

**Oracle  
Data Guard Observer  
Der vergessene Helfer**

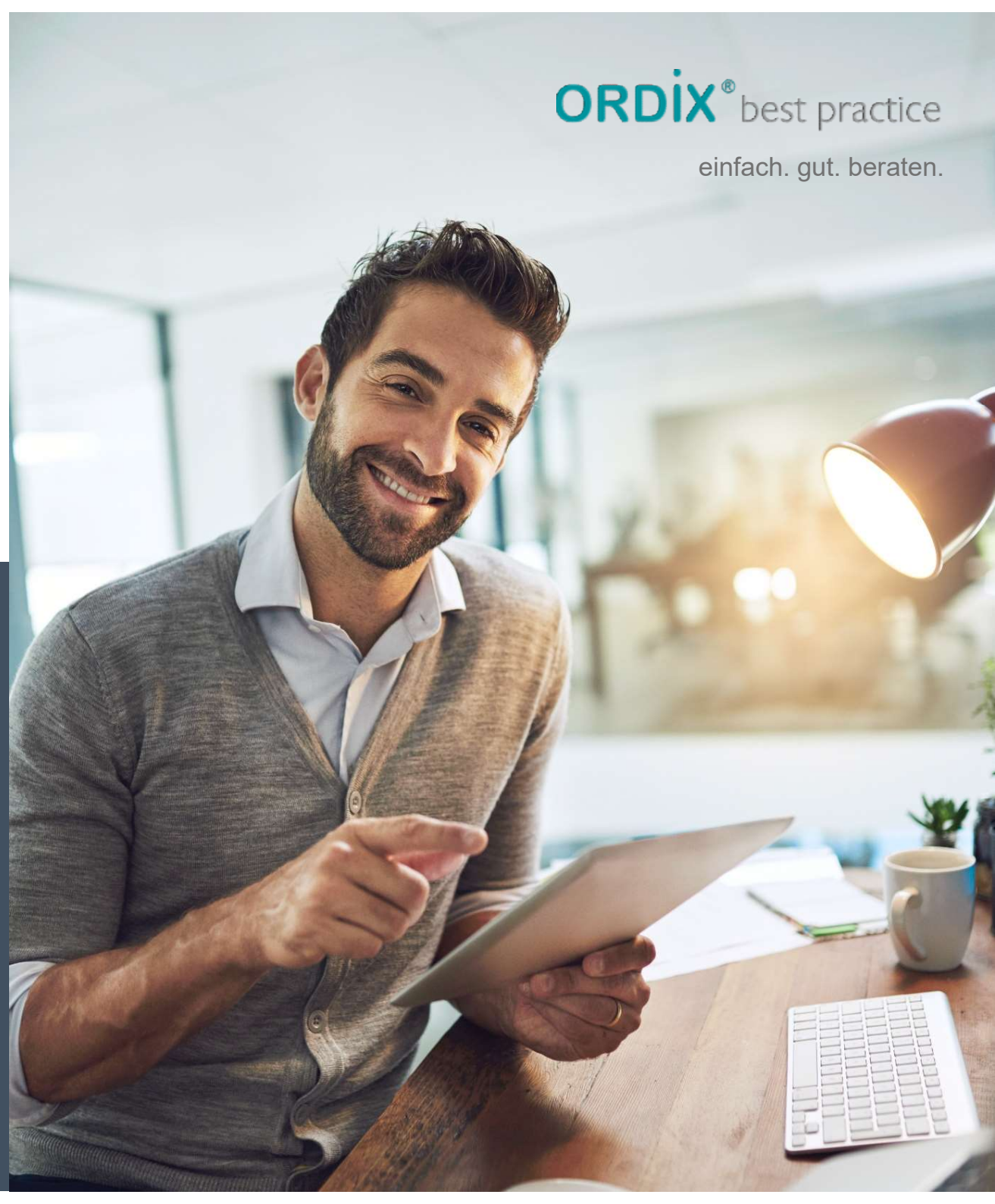
DOAG  
Konferenz + Ausstellung 2018  
Nürnberg  
Dirk Krautschick

[info@ordix.de](mailto:info@ordix.de)

[www.ordix.de](http://www.ordix.de)

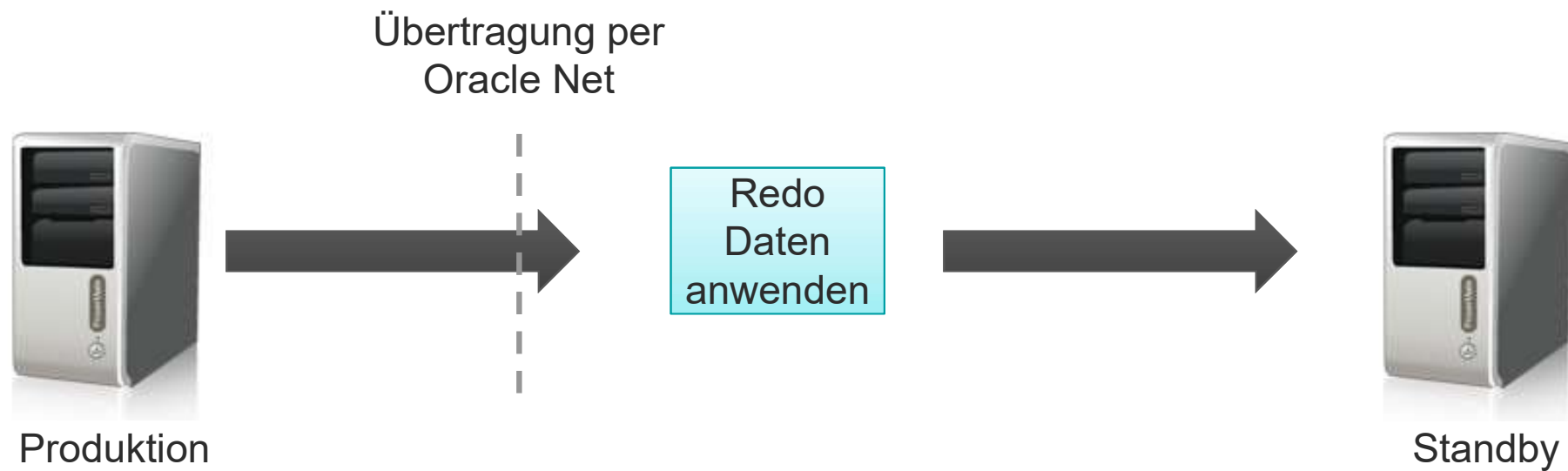
**ORDIX**<sup>®</sup> best practice

einfach. gut. beraten.



- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

- Prinzip
  - Übertragung und Nachfahren von Redo-Informationen
  - Echtzeit (Realtime Apply) oder mit Verzögerung



- Failover erfordert manuellen Eingriff durch den DBA
- Abhängigkeit von 24/7 Rufbereitschaft
- Auslöser
  - Problem-Meldung von User
  - Monitoring
- Je nach Situation lange Ausfallzeiten
  - Hochverfügbarkeit
  - Service Level Agreements

# Failover

- Kein Failover bei manuellem
  - shutdown (normal)
  - shutdown transactional
  - shutdown immediate
  
- automatischer Failover?
  - shutdown abort
  - Instanz-Fehler
  - Offline-Datenbank-Dateien
  - Netzwerk-Verbindungsabbruch
    - zwischen Clients und Produktions-Datenbank

- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

# Fast-Start Failover

- Automatisierung von Failover durch
  - Schwellwerte
  - Bedingungen
- Automatischer Reinstat
- Einführung mit Oracle Database 10 Release 2
- Failover
  - Entscheidung?
  - Auslöser?
  - Wo?

- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

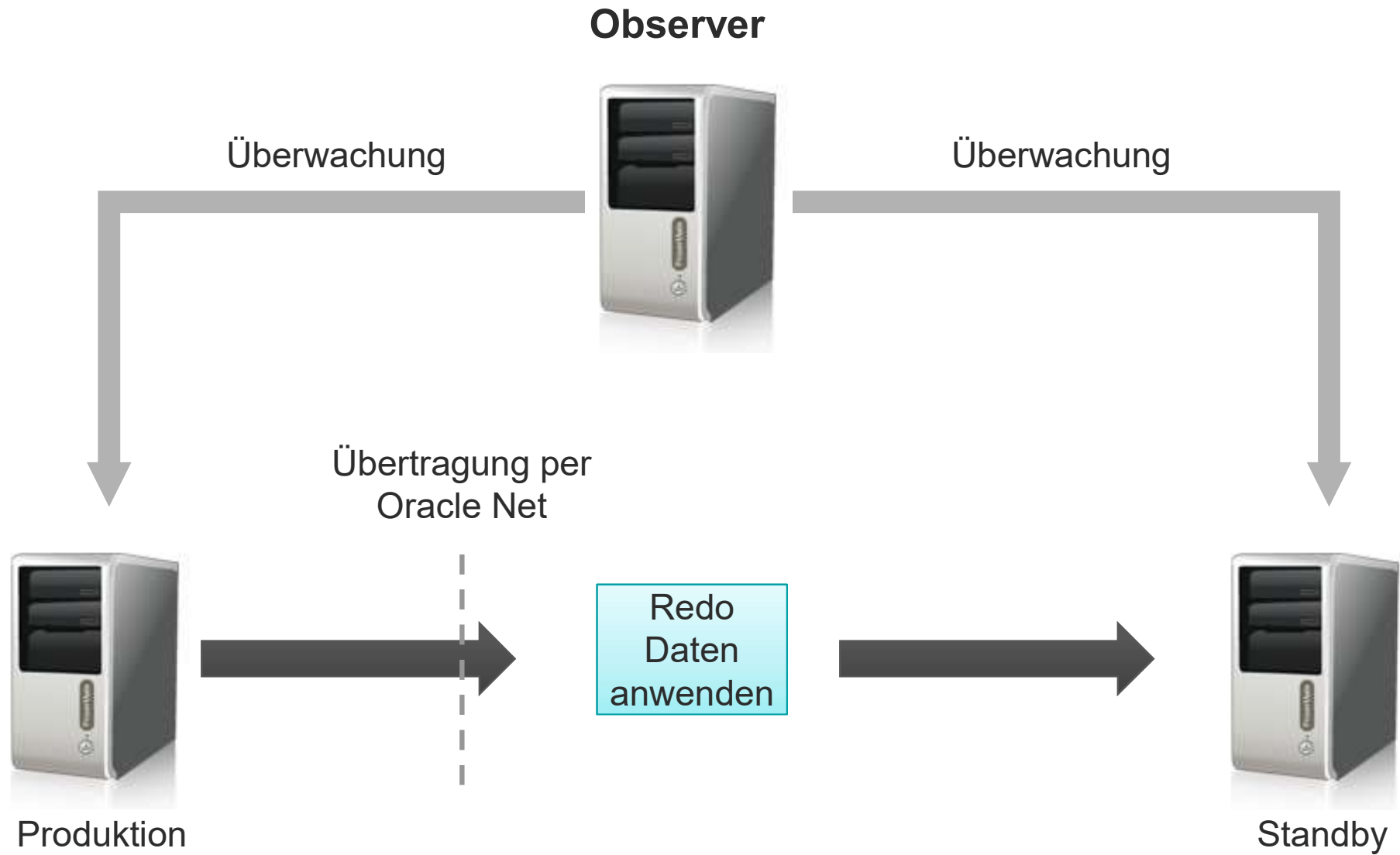


# Oracle Data Guard Observer

- Observer
  - „unabhängige“ Überwachung
  - Kontinuierlicher Vordergrundprozess



# Oracle Data Guard Observer



# Observer - Voraussetzungen

- Für die Datenbanken
  - Protection Mode
    - Maximum Availability
    - Maximum Performance
    - Maximum Protection (ab 12.2)
  - Flashback Database
  
- Für den Observer
  - Installierter Oracle Administrator Client
  - Oracle Net Konfiguration

## Observer - starten/stoppen

- Broker
  - DGMGRL> start observer
  - DGMGRL> stop observer
  
- Konsole
  - mit nohup im Hintergrund
  - nohup dgmgrl  
sys/password@<tns\_prod>  
"start observer  
file='/home/oracle/observer\_config\_file.dat';"  
> /home/oracle/observer.log 2>&1 &
  
- DAT File
  - Konfigurations- und Status-Informationen

## Observer - starten/stoppen

- Security?
- Rollentrennung
  - SYS vermeiden
  - Bevorzugt Data Guard Administrator SYSDG
- Password in der Konsole auslesbar
  - Abhilfe mit External Password Store

## Observer - starten/stoppen

```
mkstore -wrl /opt/oracle/admin/wallet -create
```

```
mkstore -wrl /opt/oracle/admin/wallet -createCredential orcl sys password
```

- sqlnet.ora

```
SQLNET.WALLET_OVERRIDE = TRUE
```

```
WALLET_LOCATION =
```

```
  (SOURCE =
```

```
    (METHOD = FILE)
```

```
    (METHOD_DATA =
```

```
      (DIRECTORY = /opt/oracle/admin/wallet)
```

```
    )
```

```
  )
```

```
nohup dgmgrl /@orcl "start observer" &
```

## Observer - Herausforderung

- potentieller Single Point of Failure
- Zusätzliches Monitoring der Observer-Server
- Wahl der Location
  - Split-Brain-Gefahr
  - Kommunikationsweg identisch zu Clients
- Redundante Observer

- Multiple Observer (ab 12.2)
  - bis zu 3 Observer möglich
  - ein Master Observer
  - maximal zwei Backup Observer

```
connect sys/password@server1 → start observer OBSERVER1;
```

```
connect sys/password@server2 → start observer OBSERVER2;
```

```
connect sys/password@server3 → start observer OBSERVER3;
```

...

```
connect sys/password@server4 → start observer OBSERVER4;
```

```
Error: ORA-16647: could not start more than three observers
```



## Observer - Hochverfügbarkeit

- Master wird automatisch gewählt
- Wechsel mit: **SET MASTEROBSERVER TO observer3;**
- Master Observer lässt sich nicht beenden, wenn Backup Observer existieren  
**ORA-16878: master observer cannot be stopped**

# Observer - Hochverfügbarkeit

```
DGMGRL> show observer
```

```
Configuration - orcl_dg_config
```

```
Primary:          orcl_stby
```

```
Active Target:    orcl
```

```
Observer „server1" - Master
```

```
Host Name:        server1
```

```
Last Ping to Primary: 1 seconds ago
```

```
Last Ping to Target:  2 seconds ago
```

```
Observer "observer_backup" - Backup
```

```
Host Name:        server2
```

```
Last Ping to Primary: 0 second ago
```

```
Last Ping to Target:  3 second ago
```

```
Observer "observer_nocheinbackup" - Backup
```

```
Host Name:        server3
```

```
Last Ping to Primary: 2 seconds ago
```

```
Last Ping to Target:  2 seconds ago
```

- Überwachung/Steuerung von mehreren Konfigurationen (FSFO) (ab 12.2)
  - observer.ora
  - DGMGRL> set observerconfigfile='/u01/app/oracle/observer.ora';
  - DGMGRL> show observerconfigfile;  

```
ObserverConfigFile=/u01/app/oracle/observer.ora
```
- tnsnames.ora auf Observer Host mit Alias
- External Password Store notwendig
- BROKER\_CONFIGS
  - Definition der verschiedenen Konfigurationen
- CONFIG GROUPS
  - Optional
  - Gruppendifinition mit mindestens einer Broker Konfiguration

# Observer - Multiple Konfigurationen

```
BROKER_CONFIGS = (  
  (CONFIG = (NAME=PROD1)  
    (CONNECT_ID=dgconfig1)  
    (config_home=/u01/app/oracle/))  
  (CONFIG = (NAME=PROD2)  
    (CONNECT_ID=dgconfig2)  
    (config_home=/u01/app/oracle/)))  
CONFIG_GROUPS = (  
  (GROUP=(NAME=A) (CONFIG_LIST = (NAME=PROD1) (NAME=PROD2)))  
  (GROUP=(NAME=B) (CONFIG_LIST = (NAME=PROD1)))  
  (GROUP=(NAME=C) (CONFIG_LIST = (NAME=PROD2))))
```

- CONNECT\_ID = Primary und alle FSFO Targets (tnsnames.ora)

```
DGMGRL> start observing A
```

```
ObserverConfigFile=observer.ora
```

```
observer configuration file parsing succeeded
```

```
Submitted command "START OBSERVER" using connect identifier  
„dgconfig1„
```

```
Submitted command "START OBSERVER" using connect identifier  
„dgconfig2"
```

- **MOS 2285891.1**

- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

- FastStartFailoverTarget definieren

```
edit database '<tns_alias_prod>'
```

```
  set property FastStartFailoverTarget= '<tns_alias_stdby>';
```

```
edit database '<tns_alias_stdby>'
```

```
  set property FastStartFailoverTarget='<tns_alias_prod>'
```

```
enable fast_start failover;
```

```
DGMGRL> show configuration
```

```
Configuration - orcl_dg_config
```

```
Protection Mode: MaxPerformance
```

```
Members:
```

```
orcl - Primary database
```

```
orcl_stby - (*) Physical standby database
```

```
Fast-Start Failover: ENABLED
```

```
Configuration Status:
```

```
SUCCESS (status updated 93 seconds ago)
```



- Multiple FastStartFailoverTarget (ab 12.2)

```
edit database ,orcl'  
  set property FastStartFailoverTarget= ,orcl_stby1, orcl_stby2';
```

```
edit database `orcl`  
  set property FastStartFailoverTarget='ANY';
```

```
show fast_start failover;
```

```
Fast-Start Failover: ENABLED
```

```
Threshold: 30 seconds
```

```
Target: orcl_stby1
```

```
Candidate Targets: orcl_stby1, orcl_stby2
```

```
Observer: server3.localdomain
```

```
...
```

```
DGMGRL> show fast_start failover
```

```
Fast-Start Failover: ENABLED
```

```
Threshold:          30 seconds  
Target:             standby  
Observer:           dataguard1  
Lag Limit:          30 seconds (not in use)  
Shutdown Primary:  TRUE  
Auto-reinstate:     TRUE
```

...

```
Beispiel: EDIT CONFIGURATION SET PROPERTY FastStartFailover Threshold=45;
```

- Bei Verbindungsabbruch 30s Verzögerung bis Failover
- Automatisches Reinstate wenn primäre DB wieder erreichbar

- Weitere Properties

  - `LagLimit`

  - `ShutdownPrimary`

  - `ObserverReconnect`

  - `ObserverOverride`

  - `OperationTimeout`

  - `TraceLevel`

  - `CommunicationTimeout`

  - `BystandersFollowRoleChange`

  - `PrimaryLostWriteAction`

# Konfiguration - Failover Conditions

```
DGMGRL> show fast_start failover
```

...

```
Configurable Failover Conditions
```

```
Health Conditions:
```

Corrupted Controlfile	YES
Corrupted Dictionary	YES
Inaccessible Logfile	NO
Stuck Archiver	NO
Datafile Offline	YES

...

```
Beispiel: ENABLE FAST_START FAILOVER CONDITION "Stuck Archiver";
```

- Beim Auftreten kein Treshold bis Failover
- Kein Automatisches Reinstat

# Konfiguration - Failover Error Conditions

```
DGMGRL> show fast_start failover
```

...

```
Oracle Error Conditions:
```

```
ORA-27102: out of memory
```

Beispiel

```
ENABLE FAST_START FAILOVER CONDITION 27102;
```

- **V\$DATABASE**
  - Falls kein DGMGRL zur Hand

```
select
  DBID, NAME, DB_UNIQUE_NAME, DATABASE_ROLE,
  SWITCHOVER_STATUS, DATAGUARD_BROKER, GUARD_STATUS,
  FS_FAILOVER_STATUS,
  FS_FAILOVER_CURRENT_TARGET,
  FS_FAILOVER_THRESHOLD,
  FS_FAILOVER_OBSERVER_PRESENT,
  FS_FAILOVER_OBSERVER_HOST,
  ...
from v$database;
```

- **V\$FS\_FAILOVER\_STATS**

- Zeitpunkt und Grund für Failover

```
SQL> select * from v$fs_failover_stats;
```

LAST_FAILOVER_TIME	LAST_FAILOVER_REASON	CON_ID
11/12/2018 11:32:28	Test Failover	0
11/14/2018 11:32:28	Application Failover Requested	0



- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

- Verwendung von Service Names und mehreren Connection Strings

```
ORCL =  
  (DESCRIPTION =  
    ADDRESS = (PROTOCOL = TCP) (HOST = server1) (PORT = 1521)  
    ADDRESS = (PROTOCOL = TCP) (HOST = server2) (PORT = 1521)  
  (CONNECT_DATA) =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = APP_ORCL)  
  )  
)
```

- Startup Trigger für Service Name Wechsel

```
create or replace trigger manage_OCIService after startup on database
declare
    role varchar(39);
    dg_start varchar(10);
begin
    select database_role into role from v$database;
    select value into dg_start from v$parameter
        where name = 'dg_broker_start';
    if     role = 'PRIMARY' AND dg_start = 'TRUE'
    then  dbms_service.start_service('APP_ORCL ');
    else  dbms_service.stop_service('APP_ORCL ');
    end if;
END;
```

- Auslösen von Failover durch die Anwendung
  - `DBMS_DG.INITIATE_FS_FAILOVER`
  
- SYSDBA Privileg
  
- Syntax
  - `DBMS_DG.INITIATE_FS_FAILOVER ( condstr IN VARCHAR2) RETURN BINARY_INTEGER;`
  - condstr
    - Ursache für geforderten Failover
    - Anzeige in Broker bzw. alert.log
    - Default: „Application Failover Requested“

- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

- Ändern von
  - Protection Modus
  - LogXptMode der Produktions- oder Standby Datenbank
  - Die FastStartFailoverTarget Eigenschaft der Produktions- oder Standby-Datenbank
  
- Deaktivieren oder Löschen von
  - Der Broker-Konfiguration
  - Der Standby-Datenbank, die das Ziel des Fast-Start Failover ist
  
- Manueller Failover ist nur unter folgenden Bedingungen möglich
  - Ziel Datenbank ist das Fast-Start Failover Target
  - Standby Datenbank ist synchron mit der Primary
  - Observer ist gestartet, Verbindung zur Standby Datenbank besteht

- Oracle Data Guard
- Fast-Start Failover
- Data Guard Observer
- Konfiguration
- FSFO und Applikationen
- Einschränkungen
- Fazit

## Fazit

- Minimale Downtime durch automatischen Failover
- Geringere Fehleranfälligkeit gegenüber manuellen Aktionen
- Konfiguration muss wohlüberlegt sein
  - Testen von Tresholds
  - Strategien
- Datenbankadministratoren geben Kontrolle ab



**Vielen Dank für  
Ihre Aufmerksamkeit!**

**ORDIX AG**

Zentrale Paderborn  
Karl-Schurz-Straße 19a  
33100 Paderborn  
Tel.: 05251 1063-0  
Fax: 0180 1 67349 0

Seminarzentrum Wiesbaden  
Kreuzberger Ring 13  
65205 Wiesbaden  
Tel.: 0611 77840-00

Weitere Geschäftsstellen  
in Essen, Gersthofen,  
Köln und Münster

[info@ordix.de](mailto:info@ordix.de)  
[www.ordix.de](http://www.ordix.de)