

Failover-Betrieb für PostgreSQL-Datenbanken



Installationsanleitung für Systembetreiber

25.03.2020

Originalanleitung

Produktlinie neo, Version 6.x

Die beschriebenen Funktionen können mit folgenden ASC-Produkten verwendet werden:

EVOIPneo

EVOLUTIONneo / XXL / eco

EVOflex (länderspezifisch)

Im Partnerbereich unserer Webseite <http://www.asctechnologies.com> finden Sie immer die aktuellsten technischen Dokumente und Produktaktualisierungen.

Copyright © 2019 ASC Technologies AG. Alle Rechte vorbehalten.

Windows ist ein eingetragenes Markenzeichen der Microsoft Corporation. VMware® ist ein eingetragenes Markenzeichen von VMware, Inc. Alle anderen hier erwähnten Marken und Produktnamen sind das Eigentum ihrer jeweiligen Inhaber.

Inhaltsverzeichnis

1	Allgemeine Hinweise	5
2	Einleitung	6
3	Anmerkungen	7
4	Installation der Datenbanken	8
4.1	Primär-Datenbank	8
4.2	Standby-Datenbank	11
5	Failover-Betrieb konfigurieren	15
5.1	Voraussetzungen	15
5.1.1	Netzwerkfreigabe	15
5.1.2	Postgres-Dienst unter dem PostgreSQL-Konto starten	16
5.2	Failover-Betrieb mit Failover Configuration Tool konfigurieren	17
5.2.1	Weiteren Applikationsserver hinzufügen	21
5.3	Failover-Betrieb manuell konfigurieren	22
5.3.1	Konfiguration der Datenbankserver	22
5.3.1.1	Primär-Datenbank einrichten	22
5.3.1.2	Standby-Datenbank einrichten	26
5.3.1.3	Datei setup.xml anpassen	28
5.3.2	Konfiguration der Applikationsserver	29
5.3.2.1	Datei setup.xml anpassen	29
5.3.2.2	Datenbankverbindung konfigurieren	29
5.4	Funktionstest	34
5.4.1	Replikation auf dem Primär-Server kontrollieren	34
5.4.2	Replikation auf dem Standby-Server kontrollieren	34
5.4.3	Netzwerkfreigabe kontrollieren	34
5.4.4	Löschen der WAL-Archive kontrollieren	34
6	Failover-Betrieb Funktionsbeschreibung	36
6.1	Ausfall der Primär-Datenbank	36
6.2	Ausfall der Standby-Datenbank	37
7	Failover-Betrieb zurücksetzen	38
7.1	Failover-Betrieb manuell zurücksetzen	38
7.1.1	Standby-Server wiederherstellen	39
7.2	Failover-Betrieb mit Failover Configuration Tool zurücksetzen	40
7.2.1	Failover-Datenbank als primäre Datenbank übernehmen	40
8	Anhang	42
8.1	Datei pg_hba.conf, Primär-Server	42
8.2	Datei pg_hba.conf, Standby-Server	42
8.3	Datei postgresql.conf, Primär-Server	43
8.4	Datei postgresql.conf, Standby-Server	43

8.5	Datei recovery.conf, Standby-Server	44
8.6	Datei setup.xml, Datenbankserver	44
8.7	Datei setup.xml, Applikationsserver	44
8.8	Datei domain.xml, Applikationsserver	45
	Abbildungsverzeichnis.....	47
	Tabellenverzeichnis	48
	Glossar	49

Allgemeine Hinweise

ASC steht im Kontext dieses Dokuments für die ASC Technologies AG, deren Tochtergesellschaften, Niederlassungen und Vertriebsbüros. Deren aktuelle Übersicht kann auf der Webseite unter <https://www.asctechnologies.com> eingesehen werden.

ASC übernimmt keinerlei Gewähr für die Aktualität, Korrektheit, Vollständigkeit oder Qualität der in den Anleitungen bereitgestellten Informationen.

ASC kontrolliert regelmäßig den Inhalt der veröffentlichten Anleitungen auf Übereinstimmung mit der beschriebenen Hard- und Software. Dennoch können Abweichungen nicht ausgeschlossen werden. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Einige Aspekte der ASC-Technologie werden in allgemeiner Form beschrieben, um das Eigentum und die vertraulichen Informationen und/oder Geschäftsgeheimnisse von ASC zu schützen.

Die Softwareprogramme und Anleitungen von ASC sind urheberrechtlich geschützt. Alle Rechte an den Anleitungen sind vorbehalten, auch die der Reproduktion und/oder Vervielfältigung in jeglicher Form, sei es fotomechanisch, drucktechnisch oder auf digitalen Datenträgern. Dies gilt auch für Übersetzungen. Nachdruck der Anleitungen, vollständig oder auszugsweise, ist nur mit schriftlicher Genehmigung von ASC gestattet.

Maßgebend ist, soweit nicht anders angegeben, der technische Stand zum Zeitpunkt der Auslieferung von Software, Geräten und Anleitungen durch ASC. Technische Änderungen ohne gesonderte Ankündigung bleiben vorbehalten. Bisherige Anleitungen verlieren ihre Gültigkeit.

Es gelten die Allgemeinen Verkaufs- und Lieferbedingungen von ASC in ihrer jeweils gültigen Fassung.

Zur Absicherung des Zugriffs auf die Aufzeichnungen bei einem Ausfall der Datenbank können Sie eine Failover-Funktion mit einer weiteren Datenbank einrichten.

Diese Anleitung beschreibt, wie Sie ein Failover-Konzept mit zwei PostgreSQL-Datenbanken einrichten und welche Schritte Sie durchführen müssen, um den Failover-Betrieb zurückzusetzen, wenn die Primär-Datenbank wieder zur Verfügung steht.

Die Failover-Funktion ist sowohl für [Single-Core-Systeme](#) als auch für [Multi-Core-Systeme](#) verfügbar.

Sie können die Konfiguration sowohl mit IPv4 als auch mit IPv6 IP-Adressen durchführen.

Die beiden PostgreSQL-Datenbanken müssen auf unterschiedlichen Servern installiert sein. Sie können dabei wahlweise auf Applikationsservern ([App-Servern](#)) oder beliebigen anderen Servern im Aufzeichnungssystem installiert sein. Auf welchen Servern in Ihrem Aufzeichnungssystem die Datenbanken installiert werden, geben Sie dem Aufzeichnungssystem während der Installation der Aufzeichnungssoftware bekannt.



Informationen zur Installation der *neo*-Software finden Sie in der Installationsanleitung für Systembetreiber *Installation der Aufzeichnungssoftware von ASC*.

Nachdem Sie die beiden Datenbanken installiert und den Failover-Betrieb konfiguriert haben, steht Ihnen in der Applikation System Configuration das Datenbank-Manager-Modul zur Verfügung. Mit diesem Modul können Sie den Failover-Betrieb überwachen.



Informationen zum Datenbank-Manager-Modul finden Sie in der Administrationsanleitung für Systembetreiber *Datenbank Manager*.



In dieser Anleitung werden die Begriffe Primär-Server und Standby-Server im folgendem Sinn verwendet:

Primär-Server = Master-Server, auf dem die Primär-Datenbank liegt

Standby-Server = Slave-Server, auf dem die Standby-Datenbank liegt

3

Anmerkungen

In dieser Anleitung werden manuell durchzuführende Konfigurationsänderungen beschrieben. Für IP-Adressen, Verzeichnisse, usw. werden Platzhalter verwendet. Diese Platzhalter müssen Sie mit den Werten der konkreten Systemumgebung ersetzen.



Ersetzen Sie dabei nicht nur den Text zwischen den spitzen Klammern, sondern den kompletten Platzhalter!

Z. B. aus *host replication postgres <<IP-ADDRESS-STANDBY-DB>>/<<CIDR>> md5* wird *host replication postgres 192.168.0.0/32 md5*.

Sie können die Konfiguration auch mit IPv6 IP-Adressen durchführen.



Generell wird die hexadezimale Darstellung verwendet, z. B.

FE80::29ED:F6FB:D58D:F92B/128md5

An Stellen, wo eine Pfadangabe erforderlich ist, verwenden Sie folgende Schreibweise

FE80--29ED-F6FB-D58D-F92B.ipv6-literal.net

Die Ausnahmefälle gelten nur für die manuelle Konfiguration und sind explizit gekennzeichnet.

Folgende Platzhalter werden verwendet:

Platzhalter	Bedeutung
<<IP-ADDRESS-PRIMARY-DB>>	IP-Adresse der Primär-Datenbank
<<IP-ADDRESS-STANDBY-DB>>	IP-Adresse der Standby-Datenbank
<<WAL-ARCHIVE>>	Name des Replikationsverzeichnisses für die Ablage der WAL-Archive (Default: <i>WAL-Archive</i>)
<<WAL-SEGMENTS>>	Anzahl der WAL -Segmente, die von der Primär-Datenbank vorgehalten werden.
<<CIDR>>	Wert für die Netzmaske bei Nutzung von Classless Inter-Domain Routing (CIDR).
<<POSTGRES-INSTALL-FOLDER>>	Laufwerk und Pfad des PostgreSQL-Installationsverzeichnisses (Default: <i>C:\Program Files\PostgreSQL\n.m</i> HINWEIS! <i>n.m</i> steht für die Versionsnummer der verwendeten PostgreSQL, z. B. <i>C:\Program Files\PostgreSQL\9.5</i> .)
<<POSTGRES-DATA-FOLDER>>	Laufwerk und Pfad des PostgreSQL-Datenverzeichnisses (Default: <i>D:\ASCDB</i>)
<<POSTGRES-DATA-DRIVE>>	Laufwerk des PostgreSQL-Datenverzeichnisses (Default: <i>D:</i>)
<<IP-ADDRESS-NEO-CORE>>	IP-Adresse eines neo -Applikationsservers
<<NEO-INSTALL-FOLDER>>	Laufwerk und Pfad des Installationsverzeichnis der neo Suite (Default: <i>C:\Program Files (x86)\ASC\ASC Product Suite</i>)
<<DELAY>>	Dauer, die bei einem Ausfall der Verbindung zum Primär-Server gewartet wird, bis das Failover zum Standby-Server erfolgt. Einheit: Millisekunden

4 Installation der Datenbanken



Stellen Sie sicher, dass Sie für die Installation die vollen Administratorrechte besitzen.



Sie müssen für beide Datenbanken die gleiche PostgreSQL-Version verwenden.



Für die Kommunikation zwischen der Primär- und der Standby-Datenbank und allen neo-Applikationsservern muss in der Firewall folgender Port geöffnet sein:

- Port 5432



Im Betriebssystem beider Datenbankserver muss die Datei- und Druckerfreigabe aktiviert sein.

4.1 Primär-Datenbank

Die Primär-Datenbank kann entweder als lokale Datenbank auf einem neo-Applikationsserver oder als externe Datenbank auf einem separaten Server installiert werden.



Bei der Installation von Multi-Server-Systemen muss der Server, auf dem die Datenbank laufen soll, zuerst installiert werden.

Bei Verwendung von Failover-Datenbanken muss der Server, auf dem die Primär-Datenbank laufen soll, zuerst installiert werden.

Die Installation erfolgt im Rahmen der Installation der ASC-Software.

1. Legen Sie das Installationsmedium für die ASC-Software ein.
2. Wählen Sie aus dem Kontextmenü der Datei *setup.exe* den Menüpunkt *Run as administrator*.

3. Übernehmen Sie in allen Installationsschritten die Standardeinstellungen, bis Sie zur Installation der Datenbank kommen.
4. Aktivieren Sie das Feature *Applikation Server*, wenn Sie alle Features auf diesem Server nutzen möchten.
5. Aktivieren Sie das Feature *Datenbank*, um eine interne PostgreSQL-Datenbank zu installieren.

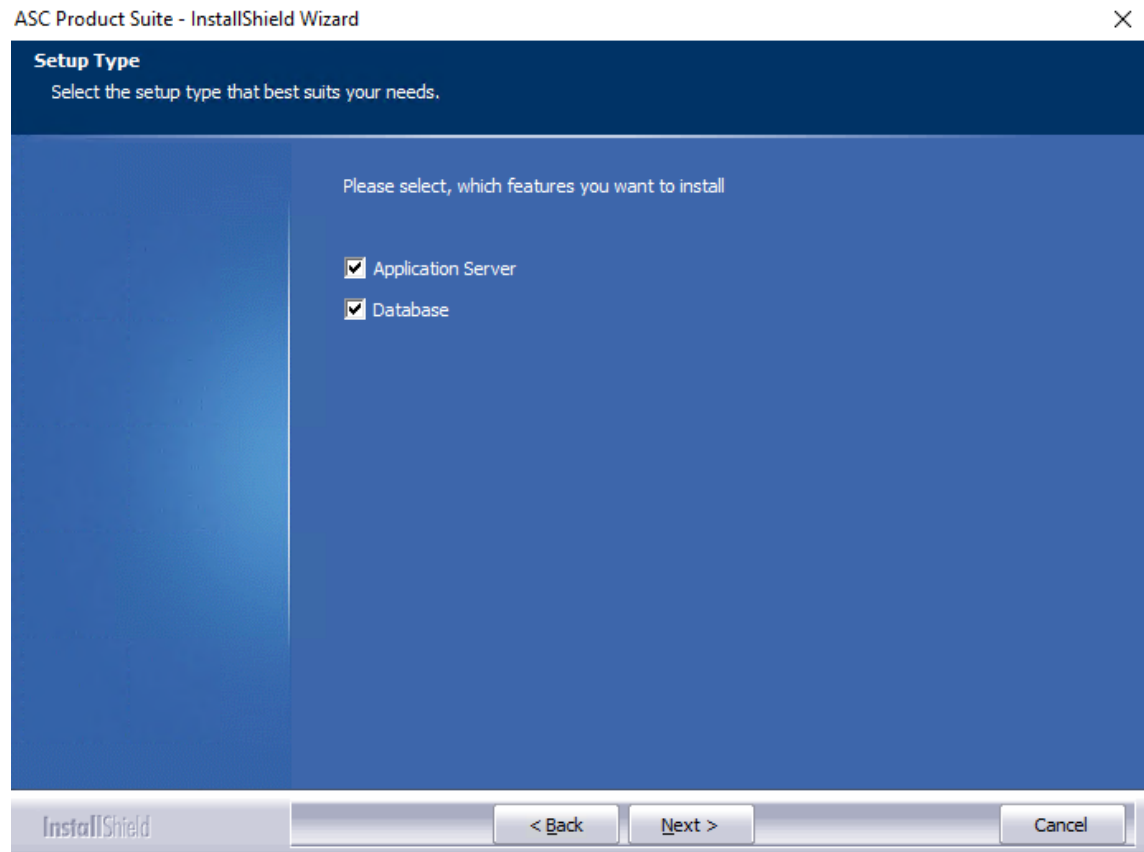


Abb. 1: Features zur Installation auswählen

6. Klicken Sie auf die Schaltfläche *Next*.
7. Klicken Sie auf die Schaltfläche *Browse*, um das Ziellaufwerk auszuwählen.

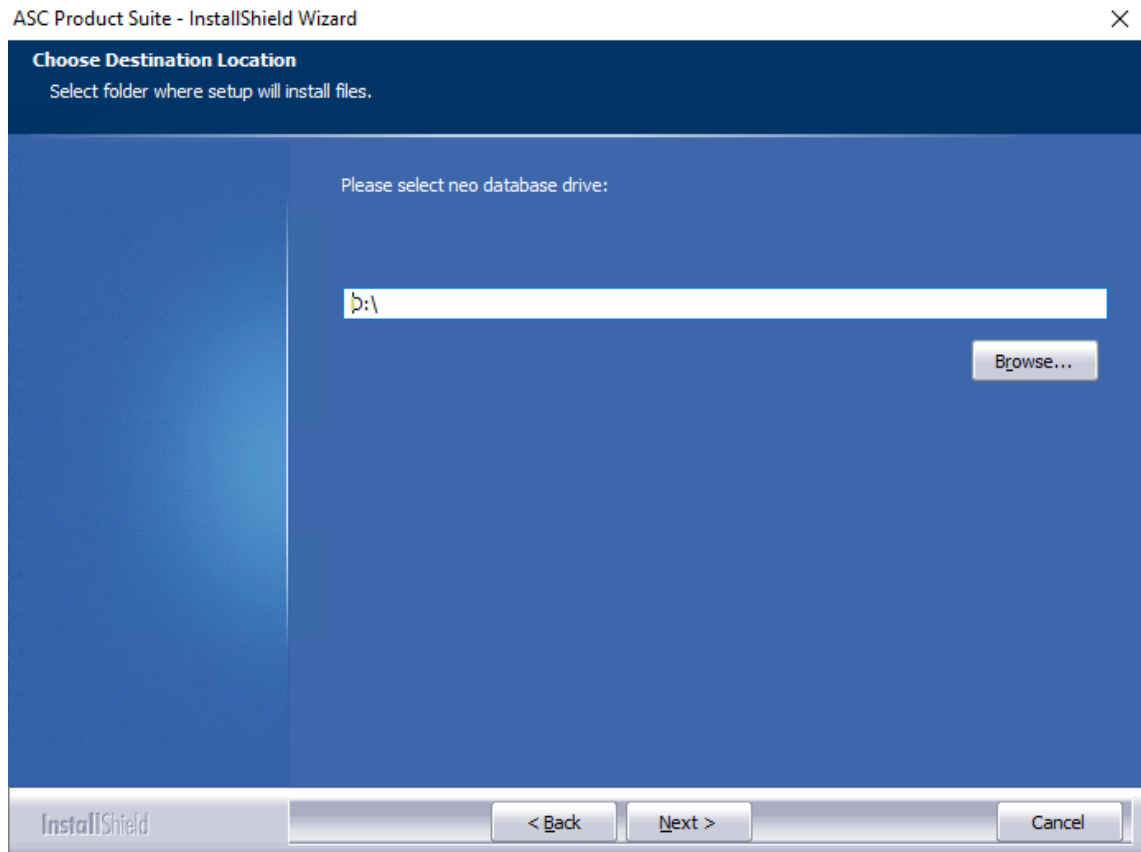


Abb. 2: Ziellaufwerk für die interne Datenbank wählen

8. Klicken Sie auf die Schaltfläche *Next*, um die nächsten Schritte der Installation durchzuführen.
9. Übernehmen Sie bis zum Ende der Installationsroutine in allen weiteren Installationsschritten die Standardeinstellungen.
10. Wählen Sie im letzten Schritt der Installationsroutine die Option *Yes, I want to restart my computer now*.

ASC Product Suite - InstallShield Wizard

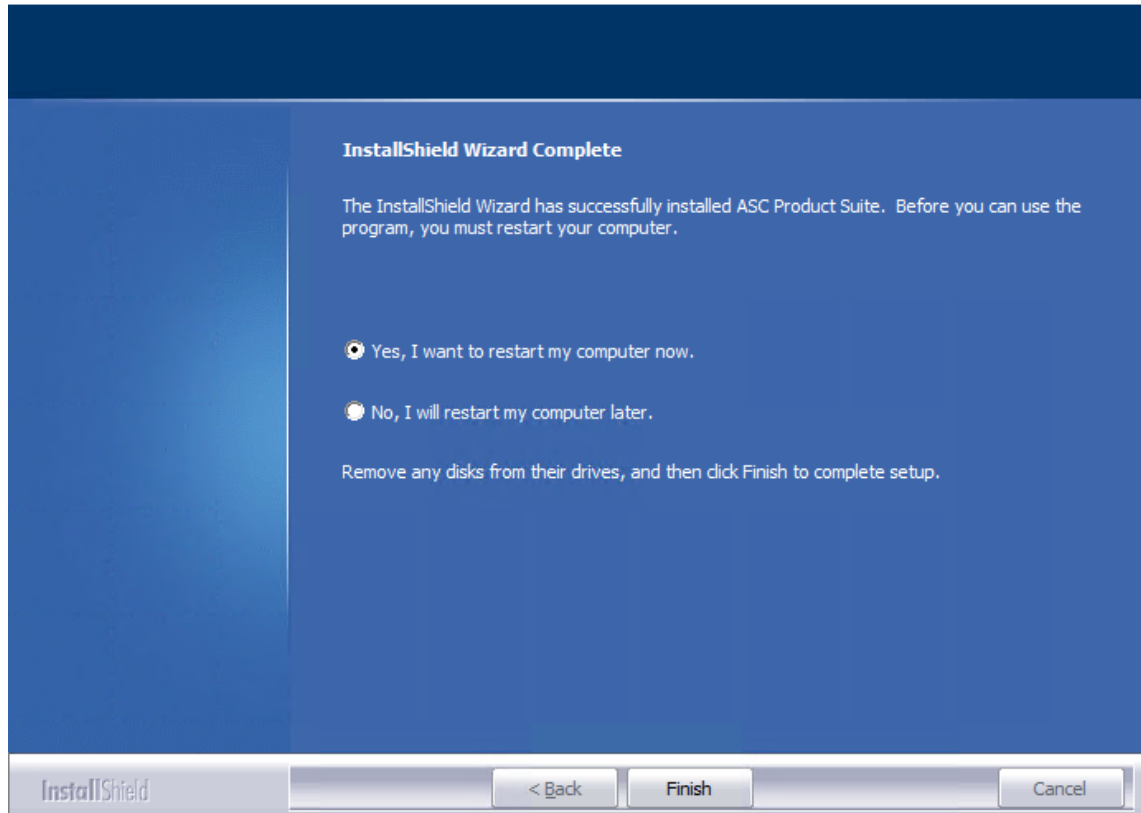


Abb. 3: Installation beenden und Server durchstarten

11. Beenden Sie die Installation der ASC-Software, indem Sie auf die Schaltfläche *Finish* klicken.
 - ⇒ Die Installation wird abgeschlossen.
 - ⇒ Der Server wird automatisch neu gestartet.
12. Stellen Sie sicher, dass auf der Primär-Datenbank die folgenden PostgreSQL-Dienste **aktiviert** sind:
 - postgresql-x64-n.m - PostgreSQL Server n.m
HINWEIS! *n.m* steht für die Versionsnummer der verwendeten PostgreSQL, z. B. *postgresql-x64-9.5 - PostgreSQL Server 9.5*.
 - PostgreSQL Scheduling Agent - pgAgent

4.2

Standby-Datenbank

Die Standby-Datenbank kann entweder als lokale Datenbank auf einem [neo-Applikationsserver](#) oder als externe Datenbank auf einem separaten Server installiert werden.

Die Installation auf einem [neo-Applikationsserver](#) erfolgt im Rahmen der Installation der ASC-Software, siehe Installationsanleitung *Installation der Aufzeichnungssoftware von ASC*.

Da auf dem Standby-Server der ASC-Dienst ServiceMan laufen muss, muss auch die Installation eines separaten Standby-Servers über das Setup der ASC-Software erfolgen.

Gehen Sie zur **Installation einer separaten Standby-Datenbank** folgendermaßen vor:

1. Legen Sie das Installationsmedium für die ASC-Software ein.
2. Wählen Sie aus dem Kontextmenü der Datei *setup.exe* den Menüpunkt *Run as administrator*.
3. Übernehmen Sie in allen Installationsschritten die Standardeinstellungen, bis Sie zur Installation der Datenbank kommen.

4. Installieren Sie eine interne PostgreSQL-Datenbank, indem Sie das Feature *Database* auswählen.

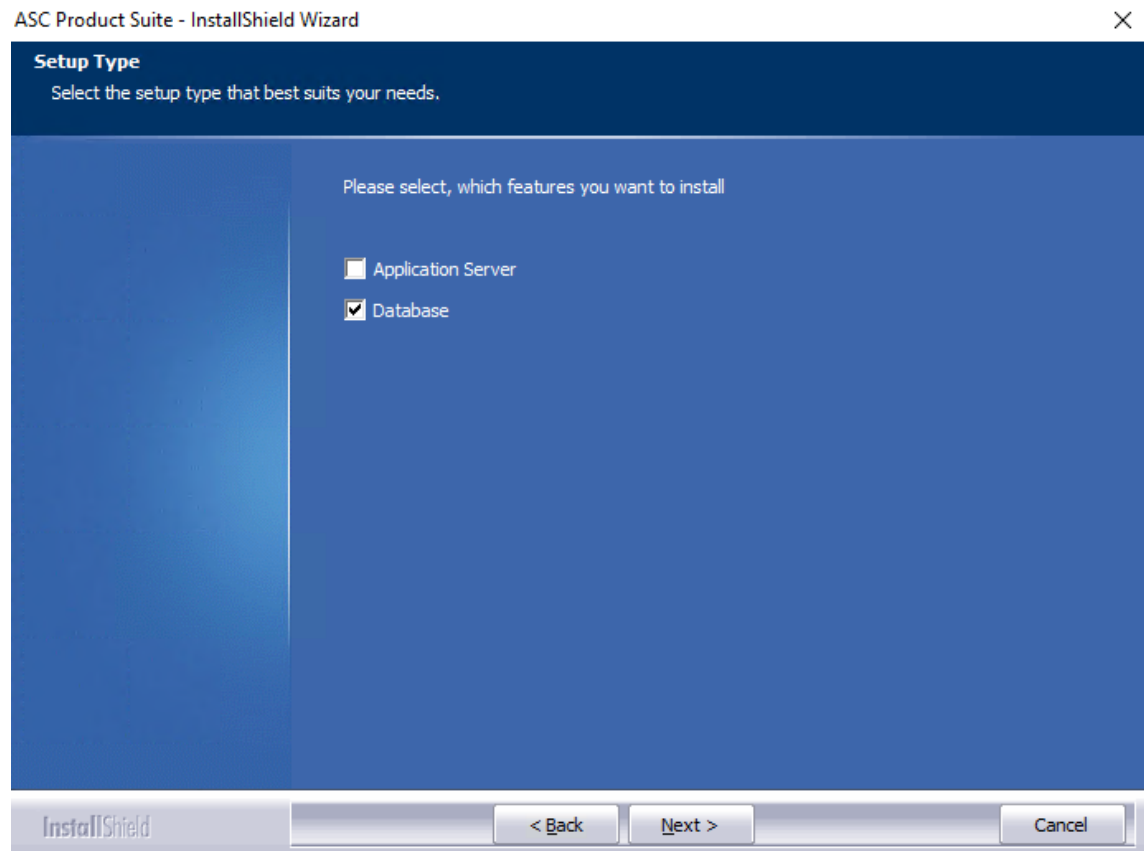


Abb. 4: Features zur Installation auswählen

5. Klicken Sie auf die Schaltfläche *Next*.
6. Klicken Sie auf die Schaltfläche *Browse*, um das Ziellaufwerk auszuwählen.

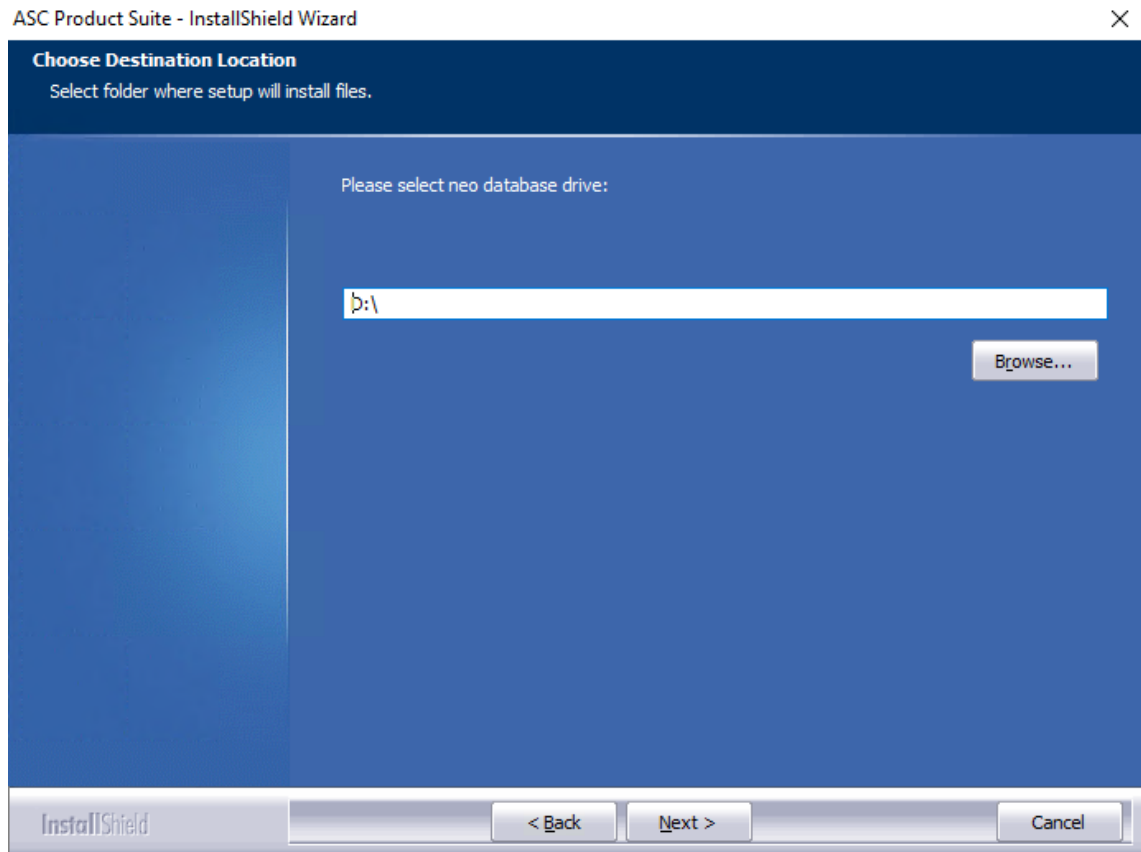


Abb. 5: Ziellaufwerk für die interne Datenbank wählen

7. Klicken Sie auf die Schaltfläche *Next*, um die nächsten Schritte der Installation durchzuführen.
8. Übernehmen Sie bis zum Ende der Installationsroutine in allen weiteren Installationsschritten die Standardeinstellungen.
9. Wählen Sie im letzten Schritt der Installationsroutine die Option *Yes, I want to restart my computer now*.

ASC Product Suite - InstallShield Wizard

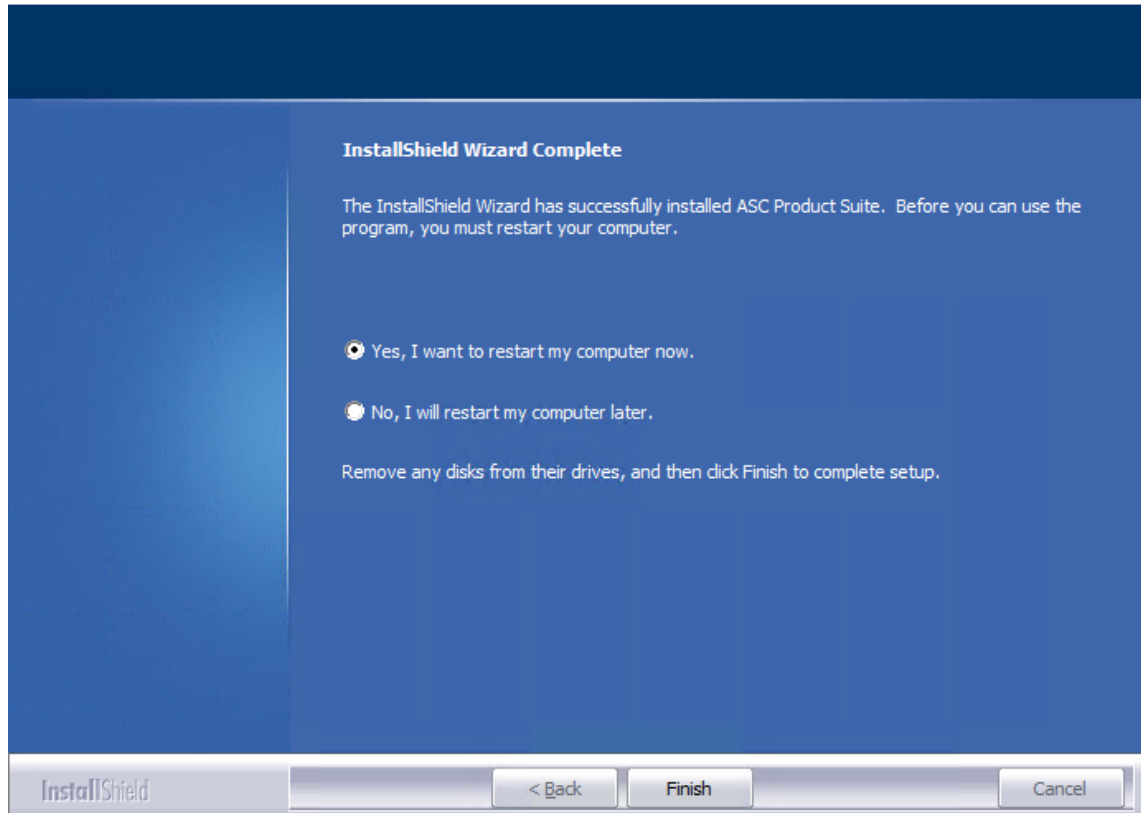


Abb. 6: Installation beenden und Server durchstarten

10. Beenden Sie die Installation der ASC-Software, indem Sie auf die Schaltfläche *Finish* klicken.
 - ⇒ Die Installation wird abgeschlossen.
 - ⇒ Der Server wird automatisch neu gestartet.
11. Stellen Sie sicher, dass auf der Standby-Datenbank der folgende PostgreSQL-Dienst **aktiviert** ist:
 - postgresql-x64-n.m - PostgreSQL Server n.m

HINWEIS! *n.m* steht für die Versionsnummer der verwendeten PostgreSQL, z. B. *postgresql-x64-9.5* - PostgreSQL Server 9.5.

5 Failover-Betrieb konfigurieren

Für die Konfiguration des Failover-Betriebs stehen 2 Möglichkeiten zur Verfügung:

- *Konfiguration mit dem Failover Configuration Tool, siehe Kapitel "Failover-Betrieb mit Failover Configuration Tool konfigurieren", S. 17.*
- *Manuelle Konfiguration der Datenbank- und Applikationsserver, siehe Kapitel "Failover-Betrieb manuell konfigurieren", S. 22*

5.1 Voraussetzungen

5.1.1 Netzwerkfreigabe

Damit ein Failover-Betrieb der beiden Datenbank-Server funktionieren kann, müssen die Laufwerke beider Server freigegeben und dem Benutzer *postgres* vollen Zugriff gewährt werden.

1. Starten Sie den Windows Explorer.
2. Klicken Sie mit der rechten Maustaste auf das Laufwerk F:\.
3. Wählen Sie aus dem Kontextmenü den Menüpunkt *Share with*.
4. Klicken Sie auf die Schaltfläche *Advanced Sharing*.
⇒ Das Fenster *Advanced Sharing* erscheint.
5. Aktivieren Sie das Kontrollkästchen *Share this folder*.
⇒ Die Settings werden aktiv und der Freigabename *F* wird vorgeschlagen.
6. Klicken Sie auf die Schaltfläche *Permissions*, um einem Benutzer oder einer Gruppe die Berechtigungen zuzuordnen.
⇒ Das Fenster *Permissions for F* erscheint.

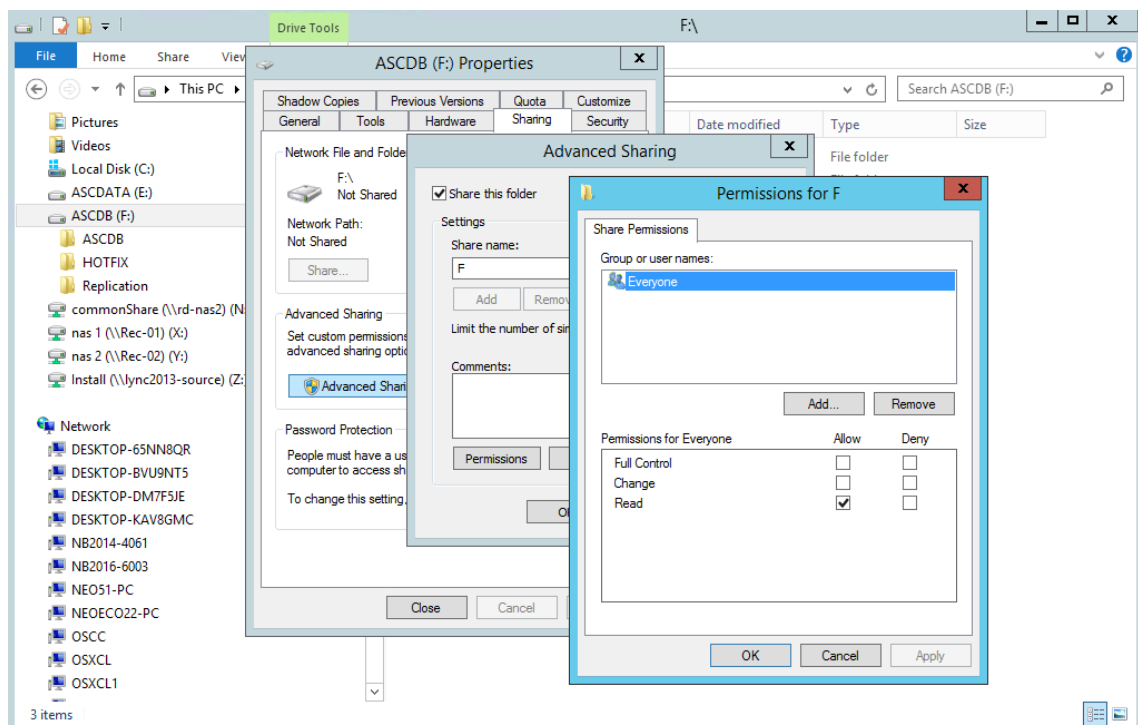


Abb. 7: Laufwerk freigeben

7. Klicken Sie auf die Schaltfläche *Add*.
⇒ Das Fenster *Select Users or Groups* erscheint.

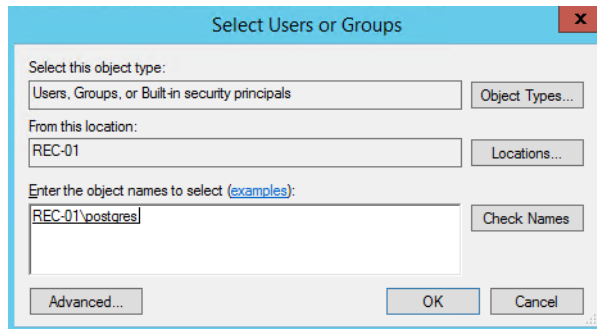


Abb. 8: Freigabe für den Benutzer *postgres* erteilen

8. Geben Sie in dem Eingabefeld *Enter the object names to select (examples)*: den Benutzer *postgres* ein.
9. Klicken Sie auf die Schaltfläche *Check Names*.
10. Der Benutzer wird als **FQDN** angezeigt.
11. Klicken Sie auf die Schaltfläche *OK*, um den Benutzer zuzuweisen und das Fenster zu schließen.
12. Aktivieren Sie das Kontrollkästchen *Full Control*, um den Vollzugriff zu erteilen.
13. Klicken Sie auf die Schaltfläche *OK*, um die Einstellungen zu übernehmen und das Fenster zu schließen.

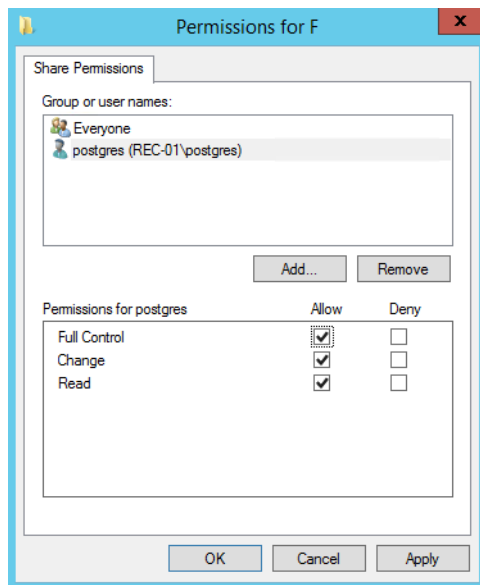


Abb. 9: Vollzugriff für den Benutzer *postgres* erteilen

14. Klicken Sie auf die Schaltfläche *OK*, um die Einstellungen zu übernehmen und das Fenster zu schließen.
15. Führen Sie die Netzwerkfreigabe auch auf dem Server mit der Standby-Datenbank durch.

5.1.2 Postgres-Dienst unter dem PostgreSQL-Konto starten

Der PostgreSQL-Dienst muss für die Replikation für beide Server unter dem Kontext des lokalen PostgreSQL-Kontos gestartet werden.

1. Wählen Sie in der Übersicht der Dienste den Dienst *postgresql-x64-9.5* aus.
2. Wählen Sie aus dem Kontextmenü den Eintrag *Properties*.
⇒ Das folgende Fenster öffnet sich.

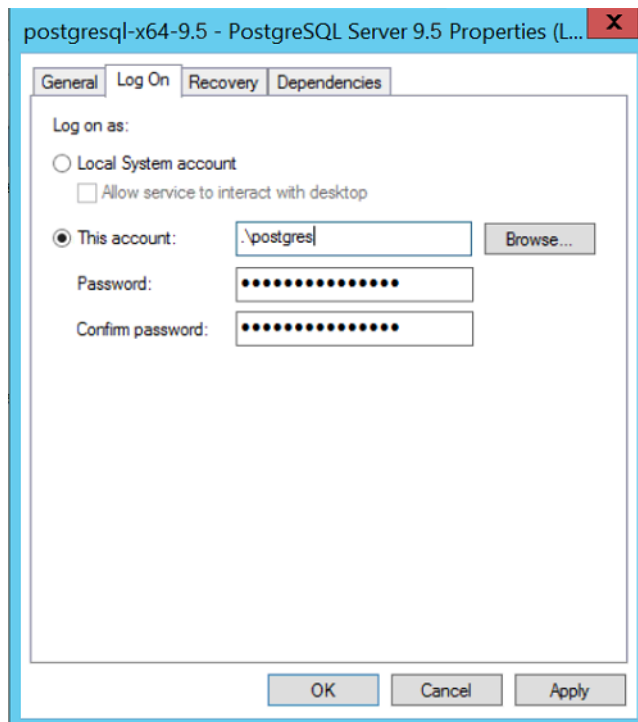


Abb. 10: Login für den Postgres-Dienst konfigurieren

3. Wählen Sie die Registerkarte *Log On*.
4. Aktivieren Sie die Option *This account* und geben Sie den Namen `.\postgres` und das Passwort für diesen Benutzer ein.

5.2 Failover-Betrieb mit Failover Configuration Tool konfigurieren

Das Failover Configuration Tool dient zur automatischen Konfiguration eines Datenbank-Failovers für zwei PostgreSQL-Datenbanken und beliebig vielen weiteren [App-Servern](#).



Verwenden Sie immer die Version des Tools, die in der jeweiligen Version mitgeliefert wird. Starten Sie das Tool immer auf dem Server, der nach der Ausführung als Primär-Server fungieren soll. Andernfalls schlägt die Konfiguration fehl.



Bei der Konfiguration mit dem Failover Configuration Tool werden die Datenbanken und die App-Server gestoppt!

1. Öffnen Sie den Windows Explorer und wechseln Sie in das Verzeichnis `C:\Program Files (x86)\ASC\ASC Product Suite\scripts`.
2. Öffnen Sie mit einem Doppelklick die Datei `DBFailoverConfig.exe`.
⇒ Das Failover Configuration Tool erscheint.

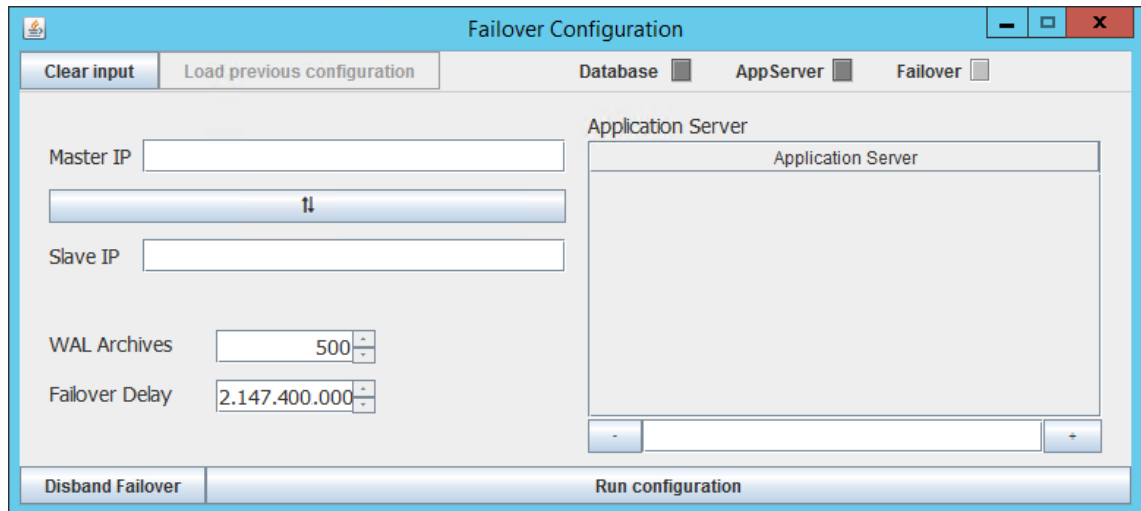


Abb. 11: Failover Configuration Tool - Failover konfigurieren

3. Geben Sie folgende Parameter ein:

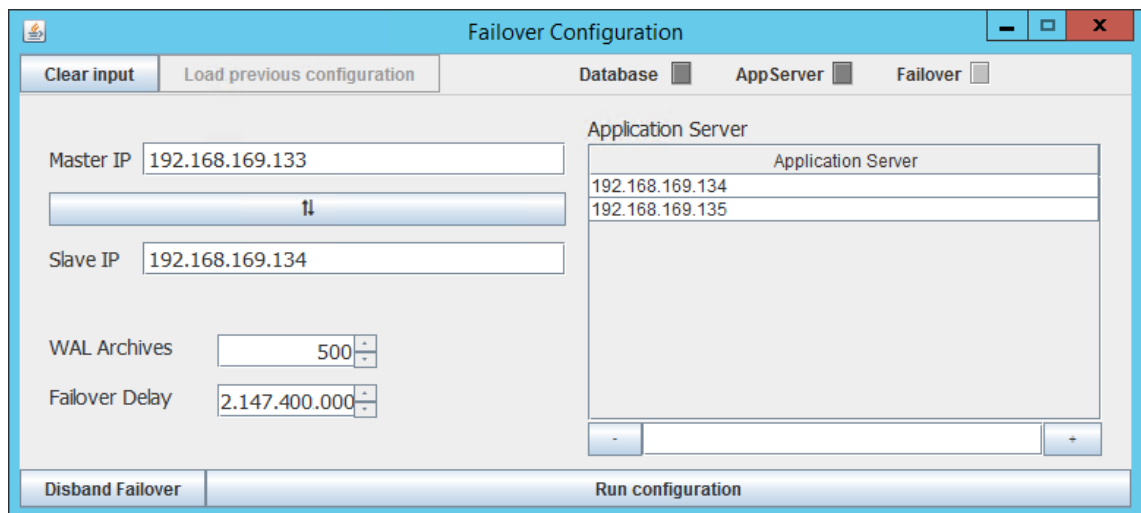


Abb. 12: Failover-Komponenten konfigurieren

Master IP	Geben Sie hier die IP-Adresse (IPv4 oder IPv6) der Master-Datenbank ein. Die Werte der Subnetzmaske /32 (IPv4) oder /128 (IPv6) werden automatisch ergänzt. Damit wird nur die angegebene IP-Adresse für die Datenbank freigeschaltet. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank konfigurieren möchten, müssen Sie diesen Wert selbst ergänzen.
Slave IP	Geben Sie hier die IP-Adresse (IPv4 oder IPv6) der Slave-Datenbank ein. Die Werte der Subnetzmaske /32 (IPv4) oder /128 (IPv6) werden automatisch ergänzt. Damit wird nur die angegebene IP-Adresse für die Datenbank freigeschaltet. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank konfigurieren möchten, müssen Sie diesen Wert selbst ergänzen.
Application Server	Falls ein Applikationsserver außerhalb der Master-IP oder Slave-IP läuft, geben Sie rechts unten in das Eingabefeld die IP-Adresse des Applikationsservers ein und fügen Sie diese über die Schaltflächen + zur Liste der Applikationsserver hinzu. Wiederholen Sie die Schritte für weitere separate Applikationsserver. Die Werte der Subnetzmaske /32 (IPv4) oder /128 (IPv6) werden automatisch ergänzt. Damit wird nur die angegebene IP-Adresse für die Datenbank freigeschaltet. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank konfigurieren möchten, müssen Sie diesen Wert selbst ergänzen.

WAL Archives	<p>Wählen Sie über die Drehfeld-Steuerelemente die Anzahl der WAL-Segmente, die gleichzeitig in der Freigabe bereitgestellt werden sollen. Die Anzahl beeinflusst die Dauer, bis die Replikation bei einem Verbindungsverlust der Datenbanken gestoppt wird.</p> <p>Das Verzeichnis <<POSTGRES-DATA-DRIVE>>\Replication, in dem die WAL-Archives liegen, wird bei der Installation der Datenbank angelegt.</p>
Failover Delay	<p>Wählen Sie über die Drehfeld-Steuerelemente die Millisekunden, die ein Applikationssserver bei Verbindungsverlust zur Master-Datenbank warten soll, bevor das Failover ausgelöst wird und der Wechsel zur Slave-Datenbank durchgeführt wird.</p> <p>Wählen Sie als Standardwert 2.147.400.000 Millisekunden.</p>

4. Klicken Sie auf die Schaltfläche *Run configuration*, um die Umsetzung der Konfiguration zu starten.

⇒ Das folgende Fenster erscheint:

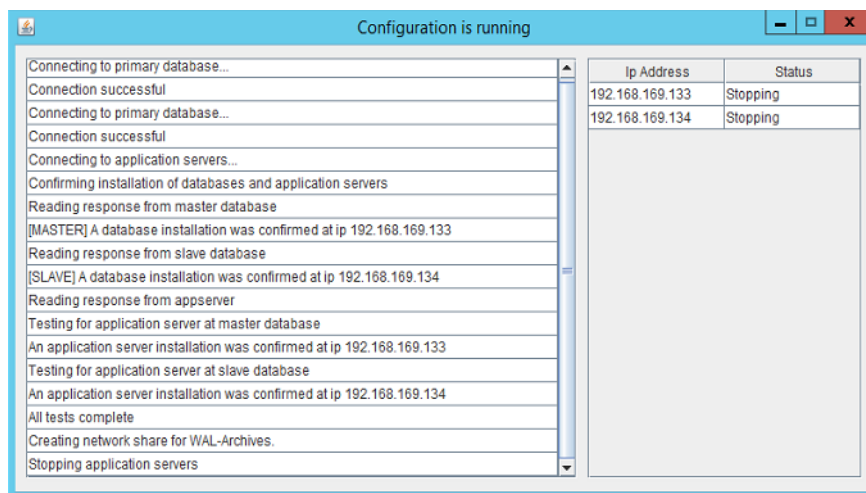


Abb. 13: Fortschritt der Konfiguration verfolgen

- ⇒ Im linken Fenster werden alle relevanten Informationen zum Ablauf des Vorgangs angezeigt.
- Im rechten Fenster werden die Verbindungen zu anderen Systemen mit deren Verbindungs- bzw. Operationsstatus angezeigt.
- ⇒ *Warnungen werden gelb markiert.*
- ⇒ *Fehler werden rot markiert.*
- ⇒ Wenn die Konfiguration durchgelaufen ist, erscheint eine Erfolgsmeldung.

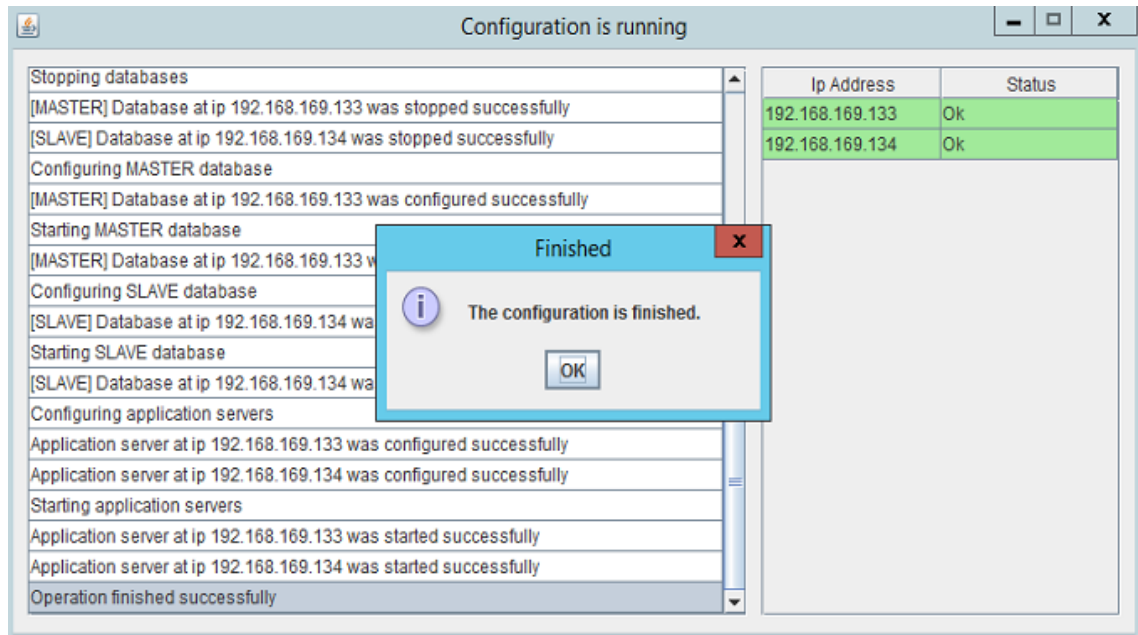


Abb. 14: Erfolgsmeldung

HINWEIS! Nach dem Vorgang werden die Applikationsserver automatisch neu gestartet. Es kann einige Zeit dauern, bis die Anwendungen wieder zur Verfügung stehen.

Nach der Konfiguration über das Configuration Tool, wird der Status der lokalen Komponenten *Database*, *AppServer* und *Replication* rechts oben angezeigt. Je nach Status der Komponente werden die Punkte farblich markiert. Im Tooltip bei Mouseover wird auch eine Textzusammenfassung angezeigt.

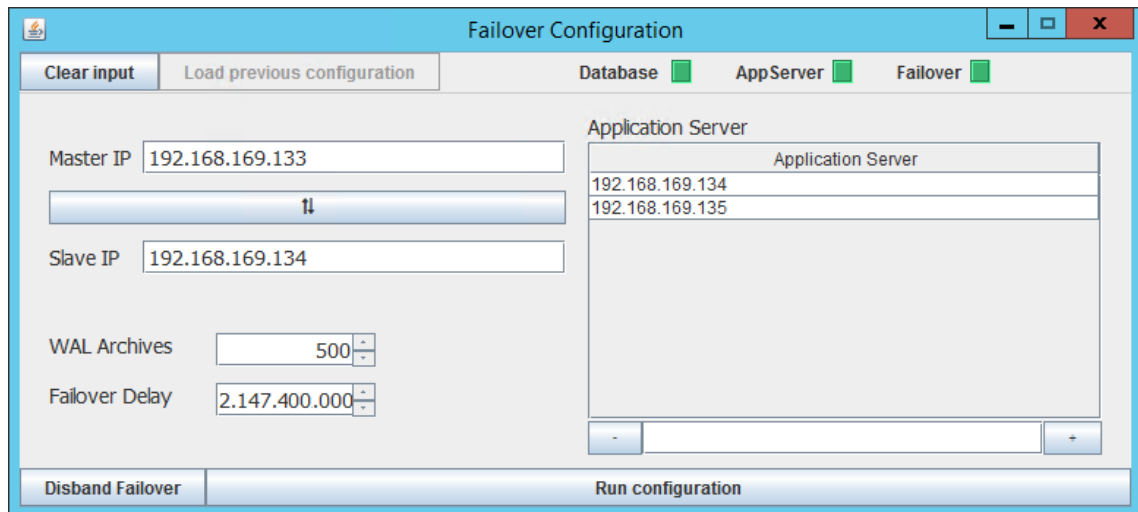


Abb. 15: Statusanzeige der Failover-Komponenten

Legende der Farbskala:

Database <input type="checkbox"/>	Keine lokale Installation gefunden
AppServer <input type="checkbox"/>	
Database <input type="checkbox"/>	Nicht konfiguriert / im Single Betrieb
AppServer <input type="checkbox"/>	
Database <input checked="" type="checkbox"/>	Failover konfiguriert
AppServer <input checked="" type="checkbox"/>	
Database <input type="checkbox"/>	Unbekannter / Fehlerhafter Zustand
AppServer <input type="checkbox"/>	
Failover <input type="checkbox"/>	Keine Datenbank / AppServer
Failover <input type="checkbox"/>	Nicht konfiguriert
Failover <input checked="" type="checkbox"/>	Failover aktiv
Failover <input type="checkbox"/>	Replikation gestoppt (Failover evtl. ausgelöst)
Failover <input type="checkbox"/>	Unbekannter / Fehlerhafter Zustand

Abb. 16: Farbskala

5.2.1 Weiteren Applikationsserver hinzufügen

Um einen weiteren Server in den Verbund aufzunehmen, müssen Sie das Failover Configuration Tool erneut aufrufen.

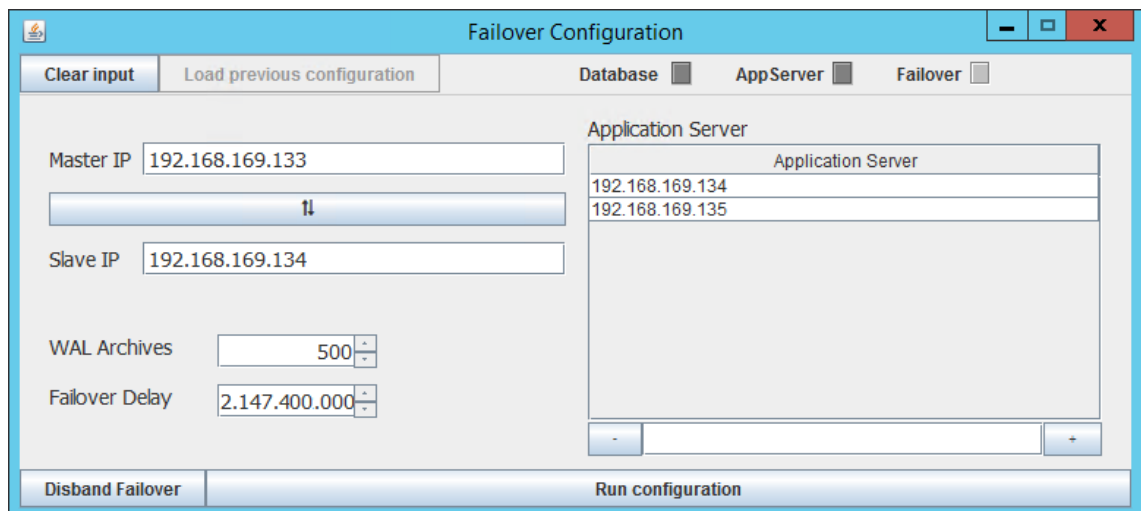


Abb. 17: Failover Configuration Tool erneut aufrufen

1. Klicken Sie in der Menüleiste oben links auf die Schaltfläche *Load previous configuration*, um die vorherige Konfiguration in das Dialogfenster zu laden.
2. Fügen Sie auf der rechten Seite unten über die Schaltfläche *+* einen weiteren Applikationsserver in die Liste der Applikationsserver hinzu.
3. Klicken Sie auf die Schaltfläche *Run Configuration*, um die Konfiguration erneut auszuführen.

Wenn Sie die Konfiguration über das Failover Configuration Tool erstellt haben, können Sie die manuelle Konfiguration überspringen und direkt wechseln zum [Kapitel "Failover-Betrieb Funktionsbeschreibung"](#), S. 36.

5.3 Failover-Betrieb manuell konfigurieren

5.3.1 Konfiguration der Datenbankserver

5.3.1.1 Primär-Datenbank einrichten

Gehen Sie in der Reihenfolge der folgenden Abschnitte vor, um den Failover-Betrieb auf dem Primär-Server einzurichten.

5.3.1.1.1 Replikationsverzeichnis einrichten

Legen Sie ein Replikationsverzeichnis an. In dieses Verzeichnis schreibt die Primär-Datenbank die **WAL**-Archive. Die Standby-Datenbank liest die **WAL**-Archive aus diesem Verzeichnis aus und löscht die ausgelesenen **WAL**-Archive anschließend.

1. Legen Sie parallel zum Verzeichnis <<PGTRES-DATA-FOLDER>> das Verzeichnis <<WAL-ARCHIVE>> an.
2. Geben Sie das Verzeichnis <<WAL-ARCHIVE>> im Netzwerk mit der Option *Full Access* frei.

5.3.1.1.2 Netzwerkfreigabe

Damit ein Failover-Betrieb der beiden Datenbank-Server funktionieren kann, müssen die Laufwerke beider Server freigegeben und dem Benutzer *postgres* vollen Zugriff gewährt werden.

1. Starten Sie den Windows Explorer.
2. Klicken Sie mit der rechten Maustaste auf das Laufwerk F:\.
3. Wählen Sie aus dem Kontextmenü den Menüpunkt *Share with*.
4. Klicken Sie auf die Schaltfläche *Advanced Sharing*.
⇒ Das Fenster *Advanced Sharing* erscheint.
5. Aktivieren Sie das Kontrollkästchen *Share this folder*.
⇒ Die Settings werden aktiv und der Freigabename *F* wird vorgeschlagen.
6. Klicken Sie auf die Schaltfläche *Permissions*, um einem Benutzer oder einer Gruppe die Berechtigungen zuzuordnen.
⇒ Das Fenster *Permissions for F* erscheint.

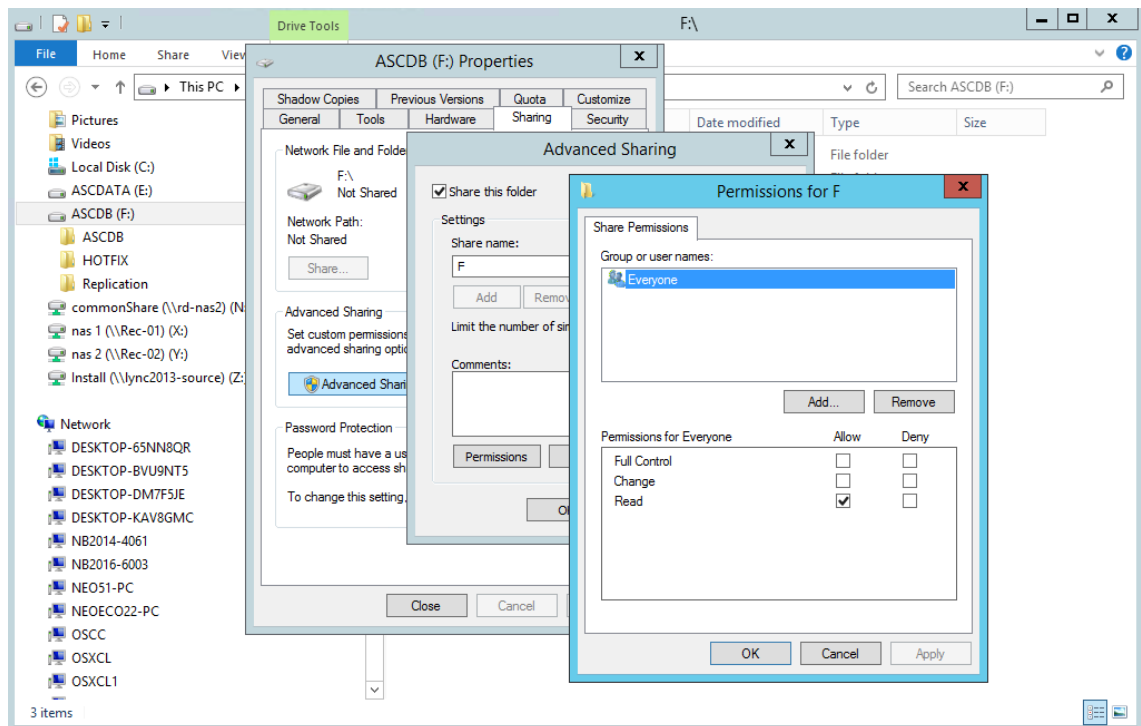
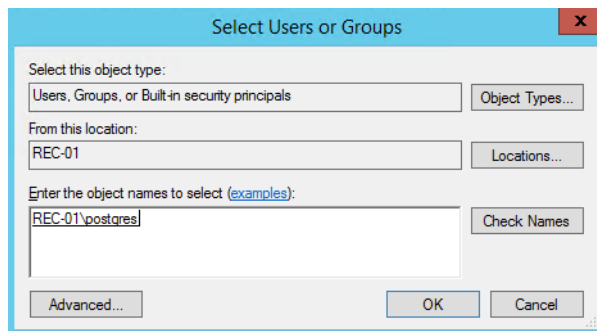


Abb. 18: Laufwerk freigeben

7. Klicken Sie auf die Schaltfläche *Add*.
 ⇒ Das Fenster *Select Users or Groups* erscheint.

Abb. 19: Freigabe für den Benutzer *postgres* erteilen

8. Geben Sie in dem Eingabefeld *Enter the object names to select (examples)*: den Benutzer *postgres* ein.
9. Klicken Sie auf die Schaltfläche *Check Names*.
10. Der Benutzer wird als **FQDN** angezeigt.
11. Klicken Sie auf die Schaltfläche *OK*, um den Benutzer zuzuweisen und das Fenster zu schließen.
12. Aktivieren Sie das Kontrollkästchen *Full Control*, um den Vollzugriff zu erteilen.
13. Klicken Sie auf die Schaltfläche *OK*, um die Einstellungen zu übernehmen und das Fenster zu schließen.

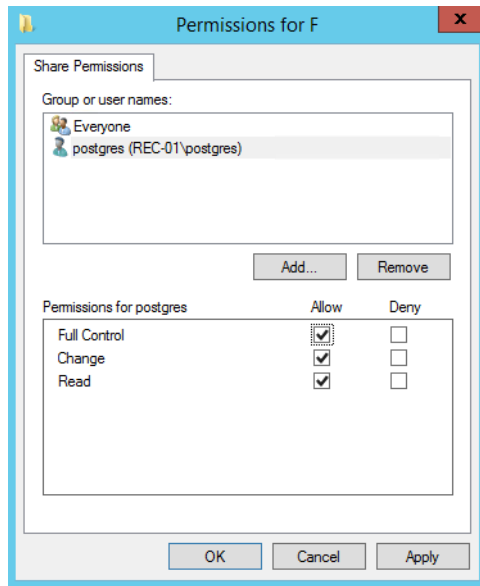


Abb. 20: Vollzugriff für den Benutzer *postgres* erteilen

14. Klicken Sie auf die Schaltfläche **OK**, um die Einstellungen zu übernehmen und das Fenster zu schließen.
15. Führen Sie die Netzwerkfreigabe auch auf dem Server mit der Standby-Datenbank durch.

5.3.1.1.3 Datei "*pg_hba.conf*" anpassen

Anhand der Einstellungen in der Datei *pg_hba.conf* erfolgt die Authentifizierung der Client-Zugriffe.

1. Öffnen Sie auf dem Primär-Server im Verzeichnis <<POSTGRES-DATA-FOLDER>> die Konfigurationsdatei *pg_hba.conf* mit einem Editor.
2. Fügen Sie am Ende des Dateiinhalts folgenden Eintrag hinzu, um der Standby-Datenbank den Replikationszugriff auf die Primär-Datenbank zu gewähren:

Typ	Datenbank	Benutzer	IP-Adresse	Methode
host	replication	postgres	<<IP-ADDRESS-STANDBY-DB>>/<<CIDR>>	md5

Beispiel:

```
host replication postgres 192.168.171.120/32 md5
```

```
host replication postgres FE80::29ED:F6FB:D58D:F92B/128 md5
```

HINWEIS! Die Subnetzmasken /32 (IPv4) oder /128 (IPv6) beziehen sich auf die angegebene IP-Adresse. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank konfigurieren möchten, müssen Sie den Wert an die Bedingungen im Kundennetzwerk anpassen.

3. Außerdem müssen alle *neo*-Applikationsserver Zugriff auf die PostgreSQL-Instanz haben. Fügen Sie dafür für jeden der angeschlossenen *neo*-Applikationsserver folgenden Eintrag hinzu:

Typ	Datenbank	Benutzer	IP-Adresse	Methode
host	asc_rs	postgres	<<IP-ADDRESS-NEO-CORE>>/<<CIDR>>	md5
host	postgres	postgres	<<IP-ADDRESS-NEO-CORE>>/<<CIDR>>	md5

Beispiel:

```
host asc_rs postgres 192.168.171.50/32 md5
```

```
host postgres postgres 192.168.171.50/32 md5
```



Als Trennzeichen zwischen den Werten für Typ, Datenbank, Benutzer, IP-Adresse und Methode müssen Sie jeweils ein Tabulator-Zeichen verwenden.



Der Wert für <<CIDR>> muss durch den Systembetreiber zur Verfügung gestellt werden. Die Werte der Subnetzmaske /32 (IPv4) oder /128 (IPv6) beziehen sich nur auf die angegebene IP-Adresse. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank freischalten möchten, müssen Sie diesen Wert anpassen.



Beispielhafter Auszug einer angepassten Datei:
[Kapitel "Datei pg_hba.conf, Primär-Server", S. 42](#)

5.3.1.1.4 Datei "postgresql.conf" anpassen

Die Datei *postgresql.conf* ist die zentrale Konfigurationsdatei für PostgreSQL. Nehmen Sie die folgenden Einstellungen vor, um die Replikation zu ermöglichen.

1. Öffnen Sie auf dem Primär-Server im Verzeichnis <<POSTGRES-DATA-FOLDER>> die Konfigurationsdatei *postgresql.conf* mit einem Editor.
2. Geben Sie in der Datei nachfolgende Werte an und achten Sie darauf, dass die Einträge nicht auskommentiert sind.
 - `max_wal_senders = 10`
 - `wal_level = hot_standby`
 - `archive_mode = on`
 - `archive_command = 'copy %p \\\<<IP-ADDRESS-PRIMARY-DB>>\\<<WAL-ARCHIVE>> \\%f'`
 - `wal_keep_segments = <<WAL-SEGMENTS>>`

Beispiel für IPv4:

```
max_wal_senders = 10
wal_level = hot_standby
archive_mode = on
archive_command = 'copy %p \\\192.168.171.110\\WAL-ARCHIVE\\%f'
wal_keep_segments = 500
```

HINWEIS! Dies ist eine Pfadangabe und erfordert für die IPv6 IP-Adresse die Schreibweise mit '-' und '.ipv6-literal.net'

Beispiel für IPv6:

```
max_wal_senders = 10
wal_level = hot_standby
archive_mode = on
archive_command = 'copy %p \\\FE80--29ED-F6FB-D58D-F92B.ipv6-literal.net \\WAL-ARCHIVE\\%f'
wal_keep_segments = 500
```

Die Anzahl der *wal_keep_segments* ist ein Maß dafür, wie lange die Replikation zur Standby-Datenbank unterbrochen und danach wiederaufgenommen werden kann. Je größer der Wert ist, umso länger kann die Unterbrechung dauern, ohne dass die Replikation erneut manuell eingerichtet werden muss.



Die Anzahl der [WAL](#)-Segmente, die pro Zeiteinheit geschrieben werden, ist abhängig von der Nutzung der Datenbank und kann nicht pauschal angegeben werden. (Default-Wert: 500)

Ein [WAL](#)-Segment benötigt 16 MB Speicherplatz auf der Festplatte. Setzen Sie daher den Wert des Parameters *wal_keep_segments* in Abhängigkeit des Speicherplatzes auf dem Laufwerk <<POSTGRES-DATA-DRIVE>>. Berücksichtigen Sie dabei, dass der Speicherbedarf der Datenbank im Laufe der Nutzungsdauer zunimmt.



Beispielhafter Auszug einer angepassten Datei:
[Kapitel "Datei postgresql.conf, Primär-Server", S. 43](#)

5.3.1.1.5 PostgreSQL-Dienst neu starten

Starten Sie folgenden Dienst neu:

- postgresql-x64-n.m - PostgreSQL Server n.m

Starttyp: *Automatisch*

HINWEIS! *n.m* steht für die Versionsnummer der verwendeten PostgreSQL, z. B. *postgresql-x64-9.5* - PostgreSQL Server 9.5

5.3.1.1.6 Datenbank-Backup erstellen

Erstellen Sie mit dem Programm *PGAdmin* ein Backup der Datenbank *asc_rs*.

5.3.1.2 Standby-Datenbank einrichten

Gehen Sie in der Reihenfolge der folgenden Abschnitte vor, um den Failover-Betrieb auf dem Standby-Server einzurichten.

5.3.1.2.1 Netzwerkfreigabe von der Standby-Datenbank auf die Primär-Datenbank kontrollieren

1. Öffnen Sie auf dem Standby-Server den Windows Explorer.
2. Versuchen Sie mit folgendem Befehl auf das Replikationsverzeichnis auf dem Primär-Server zuzugreifen \\<<IP-ADDRESS-PRIMARY-DB>>\<<WAL-ARCHIVE>>.

Beispiel:

\\192.168.171.110\WAL-ARCHIVE

\\FE80--29ED-F6FB-D58D-F92B.ipv6-literal.net\WAL-ARCHIVE

HINWEIS! Wenden Sie sich an einen Netzwerkadministrator, falls der Zugriff auf dieses Verzeichnis vom Standby-Server aus nicht möglich ist.

5.3.1.2.2 Backup der Primär-Datenbank auf der Standby-Datenbank einspielen

1. Löschen Sie auf dem Standby-Server den Inhalt des Verzeichnisses <<POSTGRES-DATA-FOLDER>>, damit das Backup vom Primär-Server dort abgelegt werden kann.
2. Öffnen Sie ein Konsolenfenster im Verzeichnis <<POSTGRES-INSTALL-FOLDER>>\bin.
3. Geben Sie den Zielpfad und den Befehl für das Backup ein:
 pg_basebackup -R -D <<POSTGRES-DATA-FOLDER>> --host=<<IP-ADDRESS-PRIMARY-DB>> --port=5432 -U postgres --checkpoint=fast --xlog-method=stream -R --verbose --progress

Beispiel:

pg_basebackup -R -D D:\ASCDB --host=192.168.171.110 --port=5432 -U postgres --checkpoint=fast --xlog-method=stream -R --verbose --progress

4. Geben Sie das Datenbank-Passwort für den Benutzer *postgres* ein.
- ⇒ Der Primär-Server erstellt ein Backup der Primär-Datenbank und kopiert es in das Verzeichnis <<POSTGRES-DATA-FOLDER>> auf dem Standby-Server.

5.3.1.2.3 Datei "pg_hba.conf" anpassen

Anhand der Einstellungen in der Datei *pg_hba.conf* erfolgt die Authentifizierung der Client-Zugriffe. Stellen Sie hier ein, dass alle neo-Applikationsservern Zugriff auf die PostgreSQL-Instanz haben.

1. Öffnen Sie auf dem Standby-Server im Verzeichnis <<POSTGRES-DATA-FOLDER>> die Konfigurationsdatei *pg_hba.conf* mit einem Editor.

- Fügen Sie am Ende des Dateiinhalts für jeden der angeschlossenen *neo*-Applikationsserver folgenden Eintrag hinzu:

Typ	Datenbank	Benutzer	IP-Adresse	Methode
host	asc_rs	postgres	<<IP-ADDRESS-NEO-CORE>>/<<CIDR>>	md5

Beispiel:

```
host asc_rs postgres 192.168.171.50/32 md5
```

```
host asc_rs postgres FE80::29ED:F6FB:D58D:F92B/128 md5
```

HINWEIS! Die Subnetzmasken /32 (IPv4) oder /128 (IPv6) beziehen sich auf die angegebene IP-Adresse. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank konfigurieren möchten, müssen Sie den Wert an die Bedingungen im Kundennetzwerk anpassen.



Beispielhafter Auszug einer angepassten Datei:

[Kapitel "Datei pg_hba.conf, Standby-Server", S. 42](#)



Als Trennzeichen zwischen den Werten für Typ, Datenbank, Benutzer, IP-Adresse und Methode müssen Sie jeweils ein Tabulator-Zeichen verwenden.



Der Wert für <<CIDR>> muss durch den Systembetreiber zur Verfügung gestellt werden. Die Werte der Subnetzmaske /32 (IPv4) oder /128 (IPv6) beziehen sich nur auf die angegebene IP-Adresse. Falls Sie einen Adressbereich für den Zugriff auf die Datenbank freischalten möchten, müssen Sie diesen Wert anpassen.

5.3.1.2.4 Datei "postgresql.conf" anpassen

Die Datei *postgresql.conf* ist die zentrale Konfigurationsdatei für PostgreSQL. Nehmen Sie die folgenden Einstellungen vor, um die Replikation zu ermöglichen.

- Öffnen Sie auf dem Standby-Server im Verzeichnis <<POSTGRES-DATA-FOLDER>> die Konfigurationsdatei *postgresql.conf* mit einem Editor.
- Tragen Sie in der Datei folgenden Wert ein und achten Sie darauf, dass der Eintrag nicht auskommentiert ist:
 - hot_standby = on



Beispielhafter Auszug einer angepassten Datei:

[Kapitel "Datei postgresql.conf, Standby-Server", S. 43](#)

5.3.1.2.5 Datei "recovery.conf" anpassen

- Öffnen Sie auf dem Standby-Server im Verzeichnis <<POSTGRES-DATA-FOLDER>> die Konfigurationsdatei *recovery.conf* mit einem Editor.
- Ergänzen Sie folgende Einträge:
 - restore_command = 'copy \\\<<IP-ADDRESS-PRIMARY-DB>>\<<WAL-ARCHIVE>>\%f %p'
 - trigger_file = '<<POSTGRES-DATA-DRIVE>>:\standby.txt'
 - archive_cleanup_command = '"<<POSTGRES-INSTALL-FOLDER>>\bin\pg_archivecleanup" \\\<<IP-ADDRESS-PRIMARY-DB>>\<<WAL-ARCHIVE>>\%r'

Beispiel für IPv4:

```
restore_command = 'copy \\192.168.171.110\WAL-ARCHIVE\%f %p'
```

```
trigger_file = 'D:\standby.txt'
```

```
archive_cleanup_command = '
"C:\Program Files\PostgreSQL\9.5\bin\pg_archivecleanup" \\
\192.168.171.110\WAL-ARCHIVE\ %r'
```

Beispiel für IPv6:

```
restore_command = 'copy \\\FE80--29ED-F6FB-D58D-F92B.ipv6-literal.net\WAL-
ARCHIVE\%f %p'
trigger_file = 'D:\standby.txt'
archive_cleanup_command = '
"C:\Program Files\PostgreSQL\9.5\bin\pg_archivecleanup" \\\FE80--29ED-
F6FB-D58D-F92B.ipv6-literal.net\WAL-ARCHIVE\ %r'
```

3. Kontrollieren Sie, ob in dem bereits vorhandenen Eintrag für den Parameter *primary_conninfo* die korrekten Werte für password und host gesetzt sind. Diese Angaben müssen den Werten des Primär-Servers entsprechen.
 host: <<IP-ADDRESS-PRIMARY-DB>>
 password: Datenbank-Passwort des Benutzers *postgres*



Beispielhafter Auszug einer angepassten Datei:

[Kapitel "Datei recovery.conf, Standby-Server", S. 44](#)

5.3.1.2.6 PostgreSQL-Dienst neu starten

1. Starten Sie folgenden Dienst neu:
 - postgresql-x64-n.m - PostgreSQL Server n.m
 Starttyp: *Automatisch*

HINWEIS! *n.m* steht für die Versionsnummer der verwendeten PostgreSQL, z. B. *postgresql-x64-9.5* - PostgreSQL Server 9.5

5.3.1.3 Datei setup.xml anpassen

Für den Failover-Betrieb müssen Sie auf beiden Datenbankservern in der Konfigurationsdatei *setup.xml* die IP-Adressen aller *neo*-Applikationsserver ([App-Server](#)) in die AIP-Liste eintragen:

1. Öffnen Sie im Verzeichnis <<NEO-INSTALL-FOLDER>>\Updater\config die Datei *setup.xml* mit einem Editor.
2. Suchen Sie bei **IPv4 IP-Adressen** nach dem Eintrag <AIPList>127.0.0.1</AIPList>.
3. Ersetzen Sie in diesem Eintrag die Standard-IP-Adresse 127.0.0.1 durch die IP-Adresse des *neo*-Applikationsservers.
 Falls Ihr System mehrere *neo*-Applikationsserver hat, tragen Sie anstelle der Standard-IP-Adresse 127.0.0.1 die IP-Adressen aller *neo*-Applikationsserver getrennt durch Semikolon ein.
 Beispiel:
 <AIPList>192.168.171.50;192.168.171.60;192.168.171.70</AIPList>
4. Suchen Sie bei **IPv6 IP-Adressen** nach dem Eintrag <AIPList>::1</AIPList>.
5. Ersetzen Sie in diesem Eintrag die Standard-IP-Adresse ::1 durch die IP-Adresse des *neo*-Applikationsservers.
 Beispiel:
 <AIPList>FE80::29ED:F6FB:D58D:F92B; FE80::29ED:F6FB:D58D:F82B;
 FE80::29ED:F6FB:D58D:F72B</AIPList>
6. Speichern Sie die Änderungen in der Datei.
7. Starten Sie den Dienst ServiceMan neu.



Beispielhafter Auszug einer angepassten Datei:
[Kapitel "Datei setup.xml, Datenbankserver", S. 44](#)

5.3.2 Konfiguration der Applikationsserver

5.3.2.1 Datei setup.xml anpassen

Für den Failover-Betrieb der Datenbanken müssen Sie auf jedem neo-Applikationsserver ([App-Server](#)) in der Konfigurationsdatei *setup.xml* folgende 2 Einstellungen manuell hinzufügen:

- `<databaseMode>FAILOVER</databaseMode>`

Diese Einstellung bewirkt, dass der [App-Server](#) im Failover-Modus betrieben wird und in der Applikation System Configuration für den Systembetreiber das Datenbank-Manager-Modul verfügbar wird.

- `<databaseFailoverDelay><<DELAY>></databaseFailoverDelay>`

Diese Einstellung bewirkt, dass die Umschaltung auf die Standby-Datenbank erst erfolgt, wenn die Primär-Datenbank für einen bestimmten Zeitraum nicht verfügbar war. Dadurch wird verhindert, dass bereits bei einem kurzen Ausfall der Netzwerkverbindung eine Umschaltung auf die Standby-Datenbank erfolgt. Die Dauer, nach der die Umschaltung erfolgen soll, wird an der Stelle des Platzhalters `<<DELAY>>` in Millisekunden angegeben.

Führen Sie folgende Schritte auf allen angeschlossenen [App-Servern](#) aus:

1. Öffnen Sie im Verzeichnis `<<NEO-INSTALL-FOLDER>>\Updater\config` die Datei *setup.xml* mit einem Editor.
2. Fügen Sie vor der Zeile `</Settings>` die beiden folgenden Zeilen ein:
`<databaseMode>FAILOVER</databaseMode>`
`<databaseFailoverDelay>60000</databaseFailoverDelay>`

Der Wert des Elements `<databaseFailoverDelay>` gibt die Dauer in Millisekunden an, die bei einem Ausfall der Verbindung zum Primär-Server gewartet wird, bis das Failover zum Standby-Server erfolgt.



Wenn der Wert sehr klein gewählt ist, können bereits kurzfristige Netzerkerausfälle zu einem Failover führen.

Wenn Sie den Wert sehr groß setzen, ist beim tatsächlichen Ausfall des Primär-Servers für die eingestellte Dauer keine Datenbank verfügbar. Während dieser Zeit stehen die Web-Applikation des Aufzeichnungssystems nicht zur Verfügung.

3. Ändern Sie die Einstellungen für die Datenbank *dbip* auf die IP-Adresse des Primär-Servers.
`<dbip><IP-ADDRESS-PRIMARY-DB></dbip>`
4. Falls die Datenbank *dbip* nicht auf dem neo-Applikationsserver installiert ist, ersetzen Sie folgende Zeile:
`<ascdatabase>...</ascdatabase>`
 durch
`<ascdatabase>EXTERNAL</ascdatabase>`
5. Speichern Sie die Änderungen in der Datei.



Beispielhafter Auszug einer angepassten Datei:
[Kapitel "Datei setup.xml, Applikationsserver", S. 44](#)

5.3.2.2 Datenbankverbindung konfigurieren

Alle neo-Applikationsserver müssen Zugriff auf die Primär- und die Standby-Datenbank erhalten. Die Datenbankverbindung wird in einer Konfigurationsdatei jedes neo-Applikationsservers verwaltet.

Führen Sie folgende Schritte auf allen angeschlossenen *neo*-Applikationsservern aus:

- Öffnen Sie auf dem *neo*-Applikationsserver die Konfigurationsdatei *domain.xml* mit einem Editor.
Sie finden die Datei in folgendem Verzeichnis:
<<NEO-INSTALL-FOLDER>>\glassfish4\glassfish\domains\enterprisecore\config\
- Suchen Sie in dieser Datei das XML-Element *jdbc-connection-pool* mit dem Attribut *name="ASCSuitePoolPrimary"*.

Es gibt zwei gleichwertige Darstellungen der Konfiguration der Datenbankverbindung:

Variante 1 IPv4:

IP-Adresse, Port und Datenbankname sind gemeinsam in das Attribut *value* des Elements *property name="URL"* eingetragen.

Beispiel:

```
<property name="URL" value="jdbc:postgresql://192.168.171.110:5432/asc_rs"></property>
```

Variante 1 IPv6:

IP-Adresse, Port und Datenbankname sind gemeinsam in das Attribut *value* des Elements *property name="URL"* eingetragen.

Beispiel:

```
<property name="URL" value="jdbc:postgresql://[FE80::29ED:F6FB:D58D:F92B]:5432/asc_rs"></property>
```

Variante 2 IPv4:

IP-Adresse, Port und Datenbankname sind in getrennten *property*-Elementen eingetragen.

Beispiel:

```
<property name="ServerName" value="postgresql://192.168.171.110"></property>
<property name="databasename" value="asc_rs"></property>
<property name="portname" value="5432"></property>
```

Variante 2 IPv6:

IP-Adresse, Port und Datenbankname sind in getrennten *property*-Elementen eingetragen.

Beispiel:

```
<property name="ServerName" value="postgresql://FE80::29ED:F6FB:D58D:F92B"></property>
<property name="databasename" value="asc_rs"></property>
<property name="portname" value="5432"></property>
```

Falls in Ihrer Datei *domain.xml* Variante 2 verwendet wird, entfernen Sie die *property*-Elemente mit den Namen *ServerName*, *databasename* und *portname* und fügen Sie das Element *<property name="URL" value="jdbc:postgresql://localhost:5432/asc_rs"></property>* an deren Stelle ein, so dass Sie die Darstellungsform von Variante 1 erreichen.

- Tragen Sie im Unterelement *property name="URL"* im Attribut *value* die IP-Adresse, den Port und den Namen der Primär-Datenbank ein.

Beispiel IPv4:

```
<property name="URL" value="jdbc:postgresql://<<IP-ADDRESS-PRIMARY-DB>>:5432/asc_rs"></property>
```

Beispiel:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
name="ASCSuitePool" validation-table-name="asc_rs.db_connection_control"
is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
<property name="URL" value="jdbc:postgresql://192.168.171.110:5432/asc_rs">
</property>
<property name="user" value="postgres"></property>
```




```
<property name="password" value="${ALIAS=db_password_alias}">
</property>
</jdbc-connection-pool>
```

HINWEIS! Beachten Sie dabei die folgende Formatierung:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true" validate-atmost-once-period-in-seconds="60"
  <property name="URL" value="jdbc:postgresql://192.168.171.110:5432/asc_rs"></property>
  <property name="user" value="postgres"></property>
  <property name="password" value="${ALIAS=db_password_alias}"></property>
</jdbc-connection-pool>
```

Abb. 21: Formatierungsprinzip

HINWEIS! Falls bei der Verwendung von IPv6 IP-Adressen die Eingabe eines Ports erforderlich ist, müssen Sie die IP-Adresse in eckige Klammern setzen, um sie von dem Port abzugrenzen.

Beispiel IPv6:

```
<property name="URL" value="jdbc:postgresql://<<IP-ADDRESS-PRIMARY-DB>>:5432/
asc_rs"></property>
```

Beispiel:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
name="ASCSuitePool" validation-table-name="asc_rs.db_connection_control"
is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
  <property name="URL" value="jdbc:postgresql://[FE80::29ED:F6FB:D58D:F92B]:5432/asc_rs">
  </property>
  <property name="user" value="postgres"></property>
  <property name="password" value="${ALIAS=db_password_alias}">
  </property>
</jdbc-connection-pool>
```

- Suchen Sie in der gleichen Datei das XML-Element *jdbc-connection-pool* mit dem Attribut *name="ASCSuitePoolStandby"*



Auch für diesen Eintrag gibt es die beiden oben beschriebenen Darstellungsvarianten.

Falls in Ihrer Datei *domain.xml* Variante 2 verwendet wird, entfernen Sie auch hier die *property*-Elemente mit den Namen *ServerName*, *databasename* und *portname* und fügen Sie den Eintrag in der Darstellungsform von Variante 1 wieder ein.

- Passen Sie den Eintrag analog zum vorangehenden Schritt an, indem Sie dabei die IP-Adresse des Standby-Servers verwenden.

Beispiel IPv4:

```
<property name="URL" value="jdbc:postgresql://<<IP-ADDRESS-STANDBY-DB>>:5432/
asc_rs"></property>
```

Beispiel:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60" steady-pool-size="0"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
validation-table-name="asc_rs.db_connection_control"
name="ASCSuitePoolPrimary" is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
  <property name="URL" value="jdbc:postgresql://192.168.171.120:5432/asc_rs">
  </property>
```

```
<property name="user" value="postgres"></property>
<property name="password" value="{ALIAS=db_password_alias}"></property>
</jdbc-connection-pool>
```

Beispiel IPv6:

```
<property name="URL" value="jdbc:postgresql://<<IP-ADDRESS-STANDBY-DB>>:5432/
asc_rs"></property>
```

Beispiel:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60" steady-pool-size="0"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
validation-table-name="asc_rs.db_connection_control"
name="ASCSuitePoolPrimary" is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
<property name="URL" value="jdbc:postgresql://[FE80::29ED:F6FB:D58D:F92B]:5432/asc_rs">
</property>
<property name="user" value="postgres"></property>
<property name="password" value="{ALIAS=db_password_alias}"></property>
</jdbc-connection-pool>
```

6. Suchen Sie in der Datei nach dem Eintrag

```
<resource-ref ref="jms/ConnectionFactory"></resource-ref>
```
7. Fügen Sie folgende Zeile hinzu:

```
<resource-ref ref="jdbc/ASCSuitePoolQuartz"></resource-ref>
```

Beispiel:

```
<resource-ref ref="jms/ConnectionFactory"></resource-ref>
<resource-ref ref="jdbc/ASCSuitePoolQuartz"></resource-ref>
```
8. Suchen Sie in der Datei nach dem Eintrag

```
<jdbc-resource pool-name="ASCSuitePoolStandby"></jdbc-resource>
```
9. Fügen Sie folgende Zeile hinzu:

```
<jdbc-resource pool-name="ASCSuitePoolQuartz" jndi-name="jdbc/
ASCSuitePoolQuartz"></jdbc-resource>
```

Beispiel:

```
<jdbc-resource pool-name="ASCSuitePoolStandby"
description="Jdbc Resource for managed connection pool"
jndi-name="jdbc/ASCSuiteStandby"></jdbc-resource>
<jdbc-resource pool-name="ASCSuitePoolQuartz"
jndi-name="jdbc/ASCSuitePoolQuartz"></jdbc-resource>
```

10. Fügen Sie unterhalb des XML-Elements `jdbc-connection-pool` mit dem Attribut `name="ASCSuitePoolStandby"` den nachfolgenden Eintrag hinzu und setzen Sie dabei die IP-Adressen der Primär- und der Standby-Datenbank.

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
validation-table-name="asc_rs.db_connection_control"
name="ASCSuitePoolQuartz" is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
<property name="URL" value="jdbc:postgresql://<<IP-ADDRESS-PRIMARY-
DB>>:5432,<<IP-ADDRESS-STANDBY-DB>>:5432/asc_rs"></property>
```



```
<property name="user" value="postgres"></property>
<property name="password" value="\${ALIAS=db_password_alias}"></property>
</jdbc-connection-pool>
```

Beispiel IPv4:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
validation-table-name="asc_rs.db_connection_control"
name="ASCSuitePoolQuartz" is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
<property name="URL" value="jdbc:postgresql://192.168.171.110:5432,192.168.171.120:5432/
asc_rs"></property>
<property name="user" value="postgres"></property>
<property name="password" value="\${ALIAS=db_password_alias}"></property>
</jdbc-connection-pool>
</resources>
```

Beispiel IPv6:

```
<jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
validate-atmost-once-period-in-seconds="60"
datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
validation-table-name="asc_rs.db_connection_control"
name="ASCSuitePoolQuartz" is-connection-validation-required="true"
res-type="javax.sql.ConnectionPoolDataSource">
<property name="URL" value="jdbc:postgresql://[FE80::29ED:F6FB:D58D:F92B]:5432,
[FE80::29ED:F6FB:D58D:F82B]:5432/asc_rs"></property>
<property name="user" value="postgres"></property>
<property name="password" value="\${ALIAS=db_password_alias}"></property>
</jdbc-connection-pool>
</resources>
```



Beispielhafter Auszug einer angepassten Datei,
siehe [Kapitel "Datei domain.xml, Applikationsserver", S. 45](#)

11. Stoppen Sie den Dienst *ServiceMan*.
12. Starten Sie den Dienst *ApplicationServer* bzw. *enterprisecore* neu und stellen Sie den Starttyp des Dienstes auf *Automatisch*.

ACHTUNG!**Möglicher Datenverlust**

Beim Starten des *neo*-Applikationsservers müssen beide Datenbanken verfügbar sein. Ist eine Datenbank nicht verfügbar, wird der Startprozess abgebrochen und der *neo*-Applikationsserver ist nicht verfügbar. In diesem Fall kann keine Aufzeichnung stattfinden.

Sollte eine der Datenbanken für einen längeren Zeitraum nicht verfügbar sein, so setzen Sie als IP-Adresse für beide JDBC-Connection-Pools die IP-Adresse der verfügbaren Datenbank.

Nachdem die ausgefallene Datenbank wieder verfügbar ist, können Sie die IP-Adresse wieder ändern. Stellen Sie dabei sicher, dass die Replikation funktionsfähig ist, siehe [Kapitel "Funktionstest", S. 34](#).

Falls Sie die Replikation manuell wieder starten müssen, gehen Sie vor wie in [Kapitel "Failover-Betrieb zurücksetzen", S. 38](#) beschrieben.

13. Warten Sie bis der Dienst ApplicationServer wieder komplett gestartet ist. Überprüfen Sie die Funktion des Dienstes, indem Sie eine beliebige *neo*-Web-Applikation aufrufen. Wenn Sie die *neo*-Web-Applikation starten können, bedeutet dies, dass der Dienst ApplicationServer läuft.
14. Starten Sie den Dienst ServiceMan wieder und stellen Sie den Starttyp des Dienstes auf *Automatisch*.

5.4 Funktionstest

5.4.1 Replikation auf dem Primär-Server kontrollieren

1. Öffnen Sie auf dem Primär-Server das Programm *PGAdmin* und verbinden Sie sich mit der Datenbank *asc_rs*.
2. Führen Sie den folgenden Befehl aus:

```
select * from pg_stat_replication;
```

⇒ Erwartete Ergebnisse:

 - Genau eine Zeile wird als Ergebnis zurückgeliefert.
 - Der Wert des Attributs *client_addr* entspricht der IP-Adresse des Standby-Servers.
 - Der Wert des Attributs *state* lautet *streaming*.

5.4.2 Replikation auf dem Standby-Server kontrollieren

1. Öffnen Sie auf dem Standby-Server das Programm *PGAdmin* und verbinden Sie sich mit der Datenbank *asc_rs*.
2. Führen Sie den folgenden Befehl aus:

```
select pg_is_in_recovery(), pg_last_xlog_receive_location(),  
pg_last_xlog_replay_location();
```



- *pg_is_in_recovery* gibt an, ob der Wiederherstellungsprozess läuft.
 - *pg_last_xlog_receive_location* gibt den letzten *WAL* Record an, den der Standby-Server vom Primär-Server erhalten hat.
 - *pg_last_xlog_replay_location* gibt den letzten *WAL* Record an, der auf dem Standby-Server verarbeitet wurde.
- ⇒ Erwartete Ergebnisse:
- Der Wert des Attributs *pg_is_in_recovery* ist *t* (für *true*).
 - Die Werte der Attribute *pg_last_xlog_receive_location* und *pg_last_xlog_replay_location* werden hexadezimal dargestellt und haben identische oder zumindest numerisch ähnliche Werte.

5.4.3 Netzwerkfreigabe kontrollieren

1. Öffnen Sie auf dem Standby-Server den Datei-Explorer.
2. Versuchen Sie auf das Replikationsverzeichnis auf dem Primär-Server zuzugreifen (*\\<IP-ADDRESS-PRIMARY-DB>\<WAL-ARCHIVE>*).
3. Wenden Sie sich an einen Netzwerkadministrator, wenn der Zugriff auf dieses Verzeichnis vom Standby-Server aus nicht möglich ist.

5.4.4 Löschen der WAL-Archive kontrollieren

Kontrollieren Sie, ob die *WAL*-Archive auf dem Primär-Server gelöscht werden.

- ✓ Die *neo*-Applikationsserver müssen mit den Datenbanken verbunden sein, siehe [Kapitel "Datenbankverbindung konfigurieren", S. 29](#).

1. Kontrollieren Sie bei laufendem Betrieb, ob ältere Dateien im Replikationsverzeichnis \`<<IP-ADDRESS-PRIMARY-DB>>\<<WAL-ARCHIVE>>` gelöscht werden.
2. Falls die Anzahl der **WAL**-Archive im Replikationsverzeichnis stetig zunimmt, kann es sein, dass der Benutzer, unter dessen Account der Postgres-Dienst "*postgresql-x64-n.m - PostgreSQL Server n.m*" läuft, nicht das Recht zum Löschen der Dateien im Replikationsverzeichnis hat.

Lassen Sie in diesem Fall den Postgres-Dienst unter dem Account des lokalen Benutzers *postgres* oder dem Account eines Domänen-Benutzers laufen, der über ausreichende Rechte verfügt.

HINWEIS! Wenn PostgreSQL mit dem Setup der *neo* Suite installiert wird, läuft der Dienst unter einem lokalen System-Account.

6 Failover-Betrieb Funktionsbeschreibung

Die Replikation unterstützt die Konsistenz und Verfügbarkeit der Daten. Die [Partitionstoleranz](#) ist dagegen nicht gegeben.

Mit der Failover-Funktion können Sie folgende Ausfallszenarien abfangen:

- *Ausfall der Primär-Datenbank*

Bei Systemen mit mehreren Enterprise Cores wird ständig abgefragt, ob die primäre Datenbank von allen Enterprise Cores aus verfügbar ist. Nur wenn keiner der Enterprise Cores mehr auf die Datenbank zugreifen kann, greift die Failover-Funktion und es wird auf die Standby-Datenbank umgeschaltet. Siehe [Kapitel "Ausfall der Primär-Datenbank", S. 36](#)

- *Ausfall der Standby-Datenbank*

Wenn die Standby-Datenbank nicht verfügbar ist, kann im Fehlerfall der Primär-Datenbank nicht auf die Standby-Datenbank umgeschaltet werden. Siehe [Kapitel "Ausfall der Standby-Datenbank", S. 37](#).

- *Ausfall eines Enterprise Cores*

Wenn ein Enterprise Core ausfällt, müssen Sie die Failover-Funktion manuell auslösen. Siehe [Ausfall eines Enterprise Cores](#)

6.1 Ausfall der Primär-Datenbank

Wenn die Primär-Datenbank ausfällt, erfolgt automatisch eine Umschaltung auf die Standby-Datenbank.

ACHTUNG!

Möglicher Datenverlust

Da die Replikation unterbrochen wird, kann es zu Datenverlust kommen. Der Umfang des Datenverlusts kann jedoch nicht pauschal angegeben werden, sondern muss im Bedarfsfall individuell ermittelt werden.

Wenn die Primär-Datenbank nicht mehr zur Verfügung steht, werden die folgenden Prozesse durch einen der konfigurierten *neo*-Applikationsserver gestartet:

1. Der Dienst ServiceMan des Standby-Servers schreibt die Datei *standby.txt* in das Stammverzeichnis des Laufwerks <<POSTGRES-DATA-FOLDER>>.
2. Diese Datei *standby.txt* wird von der Standby-Datenbank gelesen und anschließend gelöscht.
3. Die Replikation wird abgebrochen.
4. Die Standby-Datenbank wird automatisch mit Lese-/Schreibzugriff neu gestartet. Sie übernimmt damit die Rolle der produktiven Datenbank.
5. Wenn ein Ausfall der Primär-Datenbank registriert wird, wird auf jedem *neo*-Applikationsserver die Datei *DBSlaveTriggerFile.txt* angelegt. Sie finden die Datei in folgendem Verzeichnis:
 <<NEO-INSTALL-FOLDER>>\glassfish4\glassfish\domains\enterprisecore\config\



Die Datei *DBSlaveTriggerFile.txt* führt dazu, dass die Primär-Datenbank nicht verwendet wird.

Wenn die Standby-Datenbank verfügbar ist und ein Failover erfolgt, werden die Daten der folgenden Konversationen in der Standby-Datenbank gespeichert.

Nachdem der Fehler behoben wurde, muss der Failover-Betrieb zurückgesetzt werden. Das Zurücksetzen kann manuell oder mit dem Configuration Tool erfolgen.

- Failover-Betrieb mit dem Failover Configuration Tool zurücksetzen, siehe [Kapitel "Failover-Betrieb mit Failover Configuration Tool zurücksetzen", S. 40](#).

- Failover-Betrieb manuell zurücksetzen, siehe [Kapitel "Failover-Betrieb manuell zurücksetzen"](#), S. 38.

6.2 Ausfall der Standby-Datenbank

Der Ausfall der Standby-Datenbank hat auf den Betrieb der neo Suite keine Auswirkungen, sofern die Primär-Datenbank während der Ausfalldauer zur Verfügung steht.

Wenn die Standby-Datenbank ausfällt, wird allerdings die Replikation unterbrochen. Abhängig von der Dauer des Ausfalls, kann die Replikation automatisch wieder gestartet werden, sobald die Standby-Datenbank wieder zur Verfügung steht. Wird die Replikation nicht automatisch wieder gestartet, können Sie die Replikation manuell starten.

1. Wenn die Standby-Datenbank wieder zur Verfügung steht, führen Sie folgende Schritte durch:
 - Kontrollieren Sie, ob die Replikation funktioniert, siehe [Kapitel "Funktionstest"](#), S. 34.
 - Falls die Replikation **nicht** ausgeführt wird, müssen Sie die Replikation manuell wieder starten, siehe [Kapitel "Standby-Server wiederherstellen"](#), S. 39.

Failover-Betrieb zurücksetzen

Wenn die Primär-Datenbank wieder zur Verfügung steht, stehen Ihnen 2 Möglichkeiten zur weiteren Verwendung zur Verfügung.

Wählen Sie je nach Aufbau der Systemarchitektur die passende Konfiguration aus:

1. Sie können von der Standby-Datenbank manuell auf die Primär-Datenbank zurückschalten. Siehe [Kapitel "Failover-Betrieb manuell zurücksetzen"](#), S. 38.
2. Sie können die Standby-Datenbank primär nutzen und die Replikations-Richtung umkehren. Siehe [Kapitel "Failover-Betrieb mit Failover Configuration Tool zurücksetzen"](#), S. 40.

Failover-Betrieb manuell zurücksetzen

Um nach einem Failover den Ursprungszustand wiederherzustellen, müssen Sie sowohl auf den beiden Datenbankservern als auch auf den *neo*-Applikationsservern manuelle Konfigurationsänderungen durchführen. Gehen Sie dazu in der Reihenfolge der im Folgenden beschriebenen Abschnitte vor.

ACHTUNG!

Möglicher Datenverlust

Während der Wiederherstellung des Ursprungszustandes darf keine Änderung des Datenbestands erfolgen, da beide Datenbanken für kurze Zeit nicht zur Verfügung stehen.

Damit Datenverlust vermieden wird, ist deshalb ein [Wartungsfenster](#) erforderlich, während die Datenbanken aus dem Failover-Betrieb in den Ursprungszustand zurückgesetzt werden.

Dienste auf den Applikationsservern stoppen

Führen Sie folgende Schritte auf allen *neo*-Applikationsservern durch:

1. Stoppen Sie den Dienst ServiceMan und setzen Sie den Starttyp für diesen Dienst auf *Deaktiviert*.
2. Stoppen Sie den Dienst ApplicationServer und setzen Sie den Starttyp für diesen Dienst auf *Deaktiviert*.
3. Löschen Sie im Verzeichnis `<<NEO-INSTALL-FOLDER>>\glassfish4\glassfish\domains\enterpriseconfig\` die Datei *DBSlaveTriggerFile.txt*, falls vorhanden.
Beispielhafter Verzeichnispfad:
`C:\Program Files (x86)\ASC\ASC Product Suite\glassfish4\glassfish\domains\enterpriseconfig\`



Die Datei *DBSlaveTriggerFile.txt* ist nicht auf allen *neo*-Applikationsservern vorhanden.

Backup der Standby-Datenbank erstellen

1. Erstellen Sie auf dem Standby-Server mit dem Programm *PGAdmin* ein Backup der Datenbank *asc_rs*.
2. Stoppen Sie auf dem Standby-Server die folgenden Dienste:
 - postgresql-x64-n.m - PostgreSQL Server n.m
Starttyp: *Automatisch*
n.m steht für die Versionsnummer der verwendeten PostgreSQL, z. B. *postgresql-x64-9.5 - PostgreSQL Server 9.5*.
 - PostgreSQL Scheduling Agent - pgAgent
Starttyp: *Automatisch*

Primär-Server wiederherstellen

1. Je nachdem, aus welchem Grund die Primär-Datenbank ausgefallen ist, kann es erforderlich sein, das Datenbank-Management-System PostgreSQL neu zu installieren, siehe [Kapitel "Primär-Datenbank", S. 8](#).
2. Unabhängig davon, ob das Datenbank-Management-System neu installiert wurde oder weiterbetrieben werden kann, müssen Sie auf dem Primär-Server auf jeden Fall das Backup des Standby-Servers (Datenbank-Backup der Standby-Datenbank erstellen) einspielen. Nutzen Sie dazu das Programm *pgAdmin*.
3. Falls Sie die Datenbank neu installieren mussten, führen Sie die in [Kapitel "Primär-Datenbank einrichten", S. 22](#) beschriebenen Konfigurationsänderungen erneut aus.

Standby-Server wiederherstellen

1. Stellen Sie die Replikation auf dem Standby-Server wieder her, wie in [Kapitel "Standby-Server wiederherstellen", S. 39](#) beschrieben.

Dienste auf den Applikationsservern starten

Führen Sie folgende Schritte auf allen *neo*-Applikationsservern durch:

1. Ändern Sie den Starttyp des Dienstes ApplicationServer auf *Manuell*.
 2. Ändern Sie den Starttyp des Dienstes ServiceMan auf *Automatisch*.
 3. Starten Sie den Dienst ServiceMan, falls er nicht automatisch gestartet wird.
- ⇒ Der Dienst ServiceMan startet automatisch den Dienst ApplicationServer. Der Dienst ApplicationServer muss deshalb nicht manuell gestartet werden.

Funktionstest

1. Kontrollieren Sie, ob die Replikation funktioniert, siehe [Kapitel "Funktionstest", S. 34](#).

7.1.1 Standby-Server wiederherstellen

Sie können den Status der Replikation nach den in [Kapitel "Funktionstest", S. 34](#) beschriebenen Verfahren kontrollieren.

Wenn die Replikation ausgefallen ist, was z. B. durch einen längeren Ausfall der Standby-Datenbank oder einer längeren Unterbrechung der Netzwerkverbindung zwischen Primär- und Standby-Server verursacht werden kann, sind die folgenden Schritte erforderlich, um die Replikation wieder zu starten.



Die im Folgenden beschriebenen Änderungen müssen ausschließlich auf dem Standby-Server durchgeführt werden.

- ✓ Die Primär-Datenbank wird wieder von allen *neo*-Applikationsservern als Produktivdatenbank verwendet.
1. Legen Sie das Verzeichnis `<<POSTGRES-DATA-DRIVE>>\pg_conf_backup` an.
 2. Kopieren Sie die folgenden Dateien in das neu angelegte Verzeichnis:
`<<POSTGRES-DATA-FOLDER>>\pg_hba.conf`
`<<POSTGRES-DATA-FOLDER>>\postgresql.conf`
`<<POSTGRES-DATA-FOLDER>>\recovery.conf`
`<<POSTGRES-DATA-FOLDER>>\DataBase.conf`
 3. Löschen Sie den Inhalt des Verzeichnisses `<<POSTGRES-DATA-FOLDER>>`.
 4. Öffnen Sie ein Konsolenfenster im Verzeichnis `<<POSTGRES-INSTALL-FOLDER>>\bin`.

5. Geben Sie den Zielpfad und den Befehl für das Backup ein:
`pg_basebackup -R -D <<PGRES-DATA-FOLDER>> --host=<<IP-ADDRESS-PRIMARY-DB>> --port=5432 -U postgres --checkpoint=fast --xlog-method=stream -R --verbose --progress`
 Beispiel: `pg_basebackup -R -D D:\ASCDB --host=192.168.171.110 --port=5432 -U postgres --checkpoint=fast --xlog-method=stream -R --verbose --progress`
6. Geben Sie das Datenbank-Passwort für den Benutzer `postgres` ein.
 ⇒ Der Primär-Server erstellt ein Backup der Primär-Datenbank und kopiert es in das Verzeichnis `<<PGRES-DATA-FOLDER>>` auf dem Standby-Server.
7. Kopieren Sie die Dateien aus Schritt 2 in das Verzeichnis `<<PGRES-DATA-FOLDER>>` und überschreiben Sie die vorhandenen Dateien.
8. Löschen Sie die Datei `standby.txt` im Verzeichnis `<<PGRES-DATA-DRIVE>>`, sofern sie vorhanden ist.
9. Starten Sie den Dienst `postgresql-x64-n.m- PostgreSQL Server n.m` neu.
HINWEIS! `n.m` steht für die Versionsnummer der verwendeten PostgreSQL, z. B. `postgresql-x64-9.5 - PostgreSQL Server 9.5`.
10. Kontrollieren Sie, ob die Replikation funktioniert, siehe [Kapitel "Funktionstest", S. 34](#).

7.2 Failover-Betrieb mit Failover Configuration Tool zurücksetzen

Falls Sie die Failover-Datenbank nicht wieder zur Standby-Datenbank zurücksetzen möchten, sondern diese zukünftig als primäre Datenbank verwenden möchten, können Sie die Replikations-Richtung umkehren.

Damit wird die Datenbank, die vor dem Einsatz des Failover-Betriebs als Standby-Datenbank verwendet wurde, nach dem Failover-Betrieb künftig als aktive primäre Datenbank verwendet.

7.2.1 Failover-Datenbank als primäre Datenbank übernehmen

Für die Umkehrung der Replikations-Richtung steht im Failover Configuration Tool eine Umkehrfunktion zur Verfügung.

1. Öffnen Sie den Windows Explorer und wechseln Sie in das Verzeichnis `C:\Program Files (x86)\ASC\ASC Product Suite\scripts`.
2. Öffnen Sie mit einem Doppelklick die Datei `DBFailoverConfig.exe`.
 ⇒ Das Failover Configuration Tool erscheint.

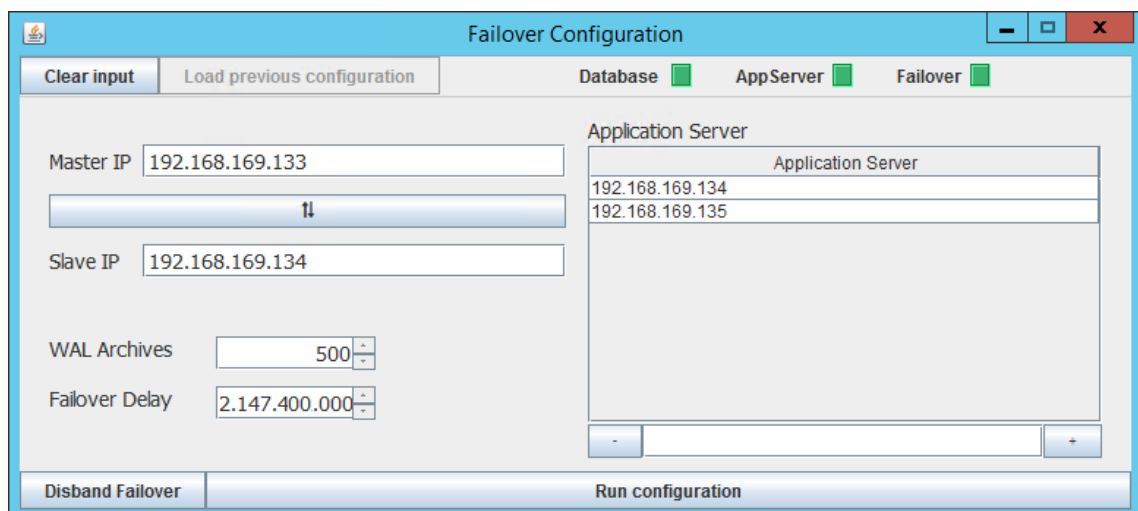
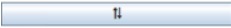


Abb. 22: Failover Configuration Tool - Konfiguration umkehren

3. Klicken Sie auf die Registerkarte `Load previous configuration`, um die vorherige Konfiguration zu laden.

4. Klicken Sie auf die Schaltfläche .
⇒ Damit werden die Einträge von *Master-IP* und *Slave-IP* getauscht.
5. Klicken Sie auf die Schaltfläche *Run configuration*, um die Umkehrung der Konfiguration anzustoßen.
6. Über das Fortschrittsfenster können Sie den Status der Umkehrung verfolgen.
⇒ Wenn die Umkehrung erfolgreich abgeschlossen ist, erscheint eine Erfolgsmeldung.

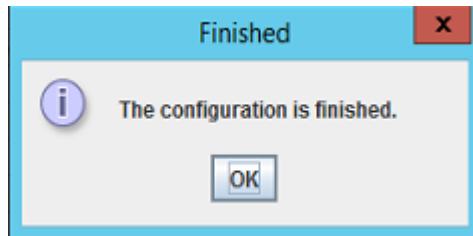


Abb. 23: Erfolgsmeldung

7. Kontrollieren Sie, ob die Replikation funktioniert, siehe [Kapitel "Funktionstest", S. 34](#).

HINWEIS! Das Verzeichnis `<<POSTGRES-DATA-DRIVE>>\Replication`, in dem die WAL-Archives liegen, wird bei der Umkehrung nicht gelöscht. Da das Verzeichnis im Netzwerk freigegeben ist, muss es manuell gelöscht werden.

Anhang

In den folgenden Kapiteln finden Sie beispielhafte Auszüge aus den Dateien, die für die Konfiguration des Failover-Betriebs angepasst werden müssen.

In diesen Auszügen sind die Einträge, die für die Konfiguration des Failover-Betriebs auf jeden Fall ergänzt werden müssen, **fett** markiert.

Abhängig von der Struktur des Gesamtsystems können weitere Einträge vorhanden sein, die durch den Platzhalter "..." gekennzeichnet sind.

Für die verschiedenen Server werden in den Beispielen folgende IP-Adressen verwendet:

- neo-Applikationsserver: **192.168.171.50**, **192.168.171.60**, **192.168.171.70**
- Primär-Server: **192.168.171.110**
- Standby-Server: **192.168.171.120**



Bitte beachten Sie, dass es bei den folgenden Angaben formatierungsbedingt zu ungewollten Zeilenumbrüchen kommen kann, die Sie bei der Eingabe der Befehle nicht übernehmen dürfen.

8.1

Datei pg_hba.conf, Primär-Server

```
...
# TYPE      DATABASE      USER      ADDRESS      METHOD
# IPv4 local connections:
host        all          all        127.0.0.1/32    md5
host        asc_rs        postgres   192.168.171.50/32    md5
host        postgres      postgres   192.168.171.50/32    md5
# IPv6 local connections:
host        all          all        ::1/128        md5
# Allow replication connections from localhost, by a user with the replication privilege.
host        replication    postgres   192.168.171.120/32    md5
...
```

8.2

Datei pg_hba.conf, Standby-Server

```
...
# TYPE      DATABASE      USER      ADDRESS      METHOD
# IPv4 local connections:
host        all          all        127.0.0.1/32    md5
host        asc_rs        postgres   192.168.171.50/32    md5
# IPv6 local connections:
...
```

8.3 Datei postgresql.conf, Primär-Server

```
...
#- Sending Server(s) -
#Set these on the master and on any standby that will send replication data.
max_wal_senders = 10          #max number of walsender processes
                               #change requires restart
wal_keep_segments = 500      #in logfile segments, 16MB each; 0 disables

...

#-----
#WRITE AHEAD LOG
#-----
#- Settings -
wal_level = hot_standby      #minimal, archive, hot_standby, or logical
                               #change requires restart
#fsync = on                  #turns forced synchronization on or off

...

#- Archiving -
archive_mode = on            #allows archiving to be done
                               #change requires restart
archive_command = 'copy %p \\\192.168.171.110\\WAL-ARCHIVE\\%f'
                               #command to use to archive a logfile segment
                               #placeholders: %p = path of file to archive
                               #%f = file name only
                               #e.g. 'test ! -f /mnt/server/archivedir/%f && cp %p /mnt/server/archivedir/%f'
#archive_timeout = 0         #force a logfile segment switch after this
                               #number of seconds; 0 disables

...
```

8.4 Datei postgresql.conf, Standby-Server

```
...
#- Standby Server(s) -
#These settings are ignored on a master server.
hot_standby = on             #"on" allows queries during recovery
```

```

                                #change requires restart
#max_standby_archive_delay = 30s      #max delay before canceling queries
                                #when reading WAL from archive
                                #-1 allows indefinite delay

...

```

8.5 Datei recovery.conf, Standby-Server

```

...
standby_mode = 'on'
primary_conninfo =
'user=postgres password=DatabaseAdmin123# host=192.168.171.110 port=5432 sslmode=prefer sslcompression=1 krbsrvna-
me=postgres'
restore_command = 'copy \\\192.168.171.110\\WAL-ARCHIVE\\%f %p'
trigger_file = 'D:\\standby.txt'
archive_cleanup_command = '"C:\\Program Files\\PostgreSQL\\9.4\\bin\\pg_archivecleanup" \\\192.168.171.110\\WAL-ARCHI-
VE\\ %r'
...

```

8.6 Datei setup.xml, Datenbankserver

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><Setup>
  <Settings>
    ...
    <AIPList>192.168.171.50;192.168.171.60;192.168.171.70</AIPList>
    ...
  </Settings>
</Setup>

```

8.7 Datei setup.xml, Applikationsserver

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><Setup>
  <Settings>
    ...
    <dbip>192.168.171.110</dbip>
    ...
    <databaseMode>FAILOVER</databaseMode>
    <databaseFailoverDelay>60000</databaseFailoverDelay>
  </Settings>

```

</Setup>

8.8 Datei domain.xml, Applikationsserver



Im folgenden Beispiel sind zur besseren Lesbarkeit teilweise Zeilenumbrüche eingefügt. Richten Sie sich nach dem Aufbau der vorhandenen Datei *domain.xml*, wenn Sie die hier gezeigten Beispiele übertragen wollen.

```
...
<resources>
...
    <jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
        validate-atmost-once-period-in-seconds="60"
        datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
        validation-table-name="asc_rs.db_connection_control"
        name="ASCSuitePoolPrimary"
        is-connection-validation-required="true" res-type="javax.sql.ConnectionPoolDataSource">
        <property name="URL" value="jdbc:postgresql://192.168.171.110:5432/asc_rs"></property>
        <property name="user" value="postgres"></property>
        <property name="password" value="{ALIAS=db_password_alias}"></property>
    </jdbc-connection-pool>
    <jdbc-connection-pool fail-all-connections="true"
        validate-atmost-once-period-in-seconds="60"
        datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
        validation-table-name="asc_rs.db_connection_control"
        name="ASCSuitePoolStandby"
        is-connection-validation-required="true" res-type="javax.sql.ConnectionPoolDataSource">
        <property name="URL" value="jdbc:postgresql://192.168.171.120:5432/asc_rs"></property>
        <property name="user" value="postgres"></property>
        <property name="password" value="{ALIAS=db_password_alias}"></property>
    </jdbc-connection-pool>
    <jdbc-connection-pool max-pool-size="60" fail-all-connections="true"
        validate-atmost-once-period-in-seconds="60"
        datasource-classname="org.postgresql.ds.PGConnectionPoolDataSource"
        validation-table-name="asc_rs.db_connection_control"
        name="ASCSuitePoolQuartz"
        is-connection-validation-required="true" res-type="javax.sql.ConnectionPoolDataSource">
        <property name="URL"
```

```

        value="jdbc:postgresql://192.168.171.110:5432,192.168.171.120:5432/asc_rs"></property>
        <property name="user" value="postgres"></property>
        <property name="password" value="${ALIAS=db_password_alias}"></property>
    </jdbc-connection-pool>
    <jdbc-resource pool-name="ASCSuitePoolPrimary"
jndi-name="jdbc/ASCSuitePrimary"></jdbc-resource>
    <jdbc-resource pool-name="ASCSuitePoolStandby"
jndi-name="jdbc/ASCSuiteStandby"></jdbc-resource>
    <jdbc-resource pool-name="ASCSuitePoolQuartz"
jndi-name="jdbc/ASCSuitePoolQuartz"></jdbc-resource>
</resources>
<servers>
    <server config-ref="server-config" name="server">
        ...
        <resource-ref ref="jdbc/ASCSuitePrimary"></resource-ref>
        <resource-ref ref="jdbc/ASCSuiteStandby"></resource-ref>
        <resource-ref ref="jdbc/ASCSuitePoolQuartz"></resource-ref>
    </server>
</servers>
...

```

Abbildungsverzeichnis

Abb. 1	Features zur Installation auswählen.....	9
Abb. 2	Ziellaufwerk für die interne Datenbank wählen	10
Abb. 3	Installation beenden und Server durchstarten	11
Abb. 4	Features zur Installation auswählen.....	12
Abb. 5	Ziellaufwerk für die interne Datenbank wählen	13
Abb. 6	Installation beenden und Server durchstarten	14
Abb. 7	Laufwerk freigeben.....	15
Abb. 8	Freigabe für den Benutzer postgres erteilen.....	16
Abb. 9	Vollzugriff für den Benutzer postgres erteilen	16
Abb. 10	Login für den Postgres-Dienst konfigurieren.....	17
Abb. 11	Failover Configuration Tool - Failover konfigurieren	18
Abb. 12	Failover-Komponenten konfigurieren	18
Abb. 13	Fortschritt der Konfiguration verfolgen	19
Abb. 14	Erfolgsmeldung	20
Abb. 15	Statusanzeige der Failover-Komponenten.....	20
Abb. 16	Farbskala	21
Abb. 17	Failover Configuration Tool erneut aufrufen.....	21
Abb. 18	Laufwerk freigeben.....	23
Abb. 19	Freigabe für den Benutzer postgres erteilen.....	23
Abb. 20	Vollzugriff für den Benutzer postgres erteilen	24
Abb. 21	Formatierungsprinzip	31
Abb. 22	Failover Configuration Tool - Konfiguration umkehren.....	40
Abb. 23	Erfolgsmeldung	41

Tabellenverzeichnis

Glossar

App-Server

Applikationsserver bzw. Web-Server. In den Systemarchitekturen ist das der Server, auf dem der Enterprise Core und die GlassFish-Software installiert sind.

CIDR

Classless Inter-Domain Routing beschreibt ein Verfahren zur effizienteren Nutzung des bestehenden 32-Bit-IP-Adress-Raumes für IPv4. Mit CIDR entfällt die feste Zuordnung einer IPv4-Adresse zu einer Netzkategorie, aus welcher die Präfixlänge hervorging. Die Präfixlänge ist mit CIDR frei wählbar und muss deshalb beim Aufschreiben eines IP-Subnetzes mit angegeben werden. Dazu verwendet man häufig eine Netzmaske. (Quelle: Wikipedia 03.04.2017)

FQDN

Fully Qualified Domain Name

IP

Internet Protocol, Basisprotokoll für die Internetkommunikation

Multi-Core-System

Aufzeichnungssystem, in dem mehrere Applikationsserver (Enterprise Core) verwendet werden.

Partitionstoleranz

Begriff aus dem CAP-Theorem in Bezug auf verteilte Systeme. In Bezug auf das Failover-Konzept der Datenbank steht der Begriff dafür, dass bei einer physikalischen Trennung der beiden Datenbankserver die Datenbestände in beiden Datenbanken (primär und standby) verfügbar UND konsistent bleiben sollten.

Single-Core-System

Aufzeichnungssystem, in dem nur 1 Applikationsserver (Enterprise Core) verwendet wird. Es ist aber möglich, dass andere Komponenten (z.B. Aufzeichnungskomponenten) auf weiteren Servern installiert sind. Ein solches System wäre ein Multi-Server-/Single-Core-System.

WAL

Write-Ahead Logging ist ein Verfahren der Datenbanktechnologie, das zur Gewährleistung der Atomarität und Dauerhaftigkeit von Transaktionen beiträgt. Es besagt, dass Modifikationen vor dem eigentlichen Schreiben (dem Einbringen in die Datenbank) protokolliert werden müssen. (Quelle: Wikipedia 22.02.2017)

Wartungsfenster

Zeitraum (Zeitfenster), in dem eine Wartung durchgeführt wird. In diesem Zeitraum steht in der Regel der Betrieb der gewarteten Komponenten still oder ist nur eingeschränkt möglich.