OpenLink Universal Data Access Driver Suite

Multi-Tier User Guide
Version 3.2

Chapter1: Release Notes & Bugs Fixed

General

PTR:1000339 **Logging to file doesn't work in NT Broker CVSID 42**

Systems General *Databases*, on Windows NT (Intel) 4.0 *Platform*, to General *Client*

Notes Start the CVS00042 archive broker with the options: -dvL <filename>

The Broker creates the log file but does not write any Trace output to the File.
Status: Fixed

PTR:1000378 **Allow customers to do a silent installation of our client software on Windows or NT**

Systems General *Databases*, on General *Platforms*, to Windows 95 *Client*

Notes The new version of install shield has an option that does a silent install & uses all defaults.
Status: Fixed

PTR:1000381 **SQLGetInfo(SQL_QUOTED_IDENTIFIER_CASE) returns an invalid value**

Systems General *Databases*, on General *Platforms*, to Windows 95 *Client*

Notes This problem appears to be a general one across all databases as the same error was returned when tested against Oracle, OpenIngres & Progress. Below is the output from an ODBCTest session against OpenIngres

```
Successfully connected to DSN 'oig2'.
SQLGetInfo:
In: ConnectionHandle = 0x007F0BD8,
InfoType = SQL_QUOTED_IDENTIFIER_CASE=93, InfoValuePtr = 0x004B2048, BufferLength = 300,
StringLengthPtr = 0x004B2018, Information Value Type = SQL_C_SLONG=-16
Return: SQL_SUCCESS=0
Out: *InfoValuePtr = -892731389
TST0006: The value -892731389 was not a valid named constant for SQL_QUOTED_IDENTIFIER_CASE.
*StringLengthPtr = 2
```

Status: Fixed

PTR:1000386 **Add routine to handle LIKE escape sequence**

Systems General *Databases*, on General *Platforms*, to Windows 95 *Client*

Notes When a select statement is parsed, the like escape sequence is not handelled.
Status: Fixed

PTR:1000403 **{fn now( )} not supported on scrollable cursors**

Systems General *Databases*, on General *Platforms*, to Windows 95 *Client*

Notes Scrollable cursor parser rejects "select {fn now()} from emp" - It does not recognize {fn now()} clause.
Status: Fixed

PTR:1000424 **Open Agent API Completion**

Systems General *Databases*, on General *Platforms*, to General *Client*

Notes The Open Agent API is a key element in our Agent Development process as it process a better abstraction between our Agent and the Comms Layer.
Status: Fixed

PTR:1000436 Build NT ODBC agent installer

Systems General *Databases*, on Windows NT (Intel) *Platform*, to General *Client*

Notes This installer will effectively just install the Broker and Service agent , ie: ODBC, JDBC & Proxy agents to enable users to use desktop ODBC Driver like Dbase, FoxPro etc to access these tools via our drivers.
Status: Completed

PTR:1000441 Add oplshut -s functionality to the web configurator

Systems General *Databases*, on General *Platforms*, to General *Clients*

Notes The functionality has been added to the web configurator code and has been merged with New Web Configurator wizard code.

The functionality of oplshut -s is already present in the Web Configurator. Need a script to expose it.
Status: Fixed

PTR:1000446 Using JDBC agent to retrieve Null values from a Text Column results in the JDBC agent crashing in NT and Segmentation Violation on Unix

Systems General *Databases*, on General *Platforms*, to Java *Client*

Notes Bug fetching NULLs in the JDBC agent.
Status: Fixed

PTR:1000476 WWW_SV fails to run on CVS 45

Systems General *Databases*, on Windows NT (Intel) 4.0 *Platform*, to General *Clients*

Notes WWW_SV fails to run when there is a space in the installed path.
Status: Fixed

PTR:970389 Install.sh Problems

Systems General *Databases*, on General *Platforms*, to General *Clients*

Notes Install.sh has a line of code that potentially breaks installations.
The script has now been extensively modified for better functionality.
Status: Fixed

Informix

PTR:1000380 Informix agent strips trailing spaces from CHAR columns when fetching

Systems Informix *Databases*, on General *Platforms*, to Windows 95 *Clients*

Notes Informix agent wrongly strips off trailing spaces when fetching from Informix CHAR columns.
Status: Fixed

PTR:1000479 Informix agent instance only supports 1 DBMS session

Systems Informix *Database*, on General *Platforms*, to General *Clients*

Notes At present, each Informix agent instance can only support one database connection. If a re-use rule is specified it must be 'ifsame database, 'ifsame user'. (This rule was presumably enforced by PMN because ESQL/C automatically includes a single, global sqlca.)

Informix 7.2 supports 'SET CONNECTION ...' to make the current connection dormant, or a dormant connection current. So, it appears to have the facility to support more than one connection. We currently don't make use of this command. This facility needs testing and if feasible, the Informix agent should be modified to support > 1 database session per instance.
Status: Fixed

PTR:1000509 Different column names are returned to SQLTables if a parameter is supplied.

Systems Informix 7 *Database*, on General *Platforms*, to General *Clients*

Notes Different column names are returned if all parameters are null, than when one parameter is specified (such as 'TABLE'). The same column names should be returned in both instances. This behavioral difference is causing problems with programs that use the stated catalog function.

If you look at the names of the columns, they have changed from
"table_qualifier", "table_owner", "table_name", "table_type", "remarks"
when there is no parameter to
"(constant)", "owner", "tabname", "tabtype", "(constant)"
when the parameters are supplied.

This confuses the development tool which is probably interrogating the resultset metatdata, or simply looking for the column in a row named "table_name".

```
When one parameter is specified:
SQLTables:
In: StatementHandle = 0x00A31878, CatalogName = SQL_NULL_HANDLE,
NameLength1 = 0, SchemaName = SQL_NULL_HANDLE, NameLength2 = 0,
TableName = SQL_NULL_HANDLE, NameLength3 = 0, TableType = "TABLE",
NameLength4 = 5
Return: SQL_SUCCESS=0

Get Data All:
"(constant)", "owner", "tabname", "tabtype", "(constant)"
<Null>, "informix", "Tt", "TABLE", ""
<Null>, "informix", "account", "TABLE", ""
...
When all parameters are null:
Get Data All:

"table_qualifier", "table_owner", "table_name", "table_type", "remarks"
<Null>, "informix", "Tt", "TABLE", ""
<Null>, "informix", "account", "TABLE", ""
<Null>, "informix", "branch", "TABLE", ""
<Null>, "informix", "call_type", "TABLE", ""
…
```

Status: Fixed

PTR:1000392 Adding to tables with datetime fields using the JDBC Scrollable Cursor demo

Systems Informix Possibly all agents *Database*, on DGUX (Intel) Possible all OS's *Platform*, to Java JDK 1.1 *Client*

Notes When you try to Add a row using the scrollable cursor demo, the database complains about the 'ts' escape. The file (udbc.out) demonstrates it in full, but the database message:
'{ts '1996-02-12 00:0' shows where the problem lies. See below
for more:
Replicated the behavior on Informix DG-UX Intel. The customer was using Ingres 6.4 on Solaris.

```
SQLExtendedFetch hStmt=DBD0003
fFetchType=1 irow=1 pcrow=DFFFF004 rgfRowStatus=5B6A0
ConvertCHAR: retBufferLength(41) to SQL_C_CHAR
ConvertCHAR: retBufferLength(21) to SQL_C_CHAR
ConvertTIMESTAMP: precision,scale (20,0) to SQL_C_CHAR
ConvertTIMESTAMP temBuf string = %04u-%02u-%02u %02u:%02u:%02u
ConvertTIMESTAMP: precision,scale (20,0) to SQL_C_CHAR
ConvertTIMESTAMP temBuf string = %04u-%02u-%02u %02u:%02u:%02u
ConvertTIMESTAMP: precision,scale (20,0) to SQL_C_CHAR
ConvertTIMESTAMP temBuf string = %04u-%02u-%02u %02u:%02u:%02u
ConvertCHAR: retBufferLength(23) to SQL_C_CHAR
ConvertCHAR: retBufferLength(2) to SQL_C_CHAR
ConvertCHAR: retBufferLength(6) to SQL_C_CHAR
ConvertCHAR: retBufferLength(41) to SQL_C_CHAR
ConvertCHAR: retBufferLength(101) to SQL_C_CHAR
SQLExtendedFetch: returning SQL_SUCCESS

SQLSetPos hStmt=DBD0003
irow=1 fOption=4 fLock=0
SQLSetPos: returning SQL_ERROR

SQLError: szErrorMsg=DFFFEB94 cbErrorMsgMax=256, pcbErrorMsg=0
```

```
szSqlState=DFFFEC94 pfNativeError=DFFFEF94
Returning [Ingres Server]Error in row

SQLError: szErrorMsg=DFFFEB94 cbErrorMsgMax=256, pcbErrorMsg=0
szSqlState=DFFFEC94 pfNativeError=DFFFEF94
Returning [Ingres Server]E_US10D4 bad character found in date string beginning with '{ts '1996-02-12
00:0'. (-40206)
```

This problem has been recreated against Ingres 6.4 on SCO 4 using the method described above with the JDBC ScrollDemo program by attempting to update or insert a row with a datetime field in it.
Have also recreated this problem with MSSQLServer although it does not occur with Oracle. On further investigation it is apparent that inserts o datatypes are passed to the JDBC agent using the ODBC escape syntax for dates ie {ts <timestamp>} and is thus dependant on the Database agent being able to convert these to native Datbase format before being presented to the database. For paramertised queries only Oracle has support for converting from the escape syntax to native database syntax
Status: Fixed

PTR:1000514 VAX VMS Broker crashes when a single quote (') is placed at the end of the username

Systems Ingres *Database*, on VAX VMS 5.5-2 *Platform*, to General *Client*

Notes **Status: Fixed**

PTR:1000510 oplrqb.exe leaves one handle unrelinquished per agent in NT

Systems Ingres 2.0 *Database*, on Windows NT (Intel) NT 4 Service Pack 3 *Platform*, to General *Client*

Notes With ReUse=never, connect to a request broker on NT, while monitoring the number of handles open in the task monitor.

For 10 connections & disconnections from udbctest, the number of handles used by oplrqb.exe increases by 10, i.e. the handles are not being closed. The loop used is:

```
zsh, oplfbsd 3:51PM examples > repeat 10 ./udbctest <<EOF
dsn=ing7
EOF
zsh, oplfbsd 3:56PM examples >
```

This is independent of ReUse (Never or 'upto 2'). Also, it is not related to the 2 minute TIME_WAIT socket problem - we waited until all the sockets had died on the client and server, and the handles are still in use.

There are also know leaks in the NT system files.
Status: Fixed

PTR:1000519 SQLTables() against Ingres gives blank list of tables in MS Query

Systems Ingres 6.4 *Database*, on VAX VMS *Platform*, to Windows NT (Intel) NT4 SP3 *Client*

Notes Using MS Query, under Data / Get external Data, select 'create new query'. Create a new File DSN using the dialog Query presents. Fill in points 1, 2 and 3 (selecting a driver for the connection and attempting to connect). Once the connection is established, Query calls SQLTables() with a % in the owner field to make a list of tables in the datasource. This list comes over as blank.
On Vax, there is also a 'syntax error or access' at this point.

Client-side debug trace follows:

```
SQLTables hStmt=DBD0003
szTableQualifier=823060 cbTableQualifier=0 szTableOwner=823070 cbTableOwner=-3
szTableName=823080 cbTableName=0 szTableType=823090 cbTableType=0
Qualifier='' Owner='%' Name='' Type=''
SQLTables: returning SQL_SUCCESS

SQLBindCol hStmt=DBD0003
icol=2 fCType=1 rgbValue=3D894 cbValueMax=65 pcbValue=3DE08
SQLBindCol: returning SQL_SUCCESS

SQLFetch hStmt=DBD0003
SQLFetch: returning SQL_NO_DATA_FOUND
```

This is reported on Vax, confirmed on HP9 and Sequent. The problem lies in the SQLTables() call where instead of using NULL pointers to strings, it uses NULL strings.

(This is reproduced in ODBCTest / Gator by specifying empty string instead of null string in a call to SQLTables().)
Status: Fixed

PTR:970225 **Unable to connect as many users when VMS Broker is running in detached mode as opposed to non-detached mode**

Systems Ingres *Database*, on General VAX VMS *Platform*, to Windows NT (Intel) *Client*

Notes The Broker running in detached mode does not appear to be inheriting the process Quotas it inherits when running in non-detached mode. The errors typically received when the Broker fails to start an agent session are: oplrqb: cannot create internal connection: RPC: Remote system error - Error 65535 (3506252) oplrqb: cannot create internal connection: RPC: Remote system error - Error 65535 (28) oplrqb: File "OPENLINK_ROOT:[BIN]ing6_sv.exe" not found (status = 114730)
Status: Fixed

MS SQL Server

PTR:1000358 **DB-Lib Agents - Share DBPROCESS between sibling HSTMTs**

Systems MS SQL Server *Database*, on General *Platform*, to General *Client*

Notes Re-architecting of DB-Library agents was required to avoid potential deadlock when performing batch updates across several hstmt's with autocommit off.
Status: Fixed

PTR:1000431 **JUDBC agent dies retrieving a null text field**

Systems MS SQL Server 6.5 *Database*, on Windows NT (Intel) 4.0 *Platform*, to Java 1.1 *Client*

Notes There was a bug in transferring NULLs in the JDBC agent.

With a type-3 judbc connection to MS SQLServer 6.5 the judbc agent dies retrieving a null text field.

There is no problem retrieving a null varchar, nor a text field if it contains data.
Status: Fixed

PTR:1000371 **MS SQL Server Stored Procedure Support**

Systems MS SQL Server All *Database*, on Windows NT (Intel) *Platform*, to Windows NT (Intel) *Client*

Notes DBLIB Agent does not handle OUTPUT or MIXED Parameters Correctly
Status: Fixed

OpenIngres

PTR:1000393 **DG/UX (Intel) release 1.5.5 OpenIngres agent archive is corrupt on the WEB**

Systems OpenIngres *Database*, on DGUX (Intel) *Platform*, to Windows 95 *Client*

Notes **Status: Fixed**

Oracle

PTR:970220 — Timestamp Type Handling Problems (INSERTS) & Parameter Binding

Systems Oracle *Database*, on General *Platform*, to Windows 95 *Client*

Notes If the insert statement has an ODBC date-time escape sequence ({ts ...}) it works , or if it is coded a "to_date(...)" the insert statement works. When a parameter is used instead of a hard coded date then it doesn't work. Support for ODBC escape syntax handling for parameters has been added.
Status: Fixed

PTR:1000433 — OPTIONS field in our OpenLink DSN Config Utility should hold the SQL*NetAlias name for Oracle sessions

Systems Oracle *Database*, on General *Platform*, to General *Client*

Notes A sqlnet alias can now be specified in the multitier options dialog. However, there is still a problem with making two connections to different aliases from a single application(ptr1000480).

The OPTIONS field in our OpenLink DSN Config Utility should hold the SQL*Net
Alias name for Oracle sessions.
Status: Fixed

PTR:1000471 — Oracle Lite DLL crashed when runing against a Multithreaded application

Systems Oracle *Database*, on General *Platform*, to Windows NT (Intel) *Client*

Notes Oracle Lite and MT Generic client crashes when running against an application running multiple threads continuosly in a loop.
Status: Fixed

PTR:1000473 — Oracle Lite crashes after 16 connections on the same handle

Systems Oracle *Database*, on General *Platform*, to General *Client*

Notes Modified source so the lite drivers could be built without the MULTIPLEX LDA feature.
The Oracle Lite driver was being built with the define MULTIPLEXLDA which is a feature not supported with the lite driver.
Status: Fixed

PTR:1000448 — Blob failure in Oracle using ASP

Systems Oracle 7.2 *Database*, on Windows NT (Intel) 4.0 *Platform*, to Windows NT (Intel) 4.0 *Client*

Notes using ADO in Active server page in IIS 4.0

when trying to connect to Oracle database using cursor, if the table contains BLOB fields an error is returned.
Status: Fixed

Postgres95

PTR:1000466 — SQLDescribeCol returns SQL_CHAR instead of SQL_DATE or SQL_TYPE_DATE for date columns

Systems Postgres95 6.3.2 *Database*, on Linux/Elf *Platform*, to Linux/Elf *Client*

Notes Returns SQL_CHAR for postgres 'date' table columns.
SQLDescribeCol() should return SQL_DATE or SQL_TYPE_DATE.
The postgres agent fetches all data as character data and currently relies on the client to do all conversions. At the moment the time and data datatypes are not converted. The data set containing the character representation of the data will be re-written on the server before being sent to the client. This work is required for the cursor library but a side effect should be that the time and date types will be supported.

Status: Fixed

Progress

PTR:1000477 3 Issues with Progress Lite data source setup box

Systems Progress *Database*, on General *Platform*, to Windows NT (Intel) *Client*

Notes Issue 1. password text box in DSN setup:
In addition to the User ID text box, can we add a Password text field as well.

Issue 2
In ODBC Admin when configuring a DSN, the box where you enter the TABLEVIEW parameter is restricted to 35 characters, where as the other two are restricted to 255. Can the size of this also be restricted to 255?

Issue 3
The tab Order at present is incorrectly set. It currently moves from Session options to Tableview and then to Database options
Status: Fixed

PTR:1000502 Registering Progress Lite Large with a Large key still shows 'Software has expired'

Systems Progress *Database*, on General *Platform*, to Windows NT (Intel) *Client*

Notes A problem when registering the Progress Lite large drivers.

When trying to register progress large with a large key, it showed a successful registration but when a connection was attempted, a dialog box appeared showing that the software had expired.
Status: Fixed

PTR:1000545 Progress 8.2c Lite Driver does not return high-byte characters (², ³, etc.).

Systems Progress *Database*, on General *Platform*, to Windows NT (Intel) 4 *Client*

Notes Unable to replicate this problem using a cvs48 lite driver connecting to Progress on NT and on oplusdgint. In both cases the high byte characters were returned correctly.

When using the 3.2 (csv48) drivers, the Progress 8.2c Lite
Driver does not return high-byte characters (², ³, etc.). The Multi-Tier driver returns these characters properly.
Status: Fixed

PTR:1000511 Cannot drop table using Progress 7 agent

Systems Progress 7 *Database*, on General *Platform*, to General *Client*

Notes Removed the -NL connection parameter and reverted to appending "FOR READ ONLY" to select statements so drop statements now work. Errors when selecting "FOR READ ONLY" from views are trapped and "FOR READ ONLY" is removed.

When parsing sql in the Progress agent we append "FOR READ ONLY" to select statements. However, in some versions of Progress 7 we cannot use "FOR READ ONLY" with views so we use the -NL connection parameter as a workaround in all Progress 7 agents.

Connecting with the -NL parameter means that when Progress translates the DROP TABLE code into 4GL, it leaves the lock requested as No-Lock and the statement fails.
Status: Fixed

PTR:970385 Agent dies after any simple select statement is issued.

Systems Progress 8.2 *Database*, on Dec Unix OSF/1 OSF1 V4.0 564 alpha *Platform*, to Windows 95 *Client*

Notes **Status: Fixed**

PTR:1000414 **Installation and Portation of Progress 8.2C on NT Alpha**

Systems Progress 8.2C *Database*, on Windows NT (Alpha) 4.0 *Platform*, to General *Client*

Notes **Status: Fixed**

PTR:1000382 **Character Space Padding Problems**

Systems Progress All *Databases*, on General *Platforms*, to Windows 95 *Client*

Notes The customer wanted the first 3 characters but the application was sizing the columns to 2k. The environment variable created for ptr981080 can be used to reduce this size from 2k.

SQL column display size from Openlink is a lot wider (containing blanks?) than Progress'. This is giving some ODBC & JDBC Applications Dynamic Sizing problems.

Note: similar to PTR100380
Status: Fixed

Sybase

PTR:1000494 **Problems with Sybase/SQLServer stored procedure support**

Systems Sybase *Database*, on General *Platform*, to General *Client*

Notes Sybase dblib, ctlib and SQLServer drivers crash when fetching row results from a stored procedure containing timestamp data (only when parameter markers are used)
Stored procedures containg the Transact SQL print command generate an error that stops any more results being fetched.(dblib only)
Status: Fixed

PTR:1000559 **Sybase /SQLServer do not allow duplicate fields in an orderby clause**

Systems Sybase *Database*, on General *Platform*, to General *Client*

Notes This has been seen working with Sybase and is a bug in the cursor library as the cursor library appends an order by clause to the query internally which results in the query sent to Sybase being of the form: 'SELECT id,datecreated,... FROM info ORDER by id, id assuming a key field named 'id'.

Sybase / SQLServer do not allow duplicate fields in an order by clause whereas every other database tested against does not complain. The fix required is thus to scan the query to check whether key columns are already present in the original order by, and if so don't attempt to add them to the end of the ordering list.
Status: Fixed

Unify 2000

PTR:1000383 **Portation of Unify DataServer 6 on Sequent at customer Site**

Systems Unify 2000 *Database*, on Dynix/Ptx *Platform*, to Windows 95 *Client*

Notes **Status: Fixed**

PTR:970259 **Unify ORDER BY clauses from Access**

Systems Unify 2000 *Database*, on General *Platform*, to Windows 95 *Client*

Notes Fails with a syntax error (-2131) or a "Invalid ORDER BY specification (-2042) based on the mode and query that is run.
Status: Fixed

Chapter2: Client Components Installation

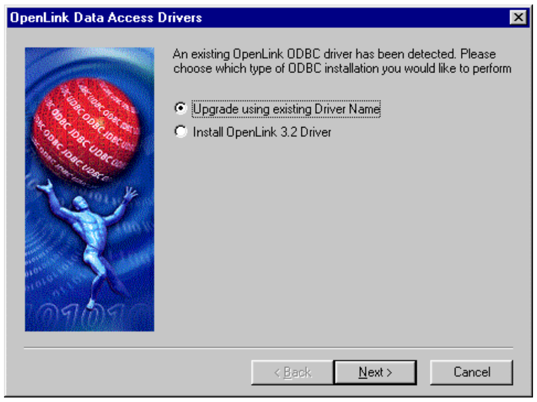
Generic ODBC Client For Windows Installation

Client Components Installation

1. After downloading the OpenLink Data Access Driver Suite client components from our Website, Unzip the contents of the ZIP format archive into a temporary installation directory on your client machine.

2. Run the setup.exe installer program from the temporary installation directory and then follow the on screen prompts.

3. The OpenLink ODBC Driver installer automatically determines if earlier versions of the OpenLink ODBC client components have been installed on your machine. By default it will install a new set of client components alongside your existing components as opposed to replacing them (you can choose to replace your older client). The dialog depicted below illustrates this step



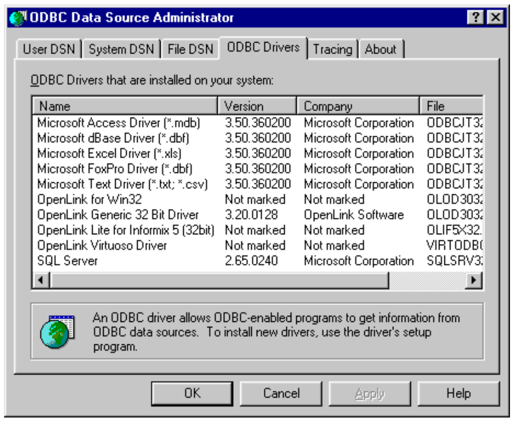
An existing OpenLink ODBC driver has been detected. Please choose which type of ODBC installation you would like to perform

[✓ Upgrade using existing Driver Name]
Install OpenLink 3.2 Driver

4. After installation you will be able to safely remove the files in the temporary directory.

ODBC Data Source Name (DSN) Configuration

Once you have completed the installation process, proceed to create an new ODBC DSN using the ODBC Driver Manager (the ODBC v3.0 Driver Manager is depicted below displaying a list of installed ODBC Drivers) situated in the Windows Control panel.



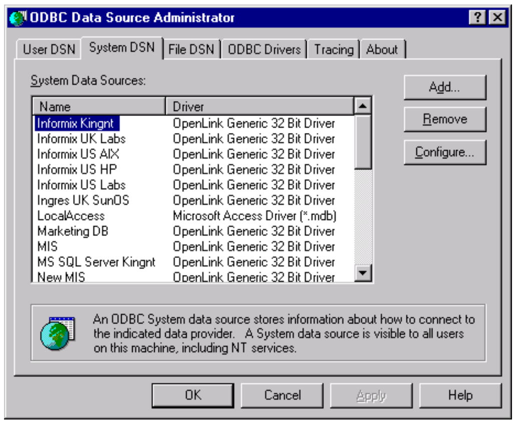
ODBC Drivers that are installed on your system:

Name	Version	Company	File
Microsoft Access Driver (*.mdb)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft dBase Driver (*.dbf)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft Excel Driver (*.xls)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft FoxPro Driver (*.dbf)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft Text Driver (*.txt; *.csv)	3.50.360200	Microsoft Corporation	ODBCJT3
OpenLink for Win32	Not marked	Not marked	OLOD303
OpenLink Generic 32 Bit Driver	3.20.0128	OpenLink Software	OLOD303
OpenLink Lite for Informix 5 (32bit)	Not marked	Not marked	OLIF5X32.
OpenLink Virtuoso Driver	Not marked	Not marked	VIRTODB
SQL Server	2.65.0240	Microsoft Corporation	SQLSRV3

An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

Configuring ODBC System & User Data Source Names (DSNs)

ODBC 2.0 & 3.0 Driver Manager (depicted below with a list of existing ODBC System DSNs) allows the creation of User, System, and File DSNs. The steps for creating a User or System DSN are as follows:-

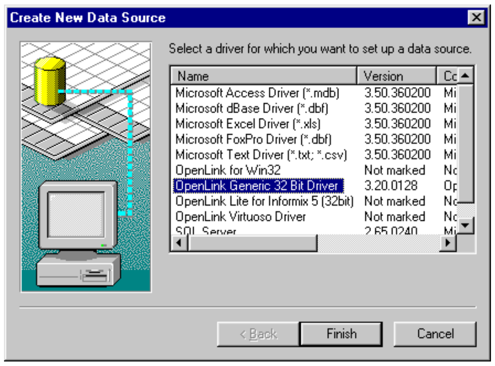


System Data Sources:

Name	Driver
Informix Kingnt	OpenLink Generic 32 Bit Driver
Informix UK Labs	OpenLink Generic 32 Bit Driver
Informix US AIX	OpenLink Generic 32 Bit Driver
Informix US HP	OpenLink Generic 32 Bit Driver
Informix US Labs	OpenLink Generic 32 Bit Driver
Ingres UK SunOS	OpenLink Generic 32 Bit Driver
LocalAccess	Microsoft Access Driver (*.mdb)
Marketing DB	OpenLink Generic 32 Bit Driver
MIS	OpenLink Generic 32 Bit Driver
MS SQL Server Kingnt	OpenLink Generic 32 Bit Driver
New MIS	OpenLink Generic 32 Bit Driver

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

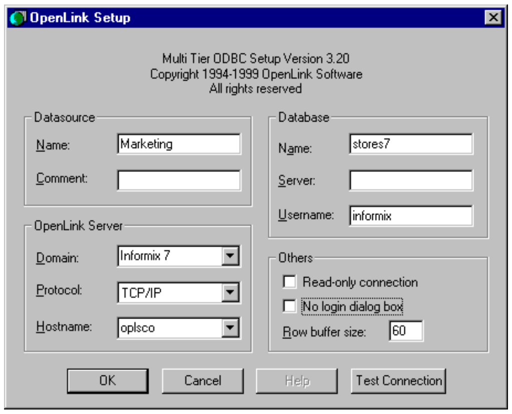
1. Pick the ODBC Driver to be used to create your ODBC DSN, in the case of the OpenLink Data Access Driver Suite 3.2, this would be the Driver identified below as "OpenLink Generic 32 Bit Driver".



Select a driver for which you want to set up a data source.

Name	Version	Co
Microsoft Access Driver (*.mdb)	3.50.360200	Mi
Microsoft dBase Driver (*.dbf)	3.50.360200	Mi
Microsoft Excel Driver (*.xls)	3.50.360200	Mi
Microsoft FoxPro Driver (*.dbf)	3.50.360200	Mi
Microsoft Text Driver (*.txt; *.csv)	3.50.360200	Mi
OpenLink for Win32	Not marked	No
OpenLink Generic 32 Bit Driver	3.20.0128	Op
OpenLink Lite for Informix 5 (32bit)	Not marked	No
OpenLink Virtuoso Driver	Not marked	No
SQL Server	2.65.0240	Mi

2. Once you have selected the "OpenLink Generic 32 Bit Driver", you will be presented with the OpenLink ODBC Data Source Configuration dialog (depicted below).



Multi Tier ODBC Setup Version 3.20
Copyright 1994-1999 OpenLink Software
All rights reserved

Datasource
Name: Marketing
Comment:

OpenLink Server
Domain: Informix 7
Protocol: TCP/IP
Hostname: oplsco

Database
Name: stores7
Server:
Username: informix

Others
☐ Read-only connection
☐ No login dialog box
Row buffer size: 60

3. Configure the fields of the ODBC "System" or "User" DSN Configuration dialog in the manner explained below :-

Name: (Datasource) The name of the ODBC DSN, this is how you will interact with the OpenLink ODBC Driver from within ODBC compliant applications once your ODBC DSN has been created.

Comment: Additional information that further describes the ODBC DSN that you are creating.

Domain: This is how you pick the Database Engine Type that your ODBC DSN is to be associated with e.g. Informix 7, Oracle 7, Progress 7 etc.

Protocol: Select the network transport protocol used to connect your ODBC DSN with a remote database engine.

Hostname: Enter the hostname or IP address that identifies a Server Machine running OpenLink Server Components, that speak the OpenLink Data Access Protocol.

Name: (Database) This is how an actual database name within the Provider Type Domain is identified, for instance "stores7" indicates an "Informix 7" database called "stores7".

Server: This is where you place any database specific database connection options. This field in a majority of cases should be left blank by default.

Username: The default database UserID to used when logging on to a remote database engine (identified by the Provider Type above).

Read-only connection: Specify whether the connection is to be "Read-only". Make sure the checkbox is unchecked to request a "Read/Write" connection.

No Login Dialog Box: Suppress the ODBC "Username" and "Password" login dialog box when interacting with your ODBC DSN from within an ODBC compliant application.

Row Buffer Size: This attribute specifies the number of records to be transported over the network in a single network hop. Values can range from 1 to 99.

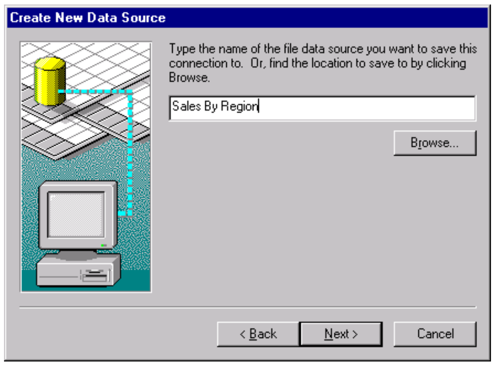
Creating a File DSN

This format of ODBC DSN enables the creation of centralized ODBC DSNs on a designated NT or Windows 95 Server machine, thereby reducing the administrative overhead of ODBC DSN configuration for every single machine within your infrastructure.

The steps for creating an ODBC File DSN are as follows: -

1. Select the File DSN Tab from the ODBC Driver Manger (see Screen Shot of Driver Manager ODBC DSN listing above)

2. Enter a name that describes your File DSN (e.g. "Sales Region " as depicted below) for future use when interacting with it from ODBC compliant application and environments.

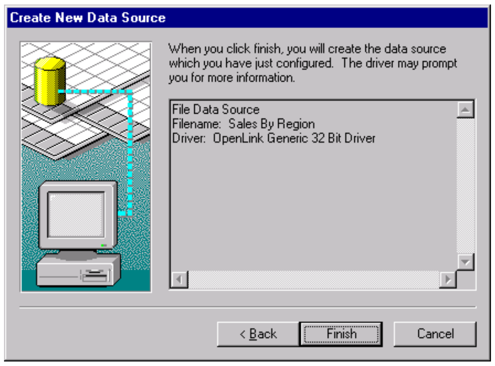

Create New Data Source

Type the name of the file data source you want to save this connection to. Or, find the location to save to by clicking Browse.

Sales By Region

Once you hit the "Next" button you will be presented with another dialog as depicted in the next screen shot.

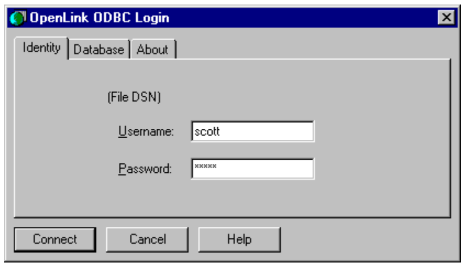
When you hit the "Finish" Button you will be presented with the OpenLink ODBC File DSN configuration dialog and then configure your File DSN as follows :



When you click finish, you will create the data source which you have just configured. The driver may prompt you for more information.

File Data Source
Filename: Sales By Region
Driver: OpenLink Generic 32 Bit Driver

3. Enter a Username and Password in the designated fields of the dialog depicted below.

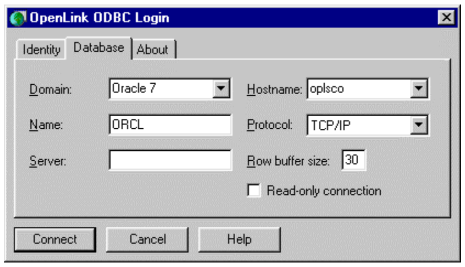


(File DSN)

Username: scott

Password: *****

4. Configure the fields of the ODBC File DSN Configuration dialog in the manner explained below :-



Domain: Oracle 7

Hostname: oplsco

Name: ORCL

Protocol: TCP/IP

Server:

Row buffer size: 30

☐ Read-only connection

Domain: This is how you pick the Database Engine Type that your ODBC DSN is to be associated with e.g. Informix 7, Oracle 7, Progress 7 etc.

Hostname: Enter the hostname or IP address that identifies a Server Machine running OpenLink Server Components, that speak the OpenLink Data Access Protocol.

Protocol: Select the network transport protocol used to connect your ODBC DSN with a remote database engine.

Name: This is how an actual database name within the Provider Type Domain is identified, for instance "stores7" indicates an "Informix 7" database called "stores7".

Server: This is where you place any database specific database connection options. This field in a majority of cases should be left blank by default.

Row Buffer Size: This attribute specifies the number of records to be transported over the network in a single network hop. Values can range from 1 to 99.

Read-only connection: Specify whether the connection is to be "Read-only". Make sure the checkbox is unchecked to request a "Read/Write" connection

Advanced Settings

The settings in this section have default values which are recommended in most situations.

If you run a complex query through the database it may take a long time before any data is actually returned. In this situation you may need to increase the ReceiveTimeout value, described below, only to cater for it. 2 Minutes (default) is usually more than adequate.

16bit Client Advanced Settings.

Client global configuration information is contained in the file OPENLINK.INI in the OpenLink Client installation directory, e.g c:\openlink.

The file is a Windows type INI file which can be opened in any text editor such as notepad. Please ensure that the file is saved in pure text only.

Important values that you may need to change are listed in the table below. The default values are recommended in most cases:

Section	Key	Description
[Communications]		
	BrokerTimeout	The amount of time the client will wait for a connection to an agent. The default value is 30. Value is measured in seconds.
	ReceivETimeout	The amount of time the client will wait for a query to return with data. The default value is 120. Value is measured in seconds.
	ShowErrors	With this set to yes any OpenLink messages will be displayed in a dialog box. If this is set to no, OpenLink messages will not be displayed.
[ServerTypes]		
	<Name>=	Each key name in this section represents a different provider type. To add a provider type simply add a new line using the same syntax as the default types.

32bit Client Advanced Settings.

Client global configuration information is contained in the sub section OPENLINK.INI of the registry. Windows 32bit platforms use the registry to store all configuration information. To edit or view to registry, run the '*regedit*' program. To view the OpenLink settings expand the following sections:

HKEY_LOCAL_MACHINE…
SOFTWARE…
ODBC…
OPENLINK.INI

Important values that you may need to change are listed in the table below. The default values are recommended in most cases:

Sub Section	Key	Description
Communications		
	BrokerTimeout	The amount of time the client will wait for a connection to an agent. The default value is 30. Value is measured in seconds.
	ReceiveTimeout	The amount of time the client will wait for a query to return with data. The default value is 120. Value is measured in seconds.
	ShowErrors	With this set to yes any OpenLink messages will be displayed in a dialog box. If this is set to no, OpenLink messages will not be displayed.
ServerTypes		
	<Name>=	Each key name in this section represents a different provider type. To add a provider type simply add a new line using the same syntax as the default types.

Server Components Installation

1. Download into a temporary directory on your local hard drive OpenLink Server components for Windows NT, using our product download selection wizard to choose one or more of the DBMS Engines that you intend access via ODBC/JDBC/UDBC
2. Extract the ZIP archive into your temporary installation directory
3. Run the program "setup.exe", and follow the on screen prompts
4. Proceed to create an ODBC, JDBC, or UDBC Data Source

Attempt a connection to the Data Source you have just created, from an ODBC, JDBC, or UDBC based application or environment.

OpenLink 3.2 Drivers for Java Database Connectivity (JDBC™)

OpenLink 3.2 for JDBC™ is a new release. Thus, the JDBC client and server components for JDBC™ aren't backward compatible with prior releases.

New JDBC Driver Packages

OpenLink now has a broader range of JDBC Driver types which also transcend JDBC versions.

The drivers are now categorized as follows:

- Generic Drivers for JDBC™ 1.0.2 (compatible with Java Virtual Machine version 1.0.2)
- Generic Drivers for JDBC™ 1.1.x (compatible with Java Virtual Machine version 1.1.x)
- Generic Drivers for JDBC™ 2.0 (compatible with Java Virtual Machine version 2.0)

- Megathin Drivers™ for JDBC™ 1.0.2 (compatible with Java Virtual Machine version 1.0.2.)
- Megathin Drivers™ for JDBC™ 1.1.x (compatible with Java Virtual Machine version 1.1.x)
- Megathin Drivers™ for JDBC™ 2.0 (compatible with Java Virtual Machine version 1.2 a.k.a. Java Virtual Machine 2.0)

The following table depicts how each of these drivers is packaged, what JDBC Driver formats are supported, and whether they are certified 100% Pure Java or Not:

Product Name	Java Package	JDBC Driver Name	JDBC Driver Formats	100% Pure Java ?
Generic Drivers for JDBC™ 1.0.2	opljdbc.jar	openlink.jdbc.Driver	1,2,3	N
Generic Drivers for JDBC™ 1.1.x	opljdbc.jar	openlink.jdbc.Driver	1,2,3	N
Generic Drivers™ for JDBC 2.0	opljdbc2.jar	openlink.jdbc2.Driver	1,2,3	N
Megathin Drivers™ for JDBC™ 1.0.2	megathin.jar	openlink.megathin.Driver	3	Y
Megathin Drivers™ for JDBC™ 1.1.x	megathin.jar	openlink.megathin.Driver	3	Y
Megathin Drivers™ for JDBC 2.0	megathin2.jar	openlink.megathin2.Driver	3	Y

New Features & Enhancements

1. **Scrollable Cursors**
   A new OpenLink RowSet class enables JDBC applications to take advantage of ODBC-style scrollable cursors functionality, with the ability to: retrieve rowsets, go to any row in the resultset or rowset, add new rows to the database, refresh and update any row with a single method invocation, lock and unlock any row, retrieve the current row number, as well as use ODBC-style bookmarks. This is an OpenLink extension to JDBC.

   The Drivers for JDBC™ 2.0 implement the Scrollable Cursor Interfaces introduced in JDBC™ 2.0. They also retain support for the OpenLink Scrollable Cursor extension so as to provide access to those Scrollable Cursor features that exist in ODBC but are missing from the JDBC™ 2.0 specification.

2. **Array Binding**
   As part of the new RowSet class. It is now possible to bind data arrays to the columns of the OpenLink RowSet object, and retrieve the data directly into the arrays with a single invocation of the RowSet.next() method. Please see the accompanying demonstration application for an example of its usage.

   This approach enables you to retrieve multiple records with a reduced number of calls to the RowSet.next() method. It basically improves JDBC Application performance.

3. **openlink.sql.Statement**
   This is a new OpenLink interface that extends java.sql.Statement to provide additional methods for configuring the ODBC-Style Scrollable Cursors functionality.

   You only need this functionality when working with the OpenLink Scrollable Cursor extensions. The Drivers for JDBC 2.0 implement similar features for the JDBC 2.0 Scrollable Cursor specification.

4. **Security**
   The OpenLink client and server components for JDBC™ encrypt the data sent across the network between the client and server. This provides for enhanced security, particularly over a WAN. This is transparent to the application, and there are no driver specific properties for the application to set.

5. **Version self-checking**
   The OpenLink client component for JDBC™ now cross checks version numbers with the server Agent for JDBC at connect time, this ensures that compatible components always in use. An exception will be thrown when incompatibilities are encountered, this exception will also contain information about the component versions relating to the exception.

6. **Easier Client Version checking**

There is now an easier way of checking the OpenLink client for JDBC™ version. Make sure that the driver is in the classpath and then at your command prompt enter the command:

```
java openlink.jdbc.Driver  (for JDK 1.x Drivers)
```

or

```
java openlink.jdbc2.Driver (for JDK 1.2/2.0 Drivers)
```

or

```
java openlink.megathin.Driver (for 100% Pure Java Drivers for JDK 1.x)
```

or

```
java openlink.megathin2.Driver (for 100% Pure Java Drivers for JDK 1.2/2.0)
```

The driver will respond with appropriate version and release number information.

7. **OpenLink Native (UDBC) Bridge for JDBC™**
   UDBC is OpenLink's Universal Database Connectivity solution for cross platform data access. The new OpenLink Client UDBC Bridge for JDBC™ enables JDBC binding to Native Database Interfaces without going through an ODBC Driver Manager.

OpenLink Drivers for JDBC™ Installation & Configuration

Introduction

The OpenLink Drivers for JDBC™ enable the development, deployment, and utilization of database independent Java Applications, Applets, Servlets, and Bean Components (collectively called JDBC Clients) that conform to the JDBC 1.0.2, JDBC 1.1.x, or JDBC 2.0.x specifications from JavaSoft.

JDBC clients are built by importing the "java.sql.*" collection of classes known as the JDBC Driver Manager interface. The JDBC Driver Manager uses JDBC URLs to link JDBC clients with JDBC Drivers. It is important to note that JDBC URLs are JDBC Driver specific. Detailed information regarding JDBC is available from: http://java.sun.com/products/jdbc/index.html

Downloading Driver Software

The OpenLink Drivers for JDBC are packaged either as a bundle alongside the other OpenLink data access drivers (ODBC, UDBC, and OLE-DB) that make up the OpenLink Universal Data Access Driver Suite or as a separate release archive which contains only the OpenLink Megathin Drivers for JDBC, a 100% pure Java Driver for JDBC.

When you download the drivers as part of the data access driver suite bundle three driver types for JDBC are available to you:

- OpenLink Driver for JDBC Type 1 (JDBC-ODBC Bridge)
- OpenLink Driver for JDBC Type 2 (JDBC-Native Bridge)
- OpenLink Driver for JDBC Type 3 (Network enabled all Java Driver)

When you download the OpenLink Megathin Drivers for JDBC you only get a very thin 100% Pure Java Type 3 Driver for JDBC known as the OpenLink Megathin Driver for JDBC.

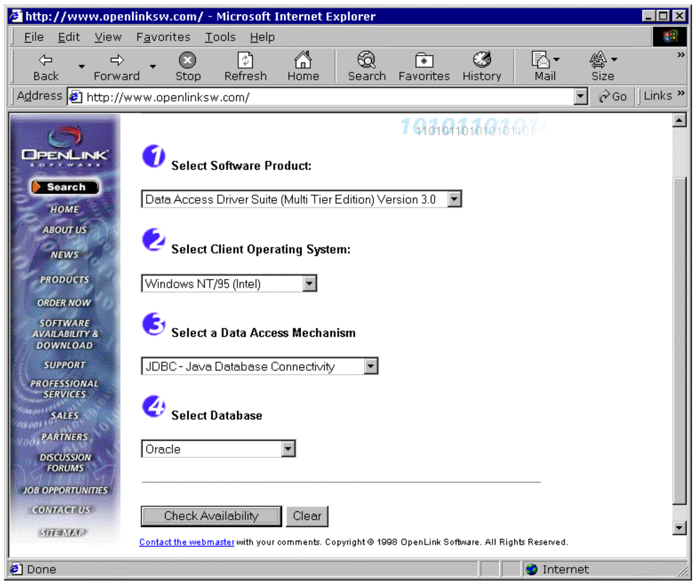
OpenLink Web Download Wizard Interaction

If you aren't installing these Drivers from a CD you would have to visit the OpenLink Web Site's download page to obtain these Drivers.

The screen shots that follow depict the OpenLink download wizard interaction that is required in order to download either the JDBC Driver bundle or the standalone Megathin Drivers.

Download Wizard Interaction for obtaining OpenLink Drivers for JDBC Bundle

1. Select a Client Operating System from the "Select Client Operating System" listbox and then select a database engine that you will be connecting to via your Driver for JDBC using the "Select Database" listbox.


http://www.openlinksw.com/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://www.openlinksw.com/
Go Links
Search HOME ABOUT US NEWS PRODUCTS ORDER NOW SOFTWARE AVAILABILITY & DOWNLOAD SUPPORT PROFESSIONAL SERVICES SALES PARTNERS DISCUSSION FORUMS JOB OPPORTUNITIES CONTACT US SITE MAP
Check Availability
Clear
Done
Internet

1 Select Software Product: Data Access Driver Suite (Multi Tier Edition) Version 3.0

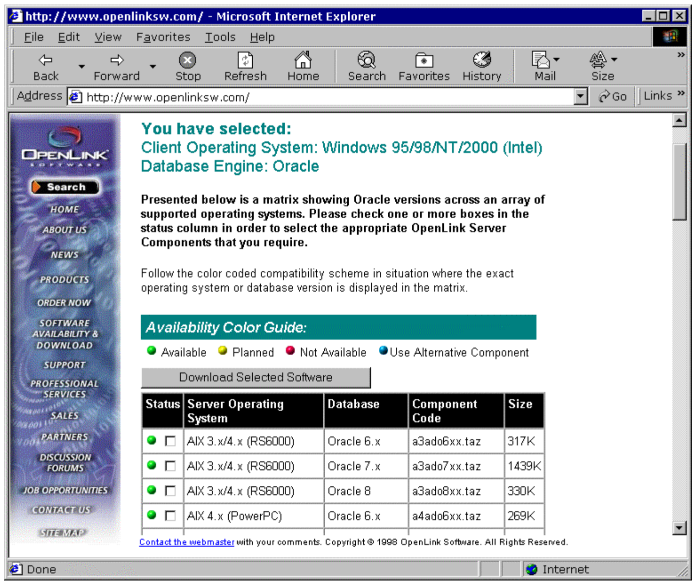
2 Select Client Operating System: Windows NT/95 (Intel)

3 Select a Data Access Mechanism: JDBC - Java Database Connectivity

4 Select Database: Oracle

Contact the webmaster with your comments. Copyright © 1998 OpenLink Software. All Rights Reserved.

2. Pick one or more server components matching the server operating system that will host the OpenLink Server components required by the Drivers for JDBC. Then click on the "Download Selected Software" button.


http://www.openlinksw.com/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://www.openlinksw.com/
Go Links
Search HOME ABOUT US NEWS PRODUCTS ORDER NOW SOFTWARE AVAILABILITY & DOWNLOAD SUPPORT PROFESSIONAL SERVICES SALES PARTNERS DISCUSSION FORUMS JOB OPPORTUNITIES CONTACT US SITE MAP
Download Selected Software
Done
Internet

You have selected:
Client Operating System: Windows 95/98/NT/2000 (Intel)
Database Engine: Oracle

Presented below is a matrix showing Oracle versions across an array of supported operating systems. Please check one or more boxes in the status column in order to select the appropriate OpenLink Server Components that you require.

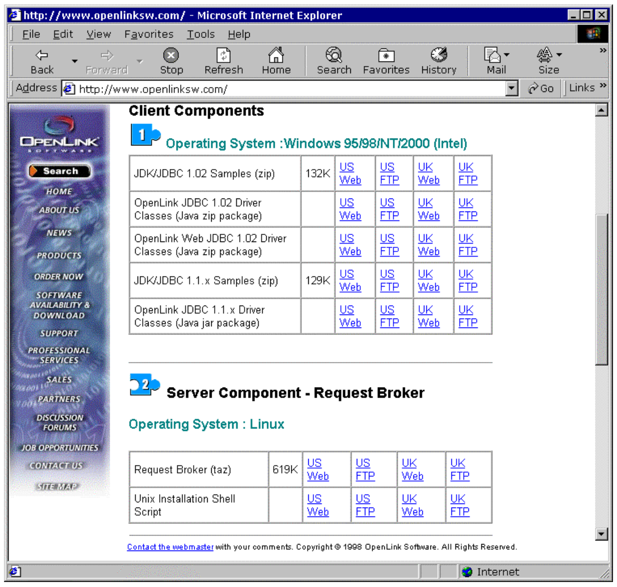
Follow the color coded compatibility scheme in situation where the exact operating system or database version is displayed in the matrix.

Availability Color Guide:
● Available ● Planned ● Not Available ● Use Alternative Component

Status	Server Operating System	Database	Component Code	Size
● ☐	AIX 3.x/4.x (RS6000)	Oracle 6.x	a3ado6xx.taz	317K
● ☐	AIX 3.x/4.x (RS6000)	Oracle 7.x	a3ado7xx.taz	1439K
● ☐	AIX 3.x/4.x (RS6000)	Oracle 8	a3ado8xx.taz	330K
● ☐	AIX 4.x (PowerPC)	Oracle 6.x	a4ado6xx.taz	269K

Contact the webmaster with your comments. Copyright © 1998 OpenLink Software. All Rights Reserved.

3. Download all the software components presented in the "Software Download" page.


http://www.openlinksw.com/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://www.openlinksw.com/ Go Links
OPENLINK SOFTWARE
Search
HOME
ABOUT US
NEWS
PRODUCTS
ORDER NOW
SOFTWARE AVAILABILITY & DOWNLOAD
SUPPORT
PROFESSIONAL SERVICES
SALES
PARTNERS
DISCUSSION FORUMS
JOB OPPORTUNITIES
CONTACT US
SITE MAP
Internet

Client Components

1 Operating System :Windows 95/98/NT/2000 (Intel)

JDK/JDBC 1.02 Samples (zip)	132K	US Web	US FTP	UK Web	UK FTP
OpenLink JDBC 1.02 Driver Classes (Java zip package)		US Web	US FTP	UK Web	UK FTP
OpenLink Web JDBC 1.02 Driver Classes (Java zip package)		US Web	US FTP	UK Web	UK FTP
JDK/JDBC 1.1.x Samples (zip)	129K	US Web	US FTP	UK Web	UK FTP
OpenLink JDBC 1.1.x Driver Classes (Java jar package)		US Web	US FTP	UK Web	UK FTP

2 **Server Component - Request Broker**

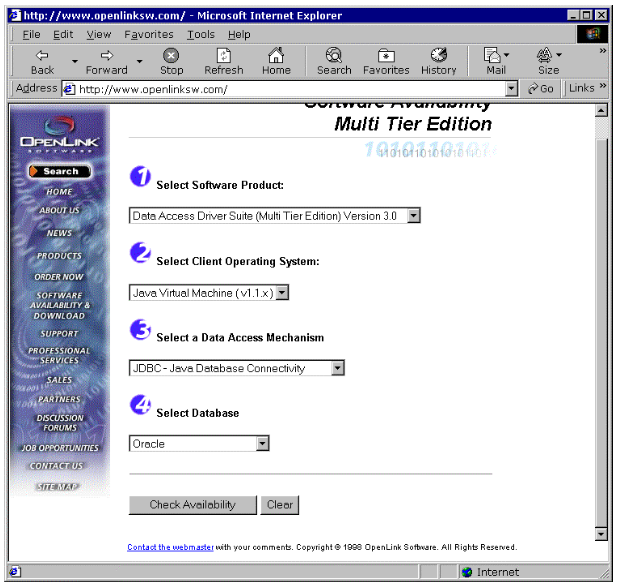
Operating System : Linux

Request Broker (taz)	619K	US Web	US FTP	UK Web	UK FTP
Unix Installation Shell Script		US Web	US FTP	UK Web	UK FTP

Contact the webmaster with your comments. Copyright © 1998 OpenLink Software. All Rights Reserved.

Download Wizard Interaction for obtaining OpenLink Megathin Drivers for JDBC

1. Select a Java Virtual Machine version from the "Select Client Operating System" listbox and then select a database engine that you will be connecting to via your Driver for JDBC using the "Select Database" listbox.


http://www.openlinksw.com/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://www.openlinksw.com/ Go Links
OPENLINK SOFTWARE
Search
HOME
ABOUT US
NEWS
PRODUCTS
ORDER NOW
SOFTWARE AVAILABILITY & DOWNLOAD
SUPPORT
PROFESSIONAL SERVICES
SALES
PARTNERS
DISCUSSION FORUMS
JOB OPPORTUNITIES
CONTACT US
SITE MAP
Check Availability
Clear
Internet

Software Availability
Multi Tier Edition

1 Select Software Product: Data Access Driver Suite (Multi Tier Edition) Version 3.0

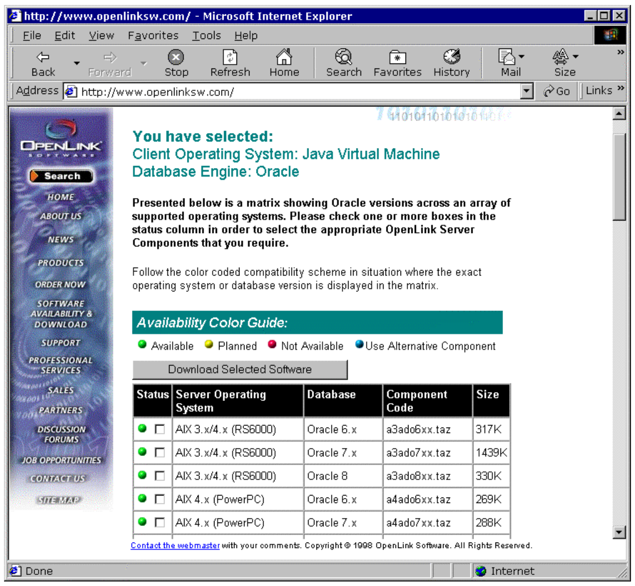
2 Select Client Operating System: Java Virtual Machine (v1.1.x)

3 Select a Data Access Mechanism: JDBC - Java Database Connectivity

4 Select Database: Oracle

Contact the webmaster with your comments. Copyright © 1998 OpenLink Software. All Rights Reserved.

2. Pick one or more server components matching the server operating system that will host the OpenLink Server components required by the Drivers for JDBC. Then click on the "Download Selected Software" button.


http://www.openlinksw.com/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://www.openlinksw.com/
Go Links
Search
HOME
ABOUT US
NEWS
PRODUCTS
ORDER NOW
SOFTWARE AVAILABILITY & DOWNLOAD
SUPPORT
PROFESSIONAL SERVICES
SALES
PARTNERS
DISCUSSION FORUMS
JOB OPPORTUNITIES
CONTACT US
SITE MAP
Done
Internet

You have selected:
Client Operating System: Java Virtual Machine
Database Engine: Oracle

Presented below is a matrix showing Oracle versions across an array of supported operating systems. Please check one or more boxes in the status column in order to select the appropriate OpenLink Server Components that you require.

Follow the color coded compatibility scheme in situation where the exact operating system or database version is displayed in the matrix.

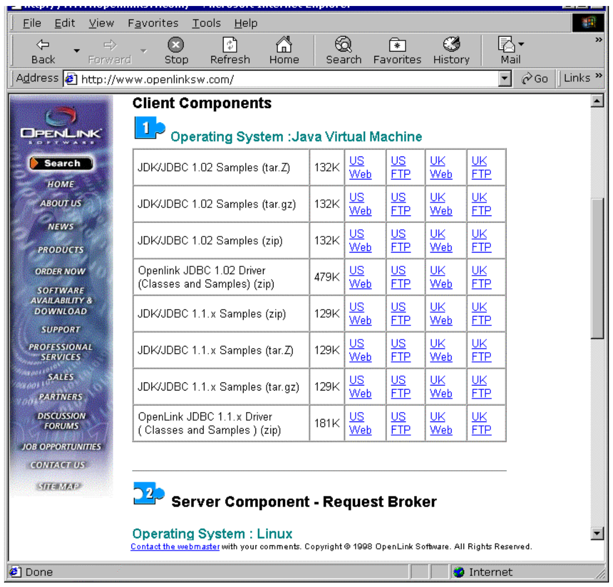
Availability Color Guide:
● Available ● Planned ● Not Available ● Use Alternative Component

Download Selected Software

Status	Server Operating System	Database	Component Code	Size
● ☐	AIX 3.x/4.x (RS6000)	Oracle 6.x	a3ado6xx.taz	317K
● ☐	AIX 3.x/4.x (RS6000)	Oracle 7.x	a3ado7xx.taz	1439K
● ☐	AIX 3.x/4.x (RS6000)	Oracle 8	a3ado8xx.taz	330K
● ☐	AIX 4.x (PowerPC)	Oracle 6.x	a4ado6xx.taz	269K
● ☐	AIX 4.x (PowerPC)	Oracle 7.x	a4ado7xx.taz	288K

Contact the webmaster with your comments. Copyright © 1998 OpenLink Software. All Rights Reserved.

3. Download all the software components presented in the "Software Download" page.


File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://www.openlinksw.com/
Go Links
Search
HOME
ABOUT US
NEWS
PRODUCTS
ORDER NOW
SOFTWARE AVAILABILITY & DOWNLOAD
SUPPORT
PROFESSIONAL SERVICES
SALES
PARTNERS
DISCUSSION FORUMS
JOB OPPORTUNITIES
CONTACT US
SITE MAP
Done
Internet

Client Components

1 Operating System :Java Virtual Machine

JDK/JDBC 1.02 Samples (tar.Z)	132K	US Web	US FTP	UK Web	UK FTP
JDK/JDBC 1.02 Samples (tar.gz)	132K	US Web	US FTP	UK Web	UK FTP
JDK/JDBC 1.02 Samples (zip)	132K	US Web	US FTP	UK Web	UK FTP
Openlink JDBC 1.02 Driver (Classes and Samples) (zip)	479K	US Web	US FTP	UK Web	UK FTP
JDK/JDBC 1.1.x Samples (zip)	129K	US Web	US FTP	UK Web	UK FTP
JDK/JDBC 1.1.x Samples (tar.Z)	129K	US Web	US FTP	UK Web	UK FTP
JDK/JDBC 1.1.x Samples (tar.gz)	129K	US Web	US FTP	UK Web	UK FTP
OpenLink JDBC 1.1.x Driver (Classes and Samples) (zip)	181K	US Web	US FTP	UK Web	UK FTP

2 **Server Component - Request Broker**

Operating System : Linux

Contact the webmaster with your comments. Copyright © 1998 OpenLink Software. All Rights Reserved.

OpenLink Drivers for JDBC Installation & Configuration

Once you have downloaded your OpenLink drivers for JDBC using the instructions provided above, the next step in the process is the actual configuration of these drivers for use within your operating environment.

Java is operating system independent by virtue of its core philosophy, but JDBC Drivers may or may not be operating system independent as this is JDBC Driver format and implementation specific. The sections that follow walk your through the OpenLink Driver for JDBC installation and configuration process.

Windows 95/98/NT/2000 Based Local Client-Server Environment

In this scenario your Windows machine is acting as the host machine for both your OpenLink client and server components, implying that you are going to install your OpenLink Client and Server components for JDBC on the same machine.

Installation Process

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard

2. As Windows 95/98/NT/200 is playing the dual role of both Client and Server machine for your OpenLink components, you would have downloaded a ZIP archive that contains both the OpenLink Client & Server components for this platform. Extract the contents of this ZIP archive to a temporary installation folder and then run the "Setup.exe" program

3. The archive you have downloaded will contain the entire suite of Data Access Drivers for this platform. If you do not require the OpenLink ODBC or OLE-DB Drivers simply uncheck these components using the installers component list dialog when presented during the install process

4. The installer will automatically determine what version of the Java Virtual Machine is installed on your machine and then automatically checks which OpenLink Drivers for JDBC java classes should be checked for installation by default. You can override this settings during the installation process so as to match your specific requirements should they differ from those derived by the installer.

5. The installer will also add the OpenLink Driver for JDBC class files that you have selected in step 3 to the CLASSPATH environment variable on your system

6. Reboot your system

7. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Generic Driver for JDBC 1.0.2	java openlink.jdbc.Driver
Generic Driver for JDBC 1.1.x	java openlink.jdbc.Driver
Generic Driver for JDBC 1.2.x or 2.x	java openlink.jdbc2.Driver
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

   If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

8. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000/

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

9. An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent

inf7_sv -? : this will verify the Informix Database Agent
ing7_sv -? : this will verify the Ingres II Database Agent
db2_sv -? : this will verify the IBM DB2 Database Agent

See the detailed section about OpenLink Database Agents for additional information.

Windows 95/98/NT/2000 Based Client-Server (2-Tier Configuration) Environment

In this scenario one or more Windows machines act as the host machine for your OpenLink client components, while a separate Windows server machine hosts your OpenLink server components. This Windows server machine also hosts the database engine that you will be connecting to via JDBC, this machine is typically referred to as your Database Server machine.

Client Components Installation

1. Download appropriate driver software using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard on to your designated client machine

2. As Windows 95/98/NT/200 is playing the single role of Client machine for your OpenLink Drivers for JDBC, you would have downloaded a ZIP archive that contains only the OpenLink Client components. Extract the contents of this ZIP archive to a temporary installation folder and then run the "Setup.exe" program

3. The archive you have downloaded will contain the entire suite of Data Access Drivers for this platform. If you do not require the OpenLink ODBC or OLE-DB Drivers simply uncheck these components using the installers component list dialog when presented during the install process

4. The installer will automatically determine what version of the Java Virtual Machine is installed on your machine and then automatically checks which OpenLink Drivers for JDBC java classes should be checked for installation by default. You can override this settings during the installation process so as to match your specific requirements should they differ from those derived by the installer

5. The installer will also add the OpenLink Driver for JDBC class files that you have selected in step 3 to the CLASSPATH environment variable on your system

6. Reboot your system

7. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	**Verification Command**
Generic Driver for JDBC 1.0.2	java openlink.jdbc.Driver
Generic Driver for JDBC 1.1.x	java openlink.jdbc.Driver
Generic Driver for JDBC 1.2.x or 2.x	java openlink.jdbc2.Driver
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

Server Components Installation

1. Download appropriate server components software using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard on to your designated server machine

2. As Windows 95/98/NT/200 is playing the single role of Server machine for your OpenLink Drivers for JDBC, you would have downloaded a ZIP archive that contains only the OpenLink Server components. Extract the contents of this ZIP archive to a temporary installation folder and then run the "Setup.exe" program

3. The archive you have downloaded will contain both OpenLink client and Server components for this platform. Since you

are setting up a Server machine simply uncheck the OpenLink Client components (ODBC, JDBC, OLE-DB) using the installers component list dialog when presented during the install process, this ensures that you only install OpenLink Server components on your Server machine(s)

4. If you are an existing OpenLink user please ensure that you do not have an OpenLink Request Broker process running (check your services   control panel item), if there is a Request Broker process running please shut it down at this point

5. Run the "setup.exe" program

6. Start the OpenLink Request Broker, you this by either going into your "Services" control panel (for Windows NT) or to the "OpenLink Data Access Drivers" Windows Start Menu, and then click on the "Broker Startup" menu item

7. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session from either your OpenLink Client or Server machine and then enter one of the following URLs:

   From Client Machine: http://<server name or IP address>:8000

   From Server Machine: http://localhost:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

8. An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Windows 95/98/NT/2000 Based Application-Server (3-Tier Configuration) Environment

In this scenario your OpenLink Client and Server components for JDBC are installed on an Application Server, as this is where your JDBC based application will be hosted and developed (if you are building a JDBC based 3-Tier solution). Thus, the installation process is broken down into two parts, Application Server, and Database Server components installation. You will not need to install any software on the client machines being used by your JDBC solutions end-users.

Application Server Components Installation

1. Download appropriate server components software using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard on to your designated server machine

2. As this machine needs to host both Client and Server components (by virtue of this machine playing the role of Application Server), you would have downloaded a ZIP archive that contains both the OpenLink Client & Server components for this platform. Extract the contents of this ZIP archive to a temporary installation folder and then run the "Setup.exe" program

3. The archive you have downloaded will contain the entire suite of Data Access Drivers for this platform. If you do not require the OpenLink ODBC or OLE-DB Drivers simply uncheck these components using the installers component list dialog when presented during the install process.

4. If you choose to use OpenLink's Database Independent Networking to connect to remote database engines hosted on one

or more dedicated Database Server machine, then ensure that an OpenLink Database Agent checkbox for each Database Engine type is checked from the component list presented by the installer. If on the other hand you choose to use Database Specific Networking provided by your database vendor(s) when connecting to your remote Database Engine(s) hosted on your dedicated Database Server machines, then simply leave all the OpenLink Database Agent checkboxes unchecked.

SQL*Net, Open Client, Progress Client. I-Connect, Ingres Net, and Netlib are database specific networking products for Oracle, Sybase, Progress, Informix, Ingres, and Microsoft SQL Server respectively.

5. The installer will automatically determine what version of the Java Virtual Machine is installed on your machine and then automatically checks which OpenLink Drivers for JDBC java classes should be checked for installation by default. You can override this settings during the installation process so as to match your specific requirements should they differ from those derived by the installer

6. The installer will also add the OpenLink Driver for JDBC class files that you have selected in step 3 to the CLASSPATH environment variable on your system

7. Reboot your system

8. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Generic Driver for JDBC 1.0.2	java openlink.jdbc.Driver
Generic Driver for JDBC 1.1.x	java openlink.jdbc.Driver
Generic Driver for JDBC 1.2.x or 2.x	java openlink.jdbc2.Driver
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

9. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session from either your OpenLink Client or Server machine and then enter one of the following URLs:

From Client Machine: http://<server name or IP address>:8000

From Server Machine: http://localhost:8000

If you are presented with the Home Page of the "OpenLink Admin Assistant" then this confirms that your OpenLink Server environment is also correctly setup.

An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run the following commands:

jdbc_sv -? - this will verify the default JDBC Agent
jodbc_sv -? - This will verify the JDBC-ODBC Agent
judbc_sv -? - This will verify the JDBC-UDBC Agent

10. If you are going to be connecting to your remote database servers using database specific networking provided by one or more database vendors then you need to perform an additional check to ensure that your database agent have been installed properly. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

ora8_sv -? :this will verify the Oracle Database Agent
pro83a_sv -? :this will verify the Progress Database Agent
sql6_sv -? :this will verify the Microsoft SQL Server Database Agent
syb10_sv -? : this will verify the Sybase Database Agent
db2_sv -? : this will verify the IBM DB2 Database Agent

See the detailed section about OpenLink Database Agents for additional information.

Database Server Components Installation

This step is only required if your are connecting your Application Server components installed in the prior section to a remote database engine hosted on a dedicated Database Server machine using OpenLink's Database Independent Networking.

1. Download appropriate server components software using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard on to your designated server machine

2. As Windows 95/98/NT/2000 is playing the single role of a dedicated Database Server machine for your OpenLink Application Server components for JDBC, you would have downloaded a ZIP archive that contains only the OpenLink Server components. Extract the contents of this ZIP archive to a temporary installation folder and then run the "Setup.exe" program

3. The archive you have downloaded will contain both OpenLink client and Server components for this platform. Since you are setting up a Server machine simply uncheck the OpenLink Client components (ODBC, JDBC, OLE-DB) using the installers component list dialog when presented during the install process, this ensures that you only install OpenLink Server components on your Server machine(s)

4. Ensure that an OpenLink Database Agent checkbox for each Database Engine type is checked from the component list presented by the installer. Uncheck all JDBC component related checkboxes unless you anticipate using this dedicated Database Server as an Application Server at a later date.

5. If you are an existing OpenLink user please ensure that you do not have an OpenLink Request Broker process running (check your services control panel item), if there is a Request Broker process running please shut it down at this point

6. Run the "setup.exe" program

7. Start the OpenLink Request Broker, you do this by either going into your "Services" control panel (for Windows NT) or to the "OpenLink Data Access Drivers" Windows Start Menu, and then click on the "Broker Startup" menu item

8. Verify your OpenLink Database server components installation by starting a Web Browser session from either your OpenLink Client, Application or Server machine and then enter one of the following URLs:

   From Client or Application Server Machine: http://<server name or IP address>:8000

   From Database Server Machine: http://localhost:8000

   If you are presented with the Home Page of the "OpenLink Admin Assistant" then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ```
   ora8_sv -?  :this will verify the Oracle Database Agent
   pro83a_sv  -? :this will verify the Progress Database Agent
   sql6_sv -?  :this will verify the Microsoft SQL Server Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent
   ```

   See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Based Local Client-Server Configuration

In this scenario your Linux or UNIX machine is acting as the host machine for both your OpenLink client and server components, implying that you are going to install your OpenLink Client and Server components for JDBC on the same machine.

Installation Process

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Move the Request Broker and Database Agent archives into a temporary installation folder on your Linux or UNIX machine then run the following commands from the command line prompt:

   Linux:
   ```
   rpm -ivh openlink-3.2-2.rpm
   ```

```
rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
or
rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)
```

Linux (if your Linux system does not have the RPM facility) and UNIX:

```
sh install.sh
```

3. Follow the instructions presented by the installer for configuring your OpenLink Database Agents, if you installed via a Linux RPM archive, post RPM installation you will need to run the "oplcfg" located in the "openlink/bin" sub-directory of your OpenLink base installation directory

4. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation's base installation directory. This files contains the following entries which you can modify so as to match the OpenLink Drivers for JDBC to the appropriate Java environment on your machine:

```
#CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.0.2/opljdbc.zip
CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.1.x/opljdbc.jar
#CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.2.x/opljdbc2.zip
```

5. Run the script "openlink.sh" (you may also want to add a reference to this in your .profile file) by executing the following command from your Linux or UNIX command line prompt:

```
. openlink.sh
```

6. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Generic Driver for JDBC 1.0.2	java openlink.jdbc.Driver
Generic Driver for JDBC 1.1.x	java openlink.jdbc.Driver
Generic Driver for JDBC 1.2.x or 2.x	java openlink.jdbc2.Driver
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

7. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

http://localhost:8000 or http://<hostname of current machine>:8000

If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

```
jdbc_sv -? - this will verify the default JDBC Agent
jodbc_sv -? - This will verify the JDBC-ODBC Agent
judbc_sv -? - This will verify the JDBC-UDBC Agent
```

8. An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

```
ora8_sv -?  :this will verify the Oracle Database Agent
pro83a_sv  -? :this will verify the Progress Database Agent
```

syb10_sv -? : this will verify the Sybase Database Agent
inf7_sv -? : this will verify the Informix Database Agent
ing7_sv -? : this will verify the Ingres II Database Agent
db2_sv -? : this will verify the IBM DB2 Database Agent

See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Based Client-Server (2-Tier Configuration) Installation

In this scenario one or more Linux or UNIX machines act as the host machine for your OpenLink client components, while a separate Linux or UNIX server machine hosts your OpenLink server components. This Linux or UNIX server machine also hosts the database engine that you will be connecting to via JDBC, this machine is typically referred to as your Database Server machine.

Client Components Installation

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard

2. Although Linux or UNIX is only playing role of both Client machine for your OpenLink components, you still need to download a Linux RPM or a UNIX compressed TAR archive containing the OpenLink Request Broker (the download page clearly identifies this archive), this contains both the OpenLink Request Broker and the OpenLink Driver for JDBC components. Move this archive to a temporary installation folder and then run the following installation programs:

   Linux:

   ```
   rpm -ivh openlink-3.2-2.rpm
   ```

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   ```
   sh install.sh
   ```

3. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation base installation directory. This files contains the following entry which you can modify so as to match the OpenLink Drivers for JDBC to the appropriate Java environment on your machine:

   ```
   #CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.0.2/opljdbc.zip
   CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.1.x/opljdbc.jar
   #CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.2.x/opljdbc2.zip
   ```

4. Run the script "openlink.sh" (you may also want to add a reference to this in your .profile file) by executing the following command from your Linux or UNIX command line prompt:

   ```
   . openlink.sh
   ```

5. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Generic Driver for JDBC 1.0.2	java openlink.jdbc.Driver
Generic Driver for JDBC 1.1.x	java openlink.jdbc.Driver
Generic Driver for JDBC 1.2.x or 2.x	java openlink.jdbc2.Driver
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

   If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

6. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000 or http://<hostname of current machine>:8000

If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

jdbc_sv -? - this will verify the default JDBC Agent
jodbc_sv -? - This will verify the JDBC-ODBC Agent
judbc_sv -? - This will verify the JDBC-UDBC Agent

7. An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Database Server Components Installation

Only perform these steps if you are connecting to database engines hosted on your dedicated Database Server using OpenLink's Database Independent Networking:

1. Download appropriate server components installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Move the Request Broker and Database Agent archives into a temporary installation folder on your Database Server machine then run the following commands from the command line prompt:

   Linux:
   rpm -ivh openlink-3.2-2.rpm
   rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
   or
   rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   sh install.sh

3. Follow the instructions presented by the installer for configuring your OpenLink Database Agents

4. An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Based Application-Server (3-Tier Configuration) Installation

In this scenario your OpenLink Client machine plays the role of an Application Server, as this is where your JDBC based application will be hosted and developed (if you are building a 3-Tier JDBC solution). Thus, the installation process is broken down into two parts, Application Server, and Database Server components installation. You will not need to install any software on the machines being used by your JDBC solution's end-users.

Application Server Components Installation

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Although Linux or UNIX is only playing role of Client machine for your OpenLink components, you still need to download Linux RPMs or a UNIX compressed TAR archives containing the OpenLink Request Broker and the Database Agents for each database engine that you will be connecting to via JDBC (the download page clearly identifies these archives).

3. Move the Request Broker and this archive to a temporary installation folder, if you choose to use OpenLink's Database Independent Networking to connect to remote database engines hosted on one or more dedicated Database Server machines, do not move the Database Agent archives into the temporary installation directory on the Application Server. Run the following installation programs from the temporary installation directory on your Application Server machine:

   Linux:

   ```
   rpm -ivh openlink-3.2-2.rpm
   ```

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   ```
   sh install.sh
   ```
   Linux:

   ```
   rpm -ivh openlink-3.2-2.rpm
   ```

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   ```
   sh install.sh
   ```

   * Ignore the Database Agent configuration menu when presented to you by the installer. *

   If on the other hand you choose to use Database Specific Networking provided by your database vendor(s) when connecting to your remote Database Engine(s) hosted on your dedicated Database Server machines, then then move each Database Agent archive into a temporary installation directory alongside the Request Broker archive and then run the following installation programs:

   Linux:
   ```
   rpm -ivh openlink-3.2-2.rpm
   rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
   or
   rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)
   ```

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   ```
   sh install.sh
   ```

4. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation base installation directory. This files contains the following entry which you can modify so as to match the OpenLink Drivers for JDBC to the appropriate Java environment on your machine:

   ```
   #CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.0.2/opljdbc.zip
   CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.1.x/opljdbc.jar
   #CLASSPATH=$CLASSPATH:/dbs/openlink/v32/openlink/jdk1.2.x/opljdbc2.zip
   ```

5. Run the script "openlink.sh" (you may also want to add a reference to this in your .profile file) by executing the following command from your Linux or UNIX command line prompt:

   ```
   . openlink.sh
   ```

6. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Generic Driver for JDBC 1.0.2	java openlink.jdbc.Driver
Generic Driver for JDBC 1.1.x	java openlink.jdbc.Driver
Generic Driver for JDBC 1.2.x or 2.x	java openlink.jdbc2.Driver
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver

Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

7. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000 or http://<hostname of current machine>:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

8. Verify your OpenLink Database server components installation by starting a Web Browser session from either your OpenLink Client, Application or Server machine and then enter one of the following URLs:

   From Client Machine: http://<server name or IP address>:8000

   From Database Server Machine: http://localhost:8000

   If you are presented with the Home Page of the "OpenLink Admin Assistant" then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation directory. Then run the following commands:

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Database Server Components Installation

Only perform these steps if you are connecting to database engines hosted on your dedicated Database Server using OpenLink's Database Independent Networking:

1. Download appropriate server components installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Move the Request Broker and Database Agent archives into a temporary installation folder on your Database Server machine then run the following commands from the command line prompt:

   Linux:
   rpm -ivh openlink-3.2-2.rpm
   rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
   or
   rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   sh install.sh

3. Follow the instructions presented by the installer for configuring your OpenLink Database Agents

4. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation's base installation directory.

5. Run the script "openlink.sh" (you may also want to add a reference to this in your .profile file) by executing the following command from your Linux or UNIX command line prompt:

   ```
   . openlink.sh
   ```

6. Verify your OpenLink Database server components installation by starting a Web Browser session from either your OpenLink Client, Application or Server machine and then enter one of the following URLs:

   From Client or Application Server Machine: http://<server name or IP address>:8000

   From Database Server Machine: http://localhost:8000

   If you are presented with the Home Page of the "OpenLink Admin Assistant" then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation directory. Then run the following commands:

   ```
   ora8_sv -?  :this will verify the Oracle Database Agent
   pro83a_sv  -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent
   ```

   See the detailed section about OpenLink Database Agents for additional information.

Java Based Local Client-Server

In this scenario the Java Virtual Machine is acting as the host of your OpenLink client component for JDBC (a 100% Pure Java Driver for JDBC). The operating system hosting your Java Virtual Machine, also hosts the OpenLink Server server components for JDBC. Thus, you are going to install your OpenLink Client and Server components for JDBC on the same machine.

Client Components Installation Process

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard . You would have selected "Java Virtual Machine" as you client operating system when interacting with the OpenLink download Wizard and then have the files "megathin.jar" or "megathin2.jar" presented in the download results page depending on the version of the Java Virtual Machine selected

2. Place the "megathin.jar" or "megathin2.jar" file into directory of your choice then add the directory and reference to the JAR file to your CLASSPATH environment variable. See example below:

   Windows 95/98/NT/2000

   Presuming you place the "megathin.jar" file in the "\program files\openlink\jdk11" on your Windows machine, you would add the following line to your "autoexec.bat" if you are running Windows 95/98:

   ```
   set CLASSPATH=%CLASSPATH%;"c:\program files\openlink\jdk11\megathin.jar":.
   ```

   If you are using NT or Windows 2000 " then you need to open the "System Environment" properties of the "System" Control Panel applet and then add the same entry to the "System Variables" section if you want the driver to be accessible to all users, if not place the entry in the "User Variables" section.

   Linux or UNIX

   Presuming you place the "megathin.jar" file in the "/opt/openlink/jdk11" on your Linux or UNIX machine, you would need to modify the following line in the file "openlink.sh" so that they match what is listed below:

   ```
   CLASSPATH=$CLASSPATH:/opt/openlink/jdk1.1.x/megathin.jar
   ```

Server Components Installation

Windows 95/98/NT/2000

1. As Windows 95/98/NT/200 is playing the dual role of both Client and Server machine for your OpenLink components, you

would have downloaded a ZIP archive that contains both the OpenLink Client & Server components for this platform. Extract the contents of this ZIP archive to a temporary installation folder and then run the "Setup.exe" program

2. The archive you have downloaded will contain the entire suite of Data Access Drivers for this platform. If you do not require the OpenLink ODBC or OLE-DB Drivers simply uncheck these components using the installers component list dialog when presented during the install process.

3. Reboot your system

4. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

5. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run the following commands:

   ```
   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent
   ```

6. Verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run the following commands:

   ```
   ora8_sv -?  :this will verify the Oracle Database Agent
   pro83a_sv  -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent
   ```

   See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Server Components Installation

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Move the Request Broker and Database Agent archives into a temporary installation folder on your Linux or UNIX machine then run the following commands from the command line prompt:

   Linux:
   ```
   rpm -ivh openlink-3.2-2.rpm
   rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
   or
   rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)
   ```

   Linux (if your Linux system does not have the RPM facility) and UNIX:

```
sh install.sh
```

3. Follow the instructions presented by the installer for configuring your OpenLink Database Agents, if you installed via a Linux RPM archive, post RPM installation you will need to run the "oplcfg" located in the "openlink/bin" sub-directory of your OpenLink base installation directory

4. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation's base installation directory. This files contains the following entries which you can modify so as to match the OpenLink Drivers for JDBC to the appropriate Java environment on your machine:

   ```
   CLASSPATH=$CLASSPATH:/openlink/openlink/jdk1.1.x/megathin.jar
   ```

   Note: This step is only required because the Linux and UNIX installer archives automatically install all the OpenLink Driver types for JDBC, and also perform the default CLASSPATH entry configuration.

5. Run the script "openlink.sh" (you may also want to add a reference to this in your ".profile" file) by executing the following command from your Linux or UNIX command line prompt:

   ```
   . openlink.sh
   ```

6. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a Linux or UNIX command prompt:

OpenLink Driver for JDBC Type	**Verification Command**
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

7. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000 or http://<hostname of current machine>:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

   ```
   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agen
   ```

8. An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run the following commands:

   ```
   ora8_sv -?  :this will verify the Oracle Database Agent
   pro83a_sv  -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent
   ```

   See the detailed section about OpenLink Database Agents for additional information.

Java Based Client-Server (2-Tier) Installation

In this scenario the Java Virtual Machine and OpenLink Drivers for JDBC reside on separate to OpenLink Server server components for JDBC and Database Connectivity. Thus, you are going to install your OpenLink Client and Server components for JDBC on separate machines, one acting as the Client and the other the Server. The Server also hosts the actual database engine that you will be connecting to via JDBC.

Windows 95/98/NT/2000 Client Components Installation Process

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard . You would have selected "Java Virtual Machine" as you client operating system when interacting with the OpenLink download Wizard and then have the files "megathin.jar" or "megathin2.jar" presented in the download results page depending on the version of the Java Virtual Machine selected

2. Place the "megathin.jar" or "megathin2.jar" file into directory of your choice then add the directory and reference to the JAR file to your CLASSPATH environment variable. See example below:

   Windows 95/98/NT/2000

   Presuming you place the "megathin.jar" file in the "\program files\openlink\jdk11" on your Windows machine, you would add the following line to your "autoexec.bat" if you are running Windows 95/98:

   set CLASSPATH=%CLASSPATH%;"c:\program files\openlink\jdk11\megathin.jar":.

   If you are using NT or Windows 2000 " then you need to open the "System Environment" properties of the "System" Control Panel applet and then add the same entry to the "System Variables" section if you want the driver to be accessible to all users, if not place the entry in the "User Variables" section.

3. Reboot your machine

4. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

Linux or UNIX Client Components Installation

1. Presuming you place the "megathin.jar" file in the "/opt/openlink/jdk11" on your Linux or UNIX machine, you would need to modify the following line in the file "openlink.sh" so that they match what is listed below:

   CLASSPATH=$CLASSPATH:/opt/openlink/jdk1.1.x/megathin.jar

2. Run the script "openlinks.sh"

3. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a Linux or UNIX command prompt:

OpenLink Driver for JDBC Type	Verification Command
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

Server Components Installation

Only perform these steps if you are connecting to database engines hosted on your dedicated Database Server using OpenLink's Database Independent Networking:

Windows 95/98/NT/200

1. As a separate Windows 95/98/NT/200 is playing the role of Server machine, you would have downloaded a ZIP archive that contains both the OpenLink Client & Server components for this platform. Extract the contents of this ZIP archive to a temporary installation folder on the Windows Server machine and then run the "Setup.exe" program

2. The archive you have downloaded will contain the entire suite of Data Access Drivers for this platform. If you do not require the OpenLink ODBC or OLE-DB Drivers simply uncheck these components using the installers component list dialog when presented during the install process.

3. Reboot your system

4. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000/

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

5. Verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation directory. Then run the following commands:

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Server Components Installation

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Move the Request Broker and Database Agent archives into a temporary installation folder on your Linux or UNIX machine then run the following commands from the command line prompt:

   Linux:
   rpm -ivh openlink-3.2-2.rpm
   rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
   or
   rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   sh install.sh

3. Follow the instructions presented by the installer for configuring your OpenLink Database Agents, if you installed via a Linux RPM archive, post RPM installation you will need to run the "oplcfg" located in the "openlink/bin" sub-directory of your OpenLink base installation directory

4. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation's base installation directory. This files contains the following entries which you can modify so as to match the OpenLink Drivers for JDBC to the

appropriate Java environment on your machine:

```
CLASSPATH=$CLASSPATH:/openlink/openlink/jdk1.1.x/megathin.jar
```

Note: This step is only required because the Linux and UNIX installer archives automatically install all the OpenLink Driver types for JDBC, and also perform the default CLASSPATH entry configuration.

5. Run the script "openlink.sh" (you may also want to add a reference to this in your ".profile" file) by executing the following command from your Linux or UNIX command line prompt:

   ```
   . openlink.sh
   ```

6. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000 or http://<hostname of current machine>:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

7. Verify your OpenLink Database server components, you do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Java Based Application-Server (3-Tier) Installation

In this scenario the Java Virtual Machine and OpenLink Drivers for JDBC and the OpenLink Server server components for JDBC reside on the same machine which is known as the Application Server. The OpenLink Database Server components reside on a separate Database Server machine (if required) which hosts the database that you will be connecting to via JDBC.
Windows 95/98/NT/2000 Client Components Installation Process.

Windows 95/98/NT/2000 Application Server Components Installation

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard . You would have selected "Java Virtual Machine" as you client operating system when interacting with the OpenLink download Wizard and then have the files "megathin.jar" or "megathin2.jar" presented in the download results page depending on the version of the Java Virtual Machine selected

2. Place the "megathin.jar" or "megathin2.jar" file into directory of your choice then add the directory and reference to the JAR file to your CLASSPATH environment variable. See example below:

   Windows 95/98/NT/2000

   Presuming you place the "megathin.jar" file in the "\program files\openlink\jdk11" on your Windows machine, you would add the following line to your "autoexec.bat" if you are running Windows 95/98:

   ```
   set CLASSPATH=%CLASSPATH%;"c:\program files\openlink\jdk11\megathin.jar":.
   ```

   If you are using NT or Windows 2000 " then you need to open the "System Environment" properties of the "System" Control Panel applet and then add the same entry to the "System Variables" section if you want the driver to be accessible to all users, if not place the entry in the "User Variables" section.

3. Reboot your machine

4. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

5. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000 or http://<hostname of current machine>:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

6. Verify your OpenLink Database server components, you do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Application Server Components Installation

1. Presuming you place the "megathin.jar" file in the "/opt/openlink/jdk11" on your Linux or UNIX machine, you would need to modify the following line in the file "openlink.sh" so that they match what is listed below:

   CLASSPATH=$CLASSPATH:/opt/openlink/jdk1.1.x/megathin.jar

2. Run the script "openlinks.sh"

3. Verify your OpenLink Driver for JDBC client components installation by running one of the following commands (depending on your choice of driver for JDBC) from a DOS Window's command prompt:

OpenLink Driver for JDBC Type	Verification Command
Megathin Driver for JDBC 1.1.x	java openlink.megathin.Driver
Megathin Driver for JDBC 1.2.x or 2.x	java openlink.megathin2.Driver

If you receive output indicating the relevant OpenLink component branding then this indicates that the drivers have been installed correctly and are ready for use with you Java environment, anything else indicates something is wrong. Typically this would be a mismatch between the Java Virtual machine (your default Java environment) and the OpenLink Driver for JDBC classes. Correcting your PATH or CLASSPATH environment variable entries will typically resolve these problems.

Database Server Components Installation

Only perform these steps if you are connecting to database engines hosted on your dedicated Database Server using OpenLink's Database Independent Networking:

Windows 95/98/NT/200

1. As a separate Windows 95/98/NT/200 is playing the role of Server machine, you would have downloaded a ZIP archive that contains both the OpenLink Client & Server components for this platform. Extract the contents of this ZIP archive to a temporary installation folder on the Windows Server machine and then run the "Setup.exe" program

2. The archive you have downloaded will contain the entire suite of Data Access Drivers for this platform. If you do not require the OpenLink ODBC or OLE-DB Drivers simply uncheck these components using the installers component list dialog when presented during the install process.

3. Reboot your system

4. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000/

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is "c:\program files\openLink"). Then run the following commands:

   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent

5. Verify the existence and state of the OpenLink Database server components called the OpenLink Database agents. You do this by moving into the "bin" sub-directory of your OpenLink installation directory. Then run the following commands:

   ora8_sv -? :this will verify the Oracle Database Agent
   pro83a_sv -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent

   See the detailed section about OpenLink Database Agents for additional information.

Linux or UNIX Server

1. Download appropriate driver software installation archive using the instructions provided in the section that covers interaction with the OpenLink Software Download Wizard. Ensure that you hatched a checkbox for each Database Engine type that you will be connecting to via JDBC.

2. Move the Request Broker and Database Agent archives into a temporary installation folder on your Linux or UNIX machine then run the following commands from the command line prompt:

   Linux:
   rpm -ivh openlink-3.2-2.rpm
   rpm -ivh openlink-agents-3.2-2.i386-glibc2.rpm (for glibc2 based Linux Environments)
   or
   rpm -ivh openlink-agents-3.2-2.i386-libc5.rpm (for libc5 based Linux Environments)

   Linux (if your Linux system does not have the RPM facility) and UNIX:

   sh install.sh

3. Follow the instructions presented by the installer for configuring your OpenLink Database Agents, if you installed via a Linux RPM archive, post RPM installation you will need to run the "oplcfg" located in the "openlink/bin" sub-directory of your OpenLink base installation directory

4. The installer creates an OpenLink environment setup script named "openlink.sh" in the openlink installation's base installation

directory. This files contains the following entries which you can modify so as to match the OpenLink Drivers for JDBC to the appropriate Java environment on your machine:

```
CLASSPATH=$CLASSPATH:/openlink/openlink/jdk1.1.x/megathin.jar
```

Note: This step is only required because the Linux and UNIX installer archives automatically install all the OpenLink Driver types for JDBC, and also perform the default CLASSPATH entry configuration.

5. Run the script "openlink.sh" (you may also want to add a reference to this in your ".profile" file) by executing the following command from your Linux or UNIX command line prompt:

   ```
   . openlink.sh
   ```

6. Verify your OpenLink Driver for JDBC server components installation by starting a Web Browser session and then entering the following URL:

   http://localhost:8000 or http://<hostname of current machine>:8000

   If you are presented with the Home Page of the OpenLink Admin Assistant then this confirms that your OpenLink Server environment is also correctly setup.

   An additional but non compulsory check that you may perform is to actually verify the existence and state of the OpenLink JDBC server components called the OpenLink JDBC agents. You do this by starting a DOS command Window and then move into the "bin" sub-directory of your OpenLink installation directory (default is /opt/openlink for Linux and /usr/openlink for UNIX). Then run the following commands:

   ```
   jdbc_sv -? - this will verify the default JDBC Agent
   jodbc_sv -? - This will verify the JDBC-ODBC Agent
   judbc_sv -? - This will verify the JDBC-UDBC Agent
   ```

7. Verify your OpenLink Database server components, you do this by moving into the "bin" sub-directory of your OpenLink installation's base installation directory. Then run one of the following commands (depending on what database(s) you will be connecting to via JDBC):

   ```
   ora8_sv -?  :this will verify the Oracle Database Agent
   pro83a_sv  -? :this will verify the Progress Database Agent
   syb10_sv -? : this will verify the Sybase Database Agent
   inf7_sv -? : this will verify the Informix Database Agent
   ing7_sv -? : this will verify the Ingres II Database Agent
   db2_sv -? : this will verify the IBM DB2 Database Agent
   ```

   See the detailed section about OpenLink Database Agents for additional information.

Mixed Environment Installations

It is important to note that the client and server operating systems hosting your OpenLink Software do not have to be the same. The Installation instructions have only taken this approach in order to simplify understanding of the installation process. A Linux or UNIX machine can act as a client to a Windows machine and vice versa, all you have to do is follow the steps for installing either the client or server components on the appropriate platform.

OpenLink Server Components Configuration

The OpenLink JDBC agent and OpenLink Database agents form the OpenLink server components, In the prior section you would have installed these components on the appropriate server machine.

Following installation you have to configure these server components in order to enable connectivity between your backend database(s) and your OpenLink Driver for JDBC. Both of these components are exposed to your OpenLink Driver for JDBC via the OpenLink Request Broker.

JDBC Agents

These server components are available in two formats, one based on the OpenLink Generic ODBC technology and the Other based on OpenLink Generic UDBC technology.

The ODBC based agents are identified as "JODBC" agents to the OpenLink Driver for JDBC and packaged as "jodbc_sv.exe" and "jodbc_sv" on Windows and Linux/UNIX platforms respectively. The JODBC agent provides you with the flexibility of either connecting to backend databases via existing ODBC DSNs or directly (DSN-Less connections), this facility also has the added benefit of not restricting the use of your Type 3 OpenLink Driver for JDBC to OpenLink ODBC DSNs. This is how OpenLink provides remote JDBC-ODBC Bridging with out compromising the platform independence or Java Purity required of a Type 3 JDBC Driver.

The UDBC based agents are identified as "JUDBC" agents to the OpenLink Driver for JDBC and packaged as "judbc_sv.exe" and "judbc_sv" on Windows and Linux/UNIX platforms respectively. The JUDBC agent provides you with the flexibility of either connecting to backend databases via existing UDBC DSNs or directly (DSN-Less connections). This is how OpenLink provides remote JDBC-Native Interface Bridging with out compromising the platform independence or Java Purity required of a Type 3 JDBC Driver, the OpenLink UDBC layer acts as a Generic Native Database Interface, it does not have any interaction with an ODBC Driver Manager or ODBC DSNs.

Database Agents

OpenLink Database Agents are the OpenLink data access server components that actually provide database connectivity services to your OpenLink Driver for JDBC. An Database Agent exists for each database engine supported by OpenLink, the supported database list currently includes: Oracle, DB2, Informix, Sybase, Ingres, Progress, Microsoft SQL Server, OpenLink Virtuoso, Solid, PostgresSQL, and other ODBC based databases. Please follow the instructions provided in the OpenLink Database Agents configuration guide prior to attempting to use your OpenLink Drivers for JDBC (if you haven't already done so as part of the installation process).

OpenLink Drivers for JDBC™ Utilization

OpenLink Drivers for JDBC are available in three different JDBC Driver formats.

- JDBC Type 1 - Driver for JDBC is implemented as a bridge to ODBC Drivers, thereby implementing the JDBC Driver classes through native methods, this is due to the fact that ODBC is a 'C' language based data access application programming interface. Thus, this driver format is inherently part Java and part Native, implying that it is inherently platform specific rather than independent.
- JDBC Type 2 - Driver for JDBC is implemented as a bridge to Native Database Call Level Interfaces, thereby implementing the JDBC Driver classes through native methods, this is due to the fact that Native Database Call Interfaces are either C/C++ language based data access application programming interfaces. Thus, this driver format is inherently part Java and part Native, implying that it is inherently platform specific rather than independent
- JDBC Type 3 - Driver for JDBC is implemented in Java sitting atop a database independent networking layer bridge also implemented in Java. Thus, the entire driver is Pure Java and thereby operating system independent.

JDBC Applets, Applications, Bean Components, and Servlets communicate with JDBC drivers through JDBC Uniform Resource Locators (URLs). Theses URLs are service request and binding formats implemented slightly differently for each OpenLink Driver for JDBC format. The general JDBC URL format is:

jdbc:<jdbc-subprotocol>:[jdbc implementation specific URL attributes]

The "sub-protocol" component of the URL above identifies each JDBC implementation and typically identifies the JDBC driver vendor, the actual URL attributes are vendor specific. Each OpenLink Driver for JDBC type has a different JDBC URL format, the sections that follow depict and provides examples of these formats.

OpenLink Driver for JDBC Type 1

URL Format

This driver format connects you to ODBC Data Source Names (DSNs) via JDBC. The URL format is as follows:

jdbc:openlink://ODBC[/DSN][/UID][/PWD][/READONLY]

URL Attributes

/DSN - ODBC Data Source Name
/UID - Username
/PWD - Password
/READONLY - Determines session mode, read-write or read-only.

Example:

If you were attempting to connect to an ODBC DSN on your machine named "Customers Database" in read-only mode then you would enter the following JDBC URL:

jdbc:openlink://ODBC/DSN=Customer Database/UID=test/PWD=test/READONLY=Y

Note: In the case of OpenLink ODBC DSNs you do not have to provide values for the /UID and /PWD attributes since these can be controlled and configured on the database or application server using the OpenLink Session Rules Book.

OpenLink Driver for JDBC Type 2

URL Format

This driver format connects you to UDBC Data Source Names (DSNs) via JDBC. The URL format is as follows:

jdbc:openlink://UDBC[/DSN][/UID][/PWD][/READONLY]

URL Attributes

/DSN - ODBC Data Source Name
/UID - Username
/PWD - Password
/READONLY - Determines session mode, read-write or read-only.

Example:

If you were attempting to connect to an UDBC DSN on your machine named "Customers Database" in read-only mode then you would enter the following JDBC URL:

jdbc:openlink://UDBC/DSN=Customer Database/UID=test/PWD=test/READONLY=Y

Note: In the case of OpenLink ODBC DSNs you do not have to provide values for the /UID and /PWD attributes since these can be controlled and configured on the database or application server using the OpenLink Session Rules Book.

OpenLink Driver for JDBC Type 3

URL Format

This driver format connects you to remote database using remote ODBC or UDBC DSNs. It also supports direct DSN-Less connections to remote databases. The URL format is as follows:

jdbc:openlink://<Hostname>:[portnumber] [/DSN] [/UID] [/PWD] [/READONLY] [/FBS]
[/JDBCAGENT] [/SVT] [/DATABASE] [/OPTIONS] [/DRIVER]

URL Attributes

Hostname - Network Alias or IP address of server machine running an OpenLink Request Broker instance

Port Number - Port number that identifies location of OpenLink JDBC Agent Service, the default value is 5000

/DSN - ODBC Data Source Name

/UID - Username

/PWD - Password

/READONLY - Determines session mode, read-write or read-only

/FBS - Sets number of JDBC resultset rows that get packed into a single network packet

/JDBCAGENT - Determines JDBC Agent type used rather than default (JDBC Agents exist for ODBC and UDBC)

/SVT - Determines OpenLink Database Agent type (Oracle, Informix, Sybase, Progress, Ingres, SQL Sever, Sybase etc.)

/DATABASE - Actual database name within a particular database environment

/OPTIONS - Values used to connect to OpenLink Database Agents to remote database servers using database vendors networking

/DRIVER - Used when making a DSN-Less connection to remote ODBC Driver

Examples:

Connecting To Remote ODBC DSN

If you were attempting to connect to a remote ODBC DSN named "Customers Database", hosted on a database server machine with the network alias "pluto", with an OpenLink JDBC server listening at port 5001 (rather than default of 5000), and you wanted this session to be in read-only mode then you would enter the following JDBC URL:

jdbc:openlink://pluto:5001/DSN=Customer Database/UID=test/PWD=test/READONLY=YES/JDBCAGENT=jodbc/FBS=55

Note:

1. In the case of OpenLink ODBC DSNs you do not have to provide values for the /UID and /PWD attributes since these can be controlled and configured on the database or application server using the OpenLink Session Rules Book
2. If "pluto" is a Windows 95/98/NT/2000 machine then the "/JDBCAGENT" attribute defaults to "jodbc" when left out of the JDBC URL. Likewise if "pluto" is a Linux or UNIX machine the "/JDBCAGENT" attribute defaults to "judbc"
3. "/FBS" ensures that each iteration of a JDBC Resultset fetch loop returns 55 records or less until all records have been retrieved from a remote database server

Connecting To Remote UDBC DSN

If you were attempting to connect to a remote UDBC DSN named "Customers Database", hosted on a database server machine with the network alias "pluto", with an OpenLink JDBC server listening at port 5001 (rather than default of 5000), and you wanted this session to be in read-only mode then you would enter the following JDBC URL:

jdbc:openlink://pluto:5001/DSN=Customer Database/UID=test/PWD=test/READONLY=Y/JDBCAGENT=judbc/FBS=55

Note:

1. In the case of OpenLink ODBC DSNs you do not have to provide values for the /UID and /PWD attributes since these can be controlled and configured on the database or application server using the OpenLink Session Rules Book
2. If "pluto" is a Linux or UNIX machine then the "/JDBCAGENT" attribute defaults to "judbc" when left out of the JDBC URL. Likewise if "pluto" is a Windows 95/98/NT/2000 machine the "/JDBCAGENT" attribute defaults to "jodbc"
3. "/FBS" ensures that each iteration of a JDBC Resultset fetch loop returns 55 records or less until all records have been retrieved from a remote database server

Connecting To Databases Using DSN-Less Connections

You do not have to create ODBC or UDBC DSNs in order to use your OpenLink Drivers for JDBC when using the type 3 format. Instead you can specify the OpenLink Database Type and Database Name attributes as part of your JDBC URL.

To connect to a remote Microsoft SQL Server database without going via an ODBC DSN you would construct the following URL:

jdbc:openlink://saturn:5001/SVT=SQLServer 6/DATABASE=pubs/UID=sa/PWD=/FBS=55/READONLY=Y

Notes:

1. In the case of OpenLink ODBC DSNs you do not have to provide values for the /UID and /PWD attributes since these can be controlled and configured on the database or application server using the OpenLink Session Rules Book
2. As "saturn" is a Windows 95/98/NT/2000 machine the "/JDBCAGENT" attribute defaults to "jodbc"
3. This feature applies to both OpenLink JDBC Agent types: JODBC Agent and JUDBC Agent

Connecting To Database via ODBC Driver Without A DSN (DSN-Less Connection)

This JDBC URL format is currently only supported by the OpenLink JDBC Agent for ODBC DSNs (JODBC Agent). DSN-Less connections require you to determine the ODBC connect string attributes for the ODBC Driver that you are using. For OpenLink ODBC Drivers these values are:

ServerType - Database Agent Type

Host - Machine hosting the Database Agent serving an OpenLink ODBC Driver

Username - Valid Database Username

Password - Valid Password for Username

FetchBufferSize - Number of resulset records fetched during each ODBC fetch loop

Database - Actual database name within database server environment

NoLoginBox - Disables OpenLink ODBC Drivers attempt to present dialog when ODBC Driver determines an incomplete ODBC connect string (collection of ODBC attributes passed at connect time) good examples being missing or blank "Username" and "Password" attributes.

If you were connecting a remote Oracle database on a machine called "pluto" and you wanted this session to be read-only, your URL formal would be as follows:

jdbc:openlink://pluto/DRIVER={OpenLink Generic 32 Bit Driver}/Database=ORCL/Username=test/PWD=test/
ReadOnly=Yes/FBS=55/ServerType=Oracle 8/Host=pluto

Connecting To Remote Databases On Separate Server Machine (OpenLink 3-Tier Architecture)

You may choose to install your OpenLink Drivers for JDBC on an Application Server and then install your OpenLink Data Access Server components (Request Broker and Database Agents) on your dedicated database server machine. In such a scenario you will be connecting to your remote database engine using OpenLink Database Independent as opposed to your Database vendor's database specific networking middleware.

If you were connecting from your Application Server called "pluto" to a dedicated Database Server machine named "ora_server", hosting an Oracle database identified as "ORCL" you would construct the following JDBC URL:

jdbc:openlink://pluto/SVT=Oracle 6/
UID=test/PWD=test/HOST=ora_server

Notes:

1. This feature applies to both OpenLink JDBC Agent types: JODBC Agent and JUDBC Agent

2. You could also have used the "/DSN"   attribute to point to an ODBC or UDBC DSN which has been configured to connect to the Database Server machine, this simply reduces the size of your JDBC URL, but imposes the use of DSNs upon you.

Connecting To Remote Databases On Separate Server Machine Using Database Vendors Networking (Mixed 3-Tier Architecture)

Organizational standards or individual preference may present you with a scenario in which you have two server machines in use, one acting as an Application Server hosting your OpenLink Drivers for JDBC and OpenLink Data Access Server components (Request Broker & Database Agents), and the other acting as a dedicated Database Server. You may not have the necessary authority to install the OpenLink Data Access Server components on the Database Server, or you simply prefer to use your database vendors networking software which is already configured on your Application Server. This scenario can be described as a "Mixed 3-Tier" architecture, this is because you are going to use your OpenLink Database Agents atop database vendor provided networking rather than connecting to an OpenLink Database Agent using OpenLink Database independent networking.

If you were connecting to a remote Oracle database somewhere on your network from our application server called "pluto" using an ODBC DSN called "Customers" you would construct the following JDBC URL assuming a Net8 or SQL*Net "tnsname" or server alias called "ora_pluto":

jdbc:openlink://pluto/SVT=Oracle 8/
UID=test/PWD=test/OPTIONS=ora_pluto

Notes:

1. The "/OPTIONS" JDBC URL attribute provides the entry or bind point for connecting OpenLink Database agents to Database vendors networking products. This applies to all supported OpenLink databases, see the OpenLink Database Agents configuration guide for additional information relating to the database specific formats of values passed to the "/OPTIONS" JDBC URL attribute

2. This feature applies to both OpenLink JDBC Agent types: JODBC Agent and JUDBC Agent

You could also have used the "/DSN"   attribute to point to an ODBC or UDBC DSN which has been configured to connect to the Database Server machine, this simply reduces the size of your JDBC URL, but imposes the use of DSNs upon you.

OpenLink Demonstration Programs

To assist you further during your utilization or evaluation of OpenLink's Drivers for JDBC a number of demonstration JDBC compliant Applets and Applications are bundled with your OpenLink Driver for JDBC installation, these programs are provided in both binary and source code format for your free use. The sections that follow guide you through the process of using these programs

JDBC Compliant Applet Demos

Three JDBC applet samples are bundled with your OpenLink Driver for JDBC installation, each one of these demonstrating practical use of JDBC applets and highlighting OpenLink specific functionality. Each of these demos reside in the "samples\jdbc\jdk[10 or 11 or 12]" sub-directory below the directory into which you installed your OpenLink software. Each applet is accessible from the OpenLink Admin Assistant (an OpenLink agent that provides HTTP services like any Web Server does). The programs are:

1. JDBCDemo - demonstrates basic JDBC functionality via an Applet

2. ScrollDemo - demonstrates JDBC functionality via an Applet. It also demonstrates the additional Resultset navigation functionality provided by OpenLink's Scrollable Resultset & RowSet Extensions for JDBC on a Record by Record Basis.

3. ScrollDemo2 - demonstrates JDBC functionality via an Applet. It also demonstrates the additional Resultset navigation functionality provided by JDBC 2.0

4. RowSetDemo - demonstrates JDBC functionality via an Applet. It also demonstrates the additional Resultset navigation functionality provided by OpenLink's Scrollable Resultset & RowSet Extensions for JDBC on a RowSet by RowSet Basis.

JDBCDemo

This applet demonstrates basic JDBC functionality via an Applet.

Utilization Steps:

1. Start the OpenLink Request Broker (click here for instructions on how to do this under Windows and here on how to do this on Linux or UNIX machines)

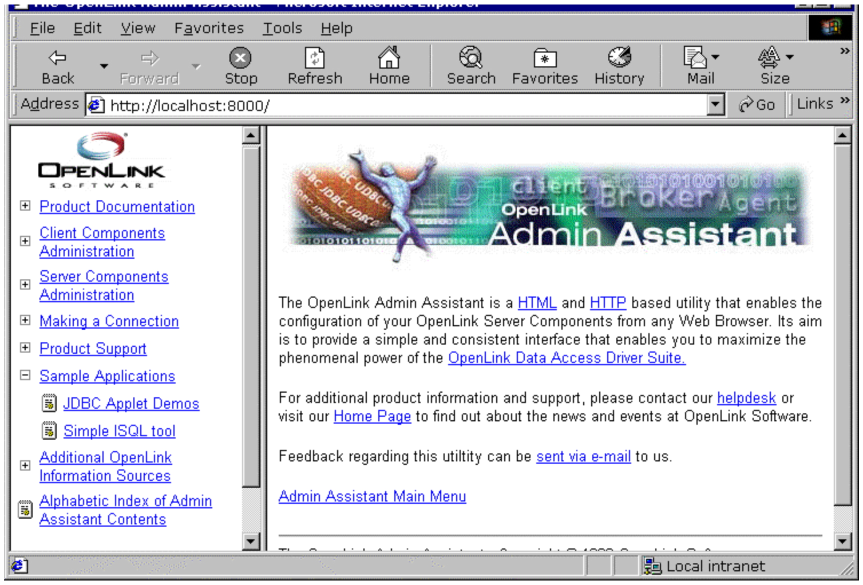
2. Start up your Web Browser

3. Enter one of the following URLs into your browser depending on the location of your OpenLink Request Broker:
   Local To you:
   http://localhost:8000

   Remote Server:
   http://<hostname or IP address of remote server>:8000

   Note: Port "8000" presumes that you provide this value when prompted during your OpenLink Sever components installation.

4. Follow the Admin Assistant's Menu tree to the location of the "Sample Applications->JDBC Applet Demos" menu item. The graphic below depicts this process.


File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://localhost:8000/
Go Links
Product Documentation
Client Components Administration
Server Components Administration
Making a Connection
Product Support
Sample Applications
JDBC Applet Demos
Simple ISQL tool
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
Client Broker Agent
OpenLink Admin Assistant
Local intranet

The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

5. Click on the hyperlink that reads "Applet demonstration with OpenLink Software JDBC Driver"

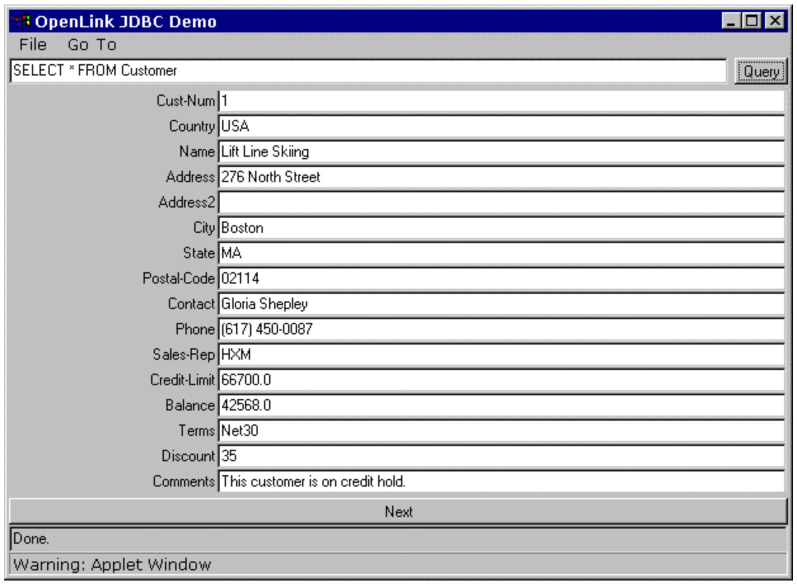
6. Use the Applet's File->Set Connection URL menu item set a URL pointing to an ODBC or UDBC DSN. If uncertain follows the

instructions laid out in the section covering OpenLink JDBC URL formats which shows you how to construct Type 1, 2, and 3 URL formats for your OpenLink Drivers for JDBC. This applet will run with non OpenLink Drivers for JDBC but you will need to obtain URL construction information from the relevant driver vendor.

The line below depicts the URL construction dialog presented:

jdbc:openlink://localhost/DSN=WebJDBCDemo

7. Enter a valid SQL statement for the backend database that you are connecting to via JDBC and then click on the "Query" button. The screen shot below depicts this process:



SELECT * FROM Customer

Cust-Num	1
Country	USA
Name	Lift Line Skiing
Address	276 North Street
Address2	
City	Boston
State	MA
Postal-Code	02114
Contact	Gloria Shepley
Phone	(617) 450-0087
Sales-Rep	HXM
Credit-Limit	66700.0
Balance	42568.0
Terms	Net30
Discount	35
Comments	This customer is on credit hold.

Done.

Warning: Applet Window

8. Basic JDBC 1.1 functionality provides Forward-Only as opposed to Bi-Directional record Scrolling, this is why the basic JDBC applet on has a "Next" button. When you click on the "Next" button you are moved to the next record in your JDBC resultset, unfortunately you have to hit the "Query" button again and re-start the Forward-Only resultset navigation if you need to see the First or Prior resultset records from your current position. The examples that follow show how OpenLink and the new release of JDBC (version 2.0) address the Bi-Directional Scrolling Limitation demonstrated by this Applet.

ScrollDemo

This program demonstrates JDBC functionality via an Applet. It also demonstrates the additional Resultset navigation functionality provided by OpenLink's Scrollable Resultset & RowSet Extensions for JDBC on a Record by Record Basis.

Utilization Steps:

1. Start the OpenLink Request Broker (click here for instructions on how to do this under Windows and here on how to do this on Linux or UNIX machines)

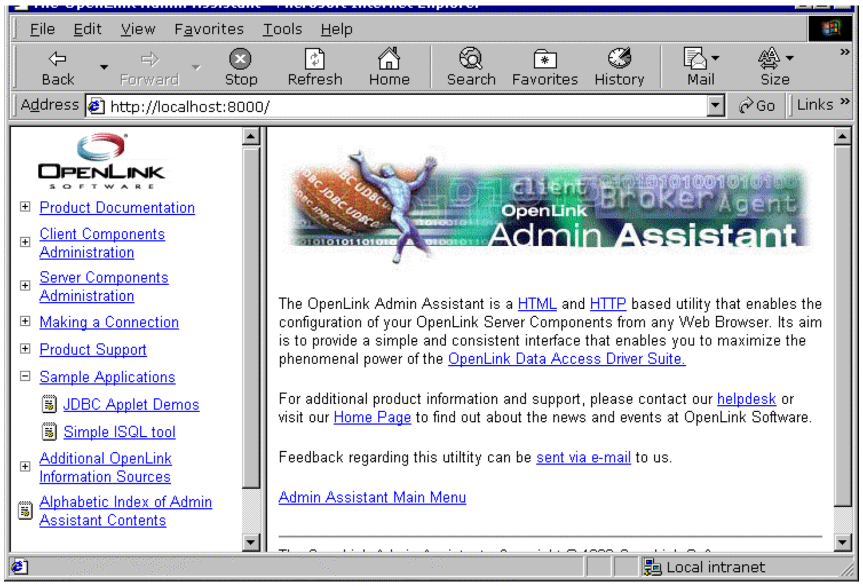
2. Start up your Web Browser

3. Enter one of the following URLs into your browser depending on the location of your OpenLink Request Broker:
   Local To you:
   http://localhost:8000

   Remote Server:
   http://<hostname or IP address of remote server>:8000

   Note: Port "8000" presumes that you provide this value when prompted during your OpenLink Sever components installation.

4. Follow the Admin Assistant's Menu tree to the location of the "JDBC Applet Demos" menu item. The graphic below depicts this process.


File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://localhost:8000/
Go Links
Product Documentation
Client Components Administration
Server Components Administration
Making a Connection
Product Support
Sample Applications
JDBC Applet Demos
Simple ISQL tool
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
OpenLink Admin Assistant
Local intranet

The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

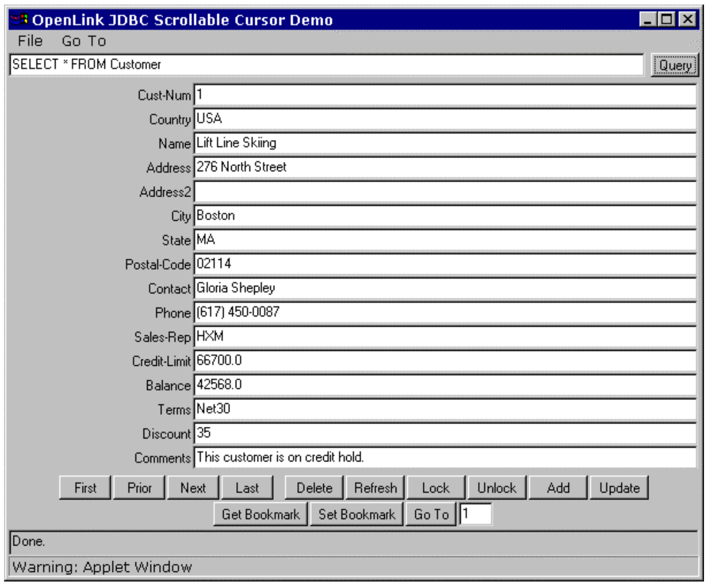
5. Click on the hyperlink that reads "Applet demonstration with OpenLink Software JDBC Scrollable Cursor extensions"

6. Use the Applet's File->Set Connection URL menu item set a URL pointing to an ODBC or UDBC DSN. If uncertain follows the instructions laid out in the section covering OpenLink JDBC URL formats which shows you how to construct Type 1, 2, and 3 URL formats for your OpenLink Drivers for JDBC.

   The line below depicts the URL construction dialog presented:

   jdbc:openlink://localhost/DSN=WebScrollDemo

7. Enter a valid SQL statement for the backend database that you are connecting to via JDBC and then click on the "Query" button. The screen shot below depicts this process:


OpenLink JDBC Scrollable Cursor Demo
File Go To
Query
First Prior Next Last Delete Refresh Lock Unlock Add Update
Get Bookmark Set Bookmark Go To

SELECT * FROM Customer

Cust-Num	1
Country	USA
Name	Lift Line Skiing
Address	276 North Street
Address2	
City	Boston
State	MA
Postal-Code	02114
Contact	Gloria Shepley
Phone	(617) 450-0087
Sales-Rep	HXM
Credit-Limit	66700.0
Balance	42568.0
Terms	Net30
Discount	35
Comments	This customer is on credit hold.

Go To: 1

Done.

Warning: Applet Window

8. JDBC 1.1 functionality provides Forward-Only as opposed to Bi-Directional Resultset Scrolling, OpenLink's Scrollable Resultset Extensions for JDBC enable Bi-Directional Resultset Scrolling. This is why this applet has an additional set of Resultset Navigation buttons: "First","Next", "Prior","Last", "Lock", "Unlock", "Add", "Update", "Get Bookmark", "Set Bookmark", "and Go To" . The existence of Bi-directional Scrollable Resultsets (or Cursors) is often presumed by end-users and developers

alike, its importance rarely understood prior to embarking upon JDBC application development or product selection, the unfortunate consequence being complex application re-writes or implementation of sub par JDBC solutions. Each of the button in the applet demo is explained below so as to understand the magnitude of this issue:

First - takes you to first record in the Resultset

Next - takes you to the next record in the Resultset from your current position

Prior - takes you to the previous record in the Resultset from your current position

Last - takes you to the last record in the Resultset

Lock - locks the current record

Unlock - unlocks the current record

Add - add a new record to database

Update - change current record

Delete - remove current record from database

Get Bookmark - mark current record position for future revisit

Set Bookmark - revisit previous marked position in current ResultSet

Go To - go directly to a specific record number within the current ResultSet

Refresh - Reopen current resultset

ScrollDemo2

This applet demonstrates JDBC functionality via an Applet. It also demonstrates the additional Resultset navigation functionality provided by JDBC 2.0

This Applet require a browser that is Java Virtual Machine version 1.2.x or 2.x compliant. If you do not have such a Browser, you can simply run the JDBC Application version of this program.

Utilization Steps:

1. Start the OpenLink Request Broker (click here for instructions on how to do this under Windows and here on how to do this on Linux or UNIX machines)

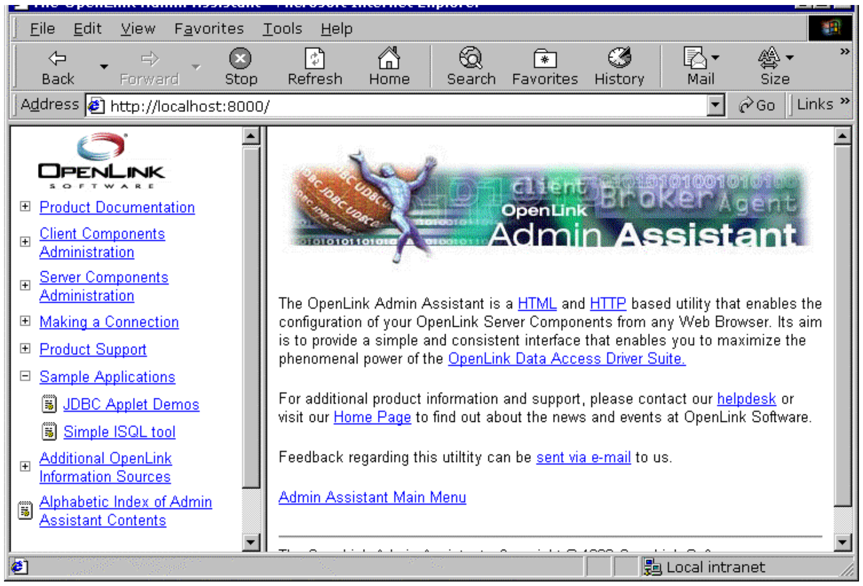
2. Start up your Web Browser

3. Enter one of the following URLs into your browser depending on the location of your OpenLink Request Broker:
   Local To you:
   http://localhost:8000/

   Remote Server:
   http://<hostname or IP address of remote server>:8000

   Note: Port "8000" presumes that you provide this value when prompted during your OpenLink Sever components installation.

4. Follow the Admin Assistant's Menu tree to the location of the "Sample Applications->JDBC Applet Demos" menu item. The graphic below depicts this process.


File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://localhost:8000/
Go Links
OpenLink Software
Product Documentation
Client Components Administration
Server Components Administration
Making a Connection
Product Support
Sample Applications
JDBC Applet Demos
Simple ISQL tool
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
OpenLink Admin Assistant
Local intranet

The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

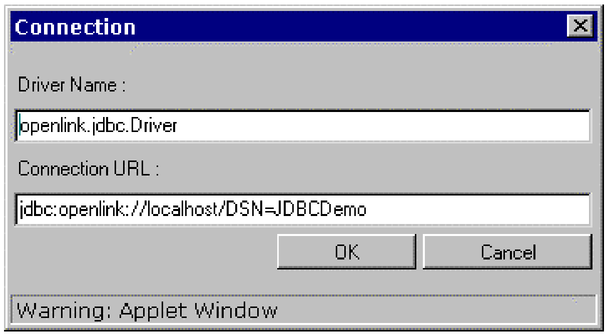
Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

5. Click on the hyperlink that reads "Applet demonstration with OpenLink Software JDBC 2.0 Scrollable Cursors"

6. Use the Applet's File->Set Connection URL menu item set register your Driver for JDBC 2.0 and then enter a URL pointing to an ODBC or UDBC DSN. If uncertain follow the instructions laid out in the section covering OpenLink JDBC URL formats which shows you how to construct Type 1, 2, and 3 URL formats for your OpenLink Drivers for JDBC. This applet will run with non OpenLink Drivers for JDBC but you will need to obtain Driver registration and JDBC URL construction information from the relevant Driver vendor.

   The screen shot below depicts the URL construction dialog presented



Driver Name : openlink.jdbc.Driver

Connection URL : jdbc:openlink://localhost/DSN=JDBCDemo

7. Enter a valid SQL statement for the backend database that you are connecting to via JDBC and then click on the "Query" button.

8. JDBC 1.1 functionality provides Forward-Only as opposed to Bi-Directional Resultset Scrolling, JDBC 2.0 on the other hand supports Bi-Directional Resultset Scrolling. As a result this applet has an additional set of Resultset Navigation buttons: "First","Next", "Previous","Last", "Insert", "Update", "Absolute", "Relative". Unfortunately JDBC 2.0 does not provide Bookmarking or Row Level concurrency control hence the exclusion of the "Lock", "UnLock", "Go To", "Set Bookmark", "Get Bookmark" navigation buttons provided in the "ScrollDemo" applet. To use this functionality in a JDBC 2.0 environment you simply make use of the OpenLink Scrollable ResultSet & RowSet Extensions.

   Each navigation button is described below so as to shed more light on the Scrollable ResultSet functionality provided by JDBC 2.0.

   First - takes you to first record in the Resultset

   Next - takes you to the next record in the Resultset from your current position

   Previous - takes you to the previous record in the Resultset from your current position

   Last - takes you to the last record in the Resultset

Add - add a new record to database

Update - change current record

Delete - remove current record from database

Relative - moves N number of records forward from the current record where N represents a value entered into the field beside the "Relative" button. IF the field contains a negative number then it indicates a backwards move.

Absolute - go directly to record number N within the current ResultSet where N represents a value entered into the field beside the "Relative" button, the actual direction of Resultset navigation depends on the actual location of the record in question

Refresh - Reopen current resultset

RowSetDemo

This applet demonstrates JDBC functionality via an Applet. It also demonstrates the additional Resultset navigation functionality provided by OpenLink's Scrollable Resultset & RowSet Extensions for JDBC on a RowSet by RowSet Basis.

Utilization Steps:

1. Start the OpenLink Request Broker (click here for instructions on how to do this under Windows and here on how to do this on Linux or UNIX machines)

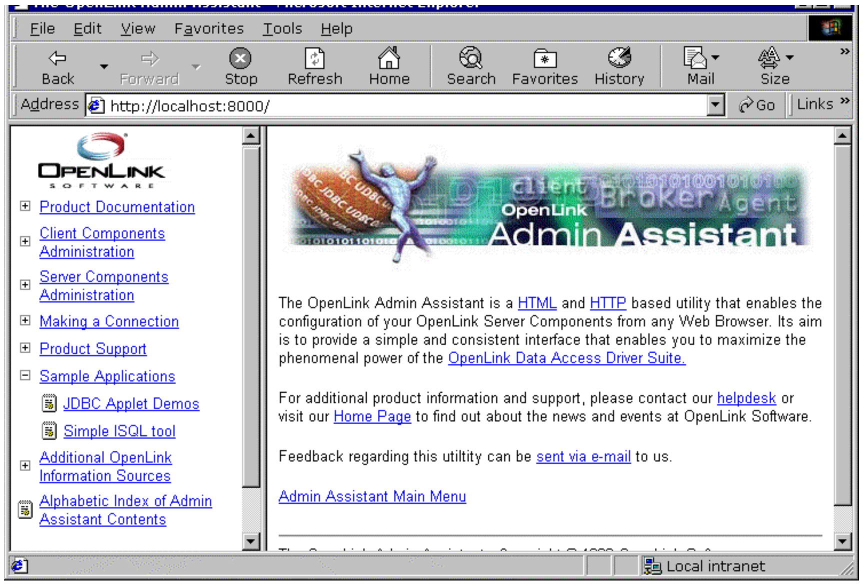
2. Start up your Web Browser

3. Enter one of the following URLs into your browser depending on the location of your OpenLink Request Broker:
   Local To you:
   http://localhost:8000

   Remote Server:
   http://<hostname or IP address of remote server>:8000

   Note: Port "8000" presumes that you provide this value when prompted during your OpenLink Sever components installation.

4. Follow the Admin Assistant's Menu tree to the location of the "Sample Applications->JDBC Applet Demos" menu item. The graphic below depicts this process.



The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

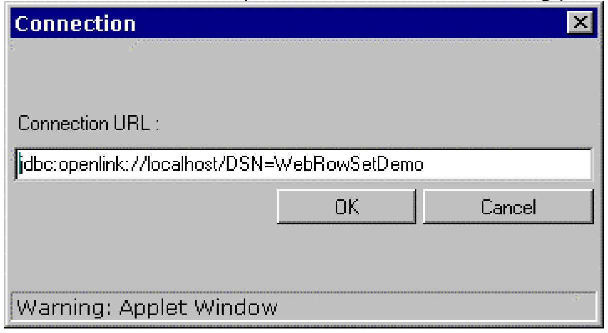
Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

5. Click on the hyperlink that reads "Applet demonstration with OpenLink Software JDBC Scrollable Cursor RowSet Extensions"

6. Use the Applet's File->Set Connection URL menu item set a URL pointing to an ODBC or UDBC DSN. If uncertain follows the instructions laid out in the section covering OpenLink JDBC URL formats which shows you how to construct Type 1, 2, and 3 URL formats for your OpenLink Drivers for JDBC. This applet will run with non OpenLink Drivers for JDBC but you will need to obtain URL construction information from the relevant driver vendor.

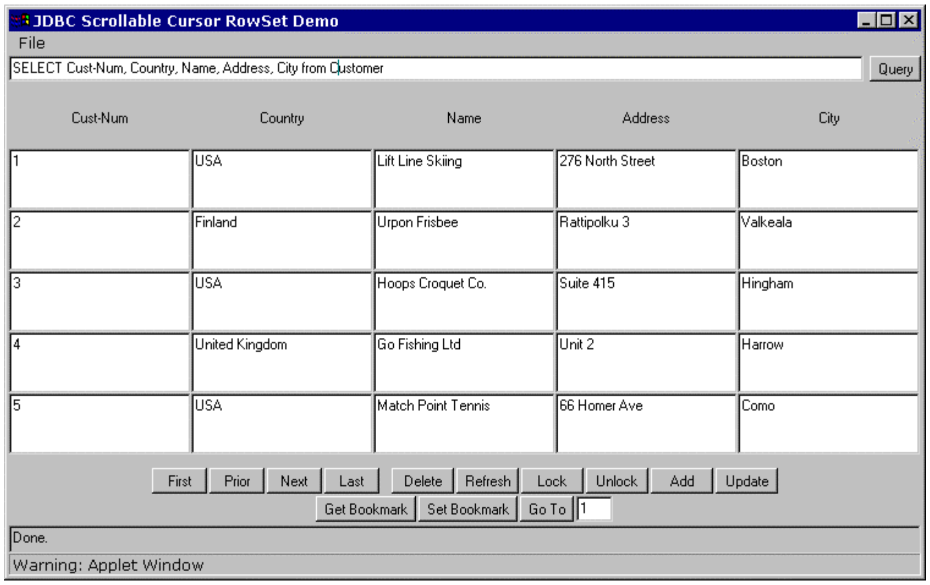
The screen shot below depicts the URL construction dialog presented:



jdbc:openlink://localhost/DSN=WebRowSetDemo

Warning: Applet Window

7. Enter a valid SQL statement for the backend database that you are connecting to via JDBC and then click on the "Query" button. The screen shot below depicts this process:



SELECT Cust-Num, Country, Name, Address, City from Customer

Cust-Num	Country	Name	Address	City
1	USA	Lift Line Skiing	276 North Street	Boston
2	Finland	Urpon Frisbee	Rattipolku 3	Valkeala
3	USA	Hoops Croquet Co.	Suite 415	Hingham
4	United Kingdom	Go Fishing Ltd	Unit 2	Harrow
5	USA	Match Point Tennis	66 Homer Ave	Como

1

Done.

Warning: Applet Window

8. JDBC 1.1 functionality provides Forward-Only as opposed to Bi-Directional Resultset Scrolling, JDBC 2.0 provides Scrollable Resultsets but does not provide Bookmarking or Attached RowSets (transient RowSets located in the same process space as the ResultSet). OpenLink's Scrollable Resultset Extensions for JDBC address these issues irrespective of JDBC version. As a result this applet has an additional set of Resultset Navigation buttons when compared to the basic JDBC 2.0 Applet in the prior section, the buttons are: "First","Next", "Prior","Last", "Lock", "Unlock", "Add", "Update", "Get Bookmark", "Set Bookmark", "and Go To" . The existence of transient RowSets due to Bi-directional Scrollable Resultsets (or Cursors) in JDBC 2.0 is more than likely presumed to exist by end-users and developers alike, it is important that you take note of this before embarking upon JDBC 2.0 application development or product selection. Each of the buttons in the Applet demo is explained below so as to assist in the understanding of these matters:

First - takes you to first RowSet in the Resultset

Next - takes you to the next RowSet in the Resultset from your current position

Prior - takes you to the previous RowSet in the Resultset from your current position

Last - takes you to the last RowSet in the Resultset

Lock - locks the current RowSet

Unlock - unlocks the current RowSet

Add - add a new record to database

Update - change current record with the current RowSet of

Delete - remove current record from within the current RowSet from the database

Get Bookmark - mark current RowSet within Resultset for future revisit

Set Bookmark - revisit previous marked RowSet position in current ResultSet

Go To - go directly to a specific record number within the current ResultSet

Refresh - Reopen current resultset

JDBC compliant Application Demos

All the JDBC compliant Applet demos described in the previous sections have also been implemented as JDBC compliant Applications Demos, you can run these programs in a number of ways depending on operating system hosting your Java Virtual Machine.

Windows 95/98/NT/200

1. Click on your Windows Start Menu Button
2. Select the "OpenLink Data Access Drivers" Start Menu Group
3. Locate the "JDBC Samples" Menu Item
4. Choose from the list of JDBC Applications presented

Linux or UNIX

1. Move into your OpenLink base installation directory
2. The move to the following directory listing output maps out the location of the various JDBC Application demos by Java Virtual Machine version:

```
JDBC/jdk1.0.2/Applications:
JDBCDemo RowSetDemo ScrollDemo

JDBC/jdk1.0.2/Applications/JDBCDemo:
DialogConnection.class JDBCDemo.class
DialogConnection.java JDBCDemo.java

JDBC/jdk1.0.2/Applications/RowSetDemo:
DataTextField.class DialogConnection.java readme.txt
DataTextField.java RowSetDemo.class
DialogConnection.class RowSetDemo.java

JDBC/jdk1.0.2/Applications/ScrollDemo:
DialogConnection.class ScrollDemo.class readme.txt
DialogConnection.java ScrollDemo.java

JDBC/jdk1.1.x/Applications:
JDBCDemo RowSetDemo ScrollDemo

JDBC/jdk1.1.x/Applications/JDBCDemo:
DialogConnection.class JDBCDemo.class
DialogConnection.java JDBCDemo.java

JDBC/jdk1.1.x/Applications/RowSetDemo:
DataTextField.class DialogConnection.java readme.txt
DataTextField.java RowSetDemo.class
DialogConnection.class RowSetDemo.java

JDBC/jdk1.1.x/Applications/ScrollDemo:
DialogConnection.class ScrollDemo.class readme.txt
DialogConnection.java ScrollDemo.java

JDBC/jdk1.2.x/Applications:
JDBCDemo RowSetDemo ScrollDemo ScrollDemo2

JDBC/jdk1.2.x/Applications/JDBCDemo:
DialogConnection.class JDBCDemo.class
DialogConnection.java JDBCDemo.java

JDBC/jdk1.2.x/Applications/RowSetDemo:
DataTextField.class DialogConnection.java readme.txt
DataTextField.java RowSetDemo.class
DialogConnection.class RowSetDemo.java

JDBC/jdk1.2.x/Applications/ScrollDemo:
DialogConnection.class ScrollDemo.class readme.txt
DialogConnection.java ScrollDemo.java

JDBC/jdk1.2.x/Applications/ScrollDemo2:
```

```
DialogConnection.class ScrollDemo2.class
DialogConnection.java ScrollDemo2.java
```

3. Move into the appropriate directory and then execute the following command:

   java <classname>

   where "<classname>" represents the JDBC class file hosting your JDBC application demo. For instance if you wanted to run the "RowSetDemo" JDBC application you would type the following:

   java RowSetDemo

Important Multi-User JDBC Solution Development & Utilization Issues

Sensitivity To Changes In Underlying Database

It is extremely important to application developers and end-users alike to understand the degree to which the Resultsets presented to them by a JDBC solution are actually sensitive to underlying changes in the source database. JDBC 1.1 not only fails to provide you with Bi-directional Resultset Scrolling, it also presents what is basically a snapshot of the data in your database at the time a JDBC query is executed. This has the effect of increasing Multi-User JDBC solution development complexity or limiting the functionality and usability of JDBC by end-users.

Sensitive to changes in underlying database takes many forms, this includes: Static, KeySet, Dynamic, and Mixed modes of sensitivity.

Static - same as basic JDBC, records scrolling occurs over a database snapshot and is insensitive to underlying change by other users

KeySet - JDBC resultset records scroll over a set of record identifiers uniquely identifying records in the underlying database, this type of scrolling is sensitive to changes is those records with identifiers at the time of query execution. This form of scrolling is insensitive to record record additions or deletions.

Dynamic - JDBC resultset records scroll over a set of record identifiers uniquely identifying records in the underlying database, these unique identifiers are recreated before each RowSet traversal (collection of resultset records used as scrolling marker or sliding window or Cursor), rather than once at query execution time. This type of scrolling is sensitive to all changes in the underlying database but may introduce a performance penalties depending on the size of RowSets and available network bandwidth.

Mixed - JDBC resultset records scroll over a set of record identifiers uniquely identifying records in the underlying database, these unique identifiers are created to a limited size (known as the KeySet Size) at query execution time, only when RowSet traversal goes beyond the existing set of unique row identifiers is another collection of unique identifiers assembled. This type of scrolling is sensitive to all changes in the underlying database, but insensitive to Additions or Deletions affecting records in the current RowSet scrolling across a current KeySet, once KeySet boundaries are crossed Insertions or Deletions are recognized. This mode of sensitivity provides increased performance and the expense of reduced sensitivity.

Concurrency Control

In addition to being sensitive to changes in the underlying database, Multi-User applications need to be able to protect users and application processes from the effects of one another when the same record or collection of records are being manipulated at the same time. The process by which these issue are addressed is known as Concurrency Control.

Concurrency control occurs in one of two ways, Optimistic or Pessimistic control.

Optimistic Concurrency Control - presumes that probability and frequency of multiple users and processes instigating changes to the same database records is low. As result when an end-user or process attempts to change records it first of all determines if the record values at the point of change are still the same as what they were at the time of retrieval. If they are unchanged at the point of change then the change occurs otherwise the change process is rejected and then re-attempted. Although this reduces concurrent user latency, it does have the knock on effect of reducing data integrity if changes rejections aren't managed carefully.

Pessimistic Concurrency Control - presumes that the probability and frequency of multiple user processing and instigating changes to the same records is high. As a result an end-user or process attempts to changes records it first of all secures Exclusive Locks on the records in question, performs the changes, and then releases the locks. Although this increases and preserves data integrity it does introduce concurrent use latency , which is perceived as performance degradation by the end-user or application developer.

OpenLink's Scrollable ResultSet and RowSet extensions for JDBC all the Multi-User JDBC solution issues raised in this section, our bundled and live online demonstrations enable you to evaluate this for yourself and ultimately make a knowledgeable JDBC Driver product and vendor selection.

ODBC Client Components For UNIX

Introduction

ODBC stands for Open Database Connectivity, it is a data access API implementation from Microsoft that conforms to the X/Open SQL CLI specification. ODBC links ODBC compliant applications to ODBC Drivers via the ODBC Driver Manager.

ODBC Driver Manager does exist in various forms under UNIX, but the definitive Driver Manager for UNIX is known as the "iODBC Driver Manager". Additional information regarding iODBC can be obtained from: http://www.openlinksw.com/iodbc .

The OpenLink ODBC Client Components for UNIX comprise the following :

- iODBC Driver Manager - A shared library that links ODBC applications to ODBC Drivers
- Generic ODBC Driver - A shared library (the file "libODBC.so") that provides database connectivity and data access services to ODBC based clients (these are applications written using the iODBC SDK)
- Sample ODBC Application - A simple program that can be used to verify your ODBC installation and working environment.

Installation

The OpenLink ODBC Client Components for UNIX are contained within the "odbc_sv.tz" compressed TAR archive file. This file is automatically presented to you via the OpenLink Software Download wizard when you enter UNIX as you client operating system.

The steps that follow describe the installation process:

Move the files "install.sh" and "oplrqb.taz" to an installation directory of your choice

1. Type in one of the following command s to extract the contents of the file "odbc_sv.taz":

   sh install.sh

   or

   install.sh

   or

   ./install.sh

2. Setup your operating environment by executing the command: . **openlink.sh**, you can also place the following entry in your ".profile" file:

   . **openlink.sh**

3. Proceed to the configuration stage of this process.

Configuration

The main configuration activity involves setting up logical references to the actual backend database engines that you wish to access via your UNIX based ODBC Driver. These local references are called Universal Data Source Names (UDSNs) and they are responsible for linking ODBC clients with actual OpenLink Data Access Drivers.

The OpenLink Admin Assistant is a Server Based HTML utility that enables you to manage UDSNs via your Web Browser. This utility provides wizards and a forms based user interfaces for performing its tasks.

Setting Up ODBC Data Sources

In the sections that follow, a step by guide and illustrative screen shots are used to demonstrate both approaches to setting up ODBC DSNs.

In the examples below lets presume that we are trying to create a UNIX based ODBC DSN on our machine called "opllinux" that will connect us to a Microsoft SQL Database on a Windows 95/98/NT Server. The critical database connection and network information for this setup (aka connection attributes) are as follows:

Network Alias of a Windows 95/98/NT/2000 Server machine (typically your application server) running OpenLink Server Components: "ntappserver"

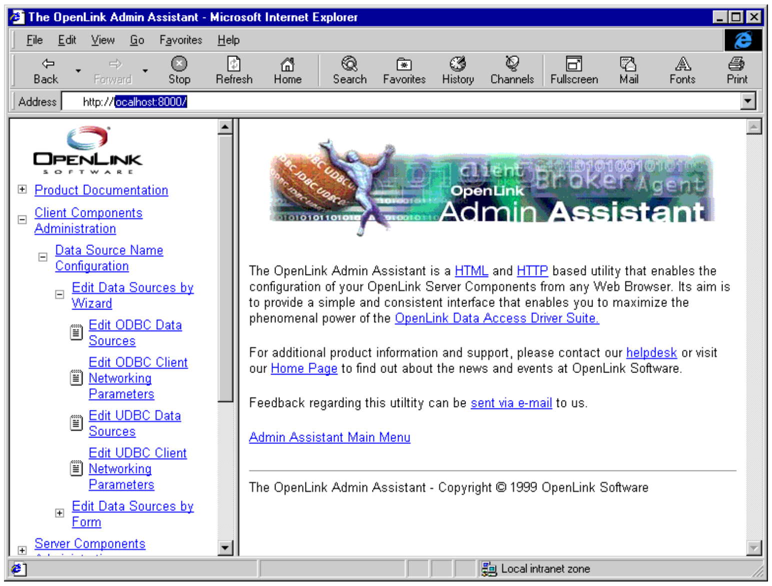
Network Alias of a Windows 95/98/NT/2000 Server machine running Microsoft SQL Server: "pluto"

Microsoft SQL Server Database Name: "pubs"

Wizard Based Data Source Management

1. Open up your Internet Browser and then enter the following URL: http://localhost:8000/ (note the OpenLink Web Assistant listens at port 8000 by default, this value is set at installation time).

2. Expand the menu by clicking on the "Client Components Administration", then "Data Source Name Configuration", and "Edit Data Sources by Wizard".


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Fonts Print
Address http://localhost:8000/
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Edit Data Sources by Form
Server Components
OpenLink Admin Assistant
Admin Assistant Main Menu
Local intranet zone

The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

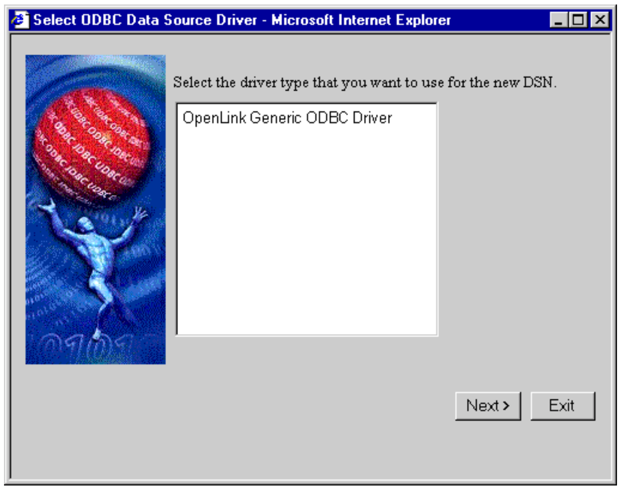
For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

Feedback regarding this utiltity can be sent via e-mail to us.

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

3. Click on the "Edit ODBC Data Sources" hyperlink, this takes you into the actual ODBC Data Source configuration wizard.

4. Click the "Add" button to commence the process of creating a new ODBC DSN, the wizard presents you with a list of ODBC Drivers installed on your system, select the driver identified as "OpenLink Generic ODBC" Driver and then click on the "Next" button.


Select ODBC Data Source Driver - Microsoft Internet Explorer
Next >
Exit

Select the driver type that you want to use for the new DSN.

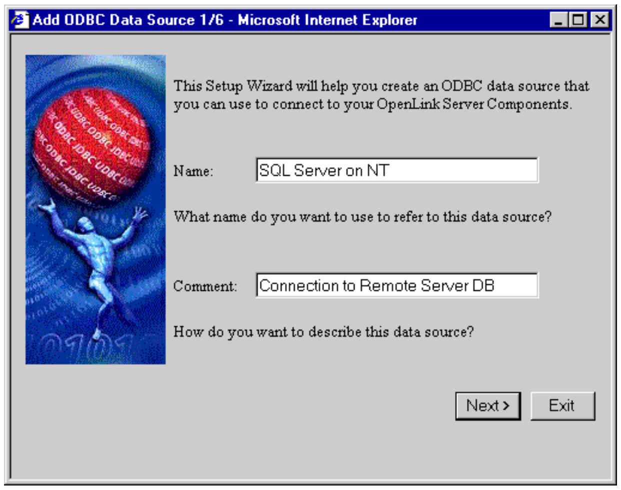
OpenLink Generic ODBC Driver

5. Enter values into the "Name" and "Description" fields as follows:

   "Name" - enter values that uniquely identify the DSN being created, our example uses the name "SQL Server on NT" to indicate that this DSN will be connecting you to a SQL Server database on an NT server.

   "Description" - enter values that provide additional information that helps in describing the purpose of the DSN that you are creating.

   Once completed click on the "Next" button.



This Setup Wizard will help you create an ODBC data source that you can use to connect to your OpenLink Server Components.

Name: SQL Server on NT

What name do you want to use to refer to this data source?

Comment: Connection to Remote Server DB

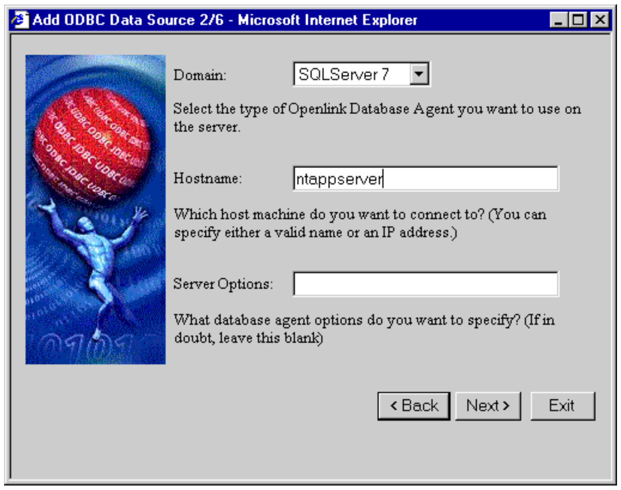
How do you want to describe this data source?

6. Enter values into the "Type", "Hostname", and "Server Options" fields as follows:

   "Domain Type" - enter a value that identifies the type of OpenLink Agent that will serve your ODBC client.

   "Hostname" - enter a value that identifies the server machine running your OpenLink Server Components.

   "Server Options" field - This field is only relevant when connecting to a "Progress" agent, this holds Progress session startup parameters such as -TM, -TB, -e, -l etc... You rarely need to enter these values on the client as they are best managed on the server within the OpenLink Session Rules Book.



Domain: SQLServer 7

Select the type of Openlink Database Agent you want to use on the server.

Hostname: ntappserver

Which host machine do you want to connect to? (You can specify either a valid name or an IP address.)

Server Options:

What database agent options do you want to specify? (If in doubt, leave this blank)

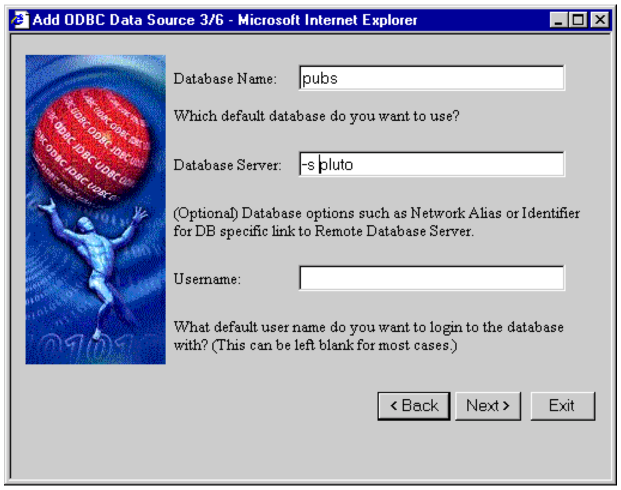
7. Enter values into the "Database", "Option", and "Server Options" fields as follows:

   "Database Name" - enter the name of an actual SQL Server database, in this case our example uses the database "pubs"

   "Database Server"- enter database server connection values for the database that your are connecting to, in this case enter valid SQL Server database server connection values (where "-s pluto" represent an actual SQL Server instance currently available on your network).

   "User ID" - enter a valid username for the database that you are connecting to, you can leave this blank and be prompted for values at actual database connect time.

   Click on the "Next" button.



Database Name: pubs

Which default database do you want to use?

Database Server: -s pluto

(Optional) Database options such as Network Alias or Identifier for DB specific link to Remote Database Server.

Username:

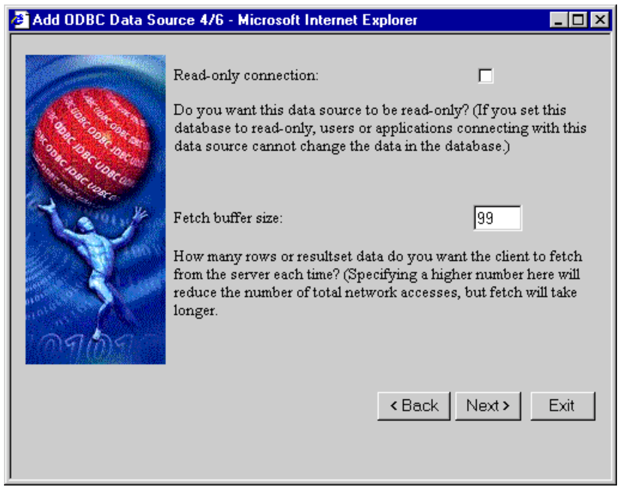
What default user name do you want to login to the database with? (This can be left blank for most cases.)

8. Enter values into the "Read-only connection" and "Fetch buffer size" fields as follows:

   "Read-only connection" - check this box if you require a read only session.

   "Fetch Buffer Size" - enter a value that represents the number of records that you would like your ODBC driver to retrieve during each network hop. A network hop represents the number of times your OpenLink ODBC send a message across the network to retrieve records from your remote database server. The feature can be used to improve ODBC record retrieval performance.

   Once completed click on the "Next" button.



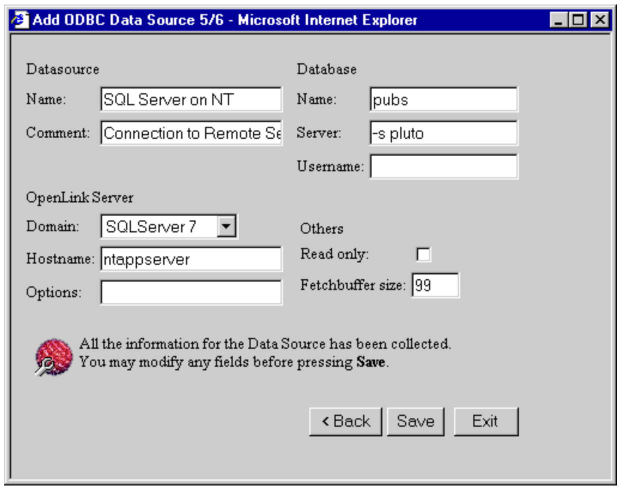
Read-only connection: ☐

Do you want this data source to be read-only? (If you set this database to read-only, users or applications connecting with this data source cannot change the data in the database.)

Fetch buffer size: 99

How many rows or resultset data do you want the client to fetch from the server each time? (Specifying a higher number here will reduce the number of total network accesses, but fetch will take longer.

9. You have now completed entering all the values that make up your new ODBC DSN, these values are collectively known as your ODBC DSN Attributes. Click on the "Save" button in order to store these values permanently on your hard disk.



Datasource
Name: SQL Server on NT
Comment: Connection to Remote Se

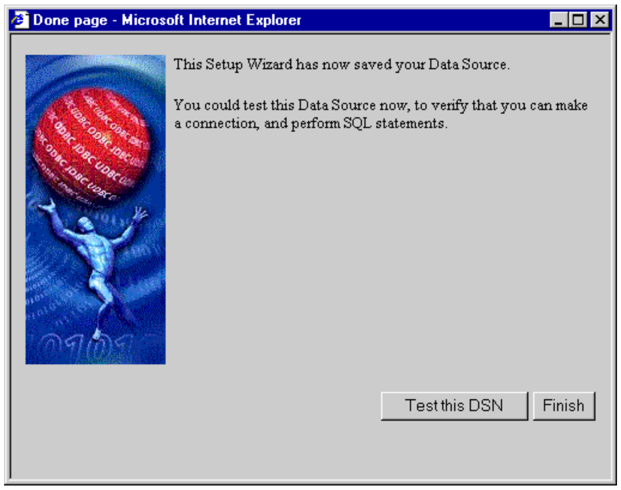
Database
Name: pubs
Server: -s pluto
Username:

OpenLink Server
Domain: SQLServer 7
Hostname: ntappserver
Options:

Others
Read only: ☐
Fetchbuffer size: 99

All the information for the Data Source has been collected.
You may modify any fields before pressing **Save**.

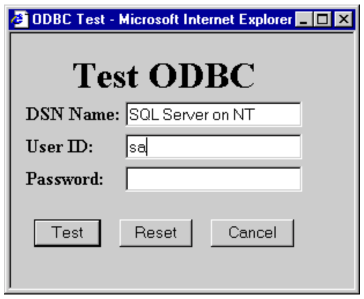
10. Click on the "Test this DSN" button, this enables you test and verify usability of the new ODBC DSN that you have created.



This Setup Wizard has now saved your Data Source.

You could test this Data Source now, to verify that you can make a connection, and perform SQL statements.

11. Click the "Test" button to actually commence the ODBC DSN Test process, you will be presented with dialogs that indicate success or failure at the end of this process.



Test ODBC

DSN Name: SQL Server on NT

User ID: sa

Password:

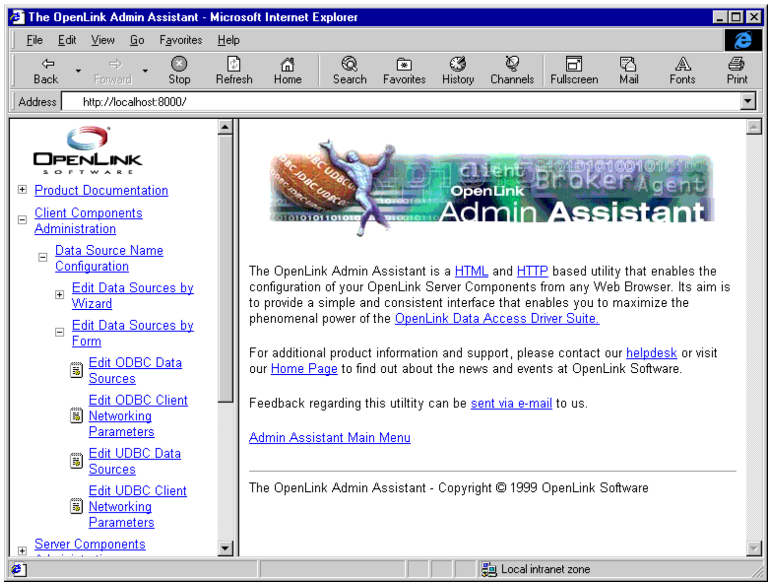
12. Click on the "exit" button to exit the ODBC DSN configuration wizard

Form Based Data Source Management

The OpenLink Admin assistant also allows the more experience OpenLink ODBC user to manage ODBC DSNs via a forms based interface. Like the wizard based approach this is done entirely from within your browser. In the sections that follow, a step by guide and illustrative screen shots are used to demonstrate the process of creating the same ODBC DSN created in the prior section using the Wizard approach.

1. Enter the following URL into your Web Browser (if the Admin Assistant isn't already initialized): http://localhost:8000 You will be presented with a screen similar to the one below. Notice that the "Client Component Administration", "Data Source Names Configuration" and "Edit Data Sources By Form" hyperlinks have been expanded.

   Click on the "Edit ODBC Data Sources" hyperlink to commence the process of creating a new ODBC DSN.



The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

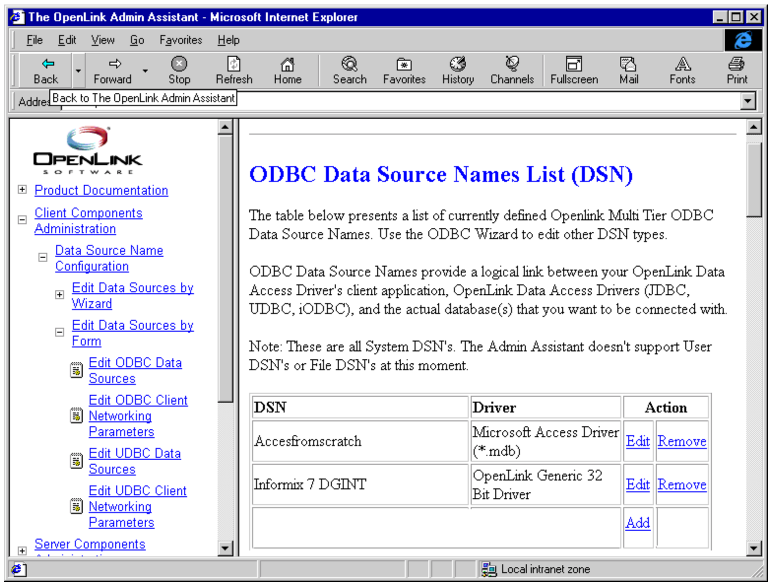
For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. On the right side of the Admin Assistant pane is your start page for configuring ODBC DSNs using the Forms approach. This page presents to you a list of currently configures ODBC DSNs on the machine "opllinux" (your UNIX client for this exercise). Click the "Add" hyperlink in the Action Column


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Fonts Print
Back to The OpenLink Admin Assistant
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit Data Sources by Form
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Server Components
Local intranet zone

ODBC Data Source Names List (DSN)

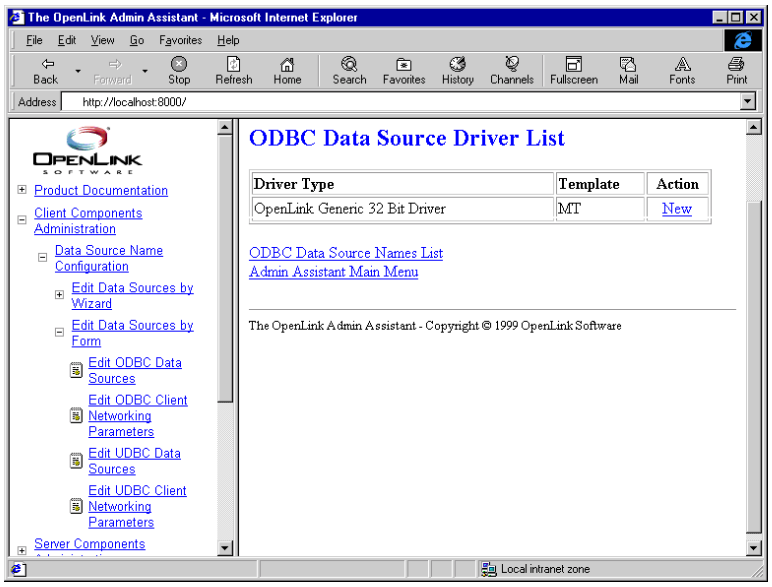
The table below presents a list of currently defined Openlink Multi Tier ODBC Data Source Names. Use the ODBC Wizard to edit other DSN types.

ODBC Data Source Names provide a logical link between your OpenLink Data Access Driver's client application, OpenLink Data Access Drivers (JDBC, UDBC, iODBC), and the actual database(s) that you want to be connected with.

Note: These are all System DSN's. The Admin Assistant doesn't support User DSN's or File DSN's at this moment.

DSN	Driver	Action	
Accesfromscratch	Microsoft Access Driver (*.mdb)	Edit	Remove
Informix 7 DGINT	OpenLink Generic 32 Bit Driver	Edit	Remove
		Add	

3. You are now presented with a table listing that comprises ODBC Drivers installed on your system, move on to the row that identifies the ODBC Driver that you will be creating your DSN for, then click on the "New" hyperlink.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Fonts Print
Address http://localhost:8000/
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit Data Sources by Form
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Server Components
Local intranet zone

ODBC Data Source Driver List

Driver Type	Template	Action
OpenLink Generic 32 Bit Driver	MT	New

ODBC Data Source Names List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

4. Enter values into the fields presented on the ODBC DSN form as follows:

   "Name" - enter values that uniquely identify the DSN being created, our example uses the name "SQL Server on NT" to indicate that this DSN will be connecting you to a SQL Server database on an NT server.

   "Description" - enter values that provide additional information that helps in describing the purpose of the DSN that you are creating.

   "UserName" - enter a valid username for the database that you are connecting to, you can leave this blank and be prompted for values at actual database connect time.

   "Password" - enter a default password to use, in most cases leave this blank and be prompted for values at actual database connect time.

   "Database Name" - enter the name of an actual SQL Server database, in this case our example uses the database "pubs"

"Read-only connection" - check this box if you require a read only session.

"No Login Dialog Box" - check this box if you do not to be prompted by your ODBC Driver for username and password dialog box at connect time.

"Database Server"- enter database server connection values for the database that your are connecting to, in this case enter valid SQL Server database server connection values (where "-s pluto" represent an actual SQL Server instance currently available on your network).

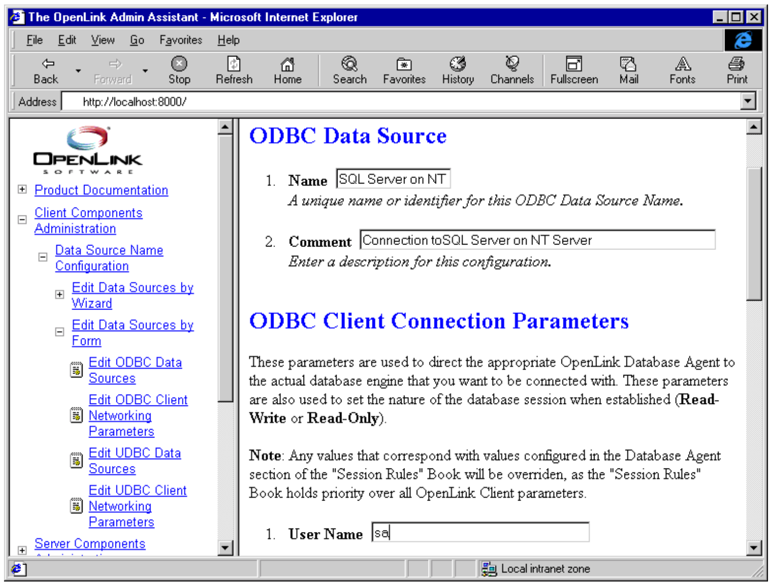
"Server Options" field - This field is only relevant when connecting to a "Progress" agent, this holds Progress session startup parameters such as -TM, -TB, -e, -l etc... You rarely need to enter these values on the client as they are best managed on the server within the OpenLink Session Rules Book.

"Server Type" - enter a value that identifies the type of OpenLink Agent that will serve your ODBC client.

"Hostname" - enter a value that identifies the server machine running your OpenLink Server Components.

"Row Buffer Size" - enter a value that represents the number of records that you would like your ODBC driver to retrieve during each network hop. A network hop represents the number of times your OpenLink ODBC send a message across the network to retrieve records from your remote database server. The feature can be used to improve ODBC record retrieval performance.

Note: The screen shot below is a snapshot of the ODBC DSN for, click on the right-hand scroll bar to see all the fields described above.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Fonts Print
Address http://localhost:8000/
OpenLink Software
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit Data Sources by Form
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Server Components
Local intranet zone

ODBC Data Source

1. **Name** SQL Server on NT
A unique name or identifier for this ODBC Data Source Name.

2. **Comment** Connection toSQL Server on NT Server
Enter a description for this configuration.

ODBC Client Connection Parameters

These parameters are used to direct the appropriate OpenLink Database Agent to the actual database engine that you want to be connected with. These parameters are also used to set the nature of the database session when established (**Read-Write** or **Read-Only**).

Note: Any values that correspond with values configured in the Database Agent section of the "Session Rules" Book will be overriden, as the "Session Rules" Book holds priority over all OpenLink Client parameters.

1. **User Name** sa

5. Click on the "Add" button at the foot of the page to complete the creation of your new ODBC DSN

ODBC Driver Session Settings

A number of configuration session parameters are available to adminstrators of OpenLink ODBC Drivers, these parameters can be managed via the Web Browser based Admin Assistant or by manually editing the file "udbc.ini" situated in the "bin" sub-directory of your OpenLink installation directory. These parameter enable you tailor the behaviour of your ODBC Drivers for UNIX in line with the requirements of your ODBC based soultions and any general infrastructural requirements that you may have.

The list of configurable session parameters and their descriptions are as follows:

Parameter	Default Value	Description

BrokerTimeout	30	The time (in secs) that the OpenLink ODBC client application will wait for the OpenLink Request Broker to accept or reject a database connection.
ReceiveTimeout	60	The time (in secs) that the OpenLink ODBC client will wait for an ODBC request to be completed.
RetryTimeout	5	The amount of wait time (in secs) before the OpenLink ODBC client attempts to re-execute a failed call. After each attempt this value is doubled. The life time of this value never exceeds the BrokerTimeout during intial connection establishment, and never exceeds the ReceiveTimeout when sessions have been established.
SendSize	4096	The size (in kilobytes) of the OpenLink ODBC client's outward bound message packets.
ReceiveSize	16000	The size (in kilobytes) of the OpenLink ODBC client's server bound message packets.
DebugFile	empty	When this variable contains a valid file and path reference, all ODBC API calls will be logged and stored in the file name referenced.

Sample Application

OpenLink also provides a sample ODBC based dynamic SQL application that enables you verify usability of your ODBC installation.The sample application is situated within the "samples" sub-directory below your OpenLink installation directory. The ODBC sample application is called "odbctest".

Using Sample ODBC Application

The following steps guide you through the process of successfully utilising this sample application. This exercise presumes that we are connecting to a DSN called "SQL Server on NT", which connects us to a remote SQL Server Database hosted on a machine called "pluto" via the OpenLink Server components on an Windows 95/98/NT/2000 application server called "ntappserver".

1. Ensure that your OpenLink Request Broker is up and running on the machine "ntappserver" (you can quickly confirm this by opening up your browser and entering the following URL: http://ntappserver:8000)

2. At your UNIX command prompt type in the following command:

   odbctest

3. Enter a full or partial ODBC connect string at the ODBC applications command prompt, some examples are listed below:

   - for a list of DSNs on your system enter "?"

   - to connect to the DSN called "SQL Server on NT" type (this is a partial connect string): DSN=SQL Server On NT

   - to enter a username and a blank password combination along with the DSN type (this is a partial connect string only becuase we have a seperate server hosting the OpenLink Server and Microsoft SQL Server components): DSN=SQL Server on NT;UID=sa;PWD=

   - to enter a directive that instructs the OpenLink Server components to connect to the remote SQL Server hosted on the machine called "pluto", type the following (this is a full connect string for this particular scenario): DSN=SQL Server on NT;UID=sa;PWD=;OPTIONS=-s pluto

4. If step 4 is successful you are now ready to execute SQL interactively against your remote database, to do this enter the following SQL command:

   select * from authors

5. To quit this application type in "exit" at the SQL command line prompt.

UDBC Client Components For UNIX

Introduction

UDBC stands for Universal Database Connectivity, it is a data access API implementation from OpenLink Software that conforms to the X/Open SQL CLI specification. Unlike the Open Database Connectivity (ODBC) specification UDBC does not include a separate Driver Manager Component, instead it provides you with direct connectivity to backend databases. In some quarters UDBC is actually seen as a Generic Native API to backend databases that adhere to the X/open SQL CLI specification.

ODBC and UDBC share identical APIs, thus applications that are written to either specification are guaranteed a high degree of compatibility.

The UDBC Client Components for UNIX comprise the following :

Generic UDBC Driver - A shared library (the file "libudbc.so") that provides database connectivity and data access services to UDBC based clients (these are applications written using the UDBC SDK)

Sample UDBC Application - A simple program that can be used to verify your UDBC installation and working environment.

Installation

The OpenLink UDBC Client Components for UNIX are contained within the "udbc_sv.tz" compressed TAR archive file. This file is automatically presented to you via the OpenLink Software Download wizard when you enter UNIX as your client operating system.

The steps that follow describe the installation process:

1. Move the files "install.sh" and "oplrqb.taz" to an installation directory of your choice

2. Type in one of the following command s to extract the contents of the file "udbc_sv.taz":

   ```
   sh install.sh
   ```

   or

   ```
   install.sh
   ```

   or

   ```
   ./install.sh
   ```

3. Setup your operating environment by executing the command: . openlink.sh, you can also place the following entry in your ".profile" file:

   ```
   . openlink.sh
   ```

4. Proceed to the configuration stage of this process.

Configuration

The main configuration activity involves setting up logical references to the actual backend database engines that you wish to access via your UNIX based UDBC Driver. These local references are called Universal Data Source Names (UDSNs) and they are responsible for linking UDBC clients with actual OpenLink Data Access Drivers.

The OpenLink Admin Assistant is a Server Based HTML utility that enables you to manage UDSNs via your Web Browser. This utility provides wizards and a forms based user interfaces for performing its tasks.

Setting Up UDBC Data Sources

In the sections that follow, a step-by guide and illustrative screen shots are used to demonstrate both approaches to setting up UDSNs.

In the examples below lets presume that we are trying to create a UNIX based UDSN that will connect us to a Microsoft SQL Database on a Windows 95/98/NT/2000 Server. The critical database connection and network information for this setup (aka connection attributes) are as follows:

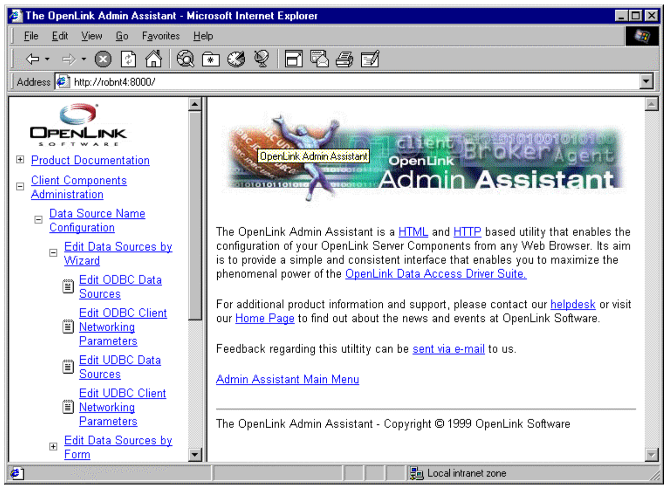
Network alias of Windows 95/98/NT/2000 machine running your OpenLink Server components: ntappserver

Network Alias of Server machine running Microsoft SQL Server (also the machine on to which you have installed the OpenLink Server Components for Windows NT): pluto

Microsoft SQL Server Database Name: pubs

Wizard Based Data Source Name Management

1. Open up your Internet Browser and then enter the following URL: http://localhost:8000/ (note the OpenLink Web Assistant listens at port 8000 by default, this value is set at installation time). Click on the "Client Components Administration" hyperlink to expand this menu option further, then "Data Source Name Configuration" then click on the "Edit Data Sources by Wizard". Now Click the "Edit UDBC Data Sources" hyperlink.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address http://robnt4:8000/
OpenLink Software
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Edit Data Sources by Form
OpenLink Admin Assistant
Local intranet zone

The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

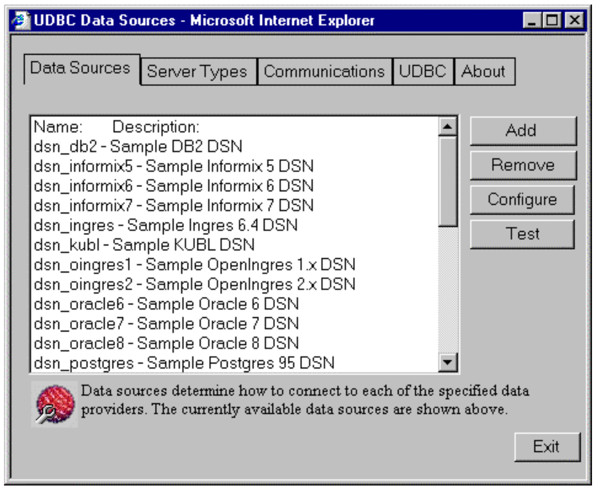
For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. You will be presented with the actual UDBC Data Source configuration wizard. Click the "Add" button



Name: Description:
dsn_db2 - Sample DB2 DSN
dsn_informix5 - Sample Informix 5 DSN
dsn_informix6 - Sample Informix 6 DSN
dsn_informix7 - Sample Informix 7 DSN
dsn_ingres - Sample Ingres 6.4 DSN
dsn_kubl - Sample KUBL DSN
dsn_oingres1 - Sample OpenIngres 1.x DSN
dsn_oingres2 - Sample OpenIngres 2.x DSN
dsn_oracle6 - Sample Oracle 6 DSN
dsn_oracle7 - Sample Oracle 7 DSN
dsn_oracle8 - Sample Oracle 8 DSN
dsn_postgres - Sample Postgres 95 DSN

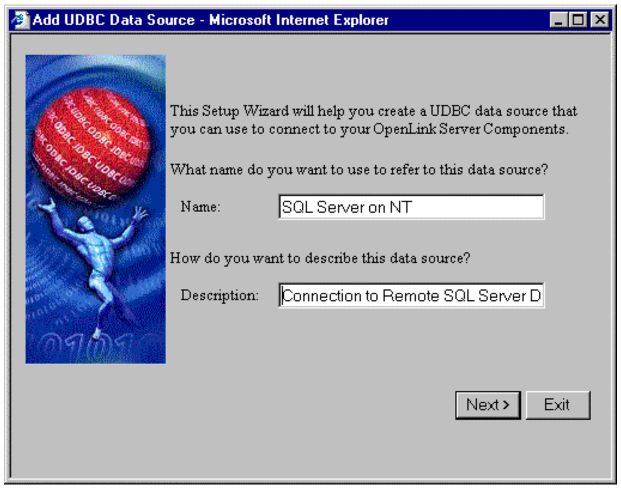
Data sources determine how to connect to each of the specified data providers. The currently available data sources are shown above.

3. Enter values into the "Name" and "Description" fields as follows:

   "Name" - enter values that uniquely identify the DSN being created, our example uses the name "SQL Server on NT" to indicate that this DSN will be connecting you to a SQL Server database on an NT server.

   "Description" - enter values that provide additional information that helps in describing the purpose of the DSN that you are creating.

   Once completed click on the "Next" button.



This Setup Wizard will help you create a UDBC data source that you can use to connect to your OpenLink Server Components.

What name do you want to use to refer to this data source?

Name: SQL Server on NT

How do you want to describe this data source?

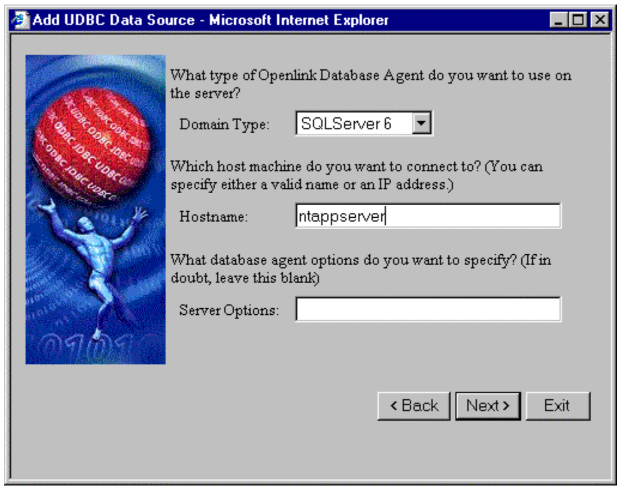
Description: Connection to Remote SQL Server D

4. Enter values into the "Domain Type", "Hostname", and "Server Options" fields as follows:

   "Domain Type" - enter a value that identifies the type of OpenLink Agent that will serve your UDBC client.

   "Hostname" - enter a value that identifies the server machine running your OpenLink Server Components.

   "Server Options" field - This field is only relevant when connecting to a "Progress" agent, this holds Progress session startup parameters such as -TM, -TB, -e, -l etc... You rarely need to enter these values on the client as they are best managed on the server within the OpenLink Session Rules Book.



What type of Openlink Database Agent do you want to use on the server?

Domain Type: SQLServer 6

Which host machine do you want to connect to? (You can specify either a valid name or an IP address.)

Hostname: ntappserver

What database agent options do you want to specify? (If in doubt, leave this blank)

Server Options:

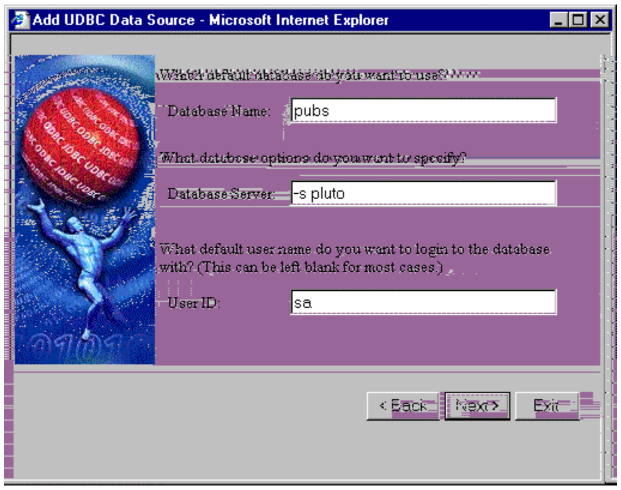
5. Enter values into the "Database Name", "Database Server", and "User ID" fields as follows:

   "Database Name" - enter the name of an actual SQL Server database, in this case our example uses the database "pubs"

   "Database Server"- enter database server connection values for the database that your are connecting to, in this case enter valid SQL Server database server connection values (where "-s pluto" represent an actual SQL Server instance currently available on your network).

   "User ID" - enter a valid username for the database that you are connecting to, you can leave this blank and be prompted for values at actual database connect time.

   Click on the "Next" button.



Add UDBC Data Source - Microsoft Internet Explorer

What default database do you want to use?

Database Name: pubs

What database options do you want to specify?

Database Server: -s pluto

What default user name do you want to login to the database with? (This can be left blank for most cases.)

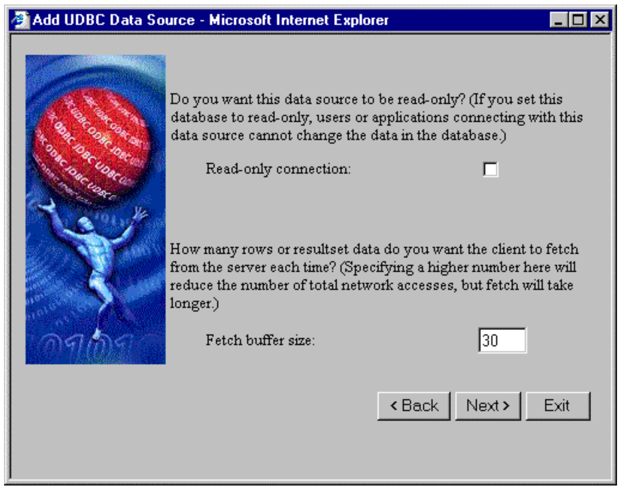
User ID: sa

6. Enter values into the "Read-only connection" and "Fetch buffer size" fields as follows:

   "Read-only connection" - check this box if you require a read only session.

   "Fetch Buffer Size" - enter a value that represents the number of records that you would like your UDBC driver to retrieve during each network hop. A network hop represents the number of times your OpenLink UDBC sends a message across the network to retrieve records from your remote database server. The feature can be used to improve UDBC record retrieval performance.

   Once completed click on the "Next" button.



Add UDBC Data Source - Microsoft Internet Explorer

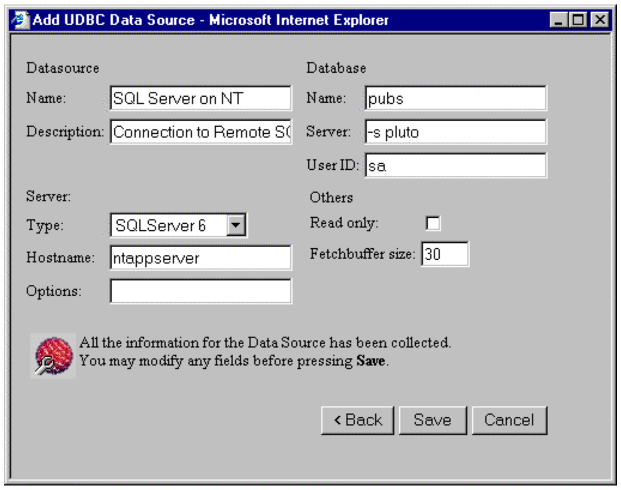
Do you want this data source to be read-only? (If you set this database to read-only, users or applications connecting with this data source cannot change the data in the database.)

Read-only connection: ☐

How many rows or resultset data do you want the client to fetch from the server each time? (Specifying a higher number here will reduce the number of total network accesses, but fetch will take longer.)

Fetch buffer size: 30

7. You have now completed entering all the values that make up your new UDBC DSN, these values are collectively known as your UDBC DSN Attributes. Click on the "Save" button in order to store these values permanently on your hard disk.



Datasource
Name: SQL Server on NT
Description: Connection to Remote S(

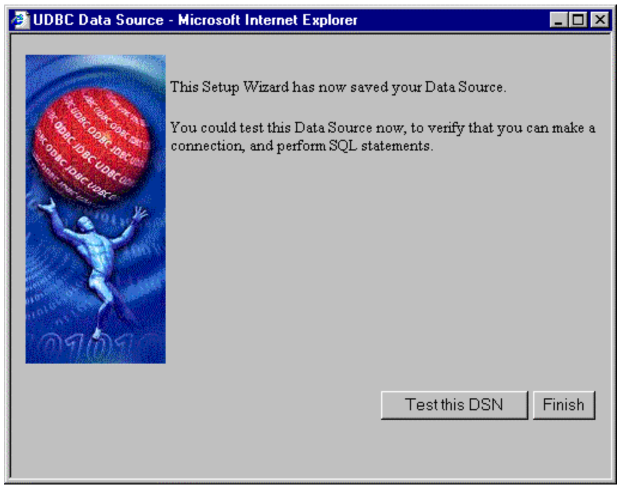
Database
Name: pubs
Server: -s pluto
User ID: sa

Server:
Type: SQLServer 6
Hostname: ntappserver
Options:

Others
Read only: ☐
Fetchbuffer size: 30

All the information for the Data Source has been collected.
You may modify any fields before pressing **Save**.

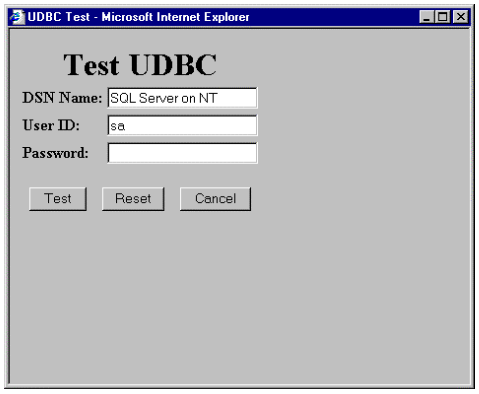
8. Click on the "Test this DSN" button, this enables you to test and verify usability of the new UDBC DSN that you have created.



This Setup Wizard has now saved your Data Source.

You could test this Data Source now, to verify that you can make a connection, and perform SQL statements.

9. Click the "Test" button to actually commence the UDBC DSN Test process, you will be presented with dialogs that indicate success or failure at the end of this process.



Test UDBC

DSN Name: SQL Server on NT
User ID: sa
Password:

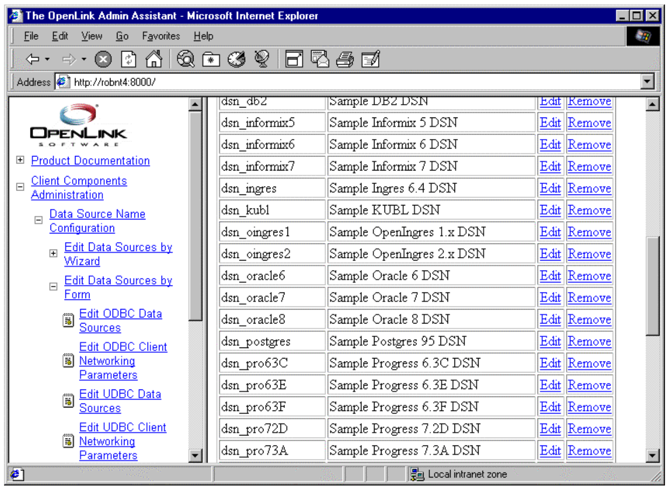
10. Click on the "exit" button to exit the UDBC DSN configuration wizard

Forms Based Data Source Name Management

The OpenLink Admin assistant also allows the more experienced OpenLink UDBC user to manage UDBC DSNs via a forms based interface. Like the wizard based approach this is done entirely from within your browser. In the sections that follow, a step by guide and illustrative screen shots are used to demonstrate the process of creating the same UDBC DSN created in the prior section using the Wizard approach.

1. Enter the following URL into your Web Browser (if the Admin Assistant isn't already initialized): http://localhost:8000 You will be presented with a screen similar to the one below. Notice that the "Data Source Names(s) Settings" and "Edit Data Sources By Form" hyperlinks have been expanded.

   On the right side of the Admin Assistant pane is your start page for configuring UDBC DSNs using the Forms approach. This page presents to you a list of currently configured UDBC DSNs on the machine "localhost" (your UNIX client for this exercise). Click on the "Add" hyperlink to commence the process of creating a new UDBC DSN.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address http://robnt4:8000/
OpenLink Software
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit Data Sources by Form
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Local intranet zone

dsn_db2	Sample DB2 DSN	Edit	Remove
dsn_informix5	Sample Informix 5 DSN	Edit	Remove
dsn_informix6	Sample Informix 6 DSN	Edit	Remove
dsn_informix7	Sample Informix 7 DSN	Edit	Remove
dsn_ingres	Sample Ingres 6.4 DSN	Edit	Remove
dsn_kubl	Sample KUBL DSN	Edit	Remove
dsn_oingres1	Sample OpenIngres 1.x DSN	Edit	Remove
dsn_oingres2	Sample OpenIngres 2.x DSN	Edit	Remove
dsn_oracle6	Sample Oracle 6 DSN	Edit	Remove
dsn_oracle7	Sample Oracle 7 DSN	Edit	Remove
dsn_oracle8	Sample Oracle 8 DSN	Edit	Remove
dsn_postgres	Sample Postgres 95 DSN	Edit	Remove
dsn_pro63C	Sample Progress 6.3C DSN	Edit	Remove
dsn_pro63E	Sample Progress 6.3E DSN	Edit	Remove
dsn_pro63F	Sample Progress 6.3F DSN	Edit	Remove
dsn_pro72D	Sample Progress 7.2D DSN	Edit	Remove
dsn_pro73A	Sample Progress 7.3A DSN	Edit	Remove

2. Enter values into the fields presented on the UDBC DSN form as follows:

   "Name" - enter values that uniquely identify the DSN being created, our example uses the name "SQL Server on NT" to indicate that this DSN will be connecting you to a SQL Server database on an NT server.

   "Description" - enter values that provide additional information that helps in describing the purpose of the DSN that you are creating.

   "Read-only connection" - check this box if you require a read only session.

   "No Login Dialog Box" - check this box if you do not want to be prompted by your UDBC Driver for username and password dialog box at connect time.

   "Type" - enter a value that identifies the type of OpenLink Agent that will serve your UDBC client.

   "Hostname" - enter a value that identifies the server machine running your OpenLink Server Components.

   "Server Options" field - This field is only relevant when connecting to a "Progress" agent, this holds Progress session startup parameters such as -TM, -TB, -e, -l etc... You rarely need to enter these values on the client as they are best managed on the server within the OpenLink Session Rules Book.

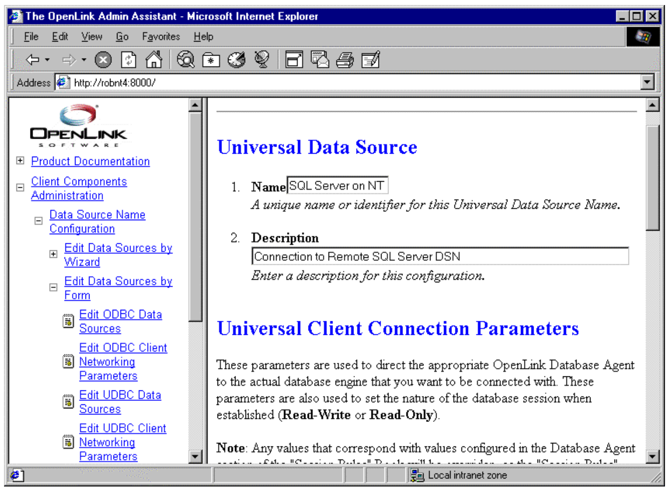
   "Database" - enter the name of an actual SQL Server database, in this case our example uses the database "pubs"

   "Connect Options" - enter database server connection values for the database that your are connecting to, in this case enter valid SQL Server database server connection values (where "-s pluto" represent an actual SQL Server instance currently available on your network).

   "UserName" - enter a valid username for the database that you are connecting to, you can leave this blank and be prompted for values at actual database connect time.

   "Fetch Buffer Size" - enter a value that represents the number of records that you would like your UDBC driver to retrieve during each network hop. A network hop represents the number of times your OpenLink UDBC driver sends a message across the network to retrieve records from your remote database server. The feature can be used to improve UDBC record retrieval performance.

Note: The screen shot below is a snapshot of the UDBC DSN for, click on the right-hand scroll bar to see all the fields described above.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address http://robnt4:8000/
OpenLink Software
Product Documentation
Client Components Administration
Data Source Name Configuration
Edit Data Sources by Wizard
Edit Data Sources by Form
Edit ODBC Data Sources
Edit ODBC Client Networking Parameters
Edit UDBC Data Sources
Edit UDBC Client Networking Parameters
Local intranet zone

Universal Data Source

1. Name: SQL Server on NT
   A unique name or identifier for this Universal Data Source Name.
2. Description: Connection to Remote SQL Server DSN
   Enter a description for this configuration.

Universal Client Connection Parameters

These parameters are used to direct the appropriate OpenLink Database Agent to the actual database engine that you want to be connected with. These parameters are also used to set the nature of the database session when established (**Read-Write** or **Read-Only**).

Note: Any values that correspond with values configured in the Database Agent

3. Click on the "Add" button at the foot of the page to complete the creation of your new UDBC DSN

UDBC Driver Session Settings

A number of configuration session parameters are available to adminstrators of OpenLink UDBC Drivers. These parameters can be managed via the Web Browser based Admin Assistant or by manually editing the file "udbc.ini" situated in the "bin" sub-directory of your OpenLink installation directory. These parameters enable you to tailor the behaviour of your UDBC Drivers for UNIX in line with the requirements of your UDBC based soultions and any general infrastructural requirements that you may have.

The list of configurable session parameters and their descriptions are as follows:

Parameter	Default Value	Description
BrokerTimeout	30	The time (in secs) that the OpenLink UDBC client application will wait for the OpenLink Request Broker to accept or reject a database connection.
ReceiveTimeout	60	The time (in secs) that the OpenLink UDBC client will wait for an UDBC request to be completed.
RetryTimeout	5	The amount of wait time (in secs) before the OpenLink UDBC client attempts to re-execute a failed call. After each attempt this value is doubled. The life time of this value never exceeds the BrokerTimeout during intial connection establishment, and never exceeds the ReceiveTimeout when sessions have been established.
SendSize	4096	The size (in kilobytes) of the OpenLink UDBC client's outward bound message packets.
ReceiveSize	16000	The size (in kilobytes) of the OpenLink UDBC client's server bound message packets.
DebugFile	empty	When this variable contains a valid file and path reference, all UDBC API calls will be logged and stored in the file name referenced.

Sample Application

OpenLink also provides a sample UDBC based dynamic SQL application that enables you verify usability of your UDBC installation.The sample application is situated within the "samples" sub-directory below your OpenLink installation directory. The UDBC sample application is called "udbctest".

Using Sample UDBC Application

The following steps guide you through the process of successfully utilising this sample application. This exercise presumes that we are connecting to a UDSN called "SQL Server on NT", which connects us to a remote SQL Server Database hosted on a machine called "pluto" via the OpenLink Server components on an Windows 95/98/NT/2000 application server called "ntappserver"..

1. Ensure that your OpenLink Request Broker is up and running on the machine "ntappserver" (you can quickly confirm this by opening up your browser and entering the following URL:

   http://ntappserver:8000/)

2. At your UNIX command prompt type in the following command:

   ```
   udbctest
   ```

3. Enter a full or partial UDBC connect string at the UDBC applications command prompt, some examples are listed below:

   - for a list of UDSNs on your system enter "?"

   - to connect to the UDSN called "SQL Server on NT" type (this is a partial connect string):

   ```
   DSN=SQL Server On NT
   ```

   - to enter a username and a blank password combination along with the UDSN type (this is a partial connect string only becuase we have a seperate server hosting the OpenLink Server and Microsoft SQL Server components):

   ```
   DSN=SQL Server on NT;UID=sa;PWD=
   ```

   - to enter a directive that instructs the OpenLink Server components to connect to the remote SQL Server hosted on the machine called "pluto", type the following (this is a full connect string for this particular scenario):

   ```
   DSN=SQL Server on NT;UID=sa;PWD=;OPTIONS=-s pluto
   ```

4. If step 4 is successful you are now ready to execute SQL interactively against your remote database, to do this enter the following SQL command:

   ```
   select * from authors
   ```

5. To quit this application type in "exit" at the SQL command line prompt.

OpenLink UDBC SDK For Unix

What is UDBC?

UDBC is OpenLink's acronym for Universal DataBase Connectivity.

In short, UDBC is the merge between the X/Open - SAG interface and the Microsoft ODBC interface for client database applications. UDBC was developed to enable software engineers to rapidly develop database independent applications without being concerned about issues like portability, network programming and database specific interfacing problems.

With UDBC it is possible to write database applications that are written according to the X/Open standard, but this SDK also provides access methods for the Microsoft ODBC programming interface.

Installation

Copy the file "udbcxxx.taz" and "install.sh" to your OpenLink base installation directory. If such a directory does not exist, please create one.

From within the OpenLink base installation directory run the "install.sh" script :

```
sh install.sh
```

This will automatically extract the contents of the .taz file into the following sub-directory structure within your OpenLink base installation directory.

Files in the SDK

```
udbcsdk
udbcsdk/lib
udbcsdk/lib/libudbc.sl.1.5
udbcsdk/lib/libudbc.sl.1
udbcsdk/lib/libudbc.sl
udbcsdk/lib/libudbc.la
udbcsdk/lib/libudbc.a
udbcsdk/include
udbcsdk/include/udbc
udbcsdk/include/udbc/udbctype.h
udbcsdk/include/udbc/udbcodbc.h
udbcsdk/include/udbc/udbcsag.h
udbcsdk/include/udbc/udbcimpl.h
udbcsdk/include/udbc/udbcprto.h
udbcsdk/include/udbc/udbcext.h
udbcsdk/include/libudbc.h
udbcsdk/examples
udbcsdk/examples/udbctest.c
udbcsdk/examples/Makefile
udbcsdk/examples/udbctest
udbcsdk/doc
udbcsdk/doc/udbc.doc
udbcsdk/doc/udbc.ini
```

Configuration

UDBC like ODBC is based on the notion of logical references to database backends through the use of Data Source Names, under UDBC these are described as Universal Data Source Names (UDSNs) due to the cross data access standard nature of these Data Source Names (they are usable by OpenLink Drivers for ODBC and JDBC™).

The OpenLink Data Access Driver Suite 1.5.5 requires you to manually create UDSNs by manually editing a designated file, and then creating your your UDSNs within this file (the OpenLink installer provides a template Universal Data Source Names file called "udbc.ini" - structured in exactly the same manner as a Windows "INI" configuration file). The OpenLink Data Access Driver Suite 3.2 on the other hand includes a HTML and HTTP based utility called the OpenLink "Web Configurator" that enables you create, configure, and manage UDSNs via any Web Browser.

You identify your UDSN file to UDBC Drivers via the Environment Variable "UDBCINI", if you do not set this variable OpenLink UDBC Drivers will use the template "udbc.ini" file that comes with your UDBC SDK installation. The UDBCINI environment variable is set using the command:

```
UDBCINI=/usr/openlink/bin/udbc.ini ; export UDBCINI
```

Note: The above presumes that your OpenLink installation's base directory is: /usr/openlink .

The example below depicts the structure of a UDSN named "dsn_oracle7", used to facilitate connections with an Oracle Database residing on a machine named "mercury".

```
[dsn_oracle7]
Database = ORCL
UserName = scott
Password = tiger
Host = mercury
ServerType = Oracle 7
FetchBufferSize = 60
Description = Oracle Database Connections
```

Your general OpenLink UDBC settings are configured within the "[UDBC]" section of your chosen UDSN file, an extract from this file is depicted below:

```
[UDBC]
;DebugFile = /tmp/udbc.out ; Write client debugging output

[Communications]
ReceiveTimeout = 120 ; max. time to complete a request
BrokerTimeout = 30 ; max. time to wait for oplrqb
RetryTimeout = 5 ; Retry time for oplrqb locating
SendSize = 4096 ; RPC send buffer size
ReceiveSize = 16000 ; RPC receive buffer size
ShowErrors = Y ; Pass RPC errors to applications
DataEncryption = N ; Encrypt outgoing data
```

Note: It is highly recommended that you use the UDSN templates provided by your OpenLink UDBC SDK installation for your initial connection, as the OpenLink installer sets up the appropriate UDBC environment for you at installation time.

The table below describes each OpenLink UDBC attribute required for making a connection from your UDBC based application to your backend database engine.

Section and Keys	***Default Value***	***Corresponding Connect String option***	***Description***
[UDBC Global]			Global UDBC client communications settings.
BrokerTimeout	30		The time (in secs) that the client application will wait for the request broker (oplrqb) to accept or reject a database connection.
ReceiveTimeout	60		The time (in secs) that the client application will wait for a database agent to finish a request.
PortMapTimeout	30		When using the ONC portmap or rpcbind name services, this is the time (in secs) that the client application will wait for a reply from the portmapper.
SendSize	4096		RPC send buffer size. A value of 0 will cause the application to use the default value.
ReceiveSize	16000		RPC receive buffer size. A value of 0 will cause the application to use the default value.
DebugFile	no debugging		When this variable contains a valid filename, the client applications will write debugging information to that file. Do not use this in the systemwide ini file! (/etc/udbc.ini) This file will contain detailed UDBC call tracing.
[DSN_name]		DSN=DSN_name	Name of the Data Source
Host		HOST=name	Name of host to contact (hostname or network address)
ServerType		SVT=type	Specify agent domain This is used by the broker to determine which agent section to connect the client request to using mapping rules.
ServerOptions			Server startup options specific to agent/database.
Database		DATABASE=db	Database to use. Agent/database specific.
Options		OPTIONS=opts	Database connect options. Agent/database specific.
UserName		UID=username	Username to connect as.
		PWD=password	Password for user.
ReadOnly		READONLY=<y|n>	Specify Y(es) for read-only (ro) or N(o) for read-write (rw) connections. Read only connections are sometimes faster, but can never modify any database.
FetchBufferSize	5	FBS=<value>	Number of rows (records) to be fetched per call from database agent. Values range from 1 to 99

UDBC Client to Server Mechanics

When connecting to a database, the application sends the UDBC engine a connect string that contains the necessary parameters to complete the database connection.

When the connect string is empty, or a null pointer, the [Default] section is used for the remaining connect parameters.

When the connect string contains a "DSN=<name>" setting, then the section [name] (case insensitive) will be picked from the "udbc.ini" file in use.

All other settings in the connect string overrule the values in the udbc.ini file when supplied.

Note: the OpenLink "Session Rules" Book will determine the ultimate database destination of your UDBC Driver.

Example UDBC Connect Strings:

"" (empty)

Use the values from the [Default] section. There must be at least a Host= definition in that section.

"DSN=dsn_Oracle7"

Use the values from the [dsn_Oracle7] section.

"DSN=dsn_Oracle7;UID=scott;PWD=tiger"

Use the values from the [dsn_Oracle7] section, and overrides the value for UserName. Also use the password "tiger" to gain access to the database.

"HOST=mercury;SVT=Oracle 7;DATABASE=ORCL;UID=scott/tiger"

Connect to an Oracle 7 database on the machine "mercury" .

"READONLY=Y"

Make a read-only connection to the [Default] database.

The Sample UDBC Application "UDBCTEST"

Your OpenLink UDBC SDK installation contains a sample interactive dynamic SQL application called "udbctest", this application is situated within your OpenLink installation's "udbcsdk/examples" sub-directory.

Then run the application by typing the following command from the command line prompt:

```
udbctest or ./udbctest
```

The application will prompt you for a connect string (assuming we are using the UDSN example depicted above):

```
Enter UDBC connect string: DSN=generic_Oracle7
```

If you connect to the database successfully you will get a SQL prompt:

```
SQL>
```

Type a valid SQL statement and then press <return> to send your request to your UDSN

```
SQL> select * from <table name>
```

or Use an OpenLink quick table list command:

```
    SQL> tables
```

If the query returns data and was successful, the data will be scrolled down the screen.

Otherwise you will receive an error message.

Type in "exit" to quit the application

Compiling Sample Program

To compile the application type the following command from the command line :

```
make -f udbctest.mk
```

Developing UDBC Applications

This UDBC SDK only implements an interface for the 'C' programming language.

To write an UDBC application,you must perform the following tasks::

a. include the file "libudbc.h" in your 'C' program(s)

b. link the application with the "libudbc.a" and the "librpc.a" libraries.

Note: Some UNIX systems also need -lsocket, -lnsl_s or both.

Further Reading:

"Data Management: SQL Call Level Interface (CLI)"

from X/Open in conjunction with SQL Access Group

ISBN: 1-872630-63-4

X/Open Document Number: S203

Microsoft ODBC API documentation:
http://www.microsoft.com/data/reference/odbc2.htm

Notes For Using The OpenLink Unix ODBC Drivers

Introduction

Welcome to the combined release of OpenLink ODBC for Unix and the free iODBC driver manager. This kit will provide you with everything you need in order to develop ODBC compliant applications under UNIX.

This kit consists of a number of parts:

1. The free iODBC driver manager. This is a complete implementation of an ODBC driver manager, released under the GNU Library General Public License (LGPL). See http://www.openlinksw.com/iodbc for additional information.

2. Generic OpenLink ODBC Driver . The driver provided in the form of a shared library file, with a system-dependent name.

3. A sample ODBC application and source code. The sample application is an Interactive Dynamic SQL Interpreter. You are free to use this application and source code as you see fit.

Installation of Run-time Distribution

Copy the file "odbcxxx.taz" and "install.sh" to your OpenLink base installation directory. If such a directory does not exist, please create one.

From within the OpenLink base installation directory run the "install.sh" script using the command:

```
sh install.sh
```

This will automatically extract the contents of the .taz file into the following sub-directory structure within your OpenLink base installation directory.

Files contained within the UNIX ODBC SDK

```
odbcsdk/lib
odbcsdk/lib/libiodbc.sl.2.12
odbcsdk/lib/libiodbc.sl.2
odbcsdk/lib/libiodbc.sl
odbcsdk/lib/libiodbc.la
odbcsdk/lib/libiodbc.a
odbcsdk/lib/oplodbc.sl.1.0
odbcsdk/lib/oplodbc.sl.1
odbcsdk/lib/oplodbc.sl
odbcsdk/lib/oplodbc.la
odbcsdk/lib/oplodbc.a
odbcsdk/include
odbcsdk/include/iodbc.h
odbcsdk/include/isql.h
odbcsdk/include/isqlext.h
odbcsdk/include/udbcext.h
odbcsdk/examples
odbcsdk/examples/odbctest.c
odbcsdk/examples/Makefile
odbcsdk/examples/odbc.ini
odbcsdk/examples/odbctest
odbcsdk/doc
odbcsdk/doc/STARTUP.DOC
odbcsdk/driver
odbcsdk/driver/odbc_src.taz
```

The next step is to ensure that your host operating system's shared library PATH variable is set correctly so that the ODBC Driver Manager and OpenLink ODBC Driver shared libraries are locatable by your UNIX based ODBC Application.

Under many UNIX flavors the environment variable LD_LIBRARY_PATH is used to set the shared library PATH. Please note that older versions of AIX use the environment variable LIBPATH, while HP-UX requires the environment variable SHLIB_PATH.

If your system has a 'C' compiler, you can verify the development environment by re-compiling the "odbctest.c" program. You can also simply run "odbctest" to test the state of your ODBC under UNIX runtime environment.

Configuration of Run-time Distribution

The iODBC driver manager links iODBC clients to iODBC Drivers via the iODBC configuration file. This file is typically the file "~/.odbc.ini" (where the tilde stands for the user's home directory. Your OpenLink iODBC SDK installation process will place this file in the $HOME directory of the account user installing the SDK. It will also set the correct references to the Driver Manager and OpenLink ODBC Driver shared libraries.

Your OpenLink UNIX based ODBC Drivers make use of iODBC Data Source Names (DSNs). These DSNs are created, configured, and deleted through the OpenLink Admin Assistant. If you so choose you can also do this manually by editing the ~/.odbc.ini file.

The example below depicts the structure of an iODBC DSN named "dsn_oracle7", used to facilitate connections with an Oracle Database residing on a machine named "mercury".

```
[dsn_oracle7]

Database = ORCL
UserName = scott
Password = tiger
Host = mercury
```

```
ServerType = Oracle 7
FetchBufferSize = 60
Description = Oracle Database Connections
```

Your general OpenLink ODBC Client for UNIX session settings are held within the "[UDBC]" section of the file "udbc.ini", this file is situated in the "bin" sub-directory of your OpenLink Server components installation. An extract from this file that deals with these settings is depicted below:

```
[UDBC]

;DebugFile = /tmp/udbc.out ; Write client debugging output

[Communications]

ReceiveTimeout = 120 ; max. time to complete a request
BrokerTimeout = 30 ; max. time to wait for oplrqb
RetryTimeout = 5 ; Retry time for oplrqb locating
SendSize = 4096 ; RPC send buffer size
ReceiveSize = 16000 ; RPC receive buffer size
ShowErrors = Y ; Pass RPC errors to applications
DataEncryption = N ; Encrypt outgoing data
```

Note: It is highly recommended that you use the iODBC DSN templates provided by your OpenLink ODBC SDK installation for your initial connection, the iODBC SDK installer program sets up the appropriate iODBC environment for you at installation time.

The table below describes each OpenLink iODBC DSN attributes required for making a connection from your ODBC based application to your backend database engine.

Section and Keys	***Default Value***	***Corresponding Connect String option***	***Description***
[UDBC Global]			Global UDBC client communications settings.
BrokerTimeout	30		The time (in secs) that the client application will wait for the request broker (oplrqb) to accept or reject a database connection.
ReceiveTimeout	60		The time (in secs) that the client application will wait for a database agent to finish a request.
PortMapTimeout	30		When using the ONC portmap or rpcbind name services, this is the time (in secs) that the client application will wait for a reply from the portmapper.
SendSize	4096		RPC send buffer size. A value of 0 will cause the application to use the default value.
ReceiveSize	16000		RPC receive buffer size. A value of 0 will cause the application to use the default value.
DebugFile	no debugging		When this variable contains a valid filename, the client applications will write debugging information to that file. Do not use this in the systemwide ini file! (/etc/udbc.ini) This file will contain detailed UDBC call tracing.
[DSN_name]		DSN=DSN_name	Name of the Data Source
Host		HOST=name	Name of host to contact (hostname or network address)
ServerType		SVT=type	Specify agent domain This is used by the broker to determine which agent section to connect the client request to using mapping rules.
ServerOptions			Server startup options specific to agent/database.
Database		DATABASE=db	Database to use. Agent/database specific.
Options		OPTIONS=opts	Database connect options. Agent/database specific.
UserName		UID=username	Username to connect as.

		PWD=password	Password for user.
ReadOnly		READONLY=<y|n>	Specify Y(es) for read-only (ro) or N(o) for read-write (rw) connections. Read only connections are sometimes faster, but can never modify any database.
FetchBufferSize	5	FBS=<value>	Number of rows (records) to be fetched per call from database agent. Values range from 1 to 99

Example ODBC Connect Strings:

"" (Empty Connect String)

This implies the use of values from the [Default] section of the iODBC configuration file. There must be at least a "Host=" definition in this section.

"DSN=dsn_oracle7"

Use the values from the [dsn_oracle7] section.

"DSN=MyOracle7dsn;UID=scott;PWD=tiger"

Use the values from the [dsn_oracle7] section, and overrides the value for UserName. Also use the password "tiger" to gain access to the database.

"HOST=mercury;SVT=Oracle 7;DATABASE=ORCL;UID=scott/tiger"

Connect to an Oracle 7 database on the machine "mercury" .

"READONLY=Y"

Make a read-only connection to the [Default] database.

The Sample ODBC Application "ODBCTEST"

Your OpenLink UDBC SDK installation contains a sample interactive dynamic SQL application called "odbctest", this application is situated within your OpenLink installation's "odbcsdk/examples" sub-directory.

Then run the application by typing the following command from the command line prompt:

```
odbctest
```

The application will prompt you for a connect string (assuming we are using the UDSN example depicted above):

```
Enter ODBC connect string: DSN=dsn_oracle7
```

If you connect to the database successfully you will get a SQL prompt:

```
SQL>
```

Type a valid SQL statement and then press <return> to send your request to your UDSN

```
SQL> select * from <table name>
```

or Use an OpenLink quick table list command:

```
SQL> tables
```

If the query returns data and was successful, the data will be scrolled down the screen.

Otherwise you will receive an error message.

Type in "exit" to quit the application

Further Reading:

"Data Management: SQL Call Level Interface (CLI)"

from X/Open in conjunction with SQL Access Group

ISBN: 1-872630-63-4

X/Open Document Number: S203

Microsoft ODBC API documentation:
http://www.microsoft.com/data/reference/odbc2.htm

Chapter3: Server Components Installation

Windows 95/98/NT/2000 Server Components Installation Guide

Introduction

The components that make up your OpenLink Server Components are:

- **OpenLink Request Broker** - the component responsible for brokering the services of OpenLink Data Access and Service Providing Agents. Its is also the component responsible for coordinating and controlling your entire OpenLink Data Access session irrespective of Data Access mechanism being used. This is the heart and soul of the OpenLink Database Independent Communications Layer, the technology that enables the OpenLink Data Access drivers to communicate with your backend database engines alleviating the need to acquire additional database specific networking software from your backend database vendor(s).

- **One or more Database Agents** (one for each supported database engine) - these are database specific components that provide the actual backend database connectivity and data access services to your OpenLink Data Access Clients (ODBC, JDBC, UDBC, OLE-DB etc.). These components are actually clients from the perspective of each supported backend database, this is because they are built using the Call Level Interfaces (CLIs) or Embedded SQL interfaces of these backend databases.

- **One or more Service Provider Agents** (ODBC, JDBC, PROXY.) - these are generic agents that provide Distributed Data Access Protocol handling services to OpenLink Data Access Drivers. The data access protocols supported are as follows:

  - **ODBC Agent** - enabling OpenLink Multi-Tier ODBC Drivers to connect to local or remote non OpenLink ODBC Drivers

  - **JDBC Agent** - enabling OpenLink JDBC Drivers to connect local or remote OpenLink or non OpenLink ODBC or UDBC Drivers

  - **PROXY Agent** - enabling all OpenLink Data Access Clients to connect to OpenLink Database Agents that do not reside on the same server machine as the prime OpenLink Request Broker. The prime Broker being the Request Broker that all your OpenLink Client are configured to request data access services from. This enables you configure your OpenLink Database Agents for use in N-Tier distributed computing environments.

An architectural overview the OpenLink Multi-tier Data Access infrastructure is available at: http://www.openlinksw.com/info/mtproduct.htm . Note that the OpenLink Request Broker is the core component that makes up the OpenLink Database Independent Communications Layer in this illustration.

The components listed above are presented to you for download at the end of your interaction with the OpenLink Software Download Wizard.

Once you have successfully downloaded the appropriate OpenLink Server components for your server operating system, you will be in a position to follow the steps outlined in the operating system specific installation sections that follow.

Installing OpenLink Server Components on Windows 95/98/NT/2000 Servers

1. Ensure you have downloaded all of the relevant OpenLink Server Components from the page created by the OpenLink Software Download wizard.

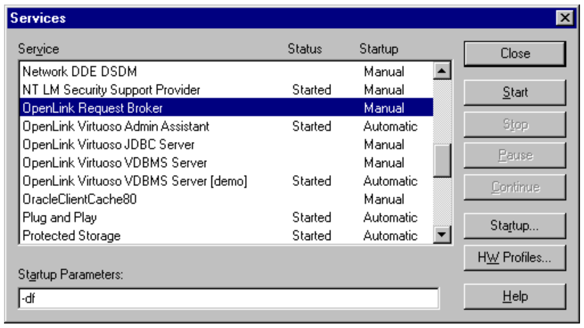
2. Run setup.exe, please note that the client and server components for Windows are packaged within the same installation archive. You distinguish the actual components that you want to install as part of your interaction with the installation program.

   Caution: please do not choose SPX/IPX protocol support during the installation process unless you have verified that this protocol is actually installed and configured on your Windows Server machine.

3. Start your OpenLink Request Broker in debug mode, from within your "services" control panel, or by opening up a DOS command window and then executing the OpenLink Request Broker startup command from within the "bin" sub-directory of the OpenLink server components installation directory:

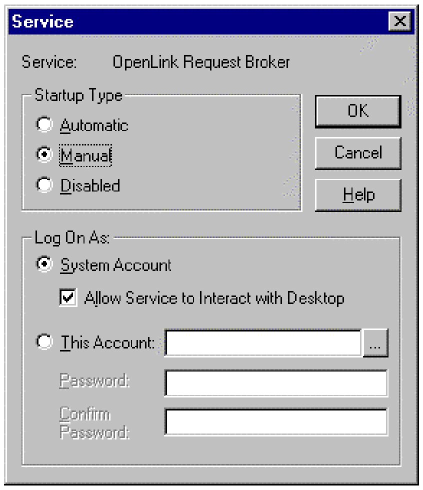
   oplrqb –dv

   Note: When doing this from the "services" control panel (Windows NT) you must stop the Broker if it is already running and then change its startup mode from "Automatic" to "Manual", then enter the required startup commands as values in the Startup Parameters field, the screen shot below demonstrates this:



Service	Status	Startup
Network DDE DSDM		Manual
NT LM Security Support Provider	Started	Manual
OpenLink Request Broker		Manual
OpenLink Virtuoso Admin Assistant	Started	Automatic
OpenLink Virtuoso JDBC Server		Manual
OpenLink Virtuoso VDBMS Server		Manual
OpenLink Virtuoso VDBMS Server [demo]	Started	Automatic
OracleClientCache80		Manual
Plug and Play	Started	Automatic
Protected Storage	Started	Automatic

Startup Parameters: -df



Service: OpenLink Request Broker

Startup Type: Automatic [✓ Manual] Disabled

Log On As: [✓ System Account]

[✓ Allow Service to Interact with Desktop]

This Account:

Password:

Confirm Password:

4. Start your Web Browser and then enter the following URL:

   http://<network alias of machine on to which you have just installed server components>:<port number you provided when prompted by OpenLink installer> . For example, if your machine has a network alias of "mainserver" and you accept the default port number at installation time, then the required URL would be constructed as follows:

   http://mainserver:8000 you can also enter the value
   http://localhost:8000 if the server in use is local rather than remote.

5. Follow the instructions provided in the "Making Your First OpenLink Connection" guide in order to verify your server components installation.

6. If step 5 is successful, shutdown the Request Broker using the command:

   CTRL-C in the debug session window

   or

   by selecting the OpenLink Request Broker entry within the "services" control panel and then clicking on the "stop" button.

You can also shutdown the Broker by executing the following command from a separate DOS command Window:

```
oplshut -f
```

7. Revert the OpenLink Request Broker startup mode back to "Automatic" from within the "Services" control panel and then restart by clicking on the "Startup" button. Note the Broker does not have to be in "Automatic" mode for regular use, its your choice as to the startup mode that best suits your operational needs.

   You can also start the Broker from a DOS Window by executing the command:

   ```
   ./oplrqb -v
   ```

UNIX & Linux Server Components Installation Guide

Introduction

The components that make up your OpenLink Server Components are:

- **OpenLink Request Broker** - the component responsible for brokering the services of OpenLink Data Access and Service Providing Agents. Its is also the component responsible for co-ordinating and controlling your entire OpenLink Data Access session irrespective of Data Access mechanism being used. This is the heart and soul of the OpenLink Database Independent Communications Layer, the technology that enables the OpenLink Data Access drivers communicate with your backend database engines, alleviating the need to acquire additional database specific networking software from your backend database vendor(s).

- **One or more Database Agents** (one for each supported database engine) - these are database specific components that provide the actual backend database connectivity and data access services to your OpenLink Data Access Clients (ODBC, JDBC, UDBC, OLE-DB etc.). These components are actually clients from the perspective of each supported backend database, this is because they are built using the Call Level Interfaces (CLIs) or Embedded SQL interfaces of these backend databases.

- **One or more Service Provider Agents** (ODBC, JDBC, PROXY.) - these are generic agents that provide Distributed Data Access Protocol handling services to OpenLink Data Access Drivers. The data access protocols supported are as follows:

  - **ODBC Agent** - enabling OpenLink Multi-Tier ODBC Drivers to connect to local or remote non OpenLink ODBC Drivers

  - **JDBC Agent** - enabling OpenLink JDBC Drivers to connect local or remote OpenLink or non OpenLink ODBC or UDBC Drivers

  - **PROXY Agent** - enabling all OpenLink Data Access Clients to connect to OpenLink Database Agents that do not reside on the same server machine as the prime OpenLink Request Broker. The prime Broker being the Request Broker that all your OpenLink Client are configured to request data access services from. This enables you configure your OpenLink Database Agents for use in N-Tier distributed computing environments.

An architectural overview the OpenLink Multi-tier Data Access infrastructure is available at:
http://www.openlinksw.com/info/mtproduct.htm .
Note that the OpenLink Request Broker is the core component that makes up the OpenLink Database Independent Communications Layer in this illustration.

The components listed above are presented to you for download at the end of your interaction with the OpenLink Software download Wizard at:
http://www.openlinksw.com/main/softdld2.htm .

Installing OpenLink Server Components on UNIX & Linux Servers

1. Ensure you have downloaded all of the relevant OpenLink Server Components from the page created by the OpenLink Software Download wizard, this must include the OpenLink Server Components Installation shell script file "install.sh" in addition to the compressed "TAR" archives for the OpenLink Request Broker and relevant Database Agent Components.

2. Log on to your Database or Application server machine using the userid and password combination that you normally use when connecting to the database in non ODBC/JDBC/UDBC mode.

3. Determine your operating system group membership by typing in the command:

   ```
   set
   ```

   the "gid" value indicates your current operating system group membership (remember this for use later on during the installation process).

4. Create an OpenLink installation directory on the server (e.g. /usr/openlink) and transfer all server related files to it. Note: If using ftp remember to transfer the ".taz" files in BINARY mode and the installation shell script "install.sh" in TEXT mode.

   You set your FTP program in TEXT or ASCII transfer mode by typing the following command at your FTP command prompt:

   ```
   ftp> ascii
   ```

   You set your FTP program in BINARY transfer mode by typing the following command at your FTP command prompt:

   ```
   ftp> bin
   ```

5. Run the install script on the server by typing the command:

   ```
   sh install.sh
   ```

   or

   ```
   ./install.sh
   ```

6. The install script will extract the files from the ".taz" files and then create all of the relevant OpenLink server component sub-directories. The install script will also place the OpenLink server components into their appropriate sub-directories

7. Once the files have been extracted and placed in the relevant sub-directories, the install script will proceed to install and configure the OpenLink Admin Assistant. This program
   enables remote configuration for all OpenLink Server Components (Rule Book,
   Service and Database Agents) from any Web Browser.

   The text below illustrates the interactions encountered during the server components installation process:

   What is the OpenLink Broker directory? [/usr/openlink/bin]

   What is the OpenLink Admin Assistant directory? [/usr/openlink/bin/w3config]

   TCP/IP Port to use? [8000]

   Log File? [www_sv.log]

   Log all requests (y/n)? [n]

8. Configure the OpenLink Request Broker and one or more OpenLink Database Agents using the OpenLink Server Components Configuration utility (the program "oplcfg"). This utility is only required for the purposes of providing interactive configuration of OpenLink Server Components at installation time, the OpenLink Admin Assistant should be used for subsequent configuration related activity after the installation process is completed.

The OpenLink Configuration Utility presents a character based user interface as depicted below:

```
OpenLink Server Components Configuration Utility

1. Request Broker               11. PostgreSQL
2.  Informix 5                  12.  Progress 6
3.  Informix 6                  13.  Progress 7
4.  Informix 7                  14.  Progress 8
5.  Ingres 6                    15.  Solid
6.  Virtuoso                    16.  Sybase 4
7.  OpenIngres                  17.  Sybase 10
8.  Oracle 6                    18.  Sybase 11
9.  Oracle 7                    19.  Unify 2000
10. Oracle 8                    20.  Velocis


U.  Undo last change            V.  View the current Rules Book
C.  Clear log file              L.  View log file
B.  Backup Rules Book           R.  Restore Rules Book
I.  Verify Rules Book           N.  Reinitialize running Broker
S.  Startup  Request Broker     D.  Shutdown Request Broker

Choose an item or type q to quit :
```

Note: You MUST select option 1 if you did not use the recommended installation directory (/usr/openlink) for your OpenLink Server Components.

Options 2 - 20 provide you with the database type you wish to configure your OpenLink "Session Rules" Book for, simply choose the appropriate option(s) and let the Server Components Configuration Utility do the rest.

For more information about the Database specific settings required please see Agent Specific Settings.

9. Start your OpenLink Request Broker in debug mode, within the OpenLink Server Components Configuration utility, or by exiting and executing the OpenLink Request Broker startup command from within the "bin" sub-directory of the OpenLink server components base installation directory:

   oplrqb –dv

   Should you receive an "Unable To Install RPC Services" error this indicates that the Broker was automatically started when you ran "oplcfg", it can be shutdown by running the command:

   oplshut –f

   or

   rqbshut

10. Start your Web Browser and then enter the following URL:

    http://<network alias of machine on to which you have just installed server components>:<port number you provided when prompted by OpenLink installer> . For example, if your machine has a network alias of "opllinux" and you accept the default port number at installation time, then the required URL would be constructed as follows:

    http://opllinux:8000

11. Follow the instructions provided in the "Making Your First OpenLink Connection" guide in order to verify your server.

12. If step 11 is successful, shutdown the Request Broker using the command:

    oplshut –f

    or

    rqbshut

    and then restart in normal mode using the command:

    oplrqb -v

    For more information about other options that can be used when starting and stopping the Request Broker see the Command Line Options section.

The program "oplcfg" (the OpenLink server components configuration utility), can also be used for starting and stopping the OpenLink Request Broker via a character mode menu based interface

Chapter4: Server Component Administration

OpenLink Request Broker Administration

Introduction

The OpenLink Request Broker is the server component responsible for brokering the services of OpenLink Data Access and Service Providing Agents. Its is also the component responsible for coordinating and controlling your entire OpenLink Data Access session, irrespective of Data Access mechanism being used. The Request Broker is the heart and soul of the OpenLink Database Independent Communications Layer, the technology that enables the OpenLink Data Access drivers communicate with your backend database engines without the need to acquire additional database specific networking software from your backend database vendor(s).

Getting To Know Your OpenLink Request Broker Components

The core components required by the Request Broker are situated within the "bin" and "bin/w3config" sub-directories under your OpenLink server components base installation directory. Each of these components is described below and grouped by directory location.

The "bin" sub-directory:

Below is a list of the important files in the *bin* directory:

oplrqb	The OpenLink Request Broker.
oplrqb.ini	The OpenLink "Session Rules" Book.
oplrqb.log	File that holds critical Broker & Startup and Shutdown audit information. All critical events that affect the Broker are written to this file irrespective of Broker startup options.
oplshut	Utility for showing Agent status and shutting down the Broker.
***xxx*_sv**	OpenLink Agent. *xxx* will be an abbreviation for the data access or protocol handling service provided by the OpenLink Agent.
release.txt	Text file with the latest information regarding the Request Broker.
register	Utility for registering the Request Broker - also requires a valid key file, register.ini.
register.ini	File containing software activation and license key for the Broker.

The "bin/w3config" sub-directory:

www_sv	The OpenLink Web Service Agent, this is basically a HTTP/Web Server implemented as an OpenLink Service Providing Agent. It is this component that forms the core engine around which the OpenLink Admin Assistant has been built.
www_sv.log	File that holds critical Web Service Agent & Startup and Shutdown audit information. All critical events that affect the Web Service Agent are written to this file irrespective of Web Service Agent startup options.
www_sv.ini	Web Service Agent configuration file.
setup	Web Service Agent installation and configuration program (you only need to run this directly if you only want to re-install the Web Service Agent component as opposed to the entire pool of OpenLink Server Components.

Registering Your OpenLink Request Broker

The OpenLink Request Broker must be registered in order to use it beyond the default license provided by OpenLink Software (2 concurrent client machines and 10 database connections).

- Place the register.ini (or any other file issued to you by OpenLink software) file in the "bin" sub-directory of your OpenLink Server Components installation directory.

- If you have to use the "ftp"utility to transfer these files please remember to perform this transfer in ASCII mode.

- Move into the "bin" sub-directory using the command: cd bin
  and then type one of the following commands:

`./register` (if the file name is "register.ini")

or

`./register <filename>`
(if you have a file name something other than "register.ini")

- A message will be displayed indicating success or failure.

The Register program must be re run every time to you upgrade concurrent usage support or actual OpenLink Versions.

Obtaining Version Numbers & Additional Descriptive Information From Your Server Components

Using The Command Line Approach

This approach provides you with information about the actual executable file, it includes:

Version Number - this is a component identifier that indicates the version number specific of a specific OpenLink Component

Release Number - this is an identifier for a collection of OpenLink Components, numerous components with different version numbers make up an OpenLink Data Access Drivers commercial release.

Compilation Date - indicates the date component was compiled.

CVSID - this is a source code archive identifier that relates to the actual source code archive from which a particular component has been assembled.

Binary Platform - indicates what platform the component has been built to run on.

To obtain the information referred to above for any OpenLink Component simply type in the program name at your command prompt with the "-?" switch.

Examples of the output produced by the OpenLink Request Broker and the OpenLink Web Service Agent are depicted below:

OpenLink Request Broker:

```
[openlink@opllinux bin]$ oplrqb -?

OpenLink Request Broker
Version 2.7A (Release 3.2) as of Thu Jan 21 1999 (cvsid 00048).
Compiled for Linux 2.0.36 (i586-pc-linux-gnulibc1)
Copyright (C) OpenLink Software.

Usage:   oplrqb [-flLdcv] [+foreground] [+loglevel num] [+logfile arg]
[+debug] [+configfile arg] [+version]
```

+foreground	run in the foreground
+loglevel	log level
+logfile	log file
+debug	debug mode
+configfile	use alternate configuration file

+version	show version number

Web Service Agent:

```
[openlink@opllinux w3config]$ www_sv -?
OpenLink Web Service Agent
Version 1.1A (Release 3.2) as of Thu Jan 21 1999 (cvsid 00048).
Compiled for Linux 2.0.36 (i586-pc-linux-gnulibc1)
Copyright (C) OpenLink Software.

Usage:  www_sv [-lLd] [+loglevel num] [+logfile arg] [+debug]
```

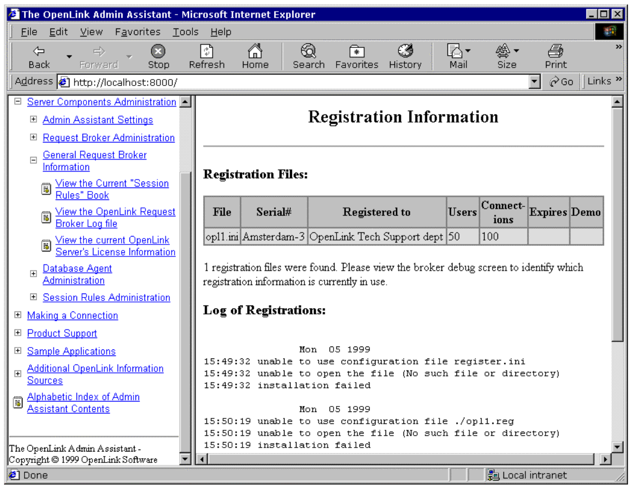
+loglevel	log level
+logfile	log file
+debug	debug mode

Obtaining Additional Information Using The Admin Assistant

You obtain additional information using the Admin Assistant, this includes:

Register File Information - enables you determine the license(s) that you have in place.
Session Rules Book - the current set of rules for your OpenLink Data Access sessions.
Request Broker Log File Information - displays the contents of your Request Broker log file.

The screen shot below shows you an example of how this is achieved from the Admin Assistant:


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print
Address http://localhost:8000/
Go Links
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
View the Current "Session Rules" Book
View the OpenLink Request Broker Log file
View the current OpenLink Server's License Information
Database Agent Administration
Session Rules Administration
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software
Done
Local intranet

Registration Information

Registration Files:

File	Serial#	Registered to	Users	Connect-ions	Expires	Demo
opl1.ini	Amsterdam-3	OpenLink Tech Support dept	50	100		

1 registration files were found. Please view the broker debug screen to identify which registration information is currently in use.

Log of Registrations:

```
                Mon  05 1999
15:49:32 unable to use configuration file register.ini
15:49:32 unable to open the file (No such file or directory)
15:49:32 installation failed

                Mon  05 1999
15:50:19 unable to use configuration file ./opl1.reg
15:50:19 unable to open the file (No such file or directory)
15:50:19 installation failed
```

OpenLink Request Broker Startup & Shutdown Options

Startup Options

The Request Broker can be started in a number of ways using various command line options.

```
Usage: oplrqb [-flLdcv] [+foreground] [+loglevel num] [+logfile arg][+debug][+configfile arg] [+version]
```

+foreground run Broker in the foreground mode

+loglevel log level where Broker session log details decrease in verbosity from 7 down to 1

+logfile full path and file name into which Broker session log output is to be written

+debug starts Broker in debug mode

+configfile use alternate configuration file

+version show version number

CommandLine Examples:

[openlink@opllinux openlink]$ oplrqb
Starts the Broker in background mode.

[openlink@opllinux openlink]$ oplrqb -v
Starts the Broker in background mode and displays version information as part of startup process.

[openlink@opllinux openlink]$ oplrqb -dv
Starts the Broker in foreground debug mode using default log level 7 (most amount of session log information). This blocks your screen and echoes the output of all OpenLink Data Access Clients to the blocked screen.

[openlink@opllinux openlink]$ oplrqb -dvl 1
Starts the Broker in foreground debug mode using default log level 1 (lowest amount of session log information). This blocks your screen and echoes the output of all OpenLink Data Access Clients to the blocked screen.

[openlink@opllinux openlink]$ oplrqb -dvL /tmp/openlink.log
Starts the Broker in foreground debug mode using default log level 7 (most amount of session log information). This blocks your screen and echoes the output of all OpenLink Data Access Clients to the file "/tmp/openlink.log" .

Shutdown Options

The OpenLink Request Broker can be shutdown in a number of ways using various command line options.

```
Usage: oplshut [-hcskryfp] [+hostname arg] [+configfile arg] [+show] [+kill] [+reinit] [+yes] [+force]
[+protocol arg] [pid ..]
```

+hostname contact oplrqb on given host

+configfile use alternate configuration file

+show show database agent connections

+kill kill a database agent

+reinit restart broker without shutting down clients (partial restart)

+yes don't ask for confirmation

+force force immediate shutdown (kill all)

+protocol specify default protocol to use when listening for client connections

CommandLine Examples:

[openlink@opllinux openlink]$ oplshut
Shuts down broker on local machine

[openlink@opllinux openlink]$ oplshut -f
Shuts down broker on local machine without waiting for clients to terminate

[openlink@opllinux openlink]$ oplshut -fy
Same functionality as prior command, but without asking for confirmation

[openlink@opllinux openlink]$ oplshut -s
Shows the current active OpenLink Data Access Client connections

oplshut -k 81 93
Try to terminate the database agent sessions with process IDs 81 and 93

System Security Enhancement Program

Due to the Unix security model certain kernel calls can only be used when the process is run by root or under the permission of root. One of these function calls is needed for the PingWatchDog protocol, thereby forcing you to run the request broker as root. This can be accomplished in two ways:

1. The user logs in as root and then fires up the request broker.

2. The request broker is flagged to run under root privileges and the user can start it up from other accounts.

The consequences of 1 are that the user(s) who need to startup the broker need to have knowledge of the root password, which is not advisable for system managers to give to ordinary users.

Therefore we have built a solution whereby the broker can be flagged as running under root privileges without the user(s) needing to know the root password. The system manager executing the following commands accomplishes this:

```
cd openlink/bin
  ./security -s
```

This will place the broker into a special security state that has the following impact:

1. The permissions of the oplrqb program are changed so it takes on the identity of "root" when started up.

2. The broker now knows the position of the oplrqb.ini file and will not accept certain startup flags so it cannot be tricked to read in another oplrqb.ini file from the command line.

3. The oplrqb.ini file can now only be written to by the "root" account, other accounts will only be able to read this file.

If a user now starts the broker the **StartupBy** clause within the **[Security]** section of the broker is checked to see whether the user is allowed to startup the broker (same for shutdown).

This accomplishes the wish of many system administrators that the broker process can be started up as root, without widespread knowledge of the root password.

As stated in item 3 above the "oplrqb.ini" file can now only be written to by the account "root". This means that ordinary users cannot modify this file, which is often an unpleasant side effect. To overcome this some system administrators change the mode of the oplrqb.ini file, thereby opening up to security risks.

OpenLink also has a proper answer for this scenario with the use of the '**includerulebook**' option inside the security section of the rule book. The system administrator can split up the rulebook into different files which than can be included from the master oplrqb.ini file. This makes specific sections available to normal users, but allowing the system administrator to decide which sections can be overruled by normal users. The included rulebooks cannot overrule any section within the oplrqb.ini file.

This is a very powerful way of dealing with high security installations within large companies.

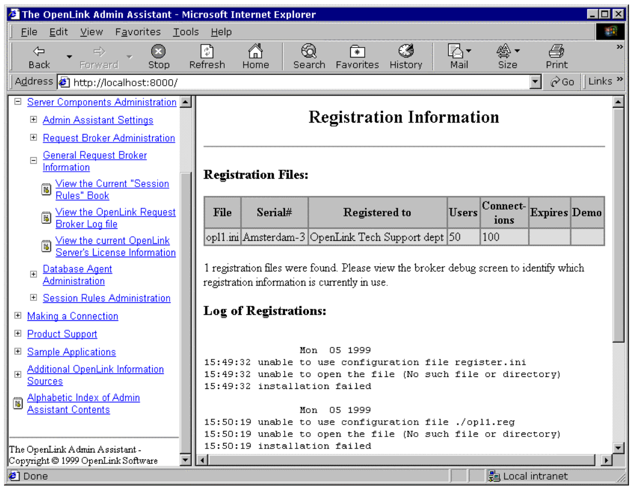
Configuring Request Broker Session Parameters

The Request Broker is responsible for setting up the profile of one of more OpenLink Data Access sessions, this includes a Transport Protocol selection, Keep Alive Packets notifications, Network Message Buffer Sizes. Initialization retry intervals, Agent initialization timeouts, and much more. Rather than have you manually set this options via the OpenLink Rule Book, it is now possible for you to configure this options via your Web Browser of choice using the Admin Assitant

The Admin assistant allows you to configure Request Broker session parameters in two ways, via a series of Wizard Interactions or via an HTML form, the sections that follow illustrate both approaches.

Using Admin Assistant Wizards

1. Expand the the "Server Components Administration " hyperlink then General Request Broker Information-->View the Current OpenLink Srever's License Information


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print
Address http://localhost:8000/
Go Links
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
View the Current "Session Rules" Book
View the OpenLink Request Broker Log file
View the current OpenLink Server's License Information
Database Agent Administration
Session Rules Administration
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software
Done
Local intranet

Registration Information

Registration Files:

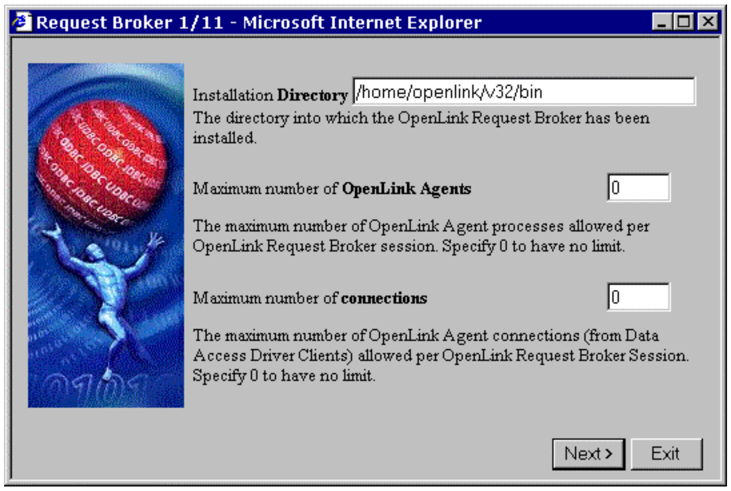
File	Serial#	Registered to	Users	Connect-ions	Expires	Demo
opl1.ini	Amsterdam-3	OpenLink Tech Support dept	50	100		

1 registration files were found. Please view the broker debug screen to identify which registration information is currently in use.

Log of Registrations:

```
                 Mon  05 1999
15:49:32 unable to use configuration file register.ini
15:49:32 unable to open the file (No such file or directory)
15:49:32 installation failed

                 Mon  05 1999
15:50:19 unable to use configuration file ./opl1.reg
15:50:19 unable to open the file (No such file or directory)
15:50:19 installation failed
```

2. Specify the location of your Request Broker and its dependent components and any Agent related restrictions you may want to apply to your OpenLink session, then click the "Next" button.


Request Broker 1/11 - Microsoft Internet Explorer

Installation **Directory** /home/openlink/v32/bin

The directory into which the OpenLink Request Broker has been installed.

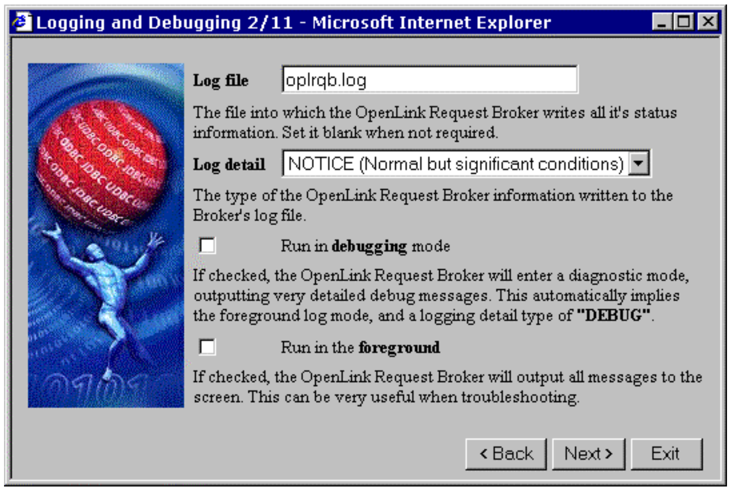
Maximum number of **OpenLink Agents** 0

The maximum number of OpenLink Agent processes allowed per OpenLink Request Broker session. Specify 0 to have no limit.

Maximum number of **connections** 0

The maximum number of OpenLink Agent connections (from Data Access Driver Clients) allowed per OpenLink Request Broker Session. Specify 0 to have no limit.

3. Enter session log values if you are running a troubleshooting as opposed to normal OpenLink session, then click on the "Next" button.


Logging and Debugging 2/11 - Microsoft Internet Explorer

Log file oplrqb.log

The file into which the OpenLink Request Broker writes all it's status information. Set it blank when not required.

Log detail NOTICE (Normal but significant conditions)

The type of the OpenLink Request Broker information written to the Broker's log file.

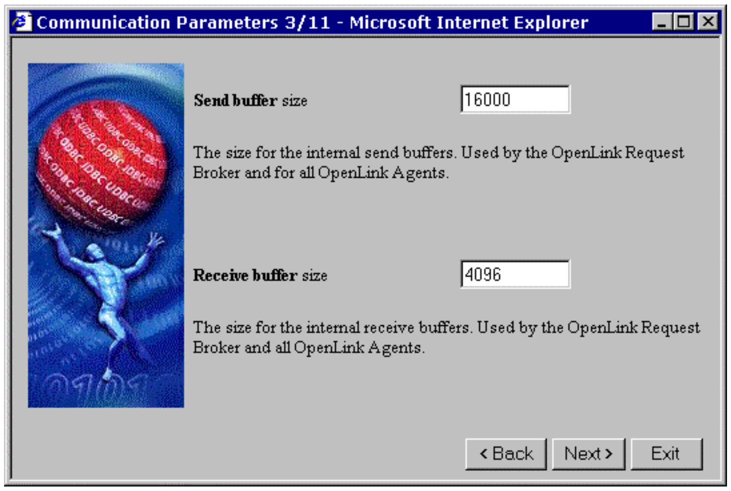
☐ Run in **debugging** mode

If checked, the OpenLink Request Broker will enter a diagnostic mode, outputting very detailed debug messages. This automatically implies the foreground log mode, and a logging detail type of **"DEBUG"**.

☐ Run in the **foreground**

If checked, the OpenLink Request Broker will output all messages to the screen. This can be very useful when troubleshooting.

4. Enter buffer size values for the inbound and outbound network messages for each OpenLink Agent (it is advisable to take the default values presented). Then click the "Next" button.



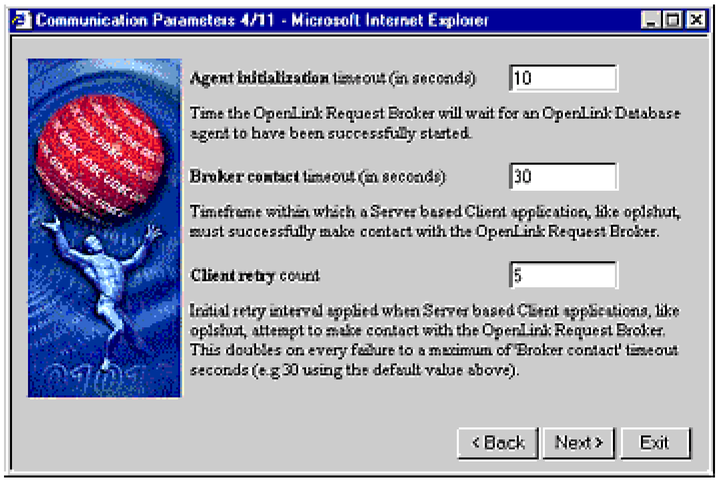
Send buffer size 16000

The size for the internal send buffers. Used by the OpenLink Request Broker and for all OpenLink Agents.

Receive buffer size 4096

The size for the internal receive buffers. Used by the OpenLink Request Broker and all OpenLink Agents.

5. Enter values for "Database Agent" session initialization, "Broker Contact", and initialization failure retry counts by entering values into the respective fields as depicted below (it is advisable take the defaults). Then click on the next button.



Agent initialization timeout (in seconds) 10

Time the OpenLink Request Broker will wait for an OpenLink Database agent to have been successfully started.

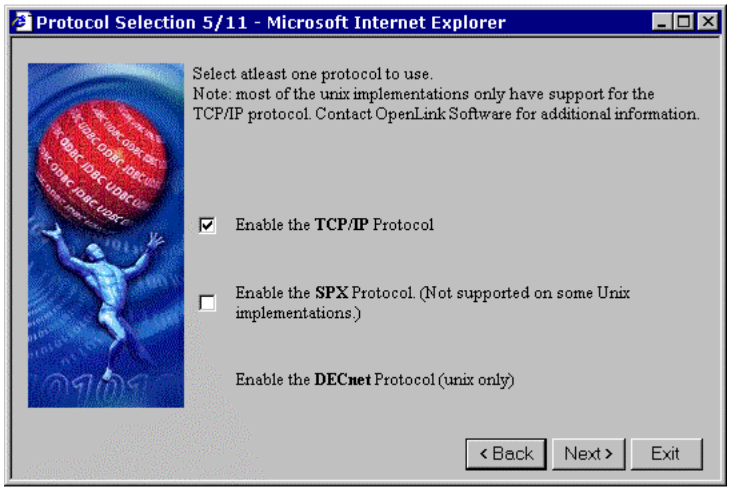
Broker contact timeout (in seconds) 30

Timeframe within which a Server based Client application, like oplshut, must successfully make contact with the OpenLink Request Broker.

Client retry count 5

Initial retry interval applied when Server based Client applications, like oplshut, attempt to make contact with the OpenLink Request Broker. This doubles on every failure to a maximum of 'Broker contact' timeout seconds (e.g 30 using the default value above).

6. Check one of more of the check-boxes presented in order select the Network Transport Protocol to be used by your Request Broker and Database Agents when communicating with your OpenLink Data Access Clients. Then click on the next button



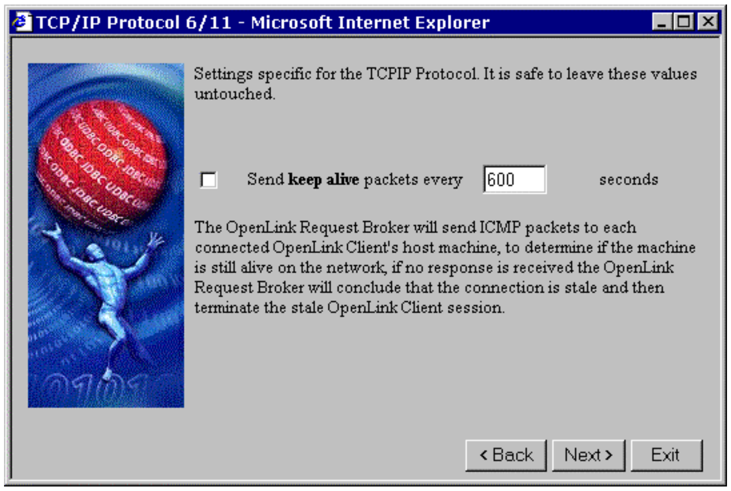
Select atleast one protocol to use.
Note: most of the unix implementations only have support for the TCP/IP protocol. Contact OpenLink Software for additional information.

[✓] Enable the **TCP/IP** Protocol

[ ] Enable the **SPX** Protocol. (Not supported on some Unix implementations.)

Enable the **DECnet** Protocol (unix only)

7. You can choose to enable OpenLink TCP session connection monitoring by checking the "Send Keep Alive Packets" check-box. this facility sends probes to all OpenLink Clients verifying if they are still "Alive". You set the intervals for these probes by entering a value in the "Every" field. The default values are adequate for most scenarios. Once completed click the "Next" button to continue.



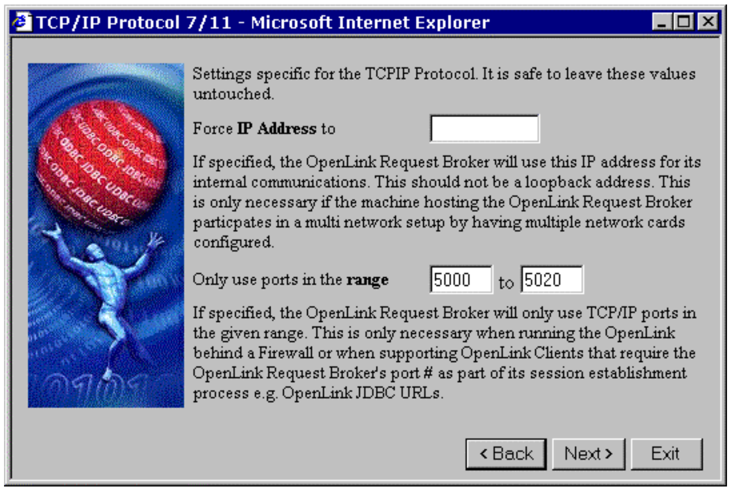
Settings specific for the TCPIP Protocol. It is safe to leave these values untouched.

[ ] Send **keep alive** packets every 600 seconds

The OpenLink Request Broker will send ICMP packets to each connected OpenLink Client's host machine, to determine if the machine is still alive on the network, if no response is received the OpenLink Request Broker will conclude that the connection is stale and then terminate the stale OpenLink Client session.

8. If you have more than one Network Interface Card (NIC) installed on your OpenLink Server machine, then you can enter a value in the "Force IP Address" field that represents the IP address that you want the Request Broker to listen at.

   You can also designate port number ranges to be used by the Broker to advertise other OpenLink services (e.g JDBC Agent used by OpenLink Drivers for JDBC). Once completed click the "Next" button to continue.


TCP/IP Protocol 7/11 - Microsoft Internet Explorer

Settings specific for the TCPIP Protocol. It is safe to leave these values untouched.

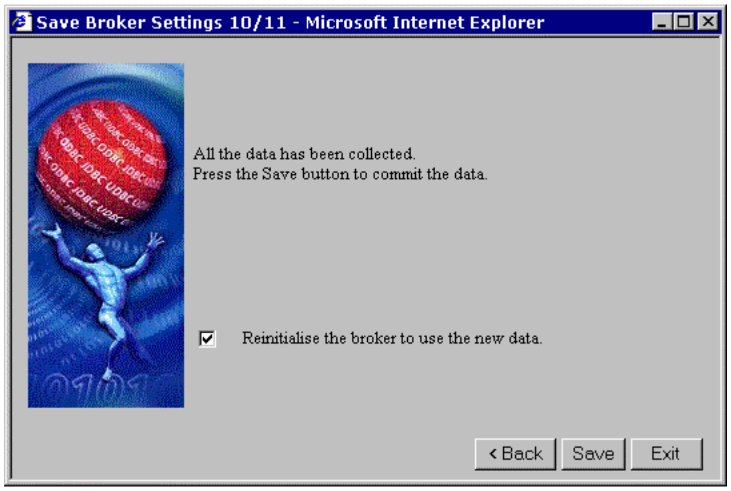
Force **IP Address** to

If specified, the OpenLink Request Broker will use this IP address for its internal communications. This should not be a loopback address. This is only necessary if the machine hosting the OpenLink Request Broker particpates in a multi network setup by having multiple network cards configured.

Only use ports in the **range** 5000 to 5020

If specified, the OpenLink Request Broker will only use TCP/IP ports in the given range. This is only necessary when running the OpenLink behind a Firewall or when supporting OpenLink Clients that require the OpenLink Request Broker's port # as part of its session establishment process e.g. OpenLink JDBC URLs.

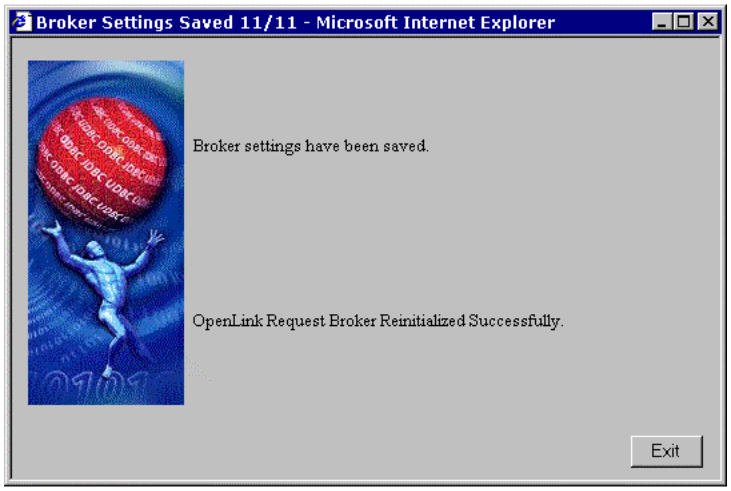
9. Click on the "Save" button to save your changes permanently.


Save Broker Settings 10/11 - Microsoft Internet Explorer

All the data has been collected.
Press the Save button to commit the data.

☑ Reinitialise the broker to use the new data.

10. Click the "Exit" button to leave the Broker Settings Wizard.


Broker Settings Saved 11/11 - Microsoft Internet Explorer

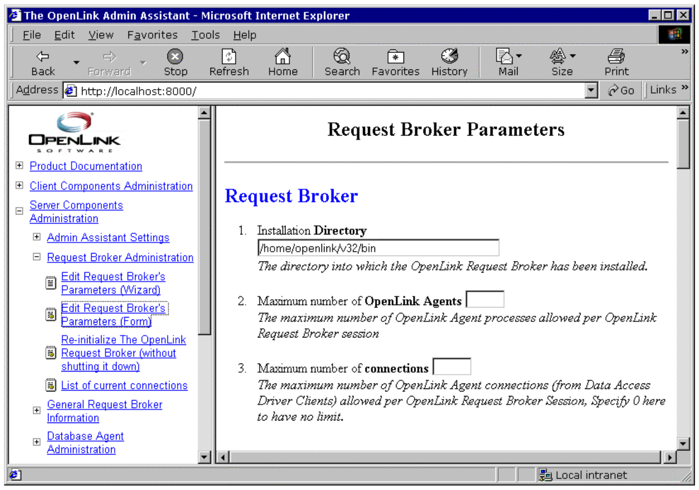
Broker settings have been saved.

OpenLink Request Broker Reinitialized Successfully.

Using Admin Assistant Forms

You can also choose to configure your Request Broker's session settings via a Forms as opposed to Wizard based interface. You select this option from the Admin Assistant and you will be presented with a screen similar to the snapshot below.

Each Broker session settings field contains a description of the values that need to be entered into each of the forms fields. In addition a brief description of the use of each of these values is provided below each field.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print
Address http://localhost:8000/
Go Links
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
Edit Request Broker's Parameters (Wizard)
Edit Request Broker's Parameters (Form)
Re-initialize The OpenLink Request Broker (without shutting it down)
List of current connections
General Request Broker Information
Database Agent Administration
Local intranet

Request Broker Parameters

Request Broker

1. Installation **Directory**
   /home/openlink/v32/bin
   The directory into which the OpenLink Request Broker has been installed.
2. Maximum number of **OpenLink Agents**
   The maximum number of OpenLink Agent processes allowed per OpenLink Request Broker session
3. Maximum number of **connections**
   The maximum number of OpenLink Agent connections (from Data Access Driver Clients) allowed per OpenLink Request Broker Session, Specify 0 here to have no limit.

OpenLink Database Agents Administration

Introduction

OpenLink Database Agents are the only database specific components within the OpenLink Multi-Tier middleware framework. It is these components that actually initiate and maintain database sessions with your OpenLink Clients, basically playing the role of a data access server.

Database Agents are servers implementing client data access interfaces such as ODBC, JDBC, UDBC, and OLE-DB using the lower level native call level interfaces provided by each supported database engine. These call level interfaces are themselves increasing based on the X/Open SQL Call Level Interface (CLI) specification. The older database engines that do not support or implement the X/Open SQL CLI specification simply provide traditional Embedded SQL interfaces.

The fact that a Database Agents are built using natives interfaces provided by database engine vendors has some very important implications:

1. Through the eyes of a backend database, a database agent is a database client, no different to any other native client provided or bundled with the backend database engine.

2. A database agent inherits all of the functionality of a traditional database client, this includes: Distributed Database, SQL syntax, Stored Procedure support, and anything else that may be specific to the relevant backend database engine.

3. Configuring a database agent is no different to configuring an native database client, they share operating system environment variable dependencies etc..

4. Resource utilization is identical to resource consumption for native clients, this means that if you have special setting for your native client sessions, you will be able to apply these when configuring your database agents.

5. Any efficiencies or deficiencies in the database engines CLI client to database server inter process communications (IPC) also affects a database agent.

Obtaining Database Agent Information

OpenLink Database agents have a specific naming convention, reflecting the fact that these components are specific to both a particular database engine, and in some case specific versions of a given database engine.

The OpenLink executable binary file naming convention consists of three distinctive logical parts:

- Database Engine - first three characters
- Database Version - next two-three characters (depending on database engine version number-functionality issues)
- OpenLink Service Provider Identifier - the characters "_sv"

The table below shows you how the current pool of OpenLink Database agents are identified based on the convention described above:

Executable Binary File	**Database Engine & Version**
virt_sv	OpenLink Virtuoso
inf5_sv	Informix 5.x
inf7_sv	Informix 7.x
ing6_sv	Ingres 6.4.x
oig1_sv	OpenIngres 1.x
ing7_sv	Ingres II
pro63e_sv	Progress 63E
pro73c_sv	Progress 73C
pro73e_sv	Progress 73E
pro82a_sv	Progress 82A
pro82c	Progress 82C
pro83a	Progress 83A
pro90a_sv	Progress 9
ora6_sv	Oracle 6.x
ora7_sv	Oracle 7.x
ora8_sv	Oracle 8.x
syb4_sv	Sybase 4.x
syb10_sv	Sybase 10.x (DBLIB based)
sybc10_sv	Sybase 10.x (CTLIB based)
sybc11_sv	Sybase 11.x
sql6_sv	Microsoft SQL Server 6.x & 7.x
db2_sv	DB2 v2.x
sol_sv	Solid
velo_sv	Velocis
uni_sv	Unify

Version, Release, and Functionality Related Information

In a manner similar to the Request Broker, you can obtain component version, release, and functionality related information about your database agent through your operating system's command line interface.

To obtain the information about your database agent simply type in the name of the binary executable file for the relevant agent and the "-?" switch. The example below shows how this is done assuming you are seeking information about the OpenLink Virtuoso™ database agent:

```
virt_sv -?
```

The output returned is depicted below:

```
OpenLink Virtuoso Database Agent
Version 1.2 (Release 3.2) as of Fri Feb 05 1999 (cvsid 00048).
Compiled for Linux 2.0.36 (i586-pc-linux-gnulibc1)
Copyright (C) OpenLink Software.

Usage:
virt_sv [-CmijrlLd] [+noautocommit] [+maxrows num] [+initsql arg] [+jetfix] [+norowsetlimit] [+loglevel num]
[+logfile arg] [+debug]

+noautocommit turn autocommit off by default
+maxrows maximum allowed rows to fetch
+initsql execute SQL from this file for every connection made
+jetfix Jet Engine Compatibility Features
+norowsetlimit tuer that relates to the actual source code archive from which a database agent has been
assembled.
```

Binary Platform - indicates what platform the database agent has been built to run on.

USAGE - describes the command line options that are to be used with a particular database agent, these can be database agent specific. The command line options are to be used within the Database Agent configuration section of "Session Rules Book" (the file oplrqb.ini), this is the section that handles database agent specific items. You can add these entries as required to the rule book, you do so by manually editing the rule book file or via the Web based Admin Assistant (see section on Configuring Database Agents).

Configuring OpenLink Database Agents

All database engines operate under a client-server paradigm, that is to say there are always two distinct processes involved in a database session, the database server and the database client. The database server must be up and running before a database client can communicate with a database. Database client and server processes may or may not reside on the same physical machine, at the same time this has no bearing on the fundamentals of database client and server process interaction as just explained.

Every database engine has a one or more key values that need to be set in order for database clients to be able to communicate with database servers. These values take the form of host operating system environment variables, database connection string formats/parameters, or a combination of both.

Configuring your OpenLink database agent is all about creating session initialization templates in the Sessions Rules Book which map key database client values with OpenLink Agent Tempalate Attributes.

OpenLink provides two user friendly utilities for configuring your database agents, namely the OpenLink Admin Assistant and the OpenLink Configuration Utility ("oplcfg"). A third option is to edit the rule book manually, but the availability of these utilities makes this a less recommended option, certainly one for experienced OpenLink users only.

OpenLink Configuration Utility ("oplcfg")

At the end of the end of your OpenLink server components installation process, the installation program will automatically start the "oplcfg" utility, thereby enabling you to configure your database agent(s) as part of the installation process.

If you have already installed your server components successfully but need to make changes to your initial database agent setting then you can initialize the "oplcfg" utility from the your host operating system's command line prompt, simply enter the following command from within the "bin" sub-directory under your base OpenLink server components installation directory:

oplcfg

You will then be presented with a character based menu system as depicted below:

```
 OpenLink Server Components Configuration Utility

1.  Request Broker                11. PostgresSQL
2.  Informix 5                    12.  Progress 6
3.  Informix 6                    13.  Progress 7
4.  Informix 7                    14.  Progress 8
5.  Progress 6                    15.  Solid
6.  Virtuoso                      16.  Sybase 4
7.  OpenProgress                  17.  Sybase 10
8.  Oracle 6                      18.  Sybase 11
9.  Oracle 7                      19.  Unify 2000
10. Oracle 8                      20.  Velocis

U.  Undo last change              V.  View the current Rules Book
C.  Clear log file                L.  View log file
B.  Backup Rules Book             R.  Restore Rules Book
I.  Verify Rules Book             N.  Reinitialize running Broker
S.  Startup  Request Broker       D.  Shutdown Request Broker
```

Choose an item or type q to quit :

You then enter a number that matches the database engine that you will configuring your database agent(s) to work with.

You will be prompted for each value.

Once this is completed you will have to edit the rule book (the file "oplrqb.ini") manually in order to set up any database session specific values. Another option is to start up the Admin Assistant which provides an intuitive GUI for configuring your database agents.

OpenLink Admin Assistant

The preferred and much more flexible way of configuring your database agents is through the OpenLink Admin Assistant. This is a powerful HTML based GUI that is usable from any Web Browser, it provides you with two approaches to configuring database agents using Wizards or forms.

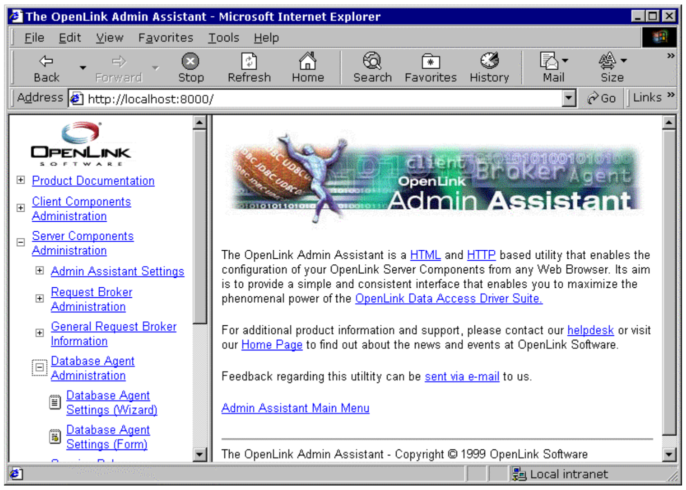
Wizard Based Database Agent Administration

This approach to database agent administration is going to be exemplified using OpenLink Virtuoso™ as the database agent that is being administered. Please note that nothing in this example is database specific beyond the actual values entered. A database specific section follows which addresses these issues on database engine specific basis.

1. Start the Admin Assistant by entering the following URL into your browser: http://<IP Address or Network Alias of machine hosting the OpenLink Database Agents to be Administered>:<Admin Assistant Port number>

   For a network machine aliased as "mainserver" using the default OpenLink Admin Assistant port "8000" the URL required would be entered as follows: http://mainserver:8000

You will be presented with a screen similar to the one below:


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://localhost:8000/
Go Links
OpenLink Software
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Client Broker Agent
OpenLink Admin Assistant
Local intranet

The OpenLink Admin Assistant is a HTML and HTTP based utility that enables the configuration of your OpenLink Server Components from any Web Browser. Its aim is to provide a simple and consistent interface that enables you to maximize the phenomenal power of the OpenLink Data Access Driver Suite.

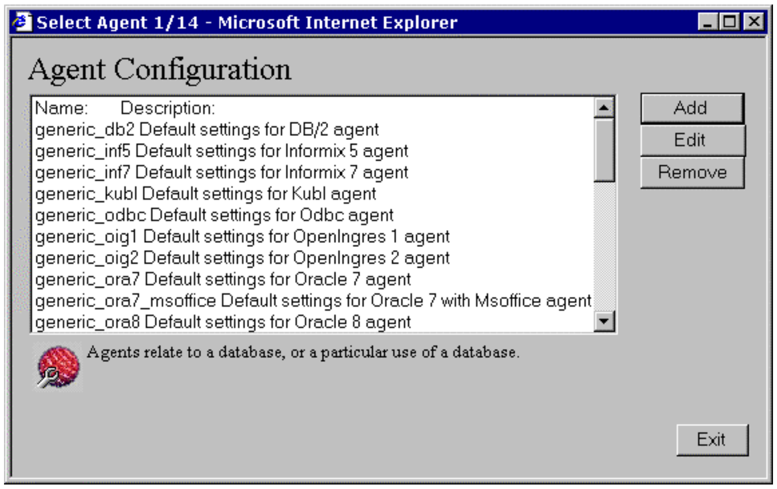
For additional product information and support, please contact our helpdesk or visit our Home Page to find out about the news and events at OpenLink Software.

Feedback regarding this utiltity can be sent via e-mail to us.

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. Expand the "Server Components Administration", and "Database Agent Administration" menus. Once this is done you need to then click on the "OpenLink Database Agent Settings (Wizard)" hyperlink. This will bring you to the first Database Agent Wizard dialog as depicted below:

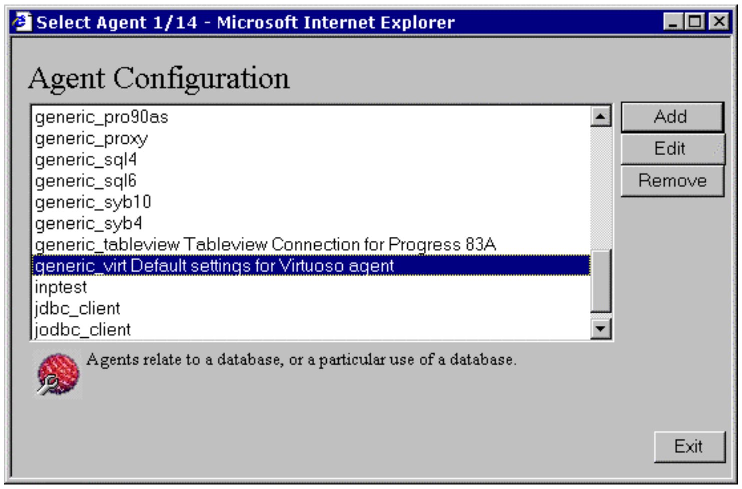


Agent Configuration

```
Name:     Description:
generic_db2 Default settings for DB/2 agent
generic_inf5 Default settings for Informix 5 agent
generic_inf7 Default settings for Informix 7 agent
generic_kubl Default settings for Kubl agent
generic_odbc Default settings for Odbc agent
generic_oig1 Default settings for OpenIngres 1 agent
generic_oig2 Default settings for OpenIngres 2 agent
generic_ora7 Default settings for Oracle 7 agent
generic_ora7_msoffice Default settings for Oracle 7 with Msoffice agent
generic_ora8 Default settings for Oracle 8 agent
```

Agents relate to a database, or a particular use of a database.

3. From the list of pre-configured database agents select one that matches the database engine that your database agent will be communicating with, note that every database supported by OpenLink will have a pre-configured default agent initialization template listed.



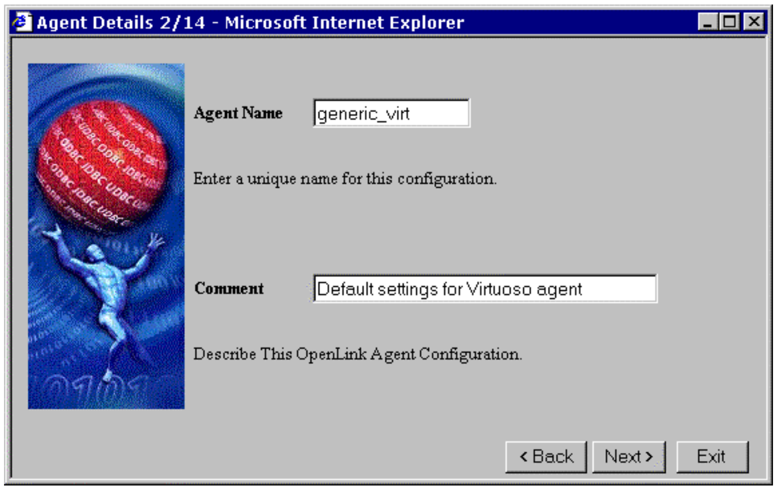
Select Agent 1/14 - Microsoft Internet Explorer

Agent Configuration

generic_pro90as
generic_proxy
generic_sql4
generic_sql6
generic_syb10
generic_syb4
generic_tableview Tableview Connection for Progress 83A
generic_virt Default settings for Virtuoso agent
inptest
jdbc_client
jodbc_client

Agents relate to a database, or a particular use of a database.

4. Click the "Edit" button to commence re configuring this database agent for your specific needs. Enter a name in the "Agent Name" field that is to be used to identify the database agent that your are configuring (it is recommended that you keep the default, if you do opt for a new name please don't enter names that contain spaces), also enter text into the "Comment" field describing your database agent. Once completed click on the "Next" button.



Agent Details 2/14 - Microsoft Internet Explorer

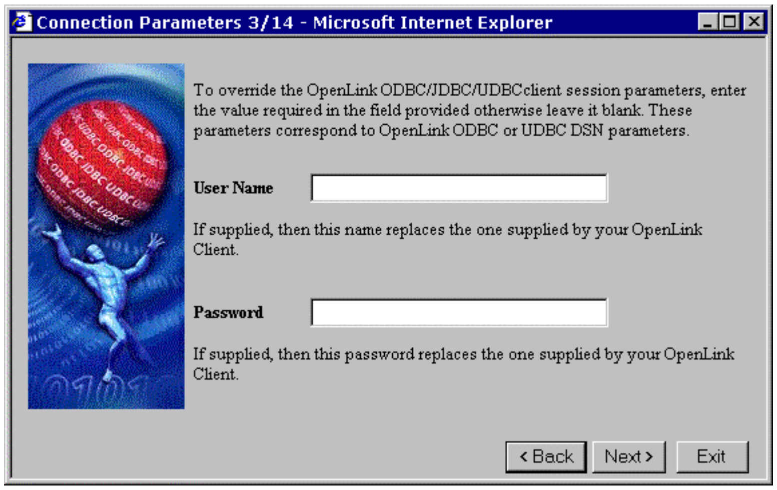
Agent Name: generic_virt

Enter a unique name for this configuration.

Comment: Default settings for Virtuoso agent

Describe This OpenLink Agent Configuration.

5. You can force your OpenLink clients to connect a backend database using a pre-assigned username and password combination, this is done by entering the values that you want to enforce into the "User Name" and "Password" fields in the Wizard dialog depicted below. This activity implies that irrespective of what username password combination is entered at OpenLink client configuration or connect time, the values that you provide will take precedence. If you do not want to enforce username and password combinations for your OpenLink clients simply leave the "User Name" and "Password" fields blank. Click on the "Next" button to continue.



To override the OpenLink ODBC/JDBC/UDBC client session parameters, enter the value required in the field provided otherwise leave it blank. These parameters correspond to OpenLink ODBC or UDBC DSN parameters.

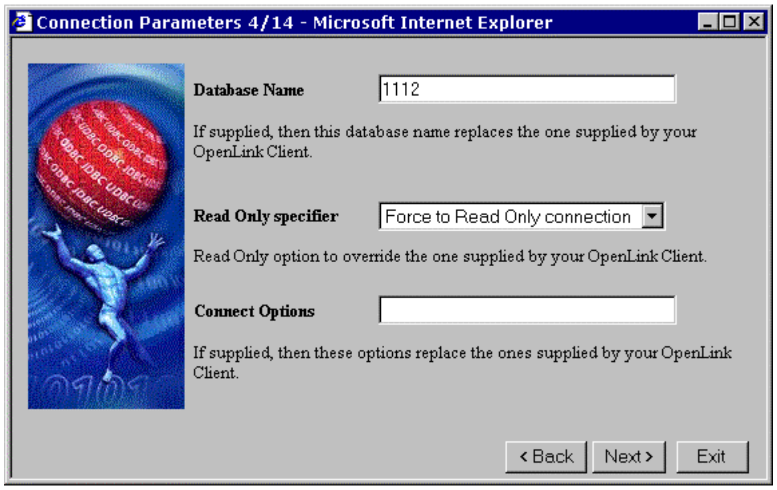
User Name

If supplied, then this name replaces the one supplied by your OpenLink Client.

Password

If supplied, then this password replaces the one supplied by your OpenLink Client.

6. Enter a value in the "Database Name" field that identifies the actual database within your database engine's environment that you want to connect your OpenLink clients with, the format of these values vary by database engine type. You can control what type of session(s) your OpenLink clients conduct with your backend database by choosing the appropriate value from the "Read Only Specifier" listbox. If you are connecting to a database that does not reside on the same machine as your database agent, or circumstances require you to use the networking middleware provided by your database vendor (which is already installed on the same machine as the database agent), then you can enter the remote database connection values required by your remote database into the "ConnectOptions" field. Once completed click the "Next" button to continue.



Database Name 1112

If supplied, then this database name replaces the one supplied by your OpenLink Client.

Read Only specifier Force to Read Only connection

Read Only option to override the one supplied by your OpenLink Client.

Connect Options

If supplied, then these options replace the ones supplied by your OpenLink Client.

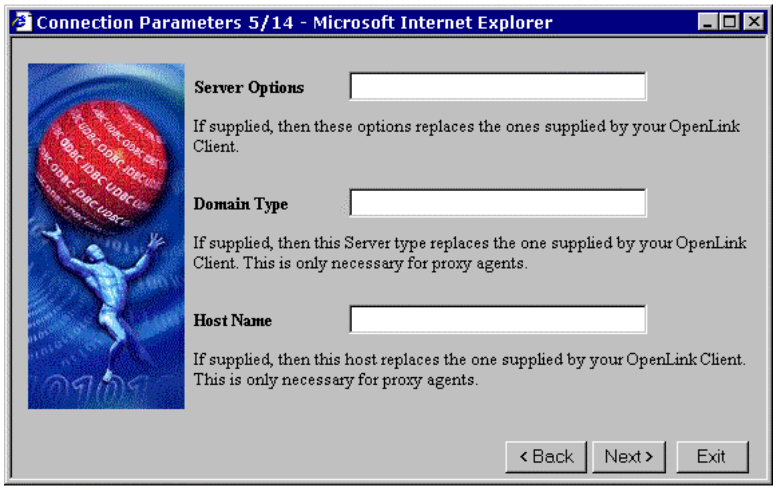
7. Some database engines allow their clients to configure session resources, in situations where this holds true you can use the "Server Options" field to set these values for your database agent sessions.

   You can also configure your database agents such that the database agent that services OpenLink clients acts as a PROXY to other database agents residing on a remote OpenLink server(s) within your network infrastructure. This is basically a 3-Tier or N-Tier Distributed Database Agent configuration and the key configuration values go into the "Domain Type" and "Host Name" fields.

   Domain Type - this is your OpenLink agent Server type (Virtuoso, Oracle. Informix, ODBC, PROXY, etc..), in the case of a typical 3-tier configuration you would enter the value PROXY, this implies that you are going to a PROXY agent rather than an database agent.

   Host Name - this is how you identify the machine hosting the Request Broker that binds your OpenLink clients with the OpenLink agent designated by the value entered into the "Domain Type" field.

   In most case all of these fields can simply be left blank. Click on the "Next button to continue.



Connection Parameters 5/14 - Microsoft Internet Explorer

Server Options

If supplied, then these options replaces the ones supplied by your OpenLink Client.

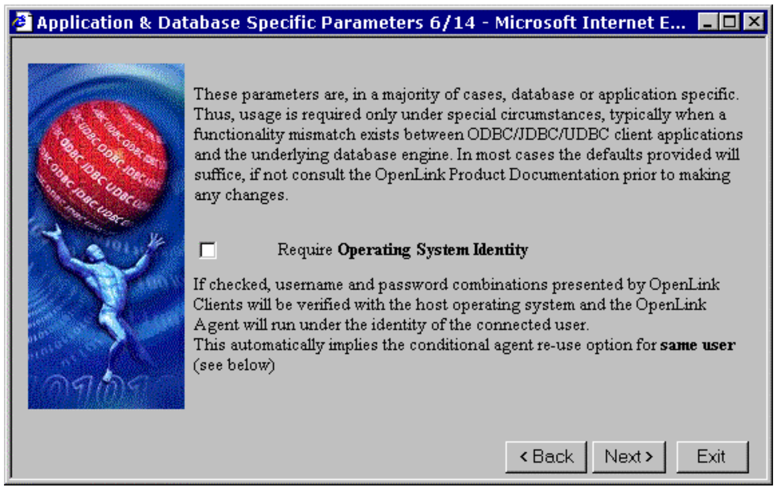
Domain Type

If supplied, then this Server type replaces the one supplied by your OpenLink Client. This is only necessary for proxy agents.

Host Name

If supplied, then this host replaces the one supplied by your OpenLink Client. This is only necessary for proxy agents.

8. For security purposes there are times when you want to validate your OpenLink clients at the operating system level before they actually initiate sessions with your backend database. In some cases backend databases presume that your ability to attempt database connection implies your being a valid operating system users, this can be a major security hole for those database engines that do not conduct their own username and password verification. By hatching the "Require Operating System Identity" checkbox your database agent will validate OpenLink clients at the operating system level before attempting a connection to your backend database. Click the "Next" button to continue.



Application & Database Specific Parameters 6/14 - Microsoft Internet E...

These parameters are, in a majority of cases, database or application specific. Thus, usage is required only under special circumstances, typically when a functionality mismatch exists between ODBC/JDBC/UDBC client applications and the underlying database engine. In most cases the defaults provided will suffice, if not consult the OpenLink Product Documentation prior to making any changes.

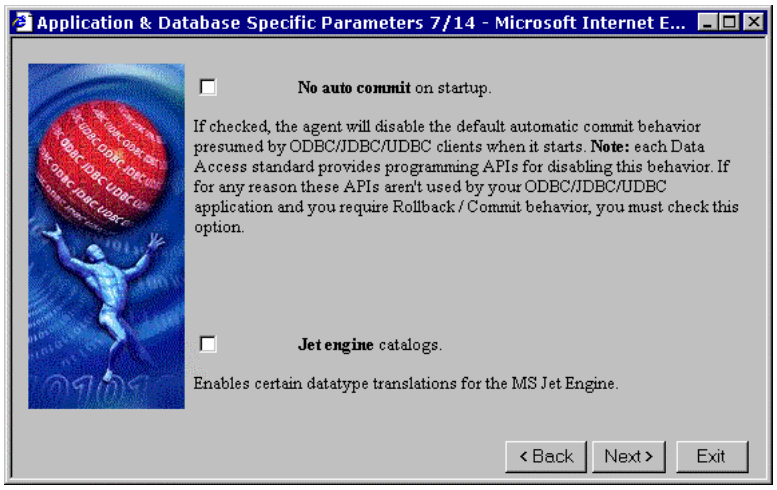
☐ Require **Operating System Identity**

If checked, username and password combinations presented by OpenLink Clients will be verified with the host operating system and the OpenLink Agent will run under the identity of the connected user.
This automatically implies the conditional agent re-use option for **same user** (see below)

9. You can enforce consistent transaction behavior across your OpenLink clients through the "No Auto Commit On Startup" field, this ensures that database commands originating from OpenLink clients do not contain trailing "Commit" instructions. Without this restriction database integrity could easily be compromised by incomplete transactions creating broken records. You enable this protection by hatching the "No Auto Commit On Startup" checkbox.

   You can also enforce application specific datatype translation handling specifically for Microsoft Jet Engine based OpenLink ODBC clients by hatching the "Jet Engine Catalogs" checkbox. This is a very application specific feature that is only required for when Microsoft's Jet engine is the OpenLink ODBC client.

   Click the "Next" button to continue.


Application & Database Specific Parameters 7/14 - Microsoft Internet E...
< Back
Next >
Exit

☐ **No auto commit** on startup.

If checked, the agent will disable the default automatic commit behavior presumed by ODBC/JDBC/UDBC clients when it starts. **Note:** each Data Access standard provides programming APIs for disabling this behavior. If for any reason these APIs aren't used by your ODBC/JDBC/UDBC application and you require Rollback / Commit behavior, you must check this option.

☐ **Jet engine** catalogs.

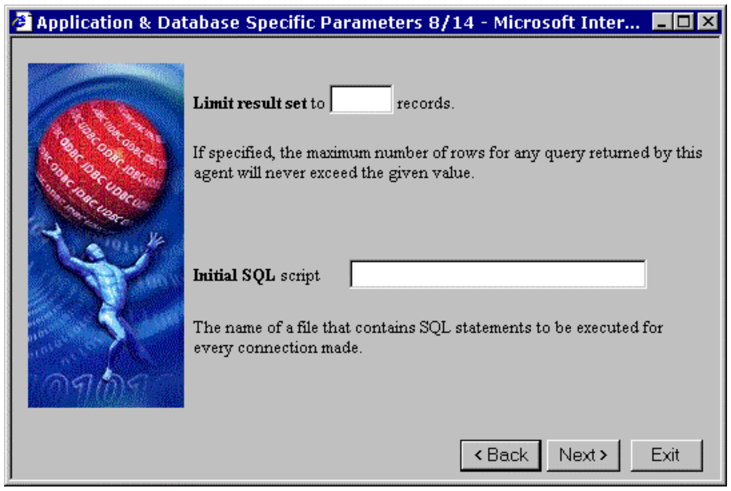
Enables certain datatype translations for the MS Jet Engine.

10. There are times when you may want to restrict the number of records that your database agent transmits to your OpenLink clients. In such scenarios you can enter a numeric value into the "Limit result set" field that represents the maximum number of records from a database query resultset that your database agent should transmit back to your OpenLink clients. This features protects your network from "Innocent Queries From Hell", these are queries that end-user unknowingly generate when using visual query tools, especially as they familiarize themselves with the concepts of SQL querying.

    There are also times when critical database functionality may not be implemented as part of the ODBC. JDBC., UDBC,or OLE-DB specifications, but you need to make use of such functionality in order to run your database infrastructure smoothly. When this situation arises you are able to use the "Initial SQL script" field to enter a value that points to a script file containing a set of SQL instructions that implemented the desired functionality.

    In some cases you may find yourself having to deal with the fact that although the functionality may be implemented at the ODBC, UDBC, JDBC, OLE-DB specification level, the client application connecting to your database via OpenLink simply isn't making use of this functionality. In this scenario the "Initial SQL Script" field comes in handy. A typical example is default Transaction Isolation Levels handling.

    Click "Next" to continue.


Application & Database Specific Parameters 8/14 - Microsoft Inter...
< Back
Next >
Exit

Limit result set to [ ] records.

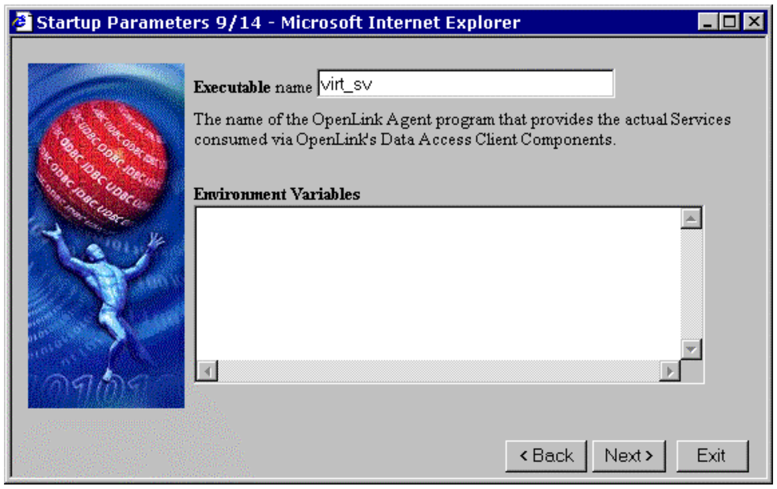
If specified, the maximum number of rows for any query returned by this agent will never exceed the given value.

Initial SQL script [ ]

The name of a file that contains SQL statements to be executed for every connection made.

11. You identify the binary executable file that represents the specific database agent for your backend database by entering the binary files name in the "Executable Name" field. This set for you by default and unless you rename this file yourself it does not need to be changed.

    Many database environments are driven by operating system level environment variables, the "Environment Variables" field allows you to set or re-configure these values (see database specific settings section). Click the "Next" button to continue.



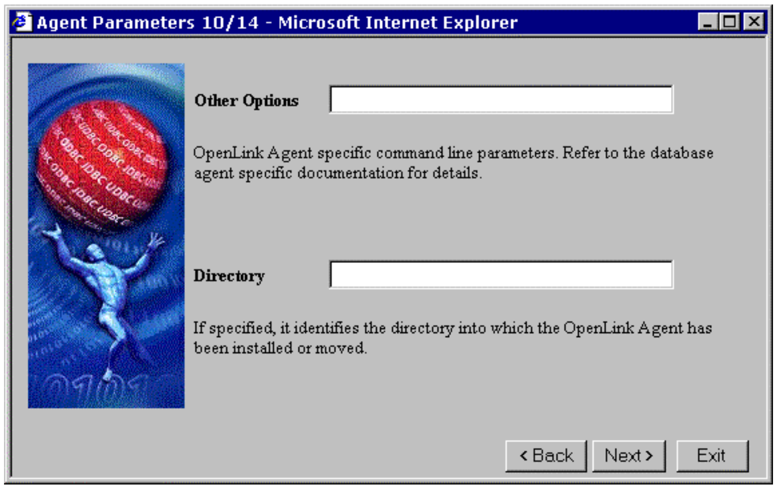
Executable name virt_sv

The name of the OpenLink Agent program that provides the actual Services consumed via OpenLink's Data Access Client Components.

Environment Variables

12. There may be some database agent specific options that you need to apply specifically to sessions with your backend database (see usage part of "obtaining database agent information" section for list of values), you enter these values into the "Other Options" field.

    You also need to indicate the directory in which your database agent executable binary file resides, this is only required if you have moved these files from their default location after installation. Click the "Next Button" to continue.



Other Options

OpenLink Agent specific command line parameters. Refer to the database agent specific documentation for details.

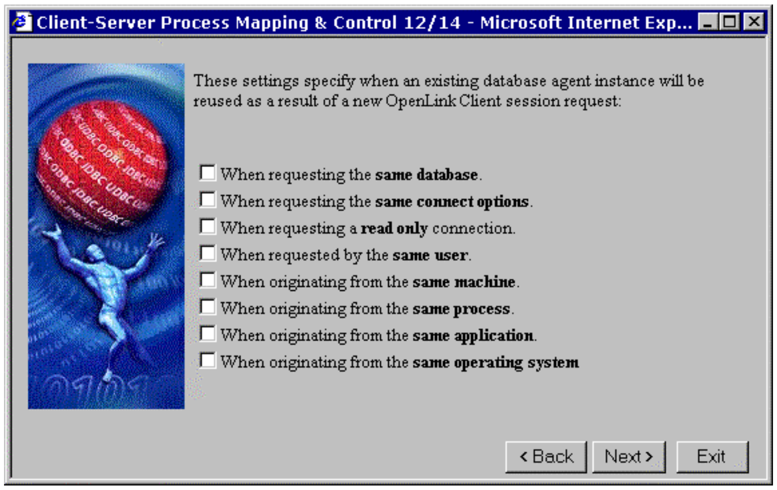
Directory

If specified, it identifies the directory into which the OpenLink Agent has been installed or moved.

13. Your database agents act as servers to your OpenLink clients, this implies that as server processes they do consume server operating system resources. You can control resource consumption by predefining how many database agent instances are to be started as a result of OpenLink clients connections, and the basis upon which these instance are reused by subsequent OpenLink client connections. The "Never", "Always", and "Conditionally" radio button allows you to choose the option that best suits your infrastructure's needs.

    The "Accept Up To" field allows you to enter a numeric value that indicates the maximum number of new database agent processes that can be instantiated on a given OpenLink server operating environment.

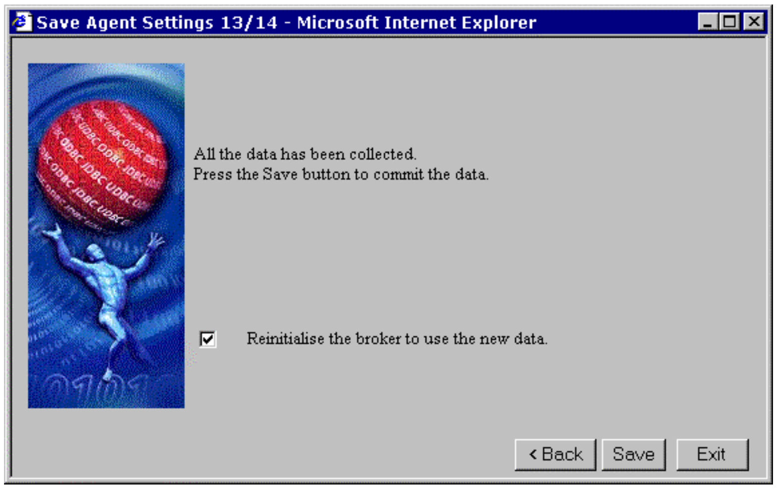
    If you choose the "Conditionally" radio button and then click the "Next" button you will be presented with an additional dialog with a list of checkboxes. These checkboxes allow you to customize the sets of circumstances under which you want an OpenLink client connection to result in the instantiation of a new database agent instances. Click the next "Button" to continue.


Client-Server Process Mapping & Control 12/14 - Microsoft Internet Exp...

These settings specify when an existing database agent instance will be reused as a result of a new OpenLink Client session request:

- ☐ When requesting the **same database**.
- ☐ When requesting the **same connect options**.
- ☐ When requesting a **read only** connection.
- ☐ When requested by the **same user**.
- ☐ When originating from the **same machine**.
- ☐ When originating from the **same process**.
- ☐ When originating from the **same application**.
- ☐ When originating from the **same operating system**

14. At the end of this Wizard interaction you can opt to make you database agent settings available to the next OpenLink client without shutting down and restarting the Request Broker, you do this by hatching the "Reinitialize" checkbox and then clicking the "Save" button to save your settings.



All the data has been collected.
Press the Save button to commit the data.

☑ Reinitialise the broker to use the new data.

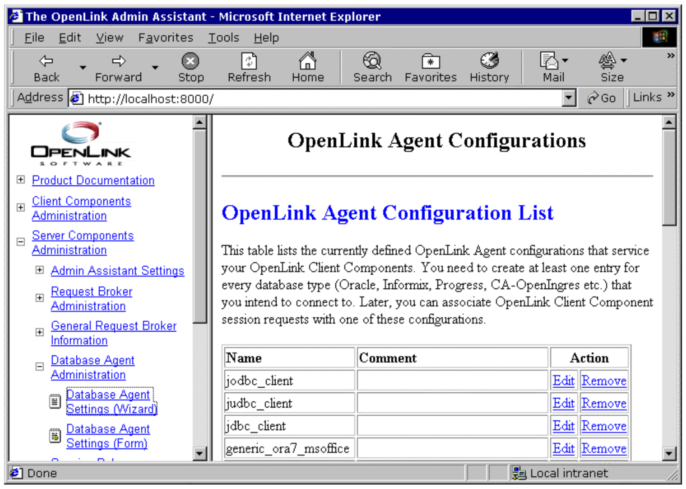
Forms Based Database Agent Administration

You can also administer database agents through the Admin Assistant using a "Forms" as opposed to "Wizards" based approach, the steps that follow guide you through this process using the same example (configuring the a database agent for the OpenLink Virtuoso™ database).

1. Start the Admin Assistant by entering the following URL into your browser: http://<IP Address or Network Alias of machine hosting the OpenLink Database Agents to be Administered>:<Admin Assistant Port number>

   For a network machine aliased as "mainserver" using the default OpenLink Admin Assistant port "8000" the URL required would be entered as follows: http://mainserver:8000

   Expand the menus for "Server Components Administration", and "Database Agent Administration". Once this is done you need to then click on the "Database Agent Settings (Form)" hyperlink. This will bring you to a database agent listing as depicted below:



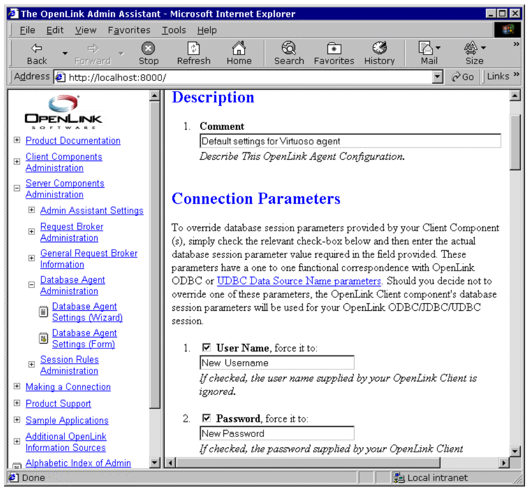
OpenLink Agent Configurations

OpenLink Agent Configuration List

This table lists the currently defined OpenLink Agent configurations that service your OpenLink Client Components. You need to create at least one entry for every database type (Oracle, Informix, Progress, CA-OpenIngres etc.) that you intend to connect to. Later, you can associate OpenLink Client Component session requests with one of these configurations.

Name	Comment	Action	
jodbc_client		Edit	Remove
judbc_client		Edit	Remove
jdbc_client		Edit	Remove
generic_ora7_msoffice		Edit	Remove

2. As our example for this exercise is based on the OpenLink Virtuoso database agent, click on the "Edit" hyperlink for the "generic_virt" default database agent (note you can substitute this with the default database agent that matches your backend database). Once these actions are completed you will be presented with the main database agent initialization template form. Complete the values for the fields that apply to needs and then click on the "Update" button at the bottom of the form to save your changes.



Description

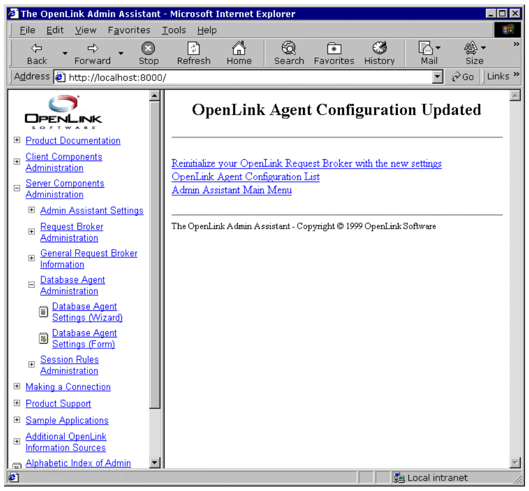
1. **Comment**
   Default settings for Virtuoso agent
   Describe This OpenLink Agent Configuration.

Connection Parameters

To override database session parameters provided by your Client Component (s), simply check the relevant check-box below and then enter the actual database session parameter value required in the field provided. These parameters have a one to one functional correspondence with OpenLink ODBC or UDBC Data Source Name parameters. Should you decide not to override one of these parameters, the OpenLink Client component's database session parameters will be used for your OpenLink ODBC/JDBC/UDBC session.

1. ☑ **User Name**, force it to:
   New Username
   If checked, the user name supplied by your OpenLink Client is ignored.
2. ☑ **Password**, force it to:
   New Password
   If checked, the password supplied by your OpenLink Client

3. Click on the "Reinitialize Your OpenLink Request Broker With New Settings" hyperlink, this enables your new setting to be applied to subsequent connections from OpenLink clients without disrupting existing OpenLink client sessions.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size
Address http://localhost:8000/
Go Links
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Alphabetic Index of Admin
Local intranet

OpenLink Agent Configuration Updated

Reinitialize your OpenLink Request Broker with the new settings
OpenLink Agent Configuration List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

Database Specific Settings

OpenLink database agents are database clients built using the SQL Call Level or Embedded SQL interfaces of the respective supported backend database engines. Thus, the process of configuring or administering a database agent is similar in essence to what you would have to do if you were administering a native database client.

Database engines use environment variables to creating a database specific operating space within which database clients and servers interact, these environment typically address the following important database session related issues:

- Database server identification - your database client needs to be able to connect to the appropriate database server, many database implementations support multiple database servers instances listening for client connections at different network ports on the same machine (e.g. OpenLink Virtuoso, Microsoft SQL Server, Sybase, Progress, Informix etc.).

- Database engine base installation directory - many database engine environments comprise shared or dynamically linked libraries and other runtime components that are required by database clients. Thus, there is a need for database clients to have a sense of what the actual base or root point of the database engine installation is, this enables the construct of a component search path (similar to the "PATH" operating system environment variable) at runtime.

- Database session resource allocation - most database engines allow database session resources to be configured for clients via environment variables (sometimes these variables simply identify resource configuration files).

- Database session optimization - some database environments allow query optimizers, network packet sizes, and records transmission values for database clients to be configured via environment variables.

The sections that follow address specific environment settings that affect the configuration of your OpenLink database agents, the values provided can supplanted values used in the Admin Assistant configuration examples provided in the prior section.

Virtuoso Specific Configuration Issues

When configuring a Virtuoso database agent the critical configuration items are:

- Database Identification - this is the Virtuoso Database Server's port number, which identifies the actual Virtuoso server process that links you to an actual virtuoso database file. This is the value that you enter into the "Database Name" field of either your Admin Assistant form or wizard dialog.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Virtuoso database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Virtuoso database on your Database Server machine using virtuoso's database specific networking as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink's Database independent Networking, while the communication between the Virtuoso database agent and the Virtuoso database server uses Virtuoso database specific networking.

Configuration Steps:

Assuming you have a Virtuoso Database Server called "mainserver2" that has a Virtuoso server process listening for clients at port 1112

1. Ensure that you have a usable connection to Virtuoso using its native networking.

2. Add the following value to the "Database Name" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same values into the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

   mainserver2:1112

Informix Specific Configuration Issues

When configuring an Informix database agent the critical configuration items are:

- Database Identification - this is an actual database name e.g "stores7", which identifies the actual Informix database file that you want to be connected with. This is the value that you enter into the "Database Name" field of either your Admin Assistant's database agent configuration form or wizard dialog. If you choose to have database identification take place at the client rather than server, you enter this value into the "Database Name" field or connection attribute when configuring your OpenLink client.

Informix provides a number of environment variables for configuring database clients, the basic set required for successfully connecting your database agent to an Informix database server are tabulated below:

[Environment INFORMIX5]	**Default Rule Book Settings**	**Notes**
INFORMIXDIR=	/dbs/informix5	Full path to the base directory for the Informix 5 installation.
[Environment INFORMIX6]	**Default Rule Book Settings**	**Notes**
INFORMIXDIR=	/dbs/informix6	Full path to the base directory for the Informix 6 installation.
[Environment INFORMIX7]	**Default Rule Book Settings**	**Notes**
INFORMIXDIR=	/dbs/informix7	Full path to the base directory for the Informix 7 installation.
INFORMIXSERVER=	Alpha	The name of Informix 7 server that you want the agent to attach to. As long as you have I-Connect or I-Net installed, configured and up and running this value can connect your database agent with remote Informix database servers.

Security Enhancement

Due to the fact that Informix leaves username and password verification to the host operating system, it is possible to close what could be an ODBC, UDBC, JDBC, or OLE-DB security loophole by utilizing the OpenLink database agent "OpsysLogin" facility which can be enabled through the Admin Assistant. By enabling this feature your Informix database agent will verify user accounts at the operating system level before attempting to connect to your Informix database. It is important to note that "super-user" or Administrator (depending on operating system) privileges are required to successfully use this feature. This implies that the account that starts the request broker must possess one of the aforementioned system level privileges, on the other hand these privileges aren't required for your actual OpenLink client sessions.

Rebuilding Informix Database agents

OpenLink provides a relinkable library and script files that enable you to rebuild your database agents as shared, as opposed to statically linked binaries, or for the purposes of getting a closer database implementation fit, should your Informix database environment be a more recent release than the actual version used by OpenLink to build the database agent installed on your system. Please read the Relinking OpenLink Database Agents document for details on how to perform this task.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Informix database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Informix database on your Database Server machine using Informix database specific networking (I-Connect or I-Net) as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink Database independent Networking, while the communication between the Informix database agent and the Informix database server uses I-Connect or I-Net (depending on Informix version).

Configuration Steps:

Assuming you have an Informix Database Server machine called "mainserver2" that has an Informix I-Connect or I-Net server process running (this is setup and configured via the SQLHOSTS file on the database server machine).

1. On your Application Server (the machine hosting your database agent) create an I-connect or I-Net Connection Alias called "mainserver2" (for purpose of this example only) if a working Connection Alias doesn't already exist on this machine.

2. Ensure that you have a usable connection to your remote Informix database using Connection Alias "mainserver2".

3. Add the following values to the "Database Server Options" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same value into to the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

   mainserver2

   You can also set the INFORMIXSERVER environment variable to "mainserver2".

You can see illustrations of the various distributed client-server architectures supported by OpenLink database agent by clicking here.

Ingres Specific Configuration Issues

When configuring an Progress database agent the critical configuration items are:

- Database Identification - this is an actual database name e.g "demo", which identifies the actual Progress database file that you want to be connected with. This is the value that you enter into the "Database Name" field of either your Admin Assistant's database agent configuration form or wizard dialog. If you choose to have database identification take at the client rather than server, you enter this value into the "Database Name" field or connection attribute when configuring your OpenLink client.

Progress provides a number of environment variables for configuring database clients, the basic set required for successfully connecting your database agent to an Progress database server are tabulated below:

[Environment Progress7]	**Default Rule Book Settings**	**Notes**
II_DATE_FORMAT=	US	Defines the output format for dates as *dd-mmm-yyyy.* This should not be changed inside the Rule Book since it enables the best compatibility with OpenLink. This will not affect any other Progress applications.
II_SYSTEM=	/dbs	Full path to the directory immediately below the Progress/ directory e.g. if your Progress installation directory is /dbs/Progress then set this to /dbs

Security Enhancement:

Due to the fact that Ingres 6.4 leaves username and password verification to the host operating system (Ingres II does not have this problem), it is possible to close what could be an ODBC, UDBC, JDBC, or OLE-DB security loophole by utilizing the OpenLink database agent "OpsysLogin" facility which can be enabled through the Admin Assistant. By enabling this feature your Ingres database agent will verify user accounts at the operating system level before attempting to connect to your Ingres database. It is important to note that "super-user" or Administrator (depending on operating system) privileges are required to successfully use this feature. This implies that the account that starts the request broker must possess one of the aforementioned system level privileges, on the other hand these privileges aren't required for your actual OpenLink client sessions.

Rebuilding Ingres Database Agents

OpenLink provides a relinkable library and script files that enable you to rebuild your database agents as shared, as opposed to statically linked binaries, or for the purposes of getting a closer database implementation fit, should your Ingres database environment be a more recent release than the actual version used by OpenLink to build the database agent installed on your system. Please read the Relinking OpenLink Database Agents document for details on how to perform this task.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Ingres database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Ingres database on your Database Server machine using Ingres database specific networking (Ingres Net) as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink Database independent Networking, while the communication between the Ingres database agent and the Ingres database server uses Ingres Net.

Configuration Steps:

Assuming that you have an Ingres Database server machine called "mainserver2" that has an Ingres Net server process running.

1. On your Application Server (the machine hosting your database agent) create an Ingres Net vnode called "mainserver2" (for purpose of this example only) if you do not have a working vnode on this machine.

2. Ensure that you have a usable connection to your remote Ingres database using the vnode "mainserver2".

3. Add the following values to the "Server Options" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same value into to the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

   mainserver2

You can see illustrations of the various distributed client-server architectures supported by OpenLink database agent by clicking here.

Progress Specific Configuration Issues

When configuring a Progress database agent the critical configuration items are:

- Database Identification - this is an actual database name e.g "demo" or "isports", which identifies the actual Progress database file that you want to be connected with. This is the value that you enter into the "Database Name" field of either your Admin Assistant form or wizard dialog. If you choose to have database identification take at the client rather than server, you enter this value into the "Database Name" field or connection attribute when configuring your OpenLink client.

Progress provides a number of environment variables for configuring database clients, the basic set required for successfully connecting your database agent to an Progress database server are tabulated below:

[Environment PROGRESS8]	**Default Rule Book Settings**	**Notes**
DLC=	/dbs/dlc8	Must be full path to the Progress *dlc* directory.
PROCFG=	/dbs/dlc8/progress.cfg	Must be the full path and filename to the progress.cfg file.
TABLEVIEW=		Must be the full path and filename to the table view file. See detailed TABLEVIEW document for more information

To connect to multiple databases through a single OpenLink client connection and/or to make use array type columns you must run the OpenLink provided "setup.p" utility. Please refer to the setup.p document for detailed information on the use of this script.

Configuring Progress Session Resources

You can control default behavior and progress server session resource allocation by entering standard progress session parameters in the "Server Options" field within the Admin Assistant's database agent configuration wizard dialogs or forms.

The following values are set for you by default at installation time and displayed as depicted below within the "Server Options" fields of the Admin Assistant Forms and Wizard dialogs.

-T /tmp -d mdy -TB 31 -TM 31

Database Agent Specific Issues

Progress database servers support sockets and shared memory based methods of Inter Process Communication (IPC), unfortunately the shared memory approach which is much faster than sockets and the preferred approach by many users bears a cost of version incompatibility. This implies that your OpenLink database agents need to be an exact version match with your backend Progress database server in order to successfully initiate shared memory based database sessions (note: these agents are built using the Progress Embedded SQL package).

Rebuilding Progress Database agents

To get around the issue explained above OpenLink provides a relinkable library and script file that enables you to build an OpenLink database agent that has an exact match to the version of Progress that you have installed. See the document on Relinking Progress Agents for details.

If shared memory based IPC isn't an issue for you then start your Progress server with the -S, -N, and -H options indicating the use of a sockets based Progress database server. This mode of operation is Progress version independent.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Progress database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Progress database on your Database Server machine using Progress database specific networking (Progress Client Networking) as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink Database independent Networking, while the communication between the Progress database agent and the Progress database server uses Progress Client Networking.

Configuration Steps:

Assuming you have an Progress Database Server machine called "mainserver2" that has a sockets based Progress Server process running, you would enter the following (assuming a TCP/IP based network):

1. Ensure that you have a usable connection to Progress using its native networking (Progress Client Networking) using the following remote database connection parameters:-S mainserver2 -H mainserver -N tcp .

2. Add the following values to the "Database Server Options" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same value into to the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

```
-S mainserver2 -H mainserver -N tcp
```

You can see illustrations of the various distributed client-server architectures supported by OpenLink database agent by clicking here.

Oracle Specific Configuration Issues

When configuring an Oracle database agent the critical configuration items are:

- Database Identification - this is an actual Oracle System Identifier (SID) e.g "ORCL", which identifies the actual Oracle environment that you want to be connected with. This is the value that you enter into the "Database Name" field of either your Admin Assistant's database agent configuration form or wizard dialog. If you choose to have database identification take at the client rather than server, you enter this value into the "Database Name" field or connection attribute when configuring your OpenLink client.

Oracle provides a number of environment variables for configuring database clients, the basic set required for successfully connecting your database agent to an Oracle database server are tabulated below:

[Environment ORACLE8]	**Default Rule Book Settings**	**Notes**
ORACLE_HOME=	/dbs/oracle8	The home directory for the Oracle installation.
ODBC_CATALOGS=	Y	Uncomment after loading the "odbccat7.sql" script.
MULTIPLEX_LDA=	5	Allow 5 OpenLink clients via a single database session

Database Agent Specific Settings

The "odbccat.sql" scripts explained:

These scripts exist for each version of Oracle supported, the files "odbccat6.sql", "odbccat7.sql", and "odbccat8.sql" representing Oracle versions 6 up to version 8 respectively. These scripts are to be applied to your Oracle instance to enable efficient and extended functionality between OpenLink and Oracle when handling ODBC, JDBC, UDBC, and OLE-DB catalog calls such as SQLForeignKeys() and SQLPrimaryKeys() functions. These functions have significant impact on the performance of your OpenLink clients.

To run these scripts you need to start the Oracle server manager (svrmgr or sqldba if you do this from the command line). Connect as internal and run the script by locating the relevant script file as you would any other Oracle SQL script file.

Rebuilding Oracle Database Agents

OpenLink provides a relinkable library and script files that enable you to rebuild your database agents as shared, as opposed to statically linked binaries, or for the purposes of getting a closer database implementation fit if your Oracle database environment is a more recent release than the actual version used by OpenLink to build the database agent installed on your system. Please read the Relinking Oracle Agents document for details on how to perform this task.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Oracle database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Oracle database on your Database Server machine using Oracle database specific networking (SQL*Net or Net8) as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink Database independent Networking, while the communication between the Oracle database agent and the Oracle database server uses Oracle SQL*Net or Net8.

Configuration Steps:

Assuming you have an Oracle Database Server machine called "mainserver2" that has an Oracle Listener process running, you would enter the following (presuming that your SQL*Net or Net8 alias for this Listener is also named "mainserver2"):

1. On you Application Server (the machine hosting your database agents) create a SQL*Net or Net8 Alias named "mainserver2" (for purposes of this example only).

2. Ensure that you have a usable connection to the remote Oracle database server using the SQL*Net or Net8 alias "mainserver2"

3. Add the following values to the "Server Options" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same value into to the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

   mainserver2

You can see illustrations of the various distributed client-server architectures supported by OpenLink database agent by clicking here.

Sybase Specific Configuration Issues

When configuring an Sybase database agent the critical configuration items are:

- Database Identification - this is an actual Sybase System Identifier (SID) e.g "ORCL", which identifies the actual Sybase environment that you want to be connected with. This is the value that you enter into the "Database Name" field of either your Admin Assistant's database agent configuration form or wizard dialog. If you choose to have database identification take at the client rather than server, you enter this value into the "Database Name" field or connection attribute when configuring your OpenLink client.

Sybase provides a number of environment variables for configuring database clients, the basic set required for successfully connecting your database agent to an Sybase database server are tabulated below:

[Environment SYBASE4]	**Default Rule Book Settings**	**Notes**
SYBASE=	/dbs/sybase4	Full path to the base directory for the Sybase installation.
DSQUERY=	SYBASE	Name of the Sybase Query Server that you are connecting to.
[Environment SYBASE10]	**Default Rule Book Settings**	**Notes**
SYBASE=	/dbs/sybase10	Full path to the base directory for the Sybase installation.
DSQUERY=	SYBASE	Name of the Sybase Query Server that you are connecting to.

Rebuilding Sybase Database Agents

OpenLink provides a relinkable library and script files that enable you to rebuild your database agents as shared, as opposed to statically linked binaries, or for the purposes of getting a closer database implementation fit, should your Sybase database environment be a more recent release than the actual version used by OpenLink to build the database agent installed on your system. Please read the Relinking OpenLink Database Agents document for details on how to perform this task.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Sybase database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Sybase database on your Database Server machine using Sybase database specific networking (Open Client) as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink Database independent Networking, while the communication between the Sybase database agent and the Sybase database server uses Sybase Open Client.

Configuration Steps:

Assuming you have an Sybase Database Server machine called "mainserver2" that has an Sybase Server named "mainserver2" up and running:

1. On you Application Server (the machine hosting your database agents) create an Open Client Database Connection Alias named "mainserver2" (for purposes of this example only).

2. Ensure that you have a usable connection to the remote Sybase database server using the Open Client Database alias "mainserver2"

3. Add the following values to the "Server Options" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same value into to the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

   mainserver2

   You may also enter the following values into the "Database Server Options" field: -s mainserver2

You can see illustrations of the various distributed client-server architectures supported by OpenLink database agent by clicking here.

Microsoft SQL Server Specific Configuration Issues

There aren't any specific environment variables that need to be configured the enable a basic database connection. All of the key setting are automatically resolved by the OpenLink installation program.

Application Server & 3-Tier Architecture Configuration

There may be situations in which you are unable to install your OpenLink Request Broker and Database Agents on the same machine as the one hosting your Microsoft SQL Server database server. Irrespective of the reasons that lead you to this scenario, it is possible to configure your OpenLink database agents hosted on your Application Server machine such that they connect to a remote Microsoft SQL Server database on your Database Server machine using Microsoft SQL Server database specific networking (NETLIB) as opposed to OpenLink's Database Independent Networking. The end result being a 3-tier distributed OpenLink architecture in which the communication between OpenLink clients and database agents use OpenLink Database independent Networking, while the communication between the Microsoft SQL Server database agent and the Microsoft SQL Server database server uses Microsoft SQL Server's NETLIB.

Configuration Steps:

Assuming you have an Microsoft SQL Server Database Server machine called "oplwinnt" that has a Microsoft SQL Server Server named "oplwinnt" up and running:

1. On you Application Server (the machine hosting your database agents) create a NETLIB Database Connection Alias named "oplwinnt" (for purposes of this example only).

2. Ensure that you have a usable connection to the remote Microsoft SQL Server database server using the Open Client Database alias "oplwinnt" (this the value you provide whenever you are prompted for a Server Name by native SQL Server utilities)

3. Add the following values to the "Server Options" field within the Admin Assistant Forms or Wizards used to configure your database agent. If you choose to set this value on the client simply enter the same value into to the "Database Name" Attribute associated with the configuration of your OpenLink client (see OpenLink ODBC or JDBC or UDBC client configuration for more details):

   oplwinnt

   You may also enter the following values into the "Database Server Options" field: -s oplwinnt

DB2 Specific Configuration Issues

[Environment DB2]	Default Rule Book Settings	Notes

DB2PATH=	/DB2	Full path to the base directory for the DB2 installation.
DB2INSTANCE=	DB2	Name of the instance you want to connect to. DB2 is the default DB2 instance name.

Agent Section

OpsysLogin=No; Validation of users is left to DB2

Database Agent default name: db2_sv

PostgresSQL Specific Configuration Issues

[Environment POSTGRES]	**Default Rule Book Settings**	**Notes**
;ODBC_CATALOGS=	Y	Uncomment after loading odbccat defs

Agent Section

OpSysLogin= Yes; Users are validated against the operating system.

Database Agent default name: pgr95_sv

Unify

[Environment UNIFY2000]	**Default Rule Book Settings**	**Notes**
UNIFY=	/dbs/unify/lib	Full path to the unify/lib directory of the Unify installation.
DBPATH=	/dbs/unify/database	Full path to the unify/database directory of the Unify installation. This will be the directory containing the Unify database files.
PATH=	/dbs/unify/bin	Full path to the unify/bin directory of the Unify installation.

Agent Section

OpsysLogin=Yes; Users are validated against the operating system.

Database Agent default name: uni_sv

Velocis

[Environment UNIFY2000]	**Default Rule Book Settings**	**Notes**
LD_LIBRARY_PATH =		

Agent Section

OpsysLogin=No; Users validated against the DBMS.

Database Agent default name: velo_sv

OpenLink ODBC Agent Installation & Configuration

Introduction:

The OpenLink ODBC Agent is an ODBC Proxy Service that facilitates the integration of non OpenLink ODBC Drivers in the Sophisticated OpenLink Multi Tier ODBC Architecture, thereby extending the benefit of this architecture beyond the scope of OpenLink ODBC drivers.

Typical Utilization:

A typical and very popular use of the OpenLink ODBC Agent is the exposure of tradition desktop database engines within your organization to your new Intranet or Internet based infrastructures, using a client-server distributed computing model. This implies that you can have multiple concurrent clients within your Intranet or remote Internet clients connecting to your Microsoft Access, DBASE, Fox PRO, Paradox database engines without any compromises in security and with astonishing performance.

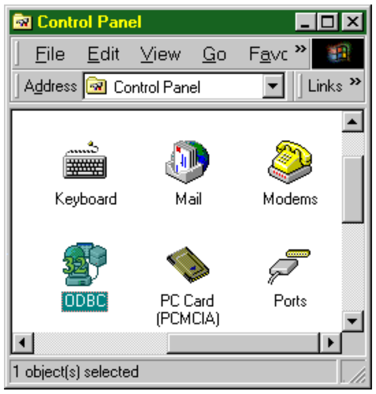
Installation:

After downloading the OpenLink Data Access Driver Suite for your chosen desktop operating system please perform the following steps:-

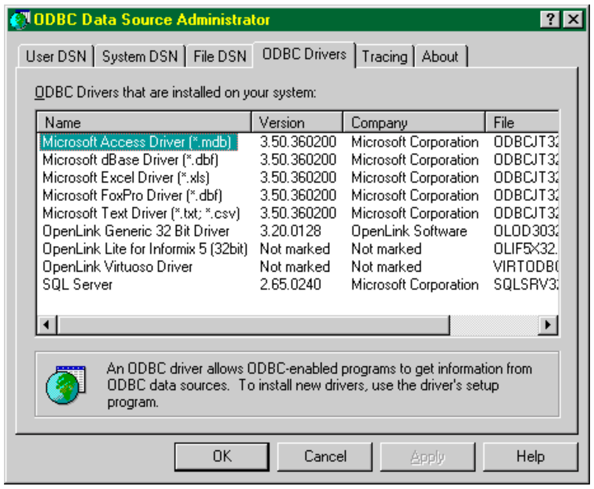
1. Move into your temporary installation directory
2. Extract the contents of the OpenLink ZIP archive into the directory in step 1
3. Double click on the program "setup.exe"
4. Follow the on-screen instructions

Post Installation & Pre Configuration Check List:

1. Verify that the ODBC Driver Manager exists on your system by opening up your desktop's control panel group



2. Verify the existence of an ODBC Driver for the desktop database engine that you will be connecting with (the example below shows the ODBC Driver for Microsoft Access from Microsoft Corporation and other installed ODBC Drivers)



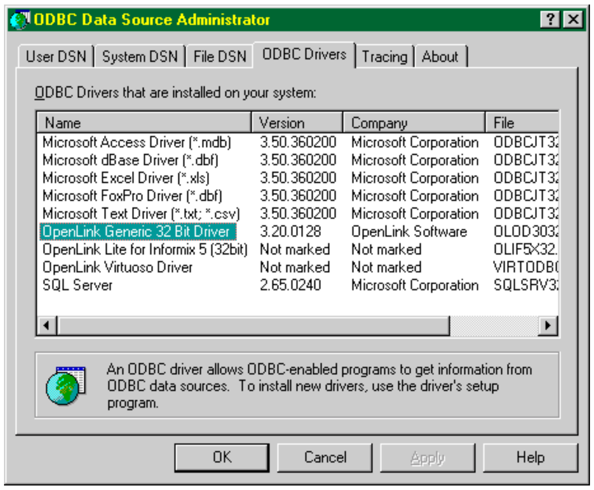
ODBC Drivers that are installed on your system:

Name	Version	Company	File
Microsoft Access Driver (*.mdb)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft dBase Driver (*.dbf)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft Excel Driver (*.xls)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft FoxPro Driver (*.dbf)	3.50.360200	Microsoft Corporation	ODBCJT3
Microsoft Text Driver (*.txt; *.csv)	3.50.360200	Microsoft Corporation	ODBCJT3
OpenLink Generic 32 Bit Driver	3.20.0128	OpenLink Software	OLOD303
OpenLink Lite for Informix 5 (32bit)	Not marked	Not marked	OLIF5X32.
OpenLink Virtuoso Driver	Not marked	Not marked	VIRTODB
SQL Server	2.65.0240	Microsoft Corporation	SQLSRV3

An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

3. Verify the existence of an ODBC System Data Source Name (DSN) for the database engine that you will be exposing via the OpenLink ODBC Agent, this ODBC DSN must be associated with the appropriate ODBC Driver for your desktop database

4. Verify the existence of an OpenLink ODBC Driver installation on your PC



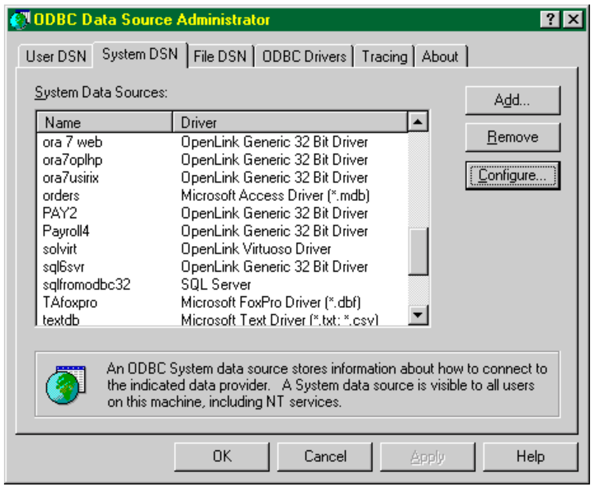
ODBC Drivers that are installed on your system:

Name	Version	Company	File
Microsoft Access Driver (*.mdb)	3.50.360200	Microsoft Corporation	ODBCJT32
Microsoft dBase Driver (*.dbf)	3.50.360200	Microsoft Corporation	ODBCJT32
Microsoft Excel Driver (*.xls)	3.50.360200	Microsoft Corporation	ODBCJT32
Microsoft FoxPro Driver (*.dbf)	3.50.360200	Microsoft Corporation	ODBCJT32
Microsoft Text Driver (*.txt; *.csv)	3.50.360200	Microsoft Corporation	ODBCJT32
OpenLink Generic 32 Bit Driver	3.20.0128	OpenLink Software	OLOD3032
OpenLink Lite for Informix 5 (32bit)	Not marked	Not marked	OLIF5X32.
OpenLink Virtuoso Driver	Not marked	Not marked	VIRTODB(
SQL Server	2.65.0240	Microsoft Corporation	SQLSRV32

An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

Configuring An OpenLink ODBC Agent Based ODBC Data Source Name (DSN)

1. Open up the ODBC Administrator within your desktop control panel, and then Click on the "Add" button to indicate that you want to add a new ODBC DSN to the current list of installed ODBC DSNs

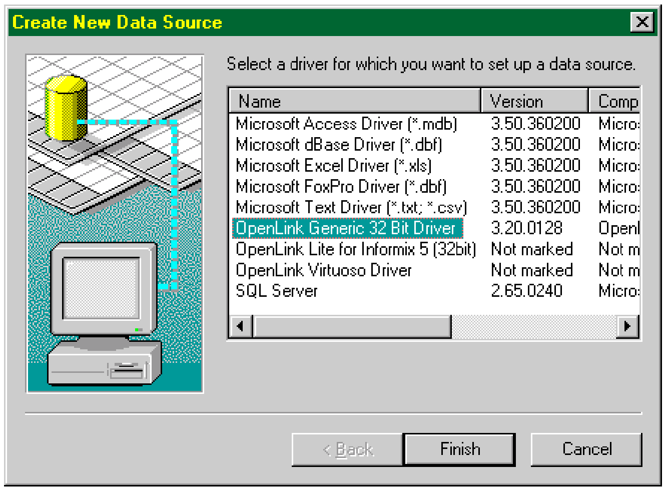


System Data Sources:

Name	Driver
ora 7 web	OpenLink Generic 32 Bit Driver
ora7oplhp	OpenLink Generic 32 Bit Driver
ora7usirix	OpenLink Generic 32 Bit Driver
orders	Microsoft Access Driver (*.mdb)
PAY2	OpenLink Generic 32 Bit Driver
Payroll4	OpenLink Generic 32 Bit Driver
solvirt	OpenLink Virtuoso Driver
sql6svr	OpenLink Generic 32 Bit Driver
sqlfromodbc32	SQL Server
TAfoxpro	Microsoft FoxPro Driver (*.dbf)
textdb	Microsoft Text Driver (*.txt; *.csv)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

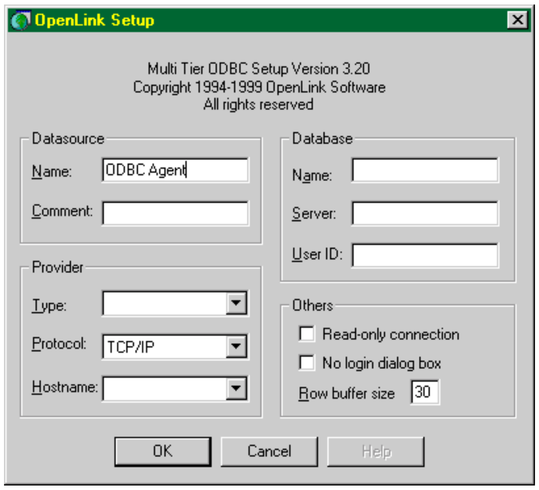
2. Click on the appropriate ODBC Driver that you will be associating this new ODBC DSN with, in this case the "OpenLink For Win32" or "OpenLink Generic 32 Bit" Driver (note: both of these are the same ODBC Driver identified differently for compatability reasons)


Create New Data Source

Select a driver for which you want to set up a data source.

Name	Version	Comp
Microsoft Access Driver (*.mdb)	3.50.360200	Micro:
Microsoft dBase Driver (*.dbf)	3.50.360200	Micro:
Microsoft Excel Driver (*.xls)	3.50.360200	Micro:
Microsoft FoxPro Driver (*.dbf)	3.50.360200	Micro:
Microsoft Text Driver (*.txt; *.csv)	3.50.360200	Micro:
OpenLink Generic 32 Bit Driver	3.20.0128	Openl
OpenLink Lite for Informix 5 (32bit)	Not marked	Not m
OpenLink Virtuoso Driver	Not marked	Not m
SQL Server	2.65.0240	Micro:

3. Choose an a Name for your OpenLink ODBC DSN and then type it into the "Name" field, the example below presumes the DSN is to be called "ODBC Agent"


OpenLink Setup

Multi Tier ODBC Setup Version 3.20
Copyright 1994-1999 OpenLink Software
All rights reserved

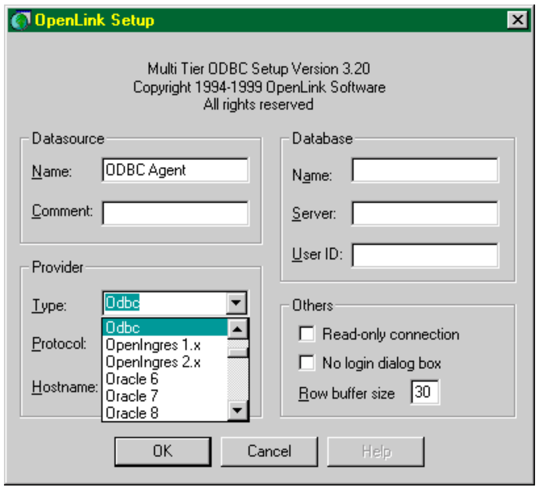
Datasource
Name: ODBC Agent
Comment:

Database
Name:
Server:
User ID:

Provider
Type:
Protocol: TCP/IP
Hostname:

Others
Read-only connection
No login dialog box
Row buffer size 30

4. Choose an OpenLink "Provider Type" of "ODBC" from the "Provider Type" listbox


OpenLink Setup

Multi Tier ODBC Setup Version 3.20
Copyright 1994-1999 OpenLink Software
All rights reserved

Datasource
Name: ODBC Agent
Comment:

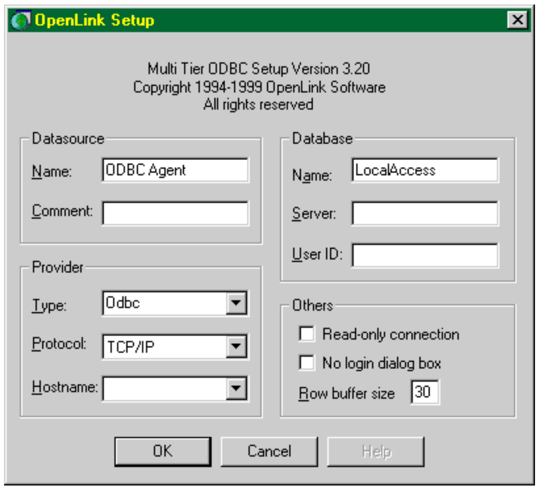
Database
Name:
Server:
User ID:

Provider
Type: Odbc
Protocol:
Hostname:

Odbc
OpenIngres 1.x
OpenIngres 2.x
Oracle 6
Oracle 7
Oracle 8

Others
Read-only connection
No login dialog box
Row buffer size 30

5. Type the name of the non OpenLink ODBC DSN that you would like to associate this OpenLink ODBC DSN with into the "Path" field. The example below presumes the existence of a non OpenLink ODBC DSN named "LocalAccess" that is bound to the Microsoft ODBC Driver for MS Access



Multi Tier ODBC Setup Version 3.20
Copyright 1994-1999 OpenLink Software
All rights reserved

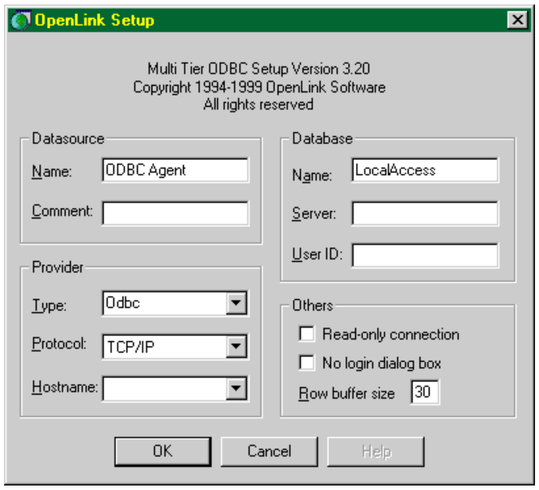
Datasource
Name: ODBC Agent
Comment:

Provider
Type: Odbc
Protocol: TCP/IP
Hostname:

Database
Name: LocalAccess
Server:
User ID:

Others
☐ Read-only connection
☐ No login dialog box
Row buffer size 30

6. Enter the name of the machine hosting the OpenLink ODBC agent, the example below presumes that the machine network alias for your desktop computer is "MyPC" (note: you can also use the machines actual IP address or even use the "localhost" account if you are connecting to a local as opposed to remote non OpenLink ODBC DSN)



Multi Tier ODBC Setup Version 3.20
Copyright 1994-1999 OpenLink Software
All rights reserved

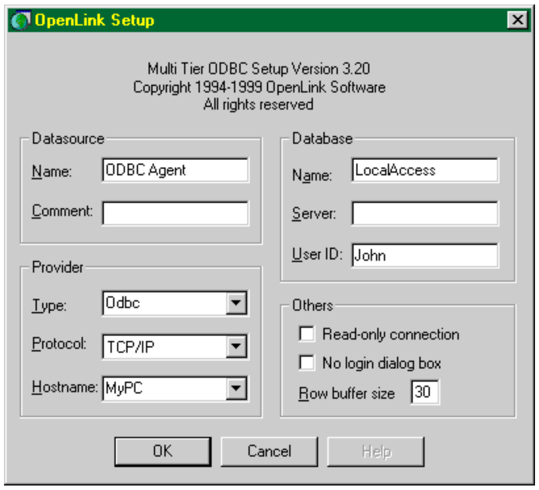
Datasource
Name: ODBC Agent
Comment:

Provider
Type: Odbc
Protocol: TCP/IP
Hostname:

Database
Name: LocalAccess
Server:
User ID:

Others
☐ Read-only connection
☐ No login dialog box
Row buffer size 30

7. If a username is required when connecting to your desktop database please enter the username value into the "User ID" field. The example below presumes a database user login of "John"



OpenLink Setup

Multi Tier ODBC Setup Version 3.20
Copyright 1994-1999 OpenLink Software
All rights reserved

Datasource
Name: ODBC Agent
Comment:

Provider
Type: Odbc
Protocol: TCP/IP
Hostname: MyPC

Database
Name: LocalAccess
Server:
User ID: John

Others
☐ Read-only connection
☐ No login dialog box
Row buffer size 30

8. Optimize record retrieval throughput by setting the "Row Buffer Size" value to a number representing how many records you would like your OpenLink ODBC Driver to retrieve during a single iteration of an ODBC Fetch call and Network hop (when connecting to remote ODBC DSNs), the example above presumes 30 records

9. Click on the "OK" button to complete the creation of your new OpenLink ODBC DSN

Making A Test Connection To Your OpenLink ODBC Agent Based ODBC DSN

To verify that your installation and configuration is ready for use, please follow the steps below in order to make a test connection to the OpenLink ODBC DSN that you have just created:-

1. Start the OpenLink Request Broker in debug mode, this can be done from a DOS shell by executing the command : oplrqb -dv or from your Services Panel (note you must change the startup mode to manual to enable the OpenLink Request Broker run in Debug Mode)

2. Locate the program "VBDemo" or "C++ Demo" situated within the "OpenLink Data Access Drivers" group on your desktop (Windows start menu item)

3. Attempt to make a connection to the OpenLink ODBC DSN

4. If step 3 is successful and you see data exchanged between your ODBC Client and your OpenLink ODBC DSN, exit the ODBC application, and the shutdown and restart the Broker without the Debug Mode options using the command: oplrqb -v. If step 3 is unsuccessful repeat step 3 and then capture the Request Broker output and proceed to instigating contact with OpenLink Technical support via the OpenLink Support Page

5. Establish connection between your own ODBC Applications and the OpenLink ODBC DSN created in step 3

6. Shutdown the Request Broker using the command: oplshut -f

Adding An OpenLink ODBC Agent To A Pre OpenLink v1.5.5 or v3.0 System

1. Place the ODBC Agent executable in the openlink/bin sub-directory

2. Edit the OpenLink "Session Rules" Book (the file oplrqb.ini) using a text editor

3. Insert a new OpenLink Session Mapping rule to the top of the "[Mapping Rules]" section of the Rule Book in the manner

depicted below:

odbc:*:*:*:*:*:* = accept odbc_agent

4. Then create a new OpenLink Agent section as follows:

   [odbc_agent]

   Program = odbc_sv

   Note: Program equals must be set to the exact file name. (For NT this would be odbc_sv.exe)

5. Save the file

6. Shutdown and restart your OpenLink Request Broker.

OpenLink Proxy Agent

Introduction

An OpenLink Proxy agent is a specialized Agent that acts on behalf of another remotely or locally situated OpenLink Database Agent. This Agent format is typically used in 3-Tier Internet based environments in which you place an Agent on an external machine (typically running your Web Server) and then have it masquerade for an actual database agent behind your organization's firewall.

A Proxy Agent can also be used in conjunction with the Session Rules Book for centralized configuration and control of all of your OpenLink Clients, by controlling the configuration of all of the OpenLink Session Elements on one or more server machines.

Like other Proxy services, an OpenLink Client connects to the Proxy Agent instead of to the actual service; the Proxy Agent then connects to the actual service which presumably lies on a machine that shouldn't normally be accessible from outside the network. With this setup, it is possible to grant selective access to databases that are otherwise not accessible from the Internet; this greatly enhances the functionality of data access standards like JDBC, ODBC, OLE-DB.

Installation

The OpenLink Proxy Agent is automatically installed with your Request Broker on any platform.

You only have to install a Request Broker installation archive on the machine that is to act as a host for the OpenLink Proxy agent (typically the middle tier machine in a three 3-tier architecture). You then install another Request Broker archive and relevant Database Agents archives on the machine(s) hosting the backend database engine(s) that you are connecting to via an OpenLink Client.

Configuration

You configure the Proxy Agent like all other OpenLink Agents using the Admin Assistant. The process is broken into two parts, the first part involves creating a Proxy Agent Template, the second part involves a Session Rules that conditionally associates OpenLink Clients with the Proxy Agent Template that you have created.

The configuration guide that follows presumes that we are creating a Proxy on a middle-tier server for the OpenLink Virtuoso Database engine such that any OpenLink Client connection (ODBC, JDBC, UDBC, or OLE-DB) to this Server ends up being connected to Virtuoso. The steps that follow guide you through this process.

Creating Proxy Agent Initialization Template

1. Start Request Broker your middle-tier server machine

2. Start a Web Browser session

3. Enter the following URL into your browser:

   If you started the Request Broker on your local machine enter:

   http://localhost:8000/

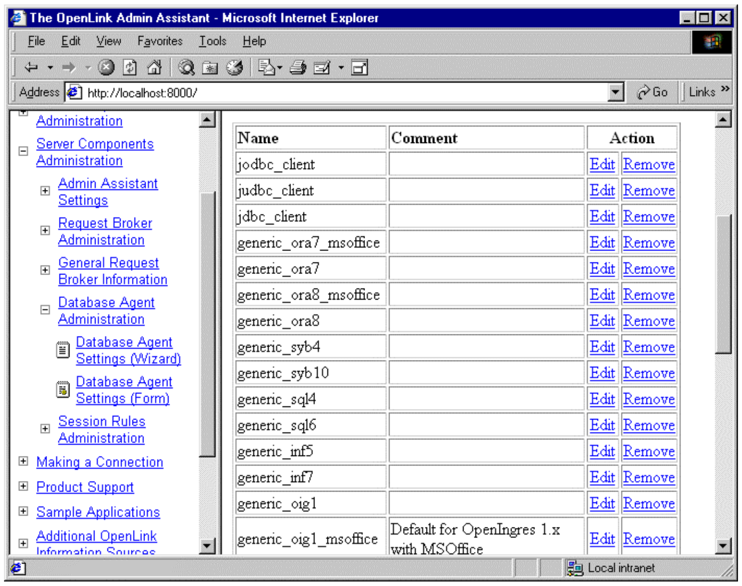
   (assuming you accepted port 8000 as the Admin Assistant port number at installation time).

   If the Request Broker in on another machine enter:

   http://<hostname or IP address>:8000

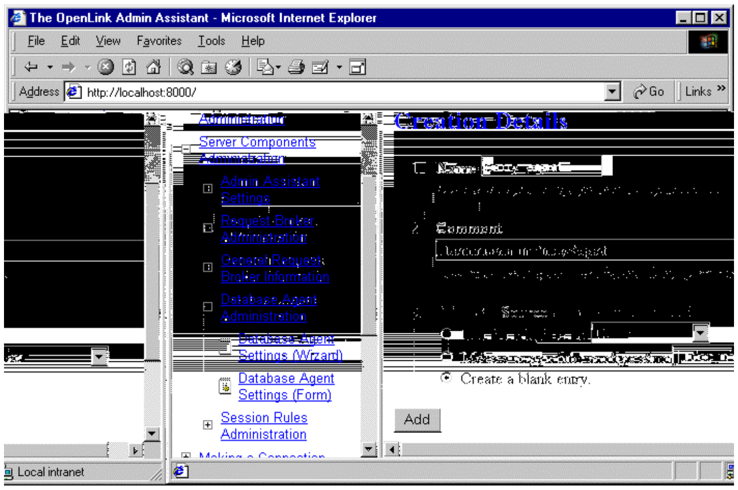
   (assuming you accepted port 8000 as the Admin Assistant port number at installation time).

4. Navigate the Admin Assistant menu tree as follows: OpenLink Database Agent(s) Settings-->Database Agent(s) Settings (Form).


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Go Links
Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Local intranet

Name	Comment	Action
jodbc_client		Edit Remove
judbc_client		Edit Remove
jdbc_client		Edit Remove
generic_ora7_msoffice		Edit Remove
generic_ora7		Edit Remove
generic_ora8_msoffice		Edit Remove
generic_ora8		Edit Remove
generic_syb4		Edit Remove
generic_syb10		Edit Remove
generic_sql4		Edit Remove
generic_sql6		Edit Remove
generic_inf5		Edit Remove
generic_inf7		Edit Remove
generic_oig1		Edit Remove
generic_oig1_msoffice	Default for OpenIngres 1.x with MSOffice	Edit Remove

5. Scroll to the bottom of the Agent Templates listing page and then click on the "Add" button, this opens up a default agent initialization template page, enter a Name and Description for your new Agent Initialization Template, then select the "create blank entry" radio button and then click the "Add" hyperlink. Reinitialize the Broker when prompted.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Go Links
Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Making a Connection
Creation Details
Name
Comment
Create a blank entry.
Add
Local intranet

6. Navigate the Admin Assistant menu tree as follows: OpenLink Database Agent(s) Settings-->Database Agent(s) Settings

(Form). Then locate the new Agent Template created in the previous step. Now click on the "Edit" hyperlink.

Enter values into the following fields representing key OpenLink Session Elements:

"User Name" - Leave empty (this is handled on the server using the Virtuoso Agent initialization Template)

"Password" - ditto

"Database Name" - ditto

"Read Only" - ditto

"Connect Options" - ditto

"Server Options" - ditto

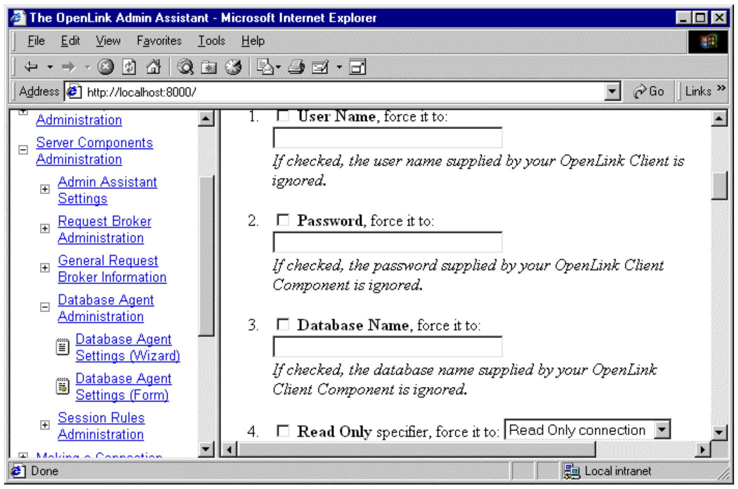
"Server Type" - Virtuoso (you enter an valid OpenLink Domain values here, e.g Oracle 8, Informix 7, Progress 83B etc.)

"Host Name" - enter IP address or network alias of database server machine. this examples presumes the IP address of the database server 123.123.123.100

"Executable Name" - enter "proxy_sv" (Linux or UNIX) or "proxy_sv.exe" (for Windows)

In the "Client-Server Mapping Process & Control" section of this page select the "Conditionally" radio button and then hatch the "When originating from same machine" checkbox. This ensures that each new OpenLink client machine has a distinct proxy agent process servicing all the ODBC, JDBC, UDBC, OLE-DB clients on that machine thereby restricting the number of new proxy agents processes initialized.

Click on the "Update" button and the reinitialize the Request Broker.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Go
Links
Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Making a Connection
Done
Local intranet

1. ☐ **User Name**, force it to:

If checked, the user name supplied by your OpenLink Client is ignored.

2. ☐ **Password**, force it to:

If checked, the password supplied by your OpenLink Client Component is ignored.

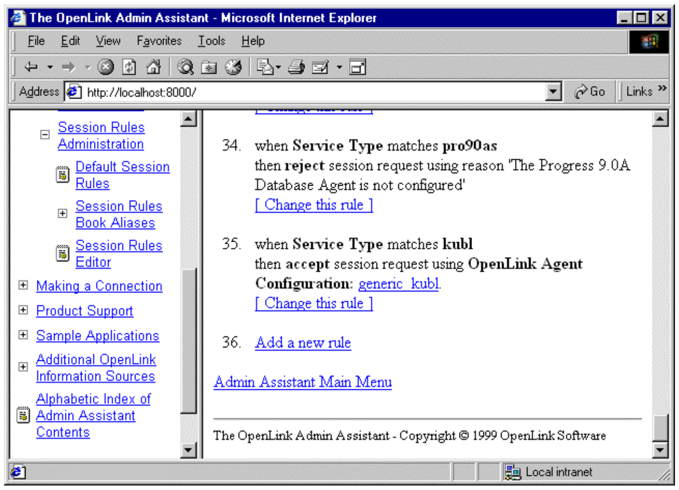
3. ☐ **Database Name**, force it to:

If checked, the database name supplied by your OpenLink Client Component is ignored.

4. ☐ **Read Only** specifier, force it to: Read Only connection

Creating Session Rule That Maps Connections to Proxy Agent's Template

1. Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Editor" menu path which brings you to a screen identical to the one depicted below, this presents you with a list of existing session rules (all OpenLink installations come with a set of pre-configured session rules). Click on the "Add new rule" hyperlink to open up the session rule creation page.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Session Rules Editor
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
Local intranet

34. when **Service Type** matches **pro90as**
then **reject** session request using reason 'The Progress 9.0A Database Agent is not configured'
[Change this rule]

35. when **Service Type** matches **kubl**
then **accept** session request using **OpenLink Agent Configuration:** generic_kubl.
[Change this rule]

36. Add a new rule

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

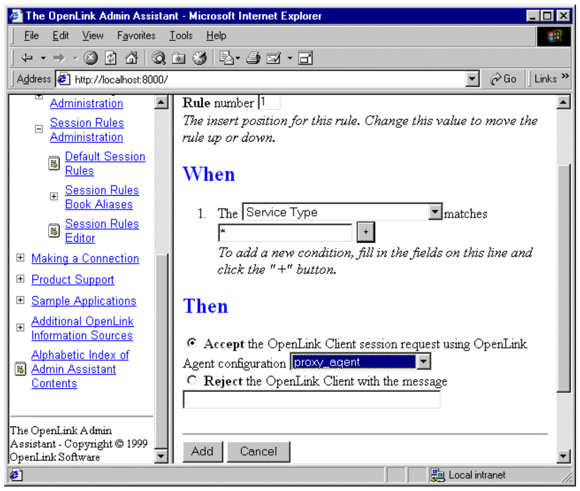
2. Create a new session rule by doing the following:

   Set the Rule Number field to 1 (this means that this rule gets evaluated before others)

   Leave the default value of "*" in the Server Type field this ensure that this rule applies to any Domain Type.

   Pick the "proxy_agent" initialization template from the agent initialization template list box used by the "Then" processing instruction to determine how calls associated with this rule are to be handled.

   Click on the "Add" button to save your new rule to the rule book. Then reinitialize the Request Broker.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Session Rules Editor
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
Add
Cancel
Local intranet

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

Rule number 1
The insert position for this rule. Change this value to move the rule up or down.

When

1. The Service Type matches
*
To add a new condition, fill in the fields on this line and click the "+" button.

Then

Accept the OpenLink Client session request using OpenLink Agent configuration proxy_agent
Reject the OpenLink Client with the message

3. Create an OpenLink ODBC, JDBC, UDBC or OLE-DB client session with the Domain Type attribute set to Proxy and the Host attribute set to your middle-tier server.

If your database server is behind a firewall you need to perform the following additional additional steps:

1. Enable UDP support, and then make port 60001 available, this is the port number used by the OpenLink Request Broker. Since we are now connecting to a database server running the Request Broker that resides behind your firewall we need to open up this port.

2. Start the Request Broker on the Database Server

3. Start a Web Browser session and then initialize the Admin Assistant running on the Database Server machine by entering the following URL:

   http://<hostname or IP address of Database Server machine>:8000 (presuming you took the default number of 8000 for the Web Assistant at install time on the Database Server).

4. Navigate the Admin Assistant menu tree to: Server Components Administration--> Request Broker Administration-->Edit Request Brokers Parameters (Form).

   Locate item number 4 on the form which reads "Only use ports in the range...", enter a range of TCP port numbers that you have enabled within your firewall software. The Broker will then automatically starts the Virtuoso agent (or other database agents depending on your settings) on the first available port in the range.

OpenLink Session Rules Administration & Configuration Guide

Introduction

One of the most important features and benefits of your OpenLink Multi Data Access Drivers, is the ability to configure and control your entire OpenLink infrastructure from a central point using "OpenLink Session Rules". These session rules are stored and maintained in a text based repository called the "Session Rules Book" (the file "oplrqb.ini").

You administer these rules from your web browser using the OpenLink Admin Assistant. The rules that you create are enforced by the OpenLink Request Broker, giving you phenomenal control over your distributed computing infrastructure.

OpenLink Session Rules Concepts

Session rules are declarative in nature and template driven. You build a template that determines how one or more OpenLink Client components are going to interact with a particular instance of an OpenLink Server component (Database Agent or Service Provider Agent). Session rules also determine what OpenLink Server is instantiated and how it is to be instantiated for a particular OpenLink Client.

The basic Session Rule unit is an OpenLink Connection Attribute, each representing a key aspect of an OpenLink Client's connection to an OpenLink Agent Session. There are six OpenLink Connection Attributes: Domain, User, OpSys, Machine, Application and Mode.

OpenLink Connection Attributes

There are two types of OpenLink Connection Attributes, these are "User Configurable" and "Non User Configurable" Connection Attributes.

User Configurable Connection Attributes

These attributes are configurable by the OpenLink Client user. When using ODBC this is handled via the ODBC Administrator or via the Admin Assistant ODBC Data Source Name Configuration Menu (this also applies to UDBC Clients). JDBC handles this through the use of JDBC URLs and in the case of JDBC 2.0 via Data Source Names. OLE-DB handles this via a Connection String Building Wizard.

Domain Attribute

Identifies the OpenLink Agent to which all OpenLink Connection Attributes apply.

This attribute describes a logical reference for a database type, OpenLink agent type, or anything else you would like to use as a logical identifier for all the other OpenLink Connection Attributes.

This attribute is also referred to as the "Server Type" or "SVT" attribute of an OpenLink Client connect string. It is also referred to in older product documentation as the OpenLink "Provider Type".

User Attribute

Identifies the User making use of an OpenLink Client.

Database Attribute

Identifies a specific database name within a database environment, e.g ORCL within Oracle, "stores7" within Informix, "pubs" with Sybase or Microsoft SQL Server etc. The values associated with this Connection Attribute aren't overridden by Aliases, instead they are overridden by the "Database" attribute within an Agent Configuration template.

ServerOpts (Database Sever Environment Options) Attribute

Identifies a set of database environment initialization parameters, currently this attribute only applies to Progress environments. It is used to set self serving client initialization parameters such as: -TB, -TB, -q, -D mdy etc. The values associated with this Connection Attribute aren't overridden by Aliases, instead they are overridden by the "SeverOptions" attribute within an Agent Configuration template.

ConnectOpts (Database Server Connection Options) Attribute

Identifies a set of database server connection parameters used to initiate a connection with a backend database server process. This is how an OpenLink Database Agent makes a connection with a backend database server using a particular database vendors networking middleware (e.g. Net8 or SQL*Net for Oracle, I-Connect or I-Net for Informix, Progress Client Networking for Progress, Open Client for Sybase etc.) . The values associated with this Connection Attribute aren't overridden by Aliases, instead they are overridden by the "ConnectOptions" attribute within an Agent Configuration template.

Non User Configurable Connection Attributes

These attributes are environmental in nature and derived automatically by an OpenLink Client.

OpSys (Operating System) attribute

Identifies the Client operating system from which the OpenLink Client is being executed.

Machine attribute

Identifies the Network Alias or IP address of the machine or device from the OpenLink Client is being executed.

Application attribute

Identifies the ODBC, JDBC, UDBC, OLE-DB Client Application from which the OpenLink Client is being executed.

Mode attribute

Identifies the session Mode required by an OpenLink Client, this may be Read-Only or Read-Write.

OpenLink Session Templates

Connection attributes are conditionally post processed during the initialization of session between an OpenLink Client and and OpenLink Agent, the rule book consists of a number of templates that play different roles during this process. The rule book is made up of the following templates: Session Aliases, Mapping Rules, and Agent Configuration

Session Aliases Templates

These are rule book templates used for post processing OpenLink Connection Attribute values prior to Mapping Rules evaluation. This is the facility used by the Request Broker for overriding Connection Attributes from OpenLink Clients with values on configured an OpenLink server.

Mapping Rules Templates

These are rule book templates used to determine which OpenLink Agents are instantiated in line with an OpenLink Client's session request.

OpenLink Agent Configuration Templates

Templates used for setting key OpenLink Agent Configuration parameters. See the OpenLink Agent Administration section for detailed information.

Session Rules Execution Process

When an OpenLink Client makes contact with an OpenLink Request Broker a series of events occur culminating in the identification and execution of a session rule. The step as follow guide you through this process:

1. Request Broker receives session request from one or more OpenLink Clients (Drivers for ODBC, UDBC, JDBC, OLE-DB).
2. Request Broker parses the session request data stream received from the relevant OpenLink Client isolating each OpenLink Connection Attribute type and associated attribute values.
3. Request Broker then performs a regular expression search through the rule book looking for Session Aliases that match the parsed OpenLink Connection Attributes.
4. For each OpenLink Connection Attribute that the Request Broker find a matching Session Alias it determines if the Alias has a non NULL assigned value, if this is true the OpenLink Connection Attribute values are reassigned to those of the matching Session Alias, otherwise they retain their existing values.
5. Request Broker then performs a regular expression search using a combination of all the OpenLink Connection Attributes across the session rules books "Mapping Rules" template.
6. The Request Broker scans through each "Mapping Rule" in ascending order, If it finds a "Mapping Rule" match it then applies the matching rule to the appropriate Client Session request, otherwise it reports an error condition back to the OpenLink client.
7. When a "Mapping Rules" occurs the Request Broker evaluates the "Mapping Rule". This evaluation results in the Acceptance or Rejection of an OpenLink Client request.
8. OpenLink Client request acceptance results in the OpenLink Clients session request being associated with an OpenLink Agent template, this template then applies all of its attribute Attributes to the OpenLink Agent Configuration process.
9. Session Rules rejection results in a user/administrator definable error message being relayed back to the OpenLink Client.
10. OpenLink Session is fully initialized. This means that the OpenLink Client and Server (agents) components are now linked and operating in a connected state.
11. OpenLink Agent evaluates subsequent OpenLink Client session requests to see if they are in line with the ReUse attribute of its Agent Configuration template.

Creating Custom Aliases For Use By OpenLink Data Access Clients

Understanding how to maintain Session Aliases is a critical part of understanding how to create session rules. You can create, modify, edit, and delete Session Aliases in two ways, you either use the Admin Assistant's GUI interface or manually edit the session rule book using a text editor (only recommended for advanced users).

The steps that follow guide you through the Session Alias management process using the Admin Assistant's GUI. Before performing any of these steps you need to start the Admin Assistant, this is done by following the steps below:

1. Start Request Broker

2. Start a Web Browser session

3. Enter the following URL into your browser:

If you started the Request Broker on your local machine enter:

http://localhost:8000/

(assuming you accepted port 8000 as the Admin Assistant port number at installation time).

If the Request Broker in on another machine enter:

http://<hostname or IP address>:8000

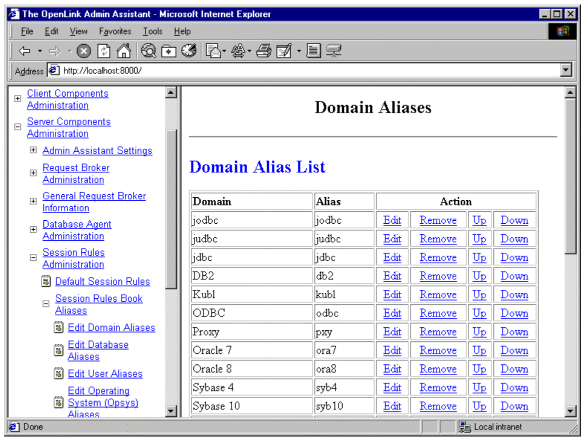
(assuming you accepted port 8000 as the Admin Assistant port number at installation time).

Domain Aliases

The steps that follow show you how to manage Domain Aliases:

1. Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Book Aliases"->"Edit Domain Aliases" menu tree which brings you to a screen identical to the one depicted below.

   The Admin Assistant presents you with a list of pre-configured Domain Aliases, click on the "Edit" hyperlink to change settings, the "Remove" hyperlink to delete an Alias, and the "Add" button at the bottom of the screen to create a new Domain Alias.

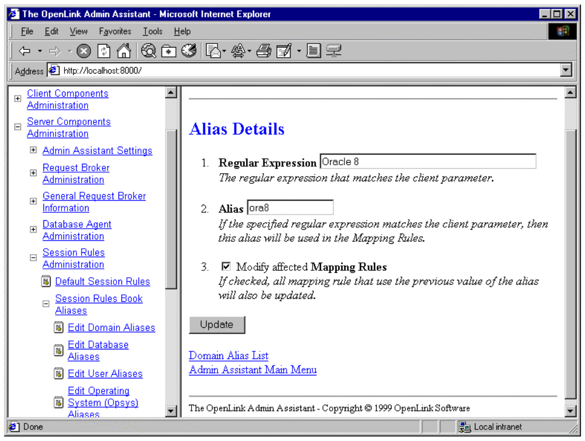

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Done
Local intranet

Domain Aliases

Domain Alias List

Domain	Alias	Action			
jodbc	jodbc	Edit	Remove	Up	Down
judbc	judbc	Edit	Remove	Up	Down
jdbc	jdbc	Edit	Remove	Up	Down
DB2	db2	Edit	Remove	Up	Down
Kubl	kubl	Edit	Remove	Up	Down
ODBC	odbc	Edit	Remove	Up	Down
Proxy	pxy	Edit	Remove	Up	Down
Oracle 7	ora7	Edit	Remove	Up	Down
Oracle 8	ora8	Edit	Remove	Up	Down
Sybase 4	syb4	Edit	Remove	Up	Down
Sybase 10	syb10	Edit	Remove	Up	Down

2. The example below assumes that you are modifying a Domain Alias named "ora8" with the attribute values "Oracle 8". This implies that you want to take note of Domain Connection Attributes from an OpenLink Client that start with the value "Oracle 8" evaluation against a mapping rules template. Once you have completed your input click the "Update" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Done
Local intranet

Alias Details

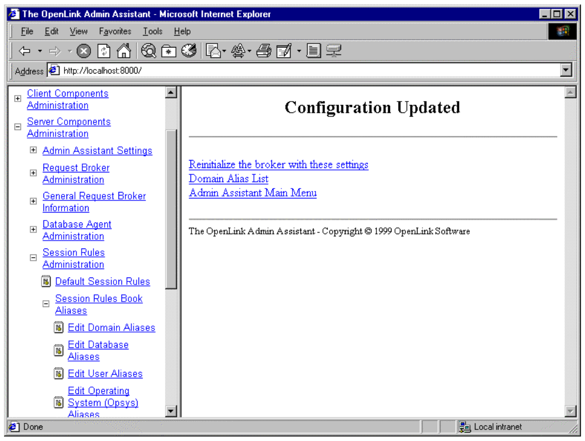
1. **Regular Expression** Oracle 8
   The regular expression that matches the client parameter.
2. **Alias** ora8
   If the specified regular expression matches the client parameter, then this alias will be used in the Mapping Rules.
3. ☑ Modify affected **Mapping Rules**
   If checked, all mapping rule that use the previous value of the alias will also be updated.

Update

Domain Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

3. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink. See screen shot below:

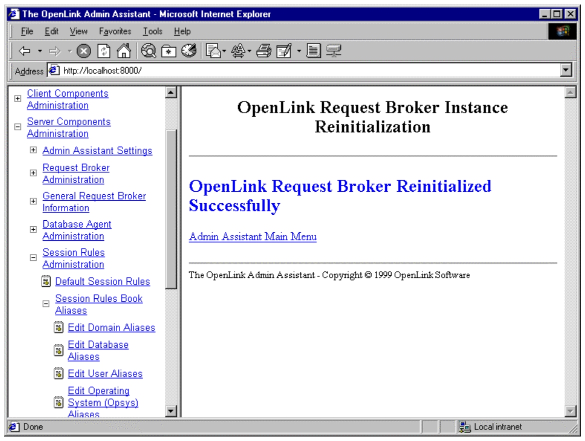

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Done
Local intranet

Configuration Updated

Reinitialize the broker with these settings
Domain Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

4. Admin Assistant confirms commitment of your changes to the rule book.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Done
Local intranet

OpenLink Request Broker Instance Reinitialization

OpenLink Request Broker Reinitialized Successfully

Admin Assistant Main Menu

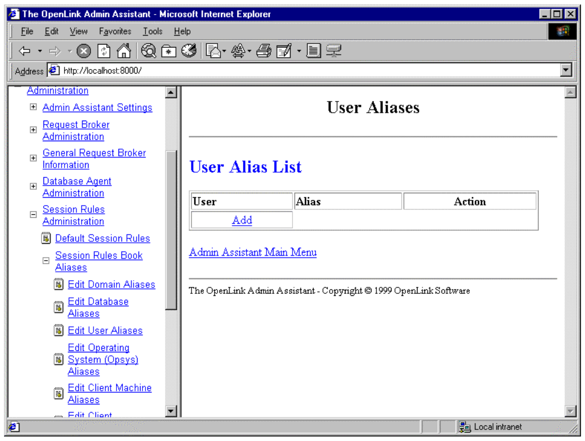
The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

User Aliases

The steps that follow show you how to manage User Aliases using the Admin Assistant:

1. Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Book Aliases"->"Edit User Aliases" menu tree which brings you to a screen similar to the one depicted below.

   The Admin Assistant presents you with a list of pre-configured User Aliases if any exist, if this is your first time there will be not items listed. Click on the "Add" hyperlink to create a new User Alias.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Local intranet

User Aliases

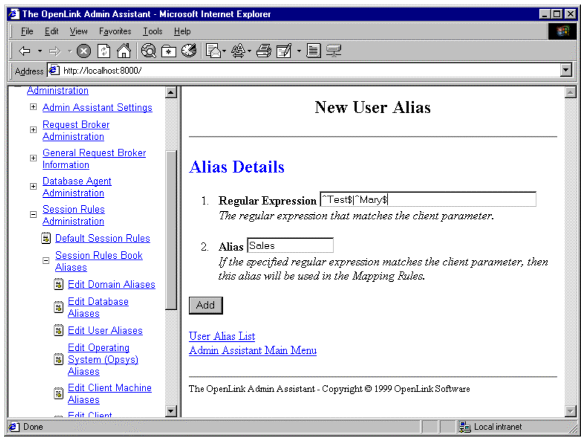
User Alias List

User	Alias	Action
Add		

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. The example below assumes that you are creating an User Alias named "Sales" with attribute values of "Test" or "Mary", the use of the caret symbol (character "^") and the Dollar symbol indicate the start and end of regular expression evaluation values respectively. The Pipe symbol (character "|") indicates an OR condition. Thus the entry depicted below implies that you want to take note of User Connection Attributes from an OpenLink Client that hold the values "Test" or "Mary" for evaluation against a mapping rules template. Once you have completed your input click the "Add" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

New User Alias

Alias Details

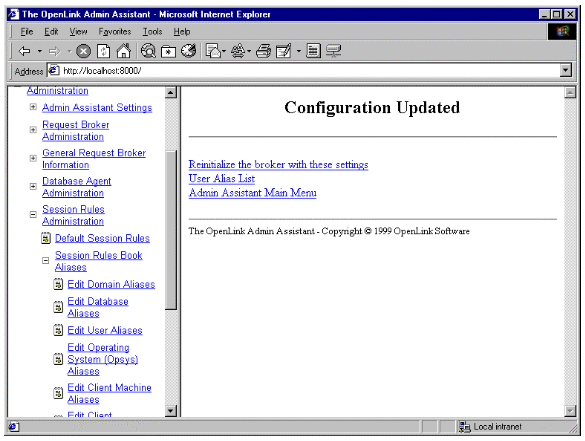
1. **Regular Expression** ^Test$|^Mary$
   The regular expression that matches the client parameter.
2. **Alias** Sales
   If the specified regular expression matches the client parameter, then this alias will be used in the Mapping Rules.

Add

User Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

3. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink. See screen shot below:

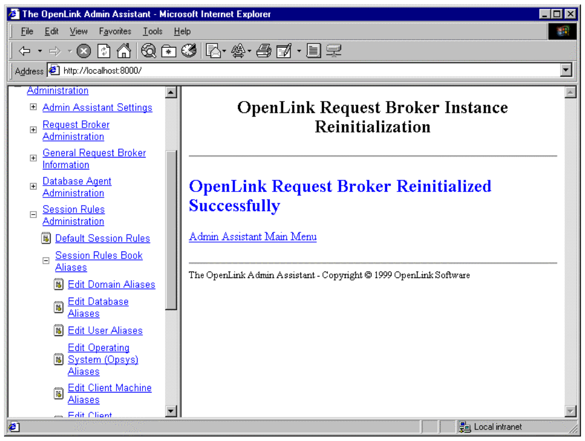

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Local intranet

Configuration Updated

Reinitialize the broker with these settings
User Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

Admin Assistant confirms commitment of your changes to the rule book.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Local intranet

OpenLink Request Broker Instance Reinitialization

OpenLink Request Broker Reinitialized Successfully

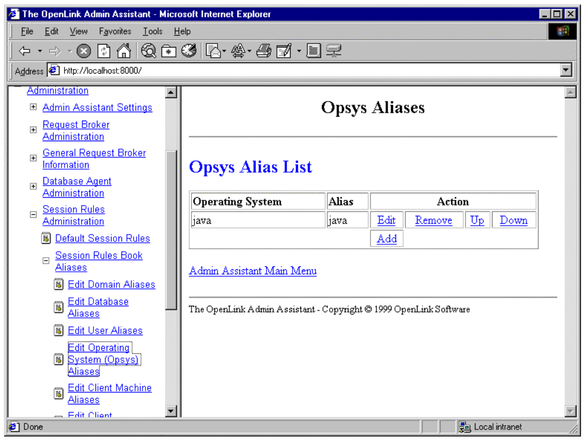
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

OpSys Aliases (Operating System Aliases)

1. The steps that follow show you how to manage OpSys Aliases using the Admin Assistant:
   Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Book Aliases"->"Edit Operating System Aliases" menu tree which brings you to a screen identical to the one depicted below.

   The Admin Assistant presents you with a list of pre-configured OpSys Aliases if any exist, if this is your first time no items will be listed. Click on the "Add" hyperlink to create a new OpSys Alias.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

Opsys Aliases

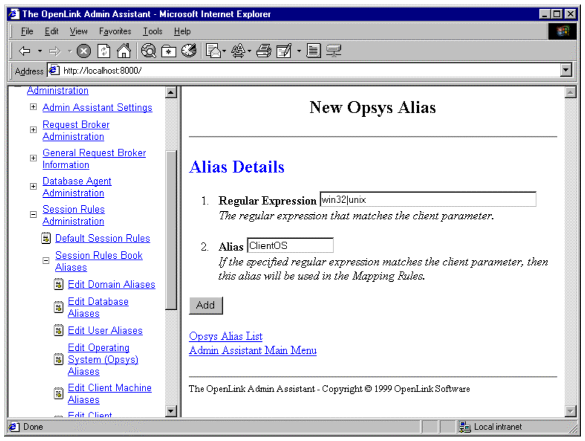
Opsys Alias List

Operating System	Alias	Action			
java	java	Edit	Remove	Up	Down
		Add			

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. The example below assumes that you are creating an OpSys Alias named "ClientOS" with attribute values of "win32" or "unix". The Pipe symbol (character "|") indicates an OR condition. Thus the entry depicted below implies that you want to take note of OpSys Connection Attributes from an OpenLink Client that start with the values "win32" or "unix" for evaluation against a mapping rules template. Once you have completed your input click the "Add" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Add
Done
Local intranet

New Opsys Alias

Alias Details

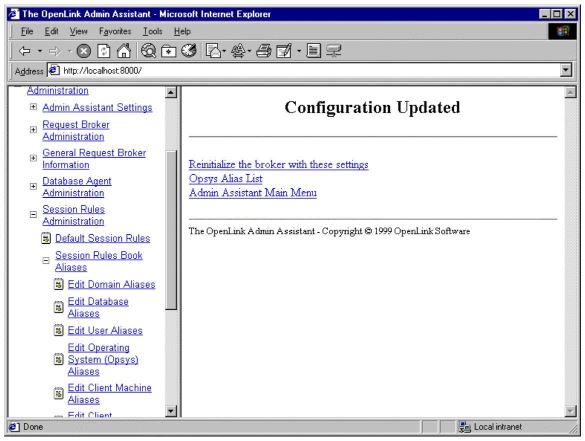
1. **Regular Expression** win32|unix
   The regular expression that matches the client parameter.

2. **Alias** ClientOS
   If the specified regular expression matches the client parameter, then this alias will be used in the Mapping Rules.

Opsys Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

3. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink. See screen shot below:

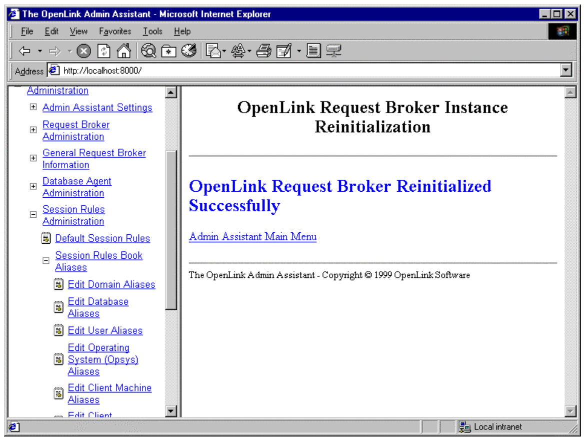

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

Configuration Updated

Reinitialize the broker with these settings
Opsys Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

4. Admin Assistant confirms commitment of your changes to the rule book.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Local intranet

OpenLink Request Broker Instance Reinitialization

OpenLink Request Broker Reinitialized Successfully

Admin Assistant Main Menu

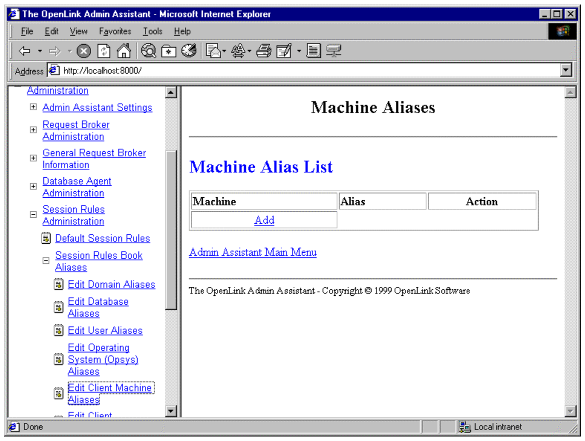
The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

Machine Aliases

The steps that follow show you how to manage User Aliases using the Admin Assistant:

1. Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Book Aliases"->"Edit Operating Machine Aliases" menu tree which brings you to a screen identical to the one depicted below.

   The Admin Assistant presents you with a list of pre-configured OpSys Aliases if any exist, if this is your first time no items will be listed. Click on the "Add" hyperlink to create a new Machine Alias.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

Machine Aliases

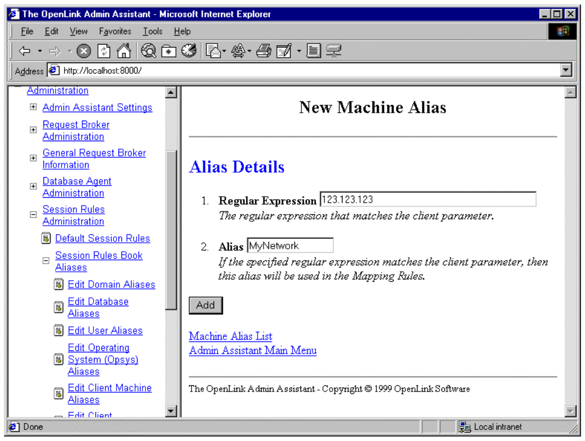
Machine Alias List

Machine	Alias	Action
Add		

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. The example below assumes that you are creating an Machine Alias named "MyNetwork" with an attribute value of "123.123.123". This implies that you want to take note of Machine Connection Attributes from an OpenLink Client that start with the value "123.123.123" (you would do this to identify the Network portion of the client machines IP address). Once you have completed your input click the "Add" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

New Machine Alias

Alias Details

1. **Regular Expression** 123.123.123
The regular expression that matches the client parameter.

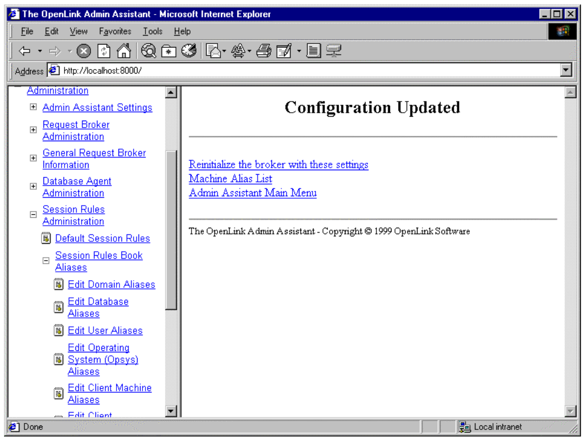
2. **Alias** MyNetwork
If the specified regular expression matches the client parameter, then this alias will be used in the Mapping Rules.

Add

Machine Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

3. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink. See screen shot below:

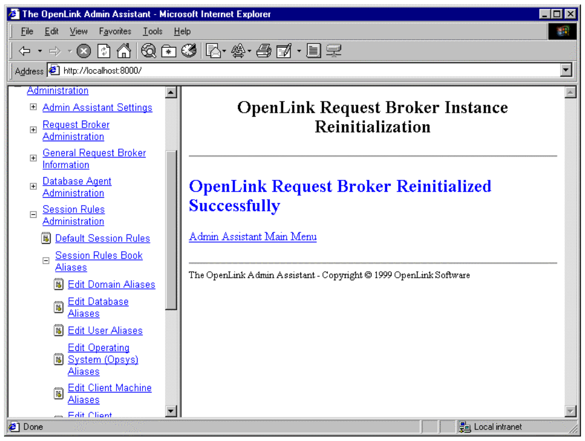

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

Configuration Updated

Reinitialize the broker with these settings
Machine Alias List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

4. Admin Assistant confirms commitment of your changes to the rule book.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

OpenLink Request Broker Instance Reinitialization

OpenLink Request Broker Reinitialized Successfully

Admin Assistant Main Menu

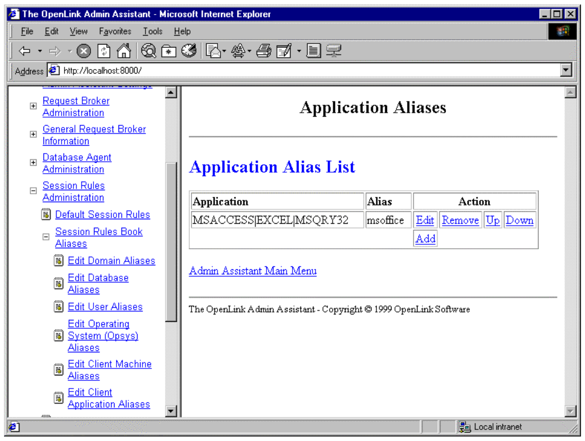
The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

Application Aliases

The steps that follow show you how to manage User Aliases using the Admin Assistant:

1. Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Book Aliases"->"Edit Application Aliases" menu tree which brings you to a screen identical to the one depicted below.

   The Admin Assistant presents you with a list of pre-configured Application Aliases, click on the "Edit" hyperlink to change settings, the "Remove" hyperlink to delete an Alias, and the "Add" button at the bottom of the screen to create a new Application Alias.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Local intranet

Application Aliases

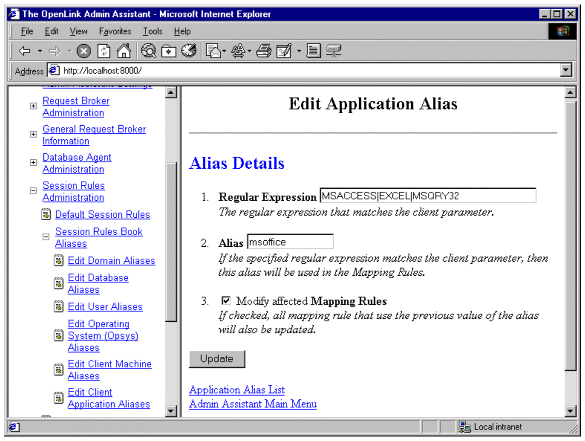
Application Alias List

Application	Alias	Action			
MSACCESS|EXCEL|MSQRY32	msoffice	Edit	Remove	Up	Down
		Add			

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

2. The example below assumes that you are modifying an Application Alias named "msoffice" with attribute values of "MSACCESS" or "EXCEL" or "MSQRY32". The Pipe symbol (character "|") indicates an OR condition. Thus, the entry depicted below implies that you want to take note of Application Connection Attributes from an OpenLink Client that hold the values "MSACCESS" or "EXCEL" or "MSQRY32" for evaluation against a mapping rules template. Once you have completed your input click the "Update" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Local intranet

Edit Application Alias

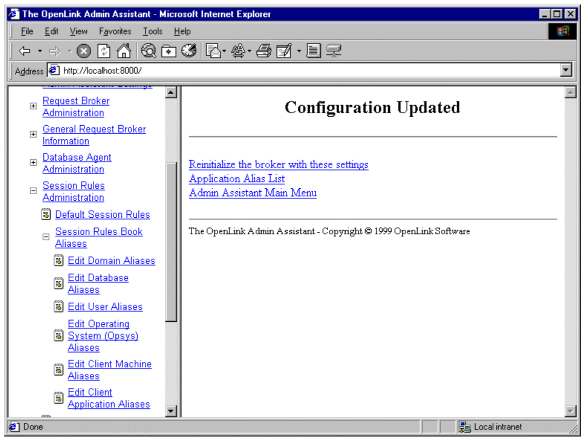
Alias Details

1. **Regular Expression** MSACCESS|EXCEL|MSQRY32
   The regular expression that matches the client parameter.
2. **Alias** msoffice
   If the specified regular expression matches the client parameter, then this alias will be used in the Mapping Rules.
3. [✓] Modify affected **Mapping Rules**
   If checked, all mapping rule that use the previous value of the alias will also be updated.

Update

Application Alias List
Admin Assistant Main Menu

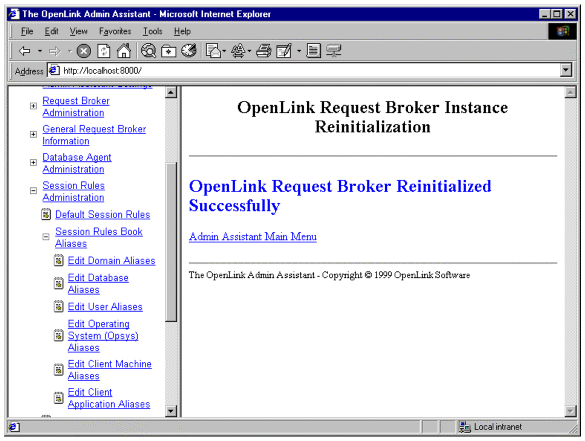
3. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Reinitialize the broker with these settings
Application Alias List
Admin Assistant Main Menu
Done
Local intranet

Configuration Updated

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

4. Admin Assistant confirms commitment of your changes to the rule book.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Admin Assistant Main Menu
Local intranet

OpenLink Request Broker Instance Reinitialization

OpenLink Request Broker Reinitialized Successfully

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

Determining The Values of OpenLink Client Generated Connection Attributes

When managing Aliases for non user configurable Connection Attributes you need to do the following in order to determine the appropriate values:

1. Start Request Broker in Debug Mode

2. Then make a connection from your OpenLink Client

3. The debug output in your Request Brokers debug window will contain the following:

```
oplrqb: client-machine.YourDomain called (123.123.123.100.1510)
oplrqb: request: domain=Oracle 8 database= serveropts=
oplrqb: connectopts= user= opsys=win32 readonly=0
oplrqb: application=OPLSCAN processid=384
```

4. Take note of the values assigned to the following Connection Attributes: opsys, readonly, and application. Apply these values to the appropriate Aliases that your are attempting to configure.

Using the Admin Assistant To Create Session Rules

There are different type of session rules that you may want to create the examples that follow guide through the process of creating some very common session rules.

Security Oriented "Session Rules" Examples

A major concern with ODBC, JDBC, UDBC and OLE-DB is the fact that you could potentially lose control over the number and types of applications that have full read-write access to your organizations databases, unfortunately database engine based security is User, Group and in some case Role driven, database engines do not have the ability to decipher the type of Client application or data access mechanism being used by a client process. Thus ODBC, UDBC, JDBC, and OLE-DB can potentially introduce serious security loopholes.

The security and configuration control attributes of the session rules book is OpenLink's single most important and distinguishing feature when compared with other data access technologies (standards or non standards based).

Session rules enable you enforce rules that protect you from many form security related exposure in a time when connectivity is increasing at an alarming rate. The examples that follow use typical security scenarios to demonstrate how session rules can be devised to address security and/or resource consumption issues introduced by the use of ODBC, JDBC, UDBC, and OLE-DB compliant solutions.

Creating A Session Rule Enforcing READ-ONLY sessions unconditionally across all OpenLink Clients

The steps that follow guide you through the process of building a session rule that enforces read-only data access across all OpenLink ODBC, JDBC, UDBC, and OLE-DB sessions:

1. Start Request Broker

2. Start a Web Browser session

3. Enter the following URL into your browser:

   If you started the Request Broker on your local machine enter:

   http://localhost:8000/

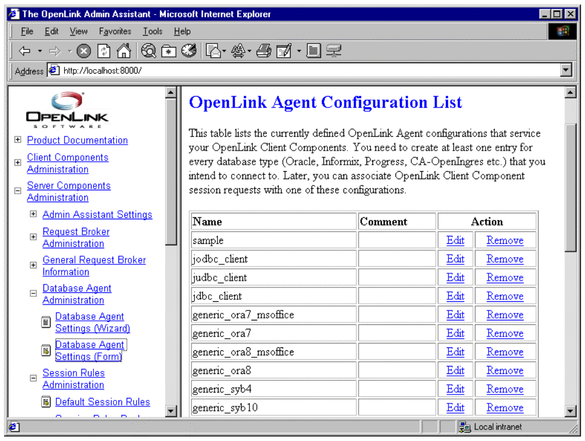
   (assuming you accepted port 8000 as the Admin Assistant port number at installation time).

   If the Request Broker in on another machine enter:

   http://<hostname or IP address>:8000

   (assuming you accepted port 8000 as the Admin Assistant port number at installation time).

4. Since you are attempting to create a read-only session rule for a particular database, you need to ensure that you actually have an OpenLink Agent template that has its ReadOnly attribute set to "Y" or "Yes". To do this you navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Database Agent Administration"->"Database Agent Settings (Form)" . Locate the database agent for your database engine type from the list presented and then click on the "Edit" hyperlink.

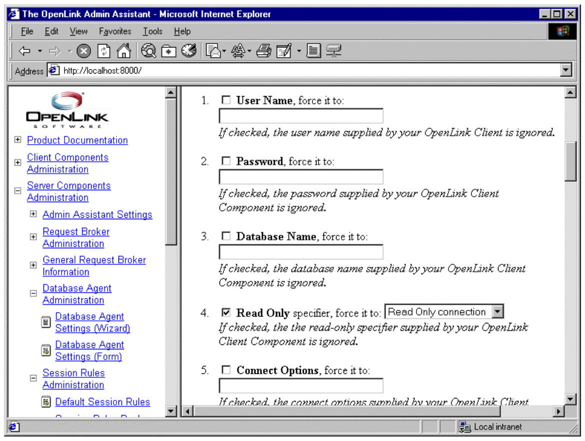

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
OPENLINK SOFTWARE
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Local intranet

OpenLink Agent Configuration List

This table lists the currently defined OpenLink Agent configurations that service your OpenLink Client Components. You need to create at least one entry for every database type (Oracle, Informix, Progress, CA-OpenIngres etc.) that you intend to connect to. Later, you can associate OpenLink Client Component session requests with one of these configurations.

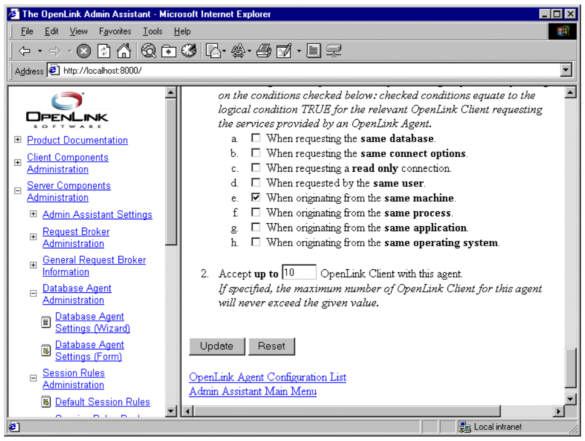
Name	Comment	Action	
sample		Edit	Remove
jodbc_client		Edit	Remove
judbc_client		Edit	Remove
jdbc_client		Edit	Remove
generic_ora7_msoffice		Edit	Remove
generic_ora7		Edit	Remove
generic_ora8_msoffice		Edit	Remove
generic_ora8		Edit	Remove
generic_syb4		Edit	Remove
generic_syb10		Edit	Remove

5. Locate the "Read Only" checkbox and then hatch the box which indicates the enforcement of Read-Only session when this agent is connected to backend database.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
OPENLINK SOFTWARE
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Local intranet

1. ☐ **User Name**, force it to:
   If checked, the user name supplied by your OpenLink Client is ignored.
2. ☐ **Password**, force it to:
   If checked, the password supplied by your OpenLink Client Component is ignored.
3. ☐ **Database Name**, force it to:
   If checked, the database name supplied by your OpenLink Client Component is ignored.
4. ☑ **Read Only** specifier, force it to: Read Only connection
   If checked, the the read-only specifier supplied by your OpenLink Client Component is ignored.
5. ☐ **Connect Options**, force it to:
   If checked, the connect options supplied by your OpenLink Client

6. Proceed to the end of this page and then click on the "Update" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
OPENLINK SOFTWARE
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Update
Reset
Local intranet

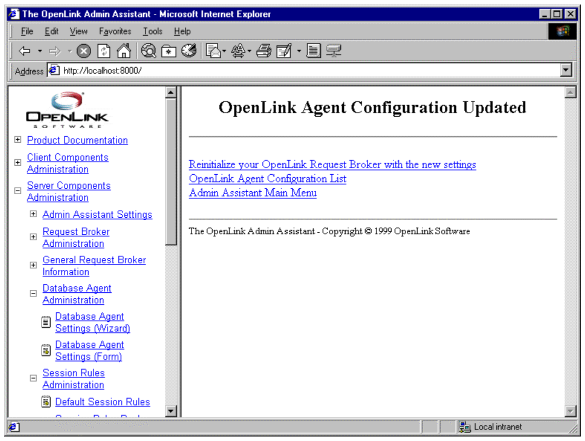
on the conditions checked below: checked conditions equate to the logical condition TRUE for the relevant OpenLink Client requesting the services provided by an OpenLink Agent.

a. ☐ When requesting the **same database**.
b. ☐ When requesting the **same connect options**.
c. ☐ When requesting a **read only** connection.
d. ☐ When requested by the **same user**.
e. ☑ When originating from the **same machine**.
f. ☐ When originating from the **same process**.
g. ☐ When originating from the **same application**.
h. ☐ When originating from the **same operating system**.

2. Accept **up to** 10 OpenLink Client with this agent.
   If specified, the maximum number of OpenLink Client for this agent will never exceed the given value.

OpenLink Agent Configuration List
Admin Assistant Main Menu

7. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink.

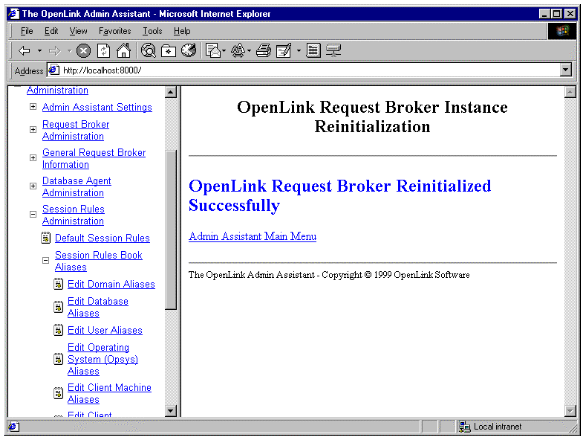

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Local intranet

OpenLink Agent Configuration Updated

Reinitialize your OpenLink Request Broker with the new settings
OpenLink Agent Configuration List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

8. Admin Assistant confirms commitment of your changes to the rule book.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Local intranet

OpenLink Request Broker Instance Reinitialization

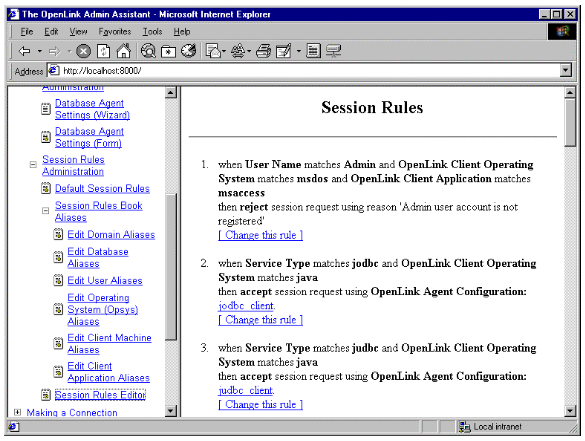
OpenLink Request Broker Reinitialized Successfully

Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

9. Now that you have created a OpenLink Agent Template for Read-Only sessions against your backend database you can now proceed to the creation of your session rule.

Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Editor" menu path which brings you to a screen identical to the one depicted below, this presents you with a list of existing session rules (all OpenLink installations come with a set of pre-configured session rules).

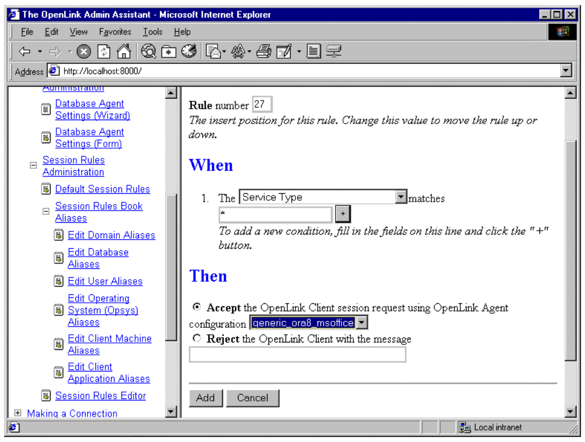

The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Session Rules Editor
Making a Connection
Local intranet

Session Rules

1. when **User Name** matches **Admin** and **OpenLink Client Operating System** matches **msdos** and **OpenLink Client Application** matches **msaccess**
   then **reject** session request using reason 'Admin user account is not registered'
   [Change this rule]
2. when **Service Type** matches **jodbc** and **OpenLink Client Operating System** matches **java**
   then **accept** session request using **OpenLink Agent Configuration:** jodbc_client.
   [Change this rule]
3. when **Service Type** matches **judbc** and **OpenLink Client Operating System** matches **java**
   then **accept** session request using **OpenLink Agent Configuration:** judbc_client.
   [Change this rule]

10. Scroll to the bottom of the session rules editor page and then click on the "Add a new rule" hyperlink, this opens up a

new session rules page.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Session Rules Editor
Making a Connection
Add
Cancel
Local intranet

Rule number 27

The insert position for this rule. Change this value to move the rule up or down.

When

1. The Service Type matches *

To add a new condition, fill in the fields on this line and click the "+" button.

Then

[✓ **Accept**] the OpenLink Client session request using OpenLink Agent configuration generic_ora8_msoffice

Reject the OpenLink Client with the message

11. Create your new session rule.

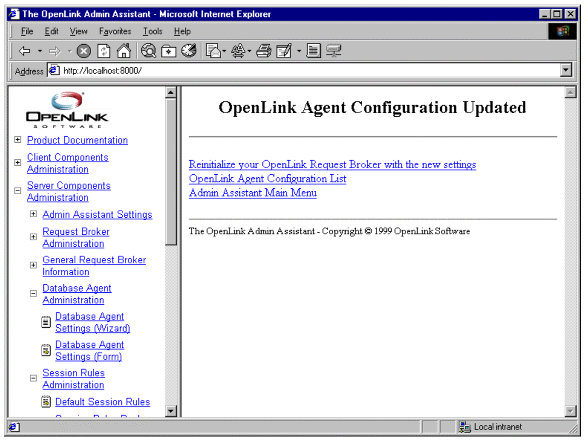
    A session rule is broken down into three parts a "Rule Number", a "When" predicate and a "Then" processing instruction.

    Rule Number:
    This sets the order in which the session rules are evaluated. The Request Broker reads scans session rules in ascending "Rule Number".

    "When" Predicate:
    Enter a regular expression value into the field beside the "+" button, this identifies to the Request Broker an OpenLink Service Type (a Domain attribute) value that forms one of the Connection Attributes that make up the session rule that you are constructing. Simply click on the "+" button to Add other Connection Attributes that will make up the session rule that you are constructing. In this example we want our session rule to apply to all OpenLink clients thus entering an "*" (which implies any Domain attribute and sub attributes) serves our needs adequately.

    "Then" Processing Instruction:
    Check the "Accept" radio button and then select an OpenLink Agent template to associate with your rule by clicking on the "configuration" Listbox. The template that you choose needs to match the one that you placed in Read-Only mode at the start of this exercise. Then click on the "Add" button.

12. You then commit your changes to the rule book by clicking on the "Reinitialize the OpenLink Request Broker with these settings" hyperlink.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
OPENLINK SOFTWARE
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Local intranet

OpenLink Agent Configuration Updated

Reinitialize your OpenLink Request Broker with the new settings
OpenLink Agent Configuration List
Admin Assistant Main Menu

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

13. Attempt a new OpenLink Client connection to you backend database and then attempt a record update, you will observe that your update attempt will be rejected and an error condition communicated to you via your ODBC, JDBC, UDBC,or OLE-DB based application.

Creating A Session Rule That Enforces READ-ONLY sessions for a specific OpenLink Client User

The steps that follow guide you through the process of building a session rule that enforces read-only data access for a specific User account across all OpenLink ODBC, JDBC, UDBC, and OLE-DB sessions.

1. Follow Steps 1 - 10 of the "Read-Only" session rules exercise above.

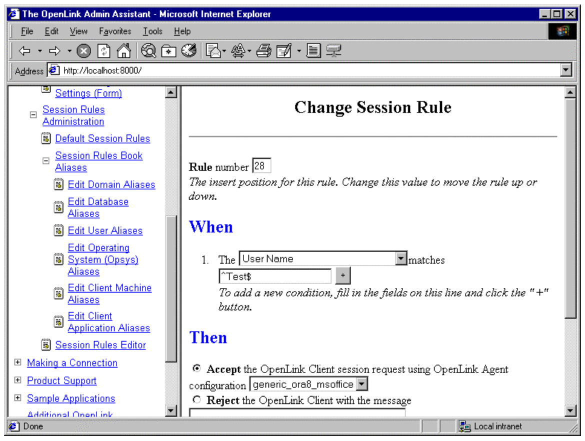
2. Create your new session rule.

   A session rule is broken down into three parts a 'Rule Number", a "When" predicate and a "Then" processing instruction.

   Rule Number:
   This sets the order in which the session rules are evaluated. The Request Broker reads scans session rules in ascending "Rule Number".

   "When" Predicate:
   Enter a regular expression value into the field beside the "+" button, this identifies to the Request Broker an OpenLink Client User (the User attribute) value that forms one of the Connection Attributes that make up the session rule that you are constructing. Simply click on the "+" button to Add other Connection Attributes that will make up the session rule that you are constructing. In this example we want our session rule to apply to a specific OpenLink Client User irrespective of anything else. Thus, entering the value "^Test$" implies that anyone that makes an ODBC, JDBC, UDBC, or OLE-DB connection using the user name "Test" operates against your backend database in Read-Only mode.

   "Then" Processing Instruction:
   Check the "Accept" radio button and then select an OpenLink Agent template to associate with your rule by clicking on the "configuration" Listbox. The template that you choose needs to match the one that you placed in Read-Only mode at the start of the Read-Only" session rules exercise. Then click on the "Add" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Session Rules Editor
Making a Connection
Product Support
Sample Applications
Additional OpenLink
Done
Local intranet

Change Session Rule

Rule number 28
The insert position for this rule. Change this value to move the rule up or down.

When

1. The User Name matches
   ^Test$ +
   To add a new condition, fill in the fields on this line and click the "+" button.

Then

[✓ **Accept** the OpenLink Client session request using OpenLink Agent configuration generic_ora8_msoffice]
Reject the OpenLink Client with the message

3. Click on the "Update" button to complete the rule creation exercise and then reinitialize the Request Broker when prompted.

4. Attempt a connection under the user name "Test" and attempt a database record update. Note that "Test" must be a valid user account in your database environment.

Creating A Session Rule That Enforces READ-ONLY sessions for a Group of Users

The steps that follow guide you through the process of building a session rule that enforces read-only data access for a specific group of users across all OpenLink ODBC, JDBC, UDBC, and OLE-DB sessions.

1. Follow Steps 1 - 10 of the "Read-Only" session rules exercise above.

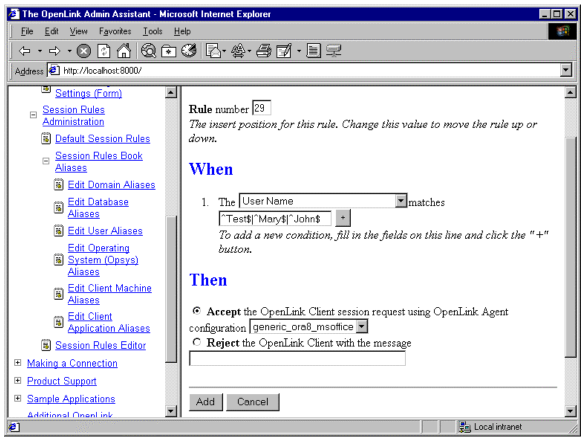
2. A session rule is broken down into three parts a "Rule Number", a "When" predicate and a "Then" processing instruction.

   Rule Number:
   This sets the order in which the session rules are evaluated. The Request Broker reads scans session rules in ascending "Rule Number".

   "When" Predicate:

Enter a regular expression value into the field beside the "+" button, this identifies to the Request Broker an OpenLink Client User (the User attribute) value that forms one of the Connection Attributes that make up the session rule that you are constructing. Simply click on the "+" button to Add other Connection Attributes that will make up the session rule that you are constructing. In this example we want our session rule to apply to a group of OpenLink Client Users irrespective of anything else. Thus, entering the value: "^Test$|^Mary$|^John$" implies that anyone that makes an ODBC, JDBC, UDBC, or OLE-DB connection using either the user name "Test" or "Mary" or "John" operates against your backend database in Read-Only mode.

"Then" Processing Instruction:
Check the "Accept" radio button and then select an OpenLink Agent template to associate with your rule by clicking on the "configuration" Listbox. The template that you choose needs to match the one that you placed in Read-Only mode at the start of the Read-Only" session rules exercise. Then click on the "Add" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Session Rules Editor
Making a Connection
Product Support
Sample Applications
Additional OpenLink
Add
Cancel
Local intranet

Rule number 29
The insert position for this rule. Change this value to move the rule up or down.

When

1. The User Name matches
^Test$|^Mary$|^John$ +
To add a new condition, fill in the fields on this line and click the "+" button.

Then

[✓ **Accept**] the OpenLink Client session request using OpenLink Agent configuration generic_ora8_msoffice
Reject the OpenLink Client with the message

3. Click on the "Update" button to complete the rule creation exercise and then reinitialize the Request Broker when prompted.

4. Attempt a connection under the user name "Test" and attempt a database record update. Note that "Test" must be a valid user account in your database environment.

Creating A Session Rule That Enforces READ-ONLY sessions for a specific OpenLink Client Application

The steps that follow guide you through the process of building a session rule that enforces read-only data access for a specific Application.

1. Follow Steps 1 - 10 of the "Read-Only" session rules exercise previously.

2. Create your new session rule.

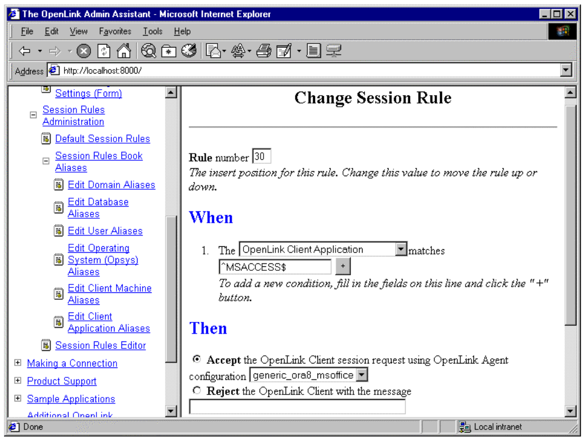
   A session rule is broken down into three parts a "Rule Number", a "When" predicate and a "Then" processing instruction.

   Rule Number:
   This sets the order in which the session rules are evaluated. The Request Broker reads scans session rules in ascending "Rule Number".

   "When" Predicate:
   Enter a regular expression value into the field beside the "+" button, this identifies to the Request Broker an OpenLink Client Application (the Application attribute) value that forms one of the Connection Attributes that make up the session rule that you are constructing. Simply click on the "+" button to Add other Connection Attributes that will make up the session rule that you are constructing. In this example we want our session rule to apply to a specific OpenLink Client Application irrespective of anything else. Thus, entering the value "^MSACCESS$" implies that anyone that makes an ODBC, JDBC, UDBC, or OLE-DB connection using the Application presented by an OpenLink Client as "MSACCESS" operates against your backend database in Read-Only mode.

   "Then" Processing Instruction:

Check the "Accept" radio button and then select an OpenLink Agent template to associate with your rule by clicking on the "configuration" Listbox. The template that you choose needs to match the one that you placed in Read-Only mode at the start of the Read-Only" session rules exercise. Then click on the "Add" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Session Rules Editor
Making a Connection
Product Support
Sample Applications
Additional OpenLink
Done
Local intranet

Change Session Rule

Rule number 30

The insert position for this rule. Change this value to move the rule up or down.

When

1. The OpenLink Client Application matches ^MSACCESS$ +

To add a new condition, fill in the fields on this line and click the "+" button.

Then

[✓ **Accept** the OpenLink Client session request using OpenLink Agent configuration generic_ora8_msoffice]

Reject the OpenLink Client with the message

3. Click on the "Update" button to complete the rule creation exercise and then reinitialize the Request Broker when prompted.

4. Attempt a connection under the user name "Test" and attempt a database record update. Note that "Test" must be a valid user account in your database environment.

Creating A Session Rule That Enforces READ-ONLY sessions for OpenLink Client connections outside of your Local Area Network (LAN)

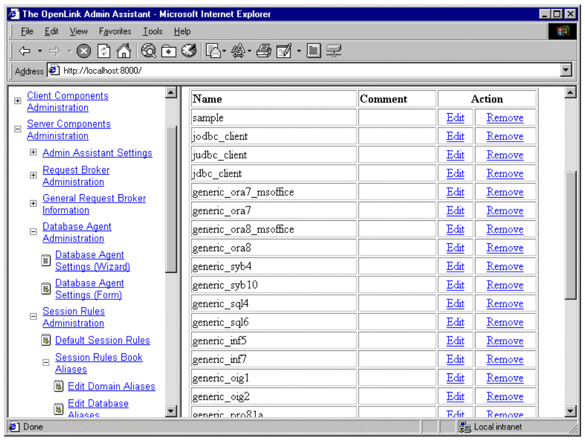
Imagine a scenario in which you staff have access to your corporate databases over the internet due to the fact that you permit employees to work remotely. It may so happen that the work that your require them to do of site requires only a read-only session. The steps that follow guide you through the process of creating a session rule that only gives read-write database access to employees within your network domain, once they connect outside of this domain the sessions are automatically read-only.

This exercise assumes that your LAN is a class C type TCP/IP network with 123.123.123 identifying your LAN.

1. Follow Steps 1 - 3 of the "Read-Only" session rules exercise previously.

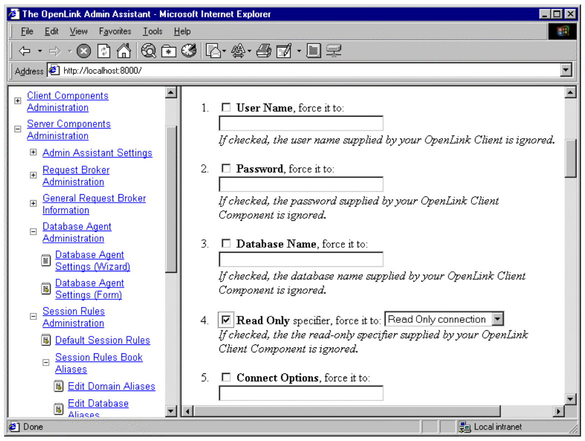
2. Since you are attempting to create a read-only session rule for a particular type of connection, you need to ensure that you actually have an OpenLink Agent template for your Read-Only database sessions and another Agent template for your Read-Write database sessions. The ReadOnly attribute of the Read-Only Agent template should be set to "Y" or "Yes", while that of the Read-Write session left unchanged.

   To set things up navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Database Agent Administration"->"Database Agent Settings (Form)" . Locate the database agent for your database engine type from the list presented and then click on the "Edit" hyperlink.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Done
Local intranet

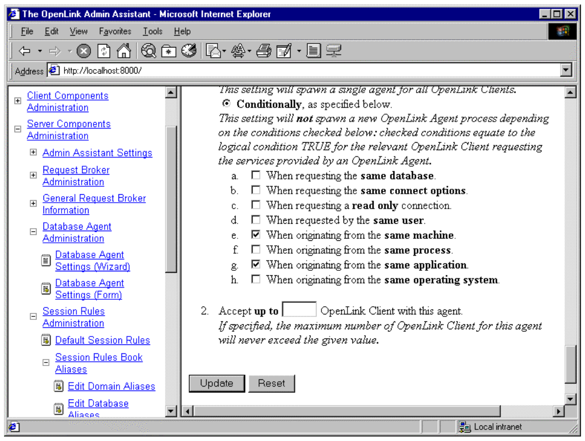
Name	Comment	Action	
sample		Edit	Remove
jodbc_client		Edit	Remove
judbc_client		Edit	Remove
jdbc_client		Edit	Remove
generic_ora7_msoffice		Edit	Remove
generic_ora7		Edit	Remove
generic_ora8_msoffice		Edit	Remove
generic_ora8		Edit	Remove
generic_syb4		Edit	Remove
generic_syb10		Edit	Remove
generic_sql4		Edit	Remove
generic_sql6		Edit	Remove
generic_inf5		Edit	Remove
generic_inf7		Edit	Remove
generic_oig1		Edit	Remove
generic_oig2		Edit	Remove
generic_pro81a		Edit	Remove

3. Locate the "Read Only" checkbox and then hatch the box which indicates the enforcement of Read-Only session when this agent is connected to backend database.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Done
Local intranet

1. ☐ **User Name**, force it to:
   If checked, the user name supplied by your OpenLink Client is ignored.
2. ☐ **Password**, force it to:
   If checked, the password supplied by your OpenLink Client Component is ignored.
3. ☐ **Database Name**, force it to:
   If checked, the database name supplied by your OpenLink Client Component is ignored.
4. ☑ **Read Only** specifier, force it to: Read Only connection
   If checked, the the read-only specifier supplied by your OpenLink Client Component is ignored.
5. ☐ **Connect Options**, force it to:

4. Proceed to the end of this page and then click on the "Update" button.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Update
Reset
Local intranet

This setting will spawn a single agent for all OpenLink Clients.
● **Conditionally**, as specified below.
*This setting will **not** spawn a new OpenLink Agent process depending on the conditions checked below: checked conditions equate to the logical condition TRUE for the relevant OpenLink Client requesting the services provided by an OpenLink Agent.*

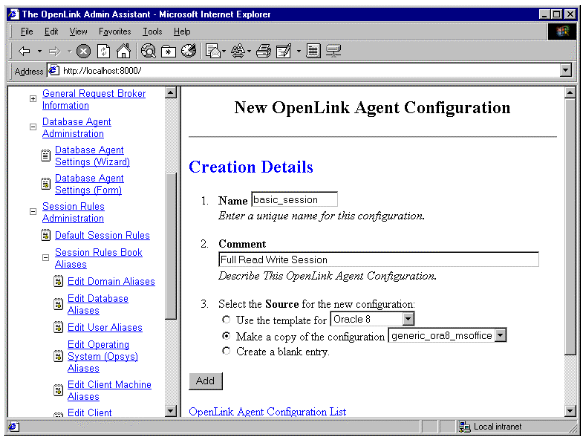
a. ☐ When requesting the **same database**.
b. ☐ When requesting the **same connect options**.
c. ☐ When requesting a **read only** connection.
d. ☐ When requested by the **same user**.
e. ☑ When originating from the **same machine**.
f. ☐ When originating from the **same process**.
g. ☑ When originating from the **same application**.
h. ☐ When originating from the **same operating system**.

2. Accept **up to** ___ OpenLink Client with this agent.
If specified, the maximum number of OpenLink Client for this agent will never exceed the given value.

5. Create a new OpenLink Agent template for the same database, but this time around we will not alter it Read-Only setting since this is disabled by default.

   To set things up by navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Database Agent Administration"->"Database Agent Settings (Form)" . Scroll to the bottom of the page and then click on the "Add" hyperlink.

Enter a name for your Agent template in the "Name" field, a descriptive comment in the "Comment" field, then click on the "Make a copy of the configuration" and select the template name that matches your Read-Only template.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Local intranet

New OpenLink Agent Configuration

Creation Details

1. **Name** basic_session
Enter a unique name for this configuration.

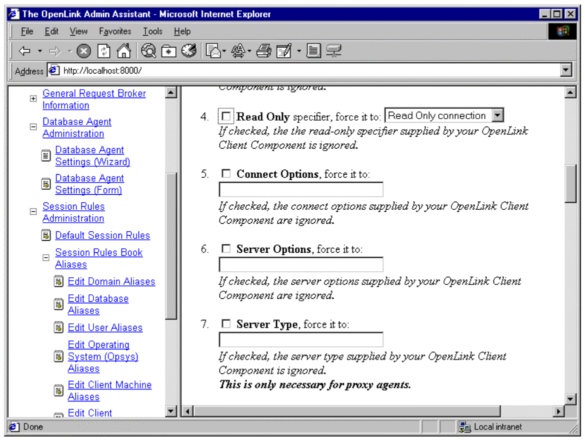
2. **Comment**
Full Read Write Session
Describe This OpenLink Agent Configuration.

3. Select the **Source** for the new configuration:
○ Use the template for Oracle 8
● Make a copy of the configuration generic_ora8_msoffice
○ Create a blank entry.

Add

OpenLink Agent Configuration List

6. You now have a new Agent template named "basic_session", but since this is based on your Read-Only template (which was created earlier) you need to change its Read-Only attributes to that Read-Only session aren't enforced when OpenLink client connections are associated with this template after the "Mapping Rules" evaluation process. To do this navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Database Agent Administration"->"Database Agent Settings (Form)" . Locate the Agent template named "basic_session" and then click on the "Edit" button. Uncheck the "Read Only" checkbox.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client
Done
Local intranet

Component is ignored.

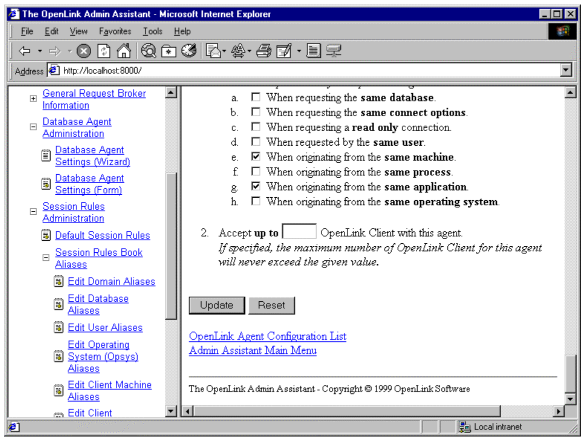
4. ☐ **Read Only** specifier, force it to: Read Only connection
If checked, the the read-only specifier supplied by your OpenLink Client Component is ignored.

5. ☐ **Connect Options**, force it to:
If checked, the connect options supplied by your OpenLink Client Component are ignored.

6. ☐ **Server Options**, force it to:
If checked, the server options supplied by your OpenLink Client Component are ignored.

7. ☐ **Server Type**, force it to:
If checked, the server type supplied by your OpenLink Client Component is ignored.
This is only necessary for proxy agents.

7. Proceed to the end of this page and then click on the "Update" button.

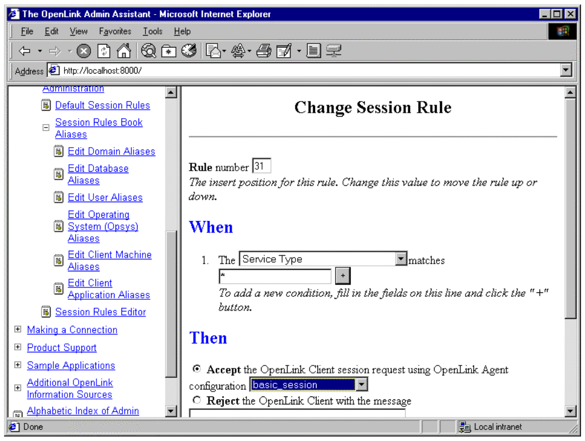

a. ☐ When requesting the **same database**.
b. ☐ When requesting the **same connect options**.
c. ☐ When requesting a **read only** connection.
d. ☐ When requested by the **same user**.
e. ☑ When originating from the **same machine**.
f. ☐ When originating from the **same process**.
g. ☑ When originating from the **same application**.
h. ☐ When originating from the **same operating system**.

2. Accept **up to** ___ OpenLink Client with this agent.
If specified, the maximum number of OpenLink Client for this agent will never exceed the given value.

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

8. Proceed to the creation of your session rule.

Follow the "Server Components Administration"->"Session Rules Administration"->"Session Rules Editor" menu path which brings you to the session rules editor, click on the "Add new rule" hyperlink so that the new session rule page is presented to you.


Change Session Rule

Rule number 31
The insert position for this rule. Change this value to move the rule up or down.

When

1. The Service Type matches *
To add a new condition, fill in the fields on this line and click the "+" button.

Then

● **Accept** the OpenLink Client session request using OpenLink Agent configuration basic_session
○ **Reject** the OpenLink Client with the message

Create your new session rule.

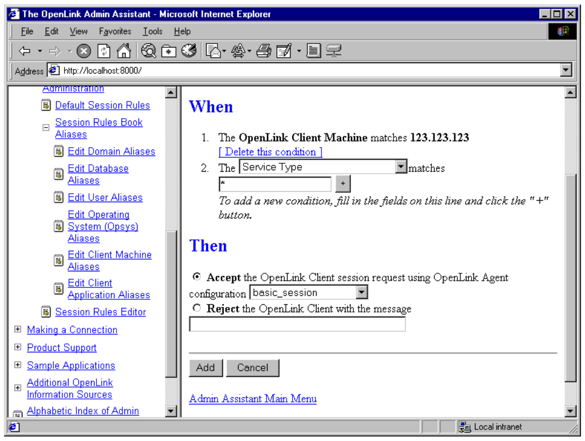
A session rule is broken down into three parts a "Rule Number", a "When" predicate and a "Then" processing instruction.

Rule Number:
This sets the order in which the session rules are evaluated. The Request Broker reads scans session rules in ascending "Rule Number". Since you are going to be creating two rules (one for read-write sessions and another for read-only sessions), you need to order your rules appropriately, remember Request Broker reads these rules for evaluation in ascending order. In this case the rule for read-write sessions is the exception to the norm so we change the number of this rule to 1.

"When" Predicate:
Enter a regular expression value into the field beside the "+" button, this identifies to the Request Broker an OpenLink Client Machine (the Machine attribute) value that forms one of the Connection Attributes that make up the session rule that you are constructing. Simply click on the "+" button to Add other Connection Attributes that will make up the session rule that you are constructing. In this example we want our session rule to apply to a specific OpenLink Client Machines irrespective of anything else. Thus, entering the value "123.123.123" implies that any machine that makes an ODBC, JDBC, UDBC, or OLE-DB connection with an from an IP Address that starts with "123.123.123" operates against your backend database in Read-Write mode.

"Then" Processing Instruction:
Check the "Accept" radio button and then select an OpenLink Agent template to associate with your rule by clicking on the "configuration" Listbox. The template that you choose needs to match the one that you placed in Read-Only mode at the start of the Read-Only" session rules exercise. Then click on the "Add" button.

Logically all other connections will end up being in Read-Only mode since they will not be resolved to having an IP Address that starts with "123.123.123". This also means that they will not be associated with the "basic_agent" Agent Template which is the only means of establishing a Read-Write session with your backend database.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Administration
Default Session Rules
Session Rules Book Aliases
Edit Domain Aliases
Edit Database Aliases
Edit User Aliases
Edit Operating System (Opsys) Aliases
Edit Client Machine Aliases
Edit Client Application Aliases
Session Rules Editor
Making a Connection
Product Support
Sample Applications
Additional OpenLink Information Sources
Alphabetic Index of Admin
Local intranet

When

1. The **OpenLink Client Machine** matches **123.123.123**
   [Delete this condition]
2. The Service Type matches
   * +
   To add a new condition, fill in the fields on this line and click the "+" button.

Then

[✓ **Accept** the OpenLink Client session request using OpenLink Agent configuration basic_session]
Reject the OpenLink Client with the message

Add Cancel

Admin Assistant Main Menu

9. Click on the "Update" button to complete the rule creation exercise and then reinitialize the Request Broker when prompted.

10. Attempt a connection from outside your LAN and attempt a database record update.

Resource Management Oriented "Session Rules" Examples

The examples addressed in this section focus on the the management of OpenLink Agent processes with the view to providing systems administrators with the ability to tune OpenLink Agents in line with operating system resource constraints.

Unconditionally Sharing One Database Agent Instance Across Numerous OpenLink Clients

You might want to reduce the number of OpenLink Agents in use at any given time, a good reason for doing this stems from the fact that OpenLink Agent instances are processes which like other processes on your machine will consume a chunk of resources per instance. The more Agents you have running the more operating system kernel resources are consumed. Thus, by setting a ReUse attribute value of "always" you could ensure that under no condition is more than one Agent instance running.

Note: An exception to this rule only occurs if the Request Broker attempts to associate a new OpenLink Client session with an existing agent that hasn't completed its processing cycle, under these circumstance the Request Broker will spawn a new Agent instance.

The steps that follow guide you through the process of creating the appropriate session rule using the Admin Assistant:

1. Start Request Broker

2. Start a Web Browser session

3. Enter the following URL into your browser:

   If you started the Request Broker on your local machine enter:

   http://localhost:8000/

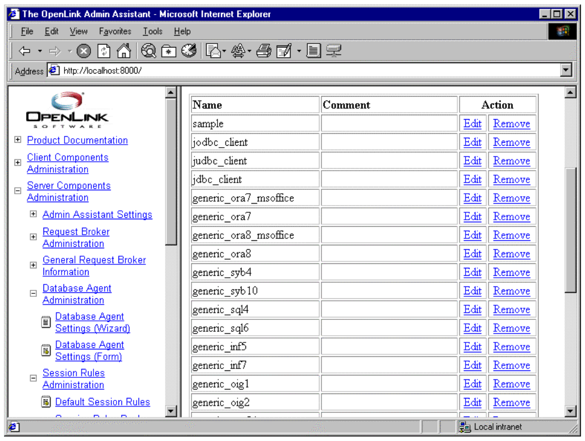
   (assuming you accepted port 8000 as the Admin Assistant port number at installation time).

   If the Request Broker in on another machine enter:

   http://<hostname or IP address>:8000

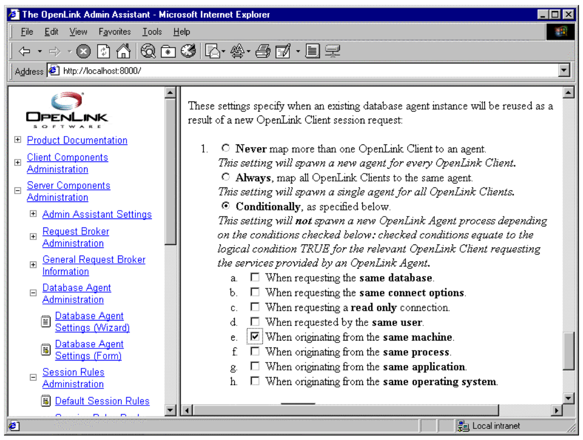
   (assuming you accepted port 8000 as the Admin Assistant port number at installation time).

4. Since you are attempting to modify the ReUse attribute of an existing Agent Template simply navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Database Agent Administration"->"Database Agent Settings (Form)" . Locate the database agent for your database engine type from the list presented and then click on the "Edit" hyperlink.



Name	Comment	Action	
sample		Edit	Remove
jodbc_client		Edit	Remove
judbc_client		Edit	Remove
jdbc_client		Edit	Remove
generic_ora7_msoffice		Edit	Remove
generic_ora7		Edit	Remove
generic_ora8_msoffice		Edit	Remove
generic_ora8		Edit	Remove
generic_syb4		Edit	Remove
generic_syb10		Edit	Remove
generic_sql4		Edit	Remove
generic_sql6		Edit	Remove
generic_inf5		Edit	Remove
generic_inf7		Edit	Remove
generic_oig1		Edit	Remove
generic_oig2		Edit	Remove

5. Scroll down to the "Client-Server Process Mapping & Control section, and then check the "Always" radio button as depicted in the screen shot below.



These settings specify when an existing database agent instance will be reused as a result of a new OpenLink Client session request:

1. **Never** map more than one OpenLink Client to an agent.
This setting will spawn a new agent for every OpenLink Client.
Always, map all OpenLink Clients to the same agent.
This setting will spawn a single agent for all OpenLink Clients.
[✓ **Conditionally**, as specified below.]
*This setting will **not** spawn a new OpenLink Agent process depending on the conditions checked below: checked conditions equate to the logical condition TRUE for the relevant OpenLink Client requesting the services provided by an OpenLink Agent.*
   a. ☐ When requesting the **same database**.
   b. ☐ When requesting the **same connect options**.
   c. ☐ When requesting a **read only** connection.
   d. ☐ When requested by the **same user**.
   e. ☑ When originating from the **same machine**.
   f. ☐ When originating from the **same process**.
   g. ☐ When originating from the **same application**.
   h. ☐ When originating from the **same operating system**.

6. Click on the "Update" button to complete the rule creation exercise and then reinitialize the Request Broker when prompted.

7. Restart your Broker in debug mode.

8. Attempt multiple OpenLink ODBC, JDBC, UDBC, OLE-DB connections from the same and different machines.

9. Then navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Request Broker Administration"->"List of Current connections" . This will show you how many OpenLink Client connections that you have open and how many OpenLink Agents are serving these sessions.

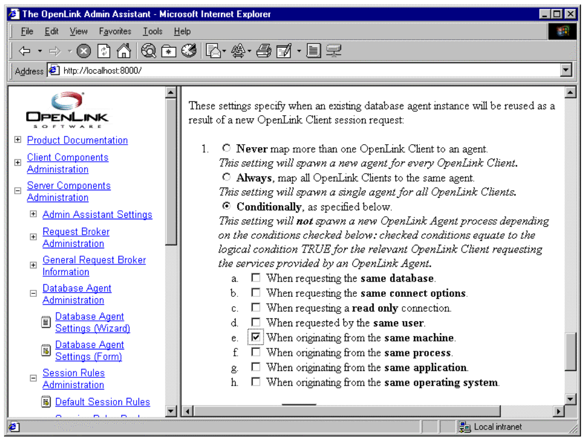
Conditionally Sharing One Database Agent Instance Across Numerous Data Access Clients

There may be times where you only want to conditionally pool connections from your OpenLink Clients against a particular OpenLink Agent instance. The steps that follow show you how to pool all OpenLink Client connections from a specific machine to a single Agent instance. Thereby creating a scenario in which an one OpenLink Agent instance is spawned per OpenLink Client machine.

The steps that follow guide you through the process of creating the appropriate session rule using the Admin Assistant:

1. Follow steps 1 -4 in the prior section.

2. Scroll down to the "Client-Server Process Mapping & Control section, then check the "Conditionally" radio button, then check the checkbox field labeled "When originating from the same machine" as depicted in the screen shot below.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Product Documentation
Client Components Administration
Server Components Administration
Admin Assistant Settings
Request Broker Administration
General Request Broker Information
Database Agent Administration
Database Agent Settings (Wizard)
Database Agent Settings (Form)
Session Rules Administration
Default Session Rules
Local intranet

These settings specify when an existing database agent instance will be reused as a result of a new OpenLink Client session request:

1. ○ **Never** map more than one OpenLink Client to an agent.
This setting will spawn a new agent for every OpenLink Client.
○ **Always**, map all OpenLink Clients to the same agent.
This setting will spawn a single agent for all OpenLink Clients.
◉ **Conditionally**, as specified below.
*This setting will **not** spawn a new OpenLink Agent process depending on the conditions checked below: checked conditions equate to the logical condition TRUE for the relevant OpenLink Client requesting the services provided by an OpenLink Agent.*
   a. ☐ When requesting the **same database**.
   b. ☐ When requesting the **same connect options**.
   c. ☐ When requesting a **read only** connection.
   d. ☐ When requested by the **same user**.
   e. ☑ When originating from the **same machine**.
   f. ☐ When originating from the **same process**.
   g. ☐ When originating from the **same application**.
   h. ☐ When originating from the **same operating system**.

3. Click on the "Update" button to complete the rule creation exercise and then reinitialize the Request Broker when prompted.

4. Attempt multiple OpenLink ODBC, JDBC, UDBC, OLE-DB connections from the same and different machines.

5. Then navigate the Admin Assistant menu down the following path: "Server Components Administration"->"Request Broker Administration"->"List of current connections" . This will show you how many OpenLink Client connections that you have open and how many OpenLink Agents are serving these sessions.

Chapter5: ODBC & JDBC Sample Applications

A number of sample applications are bundled with your Universal Data Access Driver Suite installation for the following purposes:

- To simplifying the process of getting the product up and running
- To accelerate the support case creation and resolution process
- To demonstrate the Data Access Driver Suite's unique product features highlighting the benefits it brings to your organization
- To demonstrate application programming techniques that can used to aid and assist your ODBC and JDBC programmers

The Universal Data Access Driver Suite services are consumed primarily via ODBC and JDBC applications (OLE-DB applications connect via ODBC Data Providers for OLE-DB), thus separate ODBC & JDBC sample applications (including source code) have been packaged and integrated into the installer. The current list of sample applications include:

- **C++ Demo** - an ODBC based Interactive SQL processor written in C++.
- **ODBC Bench Test** - a 32 Bit C++ program based on the industry standard TPC-A benchmark (we will be extending this program to include the TPC-C and TPC-D benchmarks also). This program helps you compare the performance of Virtuoso against other backend database engines as well as compare the performance of various ODBC Drivers connecting to any ODBC compliant backend database.
- **ODBCTEST** - ODBC based Interactive SQL processor written in 'C' for Linux & UNIX
- **ScrollDemo2** - a JDBC 2.0 sample application that demonstrates Virtuoso's support of Scrollable Cursors and its ability to perform scrollable cursor operations across heterogeneous databases.
- **JBench** - a Java and JDBC based adaptations of the industry standard TPC-A and TPC-C benchmarks. This program helps you compare the performance of Virtuoso against other backend database engines, it also helps you to compare the performance of various JDBC Drivers connecting to any JDBC compliant backend database.
-

Binary & Source File Locations

ODBC Demonstration Applications for Windows 95/98/NT/2000, Linux & UNIX:

The binary executables of these sample applications reside under the following directory structure:

<OPENLINK_INSTALLATION_DIRECTORY>\samples\odbc

The source code of some of these sample applications, when available, reside under the following directory structure, for example:

<OPENLINK_INSTALLATION_DIRECTORY>\samples\odbc\cppdemo

JDBC Demonstration Applications for Windows 95/98/NT/2000, Linux & UNIX:

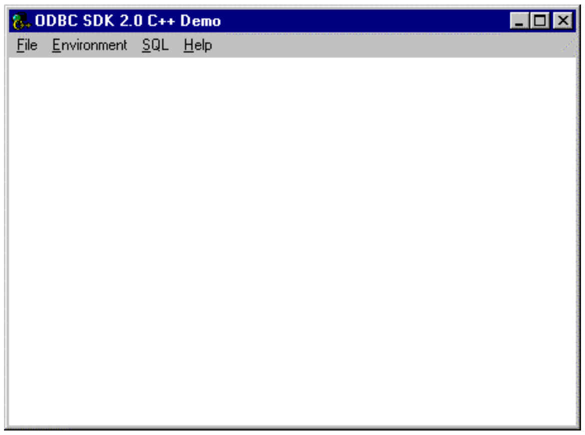
The binary executables (Java class files), and sources for these sample applications reside under the following directory structure:

< OPENLINK_INSTALLATION_DIRECTORY>\samples\jdbc\<JDK_Version>\<Demo_name>

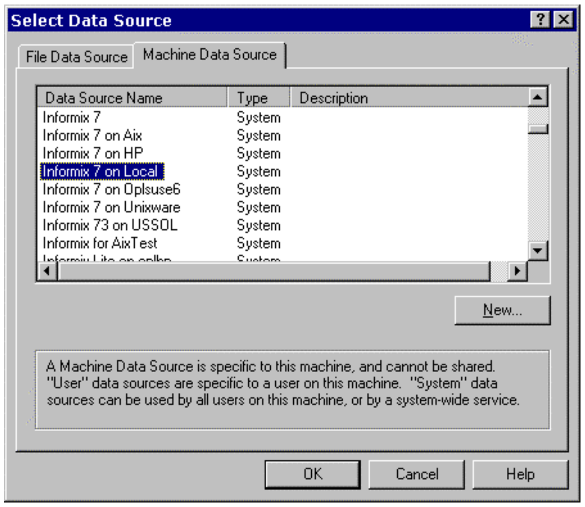
Windows 95/98/NT/2000 Based ODBC Sample Applications

C++ Demo

1. Go to the OpenLink Data Access Drivers "Start Menu" item, then click on the "C++ Demo 32 Bit" menu item.



2. Follow the Environment->Open Connection menu path. Selecting the "Open Connection" menu item results in the ODBC Driver Manager presenting you with a list of ODBC DSNs on your machine as depicted by the screen capture below:

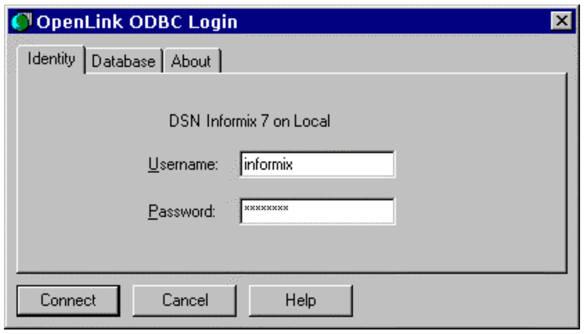


Data Source Name	Type	Description
Informix 7	System	
Informix 7 on Aix	System	
Informix 7 on HP	System	
Informix 7 on Local	System	
Informix 7 on Oplsuse6	System	
Informix 7 on Unixware	System	
Informix 73 on USSOL	System	
Informix for AixTest	System	

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

3. Select the ODBC DSN that you wish to connect to, (in this case "Informix 7 on Local" has been chosen. This will connect you to the Informix 7 database.)

4. You are then presented with a Login Dialog by the OpenLink Driver for ODBC, enter a valid user name and password into the appropriate fields.

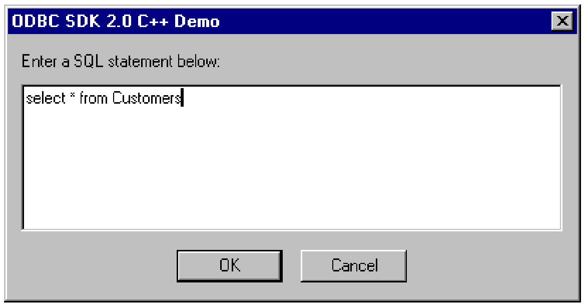


DSN Informix 7 on Local

Username: informix

Password: ********

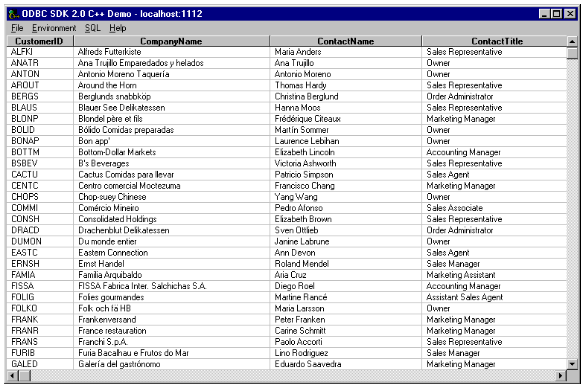
5. At this point you will be connected to the chosen datasource, you can now use the SQL-->Execute SQL menu path to open up the Interactive SQL input dialog. Enter a valid SQL statement (see example in screen shot) and then click on the "OK" button.



Enter a SQL statement below:

```
select * from Customers
```

6. You will be presented with the results of your query.

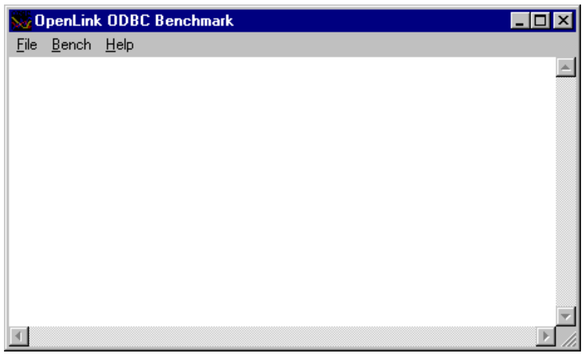


CustomerID	CompanyName	ContactName	ContactTitle
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner
AROUT	Around the Horn	Thomas Hardy	Sales Representative
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative
BLONP	Blondel père et fils	Frédérique Citeaux	Marketing Manager
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner
BONAP	Bon app'	Laurence Lebihan	Owner
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative
CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent
CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager
CHOPS	Chop-suey Chinese	Yang Wang	Owner
COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate
CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator
DUMON	Du monde entier	Janine Labrune	Owner
EASTC	Eastern Connection	Ann Devon	Sales Agent
ERNSH	Ernst Handel	Roland Mendel	Sales Manager
FAMIA	Familia Arquibaldo	Aria Cruz	Marketing Assistant
FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Accounting Manager
FOLIG	Folies gourmandes	Martine Rancé	Assistant Sales Agent
FOLKO	Folk och fä HB	Maria Larsson	Owner
FRANK	Frankenversand	Peter Franken	Marketing Manager
FRANR	France restauration	Carine Schmitt	Marketing Manager
FRANS	Franchi S.p.A.	Paolo Accorti	Sales Representative
FURIB	Furia Bacalhau e Frutos do Mar	Lino Rodriguez	Sales Manager
GALED	Galería del gastrónomo	Eduardo Saavedra	Marketing Manager

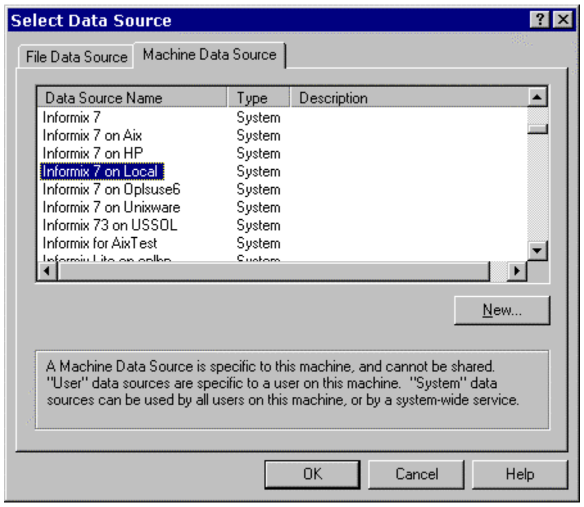
7. You exit this demo by following the Environment-->Close Connection menu path.

ODBC Bench Test 32:

1. Go to the OpenLink Data Access Drivers "Start Menu" item, then click on the "ODBC Bench Test 32 Bit" menu item. You will be presented with the "Bench Test" interface.



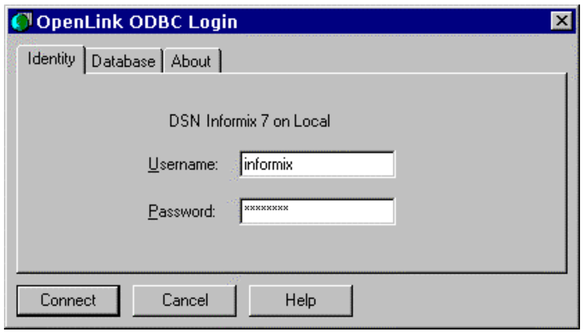
2. Follow the File-Connect menu path which initializes the ODBC Driver Manager, which in turn presents you with a list of ODBC DSNs installed on your machine. Select the DSN that you want to benchmark, remember that by benchmarking a DSN you are benchmarking the ODBC Driver that serves the DSN in question and the backend database engine that serves the ODBC Driver. Choose the name of the datasource you want to benchmark.



Data Source Name	Type	Description
Informix 7	System	
Informix 7 on Aix	System	
Informix 7 on HP	System	
Informix 7 on Local	System	
Informix 7 on Oplsuse6	System	
Informix 7 on Unixware	System	
Informix 73 on USSOL	System	
Informix for AixTest	System	

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

3. You will then be presented with a Login Dialog by the OpenLink Generic Driver for ODBC, enter a valid user name and password into the appropriate fields.

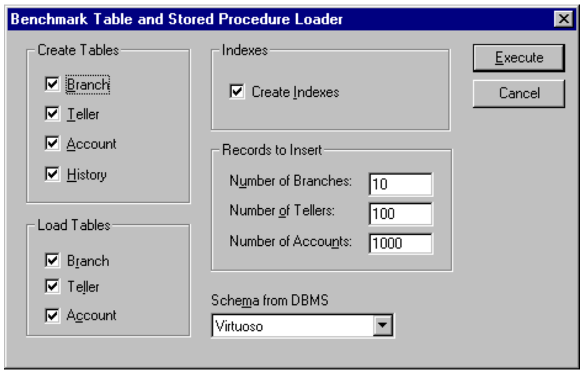


DSN Informix 7 on Local

Username: informix

Password: ********

4. Now follow the Bench-->Load Tables menu path and you will be presented with a dialog that enables you to configure key elements of your benchmark. Click the "Execute" button to commence the process of setting up your database for the benchmark tests. Please make sure you select the appropriate schema for the DBMS that you are connecting to so that the benchmark tables are created properly.



Create Tables: ☑ Branch ☑ Teller ☑ Account ☑ History

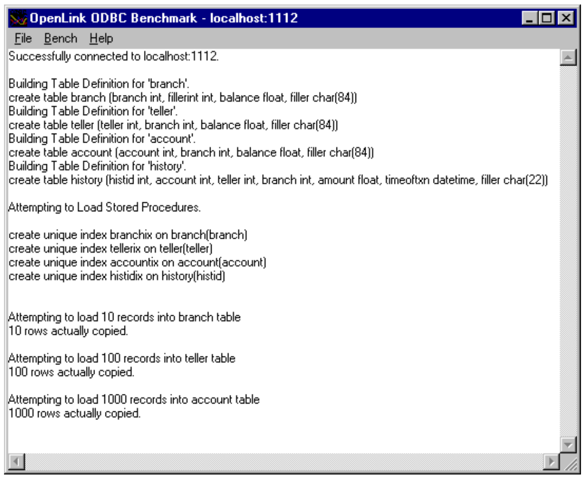
Indexes: ☑ Create Indexes

Records to Insert:
Number of Branches: 10
Number of Tellers: 100
Number of Accounts: 1000

Load Tables: ☑ Branch ☑ Teller ☑ Account

Schema from DBMS: Virtuoso

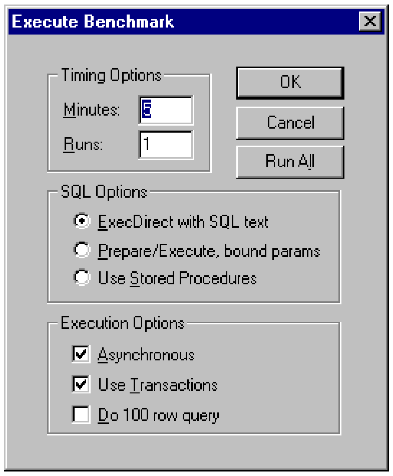
5. As the process of loading data occurs, all the way up to completion, the benchmark program will provide status information into the benchmark output pane as shown below:



```
Successfully connected to localhost:1112.

Building Table Definition for 'branch'.
create table branch (branch int, fillerint int, balance float, filler char(84))
Building Table Definition for 'teller'.
create table teller (teller int, branch int, balance float, filler char(84))
Building Table Definition for 'account'.
create table account (account int, branch int, balance float, filler char(84))
Building Table Definition for 'history'.
create table history (histid int, account int, teller int, branch int, amount float, timeoftxn datetime, filler char(22))

Attempting to Load Stored Procedures.

create unique index branchix on branch(branch)
create unique index tellerix on teller(teller)
create unique index accountix on account(account)
create unique index histidix on history(histid)


Attempting to load 10 records into branch table
10 rows actually copied.

Attempting to load 100 records into teller table
100 rows actually copied.

Attempting to load 1000 records into account table
1000 rows actually copied.
```

6. Now that all the benchmark data has been loaded into your database, follow the Bench-->Run Benchmark menu path and then configure your actual benchmark session parameters:

   These benchmark parameters fall into 3 categories, Timing Options, SQL Options, and Execution Options.



Timing Options:
These setting allow you to configure the duration related aspects of this benchmark program

Minutes - this is the duration of each benchmark run

Runs - this controls how many iterations of the benchmarks you actually run (the default is one benchmark iteration
with a duration of 5 minutes)

SQL Options:
These settings allow you to configure how your benchmark's SQL instructions are actually handled.

ExecDirect with SQL Text - this means that no form of repetitive SQL execution optimization is being applied (SQL statements are prepared and executed repetitively)

Prepare/Execute Bound Params - this means that the Parameter Binding SQL execution optimization is being applied (SQL is prepared once but executed many times without the overhead of re-preparing statements prior to execution)

Use Stored Procedures - this means that the Stored Procedure SQL optimization is being applied (benchmark instructions are stored within database being benchmarked)

Execution Options:

These settings allow you to configure the tone of your benchmark, for instance it could have Transaction scoping and a mix of record retrieval queries, or it could simply be input and update intensive with a minimal amount of record retrieval queries (the case when the 100 row query checkbox is unchecked a typical OLTP scenario)

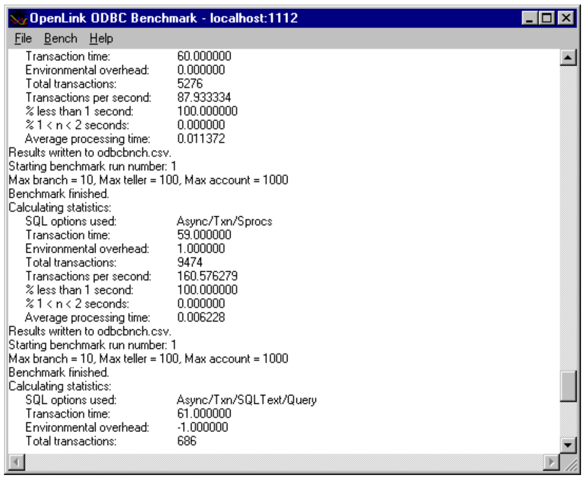
Asynchronous - execute the benchmark instructions asynchronously

Use Transactions - make the benchmark use transaction control (instructions are scoped to transaction blocks)

Do 100 row Query - perform a simulation of a 100 record retrieval as part of the benchmark activity.

7. Click on the "Run All" button if you would like all the different benchmark type combinations to be performed.

8. When benchmark run complete benchmark data is written to the benchmark program's output pane.



```
    Transaction time:            60.000000
    Environmental overhead:      0.000000
    Total transactions:          5276
    Transactions per second:     87.933334
    % less than 1 second:        100.000000
    % 1 < n < 2 seconds:         0.000000
    Average processing time:     0.011372
Results written to odbcbnch.csv.
Starting benchmark run number: 1
Max branch = 10, Max teller = 100, Max account = 1000
Benchmark finished.
Calculating statistics:
    SQL options used:            Async/Txn/Sprocs
    Transaction time:            59.000000
    Environmental overhead:      1.000000
    Total transactions:          9474
    Transactions per second:     160.576279
    % less than 1 second:        100.000000
    % 1 < n < 2 seconds:         0.000000
    Average processing time:     0.006228
Results written to odbcbnch.csv.
Starting benchmark run number: 1
Max branch = 10, Max teller = 100, Max account = 1000
Benchmark finished.
Calculating statistics:
    SQL options used:            Async/Txn/SQLText/Query
    Transaction time:            61.000000
    Environmental overhead:      -1.000000
    Total transactions:          686
```

The key pieces of benchmark data that you need to look out for are:

Total Transactions - total number of transactions completed during the benchmark run

Transactions Per Second - number of transaction completed per second for the benchmark run

Information from this benchmark is automatically written to an Excel format CSV (the file c:\odbcbnch.csv) which makes it easy for you to graph and pivot data collated from several benchmark runs. A later version of this demo will actually write the benchmark data into an ODBC DSN that you provide thereby offering even more flexibility and accessibility to benchmark data.

Linux & UNIX Based ODBC Sample Applications

ODBCTEST:

This is a simple 'C' based and ODBC compliant Interactive SQL processor.

1. Run the script openlink.sh to set up your environment:

   ```
   . openlink.sh
   ```

2. Start ODBCTEST by executing the following command:

   ```
   odbctest
   ```

3. At the SQL command prompt enter "?" for a list of ODBC DSNs on your machine or enter a valid ODBC Connect String e.g.

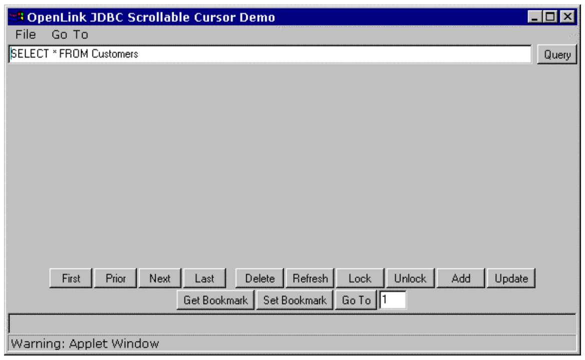
   If you have a DSN named "Marketing" you would enter: DSN=Marketing

ODBC Benchmark Application The TPC-A, TPC-C, and TPC-D benchmarks are currently under development, please monitor our Web site (http://www.openlinksw.com) for updates on these applications.

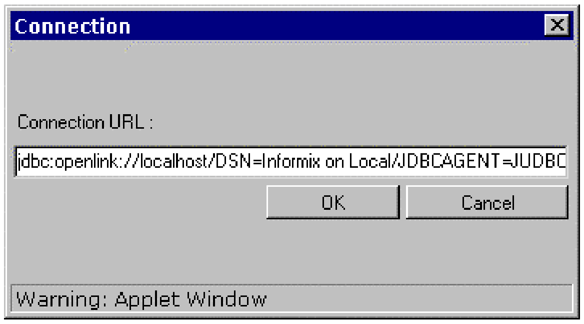
JDBC Sample Applications & Applets

ScrollDemo2 Java Application

1. Go to the OpenLink Data Access Drivers "Start Menu" program group and then follow the JDBC Samples-->ScrollDemo2 (JDK1.2) menu path, this will execute a DOS batch program that initializes the Java demo application.

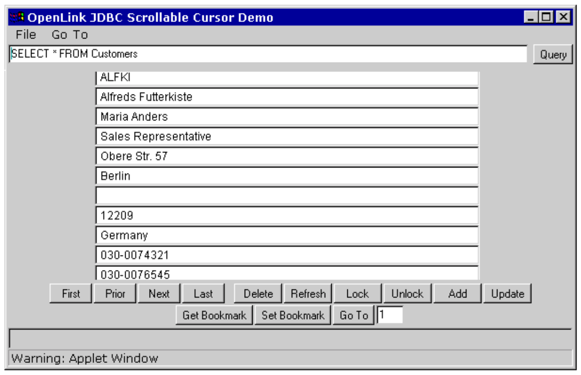

OpenLink JDBC Scrollable Cursor Demo
File Go To
SELECT * FROM Customers
Query
First Prior Next Last Delete Refresh Lock Unlock Add Update
Get Bookmark Set Bookmark Go To 1
Warning: Applet Window

2. Set the JDBC Driver Name and URL settings for your connection to a database of your choice. The "Driver Name" field identifies the OpenLink Driver. If it is left blank, then it defaults to "openlink.jdbc2.Driver", which is the OpenLink Driver for JDBC 2.0. The "Connection URL" field requires a valid OpenLink JDBC URL.



Connection URL : jdbc:openlink://localhost/DSN=Informix on Local/JDBCAGENT=JUDBC

3. Enter a valid SQL statement and then click the "Query" button.


OpenLink JDBC Scrollable Cursor Demo
File Go To
Query
First Prior Next Last Delete Refresh Lock Unlock Add Update
Get Bookmark Set Bookmark Go To 1
Warning: Applet Window

SELECT * FROM Customers

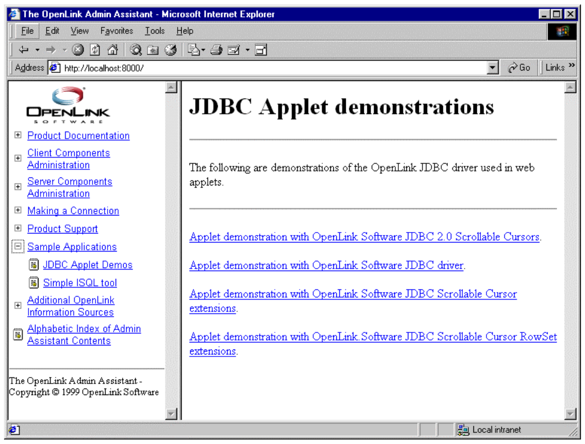
ALFKI
Alfreds Futterkiste
Maria Anders
Sales Representative
Obere Str. 57
Berlin

12209
Germany
030-0074321
030-0076545

4. You can now use the navigational buttons to Scroll backwards and forwards, each of these navigational buttons highlights OpenLink's full implementation of the JDBC 2.0 Scrollable Cursors specifications.

ScrollDemo2 Java Applet

1. Start the OpenLink Admin Assistant and then follow the Sample Applications-->JDBC Applet Demos menu path.


The OpenLink Admin Assistant - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:8000/
Go Links
Product Documentation
Client Components Administration
Server Components Administration
Making a Connection
Product Support
Sample Applications
JDBC Applet Demos
Simple ISQL tool
Additional OpenLink Information Sources
Alphabetic Index of Admin Assistant Contents
Local intranet

JDBC Applet demonstrations

The following are demonstrations of the OpenLink JDBC driver used in web applets.

Applet demonstration with OpenLink Software JDBC 2.0 Scrollable Cursors.

Applet demonstration with OpenLink Software JDBC driver.

Applet demonstration with OpenLink Software JDBC Scrollable Cursor extensions.

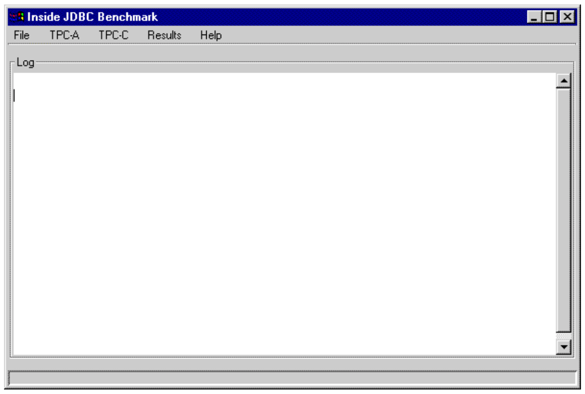
Applet demonstration with OpenLink Software JDBC Scrollable Cursor RowSet extensions.

The OpenLink Admin Assistant - Copyright © 1999 OpenLink Software

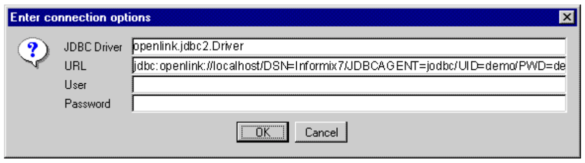
2. Click on the "ScrollDemo2" hyperlink which initializes the ScrollDemo2 applet, if you do not have a Java 1.2 or Java 2.0 compliant browser you will not be able to run this Applet demo. The other way to experience this demo is to run the Application version which uses your operating systems Java Virtual Machine (JVM) instead of a JVM inherently linked to a Web Browser.

JBench Application

1. Go to the OpenLink Data Access Drivers "Start Menu" program group and then follow the JDBC Samples-->Jbench (JDK1.1) or Jbench (JDK1.2) menu path, depending on the JVM you have installed. This will execute a DOS batch program that initializes the JBench application.



2. The follow the File-->Connect menu path to make your initial connection. You will need to identify your JDBC Driver (by providing appropriate Driver Name values in the JDBC Driver field) and then provide a valid JDBC URL for your specific JDBC Driver.

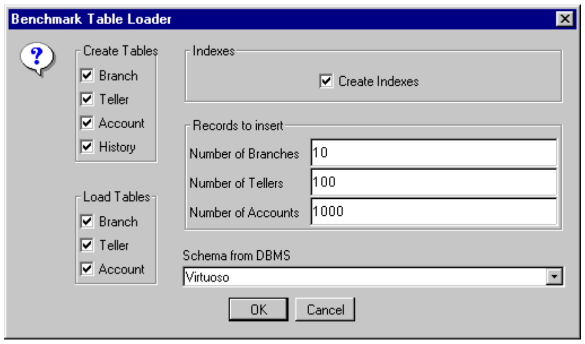

Enter connection options

JDBC Driver	openlink.jdbc2.Driver
URL	jdbc:openlink://localhost/DSN=Informix7/JDBCAGENT=jodbc/UID=demo/PWD=de
User	
Password	

3. Follow the Results-->Table URL menu path, this is how you identify (using a JDBC URL) the location of the "Results" tables into which you will be storing your benchmark data. The default URL is the current JDBC URL (the one used to establish your initial connection), but this can be a totally different database to the one being benchmarked.

4. Follow the Results-->Create menu path, this is how you actually perform the "Results" table creation in the database identified by the URL in the previous step.

5. Follow the TPC-A-->Load Tables menu path to prepare your database for the TPC-A benchmark, select a database schema type that matches the database engine that you are benchmarking. If your database isn't listed ANSI should suffice (as long as this is an ANSI SQL compliant database).


Benchmark Table Loader

Create Tables: ✓ Branch, ✓ Teller, ✓ Account, ✓ History

Load Tables: ✓ Branch, ✓ Teller, ✓ Account

Indexes: ✓ Create Indexes

Records to insert

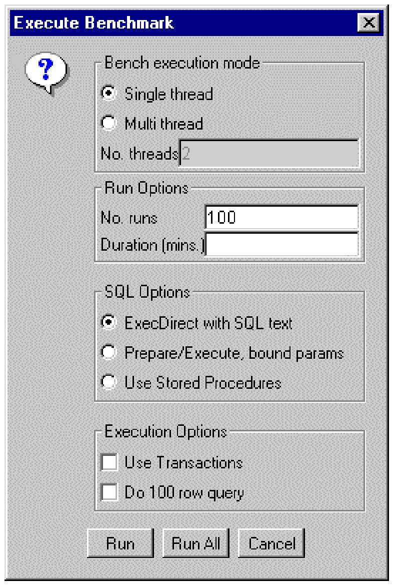
Number of Branches	10
Number of Tellers	100
Number of Accounts	1000

Schema from DBMS: Virtuoso

6. Follow the TPC-A-->Load Procedures menu path to Load the TPC-A stored procedures.

7. Now that all the benchmark data and stored procedures have been loaded into your database, follow the TPC-A-->Run Benchmark menu path and then configure your actual benchmark session parameters:

   The benchmark parameters fall into 4 categories, Bench execution mode, Run Options, SQL Options, and Execution Options.



Bench execution mode:
These setting allow you to configure the threads used for the benchmark.

Decide on a single or multiple threads test.
No. Threads - this is the number of concurrent threads to be used during the benchmark.

Run Options:
These setting allow you to configure the duration related aspects of this benchmark program.

No. runs - this controls how many iterations of the benchmarks you actually run (the default is 100 benchmark iterations).

Duration (mins.) - this is the duration in minutes of each benchmark run.

SQL Options:
These setting allow you to configure how your benchmark's SQL instructions are actually handled.

ExecDirect with SQL Text - this means that no form of repetitive SQL execution optimization is being applied (SQL statements are prepared and executed repetitively)

Prepare/Execute Bound Params - this means that the Parameter Binding SQL execution optimization is being applied (SQL is prepared once but executed many times without the overhead of re-preparing statements prior to execution)

Use Stored Procedures - this means that the Stored Procedure SQL optimization is being applied (benchmark instructions are stored within database being benchmarked)

Run All - this implies you want to perform all of the above benchmarks

Execution Options:

These setting allow you to configure the tone of your benchmark, for instance it could have Transaction scoping and a mix of record retrieval queries, or it could simply be input and update intensive with a minimal amount of record retrieval queries (the case when the 100 row query checkbox is unchecked a typical OLTP scenario)

Use Transactions - make the benchmark use transaction control (instructions are scoped to transaction blocks)

Do 100 row Query - perform a simulation of a 100 record retrieval as part of the benchmark activity.

8. Run your TPC-A benchmark.

9. Follow the TPC-A-->Cleanup menu path to clean up your database so that you can then run other benchmarks (TPC-C like benchmark).

10. To run the TPC-C benchmark simply follow the appropriate menu path, create the benchmark tables & stored procedures, load the benchmark data and then run the TPC-C benchmark.