Subject: Re: SQLSRV Driver
Posted by rafs on Mon, 11 May 2015 06:53:55 GMT
View Forum Message <> Reply to Message

Tony! Thanks for replying. I am a huge fan. Thank you, so much, for putting this project together and out for the public. The documentation is very helpful. I appreciate all of your insightful explanations about your design decisions, and critique of other more bloated alternatives, (with cutting comments for entertainment value). I am hoping to get deeper into it, so I can modify it for my project.

My project is to provide a script that can extract, transform, and load (ETL) to and from the production MSSQL2008 DB to a newly redesigned schema running on Linux/Postrgres 9.4, in both directions--easing the transition to the new database by allowing synchronization to the old database until all of the dependent services can be migrated to the new. I do not need the web UI components, but the import database (from both pgsql and sqlsrv), and then export PHP code part is what I am after, so I can use the classes.

My current sticking point is when I go to "Dictionary >> Databases >> Import Database". I am using the localhost/mysql db to store audit, dict, menu, and workflow. This works. So now I want to connect to two other databases, MSSQL2008, and PGSQL, in order to import them. I assumed that I would just add the two extra dbs into the '$servers' array. I started by just trying to add the MSQL2008 DB to the servers array, but then I get an error, "Fatal Error: Entry missing for database 'audit' in SERVERS array in CONFIG file (# 256)". So, then it seems that the mysql connection needs to be in $servers array as well. (I left out the pgsql db for the time being.) So, when I add localhost/mysql db as the first entry (see below config) I am getting a driver error, below when trying to import MSSQL2008 db. (A capture using wireshark shows that there is no TCP connection even attempted by the localhost to the MSSQL2008 remote host, 10.100.150.86, despite the warning below). (I tried both IPADDR and DNS hostname.)

Here is my warning, via webpage:
Fatal Error: [Microsoft][ODBC Driver 11 for SQL Server]Named Pipes Provider: Could not open a connection to SQL Server [2].
[Microsoft][ODBC Driver 11 for SQL Server]Login timeout expired
[Microsoft][ODBC Driver 11 for SQL Server]A network-related or instance-specific error has occurred while establishing a connection to SQL Server. Server is not found or not accessible. Check if instance name is correct and if SQL Server is configured to allow remote connections. For more information see SQL Server Books Online. (# 2)
Error in line 550 of file 'C:\xampp\radicore\includes\dml.sqlsrv.class.inc'.
PHP_SELF: /radicore/dict/database(add1)import.php
CURRENT DIRECTORY: C:\xampp\htdocs\radicore\dict
SERVER_ADDR: ::1
SERVER_NAME: localhost
HTTP_HOST: localhost
User Id: MGR
Role Id: GLOBAL
REMOTE_ADDR: ::1
REQUEST_URI: /radicore/dict/database(add1)import.php?session_name=menu21

Actually, I am not sure which driver is the one I should be connecting to. Right now, I am loading in php.ini these two:

php.ini
---
extension=php_sqlsrv_56_ts.dll
extension=php_pdo_sqlsrv_56_ts.dll

In addition to the two DLLs listed above, I have the following drivers installed, and all seem to work when testing via the Admin Tools >> ODBC Data Source Admin tool.
ODBC Driver 11 for SQL Server (MSODBCSQL11.DLL)
SQL Server (SQLSRV32.DLL)
SQL Server Native Client 11.0 (SQLNCLI11.DLL)

From command line, I can see that 'sqlsrv' is loaded.

I am not sure what to make of the sqlsrv-specific fields. I tried adding them and using special 'User DSN' and 'System DSN' names for serverName, as well as using the servers proper DNS name, but it didn't help:

```
    $servers[1]['serverName']     = 'MYUSERDSNNAME';
    $servers[1]['SQLSRV_schema']  = 'dbo';
    $servers[1]['connectionInfo'] = array('CharacterSet' => 'UTF-8',
                                          'ReturnDatesAsStrings' => true)
```

Here is my config.inc:

```
<?php
//  ************************************************************ *****************
// Copyright 2003-2005 by A J Marston <http://www.tonymarston.net>
// Copyright 2006-2014 by Radicore Software Limited <http://www.radicore.org>
//  ************************************************************ *****************

// This file contains database access details and values for optional constants/globals
$GLOBALS['dbms']   = 'mysql';      // database engine is MySQL
//$GLOBALS['dbms']   = 'pgsql';      // database engine is PostgreSQL
//$GLOBALS['dbms']   = 'oracle';     // database engine is Oracle
//$GLOBALS['dbms']   = 'sqlsrv';     // database engine is SQL Server (Microsoft)

if ($GLOBALS['dbms'] == 'oracle') {
    $GLOBALS['dbhost'] = '//localhost/xe';

} elseif ($GLOBALS['dbms'] == 'sqlsrv') {
    // JOIN clauses in SQL Server require 'dbname.<schema>.tblname'
    $GLOBALS['SQLSRV_schema'] = 'dbo';

} elseif ($GLOBALS['dbms'] == 'pgsql') {
```

```
    // NOTE: with MYSQL there are tables withing databases, and within a single connection
    // it is possible to access tables in any database.
    // But with PostgreSQL there are tables within schemas within databases, and within
    // a single connection it is only possible to access a single database, but any number
    // of schemas within that database.
    // When using PostgreSQL you must supply a value for $PGSQL_dbname for the single
database
    // connection, and what is known as 'dbname' to MySQL becomes 'schema' to PostgreSQL.
    $GLOBALS['dbhost']       = 'localhost';
    $GLOBALS['PGSQL_dbname'] = 'radicore';

} else {
    $GLOBALS['dbhost'] = 'localhost';
} // if

// NOTE: $dbprefix is for my web host (shared) where my databases are prefixed
// with my account name to keep them separate from other accounts.

if (preg_match('/^(127.0.0.1|localhost|desktop|laptop)$/i', $_SERVER['SERVER_NAME'])) {    //
this is for my PC
    if ($GLOBALS['dbms'] == "sqlsrv") {
        $GLOBALS['serverName']     = '(local)';
        $GLOBALS['connectionInfo'] = array('CharacterSet' => 'UTF-8',
                                           'ReturnDatesAsStrings' => true);
        $GLOBALS['SQLSRV_schema']  = 'dbo';
    } else {
        $GLOBALS['dbusername'] = 'root';
        $GLOBALS['dbuserpass'] = '';
        $GLOBALS['dbprefix']   = null;
        $GLOBALS['dbport']     = null;
        $GLOBALS['dbsocket']   = null;
    } // if
    // set these only if secure HTTPS protocol is available on your server
    $GLOBALS['http_server']  = '';
    $GLOBALS['https_server'] = '';
    $GLOBALS['https_server_suffix'] = '';
} else {
    // this is for my shared web host
    $GLOBALS['dbusername'] = '****';
    $GLOBALS['dbuserpass'] = '****';
    $GLOBALS['dbprefix']   = '****';
    $GLOBALS['dbport']     = null;
    $GLOBALS['dbsocket']   = null;
    // set these only if secure HTTPS protocol is available on your server
    $GLOBALS['http_server']  = '';
    $GLOBALS['https_server'] = '';
    $GLOBALS['https_server_suffix'] = '';
} // if
```

```
// this demonstrates the multi-server option (see FAQ92)
if (preg_match('/^(127.0.0.1|localhost|desktop|laptop)$/i', $_SERVER['SERVER_NAME'])) {
    global $servers;
    // server 0
    $servers[0]['dbhost']     = 'localhost';
    $servers[0]['dbengine']   = 'mysql';
    $servers[0]['dbusername'] = 'root';
    $servers[0]['dbuserpass'] = '';
    $servers[0]['dbport']     = '';
    $servers[0]['dbsocket']   = '';
    $servers[0]['dbprefix']   = '';
    $servers[0]['dbnames']    = '*';
    $servers[0]['ssl_key']    = '';
    $servers[0]['ssl_cert']   = '';
    $servers[0]['ssl_ca']     = '';
    $servers[0]['ssl_capath'] = '';
    $servers[0]['ssl_cipher'] = '';
    // server 1
    $servers[1]['dbhost']        = '10.100.150.86';
    $servers[1]['dbengine']      = 'sqlsrv';
    $servers[1]['dbusername'] = 'myusername';
    $servers[1]['dbuserpass'] = 'mypassword';
    $servers[1]['dbport']     = '';
    $servers[1]['dbsocket']   = '';
    $servers[1]['dbprefix']      = '';
    $servers[1]['dbnames']       = 'mydbname';
//    $servers[1]['serverName']     = 'MYUSERDSNNAME';
//    $servers[1]['SQLSRV_schema']  = 'dbo';
//    $servers[1]['connectionInfo'] = array('CharacterSet' => 'UTF-8',
//                                          'ReturnDatesAsStrings' => true);
} // if

// set this to true if you want all XSL transformations to be done by the client
// (NOTE: only valid if supported by your client browser)
$GLOBALS['XSLT_client_side'] = false;

// set this to TRUE to write all sql queries to file 'sql/<script_id>.sql' (for debugging)
$GLOBALS['log_sql_query'] = false;

// set this to TRUE to write all XML documents to file 'xsl/<script_id>.xml' (for debugging)
$GLOBALS['log_xml_document'] = false;

// set date format - choose one of the following (may be overridden on the MNU_LANGUAGE
table)
$GLOBALS['date_format'] = 'dmy';        //  '21 Jan 2001'
//$GLOBALS['date_format'] = 'mdy';        // 'Jan 21 2001'
//$GLOBALS['date_format'] = 'ymd';        // '2001 Jan 21'
```

```
//$GLOBALS['date_format'] = 'dd.mm.yyyy';  // '21.06.2001'
//$GLOBALS['date_format'] = 'dd/mm/yyyy';  // '21/06/2001'
//$GLOBALS['date_format'] = 'dd/mm/yy';    // '21/06/01'
//$GLOBALS['date_format'] = 'yyyy-mm-dd';  // '2013-03-02'

// use HTTPS for all web pages
$GLOBALS['use_https'] = false;

// define this only if content such as CSS or JS files are not to be referenced using HTTPS
//define('HTTPS_NOT_FOR_FILES', true);

// use to access files in the 'classes/custom-processing/<project>/' folder
//$GLOBALS['project_code'] = 'unknown';

// converts datetime between client and server timezones (PHP version >= 5.2)
//$GLOBALS['server_timezone'] = 'Europe/London';

// defines email address(es) to which any error_log will be sent
//define('ERROR_LOG_MAILTO', 'me@domain.com, someone.else@domain.co.uk');

// defines directory into which error logs will be written
define('ERROR_LOG_DIR', $_SERVER['DOCUMENT_ROOT'].'/radicore/error_logs');

// defines Return-Path to be used on all outgoing emails
//define('MAIL_RETURN_PATH', 'postmaster@domain.com');

// this causes the errorHandler to automatically activate the DBG debugger
//define('DEBUGBREAK_ON_ERROR', true);

// causes the logstuff() function to write details to the log file
//define('LOGSTUFF', true);

// causes the database session handler to be ignored
//define('NO_SESSION_HANDLER', true);

?>
```