Microsoft

SQL Server 2025: the AI-ready enterprise database from ground to cloud

CLUSTERPRO® X *for Windows*
ソフトウェア構築ガイド
(Microsoft SQL Server)

2026.6.29

第 8 版

CLUSTERPRO

改版履歴

版数	改版日付	内 容
1	2018/06/25	新規作成
2	2021/03/09	Microsoft SQL Server 2019 関連内容を追記 Microsoft SQL Server 2008、Microsoft SQL Server 2008 R2 対応に関する記載を変更 SQL Server サービス起動アカウントに関する記載を変更
3	2021/12/24	Microsoft SQL Server Reporting Services 関連内容を追記
4	2022/07/20	アタッチ時のデータベースの所有者に関する内容を追記 SQL Server 2012 対応に関する記載を変更
5	2023/05/26	Microsoft SQL Server 2022 関連内容を追記 ARM コマンド廃止のためスクリプトを修正
6	2024/07/31	CLUSTERPRO X 5.2 関連内容を追記
7	2025/05/31	CLUSTERPRO X 5.3 関連内容を追記
8	2026/06/04	CLUSTERPRO X 6.0、Microsoft ODBC Driver 18 for SQL Server Power BI Report Server (PBIRS) Web ポータル関連内容を追記

© Copyright NEC Corporation 2012-2026. All rights reserved.

免責事項

本書の内容は、予告なしに変更されることがあります。
日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。
また、お客様が期待される効果を得るために、本書に従った導入、使用および使用効果につきましては、
お客様の責任とさせていただきます。
本書に記載されている内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部または全部を日本電気株式会社の許諾なしに複製、改変、および翻訳することは禁止されています。

商標情報

CLUSTERPRO® X は日本電気株式会社の登録商標です。
Microsoft、Microsoft Windows、Microsoft SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
本書に記載されたその他の製品名および標語は、各社の商標または登録商標です。
その他のシステム名、社名、製品名等はそれぞれの会社の商標及び登録商標です。

目次

目次 5
はじめに 6
1. 対象読者と目的 6
2. 適用範囲 6
3. お問い合わせについて 7
4. CLUSTERPRO マニュアル体系 7
5. 最新情報の入手先 7

Microsoft SQL Server 8
機能概要 8
1. 片方向スタンバイ型 8
2. 双方向スタンバイ型 9
機能範囲 12
構築手順 13
1. フェイルオーバグループの作成 14
2. Microsoft SQL Server のインストール 14
3. ユーザデータベースの作成 14
4. Microsoft SQL Server のスクリプト作成 16
5. CLUSTERPRO への Microsoft SQL Server サービスの組み込み 18
6. 監視リソースの設定 34
7. ログイン情報の引き継ぎ 34
8. 暗号化設定の引き継ぎ 37
9. Microsoft SQL Server Reporting Services の組み込み 39
注意事項 49
1. CLUSTERPRO によるフェイルオーバが利用できない機能について 49
2. Microsoft SQL Server のクラスタ構成の注意事項について 50
3. 片方向スタンバイ構成における注意事項について 51
4. 双方向スタンバイ構成における注意事項について 51
5. データファイル格納ディスク破損時のログ末尾のバックアップに関する注意事項について 56
6. Microsoft SQL Server Agent の機能を使用する場合の注意事項について 57
7. ポリシーベースの管理機能を使用する場合の注意事項について 58
8. FILESTREAM 機能を使用する場合の注意事項について 58
9. 変更データキャプチャ（CDC）機能を使用する場合の注意事項について 58
10. その他の機能を使用する場合の注意事項について 58
11. Microsoft ODBC Driver 18 for SQL Server 以降をご利用される場合 59
12. SSRS および PBIRS の Web ポータル 表示について 59
その他 60

はじめに

1. 対象読者と目的

『CLUSTERPRO®ソフトウェア構築ガイド』は、クラスタシステムに関して、システムを構築する管理者、およびユーザサポートを行うシステムエンジニア、保守員を対象にしています。
本書では、CLUSTERPRO 環境下での動作確認が取れたソフトウェアをご紹介しています。ここでご紹介するソフトウェアや設定例は、あくまで参考情報としてご提供するものであり、各ソフトウェアの動作保証をするものではありません。

2. 適用範囲

本書は、以下の製品を対象としています。
CLUSTERPRO X 6.0 for Windows
CLUSTERPRO X 5.3 for Windows
CLUSTERPRO X 5.2 for Windows
CLUSTERPRO X 5.1 for Windows
CLUSTERPRO X 5.0 for Windows
CLUSTERPRO X 4.3 for Windows
CLUSTERPRO X 4.2 for Windows
CLUSTERPRO X 4.1 for Windows
CLUSTERPRO X 4.0 for Windows

Microsoft SQL Server 2025 Standard / Enterprise / Standard Developer / Enterprise Developer
Microsoft SQL Server 2022 Standard / Enterprise / Developer
Microsoft SQL Server 2019 Standard / Enterprise / Developer
Microsoft SQL Server 2017 Standard / Enterprise / Developer
Microsoft SQL Server 2016 Standard / Enterprise / Developer
Microsoft SQL Server 2014 Standard / Enterprise / Business Intelligence
Microsoft SQL Server 2012 Standard / Enterprise / Business Intelligence

Power BI Report Server (PBIRS)
Microsoft SQL Server 2022 Reporting Services
Microsoft SQL Server 2019 Reporting Services
Microsoft SQL Server 2017 Reporting Services

※ 上記に記載のないエディションは、本書の適用対象外となります。

3. お問い合わせについて

本書の Microsoft SQL Server 製品に関する記載内容のお問い合わせには、原則として CLUSTERPRO の保守契約と Microsoft SQL Server の弊社での保守契約が必要です。

Microsoft SQL Server 製品の障害発生時には、保守契約に則り、以下の NEC サポートポータルから NEC SQL Server Response Center までお問い合わせください。
NEC サポートポータル（https://www.support.nec.co.jp/）

4. CLUSTERPRO マニュアル体系

CLUSTERPRO のマニュアルは、以下の 4 つに分類されます。各ガイドのタイトルと役割を以下に示します。

『CLUSTERPRO X スタートアップガイド』（Getting Started Guide）
CLUSTERPRO を使用するユーザを対象読者とし、製品概要、動作環境、アップデート情報、既知の問題などについて記載します。

『CLUSTERPRO X インストール & 設定ガイド』（Install and Configuration Guide）
CLUSTERPRO を使用したクラスタシステムの導入を行うシステム エンジニアと、クラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO を使用したクラスタシステム導入から運用開始前までに必須の事項について説明します。実際にクラスタシステムを導入する際の順番に則して、CLUSTERPRO を使用したクラスタシステムの設計方法、CLUSTERPRO のインストールと設定手順、設定後の確認、運用開始前の評価方法について説明します。

『CLUSTERPRO X リファレンス ガイド』（Reference Guide）
管理者、および CLUSTERPRO を使用したクラスタシステムの導入を行うシステム エンジニアを対象とし、CLUSTERPRO の運用手順、各モジュールの機能説明およびトラブルシューティング情報等を記載します。『インストール & 設定ガイド』を補完する役割を持ちます。

『CLUSTERPRO X メンテナンスガイド』（Maintenance Guide）
管理者、および CLUSTERPRO を使用したクラスタシステム導入後の保守・運用を行うシステム管理者を対象読者とし、CLUSTERPRO のメンテナンス関連情報を記載します。

5. 最新情報の入手先

最新の製品情報については、以下の Web サイトを参照してください。

https://jpn.nec.com/clusterpro

Microsoft SQL Server

機能概要

Microsoft SQL Server を、CLUSTERPRO 環境下で利用する際の機能概要について以下に記述します。なお、Microsoft SQL Server のバージョンにより異なる箇所については、それぞれ個別に説明します。
CLUSTERPRO 環境下での Microsoft SQL Server の運用は、片方向スタンバイ型と双方向スタンバイ型があります。
クライアントは、通常、ODBC などを使用して現用系にアクセスします。現用系に障害が発生した場合、クライアントは待機系に接続し、運用することになります。(双方向スタンバイ型ではそれぞれが現用系、待機系となります。)

1. 片方向スタンバイ型

図 1.1 は、サーバ 1 を現用系、サーバ 2 を待機系とした片方向スタンバイ型の CLUSTERPRO 環境を構成して動作させるときのイメージ図です。
クライアントからは、フローティング IP アドレスや仮想コンピュータ名を使用して、ODBC などにより接続します。

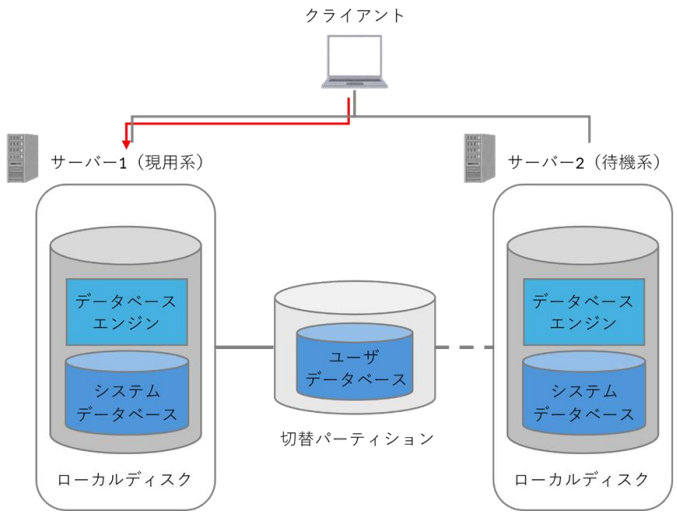


図 1.1

サーバ 1 に障害が発生すると図 1.2 のようになります。

フェイルオーバが完了すると、サーバ 2 上で Microsoft SQL Server のサービスが立ち上がり、切替パーティションのリソースがサーバ 2 へ移行するため、クライアントはサーバ 2 へ接続し、運用することになります。
フローティング IP アドレスにてサーバへ接続をしている場合は、フェイルオーバにてフローティング IP アドレスがサーバ 2 へ移行するため、クライアントはサーバが切り替わったことを意識せずに、同一の IP アドレスで再接続することが可能です。

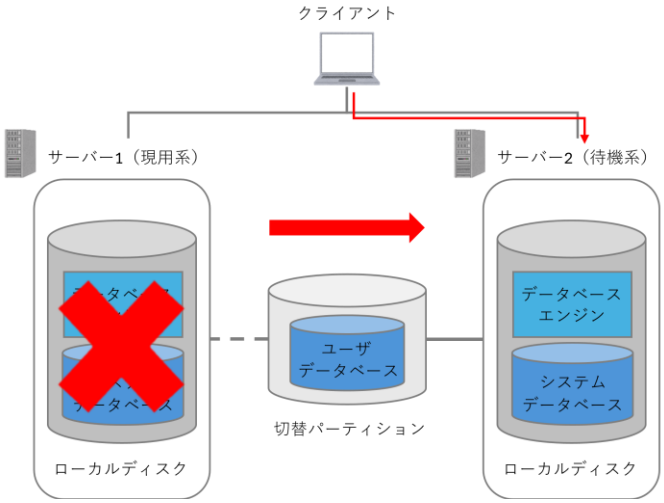


図 1.2

2. 双方向スタンバイ型

図 1.3 は、双方向スタンバイ型を CLUSTERPRO 環境下で動作させるときのイメージ図です。

双方向スタンバイ型の場合は以下のように構成します。

- サーバ 1 を現用系、サーバ 2 を待機系とするクラスタ グループを作成する。(図 1.3 の場合、切替パーティション 1 を使用します。)
- サーバ 2 を現用系、サーバ 1 を待機系とするクラスタ グループを作成します。(図 1.3 の場合、切替パーティション 2 を使用します。)

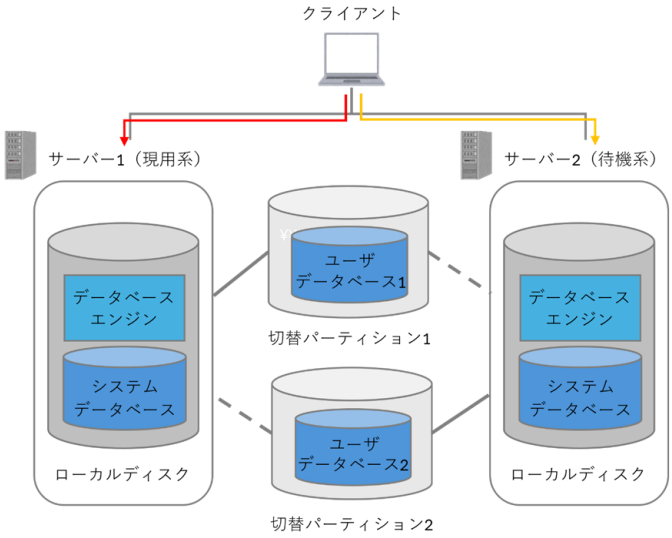


図 1.3

サーバ 1 に障害が発生すると、図 1.4 のようになります。

フェイルオーバが発生すると、サーバ 1 の切替パーティションのリソースがサーバ 2 に移行します。
この時、サーバ 2 の Microsoft SQL Server は 2 つのクラスタグループのユーザデータベースを持つことになります。サーバ 1（ユーザデータベース 1）にアクセスしていたクライアントは、サーバ 2 へ接続し、運用することになります。
フローティング IP アドレスにてサーバへ接続している場合は、フェイルオーバにてフローティング IP アドレスがサーバ 2 へ移行する為、クライアントはサーバが切り替わったことを意識せずに、同一の IP アドレスで再接続することが可能です。

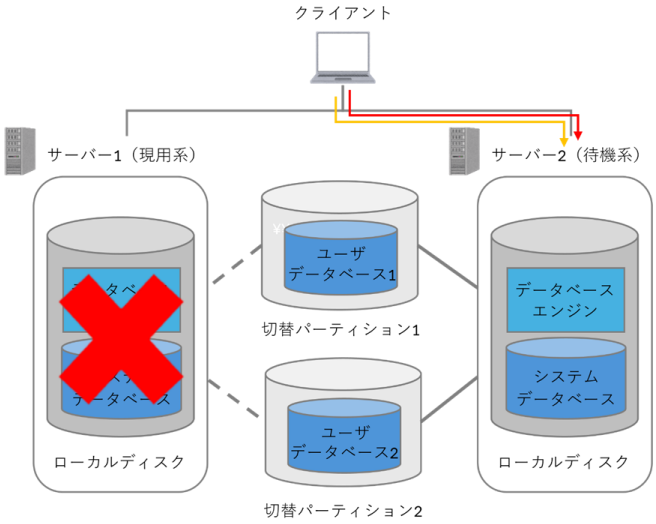


図 1.4

なお、図 1.5 のようにマルチインスタンス構成において片方向スタンバイ型を組み合わせ、双方向スタンバイ型として運用することも可能です。
右図の場合は、サーバ 1 を現用系とする片方向スタンバイ型のインスタンスと、サーバ 2 を現用系とする片方向スタンバイ型のインスタンスをそれぞれ作成し、双方向スタンバイ型を実現しています。

そのため、下図のようなマルチインスタンス構成における双方向スタンバイ型を構築する場合は、インスタンスごとに片方向スタンバイ型の作成手順に沿って構築する必要があります。

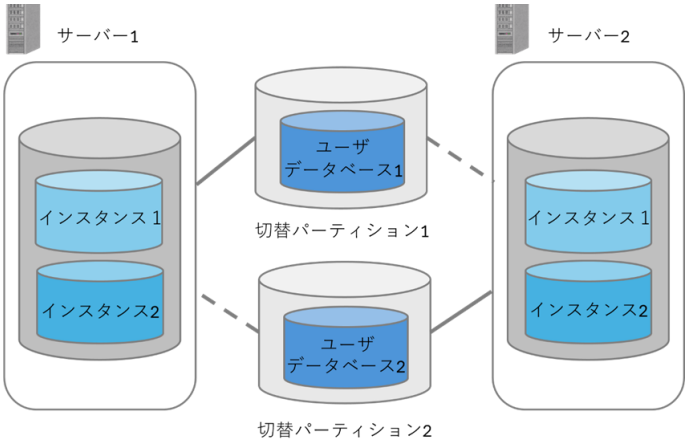


図 1.5

機能範囲

- CLUSTERPRO環境で Microsoft SQL Serverを利用する場合、システムデータベース（master、msdbなど）は、それぞれのノードのローカルディスク上に格納する必要があります。
  切替パーティション上にシステムデータベースを配置することはできません。

- システムデータベースで管理される情報（ログインやジョブ情報等）はフェイルオーバにより待機系サーバへ引き継がれません。

- Microsoft SQL Server 2014 Standard / Enterprise / Business Intelligenceから
  Microsoft SQL Server 2022 Standard / Enterpriseまでは、既定インスタンス、および名前付きインスタンスでの動作を確認しております。
  なお、以下のバージョンに関しては、Microsoft SQL ServerのPP・サポートサービス契約を締結の上、当該窓口へお問い合わせください。

  ・Microsoft SQL Server 2005 Standard / Enterprise
  ・Microsoft SQL Server 2008 Standard / Enterprise
  ・Microsoft SQL Server 2008 R2 Standard / Enterprise / Datacenter
  ・Microsoft SQL Server 2012 Standard / Enterprise / Business Intelligence

- Microsoft SQL Server 2017 Reporting Services、Microsoft SQL Server 2019 Reporting Services、Microsoft SQL Server 2022 Reporting Services での動作を確認しております。
  なお、以下のバージョンに関しては Microsoft SQL ServerのPP・サポートサービスを締結の上、当該窓口へお問い合わせください。

  ・Microsoft SQL Server 2005 Reporting Services
  ・Microsoft SQL Server 2008 Reporting Services
  ・Microsoft SQL Server 2008 R2 Reporting Services
  ・Microsoft SQL Server 2012 Reporting Services
  ・Microsoft SQL Server 2014 Reporting Services
  ・Microsoft SQL Server 2016 Reporting Services

構築手順

Microsoft SQL Server の CLUSTERPRO 環境構築は以下の流れで行います。

1. フェイルオーバグループの作成
2. Microsoft SQL Server のインストール
3. ユーザデータベースの作成
4. Microsoft SQL Server のスクリプト作成
5. CLUSTERPRO への Microsoft SQL Server サービスの組み込み
6. 監視リソースの設定
7. ログイン情報の引き継ぎ
8. 暗号化設定の引き継ぎ
9. Microsoft SQL Server Reporting Services の組み込み

1. フェイルオーバグループの作成

CLUSTERPRO でフェイルオーバグループを作成します。フェイルオーバグループには、以下のリソースが必要です。

- フローティングIPアドレス／仮想コンピュータ名
- 切替パーティション（ユーザデータベースファイルを格納する十分な容量をもったもの）

2. Microsoft SQL Server のインストール

各サーバのローカルディスク上に Microsoft SQL Server をインストールします。
Microsoft SQL Server 本体、およびデータディレクトリで指定するフォルダは、必ずローカルディスクを指定するようにしてください。
名前付きインスタンスをインストールする場合、現用系と待機系でインスタンス名は同じ名前に統一してください。
CLUSTERPRO 環境では、ユーザデータベースファイルのみを切替パーティションに作成します。
また、Microsoft SQL Server 関連のサービスの開始モードは全て「手動」に設定（自動起動を行わないように構成）します。

3. ユーザデータベースの作成

[1] 現用系での作業

フェイルオーバ対象となるユーザデータベースの作成は、現用系から行います。
ユーザデータベースは、切替パーティション上に作成します。
以下の例では、切替パーティション上（ここではドライブ文字を「Y」に設定）に、TESTDB という名前のデータベース（データファイル初期サイズ 10MB、ログファイル初期サイズ 10MB）を作成しています。

【データベース作成例】

以下のクエリを Microsoft SQL Server Management Studio（以降、SSMS と表記）から実行します。

```
/* TESTDB_Data、TESTDB_Log の 2 つのファイルから TESTDB という DB を作成 */
create database TESTDB on PRIMARY (
   name = 'TESTDB_Data',
   filename = 'Y:¥sql¥data¥TESTDB_Data.mdf',
   size = 10
)
LOG ON (
   name = 'TESTDB_Log',
   filename = 'Y:¥sql¥data¥TESTDB_Log.ldf',
   size = 10
)
go
CHECKPOINT
go
```

※「透過的データ暗号化」機能を使用して対象のデータベースの暗号化を行いたい場合、ここではまだ暗号化設定を行わないようにします。暗号化設定を行う手順については、後述の「8. 暗号化設定の引き継ぎ」にて記載しております。

データベースは、SSMS による GUI 操作から作成することもできます。データファイルとログファイルを切替パーティション上に作成する以外は、通常のデータベース作成と違いはありません。

なお、双方向スタンバイ型の構成の場合、2 台のサーバでそれぞれユーザデータベースを作成する必要がありますが、データベース ID（dbid）を現用系と待機系で一致させる運用とすることを推奨しています。（注 1）
たとえば、サーバ 1 を現用系とするフェイルオーバグループのユーザデータベースとして db1、サーバ 2 を現用系とするフェイルオーバグループのユーザデータベースとして db2 を作成する状況を考えます。以下は、サーバ 1 で db1 を作成した際の dbid が 7 となる場合の作成例となります。（注 2）

1. サーバ 1 で db1 を作成（dbid=7）
2. サーバ 2 でダミーのデータベースを作成（dbid=7）
3. サーバ 2 で db2 を作成（dbid=8）
4. サーバ 2 でダミーのデータベースを削除

（注 1）フェイルオーバにより待機系へ切り替わった際にも、現用系と同じ dbid となるよう構成することを目的としています。
データベースの dbid は、以下のクエリを実行することで確認できます。以下のクエリの実行結果から、対象データベースの[dbid]列の値を確認します。

```
exec sp_helpdb
go
```

（注 2）dbid 1 ～ 6 に割り当てられているデータベースがフェイルオーバ対象のデータベースではない（デタッチ/アタッチが行われない）データベースであることが前提となります。

[2] 待機系での作業

待機系では、データベースの作成を行う必要はありません。

4. Microsoft SQL Server のスクリプト作成

CLUSTERPRO によるフェイルオーバ、およびフェイルバックが行われる際には、対象となるユーザデータベース（フェイルオーバデータベース）のデタッチ／アタッチが必要となります。
以下は、アタッチを行うスクリプト（ACT.SQL）とデタッチを行うスクリプト（DEACT.SQL）の記述例となります。作成した各スクリプトを、各ノードの任意のフォルダに格納します。（注 3）
なお、Microsoft SQL Server のスクリプトは、ご利用の Microsoft SQL Server バージョンによって使用するスクリプトが異なっておりますので、ご注意ください。

（注 3）ローカルドライブに格納します。切替パーティション上（共有ディスク、ミラーディスク）には格納しないでください。

[A] 片方向スタンバイ型

- フェイルオーバデータベースが複数存在している場合は、そのそれぞれについて "create database for attach"／"sp_detach_db" を実行する必要があります。

- DEACT.SQLでalter database 〜 set offlineを記載しなかった場合、フェイルオーバ時にMicrosoft SQL Serverサービスの起動に失敗する恐れがあります。

- ACT.SQL

```
create database [<サーバ 1 上フェイルオーバデータベース名>] on
   (filename='<物理ファイル名>'),
   (filename='<物理ファイル名>')
for attach
```

例）『ユーザデータベースの作成』 で作成した TESTDB を使用する場合

```
create database [TESTDB] on
   (filename = 'Y:¥sql¥data¥TESTDB_Data.mdf'),
   (filename = 'Y:¥sql¥data¥TESTDB_Log.ldf')
for attach
```

- DEACT.SQL

```
alter database [<サーバ 1 上フェイルオーバデータベース名>] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db '<サーバ 1 上フェイルオーバデータベース名>',TRUE
```

例）『ユーザデータベースの作成』 で作成した TESTDB を使用する場合

```
alter database [TESTDB] set offline with ROLLBACK IMMEDIATE
exec sp_detach_db 'TESTDB',TRUE
```

[B] 双方向スタンバイ型

◆ フェイルオーバグループごとに ACT.SQL と DEACT.SQL を作成する必要があります。

◆ フェイルオーバデータベースが複数存在している場合は、そのそれぞれについて “create database for attach”／"sp_detach_db" を実行する必要があります。

◆ DEACT1.SQLとDEACT2.SQLで alter database ～ set offline を記載しなかった場合、フェイルオーバ時にMicrosoft SQL Serverサービスの起動に失敗する恐れがあります。

◆ ACT1.SQL

```
create database [<サーバ 1 上フェイルオーバデータベース名>] on
   (filename='<物理ファイル名>'),
   (filename='<物理ファイル名>')
for attach
```

◆ DEACT1.SQL

```
alter database [<サーバ 1 上のフェイルオーバデータベース名>] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db '<サーバ 1 上のフェイルオーバデータベース名>',TRUE
```

◆ ACT2.SQL

```
create database [<サーバ 2 上フェイルオーバデータベース名>] on
   (filename='<物理ファイル名>'),
   (filename='<物理ファイル名>')
for attach
```

◆ DEACT2.SQL

```
alter database [<サーバ 2 上のフェイルオーバデータベース名>] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db '<サーバ 2 上のフェイルオーバデータベース名>',TRUE
```

5. CLUSTERPRO への Microsoft SQL Server サービスの組み込み

Microsoft SQL Server サービスの起動制御を CLUSTERPRO から行う様に設定します。
なお、Microsoft SQL Server サービスの起動をスクリプトリソースで行う方法と、サービスリソースで行う方法があり、構築手順が一部異なりますので、注意してください。
「4. Microsoft SQL Server のスクリプト作成」で作成したスクリプトファイル（ACT.SQL／DEACT.SQL または ACT1.SQL／DEACT1.SQL／ACT2.SQL／DEACT2.SQL）を、ローカルドライブ上の任意のフォルダへ格納します。
以下の CLUSTERPRO スクリプトの記述例では、スクリプトファイルの格納先を C:¥mssql としています。

【方法 1】 Microsoft SQL Server サービスの起動をサービスリソースで制御する場合

サービスリソースは、CLUSTERPRO でサービスの管理を行う機能です。
サービスリソース機能を使用する際にサービス監視を行う場合には、別途サービス監視リソースの設定を行う必要がありますので、ご留意ください。サービス監視リソースの設定を行う手順等の詳細については、「CLUSTERPRO X リファレンスガイド」 のサービス監視リソースの箇所を参照してください。

サービスリソースで Microsoft SQL Server サービスの起動を管理するには、以下 2 つのリソースを登録します。

① サービスリソース
「MSSQLSERVER」 を登録します。（名前付きインスタンスの場合は、「MSSQL$<対象のインスタンス名>」 を登録します。）

② スクリプトリソース
データベースのデタッチとアタッチを実行するためのスクリプトリソースを作成し、サービスリソースとの依存関係を設定します。

※ 各リソースの依存関係として、Microsoft SQL Server では以下のように設定する必要があります。そのため、サービスリソース、スクリプトリソースでは「既存の依存関係に従う」のチェックをオフにして個別に設定を行います。

[1] サービスリソースの依存関係に、ディスクリソースを追加します
[2] スクリプトリソースの依存関係に、サービスリソースおよびディスクリソースを追加します

サービスリソースで Microsoft SQL Server サービスの起動を管理する場合、「対象 VCOM リソース名」は設定しないでください。

サービスリソースを利用する場合、スクリプトリソースでは Microsoft SQL Server サービスの制御を行わないため、開始スクリプト、および終了スクリプト内に起動（net start）／停止（net stop）のコマンドを記述する必要はありません。
なお、サービスリソースを設定する際の各設定項目の詳細につきましては、「CLUSTERPRO X リファレンスガイド」 のサービスリソースに関する箇所を参照してください。

また、データベースのデタッチとアタッチを実行するスクリプトリソースでは、実行時の状況に応じて処理を変更できるように、環境変数に実行状況を示す値が設定される構成となっています。デフォルトで作成されるテンプレートに、これらの環境変数の値による条件分岐が用意されています。
以下の開始／終了処理がそれぞれの環境変数の値に応じて実行されるように構成します。

次ページに、スクリプトリソースの記載例を示します。
※以降、sqlcmd コマンド例では Microsoft SQL Server 認証を利用しています。
※Microsoft ODBC Driver 18 for SQL Server を利用される場合は、注意事項の
「11.Microsoft ODBC Driver 18 for SQL Server を利用する場合」 をご参照ください。

[A] 片方向スタンバイ型

◆ 開始スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .¥<インスタンス名>
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .¥<インスタンス名>
```

※ 次の例は、上記の記述を行った状態の開始スクリプト（start.bat）の一部です。環境に応じて適宜修正してください。
※ データベースの所有者は、アタッチ毎に指定されたログインアカウントが設定されます。次の例では、赤線枠内で記載の sa になります。

```
rem ***************************************
rem  通常起動対応処理
rem ***************************************
:NORMAL


rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************


rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT

:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT
```

```
rem ***************************************
rem  フェイルオーバ対応処理
rem ***************************************
:FAILOVER

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  フェイルオーバ後の業務起動ならびに復旧処理
rem *************


rem  プライオリティ  のチェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER2

rem *************
rem  最高プライオリティ  での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT

:ON_OTHER2
rem *************
rem  最高プライオリティ  以外での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT
```

◆ 終了スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .¥<
インスタンス名>
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .¥<
インスタンス名>
```

※ 次の例は、上記の記述を行った状態の終了スクリプト（stop.bat）の一部です。環境に応じて適宜修正してください。

```
rem ***************************************
rem  通常終了対応処理
rem ***************************************
:NORMAL

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************


rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .

GOTO EXIT


:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .

GOTO EXIT
```

```
rem ************************************
rem  フェイルオーバ対応処理
rem ************************************
:FAILOVER

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  フェイルオーバ後の業務起動ならびに復旧処理
rem *************


rem  プライオリティ  のチェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER2

rem *************
rem  最高プライオリティ  での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .

GOTO EXIT

:ON_OTHER2
rem *************
rem  最高プライオリティ  以外での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .

GOTO EXIT
```

[B] 双方向スタンバイ型

◆ 開始スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .¥<インスタンス名>
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .¥<インスタンス名>
```

◆ 終了スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -S .¥<インスタンス名>
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -S .¥<インスタンス名>
```

◆ サーバ2 を現用系とするフェイルオーバグループのスクリプトにも、同様に上記の処理を追加してください。
なお、サーバ2 のスクリプトを作成する際は、上記の例を以下のように読み替えてください。

- ACT1.SQL → ACT2.SQL
- ACT1.LOG → ACT2.LOG
- DEACT1.SQL → DEACT2.SQL
- DEACT1.LOG → DEACT2.LOG

◆ サービスリソース調整プロパティの［サービス］タブにて、「サービスが起動済みの場合、エラーとしない」にチェックしてください。
双方向スタンバイ型構成の場合、フェイルオーバ先で既にMicrosoft SQL Serverサービスが稼働中の状態となりますので、これをチェックしない場合、フェイルオーバ時にサービスリソースのアクティブ化でエラーとなり、フェイルオーバに失敗します。

【方法 2】 Microsoft SQL Server サービスの起動をスクリプトリソースで制御する場合

スクリプトリソースでは、実行時の状況に応じて処理を変更できるように、環境変数に実行状況を示す値が設定される構成となっています。デフォルトで作成されるテンプレートに、これらの環境変数の値による条件分岐が用意されています。
開始／終了処理がそれぞれの環境変数の値に応じて実行されるように構成します。

以下に、スクリプトリソースの記載例を示します。

[A] 片方向スタンバイ型

◆ 開始スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
net start MSSQLSERVER
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
```

【名前付きインスタンスの場合】

```
net start MSSQL$<インスタンス名>
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .¥<インスタンス名>
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
net start MSSQLSERVER
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
```

【名前付きインスタンスの場合】

```
net start MSSQL$<インスタンス名>
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .¥<インスタンス名>
```

※ 次の例は、上記の記述を行った状態の開始スクリプト（start.bat）の一部です。環境に応じて適宜修正してください。
※　データベースの所有者は、アタッチ毎に指定されたログインアカウントが設定されます。次の例では、赤線枠内で記載の sa になります。

```
rem ************************************
rem  通常起動対応処理
rem ************************************
:NORMAL


rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************


rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S

GOTO EXIT

:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT
```

```
rem ************************************
rem  フェイルオーバ対応処理
rem ************************************
:FAILOVER

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  フェイルオーバ後の業務起動ならびに復旧処理
rem *************


rem  プライオリティ  のチェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER2

rem *************
rem  最高プライオリティ  での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT

:ON_OTHER2
rem *************
rem  最高プライオリティ  以外での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .

GOTO EXIT
```

◆ 終了スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .¥<インスタンス名>
net stop MSSQL$<インスタンス名>
timeout /t 10 > nul
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .¥<インスタンス名>
net stop MSSQL$<インスタンス名>
timeout /t 10 > nul
```

◇ timeout コマンドは、Microsoft SQL Server サービスの停止時にキャッシュ上の情報がディスクに書き込まれるのを待ち合わせるためのものとなります。
引数に指定する待ち合わせ時間はサービス停止時にディスクにフラッシュされていない情報量に依存します。基本的には 10 秒程度で問題ありませんが、検証の上、判断する必要があります。

※ 次の例は、上記の記述を行った状態の終了スクリプト（stop.bat）の一部です。環境に応じて適宜修正してください。

```
rem ************************************
rem  通常終了対応処理
rem ************************************
:NORMAL

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************


rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************
sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul

GOTO EXIT


:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************
sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul

GOTO EXIT
```

```
rem ************************************
rem  フェイルオーバ対応処理
rem ************************************
:FAILOVER

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  フェイルオーバ後の業務起動ならびに復旧処理
rem *************


rem  プライオリティ  のチェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER2

 rem *************
 rem  最高プライオリティ  での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul

GOTO EXIT

:ON_OTHER2
rem *************
rem  最高プライオリティ  以外での処理
rem *************

sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul

GOTO EXIT
```

[B] 双方向スタンバイ型

◆ 開始スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
net start MSSQLSERVER
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
net start MSSQL$<インスタンス名>
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .¥<インスタンス名>
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥ACT1.SQL -o C:¥mssql¥ACT1.LOG -S .¥<インスタンス名>
```

◆ 終了スクリプト例

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「HOME」(「OTHER」 ではない)

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o
C:¥mssql¥DEACT1.LOG -S .¥<インスタンス名>
net stop MSSQL$<インスタンス名>
timeout /t 10 > nul
```

◇ %CLP_EVENT% が 「START」 または 「FAILOVER」
%CLP_SERVER% が 「OTHER」

【既定インスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -S .
```

【名前付きインスタンスの場合】

```
sqlcmd -U sa -P <パスワード> -i C:¥mssql¥DEACT1.SQL -o C:¥mssql¥DEACT1.LOG -
S .¥<インスタンス名>
```

◇ サーバ 2 を現用系とするフェイルオーバグループのスクリプトにも、同様に上記の処理を追加してください。
なお、サーバ 2 のスクリプトを作成する際は、上記の例を以下のように読み替えてください。

・ ACT1.SQL → ACT2.SQL
・ ACT1.LOG → ACT2.LOG
・ DEACT1.SQL → DEACT2.SQL
・ DEACT1.LOG → DEACT2.LOG

6. 監視リソースの設定

フェイルオーバ対象とするデータベースを監視するため、CLUSTERPRO 上で Microsoft SQL Server 監視リソースを設定します。Microsoft SQL Server 監視リソースの詳細や設定方法は、「CLUSTERPRO X リファレンスガイド」の Microsoft SQL Server 監視リソースの箇所を参照してください。

7. ログイン情報の引き継ぎ

現用系側で Microsoft SQL Server 認証ログインを作成した場合、待機系側へのフェイルオーバ後に当該ログインを有効にするには、以下のいずれかの方法を実施する必要があります。
sa については SID が固定のため、本対処は不要です。

【方法 1】 現用系と待機系で同じログインを作成する
【方法 2】 包含データベースを使用する

【方法 1】では稼動系と待機系それぞれで、同名かつ同一 SID のログインを作成する必要があります。【方法 2】ではフェイルオーバ対象の包含データベースにログインを作成するため、各データベースにログインを作成する必要があります。このため運用にあわせて対処方法を選択してください。

各方法の詳細手順を以下に記載しています。

【方法 1】の手順 1)、3）および【方法 2】の手順 1)、3)、5)、6）については SSMS からの GUI 操作でも実施可能です。ただし、【方法 1】の手順 2)、4）については、クエリによる実行が必要となります。

【方法 1】 現用系と待機系で同じログインを作成する

手順 1）

現用系側でログインを作成します。

ここでは、ログイン名を「TestLogin」、パスワードを「PassWord」、既定のデータベースを「TESTDB」としてログインを作成する例を示します。

```
create login TestLogin with password = 'PassWord', default_database = TESTDB
```

手順 2）

手順 1）で作成したログインの SID を記録します。この SID は、待機系側で同一のログインを作成するために必要となります。

ログインの SID は以下のクエリを実行することで確認することができます。

```
select SUSER_SID('TestLogin')
```

手順 3）

現用系側にて、フェイルオーバ対象のデータベース上にユーザを作成します。

ここでは、手順 1）で作成したログイン「TestLogin」 に対するユーザ「TestUser」をデータベース「TESTDB」に作成する例を示します。

```
use TESTDB
go
create user TestUser for login TestLogin
go
```

手順 4）

対象のデータベースが存在するフェイルオーバグループを待機系側へフェイルオーバします。

フェイルオーバ完了後、待機系側で対象のデータベースへアクセスできることを確認してください。

手順 5）

待機系側にて、現用系側と同一のログインを作成します。

ここでは、手順 1）で作成したログイン「TestLogin」と同一のログインを作成する例を示します。

```
create login TestLogin with password = 'PassWord',
   SID = <手順 2 で記録した SID>, default_database = TESTDB
go
```

上記の通り、同一のログインを作成するには、ログインの SID を一致させてログインを作成する必要があります。SID 以外の項目の設定が同じであっても、SID が一致していない場合は異なるログインと認識されます。

【方法 2】 包含データベースを使用する

手順 1）

現用系側で包含データベースを有効化します。

```
sp_configure 'contained database authentication', 1
reconfigure
go
```

手順 2）

対象の Microsoft SQL Server 用のフェイルオーバグループを待機系側へフェイルオーバします。

手順 3）

待機系側で包含データベースを有効化します。

```
sp_configure 'contained database authentication', 1
reconfigure
go
```

手順 4）

対象の Microsoft SQL Server 用のフェイルオーバグループを現用系側へフェイルバックします。

手順 5）

現用系側で、フェイルオーバ対象のデータベースを部分的包含に設定します。
ここでは、データベース「TESTDB」 に対して設定する例を示します。

```
use master
go
alter database TESTDB set containment = partial
go
```

手順 6）

現用系側でフェイルオーバ対象のデータベース上に包含データベース ユーザを作成します。
ここでは、ユーザ名「TestUser」、パスワード「PassWord」としてデータベース「TESTDB」上にユーザを作成する例を示します。

```
use TESTDB
go
create user TestUser with password = 'PassWord'
go
```

手順 7）

対象の Microsoft SQL Server 用のフェイルオーバグループを待機系側へフェイルオーバします。正常にデータベースがアタッチされ、手順 6）で作成したユーザがログインできることを確認します。

8. 暗号化設定の引き継ぎ

「透過的なデータ暗号化」機能を使用する場合、現用系と待機系で同じサーバ証明書が作成されている必要があります。
待機系側に現用系と同じサーバ証明書が存在していない状態でフェイルオーバが発生すると、エラー 33111 が発生してデータベースのアタッチに失敗します。

現用系と待機系で同じサーバ証明書を作成し、正しくフェイルオーバが行われるようにデータベースを構成するには、以下の手順を実行します。
※「透過的なデータ暗号化」機能を使用しない場合、本設定は不要です。

手順 1）

現用系側でマスターキーを作成します。
ここでは、パスワードを「PassWord!」に設定してマスターキーを作成する例を示します。

```
use master
go
create master key encryption by password = 'PassWord!'
go
```

手順 2）

現用系側でサーバ証明書を作成します。
ここでは、証明書名を「TestCert」、サブジェクトを「Server Certificate Test」としてサーバ証明書を作成する例を示します。

```
create certificate TestCert with subject = 'Server Certificate Test'
go
```

手順 3）

現用系側でサーバ証明書をバックアップします。
ここでは、手順 2）で作成したサーバ証明の秘密キーをパスワード「##pasS」で暗号化して「C:¥temp¥TestCertKey」に保存し、サーバ証明書を「C:¥temp¥TestCert」へバックアップする例を示します。

```
backup certificate TestCert to file = 'C:¥temp¥TestCert'
with private key (file = 'C:¥temp¥TestCertKey', encryption by password =
  '##pa$sS$')
```

手順 4）

対象の Microsoft SQL Server 用フェイルオーバグループを、現用系から待機系へフェイルオーバします。
また、併せて手順 3）でバックアップしたサーバ証明書（バックアップファイル、秘密キーファイル）を待機系側へコピーします。
ここでは、現用系側と同じフォルダ（C:¥temp）配下にサーバ証明書をコピーし、以降の手順を実施するものとします。

手順 5）

待機系側でマスターキーを作成します。

マスターキーの作成時に指定するパスワードは現用系側と同じパスワードとする必要があります。

ここでは、手順 1）で指定したパスワード「PassWord!」を指定してマスターキーを作成する例を示します。

```
use master
go
create master key encryption by password = 'PassWord!'
go
```

手順 6）

待機系側で手順 4）でコピーしたサーバ証明書のリストアを行います。

```
create certificate TestCert from file = 'C:¥temp¥TestCert'
with private key (file = 'C:¥temp¥TestCertKey',
decryption by password = '##pa$sS$')
```

（※）「decryption by password」に指定するパスワードは、手順 3）で指定したパスワードと同じパスワードを指定します。

手順 7）

対象の Microsoft SQL Server 用フェイルオーバグループを、待機系から現用系へフェイルバックします。

手順 8）

フェイルオーバ対象のデータベース上に暗号化キーを作成します。

ここでは、データベース「TESTDB」に、手順 2）で作成したサーバ証明書「TestCert」と、暗号化アルゴリズム「AES_256」を使用して暗号化キーを作成する例を示します。

```
use TESTDB
go
create database encryption key with algorithm = AES_256
encryption by server certificate TestCert
go
```

（※）暗号化アルゴリズムに指定可能な値は、以下の 4 つです。推奨値はないため、環境に応じて選択してください。

・AES_128
・AES_192
・AES_256
・TRIPLE_DES_3KEY

手順 9）

フェイルオーバ対象のデータベースに対して、暗号化設定を有効化します。
ここでは、手順 8）で暗号化キーを作成したデータベース「TESTDB」の暗号化設定を有効化する例を示します。

```
alter database TESTDB set encryption on
go
```

手順 10）

現用系から待機系へフェイルオーバし、待機系側で正しくアタッチが行われることを確認します。

9. Microsoft SQL Server Reporting Services の組み込み

CLUSTERPRO へ Microsoft SQL Server Reporting Services（以下 SSRS）を組み込んでクラスタ化構成を行うには、次の設定を実施します。
また、本項では既に構築手順の「1. フェイルオーバグループの作成」から「5. CLUSTERPRO への Microsoft SQL Server サービスの組み込み」までが完了している環境を前提としています。

手順 1）

現用系と待機系それぞれで SSRS をインストールします。
インストール先フォルダは、必ずローカルディスクを指定するようにしてください。
また、現用系と待機系それぞれで SQL Server Reporting Services の開始モードは全て「手動」に設定（自動起動を行わないように構成）します。

手順 2）

現用系側で Report Server Configuration Manager を起動して次の設定を行います。

a. SQL Server Reporting Services が停止中の場合のみ、左メニューの［＜サーバ名＞¥SSRS］にて「開始」をクリックして、サービスを起動します。
b. 左メニューの［データベース］を選択して、「データベースの変更」をクリックします。
ウィザードにて［新しいレポート サーバデータベースを作成する］を選択して、レポートサーバデータベースを作成します。
なお、以降のレポート サーバデータベース名は既定値の ReportServer 、一時データベースは ReportServerTempDB と記載します。他の名前で作成した場合は、該当環境のデータベース名に読み替えてください。
c. 左メニューの［Web サービス URL］を選択して、「適用」をクリックします。
d. 左メニューの［Web ポータル URL］を選択して、「適用」をクリックします。
e. 左メニューの［暗号化キー］を選択して、「バックアップ」をクリックし、任意のフォルダにバックアップします。
なお、ローカルディスク上にバックアップした場合は、待機系側の同一パスへファイル（*.snk）をコピーします。
f. その他の構成を必要に応じて設定します。

g. 左メニューの［<サーバ名>¥SSRS］にて「停止」をクリックして、サービスを停止します。

手順 3）

現用系側で次のスクリプトを実行して、レポート サーバデータベースをデタッチします。

```
alter database [ReportServer] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db 'ReportServer',TRUE

alter database [ReportServerTempDB] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db 'ReportServerTempDB',TRUE
```

また、現用系側で、デタッチした ReportServer と ReportServerTempDB のデータベースファイルを切替パーティション上の任意のパスに移動後、次のスクリプトを実行して、レポート サーバデータベースをアタッチします。

```
create database [ReportServer] on
(filename='<物理ファイル名>'),
(filename='<物理ファイル名>')
for attach

create database [ReportServerTempDB] on
(filename='<物理ファイル名>'),
(filename='<物理ファイル名>')
 for attach
```

手順 4）

CLUSTERPRO によるフェイルオーバ、およびフェイルバックが行われる際に、ReportServer と ReportServerTempDB をアタッチ・デタッチするスクリプトを設定します。

ユーザデータベースのアタッチを行うスクリプト（ACT.SQL）とデタッチを行うスクリプト（DEACT.SQL）の末尾に、以下のスクリプトを追記します。（注 4）

◆ ACT.SQL

```
create database [ReportServer] on
(filename='<物理ファイル名>'),
(filename='<物理ファイル名>')
for attach

create database [ReportServerTempDB] on
(filename='<物理ファイル名>'),
(filename='<物理ファイル名>')
for attach
```

◆ DEACT.SQL

```
alter database [ReportServer] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db ‘ReportServer',TRUE

alter database [ReportServerTempDB] set offline
with ROLLBACK IMMEDIATE
exec sp_detach_db 'ReportServerTempDB',TRUE
```

（注 4）アタッチを行うスクリプトおよびデタッチを行うスクリプトを新規で作成した場合、スクリプト実行のための Microsoft SQL Server への接続でタイムアウトエラーが発生する可能性があります。このため、必ず既存のスクリプト（ACT.SQL、DEACT.SQL）に追記する必要があります。

手順 5）（Standard Edition を使用している場合のみ実施します。）

フェイルオーバ後にレポート サーバデータベースからフェイルオーバ元のサーバ情報を削除するスクリプトを設定します。

ユーザデータベースのアタッチを行うスクリプト（ACT.SQL）の末尾に、以下のスクリプトを追記します。

◆ ACT.SQL（現用系で記載します。）

```
DELETE FROM ReportServer.dbo.Keys
  WHERE MachineName='<待機系のサーバ名>'
```

◆ ACT.SQL（待機系で記載します。）

```
DELETE FROM ReportServer.dbo.Keys
        WHERE MachineName='<現用系のサーバ名>'
```

手順 6）

CLUSTERPRO の WebUI から、待機系にフェイルオーバします。

手順 7）

新たに現用系となったサーバ側で次のスクリプトを実行して、RSExecRole ロールを作成します。

```
USE master;
GO
IF NOT EXISTS (SELECT 1 FROM sys.database_principals WHERE [type] = 'R' AND
[name] = 'RSExecRole') BEGIN
CREATE ROLE [RSExecRole];
END
GRANT EXECUTE ON dbo.xp_sqlagent_enum_jobs TO [RSExecRole];
GRANT EXECUTE ON dbo.xp_sqlagent_is_starting TO [RSExecRole];
GRANT EXECUTE ON dbo.xp_sqlagent_notify TO [RSExecRole];
GO
USE msdb;
GO
IF NOT EXISTS (SELECT 1 FROM sys.database_principals WHERE [type] = 'R' AND
[name] = 'RSExecRole') BEGIN
CREATE ROLE [RSExecRole];
END
GRANT EXECUTE ON dbo.sp_add_category TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_add_job TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_add_jobschedule TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_add_jobserver TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_add_jobstep TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_delete_job TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_help_category TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_help_job TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_help_jobschedule TO [RSExecRole];
GRANT EXECUTE ON dbo.sp_verify_job_identifiers TO [RSExecRole];
GRANT SELECT ON dbo.syscategories TO [RSExecRole];
GRANT SELECT ON dbo.sysjobs TO [RSExecRole];
GO
```

手順 8）

Report Server Configuration Manager を起動して次の設定を行います。

a. SQL Server Reporting Services が停止中の場合のみ、左メニューの［<サーバ名>¥SSRS］にて「開始」をクリックして、サービスを起動します。
b. 左メニューの［データベース］を選択して、「データベースの変更」をクリックします。ウィザードにて［既存のレポート サーバデータベースを作成する］を選択して、レポートサーバデータベースを設定します。
c. 左メニューの［Web サービス URL］を選択して、「適用」をクリックします。
d. 左メニューの［Web ポータル URL］を選択して、「適用」をクリックします。
e. 左メニューの［暗号化キー］を選択して、「復元」をクリックし、元現用系でバックアップしたファイルを復元します。
f. その他の構成を必要に応じて設定します。

手順 9）

CLUSTERPRO の WebUI から、「5. CLUSTERPRO への Microsoft SQL Server サービスの組み込み」の項目で作成した、開始スクリプト（start.bat）および終了スクリプト（stop.bat）に SSRS の起動/停止を制御するスクリプトを追記します。

SSRS の起動を行うスクリプトは、start.bat 内のアタッチを行うスクリプトの直後に追記します。
SSRS の停止を行うスクリプトは、stop.bat 内のデタッチを行うスクリプトの直前に追記します。

なお、使用している Microsoft SQL Server のエディションによって作成する開始スクリプトの記述が異なるので注意してください。

◆ 開始スクリプト例

◇ Standard Edition の場合（注 5）

```
net start SQLServerReportingServices

:SSRS_KEY_ADD1
timeout /t 1 > nul

"C:¥Program Files¥Microsoft SQL Server Reporting Services¥Shared
Tools¥RSKeyMgmt.exe" -a -f "<暗号化キーのバックアップのパス>" -p <パスワード> -i
SSRS

if %errorlevel% neq 0 GOTO SSRS_KEY_ADD1
```

（注 5）末尾で実行している “RSKeyMgmt.exe” は環境によってフォルダパスが異なる場合がありますので、必要に応じて変更してください。

◇ Enterprise Edition の場合

```
net start SQLServerReportingServices
```

◆ 終了スクリプト例

```
net stop SQLServerReportingServices
```

※ 次の例は、上記の記述を行った状態のスクリプトの一部です。環境に応じて適宜修正してください。

◆ 開始スクリプト例

◇ Standard Edition の場合

```
rem ****************************************
rem  通常起動対応処理
rem ****************************************
:NORMAL

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************

rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
net start SQLServerReportingServices
:SSRS_KEY_ADD1
timeout /t 1 > nul
"C:¥Program Files¥Microsoft SQL Server Reporting Services¥Shared
Tools¥RSKeyMgmt.exe" -a -f "Y:¥sql¥ssrs.snk" -p Passwd -i SSRS
if %errorlevel% neq 0 GOTO SSRS_KEY_ADD1

GOTO EXIT

:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
net start SQLServerReportingServices
:SSRS_KEY_ADD2
timeout /t 1 > nul
"C:¥Program Files¥Microsoft SQL Server Reporting Services¥Shared
Tools¥RSKeyMgmt.exe" -a -f "Y:¥sql¥ssrs.snk" -p Passwd -i SSRS
if %errorlevel% neq 0 GOTO SSRS_KEY_ADD2
GOTO EXIT
```

◇ Enterprise Edition の場合

```
rem ************************************
rem  通常起動対応処理
rem ************************************
:NORMAL


rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************


rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
net start SQLServerReportingServices
:
GOTO EXIT

:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************

net start MSSQLSERVER
sqlcmd -U sa -P passWORD! -i C:¥mssql¥ACT.SQL -o C:¥mssql¥ACT.LOG -S .
net start SQLServerReportingServices

GOTO EXIT
```

◆ 終了スクリプト例

```
rem ***************************************
rem  通常終了対応処理
rem ***************************************
:NORMAL

rem  ディスクチェック
IF "%CLP_DISK%" == "FAILURE" GOTO ERROR_DISK


rem *************
rem  業務通常処理
rem *************


rem  プライオリティ  チェック
IF "%CLP_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *************
rem  最高プライオリティ  での処理
rem *************

net stop SQLServerReportingServices
sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /T 10 > nul

GOTO EXIT


:ON_OTHER1
rem *************
rem  最高プライオリティ  以外での処理
rem *************

net stop SQLServerReportingServices
sqlcmd -U sa -P passWORD! -i C:¥mssql¥DEACT.SQL -o C:¥mssql¥DEACT.LOG -S .
net stop MSSQLSERVER
timeout /t 10 > nul

GOTO EXIT
```

手順 10）

SSRS サブスクリプション機能を使用する場合は、定期実行のために Microsoft SQL Server Agent サービスが必要です。

このため、注意事項「6. Microsoft SQL Server Agent の機能を使用する場合の注意事項について」を参照して、Microsoft SQL Server Agent サービスをフェイルオーバ対象のサービスとして制御するよう構成します。

なお、SSRS サブスクリプション機能で使用される Microsoft SQL Server Agent ジョブは待機系に対して手動作成する必要はありません。フェイルオーバ時に必要に応じて自動的に該当ジョブが作成されます。

補足） Power BI Report Server (PBIRS) について

Microsoft SQL Server 2025 以降、SSRS は Power BI Report Server (PBIRS) に統合されています。

基本的な設定手順は、上述の 「9. Microsoft SQL Server Reporting Services の組み込み」 と同様ですが、サービス名が異なります。

・SSRS
SQLServerReportingServices

・PBIRS
PowerBIReportServer

そのため、PBIRS をご利用される場合は、各スクリプト内のサービス名を PowerBIReportServer に書き換える必要があります。

なお、PBIRS への統合により SSRS の新しいバージョンは、リリースされません。
既存の SSRS については、2033 年 1 月 11 日まで引き続きセキュリティ更新プログラムとサポートを受け取ります。

PBIRS への統合に関しては、次のマイクロソフト社公開情報をご参照ください。

Reporting Services の統合に関する FAQ
https://learn.microsoft.com/ja-jp/sql/reporting-services/reporting-services-consolidation-faq?view=sql-server-ver17

注意事項

1. CLUSTERPRO によるフェイルオーバが利用できない機能について

◆ システムデータベース（master、msdb 等）を使用する機能は、フェイルオーバすることはできません。フェイルオーバが利用できない主な機能は以下の通りです。

- **Microsoft SQL Server 2025** Standard / Enterprise
  Microsoft SQL Server 2022 Standard / Enterprise に記載の各機能、Tempdb 領域リソースのガバナンス 等
- **Microsoft SQL Server 2022** Standard / Enterprise
  Microsoft SQL Server 2019 Standard / Enterprise に記載の各機能、Azure Synapse Link for SQL 等
- **Microsoft SQL Server 2019** Standard / Enterprise
  Microsoft SQL Server 2017 Standard / Enterprise に記載の各機能、ビッグ データ クラスター等
- **Microsoft SQL Server 2017** Standard / Enterprise
  Microsoft SQL Server 2016 Standard / Enterprise に記載の各機能、自動チューニング、Machine Learning Services 等
- **Microsoft SQL Server 2016** Standard / Enterprise
  Microsoft SQL Server 2014 Standard / Enterprise / Business Intelligence に記載の各機能、ライブクエリ統計、クエリストア、R Services 等
- **Microsoft SQL Server 2014** Standard / Enterprise / Business Intelligence
  Microsoft SQL Server Analysis Services、データベース スナップショット、AlwaysOn 可用性グループ、データベース ミラーリング、ログ配布、レプリケーション、マルチサーバ管理、データ層アプリケーション、マスター データ サービス、データコレクション、パフォーマンス データコレクション、リソースガバナ、Microsoft SQL Server Audit 監査機能、データ プロファイル タスク、インメモリ OLTP、バッファプール拡張 等

  これらの機能はフェイルオーバ後に情報が引き継がれないため、CLUSTERPRO 環境上で正常な動作を保証することができません。このため、NEC SQL Server Response Center では、CLUSTERPRO 環境における上記の機能はサポートしていません。

2. Microsoft SQL Server のクラスタ構成の注意事項について

- 現用系、待機系のMicrosoft SQL Server関連サービスのスタートアップの種類は全て「手動」に設定してください。(サーバ起動時に自動起動しないよう構成する必要があります。)

- Microsoft SQL Serverのフェイルオーバグループに登録する切替パーティション上には、フェイルオーバ対象となるデータベースのデータファイル（*.mdf、*.ndf）とトランザクションログファイル（*.ldf）のみを格納してください。これら以外のファイルを切替パーティション上に格納し、現用系と待機系で同じファイルを使用する構成はサポートされません。

- CLUSTERPRO終了スクリプトファイル（stop.bat）内に記述するtimeoutコマンドのパラメータ（スリープ時間）は、システムの状態やMicrosoft SQL Serverの状態により異なるため、実機での評価後、調整する必要があります。

- 「4. Microsoft SQL Serverのスクリプト作成」に記載しているクエリで使用するフェイルオーバデータベース名は、Microsoft SQL Server上で認識されているデータベース名と大文字／小文字を一致させて記述してください。
  Microsoft SQL Serverインスタンスレベルの照合順序の設定によっては、大文字／小文字が区別され、クエリの実行に失敗する可能性があります。
  Microsoft SQL Server上で認識されているデータベース名は、以下のクエリを実行することで確認できます。以下のクエリの実行結果から、[name] 列の値を確認します。

```
exec sp_helpdb
go
```

- フェイルオーバデータベースに 3つ以上のファイルを使用する場合は、
  「4. Microsoft SQL Server のスクリプト作成」に記載している ACT.SQLを以下のように修正してください。

```
create database '<現用系サーバ上フェイルオーバデータベース名>' on
  (filename='<物理ファイル名>'),
  (filename='<物理ファイル名>'),
  (filename='<物理ファイル名>'),
    ・・・
  (filename='<物理ファイル名>')
for attach
```

例）TESTDB データベースに、TESTDB.mdf、TESTDB_1.ndf、
TESTDB_2.ndf、TESTDB_log.ldf の 4 つのファイルが存在する場合

```
create database TESTDB on
  (filename = 'Y:¥sql¥data¥TESTDB.mdf'),
  (filename = 'Y:¥sql¥data¥TESTDB_1.ndf'),
  (filename = 'Y:¥sql¥data¥TESTDB_2.ndf'),
  (filename = 'Y:¥sql¥data¥TESTDB_log.ldf')
for attach
```

- フェイルオーバデータベースの物理ファイルの構成変更（*.ndf や *.ldf の追加や削除）を行う場合は、もう一方のサーバに当該データベースのエントリが残存していない状態で実施してください。たとえば、現用系側にて OS ダウンなどの障害発生により、データベースのデタッチが行われなかった場合、当該データベースのエントリが残存します。この状態で、待機系側で物理ファイルの構成変更を行い、フェイルバックを行うと、現用系側で当該データベースのオープン処理が失敗します。

- 名前付きインスタンスを使用する場合、現用系と待機系でインスタンス名に同じ名前を設定してください。

3. 片方向スタンバイ構成における注意事項について

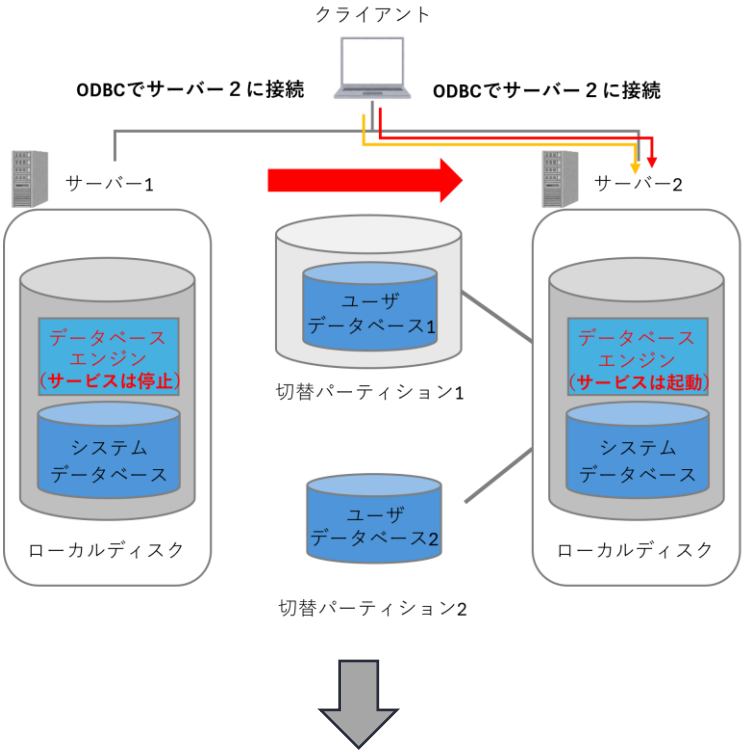
- フェイルオーバ対象のデータベースが複数存在する場合、現用系と待機系で同一のdbidで登録するために、現用系のdbid順に待機系にcreate database for attachを実行してください。データベースのdbidは、以下のクエリを実行することで確認できます。以下のクエリの実行結果から、対象データベースの［dbid］列の値を確認します。

```
exec sp_helpdb
go
```

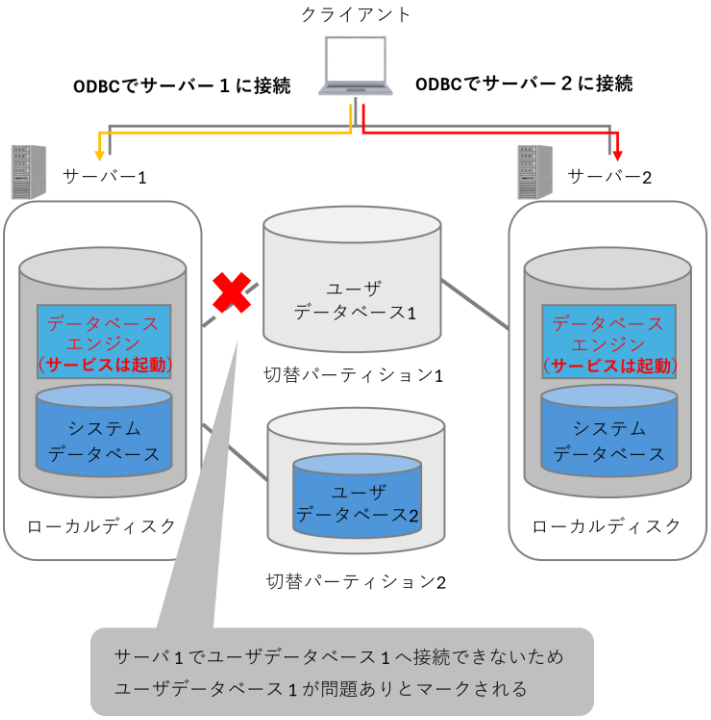
4. 双方向スタンバイ構成における注意事項について

- 双方向スタンバイ型構成において 1つのサーバでフェイルオーバが発生した場合、まずフェイルオーバされたデータベースをフェイルバックしてください。フェイルバックを実行する前にもう 1つのデータベースをフェイルオーバすることはできません（以下の操作を行ってはいけません）。

① ユーザデータベース 1 をサーバ 2 へフェイルオーバ



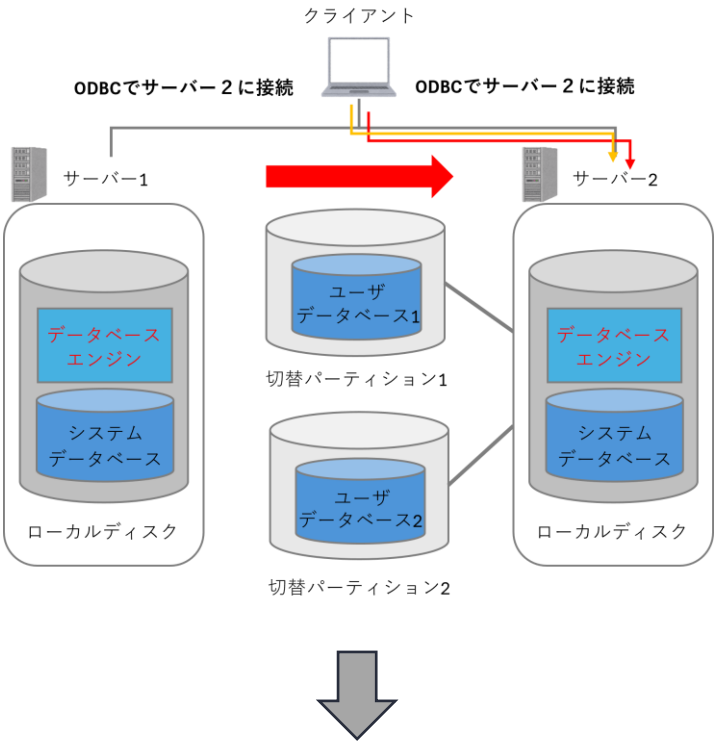
② ユーザデータベース 1 をサーバ 1 へフェイルバックを実行する前に、
ユーザデータベース 2 をサーバ 1 にフェイルオーバ



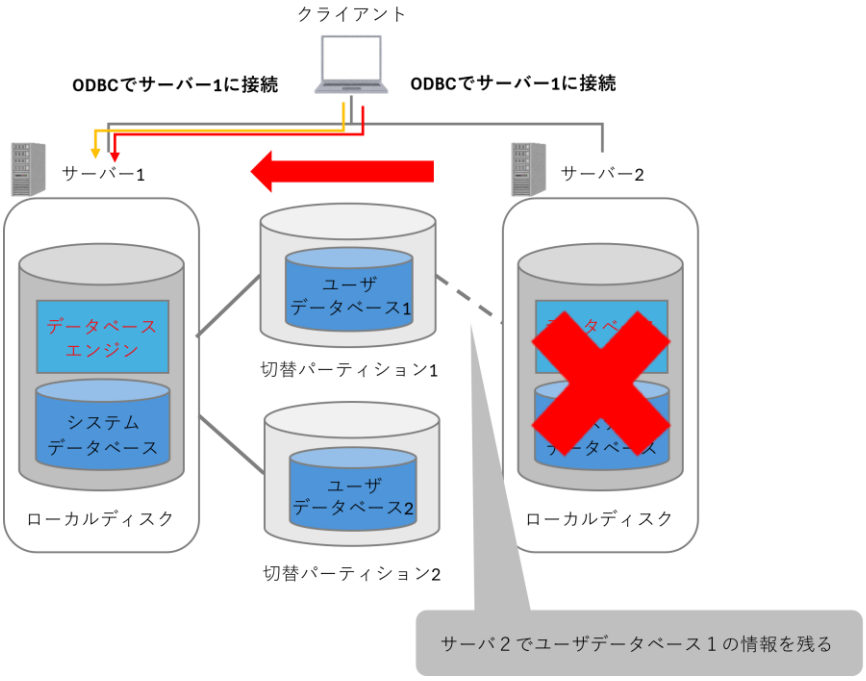
サーバ 1 でのユーザデータベース 1、サーバ 2 でのユーザデータベース 2 は、それぞれのサーバ上の Microsoft SQL Server で定義されたデータベースです。
したがって、Microsoft SQL Server のサービスが起動中にアクセスできない状態では問題あり（「未確認」ステータス）となり、以降アクセスできない状態となります。このような操作は運用上サポートされません。

◆ 双方向スタンバイ型で運用中、サーバ1で障害が発生し、サーバ2上で運用している（サーバ 1 は復旧されているが、フェイルバックは実行されていない）状態で、サーバ2に障害が発生した場合、サーバ2を起動、復旧し、Microsoft SQL Serverが起動される（クラスタに復帰）と、サーバ1からフェイルオーバで引き継いだデータベースが問題ありとマークされます。

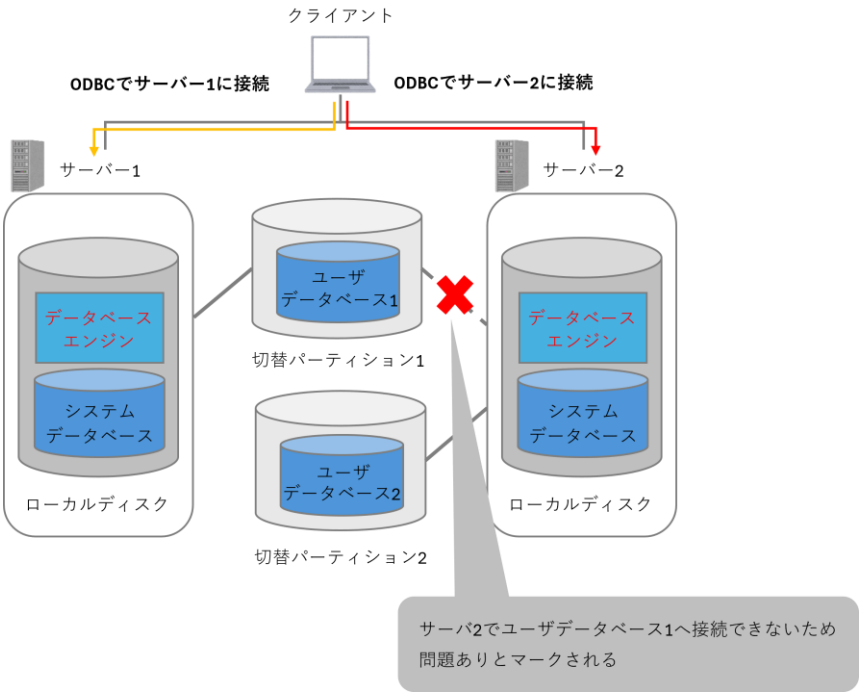
① ユーザデータベース 1 をサーバ 2 へフェイルオーバ



② サーバ 1 へフェイルバックを実行する前にサーバ 2 に障害発生



③ サーバ 2 を復旧し、クラスタへ復旧後、切替パーティション 2 のデータベースを
サーバ 2 へフェイルバック



上記図のような状況となった場合には、サーバ 2 上で以下のクエリを実行し、サーバ 2 上からユーザデータベース 1 の情報を削除してください。

```
use master
go
alter database <ユーザデータベース 1 のデータベース名> set offline
go
exec sp_detach_db '<ユーザデータベース 1 のデータベース名>', true
go
```

◆ サーバ1のフェイルオーバデータベースのdbidが7の場合、サーバ2にて一度ダミーデータベースをdbid 7で登録し、その状態でサーバ2のフェイルオーバデータベースをdbid 8で登録してください。dbid 8で登録後、ダミーデータベースを削除し、dbid 7がサーバ2上に存在しない状態とします。
また、サーバ1上にはdbid 8が登録されていない状態としてください。
データベースのdbidは、以下のクエリを実行することで確認できます。以下のクエリの実行結果から、対象データベースの［dbid］列の値を確認します。

```
exec sp_helpdb
go
```

5. データファイル格納ディスク破損時のログ末尾のバックアップに関する注意事項について

- データベースを構成する物理ファイルのうち、データファイル（*.mdf、*ndf）を格納しているディスクが破損してデータファイルへアクセス不可となった場合、その時点でトランザクションログ末尾のバックアップを取得できれば、障害発生直前の状態まで復旧することが可能です。

- 障害発生によりフェイルオーバが発生すると、フェイルオーバ処理によって障害が発生したサーバからデータベースがデタッチされます。デタッチされた後にトランザクションログ末尾のバックアップを取得することはできませんので、フェイルオーバ発生前にトランザクションログ末尾のバックアップを取得するよう構成する必要があります。

  以下のいずれかの方法でフェイルオーバ発生前にトランザクションログ末尾のバックアップ取得を行うよう構成することが可能です。（どちらの方法でもトランザクションログバックアップを行う操作であることは同じとなりますので環境に応じてご選択ください。）

  - デタッチを行うスクリプト（DEACT.SQL）の 1 行目（alter database ステートメント）の前に、backup log ステートメント（with CONTINUE_AFTER_ERROR オプション付き）を記述する。
  - データファイル格納ディスクが破損したことを検知した際に実行可能な CLUSTERPRO スクリプトに backup log ステートメント（with CONTINUE_AFTER_ERROR オプション付き）を実行する sqlcmd コマンドを記述する。

- backup logステートメントにCONTINUE_AFTER_ERRORオプションを指定するのは、破損により万一バックアップ時にエラーが検出された場合でも、バックアップを継続して実行することを想定しているためです。エラーが発生しない場合は、当該オプションを指定しない場合と同等の動きとなります。

- CONTINUE_AFTER_ERRORオプションを指定すれば、必ずログ末尾のバックアップが取得できるということを保証するものではありません。当該オプションを指定した状態においてもバックアップが行えない可能性もあります。
  当該オプションを指定してもバックアップが取得できない場合には、ログ末尾のバックアップを行う方法はありません。

6. Microsoft SQL Server Agent の機能を使用する場合の注意事項について

- Microsoft SQL Server Agentジョブ、警告等は、現用系のみの設定ではフェイルオーバ後、継続して利用することができません。
  フェイルオーバ後も待機系でMicrosoft SQL Server Agentジョブ、警告等を使用する場合は、現用系と待機系でそれぞれ同じMicrosoft SQL Server Agentジョブ、警告等を作成する必要があります。

- Microsoft SQL Server Agentの機能をフェイルオーバ後も利用する場合、
  Microsoft SQL Server Agentサービスをフェイルオーバ対象のサービスとして制御する必要がありますので、環境構成によって以下のいずれかの対応を実施する必要があります。
  - スクリプトリソースに Microsoft SQL Server Agent サービスに対する net start／net stop コマンドを記述します。
  - サービスリソースに「SQLSERVERAGENT」を登録します。(名前付きインスタンスの場合は「SQLAGENT$<対象のインスタンス名>」を登録します。)

- 上記いずれかの方法によりCLUSTERPRO側でサービス制御を行う場合、
  Microsoft SQL Serverサービスを停止する処理の前にMicrosoft SQL Server Agentサービスが停止するよう構成してください。
  Microsoft SQL Server Agentサービスを停止させずにフェイルオーバ／フェイルバックを実行すると、Microsoft SQL Serverサービスは、Microsoft SQL Server Agentサービスの停止待ち状態となり、フェイルオーバ／フェイルバックの実行中にハングしたような状態となります。

  また、サービス起動時にはMicrosoft SQL Serverサービス起動後にMicrosoft SQL Server Agentサービスを起動するよう構成します。

  サービスリソースを使用している場合は、Microsoft SQL Serverサービスの後に
  Microsoft SQL Server Agentサービスが起動するようサービスリソースの依存関係を設定する必要があります。この依存関係を設定しておけば終了時は逆の順番でサービスが停止されますので、停止時の順序を設定する必要はありません。
  なお、必要に応じてサービス監視リソースを設定してください。

7. ポリシーベースの管理機能を使用する場合の注意事項について

- 「ポリシーベースの管理」機能をCLUSTERPRO環境で使用する場合、現用系で作成したポリシーを待機系へ移行することで、待機系でも問題なくポリシーを適用することが可能となります。
  また、現用系でポリシーの評価を実行するジョブを作成している場合においても、ポリシーの移行を行うことで併せて作成されます。(ジョブを個別に移行する必要はありません。)
  ポリシーの移行方法は以下の通りです。

(1) 現用系で SSMS を開きます。
(2) SSMS 上で、作成したポリシーを右クリックして、[ポリシーのエクスポート] を選択し、表示されたダイアログで任意の名前と保存場所を指定して保存します。
(3) (2)で保存したファイルを待機系へコピーします。
(4) 待機系で SSMS を開きます。
(5) SSMS 上で、[管理] – [ポリシー管理] – [ポリシー] を右クリックして、[ポリシーのインポート] を選択し、(3) でコピーしたファイルを指定します。
(6) [OK] をクリックし、正常にポリシーが作成されたことを確認します。

8. FILESTREAM 機能を使用する場合の注意事項について

- 「FILESTREAM」機能を CLUSTERPRO 環境で使用する場合、FILESTREAMファイルグループ（データベース作成時に指定するFILESTREAM格納先）を切替パーティション上に指定することで、待機系へフェイルオーバ後も継続して使用することが可能となります。

9. 変更データキャプチャ（CDC）機能を使用する場合の注意事項について

- 「変更データキャプチャ（CDC)」機能をCLUSTERPRO環境で使用する場合、切替パーティション上のユーザデータベースに対して設定することで、待機系へフェイルオーバ後も継続して使用することが可能となります。
- 本機能を使用する場合、Microsoft SQL Server Agentサービスが起動されている必要がありますので、Microsoft SQL Server AgentサービスをCLUSTERPROによるサービス制御の対象としてください。
  併せて「Microsoft SQL Server Agent機能を使用する場合の注意事項について」の項目もご参照ください。

10. その他の機能を使用する場合の注意事項について

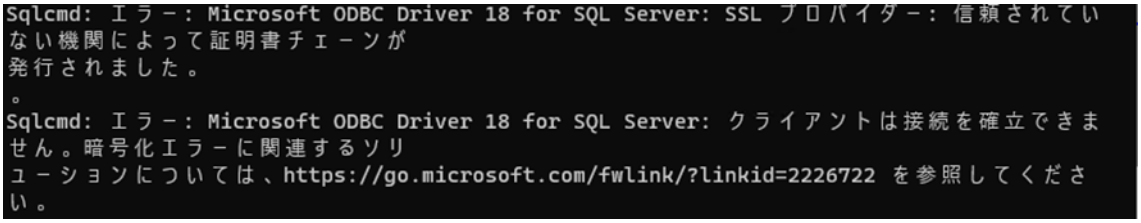
- Microsoft SQL Server 2012 Standard / Enterprise / Business Intelligenceから
  Microsoft SQL Server 2025 Standard / Enterpriseまでで追加された新機能における
  CLUSTERPRO上での利用可否については、利用可否が判明次第、情報を公開いたします。

11. Microsoft ODBC Driver 18 for SQL Server 以降をご利用される場合

◆ 以前のバージョン（ODBC Driver 17 まで）では、ユーザが明示的に指定しない限り、接続の暗号化はオプション扱いでしたが、ODBC Driver 18 からセキュリティ強化のため、以下のデフォルト値が変更されました。

Encrypt: No → Yes (Mandatory)
TrustServerCertificate: No (デフォルト)

この変更により、sqlcmd などによる Microsoft SQL Server への接続時に何も指定せずに接続しようとすると、「信頼された証明書による暗号化接続」 が行われ、次のエラーが発生します。



上記のエラーを解決するには、sqlcmd の接続コマンドに “-C” オプションを付けるか、公的または社内 CA の証明書をクライアントにインポートする必要があります。

-C は、Microsoft SQL Server への接続時に 「サーバ側の証明書が本物かどうかを確認するプロセスをスキップする」 というフラグです。(TrustServerCertificate : Yes と同様)

-C オプションを使用する場合、サーバ側の証明書検証をスキップすることになります。これには、以下のようなセキュリティ上の重大な懸念とリスクが伴います。

12. SSRS および PBIRS の Web ポータル 表示について

◆ Microsoft Edge のバージョンが古い場合、最新の SSRS および PBIRS の Web ポータル画面が表示されない(真っ白のままになる)ケースが報告されています。

上記の問題が発生した場合は、Microsoft Edge を最新バージョンに更新したうえ、Web ポータル画面が表示されるかご確認ください。

Microsoft Edge のバージョンは、上部の［・・・］–［ヘルプとフィードバック］–［Microsoft Edge について］から確認可能です。

その他

- Microsoft SQL Serverの各スクリプトの詳細については、Microsoft SQL Server Books Online に記載されていますのでご参照願います。
- 以下のURLにマイクロソフトサポート技術情報が公開されていますので、併せてご参照ください。

  マイクロソフトサポートオンライン
  http://support.microsoft.com/default.aspx?LN=JA