Chapter 12. December 2007

Welcome to the December 2007 edition of IBM Information Server Developer’s Notebook. This month we answer the question;

> How do I install IBM Information Server on RedHat Linux Advanced Server and the MySQL database server?
>
> *Excellent question! Linux is the favorite operating system of this month’s author, having enjoyed watching Linux grow up from a student project, to a largely PC operating system, to now a major force in the data center, especially in grid configurations. IBM Information Server works especially well and in a highly cost effective manner on Linux; 32 bit or 64 bit, Intel or Series/z chips, RedHat or SuSE, SMP (symmetric multi-processor) or MPP (massively parallel, shared nothing architectures) and grid.*

In short, we are going discuss all of the relevant terms and technologies above, and provide examples that detail how to deliver this functionality using IBM Information Server.

Software versions

All of these solutions were *developed and tested* on IBM Information Server version 8.01, FixPak 1A, on the Microsoft Windows XP/SP2 platform. In the case of a Unix or Linux dependency, examples were *developed and tested* on a RedHat Linux Advanced Server 4 (RHEL 4) FixPak U6 32 bit SMP server (Linux kernel version 2.6.9-67.EL-smp), while still using IBM Information Server client programs on Microsoft Windows XP/SP2.

IBM Information Server allows for a single, consistent, and accurate view of data across the full width of the corporate enterprise, be it relational or non-relational, staged or live data. As a reminder, the IBM Information Server product contains the following major components;

> WebSphere Business Glossary Anywhere™, WebSphere Information Analyzer™, WebSphere Information Services Director™, WebSphere DataStage™, WebSphere QualityStage™, WebSphere Metadata Server and Metabridges™, WebSphere Metadata Workbench™, WebSphere Federation Server™, Classic Federation™, Event Publisher™, Replication Server™, Rational Data Architect™, DataMirror Transformation Server™, and others.

Obviously, IBM Information Server is a large and capable product, addressing many strategic needs across the enterprise, and supporting different roles and responsibilities.

12.1 Terms and core concepts

This month's question, as it is written above, really covers two independent topics; IBM Information Server installation, and then post install product configuration. The primary reason to make this distinction, is to reinforce that we are using two distinct problem solving techniques, one first to install (read and follow instructions carefully with comprehension of what you are doing, and take notes), and then another to configure (decompose the problem into smaller parts and verify everything).

Of the two topics in this document (install then configure), we are going to cover configuring data sources and MySQL in great detail. Installation is really a much easier problem, with good supporting documentation and technical support services from IBM. Installation is the topic we cover first.

IBM Information Server installation

Even though this month's author has installed IBM Information Server 80 or more times on numerous operating systems, we still print the *IBM Information Server Planning, Installation, and Configuration Guide* every time we do an install. We read each section twice, draw on the sections we were instructed to complete, cross out the sections that don't pertain to us, we write down the values we supplied, and write down the values we were given in response (port numbers and the like). A successful install on even the largest systems can take as little as 20 minutes to complete, and its not hard to do. If you don't confirm the operating system parameters as instructed, and test the steps you were asked to complete prior to or during the install, a successful installation can take days.

Note: Installing IBM Information Server really isn't that hard. Somewhere between the difficulty of installing the Macromedia Flash plug in into your Web browser, and implementing an SAP ERP system, lies the challenge of installing IBM Information Server. You would never implement SAP without training and assistance, and you never think twice about installing a plug in with a single click. Its the folks who just click through the IBM Information Server install, without really checking or understanding what they are doing, who have difficulty.

The *IBM Information Server Planning, Installation, and Configuration Guide* is located on the product installation media (compacted file, DVD, or CD-ROM). From the root directory of the install media, this guide is located in; /is-ia-suite/InstallationGuide.pdf. The current version is marked, *Revised March 2007*. Updates to this document are available on line at;

http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp?topic=/com.ibm.swg.im.iis.productization.iisinfsv.install.doc/concepts/wsisinst_overview_container.html

The online document differs from the document that comes with the installation media in at least two ways;

- The online document is burst into multiple sections.
- The online document has been updated since March 2007.

Still, we like a hard copy of this document best (any hard copy), with instructions that flow sequentially, one page following the next. It is easier to complete each section in order, without missing steps.

Installation overview

The following offers an overview of the IBM Information Server installation in general, and also with specific comments towards Linux/Unix;

- IBM Information Server installation includes graphical two-tier rich client programs; hence, there is a server side install, and a client side install. We always install the server, then the client.

  The server side install and client side install are done from two different files, to account for differences in the client and server operating systems. Both files, both installers, use a Java, graphical or command line, based install program. We always use the graphical installer.

  While your supplied files names may differ, the files we install from are named-

  - RedHat Linux Advanced Server (the server side), InfoSvr_v8.0.1_RHEL_EN.tar.

    (While they are both Linux, RedHat and SuSE have different installation requirements, and each install from a different binary installation program.)

  - Microsoft Windows XP/SP2 (the client side), InfoSvr_Client_v8.0.1_Win_EN.zip.

  The server side install will require an XML formatted license file. The client side does not. License files are not operating system specific.

Note: We are discussing Linux here, however, we will mention that on Windows you can install both the server side and client side programs in one step, one install. The Windows server side install distribution includes the graphical client programs.

- The server side install involves three major components (three major software products that are bundled into the IBM Information Server install). Installed in order, these are;
  - A relational database server.
  - A Java/J2EE application server.
  - IBM Information Server proper, and the DSEngine.
- The relational database server hosts the shared IBM Information Server meta data database, also referred to as the meta data repository.
  - The default name of the meta data database is xmeta. Another database, default name iadb, is used by the Information Analyzer component of IBM Information Server.
  - IBM DB2 UDB V 9.1 32 bit software is supplied with the IBM Information Server software, and will automatically installed by default. You can also use specific versions of Oracle or Microsoft SQL/Server to host the meta data and Information Analyzer databases.

Note: At this time, you can not use MySQL to host the IBM Information Server meta data repository. Only specific versions of DB2 UDB, Oracle or SQL/Server can host the meta data repository.

  - The DB2 UDB software install is included in the IBM Information Server install program. Oracle and SQL/Server have to have specific pre-work done prior to the IBM Information Server install.

    Generally with Oracle or SQL/Server, you have to make and specifically tune the server instance/database, as well as make the database, users, and related.
  - As a general rule, 32 bit programs run on 64 bit operating systems. However, there are also support and installation dependencies to consider when running 32 bit programs on 64 bit operating systems.

    The net/net of this sub-topic is, if you are running on 64 bit Linux, and you are using a DB2 UDB server as the host for the meta data repository, you must install a 64 bit version of the DB2 UDB server. (This is a DB2 UDB supported platform issue.)

    The 64 bit version of DB2 UDB server is installed prior to beginning the IBM Information Server install.

    A 64 bit version of DB2 UDB server is not on the IBM Information Server installation media. You must contact IBM support or your sales representative to get a (free) copy of 64 bit DB2 UDB.

  - While this embedded version of the DB2 UDB server is licensed only for use with IBM Information Server, the DB2 UDB server may be installed on a host separate from IBM Information Server, per your license agreement. Most commonly we do this in clustered or grid configurations.
  - Oracle and SQL/Server work perfectly fine as hosts for the meta data repository. Every support case we have seen identifying Oracle or SQL/Server as the host of the meta data repository, has involved resource inadequacy; for example, a disk fills inside the Oracle database server, and this impacts the operation of IBM Information Server. The first notice we tend to observe in this case is an IBM Information Server related error, because IBM Information Server is dependent on Oracle as the host of the meta data repository.

    While IBM Information Server is database agnostic, and we are not beholden to DB2 UDB is any way, we will say that the DB2 UDB server works pretty well. DB2 UDB has an automatic maintenance and self tune mode, which is how the meta data repository configures itself; To a large extent, DB2 UDB is largely set it and forget it.

– IBM Information Server is inherently Web services and Service Oriented Architecture aware and enabled. A Java/J2EE compliant application server is the second major component included in the IBM Information Server install.

  - Similar to DB2 UDB, a 32 bit version of IBM WebSphere Application Server (WAS) V6 is bundled with the IBM Information Server install program.

    As with DB2 UDB, if you are on 64 bit Linux, you must install a 64 bit version of WAS prior to beginning the IBM Information Server install. As with DB2 UDB, the 64 bit version of WAS is not included on the IBM Information Server installation media, and you must contact IBM support or your sales representative to get a (free) copy of 64 bit WAS.

The 64 bit version of WAS is installed prior to beginning the IBM Information Server install.

Note: IBM Information Server is comprised of a number of base (always present and installed) components, and a number of optional components. Base components include the meta data repository, and the Java/J2EE compliant application server.

Even if you (own) zero IBM Information Server Web related components (like WISD, WebSphere Information Services Director, formerly known as RTI, Real Time Interface), you will still install a Java/J2EE compliant application server. IBM Information Server uses Web services internally to perform the meta data repository service fulfillments to clients and run time programs.

While there has been public mention to support future Java/J2EE compliant application servers other than WAS, WAS is currently the only supported application server.

- While this embedded version of WAS is licensed only for use with IBM Information Server, WAS may be installed on a host separate from IBM Information Server, per your license agreement. Most commonly we do this in clustered or grid configurations.
- The WAS server instance is always password protected (not the default condition, and uncommon for development only areas).

  Generally, the WAS Administrative Console (a Web browser based application) is accessible at the IP on which WAS is operating, and port 9060. The username and password are those as specified in Table 12-1, below, under user wasadmin.

– IBM Information Server proper, and the DSEngine, constitute the last portion of the installation process.

Installation, things to watch for

There are pre-requisites, settings that must be made or confirmed on the server side especially, before starting the IBM Information Server server side install. These include;

– You must be on a supported operating system. RedHat Linux Advanced Server V4 is one supported operating system, 32 or 64 bit.

  If you successfully install on another distribution of Linux that is not supported, it is not supported. (For example, SuSE Enterprise Server is supported, but SuSE Enterprise Desktop is not.) Currently the supported platform matrix is available on line at, http://www-1.ibm.com/support/docview.wss?uid=swg27008923#platformlinux

We’d recommend you also contact IBM sales or technical support for a more current list, as even the on line guide above tends to lag by a few weeks.

– There is also a specific type and version of C++ language compiler, and compiler options that must be in effect. This too is listed on the supported platforms matrix, listed above.

Note: The C++ compiler is required more for operation of IBM Information Server than installation. As an install verification test related to the C++ compiler, make a IBM Information Server DataStage Parallel Job with a Transformer operator; if that Job compiles and runs, generally you are done.

– Some persons have observed that a RedHat Linux Advanced Server Version 3 (V3) that was officially upgraded to Version 4 (V4), is not equivalent to newly installed V4, and that the installation of IBM Information Server would not succeed. Based on the number of reports of this type, we recommend always starting with a newly installed RHEL V4.

– While IBM Information Server officially supports 32 bit or 64 bit Linux, the 64 bit installs seem to be harder, as these operating system libraries (32 bit or 64 bit) have often been incorrectly placed or made reference to.

The Unix/Linux file(C) command will report what type of file a given target is, and 32 bit or 64 bit. Example, “file myfile.so”.

– The IBM Information Server installation guide states 8 or more Linux kernel parameters that must be set. In effect, the installation is asking for *available* kernel resource. If the kernel parameter/resource is set, but at the same time it is all being currently consumed by some other server software, the IBM Information Server install will fail.

Consider setting these kernel parameters much higher than requested; going from 1000 open files per user to 4000 consumes very little additional kernel resource (memory), something in the order of kilobytes.

And while we are discussing open files, some Linux shells artificially limit the number of open files, regardless of the kernel setting; this is true for root as well as normal users. This limit needs to be checked as well.

Note: You can read most RedHat Linux kernel parameters from the /proc/sys/kernel/ directory. A number of ASCII text files in this directory contain the setting (value) for the same named file. For example, a file named semmni, contains a single integer value which is the current Linux kernel parameter, semmni.

With the Linux kernel version 2.2.1 and above, most kernel parameters can be set dynamically. On RedHat Linux, we prefer to edit the /etc/sysclt.conf file, and then reboot. Changes made here are also persistent across system reboots.

Lastly, the Linux shell being discussed above is bash(c). You can view settings of this type as the output of a, "ulimit -a" command. To increase the number of open files as allowed by the bash(c) shell, execute a "ulimit -n ##", as in "ulimit -n 2048", to set this value to 2048.

- While the installer comes with its own Java virtual machine (JVM) used only during the install, you should manage your PATH environment variable to be certain that this is the JVM used for the install.

  The comment is directed more for installation of the FixPak, and is documented in the install guide for that package.
- When adding operating system users and groups, be certain of the following;
  - After you make the users, test the login of every user made. Be certain that you have a shell, that you are not prompted to change the password upon log in, that you can actually log in, etcetera.
  - Consider disabling password aging.

    If you stay ahead of password aging (always change passwords before you receive warnings or they expire), fine and no worries. It is the prompting to change passwords, and actual expired passwords that can cause any server, including IBM Information Server, to have grief.

    On Windows, at least, if any passwords expire, not only do you have fix the offending passwords, you have to take corrective administrative action within IBM Information Server. As a support keyword on the expired password topic, we offer the IBM Information Server command, AppServerAdmin.bat. (As a BAT file, obviously that is a Windows platform command only.) Do not execute this command without further study.

- User and group names are actually important. On Unix/Linux at least, these names (roles) have inter-dependencies. We use,

Table 12-1 Suggested user and group names.

Function	User name	Group Name
xmeta db owner	xmeta	xmeta
DB2 server instance owner	db2inst1	db2iadm1
DB2 fenced user	db2fenc1	db2fadm1
DB2 DAS server owner	dasusr1	dsadm1
WAS admin	wasadmin	wasadmin
iadb owner	iauser	iauser
DS engine admin	dsadm	dstage
Info Server admin	iisadmin	dstage ** Same primary Linux group as “dsadm”.

The most important element of Table 12-1 is to have the dsadm and iisadmin users share the same primary operating system group. (Primary operating system group as in, the first group listed for each in /etc/passwd and /etc/group.)

This condition will allow both dsadm and iisadmin to make new Projects through the Project Administrator, a graphical two-tier client program. Net/net, you only have that ability if you are part of the same primary group as dsadm.

As stated during the installation process, and in the installation guide, wasadmin can also be the iisadmin. We recommend using two separate user names.

Note: On the topic of users and groups-

Every software server ever created has its own hierarchy of users and groups, roles and permissions, property files (profiles and environment variables), and the like. Some have an all powerful super user, some don’t. Some validate users via LDAP, via the operating system password file, via their own internal password file, or a combination of any/all of the above.

On this topic, IBM Information Server is no different.

To successfully install and operate IBM Information Server, you will need at least a basic understanding of its roles and permissions. In some cases, you will not get what you consider an obvious and useful diagnostic message if you try to perform a given task without permission or as the wrong user.

- Be certain that the Linux host name resolves cleanly; meaning, you can ping the IP associated with the host name, it is addressable if this server is to be accessible by clients (the IP is not 127.0.0.1), the firewall and/or proxy server in place allow access.

  If Gnome or KDE complain or give warnings about resolving the host name, that is a bad sign.
- Two IBM Information Server (DataStage) Projects are created as part of the product install. The default name for the Information Analyzer related Project is, ANALYZERPROJECT. Don’t delete that Project without guidance.
- And lastly, IBM Information Server Version 8.01 FixPak 1A-
  - This should be considered a mandatory FixPak installation for all new installs.
  - This FixPak comes with a 16 page HTML Read Me type document in the root directory of its packaging.

    Even if you never plan to install FixPak 1A, read this document as it has many good tunables and related (easy to put in place, zero negative impact type recommendations).
  - By page 4 or so of the Read Me guide, they have you installing this FixPak. Be certain to read pages 5 through 16 before starting the install; there is a lot of critical guidance on those pages, guidance you want to review before you install the FixPak.

- As an overview, we offer-

  Unlike many other software packages as it relates to a new product install, you do not install a brand new IBM Information Server, and then immediately install FixPak 1A. First you install IBM Information Server, then you configure a small portion of it, then you install FixPak 1A.

  The type of configuration we are discussing is that as found on page 57 of the *IBM Information Server Planning, Installation, and Configuration Guide*, under the section entitled, *Configuring credential mappings for WebSphere DataStage and WebSphere QualityStage users*. This section contains the minimum IBM Information Server configuration you must perform before installing FixPak 1A.

If you can successfully install IBM Information Server, administering it on a daily basis seems relatively easy. A successful install of IBM Information Server on Linux requires moderate level expertise from a Linux system administrator type, entry level to moderate level expertise regarding the database server hosting the meta data repository (entry level for DB2 UDB, moderate for Oracle and SQL/Server), and an ability to follow directions with care for the install of IBM Information Server proper.

12.2 Configuring IIS for use with MySQL

In the previous section, we overviewed installing IBM Information Server (IIS) on the RedHat Linux platform. IBM Information Server relies on a relational database server to host its meta data repository, and this database server must be a specific version of either DB2 UDB, Oracle, or Microsoft SQL/Server. IBM Information Server administrators, operators, end users, etcetera, never directly interact with the meta data repository; the meta data repository serves to configure IBM Information Server, and record the analysis data and other discoveries and enhanced information, that IBM Information Server records or makes available.

> *When we discuss configuring IBM Information Server for use with MySQL (or any data source for that matter), we are discussing the configuration of IBM Information Server and (MySQL) both; MySQL as an ODBC data source provider, a data end point, be it source or target. In this context, IBM Information Server is considered an ODBC client to MySQL.*
>
> *For example, a given IBM Information Server DataStage Job may read from MySQL, perform some analysis or transformation of data, and then write that data back to MySQL. The same Job could also read and/or write from sources other than MySQL, including a mixture of numerous and differing data*

> *sources, relational, non-relational, real time, batch, whatever. IBM Information Server is a client program to these database servers.*

This section of this document details how to configure IBM Information Server for use with MySQL (as a data source), including conditions and settings that may have to be researched or even reconfigured on MySQL, or IBM Information Server both.

Note: While we are using MySQL as the example on the pages that follow, IBM Information Server is very consistent in the area of configuring an ODBC data source. Once you learn how to configure one ODBC data source with IBM Information Server, you have 95% or more of what you need to configure any remaining (new) ODBC data sources. They all follow the same basic format and procedure.

RedHat Linux, what exactly is installed

Specifically when discussing the MySQL database server, a large number of folks will attempt to use the MySQL server software that comes bundled with the RedHat Linux operating system. A handy skill to have when working on RedHat Linux, is to be able to reverse engineer exactly what is installed on a given Linux server; what versions, what optional components (of MySQL), what compilations or distributions. Linux is an open source world, just because it says MySQL doesn't mean it came from MySQL the relational database software company. Also, MySQL has 8 or more (packages) that may be installed, each providing a specific area of functionality. Is a MySQL ODBC client connectivity package even installed?

> *A package is loosely defined as one indivisible system or subsystem that has been installed on Linux. A robust software package like MySQL may be composed of many packages, offering the system administrator the ability to install basically all, or just a subset of MySQL.*
>
> *You can install software packages on RedHat without using packages or the RedHat Package Manager. MySQL, and most Linux centric software packages do use the RedHat Package Manager.*

When we first encountered the RedHat Package Manager 10+ years ago, we hated it. It seemed so much easier to just untar or unzip files to install them; placing them where ever you wanted. The advantage to using something like a RedHat Package Manager is that it maintains a database of everything you install on the operating system, including versions and whatever dependencies that may exist between two or more installed packages. The RedHat Package Manager has become so popular, folks even install it on the SuSE Linux server.

Assuming you have no prior knowledge of what has been installed on a given Linux system that has the RedHat Package Manager, below are the steps you would follow to completely reverse engineer what packages are installed, (versions, locations, dependencies, member files, and more);

(Assume all of this work is done by the Linux root user id, the super user.)

- rpm -q -all -i

  The command above will output every package installed on the given Linux server. This command will also output every file name in every package, along with other data. This command will output data for every package, because no package names were specified at the end of this command.

  It would not be uncommon for the output of this command to produce 300,000 or more lines of output; possibly every system and application file on the entire Linux server.

  Since we really only have interest in MySQL and ODBC at this point, we actually prefer to run the command listed below.

- rpm -q -all -i | grep "^Name" | egrep -i "mysql|odbc"

  The command above builds on the first rpm command offered. This command filters the 300,000 lines of output from the first rpm command, giving us only the ones we have interest in. The command above is case sensitive and whitespace sensitive.

  A code review of this full command is offered below;

  - rpm -q -all -i

    Gives us the full output of all packages and member files present on this Linux system.

    "-q" stands for query, or read.

    "-all" says give us every entry, a full listing for every package found.

    "-i" calls to give us the package names in this output. "-i" is short for, identify.

  - "|", the vertical bar, aka ASCII 127, says feed the output of this command, to the command that follows on the same line. There are two such "|" characters in this command sequence. In this context, the "|" character is also called the pipe symbol.

  - grep is a Linux text processing, pattern search command.

    With the "^Name" argument that follows, this grep command is instructed to only output lines which have the fixed character string "Name", on the left margin of output. (These happen to be the package names only.)

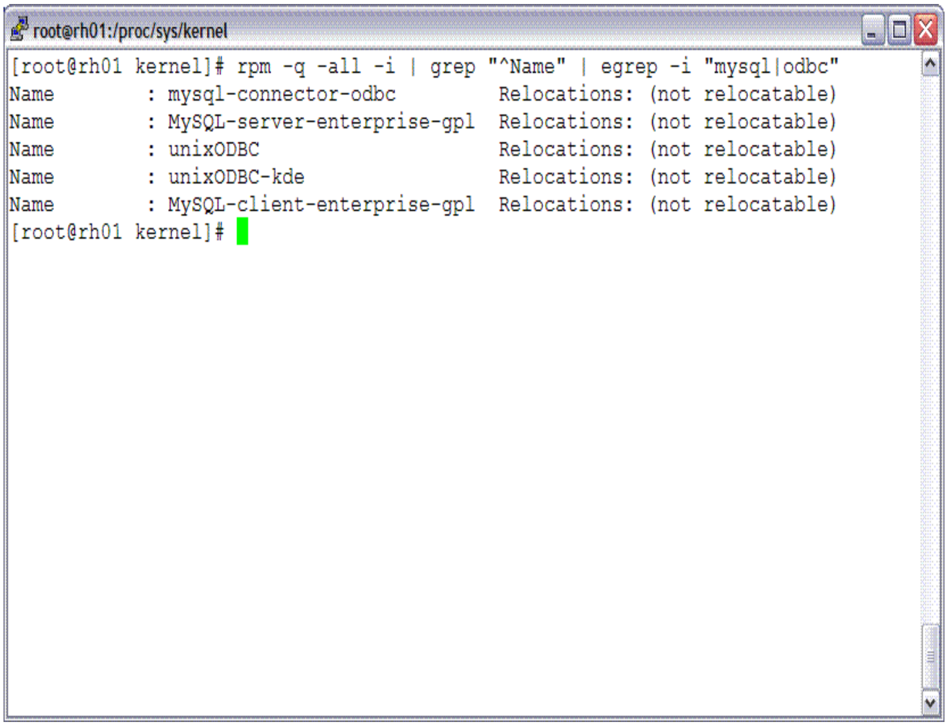
The carat symbol is a special character to grep, and specifies the left margin, left justified text.

- Another “|”, pipe symbol; send this output to the next command that follows.
- egrep is a more advanced form of grep, offering logical OR and AND comparisons.

  In this context, the egrep command is instructed to ignore case sensitivity when examining its input; as specified by the “-i”. (Here “-i” is short for, ignore case sensitivity.)

  Within the double quotes, the vertical present is not a pipe symbol, but takes on another meaning. Here the vertical bar acts as a logical “OR” operator; as in, give us records with a possible mixed case of “mysql” OR “odbc”.

Sample output from this command is displayed in Figure 12-1.



```
[root@rh01 kernel]# rpm -q -all -i | grep "^Name" | egrep -i "mysql|odbc"
Name        : mysql-connector-odbc         Relocations: (not relocatable)
Name        : MySQL-server-enterprise-gpl  Relocations: (not relocatable)
Name        : unixODBC                     Relocations: (not relocatable)
Name        : unixODBC-kde                 Relocations: (not relocatable)
Name        : MySQL-client-enterprise-gpl  Relocations: (not relocatable)
[root@rh01 kernel]# 
```

Figure 12-1   Output from rpm command, with options.

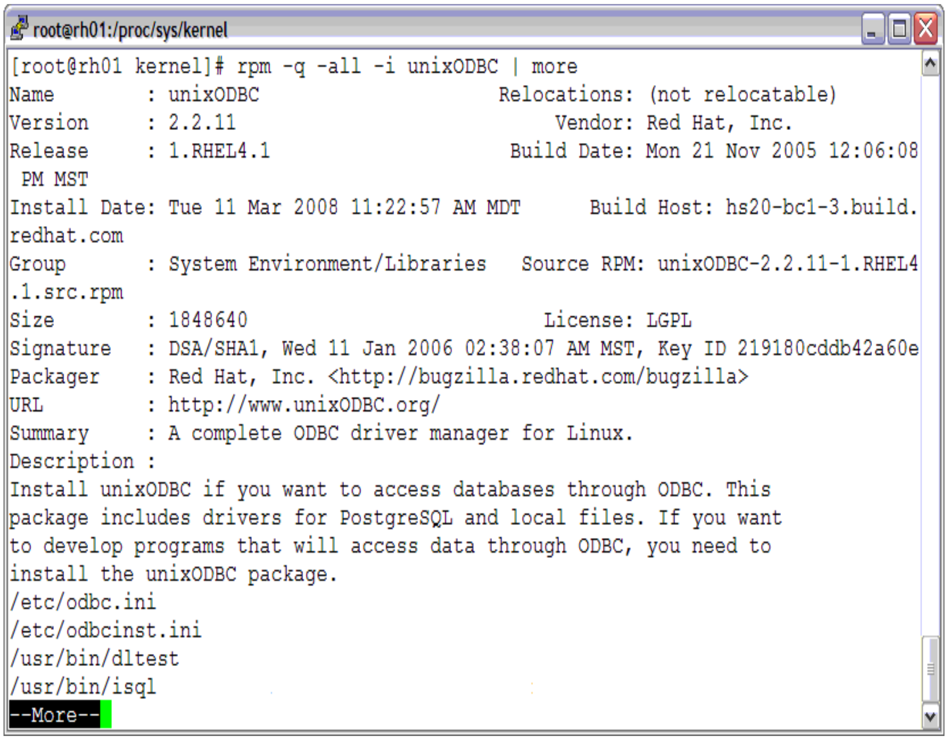
- In Figure 12-1, we are told of five packages that are installed on this Linux server, (of the possibly hundreds). The package names appear in column three, after the colon.
- Three of the packages in Figure 12-1 are related to MySQL, and two are related to the unixODBC package, which we will discuss shortly.

  We can determine the associations of the five packages names, by the data presented in Figure 12-2.
- If we wish to see the full contents of a given package, we execute a command similar to,

  rpm -q -all -i {package name}, a direct example being,

  rpm -q -all -i unixODBC

  Sample output from this command is listed in Figure 12-2.



```
[root@rh01 kernel]# rpm -q -all -i unixODBC | more
Name        : unixODBC                     Relocations: (not relocatable)
Version     : 2.2.11                            Vendor: Red Hat, Inc.
Release     : 1.RHEL4.1                     Build Date: Mon 21 Nov 2005 12:06:08
 PM MST
Install Date: Tue 11 Mar 2008 11:22:57 AM MDT      Build Host: hs20-bc1-3.build.
redhat.com
Group       : System Environment/Libraries   Source RPM: unixODBC-2.2.11-1.RHEL4
.1.src.rpm
Size        : 1848640                          License: LGPL
Signature   : DSA/SHA1, Wed 11 Jan 2006 02:38:07 AM MST, Key ID 219180cddb42a60e
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://www.unixODBC.org/
Summary     : A complete ODBC driver manager for Linux.
Description :
Install unixODBC if you want to access databases through ODBC. This
package includes drivers for PostgreSQL and local files. If you want
to develop programs that will access data through ODBC, you need to
install the unixODBC package.
/etc/odbc.ini
/etc/odbcinst.ini
/usr/bin/dltest
/usr/bin/isql
--More--
```

Figure 12-2   Further output of the rpm command, with options.

- The data in Figure 12-2 is a partial listing, the full output scrolls off of the page. Displayed in Figure 12-2, we see;
  - The package name (Name).

- The version, install date, name of the installation file from which this package came.
- We also see the source organization or company under (URL).
- At this end of Figure 12-2, we see the output includes a complete file listing for this package, of which only 4 files are displayed in Figure 12-2.

MySQL packages

After receiving the output in Figure 12-1 above, we have isolated the five package names we have interest in. (You’re number may vary.) If we run the example in Figure 12-2 for each of these five package names, we confirm that three of the packages we found are from MySQL.

Through a Google query, we found the following URL that explains each of the MySQL packages, http://dev.mysql.com/doc/refman/5.0/en/linux-rpm.html. This URL offers the following;

- MySQL-server-enterprise-gpl
    - Contains the MySQL server software itself; reference to the word “server”.
    - “enterprise” is reference to a version, a licensing of the server software. (The numeric software version is 5.0.50.sp1a, which we can retrieve from the rpm command above.)
    - “gpl” is reference to the C compiler/libraries which are supported.
- MySQL-client-enterprise-gpl
    - “client” is to the MySQL client and administration tools; for example, a character based command line interface by in which you can run basic SQL commands.
    - “enterprise” and “gpl” are the same as above.
- mysql-connector-odbc

  “Connector” and “odbc” are references to the MySQL ODBC client connector pack. Like most database servers, MySQL supports end user connectivity through a number of communication protocols, some local, some networked.

  You really want to run a version 3.51 of the MySQL ODBC driver or above. See below.
- (Others)

  Other MySQL package names you may encounter include;

- “bench”, includes various performance and benchmark scripts and tools.
- “devel”, includes development libraries for Perl, and others.
- “debug”, used to analyze MySQL software crashes.
- “shared”, used by languages that dynamically load libraries.
- “shared-compat”, similar to above, but for older releases of MySQL.
- “cluster*”, and/or “ndb”, a number of packages to support clustering of MySQL.
- “test” and “src”, the source code to, and test suite for MySQL.

Note: Minimally you need to have the “server” and client” packages to effectively run MySQL, and we install them in order of; server, then client.

Also, if you want to support ODBC client connectivity (as we must with IBM Information Server), you have to locate an ODBC driver package for MySQL. Generally the MySQL ODBC download page is separate from the remaining areas where you can download MySQL packages. We install the ODBC connectivity package after installing the server, and then client packages.

In September 2007, MySQL released a much improved version of its ODBC driver package. See, http://blogs.mysql.com/kaj/2007/09/11/re-engineered-odbc-51-driver-for-mysql/

Other MySQL near, or related topics

Other topics, URL’s, or comments (including tips and feedback we have received from other persons) that may be helpful when working with MySQL, are listed below;

– Issues and dependencies between multiple RedHat packages-

  If you install every package available with RedHat Linux Advanced Server, you are going to see dozens of packages with MySQL in all or part of the package name. Most of these are programming APIs to support MySQL; for example, you’ll see packages to support Perl, Tcl/Tk, SmallTalk, etcetera.

  If you then try to install whatever version of MySQL you prefer to work with, you will then enter a condition where installing your (new) version would conflict with any previously installed versions of packages. This is actually a good thing, the RedHat Package Manager is doing its job; keeping you from accidently causing issues between dependent packages.

If you are just working with or testing MySQL, we found it easier to only install packages we need (go with a minimal RedHat install), and then install the specific MySQL packages we wish to work with from a fresh download from MySQL. A minimal install of RedHat also boots a whole lot quicker.

If you still have conflicting package references, rpm does support overrides and related of installs and updates of packages. For example, we were receiving a complaint about a version conflict with some programming language that we had never heard of, were never going to use, etcetera. We issued the following command to delete that super obscure package, even though it had dependencies;

```
rpm -e --nodeps {Package Name}
```

"-e" is short for erase (delete package). "--nodeps" is a reference to ignore dependencies.

– MySQL not starting-

The MySQL server wants to be started and stopped by the mysql user id. It looks like our RedHat server install would configure to automatically boot MySQL upon Linux boot, by the root user id, and this left the MySQL server is a non-working state. Some comments on this topic include;

- MySQL uses a two process architecture, and keeps a daemon process running to support new user connections and related. This daemon process is located in /etc/rc.d/init.d/, and is invoked by the mysql user id. Examples being-

```
/etc/rc.d/init.d/mysqld start # start MySQL
/etc/rc.d/init.d/mysqld stop # stop MySQL
/etc/rc.d/init.d/mysqld status # Determine operating state of MySQL
```

- If you are receiving errors when starting MySQL about not being able to create a MySQL PID file, See, http://www.linuxquestions.org/questions/linux-server-73/mysql-cant-createwrite-to-mysql.pid-612467/
- If you are receiving errors when starting MySQL including text similar to, "mysqld dead but subsys locked", this is the root user id issue. See, http://www.linuxquestions.org/questions/linux-general-1/problem-with-mysql-262262/
- As a topic, how to mange MySQL passwords is detailed at, http://www.netadmintools.com/art90.html
- Still can't start MySQL, see the following all purpose MySQL primer page at, http://www.yolinux.com/TUTORIALS/LinuxTutorialMySQL.html

This is a really good MySQL tutorial and is Linux specific.

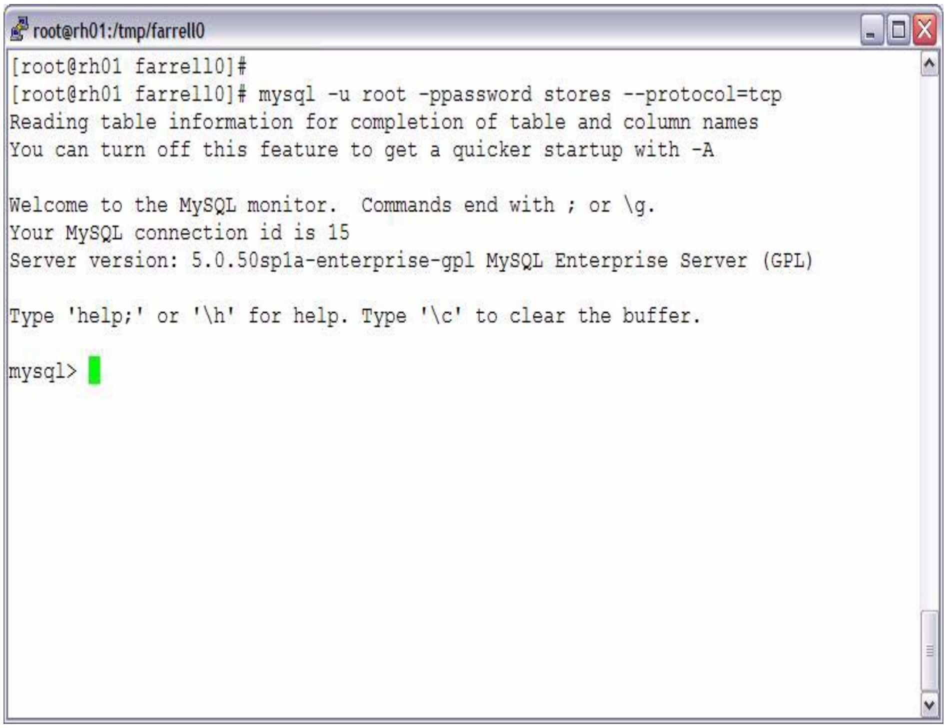
Another we like is at, http://www.science.uva.nl/ict/ossdocs/mysql/manual_toc.html

This URL gives great detail about some of the MySQL SQL DDL commands.

Last on MySQL, install verify, version, TCP ports, other

Just because the MySQL server is installed and running, doesn’t mean the MySQL server is configured to support network connectivity; MySQL might be configured to support only local communication protocols. Also, there are some MySQL configuration parameters we need to have recorded for our use later in this document.

With the MySQL sever and client packages installed, and with the MySQL server in an operational state, we can run the character based MySQL SQL prompt utility to perform some additional test and research. That program name is mysql, and is most often located in /usr/bin/. Figure 12-3 below demonstrates use of mysql.



```
[root@rh01 farrell0]#
[root@rh01 farrell0]# mysql -u root -ppassword stores --protocol=tcp
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.0.50sp1a-enterprise-gpl MySQL Enterprise Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> 
```

Figure 12-3   Logging on to MySQL via mysql command line interface

A code review of the command executed in Figure 12-3, is offered below;

- The command being run is mysql, which offers a command line interface to the MySQL server.
- "-u root"

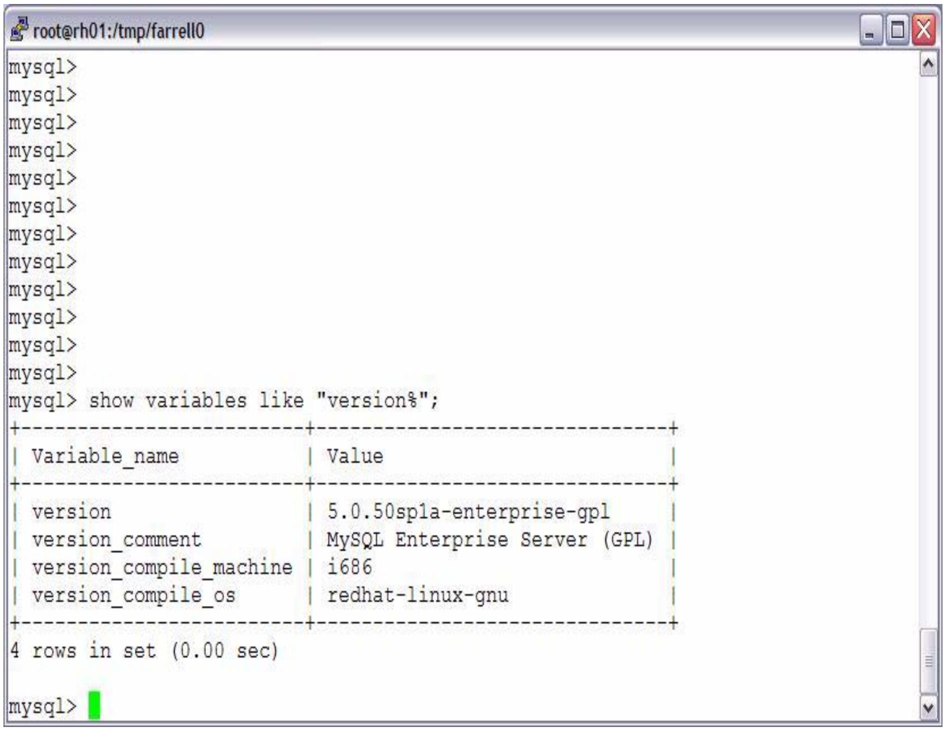
  The "-u flag argument we wish to specify our connection user id. the argument is following by white space, and then the user name.

  While you can't boot MySQL as root, root can do most everything else with MySQL.
- "-ppassword"

  The -p{word}" argument says we wish to specify our authentication password. *Unlike the user id, there is zero white space between the "-p" switch and the password value.*
- "stores" is the name of the database we wish to connect to.
- "--protocol=tcp" indicates we wish to connect via a TCP/IP protocol.

  This is our attempt to install verify that MySQL is in fact configured to support TCP/IP end user connections.
- *All of the arguments above are optional.*

Figure 12-4 displays how to get and/or confirm the MySQL version from mysql. This is an excellent portion of in install verification test.



```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> show variables like "version%";
+-------------------------+------------------------------+
| Variable_name           | Value                        |
+-------------------------+------------------------------+
| version                 | 5.0.50sp1a-enterprise-gpl    |
| version_comment         | MySQL Enterprise Server (GPL) |
| version_compile_machine | i686                         |
| version_compile_os      | redhat-linux-gnu             |
+-------------------------+------------------------------+
4 rows in set (0.00 sec)

mysql>
```

Figure 12-4   Getting MySQL server version from mysql.

A code review of the command executed in Figure 12-4, is offered below;

– The basic syntax is, show variables like ‘version%’;

Sample output as shown.

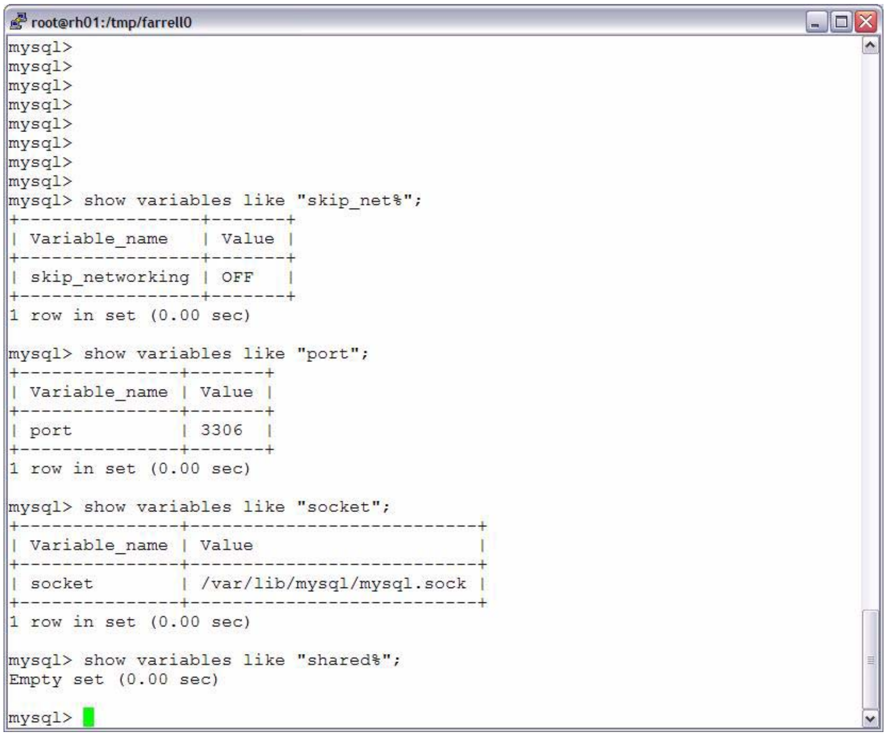
- – This is another means to confirm or discover the version and source of the MySQL distribution you are using, especially if you had not previously done so, or were not able to, via the RedHat Package Manager.

Note: The version and distribution of the MySQL server that was installed from our RedHat Advanced Server V4 installation media was not from MySQL, but from a company called SleepyCat Software, and actually reported as BerkeleyDB. Even though the package name said MySQL, and the command prompt we ran above said mysql, this was/is a modified version of MySQL. The SleepyCat Web site said they produced a highly optimized version of MySQL, with the implied condition being *modified/different*.

SleepyCat was acquired by Oracle in 2006 for reasons not related to MySQL. See, http://www.oracle.com/corporate/press/2006_feb/sleepycat.html

As a modified version of MySQL, we had trouble getting the SleepyCat version of MySQL to work with ODBC drivers we had found. We uninstalled the SleepyCat version and went with a distribution of MySQL from MySQL.

Figure 12-5 displays a number of commands to gather the MySQL communication related parameters.



```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> show variables like "skip_net%";
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| skip_networking | OFF   |
+-----------------+-------+
1 row in set (0.00 sec)

mysql> show variables like "port";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3306  |
+---------------+-------+
1 row in set (0.00 sec)

mysql> show variables like "socket";
+---------------+---------------------------+
| Variable_name | Value                     |
+---------------+---------------------------+
| socket        | /var/lib/mysql/mysql.sock |
+---------------+---------------------------+
1 row in set (0.00 sec)

mysql> show variables like "shared%";
Empty set (0.00 sec)

mysql>
```

Figure 12-5 Getting MySQL communication related parameters through mysql.

A code review of the commands executed in Figure 12-5, is offered below;

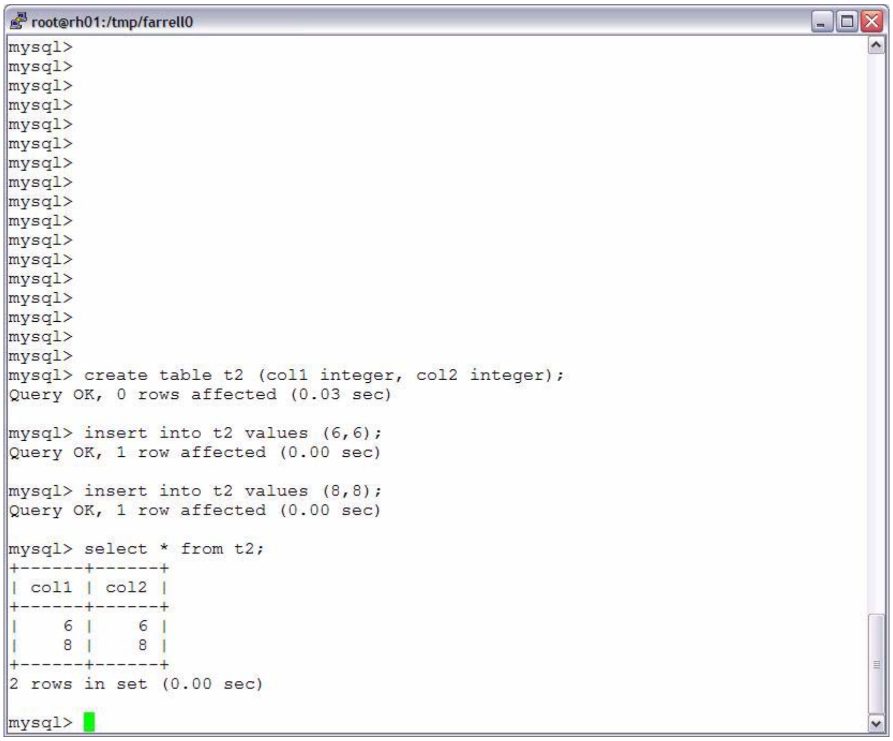
- show variables like “skip_net%”;

  This is the basic command format we repeat four times, each with differing MySQL server control variables we wish to display.

  SKIP_NETWORKING = ON would indicate that this MySQL server does not support TCP/IP networking. Per our intent, that would be bad.
- “port” shows the TCP/IP communication port number value. We will need this value later in order to configure the ODBC driver.
- “socket” supports a local communication protocol; we don’t need this value for our purposes.
- “shared%” would displays the shared emory communication protocol, also local, and also something we don’t need.
- Commands in this area are documented very well at the following URL, http://books.google.com/books?id=s_87mv-Eo4AC&pg=PA263&lpg=PA263&dq=how+to+determine+what+communication+protocol+are+available

+for+a+mysql+server&source=web&ots=tbRDy9LCvk&sig=ioopDS4SLqpILNiAWQDXTa49TbY&hl=en#PPA264,M1

Lastly, since we are this close, we should test some actual SQL workings with MySQL and the mysql command interface. Example as shown in Figure 12-6.



root@rh01:/tmp/farrell0

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> create table t2 (col1 integer, col2 integer);
Query OK, 0 rows affected (0.03 sec)

mysql> insert into t2 values (6,6);
Query OK, 1 row affected (0.00 sec)

mysql> insert into t2 values (8,8);
Query OK, 1 row affected (0.00 sec)

mysql> select * from t2;
+------+------+
| col1 | col2 |
+------+------+
|    6 |    6 |
|    8 |    8 |
+------+------+
2 rows in set (0.00 sec)

mysql> 
```

Figure 12-6   Creating a table, inserting data, selecting data from MySQL.

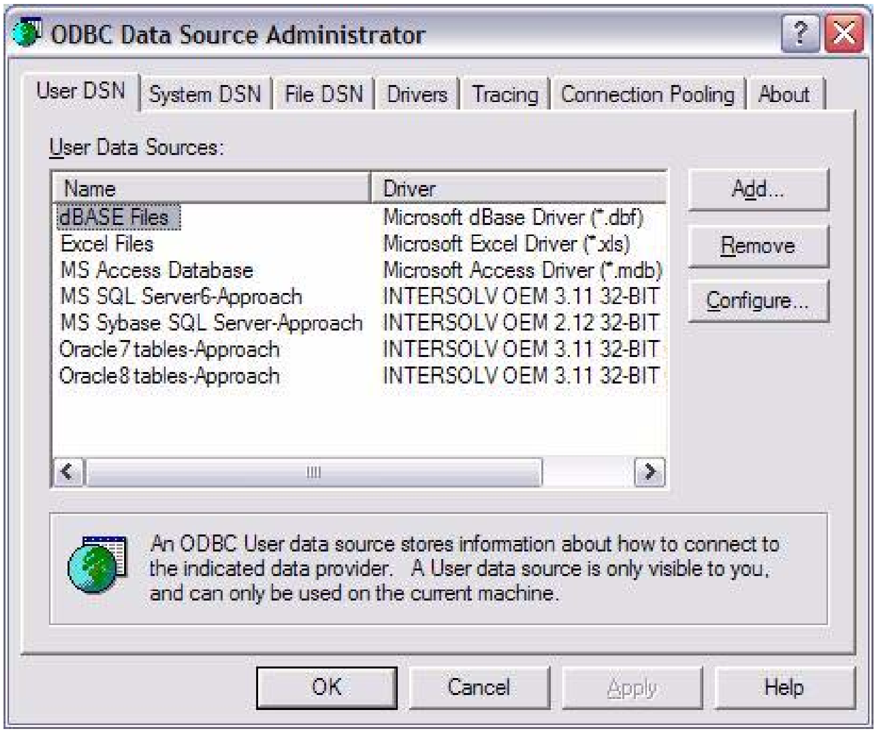
In Figure 12-6, we create a table named "t2", add two rows of data, and then read that data back. This occurs in the "stores" database, as determined by our connection string in Figure 12-3.

To exit the mysql command line interface, type "quit", and then ENTER, or Carriage Return.

unixODBC, and unixODBC-kde packages

In Figure 12-1, we not only saw three packages installed on our Linux server related to MySQL, we also saw two additional packages entitled, unixODBC and unixODBC-kde. "kde" is a reference to the Linux Kommon (Common) Desktop Environment (KDE), the Linux graphical desktop similar to Microsoft Windows. These two packages together provide a functionality similar to the Windows

ODBC Data Source Administrator. Figure 12-7 displays the Windows Platform ODBC Data Source Administrator program, while Figure 12-8 displays the Linux version of this same program.

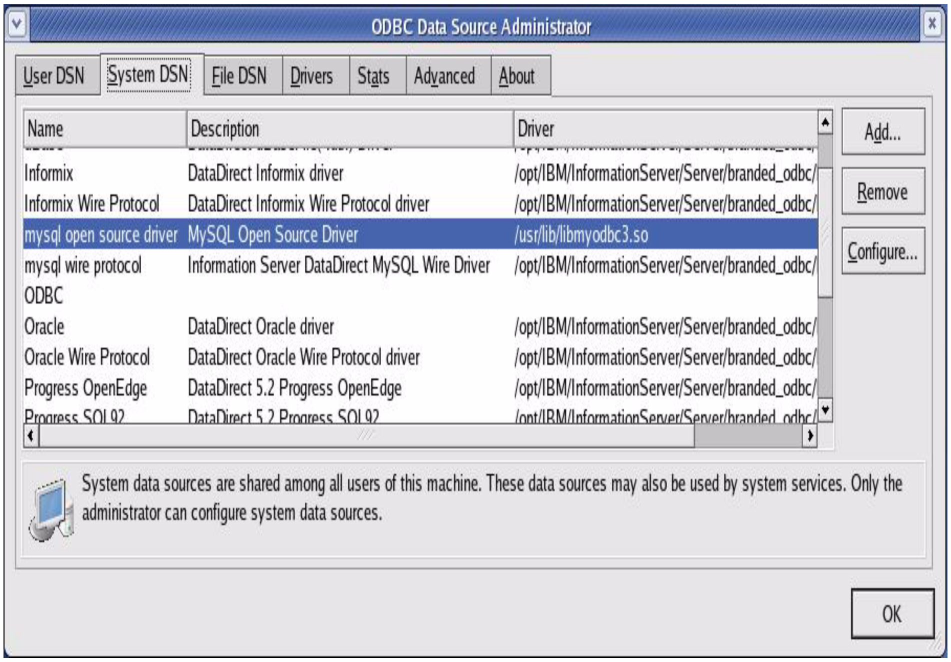


User Data Sources:

Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
MS Access Database	Microsoft Access Driver (*.mdb)
MS SQL Server6-Approach	INTERSOLV OEM 3.11 32-BIT
MS Sybase SQL Server-Approach	INTERSOLV OEM 2.12 32-BIT
Oracle7 tables-Approach	INTERSOLV OEM 3.11 32-BIT
Oracle8 tables-Approach	INTERSOLV OEM 3.11 32-BIT

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

Figure 12-7 Windows ODBC Data Source Administrator program.



Name	Description	Driver
Informix	DataDirect Informix driver	/opt/IBM/InformationServer/Server/branded_odbc/
Informix Wire Protocol	DataDirect Informix Wire Protocol driver	/opt/IBM/InformationServer/Server/branded_odbc/
mysql open source driver	MySQL Open Source Driver	/usr/lib/libmyodbc3.so
mysql wire protocol	Information Server DataDirect MySQL Wire Driver	/opt/IBM/InformationServer/Server/branded_odbc/
ODBC		
Oracle	DataDirect Oracle driver	/opt/IBM/InformationServer/Server/branded_odbc/
Oracle Wire Protocol	DataDirect Oracle Wire Protocol driver	/opt/IBM/InformationServer/Server/branded_odbc/
Progress OpenEdge	DataDirect 5.2 Progress OpenEdge	/opt/IBM/InformationServer/Server/branded_odbc/
Progress SQL92	DataDirect 5.2 Progress SQL92	/opt/IBM/InformationServer/Server/branded_odbc/

System data sources are shared among all users of this machine. These data sources may also be used by system services. Only the administrator can configure system data sources.

Figure 12-8 Linux ODBC Data Source Administrator program.

If effect, the Linux version of the ODBC Data Source Administrator gives you a graphical user interface to edit the /etc/odbc.ini configuration file, which is in ASCII text. The /etc/odbc.ini file is a Linux system configuration file. (IBM Information Server has its own copy of this same basic file, which is nearly identical in its use, format, etcetera.)

Comments related to unixODBC and unixODBC-kde include;

– We consider these packages and the functionality they provide to be must haves.

  These packages can be downloaded from, http://www.easysoft.com/developer/interfaces/odbc/linux.html

  This Web form also includes a full tutorial on how to install and use these packages.

– You run the program in Figure 12-8 by invoking the following command from the Linux command line prompt; `ODBCConfig`

– ODBCConfig has no magic properties; it simply gives you a graphical means to edit the `/etc/odbc.ini` file.

- – After you run ODBCConfig, or directly edit the /etc/odbc.ini file, you can test your work with odbctest, another graphical user interface program in these packages.
- – Changes to the /etc/odbc.ini file take effect immediately; no restarts, etcetera, are required after editing this file.

Note: We use ODBCConfig (or directly edit the /etc/odbc.ini file), and then run odbctest to verify that MySQL (or whatever ODBC data source you are working with) can accept a networked ODBC end user connection.

This is huge. When this test works, we know we have everything outside of IBM Information Server working correctly.

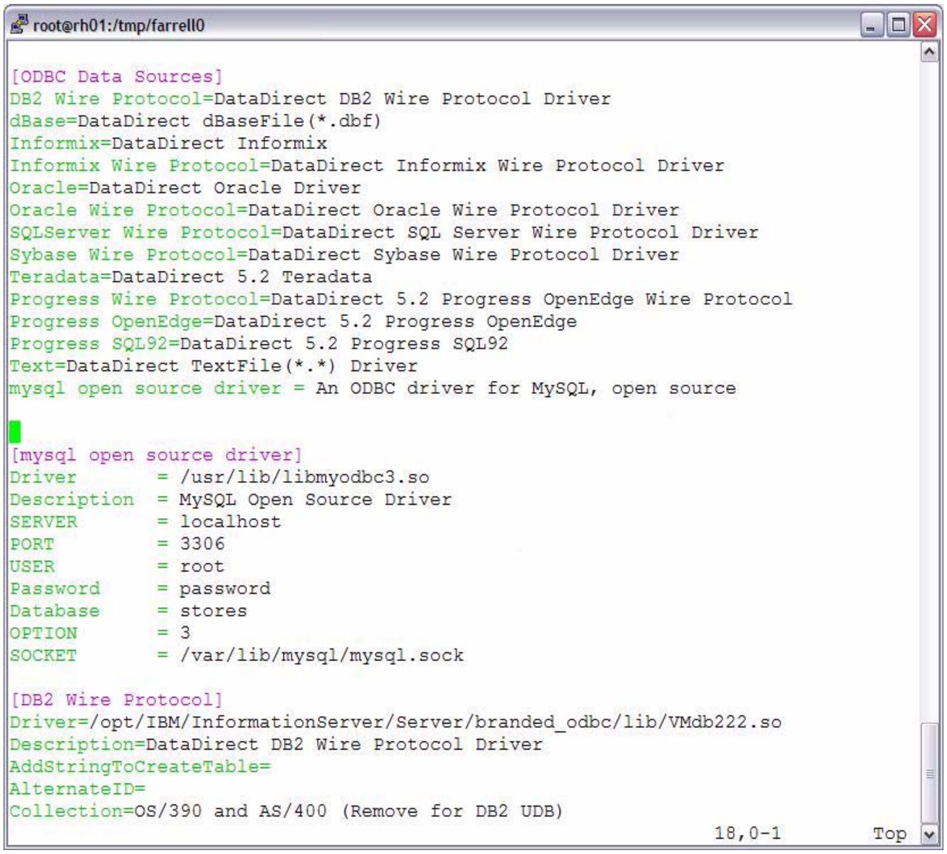
odbctest tests the connection, based on the entries provided in /etc/odbc.ini. /etc/odbc.ini is a Linux/Unix system configuration file owned by root.

IBM Information Server (IIS) uses another file, its own version of odbc.ini, but these two files are nearly identical. (The IIS version of this file is owned by dsadm, and offers a small amount of additional capabilities over its Linux counterpart.)

This files are so similar, however, we often just cut and paste entries from one file to another.

Sample /etc/odbc.ini file

Figure 12-9 displays sample contents of the Linux/Unix /etc/odbc.ini file.



```
[ODBC Data Sources]
DB2 Wire Protocol=DataDirect DB2 Wire Protocol Driver
dBase=DataDirect dBaseFile(*.dbf)
Informix=DataDirect Informix
Informix Wire Protocol=DataDirect Informix Wire Protocol Driver
Oracle=DataDirect Oracle Driver
Oracle Wire Protocol=DataDirect Oracle Wire Protocol Driver
SQLServer Wire Protocol=DataDirect SQL Server Wire Protocol Driver
Sybase Wire Protocol=DataDirect Sybase Wire Protocol Driver
Teradata=DataDirect 5.2 Teradata
Progress Wire Protocol=DataDirect 5.2 Progress OpenEdge Wire Protocol
Progress OpenEdge=DataDirect 5.2 Progress OpenEdge
Progress SQL92=DataDirect 5.2 Progress SQL92
Text=DataDirect TextFile(*.*) Driver
mysql open source driver = An ODBC driver for MySQL, open source


[mysql open source driver]
Driver         = /usr/lib/libmyodbc3.so
Description    = MySQL Open Source Driver
SERVER         = localhost
PORT           = 3306
USER           = root
Password       = password
Database       = stores
OPTION         = 3
SOCKET         = /var/lib/mysql/mysql.sock

[DB2 Wire Protocol]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/VMdb222.so
Description=DataDirect DB2 Wire Protocol Driver
AddStringToCreateTable=
AlternateID=
Collection=OS/390 and AS/400 (Remove for DB2 UDB)
                                                            18,0-1         Top
```

Figure 12-9   Linux/Unix /etc/odbc.ini file.

Related to Figure 12-9, the following is offered;

- The first paragraph in Figure 12-9 begins with the string, “[ODBC Data Sources]”.

  Following this declaration, is a one line entry for every remaining paragraph in this file. Notice the last entry in paragraph one with the value, “mysql open source driver”. This value corresponds with the value that opens the paragraph for our actual driver description, just below.

  We can use any value we want as this string; we could have called this driver entry “Accounting”. The key point is that these values are paired from the header paragraph, to the individual driver definition paragraphs.

- Entries in the first paragraph have the format; tag = description

- The second paragraph in Figure 12-9 is our actual ODBC driver entry definition. A code review of this paragraph follows;

- “mysql open source driver” is the identifier (the connection handle) for this ODBC driver, and the value that matches an entry in the paragraph above.
- Driver = /usr/lib/libmyodbc3.so

  This is the actual driver file, always a .DLL file (on Windows), or a .SO file (on Linux or Unix).

  The name and pathname to this file were determined by the supplier of the driver.

  We placed this value here, either by running ODBCConfig, or by editing the /etc/obc.ini file. We knew what value to enter for this line by the work we did in Figure 12-1 and Figure 12-2.
- Description = { Stuff }

  A user friendly description to the entry.
- SERVER = localhost

  The Linux host name where the MySQL database server is listening for connections. Using system libraries, this name will resolve into an IP address. Since IBM Information Server and MySQL will reside on the same Linux host in our test, localhost is a valid host name.

  If they were on different hosts, you would need a value other than localhost.
- PORT = 3306

  The port number that MySQL is listening on. An IP address and a port number together form a socket, which is the basis of all TCP/IP based communication.

  We got this value from Figure 12-5.
- USER, Password, and Database-

  Through our tests and configuration of MySQL, we know these values to be those we wish to use for our network connection to MySQL.
- OPTION = 3

  This value has meaning in the ODBC driver world; remembering that ODBC is an open and standard programming API. We always set this value to 3 when using MySQL.

  This topic is not expanded upon further here.
- Socket = /var/lib/mysql/mysql.sock.

  Again, we gathered this value from Figure 12-5.

Thus far we have altered and saved the contents of the /etc/odbc.ini file via the ODBCConfig graphical program, or by editing this file directly. We knew what values to supply be research performed earlier. Now it is time to test our work by using odbctest.

odbctest graphical program

odbctest is a graphical program similar to ODBCConfig. Whereas ODBCConfig changes the contents of the /etc/odbc.ini Linux/Unix configuration file, odbctest is non-destructive; odbctest merely reports, it doesn't change any file contents.

To run odbctest from the Linux/Unix prompt, enter; odbctest

The result of this command is displayed in Figure 12-10.

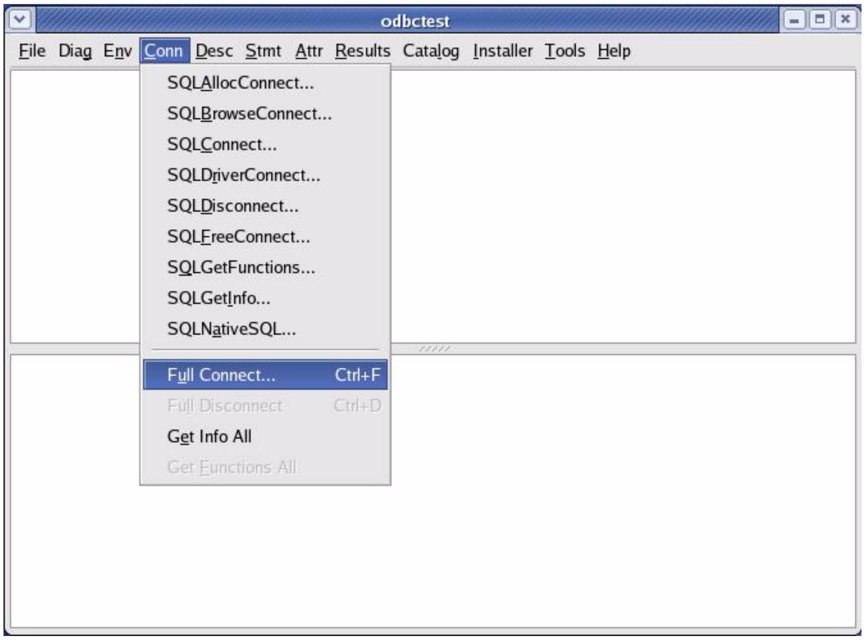
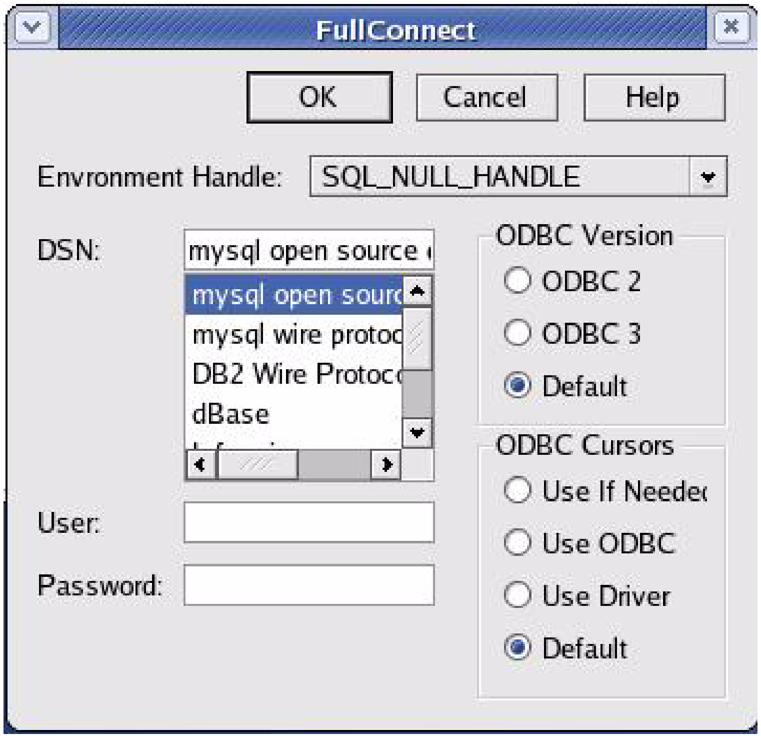


Figure 12-10   Running odbctest, then Conn -> Full Connect

In Figure 12-10 and from the menu bar, we select; Conn -> Full Connect. This action produces the result as shown in Figure 12-11.



Envronment Handle: SQL_NULL_HANDLE

DSN: mysql open source

User:

Password:

ODBC Version

○ ODBC 2

○ ODBC 3

● Default

ODBC Cursors

○ Use If Needec

○ Use ODBC

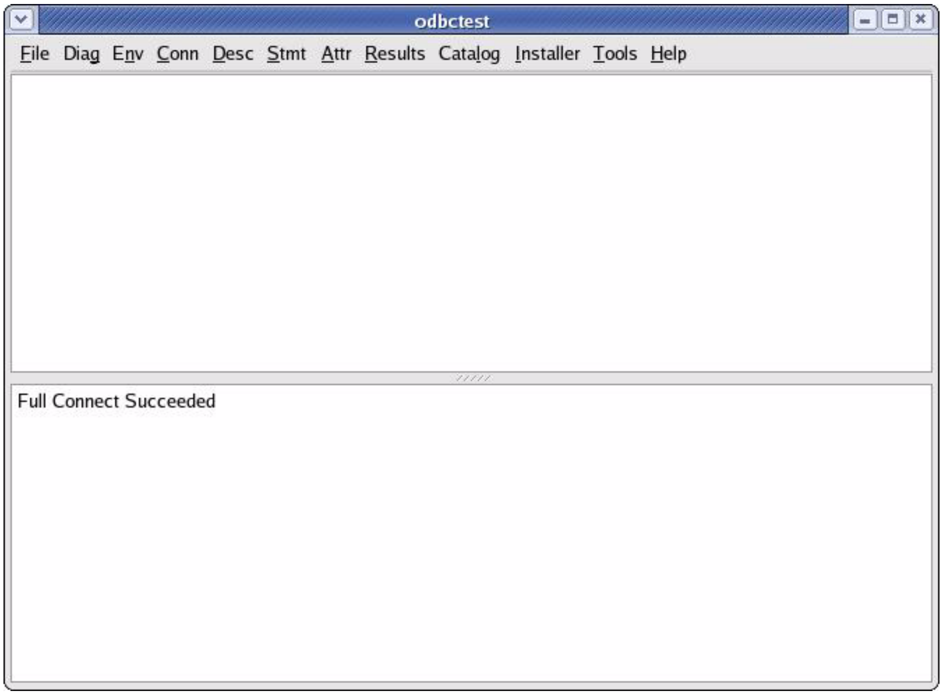
○ Use Driver

● Default

Figure 12-11 odbctest, Full Connect dialog box.

In Figure 12-11 and in the DSN list box visual control, we browse until we find the specific entry we wish to test. The entries in this visual control come from the /etc/odbc.ini Linux/Unix configuration file. After we select the desired DSN value, and enter a User name and Password, we Click [OK] to test.

This action produces the result as displayed in Figure 12-12.


odbctest
File Diag Env Conn Desc Stmt Attr Results Catalog Installer Tools Help

Full Connect Succeeded

Figure 12-12 Successful result of test

Figure 12-12 displays the result of a successful test. A successful test is the result of;

- MySQL is up and running, and accepting TCP (networked) communications.
- The version of the ODBC driver is compatible with the operating system and MySQL.
- The /etc/odbc.ini is edited correctly.

Likely sources of error include;

- If odbctest dumps core or abnormally terminates, look at the .SO file you are using; you probably have a bad or incorrect version.
- Otherwise, look at the diagnostic output in the window displayed in Figure 12-12. Google whatever you text you receive and work it from there.

A really good URL for configuring a Linux/Unix ODBC data source is also available from MySQL at;
http://dev.mysql.com/doc/refman/5.0/es/myodbc-configuration-dsn-unix.html

IIS, ODBC concepts and configuration

IBM Information Server (IIS) ships with a number of popular and IBM fully supported ODBC drivers. Generally these drivers are referred to as the *Data Direct driver pack*; Data Direct being the software vendor specializing in ODBC drivers, whom IBM re-licenses this software from.

Note: IBM Information Server is associated with an installation directory, also referred to as $DSHOME. The default $DSHOME on Linux/Unix is, /opt/IBM/InformationServer/.

The Data Direct drivers are located under $DSHOME/branded_odbc/, more specifically, $DSHOME/branded_odbc/lib/. On Windows platforms, the actual ODBC driver files appear with a .DLL file suffix, and on Linux and Unix, these files appear with a .SO file suffix. The Data Direct drivers are also those that appear as named SQL Connectors in the DataStage and QualityStage Designer program, Palette view, Database drawer; for example, DB2, Informix, Teradata, Oracle, MS SQL/Server, and others.

Using the Data Direct driver pack, one can create DataStage and QualityStage Jobs that read and write to numerous Data Direct driver pack supported data sources within one Job. The ODBC data sources supported by the Data Direct driver pack come pre-configured; meaning, one needs only to specify connection parameters inside the specific SQL Connector operator (Stage) on the Parallel Canvas. For most operations, you should not need to edit or make direct changes to any of the files located beneath the $DSHOME/branded_odbc/ subdirectory. Similarly and for most operations, you should not need to edit the $DSHOME/DSEngine/.odbc.ini file for Data Direct driver pack supported data sources, as the .odbc.ini file comes largely pre-configured.

IBM Information Server also has the ability to connect to data sources which are not included in the Data Direct driver pack; examples including the Ingres or Postgres relational database servers, RedBrick DataWarehouse Server, and others. Inside IBM Information Server, more specifically the DataStage Designer program, data sources of this type are accessed using the (generic) ODBC Connector, in the Palette view, Database drawer. IBM Information Server is extensible in that it allows for use of ODBC drivers from external, non-IBM supported sources. One may also use newer or variant versions of ODBC drivers for data sources that are supported under the Data Direct driver pack, detailed above. In the case of any third party, or non-IBM included ODBC driver or driver version, a portion your customer support will come from IBM, and a portion will come from that third party.

3 control files; dsenv, .odbc.ini, and uvodbc.config

IBM Information Server has three primary ASCII text control files used to configure Linux/Unix ODBC connectivity. Comments related to these files, and IBM Information Server ODBC connectivity include;

– All three of these files are located in; $DSHOME/Server/DSEngine/.

All three of these files are ASCII text, and changes to their contents are made using a program editor.

– Only dsenv requires that a subset of IBM Information Server be restarted in order for changes to take effect. (The DSEngine needs to be restarted.)

In the context of this discussion (Linux/Unix ODBC connectivity), dsenv contains operating system environment variables that are referenced by the ODBC client connectivity libraries. *The fact that these are operating system environment variables is what causes the need to restart DSEngine. (Operating system environment variables suffer famously from this condition.)*

Changes to .odbc.ini and uvodbc.config take place immediately.

– As mentioned and in this context (Linux/Unix ODBC connectivity), dsenv contains any environment variables that the ODBC client libraries you are using may require.

For example, if a given ODBC driver file you were using were to look at an environment variable to ignore case when joining on character columns, you would set this environment variable inside dsenv.

We mention the dsenv file for completeness towards this discussion; that being stated, we rarely need to edit this file for the purpose of configuring Linux/Unix ODBC.

– The IBM Information Server $DSHOME/Server/DSEngine/.odbc.ini file is *very similar* to its Linux/Unix counterpart, /etc/odbc.ini.

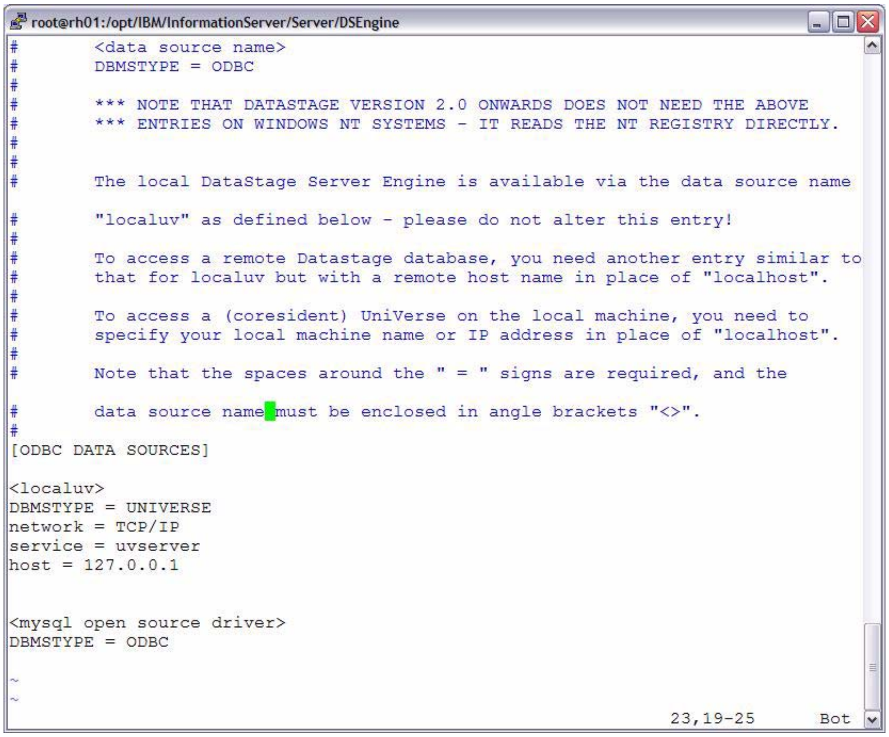
Again, if you are using an IBM Information Server Data Direct driver data source, changes to this file are often minimal.

While it was presented as an example /etc/odbc.ini file, the image in Figure 12-9 and its associated code review serves also as a $DSHOME/Server/DSEngine/.odbc.ini example.

Note: If you placed the MySQL ODBC driver entry from Figure 12-9 in $DSHOME/Server/DSEngine/.odbc.ini, Information Server would be able to access MySQL.

– If you are using a non Data Direct driver pack supported ODBC data source, you must also make a new entry in the

$DSHOME/Server/DSEngine/uvodbc.config control file. Example as shown in Figure 12-13;



```
#       <data source name>
#       DBMSTYPE = ODBC
#
#       *** NOTE THAT DATASTAGE VERSION 2.0 ONWARDS DOES NOT NEED THE ABOVE
#       *** ENTRIES ON WINDOWS NT SYSTEMS - IT READS THE NT REGISTRY DIRECTLY.
#
#
#       The local DataStage Server Engine is available via the data source name

#       "localuv" as defined below - please do not alter this entry!
#
#       To access a remote Datastage database, you need another entry similar to
#       that for localuv but with a remote host name in place of "localhost".
#
#       To access a (coresident) UniVerse on the local machine, you need to
#       specify your local machine name or IP address in place of "localhost".
#
#       Note that the spaces around the " = " signs are required, and the

#       data source name must be enclosed in angle brackets "<>".
#
[ODBC DATA SOURCES]

<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = 127.0.0.1


<mysql open source driver>
DBMSTYPE = ODBC

~
~
                                                                     23,19-25      Bot
```

Figure 12-13   Sample uvodbc.config file

The following code review pertains to the $DSHOME/Server/DSEngine/uvodbc.config displayed in Figure 12-13;

– The first entry in this file “<localuv>” is special to IBM Information Server and should not be changed; this entry comes with this file by default.

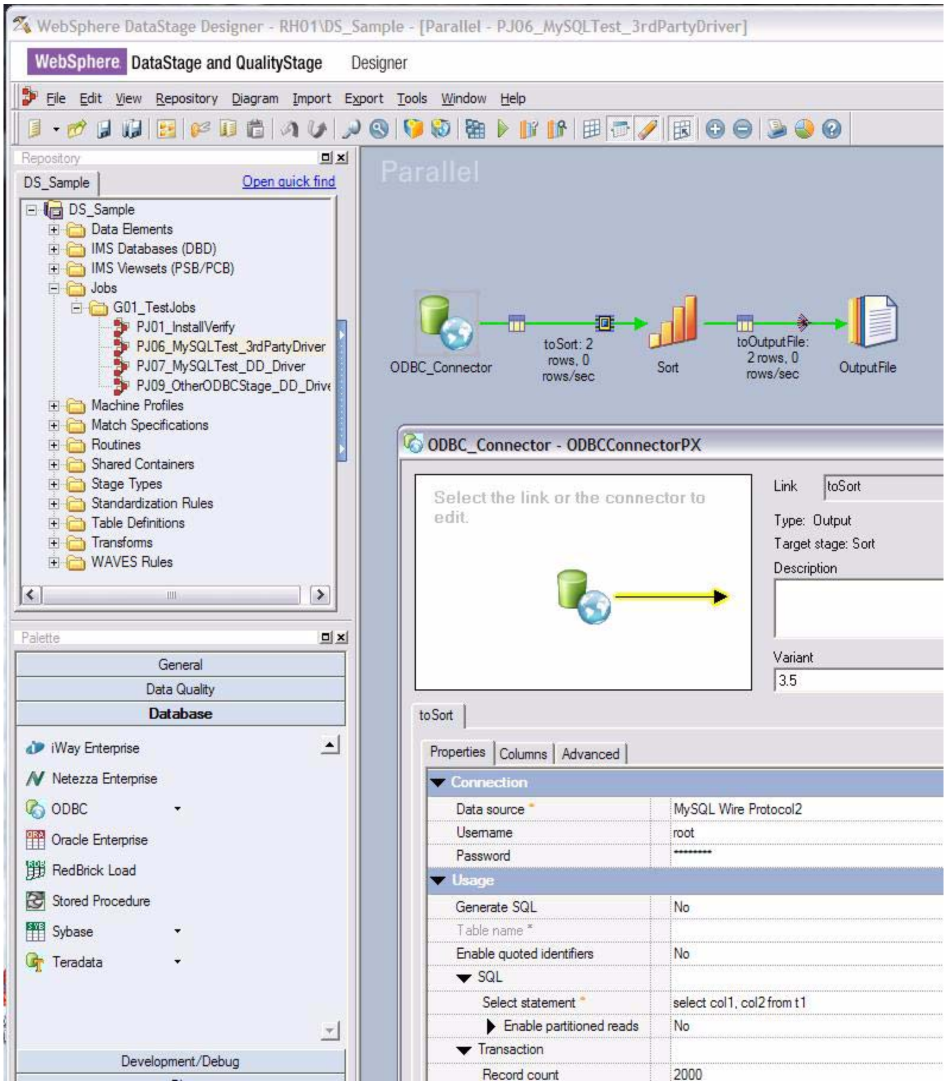
  The second entry, entitled “<mysql open source driver>”, is the one we needed to add.

– This second entry has a tag name of our choosing “<mysql open source driver>”, and corresponds to a same named entry in $DSHOME/Server/DSEngine/.odbc.ini.

  While this new entry may seem sparse, it is required.

Final Testing, IIS and MySQL

We can use the DataStage component of IBM Information Server to test our work, as displayed in Figure 12-14.



Link: toSort

Type: Output

Target stage: Sort

Variant: 3.5

Connection	
Data source *	MySQL Wire Protocol2
Username	root
Password	*********
Usage	
Generate SQL	No
Table name *	
Enable quoted identifiers	No
SQL	
Select statement *	select col1, col2 from t1
Enable partitioned reads	No
Transaction	
Record count	2000

Figure 12-14 Test Job for ODBC connectivity.

From Figure 12-14, the following is offered;

– In Figure 12-14, we have created a new DataStage EE Job with three operators; the ODBC Connector from the Palette -> Database drawer, and a Sort and a Sequential File for output.

– The Properties view (dialog box) of the ODBC Connector is open in Figure 12-14.

In the upper left portion of the display, it is important to note that you can highlight the Arrow (shown in Yellow), or the Cylinder/Globe icon to its left; highlighting one or the other affects how much detail is available below.

Under Connection -> Data source is a drop down list box visual control; the values available here comes directly from the $DSHOME/Server/DSEngine/.odbc.ini configuration file.

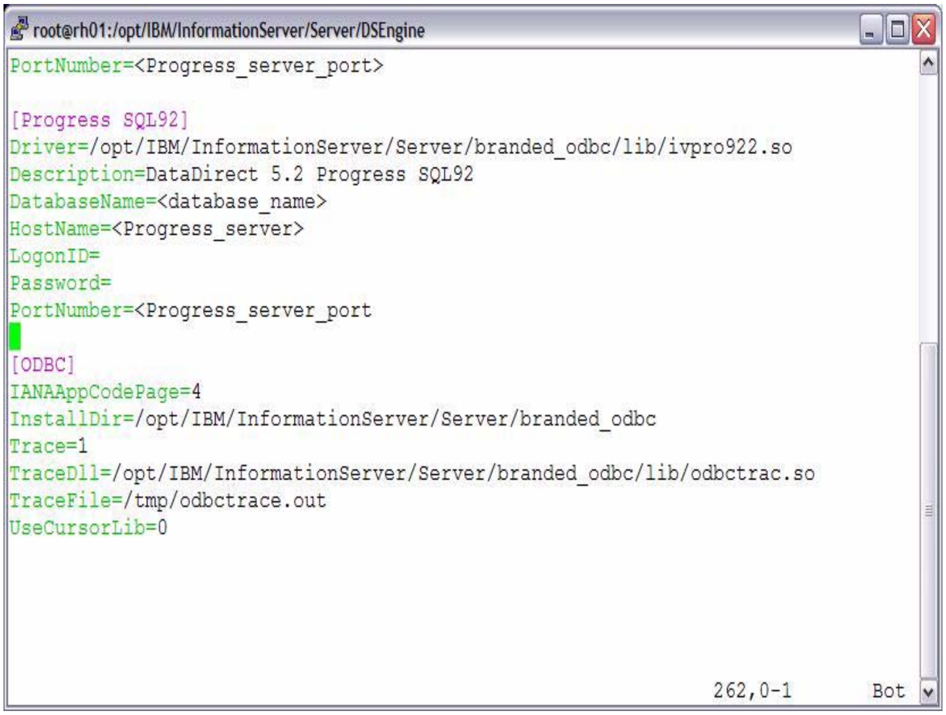
A “[Test]” (connection) button is also available.

In addition to using the Test button mentioned above, compile and run this sample Job. It this job outputs data successfully, you are done. If you observe an error, reference the next section of this document.

Diagnosing errors with IIS and an ODBC driver

If you are not able to complete a successful test as outlined in Figure 12-14 above, retrace or introduce the following;

- Reconfirm successful configuration of the ODBC data source outside of IBM Information server using odbctest.
- Add debug only configuration parameters to the $DSHOME/Server/DSEngine/.odbc.ini, as shown in Figure 12-15.



```
PortNumber=<Progress_server_port>

[Progress SQL92]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/ivpro922.so
Description=DataDirect 5.2 Progress SQL92
DatabaseName=<database_name>
HostName=<Progress_server>
LogonID=
Password=
PortNumber=<Progress_server_port

[ODBC]
IANAAppCodePage=4
InstallDir=/opt/IBM/InformationServer/Server/branded_odbc
Trace=1
TraceDll=/opt/IBM/InformationServer/Server/branded_odbc/lib/odbctrac.so
TraceFile=/tmp/odbctrace.out
UseCursorLib=0




                                                  262,0-1       Bot
```

Figure 12-15   Tracing ODBC issues, .odbc.ini optional settings

Related to Figure 12-15, the following is offered;

– In Figure 12-15, we are editing the $DSHOME/Server/DSEngine/.odbc.ini file.

– In a paragraph entitled “[ODBC]” (generally near the bottom of the file as it arrives from IBM), two optional parameters have been changed. (The remainder of these parameters were already in this file with their given settings.)

  We change the value; Trace = 1

  And we change the value; TraceFile = /tmp/odbctrace.out.

Note: Changes to $DSHOME/Server/DSEngine/.odbc.ini are global to all users of this system. Making this change will disable all ODBC Jobs, putting all Jobs in trace mode and causing them not to complete normally as intended.

If you have to test this in a scope smaller than global, you may make a copy of the .odbc.ini in your home directory. then changes to .odbc.ini will only affect the given user.

After diagnosing the problem at hand, be certain to reverse the modifications made to this file.

These two modifications will change the way our test Job will function; the Job, even if it were to execute successfully, will not complete, and will instead generate diagnostic output to the named file.

This diagnostic file (/tmp/odbctrace.out, from the example in Figure 12-15), is in ASCII text and is available for review. If you are not able to diagnose your issue from this text, contact IBM technical support.

12.3 Summation

In this document, we reviewed or created:

We overviewed how to install IBM Information Server on the RedHat Linux platform. We gave detailed attention to configuring IBM Information Server for use with MySQL as an ODBC data source.

Generally we covered the topics;

- MySQL and network connectivity.
- Linux/Unix ODBC, and testing thereof.
- Configuring IBM Information Server as it related to Linux/Unix ODBC.
- IBM Information Server installation on RedHat Linux.

While MySQL is currently not included in the IBM Information Server Data Direct driver pack, the unofficial position is that MySQL soon will be. The primary impact to customers then would be a consolidated technical support service.

Additional resources:

The IBM Information Server Install Planning, Installation, and Configuration Guide - Revised March 2007, located on line at,

http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp?topic=/com.ibm.swg.im.iis.productization.iisinfsv.install.doc/concepts/wsisinst_overview_container.html

The IBM Information Server FixPak 1A Read Me file is available in the root directory of its installation media.

The IBM Information Server supported platform list, located on line at,

http://www-1.ibm.com/support/docview.wss?uid=swg27008923#platformlinux

In this document, we used the MySQL ODBC driver located at,

http://www.mysql.com/products/connector/odbc/

http://dev.mysql.com/downloads/connector/odbc/3.51.html

http://dev.mysql.com/downloads/connector/odbc/3.51.html#linux-x86-32bit-rpms

A good Unix ODBC driver file site, including detail on the Unix /etc/odbc.ini file is located at,

http://dev.mysql.com/doc/refman/5.0/es/myodbc-configuration-dsn-unix.html

Another good Unix ODBC reference is at,

http://www.openlinksw.com/info/docs/odbcstory.htm

How to compile a MySQL C language program, albeit one that uses a proprietary MySQL API (not standard ODBC) is documented at,

http://www.cyberciti.biz/tips/linux-unix-connect-mysql-c-api-program.html

http://dev.mysql.com/doc/refman/5.0/en/mysql-real-connect.html

http://www.science.uva.nl/ict/ossdocs/mysql/manual_Clients.html#Clients

The MySQL Enterprise Server information and download page is at,

https://enterprise.mysql.com/software/server_4_1.php?show=all

This site is password protected. MySQL will send you a username and password within minutes of registering. You also receive a free 30 day trial to MySQL Enterprise.

Legal statements:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating trademarks that were owned by IBM at the time this information was published. A complete and current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

Other company, product or service names may be trademarks or service marks of others.

Special attributions:

The listed trademarks of the following companies require marking and attribution:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Microsoft trademark guidelines

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel trademark information

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Other company, product, or service names may be trademarks or service marks of others.