

Reference Manual

Version E.57



M.B. Foster Software Labs Inc.

Head Office:	82 Main Street South Chesterville, Ontario K0C 1H0 Canada
Telephone:	(613) 448-2333
Fax:	(613) 448-2588

NOTICE

The information in this manual is subject to change without notice.

LIMITATIONS ON WARRANTIES AND LIABILITY

M.B. Foster Software Labs makes no warranties, either express or implied, regarding this manual or the computer software package described in this manual, its merchantability or its fitness for any particular purpose. The exclusion of implied warranties is not permitted by some states.

COPYRIGHT

This manual is copyrighted by M.B. Foster Software Labs Inc., with all rights reserved. Under the copyright laws, this manual may not be copied, in whole or in part, without the written consent of M.B. Foster Software Labs Inc. Under the law, copying includes translating to another language.

ODBCLink/SE is a trademark of M. B. Foster Software Labs, Inc.

Impromptu, PowerPlay, Axiant and PowerHouse PDL are trademarks of Cognos Incorporated.

Jetform is a trademark of Jetform Corporation.

PowerBuilder is a trademark of Powersoft Corporation.

Reflection, Reflection for Windows, Reflection Network Series and PPL (Process-to-Process Link) are trademarks of WRQ, Inc.

IMAGE/SQL, ALLBASE/SQL, Query, MPE/XL and MPE/iX are trademarks of the Hewlett-Packard Company.

Windows, Windows for Workgroups, Microsoft Access, Visual Basic, Visual C++, Visual FoxPro, Excel 5.0, MS-Query and Word for Windows are trademarks of Microsoft Corporation.

Dr. DeeBee Spy © 1995 Syware, Inc., All rights reserved.



ODBCLink/SE offers the following features:

- ⊗ **ODBC Level 2 compliance (except SQLBrowseConnect, SQLExtendedFetch and SQLSetPos)**
- ⊗ **Access to IMAGE/SQL database environments (DBE)**
- ⊗ **Access to ALLBASE/SQL DBE**
- ⊗ **Access to a single DBE**
- ⊗ **Support for 16 and 32 bit desktop clients**

The additional features of DataExpress for Client-Server are:

- ⊗ **Connectivity to TurboIMAGE and Oracle databases**
- ⊗ **Serial/Modem connection in both 16 and 32 bit**
- ⊗ **Create a mini-dictionary for access to Image data sets, KSAM or MPE files.**
- ⊗ **Read/Write KSAM, MPE and Suprtool files**
- ⊗ **Support for PowerHouse PDL dictionaries**
- ⊗ **Support for TPI keyword indexes used by Omnidex and Superdex**
- ⊗ **Enhanced security via field level access controls**
- ⊗ **Multi-threaded access to host-based data ensures fast and reliable data access**
- ⊗ **Use-able in MTS/IIS environments**
- ⊗ **Secure reliable access to data through browser clients**

Introducing MBF-Console

For monitoring and controlling ODBCLink/SE or ODBCLink activities such as:

- ⊗ **Providing a list of each client connection to the server**
- ⊗ **Changing logging levels in real time**
- ⊗ **Start/stop logging for a single process**
- ⊗ **Kill a single process as required**

Require access to these features? Upgrade to DataExpress for Client-Server before November 30, 1999 and receive 15% off the full retail price AND our two-day Express Start ODBCLink training class FREE to help get you up and running!

(This offer only valid in Canada and the USA. For international program details, contact

globalsales@mbfoster.com)

ODBCLink/SE offers the following features:

- ODBC Level 2 compliance (except SQLBrowseConnect, SQLExtendedFetch and SQLSetPos)
- Access to an IMAGE/SQL database environment (DBE)
- Access to an ALLBASE/SQL DBE
- Access to a single DBE
- Support for 16 and 32 bit desktop clients

If you require any of the following features:

- Support for Serial and/or Modem connections
- Read/Write access to KSAM files
- Read/Write access to MPE files
- Native read/write access to TurboIMAGE databases
- Read/Write access to Suprtool files
- Support for PowerHouse dictionaries including PDL and PowerHouse subfiles support
- Support for TPI keyword indexes used by Omnidex and Superdex
- Access to Oracle databases on HP3000 and HP9000 platforms
- Access to Multiple DBE's
- Access to HP intrinsics and procedures via a remote procedure call mechanism
- Enhanced security via field level access controls

Please contact M. B. Foster Associates at 613-448-2333 or 1-800-ANSWERS (800-267-9377) or use the enclosed faxback sheet to order a fully functional evaluation copy of ODBCLink.



FAX BACK
to
M. B. FOSTER ASSOCIATES LIMITED
613-448-2588

NAME _____
TITLE _____
COMPANY _____
ADDRESS _____
CITY/STATE/ZIP _____
COUNTRY _____ E-MAIL _____
PHONE _____ FAX _____

Please check the features that you are most interested in:

- | | |
|--|--|
| <input type="checkbox"/> Serial/Modem connection | <input type="checkbox"/> KSAM file access |
| <input type="checkbox"/> MPE file access | <input type="checkbox"/> TurboIMAGE access |
| <input type="checkbox"/> PowerHouse PDL support | <input type="checkbox"/> Suprtool file access |
| <input type="checkbox"/> Oracle access | <input type="checkbox"/> TPI interface support |
| <input type="checkbox"/> Multithreading | <input type="checkbox"/> Enhanced security (field level) |
| <input type="checkbox"/> MBF Console | |

I am using the following desktop product(s):

- | | | |
|--------------------------------------|---------------------------------------|--|
| <input type="checkbox"/> MS Access | <input type="checkbox"/> Visual Basic | <input type="checkbox"/> Fox Pro |
| <input type="checkbox"/> MS Query | <input type="checkbox"/> Paradox | <input type="checkbox"/> Jetform |
| <input type="checkbox"/> Impromptu | <input type="checkbox"/> Axiant | <input type="checkbox"/> PowerBuilder |
| <input type="checkbox"/> Lotus | <input type="checkbox"/> Excel | <input type="checkbox"/> Crystal Reports |
| <input type="checkbox"/> Other _____ | | |

- 16-bit 32-bit

I am interested in a demo, please send one immediately!

HPCPUNAME: _____
HPSUSAN NUMBER: _____
Media Required: DAT Tape

**IN A HURRY? GET YOUR QUESTIONS READY AND
CALL 1-800-ANSWERS (267-9377) NOW!**

We can also be reached at:

PHONE: 613-448-2333
FAX: 613-448-2588
sales@mbfoster.com

Table Of Contents

Overview and Specifications	1 - 1
ODBCLink/SE Overview	1 - 1
ODBC Compliancy Level.....	1 - 2
Client Environment	1 - 2
Server Environment.....	1 - 2
The Connecting Link.....	1 - 2
Microsoft ODBC Overview	1 - 3
ODBCLink/SE System Requirements	1 - 4
Preparing the Database Server.....	2 - 1
Verify Software Version Numbers	2 - 1
Verify the Connection Assurance Parameters	2 - 1
Starting and Stopping the Listener	2 - 2
Installing ODBCLink/SE on the PC.....	3 - 1
Downloading ODBCLink/SE to the PC.....	3 - 1
Extracting the ODBCLink/SE file	3 - 3
Running the SETUP program	3 - 3
Notes on 16 or 32-bit Access	3 - 6
Configuring Data Sources.....	4 - 1
Data Source Configuration	4 - 2
Choosing an ODBC Driver	4 - 2
ODBCLink/SE Setup	4 - 3
Setup for MPE/iX	4 - 4
Setup for HP-UX.....	4 - 5
Select Translator	4 - 7
ODBC Translator	4 - 7
Successful Completion	4 - 7
Migrating HP PCAPI Data Sources.....	4 - 8
Application Development	5 - 1
Using Stored Procedures	5 - 2
Performance Considerations	5 - 3
Turning AutoCommit On/Off.....	5 - 4
Supported Functions.....	5 - 5
Isolation Levels	5 - 7
Supported Data Types.....	5 - 8

Note on Using BLOBs	5 - 9
Using ODBCLink/SE With ODBC Applications	6 - 1
Updating Data	6 - 1
Using Cognos Impromptu	6 - 2
Special Notes for Users of Impromptu	6 - 3
Using Lotus 123 Release 5.0	6 - 6
Using Microsoft Access 7.0	6 - 7
Notes for Users of MS-Access	6 - 9
Using MS-Query	6 - 10
Using Visual Basic (2.0 or higher)	6 - 12
Connection Examples	6 - 13
Connect Using SQL API	6 - 13
Connect Using SQLAPI in VB 2.0-6.0	6 - 13
Connecting with DAO's	6 - 16
Connecting with RDO's	6 - 17
Calling ODBC Functions Programmatically	6 - 18
Troubleshooting	7 - 1
Terminating an ODBC Connection	7 - 1
Using Microsoft ODBC Test (32-bit)	7 - 2
Verifying Software Version Numbers	7 - 3
Verifying the Client-Side Components	7 - 3
Deleting Unwanted Drivers	7 - 4
Deleting a Translation DLL	7 - 5
Tools on the Database Server	7 - 7
Host Logging	7 - 9
Tools on the PC Client	7 - 10
ODBC Call Tracing using ODBCLink.LOG	7 - 10
ODBC Call Tracing using Dr. DeeBee Spy	7 - 10
Modifying the System Registry and ODBC.INI Files	7 - 12
Error Messages	7 - 13
Implementation Notes	A - 1
Primary key name returned by SQLStatistics	A - 1
Reading or writing to ALLBASE/SQL LongVarBinary items	A - 1
Maximum number of statements	A - 1
ALLBASE/SQL and IMAGE/SQL Restrictions on the ODBC Grammar	A - 2
Using the ANSI Character Set	A - 2
Unsupported ALLBASE/SQL and IMAGE/SQL Statements	A - 3
ALLBASE/SQL and IMAGE/SQL Statements That Work Only with Embedded SQL	A - 4

ALLBASE/SQL and IMAGE/SQL Statements Replaced by Functions on the PC Client	A - 4
Creating a DBEnvironment	B - 1
On the HP3000 Database Server	B - 1
On the HP9000 Database Server	B - 3
List of Installed Files	C - 1
16 Bit Driver Client - For Windows 3.1 or Windows 3.11	C - 1
32 Bit Driver Client - For Windows 95	C - 3
32 Bit Driver Client - For Windows NT	C - 5
MPE/iX Host	C - 7
HP-UX 9.x Host	C - 7
HP-UX 10.x Host	C - 7
Index	I - i

Chapter 1 Overview and Specifications

ODBCLink/SE Overview

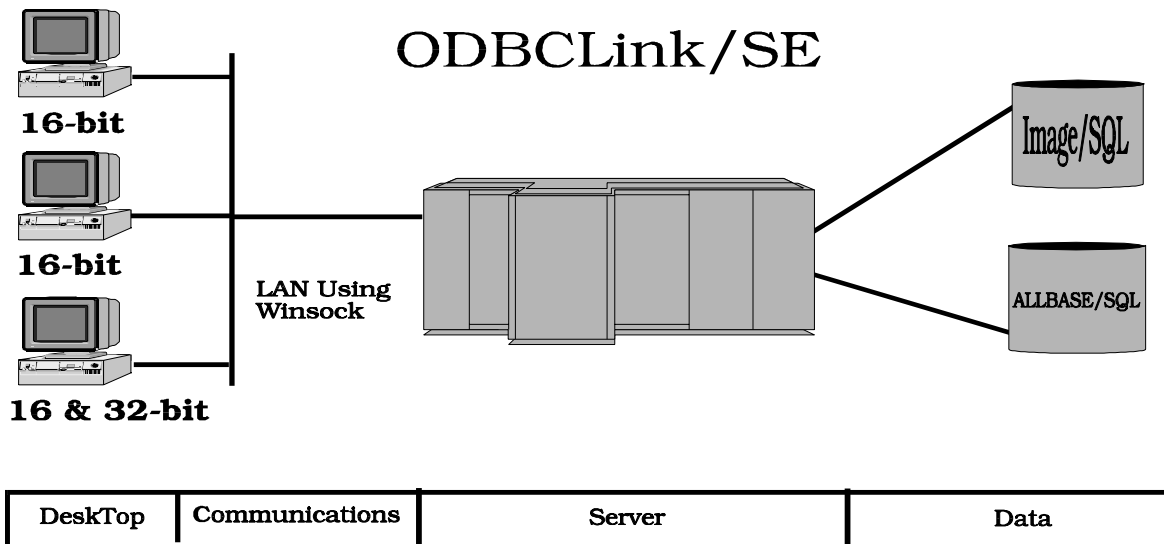
ODBCLink/SE is an implementation of Microsoft's Open Database Connectivity (ODBC) interface that enables Microsoft Windows based applications and tools to access IMAGE/SQL on the HP3000 and ALLBASE/SQL on the HP3000 and HP9000, in a client/server environment.

In this environment, application developers and end-users can take advantage of the PC's graphical user interface (GUI) and processing power, while relying on the security, integrity, and database management capabilities of ALLBASE/SQL and IMAGE/SQL.

ODBCLink/SE runs under MPE/iX 5.0 or higher on the HP3000 server, or under HP-UX Versions 9.x and 10.x on the HP9000 Series 7xx and 8xx servers. ODBCLink/SE runs under Microsoft Windows 3.1 and 3.11, Windows95, or WindowsNT, on the client. Connection via Winsock is available in 16 and 32-bit versions, regardless of the host environment.

ODBCLink/SE can be used in two ways: either by direct calls to the Windows DLL from a Windows program (Foxbase, etc.) or else through an ODBC-compliant application such as Microsoft Access, Visual Basic, etc.

ODBCLink/SE supports the ODBC LongVarBinary data type for binary large objects (BLOBs), such as compressed photographs or document images. These are stored on the host as ALLBASE/SQL LongVarBinary items.



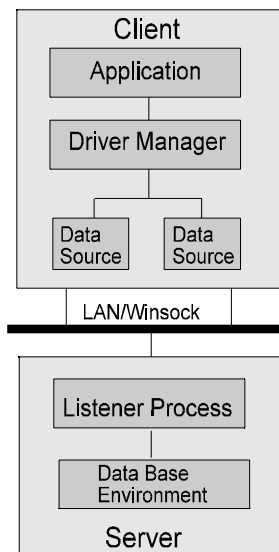
ODBC Compliancy Level

ODBCLink/SE is ODBC Level 2 with the following exceptions:

SQLBrowseConnect, SQLExtendedFetch, SQLSetPos are not supported. Scalar Functions are not support.

SQL_BIT, SQL_TINYINT and SQL_BIGINT data types are not supported.

Client Environment



ODBCLink/SE and application software reside on the PC client. With ODBCLink/SE and applications software you can develop SQL applications, generate reports, and query ALLBASE/SQL or IMAGE/SQL on the database server. The tested client applications include:

Cognos Axiant
 Jetform
 Lotus 123
 Crystal Reports
 MSAccess
 Visual Basic
 Visual FoxPro
 and many more.

Cognos Impromptu
 Cold Fusion
 PowerBuilder
 MSEXcel
 MSQuery
 Visual C++
 Paradox

Server Environment

The networked HP 3000 Series 900 with ALLBASE/SQL and/or IMAGE/SQL or the HP9000 Series 7xx or 8xx with ALLBASE/SQL provide the relational database environment on the server. Security is provided by the MPE/iX or HP-UX logon system and ALLBASE/SQL or IMAGE/SQL.

The Connecting Link

ODBCLink/SE links the client to the server with a collection of dynamic link libraries (DLLs) and other files that reside on the client and run under Microsoft Windows 3.11, Microsoft Windows95 or Microsoft WindowsNT. ODBCLink/SE routes requests, made from the PC client application, over the network to an ALLBASE/SQL or IMAGE/SQL database on the database server, and returns replies to the client application.

A listener process establishes the connection between the PC client application and the target database on the server. The listener works with HP ThinLAN 3000/iX on the HP3000 or ARPA Services on the HP9000.

Microsoft ODBC Overview

ODBC applications make functions calls into ODBC.DLL (the Microsoft 16-bit ODBC driver manager), or ODBC32.DLL (the 32-bit driver manager). Since a single application can make calls to more than one ODBC database (e.g. Sybase, Microsoft SQL server) the driver manager is in charge of routing ODBC calls to the appropriate driver. The ODBC.INI file contains a list of drivers and associated data sources (one driver may have several data-sources). In Windows 95 or NT, some of this information is kept in the system registry.

To connect to an ODBC database, you call SQLDriverConnect (or you let your ODBC-compliant application call it for you) and specify your data-source name in the connection string variable, which is in the form "DSN=data_source". The driver manager reads ODBC.INI or the registry, to find out which driver handles that data source and passes the call on to the driver. The driver opens the communication channel and does all necessary initialization.

The ODBC.INI file (or the ODBC System Registry) is maintained through the ODBC Control Panel application ODBCADM.EXE , or the 32-bit ODBC Control Panel application ODBCAD32.EXE. The control panel application will call the setup DLL for the driver it is accessing (ODBCLS16.DLL or ODBCLS32.DLL in the case of ODBCLink/SE) which will update ODBC.INI or the system registry.

ODBCLink/SE System Requirements

HP3000 Server Requirements:

Operating System Requirements:

MPE/iX Version 5.0 or Version 5.5

Database Requirements :

IMAGE/SQL Version B.G1.10

ALLBASE/SQL Version A.G1.15

Network Requirements:

ThinLANLink/XL must be configured and turned on.

HP9000 Server Requirements:

Operating Systems Requirements

HP-UX Version 9.0 and Version 10.0

Database Requirements:

ALLBASE/SQL Version A.G1.15

Network Requirements:

ARPA Services must be configured and turned on.

Client Requirements:

Platform:

486 processor at a minimum, preferably a Pentium

8 mg memory at a minimum, preferably 16 mg

5 mgs of free disc space

Operating Systems:

Windows 3.1

Windows 3.11

Windows 95

Windows NT Version 3.51

Windows NT Version 4.0

Network Requirements

Winsock or other TCP/IP software

Software provided:

Server program for the HP3000 or HP9000 as appropriate

Client programs both 16 & 32-bit platforms.

Field types supported:

All data types supported by Allbase and Image/SQL.

SQL Commands supported:

ODBCLink/SE supports the complete ALLBASE/SQL and IMAGE/SQL syntax dynamic SQL.

Chapter 2

Preparing the Database Server

Before you install ODBCLink/SE on the PC client, perform the following steps to prepare the database server:

- ⊗ Verify the software version numbers.
- ⊗ Verify the Connection Assurance Parameters.
- ⊗ Starting and Stopping the listener.
- ⊗ Gather information.
- ⊗ Create the DBEnvironment

Verify Software Version Numbers

Note: *If you are using ALLBASE/SQL, version A.G1.15 or higher is required.*

On the HP3000 enter:

```
: SQLVER.PUB.SYS
```

On the HP9000 enter:

```
$sqlver
```

Note: *If you are using IMAGE/SQL, the version number should be B.G1.10 or higher. Type:*

```
:IMAGESQL.PUB.SYS  
>>exit
```

Note: *If both ALLBASE/SQL and IMAGE/SQL are installed, both must be either the respective version state above or newer in order to use ODBCLink/SE.*

Verify the Connection Assurance Parameters

When the connected PC client application terminates abnormally or aborts, connection assurance parameters, set under MPE/iX, determine the length of time that the server takes to time out the connection. Because the server process may be holding locks when the PC aborts, it is important that these parameters are set properly.

Your network administrator should verify that the settings for Connection Assurance Interval and Maximum Connection Assurance Retransmissions parameters are set appropriately.

The Connection Assurance Interval specifies the amount of time between each polling event. The default value is 600 seconds.

The Maximum Connection Assurance Retransmissions specifies the number of retries the server is polled after a PC client abort has been detected by the polling event. The default value is four times.

After the connection abort, the server polls the connection five times (the polling event plus four retries). The

connection times out after 4 ten-minute intervals plus the amount of time between the polling event and the first retry after the connection aborted.

To change the network assurance parameters, you must bring the network down. The sequence of commands used to set network assurance parameters are as follows:

```
:HELLO MANAGER.SYS
:NMMGR
```

Next, press the function keys in the sequence shown in the following example (these function key labels are for MPE/iX 5.0):

```
F1 Open Config
F2 NS
F2 Unguided Config
F1 Go To Netxport
F2 Go To GPROT
F2 Go To TCP
:
```

Modify values for Connection Assurance Interval (e.g. 60) and Maximum Connection Assurance Retransmissions (e.g. 2).

```
:
F6 Save Data
```

Note: *Connection assurance parameter values control polling times on all active server LAN connections (not just ODBCLink/SE connections). More frequent polling uses more server CPU time and increases LAN traffic.*

Starting and Stopping the Listener

To start the listener on the HP3000 data base server, enter:

```
STREAM ODBCJOB.ODBCSE.SYS
```

Note: *PM and NM capability is required to start the listener job, therefore it is normal for the job to be started by MANAGER.SYS.*

To start the listener on the HP9000 database server, enter:

```
/usr/bin/odbcse/odbcInse server          On a 9.x server
/opt/allbase/bin/odbcse/odbcInse server  On a 10.x server
```

Note: *You should start the listener as root user-id preferably in your system startup file.*

Note: *When the server is started it makes a call to sqlver and puts the output into a file called odbcver in the same directory that the server is running from. For this reason, upon startup, it may take 20 or more seconds for the program to return. The odbcver file is used to determine the DBMS version for a SQLGetInfo call. If the odbcver file is deleted the SQLGetInfo call for DBMS version will return "UNKNOWN"*

To determine whether or not the listener job is running:

On the HP3000 database server, enter:

```
SHOWJOB JOB=@J
```

The resulting display will be something like this:

```
JOBNUM  STATE  IPRI  JIN  JLIST  INTRODUCED  JOB NAME
#Jnnn   EXEC           10S  LP    THU 12:01P  ODBCLNSE,MANAGER.SYS
```

On the HP9000 database server, enter:

```
# ps -ef | grep odbclnse
```

The resulting display will be something like this:

```
root    7223      1  0    15:37:15  ?    0:00  /usr/bin/odbcse server
paw     7226     7223  0    15:40:56  ?    0:00  /usr/bin/odbcse server
root    7238     7230  1    15:41:29  ttys2 0:00  grep odbcse
```

The first line shows the server running. The second line shows a user with a login of "paw" is using the server and the last line shows the command you just entered.

To stop the listener:

On the HP3000 database server, enter:

```
ODBCLNSE.ODBCSE.SYS  STOP
```

Note: *If there is a connection open when the stop command is issued the process will not be stopped. If you are not able to stop the listener in the normal way, or if you want to terminate active ODBC client sessions, you may abort the listener Job with the :ABORTJOB command*

On the HP9000 database server, enter:

```
/usr/bin/odbcse/odbclnse stop          On HP-UX 9.0
/opt/allbase/bin/odbcse/odbclnse stop  On HP-UX 10.0
```

Note that in Unix, the STOP command will prevent any new connections but will not affect connections that were active at the time the command was issued.

You may also use the following command to stop the listener. It is, however, strongly suggested that you use the "odbcse stop" command to stop the listener.

```
kill -16 "pid" where "-16" is the "siguser1" and "pid" is the process ID number
```

In the example above, if the "pid" was 7223" then "kill -16 7223" would stop the server. Using a "kill -16" is less

dangerous than using “kill (pid)”. If you issue a kill command without the -16 and enter the wrong “pid” number then you may stop a server process that does not belong to you. Using “kill -16” will prevent this from happening.

Gather Information

When you install ODBCLink/SE on the PC client, you will need database server information. Now is a good time to Note it. The database administrator will have information about the database.

- ⊗ **Host or node name of the HP3000 or HP9000 database server, or its IP address (For example: HP3000 or HP9000 or 192.9.2.10)**

To obtain the host name used by the ARPA interface:

- ⊗ **On the HP3000 check the HOSTS.NET.SYS file. If the hosts file does not exist, check the domain name service file. This file contains the names of the systems that have hosts files.**
- ⊗ **On the HP9000 check the /etc/hosts file. If the hosts file does not exist, check the domain name service file, /etc/resolv.conf. This file contains the names of the systems that have hosts files.**
- ⊗ **Fully qualified name or the absolute pathname of the DBEnvironment on the server: (For example: PartsDBE.SomeGrp.SomeAcct on the HP3000 or /usr/hpsql/sampled/PartsDBE on the HP9000)**
- ⊗ **Logon string and password used to connect to the DBEnvironment:(For example: SomeUser/passwd.SomeAcct/AcctPass,SomeGrp/GrpPass on the HP3000 or User1,passwd on the HP9000)**

Create the DBEnvironment

For details, see Appendix B - Creating a DBEnvironment.

Chapter 3

Installing ODBCLink/SE on the PC

ODBCLink/SE is contained in self-extracting archive files:

- ⊗ **ODBCCL16 for the 16 bit driver and**
- ⊗ **ODBCCL32 for the 32 bit driver.**

Setting up ODBCLink/SE involves:

- ⊗ **Downloading the appropriate ODBCLink/SE file to a temporary sub-directory.**
- ⊗ **Extracting the files on your PC client.**
- ⊗ **Running the SETUP program to install the driver.**

Once the driver has been successfully installed, the sub-directories created as part of the install may be deleted.

Downloading ODBCLink/SE to the PC

In the following section, replace *Server* with the name of your HP3000 or HP9000 server and *nn* with 16 or 32 depending on which driver is being installed.

Using Reflection® software:

From your PC:

- ⊗ **Create a directory:**
 - MD ODBCSEnn
- ⊗ **Move to the new directory:**
 - CD ODBCSEnn
- ⊗ **Start Reflection and copy the appropriate ODBCLink/SE self-extracting file to your PC.**

For the HP3000 server, press Alt/Y to open the Command Window and enter

```
receive \odbcse $nn$ \odbccl $nn$ .exe from odbccl $nn$ .odbcse.sys binary
```

For the HP9000 server use any login as the Username:

For HP-UX Version 9.x

```
receive \odbcse $nn$ \odbccl $nn$ .exe from \usr\bin\odbcse\odbccl $nn$  binary
```

For HP-UX Version 10.x:

receive\odbcse\odbcclnn.exe from \opt\allbase\bin\odbcse\odbcclnn binary

Using ARPA Services (FTP):

From your PC:

- ⊗ **Create a directory:**
 - MD ODBCSEnn
- ⊗ **Move to the new directory:**
 - CD ODBCSEnn
- ⊗ **Copy the appropriate ODBCLink/SE self-extracting file to your PC**

For the HP3000 server, use MANAGER.SYS,ODBCSE as the Username

```
C:\ODBCSEnn>
ftp Server
binary
get odbclnn odbclnn.exe
quit
```

For the HP9000 server use any login as the Username:

```
C:\ODBCSEnn>
ftp Server
binary
get /usr/bin/odbcse/odbcclnn odbclnn.exe (for HP-UX V9.x)
```

or

```
get /opt/allbase/bin/odbcse/odbcclnn odbclnn.exe (for HP-UX V10.x)
```

Using NS Services (DSCOPY) on the HP3000:

From your PC:

- ⊗ **Create a directory:**
 - MD ODBCSEnn
- ⊗ **Move to the new directory:**
 - CD ODBCSEnn
- ⊗ **Copy the appropriate ODBCLink/SE self-extracting file to your PC**
dscopy -r -F -B -L256 Server#manager.sys,odbcse# odbclnn odbclnn.exe

Extracting the ODBCLink/SE file

From your PC, type the following to extract the ODBCLink/SE files from the self-extracting file (replace *nn* with 16 or 32 depending on which driver is being installed):

```
C:\CD \ODBCSEnn
```

```
C:\ODBCSEnn> ODBCCLnn.EXE
```

Note: *If you are installing ODBCLink/SE on a number of PC's, after extracting the files you might wish to create installation disks by copying the extracted files onto a diskette.*

Where *nn* is 16 for the 16 bit driver and 32 for the 32 bit driver.

Running the SETUP program

A problem associated with Microsoft ODBC 2.10 setup causes the setup on a WindowsNT 4.0 system to install as if it was a Windows95 system. We have therefore created two information files to be used with the 32-bit setup program. A batch file copies the appropriate file to `odbc.inf` **depending on the parameters given**.

To allow the appropriate file to be installed do the following from the setup directory (eg `\ODBCSEnn`). It is preferable that this procedure be performed at the DOS prompt so that the window will not disappear before you have a chance to read it. The process can however be performed from the Program Manager.

On a WindowsNT system run "infsetup wnt"

On a Windows95 systems run "infsetup w95"

The batch file will read the parameter entered and copy the appropriate file to `odbc.inf`. Entering `infsetup` with no parameter specified will simply describe the usage but **will not copy the file**.

Note: *We have left the original `odbc.inf` file on the disk so if `infsetup` is not run, the install will still work in most cases. If the wrong options is entered or you are installing on both a WindowsNT and Windows95 platform, simply re-run the `infsetup.bat` file with the proper option.*

Once the above has been accomplished proceed with the setup program as follows:

From the Program Manager:

Click on File then Run

```
enter \ODBCSEnn\SETUP.EXE
```

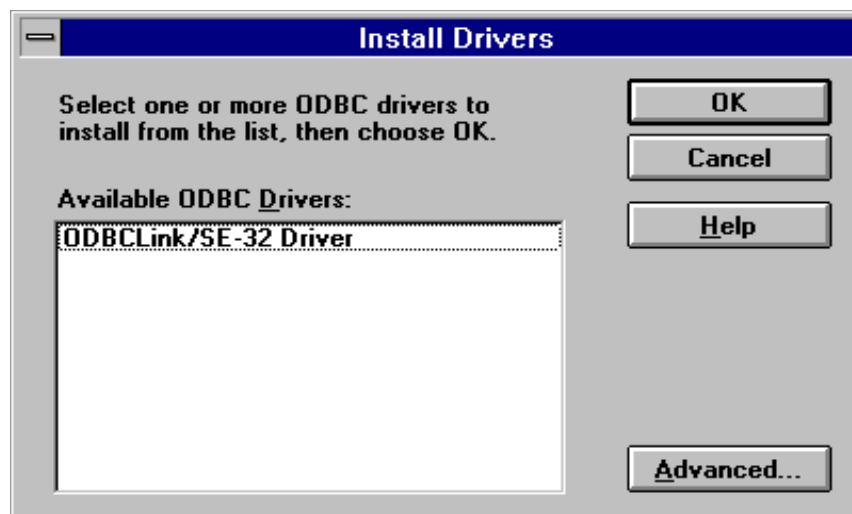
From DOS:

```
enter \ODBCSEnn\SETUP.EXE
```

The Driver Setup Program will display this window.



When you click on Continue, the Install Drivers window appears.

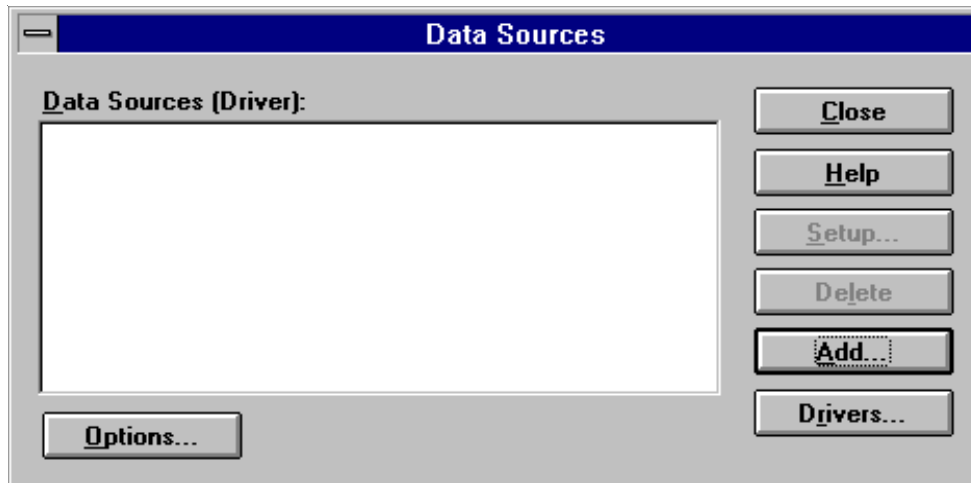


Note: You **MUST** click on a driver name, even if only one driver name appears in the list, then click OK. The driver names for ODBCLink/SE are:

"ODBCLink/SE-16 Driver" for 16 bit applications

"ODBCLink/SE-32 Driver" for 32 bit applications.

The driver is now installed. If this is the first driver you have installed, the Data Source name list will be empty.



You may now click on:

- ⊗ **Close to end the installation.**
- ⊗ **Add to add a new Data Source**
- ⊗ **Setup to modify existing Data Sources**
- ⊗ **Delete to delete the highlighted data source (16 bit only)**
- ⊗ **Drivers to do driver maintenance (16 bit only)**
- ⊗ **Options to establish ODBC tracing rules**



The driver has been successfully installed. You may now create data sources for use in applications that use this driver. *See Section 4* for a complete description of creating data sources.

The sub-directories ODBCSE16 and ODBCSE32 are no longer required and may be deleted, if you wish. If you need to re-install a driver you can always download the files from the host.

Notes on 16 or 32-bit Access

On a 32-bit operating system (Win95 or WinNT) you can run both 16 and 32-bit versions of the ODBCLink/SE driver. The server software runs the same for both.

You should always use the 32-bit driver, if you are using a 32-bit operating system. In some cases however you may not have a choice, since a 16-bit application in Win95 or WinNT **cannot use a 32-bit driver**. If you want ODBC access from a 16-bit application on Win95 or WinNT, you must install the 16-bit version of ODBCLink/SE.

16 and 32-bit data-sources are maintained separately and can coexist. 32-bit data-source information is kept in the registry, and maintained by the 32-bit ODBC control panel application ODBCAD32.EXE. 16-bit data-source information is kept in the ODBC.INI and ODBCINST.INI files in your windows root directory (not the windows\system directory) and is maintained by the 16-bit control panel application ODBCADM.EXE. The 16-bit ODBC control panel application will show 32-bit data-sources, but the 32-bit ODBC control panel will **not** show 16-bit data-sources.

To distinguish between 16 and 32-bit data-sources, look at the driver name in the data-source window in the ODBC control panel. You will either see ODBCLink/SE-16 or ODBCLink/SE-32

You do not normally make entries manually into either the registry or the ODBC.INI file. However information has been provide in the manual to help you do this.

Chapter 4

Configuring Data Sources

Before you access data defined to ALLBASE/SQL or IMAGE/SQL you must configure one or more *data sources* for each PC you wish to access your server.

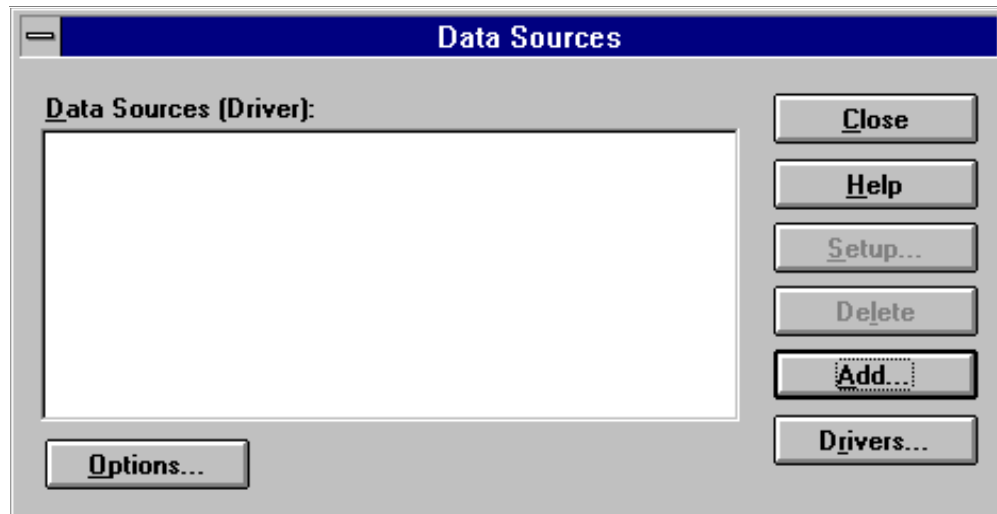
A data source includes the name you wish to give for a database environment and the information necessary to find it.

You configure data sources with the "ODBC" application in the Windows Control Panel.

- ⊗ **In Windows 3.11, double click on Main, Control Panel and ODBC**
- ⊗ **In Windows95 and WindowsNT 4.0 , click on Start, Settings, Control Panel and double click on ODBC.**
- ⊗ **In Windows NT V3.51, double click on Main, Control Panel and ODBC for 32-bit data sources.**
- ⊗ **For 16-bit data sources use the ODBC Administrator in the ODBC Program Group.**

Data sources created in HP ALLBASE/SQL PCAPI can be migrated to ODBCLink/SE using the Data Set Migration Tool described later in this section.

Data Source Configuration

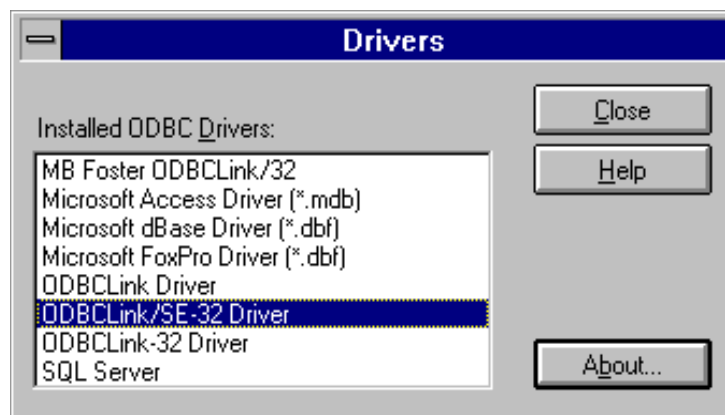


You may now click on:

- ⊗ Close to return to the Control Panel
- ⊗ Add to add a new Data Source
- ⊗ Setup to modify existing Data Sources
- ⊗ Delete to delete the highlighted data source (16 bit only)
- ⊗ Drivers to do driver maintenance (16 bit only)
- ⊗ Options to establish ODBC tracing rules

Choosing an ODBC Driver

You MUST click on a driver name even if only one name appears in the list. Click **Close**.



ODBCLink/SE Setup

On this screen you give the database environment a name and indicate where it is located. The Data Source Name may be any combination of alphanumeric characters including blanks.

The maximum length is 30 characters. If 16 and 32 bit applications are being used on the same PC, it is useful to know which data source is for which driver; it may help to include '16' or '32' in the name. You may choose your own naming convention.

The Description is optional and may contain any string of characters you wish. It may be longer than the display window.

The Database Name is the name of the ALLBASE/SQL Database Environment you want to access with this data source. The name will usually include the location of the DBE as shown in the sample screen. If the logon provided in the next screen does not include the same logon group, the location **MUST** be provided. A sample HP/UX name might be:

`/usr/users/data/PartsDBE`

The Server Name or IP Address is the name or address of the computer where your database environment resides. If you don't know this name, contact your network administrator.

The Server Type radio button indicates the type of computer on which the DBE is located. Click on:

MPE/iX if it is an HP3000
HP-UX if it is an HP9000

Setup for MPE/iX

Session ID (optional)	User Name (mandatory)	Acct Name (mandatory)	Group Name (optional)
HELLO	johnm	odblink	
	User Password (?' will prompt)	Acct Password (?' will prompt)	Group Password (?' will prompt)
	/	/	/
		xxxxxx	

Buttons: Continue, Cancel, Back

The SessionID, User Name, Acct Name, Group Name, User Password, Acct Password and Group Password entries correspond to those used if you were manually connecting to the HP3000 or HP9000. If you don't know this information, contact your System Manager.

If you want the client application to prompt for the passwords at run-time, enter a "?" in the password field. If you put a question mark in any password field, you will get a dialogue box and be prompted to enter the password **every time you run your application and request a connection**. For an attached table in MS-Access, ODBCLink/SE will **not** store the password anywhere within the application, so you will have to enter the password the first time you open the table. However, many applications, including Access, connect more than once to the same data-source, and if you have already entered the passwords and created a new connection, you will not normally have to enter it again for additional connections to the same data-source.

Note: *As you might expect, "mandatory" means you must enter a value. However, "optional" doesn't necessarily mean you don't have to enter a value; it means there may not be a value required. For example, on the MPE/iX logon, "Session ID" is any 8 characters you choose, starting with an alphabetic character with no embedded blanks. You may choose not to enter a "Session ID". On the other hand, "Group Name" will default you to your "home group" so may be left blank; if your "User Name" does not have a "home group" this field is required or the connection will fail.*

Setup for HP-UX



ODBCLink/SE Setup (HP-UX)

Login:
(mandatory)

johnm

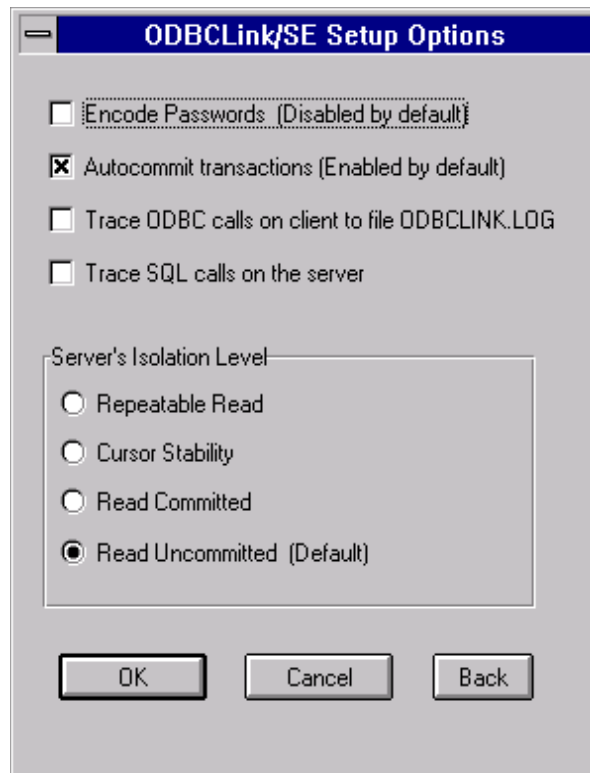
Password: (optional)
? Will Prompt

Continue Cancel Back

Note: *On HP-UX, the Login name and password are case sensitive.*

If you want the client application to prompt for the passwords at run-time, enter a “?” in the password field. If you put a question mark in any password field, you will get a dialogue box and be prompted to enter the password **every time you run your application and request a connection**. For an attached table in MS-Access, ODBCLink/SE will **not** store the password anywhere within the application, so you will have to enter the password the first time you open the table. However, many applications, including Access, connect more than once to the same data-source, and if you have already entered the passwords and created a new connection, you will not normally have to enter it again for additional connections to the same data-source.

Setup Options



Encoded passwords: Check this box to encrypt the connection passwords in the ODBC.INI file or system registry. For security reasons once the passwords are encoded if you decide to un-click the box you must DELETE the current passwords first.

AutoCommit transactions: Check this box to have UPDATE and INSERT transactions committed as soon as they are executed. If this box is not checked, transactions must be explicitly committed by the client application.

Note: *It is recommended that AutoCommit be left enabled*

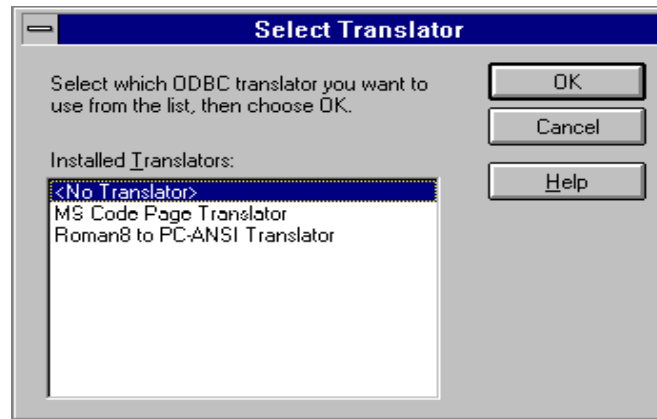
Trace ODBC calls on client: Check this box to have client calls to ODBC logged in the file ODBCLINK.LOG on the PC.

Trace ODBC calls on server: check this box to have server calls to ODBC logged in the file ODBCLOG.ODBCSE.SYS on the HP3000 or /tmp/odbclog on the HP-UX.

Isolation levels allow you to control the degree of concurrency by regulating the extent to which operations performed by one user in a multi-user environment can be affected by operations performed by other users.

Note: *The recommended isolation level is "RU" for Read Uncommitted. This minimizes the number of locks that are held on your database.*

Select Translator



ODBC Translator

Select “Roman8 to PC-ANSI Translator” if you want extended characters on your HP3000 (such as foreign accented characters) to appear correctly on your PC. The translation goes both ways. For example; extended characters are entered on the PC and sent to the host in an SQL update or Insert operation will be translated.

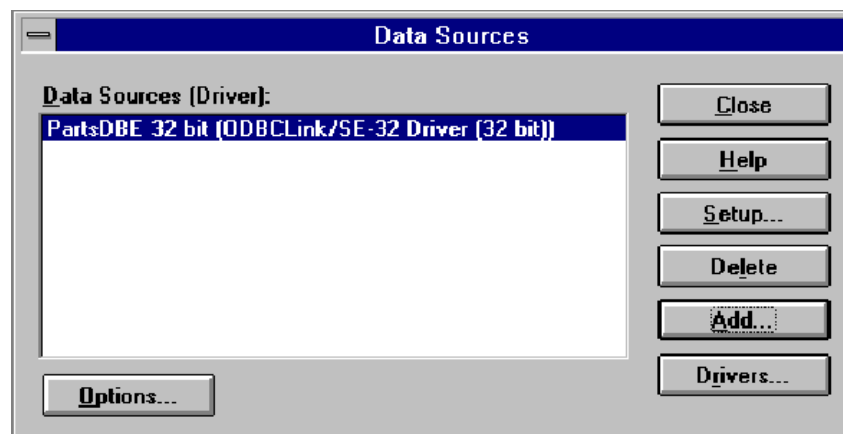
Note: *Roman9 to PC-ANSI Translator will recognize the symbol for the European Currency Symbol, known as the EURO.*

Successful Completion

The new data source is now ready for use in your client application.

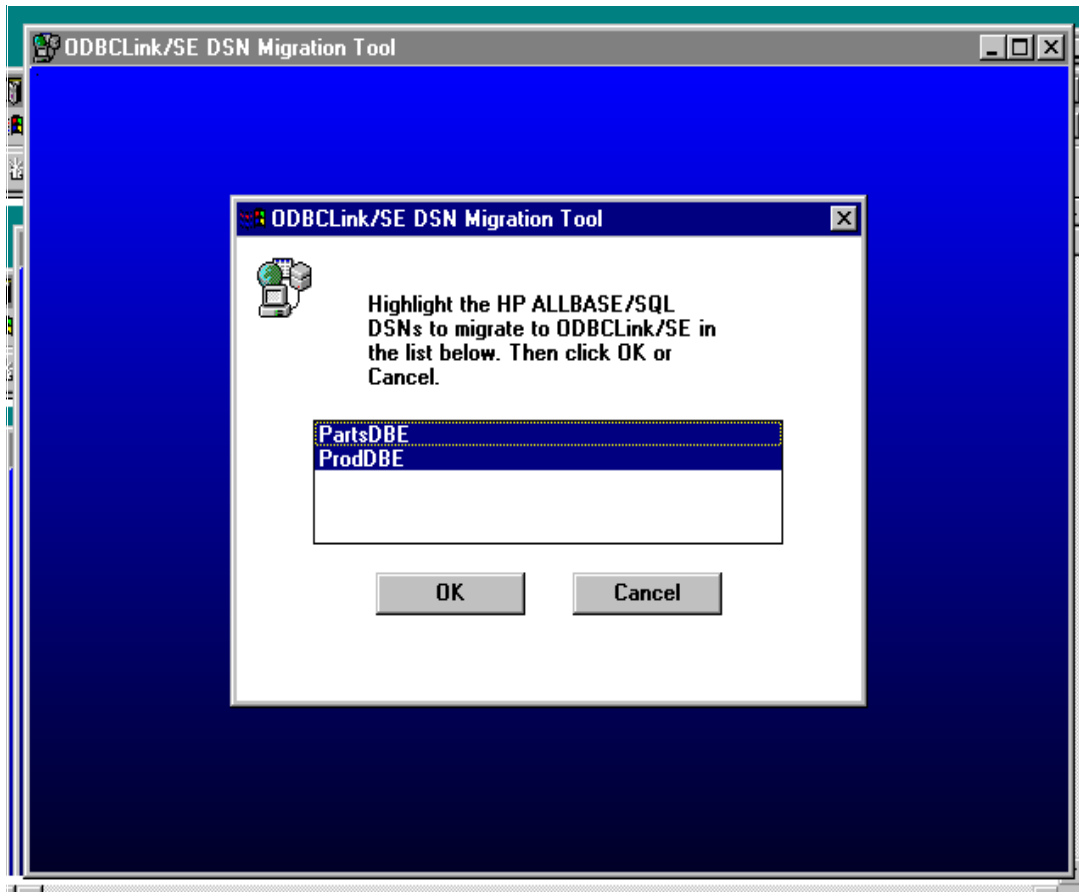
It may appear that the "32 bit" included in the data source description is redundant, as it is also included in the driver name. However in a mixed environment; some clients show only the data source name. Adding “32bit” make it easier to distinguish between 32-bit and 16-bit drivers.

Some applications may allow you to go directly to the "Data Sources" menu without going to Control Panel to click on the ODBC Administrator. Check you client application documentation for details.



Migrating HP PCAPI Data Sources

Data sources created in the HP ALLBASE/SQL PCAPI can be migrated to ODBCLink/SE by running the data set migration tool DXNMIG16.EXE either from the RUN window or in an MS-DOS prompt window.



Click on all the data source names you wish to migrate and then click OK. Data sources will be migrated to ODBCLink/SE data sources with the same name. Copies of the PCAPI data sources will be made with the same name followed by "-orig".

When the copies of the originals are no longer required, they may be deleted with the 16-bit ODBC Administrator program.

Chapter 5

Application Development

If you are developing applications that use the ODBC interface, you may want to order the Microsoft Developer Network (MSDN) Professional (Level 2) from Microsoft. The MSDN contains the ODBC Software Developers Kit (SDK), which includes the ODBC API Reference, sample ODBC applications, and other technical information to assist you in developing your applications. There are also many good books written on the use of ODBC.

Using Stored Procedures

ODBCLink/SE implements all types of ALLBASE/SQL stored procedures, including those which return multiple-format results sets. You can list the stored procedures available to your ODBC session by calling `SQLProcedures` and you can get parameter information for stored procedures by calling `SQLProcedureColumns`.

Stored procedures return data either through bound parameters or through one or more results sets. To get a return value that is a procedure parameter, prepare an SQL statement (with `SQLPrepare`) of the form:

```
{?=call owner.procedure (?,?..)}
```

Then bind the parameters with `SQLBindParameter`, specifying an output parameter where appropriate (`SQLProcedureColumn` will tell you which are input, which are output, and which are input/output parameters). Then call `SQLExecute`. The return value(s) will be copied to the storage locations that you specify. Note that all stored procedures return an integer return status called `RETURN_STATUS`.

If your stored procedure also returns one or more result sets, you call `SQLNumResultCols`, `SQLDescribeCol`, `SQLFetch`, `SQLGetData` in the normal way you would for any Select statement. If the procedure returns more than 1 result set, you call `SQLMoreResults` when you get to the end of the first result set, and, if successful, you can then retrieve another result set. It is permissible to call `SQLMoreResults()` at any time to switch to a different result set.

Performance Considerations

Many users find ODBC access to their server very fast. You should however be aware of the following:

- ⊗ **If you are reading and sorting a large table, the host may not return control to the client until the entire data-base has been read and sorted. If this occurs, even restarting your PC will have no effect and you will have to manually abort the server, or terminate the connection with the ISQL “Terminate User” command.**

- ⊗ **When you are opening a large table in MS-Access, the program typically displays a screenful of data and then seems to wait for user input. In many cases, however, it will actually be downloading data to the client in the background. You should think about this when giving your users ODBC access to large tables on the server. You may give or withhold access to certain tables with the ALLBASE/SQL GRANT and REVOKE commands. There is no way currently of limiting the number of rows or the time the server is allowed to execute a request.**

- ⊗ **As a rule of thumb, ODBCLink/SE will not complete any request faster than ISQL will, and could be considerably slower due to network overhead. If you are writing your own SQL, you may want to verify in ISQL how ALLBASE/SQL optimizes your queries by reading the SYSTEM.PLAN pseudo-table.**

- ⊗ **If you are writing your own SQL to do multi-row inserts or updates, you can speed up your application by using dynamic substitution parameters. See the ODBC SDK manual (from Microsoft Press) for details.**

Turning AutoCommit On/Off

AutoCommit “on” is the default value for all ODBC drivers. Turning AutoCommit on/off is normally accomplished by the application calling `SQLSetConnectOption`. The AutoCommit on/off setting in ODBC Administrator is there for historical reasons only and should not be used. Setting AutoCommit “off:” in the ODBC Administrator is not a good idea as the application can get confused. For instance, it can incorrectly assume that AutoCommit is on, and then emit SQL that never gets committed.

Supported Functions

The ODBC functions are listed here for application software developers. These functions are supported by the ODBCLink/SE driver at the time of publication of this manual. Refer to the README.HLP file included with ODBCLink/SE for further developments.

You can use the ODBC SQLGetFunctions call for a list of the supported ODBCLink/SE driver functions. Descriptions and conformance designations are listed in ODBC API Reference included in the Microsoft Software Development Kit (SDK).

Functions that are not completely implemented are noted by an R in parenthesis and are discussed below this list.

SQLAllocConnect
SQLAllocEnv
SQLAllocStmt
SQLBindCol
SQLFindParam
SQLCancel
SQLColAttributes (R1)
SQLColumns
SQLColumnPrivileges
SQLConnect
SQLDataSources
SQLDescribeCol
SQLDisconnect
SQLDriverConnect
SQLError
SQLExecute
SQLExecDirect
SQLFetch
SQLForeignKeys
SQLFreeConnect
SQLFreeEnv
SQLFreeStmt
SQLGetConnectOption
SQLGetCursorName
SQLGetFunctions
SQLGetInfo (R3)
SQLGetTypeInfo
SQLNumResultCols
SQLPrepare
SQLPrimaryKeys
SQLProcedureColumns
SQLProcedures
SQLRowCount
SQLSetConnectOption (R2)
SQLSetCursorName
SQLSetParam
SQLSetStmtOption (R4)
SQLSpecialColumns
SQLStatistics
SQLTablePrivileges
SQLTables
SQLTransact

- ⊗ **R1:** The following SQLColAttributes options are implemented; the others will receive a return value of 'Not Implemented'.

SQL_COLUMN_COUNT
 SQL_COLUMN_NAME
 SQL_COLUMN_TYPE
 SQL_COLUMN_LENGTH
 SQL_COLUMN_PRECISION
 SQL_COLUMN_SCALE
 SQL_COLUMN_DISPLAY_SIZE
 SQL_COLUMN_NULLABLE

- ⊗ **R2:** The following SQLSetConnectOptions are supported. All other will return a “Driver not Capable (SQLState S1C00)”.

SQL_AUTOCOMMIT	1=ON (default) 0=OFF
SQL_MAX_ROWS	0=All Rows (default)
SQL_OPT_TRACE	supported by Driver Manager
SQL_OPT_TRACEFILE	supported by Driver Manager
SQL_QUERY_TIMEOUT	0=No Timeout (default) Use ALLBASE Timeout
SQL_TRANSLATE_DLL	supported
SQL_TRANSLATE_OPTION	supported
SQL_TXN_ISOLATION	supported: Cursor Stability isolation level (CS) is not supported by SQLSetConnectOption but you may set it from the ODBC Administrator
SQL_TXN_READ_UNCOMMITTED	(Read Uncommitted (RU))
SQL_TXN_READ_COMMITTED	(Read Committed (RC))
SQL_TXN_REPEATABLE_READ	(Repeatable Read (RR))
SQL_TXN_SERIALIZABLE	(Note: Sets to Repeatable Read (RR))
SQL_TXN_VERSIONING	Not Implemented - DO NOT USE

Note: For information on isolation levels, refer to the “Isolation Levels” section in this chapter.

- ⊗ **R3:** All SQLGetInfo Options are supported.
- ⊗ **R4:** SQLSetStmtOptions are not supported. The driver will return either a “Driver not Capable (SQLState S1C00)” or “Option value changed (SQLState 01S02)”.

Isolation Levels

Isolation levels allow you to control the degree of concurrency by regulating the extent to which operations performed by one user in a multi-user environment can be affected by operations performed by another user.

ALLBASE/SQL allows four different isolation levels:

- ⊗ **Cursor Stability (CS)**
- ⊗ **Repeatable Read (RR)**
- ⊗ **Read Committed (RC)**
- ⊗ **Read Uncommitted (RU)**

ODBCLink/SE uses the isolation level specified in the ODBC setup screen. The application can later change this by calling `SQLSetConnectOptions` with `SQL_TXN_ISOLATION` option.

The recommended isolation level is “RU” for Read Uncommitted. This minimizes the number of locks that are held on your database.

Note: *Refer to the ALLBASE/SQL Reference Manual for further information.*

Supported Data Types

The following is a table of correspondence between HP data types and ODBC data types.

ALLBASE/SQL or Image or IMAGE/SQL Data Type	ODBC Data Type	Comments
CHAR, max length <255	SQL_CHAR	
CHAR, max length >=255	SQL_LONGVARCHAR	
VARCHAR, max length <255	SQL_VARCHAR	
VARCHAR, max length >=255	SQL_LONGVARCHAR	
BINARY, length <256	SQL_BINARY	
BINARY, length >=256	SQL_LONGVARBINARY	
VARBINARY <256	SQL_VARBINARY	
VARBINARY >=256	SQL- LONGVARBINARY	
LONG BINARY	SQL_LONGVARBINARY	
LONG VARBINARY	SQL_LONGVARBINARY	
INTEGER (32-bit)	SQL_INTEGER	
SMALLINT (16-bit)	SQL_SMALLINT	
DECIMAL (Internal representation is packed decimal)	SQL_DECIMAL	Character representation with leading sign and decimal point
Image Zoned (Z)	SQL_DECIMAL	Converted by Image/SQL
Image Packed (P)	SQL_DECIMAL	Converted by Image/SQL
Image I3 (48-bit integer)	SQL_DECIMAL	Converted by Image/SQL
Image I4 (64-bit integer)	SQL_DECIMAL	Converted by Image/SQL
FLOAT(24) or REAL or Image E2	SQL_REAL	
FLOAT(53) or DOUBLE PRECISION or Image E4	SQL_DOUBLE	
Image R2 (non-IEEE float)	SQL_REAL	Converted by Image/SQL

Image R4 (non-IEEE float)	SQL_DOUBLE	Converted by Image/SQL
DATE	SQL_DATE	6-byte with year, month, day in 2-byte binary fields
TIME	SQL_TIME	6-byte with hour, minute, second in 2-byte binary fields
DATETIME	SQL_TIMESTAMP	16-byte made up of year(2), month(2), day (2), hour (2), minute (2), second (2), fraction (4)
INTERVAL	SQL_CHAR	Format is “ddddddd hh:mm:ss.fff”

Note: *When creating a table using CHAR and BINARY data types that are greater than 255 characters the resulting data types used will be LONGVARCHAR and LONGVARBINARY, which may not be the data type expected by the application.*

Note on Using BLOBs

It is strongly recommended that when creating an ALLBASE table for storage of BLOBs that you use a LONG VARBINARY column. ALLBASE will allocate storage space according to the actual size of the BLOB. Eg. Creating a table with a column defined as LONG VARBINARY (200000000) and writing a 10K BLOB to it will result in only 10K of space being used (not the maximum of 2 GB as specified when the column was created).

Although you can store a BLOB using a LONG BINARY column, this is not recommended, because ALLBASE will allocate storage space according to the specified column size for each BLOB regardless of the size of the actual data. Eg. Creating a table with a column defined as LONG BINARY(1000000) and writing a 10K BLOB will result in 1000000 bytes of space being used for each BLOB. You would quickly run out of space in your database.

Chapter 6

Using ODBCLink/SE With ODBC Applications

To access your host through ODBCLink\SE, you will first have to set up one or more data sources via the ODBC Control Panel. You also have to configure host security using the HOSTCONF program.

Before you can connect from an ODBC client you must specify a User-ID and Password. This information is set up on the host. You must also enter this information locally on the PC. The host uses the User-ID to determine which tables the client can access. You can either enter the information locally in the data link setup of the application you are using, or you may enter it as part of the data-source configuration. Not all clients will let you enter a User-ID and password, in particular MS-Access, MS-Query and Lotus-123. For these clients you must enter the User-ID and password in your data-source configuration. Alternatively, you may let your client application connect with a null (or blank) User-ID and set-up a User-ID on the host named "DEFAULT".

Updating Data

Most ODBC applications require a unique primary key, existing on a table or data set, to be able to update the table. ODBCLink will attempt to locate a primary key using the algorithm described in Appendix A. For TurboIMAGE masters or ALLBASE/SQL tables created with a unique primary key this is not a problem, but in many cases you will have to enter the primary key name, if one exists, in the user-tables screen in HOSTCONF. TurboIMAGE detail data sets normally cannot be updated unless you manually enter the primary key. You will also have to disable the "SnapShot Only" check box in HOSTCONF.

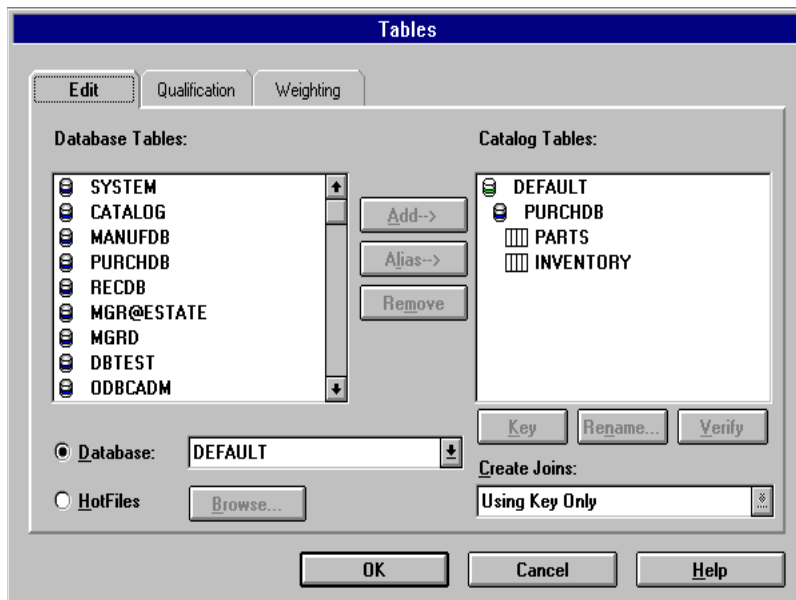
Using Cognos Impromptu

Select Catalog, Database, Add. Enter *HP3000-1* as the logical database name, *ODBC Gateway* as the Gateway, and the name of your data source, e.g. HP3000. Enter your User ID and password, and click OK.

The screenshot shows the "Database Definition" dialog box. It has a blue title bar with the text "Database Definition". The dialog is divided into several sections:

- Logical Database Name:** A text box containing "DEFAULT".
- Database Type/Gateway:** A dropdown menu showing "ODBC".
- Database Details:** A section containing:
 - ODBC Data Source:** A dropdown menu showing "hp-3000".
 - ODBC Connect String:** A text box containing "uid=default;pwd=pass".
- User Prompts:** Two checkboxes: User ID and Password.
- Timeouts:** Two spin boxes: "Connect" set to "0" and "Reply" set to "0".
- Use Asynchronous Open:** A checkbox that is currently unchecked.

On the right side of the dialog, there is a vertical stack of buttons: "OK", "Cancel", "Help", "Test", "Admin...", and "Options...".



Select Catalog, New. Enter a catalog name and your logical database name (HP3000-1). Impromptu should then establish the connection and put you into the Edit Tables Screen.

The left side of the Edit Tables screen is a list of table owners. In ALLBASE/SQL, a table owner is the owner name assigned by the database administrator (default is *UserAccount*). For a TurboIMAGE dataset, the owner name is the database name. To get a list of all the tables for a specific owner name, double click it.

You may now select the tables you want included in your catalog. After you click OK, Impromptu will load the table definition into the catalog.

If you want to add tables from another database to your catalog, select Catalog, Edit Tables, and follow the same procedure again.

You may now create a report through the File, New command. You will be prompted for the fields to include in your report, and Impromptu will generate the report on your screen.

Note: *You cannot create summary reports on TurboIMAGE tables with this version of the driver. The GROUP BY command is not implemented. You can, however, group rows and compute subtotals..*

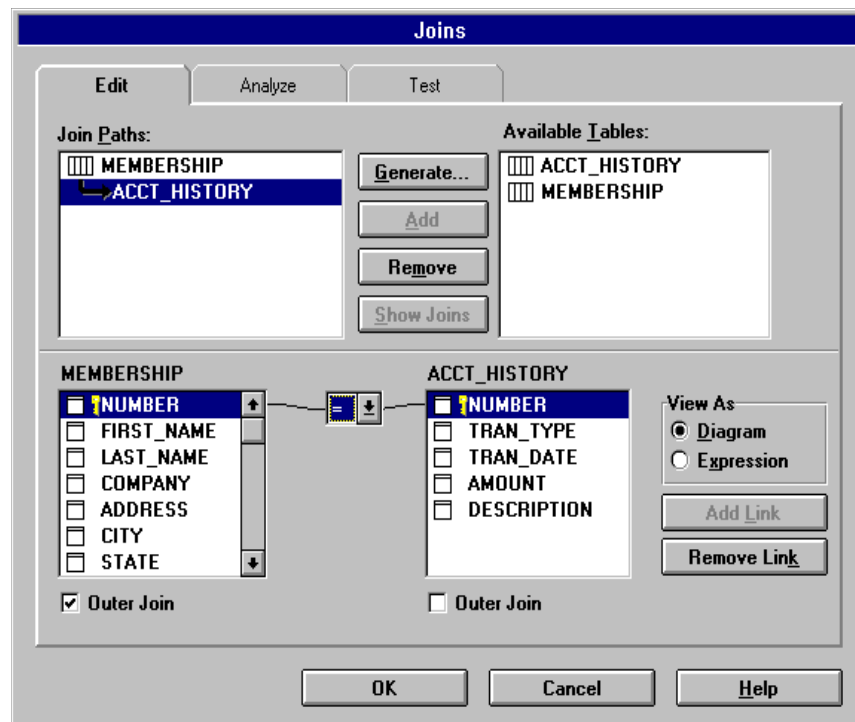
Special Notes for Users of Impromptu

Viewing the SQL generated by Impromptu

Enter a new report and select Report, Query, Profile, View SQL. You can also write your own SQL and override that generated by Impromptu. Search for help on topic "Writing your own SQL".

Joins

ODBCLink supports two kinds of joins: inner joins, sometimes called natural joins, and Left Outer joins. An inner join will return no records if any of the secondary records are not found; a left outer joins will still return the primary record and give nulls (or zeros) on the secondary table. To add or modify a join definition, select Catalog, Joins. The following figure illustrates this process by creating a Left Outer join from table MEMBERSHIP to table ACCT-HISTORY.



Note: To do left outer joins on the host (as opposed to Impromptu downloading both tables and doing the joins locally) you must modify file COGDMOD.INI in directory \COGNOS\COGAPPS as follows:

```
[Exception tables]
Joined=T
```

Impromptu has been known to generate incorrect SQL when joining three tables or more using a left outer join. To correct, modify the SQL that is generated and re-run the report.

Changing the display format

Impromptu has a limited number of display formats available. To change the display format, highlight the column and select Format Data. You cannot specify edit masks, however you have a choice of numeric display formats. To add dashes or other constants to your output, you must build an expression with SUBSTRING functions. To add a decimal point to a number that is not defined as a decimal, you must create a calculated field by dividing the number by 100.

Creating Joins Using Calculated Columns

It is possible in Impromptu to create a join using a column that is calculated from two or more fields in one or more tables. To do this, select Catalog, Edit joins; select your primary table and the table you want to join in the upper left part of the screen; click "View as Expression"; and then enter an equation such as:

$$\text{Table3.Column3} = \text{Table1.column1} + \text{Table2.Column2} \dots$$

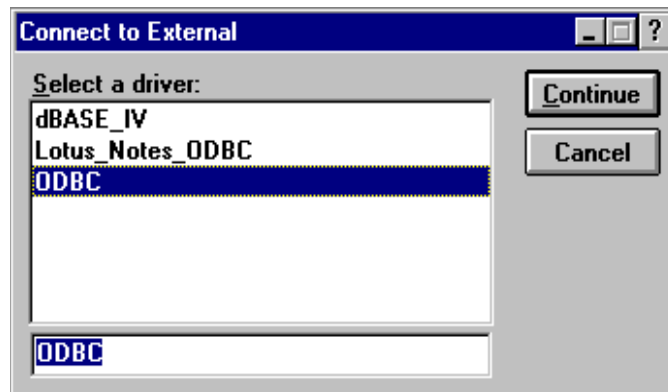
The left hand side of this expression is the table you want to join to, and the right hand side is the expression to generate the key. An expression can consist of concatenated fields, SUBSTRING statements, constants, or all of these.

Using Lotus 123 Release 5.0

To enable Lotus123 to use ODBC, you will have to add the following line to the file \LOTUSAPP\DATALENS\LOTUS.BCF. (Note: The following information is summarized from the file \123R5W\readme.txt):

```
DN="ODBC" DL="DLODBC" DD="All ODBC Sources";
```

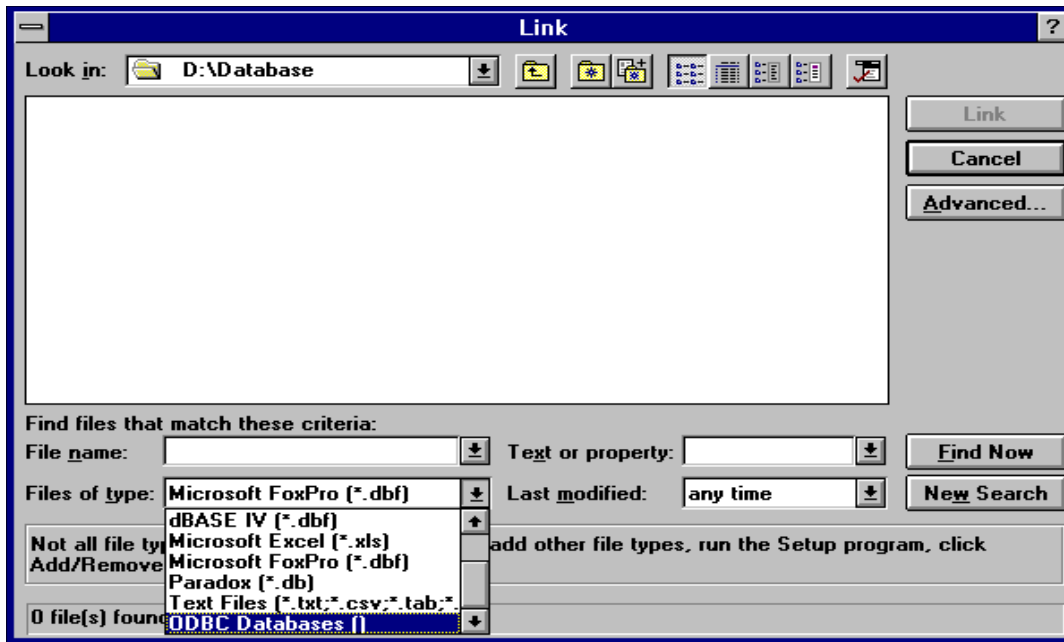
Once you have done this, you may access ODBC data sources.



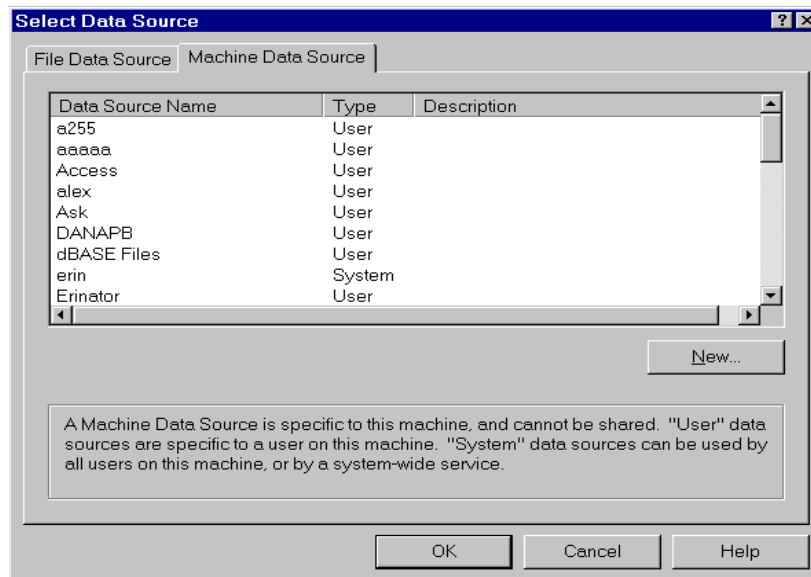
- ⊗ Select Tools, Database, New Query, External, ODBC.
- ⊗ Select a data-source from the list box.
- ⊗ Select a table to query.
- ⊗ You can then choose fields to include in the query, set criteria, enter joins, etc.

Using Microsoft Access 7.0

To access data through MSAccess 7.0, open a database and click on File, Get External Data, Link files and select ODBC Databases from the "files of type:" pull down list.

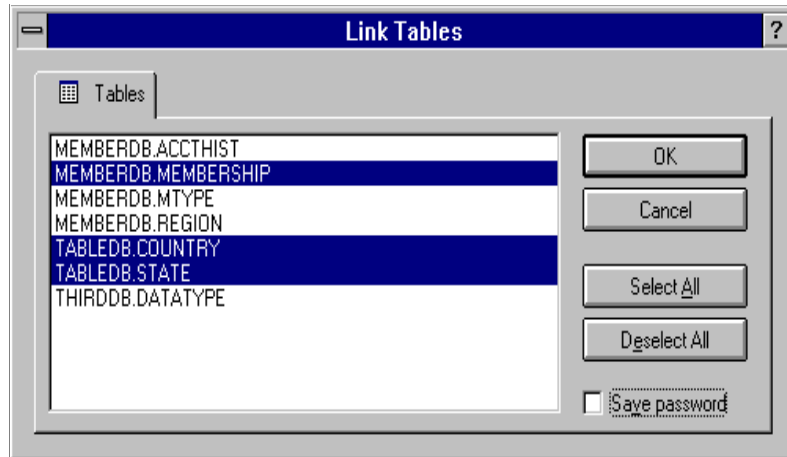


The Data Sources Window is displayed.



If the data source you require is not in the list, click on New and create a new data source.

Note: *You must click on the name of the data source you wish to use even if there is only one.*



You may highlight one or more tables and then click "OK". This will create an "attached table" which you may later open and modify. An attached table is treated like a local table in every way (you may read it, update it, delete it, import it, or join it to other local or remote tables.)

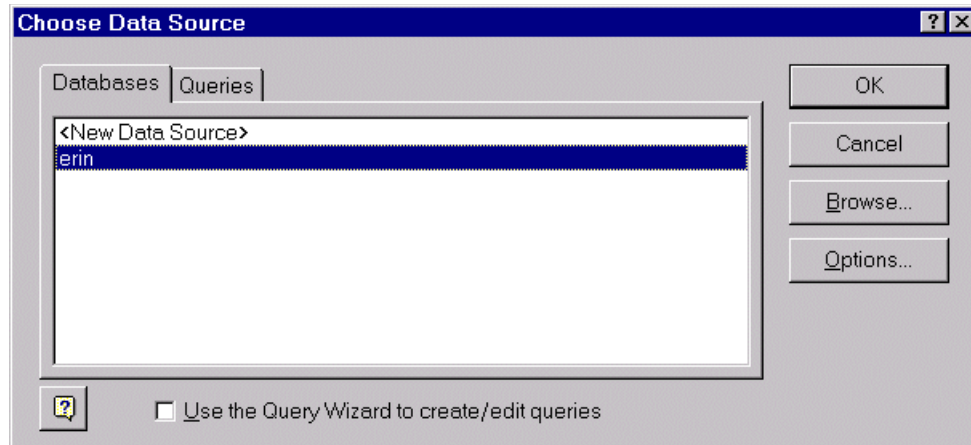
Note: *Ensure you have checked the SAVE PASSWORD box. This stores the User-ID returned by the driver in the Table Properties field and allows you to connect afterwards with the same User-ID. You should get a list of tables on the HP3000. When you select one of them, it will load the table structure (which you may examine by opening the table in Design View) and store the table in the database as an attached table. Once the table is attached, it is treated like a local table in every way, except that you may not modify the table structure. You may open the table in taew, run reports or queries against it, or link it to other local or remote tables.*

Notes for Users of MS-Access

- ⊗ For a table to be updatable in Access, it must have a unique key and Access must know about it. If the table is defined in ALLBASE/SQL with a unique key, this will be reported by ODBCLink (in the SQLStatistics call that Access makes) and the table will be updatable. If Access cannot locate a primary key, it will prompt for the column or combination of columns that make up a unique key. You can ignore this prompt, and the table will not be updatable from Access; or you can specify one or more columns that make a unique key. If this combination of columns is not unique (i.e. if there are duplicate records with the same key combination) Access will not work properly. When you enter a unique key combination, ensure that it really is unique within the table. To see the column or columns that Access uses as the unique key, look at the table in Design view. The column will have “key” symbol in the left most column.
- ⊗ Access uses a different algorithm to access a table depending on whether a unique key exists or not. If it finds a unique key, it downloads the key values and then issues an SQL statement of the form “select...From...Where key=? Or key=?...”. It then displays a screenful of data and stops. (Note, however, that it may continue to download the keys in the background so long as the table is open in “table view”.) If a unique key is not defined, Access just does a regular Select statement to read the table. It will display a screenful of data but will continue to read the table in the background. You may view the SQL that Access (and any ODBC application) generates by turning on logging in the ODBC Setup Screen .
- ⊗ To update a record in Access, you highlight the column you want to change, make the change, and then click on any other record. Access uses an algorithm called “Optimistic concurrency control” to verify that a record has not been changed by another user. This is a less secure method than the SELECT FOR UPDATE used by other applications, however it minimizes the number of locks that are held on the database. Optimistic concurrency control works by Access generating and Update statement of the form “Update Table Set column1=?, column2=?,...WHERE column1=? AND column2=? AND column3=? And”. This ensures that the update will fail if the record has been changed by another user since the time it was last read in by Access.
 - The default type of join used by Access is the Inner Join. When creating a Query with a join between two tables, you can change this to a Left Outer Join by double-clicking on the link between the tables. You may also use the Edit Joins screen to change the default join between specific tables, so that you don’t have to edit the join every time you create a new Query.
- ⊗ You cannot join a local table to a remote table efficiently in Access unless there is a one-to-one correspondence between records in the local table and records in the remote table; in other words, you must have all the columns in the local table that are necessary to form a unique key on the remote table. If this is not the case, Access will attempt to download the remote table to the local machine.

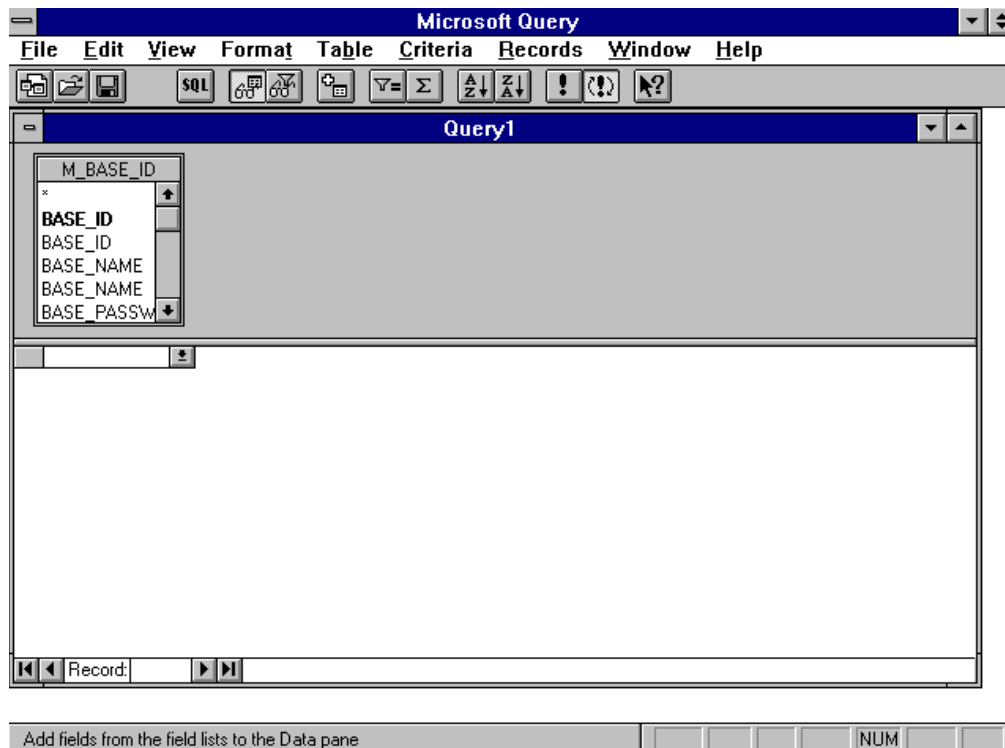
Using MS-Query

To Run MS-QUERY from Microsoft Excel, select Data, Get External Data, Create New Query. The following screen is displayed.



Select the data-source you are going to use to connect (you must have pre-configured a data source with ODBC administrator).

After connecting, a list of the tables on your host you have access to is displayed. A User-ID and password cannot be configured in MS-Query. The ability to enter a User-ID and password has been added to the ODBCLink as part of the data source specifications.



Select one or more tables and click OK. You may join tables by dragging the column you want to join on from one table to another in the upper part of the screen. You may view the data by double-clicking on any column name or by dragging it from the upper to the lower part of the screen.

Note: *MS QUERY must be installed during the installation of Microsoft Excel. Do a custom installation selecting MS Query as one of the installed options.*

Using Visual Basic (2.0 or higher)

You can use ODBCLink either with VB Database controls or with Database and dynaset variables. In addition, of course, you may call ODBC functions directly from the DLL.

To use VB database controls, define a data control and assign it a Connect property of the form "DSN=HP3000;UID=users;PWD=password" and a RecordSource property of the form "SELECT * FROM TableName". If you want to create a SnapShot (no update) set the Options property to 64 (SQL_PASSTHROUGH). You may then place text controls on the screen, set the DataSource property to the name of the data control, and set the DataField property to the name of the column in the Select statement. When the application is run, it will download all records qualified by the Select statement (or all primary key values, depending on the SQL_PASSTHROUGH option) and display the first record on the screen. You may scroll through the records by using the "Up", "Down", "Top", "Bottom" buttons on the data control. If it's a large table, it may take a long time to load (perhaps hours); you may prefer to use direct calls to ODBC.DLL like the VBDEMO application. There is no way to tell Visual Basic to stop loading a table once it has started other than by closing the form.

A second option is to use dynaset variables which gives the program more control, such as for loading a grid with data. Define a database object, assign it Connect and Options properties as above, and open it with an OpenDatabase call, as in SET DBVar=OpenDatabase(...). Then define a dynaset variable (DIM DSVar as DYNASET) and open it with a statement of the form "Set DSVar=DBVar.OpenDynaset (SQLstatement)." If *SQL Statement* is a Select statement, you can read the results from "DSVar.FieldName" and move to the next record with "DSVar.MoveNext". If *SQLStatement* is an Insert or Update statement, no results are returned. You can update records with the *Update* method, and you can get the selection count by calling SQLRowCount or with a query of the form "Select COUNT(*) From *Table* Where *Conditions*". To update the table, you will of course need a primary key defined and you will need to disable the SnapShot only button.

A third option is to call ODBC functions directly. Generally you will do an SQLAllocEnv, SQLAllocConnect and SQLDriverConnect to open the database, followed by SQLExecDirect, followed by multiple SQLFetch and SQLGetData calls. If you plan to use SQLBindCol to bind fetch results to local variables you must use the following procedure:

- A) Declare strings variables as: Dim mystr as String
- B) Initialize your string variables before use: mystr=String(255,0)
- C) Use the ByVal keyword when passing the variable to the ODBC driver. Do not use the ByVal keyword in the "lpbBuf" parameter except for character data, as the goal is to pass the address of the parameter, not the value. Declare SQLBindCol as follows:

Declare Function SQLFindCol Lib "odbc.dll" (ByVal hstmt AS long, ByVal col As Integer, By Val wConvType As Integer, lpbBuf As Any, ByVal dwbuflen As Long, lpcbout As Long) As Integer.

Connection Examples

Connect Using SQL API

1. DECLARE YOUR VARIABLES FOR THE CONNECTION
2. CREATE YOUR CONNECTION STRING
3. ALLOCATE A ENVIROMENT HANDLE
4. ALLOCATE A CONNECTION HANDLE
5. CONNECT TO YOUR DSN WITH SQLDriverConnect
6. ALLOCATE A STATEMENT HANDLE
7. QUERY THE DATASOURCE USING SQLExecDirect
8. CALL AN ERROR ROUTINE IF AN ERROR OCCURS See Sub GetError
9. MAKE SURE YOU RELEASE ANY ENVIROMENT THAT YOU HAVE CREATED WHEN ERRORS OCCUR(or on exiting)

Connect Using SQLAPI in VB 2.0-6.0

The following example demonstrates how to use SQLAPI in VB 2.0-6.0. It is to be used as a basic guide line for creating a project to use SQLAPI. This examples assumes you have the necessary SQL Declare functions and Constants etc.

1.


```
Global sDSNConnect As String 'Connection string
Global henv As Long 'handle to the environment
Global hdbc As Long 'handle to the connection
Global hstmt As Long 'handle to the statement
Global rc As Integer 'Return code
Dim outstr As String * 256
Dim outlen As Integer

Function DBConnect() As Integer
```
2.


```
'Add your connection string to be used

      Example "DSN=MEMBER;UID=USERPWD=PASSWORD;"

      sDSNConnect = "YOUR_DSN_STRING"
```
3.


```
'First Allocate an Environment Handle
      rc = SQLAllocEnv(henv)
      If rc <> SQL_SUCCESS Then
      MsgBox ("SQLAllocEnv failed rc=" + Str(rc))
      Exit Function
      End If
```

4. `Second allocate a connection handle
 rc = SQLAllocConnect(henv, hdbc)
 If rc <> SQL_SUCCESS Then
 MsgBox ("SQLAllocConnect failed rc=" + Str(rc))
 Call GetError
 Call FreeEnv
 Exit Function
 End If

5. `Third allocate the connection and pass in the Connection string
 rc = SQLDriverConnect(hdbc, Form1.hWnd, sDSNConnect, Len(sDSNConnect),
 oustr, 256, outlen, 3)
 If rc <> SQL_SUCCESS Then
 If rc = SQL_NO_DATA_FOUND Then
 Exit Function `User cancelled dialogue
 End If
 Call GetError
 Call Freeconnect
 Call FreeEnv
 Exit Function
 End If

6. `After connecting, allocate a statement handle
 rc = SQLAllocStmnt(hdbc, hstmt)
 If rc <> SQL_SUCCESS Then
 Call GetError
 Call Disconnect
 Exit Function
 End If

7. `Now call an SQL query to select your data
 rc = SQLExecDirect("Select * from member.membership")
 If rc <> SQL_SUCCESS Then
 Call GetError

 MsgBox "Unable to connect to the Database Environment!", vbCritical
 Call Disconnect
 Exit Function
 End If

 End Function

```
8. Sub GetError()
    Dim error_str As String * 256
    Dim SQLState As String * 20
    Dim outlen As Integer
    Dim NativeError As Long
    Dim msg As String

    rc = SQLError(henv, hdbc, SQL_NULL_HSTMT, SQLState, NativeError, error_str,
    256, outlen)sDSNConnect = Space(256)

    If rc <> SQL_NO_DATA_FOUND Then
        msg = Left(error_str, outlen)
        If gl_SQLStatement <> "" Then
            msg = msg + " (" + gl_SQLStatement + ")"
        End If
        MsgBox msg
    End If
End Sub

9. Sub Freeconnect()
    rc = SQLFreeConnect(hdbc)
    If rc <> SQL_SUCCESS Then
        Call GetError
    End If

End Sub

Sub FreeEnv()
    rc = SQLFreeEnv(henv)
    If rc <> SQL_SUCCESS Then
        Call GetError
    End If

End Sub

Sub Freestmt()
    rc = SQLFreeStmt(hstmt, SQL_DROP)
    If rc <> SQL_SUCCESS Then
        Call GetError
    End If

End Sub

Sub Disconnect()

    rc = SQLDisconnect(hdbc)
    If rc <> SQL_SUCCESS Then
        Call GetError
    End If
```

```

rc = SQLFreeConnect(hdbc)
If rc <> SQL_SUCCESS Then
MsgBox ("SQLFreeConnect failed rc=" + Str(rc))
Exit Sub
End If

```

```

rc = SQLFreeEnv(henv)
If rc <> SQL_SUCCESS Then
MsgBox ("SQLFreeEnv failed rc=" + Str(rc))
Exit Sub
End If

```

```
End Sub
```

Connecting with DAO's

1. DECLARE YOUR VARIABLES FOR THE CONNECTION
2. SET THE DAO WORKSPACE
3. SET AND OPEN THE CONNECTION TO THE DATABASE
4. SET AND OPEN THE RECORDSET FOR THE TABLE
5. SE THE RECORDSET INFORMATION YOU NEED AND PROCESS IT ACORDINGLY
6. MAKE SURE YOU CLOSE ANY OBJECTS THAT YOU HAVE CREATED WHEN ERRORS OCCUR(OR WHEN EXITING)

1. Dim DAOWS As DAO.Workspace
Dim DAORS As DAO.Recordset
Dim DAODB As DAO.Database
2. 'SET A WORKSPACE ENVIROMENT FOR YOUR CONNECTION
'AND SET THE TYPE TO DBUSEODBC
'Set DAOWS = CreateWorkspace("NewODBCWorkspace", "", "", dbUseODBC)
3. 'OPEN THE CONNECTION PASS IN YOU CONNECTION STRING
'Set DAODB = Workspaces(0).OpenDatabase("", dbDriverComplete, True,
"ODBC;DSN=YOURDSN;UID=YOURUID;PWD=YOURPASSWORD;")
4. 'OPEN THE TABLE AND ENTER THE SQL QUERY
'Set DAORS = DAODB.OpenRecordset("Select * from membership", dbOpenDynaset,
dbSQLPassThrough)
5. While Not DAORS.EOF
' Insert your query processing code here.
DAORS.MoveNext
Wend
6. 'Close Resultset
DAORS.Close
'Close Data Base Connection
DAODB.Close

Connecting with RDO's

1. DECLARE YOUR VARIABLES FOR THE CONNECTION
2. SET THE RDO ENVIRONMENT
3. SET AND OPEN THE CONNECTION TO THE DATABASE
4. SET AND OPEN THE RECORDSET FOR THE TABLE
5. USE THE RECORDSET INFORMATION YOU NEED AND PROCESS IT ACCORDINGLY
6. MAKE SURE YOU CLOSE ANY OBJECTS THAT YOU HAVE CREATED WHEN ERRORS OCCUR(OR WHEN EXITING)

1. Dim RDOEN As RDO.rdoEnvironment
Dim RDOCN As New RDO.rdoConnection
Dim RDORS As RDO.rdoResultset
Dim RDOCL As RDO.rdoColumn
2. 'Create an RDO Environment and specify the type
'of cursor we want. The Client Side cursor seems
'to give the most accurate results.
Set RDOEN = rdoEngine.rdoEnvironments(0).CursorDriver = rdUseOdbc
3. 'Open a connection to the database
Set RDOCN = RDOEN.OpenConnection(YOURDSN)
4. 'Open the table
Set RDORS = RDOCN.OpenResultset
(“Select * from Member.Membership”,rdOpenKeyset)
5. While Not RDORS.EOF
' Insert your query processing code here.
RDORS.MoveNext
Wend
6. 'Close resultset
RDORS.Close

'Close db connection
RDOCN.Close

Calling ODBC Functions Programmatically

Any program that can call a DLL can use ODBC. This includes "C" and Visual Basic. Note that Visual Basic has built-in ODBC support through Database and Dynaset objects, however it is usually faster to call ODBC functions directly.

First acquaint yourself with ODBC 2.0 specification. The ODBC 2.0 Winhelp has been installed on the host as ODBCAPL.DX.MBFOSTER (this file can be transferred to the PC and called ODBCAPL.HLP). It contains information on all ODBC function calls. A printed version of the ODBC specification is available from Microsoft Press. If you have Visual Basic 3.0, you may also want to look at the Visual Basic samples called VBSAMPL1, VBSAMPL2, and VBSAMPL3 located on the host in the DX.MBFOSTER group.

They are, respectively, self-extracting archives of Microsoft's VBDEMO application, the complete sources of HOSTCONF, and a Visual Basic program that demonstrates the use of Remote Data Objects.

Each file should be down loaded to the PC, renamed respectively, VBSAMPL1.EXE, VBSAMPL2.EXE and VBSAMPL3.EXE, and executed.

Chapter 7

Troubleshooting

Terminating an ODBC Connection

Occasionally it may be necessary to terminate an ODBC session that has deadlocked or is attempting to read a very large table (note that rebooting the PC will not be sufficient as the server has control at this point). You may do this from ISQL with the command `TERMINATE USER user@account`.

When an ODBC client terminates abnormally, the server process that handles the connection will normally go down after the timeout interval has expired. This is typically 30 minutes to 2 hours.

On an HP-UX system, you can kill the process manually (if you know the process-id) with the command “`kill -16 pid`” (*pid* is the process id).

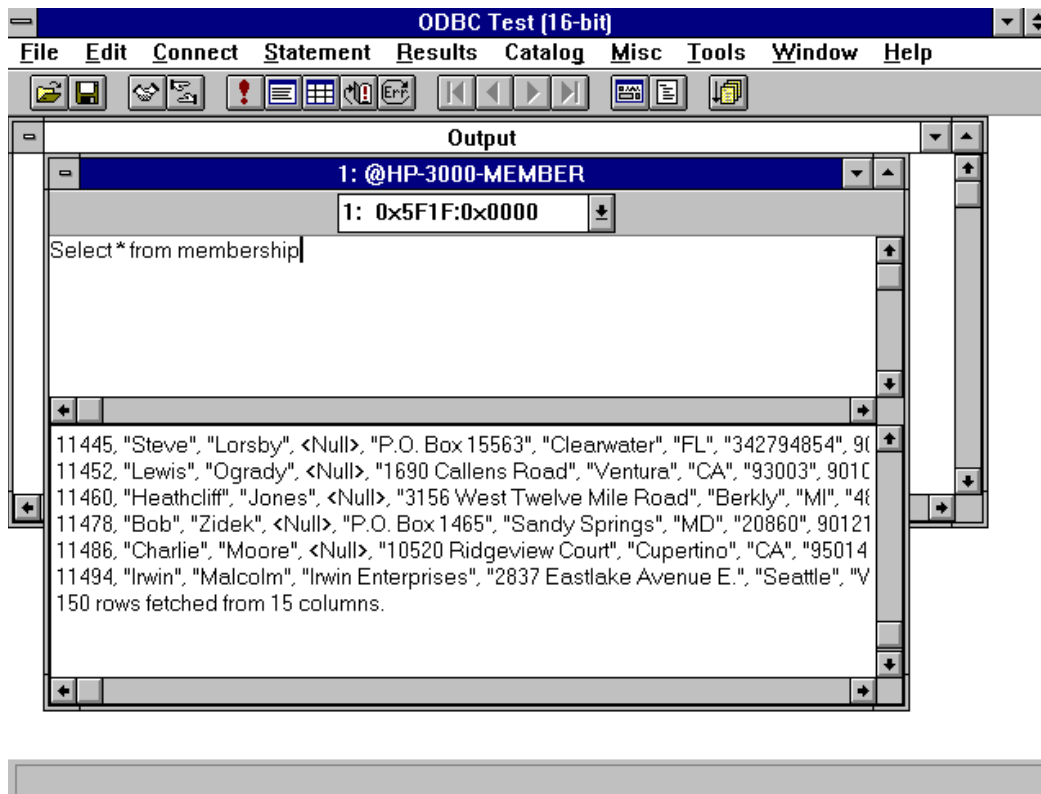
On MPE/iX you must either use the ISQL `TERMINATE USER` command or abort the listener job using the `ABORTJOB` command.

Using Microsoft ODBC Test (32-bit)

You can use ODBC Test to verify correct functioning of the driver and to enter ad-hoc SQL commands. ODBCTEST and ODBCTE32 are part of the ODBC SDK that comes with the Microsoft Developers Network.

To run ODBCTE32.EXE:

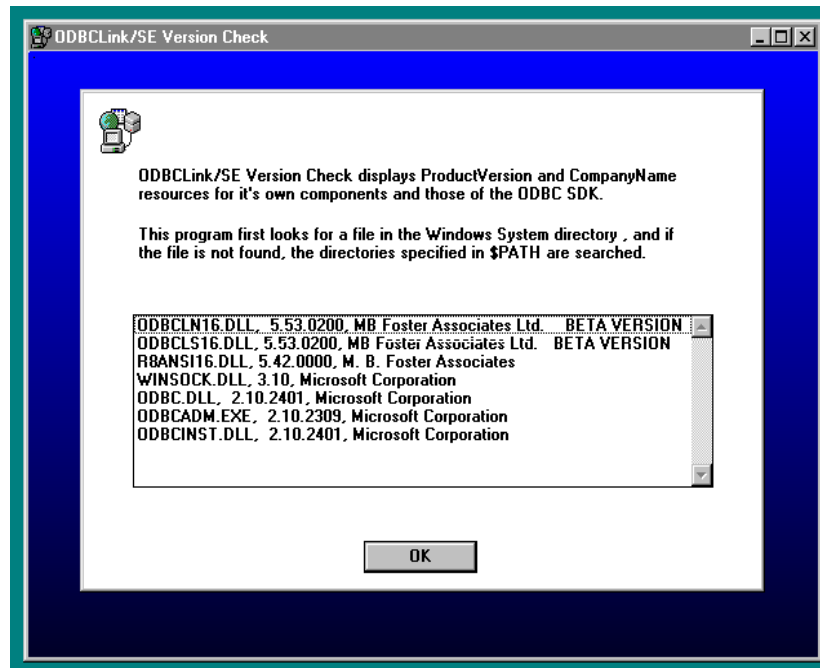
- ⊗ **Select Connect, Full Connect. You must enter a data source name, User-ID and a password, and click “OK”. You should get a message “Successfully connected to *databasename*”.**
- ⊗ **You can get a list of tables by selecting Catalog, SQL Tables, then click OK and then Results, Get Data All.**
- ⊗ **You can enter an SQL statement, such as “Select * from *tablename*” in the Query Window and then execute it by choosing Statement, SQLExecdirect and view the results with “Results, Get Data All”.**



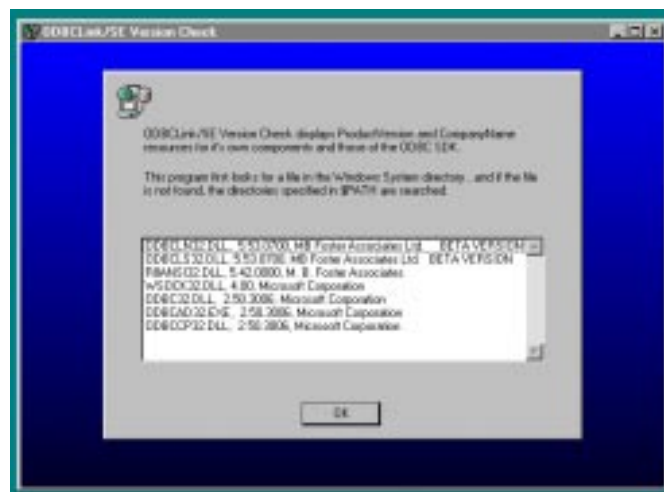
Verifying Software Version Numbers

Verifying the Client-Side Components

To determine which version of ODBCLink/SE is installed, run either WHAT16.EXE or WHAT32.EXE from the RUN window for from an MS-DOS prompt window.



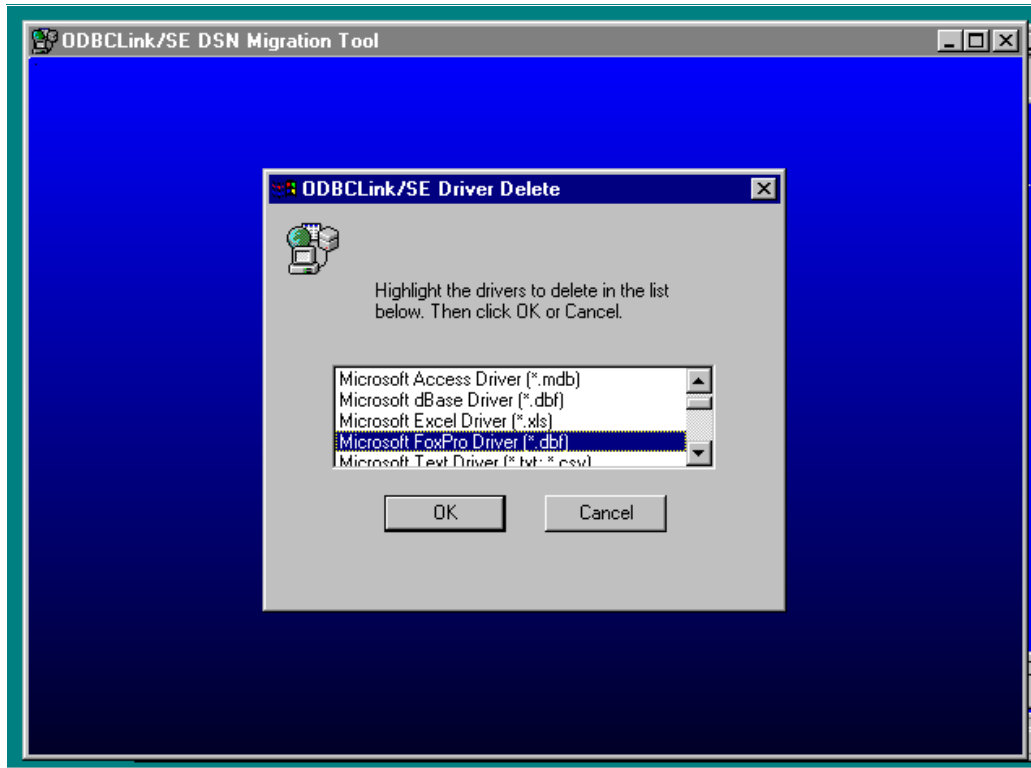
Note: Press OK to exit before running the other program when checking both 16 and 32-bit versions.



Deleting Unwanted Drivers

You can delete unwanted 16-bit drivers from the 16-bit ODBC Administrator on the Control Panel. Click on Drivers, highlight and unwanted driver and click on Delete.

To delete unwanted 32-bit drivers, run the driver delete tool DRVDEL32.EXE from either the RUN window or an MS-DOS prompt window.



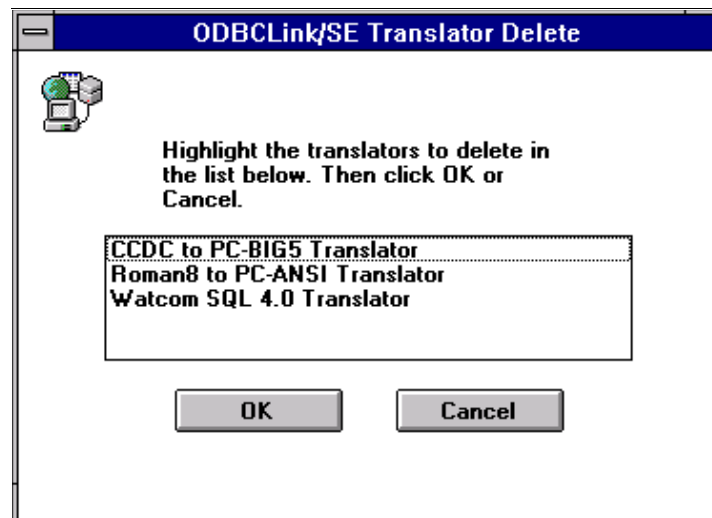
Highlight the drivers you wish to delete and click OK. The driver name and all related information will be removed from the system registry.

Deleting a Translation DLL

Run TRNDEL16.EXE or TRNDEL32.EXE from Program Manager by selecting FILE then RUN and entering the directory path and program name, or from File Manager by double clicking on the file name.



A welcome screen will be shown first. Click on OK to display the next screen.



Select the translator you want to uninstall and press OK. The translator information will then be deleted from the registry. If the program is successful it will show a window indicating the successful removal.



Once the translator has been deleted it will no longer be displayed in the translator list in the driver setup. To add the translator back in simply rerun the ODBC driver install program.

Note: *The translator DLL file will still be in the windows directory after the program is run, but it will not be shown in the registries translator information.*

Tools on the Database Server

Monitoring Activity with HP GlancePlus/XL (HP3000) Program.

HP GlancePlus/XL allows you to monitor a job, session, or process on MPE/iX. The GlancePlus screens display statistics on how different system resources are used. For information, refer to the HP GlancePlus/XL User's Manual.

Monitoring Activity with HP GlancePlus/UX (HP9000) Program.

HP GlancePlus/UX allows you to monitor a login or process on HP-UX. The GlancePlus screens display statistics on how different system resources are used. For information, refer to the HP GlancePlus/UX User's Manual.

Displaying Active Processes with HP-UX Process Status (ps) Command.

You can use the HP-UX ps command to display the status of your active processes. For information, refer to the HP-UX man page.

Monitoring ALLBASE/SQL Activity with SQLMON.

SQLMON is a component of ALLBASE/SQL and monitors the activity of an ALLBASE/SQL DBEnvironment. SQLMON summarizes the activity for the entire DBEnvironment, or focuses on individual sessions, programs, or database components. For information, refer to the ALLBASE/SQL Performance and Monitoring Guidelines.

Checking the Listener Log File on the HP3000

Look for ODBCLOG where the listener is running.

Checking the Listener Log File on the HP9000

Look for odbcblog in the /tmp/ directory.

Using the Host Testing Utility ODBCUTSE

If you are having problems connecting or accessing a table on the host, you can access it directly on the host in the same way the ODBC server accesses it, by running the ODBCUTSE utility.

To run the program on MPE/iX, enter:

```
ODBCUTSE.ODBCSE.SYS "dbe name"
```

To run the program on HP/UX, enter:

```
or /usr/bin/odbcse/odbcutse "dbe name" (for HP-UX V9.x systems)
```

```
/opt/allbase/bin/odbcutse "dbe name" (for HP-UX V10.x systems)
```

You should now be connected to the database. To see a menu of available commands, enter HELP:

```

0>HELP

SHOW Display tables and table structures
EXIT Exit program
SELECTSQL Select
FETCH SQL Fetch [ALL|##]
INSERTSQL Insert
UPDATESQL Update
DELETESQL Delete
SQLExecute immediate SQL command
COMMITSQL Commit
PARAMParameter substitution on a prepared statement
STMT Change current statement/cursor number
ROLLBACK SQL Rollback
EXECUTEExecute stored procedure
QUIT Exit program
FOREIGNDisplay foreign Keys
TABPRIVDisplay table privileges
0>
    
```

Enter the command you wish to use. For example, enter SHOW to see a list of available tables:

```

>show
Dataset/Table      Database           Type
MANUFDB.SUPPLYBATCHES  partsdbe.sql.mbfest  ALLBASE TABLE
MANUFDB.TESTDATA      partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.PARTS         partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.INVENTORY     partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.SUPPLYPRICE   partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.VENDORS       partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.ORDERS        partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.ORDERITEMS    partsdbe.sql.mbfest  ALLBASE TABLE
PURCHDB.PARTINFO      partsdbe.sql.mbfest  ALLBASE VIEW
PURCHDB.VENDORSTATS  partsdbe.sql.mbfest  ALLBASE VIEW
RECDB.CLUBS           partsdbe.sql.mbfest  ALLBASE TABLE
RECDB.MEMBERS         partsdbe.sql.mbfest  ALLBASE TABLE
RECDB.EVENTS          partsdbe.sql.mbfest  ALLBASE TABLE
    
```

To see the list of columns and indexes for a table, enter SHOW followed by the name of the table. For example:

```

>sho recdb.clubs

Field      ODBC-Datatype  HP-Datatype  Length  Offset  Nulls
CLUBNAME   CHAR(15)       15           0
CLUBPHONE  SMALLINT       2            16      1
ACTIVITY   CHAR(18)       18           20      1

Indexed field      Index Type      Index Name
CLUBNAME           CONSTRAINT PRIMARY  CLUBS_PK
    
```

Host Logging

Normally, only connections and errors are logged to the ODBCLOG file on the host. If you check the box labelled "Trace SQL calls on the server" in the data-source setup dialogue, all the SQL sent from the host and the ALLBASE operation will be logged as well.

If a listener error occurs, an error message is logged in the ODBCLOG file

Here is the format of the ODBCLOG file (this file can be found in group and account of the listener job, typically ODBCLOG.ODBCSE.SYS on MPE systems and /tmp on Unix systems)

```
96/11/20 14:12:09 068ODBC listener started
96/11/20 14:13:13 068 Listener created new connection as SYS1 (PIN 77), IP=192.9.3.102
96/11/20 14:13:13 077 ODBCLNSE E.52.00 started from sockets
96/11/20 14:13:22 077 Allbase Prepare[49]SELECT NAME, OWNER, TYPE, NPAGES, NROWS FROM
    CATALOG.TABLE;
96/11/20 14:13:26 077 Allbase SetParams[49]
96/11/20 14:13:27 077 Allbase Fetch[49] Rows=119 MaxRows=200
96/11/20 14:13:27 077 Allbase Fetch[49] Rows=0 MaxRows=200
96/11/20 14:13:27 077 Allbase Close cursor[49]
96/11/20 14:13:27 077 Allbase Commit
96/11/20 14:13:27 077 Allbase Prepare[49]SELECT NAME,OWNER,NUMP,NUMR,MULTIRESET FROM
    CATALOG.PROCEDURE;
96/11/20 14:13:27 077 Allbase SetParams[49]
96/11/20 14:13:28 077 Allbase Fetch[49] Rows=6 MaxRows=200
96/11/20 14:13:28 077 Allbase Fetch[49] Rows=0 MaxRows=200
96/11/20 14:13:28 077 Allbase Close cursor[49]
96/11/20 14:13:28 077 Allbase Commit
96/11/20 14:13:28 077 Connected to partsdbe, ClientVersion=E.52.00
96/11/20 14:13:35 077 Prepare[0]SELECT _TABLES
96/11/20 14:13:35 077 SetParams[0] nparam=0
96/11/20 14:13:53 077 FreeStmt[0]
96/11/20 14:13:54 077 Normal exit, Received 3 requests (133 bytes), Sent 0 records (5141 bytes), CPU=2
    (5.6%)
```

The first three columns of the list are the date, time, and pin (process id number) of the requesting connection. The last column is the Allbase operation being performed, with the statement or cursor number in square brackets. The number in square brackets (as in Allbase prepare[49]) is the statement number.

Tools on the PC Client

The preferred method to trace ODBC calls on the client is by setting the “Trace ODBC Calls” check box in ODBC setup. However, Syware’s Dr. DeeBee Spy is installed with ODBCLink/SE (in the Windows directory). It can be used to trace calls to any ODBC driver.

ODBC Call Tracing using ODBCLink.LOG

To log all the ODBC calls made by the client, check the box “Trace ODBC calls on the client” on the data-source setup screen. This will create a file, called ODBCLINK.LOG, of all the ODBC calls made by the client. This file normally resides in the directory where the ODBC application was started from. Some applications however change their working directory, for MS-Access for instance look for the log file in “My Documents” directory.

Here is the format of the ODBCLINK.LOG file:

```

SQLDriverConnect(0) hstmt=0 Ver= 5.52.0000 szConnStrIn=<DSN=DBA
MEMBER; UID=Admin; PWD=>,Completed_string=<DSN=DBA MEMBER;
UID=#mpeix/192.9.3.10:MembrDBE.SQL.mbfest,,an# DBA,dba/.MBFTEST/
password,work/> Login=<>

SQLError(100) hstmt=0 pfNativeError=0,szErrorMsg=
SQLGetInfo(0) hstmt=0 flInfoType=23, rgbInfoValue= [1,0,0,0]
SQLSetConnectOption(0) hstmt=0 fOption=101 vParam=1
SQLAllocStmt(0) hstmt=0
SQLGetStmtOption(0) hstmt=0 fOption=0 vParam=0
SQLSetStmtOption(-1) hstmt=0 fOption=0 vParam=60
SQLError(0) hstmt=0 pfNativeError=-57,szErrorMsg=[ODBCLN32.DLL] Driver not
capable
SQLError(100) hstmt=0 pfNativeError=0,szErrorMsg=

SQLExecDirect(0) hstmt=0 SELECT
MEMBERDB.ACCTHIST.NUMBER,MEMBERDB.ACCTHIST.TRANTYPE,MEMB
ERDB.ACCTHIST.
TRANDATE FROM MEMBERDB.ACCTHIST

SQLFetch(0) hstmt=0
SQLGetData(0) hstmt=0 icol=1 fCType=99 pcbValue=4 data= '[25,39,0,0]
SQLGetData(0) hstmt=0 icol=2 fCType=99 pcbValue=3 data=INV[73,78,86,0]
SQLGetData(0) hstmt=0 icol=3 fCType=99 pcbValue=6 data=É [-56,7,1,0]

```

Each line starts with the name of the function being called, with the return code in brackets, followed by the different parameters of the call. Refer to the ODBC SDK manual for a description of the parameters of the call.

ODBC Call Tracing using Dr. DeeBee Spy

Dr. DeeBee Spy traces calls to the ODBC driver. Dr. DeeBee Spy logs each function call along with the input and output values. To turn on logging, do the following:

- ⊗ **From the Program Manager Menu in Windows, click on RUN.**
- ⊗ **Type DRDBSP.**
- ⊗ **Choose OK.**
- ⊗ **Select the Data Source Name to be traced.**
- ⊗ **Choose OK.**

The Dr. DeeBee Spy icon will appear at the bottom of the screen. It will flash when it is logging.

To turn off logging:

- ⊗ Click on the Dr. DeeBee Spy icon.
- ⊗ Choose CLOSE.

The log is located in C:\WINDOWS\DRDEEBEE.LOG.

Note: *“Dr. DeeBee Spy is © 1995 Syware, Inc., All rights reserved”*

Modifying the System Registry and ODBC.INI Files

You only need to do this if you want to modify an option that you cannot configure in the ODBC control panel. An example of this is the MAXSTMT options.

To add or change an option, either edit the ODBC.INI file (for 16-bit data-sources) or run the registry editor (regedt32.exe). In the registry editor, look under HKEY_CURRENT_USERS / Software / Microsoft / ODBC.INI.

Error Messages

Errors in ODBCLink/SE can come from several sources; all messages have a prefix which indicates the source of the error, as follows:

[ODBCLN16]	Generated by the 16-bit client DLL.
[ODBCLN32]	Generated by the 32-bit client DLL.
[WINSOCK]	Comes from the Winsock driver.
[ALLBASE]	Generated by the ALLBASE/SQL command interpreter.

The first action you should take in case of a problem is to attempt to execute the command manually, e.g. try to login using a terminal emulator, or else try issuing the SQL statement that causes a problem in ODBCUTSE or ISQL. If you do not know the command causing the error (because you are using Microsoft Access for instance), turn on host logging and examine your ODBCLOG file.

Errors from ALLBASE/SQL or IMAGE/SQL are followed by DBERR or DBWARN and a number. These error messages are documented in the ALLBASE/SQL Message Manual or in the IMAGE/SQL Administration Guide.

Error messages are returned by ODBCLink/SE to the application. Each message is followed by a CAUSE which gives information as to why the warning or error occurred. This is followed by an ACTION which gives information on how to resolve the problem.

Note: *For all other errors, refer to the appropriate documentation or manuals.*

Appendix A

Implementation Notes

Following is a list of features that are unique or specific to ODBCLink/SE and some details about them.

Primary key name returned by SQLStatistics

Most ODBC applications including Microsoft Access, Visual Basic and PowerBuilder require a unique primary key for a table to be updatable.

Note: *If a unique key has not be defined in ALLBASE/SQL the table will not be updatable.*

Reading or writing to ALLBASE/SQL LongVarBinary items

Any file residing on the HP can be inserted into an ALLBASE/SQL LongVarBinary field with the SQL command:

```
INSERT INTO Table VALUES (... , '<BLOB >%' , ...);
```

Where BLOB is the name of the file on the HP and the '>%' tells ALLBASE/SQL to copy the contents to a memory location when retrieving the row. You can also have ALLBASE/SQL copy it to a file during retrieval ('<BLOB >OUTFILE'). In either case, the data can be retrieved in the standard way (SQLFetch followed by SQLGetData in chunks).

You may also create a binary column on the HP by using ODBC functions call sequence: SQLPrepare, SQLBindparameter (DATA_AT_EXECUTION), SQLExecute, SQLParamData, SQLPutData.

Maximum number of statements

ODBCLink/SE supports up to 50 concurrent statements, or cursors, per connection. However, SQLGetInfo with option SQL_MAX_STMT will only report 1 as the maximum number of concurrent statements. This is because multiple statements (or cursors) in ALLBASE/SQL on the same connection are not truly independent. Having SQLGetInfo return 1 for SQL_MAX_STMT forces MS-Access, and other applications to use multiple connections (multiple SQLDriverConnect's) instead of multiple statements on the same connection (one SQLDriverConnect and multiple SQLAllocStmt's).

You **can** use multiple statements from within the same connection (in fact, some applications ignore the SQLGetInfo and do this anyway). There is a performance advantage to doing things this way since a new statement (or cursor) in ALLBASE/SQL takes much less overhead than a new connection. However you should be aware of the following:

- ⊗ **In ALLBASE/SQL, a COMMIT or ROLLBACK operation closes all open cursors within the connection. If you have multiple Select statements going on different statements, the next time you SQLFetch on one statement, after a Commit on another statement, you will get an error from ALLBASE/SQL "ALLBASE has closed the cursor for this statement". You may not get this error immediately, as ODBCLink/SE caches a certain number of rows during a fetch.**

- ⊗ **If you are in AUTOCOMMIT mode ON (this is the default option, set through SQLSetConnectOptions and in ODBC Administrator), there is an implicit Commit done whenever you fetch the last row. This is so that locks held on the database by the open cursor are released. This may destroy other open cursors. Note also that opening a table for the first time automatically triggers various initialization calls that cause a commit to be done when AUTOCOMMIT is ON.**
- ⊗ **If you are using prepared statements with substitution parameters to insert or update you need not be concerned about this, as the prepared statement will still be valid after a Commit, and you may bind new parameters and SQLExecute again with no problem. Prepared statements are SQL statements with question marks '?' that you SQLPrepare and then call SQLBindParam or SQLBindCol.**

If you wish to use multiple statements, but are not writing your own SQL (i.e. you are using a 4GL application) you can tell ODBCLink/SE to have SQLGetInfo report more than 1 statement per connection. To do this, modify the ODBC.INI or the registry manually, and add a new parameter "MAXSTMT=50".

ALLBASE/SQL and IMAGE/SQL Restrictions on the ODBC Grammar

The following table summarizes the ALLBASE/SQL restrictions on the ODBC grammar:

Statement	Programming Considerations
CREATE TABLE	UNIQUE PRIMARY KEY must follow NOT NULL
DROP TABLE	ALLBASE/SQL does not provide CASCADE or RESTRICT
REVOKE	ALLBASE/SQL does not provide RESTRICT.

ALLBASE/SQL provides a DATETIME data type that is similar to TIMESTAMP.

ALLBASE/SQL does not implement optimistic locking. Optimistic locking means that the DBMS does not lock data until just before an update is made, thus improving concurrency because locks are not held for long.

Using the ANSI Character Set

The HP3000 and HP9000 both use the default ROMAN8 character set. Most applications running under Microsoft Windows use the 8-bit ANSI (ISO 8859/1) character set to provide support for Western European languages (including American). The first 127 characters are the same for both ROMAN8 and ANSI character sets, so only the extended and special characters are different.

If your PC client application uses these extended or special characters, then the data from the database server must be converted from ROMAN8 to ANSI, and the data returning to the database server must be converted from ANSI to ROMAN8. To activate the conversion, select the "Roman8 to PC-ANSI translation DLL during the data-source setup. The conversion takes place on the PC client.

Bind variables of datatype CHAR and VARCHAR are also converted. A bind variable in an SQL statement is associated (bound) to variables defined in a program. A bind variable is another name for a dynamic parameter.

Note: *Roman9 to PC-ANSI Translator will recognize and convert the symbol used for the European Currency Symbol, the EURO.*

Some characters cannot be converted to ROMAN8 and then back to ANSI. For example, when the copyright sign in ANSI is converted to ROMAN8, it is changed to a lowercase "c." When the data is returned to the PC client, the copyright sign cannot be recognized, as it was converted to a "c." The following characters, sent from the PC client, cannot be properly converted from ANSI to ROMAN8 in a round-trip fashion. (The hex designation is listed along with the character name.)

- _A6 broken bar
- _A9 copyright sign
- _AC not sign
- _AE registered trade mark
- _B2 superscript 2
- _B3 superscript 3
- _B8 cedilla
- _B9 superscript 1
- _D7 multiply sign
- _F7 divide sign

Similarly, the following characters, sent from the database server, cannot be properly converted from ROMAN8 to ANSI in a round-trip fashion.

- _A9 accent grave
- _AA circumflex accent
- _AC tilde accent
- _BE Dutch guilder
- _EB uppercase S caron
- _EC lowercase s caron
- _EE uppercase Y umlaut
- _F6 long dash
- _FC solid box

Unsupported ALLBASE/SQL and IMAGE/SQL Statements

There are several reasons why some ALLBASE/SQL and IMAGE/SQL statements are not supported in the ODBCLink/SE environment. They are explained here:

- ⊗ **Several ALLBASE/SQL statements work only with embedded SQL; ODBCLink/SE does not support embedded SQL.**
- ⊗ **Some ALLBASE/SQL statements relate to functions, such as connection, that have been replaced by functions resident on the PC client.**
- ⊗ **Some ALLBASE/SQL and IMAGE/SQL functions are best restricted to the DBA; statements which control these functions are accessible only through Interactive Structured Query Language (ISQL) on the server.**
- ⊗ **ODBCLink/SE internally prepares an SQL statement to be dynamically preprocessed. Therefore, you cannot use some statements such as EXECUTE IMMEDIATE.**

ALLBASE/SQL and IMAGE/SQL Statements That Work Only with Embedded SQL

PC client application software that accesses ALLBASE/SQL and IMAGE/SQL databases through ODBCLink/SE do not support embedded SQL. The application software calls the database functions directly; there is no need for preprocessing.

Because the following ALLBASE/SQL and IMAGE/SQL statements function only with embedded SQL, they are not supported by ODBCLink/SE:

```
CLOSE CURSOR  
BEGIN DECLARE SECTION  
DECLARE CURSOR  
DESCRIBE  
END DECLARE SECTION  
EXECUTE  
EXECUTE IMMEDIATE  
FETCH  
INCLUDE  
OPEN  
PREPARE  
REFETCH  
SQLEXPLAIN  
WHENEVER
```

ALLBASE/SQL and IMAGE/SQL Statements Replaced by Functions on the PC Client

Some of the functionality has been moved from the database server to the PC client with ODBCLink/SE.

The following ALLBASE/SQL and IMAGE/SQL statements are not supported because they have been replaced by other functions resident on the PC client:

```
CONNECT  
DISCONNECT  
RELEASE  
RESET  
SET CONNECT  
SET MULTITRANSACTON  
START DBE  
START DBE NEW  
START DBE NEWLOG  
STOP DBE
```

Note: *If you attempt to use any of these unsupported statements, you will receive an error message.*

Appendix B

Creating a DBEnvironment

On the HP3000 Database Server

To enable a connection between a PC client and database server, you must set up an ALLBASE/SQL DBEnvironment. If you do not have an existing database for testing purposes, you can use the ALLBASE/SQL SQLSetup tool to create a DBEnvironment called PartsDBE. If you are using IMAGE/SQL, you can use the IMSQL.SAMPLEDB.SYS command to create an IMAGE/SQL database called MusicDBE. *For more information, refer to Getting Started with HP IMAGE/SQL.*

Follow these steps to create the PartsDBE database on the HP3000:

- ⊗ **Logon to the group and account where you want to create the DBEnvironment.**
- ⊗ **From the command prompt, enter the following command:**

:SQLSETUP.SAMPLEDB.SYS

A menu like the following will appear on your screen:

Options for Setting Up ALLBASE/SQL Sample DBEnvironments

=====

Choose one:

1. Create PartsDBE without sample programs
2. Create PartsDBE, copy, preprocess and compile sample programs
3. Copy, preprocess and compile sample programs only
4. Generate a schema for PartsDBE
5. Display schema for PartsDBE
6. Purge PartsDBE and sample programs
7. Help
0. Exit

=====

Enter your choice=>

- ⊗ **Choose option 1 to create PartsDBE. This option creates the DBEnvironment, defines all of its tables, views, indexes, and security structure, and then loads it with data. As the system creates PartsDBE, you see several messages displayed. At the end of the creation process, you see the following message:**

Creation and Loading of PartsDBE is now complete!
Press Return to continue...

- ⊗ **Choose 0 to exit the menu.**

If you have designed your own database for testing purposes, you can consult your database administrator and complete the following tasks:

- ⊗ **Plan the security structure of the DBEnvironment. Decide which users will be granted access to the DBEnvironment, databases within the DBEnvironment, and tables within the databases.**
- ⊗ **Create the account where the DBEnvironment is to reside.**
- ⊗ **Grant authorizations to users who will be using the test DBEnvironment.**
- ⊗ **Create the DBEnvironment, including databases and tables.**

After you setup the DBEnvironment, ensure that you can connect to it. For example,

```
: isql
isql=> CONNECT TO 'PartsDBE.SomeGrp.SomeAcct';
isql=> SELECT * FROM SYSTEM.TABLE;
:
At this point you should see the system table information.
:
U[p],d[own],l[eft],r[ight],t[op],b[ottom],pr[int]<n>,or e[nd]
    > e;
isql=> exit;
```

On the HP9000 Database Server

To enable a connection between a PC client and database server, you must set up an ALLBASE/SQL DBEnvironment. If you do not have an existing database for testing purposes, you can use the ALLBASE/SQL SQLSetup tool to create a DBEnvironment called PartsDBE.

Follow these steps to create the PartsDBE database on the HP9000:

- ⊗ **Login to the directory where you want to create the DBEnvironment.**
- ⊗ **From the C shell, enter the following command:**

```
$ /usr/lib/allbase/hpsql/sqlsetup
```

•From the Korn shell or Bourne shell, enter the following command:

```
$ csh /usr/lib/allbase/hpsql/sqlsetup
```

A menu like the following will appear on your screen:

Options for Setting Up ALLBASE/SQL Sample DBEnvironments

```
=====
Choose one:
```

1. Create PartsDBE without sample programs
2. Create PartsDBE, copy, preprocess and compile sample programs
3. Copy, preprocess and compile sample programs only
4. Generate a schema for PartsDBE
5. Display schema for PartsDBE
6. Purge PartsDBE and sample programs
7. Help
0. Exit

```
=====
Enter your choice=>
```

- ⊗ **Choose option 1 to create PartsDBE. This option creates the DBEnvironment, defines all of its tables, views, indexes, and security structure, and then loads it with data. As the system creates PartsDBE, you see several messages displayed. At the end of the creation process, you see the following message:**

```
Creation and Loading of PartsDBE is now complete!
Press Return to continue...
```

- ⊗ **Choose 0 to exit the menu.**

If you have designed your own database for testing purposes, you can consult your database administrator and complete the following tasks:

- ⊗ **Plan the security structure of the DBEnvironment. Decide which users will be granted access to the DBEnvironment, databases within the DBEnvironment, and tables within the databases.**
- ⊗ **Create the account where the DBEnvironment is to reside.**

- ⊗ **Grant authorizations to users who will be using the test DBEnvironment.**
- ⊗ **Create the DBEnvironment, including databases and tables.**

After you setup the DBEnvironment, ensure that you can connect to it. For example,

```
$ isql
isql=> CONNECT TO '/users/hpsql/sampledb/PartsDBE';
isql=> SELECT * FROM SYSTEM.TABLE;
:
At this point you should see the system table information.
:
U[p],d[own],l[eft],r[ight],t[op],b[ottom],pr[int]<n>,or e[nd]
> e;
isql=> exit;
```

Appendix C List of Installed Files

16 Bit Driver Client - For Windows 3.1 or Windows 3.11

Note: “..\” is a short-hand that represents the location of the windows installation (e.g. “c:\windows\”)

Install Disc	Windows 3.x	Description
_bootstp.exe	not installed to hard drive	
_bootstp.obj	not installed to hard drive	
mssetup.ex	not installed to hard drive	
b5ccdc16.dl_	..\system\b5ccdc16.dll	Used for character translation for Chinese characters
cpn16ut.dl_	..\system\cpn16ut.dll	16 bit installer Universal Thunking DLL. Used by 32-bit applications with 16-bit installer DLL
ctl3dv2.dl_	..\system\ctl3dv2.dll	The DLL for 3D controls
drvsetup.exe	not installed to hard drive	16-bit ODBC driver setup program
dsnmig.ex_	..\system\dsnmig.exe	Migrates 16bit PCAPI clients to ODBCLink/SE client
drdbsp16.ex_	..\system\drdbsp16.exe	16-bit Dr. DeeBee spy used for debugging
drdbsp16.dl_	..\system\drdbsp16.dll	16-bit Dr. DeeBee spy used for debugging
odbc.dl_	..\system\odbc.dll	16-bit driver manager (Microsoft)
odbc.inf	not installed to hard drive	Setup information file
odbc16ut.dl_	[win]\system\odbc16ut.dll	16-bit ODBC Universal Thunking DLL. Used by 32-bit applications with 16-bit driver manager
odbc32.dl_	..\system\odbc32.dll	32-bit Thunking Driver manager
odbcadm.ex_	..\system\odbcadm.exe	ODBC Administrator application
odbccp32.dl_	..\system\odbccp32.dll	Windows control Panel ODBC Installer Tool
odbccurs.dl_	..\system\odbccurs.dll	Microsoft cursor library DLL. Can be used by client to implement scrollable cursors
odbcinst.dl_	..\system\odbcinst.dll	ODBC administrator DLL. Called by ODBC Control Panel Application
odbcinst.hl_	..\system\odbcinst.hlp	Help file for odbcinst.dll
odbcln16.dl_	..\system\odbcln16.dll	16-bit version of the ODBCLink/SE driver
odbcls16.dl_	..\system\odbcls16.dll	16-bit version of ODBCLink/SE setup

r8ansi16.dl_	..\system\r8ansi16.dll	16-bit character translation DLL for translation between the ANSI and roman-8 character set.
r9ansi16.dl_	..\system\r9ansi16.dll	16-bit character translation DLL for translation between the ANSI and roman-9 character set.
setup.exe	not installed to hard drive	Microsoft ODBC Setup program
setup.lst	not installed to hard drive	
trndel16.exe	..\system\trndel16.exe	Used to delete/uninstall the 16-bit translation DLL
what16.ex_	..\system\what16.exe	Utility to determine what versions of drivers required by ODBC application are currently loaded on the PC

32 Bit Driver Client - For Windows 95

Note: “..|” is a short-hand that represents the location of the windows installation (e.g. c”\windows|”)

Install Disc	Windows 95	Description
_bootstp.exe	not installed to hard drive	
mssetup.ex	not installed to hard drive	
ctl3d95.dl_	..\system\ctl3d95.dll	DLL for 3D controls in Windows95
b5ccdc32.dl_	..\system\b5ccdc32.dll	Used for character translation for Chinese characters
drdbsp32.ex_	..\system\drdbsp32.exe	32-bit Dr. DeeBee spy used for debugging
drdbsp32.dl_	..\system\drdbsp32.dll	32-bit Dr. DeeBee spy used for debugging
drvstp32.exe	not installed to hard drive	32-bit ODBC driver setup program
dvrdel32.ex_	..\system\dvrdel32.exe	Utility used to delete a 32-bit driver from the registry
ds16gt.dl_	..\system\ds16gt.dll	16-bit Driver Setup Generic Thunking DLL. 16-bit applications use this in conjunctions with the 32-bit Installer DLL
ds32gt.dl_	..\system\ds32gt.dll	32-bit Driver Setup Generic Thunking DLL.
mfc30.dl_	..\system\mfc30.dll	Microsoft Foundation Class Library
msvcrt20.dl_	..\system\msvcrt20.dll	Microsoft Run-Time Library
odbc.inf	not installed to hard drive	Installation control file
odbc16gt.dl_	..\system\odbc16gt.dll	16-bit ODBC Generic Thunking DLL
odbc32.dl_	..\system\odbc32.dll	32-bit ODBC Driver Manager
odbc32gt.dl_	..\system\odbc32gt.dll	32-bit ODBC Generic Thunking DLL
odbcad32.ex_	..\system\odbcad32.exe	ODBC Administrator Application
odbccp32.cp_	..\system\odbccp32.cpl	32-bit ODBC Installer Control Panel Tool
odbccp32.dl_	..\system\odbccp32.dll	32-bit ODBC Control Panel Installer DLL
odbccr32.dl_	..\system\odbccr32.dll	32-bit cursor library
odbcinst.hl_	..\system\odbcinst.hlp	Installation help file
odbcIn32.dl_	..\system\odbcIn32.dll	32-bit ODBCLink/SE Driver
odbcls32.dll	..\system\odbcls32.dll	32-bit ODBCLink/SE Driver Setup DLL
r8ansi32.dl_	..\system\r8ansi32.dll	32-bit character translation DLL for translations between the ANSI and roman-8 character sets.

r9ansi32.dl_	..\system\r9ansi32.dll	32-bit character translation DLL for translations between the ANSI and roman-9 character sets.
setup.exe	not installed to hard drive	Microsoft ODBC Setup program drive
setup.lst	not installed to hard drive	
trndel32.exe	...\system\trndel32.exe	Translator uninstall utility to “de-register” translator information
what32.ex_	...\system\what32.exe	Utility to determine what versions of drivers required by ODBC application are currently loaded on the PC

32 Bit Driver Client - For Windows NT

NOTE: “..|” is a short-hand that represents the location of the windows installation (e.g. c”|windows|”)

Install Disc	Windows NT	Description
_bootstp.exe	not installed to hard drive	
mssetup.ex	not installed to hard drive	
b5ccdc32.dl_	..\system32\b5ccdc32.dll	Used for character translation for Chinese characters
ctl3dnt.dl_	..\system32\ctl3dnt.dll	DLL for 3D controls in WindowsNT
drdbsp32.ex_	..\system32\drdbsp32.exe	32-bit Dr. DeeBee spy used for debugging
drdbsp32.dl_	..\system32\drdbsp32.dll	32-bit Dr. DeeBee spy used for debugging
drvstp32.exe	not installed to hard drive	32-bit ODBC driver setup program
dvrdel32.ex_	..\system32\dvrdel32.exe	Utility used to delete a 32-bit driver from the registry
ds16gt.dl_	..\system32\ds16gt.dll	16-bit Driver Setup Generic Thunking DLL. 16-bit applications use this in conjunctions with the 32-bit Installer DLL
ds32gt.dl_	..\system32\ds32gt.dll	32-bit Driver Setup Generic Thunking DLL.
mfc30.dl_	..\system32\mfc30.dll	Microsoft Foundation Class Library
msvcrt20.dl_	..\system32\msvcrt20.dll	Microsoft Run-Time Library
odbc.inf	not installed to hard drive	Installation control file
odbc16gt.dl_	..\system32\odbc16gt.dll	16-bit ODBC Generic Thunking DLL
odbc32.dl_	..\system32\odbc32.dll	32-bit ODBC Driver Manager
odbc32gt.dl_	..\system32\odbc32gt.dll	32-bit ODBC Generic Thunking DLL
odbcad32.ex_	..\system32\odbcad32.exe	ODBC Administrator Application
odbccp32.cp_	..\system32\odbccp32.cpl	32-bit ODBC Installer Control Panel Tool
odbccp32.dl_	..\system32\odbccp32.dll	32-bit ODBC Control Panel Installer DLL
odbccr32.dl_	..\system32\odbccr32.dll	32-bit cursor library
odbcinst.hl_	..\system32\odbcinst.hlp	Installation help file
odbcIn32.dl_	..\system32\odbcIn32.dll	32-bit ODBCLink/SE Driver
odbcls32.dll	..\system32\odbcls32.dll	32-bit ODBCLink/SE Driver Setup DLL
r8ansi32.dl_	..\system32\r8ansi32.dll	32-bit character translation DLL for translation between the ANSI and roman-8 character set.

r9ansi32.dl_	..\system32\r9ansi32.dll	32-bit character translation DLL for translation between the ANSI and roman-9 character set.
setup.exe	not installed to hard drive	Microsoft ODBC Setup program drive
setup.lst	not installed to hard drive	
trndel32.exe	...\system32\trndel32.exe	Translator uninstall utility to “de-register” translator information
what32.ex_	...\system32\what32.exe	Utility to determine what versions of drivers required by ODBC application are currently loaded on the PC

MPE/iX Host

ODBCJOB.ODBCSE.SYS	sample jobstream to run the listener process
ODBCLNSE.ODBCSE.SYS	the listener program
ODBCXLSE.ODBCSE.SYS	an XL library containing common code
ODBCUTSE.ODBCSE.SYS	utility for technical support purposes
ODBCCL16.ODBCSE.SYS	a self-extracting archive for the 16bit client software
ODBCCL32.ODBCSE.SYS	a self-extracting archive for the 32bit client software
ODBCLOG.ODBCSE.SYS	log file (when created).

HP-UX 9.x Host

/usr/bin/odbcse/odbclnse	the listener program
/usr/bin/odbcse/odbcutse	utility for technical support purposes
/usr/bin/odbcse/odbccl16	a self-extracting archive of the 16bit client software
/usr/bin/odbcse/odbccl32	a self-extracting archive of the 32bit client software

HP-UX 10.x Host

/opt/allbase/bin/odbcse/odbclnse	the listener program
/opt/allbase/bin/odbcse/odbcutse	utility for technical support purposes
/opt/allbase/bin/odbcse/odbccl16	a self-extracting archive of the 16bit client software
/opt/allbase/bin/odbcse/odbccl32	a self-extracting archive of the 32bit client software
/tmp/odbclog	log file (when created).

Index

A

ALLBASE	7 - 13
ALLBASE / IMAGE Data Types	5 - 8
ANSI Character Set	A - 2
Application Development	
Isolation Levels	5 - 7
Supported Functions	5 - 5

B

BLOBs	1 - 1, 5 - 9
Note on Using	5 - 9

C

Calculated Columns	6 - 5
Calling ODBC Functions Programmatically	6 - 18
Changing the display format	6 - 4
Client Configuration	4 - 1
ODBCADM	4 - 1
Client Environment	1 - 2
Configuring Data Sources	
Choosing an ODBC Drive	4 - 2
ODBC Translator	4 - 7
ODBCLink/SE Setup	4 - 3
Setup for HP-UX	4 - 5
Setup for MPE/iX	4 - 4
Setup Options	4 - 6
Setup Translator	4 - 7
Successful Completion	4 - 7
Windows 3.11	4 - 1
Windows NT V3.51	4 - 1
Windows95,	4 - 1
WindowsNT 4.0	4 - 1
Connect Using SQL API	6 - 13
Connect Using SQLAPI in VB 2.0-6.0	6 - 13
Connecting Link	1 - 2
Connecting with DAO's	6 - 16
Connecting with RDO's	6 - 17
Creating a DBEnvironment	
HP3000 Database Server	B - 1
HP9000 Database Server	B - 3
Creating Joins Using Calculated Columns	6 - 5

D

Data Source Configuration	4 - 2
Deleting a Translation DLL	7 - 5
Deleting drivers.	7 - 4
Downloading ODBCLink/SE to the PC	3 - 1
Using ARPA Services (FTP)	3 - 2
Using NS Services (DSCOPY) on the HP3000	3 - 2
Using Reflection® software	3 - 1
Dr. DeeBee Spy	7 - 10
Drivers, deleting	7 - 4

E

EURO	4 - 7, A - 3
European Currency Symbo.	14 - 7, A - 3
Extracting the ODBCLink/SE file	3 - 3

G

Gather Information	2 - 4
GlancePlus/UX	7 - 7
GlancePlus/XL	7 - 7

H

Host Logging	7 - 9
--------------------	-------

I

Implementation Notes	A - 1
ALLBASE/SQL and IMAGE/SQL Restrictions on the ODBC Grammar	A - 2
ALLBASE/SQL and IMAGE/SQL Statements and Embedded SQL	A - 4
ALLBASE/SQL and IMAGE/SQL Statment Functions on Client	A - 4
ALLBASE/SQL LongVarBinary items	A - 1
Maximum number of statements	A - 1
Primary key name returned by SQLStatistics.	A - 1
Unsupported ALLBASE/SQL and IMAGESQL Statements	A - 3
Using the ANSI Character Set	A - 2
Installation	3 - 1
ARPA Services (FTP).	3 - 2
Extracting the ODBCLink/SE file	3 - 3
Gather Information	2 - 4
NS Services (DSCOPY).	3 - 2
Set the Connection Assurance Parameters	2 - 1

Starting and Stopping the Listener	2 - 2
Installed Files	C - 1
For Windows 3.1 or Windows 3.11	C - 1
For Windows 95	C - 3
For Windows NT	C - 5
HP-UX 10.x Host	C - 7
HP-UX 9.x Host	C - 7
MPE/iX Host	C - 6

J

Joins	6 - 4
Joins - Creating	6 - 5

L

List of Installed Files	C - 1
16 Bit Driver Client	C - 1
32 Bit Driver Client - Windows NT	C - 5
32 Bit Driver Client - Windows95	C - 3
HP-UX 10.x Host	C - 7
HP-UX 9.x Host	C - 7
MPE/iX Host	C - 6, C - 7
Listener - Starting and Stopping	
Stopping the listener	2 - 2
Listener Log File - HP3000	7 - 7
Listener Log File - HP9000	7 - 7

M

Migrating Data Sources	4 - 8
PCAPI	4 - 8

N

Notes for Users of MS-Access	6 - 9
------------------------------------	-------

O

ODBC Compliancy	1 - 2
ODBC Conformance level	1 - 2
ODBC Data Type	5 - 8
ODBCAD32.EXE	1 - 3
ODBCADM.EXE	1 - 3
ODBCLinkSE	

Client Requirements	1 - 4
Compliance	1 - 2
Field types supported	1 - 5
Operating Systems	1 - 4
Overview	1 - 1
Software provided	1 - 4
Specification	s 1 - 4
SQL Commands supported	1 - 5
ODBCLinkSE Host	7 - 13
ODBCLinkSE.DLL	7 - 13
ODBCLS16.DLL	1 - 3
ODBCLS32.DLL	1 - 3
ODBCUTIL	7 - 7

P

PC Client Tools	7 - 10
PCAPI	4 - 8
Migrating data sources	4 - 8
Performance Considerations	5 - 3
Preparing the Database Server	2 - 1
Process Status (ps)	7 - 7

R

Running the SETUP program	3 - 3
---------------------------------	-------

S

Set the Connection Assurance Parameters	2 - 1
Setup	
HP-UX	4 - 5
MPE/iX	4 - 4
ODBC Translator	4 - 7
ODBCLink/SE	4 - 3
Options	4 - 6
Translator	4 - 7
Special Notes for Users of Impromptu	6 - 3
SQLColAttributes	5 - 5
SQLExtendedFetch	1 - 2
SQLGetFunctions	5 - 5
SQLGetInfo	5 - 6
SQLMON	7 - 7
SQLSetConnectOptions	5 - 6
SQLSetStmtOptions	5 - 6

Starting and Stopping the Listener 2 - 2
 Start the listener - HP3000 2 - 2
 Start the listener HP9000 2 - 2
 Stopping the listener 2 - 3
 Stored Procedures - Using 5 - 2
 Supported Functions 5 - 5

T

Terminating an ODBC Connection 7 - 1
 Tools on Database Server
 Checking Listener Log File - HP3000 7 - 7
 Checking Listener Log File - HP9000 7 - 7
 Using GlancePlus/UX 7 - 7
 Using GlancePlus/XL 7 - 7
 Using Process Status (ps) Command 7 - 7
 Using SQLMON 7 - 7
 Tools on the PC Client
 Client logging 7 - 10
 ODBC Tools 7 - 10
 Translation DLL
 Deleting 7 - 5
 Translator - ODBC 4 - 7
 Translator - Roman8 to PC-ANSI 4 - 7
 Translator - Roman9 to PC-ANS I 4 - 7, A - 3
 Troubleshooting
 Checking Listener Log File
 HP3000 7 - 7
 HP9000 7 - 7
 Using GlancePlus/UX 7 - 7
 Using GlancePlus/XL 7 - 7
 Using Microsoft ODBC Test (32-bit) 7 - 2
 Using Process Status (ps) Command 7 - 7
 Using SQLMON 7 - 7
 Using the Host Testing Utility ODBCUTSE 7 - 7

U

Updating Data 6 - 1
 Using Cognos Impromptu 6 - 2
 Using Microsoft Access 6 - 7
 Using MS-Query 6 - 10
 Using ODBCLink/SE With ODBC Applications 6 - 1
 Using Visual Basic 6 - 12

V

Verify Software Version Numbers	2 - 1, 7 - 3
Verifying the client-side components	7 - 3
Viewing the SQL generated by Impromptu	6 - 3

W

Winsock	
WINSOCK.DLL	7 - 13