

Infrastructure at your Service.

# Oracle 12c – Transaction Guard and Application Continuity



Invention & Infrastructure  
at your service.



# About me

**Alain Lacour**

Consultant

Mobile +41 79 199 96 44

[alain.lacour@dbi-services.com](mailto:alain.lacour@dbi-services.com)

[www.dbi-services.com](http://www.dbi-services.com)



# Who we are

## dbi services

### Experts At Your Service

- > 35 specialists in IT infrastructure
- > Certified, experienced, passionate

### Based In Switzerland

- > 100% self-financed Swiss company
- > Over CHF 4 mio. turnover

### Leading In Infrastructure Services

- > More than 100 customers in CH, D, & F
- > Over 30 SLAs dbi FlexService contracted



# Agenda

---

1. What's new?
2. Principles
3. Setup
4. Code sample
5. Benefits
6. Restrictions
7. Demo

# What's new?



## Intensity & Infrastructure at your service.

- > Problematic
- > Improved HA functionality

# What's new?

## Problematic

---

**Several issues must be considered upon database failures**

- > Hang
- > Error Handling
- > Outcome of In-Flight Work
- > Resubmission of In-Flight Work

# What's new?

## Improved High Availability functionality

### Since Oracle 10g

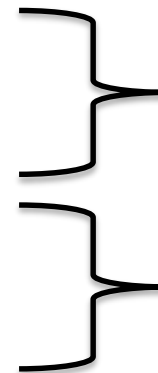
- > Fast Application Notification (FAN)
  - > Oracle Notification Service (ONS)
- > Fast Connection Failover (FCF)
- > Connection and runtime load balance (CLB - RLB)



Universal  
Connection  
Pool

### Added with 12c

- > Recoverable error
- > Logical Transaction ID (ltxid)
- > Replay through Oracle JDBC Driver
  - > Transaction kept in java application cache



Transaction  
Guard

Application  
Continuity

# Principles

## Insight & Infrastructure at your service.

- > FAN
- > CLB
- > RLB
- > FCF
- > Transaction Guard
- > Application continuity



# Principles

## FAN

---

### Fast Application Notification

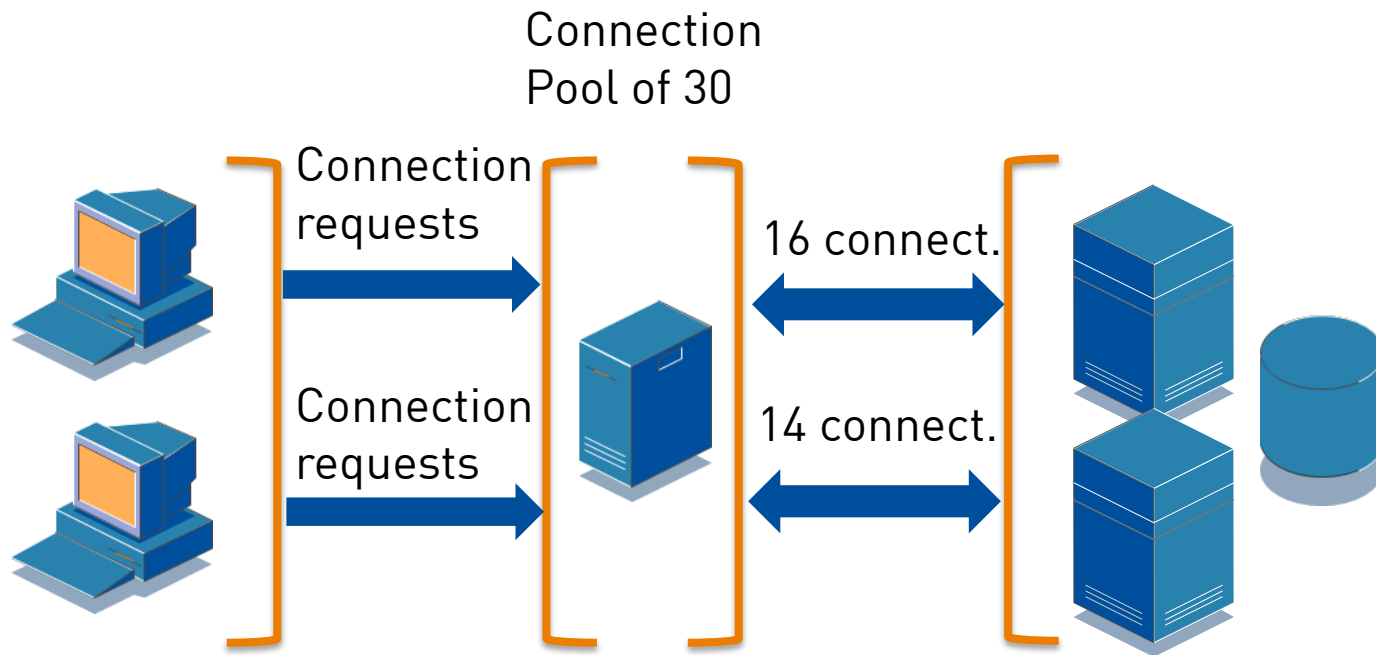
- > FAN delivers information about RAC and related services
- > Communication with other components is based on Oracle Notification System (ONS) message
- > Workload information about services enable Runtime Load Balancing for .Net, JAVA and OCI clients
- > Up / Down Events of nodes, instances, services
  - > Enables automatic Connection Pool management limiting usage of bad connections
- > FAN leverages services

# Principles

## CLB

### Connection time Load Balancing

- > At application server startup, the connections of the connection pool are distributed over the nodes based on remote listener setup
- > Even if the distribution is not perfect, the load will be well balanced by Runtime Load Balancing

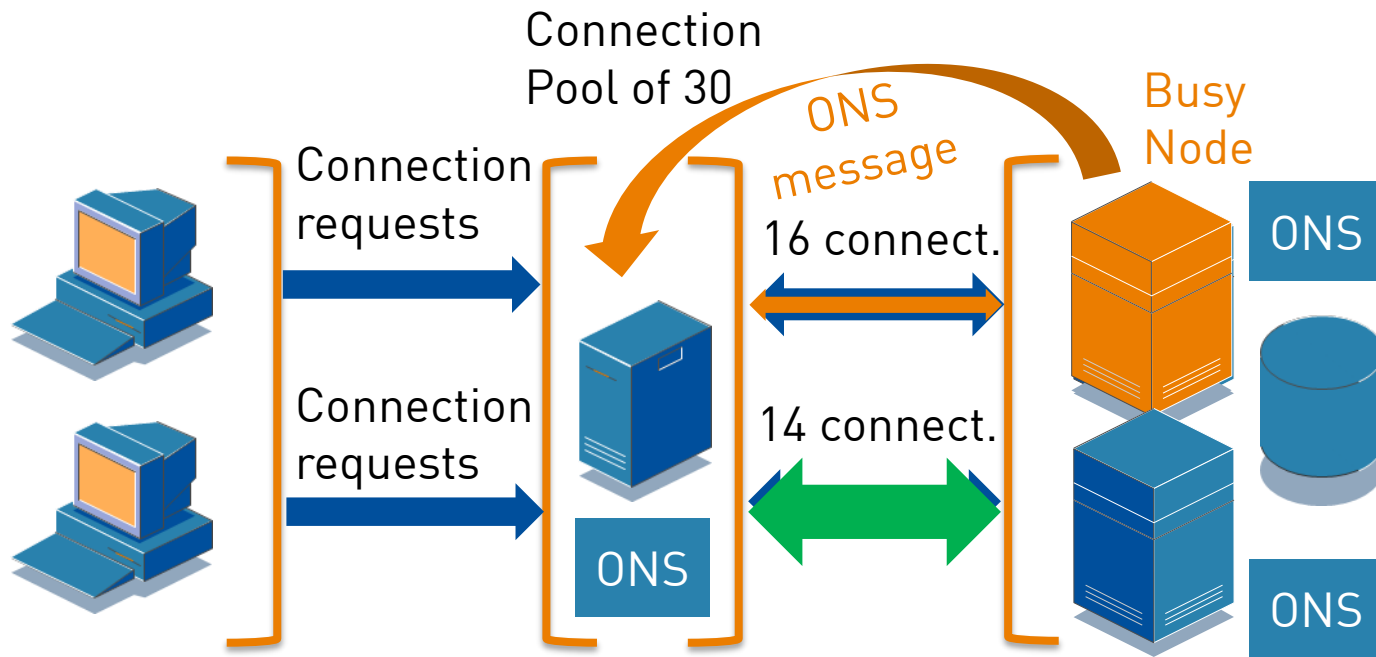


# Principles

## RLB

### Runtime Load Balancing

- > Load is distributed over RAC nodes. The connection pool will allocate connections at open time
- > Each “getConnection” accesses the **least busy** node based on ONS messages

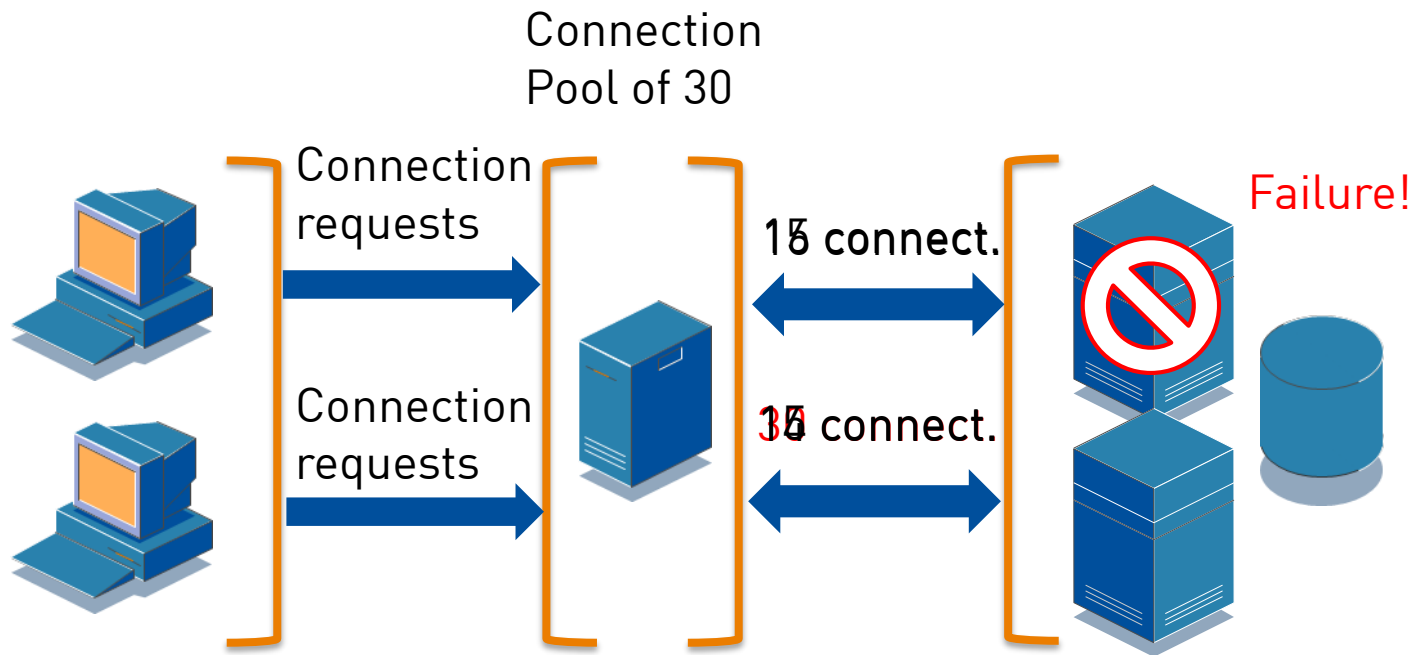


# Principles

## FCF

### Fast Connection Failover

- > After the failure of any node, the connection pool should be re-balanced over the remaining nodes



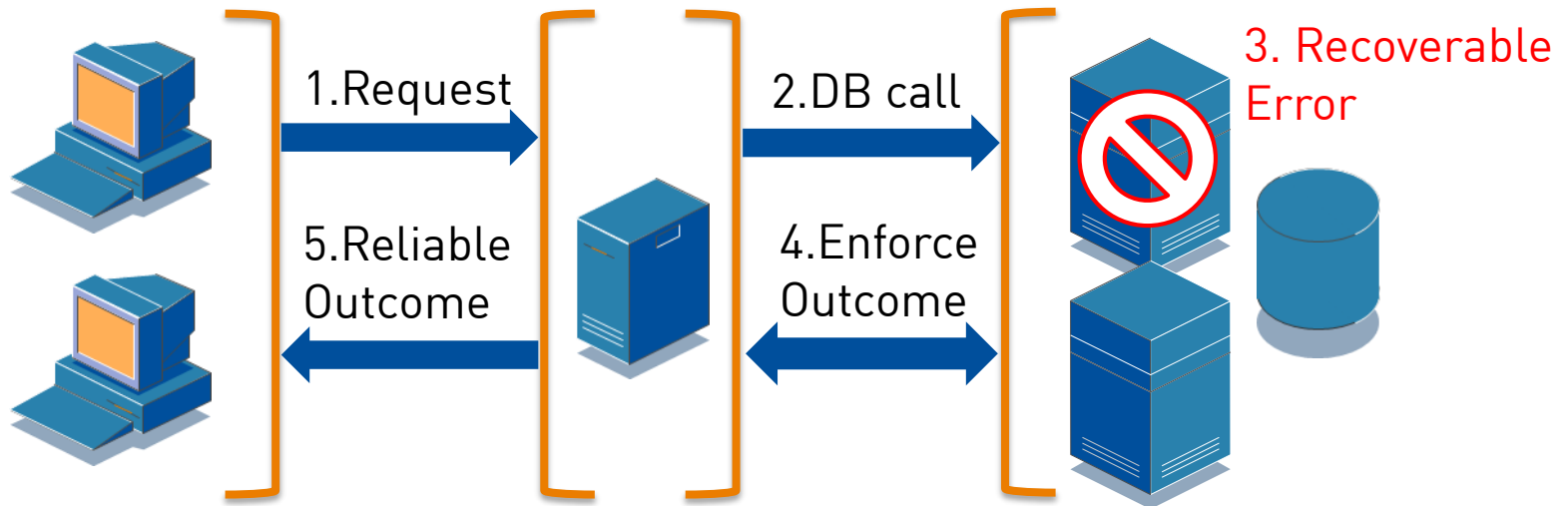
# Principles

## Transaction Guard

### Generic infrastructure for:

- > At-most-once : prevent for transaction double submission
- > Guaranteed outcome : during planned and unplanned outage

Transaction commit outcome (status) is tracked with logical ID

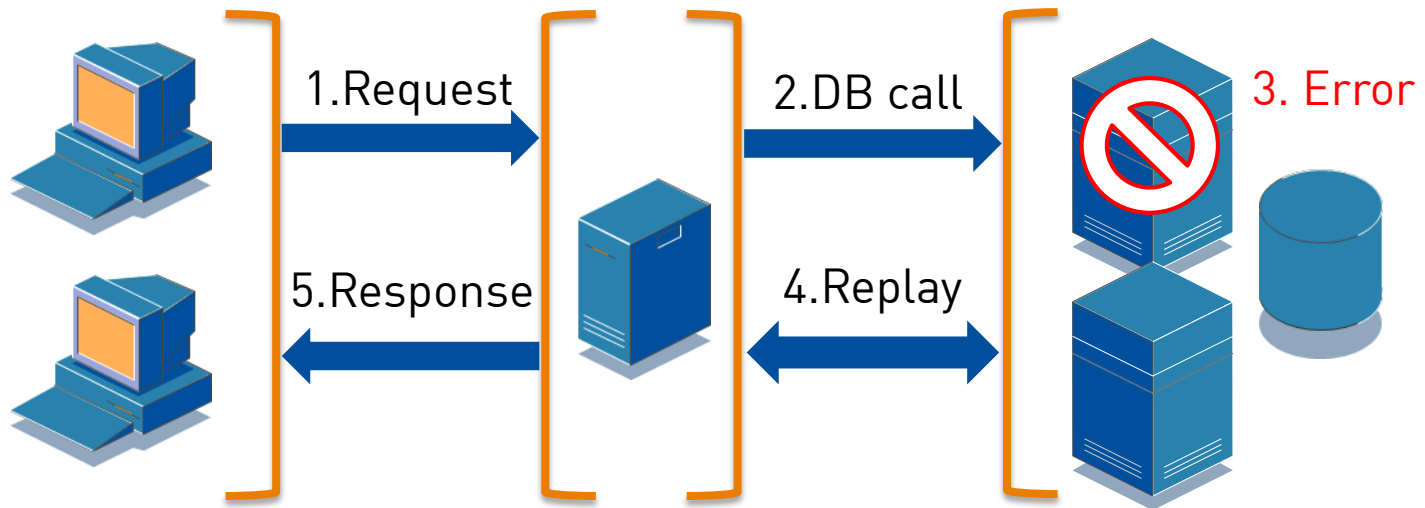


# Principles

## Application Continuity

### Moreover and conjointly with transaction guard

- > Provide transparent session recovery
- > Enable statements to be replayed
- > Allows to hide many errors (hardware, software, network, storage, ...) and outages from the user



# Setup



- > Database
- > Application Server - Weblogic

# Setup

## Fast Application Notification (FAN)

### Client side

- > ONS subscription
  - > Remotely (Java)
    - > Manually (ons.jar and ONS properties)
    - > Automatically through UCP
  - > Enable logging to see ONS and FAN events



# Setup

## Fast Application Notification (FAN)

### Server side (on each RAC node)

- > ONS daemon configuration
  - > Configuration file `${CRS_HOME}/opmn/config/ons.conf`
    - > Local port
    - > Remote port
    - > ONS nodes

```
localport=6100
remoteport=6200
nodes=vmtestrac12c1:6200,vmtestrac12c2:6200
```

# Setup

## Fast Application Notification (FAN)

A service must be setup on the DB server with following parameters

> NOTIFICATION = TRUE - for enabling ONS

On RAC service is created using srvctl

```
srvctl add service -db DBTEST -service testguard -notification TRUE
```

**DO NOT** use the database service (db\_name)  
reserved for OEM



# Setup

## Connection Load Balancing (CLB)

Must be configured depending on the application requirements

Following parameters must be set on the service

- > RLB\_GOAL = SERVICE\_TIME
- > CLB\_GOAL = SHORT

It defines how connection should be spread over the different nodes

```
srvctl modify service -db DBTEST -service testguard -clbgoal SHORT  
-rlbgoal SERVICE_TIME
```

# Setup

## Transaction Guard

### Following parameters must be set on the service

- > COMMIT\_OUTCOME = TRUE - for enabling Transaction Guard
- > RETENTION = 86400 – for specifying the commit outcome retention time in seconds

```
srvctl modify service -db DBTEST -service testguard  
-commit_outcome TRUE -retention 86400
```

# Setup

## Application Continuity

The service setup for Transaction Guard, on the DB server, must be updated with following additional parameters

- > FAILOVER\_TYPE = TRANSACTION - for using Application Continuity
- > REPLAY\_INITIATION\_TIMEOUT = 900 - for setting the duration in seconds for which replay will occur
- > FAILOVER\_RETRIES = 30 - for specifying the number of connection retries for each replay
- > FAILOVER\_DELAY = 10 - for specifying the delay in seconds between connection retries

```
srvctl modify service -db DBTEST -service testguard  
-failoverretry 30 -failoverdelay 10 -failovertype TRANSACTION  
-replay_init_time 900
```

# Setup

On single database instance, service is created using DBMS\_SERVICE package

```
declare params dbms_service.svc_parameter_array;
Begin
    params ('FAILOVER_TYPE') := 'TRANSACTION';
    params ('REPLAY_INITIATION_TIMEOUT') := 900;
    params ('RETENTION_TIMEOUT') := 86400;
    params ('FAILOVER_DELAY') := 10;
    params ('FAILOVER_RETRIES') := 30;
    params ('COMMIT_OUTCOME') := 'true';
    params ('AQ_HA_NOTIFICATIONS') := 'true';
    dbms_service.modify_service('testsrv',params);
end;
/
```

# Setup

## Application Continuity

### Oracle Weblogic Server 12c

> Following libraries must be installed on the server

Library file	Description
jdbc/lib/ojdbc7.jar	JDBC driver
ucp/lib/ucp.jar	Universal Connection Pool
opmn/lib/ons.jar	Oracle Network Server client
jlib/orai18n.jar	I18N support
jlib/orai18n-mapping.jar	I18N support
jlib/oraclepki.jar	Oracle Wallet support
jlib/osdt_cert.jar	Oracle Wallet support
jlib/osdt_core.jar	Oracle Wallet support

> Oracle 12c replay Data Source based on the JDBC replay driver

# Code sample

Insurance & Infrastructure  
at your service.





# Code sample

## Java

### Without Application Continuity facility

```
PoolDataSource pds=null;
try {
    pds = PoolDataSourceFactory.getPoolDataSource();
    pds.setConnectionFactoryClassName(
        "oracle.jdbc.pool.OracleDataSource");
    ...
}
```

### With Application Continuity facility

```
PoolDataSource pds=null;
try {
    pds = PoolDataSourceFactory.getPoolDataSource();
    pds.setConnectionFactoryClassName(
        "oracle.jdbc.replay.OracleDataSourceImpl");
    ...
}
```

# Benefits

Influence & Infrastructure  
at your service.



# Benefits

## Hide many errors and outages from the user

- > Error notified through FAN
- > Transparent reconnection through FCF

## Prevent end user mistakes

- > No double submission

## Recover in-flight transactions

- > If it's a recoverable exception



# Restrictions

Ingenuity & Infrastructure  
at your service.



# Restrictions

## These features are available on the following conditions

- > Only for Java application available through JDBC Thin (for now)
- > In conjunction with 12c driver
  - > Oracle Universal connection pool (UCP)
  - > Oracle WebLogic 12.1.2
  - > Oracle WebLogic Active Grid Link 12.1.2
- > Only supported on HA database features
  - > Oracle RAC
  - > Oracle Data Guard
  - > Oracle Active Data Guard



# Demo

---



Any questions? Please do ask.

**Alain Lacour**

Consultant

Mobile +41 79 199 96 44

[alain.lacour@dbi-services.com](mailto:alain.lacour@dbi-services.com)

[www.dbi-services.com](http://www.dbi-services.com)



We look forward to working with you!

# Backup slides



Intensity & Infrastructure  
at your service.



# Demo

## Table data verification

Application  
Continuity

Without Application  
Continuity

```
=====
LIST RECORDS IN SCOTT.EMP_REPLAY
=====

SQL> SELECT (*) FROM SCOTT.EMP_REPLAY

COUNT (*)
-----
          0

1 row selected.
```

```
=====
LIST RECORDS IN SCOTT.EMP_NOREPLAY
=====

SQL> SELECT (*) FROM SCOTT.EMP_NOREPLAY

COUNT (*)
-----
          0

1 row selected.
```

# Demo

## Launch test

Application  
Continuity

Without Application  
Continuity

```
=====
#----- dbi services Appetizer 12c -----#
#----- Application Continuity DEMO -----#
=====

Starting UCP with JDBC Replay Driver...
...(please wait)...
...
UCP started in 00.21.032
Retrieved connection from the Pool : com.sun.proxy.$
SetAutocomit : false !!!

#####
TRANSACTION PROCESS BEGIN :
#####

-----
Begin to insert Employee : 1
Elapsed time : 00.04.684
Employee 1 inserted !!

-----
Begin to insert Employee : 2
Elapsed time : 00.04.547
Employee 2 inserted !!

-----
Begin to insert Employee : 3
Elapsed time : 00.04.549
Employee 3 inserted !!
```

```
=====
#----- dbi services Appetizer 12c -----#
#----- Application Continuity DEMO -----#
=====

Starting UCP without JDBC Replay Driver...
...(please wait)...
...
UCP started in 00.21.512
Retrieved connection from the Pool : com.sun.proxy.$
SetAutocomit : false !!!

#####
TRANSACTION PROCESS BEGIN :
#####

-----
Begin to insert Employee : 1
Elapsed time : 00.04.708
Employee 1 inserted !!

-----
Begin to insert Employee : 2
Elapsed time : 00.04.547
Employee 2 inserted !!

-----
Begin to insert Employee : 3
Elapsed time : 00.04.546
Employee 3 inserted !!
```

# Demo

## Stop one instance

Application  
Continuity

Without Application  
Continuity

```
Begin to insert Employee : 18
Elapsed time : 00.04.549
Employee 18 inserted !!
-----
Begin to insert Employee : 19
Elapsed time : 00.04.550
Employee 19 inserted !!
-----
Begin to insert Employee : 20
Elapsed time : 00.04.548
Employee 20 inserted !!
-----
Begin to insert Employee : 21
Elapsed time : 00.04.740
Employee 21 inserted !!
-----
Begin to insert Employee : 22
Elapsed time : 00.04.548
Employee 22 inserted !!
-----
Begin to insert Employee : 23
```

```
-----
Begin to insert Employee : 33
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.001
Employee 33 inserted !!
-----
Begin to insert Employee : 34
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.002
Employee 34 inserted !!
-----
Begin to insert Employee : 35
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.001
Employee 35 inserted !!
-----
#####
TRANSACTION PROCESS END : Commit will be performed
#####
-----
java.sql.SQLException: The connection is closed: The
```

# Demo

## Application reconnect and replay

Application  
Continuity

Without Application  
Continuity

```
Begin to insert Employee : 23
Elapsed time : 01.25.702
Employee 23 inserted !!
-----
Begin to insert Employee : 24
Elapsed time : 00.04.547
Employee 24 inserted !!
-----
Begin to insert Employee : 25
Elapsed time : 00.04.547
Employee 25 inserted !!
-----
Begin to insert Employee : 26
Elapsed time : 00.04.547
Employee 26 inserted !!
-----
Begin to insert Employee : 27
Elapsed time : 00.04.548
Employee 27 inserted !!
-----
```

```
Begin to insert Employee : 33
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.001
Employee 33 inserted !!
-----
Begin to insert Employee : 34
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.002
Employee 34 inserted !!
-----
Begin to insert Employee : 35
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.001
Employee 35 inserted !!
-----
#####
TRANSACTION PROCESS END : Commit will be performed
#####
java.sql.SQLException: The connection is closed: The
```

# Demo

At the end, commit is performed

Application  
Continuity

Without Application  
Continuity

```
Begin to insert Employee : 32
Elapsed time : 00.04.547
Employee 32 inserted !!
-----
Begin to insert Employee : 33
Elapsed time : 00.04.548
Employee 33 inserted !!
-----
Begin to insert Employee : 34
Elapsed time : 00.04.550
Employee 34 inserted !!
-----
Begin to insert Employee : 35
Elapsed time : 00.04.549
Employee 35 inserted !!
-----
#####
TRANSACTION PROCESS END : Commit will be performed
#####
----- Transaction committed -----
=====
#----- END -----#
=====
```

```
-----
Begin to insert Employee : 33
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.001
Employee 33 inserted !!
-----
Begin to insert Employee : 34
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.002
Employee 34 inserted !!
-----
Begin to insert Employee : 35
java.sql.SQLException: The connection is closed: The
Elapsed time : 00.00.001
Employee 35 inserted !!
-----
#####
TRANSACTION PROCESS END : Commit will be performed
#####
-----
java.sql.SQLException: The connection is closed: The
G:\Soft\dbi\ApplicationContinuity\dbiInfra>
```

# Demo

## Table data verification

Application  
Continuity

```
=====
LIST RECORDS IN SCOTT.EMP_REPLAY
=====
SQL> SELECT (*) FROM SCOTT.EMP_REPLAY

COUNT (*)
-----
          35

1 row selected.
```

Without Application  
Continuity

```
=====
LIST RECORDS IN SCOTT.EMP_NOREPLAY
=====
SQL> SELECT (*) FROM SCOTT.EMP_NOREPLAY

COUNT (*)
-----
          0

1 row selected.
```