Progress OpenEdge SQL: Connection Failover

Jyothi Sundaragiri

WHITEPAPER

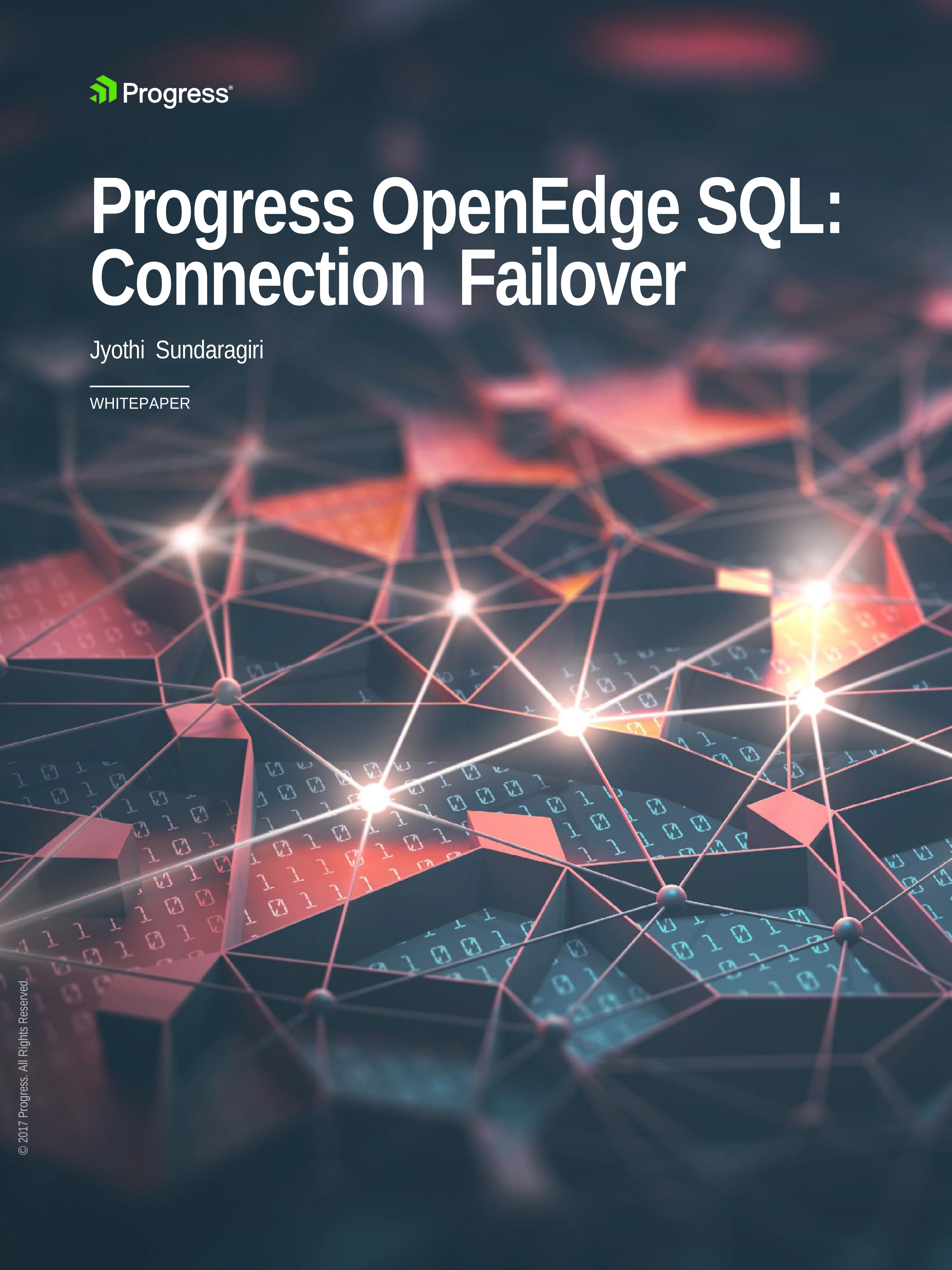


Table of Contents

Connection Failover / 3

Alternate Servers / 3

Configuring Alternate Servers for ODBC / 4

Configuring Alternate Servers for JDBC / 5

Failover Modes: Basic to Comprehensive / 6

1. Connection Failover / 6

2. Extended Connection Failover / 7

3. Select Connection Failover / 8

Summary / 9

This document provides an overview of Connection Failover for applications using the Progress® OpenEdge® SQL engine.

OpenEdge SQL supports Connection Failover provided by Progress DataDirect® Connect Series for ODBC and JDBC drivers. These ODBC and JDBC drivers are embedded in SQL client products, enabling them to use Connection Failover.

Connection Failover

To ensure uninterrupted access to the data, Connection Failover enables an application to connect to an alternate or backup database server if the primary database server is unavailable due to hardware failure or traffic overload. The DataDirect Connect Series for ODBC and JDBC drivers provides three levels of failover protection.

The three types of failover modes are:

1. Connection Failover
2. Extended Connection Failover
3. Select Connection Failover

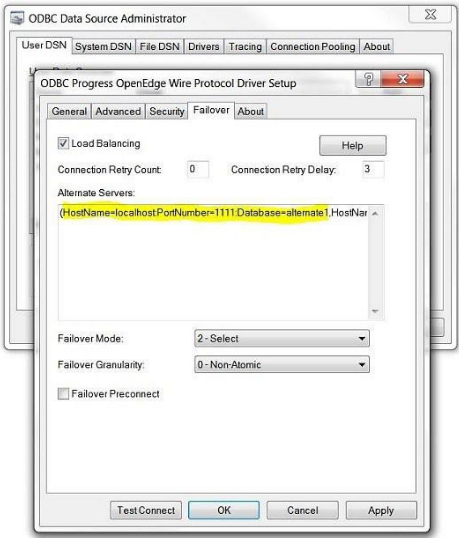
Alternate Servers

Irrespective of which failover mode you choose, you must configure one or multiple alternate servers using the Alternate Servers connection option. Alternate servers should mirror data on the primary server or be part of a configuration where multiple servers contain identical data, as in OpenEdge Replication.

Configuring Alternate Servers for ODBC

Alternate Servers can be specified in the Failover tab while creating an ODBC Data Source Name (DSN). A list of server names can be specified in a comma-separated list as follows:

```
(HostName=localhost:PortNumber=1111:
Database=alternate1,HostName=localhost:
PortNumber=2222:Database=alternate2)
```



☑ Load Balancing

Connection Retry Count: 0   Connection Retry Delay: 3

Alternate Servers:

(HostName=localhost:PortNumber=1111:Database=alternate1,HostNar

Failover Mode: 2 - Select

Failover Granularity: 0 - Non-Atomic

□ Failover Preconnect

Additionally, you can change other available options listed below:

1. Connection Retry Count defines the number of times the driver attempts to connect to the primary server and alternate database servers (if configured) after the initial unsuccessful connection attempt. It can be used with connection failover, extended connection failover and select failover.

2. Connection Retry Delay specifies the wait interval (in seconds) between attempts to connect to the primary and alternate database servers. The default interval is three seconds.

3. If **Failover Mode** is set to Extended Connection (FailoverMode=1) or Select (FailoverMode=2), set the Failover Granularity connection option to specify how you want the driver to behave if errors occur while trying to re-establish a lost connection. The default behaviour of the driver is Non-Atomic (FailoverGranularity=0), which

continues with the failover process and posts any errors on the statement on which they occur. Other values are:

a. Atomic (FailoverGranularity=1): The driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated because of repositioning a result set to the last row position, the driver continues with the failover process and generates a warning that the Select statement must be reissued.

b. Atomic Including Repositioning (FailoverGranularity=2): The driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

c. Disable Integrity Check (FailoverGranularity=3: The driver does not verify that the rows restored during the failover process match the original rows. This value applies only when Failover Mode is set to Select (FailoverMode=2).

4. Optionally, enable the Failover Preconnect connection option (FailoverPreconnect=1) if you want the driver to establish a connection with the primary and an alternate server at the same time. This value applies only when Failover Mode is set to Extended Connection (FailoverMode=1) or Select (FailoverMode=2). The default behaviour is to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection (FailoverPreconnect=0).

Configuring Alternate Servers for JDBC

For JDBC clients, Alternate Servers can be specified in JDBC URL to use any of the Failover modes.

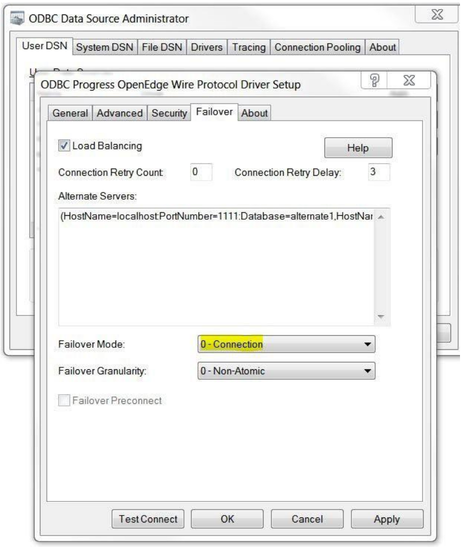
```
sqlexp -driverUrl
“jdbc:datadirect:openedge://localhost:9999;DatabaseName=primary_server;User=dbauser
1;Password=oesqldba;AlternateServers=(localhost:1111;DatabaseName=alternatedb1,local
host:2222;DatabaseName=alternatedb2);ConnectionRetryCount=3;ConnectionRetryDelay=0;F
ailoverMode=connect”
```

Failover Modes:
Basic to Comprehensive

1. Connection Failover

Provides failover protection for new connections only. The driver fails over new connections to an alternate, or backup, database server if the primary database server is unavailable. This is the default failover method.

The following image illustrates this behavior:



ODBC Data Source Administrator

User DSN | System DSN | File DSN | Drivers | Tracing | Connection Pooling | About

ODBC Progress OpenEdge Wire Protocol Driver Setup

General | Advanced | Security | Failover | About

☑ Load Balancing

Connection Retry Count: 0

Connection Retry Delay: 3

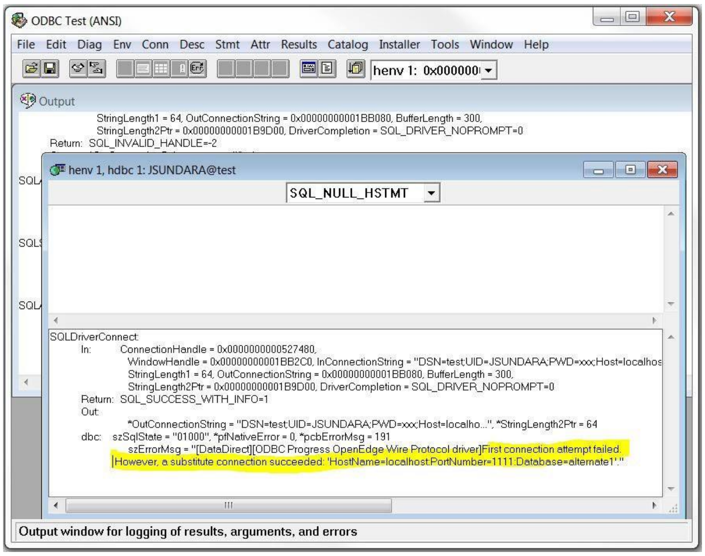
Alternate Servers:

(HostName=localhost:PortNumber=1111:Database=alternate1,HostNar

Failover Mode: 0 - Connection

Failover Granularity: 0 - Non-Atomic

□ Failover Preconnect



ODBC Test (ANSI)

```
            StringLength1 = 64, OutConnectionString = 0x00000000001BB080, BufferLength = 300,
            StringLength2Ptr = 0x00000000001B9D00, DriverCompletion = SQL_DRIVER_NOPROMPT=0
    Return:  SQL_INVALID_HANDLE=-2
```

henv 1, hdbc 1: JSUNDARA@test

```
SQLDriverConnect:
        In:         ConnectionHandle = 0x0000000000527480,
                    WindowHandle = 0x00000000001BB2C0, InConnectionString = "DSN=test;UID=JSUNDARA;PWD=xxx;Host=localhos
                    StringLength1 = 64, OutConnectionString = 0x00000000001BB080, BufferLength = 300,
                    StringLength2Ptr = 0x00000000001B9D00, DriverCompletion = SQL_DRIVER_NOPROMPT=0
        Return:  SQL_SUCCESS_WITH_INFO=1
        Out:
                    *OutConnectionString = "DSN=test;UID=JSUNDARA;PWD=xxx;Host=localho...", *StringLength2Ptr = 64
        dbc:    szSqlState = "01000", *pfNativeError = 0, *pcbErrorMsg = 191
                    szErrorMsg = "[DataDirect][ODBC Progress OpenEdge Wire Protocol driver]First connection attempt failed.
                  However, a substitute connection succeeded: 'HostName=localhost:PortNumber=1111:Database=alternate1'."
```

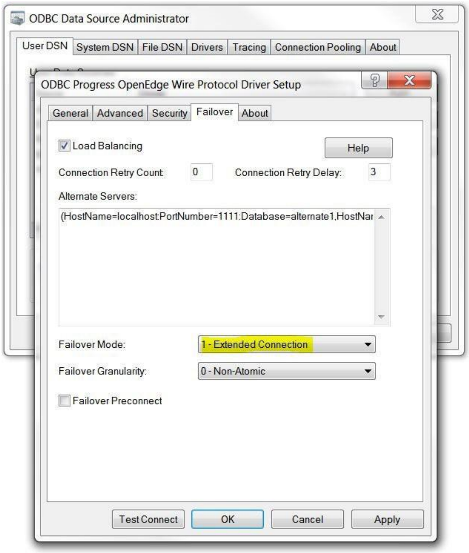
Output window for logging of results, arguments, and errors

Sample JDBC URL to use Connection Failover:

```
sqlexp -driverUrl “jdbc:datadirect:openedge://localhost:9999;DatabaseName=primary_
server;User=dbauser1;Password=oesqldba;AlternateServers=(localhost:1111;DatabaseNam
e=alternatedb1,localhost:2222;DatabaseName=alternatedb2);ConnectionRetryCount=3;Conn
ectionRetryDelay=0;FailoverMode=connect”
```

2. Extended Connection Failover

Provides failover protection for new connections and lost database connections.
If a connection to the database is lost, the driver fails over the connection to an alternate server, preserving the state of the connection at the time it was lost, but not any work in progress.



☑ Load Balancing

Connection Retry Count: 0   Connection Retry Delay: 3

Alternate Servers:

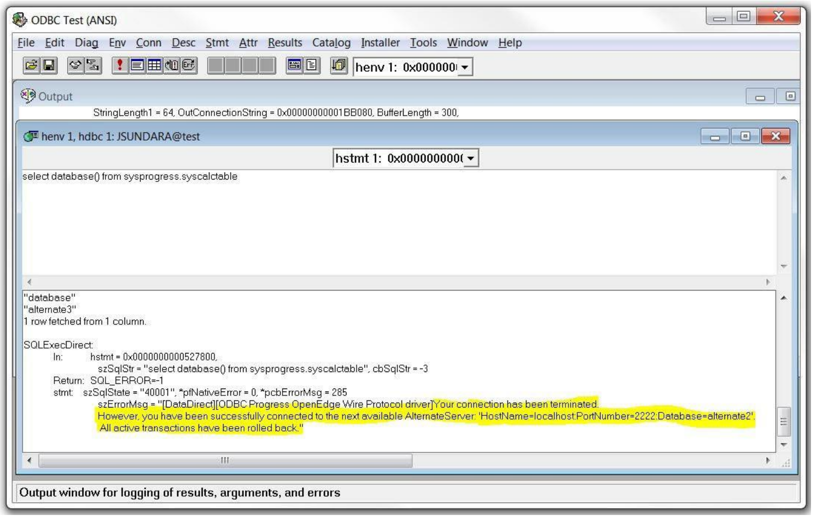
(HostName=localhost:PortNumber=1111:Database=alternate1,HostNar

Failover Mode: 1 - Extended Connection

Failover Granularity: 0 - Non-Atomic

□ Failover Preconnect

The following image illustrates this behavior:


ODBC Test (ANSI)
File Edit Diag Env Conn Desc Stmt Attr Results Catalog Installer Tools Window Help
henv 1: 0x000000(
Output
henv 1, hdbc 1: JSUNDARA@test
hstmt 1: 0x000000000(
Output window for logging of results, arguments, and errors

```
StringLength1 = 64, OutConnectionString = 0x00000000001BB080, BufferLength = 300,
```

```
select database() from sysprogress.syscalctable
```

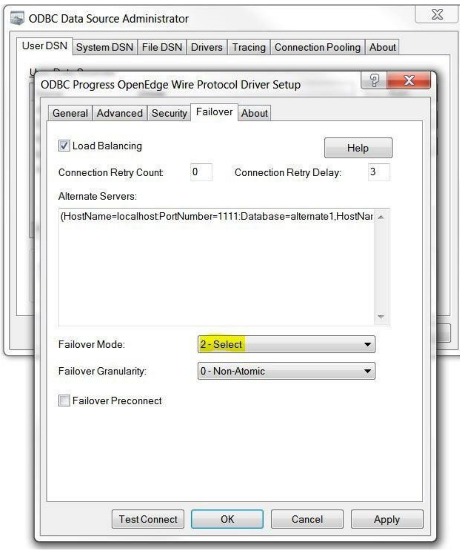
```
"database"
"alternate3"
1 row fetched from 1 column.

SQLExecDirect:
        In:         hstmt = 0x0000000000527800,
                      szSqlStr = "select database() from sysprogress.syscalctable", cbSqlStr = -3
        Return:   SQL_ERROR=-1
        stmt:    szSqlState = "40001", *pfNativeError = 0, *pcbErrorMsg = 285
                      szErrorMsg = "[DataDirect][ODBC Progress OpenEdge Wire Protocol driver]Your connection has been terminated.
                      However, you have been successfully connected to the next available AlternateServer: 'HostName=localhost:PortNumber=2222:Database=alternate2'.
                       All active transactions have been rolled back."
```

Sample JDBC URL to use Extended Connection Failover:

```
sqlexp -driverUrl “jdbc:datadirect:openedge://localhost:9999;DatabaseName=primary_
server;User=dbauser1;Password=oesqldba;AlternateServers=(localhost:1111;DatabaseNam
e=alternatedb1,localhost:2222;DatabaseName=alternatedb2);ConnectionRetryCount=3;Conn
ectionRetryDelay=0;FailoverMode=extended”
```

3. Select Connection Failover

Provides failover protection for new connections and lost database connections. Inaddition to this, it preserves the state of any work that was being performed by the Select statement at the time of connection loss. If your application was iterating through results at the time of the failure when the connection is re-established, the driver can reposition on the same row where it stopped so that the application does not have to undo all its previous result processing.



☑ Load Balancing

Connection Retry Count: 0   Connection Retry Delay: 3

Alternate Servers:

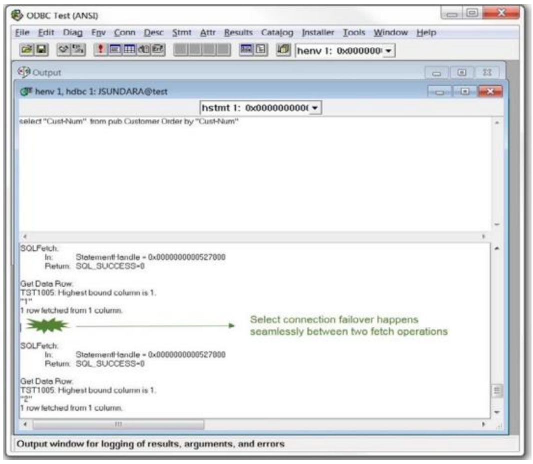
(HostName=localhost:PortNumber=1111:Database=alternate1,HostNar

Failover Mode: 2 - Select

Failover Granularity: 0 - Non-Atomic

☐ Failover Preconnect

The following image illustrates this behavior:


ODBC Test (ANSI)
File Edit Diag Env Conn Desc Stmt Attr Results Catalog Installer Tools Window Help
henv 1: 0x000000(
Output
henv 1, hdbc 1: JSUNDARA@test
hstmt 1: 0x0000000000(

```
select "Cust-Num" from pub Customer Order by "Cust-Num"
```

```
SQLFetch:
        In:       StatementHandle = 0x0000000000527000
        Return:   SQL_SUCCESS=0

Get Data Row:
TST1005: Highest bound column is 1.
"1"
1 row fetched from 1 column.

SQLFetch:
        In:       StatementHandle = 0x0000000000527000
        Return:   SQL_SUCCESS=0

Get Data Row:
TST1005: Highest bound column is 1.
"2"
1 row fetched from 1 column.
```

Select connection failover happens seamlessly between two fetch operations

Output window for logging of results, arguments, and errors

Sample JDBC URL to use Select Connection Failover:

```
sqlexp -driverUrl “jdbc:datadirect:openedge://localhost:9999;DatabaseName=primary_
server;User=dbauser1;Password=oesqldba;AlternateServers=(localhost:1111;DatabaseNam
e=alternatedb1,localhost:2222;DatabaseName=alternatedb2);ConnectionRetryCount=3;Conn
ectionRetryDelay=0;FailoverMode=select”
```

Summary

Connection Failover helps customize the drivers by configuring a list of alternate database servers that are tried if the primary server is not accepting connections. Connection attempts continue until a connection is successfully established or until all the alternate database servers have been tried for the specified number of times.

The following table summarizes how failover-related connection options work with the drivers.

OPTIONS	CHARACTERISTICS
Alternate Servers	One or multiple alternate database servers. An IP address or host name identifying each server is required.
Connection Retry Count	Number of times the driver retries the primary database server, and if specified, alternate servers until a successful connection is established.
Connection Retry Delay	Wait interval, in seconds, between connection retry attempts when the Connection Retry Count option is set to a positive integer.
Failover Granularity	The type of behaviour that the driver exhibits when errors are detected during the failover process.
Failover Mode	The type of failover that the driver attempts.
Failover Preconnect	Determines whether the driver makes a connection attempt to the next server in the Alternate Servers list at the time of the initial connection.

```
[ODBC Data Sources]
Description=Progress_SQL92_Driver
Driver=/vobs_prgs/linuxx86_64/dlc/odbc/lib/pgoe27.so
...
...
ConnectionRetryCount=4
...
ConnectionRetryDelay=3
...
AlternateServers=(HostName=oelxdev12:PortNumber=51999:Database=test1,
HostName=oelxdev12:PortNumber=61999:Database=test2)
...
FailoverMode=1
...
FailoverGranularity=0
...
FailoverPreconnect=1
...
```

Specifically, this odbc.ini configuration tells the driver to:

- Use two alternate servers as connection failover servers
- Attempt to connect four more times if the initial attempt fails
- Wait three seconds between attempts
- Try the primary and alternate servers in sequential order
- Attempt reconnecting on new and lost connections
- Establish a connection with the primary and alternate servers at the same time

About Progress

Progress (NASDAQ: PRGS) offers the leading platform for developing and deploying mission-critical business applications. Progress empowers enterprises and ISVs to build and deliver cognitive-first applications that harness big data to derive business insights and competitive advantage. Progress offers leading technologies for easily building powerful user interfaces across any type of device, a reliable, scalable and secure backend platform to deploy modern applications, leading data connectivity to all sources, and award-winning predictive analytics that brings the power of machine learning to any organization. Over 1,700 independent software vendors, 100,000 enterprise customers and 2 million developers rely on Progress to power their applications. Learn about Progress at www.progress.com or +1-800-477-6473.

At Progress, we are powering the world's data. Our connectivity solutions include the full range of relational, cloud, NoSQL and big data across industry standards, including ODBC, JDBC, ADO.NET and OData. To learn more, visit the Progress Community to network, share and learn from OpenEdge experts and developers worldwide. Contact Us

Worldwide Headquarters

Progress, 14 Oak Park, Bedford, MA 01730 USA
Tel: +1 781 280-4000 Fax: +1 781280-4095
On the Web at: www.progress.com
Find us on facebook.com/progresssw twitter.com/progresssw youtube.com/progresssw
For regional international office locations and contact information, please go to www.progress.com/worldwide

Progress®