The Progress ODBC FAQ
(Final Version 2.1 – 12/22/98)

Geoff Crawford
Innov8 Computer Solutions, LLC.
Phone: (973) 361-4224
FAX: (973) 537-6946
E-Mail: info@innov8cs.com
# Table Of Contents

What is ODBC? ................................................................. 3

ODBC Basics ................................................................. 3

What Are The Different ODBC Versions .......................... 3

ODBC Drivers Available For Progress .............................. 4
  - Version 6 .......................................................... 4
  - Version 7 .......................................................... 4
  - Version 8 .......................................................... 4
  - Version 9 .......................................................... 4

Architectures ................................................................. 5
  - Single Tier ....................................................... 5
  - Multiple Tier .................................................... 5

Setting Up The Drivers .................................................... 6
  - OpenLink ......................................................... 6
    - OpenLink Lite 16 Bit ....................................... 6
    - OpenLink 32 Bit (Small Client and Large Client) .. 6
    - OpenLink ODBC (MT) ...................................... 8
  - Intersolv ......................................................... 10
    - Intersolv DataDirect ODBC ......................... 10
    - Intersolv Version 3.01/3.10 ......................... 11
  - SCO SQL Retriever ............................................. 11
  - Progress ......................................................... 12
  - OID/OIB Setup ................................................. 12
  - ODBC Application Compatibility ....................... 14
  - Progress Version Compatibility ......................... 16

Database Triggers and ODBC ........................................... 17

Special Notes On ODBC 3.0 ........................................... 17

Some Miscellaneous Comments ...................................... 17

The ODBC DataServer .................................................. 19

JDBC and ODBC to JDBC Bridges ................................. 19

Closing Note .............................................................. 19

Resource List ............................................................ 20
  - Books ........................................................... 20
  - Web Sites ....................................................... 20
What is ODBC?

ODBC stands for Open Database Connect and is a Microsoft specification. ODBC provides a standard way of defining data sources and their data access methods. ODBC is designed around SQL and relational databases, but there is nothing preventing translation of the incoming SQL to another language.

The ODBC specification defines low level API calls that any application can make use of for database queries. By writing calls to the API, a reporting writer or other tool can portably access heterogeneous data sources with one set of source code.

ODBC Basics

From a user’s standpoint, all you need to know is that your report writer, or other tool, has been written to make use of the ODBC specification. From there, you need to select and install a Progress ODBC driver. Possible drivers are listed below, along with some notes about architecture and installation. After your driver is installed, you must define each of your Progress databases as a valid ODBC data source. This is done with the standard Microsoft ODBC administrator program.

What Are The Different ODBC Versions

The ODBC specification is currently up to version 3.0. With each different version, new functionality is added. Some of the new specification deals with threading and other behind the scenes make up of the driver, other parts like the new tracing capability are more visible.
ODBC Drivers Available For Progress

Several companies have made drivers for Progress databases, including Intersolv, OpenLink, and Progress themselves. Drivers are specific to the version of Progress:

**Version 6**
- Esker Tun SQL
- Intersolv DataDirect ODBC Version 2.1
- OpenLink ODBC (MT)
- Progress

**Version 7**
- Esker Tun SQL
- Intersolv DataDirect ODBC Version 2.5 (16 bit and 32 bit) *
- Intersolv DataDirect SQLLink (In Beta as of 7/97)
- OpenLink ODBC (MT)
- OpenLink Lite (Small Client, and Large Client) *
- SCO SQL Retriever

**Version 8**
- Esker Tun SQL
- Intersolv DataDirect ODBC Versions 2.5 (16 Bit and 32 Bit) *
- Intersolv DataDirect ODBC Version 3.0 (32 Bit)
- OpenLink ODBC (MT)
- OpenLink Lite (Small Client, and Large Client) *

Note: * - The 16 bit versions seem to be no longer available. The 32 bit versions work with their respective versions of the server and with the 32 Bit ESQL that comes with V8 Client/Networking. Version 7 databases are supported, but Version 7 Client/Networking is not specifically supported. It can be made to work, see notes later in this document.

**Version 9**
- Progress
Architectures

There are two basic architectures employed by the driver makers: single vs. multiple tier. Intersolv’s DataDirect ODBC, OpenLink Lite, and the Progress ODCB Driver are single tier drivers, while the rest are all multiple tier.

Single Tier
Single tier architectures use the driver itself to process the SQL query, implying PC side resolution. The driver connects to the database, sends SQL to the database, does any additional record selection or joining, and then passes the result to the application. Driver connections for Progress require a client product such as Client Networking, 4GL Client, or ProVision to connect to the database, or use its own network protocol for a remote database. The Progress client is responsible for getting the records to the driver where it does the rest of the work. Starting in Version 8, executables separate from the full Client Networking product are shipped for establishing this connection. This smaller client is referred to as the Open Interface Driver and is combined with the Open Interface Broker for multi-user situations.

Multiple Tier
Queries are offloaded by the driver to another application in multiple tier architectures. This secondary layer is generally a networking program talks to a server side component. The server receives SQL requests from multiple network connections, resolves the request through interaction with the database, and returns the data to the PC’s secondary layer. The secondary program must still pass the final results to the driver. While it is not required, almost all multiple tier implementations make direction connections from the server to the database. The server side execution generally provides better performance since only selected records get passed to the client PC. Under Progress Client Networking, records are sometimes passed to the PC for selection, increasing network traffic. The specific circumstances where this happens are version specific, but joins for example, are determined by the client PC under all current versions of Progress.
Setting Up The Drivers

OpenLink

OpenLink’s web site contains pages detailing installation instructions for all of their drivers, including screen shots. Instructions on Progress program setup (OID/OIB setup for example) or ODBC Admin installation is not included. See http://www.openlinksw.com/support/install/proltins.htm for all versions of the Lite Drivers, and http://www.openlinksw.com/support/install/unixserv.htm for the MT Driver server installation.

Be careful of all of OpenLink’s Lite uninstall programs. Some versions appear to entirely wipe out the datasource list, others seem to remove nothing at all. (Progress datasources, DLL’s, icons, etc. are all left alone)

OpenLink Lite 16 Bit

Currently, the instructions are incomplete as the driver installation failed. At the current time this driver also appears to be discontinued.

OpenLink Lite assumes the standard Windows 16 bit Version 2.0 ODBC administration has already been set up. If not, copy into the Windows “SYSTEM” directory ODBC.DLL, ODBSCINST.DLL and ODBCADMIN.EXE from the “Winfiles” subdirectory of the Progress DLC directory. Under Windows NT, this will need to be done even if 32 bit ODBC administration has been installed. Since the Lite Drivers are 16 bit, do not copy the files into \WINNT\SYSTEM32, but \WINNT\STSTEM.

Next run the SETUP.EXE and follow the prompts. A check will be done to ensure the 16 bit ODBC files have been installed. Run the ODBC setup program from the control panel, and choose Add from the list of Data Sources. Select OpenLink Lite 16 Progress from the list of installed drivers. This is the point where the drivers do not appear to be properly installed.

OpenLink 32 Bit (Small Client and Large Client)

There are two versions of the 32 bit driver, the Small and Large Clients. The Small Client uses the OID/OIB setup to talk to Progress, while the Large Client goes directly to the Client Networking executable. If you already have the OID/OIB setup working for some other reason (for example, you are using Actuate), then you may want to use the Small Client since the DLL needed by OpenLink is considerably smaller. Otherwise, the headache of setting up the OID/OIB can be avoided by using the Large Client.

Both drivers expect the 32 bit ODBC admin program to already be installed. It can be downloaded from the Microsoft FTP site if needed, the file name is WT1250.EXE. After downloading the file, copy it into a separate directory. Click on the Start Button to run the self-extracting EXE. Run the SETUP.EXE that was extracted. It is not necessary to
install the entire package. Select the Custom Install and deselect Desktop Drivers, SQL Server, and Oracle unless you want these drivers for some other reason. If the installation is successful, the ODBC admin program will be run automatically. Since the OpenLink Lite driver is not yet installed, simply click on the cancel button.

To install the OpenLink Lite 32 Bit Driver, simply run the self installing PROLTE32.EXE. The installation will make sure that the 32 Bit ODBC has been installed and then prompt you for the directory you would like to have the driver installed.

After choosing the directory you want and viewing the README file, go to the Control Panel and run the ODBC Admin 32. Make sure this is the Admin 32 since you may have another 16 bit ODBC icon as well. Choose the Add button to create a new ODBC data source. Highlight “OpenLink 32 Bit Lite” and click OK. A box with four tabs will appear. Give the database a name and type in a description if desired. Click on the Progress tab to fill in the database connection parameters. What parameters you will need depends whether you have the Small or Large Client version.

For the Small Client, fill in the session options with the OID/OIB connection parameters. This would typically be:

`–SV –S oibservice –H host –N TCP`

Where *oibservice* is the TCP/IP port number for the OIB, and *host* is the name of the remote server. (you must put in the server’s name as found in your “hosts” file, the actual IP address will not work) Only the TCP protocol is supported.

Important Note: *oibservice* is NOT the Progress database port number used in making a client/server connection. You will get an error message from the ODBC driver saying there is a database protocol error if you try to put the Progress database port number here.

For the Large Client, simply leave the session parameters blank.

Both versions of the driver require the same Database Options information. This is where to put the database connection parameters. They would typically be something like:

`-db dbname`

OR

`-db dbname –H host –S dbservice –N protocol`

The first example is for a local database server, and the second example for a remote database. Be careful to consider that the Small Client uses OID/OIB so that the connection parameters must be from the viewpoint of where the OID is connecting from.
For example, if the OID is running on machine “B” and the database is /usr/db/database, then you would want to use the Database Options of:

-db /usr/db/database

If you used parameters like the second options, you would be connecting to the host machine, and then using TCP/IP to loop back to the same machine. This will create a client/server connection and be considerably slower than the self-service option listed above.

Also remember that the same thing goes for the Large Client. Since it uses the Progress executable on the local machine, you want to use the same parameters you would give your 4GL client. If the database is local to the PC, then use just the –db parameter, but if the database is remote to the PC, then use –H, -N, and –S.

Keep in mind that the database server must be running to make the connection, and the “hosts” and “services” files must contain –H and –S information if client/server mode is being employed. You must also start the OIB if you are running the Small Client Version.

Both 32 bit OpenLink drivers require the 32 bit Embedded SQL DLL from Progress Version 8.2 to be installed in the \WINDOWS\SYSTEM directory. (or \WINNT\SYSTEM32 for Windows NT) The OpenLink Web Site erroneously states this file is shipped with their driver. If you have a Progress 32 bit version installed, it can be copied from the DLC “bin” subdirectory. If not, you must obtain this file from Progress or purchase a 32 bit Progress product that contains it.

OpenLink ODBC (MT)

OpenLink’s Multi Tier drivers consist of four pieces of software, the Universal PC Driver, the Request Broker, the Database Agent, and a registration key.

Download a Request Broker based on the server platform. This file will be of the form rqb*.taz. Download a Database Agent for Progress 6 or Progress 7, the file name will be pro6*.taz or pro7*.taz. Finally, fill out the registration form and load a temporary registration key.

Copy all of the files to the directory where you want to install, such as /usr/openlink. Run ./install.sh to extract the files for the archives and then run ./bin/oplcfg. This is the OpenLink configuration script that will tell the driver where you version of Progress is installed. Select either option 8 or option 9, depending on version 6 or 7 of Progress. Fill in the full path name of the DLC directory. The only other parameter OpenLink require is the directory name where the OpenLink driver was installed. This will be automatically set by the install script, but if you move the directory, simple select option 1 in oplcfg.
Lastly, start up the Request Broker by selecting option number 13. This completes the server install.

Download a PC driver based on your client PC operating system. There are basically two choices today, either the Windows 3.1 version or the Windows 95/NT version. Install the driver by running SETUP.EXE and follow the instructions.

Run the ODBC setup program to define a data source. You must type in the name of the database, and any other startup parameters. Remember that the Multi Tier driver executes on the server, so that all parameters and file names must be from the server’s viewpoint.
Intersolv

Intersolv DataDirect ODBC

Directions for installing all Intersolv DataDirect series are the same with the exception of the environment variables needed. (Please see the special note for the Version 3.01 Driver)

Run the SETUP.EXE program and enter your contact information and serial number. choose which drivers you want to install. Unlike OpenLink’s Lite driver, Intersolv will automatically install the ODBC administration if needed. By default, Intersolv installs all of the drivers in the pack, overwrites older drivers, and creates default Data Sources. Modify these parameters as desired.

***** The version of PROESQL.DLL shipped with Intersolv 2.5 will not work with Progress Version 8.1 and above. You must copy \DLC\BIN\ESQL01.DLL to your Windows directory and rename it as PROESQL.DLL. Since this is a 16 bit DLL, it is not shipped with Version 8.2 but is available on the Progress FTP site. While 8.2 does seem compatible with the Intersolv 2.5 driver, it is unsupported. *****

After all files are copied to the appropriate directories, you must create an ODBC data source.

The ODBC icon will now appear in the Windows Control Panel. Run it and select Add from the list of Data Sources. Select Intersolv 2.5 Progress from the list of install drivers. You must now enter 3 sets of parameters, one each for the ODBC information, the OID/OIB information, and the Database Connect Options.

The Data Source name is arbitrary but will be seen when you select this database in your ODBC application. Enter a description, and the database name without any directory names. This is important because most ODBC applications will use this parameter as the database alias. Enter an option Username if security is turned on.

Next enter the OIB/OID parameters starting with either a Local or Remote location. The Protocol is chosen for you based on the OIB location. Only WIPC is supported for Local connections, and only TCP/IP for Remote. Since Progress Version 8.2 no longer supports WIPC, you must always use TCP/IP with the Intersolv Driver. For standalone Windows V8.2 databases, this is the only server protocol provided. Fill in the service name used to start the OI Broker. (Remember to use the OIB service number, not the Progress database service number.) For local connections, this parameter is not used, but must be filled in anyway. (any name will work)

Lastly, fill in the parameters the OID must use to connect to the database. Select either Direct or Via Server for host mode or client/server respectively. Direct connections further require the Database Path and Operating System to be filled in. The path is only the directory name and will be concatenated with the Database Name entered in the
ODBC parameter section. Fill in the Protocol, Host and Service Names for client/server connections.

If you are using a remote OID/OIB connection, you may not have set up the necessary environment variables. These are the same variables you would need to set up a Progress client session, namely DLC, PROMSGS, and PROCFG. Since you are not executing Progress 4GL programs, PROPATH is not needed. For Windows 3.11 or Windows 95 computers, set the parameters in your AUTOEXEC.BAT. Windows NT requires the parameters to be set in the DOS Console.

After the OIB and database servers are started, the driver is ready to use. See the OID/OIB section for further notes.

Intersolv Version 3.01/3.10

***** The Intersolv 3.01 AND 3.10 drivers have been shipped with an alternate variation of the PROSESQL.DLL. This driver is meant to help you separate the version of Progress you use for the driver from the Version you program with. This is accomplished by using a different set of environment variable names. This driver requires you to set IDLC, IPROMSGS, and IPROCFG. Without these variables set, you will most likely receive the message “Unable to find Progress PROMSGS File”. *****

SCO SQL Retriever
SQL Retriever is a Multi Tier type driver, similar to OpenLink’s MT, but for Version 7 Progress only. There are two versions, the full version and the “Lite”. SQL Retriever is a part of the whole SCO Vision Windows to Unix connectivity suite. The full product contains file and print sharing facilities such as their “Unix Neighborhood” while the Lite does not. SCO says the only other feature you would lose with the Lite driver is the ability to run over serial lines.

Installation is a two step process, once for each PC, once for the server. The PC install is uneventful, just follow the prompts and the defaults should be fine. Setting up an ODBC source though you will see no spot for parameters. SCO apparently believed that you could put them in the database name, which may work for some applications. But for many this will create a database name that the application will not see as legal. (particularly spaces, and dashes) SCO does allow a spot for such parameters, but for some strange reason it is not accessible by default. You will find a Dboptions space in the DSN setup, but greyed out. The solution is to go to the registry and look for HKEY_LOCAL_MACHINE/SOFTWARE/SCO/VWODBC.INI/Progress/Dboptions and set this to yes. Although I have not tested it, SCO Tech Support says that 16 bit installs will have a VWODBC.INI file in the windows directory. In the section with [Progress], enter a line with Dboptions=yes. Other than that, creating the DSN doesn’t seem to be more complicated than setting the host name, Unix path name for the database, and the –H –N and –N parameters for the database connection.
On the server, the installation scripts are slightly more complicated. A SCO license manager will be installed that will scan the network for existing servers. Be careful, this will go through your entire hosts file and could take a while deciding each one of your PC’s are not SCO manager hosts. The installation should not be much more difficult than letting it setup the proper services.

While SQL Retriever does execute on the server, the server component comes only in very specific Progress V7 minor releases. Unlike OpenLink’s MT driver, there is no way to probuild a new server for you specific platform. SCO has chosen to simplify the setup by shipping only one version of the binaries. So under most circumstances you will need to use a client/server connection due to the differences in shared memory versions among Progress sub-releases. That means that your server must have Progress server networking installed, and the network broker started. Just follow the Progress documentation, create a port number in /etc/services and use –H –S and –N when you use proserve.

Progress

The Version 6 driver is discontinued and therefore dropped from the FAQ. The new Skywalker ODBC driver needs no special explanations for setup. There is nothing other than starting a server that needs to be done on the database side. On the client side, simple run SETUP.EXE and you’re done. Now that was the way it was meant to be all along.

OID/OIB Setup

Since the current crop of single tier drivers all use OID/OIB, an explanation is in order. The Open Interface Broker is basically a network program that accepts connections and spawns an Open Interface Driver session. The OID then sends data back to the program at the other end of the connection. OID’s can access both local and remote databases using all of the methods that a normal Progress session can.

Before actually starting any servers, it is critical to set up a set of environment variables. You must set PROOIBRK, PROOIDRV, DLC, PROMSGS, PROCFG, and PROSTARTUP. For Unix systems, these need to be set before the OIB is started. That could be either in your .profile/.login or in an OIB startup script. For Win 3.1 and Win 95 systems, put them in your AUTOEXEC. Under Windows NT you will need to go to the Environment tab of the Control Panel’s System Icon. Also be sure to include both the DLC directory, and the DLC/bin directory in your PATH.

Unix sample:

DLC=/usr/dlc
export DLC
PATH=$PATH:$DLC:$DLC/bin
export PATH
PROMSGS=$DLC/promsgs
PROCFG=$DLC/progress.cfg
PROSTARTUP=$DLC/startup.pf
PROOIBRK=$DLC/bin/_prooibk
PROOIDRV=$DLC/bin/_prooidv

Export PROMSGS PROCFG PROSTARTUP PROOIBRK PROOIDRV

Windows 95 AUTOEXEC.BAT Sample:

PATH=%PATH%;C:\DLC;C:\DLC\BIN
DLC=C:\DLC
PROMSGS=C:\DLC\PROMSGS
PROCFG=C:\DLC\PROGRESS.CFG
PROSTARTUP=C:\DLC\STARTUP.PF
PROOIBRK=C:\DLC\BIN\OIBRKR.EXE
PROOIDRV=C:\DLC\BIN\OIDRVR.EXE

To set up the OIB, first make sure your database has a multi-user server already started. On Windows standalone machines up until Version 8.2, this can be the WIPC server by using the command:

proserve C:\DATABASE\DBNAME –N WIPC

Edit your TCP/IP services file to set aside a TCP protocol port for the OIB’s use. The actual file location varies from system to system, /etc/services for most Unix variants, \WINDOWS\SERVICES or \WINNT\SYSTEM\DRIVERS\ETC\SERVICES for Microsoft TCP/IP, etc. The port you choose is arbitrary, but it must not be in use by another application. It should look something like:

oibservice 6400/TCP

Next, actually start the OIB using the name you chose above as the service:

oibrkr –SV –S oibservice –N TCP

Important Note: oibservice is NOT the Progress database port number used in making a client/server connection. You will get an error message from the ODBC driver saying there is a database protocol error if you try to put the Progress database port number here. The two service number must be unique.
**What Works With What**

Here is a list of combinations of third party applications, and whether or not they are known to work with ODBC drivers from the various vendors. Please note the list represents the experiences of users trying to run these applications. In some cases, abandoning a project or specific driver may have been considered preferable to spending the time to get the application working. Please do not take a “DOES NOT WORK” designation as the final word, YMMV. Also keep in mind that a designation of “WORKS” does not mean the environment is not without its problems.

**ODBC Application Compatibility**

<table>
<thead>
<tr>
<th>Application</th>
<th>Viability</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Basic</td>
<td>OK</td>
<td>Requires “SQL-Passthrough” with the Intersolv Driver for array support. OpenLink supports arrays only in its Lite drivers if a special program is run against the database.</td>
</tr>
<tr>
<td>Visual C++</td>
<td>OK</td>
<td>Access requires at least one unique index per file, 256 field per file limit. Too many indexes per file also seems to confuse Access. Arrays do not seem to work. The “Group By” check box appears to greatly enhance the performance of reports. YMMV.</td>
</tr>
<tr>
<td>Access</td>
<td>OK</td>
<td>Requires at least one unique index per file, 256 field per file limit. Too many indexes per file also seems to confuse Access. Arrays do not seem to work. The “Group By” check box appears to greatly enhance the performance of reports. YMMV.</td>
</tr>
<tr>
<td>Excel 95</td>
<td>OK</td>
<td>Requires 32 bit driver</td>
</tr>
<tr>
<td>Excel 97</td>
<td>OK</td>
<td>Requires ODBC 3.0 and 32 bit driver (Intersolv Version 3.01 or OpenLink Lite 32/MT 32) ODBC DLL’s Version 3.0.28.22 do not seem to allow re-editing the query. The datasource name must either be “Default”, or you must create a DSN file manually. OpenLink MT Series creates alternate file names when you use the setup.p (for array support) which do not work in the Query Wizard. Manual entry of SQL will resolve the problem.</td>
</tr>
<tr>
<td>Delphi</td>
<td>Limited</td>
<td>Sporadic reports of it not</td>
</tr>
</tbody>
</table>
working but possibly problems with outdated drivers. Delphi 2.0 requires BDE 3.51 or higher. The OpenLink MT broker on the server must be restarted if a Delphi V3 session crashes.

<table>
<thead>
<tr>
<th>Software</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BusinessObjects</td>
<td>Appears OK</td>
</tr>
<tr>
<td>Clear Access</td>
<td>OK</td>
</tr>
<tr>
<td>ReportSmith 3.0</td>
<td>OK</td>
</tr>
<tr>
<td>Crystal Reports</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>Newer versions of Crystal generate SQL-92 which Progress does not support. There is a special DLL available to solve this problem. Intersolv 3.01, and 3.10 seem to require different DLL’s. The situation keeps changing too fast to be described here. See Seagate’s Web Site for current info.</td>
</tr>
<tr>
<td>Impromptu</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>There is a Progress specific bug in the way Version 5 creates SQL. No fix at the time of writing.</td>
</tr>
<tr>
<td>Apptivity</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>Now shipped with OpenLink MT and an ODBC/JDBC bridge. Requires at least one unique index per file. Appears to work with Intersolv ODBC/JDBC bridge as well.</td>
</tr>
<tr>
<td>Actuate 2.0</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>Requires 32 bit driver, unlike native connection does not require OIB started in the same directory as the database.</td>
</tr>
<tr>
<td>Esperant 3.0</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>Will not work with tables that have a hyphen in the name.</td>
</tr>
</tbody>
</table>
## Progress Version Compatibility

<table>
<thead>
<tr>
<th>ODBC Driver</th>
<th>Version</th>
<th>Progress Server Version(s)</th>
<th>Progress Client Version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersolv ODBC</td>
<td>2.1</td>
<td>All V6</td>
<td>V6.2, V6.3</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>V6.3, All V7</td>
<td>V7.1, V7.2, V7.3</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>All V7, All V8</td>
<td>V8.0, V8.1, V8.2a(*)</td>
</tr>
<tr>
<td></td>
<td>3.01, 3.10</td>
<td>All V7, All V8</td>
<td>V8.2</td>
</tr>
<tr>
<td>OpenLink Lite</td>
<td>Lite</td>
<td>V6.3, All V7, All V8</td>
<td>All V7, V8.0, V8.1</td>
</tr>
<tr>
<td></td>
<td>Lite (32 Bit)</td>
<td>All V7, All V8</td>
<td>V8.2</td>
</tr>
<tr>
<td>MT</td>
<td>All V6, All V7, All V8</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Progress</td>
<td>V9</td>
<td>V9</td>
<td></td>
</tr>
<tr>
<td>SCO SQL Retriever</td>
<td>4.1</td>
<td>V7.3C (ODT &amp; UnixWare) and V7.3A (Sun OS) via shared memory, all other V6 and V7 via Client/Server</td>
<td>N/A</td>
</tr>
<tr>
<td>Esker Tun SQL</td>
<td>All V6</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

* - Unsupported Configuration but does work

While many of the above configurations work, not all support shared memory connections for best performance. All environments where the client is one major version of Progress, and the server is another will require a client/server connection.
Database Triggers and ODBC

ODBC does in fact cause Progress database triggers to fire. It’s very important to understand the architecture to understand where exactly they fire. For drivers that use the OID/OIB, such as Intersolv, and the OpenLink Lite series, they fire inside of the OID. Since the OIB starts the OID, it will inherit both the environment variables and the current directory. This means all triggers will be referenced (as if you set –trig) to the current directory where you were when the OIB was started, and will have a PROPATH the same as was set when the OIB was initiated.

Multi-tier drivers use a client agent to retrieve data, and will have the trigger fire inside of the agent. OpenLink uses a “probuild” client in its MT drivers and does not have full functionality. It is important to write Progress code compatible with this cut down client.

OpenLink presents a very special situation with triggers because OL renames the first connected database to “PUBLIC”. This would not normally cause any great problem except that this alias will now most likely be different from the .r code produced for your triggers. There are two possible solutions. One is to connect to the database with it aliased as “PUBLIC” (ie. mpro dbname –ld public) and compile a version of your triggers specially for ODBC. Perhaps an easier and more universal solution is to connect a blank database as the first one in your DSN. OpenLink only aliases the first database, and so any subsequent databases will appear with the correct name.

Special Notes On ODBC 3.0

MS Query
It appears that the latest version of MS Query is giving people quite a bit of a headache due to the new ODBC 3.0. This version of Query expects that the data source is a “File” DSN instead of a User DSN. I have not been able to find official documentation on this but I can did find a quick note in the help file. It seems File DSN’s are meant as a way of sharing the DSN over a network. That does not mean sharing the data over the network, but the connection options, database names, etc. That way you can do the setup on one PC and anytime you need a modification you have only one place that would need to be changed.

I have not found the exact contents of what the DSN format is, but it looks something like an older .ini file. For what ever reason they seem to be stored in “\Program Files\Common Files\ODBC”. What I’ve found that seems to work very well is to download a copy of the ODBC SDK from the Microsoft site. (http://www.microsoft.com/data/ODBC/download/SDKDownload.htm) It contains a utility called CONVDSN.EXE. After executing this program once, all User DSN’s will have a corresponding File DSN. In addition, future User DSN’s will also be copied as they are created.

Some Miscellaneous Comments
Here are some additional tidbits that may help along the way.
Intersolv and OpenLink treat the problem of arrays differently. Without any user intervention, Intersolv automatically converts the field into a field name with an underscore and the extent number. OpenLink requires you to run setup.p and then use the output as the tableview file. If you need to switch drivers for some reason, this could be quite a problem unless your tool is generating SQL on the fly. It’s not such a problem to go from Intersolv to OpenLink because you can always modify the .DAT, but the other direction really doesn’t have a quick solution.

By all means, add –NL to either the DSN Database Options, or your PROSTARTUP file when using Intersolv. It appears that when using certain functions to look at the schema, a limbo lock is left. Using the –NL option seems to solve this.

Be careful in your trigger code. SQL does have a very interesting difference in the way the undefined value (ie ?) is used. Also make sure that your triggers are in the correct path of where they are executing. That’s the OID for Intersolv and OpenLink Lite Small Client, Client Networking for OpenLink Lite Large Client, and the ODBC broker for OpenLink Lite MT, SCO SQL Retriever, or Esker.

Progress supplies the 32 bit OIB and OID (oibrkr32.exe and oidrvr32.exe) with Version 8.1, but they forgot files they depend on. Two DLL’s, category.dll and promsgs.dll are missing.

Having trouble finding a programming error? Try using ODBC’s Version 3.0 tracing facility. Just go into the ODBC 32 icon in the control panel and click on the Tracing tab. You can then specify a log file to see where and when the SQL calls occur. If that doesn’t help by all means get the ODBC SDK from Microsoft. There are a load of SQL and ODBC test utilities included.

There are some cases where it appears the name of the database is too long, and you get something like “SQL MAX OWNER NAME LEN Exceeded Error”. At least one instance of this was reported with Delphi and the BDE. It appears that this is actually a server side environment variable that can be set. A simple:

```
SQL_MAX_OWNER_NAME_LEN=80; export SQL_MAX_OWNDER_NAME_LEN
```

Reportedly fixed the problem. There are other similar style errors that have occurred, and it might be worth while to try setting an environment variable as suggested by the error message.

The OID is a sort of headless Progress client, but it has significantly been stripped down. It apparently will only receive the following parameters from the OIB: -h, –RO, -NL, and –yy. If you need any other options, be sure to start the OIB with the PROSTARTUP parameter pointing to a .pf file. See Progress KnowledgeBase entry 18088.
**The ODBC DataServer**

In addition to connecting third party tools to Progress database, ODBC is also used as a way to connect Progress to third party databases. The Progress ODBC DataServer theoretically allows use of a Progress client against any ODBC compatible data source. In reality, success will depend on the level and quality of ODBC support from both the third party database, and the ODBC driver involved. Progress officially supports Microsoft Access Versions 1.0, 1.1 and 2.0 using Microsoft ODBC Drivers, and Informix On-Line with Intersolv ODBC Version 2.0 or higher and Informix I-Net 4.1 or later.

Future directions indicate the ODBC DataServer will become an increasingly important DataServer product. Progress has suggested that future support of DB2, MS SQL Server, and perhaps Sybase may only be through the ODBC DataServer. The Progress web site should be consulted for the correct details:

http://www.progress.com/core/progress/odbcds.htm

**JDBC and ODBC to JDBC Bridges**

If you’re programming in Java, there is a standard similar to ODBC for Java. JDBC drivers are also available for Progress. They are generally of two different types, true JDBC drivers and ODBC to JDBC Bridges.

True JDBC involves your Java program making a connection to a database over a URL. ODBC bridges work the same way from the program side, but connect locally through the PC’s ODBC driver. The database can still be remote, just the ODBC must be set up locally.

**Closing Note**

As has been noted earlier, the new Progress ODBC driver that is coming out with Skywalker has none of the horrible setup problems the other drivers suffer from. Due to the enormous benefits of the easy setup (not to mention the direct SQL access), this FAQ is now moot. While ICS has not specifically endorsed a driver in the past, that is no longer the case. The native connection seems absolutely unbeatable in all categories, and is the only driver we as a company intend to deal with in the coming future.
Resource List

If you need more information on ODBC and Progress, try the following:

**Books**

**Progress Manuals**
- Inside ODBC by Kyle Geiger (Microsoft Press 1-55615-815-7)
- Microsoft ODBC 3.0 Software Development Kit and Programmer’s Reference (Microsoft Press 1-57231-516-4)

**Web Sites**

- **Intersolv** [http://www.intersolv.com/products/dataconnectivity.htm](http://www.intersolv.com/products/dataconnectivity.htm)
- **OpenLink** [http://www.openlinksw.com](http://www.openlinksw.com)
- **Microsoft** [http://www.microsoft.com](http://www.microsoft.com)

**Acknowledgements**

Several people have been kind enough to describe their ODBC experiences. In addition, ODBC is a regular point of discussion on the Progress Email Group. The distilled knowledge from these two sources, along with the resources listed above, form the basis of this document.

While a list of contributors could never be complete, but here are some people who have shared their experiences:

- George Potemkin ([G.Potemkin@CSBI.spb.su](mailto:G.Potemkin@CSBI.spb.su))
- Clinton Hastings ([chastings@gcl.com.au](mailto:chastings@gcl.com.au))
- Jeff Lischka ([105503.401@compuserver.com](mailto:105503.401@compuserver.com))
- Glen West ([gwest@corp.atl.com](mailto:gwest@corp.atl.com))
- Michael Quattlebaum ([mquattle@ecs-inc.com](mailto:mquattle@ecs-inc.com))
- Murray Hermann ([murray.hermann@star.com.au](mailto:murray.hermann@star.com.au))
- Rune Sandbakken ([rune@Oslo4GW.Norway.NCR.COM](mailto:rune@Oslo4GW.Norway.NCR.COM))
- Pat Caulfield at SCO
- Emmon and Paul at OpenLink