtuoso, being sure to use the --enable-php5 option to configure:

```
zsh/scr, gentoo  4:34PM virtuoso-opensource-5.0.11/ % ./configure
\
        --prefix=/usr/local/virtuoso \
        --enable-php5=/usr/local/php5 --enable-imagemagick --
with-readline \
        --with-iodbc=/usr/local/iODBC
```

When it's done configuring, check the BUILD_OPTS line in the summary output contains the word `php5':

```
Options
  BUILD_OPTS               xml ssl imsg pldebug php5 pthreads
readline odbc
```

Then proceed to use `make` and `make install` as usual.

Once you have installed Virtuoso, you should copy the php5 hosting library, `libphp5.so` and extensions (`*.so`) into place in the `/usr/local/virtuoso/lib/virtuoso/hosting/` directory; the php.ini file goes in the database directory, by default `/usr/local/virtuoso/var/lib/virtuoso/db/` .

Enabling the PHP Hosting plugin

Now we must enable the hosting module; Virtuoso has a standard module system for the optional inclusion of various areas of functionality, enabled through the `Plugins' section of virtuoso.ini:

```
[Plugins]
LoadPath = ../hosting
Load1    = plain, wikiv
Load2    = plain, mediawiki
Load3    = plain, creolewiki
Load4    = plain, im
Load5    = attach, libphp5.so
Load6    = Hosting, hosting_php.so
```

Here we see the wiki markup parsers, an imagemagick plugin (for use in the ODS-Gallery application), and a link to the PHP library.

To host a PHP application within Virtuoso, it should suffice to enable the relevant lines (5 and 6) above.

Using PHP5 and Virtuoso (Commercial Edition)

OpenLink's commercial Virtuoso packages already come with PHP hosting support; to use your own library, simply copy the libphp5.so and/or any extensions required into place:

```
cp /usr/local/php5/lib/libphp5.so hosting/php/
```

Aside: failure to copy libphp5.so into the same place as is referenced by the [Plugins] section in virtuoso.ini will give rise to the following error:

```
16:28:50 INFO: { Loading plugin 7: Type `attach', file
`libphp5.so' in `/usr/local/virtuoso/lib/virtuoso/hosting'
16:28:50 ERROR:   FAILED  plugin 7: Unable to locate file }
```

Compiling your own PHP extensions

If you build your PHP as above, extra extensions can be added as required afterwards.

Our commercial builds of Virtuoso include the following PHP modules as standard:

apc	dbase	ldap	soap	sysvshm
bcmath	dom	mbstring	sockets	wddx
bz2	exif	mime_magic	sqlite	xmlreader
curl	ftp	openssl	sysvmsg	xmlrpc
dba	gd	shmop	sysvsem	xmlwriter
xsl				

In this case, all you need do to enable a module is uncomment it in the `php.ini` file (search for `Extensions'):

```
;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;
;;extension=apc.so
;;extension=bcmath.so
;;extension=bz2.so
;;extension=curl.so
;;extension=dbase.so
...
```

As a worked example, we continue by compiling the Alternative PHP Cache (APC) PHP extension for ourselves.

Having downloaded and unpacked the sources, run the `phpize` command installed as part of your new build of PHP:

```
zsh/scr, gentoo  5:06PM APC-3.1.2/ % /usr/local/php5/bin/phpize
Configuring for:
PHP Api Version:         20090626
Zend Module Api No:      20090626
...

zsh/scr, gentoo  5:05PM APC-3.1.2/ % ./configure --cache-
file=config.cache \
                --prefix=/usr/local/php5 \
                --enable-apc-mmap \
                --with-php-config=/usr/local/php5/bin/php-config
\
                --disable-static
...
appending configuration tag "CXX" to libtool
configure: updating cache config.cache
configure: creating ./config.status
config.status: creating config.h

zsh/scr, gentoo  5:07PM APC-3.1.2/ % make
/bin/sh /home/tim/C/APC-3.1.2/libtool --mode=compile cc  -I. \
-I/home/tim/C/APC-3.1.2 -DPHP_ATOM_INC -I/home/tim/C/APC-
3.1.2/include \
-I/home/tim/C/APC-3.1.2/main -I/home/tim/C/APC-3.1.2 \
-I/usr/local/php5/include/php -I/usr/local/php5/include/php/main
\
-I/usr/local/php5/include/php/TSRM -
I/usr/local/php5/include/php/Zend \
-I/usr/local/php5/include/php/ext -
I/usr/local/php5/include/php/ext/date/lib \
-DHAVE_CONFIG_H  -g -O2   -c /home/tim/C/APC-3.1.2/apc.c -o
apc.lo
...
```

```
zsh/scr, gentoo  5:08PM APC-3.1.2/ % sudo make install
Password:
Installing shared extensions:
/usr/local/php5/lib/php/extensions/no-debug-zts-20090626/
zsh/scr, gentoo  5:08PM APC-3.1.2/ %
```

To use the extension,

- copy it into the `../hosting/php/` directory
- edit the php.ini file and search for `Extensions'. Add and remove lines there as required.

Aside: note that if you fail to copy the libphp5.so and/or extensions (*.so) into the right directory, you'll have a mismatch with the prebuilt versions, leading to this error:

```
16:05:04 { Loading plugin 7: Type `Hosting', file
`hosting_php.so' in `../hosting'
16:05:04   Hosting version 3034 from OpenLink Software
16:05:04   PHP engine version 5.2.7-dev
16:05:04 PHP Warning:  PHP Startup: apc: Unable to initialize
module
Module compiled with module API=20060613
PHP    compiled with module API=20090626
These options need to match
 in Unknown on line 0
***
*** Please fix the above issue(s) before trying again.
```

Testing

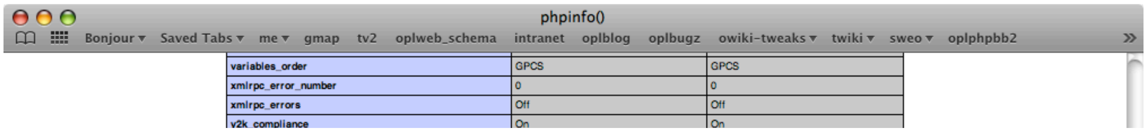
As with any other PHP environment, it makes sense to test the setup:

Create a file (say `info.php`) in the `../vsp/` directory containing the text:

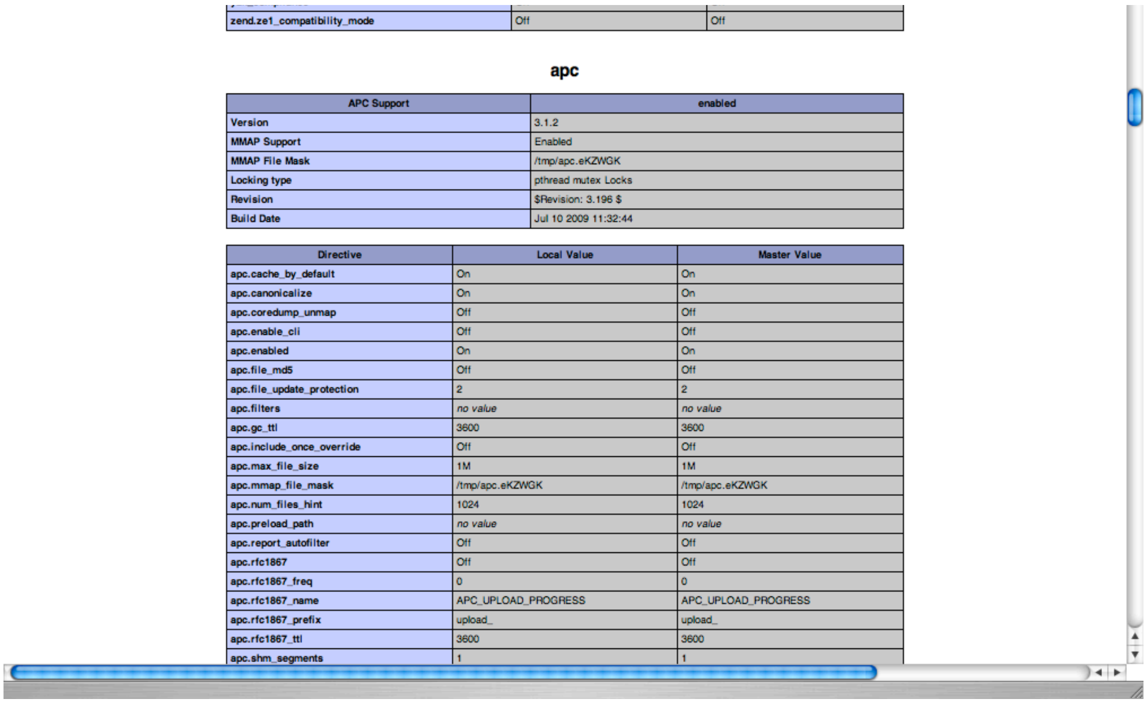
```
<?php
  phpinfo();
?>
```

This file can be downloaded: info.php

Copy this file into the ../vsp/ directory of your Virtuoso installation and point your browser at the relevant URL - eg http://localhost:8890/info.php and see what the output of phpinfo() is. You should see a section with title corresponding to the module you compiled - here `APC':



variables_order	GPCS	GPCS
xmlrpc_error_number	0	0
xmlrpc_errors	Off	Off
y2k_compliance	On	On



zend.ze1_compatibility_mode	Off	Off

apc

APC Support	enabled
Version	3.1.2
MMAP Support	Enabled
MMAP File Mask	/tmp/apc.eKZWGK
Locking type	pthread mutex Locks
Revision	$Revision: 3.196 $
Build Date	Jul 10 2009 11:32:44

Directive	Local Value	Master Value
apc.cache_by_default	On	On
apc.canonicalize	On	On
apc.coredump_unmap	Off	Off
apc.enable_cli	Off	Off
apc.enabled	On	On
apc.file_md5	Off	Off
apc.file_update_protection	2	2
apc.filters	*no value*	*no value*
apc.gc_ttl	3600	3600
apc.include_once_override	Off	Off
apc.max_file_size	1M	1M
apc.mmap_file_mask	/tmp/apc.eKZWGK	/tmp/apc.eKZWGK
apc.num_files_hint	1024	1024
apc.preload_path	*no value*	*no value*
apc.report_autofilter	Off	Off
apc.rfc1867	Off	Off
apc.rfc1867_freq	0	0
apc.rfc1867_name	APC_UPLOAD_PROGRESS	APC_UPLOAD_PROGRESS
apc.rfc1867_prefix	upload_	upload_
apc.rfc1867_ttl	3600	3600
apc.shm_segments	1	1

Bringing it together: PHP, web-apps and data

Given that Virtuoso provides not just a webserver with PHP hosting ability, but the whole backend ODBC-compatible database environment as well all in one package, it makes sense to be able to connect back to the hosting RDBMS instance from within your PHP application/script.

To ease this process, Virtuoso's PHP-hosting module adds support for a parameter to identify the "local DSN"; normally there is a DSN configured called "Local Virtuoso" but in case that has been changed, or some other instance should be used, there are parameters in php.ini that can be specified:

```
[Virtuoso]
virtuoso.logging = On
virtuoso.local_dsn = Local Virtuoso
virtuoso.allow_dba = 0
```

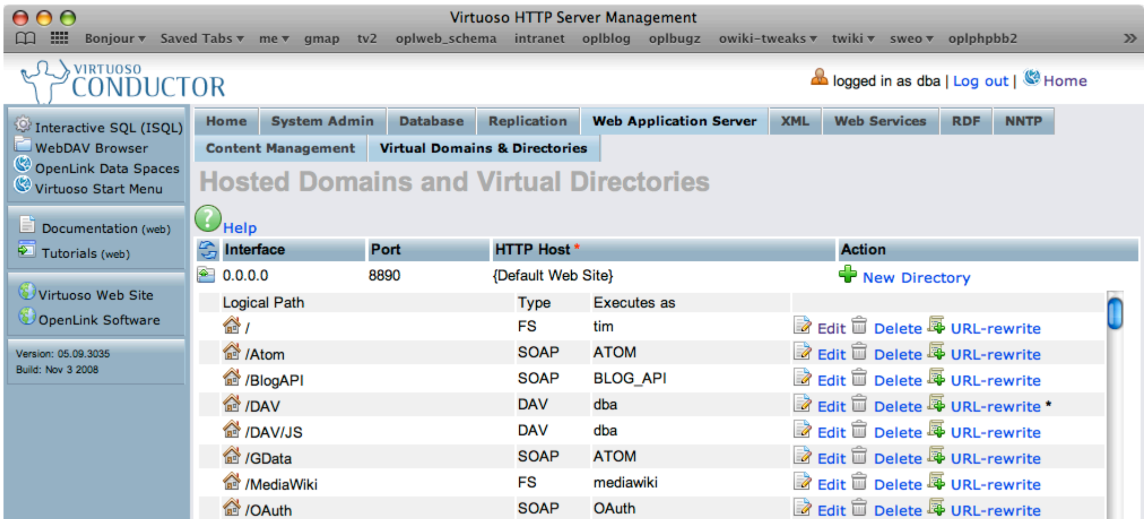
The `virtuoso.allow_dba' property controls whether the administrative dba user can be used as the username to connect to the ODBC data-source; by default, it cannot, as a security measure.

Using this in your application is simple. First, wherever you would previously have had to compose an ODBC connection-string, you can now just drop this function into place:

```
<?php
  $con=odbc_connect(__virt_internal_dsn(), "", "");
?>
```

The results of the `__virt_internal_dsn()` function may be configured through the Virtuoso Conductor:

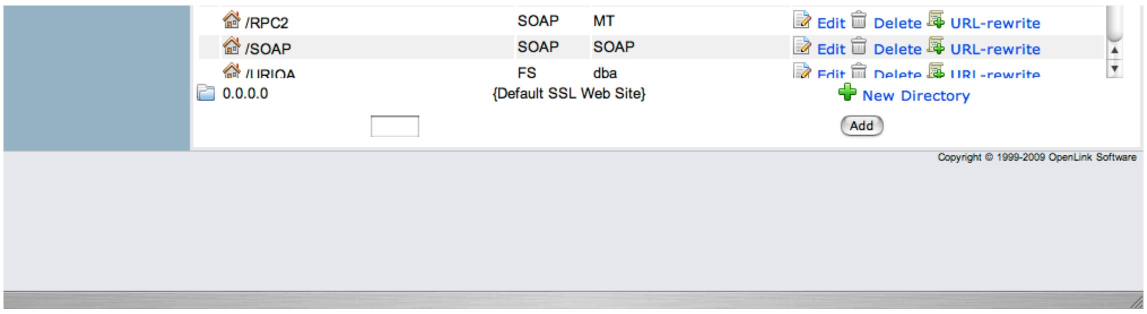
- point your web-browser at http://localhost:8890/conductor/ (or hostname as appropriate)
- log in as dba, and navigate to "web application server" and "Virtual Domains and Directories"
- expand the relevant hostname (or default) and click `edit' on the directory corresponding to the URL in which your PHP application resides



Hosted Domains and Virtual Directories

Interface	Port	HTTP Host *	Action
0.0.0.0	8890	{Default Web Site}	New Directory

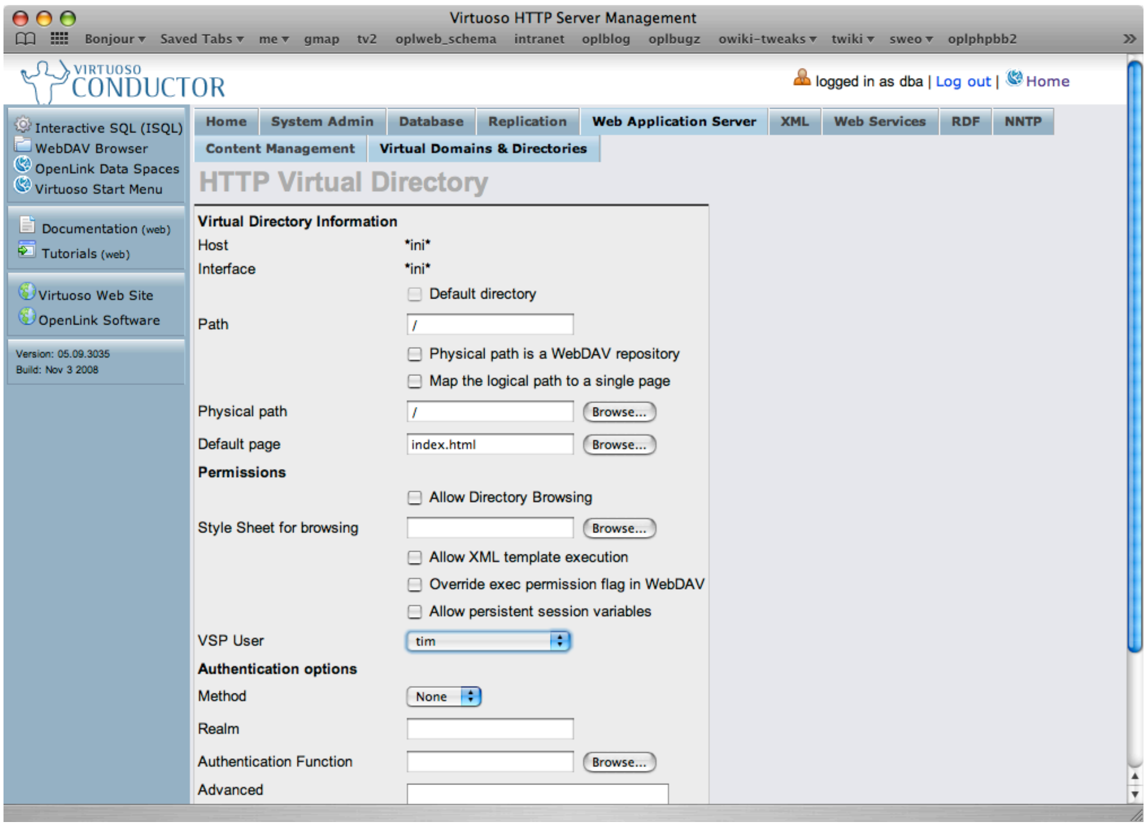
Logical Path	Type	Executes as	
/	FS	tim	Edit Delete URL-rewrite
/Atom	SOAP	ATOM	Edit Delete URL-rewrite
/BlogAPI	SOAP	BLOG_API	Edit Delete URL-rewrite
/DAV	DAV	dba	Edit Delete URL-rewrite *
/DAV/JS	DAV	dba	Edit Delete URL-rewrite
/GData	SOAP	ATOM	Edit Delete URL-rewrite
/MediaWiki	FS	mediawiki	Edit Delete URL-rewrite
/OAuth	SOAP	OAuth	Edit Delete URL-rewrite



/RPC2	SOAP	MT	Edit Delete URL-rewrite
/SOAP	SOAP	SOAP	Edit Delete URL-rewrite
/URIQA	FS	dba	Edit Delete URL-rewrite
0.0.0.0	{Default SSL Web Site}		New Directory

Copyright © 1999-2009 OpenLink Software

- Click "Edit"
- scroll down to set the VSP user



Virtuoso HTTP Server Management

Bonjour Saved Tabs me gmap tv2 oplweb_schema intranet oplblog oplbugz owiki-tweaks twiki sweo oplphpbb2
VIRTUOSO CONDUCTOR
logged in as dba | Log out | Home
Interactive SQL (ISQL)
WebDAV Browser
OpenLink Data Spaces
Virtuoso Start Menu
Documentation (web)
Tutorials (web)
Virtuoso Web Site
OpenLink Software
Home System Admin Database Replication Web Application Server XML Web Services RDF NNTP
Content Management Virtual Domains & Directories

Version: 05.09.3035
Build: Nov 3 2008

HTTP Virtual Directory

Virtual Directory Information

Host	*ini*
Interface	*ini*
	☐ Default directory
Path	/
	☐ Physical path is a WebDAV repository
	☐ Map the logical path to a single page
Physical path	/ Browse...
Default page	index.html Browse...
Permissions	
	☐ Allow Directory Browsing
Style Sheet for browsing	Browse...
	☐ Allow XML template execution
	☐ Override exec permission flag in WebDAV
	☐ Allow persistent session variables
VSP User	tim
Authentication options	
Method	None
Realm	
Authentication Function	Browse...
Advanced	

- choose an appropriate user as which your PHP application should run

The function verifies that the Virtual Directory has been properly configured with a user valid for SQL logins and not disabled, else it logs a message and returns FALSE.

Permissions Note: If you are deploying an application that runs as a specific user, permission must be granted for it to access tables.

Additionally, for deploying an application package (eg as a VAD of your own), you can export the virtual-directory specification and have it create your endpoint, including specifying a user as which it should run, all executable as SQL/PL from within your package.

```
    -- Create and use the MYAPP schema
    use MYAPP;

    -- Create a user myapp
    db.dba.user_create (
        'myapp',                                -- Account name
        uuid (),                                -- Random UUID as
password
        vector ('LOGIN_QUALIFIER', 'MYAPP',
                'SQL_ENABLE', 1,
                'DAV_ENABLE', 0,
                'FULL_NAME', 'MYAPP Administrator'));
    db.dba.vhost_remove (lpath=>'/myapp');
    db.dba.vhost_define (
        lpath=>'/myapp',
        ppath=>'/testapp',
        is_dav=>0,
        is_brws=>0,
        vsp_user=>'myapp',
        def_page=>'myapp.php');
```

The Pièce de Résistance

Finally, putting it all together, a simple sample PHP script: home-grown build of PHP, dynamic RDF data (created using an RDF view above), accessed using SPARQL over SQL, via ODBC using the `__virt_internal_dsn()` function:

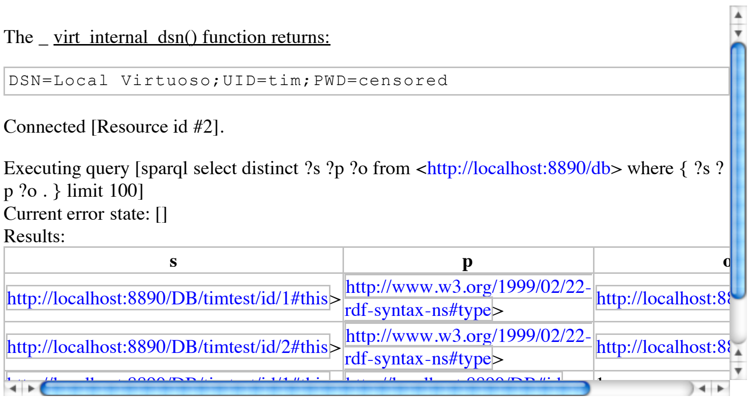
Please re-read the previous section "putting it all together", above, and note the requirements for permissions to be granted on relevant tables to the user as which your PHP application will run.

```
grant select on rdf_quad to tim;
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>PHP ODBC test</title>
  <style type="text/css">
body { padding: 1em; }
pre  { padding: 0.3em; background-color: #e0e0e0; border: solid
1px silver }
  </style>
</head>

<body>
  <h1>Test PHP / ODBC / Virtuoso connection</h1>
  <hr />

  <p>The __virt_internal_dsn() function returns:</p>
<pre>


<?php
  ini_set("odbc.default_cursortype", "0");

  $con=odbc_connect(__virt_internal_dsn(), "", "");
  if($con) {
    printf("<p>Connected [%s].\n</p>\n", $con);

    $query="sparql select distinct ?s ?p ?o from
<http://localhost:8890/db>
    where { ?s ?p ?o . } limit 100";

    print "Executing query [$query]<br />\n";
    $rs = odbc_exec($con, $query);

    $err=odbc_errormsg($con);
    print "Current error state: [$err]<br />\n";

    print "Results:<br />\n";
    odbc_result_all($rs, "border=2");
    odbc_close($con);
  } else {
    print "<p>Failed to connect!</p>\n";
  }
?>

?>

  <hr />
</body>
</html>
```

This script can be downloaded here: sparql-retrieve.php

Results:



The _ virt_internal_dsn() function returns:

```
DSN=Local Virtuoso;UID=tim;PWD=censored
```

Connected [Resource id #2].

Executing query [sparql select distinct ?s ?p ?o from <http://localhost:8890/db> where { ?s ?p ?o . } limit 100]
Current error state: []
Results:

s	p	o
http://localhost:8890/DB/timtest/id/1#this>	http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	http://localhost:8
http://localhost:8890/DB/timtest/id/2#this>	http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	http://localhost:8
[illegible]	[illegible]	[illegible]