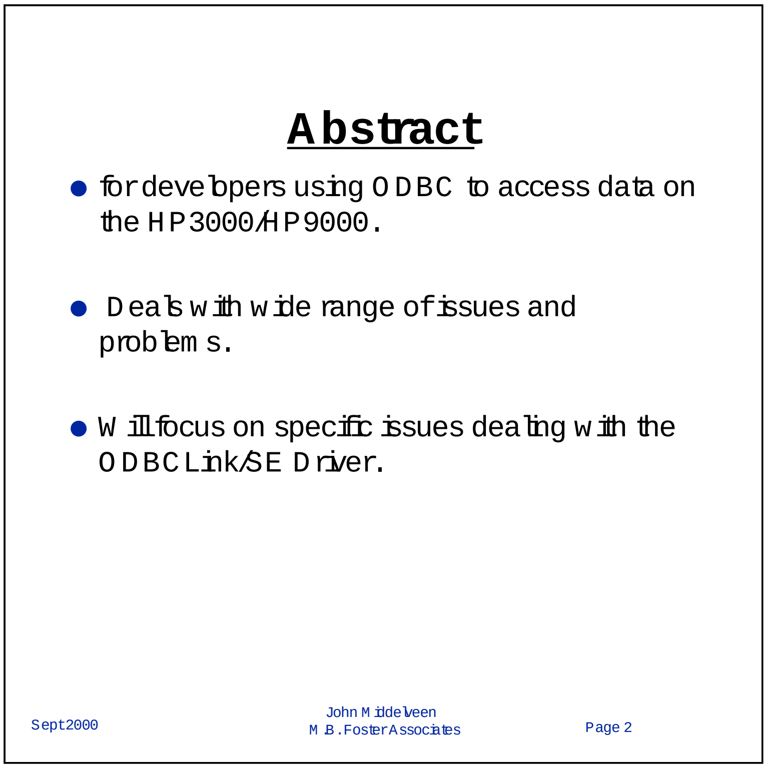
ODBC Power User Seminar

- Presentation # 263
- Presented by: John Middelveen
  - johnm@mbfoster.com

- from:
- M.B. Foster Associates
  - 82 Main St South
  - Chesterville ONT
  - K0C-1H0
  - (613) 448-2333
  - www.mbfoster.com

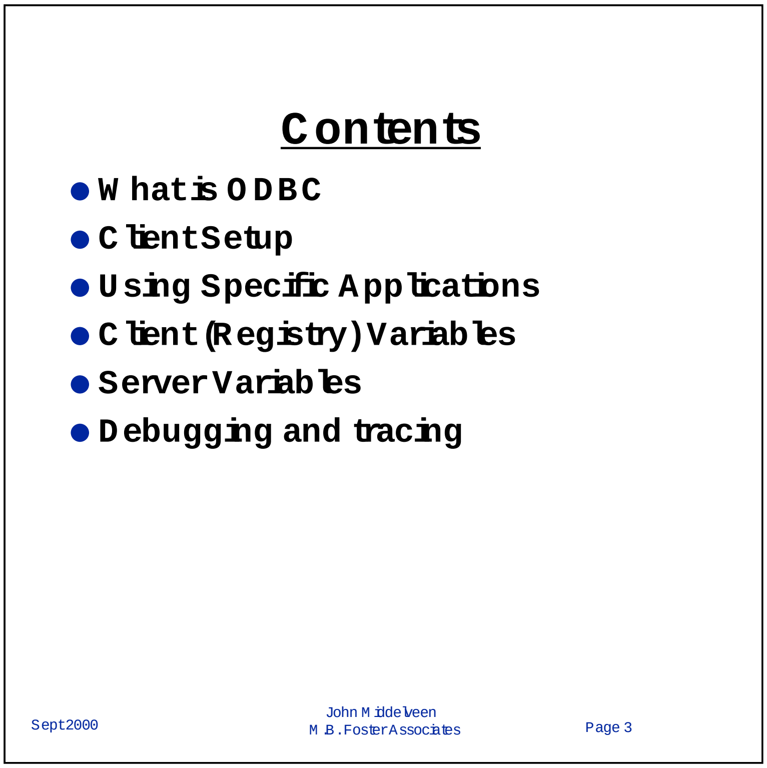


Abstract

- for developers using ODBC to access data on the HP3000/HP9000.
- Deals with wide range of issues and problems.
- Will focus on specific issues dealing with the ODBCLink/SE Driver.

This ODBC Seminar is intended for Intermediate users or developers of Client Server Applications using ODBC to gain access to data on the HP3000 / HP9000. But will cover many topics rancing from beginner to Adavnced levels.

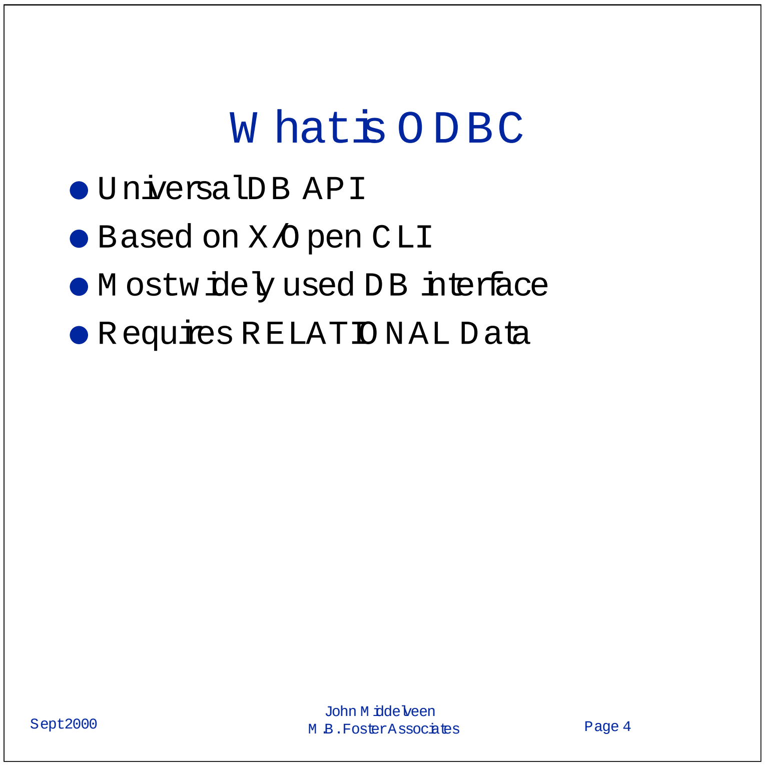
The Information will primarily deal with the how to use the ODBCLink / SE Drivers with various applications. In many cases it will focus in on specific implementation issues dealing with these drivers.



Contents

- What is ODBC
- Client Setup
- Using Specific Applications
- Client (Registry) Variables
- Server Variables
- Debugging and tracing

Notes



What is ODBC

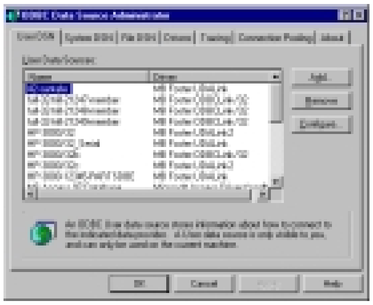
- Universal DB API
- Based on X/Open CLI
- Most widely used DB interface
- Requires RELATIONAL Data

Open Database Connectivity (ODBC) provides a Universal Database Connectivity application programming interface (API) that enables applications to access data in a wide range of proprietary databases. Based on the X/Open SQL Access Group's Call Level Interface (CLI) specification, ODBC is an open, **vendor-neutral** way to uniformly access data stored in different formats and database engines.

ODBC is the most widely used interface to relational data. It's also quite fast, but you pay for the fast access with complex application code. Fortunately much of this complexity can now be hidden to the programmer by the use of various Object Libraries.

As the standard interface to relational data, your application can access a lot of data using ODBC. But ODBC does require that your data look like a relational database, so it's not always the best way to expose data. If you don't have a relational database, it can be very difficult to write an ODBC driver to expose your data because you basically have to write a relational engine on top of the existing data structure.

Client Setup



The first program that anyone dealing with any ODBC Driver must be familiar with is the ODBC Administrator. For Windows platforms, this is provided by Microsoft. Its main function is to allow users to configure data-sources for any ODBC driver that exists on the system.

Normally this program is invoked from the ODBC Administrator (ODBC Data Sources) icon in the Control Panel.

For Windows 2000 there is a subdirectory for all the Administrative Tools, and the ODBC Administrator is involked by clicking on the “DataSources(ODBC)” icon

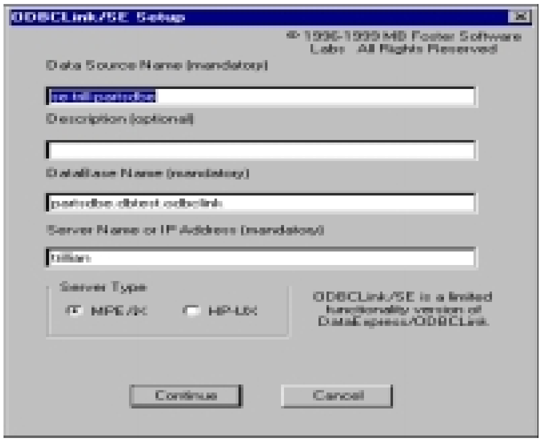
The ODBC Administrator program allows you to configure data-sources, turn logging on and off; and for ODBC3.0 versions allows you to turn connection pooling on. You can also find out about what versions of ODBC drivers are installed on your PC.

To setup an ODBCLink/SE DSN you click on the Add button of the ODBC Administrator.

Then you select the driver that you wish to setup a DSN for; in this case the ODBCLink/SE-32 Driver.

Next click <Finish>.

ODBCLink/SE DSN Setup



ODBCLink/SE Setup

© 1996-1999 MB Foster Software Labs All Rights Reserved

Data Source Name (mandatory): se.tillpartsdbe

Description (optional):

DataBase Name (mandatory): partsdbe.dbtest.odbclink

Server Name or IP Address (mandatory): tillian

Server Type

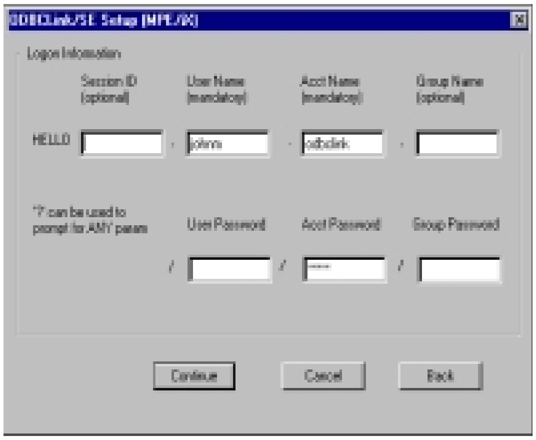
● MPE/iX ○ HP-UX

ODBCLink/SE is a limited functionality version of DataExpress/ODBCLink.

Enter the DSN Name, DBE name (including path or group/account), Server Name (or IP address) and select the proper Server Type;

Then Click <Continue>

SE Setup (Login)



ODBCLink/SE Setup (MPE/iX)

Logon Information

	Session ID (optional)		User Name (mandatory)		Acct Name (mandatory)		Group Name (optional)
HELLO		,	johnm	.	odbclink	.	

"?" can be used to prompt for ANY param

	User Password		Acct Password		Group Password
/		/	*****	/	

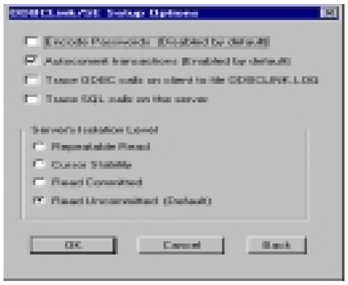
A MPE or UNIX login screen will appear. (depending on the OS type as setup in the previous screen)

Enter the appropriate Info:

You will need to login as a user who has access to whatever DBE you specified in the previous screen:

Press <Continue>

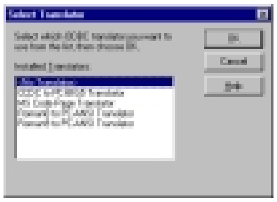
SE Setup (Misc & Translator Info)



ODBCLink/SE Setup Options

☐ Encode Passwords (Enabled by default)
☑ Autocommit transactions (Enabled by default)
☐ Trace ODBC calls on client to file ODBCLINK.LOG
☐ Trace SQL calls on the server

Server's Isolation Level
○ Repeatable Read
○ Cursor Stability
○ Read Committed
● Read Uncommitted (Default)



Select Translator

Select which ODBC translator you want to use from the list, then choose OK.

Installed Translators:

- <No Translator>
- ODBC to PC BIG5 Translator
- MS Code Page Translator
- Roman8 to PC ANSI Translator
- Roman9 to PC ANSI Translator

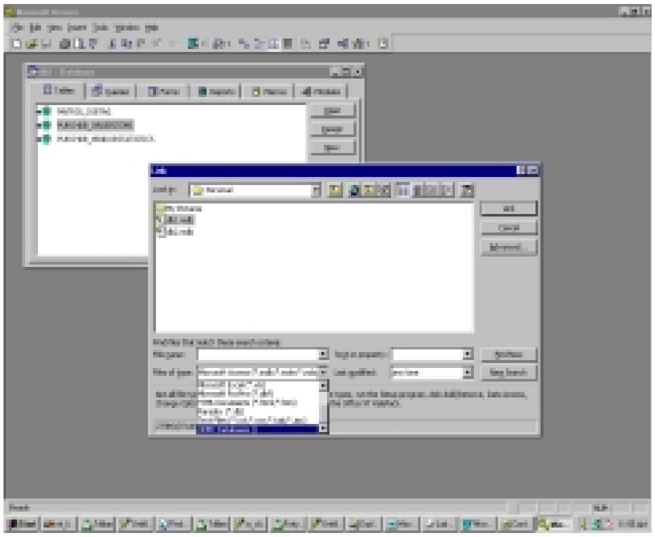
After the Login Info a window will open allowing you to add Transaction Isolation Auto-Commit and Allow you to turn Debug ON or OFF

The final screen is the Translator setup screen which allows you to select any of the translators that have been setup on your PC.

The SE Driver comes with the BIG5 translator and the r8 to PC ANSI translator.

The Roman9 to PC-ANSI is identical except that it has the EURO symbol added in place of the sputnic. (As of 5.57.08)

Connecting with MS Access



After starting MS Access, you must either select an existing DBE or create a new one.

Next….

From the main MS Access Window, select

Get External Data, -> Link Tables, (or Import if you simply wish to Import a HP3000 table)

In the ‘Files of Type’ Dropdown box select ODBC Databases (the Last Option)

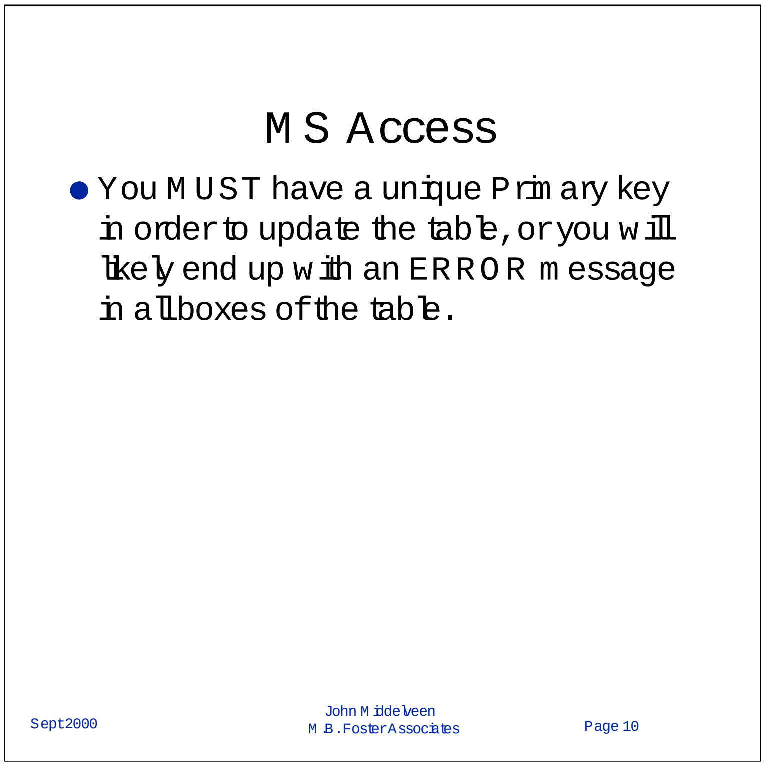
A Window will appear asking you to pick one of the configured data-sources.

Select the DSN that you created earlier.

After MS Access connects with the data source, a window will appear showing what tables are available to connect to.

When you have selected one or more tables hit <OK>

If there is no primary key detected on the table MS Access will prompt you to select one or more columns in the table which form a unique key.

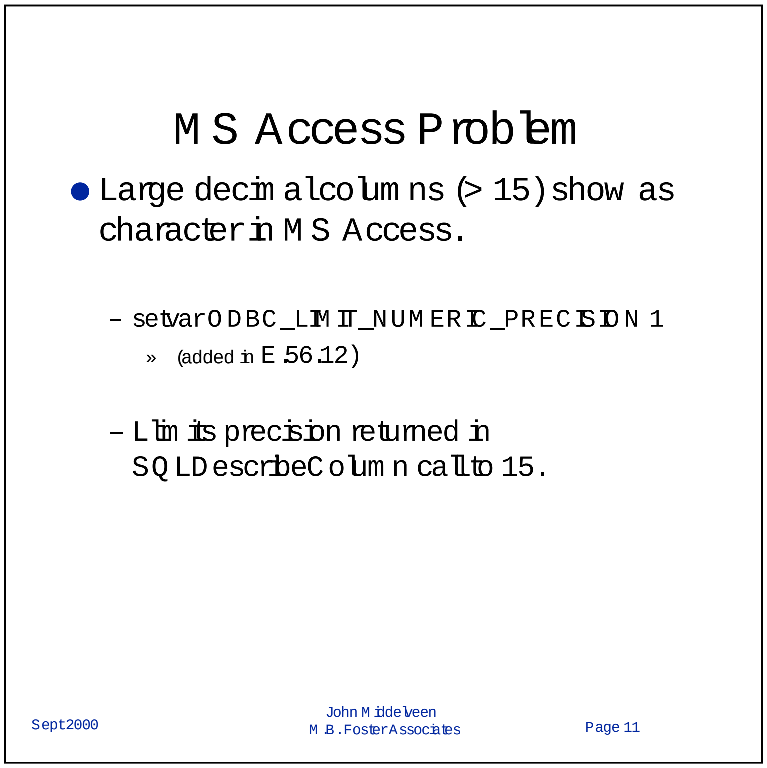


MS Access

- You MUST have a unique Primary key in order to update the table, or you will likely end up with an ERROR message in all boxes of the table.

You can select a combination of Fields that you are sure will create a unique key.

If you intend only to read the table, (or you don’t know) click cancel to not select any key, in which case MS Access will create a non - updateable table.

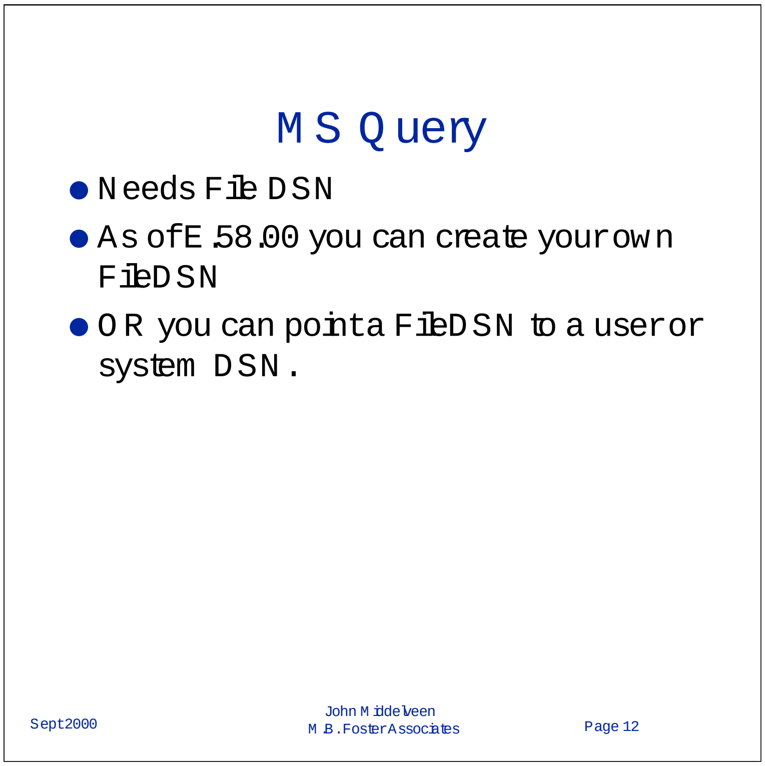


MS Access Problem

- Large decimal columns (> 15) show as character in MS Access.
  - set var ODBC_LIMIT_NUMERIC_PRECISION 1
    - » (added in E.56.12)
  - Limits precision returned in SQLDescribeColumn call to 15.

Notes:

Modifying of Server Parameters will be covered in a later section.

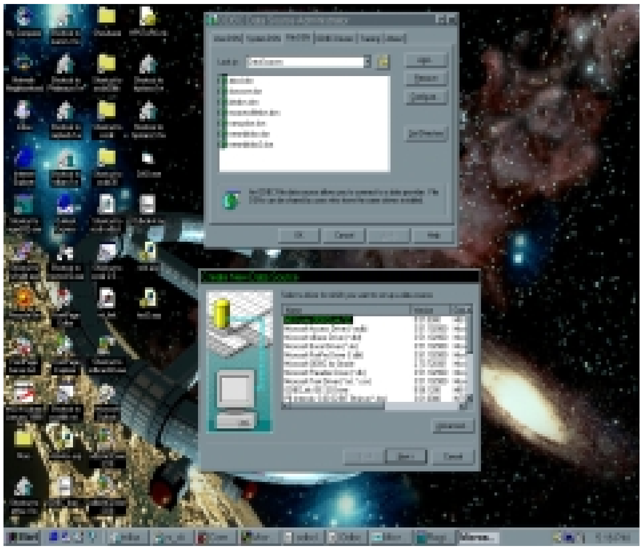


MS Query

- Needs File DSN
- As of E.58.00 you can create your own File DSN
- OR you can point a File DSN to a user or system DSN.

Notes:

Pointing FileDSN to registry DSN

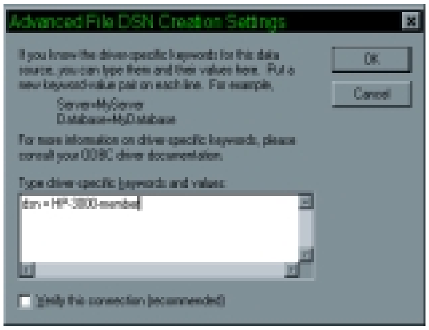


Under FileDSN Tab click Add

A window will popp up showing the available drivers to chose from
Select the Appropriate Driver Name.
(ODBCLINK/SE-32 Driver)

Click <Advanced>

Advanced File DSN Settings



Advanced File DSN Creation Settings

If you know the driver-specific keywords for this data source, you can type them and their values here. Put a new keyword-value pair on each line. For example,

Server=MyServer
Database=MyDatabase

For more information on driver-specific keywords, please consult your ODBC driver documentation.

Type driver-specific keywords and values:

```
dsn = HP-3000-member
```

□ Verify this connection (recommended)

- Add a DSN = <DSN_name> with space before =

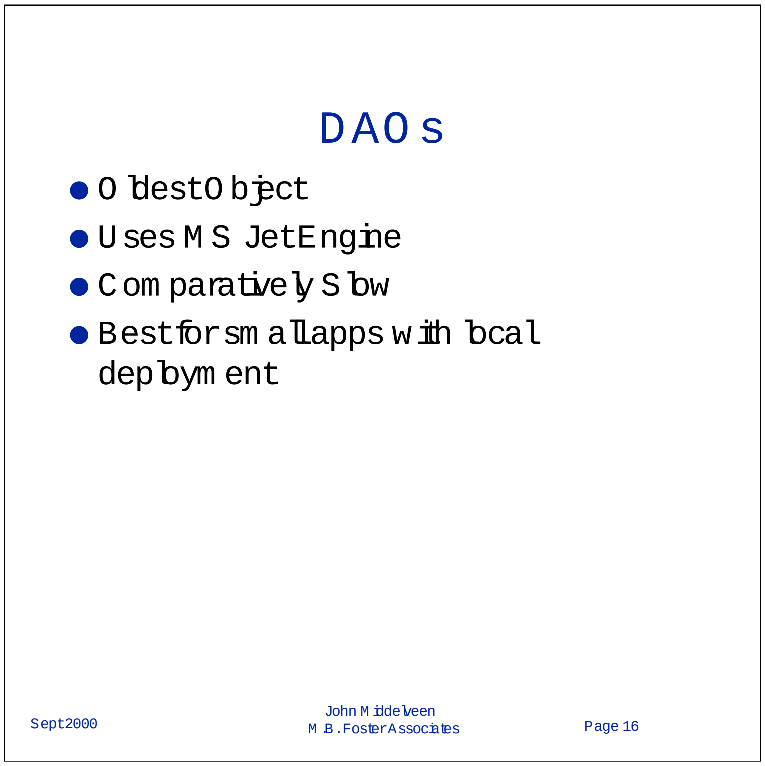
When the Advanced DSN Settings Window Opens

add a line that says DSN = <System DSN>
Where <System DSN> is a DSN that you have already setup

Note: You will likely need to put a space before and after the ‘ = ‘ sign (depending on what ODBC Administrator version is installed)

VB and Web Access

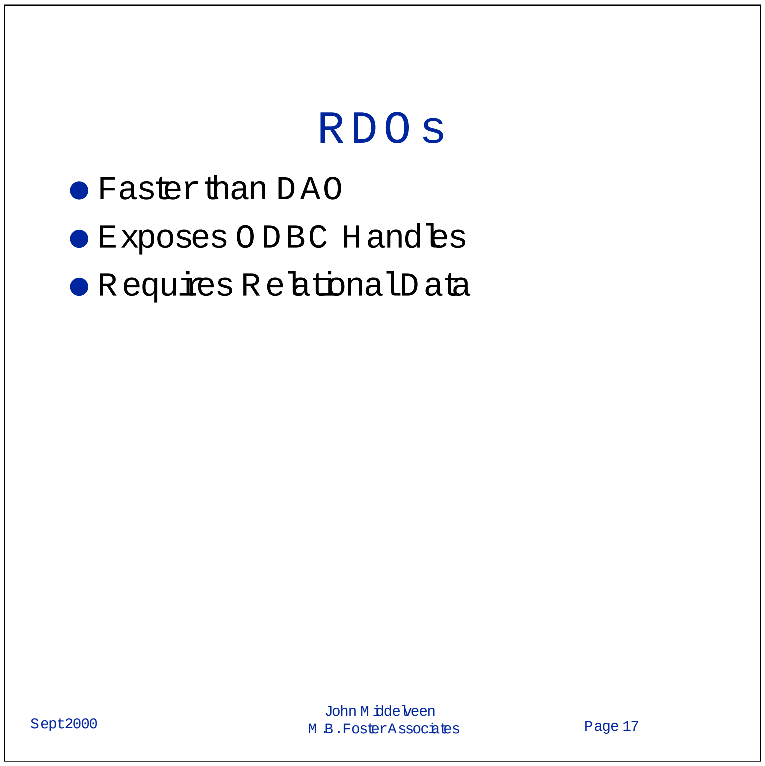
- DAO - Data Access Object
- RDO - Remote Data Object
- ADO - ActiveX Data Object



DAOs

- Oldest Object
- Uses MS Jet Engine
- Comparatively Slow
- Best for small apps with local deployment

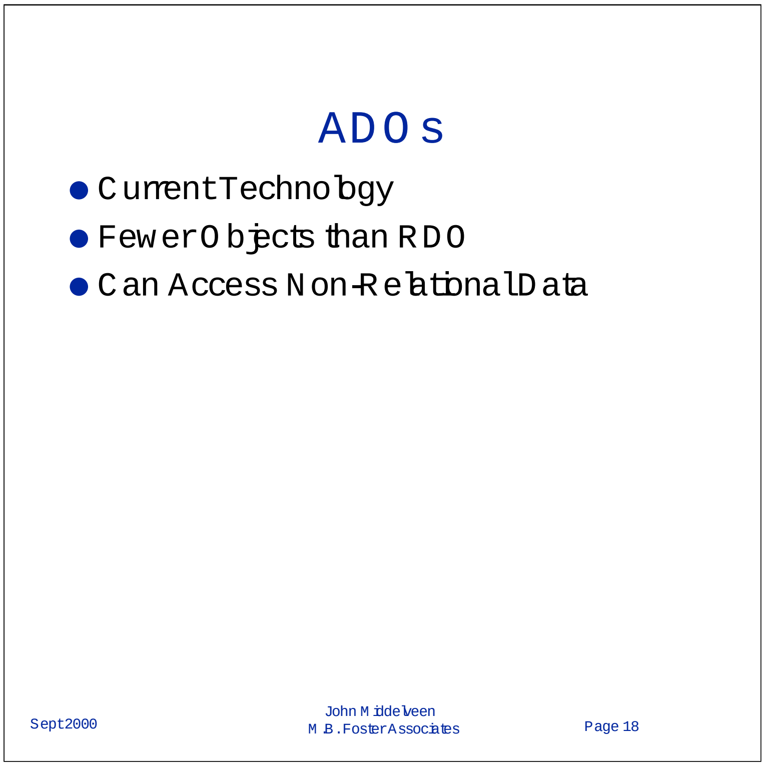
(Data Access Objects) was the first object-oriented interface that exposed the Microsoft Jet database engine (used by Microsoft Access) and allowed Visual Basic developers to directly connect to Access tables - as well as other databases - through ODBC. DAO is suited best for either single-system applications or for small, local deployments.



RDOs

- Faster than DAO
- Exposes ODBC Handles
- Requires Relational Data

(Remote Data Objects) is an object-oriented data access interface to ODBC combined with the easy-to-use style of DAO, providing an interface that exposes virtually all of ODBC's low-level power and flexibility. RDO is limited, though, in that it doesn't access Jet or ISAM databases very well, and that it can access relational databases only through existing ODBC drivers. However, RDO has proven to be the interface of choice for a large number of SQL Server, Oracle, and other large relational database developers. RDO provides the objects, properties, and methods needed to access the more complex aspects of stored procedures and complex result sets.



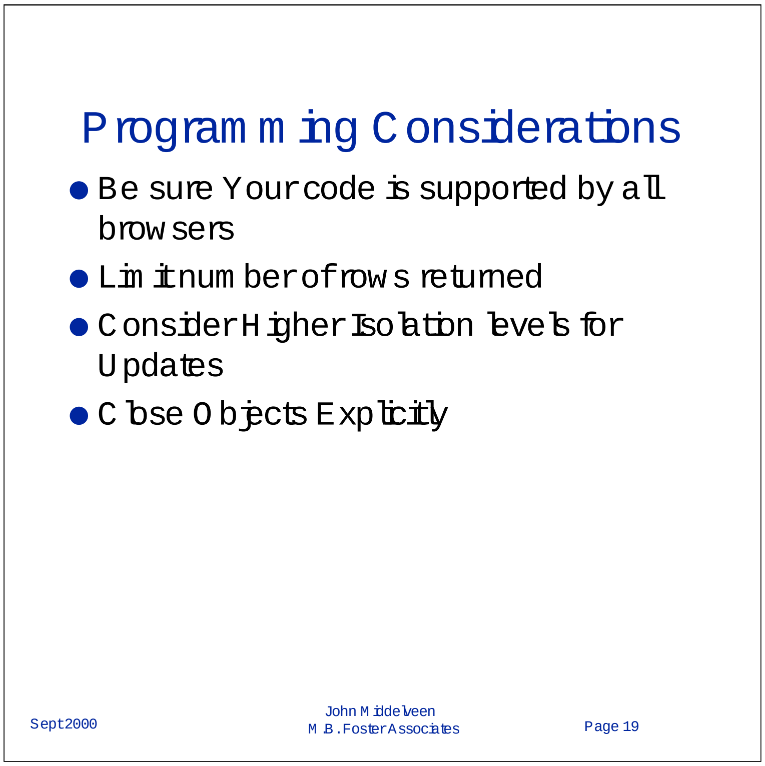
ADOs

- Current Technology
- Fewer Objects than RDO
- Can Access Non-Relational Data

(Active-X Data Object) is the successor to DAO/RDO. Functionally ADO 2.0 is most similar to RDO, and there's generally a similar mapping between the two models. ADO "flattens" the object model, containing fewer objects and more properties, methods (and arguments), and events. For example, ADO has no equivalents to the rdoEngine and rdoEnvironment objects, which exposed the ODBC driver manager and hEnv interfaces. Nor can you currently create ODBC data sources from ADO, despite the fact that your interface might be through the ODBC OLE DB service provider.

Much of the functionality contained in the DAO and RDO models was consolidated into single objects, making for a much simpler object model. Because of this, however, you might initially find it difficult to find the appropriate ADO object, collection, property, method, or event. Unlike DAO and RDO, although ADO objects are hierarchical, they can also be created outside the scope of the hierarchy.

It should be noted that ADO currently doesn't support all of DAO's functionality. ADO mostly includes RDO-style functionality to interact with OLE DB data sources, plus remoting and DHTML technology. ADO doesn't currently support data definition (DDL), users, groups, and so forth.



Programming Considerations

- Be sure Your code is supported by all browsers
- Limit number of rows returned
- Consider Higher Isolation levels for Updates
- Close Objects Explicitly

Sept 2000

John Middelveen
M.B. Foster Associates

Programming Considerations

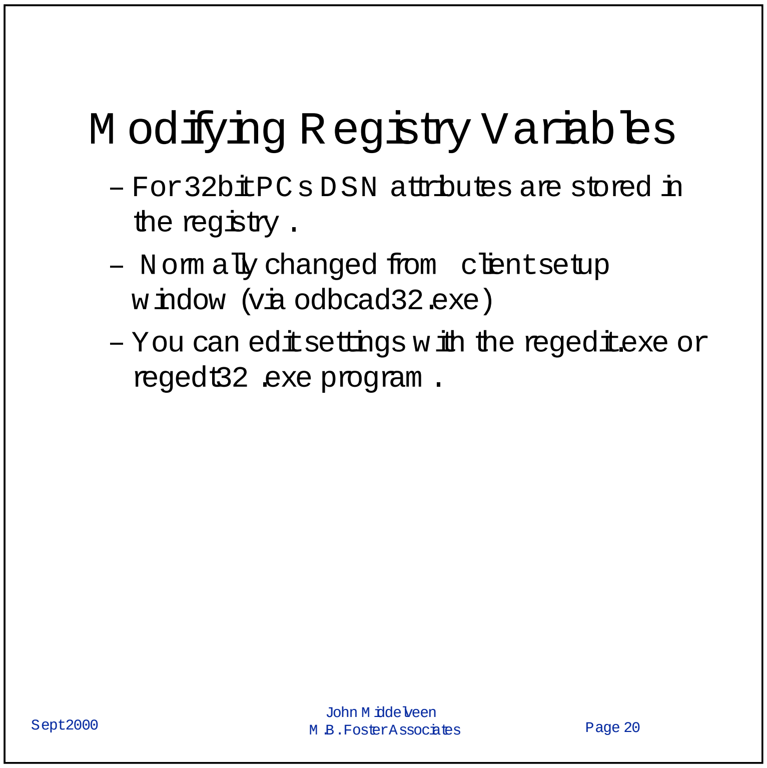
When designing an application in which uses ODBC to gain access to data on the server there are a number of things to be considered BEFORE you start coding your application.

-You can use the MSFlexGrid in an ASP but not all browsers support it.

-If you are connecting to a db that has thousands of rows you should code your select statements or set the statement options to limit the amount of the rows that will come back. If you can reduce the number of rows returned your application will run faster.

-If you are creating an application that could have many users updating data you will want a higher isolation level than if you are simply reading data.

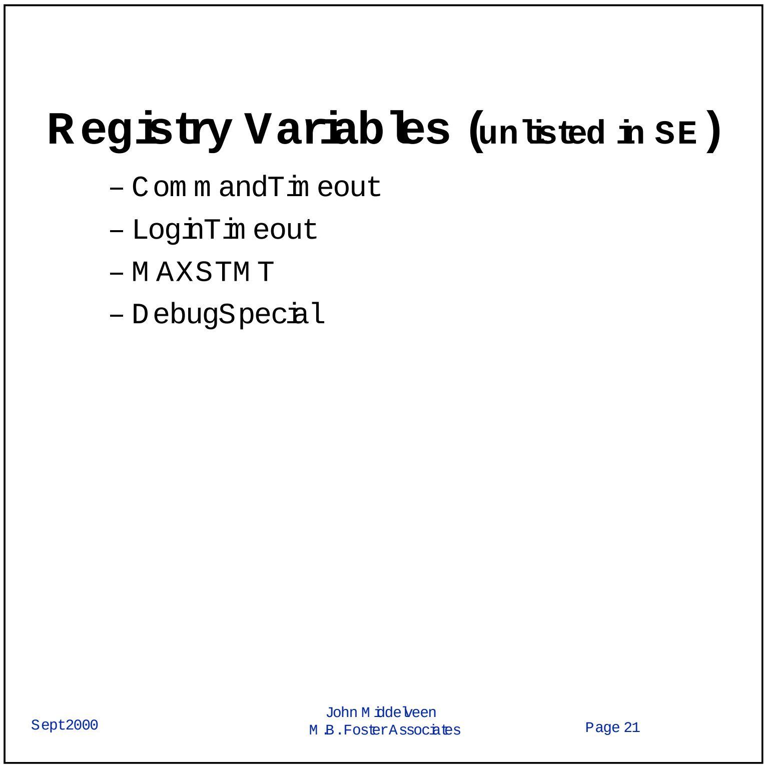
Explicitly close all your objects (with the .Close Method), rather than assuming they will be closed when the form or Application Terminates. This will reduce the chance of memory leaks



Modifying Registry Variables

- For 32bit PCs DSN attributes are stored in the registry.
- Normally changed from client setup window (via odbcad32.exe)
- You can edit settings with the regedit.exe or regedt32.exe program.

Notes:



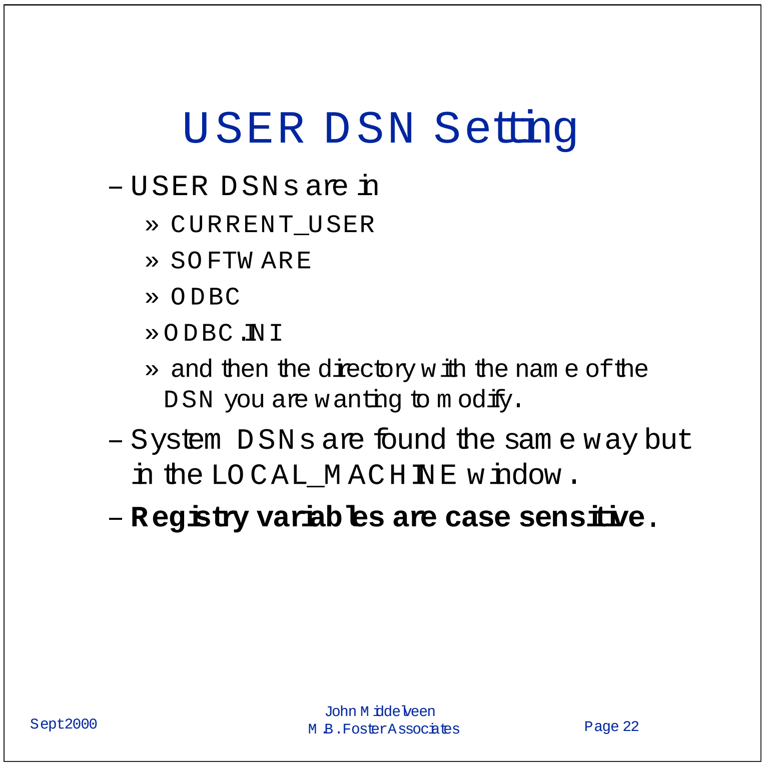
Registry Variables (unlisted in SE)

- CommandTimeout
- LoginTimeout
- MAXSTMT
- DebugSpecial

Usually you will not need to change any of these parameters.

These show up in Registry for the full version of the driver(as well as many other settings) but not the SE version

If you do want to modify the parameters you will need to add them MANUALLY to the registry.



USER DSN Setting

- USER DSNs are in
  - » CURRENT_USER
  - » SOFTWARE
  - » ODBC
  - » ODBC.INI
  - » and then the directory with the name of the DSN you are wanting to modify.
- System DSNs are found the same way but in the LOCAL_MACHINE window.
- **Registry variables are case sensitive.**

As you may know there are 3 types of DSNs

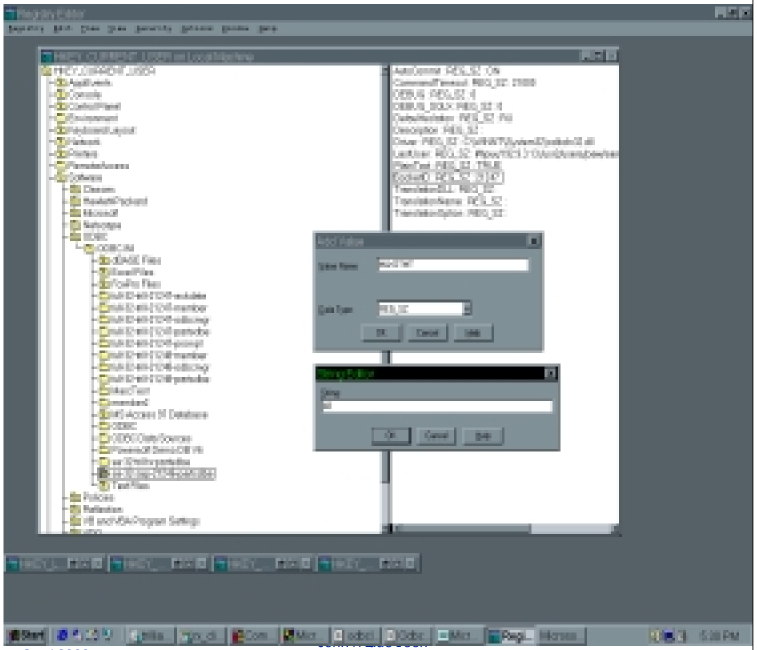
User, System and File DSN

The User and System DSN Info is stored in the registry

File DSNs info is stored in small text files with ‘.dsn’ extensions.
These could be anywhere (depending on how things are set up)

There are also ‘DSNless’ connections where all the connect information is entered in the Connection string

Editing the Registry Variables

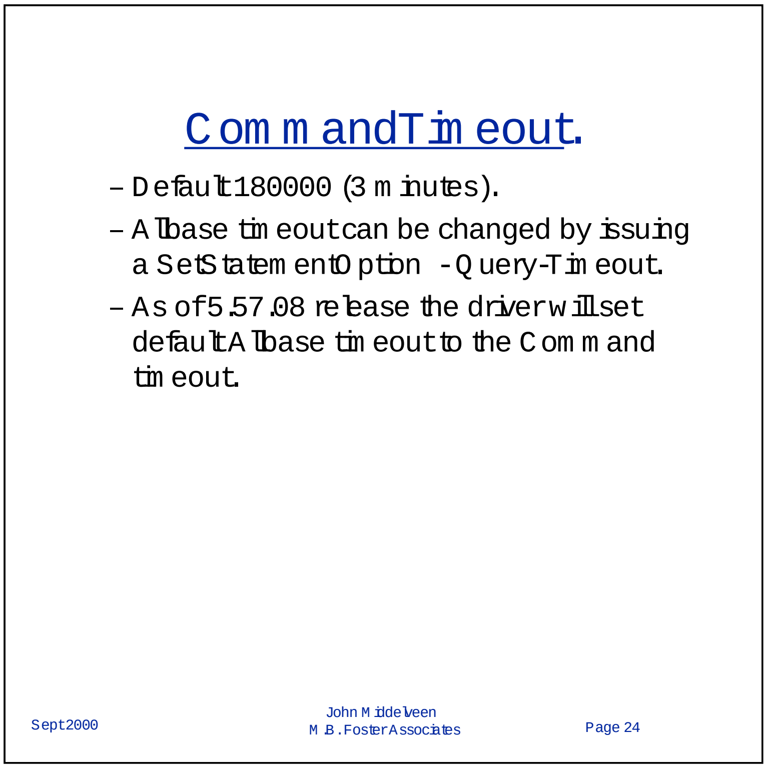


All Variables are string (REG_SZ)

Here is a Shot of what it looks like using Regedit to Add a parameter to the Registry.

Depending of what version of the Administrator you are using, once you add the new variable it may prompt you for a value or you may have to double-click on the parameter to open an edit box.

REMEMBER THAT THESE PARAMETERS ARE CASE SENSITIVE!



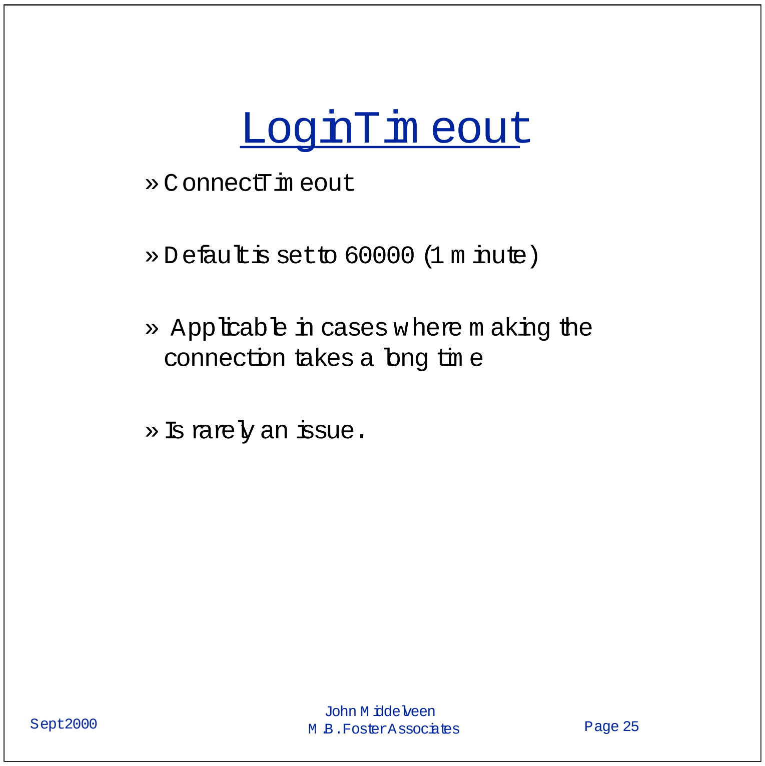
Command Timeout.

- Default 180000 (3 minutes).
- Allbase timeout can be changed by issuing a SetStatementOption -Query-Timeout.
- As of 5.57.08 release the driver will set default Allbase timeout to the Command timeout.

Notes:

In older versions (pre 5.57.08) this does NOT change ALLBASE timeout.

ALLBASE also has a MAXIMUM timeout setting (user settable). If this timeout is higher than theALLBASE MAX then ALLBASE will just use the max (and you will see an error in the Server log if Logging is ON)



LoginTimeout

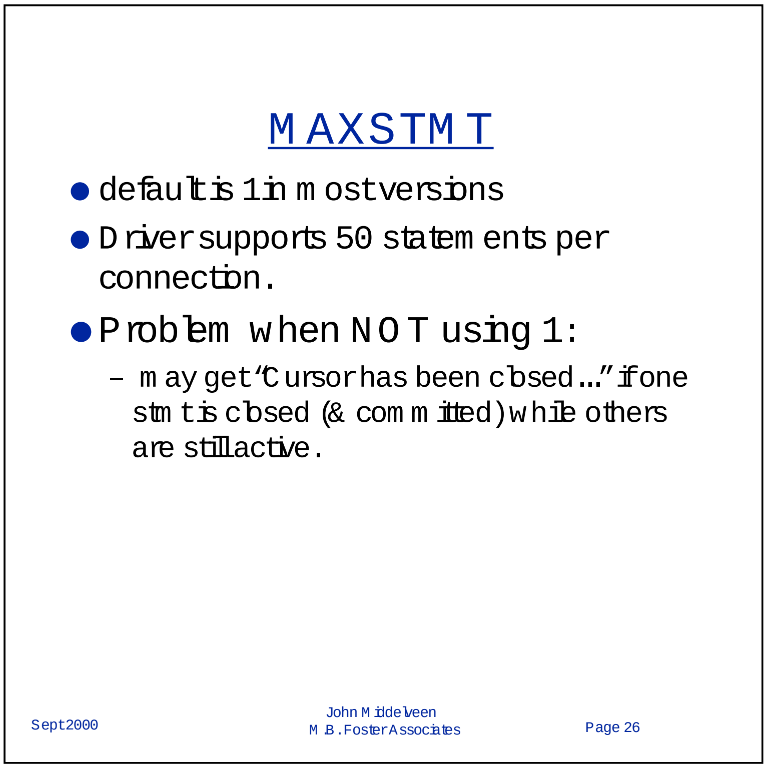
» ConnectTimeout

» Default is set to 60000 (1 minute)

» Applicable in cases where making the connection takes a long time

» Is rarely an issue.

Sometines this is usefull if you have a LOT of Inage / SQL tables attached to the DBE. In some Allbase versions these take a long time to get the info for and so the connection takes a long time.



MAXSTMT

- default is 1 in most versions
- Driver supports 50 statements per connection.
- Problem when NOT using 1:
  - may get “Cursor has been closed…” if one stmt is closed (& committed) while others are still active.

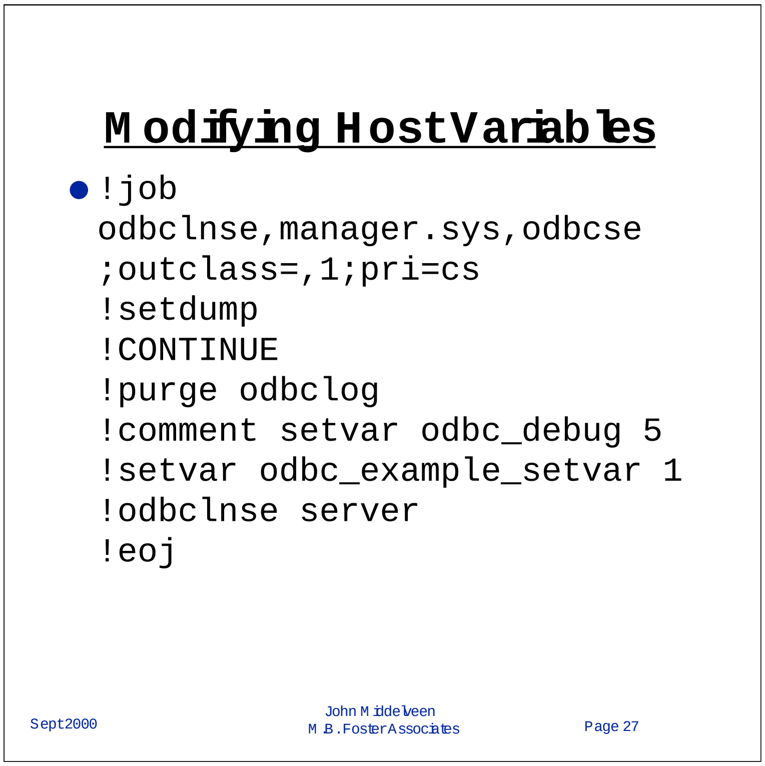
Notes:

Although Setting the MAXSTMT to 1 cures problems associated with some Applications when accessing Allbase, there are some disadvantages to doing this.

Disadvantages:

- There is more overhead required

- You will reach your max ALLBASE concurrent transaction limit sooner.

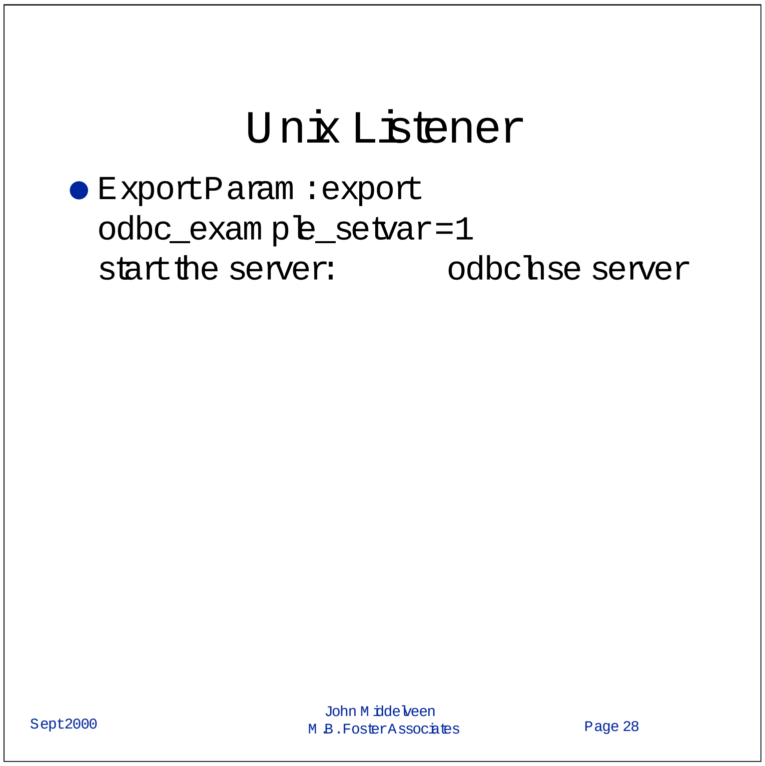


Modifying Host Variables

```
!job
odbclnse,manager.sys,odbcse
;outclass=,1;pri=cs
!setdump
!CONTINUE
!purge odbclog
!comment setvar odbc_debug 5
!setvar odbc_example_setvar 1
!odbclnse server
!eoj
```

As you can see in the example putting a ‘comment’ in front of the setvar will comment the line out.

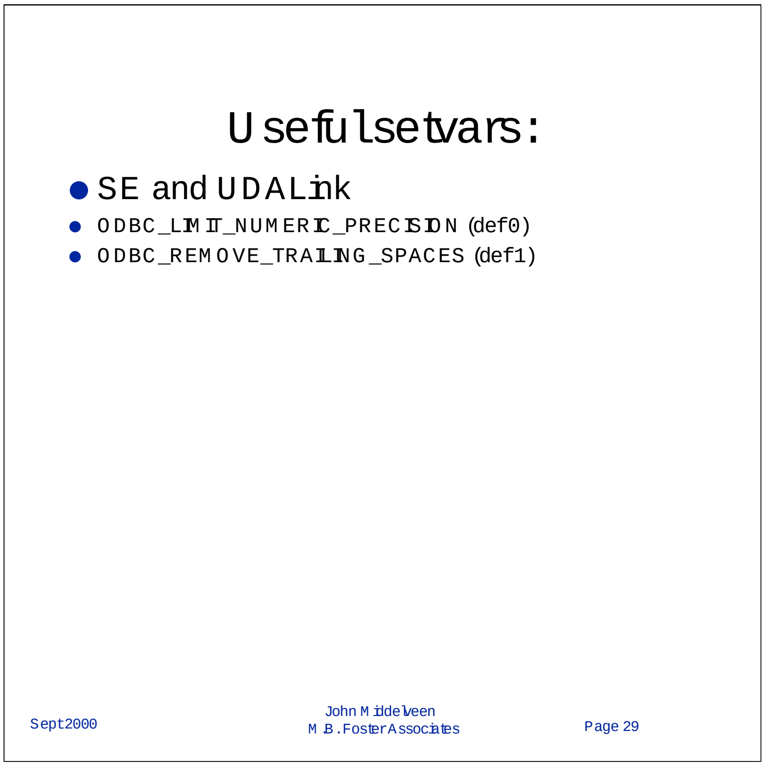
If you have a number of setvars (especially when using the full driver) it is useful to make a template with the more commonly used setvar in it, and just remove the comment as needed.



Unix Listener

- ExportParam :export odbc_example_setvar=1
  start the server: odbclnse server

The UNIX listener is started at the command line rather than streamed. You can make a script to automatically start the server and put any environment settings in the script file. In this way you only need to remember the name of the script file, and not each of the setvars you want



Useful setvars:

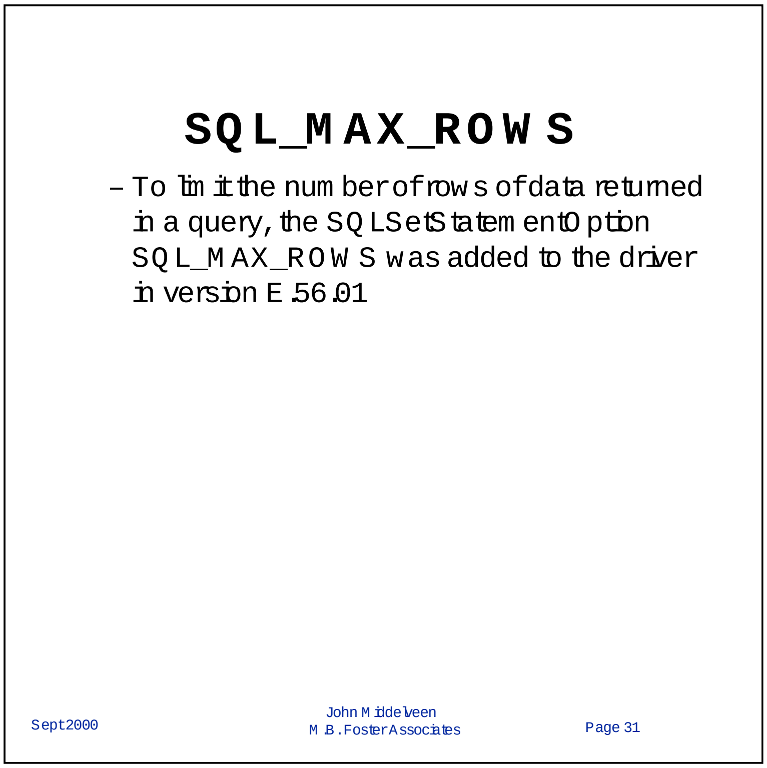
- SE and UDALink
  - ODBC_LIMIT_NUMERIC_PRECISION (def 0)
  - ODBC_REMOVE_TRAILING_SPACES (def 1)

We mentioned ODBC_LIMIT_NUMERIC_PRECISION in the MS Access Section Previously.

The ODBC_REMOVE_TRAILING_SPACES is a param that was added for SQLServer Users. If set to 0 it will include all the trailing spaces in a fixed character field that the driver normally truncates before sending.

Other Tidbits

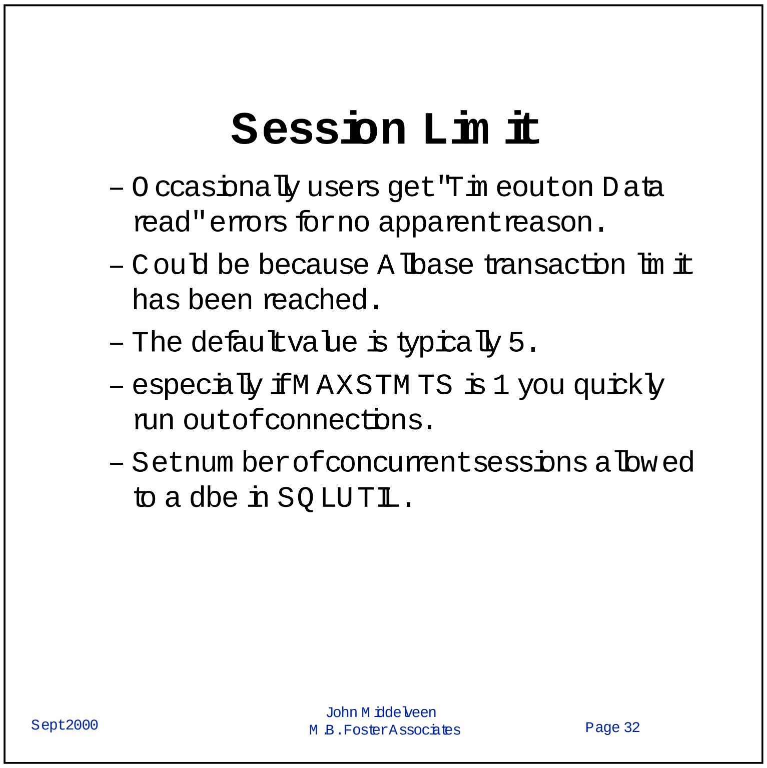
- SQL_MAXROWS
- Session Limit
- Queue Tuning
- Connecting to other DBEs
- File DSN/DSN Less Connections



SQL_MAX_ROWS

- To limit the number of rows of data returned in a query, the SQLSetStatementOption SQL_MAX_ROWS was added to the driver in version E.56.01

Notes:.



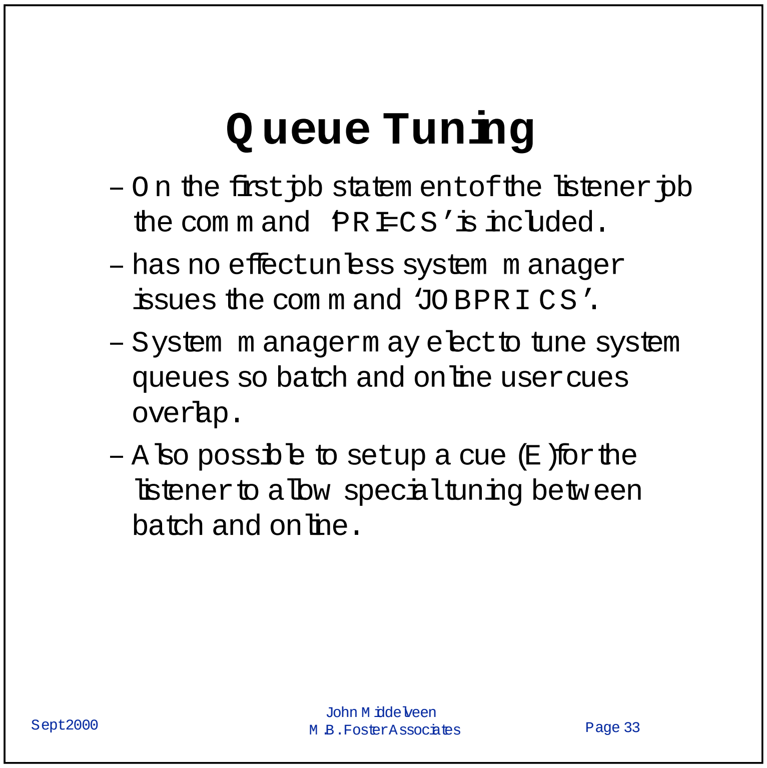
Session Limit

- Occasionally users get "Timeout on Data read" errors for no apparent reason.
- Could be because Allbase transaction limit has been reached.
- The default value is typically 5.
- especially if MAXSTMTS is 1 you quickly run out of connections.
- Set number of concurrent sessions allowed to a dbe in SQLUTIL.

Notes:

We had a customer who ran into this problem once. Their test box was set to 50 users, and the production box was set to a 5 concurrent user limit.

There was a fair bit of man hours spent trying to figure that one out, and it was a simple matter of having different configuration settings.

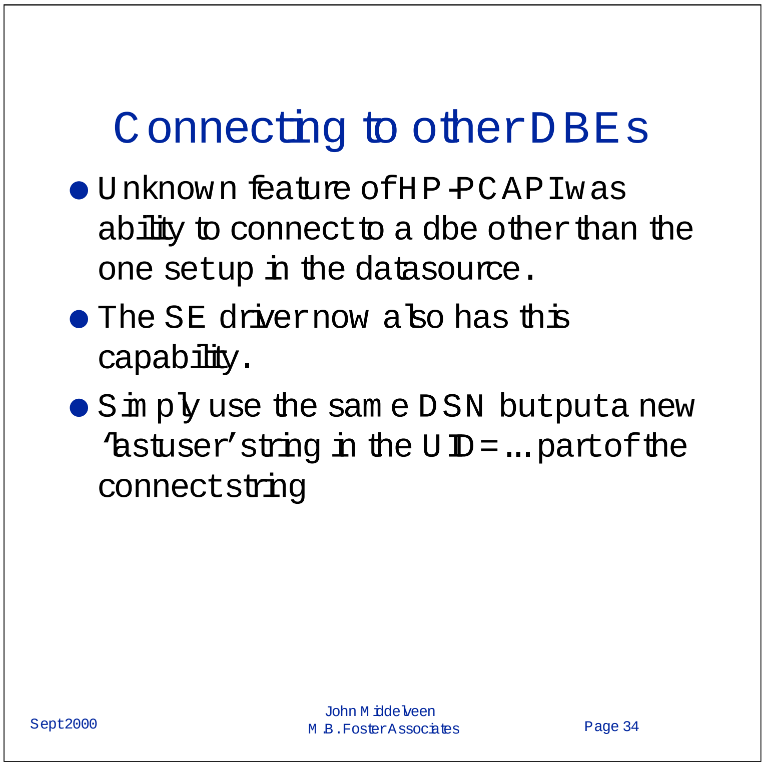


Queue Tuning

- On the first job statement of the listener job the command ‘PRI=CS’ is included.
- has no effect unless system manager issues the command ‘JOBPRI CS’.
- System manager may elect to tune system queues so batch and online user cues overlap.
- Also possible to setup a cue (E) for the listener to allow special tuning between batch and online.

Notes:.

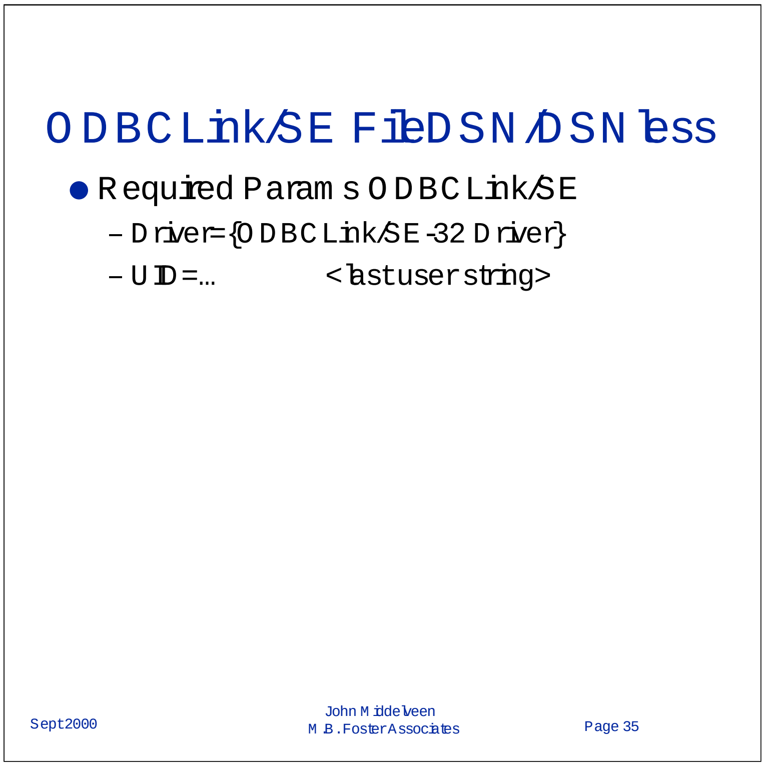
One way to tell if you are really in the Queue you expect is to look at the start of the spool file. You will see a `Priority = DS;` if you are in the D Queue instead of the C queue.



Connecting to other DBEs

- Unknown feature of HP-PCAPI was ability to connect to a dbe other than the one setup in the datasource.
- The SE driver now also has this capability.
- Simply use the same DSN but put a new 'lastuser' string in the UID=… part of the connect string

This ability was added in the early E.57.xx versions and so virtually everyone should have this ability now.



ODBCLink/SE File DSN / DSN less

- Required Params ODBCLink/SE
  - Driver={ODBCLink/SE-32 Driver}
  - UID=… <last user string>

The Last User string is the one that you would have made from a registry DSN connecting to the same datasource. (I recommend making a Registry DSN and then simply Copy and Paste the last user string.into the DSN string)

You can make up your own User Specific LastUser String by prompting the user for login information, provided you put all the information in the proper format.

The reason we require that the LastUser String is placed in the DSN String is to make the SE Driver Compatible with applications that used this hidden feature in HP-PCAPI 16 bit ODBC Driver.

Optional Parameters:

- PWD - Host Conf PWD (not for SE)
- Debug - Client Debug level (0-15)
- MAXSTMT 1 or 48
- Debug_SQLX - Host Debug Level 0-5

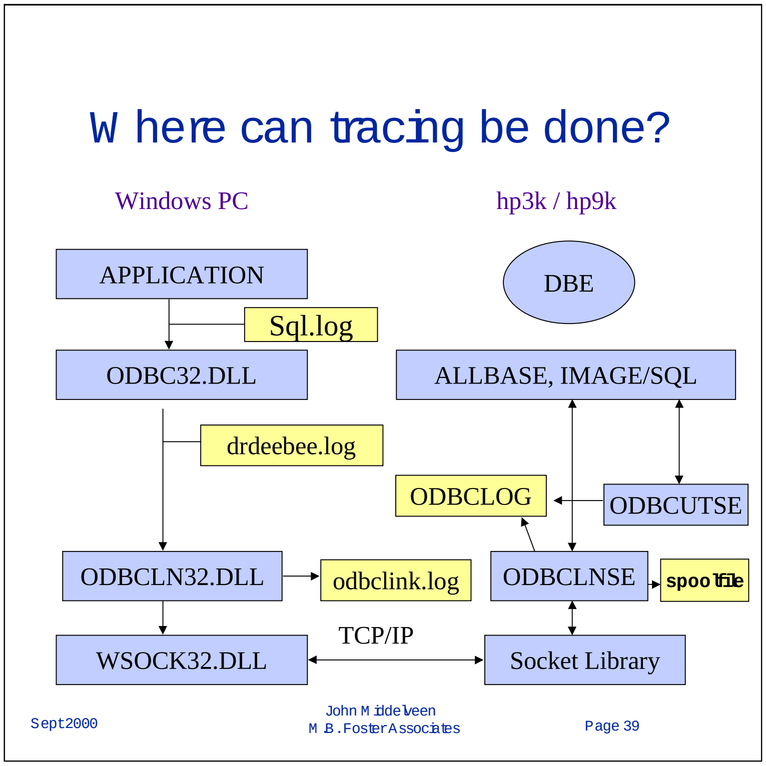
Optional Parameters (cont)

● LoginTimeout	- Connect timeout (msec)
● CommandTimeout	- Query timeout (msec)
● TranslationDLL	- Trans DLL location
● TranslationName	- Trans Name
● TranslationOption	- Trans option value

Debugging Techniques

- – Client Debugging
- – Server Debugging
- – Reading Logfile

Notes:



Where can tracing be done?

Windows PC

hp3k / hp9k

APPLICATION

Sql.log

ODBC32.DLL

drdeebee.log

ODBCLN32.DLL

odbclink.log

WSOCK32.DLL

TCP/IP

DBE

ALLBASE, IMAGE/SQL

ODBCLOG

ODBCUTSE

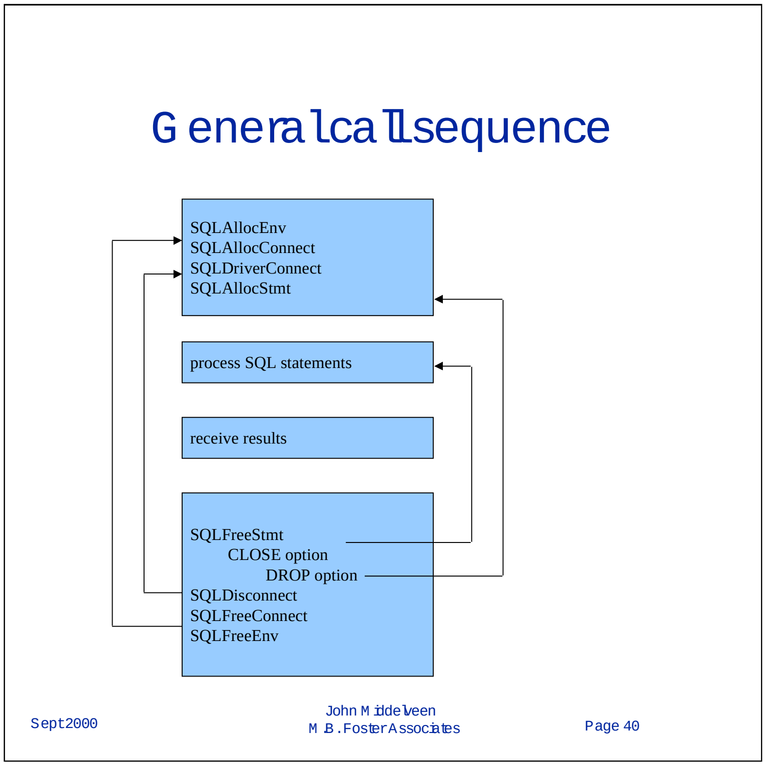
ODBCLNSE

spool file

Socket Library

Notes:

Newer versions of the sql.log will show the call both into and out of the odbc32.dll.



General call sequence

SQLAllocEnv
SQLAllocConnect
SQLDriverConnect
SQLAllocStmt

process SQL statements

receive results

SQLFreeStmt
CLOSE option
DROP option
SQLDisconnect
SQLFreeConnect
SQLFreeEnv

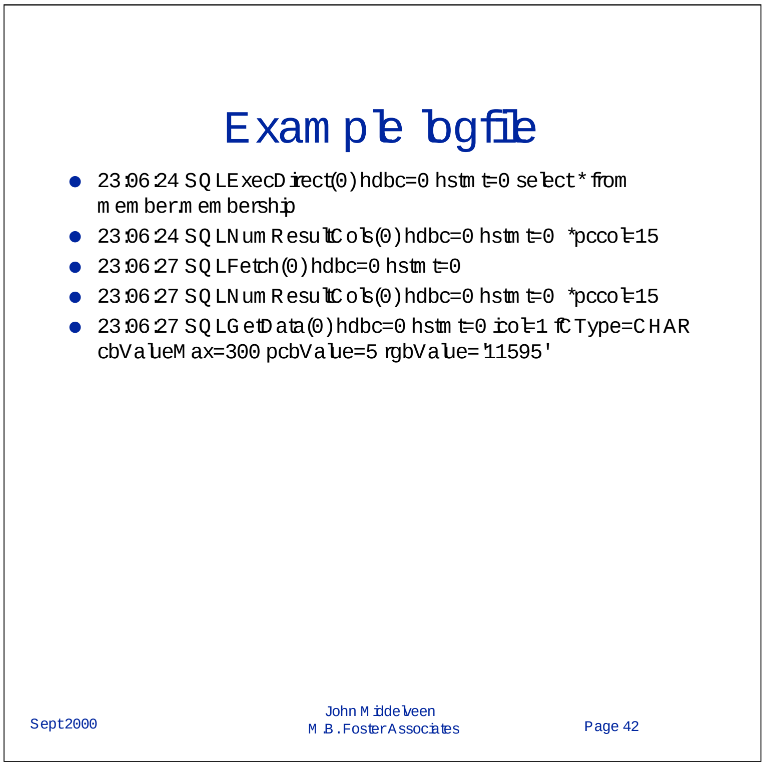
Notes:

This is extremely simplified.

If you look at a typical log of MSAccess or VB doing a single SQL Statement there will be perhaps even hundreds of ODBC calls. Most of them will be for setting and getting Connection or Statement Options.

ODBCLINK LOG

- Created when Debug Client box in the setup is checked.
- logfile 'odbclink.log' will be made in the working directory of the app. being run. (eg : MS Access will often put it in the My Documents directory)
- When the box is checked, the registry variable DEBUG for the particular datasource you are using will be set to 3. Level 3 is adequate for most users needs, and gives all ODBC Calls.



Example logfile

```
23:06:24 SQLExecDirect(0) hdbc=0 hstmt=0 select * from member.membership
23:06:24 SQLNumResultCols(0) hdbc=0 hstmt=0 *pccol=15
23:06:27 SQLFetch(0) hdbc=0 hstmt=0
23:06:27 SQLNumResultCols(0) hdbc=0 hstmt=0 *pccol=15
23:06:27 SQLGetData(0) hdbc=0 hstmt=0 icol=1 fCType=CHAR cbValueMax=300 pcbValue=5 rgbValue='11595'
```

Here is an example of an odbclink.log of a few ODBC Calls done to execute an SQL Statement and Fetch the first row of data.

What does it all mean?

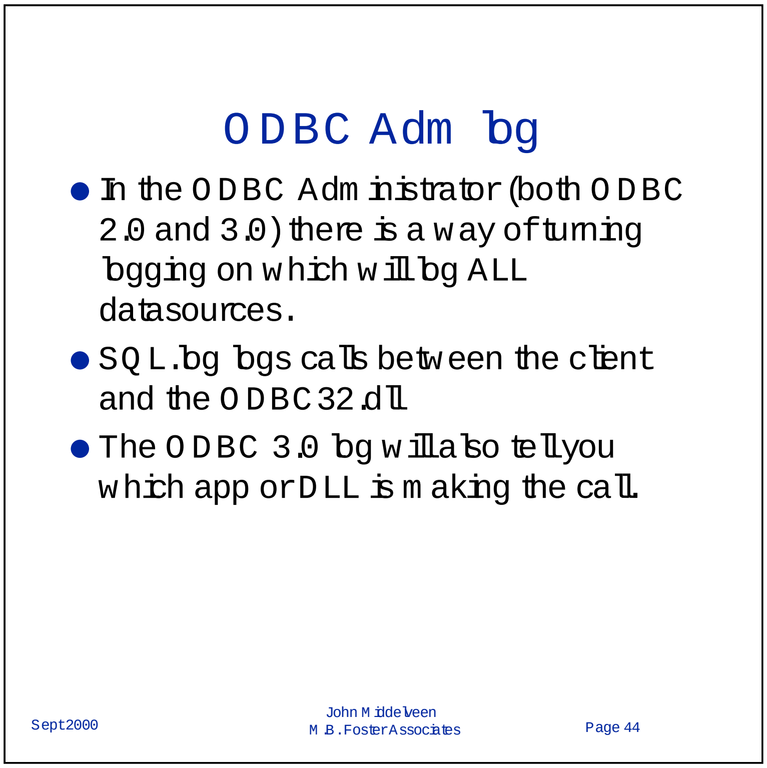
– Generally... a client log note contains

» Time - Call (return code) connection_handle, stmt_handle, other call specific information

» 14:22:25 SQLColAttributes(0) hdbc=0 hstmt=0 fDescType=7

– The most common return codes are:

- 0 Success   -1 Failure
- 1 Success with Info   100 End of Data



ODBC Admin log

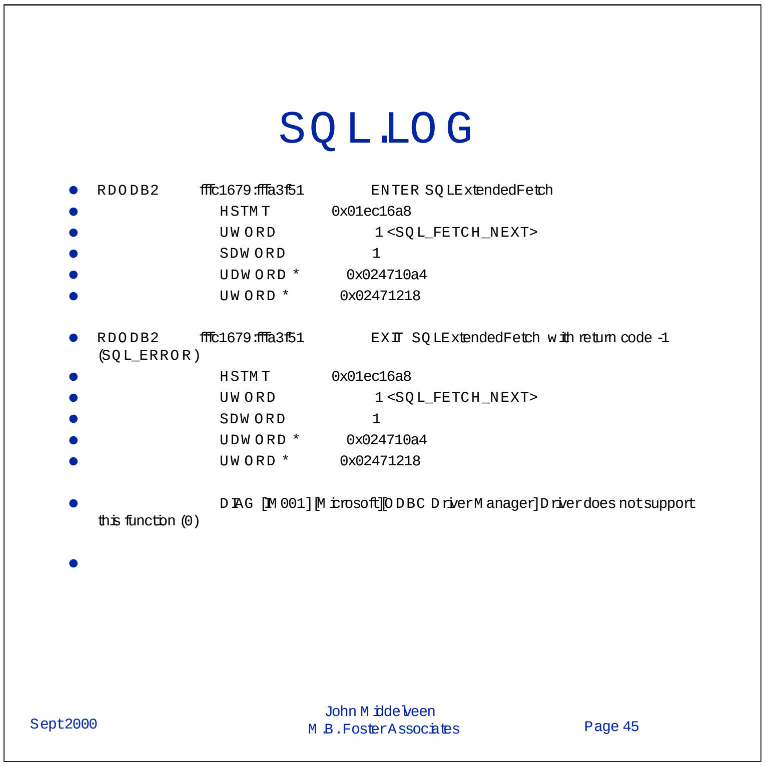
- In the ODBC Administrator (both ODBC 2.0 and 3.0) there is a way of turning logging on which will log ALL datasources.
- SQL.log logs calls between the client and the ODBC32.dll
- The ODBC 3.0 log will also tell you which app or DLL is making the call.

You use the “Tracing” Tab of the Administrator to turn on the sql.log.

The default name is SQL.LOG but you can change it (as well as where the log is written to).

One useful thing about the SQL.log is that you can reduce the output by not turning it on until just before you get to the problem you want to debug (provided you are able to control the app and that the bug happens shortly after some kind of user input)

The control button toggles between “Start Tracing Now” and “Stop Tracing Now”



SQL.LOG

```
RDODB2      fffc1679:fffa3f51       ENTER SQLExtendedFetch
                  HSTMT         0x01ec16a8
                  UWORD               1 <SQL_FETCH_NEXT>
                  SDWORD              1
                  UDWORD *        0x024710a4
                  UWORD *         0x02471218

RDODB2      fffc1679:fffa3f51       EXIT  SQLExtendedFetch  with return code -1
(SQL_ERROR)
                  HSTMT         0x01ec16a8
                  UWORD               1 <SQL_FETCH_NEXT>
                  SDWORD              1
                  UDWORD *        0x024710a4
                  UWORD *         0x02471218

                  DIAG [IM001] [Microsoft][ODBC Driver Manager] Driver does not support
this function (0)
```

Sept2000

John Middelveen
M.B.Foster Associates

In this case we see a call to SQLExtended Fetch

THE ODBCLink Driver does not support this call.

Provided that the ODBC Cursor Library is installed the odbc32.dll will translate the Extended Fetch call into a series of SQLFetch Calls.

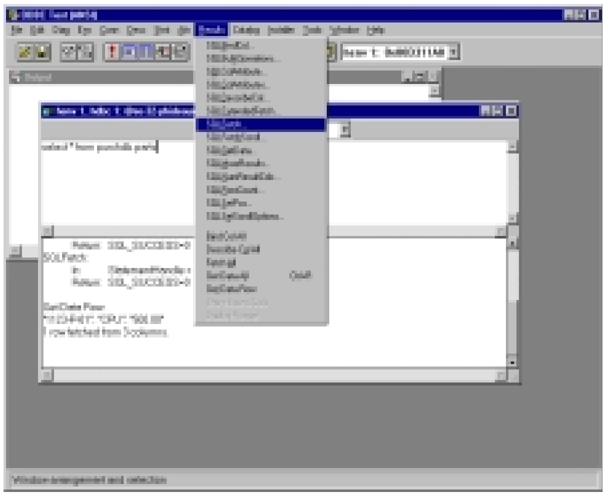
If the ODBC Cursor Library is NOT turned on then you will get an error as shown in this slide.

Note that this is one case where if you would have looked in the odbclink.log you would not have seen any indication of the error, This is because the error came from the odbc32.dll not the driver.

Client Debug Programs

- ODBC Test
- Dr DeeBee Spy
- ODBC Spy

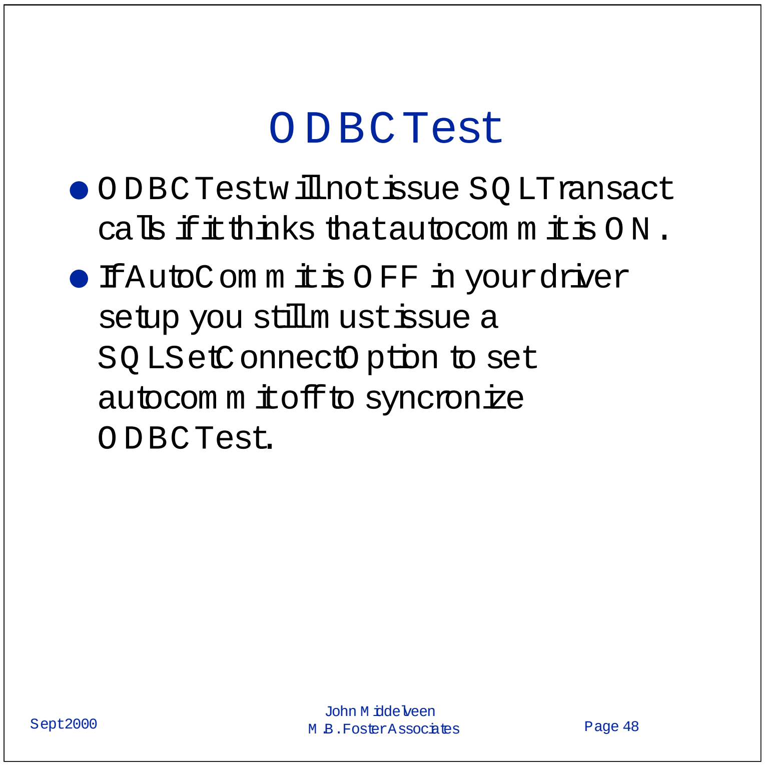
ODBCTest



This is one of the Applications I use most in the lab to debug problems.

It will allow you to call any ODBC call.

You can connect as an ODBC2 Application or an ODBC3 Application.



ODBCTest

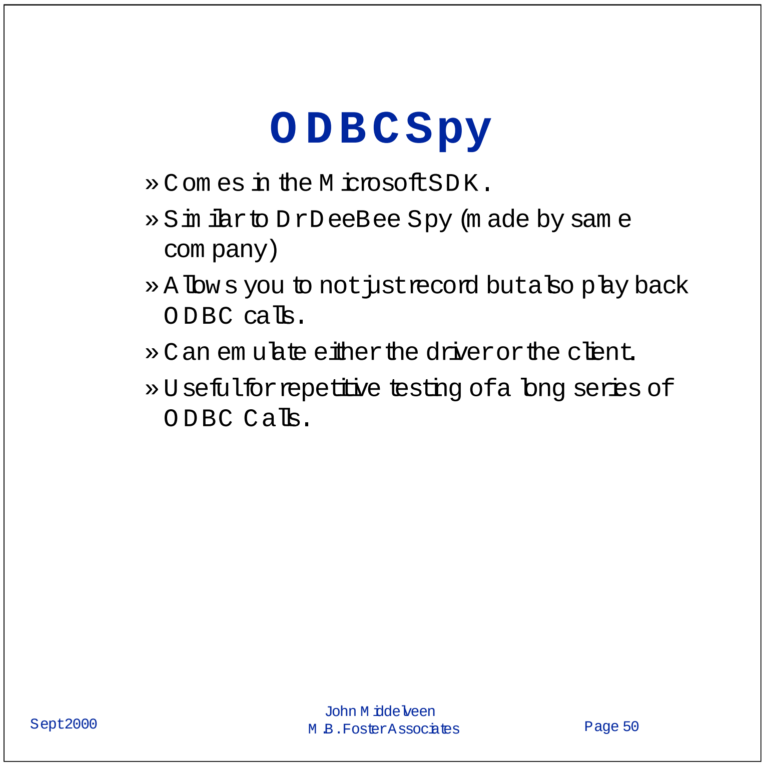
- ODBCTest will not issue SQLTransact calls if it thinks that autocommit is ON.
- If AutoCommit is OFF in your driver setup you still must issue a SQLSetConnectOption to set autocommit off to syncronize ODBCTest.

Most of You wont care about this unless you are using ODBCTest to test Updates, AND you have Auto-commit turned OFF in the Setup.

Its just one of those little GOTCHAs that a customer ran into a couple of times. Of cource silly little problems like this tend to take a lot of time to figure out

DrDeeBee Spy

- Provided with the SE Driver.
- Lists all ODBC Calls made to and from a particular datasource.
- Provides more detail than odbclink.log in that ALL parameters are listed, one parameter per line.
- BUT…Output is much larger.



ODBCSpy

» Comes in the Microsoft SDK.

» Similar to DrDeeBee Spy (made by same company)

» Allows you to not just record but also play back ODBC calls.

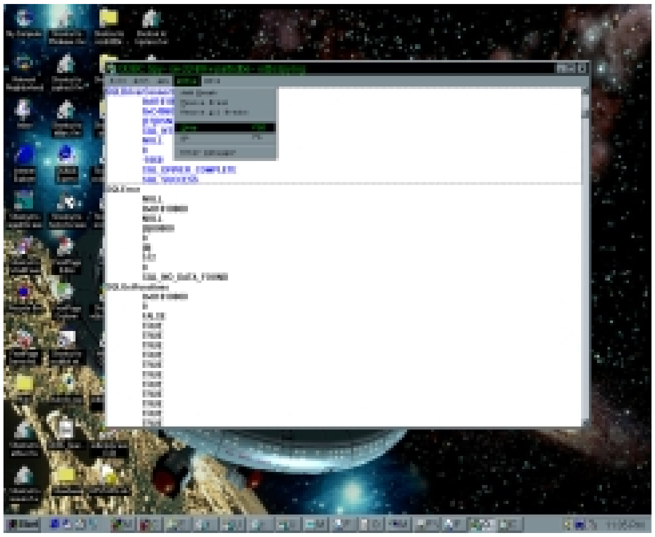
» Can emulate either the driver or the client.

» Useful for repetitive testing of a long series of ODBC Calls.

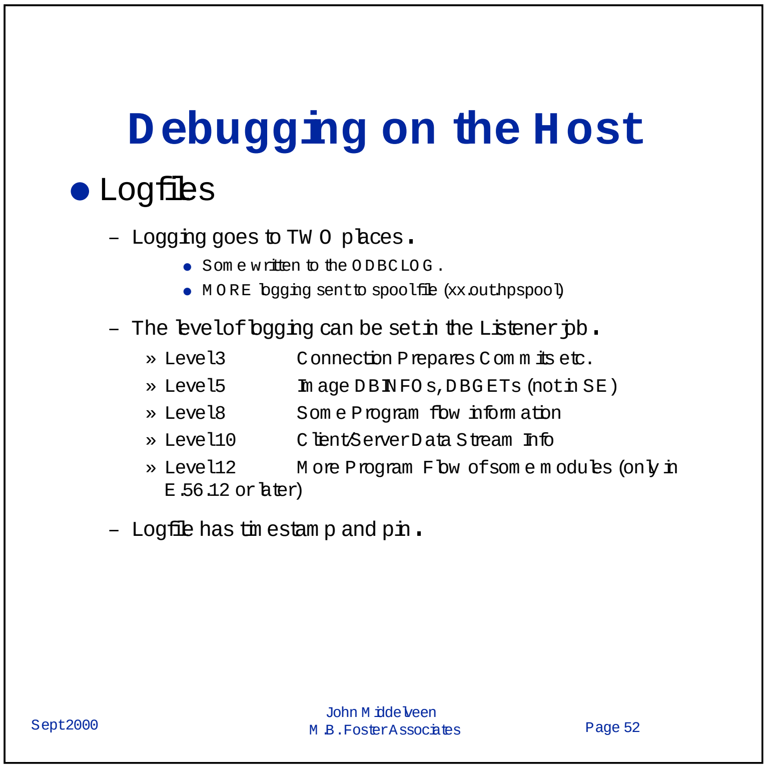
I believe you can use the log generated by DrDeeBee Spy with ODBCSpy.

This would be very usefull in Emulating someones Application to reproduce an error without actually having the application. Of cource you would still need to have the table layout so that the SQL commands could execute successfully.

ODBCSpy Program



ODBCSpy is a neet program that is probably underused, considering what it can do to help in Debugging.

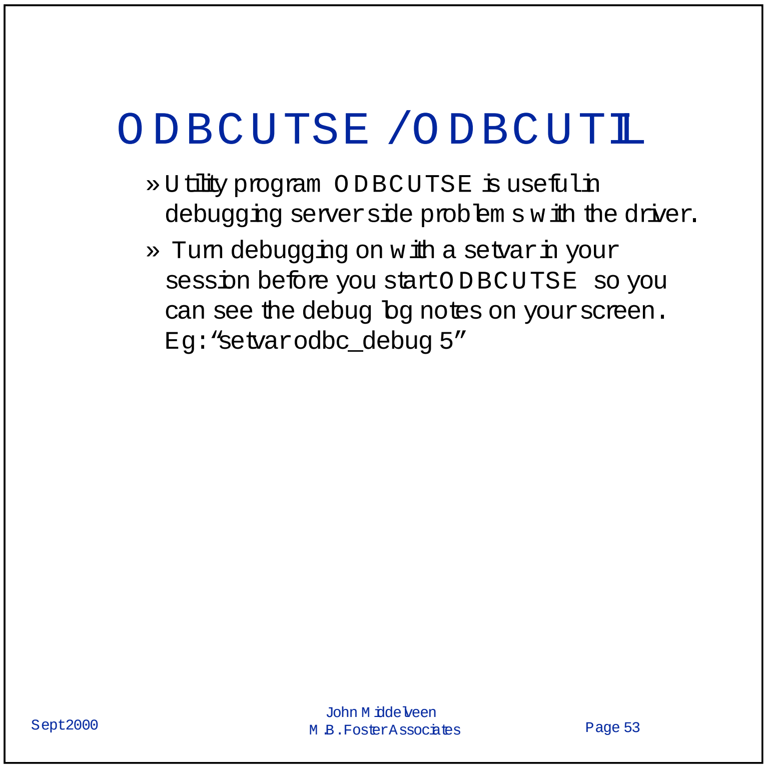


Debugging on the Host

- Logfiles
  - Logging goes to TWO places.
    - Some written to the ODBCLOG.
    - MORE logging sent to spoolfile (xx.out.hpspool)
  - The level of logging can be set in the Listener job.
    - » Level 3 Connection Prepares Commits etc.
    - » Level 5 Image DBINFOs, DBGETs (not in SE)
    - » Level 8 Some Program flow information
    - » Level 10 Client/Server Data Stream Info
    - » Level 12 More Program Flow of some modules (only in E.56.12 or later)
  - Logfile has timestamp and pin.

As mentioned before level 3 is the default(if you click debug_sqlx in the setup)

There are a few debug levels even higher than 12 but these are rarely used.



ODBCUTSE / ODBCUTIL

» Utility program ODBCUTSE is useful in debugging server side problems with the driver.

» Turn debugging on with a setvar in your session before you start ODBCUTSE so you can see the debug log notes on your screen. Eg: “setvar odbc_debug 5”

There are only a few commands that’ are likely to be of interest. We use this programoften at the lab to debug server side problems with the x debugger, in which case we use the unoptimized compile.

To get a list of commands type help.

You only need to enter enough of a command to differentiate it from another command (usually 2 letters is enough)

eg sh for show (to show a list of tables) or SH <owner.table>

to show details of a specific table.

You can then enter select statements and then a FE to fetch a row

or FE ALL to fetch all the rows