



高可用性ガイド

Version 2024.1
2024-06-03

高可用性ガイド

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

1 高可用性を実現するためのフェイルオーバー方法	1
1.1 フェイルオーバー方法なし	1
1.2 フェイルオーバー・クラスタ	2
1.3 仮想化 HA	3
1.4 InterSystems IRIS ミラーリング	3
1.5 フェイルオーバー方策での分散キャッシュの使用法	4
2 ミラーリングの概要	7
3 ミラーリングのアーキテクチャおよび計画	9
3.1 ミラー・コンポーネント	9
3.1.1 フェイルオーバー・ミラー・メンバ	9
3.1.2 非同期ミラー・メンバ	10
3.1.3 ISCAgent	13
3.1.4 アービター	14
3.2 ミラー同期	14
3.3 自動フェイルオーバーのメカニズム	15
3.3.1 安全な自動フェイルオーバーの要件	15
3.3.2 自動フェイルオーバーのルール	16
3.3.3 さまざまな停止シナリオに対するミラーの対応	17
3.3.4 ミラーの可用性を最適化するためのアービターの配置	19
3.3.5 自動フェイルオーバーのメカニズムの詳細	20
3.4 自動フェイルオーバーの阻止	25
3.5 ミラーリング通信	25
3.5.1 ネットワーク構成に関する考慮事項	26
3.5.2 ネットワーク遅延に関する考慮事項	26
3.5.3 ミラー・トラフィックの圧縮	27
3.5.4 ミラー・メンバのネットワーク・アドレス	28
3.6 ミラーリング・アーキテクチャとネットワーク構成のサンプル	29
3.6.1 単一のデータ・センタ、コンピュータ・ルーム、またはキャンパス内のミラーリング構成	29
3.6.2 2つのデータ・センタと地理的に分離された災害復旧に対するミラーリング構成	35
3.7 フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト	39
3.7.1 組み込みのメカニズム	39
3.7.2 外部テクノロジー	40
3.7.3 ミラー仮想 IP (VIP) の計画	40
3.8 仮想環境でのミラーリング	41
3.9 クラウド環境でのミラーリング	42
3.10 バックアップ・フェイルオーバー・メンバへのアクセスの制限	42
3.11 単一ホストへの複数のミラー・メンバのインストール	43
4 ミラーリングの構成	45
4.1 ミラーの自動導入方法	45
4.1.1 InterSystems Cloud Manager (ICM) を使用したミラーの導入	45
4.1.2 InterSystems Kubernetes Operator (IKO) を使用したミラーの導入	45
4.1.3 構成マージを使用したミラーの導入	46
4.2 ミラー構成のガイドライン	46
4.3 アービターのインストール	47
4.4 ISCAgent の起動	49

4.5 TLS セキュリティを使用したミラー通信の保護	49
4.6 ^MIRROR ルーチンの使用法	49
4.7 ミラーの作成	49
4.7.1 ミラーを作成し、第 1 のフェイルオーバー・メンバを構成する	50
4.7.2 第 2 のフェイルオーバー・メンバを構成する	52
4.7.3 第 2 のフェイルオーバー・メンバまたは非同期メンバを承認する (TLS ミラーのみ)	53
4.7.4 フェイルオーバー・メンバのステータスをミラー・モニタで確認する	54
4.7.5 非同期ミラー・メンバを構成する	54
4.8 ミラーへのデータベースの追加	56
4.8.1 ミラーリングされたデータベースの考慮事項	57
4.8.2 ミラーリングされるデータベースを作成する	58
4.8.3 ミラーへ既存データベースを追加する	59
4.8.4 ミラーリングされるデータベースの有効化とキャッチアップ	61
4.9 ミラーの削除	62
4.10 ミラー・メンバの編集または削除	63
4.10.1 レポート非同期ミラー・メンバの FailoverDB フラグのクリア	63
4.10.2 ミラー・メンバ削除時のミラーされるデータベースの属性の削除	64
4.10.3 非同期メンバの編集または削除	64
4.10.4 フェイルオーバー・メンバの編集または削除	67
4.10.5 ミラーリングされるデータベースをミラーから削除する	69
4.11 ミラーでのマネージド・キー暗号化の使用法	70
4.11.1 ミラーリングされたデータベースの暗号化	70
4.11.2 ミラー内ジャーナル暗号の有効化	71
4.12 ミラーへのアプリケーション・サーバ接続の構成	72
4.13 ミラー仮想 IP (VIP) の構成	74
4.13.1 ミラー VIP の InterSystems IRIS 構成	74
4.13.2 ミラー VIP の構成	75
4.14 ISCAgent の構成	75
4.14.1 ISCAgent の開始および停止	76
4.14.2 ISCAgent のカスタマイズ	79
4.15 サービス品質 (QoS) タイムアウトの構成	80
4.16 並列デジャーナリングの構成	80
4.17 ^ZMIRROR ルーチンの使用法	81
4.18 医療製品向けのミラーリングの構成	82
5 ミラーリングの管理	85
5.1 ミラーの監視	85
5.1.1 ミラー・モニタの使用法	85
5.1.2 ^MIRROR ステータス・モニタの使用法	91
5.1.3 ミラーリング通信プロセスの監視	92
5.2 ミラー・メンバのネットワーク・アドレスの更新	94
5.3 ネットワーク・アドレス検証エラーの解決	95
5.4 X.509 DN 更新の承認 (TLS のみ)	95
5.5 DR 非同期メンバのフェイルオーバー・メンバへの昇格	96
5.6 DR 非同期へのバックアップの降格	99
5.7 ミラー・メンバの再構築	100
5.8 バックアップおよび非同期メンバのミラーリングの停止	101
5.9 データベース・デジャーナリングの管理	101
5.9.1 バックアップまたは DR 非同期でのデジャーナリング管理	102
5.9.2 レポート非同期でのデジャーナリング管理	102
5.9.3 レポート非同期に対するデジャーナル・フィルタの使用	103

5.10 一般的なミラーリングに関する考慮事項	104
5.10.1 ミラーの API	104
5.10.2 プライマリ・フェイルオーバー・メンバの外部バックアップ	105
5.10.3 ミラー・メンバでの InterSystems IRIS のアップグレード	105
5.11 ミラーリングにおけるデータベースの考慮事項	105
5.11.1 InterSystems IRIS インスタンスの互換性	105
5.11.2 メンバのエンディアンに関する考慮事項	106
5.11.3 ^DATABASE ルーチンを使用したミラーリングされるデータベースの作成	106
5.11.4 ^DATABASE ルーチンを使用した既存のミラーリングされるデータベースの再作成	107
5.11.5 ミラーリングされるデータベースのマウント/ディスマウント	107
5.11.6 ミラーリングされないシステムへのミラーリングされるデータベースのコピー	107
5.12 ミラーリングにおけるプロダクションの考慮事項	108
5.12.1 InterSystems IRIS によるミラーリングされるデータを使用した相互運用対応ネームスペースの処理方法	108
5.12.2 InterSystems IRIS プロダクションの推奨ミラーリング構成	108
5.12.3 ミラーリング環境におけるプロダクションの自動開始の機能	108
5.13 インターシステムズの医療製品のミラーリングに関する考慮事項	109
6 ミラー停止の手順	111
6.1 計画的停止の手順	112
6.1.1 バックアップ・フェイルオーバー・メンバのメンテナンス	112
6.1.2 プライマリ・フェイルオーバー・メンバのメンテナンス	113
6.1.3 フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避	113
6.1.4 ミラー内 InterSystems IRIS インスタンスのアップグレード	113
6.2 計画外停止の手順	114
6.2.1 バックアップ・フェイルオーバー・メンバの計画外停止	114
6.2.2 自動フェイルオーバーによるプライマリ・フェイルオーバー・メンバの計画外停止	114
6.2.3 自動フェイルオーバーが発生しない場合のプライマリ・フェイルオーバー・メンバの計画外停止	115
6.2.4 プライマリ・フェイルオーバー・メンバの計画外分離	117
6.2.5 両方のフェイルオーバー・メンバの計画外停止	118
6.3 災害復旧の手順	119
6.3.1 災害復旧時の昇格した DR 非同期への手動フェイルオーバー	119
6.3.2 昇格した DR 非同期への計画的フェイルオーバー	123
6.3.3 昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え	124

図一覧

図 1-1: フェイルオーバー・クラスタ構成	2
図 1-2: 仮想環境でのフェイルオーバー	3
図 1-3: InterSystems IRIS ミラー	4
図 2-1: InterSystems IRIS ミラー	7
図 3-1: ミラーのフェイルオーバー・メンバ	10
図 3-2: 単一のミラーに接続された複数の DR 非同期メンバ	11
図 3-3: 複数のミラーに接続された単一レポート非同期メンバ	12
図 3-4: 単一のフェイルオーバー・メンバと複数の非同期メンバ	13
図 3-5: ミラーのフェイルオーバー・メンバとアービター	14

テーブル一覧

テーブル 3-1: アービター・モードで失われた接続に対するミラーの対応	23
テーブル 5-1: プライマリ・フェイルオーバー・メンバ上でのミラーリング・プロセス	93
テーブル 5-2: バックアップ・フェイルオーバー・メンバおよび非同期メンバ上でのミラーリング・プロセス	93

1

高可用性を実現するためのフェイルオーバー方法

団体や組織は以前にも増してネットワークベースのアプリケーションに依存しているため、データベースの可用性および信頼性をできる限り高めることは重要です。このドキュメントでは、InterSystems IRIS® データ・プラットフォームで可用性と信頼性の高いデータ・ストレージを提供する方法について、およびデータの整合性を維持しつつ、停止や障害から迅速に回復するための方法について説明します。

InterSystems IRIS® Data Platform には、高可用性 (HA) ソリューションがいくつも用意されており、オペレーティング・システム・プロバイダが提供して広く使用されているすべての HA 構成と容易に統合されます。

システムの高い可用性を維持するための主要機能は、フェイルオーバーと呼ばれています。この手法では、障害が発生したプライマリ・システムをバックアップ・システムと入れ替えることで、処理をバックアップ・システムへ障害回避 (フェイルオーバー) します。多くの HA 構成は災害復旧の機能も備えています。これは、フェイルオーバー機能でシステムを利用可能な状態に維持できなくなったときに、システムの可用性を復元する機能です。

InterSystems IRIS のインスタンスで HA を確保するためのフェイルオーバーには、5 種類の一般的な手法があります (その 1 つは、HA 対策を導入しないというものです)。ここでは、これらの手法の概要を説明し、このドキュメントの残りの部分では、その実装手順を説明します。

ミラーリングを除き、これらのすべての手法では、1 台のストレージに障害が発生しただけで深刻な状況になる可能性があることを銘記しておく必要があります。このことから、ディスクの冗長化、データベースのジャーナリング ([“ジャーナリング”](#)を参照してください)、および優れたバックアップ手順 ([“バックアップとリストア”](#)を参照してください) を、これらの手法と必ず組み合わせます。このような対応が、ディスクの障害による影響を軽減するうえで重要であるからです。

ご使用の環境に適切なフェイルオーバーや災害復旧方法の構築や現在の方法の見直しのための詳細情報が必要な場合は、[インターシステムズのサポート窓口](#)までお問い合わせください。

1.1 フェイルオーバー方法なし

InterSystems IRIS データベースの整合性は、“データ整合性ガイド”の説明にある機能により、プロダクション・システムの障害から常に保護されています。構造的なデータベースの整合性は、インターシステムズのライト・イメージ・ジャーナル (WIJ) テクノロジーによって維持され、論理的な整合性はジャーナリングとトランザクション処理によって維持されています。自動的な WIJ およびジャーナルのリカバリは、InterSystems のエラープルーフ・データベース・アーキテクチャの基本的な構成要素です。

一方、フェイルオーバー対策を講じていないと、システムの障害の原因およびそれを見極めて解決する能力によっては、その障害が重大なダウンタイムを引き起こすことがあります。業務上それほど重要ではない多くのアプリケーションでは、このリスクは許容できるかもしれません。

この方法を採用するには、システムのすべての利用者が以下の特質を備えている必要があります。

- ・ ジャーナリング、バックアップとリストアなどの明確で詳細な運用面のリカバリ手順
- ・ ディスクの冗長性 (RAID、もしくはディスク・ミラーリング)
- ・ ハードウェアをすばやく入れ替えることができる能力
- ・ すべてのベンダとの年中無休のメンテナンス契約
- ・ 障害に起因して発生するある程度のダウンタイムに対する管理上の受容とアプリケーション・ユーザの許容

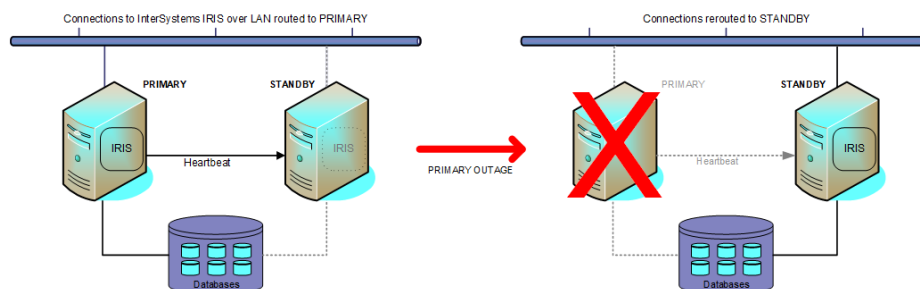
1.2 フェイルオーバー・クラスタ

HAを実現する手法として広く使用されているものにフェイルオーバー・クラスタがあります。この手法では、共有ストレージおよびアクティブなメンバをフォローするクラスタ IP アドレスを使用し、プライマリ・プロダクション・システムを（一般的に同じ構成の）スタンバイ・システムで補完します。プロダクション・システムに障害が発生すると、スタンバイ・システムがプロダクションの作業負荷を引き受けて、障害が発生したプライマリ・システムでそれまで実行されていた InterSystems IRIS などのプログラムとサービスを引き継ぎます。

InterSystems IRIS は、オペレーティング・システム・レベルで提供されているフェイルオーバー・ソリューションと容易に統合できるように設計されています。このようなオペレーティング・システムとして、具体的には IBM PowerHA SystemMirror や Red Hat Enterprise Linux HA などがあります。共有ストレージ・デバイスに InterSystems IRIS のインスタンスを 1 つだけインストールして、両方のクラスタ・メンバでそのインスタンスを認識するようにします。続いて、このインスタンスをフェイルオーバー・クラスタ構成に追加して、フェイルオーバーの構成要素として自動的に起動するようにします。アクティブ・ノードを使用できない期間が指定時間に達すると、クラスタ IP アドレスと共有ストレージのコントロールがフェイルオーバー・テクノロジーによってスタンバイ・システムに移管され、新しいプライマリ・システムで InterSystems IRIS が再起動します。ここで再起動するシステムでは、通常の起動リカバリが自動的に実行され、WIJ、ジャーナリング、およびトランザクション処理によって、障害が発生したシステム上で InterSystems IRIS が再起動した場合とまったく同様に構造上の整合性とデータの整合性が維持されます。

スタンバイ・サーバには、障害が発生したプライマリ・システムがリストアされるまでの間、通常のプロダクション作業負荷を処理できる能力が必要です。スタンバイ・システムがプライマリ・システムとなり、障害からリストアされたプライマリ・システムがスタンバイ・システムとなる構成も可能です。

図 1-1: フェイルオーバー・クラスタ構成

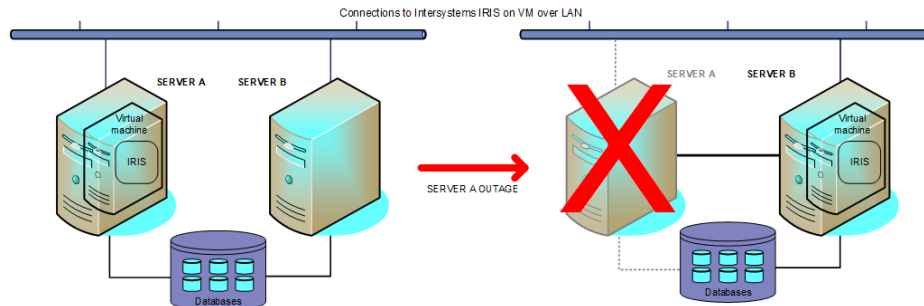


この手法では、共有ストレージ・デバイスに障害が発生すると深刻な状況となります。このことから、十分なりカバリ機能を実現するうえで、ディスクの冗長化、ジャーナリング、およびバックアップとリストアの良好な手順がきわめて重要です。

1.3 仮想化 HA

仮想化プラットフォームにはHA機能が用意されていることが一般的で、多くの場合、ゲスト・オペレーティング・システムとそこで稼働しているハードウェアの両方のステータスをHA機能で監視します。いずれかで障害が発生すると、障害が発生した仮想マシンが仮想化プラットフォームによって自動的に再起動します。必要に応じて、代替のハードウェア上で再起動することもあります。InterSystems IRIS インスタンスが再起動するときは、通常の起動リカバリが自動的に実行され、WII、グローバルなジャーナリング、およびトランザクション処理によって、物理サーバ上でInterSystems IRISが再起動した場合とまったく同様に構造上の整合性とデータの整合性が維持されます。

図 1-2: 仮想環境でのフェイルオーバー



仮想化プラットフォームでは、保守上の目的で仮想マシンを代替のハードウェアに再配置することもできます。これによって、例えばダウン・タイムを発生させずに物理サーバをアップグレードできます。一方、仮想化HAはフェイルオーバー・クラスタと並行クラスタの主な欠点も併せ持っているので、共有ストレージに障害が発生すると深刻な状況になります。

1.4 InterSystems IRIS ミラーリング

自動的なフェイルオーバーを組み合わせた InterSystems IRIS データベースのミラーリングでは、計画上の中断と計画外の中断に対応できる効果的で低コストの高可用性ソリューションを実現できます。ミラーリングはデータのレプリケーションに基づいており、共有ストレージに依存していないので、ストレージの障害に起因する深刻なサービス中断を回避できます。

InterSystems IRIS ミラーは、フェイルオーバー・メンバと呼ばれる物理的に独立した2つの InterSystems IRIS システムで構成されます。各フェイルオーバー・メンバは、各ミラーされたデータベースのコピーをミラーに維持します。アプリケーションの更新はプライマリ・フェイルオーバー・メンバ上で実行され、バックアップ・フェイルオーバー・メンバのデータベースは、プライマリのジャーナル・ファイルを適用することでプライマリとの同期状態を維持します（ジャーナリングの詳細は、「[ジャーナリングの概要](#)」を参照してください）。

ミラーでは、2つのフェイルオーバー・メンバの一方にプライマリのロールが自動的に割り当てられます。これにより、もう一方のフェイルオーバー・メンバは自動的にバックアップ・システムになります。プライマリ InterSystems IRIS インスタンスに障害が発生するか、使用不可になると、バックアップ・インスタンスが短時間で自動的に引き継いでプライマリ・インスタンスとなります。

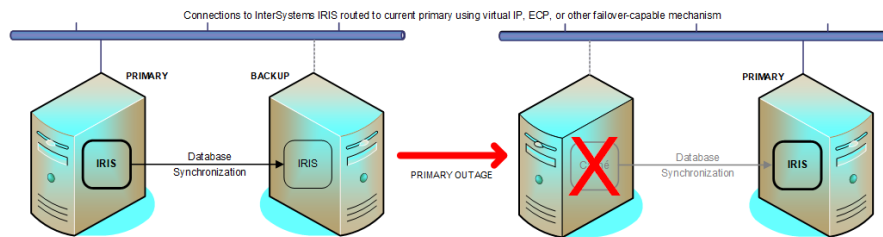
アービターと呼ばれる第3のシステムは、フェイルオーバー・メンバとの継続的な通信を維持し、メンバ間で直接通信できない場合にフェイルオーバーの決定を安全に下すために必要なコンテキストを提供します。各フェイルオーバー・システムのホストで動作しているエージェント・プロセス (ISCAgents) も自動フェイルオーバー・ロジックを支援します。プライマリが実際に停止しているか、使用不可になっていることが確認できない限り、バックアップは引き継ぐことができず、プライマリとして動作しません。アービターと ISCAgents 間では、ほぼすべての障害シナリオでこの確認を行うことができます。

別の方法として、ハイブリッド仮想化とミラーリング HA の手法 (このセクションの後半で説明) を使用している場合は、障害が発生したホスト・システムを仮想化プラットフォームから再起動できます。これにより、ミラーリング機能でプライマリ・インスタンスの以前のステータスを判断でき、その判断に応じて処理を続行できます。

仮想 IP アドレス (VIP) を使用するようにミラーを構成している場合は、アプリケーションの接続が新しいプライマリに透過的にリダイレクトされます。ECP による接続の場合は、自動的に新しいプライマリに再設定されます。アプリケーションの接続をリダイレクトする機能は、これらのほかにも用意されています。

プライマリ・インスタンスの動作が復帰した場合、このインスタンスは自動的にバックアップとなります。また、メンテナンスやアップグレードのための計画的な中断状態で、オペレータによるフェイルオーバーを使用して可用性を維持することもできます。

図 1-3: InterSystems IRIS ミラー



仮想化環境でミラーリング使用すると、その両方の利点を兼ね備えたハイブリッド高可用性ソリューションが作成されます。ミラーは計画的または計画外の停止に自動フェイルオーバーで即座に対応し、仮想 HA ソフトウェアは計画外のマシンの停止や OS の停止後にミラー・メンバをホストする仮想マシンを自動的に再起動します。これにより、障害が発生したメンバは、すぐにミラーに再参加できるようになり、バックアップとして動作します (必要であればプライマリを引き継ぎます)。

InterSystems IRIS ミラーリングの詳細は、“[ミラーリングの概要](#)” を参照してください。

1.5 フェイルオーバー方策での分散キャッシュの使用法

HA に向けてどの手法を採用する場合でも、エンタープライズ・キャッシュ・プロトコル (ECP) によって有効にされた分散キャッシュ・クラスタを使用することで、ユーザとデータベース・サーバとを分離するレイヤを実現できます。分散キャッシュ・クラスタ内のアプリケーション・サーバは、データ・サーバのフェイルオーバーが発生しても、その前後で実行中のアプリケーションの状態が維持されるように設計されています。データ・サーバに障害が発生しても、ユーザ側ではアプリケーション・サーバとの接続が維持されます。この中断の間にデータベースに頻繁にアクセスするユーザ・セッションは、フェイルオーバーの完了によってデータ・サーバが再び利用可能になるまで一時停止状態になります。アプリケーション・アクティビティとフェイルオーバー・メカニズムの特性に応じて、フェイルオーバーが完了するまで一時停止する場合がありますが、その後ワークフローを中断することなく動作を続行できます。

スタンドアロンの InterSystems IRIS インスタンスと同じように、高可用性を実現するために分散キャッシュ・クラスタ内のデータ・サーバをミラーリングし、フェイルオーバーが発生した場合に接続を自動的にバックアップにリダイレクトするようにアプリケーション・サーバを設定することもできます。分散キャッシュ・クラスタでのミラーリングの使用法の詳細は、“[ミラーへの ECP 接続の構成](#)” を参照してください。

ここで詳述されているその他のフェイルオーバーの方法は、分散キャッシュ・クラスタでも使用できます。データ・サーバに使用するフェイルオーバーの方法に関係なく、アプリケーション・サーバは、フェイルオーバーに続いて再接続してその状態を回復し、障害発生前の場所からアプリケーションのプロセスを続行できます。

ただし、分散キャッシュの主な目的は水平方向の拡張であることに注意してください。クラスタを単に HA の計画のコンポーネントとして配置することは、メリットだけでなく、複雑さの増大や新たな障害点の発生などの追加的なコストも生じる可能性があります。

分散キャッシュの詳細は、“[分散キャッシュによるユーザ数に応じた水平方向の拡張](#)”を参照してください。

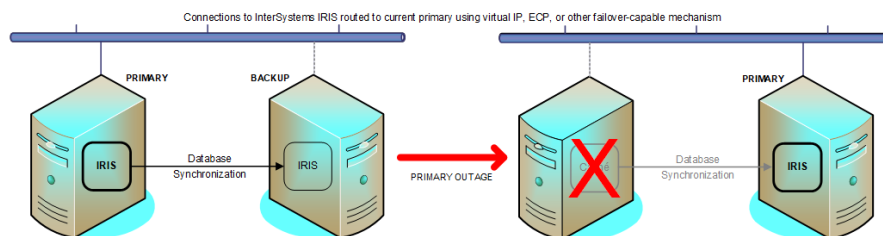
2

ミラーリングの概要

従来の高可用性ソリューションおよびデータ・レプリケーション・ソリューションは、しばしばインフラストラクチャ、配置、構成、ソフトウェアのライセンス、および計画に多額の資本投資を必要とします。InterSystems IRIS® のデータベース・ミラーリングは、2 つの InterSystems IRIS インスタンス間において、迅速で信頼性が高く強固な自動フェイルオーバーの経済的ソリューションを提供するように設計されており、企業の効果的な高可用性ソリューションを実現します。

共有リソース (共有ディスクなど) に依存した従来の可用性ソリューションは、その共有リソースの単一点障害の影響を受けることがしばしばあります。ミラーリングでは、プライマリ・ミラー・メンバとバックアップ・ミラー・メンバで独立したリソースを保持することによって、このリスクが低減されます。さらにミラーリングでは、論理データ・レプリケーションを利用することにより、SAN ベースのレプリケーションなどの物理的レプリケーション・テクノロジーに関連するリスク (不適切な更新、破損の繰り越しなど) が回避されます。

図 2-1: InterSystems IRIS ミラー



分散キャッシュとミラーリングの組み合わせにより、さらに高レベルの可用性が実現します。つまり、ミラーリングされたデータ・サーバと組み合わせた分散キャッシュ・クラスタ内のアプリケーション・サーバはミラー・フェイルオーバーをデータ・サーバの再起動として処理し、新しいプライマリで処理が遮られずに続行され、ワークフローおよびユーザの作業の中断が大幅に軽減されます。別々のデータ・センタで 2 つのフェイルオーバー・ミラー・メンバを構成すると、冗長性が向上し、致命的なイベントからも保護されるようになります。

ミラーリングは、計画外のダウンタイムに対する可用性ソリューションを提供するほかに、特定の InterSystems IRIS システムでの計画されたダウンタイム (例えば、InterSystems IRIS の構成変更、ハードウェアまたはオペレーティング・システムのアップグレードなど) を、組織のサービス水準合意 (SLA) 全体に影響を与えずに組み込む柔軟性も備えています。

最後に、ミラーにはフェイルオーバー・メンバのほかに、企業全体で複数のミラーから更新を受信するように構成できる特別な非同期メンバを使用できます。このことにより、単一のシステムが包括的なデータ・ウェアハウスとして動作できるようになり、InterSystems IRIS Business Intelligence™ を使用した企業全体のデータ・マイニングおよびビジネス・インテリジェンスが実現されます。また、非同期メンバは単一ミラーの災害復旧 (DR) に対しても構成できます。これにより、必要に応じて非同期メンバをシームレスにいずれかのフェイルオーバー・メンバの代わりとすることができます。1 つのミラーには、最大 16 のメンバを含めることができます。そのため、地理的に分散した多数の DR 非同期メンバを構成できます。このモデルは分散データ・レプリケーションのための強固なフレームワークを提供し、それによって組織にはビジネス継続性というメリットがもたらされるようになります。詳細は、“[ミラー停止の手順](#)”を参照してください。

3

ミラーリングのアーキテクチャおよび計画

ミラーとは、物理的に独立した複数の InterSystems IRIS インスタンスの論理グループで、プロダクション・データベースの同一コピーを同時に保持します。これによって、データベースへのアクセスを提供しているインスタンスが使用不可になった場合でも、別のインスタンスが引き継ぐことができます。ミラーでは、データベース（またはそのホスト・システム）へのアクセスを提供している InterSystems IRIS インスタンスに障害が発生した場合、別のインスタンスが自動的にかつ迅速に引き継ぐ処理である自動フェイルオーバーを通じた高可用性を実現できます。

ここでは、ミラーリングのコンポーネントおよびメカニズムについて説明し、ネットワーク要件、フェイルオーバー後のアプリケーション接続のリダイレクト、仮想環境でのミラーリングなど、ミラー計画におけるさまざまな問題についても説明します。

3.1 ミラー・コンポーネント

ミラーの一部として構成されている InterSystems IRIS インスタンスをホストするシステムをミラー・メンバと呼びます（同様に、InterSystems IRIS インスタンスをミラー・メンバと呼ぶ場合もあります）。ミラー・メンバには、次の 2 つのタイプがあります。

- ・ [フェイルオーバー・ミラー・メンバ](#)
- ・ [非同期ミラー・メンバ](#)

ほかに次の 2 つのコンポーネントが、1つのフェイルオーバー・メンバから別のフェイルオーバー・メンバへの自動フェイルオーバーをサポートします。

- ・ [ISCAgent](#)
- ・ [アービター](#)

3.1.1 フェイルオーバー・ミラー・メンバ

自動フェイルオーバーを有効にするには、2 つのフェイルオーバー・メンバ（物理的に独立したシステムで、それぞれが InterSystems IRIS インスタンスをホストしています）がミラーに含まれている必要があります。いつでも、一方のフェイルオーバー・インスタンスがプライマリとして動作して、ミラー内のデータベースへのアクセスをアプリケーションに提供し、他のインスタンスがバックアップとして動作して、これらのデータベースの同期コピーを保持し、プライマリとして引き継ぐことができるように準備しておきます。プライマリの InterSystems IRIS インスタンスが使用不可になった場合、バックアップが引き継いで、データ損失のリスクなしにデータベースへのアクセスを中断なく提供します。自動フェイルオーバー・プロセスに関する詳細は、“[自動フェイルオーバーのメカニズム](#)”を参照してください。

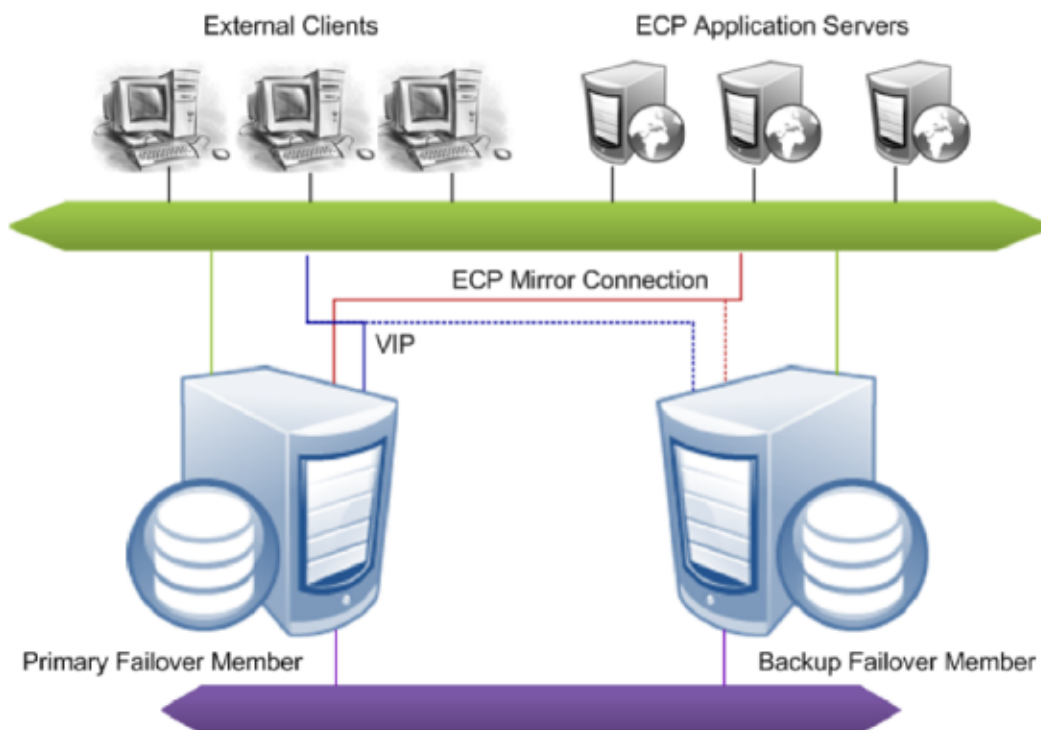
重要

プライマリではないフェイルオーバー・メンバの目的は、プライマリとして引き継ぐ準備をしておくことです。クエリやアプリケーション・コードを実行するためにこのメンバを直接使用することはサポートされていません。このメンバを使用してクエリやアプリケーション・コードを実行しようとすると、LOCK コマンドは〈PROTECT〉エラーで失敗します。

この同じ制限は、DR 非同期メンバおよびレポート非同期メンバには適用されません。

フェイルオーバー・メンバは、複数のミラー・メンバ・ネットワーク・アドレスを使用して、複数の通信チャンネル経由でお互いに通信します。外部クライアントは通常、現在のプライマリ上のインタフェースに常にバインドされる仮想 IP アドレス (VIP) 経由でミラーに接続します。ミラーリングされた分散キャッシュ・クラスタ内のアプリケーション・サーバ接続は、フェイルオーバー後に自動的に新しいプライマリにリダイレクトされます。したがって、この場合 VIP は不要です。

図 3-1: ミラーのフェイルオーバー・メンバ



ミラーのフェイルオーバー・メンバの構成に関する詳細は、“[ミラーの作成](#)”を参照してください。

重要

ミラー内の 2 つのフェイルオーバー・メンバは同等と見なされ、いずれもプライマリとして優先されることはありません。このため、プライマリとバックアップは単なる一時的な指定であると考えする必要があります。プライマリで問題が検出されてバックアップが引き継ぎ可能な場合、その引き継ぎはきわめて迅速に行われます。これは、プライマリ上の問題が、プライマリに十分な時間があれば自然に解決すると思われる場合であっても当てはまります。

フェイルオーバー・メンバ間のネットワーク遅延は、アプリケーションのパフォーマンスにおいて重要な要素であるため、接続における遅延が最小限に抑えられるように、フェイルオーバー・メンバの相対的な物理位置およびそれらのメンバ間のネットワーク接続を選択する必要があります。詳細は、“[ネットワーク遅延に関する考慮事項](#)”を参照してください。

3.1.2 非同期ミラー・メンバ

非同期メンバでは、ミラーリングされるデータベースの非同期コピーが保持されます。非同期メンバとしては、[災害復旧](#)と[レポート](#)の 2 つのタイプがあります。単一のミラーに、最大 16 のメンバを含めることができます。そのため、1 つのミラーに、1 つのフェイルオーバー・ペアと、上記の 2 つのタイプを任意に組み合わせた最大 14 の非同期メンバを構成でき

ます。自動フェイルオーバーなしで非同期メンバを利用するように、ミラーを**単一のフェイルオーバー・メンバ**で構成することもできます。

重要 非同期メンバのデータは、接続先のミラーで発生する変更に応じて継続的に非同期で更新されるため、非同期メンバ上での更新の同期およびクエリの結果の同期は保証されません。変化するデータに対するクエリの結果の整合性が保証されるかどうかは、非同期メンバに対して実行されるアプリケーション次第です。

また、非同期メンバは単一のミラーの災害復旧 (DR) 用にも構成できます。

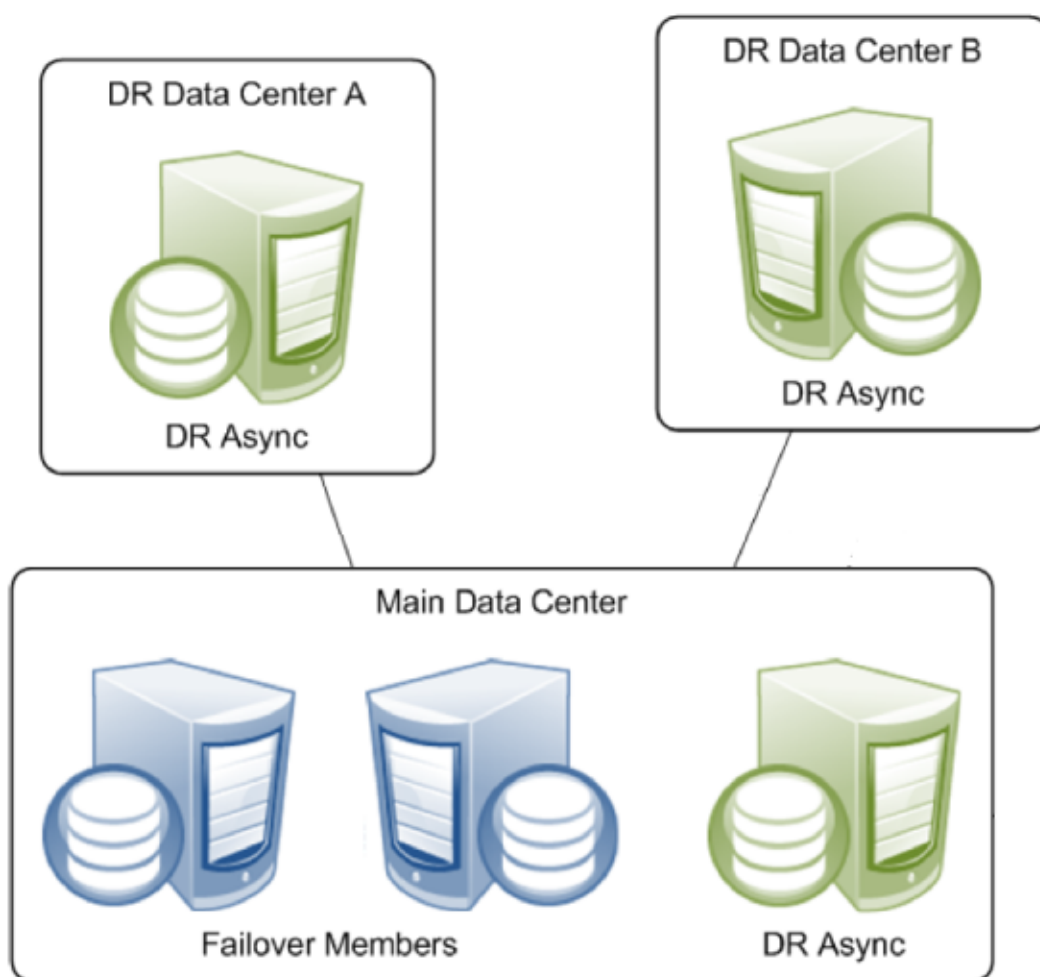
ミラーへの非同期メンバの追加に関する詳細は、“**非同期ミラー・メンバを構成する**”を参照してください。

3.1.2.1 災害復旧非同期

ミラーは、災害復旧 (DR) 非同期メンバを通して**災害復旧機能**を提供できます。DR 非同期メンバを**手動でフェイルオーバー・メンバに昇格**ことができ、両方のフェイルオーバー・メンバが災害のために使用できない場合は DR 非同期メンバをプライマリに設定することさえできます。昇格した DR は、計画的メンテナンスの実施やフェイルオーバー・メンバの一時的置き換えにおいても有用です。1 つの DR 非同期メンバは 1 つのミラーにしか属することができませんが、ミラー・メンバの制限数である 16 までなら、必要な数だけ 1 つのミラー内にメンバを構成することができます。

一般に、DR 非同期ミラー・メンバは geoレプリケーションの形式となります。

図 3-2: 単一のミラーに接続された複数の DR 非同期メンバ

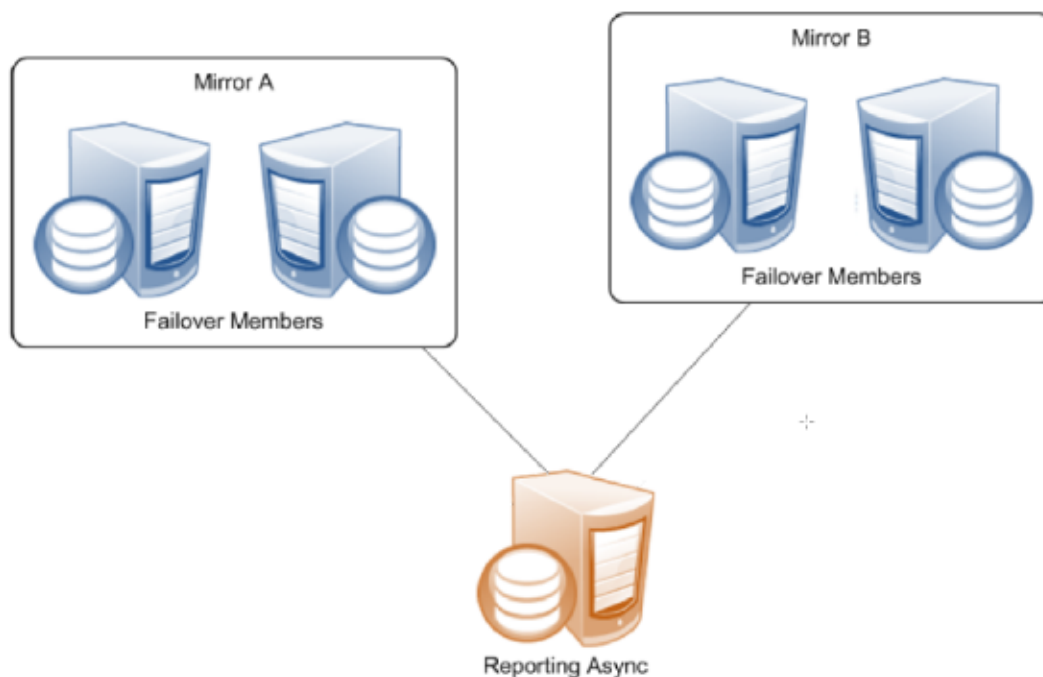


注釈 DR 非同期メンバは絶対に自動フェイルオーバーの候補とはなりません。自動フェイルオーバーは 1 つの**フェイルオーバー・ミラー・メンバ**から別のフェイルオーバー・ミラー・メンバに対してのみ行われます。

3.1.2.2 レポート非同期

レポート非同期ミラー・メンバでは、データ・マイニングやビジネス・インテリジェンスなどを目的として、選択されたデータベースの読み取り専用または読み取り/書込みのコピーが保持されます。レポート非同期メンバをフェイルオーバー・メンバに昇格することはできません。1 つのレポート非同期は最大 10 個のミラーに所属でき、別々の場所から一連の関連データベースをまとめる包括的エンタープライズ・データ・ウェアハウスとして機能できます。

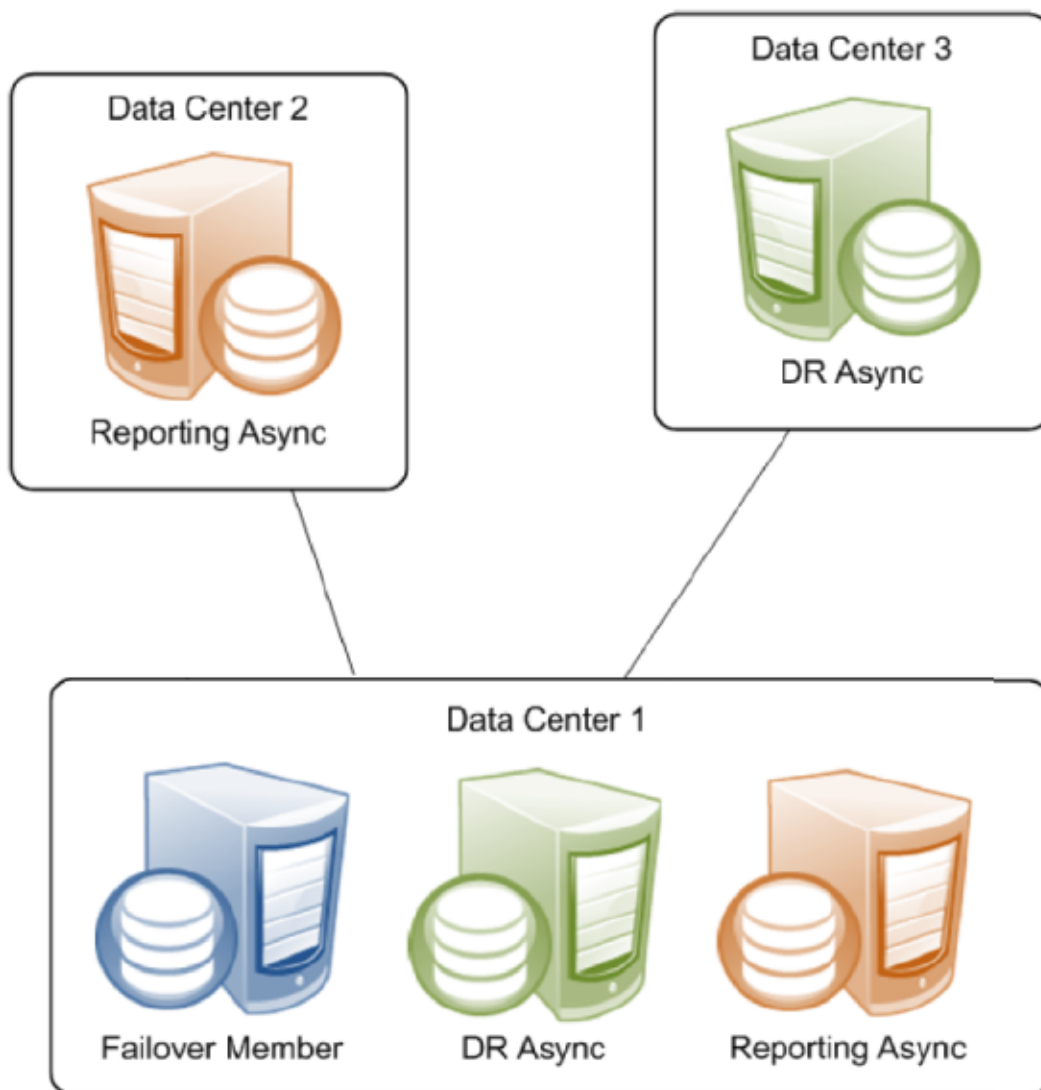
図 3-3: 複数のミラーに接続された単一レポート非同期メンバ



3.1.2.3 シングル・フェイルオーバー・ミラーの構成

1 つのミラーを、単一のフェイルオーバー・メンバと 1 つ以上の非同期メンバで構成することができます。この構成では高可用性は実現されませんが、他のニーズに対処できます。例えば、単一のフェイルオーバー・メンバ、1 つ以上の DR 非同期メンバ、および複数のレポート非同期メンバで構成されるミラーでは、データの収集と保存をサポートする一方で、データ・セキュリティおよび災害復旧を実現できます。高可用性を実現するには、フェイルオーバー・メンバを OS レベルのフェイルオーバー・クラスタまたはその他の高可用性構成に配置できます ([“高可用性を実現するためのフェイルオーバー方法”](#) を参照してください)。

図 3-4: 単一のフェイルオーバー・メンバと複数の非同期メンバ



3.1.3 ISCAgent

ISCAgent と呼ばれるプロセスは各ミラー・メンバのホスト・システム上で実行され、ミラー・メンバ間の通信に追加の手段を提供します。最も重要なのは、ISCAgent では、一方のフェイルオーバー・メンバが、もう一方のメンバに関する情報を、これら2つのメンバ間の通常の通信が中断された場合でも取得できるようにする方法が提供されていることです。ISCAgent は、ダウンしていたり切断しているミラー・メンバにデータを送信できます。このエージェントは、フェイルオーバーの決定にも関係しています。例えば、プライマリのインスタンスともアービターとも通信が途絶えたバックアップは、プライマリのインスタンスが本当にダウンしていることを引き継ぎの前に確認するために、プライマリの ISCAgent と通信できます (プライマリのホスト・システムがまだ動作していることが前提)。

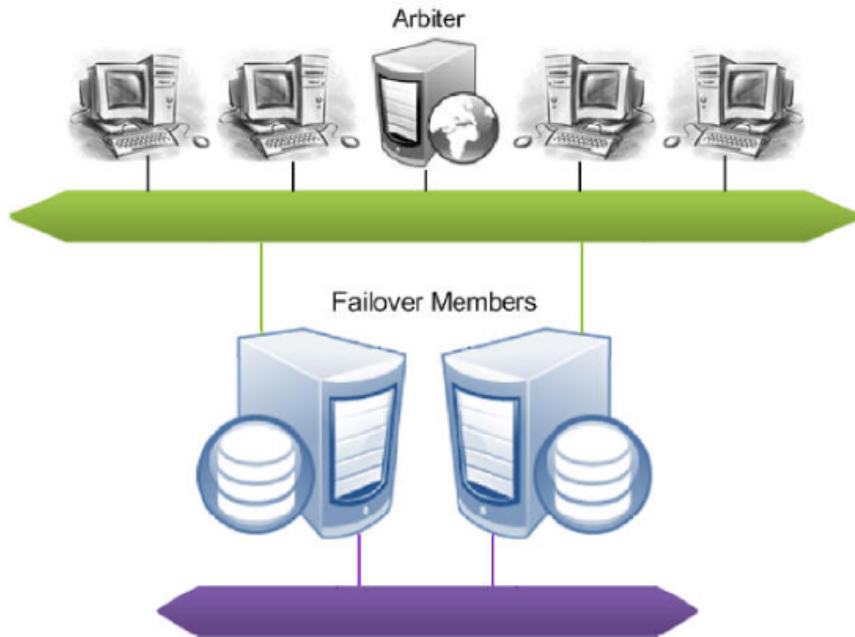
ISCAgent がまだインストールされていない場合には、InterSystems IRIS と共に自動的にインストールされます。1 つ以上のミラーに属する複数の InterSystems IRIS インスタンスが、1 つのシステムでホストされている場合、それらのインスタンスは、1 つの ISCAgent を共有します。

ISCAgent のロールおよび構成の詳細は、“[自動フェイルオーバーのメカニズム](#)” および “[ISCAgent の構成](#)” を参照してください。

3.1.4 アービター

アービターは、ミラーのフェイルオーバー・メンバとの継続的な通信が維持される [ISCAgent](#) をホストする独立システムで、直接通信できない場合にフェイルオーバーの決定を安全に下すために必要なコンテキストを提供します。1 つのアービターで複数のミラーに対処できますが、1 つのミラーでは一度に 1 つのアービターしか使用できません。アービターの使用は必須ではありませんが、[自動フェイルオーバー](#)が可能となる障害シナリオの範囲が大幅に広がるため、使用することを強くお勧めします。

図 3-5: ミラーのフェイルオーバー・メンバとアービター



注釈 バックアップがアクティブでない場合、フェイルオーバー・メカニズムでアービターが果たす役割は何もありません。

システムをアービターとして構成する場合は、最小限のソフトウェアのインストールで済み、InterSystems IRIS をインストールする必要はありません。アービターは、最小限のシステム・リソースを使用するため、他のサービスをホストしているシステムに配置でき、ワークステーションにさえ配置できます。アービターに関する第一の要件は、アービターと単一のフェイルオーバー・メンバに同時に計画外の障害が発生するリスクを最小限に抑えるようにアービターを配置して、構成する必要がありますということです。詳細は、“[ミラーの可用性を最適化するためのアービターの配置](#)”を参照してください。

3.2 ミラー同期

“[ジャーナリング](#)”で述べているように、ジャーナル・ファイルには、最後のバックアップ以後、InterSystems IRIS インスタンスのデータベースで行われた変更の記録が時間順に含まれています。ミラー内では、プライマリ上のデータベースに加えられた変更を記録するジャーナル・データが、バックアップおよび非同期上のデータベースのコピーに同じ変更を加える際の基礎となります。このため、ミラーリングされたデータベースは常にプライマリ上で[ジャーナリングされる](#)一方で、バックアップおよび DR 非同期上では、他のソースから更新されないように常に読み取り専用となります。通常、レポート非同期上でも読み取り専用です。

ミラーリングされたデータベースでグローバル更新処理（主に Set 処理と Kill 処理）を記録するデータがプライマリでジャーナルに書き込まれると、ジャーナル・レコードは他のミラー・メンバに送信されます。バックアップまたは非同期メン

バでジャーナル・レコードが受信されると、それらのレコードに記録されている処理がそのメンバ上のデータベースで実行されます。このプロセスはデジャーナリングと呼ばれます(非同期メンバでのデジャーナリングの管理に関する重要な情報は、“[データベース・デジャーナリングの管理](#)”を参照してください)。

プライマリからバックアップへのジャーナル・レコードの転送は同期的に実行され、プライマリはバックアップからの応答を重要なポイントで待機します。これによって、“[バックアップ・ステータスと自動フェイルオーバー](#)”で詳細に説明されているように、フェイルオーバー・メンバは緊密に同期され、バックアップはアクティブである状態が保持されます。一方、非同期はプライマリから非同期的にジャーナル・データを受信します。このため、非同期ミラー・メンバはプライマリよりジャーナル・レコードが数個分遅れている場合があります。

注釈 InterSystems IRIS インスタンスがミラーのメンバとなった場合、以下のミラーリング・サポート用のジャーナル変更が発生します。

- ・ InterSystems IRIS インスタンスがミラーでプライマリ・フェイルオーバー・メンバとなった場合、以下の変更が生じます。
 - **MIRROR-mirror_name** が先頭に付く新規ジャーナル・ファイル (**MIRROR-MIR21-20180921.001** など) に対してのジャーナル切り替えが発動します。その時点より、すべてのジャーナル・ファイルはミラー・ジャーナル・ファイルとして書き込まれ、**mirrorjrn-mirror_name.log** (**mirrorjrn-MIR21.log** や **journal.log** など) にログが記録されます。
 - **[エラー発生時に凍結する]** ジャーナリング構成は、ジャーナル入出力エラーが発生したときに自動的にオーバーライドされ、現在の設定に関係なく、ジャーナルされているグローバルの更新をすべて凍結します。現在の設定が [いいえ] の場合、インスタンスがプライマリ・フェイルオーバー・メンバでなくなったときに動作がこの設定に戻ります。この影響について理解するには、“[ジャーナル設定の構成](#)”と“[ジャーナル入出力エラー](#)”を参照してください。
- ・ インスタンスがバックアップまたは非同期のミラー・メンバになるとき、プライマリから受信したミラー・ジャーナル・ファイルは、ローカル・インスタンスの標準ジャーナル・ファイルと共に、構成済みのジャーナル・ディレクトリに書き込まれ、プライマリのミラー・ジャーナル・ログ (**mirrorjrn-mirror_name.log**) のコピーが **install-dir\Mgr** に作成され、継続的に更新されます。

ジャーナリングに関する一般情報は、“[ジャーナリングの概要](#)”を参照してください。

3.3 自動フェイルオーバーのメカニズム

プライマリに障害が発生したり使用できなくなった場合にバックアップへの安全な自動フェイルオーバーを提供するようにミラーリングは設計されています。このセクションでは、この処理を可能にするメカニズムについて説明します。内容は以下のとおりです。

- ・ [安全な自動フェイルオーバーの要件](#)
- ・ [自動フェイルオーバーのルール](#)
- ・ [プライマリ停止シナリオに対するミラーの対応](#)
- ・ [ミラーの可用性を最適化するためのアービターの配置](#)
- ・ [自動フェイルオーバーのメカニズムの詳細](#)

3.3.1 安全な自動フェイルオーバーの要件

バックアップの InterSystems IRIS インスタンスは、次の 2 つの条件が満たされることを確認できた場合にのみ、プライマリから自動的に引き継ぐことができます。

- ・ バックアップのインスタンスがプライマリから最新のジャーナル・データを受信した。
この要件によって、停止前にプライマリ上のミラーリング・データベースで行われた永続的な更新はすべて、バックアップ上の同じデータベースに対しても行われているか、またはこれから行われることが保証され、データが損失されることはなくなります。
- ・ プライマリのインスタンスは、もはやプライマリとしては動作しておらず、手動操作なしではプライマリとして動作できない。
この要件によって、両方のフェイルオーバー・メンバが同時にプライマリとして動作する可能性が除去されます。そのような状況が発生すると、論理データベースの性能低下と整合性の損失をまねくおそれがあります。

3.3.2 自動フェイルオーバーのルール

このセクションでは、自動フェイルオーバー・プロセスを制御し、双方の自動フェイルオーバー要件が確実に満たされるようにするルールについて説明します。

注釈 以下の条件が当てはまらない場合、いかなる状況でもバックアップはプライマリ化を試行しません。

- ・ **[開始時にマウントが必要]** が選択されたデータベースは、ミラーリングされるものもミラーリングされないものも両方ともすべてマウントされている。
- ・ **[開始時にマウントが必要]** が選択されたミラーリングされるデータベースはすべて有効化され、キャッチアップされている ("[ミラーリングされるデータベースの有効化とキャッチアップ](#)" を参照)。

[開始時にマウントが必要] に関する情報は、"[ローカル・データベースのプロパティの編集](#)" を参照してください。

3.3.2.1 バックアップ・ステータスと自動フェイルオーバー

通常のミラー処理中は、バックアップ・フェイルオーバー・メンバのジャーナル転送ステータスはアクティブになっており、プライマリからすべてのジャーナル・データを受信してプライマリと同期していることを意味します(フェイルオーバー・メンバのデータベースがジャーナル・データおよび関連する詳細を使用して同期される方法については、"[ミラー同期](#)" を参照してください。また、ミラー・メンバのステータスの監視については、"[ミラーの監視](#)" を参照してください)。アクティブなバックアップは、現在のジャーナル・データをプライマリで書き込まれたままの状態を受信し、アクティブなバックアップがジャーナル・データの受信を確認しそのデータを永続的と見なすのをプライマリは待機します。これによって、アクティブなバックアップは、フェイルオーバーの 1 つ目の条件を満たします。

アクティブなバックアップが **サービス品質 (QoS) タイムアウト** 内にプライマリからの新しいデータの受信を確認しなかった場合、プライマリはバックアップのアクティブ・ステータスを取り消し、そのバックアップの接続を解除して、一時的に障害状態に入ります。障害状態の間、プライマリは新しいジャーナル・データをコミットしないので(おそらくアプリケーションでの一時停止を引き起こす)、2 つのメンバが非同期にはならず通信をリストアする時間か適切で安全なフェイルオーバーの決定を下す時間を確保できます。

バックアップがプライマリに再接続すると、まずプライマリから最新のジャーナル・データをすべて取得することによってキャッチアップした後で、アクティブになります。バックアップが最新ジャーナル・データをプライマリから取得してその受信を確認することでキャッチアップすると、そのアクティブ・ステータスがリストアされます。

3.3.2.2 バックアップがアクティブな場合の自動フェイルオーバー

バックアップがアクティブな際、プライマリがプライマリとしては動作しておらずに操作員の操作なしでプライマリとして動作できないという 2 つ目のフェイルオーバー条件を確認できる場合、プライマリとして引き継ぐことができます。バックアップは次の 3 つの方法のいずれかで引き継ぎができます。

- ・ 引き継ぎを要求するプライマリから通信を受信する方法。

これは、プライマリ・インスタンスの通常シャットダウン時またはハング状態をプライマリが検出した際に発生します。プライマリがこのメッセージを送信すると、プライマリとして動作できなくなり、アクティブなバックアップが安全に引き継ぐことができます。前のプライマリがハングしている場合、新しいプライマリはそれを強制終了します。

- ・ プライマリとの通信を失ったという情報をアービターから受信する方法。

プライマリとバックアップの InterSystems IRIS インスタンスでは、アービターとの継続的な通信が維持されており、これによって、他方のフェイルオーバー・メンバとの通信が切断されたかリストアされた場合は必ず、それぞれが更新されます。ネットワーク・イベントによってバックアップとアービターの両方から同時にプライマリが切り離された場合、無限に障害状態に入ります。このように、アクティブなバックアップがプライマリとの通信を失い、アービターもプライマリとの通信が途絶えていることをアービターから確認すると、バックアップは安全に引き継ぐことができます。プライマリは障害が発生しているか切り離されていて、障害状態に入っているため、プライマリとして動作できないことがわかるからです。接続が回復すると、前のプライマリがハングしている場合、新しいプライマリはそれを強制終了します。

- ・ プライマリ・システムの ISCAgent から、プライマリのインスタンスがダウンまたはハングしているという情報を受信する方法。

アービターが利用できない場合や、アービターが構成されていない場合、プライマリ・インスタンスとの通信が途絶えたアクティブなバックアップは、プライマリの ISCAgent への通信を試行します（これは、プライマリのホスト・システムが動作を維持している場合のみ可能です）。これにより、プライマリ・インスタンスがダウンしているかどうかを確認し、ハング状態であった場合は強制終了させます。プライマリがプライマリとしての役割を果たせなくなっていて、そのためにフェイルオーバーが安全であることをエージェントが確認すると、バックアップがプライマリの役割を引き継ぎます。

ネットワーク・イベントによってプライマリがアクティブなバックアップから切り離されているが、バックアップでこれら3つの方法のいずれでも安全なフェイルオーバー状態を確保できない場合、バックアップはアクティブではなくなり、次のセクションで説明しているフェイルオーバー・メカニズムに従います。

重要 プライマリが強制終了されると、手動でインスタンスが再起動されるまで、終了したままになります。

3.3.2.3 バックアップがアクティブでない場合の自動フェイルオーバー

アクティブでないバックアップは、プライマリの ISCAgent との通信を試行して、プライマリ・インスタンスがダウンしているかどうかを確認するか、ハング状態の場合は強制終了させて、エージェントからプライマリの直近のジャーナル・データを取得するよう試行できます。旧プライマリの終了を確認し、ジャーナルを取得したら、バックアップはプライマリとして安全に引き継ぐことができます。

アクティブでないバックアップがプライマリの ISCAgent と通信できない場合、そのバックアップは、プライマリがプライマリとしての役割を果たせなくなっていることを確認する手段も、プライマリから最新のジャーナル更新を入手する手段もないため、プライマリの役割を引き継ぐことはできません。

バックアップがアクティブでない場合、フェイルオーバー・メカニズムでアービターが果たす役割は何もありません。

3.3.3 さまざまな停止シナリオに対するミラーの対応

このセクションでは、フェイルオーバー・メンバとアービターがさまざまな組み合わせで停止した場合のミラーの対応の概要を示します。

注釈 オペレータはフェイルオーバーを発動させずにプライマリ・システムを一時的にダウン状態にできます（“[フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避](#)”を参照）。これは、例えばメンテナンスのためにプライマリ・システムを非常に短時間ダウン状態にする必要がある場合に便利です。プライマリ・システムを稼働状態に戻した後、自動フェイルオーバーの既定の動作が復元されます。

ここで示すシナリオのいくつかでは、バックアップを手動で強制的にプライマリにするオプションに言及しています。この手順についての詳細は、“[自動フェイルオーバーを使用しないプライマリ・フェイルオーバー・メンバの計画外停止](#)”を参照してください。

3.3.3.1 プライマリ停止シナリオに対応した自動フェイルオーバー

状況も詳細も異なる中で、アクティブなバックアップ・フェイルオーバー・メンバによる自動引き継ぎが実行される主要なプライマリ停止シナリオがいくつかあり、それらを以下に示します。

1. 例えばメンテナンスのためなど、プライマリの計画停止が、プライマリの InterSystems IRIS インスタンスをシャットダウンすることによって開始しました。

プライマリがアクティブなバックアップに引き継ぎを指示したために自動フェイルオーバーが発生します。

2. プライマリの InterSystems IRIS インスタンスが、予期しない状態によりハングしました。

プライマリがハングを検出し、アクティブなバックアップに引き継ぎを指示したため、自動フェイルオーバーが発生します。

3. 予期しない状態により、プライマリの InterSystems IRIS インスタンスが強制的にダウン状態になり、完全に無応答になりました。

このシナリオでは、プライマリはバックアップに引き継ぎを指示できません。ただし、プライマリとの通信を失っていることをアービターから確認できた後か、プライマリの ISCAgent と通信してプライマリがダウンしていることを確認した場合、アクティブなバックアップが引き継ぎます。

4. プライマリのストレージ・サブシステムに障害が発生しました。

ストレージに障害が発生すると通常はその結果として、入出力エラーによりプライマリのインスタンスがハングします。この場合、プライマリはハングを検出して、アクティブなバックアップに引き継ぎを指示します（シナリオ 2 と同様）。ただし、場合によっては、シナリオ 3 またはシナリオ 5 で示した動作が当てはまることがあります。

5. プライマリのホスト・システムに障害が発生したか応答しなくなりました。

アクティブなバックアップがアービターから、プライマリとの通信が途絶えていることを確認できると、自動フェイルオーバーが発生します。

アービターが構成されていないか、またはプライマリのホストに障害が発生する前にアービターが使用不可になっていた場合、自動フェイルオーバーは発生しません。このような場合、1 つのオプションとして、バックアップを手動で強制的にプライマリにできます。

6. ネットワーク上の問題によってプライマリが切り離されました。

アービターが構成されていて、ネットワーク障害発生時に両方のフェイルオーバー・メンバがアービターに接続されていた場合、プライマリは無限に障害状態に入ります。

- ・ プライマリとの通信が途絶えていることをアクティブなバックアップがアービターから確認すると、自動フェイルオーバーが発生します。
- ・ バックアップがプライマリとの通信を失ったと同時にアービターとの通信も失った場合、自動フェイルオーバーはできません。両方のフェイルオーバー・メンバが稼働していれば、ネットワークのリストア時にバックアップはプライマリと通信します。これによってプライマリとして処理を再開します。または、プライマリを手動で指定できます。

アービターが構成されていないか、ネットワークに障害が発生する前にいずれかのフェイルオーバー・メンバが切断されていた場合、自動フェイルオーバーはできず、プライマリはプライマリとして動作し続けます。

アクティブではないバックアップ（起動中であるか、遅れを取ってしまっているため）は、前述のシナリオ 1 から 4 までの場合、プライマリの ISCAgent と通信して、最新ジャーナル・データを取得することで引き継ぐことができます。シナリオ 5 および 6 の場合、アクティブではないバックアップは、ISCAgent と通信できないため、引き継ぐことができません。このような場合、1 つのオプションとして、バックアップを手動で強制的にプライマリにできます。

3.3.3.2 アービター停止の影響

アービターの停止がミラーの機能に直接的な影響を及ぼすことはありません。ただし、アービターがリストアされる前に、“[プライマリ停止シナリオに対応した自動フェイルオーバー](#)” に示したプライマリ停止シナリオの 5 または 6 が発生すると、バックアップによる自動的な引き継ぎができなくなります。

3.3.3.3 バックアップ停止の影響

アプリケーションによっては、プライマリが処理を再開する前に、短い間（[サービス品質タイムアウト](#)程度の時間）一時停止するものがあります。アービターが構成されていないか、バックアップの停止前にアービターが使用不可になっていた場合、発生する一時停止の時間はわずかに長くなることがあります（サービス品質タイムアウトの 3 倍程度の時間）。バックアップがリストアされる前にプライマリの停止が発生すると、ミラー全体の停止になります。

3.3.3.4 プライマリの停止とアービターの停止が組み合わさった場合の影響

このシナリオの結果は、“[プライマリ停止シナリオに対応した自動フェイルオーバー](#)” で説明しています。簡単に言うと、バックアップがプライマリの ISCAgent と通信できる場合は引き継ぎ、通信できない場合はミラー全体の停止となりますが、適切なオプションとして、手動操作によりバックアップを強制的にプライマリにできます。

3.3.3.5 バックアップの停止とアービターの停止が組み合わさった場合の影響

バックアップとアービターが同時に（またはほとんど同時に）使用不可になった場合、プライマリは無限に障害状態のままです。これは、切り離されているからバックアップがプライマリになることができると見なされるためです。この結果、ミラー全体の停止となります。バックアップが再度使用可能になると、プライマリと通信します。これによってプライマリとして処理を再開します。または、プライマリを手動操作で強制的に再開することもできます。バックアップとアービターに順に障害が発生した場合、“[バックアップ停止の影響](#)” で説明した短い一時停止の後、プライマリはプライマリとして動作を続けます。これは、バックアップはプライマリになることができないと認識されるためです。

3.3.3.6 プライマリの停止とバックアップの停止が組み合わさった場合の影響

この組み合わせでは常に、ミラー全体の停止となります。この場合に使用可能なオプションについては、“[両方のフェイルオーバー・メンバの計画外停止](#)” を参照してください。

3.3.4 ミラーの可用性を最適化するためのアービターの配置

フェイルオーバー・メンバとアービターは共同でミラーリング高可用性ソリューションを提供します（アービターは最小限の役割を果たします）。アービターはクォーラム・メカニズムではなく、フェイルオーバー・メンバ間の通信が途絶えた場合にそれぞれにコンテキストを提供することにより、自動フェイルオーバーの処理で各フェイルオーバー・メンバをサポートします。どのような種類のプライマリ停止であってもその直前に両方のフェイルオーバー・メンバがアービターと通信しており、バックアップもアービターと通信している限り、自動フェイルオーバーができます。アービターに障害が発生すると、場合によっては自動フェイルオーバーが発生する可能性がなくなりますが、置換が構成されている際にミラーの動作を妨げることも、多くのプライマリ停止シナリオ（例えば、“[プライマリ停止シナリオに対応した自動フェイルオーバー](#)” のシナリオ 1 から 4）でのミラーによる自動フェイルオーバーの実行を妨げることもありません。

これらの理由により、それぞれが独立して高い可用性を持つフェイルオーバー・メンバよりもすぐれた高可用性をアービターが持つ必要はなく、アービターと単一フェイルオーバー・メンバに計画外の停止が同時に発生するリスクが最小限に抑えられるようにアービターを配置および構成することのみが必要です。（両方のフェイルオーバー・メンバに障害が発生するとミラーも失敗し、アービターのステータスは関係ないため、3 つすべてが同時に停止するリスクを考慮する必要はありません。）

この要件に基づいて、一般的に、フェイルオーバー・メンバが互いに分離されているのと同じだけ、アービターをフェイルオーバー・メンバから分離することをお勧めします。具体的には以下の処理を行います。

- ・ フェイルオーバー・メンバが1つのデータ・センタ内に配置されている場合、アービターを同じデータ・センタ内に配置できます。データ・センタ内で、フェイルオーバー・メンバが互いに物理的に分離されているのと同じだけ、アービターをフェイルオーバー・メンバから物理的に分離する必要があります。例えば、フェイルオーバー・メンバを別々のサーバ・ラックに配置して、1つのラックに電力またはネットワークの問題が発生した際に両方のメンバに影響が及ばないようにしている場合、アービターをそれら2つのラックとは別に配置する必要があります。

データ・センタでミラー内の通信に内部ネットワークが使用されている場合、アービターをネットワークのパブリック側に配置して、内部ネットワーク内の障害によって、フェイルオーバー・メンバが互いに切り離されることも、アービターから切り離されることもないようにする必要があります。

- ・ フェイルオーバー・メンバが別々のデータ・センタ内に配置されている場合、アービターを3つ目の場所に配置する必要があります。その場所としては、別のデータ・センタ、別の関係者によってホストされている場所、パブリックまたはプライベートのクラウド・サービスが考えられます。ユーザ・コミュニティを代表する場所にアービターを配置すると、ネットワーク停止時の最適なミラー対応を支援できます。

1つのシステムが複数のミラーそれぞれに適切な場所にあれば、そのシステムを複数のミラーのアービターとして構成できます。そのように構成するには、システムがアービターとして機能する対象の各ミラーを作成または編集するときに、“[ミラーの作成](#)”の説明にあるようにホストとポート番号を指定します。

新しく配置されたシステムまたは専用のシステムでアービターをホストする必要はありません。実際、信頼性が適切に確立されている既存ホストが望ましい場合があります。レポート非同期ミラー・メンバ (“[レポート非同期](#)”を参照) が適切なホストとして機能できます。ただし、DR 非同期でのホストは避ける必要があります。メンテナンスまたは障害発生シナリオでの DR 非同期の昇格 (“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”を参照) によって、アービターがフェイルオーバー・ミラー・メンバ上でホストされる可能性があり、これは間違った構成です。

注釈 “[アービターのインストール](#)”で説明するように、ISCAgent を実行しているシステムは、そのシステムが1つ以上の InterSystems IRIS インスタンスをホストしている場合を含め、アービターとして構成できます。ただし、ミラーのフェイルオーバー・メンバまたは DR 非同期メンバを1つ以上ホストしているシステムは、そのミラーのアービターとして構成してはいけません。

3.3.5 自動フェイルオーバーのメカニズムの詳細

このセクションでは、フェイルオーバーのメカニズムに関して追加の詳細を提示します。

フェイルオーバー・メンバ間またはフェイルオーバー・メンバとアービター間の通信が途絶えた場合のミラーの対応は、次の2つの異なるミラー・フェイルオーバー・モードの使用によってサポートされます。

- ・ [エージェント制御モード](#)
- ・ [アービター制御モード](#)

3.3.5.1 エージェント制御モード

ミラーの開始時に、フェイルオーバー・メンバはエージェント制御モードで動作を開始します。アービターが使用不可になっているか構成されていない場合、このモードのままです。エージェント制御モードでは、フェイルオーバー・メンバは、以下で説明しているように、お互いの間の通信の損失に対応します。

通信が途絶えた場合のプライマリの対応

プライマリからアクティブなバックアップへの接続が失われた場合、またはデータ受信の確認を待機している時間が[サービス品質タイムアウト](#)を超えた場合、プライマリはバックアップのアクティブ・ステータスを取り消して、障害状態に入り、アクティブでなくなったことがバックアップで確認するのを待機します。プライマリがバックアップからの確認を受信するか、障害タイムアウト(サービス品質タイムアウトの2倍)の時間が過ぎると、プライマリは障害状態から出て、プライマリとして動作を再開します。

プライマリからアクティブではないバックアップへの接続が失われた場合、プライマリとして動作を続け、障害状態には入りません。

通信が途絶えた場合のバックアップの対応

バックアップがプライマリとの接続を失った場合や、プライマリからのメッセージの待機時間がサービス品質タイムアウトを超えた場合、そのバックアップはプライマリの ISCAgent との通信を試行します。そのエージェントから、プライマリのインスタンスはプライマリとして引き続き動作していることが報告されると、バックアップは再接続します。プライマリがダウンしているか、プライマリが強制終了されていることをエージェントが確認すると、バックアップは以下のように動作します。

- ・ バックアップがアクティブで、プライマリが障害タイムアウトの範囲内でダウンしていることをエージェントが確認した場合、バックアップはプライマリの役割を引き継ぎます。
- ・ バックアップがアクティブでない場合や、障害タイムアウトを超過している場合、そのバックアップは、プライマリがダウンしていることをエージェントが確認して、エージェントから最新のジャーナル・データを取得できたときに、プライマリの役割を引き継ぎます。

バックアップは、アクティブであるかどうかにかかわらず、エージェント制御モードの状態で自動的にプライマリの役割を引き継ぐことはありません。ただし、プライマリ自体がプライマリのハング状態を確認した場合、またはプライマリのエージェントが（強制終了させた後で）プライマリのダウン状態を確認した場合を除きますが、どちらの状況も、プライマリのホストがダウンしている場合や、ネットワークから切り離されている場合は発生しません。

注釈 いずれかのフェイルオーバー・メンバが再起動すると、他方の ISCAgent との通信を試行し、その動作はアクティブではないバックアップで説明したものと同一になります。

3.3.5.2 アービター制御モード

フェイルオーバー・メンバが互いに接続していて、両方がアービターに接続しており、バックアップがアクティブな場合、アービター制御モードに入ります。このモードでは、フェイルオーバー・メンバは、お互いの間の通信が途絶えた場合に、アービターから提供される他方のフェイルオーバー・メンバに関する情報に基づいて対応します。それぞれのフェイルオーバー・メンバは、アービターとの接続の損失に対して、自身から他方のフェイルオーバー・メンバへの接続と、その逆の接続をテストすることによって対応するため、1つのネットワーク・イベントから生じる複数の接続損失が、1つのイベントとして処理されます。

アービター制御モードでは、いずれかのフェイルオーバー・メンバがアービターへの接続のみを失ったか、バックアップがアクティブ・ステータスを失った場合に、フェイルオーバー・メンバはエージェント制御モードへの切り替えを調整して、そのモードについて説明したように、さらなるイベントに対して対応します。

アービター制御モードで、プライマリとバックアップの間の接続が失われた場合、それぞれのフェイルオーバー・メンバは、以下に説明しているようにアービターとの接続状態に基づいて対応します。

プライマリがバックアップへの接続を失った場合

プライマリがアクティブなバックアップへの接続を失った場合や、データの受信確認を待機する時間がサービス品質タイムアウトを超過した場合に、アービターもバックアップへの接続を失っているか、バックアップからの応答を待機する時間がサービス品質タイムアウトを超過していることをアービターから通知された場合、プライマリはプライマリとしての動作を継続し、エージェント制御モードに切り替えます。

アービターがバックアップに接続されていることをプライマリで確認すると、障害状態に入り、アービターを通じてバックアップとエージェント制御モードへの切り替えの調節を試行します。調節した切り替えが実行されるか、またはバックアップがアービターにはもう接続されていないことをプライマリが確認すると、プライマリはプライマリとしての通常の動作に戻ります。

バックアップへの接続と同様にアービターへの接続もプライマリが失った場合、無限に障害状態のままになるため、バックアップは安全に引き継ぐことができます。フェイルオーバーが発生すると、接続のリストア時にプライマリはシャットダウンします。

注釈 障害タイムアウトは、アービター制御モードでは適用されません。

バックアップがプライマリへの接続を失った場合

バックアップがプライマリへの接続を失った場合や、プライマリからのメッセージを待機する時間がサービス品質タイムアウトを超過した場合に、アービターもプライマリへの接続を失っているか、プライマリからの応答を待機する時間がサービス品質タイムアウトを超過していることをアービターから通知された場合、バックアップはプライマリとして役割を引き継ぎ、エージェント制御モードに切り替えます。接続が回復すると、前のプライマリがまだダウンしていない場合は、新しいプライマリはそれを強制終了します。

アービターがプライマリに接続されていることをバックアップで確認すると、バックアップは自身がアクティブであるとは見なさなくなり、エージェント制御モードに切り替え、アービターを通じてエージェント制御モードへのプライマリの切り替えに合わせます。その後バックアップは、プライマリへの再接続を試行します。

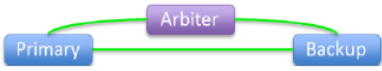
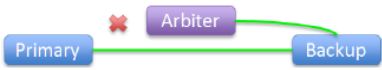

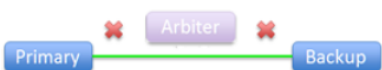
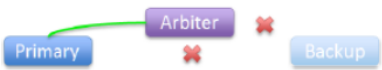
プライマリへの接続と同様にアービターへの接続もバックアップが失った場合、エージェント制御モードに切り替えて、エージェント制御メカニズムを通じてプライマリの ISCAgent への通信を試みます。



失われた接続に対するミラーの対応

次の表は、アービター制御モードで起こりうる接続の損失のすべての組み合わせに対するミラーの対応を示しています。最初の 3 つの状況は、ネットワーク障害のみを表し、それ以外には、フェイルオーバー・メンバの視点から見て、システム障害とネットワーク障害のいずれか、またはそれらの組み合わせが関係している可能性があります。説明では、1 つ以上の接続が失われる直前には、フェイルオーバー・メンバおよびアービターはすべて互いに通信し合っていて、バックアップはアクティブであったことが前提となっています。

注釈 アービター制御モードでの接続の損失のほとんどの組み合わせに対して、ミラーは、エージェント制御モードに切り替えて対応します。このため、1 つの障害イベントが処理された後、すべての接続が再確立される前に生じる後続のイベントに対する対応は、このテーブルで説明している対応ではなく“エージェント制御モード”で説明している対応と同じです。

テーブル 3-1: アービター・モードで失われた接続に対するミラーの対応

	<p>3つのシステムすべてが接続されている：</p> <ul style="list-style-type: none"> ミラーはアービター制御モードになります（まだアービター制御モードになっていない場合）。
	<p>バックアップはアービターへの接続を失うが、まだプライマリには接続されている：</p> <ul style="list-style-type: none"> ミラーはエージェント制御モードに切り替えます。 プライマリは引き続きプライマリとして動作します。 バックアップはアービターへの再接続を試みます。
	<p>プライマリはアービターへの接続を失うが、まだバックアップには接続されている：</p> <ul style="list-style-type: none"> ミラーはエージェント制御モードに切り替えます。 プライマリは引き続きプライマリとして動作します。 プライマリはアービターへの再接続を試みます。
	<p>フェイルオーバー・メンバは互いへの接続を失うが、まだアービターには接続されている：</p> <ul style="list-style-type: none"> ミラーはエージェント制御モードに切り替えます。 プライマリは引き続きプライマリとして動作します。 バックアップはプライマリへの再接続を試みます。
	<p>アービターは障害が発生しているか切り離されている－フェイルオーバー・メンバはアービターの接続を失うが、まだ互いに接続されている：</p> <ul style="list-style-type: none"> ミラーはエージェント制御モードに切り替えます。 プライマリは引き続きプライマリとして動作します。 両方のフェイルオーバー・メンバはアービターへの再接続を試みます。
	<p>バックアップは障害が発生しているか切り離されている－プライマリとアービターはバックアップへの接続を失うが、まだ互いに接続されている：</p> <ul style="list-style-type: none"> プライマリはエージェント制御モードに切り替えて、引き続きプライマリとして動作します。 バックアップ（動作中の場合）はエージェント制御モードに切り替えて、プライマリへの再接続を試みます。

	<p>プライマリは障害が発生しているか切り離されている – バックアップとアービターはプライマリへの接続を失うが、まだ互いに接続されている：</p> <ul style="list-style-type: none"> ・ プライマリ（動作中の場合）はいつまでもアービター制御モードで、障害状態のままになります。 ・ バックアップはプライマリを引き継ぎ、エージェント制御モードに切り替えて、接続が回復したら、プライマリを強制的にダウンさせます。
	<p>3 つの接続すべてが失われている：</p> <ul style="list-style-type: none"> ・ プライマリ（動作中の場合）はいつまでもアービター制御モードで、障害状態のままになります。バックアップがプライマリに通信すると、プライマリはエージェント制御モードに切り替えて、プライマリとして動作を再開します。 ・ バックアップ（動作中の場合）はエージェント制御モードに切り替えて、プライマリへの再接続を試みます。 <p>注釈 1 つのイベント（または複数の同時イベント）が原因ですべての接続が失われることはめったにありません。ほとんどの場合、ミラーはすべての接続が失われる前にエージェント制御モードに切り替えます。その場合、次のようになります。</p> <ul style="list-style-type: none"> ・ プライマリ（動作中の場合）は引き続きプライマリとして動作します。 ・ バックアップ（動作中の場合）はプライマリへの再接続を試みます。

3.4 自動フェイルオーバーの阻止

何らかの状況で、ミラーで自動フェイルオーバーが発生しないようにする場合、最善の方法として、単一のフェイルオーバー・メンバを 1 つ以上の DR 非同期で構成します（“[非同期ミラー・メンバ](#)”を参照）。DR 非同期は自動的に引き継ぐことはありませんが、必要に応じてプライマリも含めたフェイルオーバー・メンバに簡単に昇格できます（“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”を参照）。

メンテナンス処理中にバックアップに対して自動フェイルオーバーを一時的に阻止するには、バックアップを DR 非同期に一時的に降格するか、nofailover オプションを使用します。両方とも“[計画的停止の手順](#)”で説明されています。ここにはミラー処理を中断することなくフェイルオーバー・メンバにメンテナンスを実行する手順が示されています。

自動フェイルオーバー・プロセスのさまざまなポイントでアプリケーションの操作が必要な場合は、“[ZMIRROR ルーチンの使用法](#)”を参照してください。

3.5 ミラーリング通信

このセクションでは、ミラー・メンバ間の通信について詳細に説明します。内容は以下のとおりです。

- ・ [ネットワーク構成に関する考慮事項](#)

- ・ ネットワーク遅延に関する考慮事項
- ・ ミラー・トラフィックの圧縮
- ・ ミラー・メンバのネットワーク・アドレス

3.5.1 ネットワーク構成に関する考慮事項

2 つのフェイルオーバー・メンバ間でネットワークを構成する場合は、以下の一般的なネットワーク構成項目を考慮する必要があります。

- ・ 信頼性 – 信頼性を最大限に高めるために、2 つのフェイルオーバー・メンバ間のミラー通信に対して、分離された（プライベート）ネットワークを構成する必要があります（“[ミラーリング・アーキテクチャとネットワーク構成のサンプル](#)”を参照）。さらに、このネットワークは冗長性を持たせて（ポートをフェイルオーバーボンディングした複数の NIC、複数の冗長スイッチなど）構成する必要があります。
- ・ 帯域幅 – アプリケーションによって生成されるジャーナル・データ量を転送するのに十分な帯域幅を使用できる必要があります。
- ・ 遅延 – フェイルオーバー・メンバ間のネットワーク遅延は、アプリケーションのパフォーマンスにおいて重要な要素の 1 つです。詳細は、“[ネットワーク遅延に関する考慮事項](#)”を参照してください。

[ミラー同期](#)は、プライマリ・フェイルオーバー・メンバでのジャーナル書き込みサイクルの一部として行われます。ジャーナル書き込みサイクルおよびミラー同期プロセスをできる限り早く完了できることが重要です。このプロセスにおける遅延はすべて、パフォーマンスの低下につながる可能性があります。

注釈 VIP 使用時の重要なネットワーキング要件や考慮事項の詳細は、“[ミラー仮想 IP \(VIP\) の構成](#)”を参照してください。

3.5.2 ネットワーク遅延に関する考慮事項

フェイルオーバー・メンバ間のネットワーク遅延に厳格な上限はありません。遅延が長くなることによる影響は、アプリケーションごとに異なります。フェイルオーバー・メンバ間を往復する時間がディスク書き込みサービス時間と似通っている場合、予期される影響はありません。ただし、データが保存されるのをアプリケーションが待機する必要があるとき（ジャーナル同期と呼ばれることがあります）、往復する時間が問題となる場合があります。ミラーリングされていない環境では、データの保存の待機には、ジャーナル・データの同期ディスク書き込みも含まれます。アクティブなバックアップが含まれるミラーリング環境では、フェイルオーバー・メンバ間のネットワーク往復時間も含まれます。データが保存されるのを多くのアプリケーションは待機することはないですが、それら以外のアプリケーションは頻繁に待機します。

アプリケーションの待機メカニズムに、以下の動作が含まれる場合があります。

- ・ 同期コミット・モード（既定以外）でのトランザクションのコミット。
- ・ `%SYS.Journal.System` の `Sync()` メソッド。
- ・ アプリケーション・サーバ上で実行されているアプリケーションからの一般的な要求に応答する前に、分散キャッシュ・クラスタ・データ・サーバが保存を待機する動作（ロックや `$increment` などのアプリケーション同期化アクションの一部として待機）。
- ・ ビジネス・サービスの `SyncCommit` 機能（既定）

往復時間が比較的長い場合でも、アプリケーションの応答時間またはスループットに悪影響を及ぼすかどうかは、アプリケーション内で前述の動作が発生する頻度、およびアプリケーションがそのような動作を順番に処理するのか、それとも並行して処理するのかによって異なります。

ミラー・メンバ間のネットワーク遅延が問題になる場合は、オペレーティング・システムの `SO_SNDBUF` と `SO_RCVBUF` の最大値を管理する TCP パラメータを微調整することで、遅延を短縮できる可能性があります。これにより、プライマリと

バックアップ/非同期は、それぞれに適切なサイズの送信バッファと受信バッファを設置できるようになります (最大 16 MB)。必要になるバッファ・サイズは、往復時間に必要なピーク時の帯域幅 (“受信ジャーナルの転送速度” を参照) と、プロトコルのオーバーヘッドおよび将来の増加を乗算したものを約 2 倍した値になります。例えば、以下の条件が当てはまるとします。

- ・ プライマリ・ミラー・サイトと DR サイトとの間のトラフィックは、毎秒 60 MB のジャーナル・データ (ピーク時)。
- ・ 帯域幅を減らすために使用する圧縮は、ジャーナル・レート約 33% が求められている。
- ・ 往復時間は、50 ミリ秒 (一般に、1000 マイルの距離にかかる時間)。

この場合、最小のバッファサイズは、 $60 \text{ MB} \times 0.05 \times .33 \times 2 = 2 \text{ MB}$ になります。バッファ・サイズをできるだけ小さく抑える大きな理由はないため、この状況では、より大きな最小値を懸念なく試してみることができます。

3.5.3 ミラー・トラフィックの圧縮

ミラーの作成時または編集時 (それぞれ “ミラーを作成し、第 1 のフェイルオーバー・メンバを構成する” または “フェイルオーバー・メンバの編集または削除” を参照) には、以下に示すように、プライマリからバックアップに、およびプライマリから非同期メンバに転送されるミラー・トラフィックに対して圧縮モードを選択できます。

- ・ **システム選択** – ほとんどの環境に最適な圧縮方法を使用します。バックアップ・メンバへの転送の場合は、高帯域幅で低遅延の接続を前提として、応答時間を最適化することを意味します。つまり、ジャーナル・データは転送前に圧縮されるということです (圧縮により、フェイルオーバー・メンバの同期に必要な時間が短縮される場合)。非同期メンバへの転送の場合は、ネットワーク使用率を最適化することを意味します。[システム選択] は、フェイルオーバー・メンバと非同期の両方に対する既定の設定です。

現時点では、バックアップへの転送については、ミラーが TLS を必要とする場合にのみ、この設定で LZ4 圧縮が使用されます (“TLS セキュリティを使用したミラー通信の保護” を参照)。非同期への転送については、Zstd 圧縮が常に使用されます。この動作は、ネットワーク環境の分析メカニズムの向上と、圧縮動作の最適化に応じて、そのうちに変更される可能性があります。

- ・ **圧縮なし** – ミラー・トラフィックを圧縮しません。
- ・ **圧縮** – 常にミラー・トラフィックを圧縮します。この設定を選択する場合、3 つの圧縮タイプ zlib (既定)、Zstd、LZ4 のうちの 1 つを選択する必要があります。

[システム選択] モード、または [圧縮] モードでのユーザの選択のいずれかにより Zstd または LZ4 圧縮を使用している場合に、受信システムでそのタイプがサポートされていないと、zlib 圧縮が代わりに使用されます。

ほとんどの量のデータベース更新が圧縮済みまたは暗号化済みのデータで構成されている場合は、[圧縮なし] を選択することが望ましいです。この状況では、圧縮に期待できる全体的な効果はほとんどありません。この場合、圧縮にかかる CPU 時間が無駄になる可能性があります。この例として、圧縮されたイメージ、圧縮されたメディア、またはデータベースに設定される前に (InterSystems IRIS のデータ要素暗号化などの暗号化手法を使用して) 暗号化されたデータなどが挙げられます。InterSystems IRIS データベース暗号化またはジャーナル暗号化の使用は、圧縮を選択する際の要因にはなりません。

圧縮と TLS 暗号化は、どちらもスループットと遅延に影響する計算上のオーバーヘッドの原因になります。どちらが原因になるオーバーヘッドも同様のものですが、TLS 暗号化が使用されている場合は、追加の圧縮で暗号化する必要のあるデータの量を削減することで、実際にはオーバーヘッドを減らして、パフォーマンスを向上できます。詳細は、オペレーティング・システム、CPU アーキテクチャ、およびアプリケーション・データの圧縮率により変化します。具体的には、以下のようになります。

- ・ 圧縮や TLS 暗号化を使用すると、データの圧縮に必要な計算時間のために、転送速度が制限されることがあります。最大転送速度は、最大圧縮速度に制限されます。ほとんどの構成では、圧縮と TLS 暗号化によって決まる最大転送速度は、実際にミラーリングに必要な最大スループットよりも、はるかに高速になります。一例を挙げると、このドキュメントの執筆時点では、一般的なシステムにおける圧縮と暗号化に対する計算速度は、毎秒 100 MB

程度であり、大規模なエンタープライズ・アプリケーションに対するピーク時のジャーナル作成速度よりも数倍も高速です。

- ・ 圧縮や TLS 暗号化の使用により、ネットワーク遅延に加算される“計算上の遅延”が発生します（“[ネットワーク遅延に関する考慮事項](#)”を参照）。ほとんどのアプリケーションで、これは無視できます。圧縮や TLS 暗号化を有効にすることで達成可能なスループットよりも、さらに高いスループットが要求される構成の場合は、圧縮や TLS 暗号化を無効にして（無効にしても、認証には SSL が使用できます）、圧縮なしのピーク時の転送に十分に対応できる帯域幅を用意する必要があります。

3.5.4 ミラー・メンバのネットワーク・アドレス

ミラー・メンバは、お互いに通信するために、いくつかのネットワーク・アドレスを使用します。これらについては、このセクションで説明しますが、“[ミラーリング・アーキテクチャとネットワーク構成のサンプル](#)”でも説明しています。ここで説明されているミラー・アドレスの一部またはすべてに、同一のネットワーク・アドレスが使用されている可能性があることに留意してください。

- ・ ミラー・プライベート・アドレス

1 つの InterSystems IRIS インスタンスがプライマリ・フェイルオーバー・メンバとして実行されている場合、その他のミラー・メンバはそれぞれミラー・プライベート・アドレスを使用して、自身のミラー・データ・チャンネルを確立します。これは、プライマリからのジャーナル・データの受信に使用するチャンネルであり、最も頻繁に使用されるミラー通信チャンネルです。バックアップになろうとする第 2 のフェイルオーバー・メンバは、このアドレスに接続する必要があります。これは、[フェイルオーバー・メンバに昇格された DR](#) 非同期にも当てはまります。昇格された DR が他方のフェイルオーバー・メンバのプライベート・アドレスにアクセスできなくても、他方のフェイルオーバー・メンバがダウンしたときにプライマリになることは可能ですが、バックアップになることはできません。

プライマリは、非同期メンバの監視のために、ミラー・プライベート・アドレスも使用することがあります。

非同期メンバは、プライマリのミラー・プライベート・アドレスに接続しようしますが、そのミラー・プライベート・アドレスでプライマリに到達できないと、プライマリのスーパーサーバ・アドレスにフォールバックします。これにより、また ISCAgent がジャーナル・データを他のミラー・メンバに送信できることにより、場合によっては、ジャーナル・データはミラー・プライベート・ネットワーク以外のネットワーク経由で転送されます。

注釈 管理ポータルを使用して非同期メンバをミラーに追加する際に（“[非同期ミラー・メンバを構成する](#)”を参照）、[\[非同期メンバのアドレス\]](#)を入力します。このプロンプトで指定するアドレスは、非同期メンバのミラー・プライベート・アドレスおよびスーパーサーバ・アドレスになります。これらを異なったものにする場合、ミラーに追加した後に、プライマリで[非同期のアドレスを更新](#)できます。

- ・ スーパーサーバ・アドレス/ポート

このアドレスを使用して、外部ミラー認識システムをプライマリに接続できます。現時点では、そのような外部システムのみが、ミラーリングされた分散キャッシュ・クラスタ内のアプリケーション・サーバになります（“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)”を参照）。ただし、将来、これは別の接続にも拡張される可能性があります。その他のミラー・メンバも特定の制御目的および監視目的のためにメンバのスーパーサーバ・アドレスに接続することがあります。例えば、プライマリは、非同期メンバの監視に、このアドレスを使用することがあります。プライマリのミラー・プライベート・アドレスにアクセスできない場合、非同期メンバはこのアドレスを使用してプライマリへのデータ・チャンネルを確立しようとします。これは、ジャーナル・データがこのネットワーク経由で転送可能であることを意味します。

- ・ エージェント・アドレス/ポート

このメンバのエージェントに通信しようとする場合、他のメンバはまずこのアドレスを試用します。重要なエージェント関数（フェイルオーバーの決定に使用する関数など）は、このアドレスにアクセスできないと、ミラー・プライベート・アドレスおよびスーパーサーバ・アドレス（異なる場合）を使用して再試行します。エージェントはジャーナル・データを他のメンバに送信することができるので、ジャーナル・データはこのネットワーク経由で転送できます。

- ・ 仮想 IP (VIP) アドレス

“[ミラー仮想 IP \(VIP\) の計画](#)” で説明しているように仮想 IP (VIP) アドレスを使用する場合、プライマリ・フェイルオーバー・メンバを[作成](#)または[編集](#)するときに VIP アドレスを入力する必要があります。プライマリは、プライマリになる処理の一環として自身をこのアドレスに動的に登録します。2 つのフェイルオーバー・メンバは、フェイルオーバーの発生時にバックアップが VIP を取得できるように VIP に関連付けられているネットワークの同じサブネットに配置する必要があります。管理者は通常、VIP アドレスに DNS サーバの DNS 名を指定します。このアドレスをミラーリング構成の他の場所で使用してはいけません (アプリケーション・サーバの接続先のアドレスとしても使用してはいけません。ECP には、スーパーサーバ・アドレスを使用してプライマリを検索する独自のメカニズムがあります)。

- ・ アービター・アドレス/ポート (送信)

フェイルオーバー・メンバがアービターへの接続に使用するアドレスです。このアドレスは、プライマリ・フェイルオーバー・メンバを[作成](#)または[編集](#)するときに構成します。アービターのネットワーク上の場所に関する重要な情報は、“[ミラーの可用性を最適化するためのアービターの配置](#)” を参照してください。

ここで説明したアドレス間のミラー通信に対する TLS の構成は任意ですが、強くお勧めします。機密データがフェイルオーバー・メンバ間で受け渡しされるためです。TLS は、ISCAgent に対する認証を提供します。ISCAgent は、ジャーナル・ファイルへのリモート・アクセスを提供し、システムを強制的に停止したり、その仮想 IP アドレスを操作できます。インスタンスでジャーナルの暗号化が有効になっていて、そのインスタンスをミラーのプライマリ・フェイルオーバー・メンバに設定する場合、TLS を使用するようにミラーを構成する必要があります。詳細は、“[TLS セキュリティを使用したミラー通信の保護](#)” を参照してください。

3.6 ミラーリング・アーキテクチャとネットワーク構成のサンプル

このセクションでは、ミラーリング・アーキテクチャと構成のサンプルをいくつか示して説明します。

- ・ [単一のデータ・センタ、コンピュータ・ルーム、またはキャンパス内のミラーリング構成](#)
- ・ [2 つのデータ・センタと地理的に分離された災害復旧に対するミラーリング構成](#)

一部の図では、[災害復旧 \(DR\) 非同期メンバ](#)および[レポート非同期メンバ](#)がさまざまな位置に描かれています。いずれかまたは両方を省略することも、それぞれを複数許可することもでき、また概して、異なる図で描かれている位置を組み合わせることもできます。

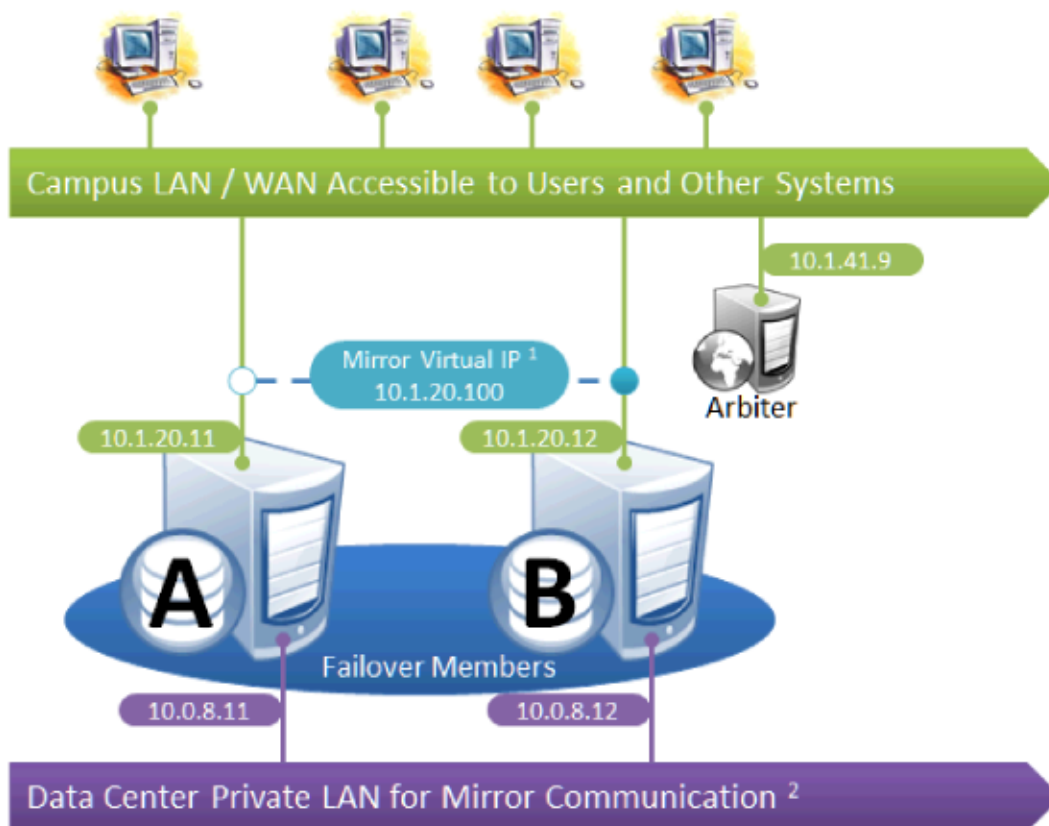
図で示すことを目的として、組織の内部ネットワークのサンプル IPv4 アドレスが示されています。サブネットは 24 ビットで指定される (つまり、CIDR 表記 a.b.c.d/24 または [ネットマスク 255.255.255.0](#)) ことが前提となっているため、同じサブネットに対して記載されているアドレスは、4 番目のドット区切り部分のみが異なります。

ミラー構成の IP アドレスの代わりに、それに相当する DNS 名も指定されている場合があります。ただし、ミラー仮想 IP (VIP) アドレスの場合は IP アドレスである必要があるため、この限りではありません。

3.6.1 単一のデータ・センタ、コンピュータ・ルーム、またはキャンパス内のミラーリング構成

以下の図は、1 つのデータ・センタ、コンピュータ・ルーム、またはキャンパス内で一般的な、さまざまなミラーリング構成を示しています。それぞれの図は、該当するネットワーク・トポロジ、およびミラー構成に指定されている[ネットワーク・アドレス](#)とのリレーションシップを示しています。バリエーションが示されており、特にミラー・メンバが同一キャンパス内の複数の場所に配置されている場合に適用できることがあります。

3.6.1.1 単純なフェイルオーバーのペア



これは、最も単純なミラー構成です。複数のフェイルオーバー・メンバはプライベート・ネットワーク経由で互いに通信し、それらへの外部接続はパブリック・ネットワークを通じて確立されます。オプションでミラー仮想 IP (VIP) が使用されます。アービターは外部ネットワーク上にありますが(“ミラーの可用性を最適化するためのアービターの配置”で推奨されているとおり)、アービターへの接続を開始するのは常にフェイルオーバー・メンバであることから、VIP はアービターへの接続には関連していません。

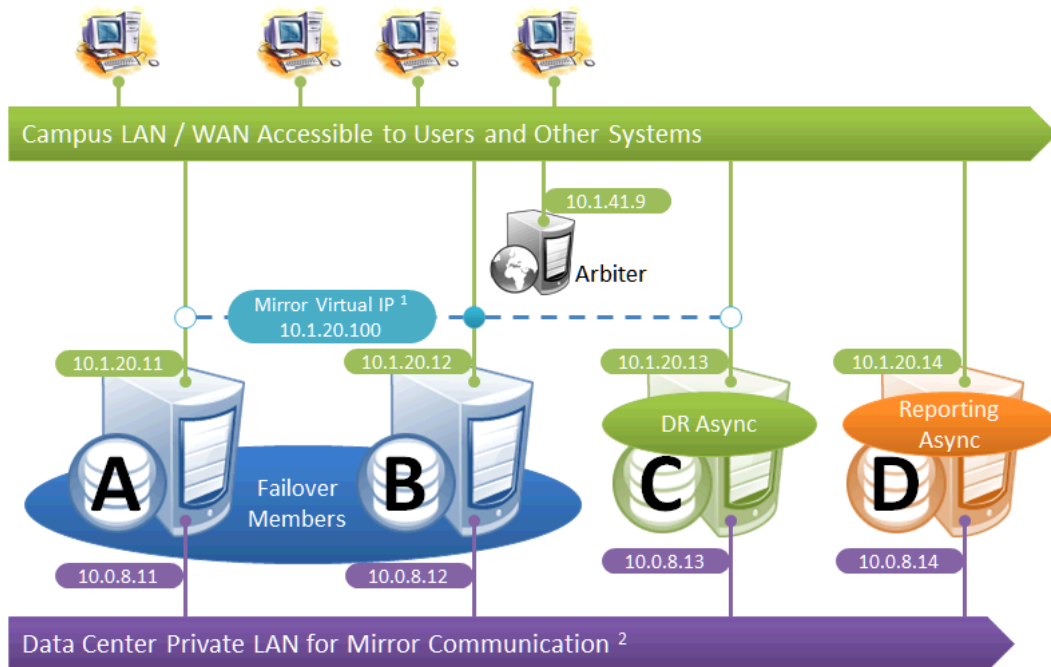
この構成では、以下の IP アドレスが使用されています。

Virtual IP Address	10.1.20.100	
Arbiter Address	10.1.41.9	
Member-Specific Mirror IP Addresses for Member	A	B
SuperServer Address	10.1.20.11	10.1.20.12
Mirror Private Address	10.0.8.11	10.0.8.12
Agent Address	10.1.20.11	10.1.20.12

メモ：

1. VIP では、両方のフェイルオーバー・メンバが同一のサブネットに配置されている必要があります。
2. ミラーリングに必須というわけではありませんが、ネットワーク使用率を最適に制御できるように、ここに示しているミラー通信の個別プライベート LAN をお勧めします。このような LAN を使用しない場合、緑の背景で示しているアドレスを使用するようにミラー構成でミラー・プライベート・アドレスを変更する必要があります。ここに示されているように、メンバがこのネットワークの同一サブネット上にあることをミラー・プライベート・アドレスは示唆していますが、これは必須ではありません。

3.6.1.2 DR およびレポートが非同期に均一に接続されたフェイルオーバーのペア



この構成では、DR 非同期で機能上の柔軟性が最大限にでき、災害復旧機能を実現できることに加えて、メンテナンスや修理のために停止しているフェイルオーバー・メンバに代わるために昇格できます。昇格された DR はバックアップまたはプライマリとして完全に機能することができ、VIP に追加されます。フェイルオーバー・メンバと DR は、VIP の同一パブリック接続サブネット上にあります。これらのプライベート・ネットワーク・アドレスが使用される場合、これらはお互いにアクセス可能です (同じサブネットではない場合は、ルーティングによって)。ネットワーク・トポロジおよびネットワーク遅延によって、DR と 2 つのフェイルオーバー・メンバとの間に可能な物理的な分離に、制約が加わる場合があります。

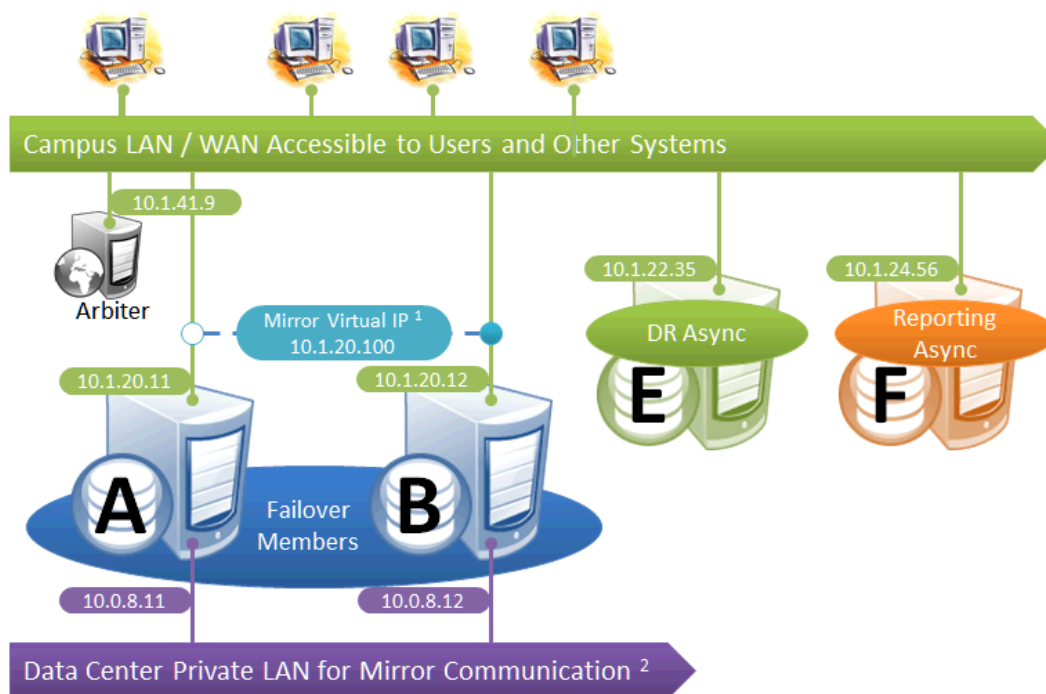
この構成では、以下の IP アドレスが使用されています。

Virtual IP Address	10.1.20.100			
Arbiter Address	10.1.41.9			
Member-Specific Mirror IP Addresses for Member	A	B	C	D
SuperServer Address	10.1.20.11	10.1.20.12	10.1.20.13	10.1.20.14
Mirror Private Address	10.0.8.11	10.0.8.12	10.0.8.13	10.1.20.14 ³
Agent Address	10.1.20.11	10.1.20.12	10.1.20.13	10.1.20.14

メモ：

- VIP を保持または取得できるメンバはすべて同じサブネット上に配置する必要があります。
- ここに示しているミラー通信用の個別プライベート LAN はミラーリングに必須ではありませんが、ネットワーク使用率の最適制御でお勧めします。このような LAN を使用しない場合、緑で示しているアドレスを使用するようにミラー構成でミラー・プライベート・アドレスを変更する必要があります。示されているミラー・プライベート・アドレスは、メンバがこのネットワークの同一サブネット上にあることを示唆していますが、これは必須ではありません。
- レポート・メンバはプライマリになることはできないため、ミラー・プライベート・ネットワークでは送信接続のみを確立します。このため、そのアドレスを、ミラー構成で個別に指定する必要はありません。

3.6.1.3 DR およびレポートがキャンパス上の任意の場所にあるフェイルオーバーのペア



この構成では、非同期メンバの位置およびそれらに接続しているネットワークにおいて最大限の柔軟性を実現できます。この構成において DR が VIP サブネット上にあることは前提となっていないため、災害復旧時の DR へのユーザ接続のリダイレクトには、他の代替方法を使用する必要があります。例えば、VIP ではなく DR 非同期の IP をポイントするように DNS 名を手動で更新することや、“フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト”で説明したメカニズムのいずれかを構成できます。さらに、DR メンバがミラー・プライベート・ネットワーク(使用されている場合)に接続されていることは前提となっていないため、稼働中のフェイルオーバー・メンバがない場合に限り、プライマリにするためのみ昇格できます。

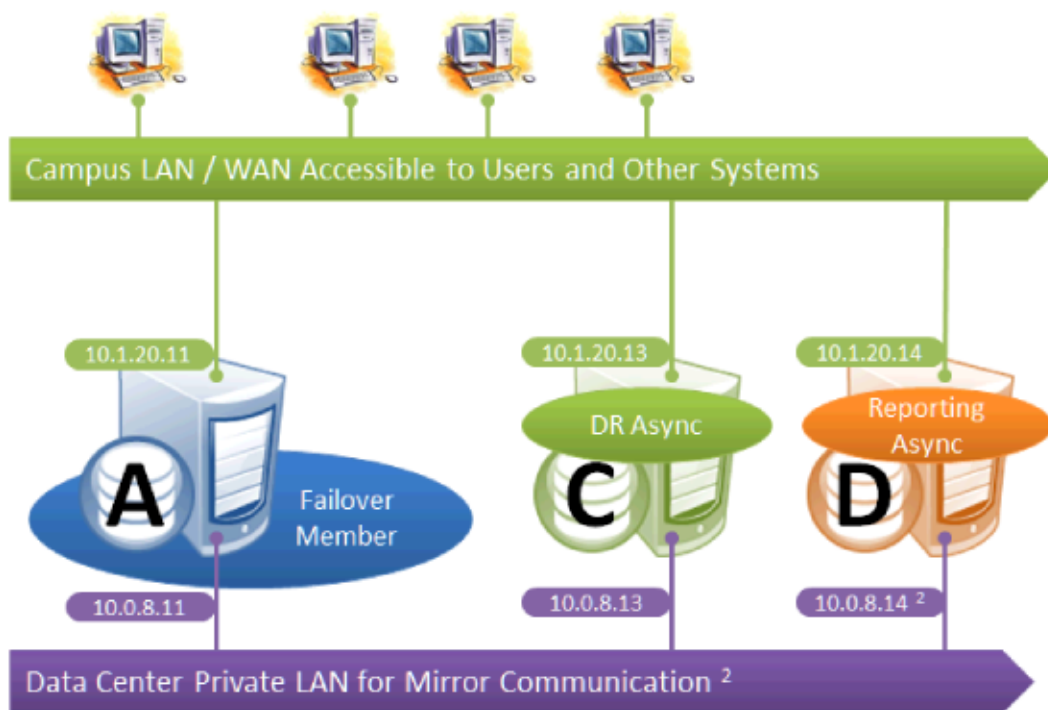
この構成では、以下の IP アドレスが使用されています。

Virtual IP Address	10.1.20.100			
Arbiter Address	10.1.41.9			
Member-Specific Mirror IP Addresses for Member	A	B	E	F
SuperServer Address	10.1.20.11	10.1.20.12	10.1.22.35	10.1.24.56
Mirror Private Address	10.0.8.11	10.0.8.12	10.1.22.35	10.1.24.56
Agent Address	10.1.20.11	10.1.20.12	10.1.22.35	10.1.24.56

メモ：

1. 仮想 IP を取得するメンバはすべて、同じサブネット上に配置する必要があります。
2. ここでは、ミラー通信の個別プライベート LAN が示されていますが、必須ではありません。このような LAN を使用しない場合、緑で示しているアドレスを使用するようにミラー構成でミラー・プライベート・アドレスを変更する必要があります。示されているミラー・プライベート・アドレスは、フェイルオーバー・メンバがこのネットワークの同一サブネット上にあることを示唆していますが、これは必須ではありません。

3.6.1.4 災害復旧およびレポート専用のミラーリング



この構成では、DR とレポートのいずれかまたは両方の機能のみを提供するミラーリングを使用します。“[高可用性を実現するためのフェイルオーバー方法](#)”で説明されているように、OS フェイルオーバー・クラスターリング、仮想 HA、またはその他のインフラストラクチャレベルのオプションを使用すると、単一のフェイルオーバー・メンバの場合に高可用性が実現されます。この構成で自動フェイルオーバーにミラーリングは使用されないため、VIP は描かれていません。必要に応じて、障害復旧時に使用するために VIP を構成できますが、これには DR メンバをフェイルオーバー・メンバと同じサブネットに配置する必要があります。または、災害復旧時の DR へのユーザ接続のリダイレクトに、“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)”で説明したような代替のテクノロジーまたは手順を使用する必要があります。

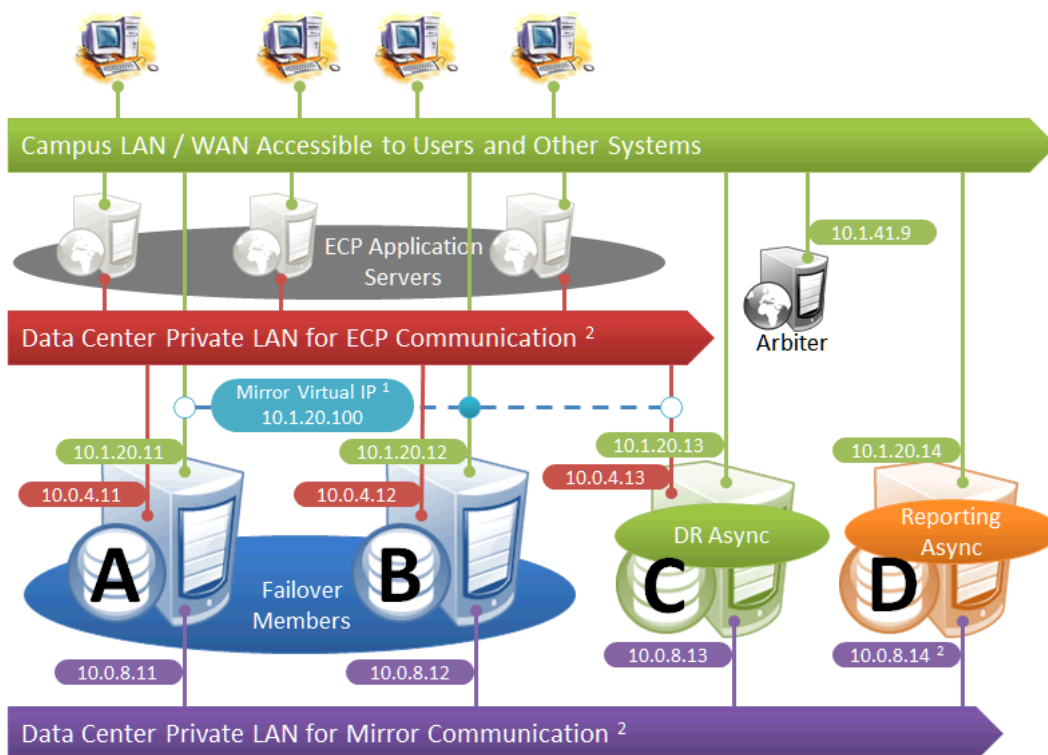
この構成では、以下の IP アドレスが使用されています。

Virtual IP Address	Not Depicted		
Arbiter Address	Not Used		
Member-Specific Mirror IP Addresses for Member	A	C	D
SuperServer Address	10.1.20.11	10.1.20.13	10.1.20.14
Mirror Private Address	10.0.8.11	10.0.8.13	10.1.20.14 ²
Agent Address	10.1.20.11	10.1.20.13	10.1.20.14

メモ：

- ここでは、ミラー通信の個別プライベート LAN が示されていますが、必須ではありません。このような LAN を使用しない場合、緑で示しているアドレスを使用するようにミラー構成でミラー・プライベート・アドレスを変更する必要があります。示されているミラー・プライベート・アドレスは、フェイルオーバー・メンバがこのネットワークの同一サブネット上にあることを示唆していますが、これは必須ではありません。
- レポート・メンバはプライマリになることはできないため、ミラー・プライベート・ネットワークでは送信接続のみを確立します。このため、そのアドレスを、ミラー構成で個別に指定する必要はありません。

3.6.1.5 分散キャッシュ・クラスタでのミラーリング



この図は、ミラーリング環境に追加されたアプリケーション・サーバを示しています。複雑性は高まりますが、アプリケーション・サーバ層によって水平スケーラビリティが可能となり、データベース・サーバのフェイルオーバーを通してユーザ・セッションが保持されます。分散キャッシュおよび分散キャッシュ・クラスタの詳細は、“[分散キャッシュによるユーザ数に応じた水平方向の拡張](#)”を参照してください。

この構成では、以下の IP アドレスが使用されています。

Virtual IP Address	10.1.20.100 ¹			
Arbiter Address	10.1.41.9			
Member-Specific Mirror IP Addresses for Member	A	B	C	D
SuperServer Address	10.0.4.11	10.0.4.12	10.0.4.13	10.1.20.14
Mirror Private Address	10.0.8.11	10.0.8.12	10.0.8.13	10.1.20.14 ³
Agent Address	10.1.20.11	10.1.20.12	10.1.20.13	10.1.20.14

メモ：

- アプリケーション・サーバはVIPを使用せず、任意のフェイルオーバー・メンバまたは昇格されてプライマリになったDRメンバに接続するため、VIPはプライマリへのユーザの直接接続がある場合は、そのみに使用されます。VIPでは、両方のフェイルオーバー・メンバが同一のサブネットに配置されている必要があります。DRメンバが昇格時にVIPを取得するには、同じサブネットに配置する必要もあります。同じサブネットに配置していない場合、“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)”を参照してください。
- ここに示されているECP通信とミラー通信のプライベートLANは両方とも必須ではありませんが、ネットワーク使用率およびECPデータのプライバシーを最適に制御するために使用することをお勧めします。いずれかのネットワークを別のネットワークに折りたたむことによって、ネットワーク数を少なくした構成が可能です。示されているプライベート・アドレスは、メンバがこれらのネットワークの同じサブネット上にあることを示唆していますが、唯一の要件は、これらのアドレスがお互いにルーティングできることです。

ネットワーク・レイアウトについて検討する場合は、すべての非同期メンバが、プライマリのミラー・プライベート・アドレスかスーパーサーバ・アドレスでプライマリに接続できる必要があることを忘れないでください。このように、示されている構成では、緑のユーザ・ネットワークにのみアクセスできる非同期メンバは機能しません。

3. レポート・メンバはプライマリになることはできないため、ミラー・プライベート・ネットワークでは送信接続のみを確立します。このため、そのアドレスを、ミラー構成で個別に指定する必要はありません。

3.6.2 2つのデータ・センタと地理的に分離された災害復旧に対するミラーリング構成

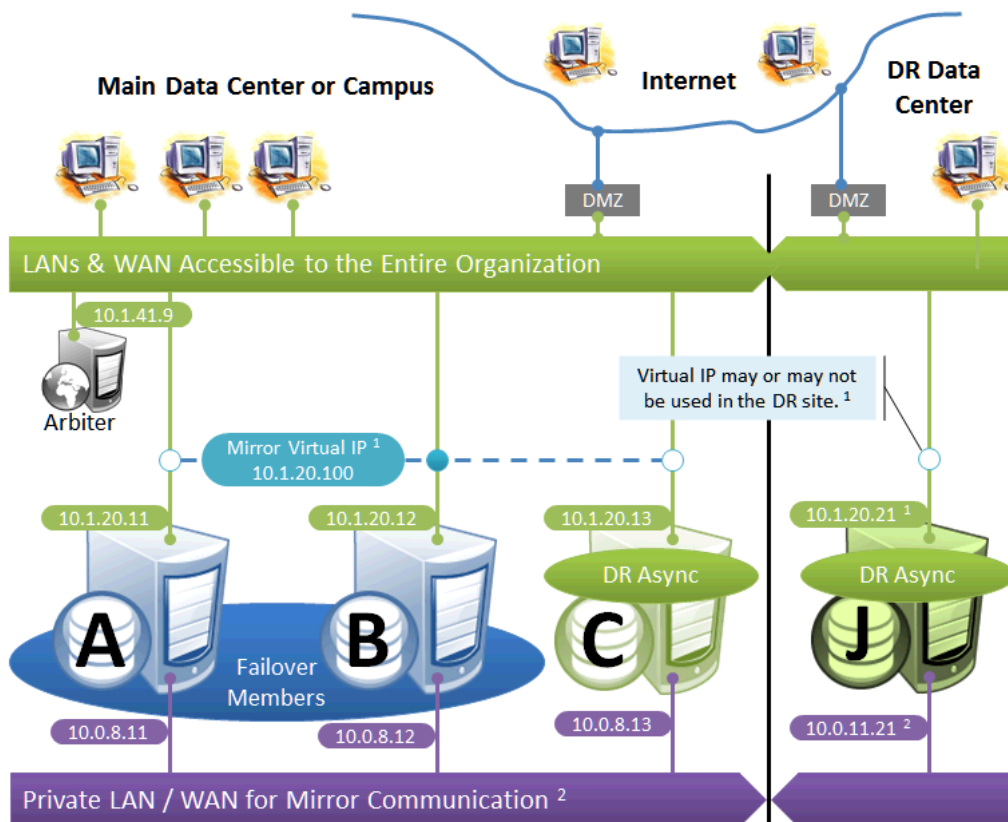
以下の図は、データ・センタ、キャンパス、または地域に影響を与える災害から復旧するために地理的な分離を利用した HA および DR 構成を示しています。図を単純にするためにこれらの図からレポート・メンバは省略されていますが、単一キャンパス構成で示したとおりに、いずれかの場所に追加できます。

以下の構成にはすべて、他方の場所のメンバがプライマリになったときに、そのプライマリに接続をリダイレクトする方法が必要です。地理的に分離されている場所の場合、これら 2 つの場所にわたってサブネットを拡張することが必要であるため、VIP を構成することは難しいか、または不可能な場合があります。構成したとしても、以下の段落で説明しているように、十分ではない可能性があります。“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)”で説明したような代替のテクノロジー、ハードウェア、または手順では、接続をリダイレクトする他の方法を使用できます。拡張されたサブネットを利用するかどうかにかかわらず、VIP は単一のデータ・センタ内の 2 つのメンバ間の自動フェイルオーバーにはきわめて有用であるため、以下の図では VIP の使用が描かれています。

VIP 用に拡張されたサブネットは通常、内部イントラネット・アプリケーションに役立ちます。これにより、緑で示されている LAN/WAN への接続（つまり VPN アクセス）があるユーザおよびシステムは、VIP 経由でいずれかの場所にあるプライマリにアクセスできます。

一方、インターネットに接続するアプリケーションの場合、VIP 用に拡張されたサブネットが、災害時に接続のソリューションを提供することはありません。メイン・データ・センタの DMZ が、内部ミラー VIP のプロキシとして、アプリケーションのインターネット接続 IP アドレスと DNS 名のいずれかまたは両方を提供します。災害時には、これらを他方のデータ・センタに外部的に転送する必要がある場合があります。ソリューションには、高度な外部ルーティングまたは“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)”で説明したいずれかの手法が使用されます。これらの手法のいずれでも、拡張されたサブネットの必要性は除去されます。

3.6.2.1 ローカル DR と地理的に分離された DR を使用したフェイルオーバーのペア



ローカル DR 非同期は、一方または両方のフェイルオーバー・メンバに影響を与えるイベントの不測の事態に備えます。ローカル DR を昇格して、メンテナンスや修理のために停止しているいずれかのフェイルオーバー・メンバの代わりとすることも、両方のフェイルオーバー・メンバに影響を及ぼす災害からの復旧に使用することもできます。地理的に分離されている DR は、メイン・データ・センタまたはキャンパス全体に影響を及ぼす災害からの復旧に使用されます。

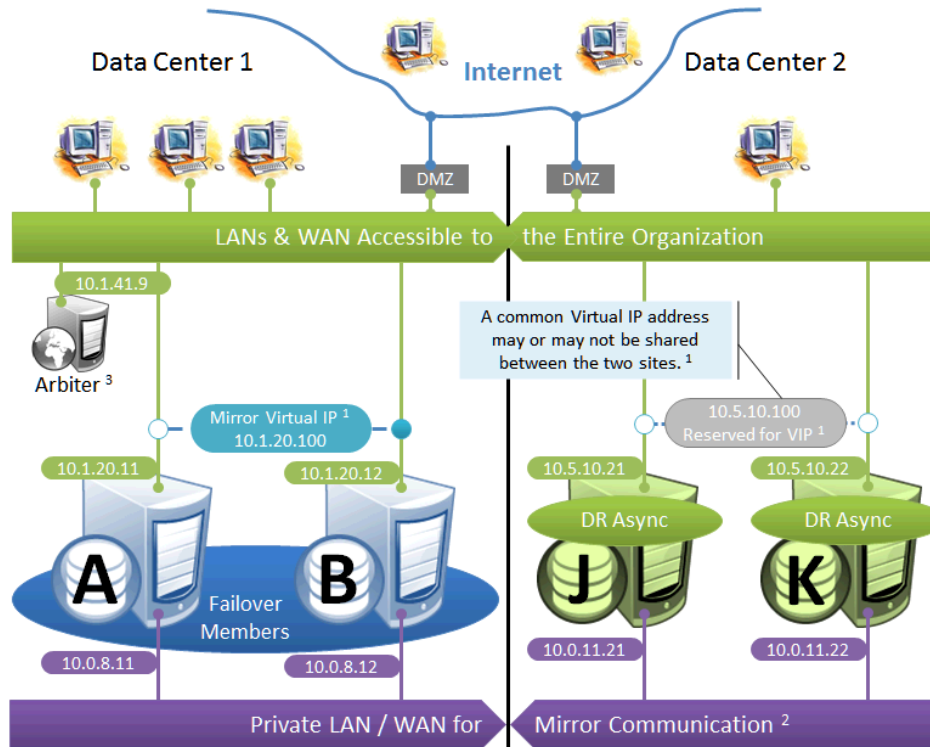
この構成では、以下の IP アドレスが使用されています。

Virtual IP Address	10.1.20.100			
Arbiter Address	10.1.41.9			
Member-Specific Mirror IP Addresses for Member	A	B	C	J
SuperServer Address	10.1.20.11	10.1.20.12	10.1.20.13	10.1.20.21
Mirror Private Address	10.0.8.11	10.0.8.12	10.0.8.13	10.0.11.21 ²
Agent Address	10.1.20.11	10.1.20.12	10.1.20.13	10.1.20.21

メモ：

1. VIP の前述の説明を参照してください。
2. 可能な場合、ミラー・プライベート・ネットワーク (使用されている場合) に、データ・センタ相互接続 (WAN) を使用して DR データ・センタからアクセスできるようにすると、メンバ J にさらに機能上の柔軟性がもたらされます。これには、サブネットの拡張は必要ではなく、対象のネットワーク上のトラフィックをデータ・センタ間でルーティングすることだけが必要です。この構成では、J を昇格した場合、メイン・データ・センタのプライマリにバックアップとして接続できます。DR にミラー・プライベート・ネットワークへのアクセス権がない場合、プライマリとして機能させるためだけに昇格でき、稼働中のフェイルオーバー・メンバがない場合にのみこれは可能です。ここで述べた柔軟性は主として、VIP が拡張されていて、アプリケーションがデータ・センタ間の遅延に実質上は影響を受けない構成において有用です。

3.6.2.2 地理的に分離されていて、完全に冗長な DR 環境を使用したフェイルオーバーのペア



データ・センタ 1 に影響を及ぼす災害の発生時には、データ・センタ 2 内の 2 つの DR メンバが昇格され、完全に冗長な代替 HA 環境が提供されます。この構成では、以下の IP アドレスが使用されています。

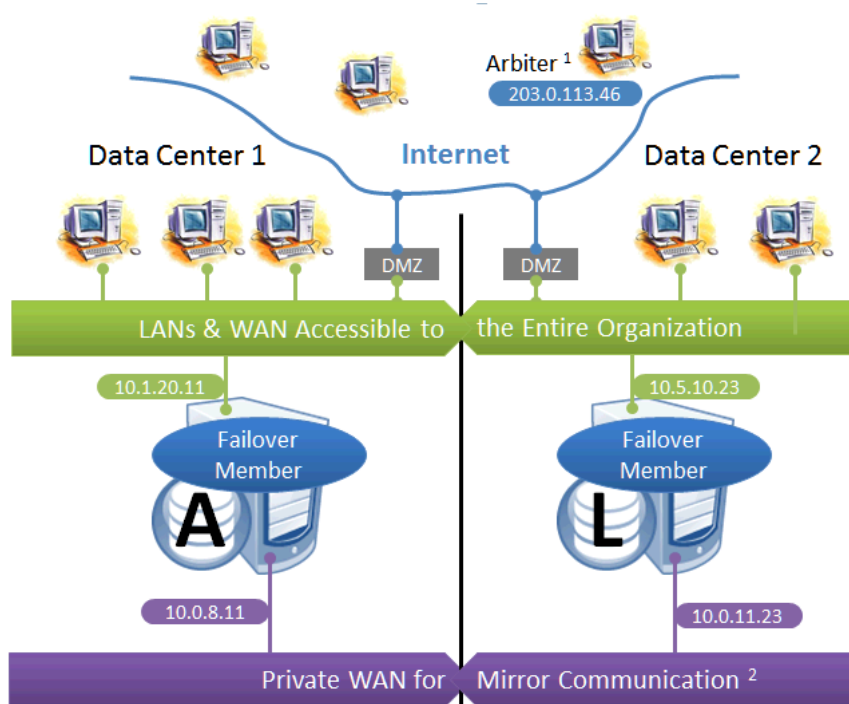
Virtual IP Address	10.1.20.100 ¹			
Arbiter Address	10.1.41.9 ³			
Member-Specific Mirror IP Addresses for Member	A	B	J	K
SuperServer Address	10.1.20.11	10.1.20.12	10.5.10.21	10.5.10.22
Mirror Private Address	10.0.8.11	10.0.8.12	10.0.11.21	10.0.11.22
Agent Address	10.1.20.11	10.1.20.12	10.5.10.21	10.5.10.22

メモ：

- VIP の前述の説明を参照してください。この図では、拡張されたサブネットは前提となっておりません。代わりに、データ・センタ 2 への移行時に、そのデータ・センタ内での後続の自動フェイルオーバーに別の VIP を使用するようミラーが再構成されます。外部のテクノロジー、ハードウェア、または手順は、“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)” で説明したように、この新しい VIP アドレスへの接続のリダイレクトに使用されます。
- 可能な場合、両方のデータ・センタから、データ・センタ相互接続 (WAN) を使用してミラー・プライベート・ネットワーク (使用されている場合) にアクセスできるようにすると、機能上の柔軟性がもたらされます。これには、サブネットの拡張は必要ではなく、対象のネットワーク上のトラフィックをデータ・センタ間でルーティングすることだけが必要です。この構成では、一方のデータ・センタ内の昇格された DR メンバが、バックアップとして他方のデータ・センタ内のプライマリに接続できます。これは主として、VIP が拡張されていて、アプリケーションがデータ・センタ間の遅延に実質上は影響を受けない構成において有効です。(DR にミラー・プライベート・ネットワークへのアクセス権がない場合、プライマリとして機能させるためだけに昇格でき、稼働中のフェイルオーバー・メンバがない場合にのみこれは可能です。)
- データ・センタ 1 が完全にオフラインになり、メンバ J および K がフェイルオーバー・メンバに昇格された場合、新しいアービターをデータ・センタ 2 で使用可能にすることができ、ミラー構成を新しいアービターの IP アドレスで更新

することができます。示されている構成では、反対側のデータ・センタで2つのフェイルオーバー・メンバを使用して長時間動作することは意図されていません。このように動作する場合、別の3つ目の場所（この描画ではインターネット）にアービターを配置することをお勧めします。詳細は、“[ミラーの可用性を最適化するためのアービターの配置](#)”を参照してください。

3.6.2.3 地理的に分離されたフェイルオーバーのペア



この構成では、別個の場所で2つのマシンを利用して、最小限のハードウェアで高可用性と災害復旧の両方のニーズを満たします。フェイルオーバー・メンバ間のネットワーク遅延は、重要な考慮事項の1つですが、その影響がある場合、それはアプリケーションによって異なります。詳細は、“[ネットワーク遅延に関する考慮事項](#)”を参照してください。

ミラーリングでは、1つのフェイルオーバー・メンバがプライマリとして動作するためにもう1つのフェイルオーバー・メンバより優先されることはなく、フェイルオーバーはあらゆるタイプの停止の結果として生じ、これはプライマリ上の問題が一時的なものであったことがわかった場合でも当てはまります。このため、この構成は、特定のデータ・センタで実行されているプライマリに暗黙的な優先順位がない場合に最適です。

この構成では、前述の説明で示されている理由により、VIPの使用は可能かどうかはわかりません。2つのデータ・センタ間のフェイルオーバーは自動的に発生するため、採用する代替方法では、ユーザを迅速に自動で新しいプライマリにリダイレクトする必要があります。手動操作が必要な方法は通常、十分ではありません。

この構成では、以下のIPアドレスが使用されています。

Virtual IP Address	Not Depicted	
Arbiter Address	203.0.113.46 ¹	
Member-Specific Mirror IP Addresses for Member	A	L
SuperServer Address	10.1.20.11	10.5.10.23
Mirror Private Address	10.0.8.11	10.0.11.23
Agent Address	10.1.20.11	10.5.10.23

メモ：

1. この構成では、アービターは、3 つ目の場所に配置することが最適です。詳細は、“[ミラーの可用性を最適化するためのアービターの配置](#)”を参照してください。
2. ここでは、データ・センタ相互接続 (WAN) 経由で実行されるミラー通信のプライベート・ネットワークが描かれていますが、これは必須ではありません。

3.7 フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト

バックアップ・フェイルオーバー・メンバが自動フェイルオーバーによってプライマリになるか、DR 非同期を手動で、[災害復旧の一部としてプライマリに昇格する場合](#)、アプリケーション接続を新しいプライマリにリダイレクトするメカニズムが必要です。これを実現する方法は多数ありますが、このページではそのいくつかを詳しく取り上げます。1 つのソリューションを自動フェイルオーバーと DR の昇格の両方に適用することも、複数のソリューションを組み合わせることもできます。例えば、自動フェイルオーバーにはミラー VIP を使用し、DR の昇格には DNS の更新を使用する方法です。

- ・ [組み込みのメカニズム](#)
- ・ [外部テクノロジー](#)
- ・ [ミラー仮想 IP \(VIP\) の計画](#)

3.7.1 組み込みのメカニズム

“[ミラーリング・アーキテクチャとネットワーク構成のサンプル](#)”で示しているように、ミラー構成に以下のメカニズムを追加して、アプリケーションのリダイレクトに対処できます。

- ・ [ミラー仮想 IP アドレス \(VIP\)](#)

ミラー VIP が使用されていて (“[ミラー仮想 IP \(VIP\) の計画](#)”を参照)、メンバがプライマリになった場合、VIP が新しいプライマリ上のローカル・インタフェースに自動的にバインドされます。これにより、外部クライアントは同じ IP アドレスに引き続き接続できます。“[ミラーリング・アーキテクチャとネットワーク構成のサンプル](#)”で示しているように、VIP の使用では、メンバが同一のサブネット上でプライマリになることが可能である必要があります。

注釈 通常、VIP はクラウド環境では使用できません。クラウドでミラーリングする場合の代替手段やその他の考慮事項は、“[クラウド環境でのミラーリング](#)”を参照してください。

- ・ [分散キャッシュ・クラスタ](#)

ミラーリングされた分散キャッシュ・クラスタ (“[ミラーへのアプリケーション・サーバ接続の構成](#)”を参照) では、フェイルオーバー・メンバをデータ・サーバとして構成し、すべてのアプリケーション・サーバの接続を具体的にミラー接続として構成します。フェイルオーバーに続き、アプリケーション・サーバが新しいプライマリ・フェイルオーバー・メンバへのそれぞれの接続を再確立し、進行中の作業負荷を引き続き処理します。フェイルオーバー・プロセスの実行中は、アプリケーション・サーバに接続しているユーザが作業に復帰する前に、接続が一時的に中断することがあります。ECP リカバリの詳細は、“[ECP リカバリ](#)”と“[ECP リカバリ・プロセス、保証、および制限](#)”を参照してください。

分散キャッシュの主な目的は水平方向の拡張であることに注意してください。クラスタを単に HA の計画のコンポーネントとして配置することは、メリットだけでなく、複雑さの増大や新たな障害点の発生などの追加的なコストも生じる可能性があります。

- ・ [Web ゲートウェイ](#)

Web ゲートウェイのサーバ・アクセス・エントリがミラー認識で構成されている場合、ゲートウェイは初期構成でフェイルオーバー・メンバのいずれかに接続され、そのメンバからミラー内のフェイルオーバーおよび DR 非同期メンバの

リストを取得します。ゲートウェイはこのリストに基づいて現在のプライマリを識別して接続します。ミラーがフェイルオーバーした場合、ゲートウェイは新しいプライマリに接続を変更します。フェイルオーバー・メンバの中でプライマリが見つからない場合、ゲートウェイはリスト内の DR 非同期の中からプライマリを見つけようと試みます。これにより、DR 非同期が**プライマリに昇格**するとゲートウェイは接続を再確立できます。ミラー認識構成のゲートウェイ接続では、スーパーサーバ・アドレスを使用してミラー・メンバと通信します（“**ミラー・メンバのネットワーク・アドレス**”を参照）。

一般に、アプリケーション接続をプライマリにリダイレクトする他の方法が有効になっている場合（VIP など）、そのメカニズムへの標準の Web ゲートウェイ接続（ミラー認識接続ではなく）を構成することが推奨されます。ミラー認識 Web ゲートウェイ接続は、アプリケーション接続をリダイレクトするための主な手段としてのみ使用する必要があります。

既定では、サーバ・アクセス・エントリはミラー認識ではありません。これは、InterSystems IRIS 管理ポータルをサポートしているゲートウェイ・サーバなど、多くのゲートウェイ・サーバ構成に適していないためです。ミラー認識 Web ゲートウェイ接続の詳細は、“**サーバ・アクセスの構成**”を参照してください。

3.7.2 外部テクノロジー

ミラーリングと併用して以下のメカニズムを実装し、アプリケーションのリダイレクトに対処できます。

- ・ ハードウェアによるロード・バランサおよびサイト選択機能
ハードウェアによるサイト選択機能などのメカニズムを使用して、ネットワーク・レベルでアプリケーション・トラフィックをリダイレクトする機能を実装できます。
- ・ DNS の更新
自動および手動のオプションを使用できます。一部のオプションは、自動フェイルオーバーで使用するには遅すぎる場合があります。
- ・ アプリケーションのプログラミング
ミラー・メンバの情報を保持して、現在のプライマリに接続するように、個々のアプリケーションを適合させることができます。
- ・ ユーザ・レベルの手順
複数のミラー・メンバに接続する手段をユーザに提供できます。例えば、災害復旧サイトに接続するための 2 つ目のアイコンなどです。

3.7.3 ミラー仮想 IP (VIP) の計画

“**組み込みのメカニズム**”で説明したように、ミラー VIP が使用されていてメンバがプライマリになると、VIP は新しいプライマリに再割り当てされます。これにより、どちらのフェイルオーバー・メンバが現行プライマリとして動作しているかに関係なく、すべての外部クライアントおよび接続が単一の静的 IP とやりとりできます。

フェイルオーバー・プロセス中に、接続しているネットワークが切断されたクライアントは、バックアップがプライマリになると再接続できます。VIP が構成されている場合、バックアップは VIP の正常割り当てが可能な場合のみフェイルオーバーを完了します。そうでない場合、フェイルオーバー・プロセスは中止され、ミラーには手動操作が必要になります。

ミラー VIP の設定準備においては、以下について検討します。

- ・ ミラー VIP を使用するには、両方のフェイルオーバー・メンバが同じサブネットに構成され、VIP が各システムで選択されているネットワーク・インタフェースと同じサブネットに属している必要があります。DR 非同期メンバは、**災害復旧の一部としてプライマリに昇格した**場合は、VIP を取得できるように同じサブネット上にネットワーク・インタフェースを有する必要があります。それ以外の場合は、災害復旧の手順に代替のリダイレクト・メカニズムを組み込む必要があります。

- ・ フェイルオーバー・メンバまたは DR 非同期メンバが別々のデータ・センタにある場合、VLAN サブネットをデータ・センタ間に拡張して、同じ VIP アドレスをサポートし続けることができます。これには、2 つのサイト間にレイヤ 2 接続が必要ですが、すべての状況で十分であるとは言えません。“[2 つのデータ・センタと地理的に分離された災害復旧に対するミラーリング構成](#)” の説明を参照してください。
- ・ クライアントの接続に使用するために、DNS サーバ上の VIP に DNS 名を割り当てる必要があります。
- ・ VIP が使用中であり、フェイルオーバー・メンバが VIP サブネットから削除される場合、そのメンバは DR 非同期に降格させる (“[DR 非同期へのバックアップの降格](#)” を参照) かまたはミラーから削除する必要があります。あるいは、VIP 構成を両方のフェイルオーバー・メンバから削除する必要があります。そのようにしないと、フェイルオーバー・メンバがプライマリとして引き継ぎを試行しても、VIP の獲得に失敗し、プライマリになることができません。

重要 1 つ以上のミラー・メンバが、UNIX® または Linux システム上の非 root InterSystems IRIS インスタンスの場合、“[InterSystems IRIS 非 root インストール](#)” で説明されているように、ミラー VIP は使用できません。

3.8 仮想環境でのミラーリング

ミラーを構成する InterSystems IRIS インスタンスが仮想ホストにインストールされている仮想化環境でミラーリングを使用すると、ミラーリングの利点と仮想化の利点を併せ持つハイブリッド高可用性ソリューションが実現します。ミラーは計画的または計画外の停止に自動フェイルオーバーで即座に対応し、仮想 HA ソフトウェアは計画外のマシン停止や OS 停止後に、ミラー・メンバをホストする仮想マシンを自動的に再起動します。これにより、障害が発生したメンバは、すぐにミラーに再参加できるようになり、バックアップとして動作します（必要であればプライマリを引き継ぎます）。

ミラーを仮想化環境で構成する場合は、次の推奨事項が適用されます。

- ・ フェイルオーバー・メンバの仮想ホストは、同じ物理ホスト上に配置するようには構成しないでください。
- ・ 単一障害点を回避するには、フェイルオーバー・メンバ上の InterSystems IRIS インスタンスによって使用されるストレージは、個別のディスク・グループまたはストレージ・アレイ上の個別のデータストアに恒久的に分離しておいてください。
- ・ 仮想プラットフォーム・レベルで実行される一部の操作（バックアップや移行など）により、フェイルオーバー・メンバが長い時間応答しなくなって不要なフェイルオーバーや望ましくない頻度の警告が発生することがあります。この問題に対処するために、サービス品質タイムアウトの設定値を大きくしてください (“[サービス品質タイムアウト](#)” を参照)。
- ・ フェイルオーバー・メンバの接続中断をもたらすような、計画的なメンテナンスを実施する場合は、不要なフェイルオーバーと警告を避けるために、一時的にバックアップでの[ミラーリングを停止](#)することができます。
- ・ スナップショット管理は、ミラー・メンバに対しては注意深く使用する必要があります。メンバを以前のスナップショットに戻すと、メンバの最新のステータス（例えば、スナップショットが取られて以降にプライマリからバックアップに変更が加えられている場合など）と、他のメンバにより継続して処理されているジャーナル・データの両方が消去されてしまうことになります。特に次の事柄には注意してください。
 - － 以前のスナップショットに戻されているフェイルオーバー・メンバは、必ず電源オフの状態から再開してください。電源オンの状態から再開すると、両方のフェイルオーバー・メンバが同時にプライマリとして動作する可能性が生じます。
 - － 以前のスナップショットに戻されたフェイルオーバー・メンバが、スナップショット以降に作成されたすべてのジャーナル・データを取得せずにプライマリになる場合（例えば、強制的にプライマリにされる）、他のすべてのミラー・メンバを再構築する必要があります (“[ミラー・メンバの再構築](#)” で説明しています)。

3.9 クラウド環境でのミラーリング

InterSystems IRIS をパブリック・クラウド・プラットフォームに導入する場合は、ミラーリングによって高可用性と災害復旧性を備えた堅牢なソリューションを実現できます。使用しているアプリケーションの要件によって導入の詳細は異なりますが、一般的な考慮事項を以下に示します。

- ・ 同じリージョンにある 2 つの可用性ゾーンにフェイルオーバー・メンバを分割配置することをお勧めします。この構成により、遅延と復元可能性の適切な組み合わせが得られます。
- ・ アプリケーションで更新のワークロードがきわめて大きい場合や同期コミット・トランザクションを使用している場合は、近接して配置したグループを導入することで、復元可能性はやや低下するものの、遅延をさらに低減できる余地が生まれます。このような配置のグループでは、単一の可用性ゾーンに仮想マシン・インスタンスを互いに物理的に近い位置に置き、遅延を最小限に抑えることができます。この構成は、復元可能性が低くなるので、アプリケーションできわめて少ない遅延を必要とする場合にのみ検討します。
- ・ 最大限の復元可能性を得るには、フェイルオーバー・メンバとは別のリージョンに DR 非同期メンバを配置する必要があります。
- ・ 通常、VIP はクラウド環境では使用できません。ただし、代替手段があります。物理的または仮想のロード・バランサなどのネットワーク・トラフィック管理アプライアンスを使用して、VIP と同程度の透過性を実現できます。これにより、クライアント・アプリケーションやデバイスに単一のアドレスを提示できます。プラットフォームによっては、専用の構成を使用して VIP を確立できることもあります。クラウド環境でフェイルオーバー後のアプリケーションのリダイレクトを構成するためのオプションの詳細は、InterSystems Developer Community で [“Database Mirroring without a Virtual IP Address”](#) および [“VIP in AWS”](#) を参照してください。
- ・ 通常、アービターの配置場所は、Web サーバと同じネットワーク層、またはクラウドベースの InterSystems IRIS 導入への入口となる場所とする必要があります。Web サーバもアプリケーション・サーバもない場合は、ミラー・メンバ自体と同じ層またはセキュリティ・グループにアービターを配置できます。
- ・ 2 つの可用性ゾーンがある導入では、どちらのゾーンにアービターを配置してもかまいません。その他の目的で 3 番目のゾーンを既に導入している場合は、そのゾーンにもアービターを配置すると役立つ可能性があります。アービター専用新たに 3 番目のゾーンを導入しても、通常はコストと管理のオーバーヘッドが増加するので、利点が相殺されます。詳細は、[“ミラーの可用性を最適化するためのアービターの配置”](#) を参照してください。
- ・ クラウドでミラーリングを実装する場合は暗号化を強くお勧めします。プライマリ・メンバだけでなく、すべてのミラー・メンバで暗号化を使用することが最良です。[データベースの暗号化](#)により、ストレージ・ボリューム上のデータは保護されますが、ミラー・メンバ間で転送するデータは保護されません。転送するデータを保護するには、[ジャーナルの暗号化](#)も構成する必要があります。パブリック・クラウド・プラットフォームもデータベース暗号化を提供しますが、ストレージ・ボリュームを構成するときに暗号化の指定が必要になることがあります。暗号化の一般的な情報は、[“暗号化ガイド”](#) を参照してください。

3.10 バックアップ・フェイルオーバー・メンバへのアクセスの制限

ミラーのバックアップ・フェイルオーバー・メンバをホストするシステムに未使用のリソースまたは容量がある場合、またはそのミラーリングされたデータベースに読み取り専用のクエリを実行する場合、そのホストをバックアップ・ミラー・メンバとしての役割に専念させることを最善の方法としてお勧めします。バックアップをミラー関連としてまたはミラー以外に使用すると、以下の影響が生じる可能性があります。

- ・ バックアップのパフォーマンス低下によって、プライマリからのジャーナル・データの受信確認が遅くなると、プライマリ上のミラーリングされたデータベースにアクセスするアプリケーションのユーザに対してパフォーマンスの低下が生じる場合があります。プライマリによる確認を待機する必要があるアプリケーション対話は、明示的なジャーナル同期

化、同期コミット・トランザクション、ECP アクティビティなどが関連するものを含め、すべてこのように影響を受ける可能性があります。

- ・ バックアップからの確認が遅くて、[サービス品質タイムアウト](#)の時間内に行われない場合、プライマリはバックアップのアクティブ・ステータスを取り消すので、プライマリ停止のタイプによっては、[自動フェイルオーバー](#)がより困難になるか不可能になります。
- ・ 自動フェイルオーバーが発生したときに、バックアップが自身の既存のリソース使用とプライマリのユーザ・アプリケーションのリソース使用の両方をサポートします。これが起こりうる場合、バックアップのホストには、これらの負荷を両方とも処理できる容量が必要です。

これらの理由により、ユーザの操作をプライマリからオフロードする必要がある場合、バックアップではなく、非同期メンバを使用する必要があります。

3.11 単一ホストへの複数のミラー・メンバのインストール

ミラーを構成する InterSystems IRIS インスタンスは通常、別個の物理ホストまたは仮想ホストにインストールされますが、それは要件ではありません。パフォーマンスを低下させることなく、関係するリソース負荷を処理するのに十分な容量がシステムにあるとすれば、複数のミラー・メンバ（複数のミラー全体を含め）を同一ホストにインストールできます。個々の状況で、これが実現可能かどうか、またいくつのミラーまたはミラー・メンバをコホストできるのかを判断します。

複数のフェイルオーバー・メンバをコホストする場合、フェイルオーバー・ミラーリングでは、メンバは同等と見なされることに留意してください。優先されるプライマリ・メンバはありません。このため、フェイルオーバー・メンバのインスタンスを個別のホストに配置する際の最善の方法は、容量においてそれらのホストを可能な限り同じ、つまりほぼ同等にすることです。フェイルオーバー・メンバのコホストでは、対象モデルの境界の外に出る可能性があります。例えば、5 つのミラーを 5 つの個別ホストに作成して、1 つのホスト上の 5 つの InterSystems IRIS インスタンスを第 2 のフェイルオーバー・メンバとしてミラーに追加した場合、ミラーは最初、個別ホスト上の各プライマリと単独システムでコホストされているすべてのバックアップとを使用して動作します。ところが、フェイルオーバーに至る 2 つの同時、またはほとんどの同時の停止があると、単独システムで 2 つのプライマリと 3 つのバックアップをホストすることになり、適切なパフォーマンスを確保しながら処理するには、システムにとって負荷が大きくなりすぎる可能性があります。

1 つ以上のミラーに属する複数の InterSystems IRIS インスタンスがコホストされている場合、それらのインスタンスは 1 つの ISCAgent を共有します。

複数のミラー・メンバをコホストする場合は、次のネットワーキングに関する考慮事項に留意してください。

- ・ 各ミラーが各マシンで一意のポート・セットを使用し（[“ミラー・メンバのネットワーク・アドレス”](#)を参照）、コホストされていない他のミラー・メンバ（存在する場合）が同じポートを使用するよう設定する必要があります。例えば、2 つの個別ホスト上で実行されている 2 つのプライマリが両方ともポート 1972 を使用しているが、コホストされている DR 非同期メンバに両方が置き換えられる可能性がある場合、前述の説明で示したように、新しいプライマリ（以前の DR 非同期）ではポート割り当てが競合します。1 つのプライマリがポート 1972 を使用し、もう 1 つのプライマリが 1973 を使用し、これらと同じポートが非同期で構成されていると、非同期は同時昇格の準備ができていますので、発生時にクライアントは停止前と同じポートを使用してミラーにアクセスできます。
- ・ 各ミラーのフェイルオーバーおよび DR 非同期メンバは（他のミラーまたはそのメンバと、全部がコホストされているか、一部がコホストされているかにかかわらず）、独自のサブネットを持つ必要があり、各ミラーは独自の VIP（VIP が使用されている場合）を持つ必要があります。これは、使用している環境によっては多少複雑であり、ミラー・メンバをコホストするノードに、複数のサブネットをサポートする複数の NIC が必要になります。ミラー・ネットワーク構成および VIP に関する重要な情報について、[“ミラーリング通信”](#)、[“ミラーリング・アーキテクチャとネットワーク構成のサンプル”](#)、[“ミラー仮想 IP \(VIP\) の計画”](#) および [“ミラー仮想 IP \(VIP\) の構成”](#) を参照してください。
- ・ ミラー・メンバのコホストは、各ミラーに対応するアービターのネットワークの位置に影響を与えません（[“ミラーの可用性を最適化するためのアービターの配置”](#)を参照）。関与するミラーは、アービターを共有することも、個別のアービ

ターを使用することもできます。ただし、フェイルオーバー・メンバとアービターが適切に配置されている必要があります。

4

ミラーリングの構成

ここでは、ミラーとミラー・メンバの設定、構成、および管理に関する情報とその手順を示します。

4.1 ミラーの自動導入方法

ここでは、管理ポータルを使用して、ミラーを作成し、既存のインスタンスをメンバとして構成する手順について説明します。InterSystems IRIS データ・プラットフォームでは、導入後に完全に機能するミラーの自動導入方法がいくつか提供されています。

4.1.1 InterSystems Cloud Manager (ICM) を使用したミラーの導入

ミラー構成を含め、InterSystems Cloud Manager (ICM) を使用して InterSystems IRIS を導入することをお勧めします。プレーン・テキストの宣言型構成ファイル、シンプルなコマンド行インタフェース、および Docker コンテナへの InterSystems IRIS の導入を組み合わせることにより、ICM では、簡単かつ直感的にクラウド・インフラストラクチャまたは仮想インフラストラクチャをプロビジョニングし、目的の InterSystems IRIS アーキテクチャを他のサービスと共にそのインフラストラクチャに導入することができます。ICM により、複雑な水平クラスタ構成の場合は特に、導入プロセスを大幅に簡素化することが可能です。

スタンドアロン・ミラー・インスタンスを導入できるほか、ICM では分散キャッシュ・クラスタをミラー・データ・サーバと共に、またシャード・クラスタをミラー・データ・ノードと共に導入することもできます。

ICM を使用して InterSystems IRIS をミラー構成で導入する方法の詳細は、“[ICM の使用](#)”と“[ICM クラスタのトポロジとミラーリング](#)”を参照してください。

4.1.2 InterSystems Kubernetes Operator (IKO) を使用したミラーの導入

[Kubernetes](#) は、コンテナ化されたワークロードとサービスの導入、拡張、および管理を自動化するためのオープンソースのオーケストレーション・エンジンです。導入するコンテナ化されたサービスと、そのサービスを管理するポリシーを定義すると、Kubernetes は、必要なリソースを可能な限り最も効率的な方法で透過的に提供します。また、導入が指定値から外れた場合は導入を修復またはリストアするほか、拡張を自動またはオンデマンドで行います。InterSystems Kubernetes Operator (IKO) は、IrisCluster カスタム・リソースで Kubernetes API を拡張します。このリソースは、InterSystems IRIS のシャード・クラスタ、分散キャッシュ・クラスタ、またはスタンドアロン・インスタンスとして、すべて任意でミラーリングした状態で、Kubernetes プラットフォームに導入できます。

Kubernetes で InterSystems IRIS を導入するのに IKO は必須ではありませんが、プロセスが大幅に簡易化され、InterSystems IRIS 固有のクラスタ管理機能が Kubernetes に追加され、クラスタにノードを追加するなどのタスクが可能になります。このようなタスクは、IKO を使わなければインスタンスを直接操作して手動で行わなければなりません。

IKO の使用法の詳細は、“[InterSystems Kubernetes Operator の使用](#)”を参照してください。

4.1.3 構成マージを使用したミラーの導入

Linux および UNIX® システムで利用可能な構成マージ機能を使用すると、宣言型構成マージ・ファイルを導入内の各インスタンスに適用することにより、同じイメージから導入した InterSystems IRIS コンテナや、同じキットからインストールしたローカル・インスタンスの構成を変更することができます。このマージ・ファイルは、既存のインスタンスを再起動したときに適用することもでき、インスタンスの構成パラメータ・ファイル (CPF) を更新します。CPF にはインスタンスのほとんどの構成設定が含まれており、これらの設定は、インスタンス導入後の最初の起動を含め、開始時に毎回、CPF から読み取られます。導入時に構成マージを適用すると、インスタンスと共に提供された既定の CPF が実質的に独自の更新バージョンに置き換えられます。

構成マージを使用すると、ミラーリングされたデータベースを含め、1 つ以上のミラーを導入 (または既存のインスタンスから構成) できます。それには、さまざまなミラー・ロールに別個のマージ・ファイルを適用し、最初のフェイルオーバー・メンバ、2 番目のフェイルオーバー・メンバ、DR 非同期メンバの順に導入または構成します。(ミラーの導入または構成後に、レポート非同期メンバを手動でミラーに追加する必要があります。)導入ホストの名前が特定のパターンに一致する場合、複数のフェイルオーバー・ペアを自動的に導入したり、既存のプライマリ複数のバックアップを導入することもできます。この場合、プライマリとバックアップの両方に 1 つのマージ・ファイルを使用し、フェイルオーバー・ペアの自動導入後に、DR 非同期メンバに別個のマージ・ファイルを使用できます。

構成マージ機能を使用して、分散キャッシュ・クラスタをミラー・データ・サーバと共に、またシャード・クラスタをミラー・データ・ノードと共に導入することもできます。

前述のように、IKO には構成マージ機能が組み込まれています。

一般的な構成マージの使用法および具体的なミラーの導入方法の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)”を参照してください。

4.2 ミラー構成のガイドライン

強固でコスト効率のよい HA ソリューションを提供するために、ミラーリングは、さまざまなシステム構成およびアーキテクチャに適応できるように設計されています。ただし、以下の一般的な構成ガイドラインに従うことをお勧めします。

- ・ InterSystems IRIS インスタンスおよびプラットフォームの互換性 – ミラーに追加するシステムを特定する前に、“[InterSystems IRIS インスタンスの互換性](#)”および“[メンバのエンディアンに関する考慮事項](#)”に記載されている要件を必ず確認してください。
- ・ フェイルオーバー・メンバの平等性 – 1 つのミラー内の 2 つのフェイルオーバー・メンバは平等であると見なされます。一方をプライマリとする優先順位を構成することはできません。また、必要に応じてプライマリとバックアップのロールは入れ替わります。このため、最適な方法として、フェイルオーバー・システムのホストが可能な限り互いに類似するように、特に類似したコンピュータ・リソースで構成します。つまり、2 つのシステムの CPU、メモリ構成、およびディスク構成に互換性を持たせて構成します。
- ・ プライマリ・インスタンス構成およびセキュリティ設定 – プライマリ・フェイルオーバー・メンバ上のユーザ、ロール、ネームスペース、およびマッピング (グローバル・マッピングとパッケージマッピングを含む) などの要素の構成は、他のミラー・メンバ上のミラーにより複製されることはありません。したがって、バックアップ・フェイルオーバー・メンバまたは DR 非同期メンバが効率的にプライマリからの引き継ぎを行うために必要な設定はいずれも、必要に応じてそれらメンバに手動で複製し、更新する必要があります。
- ・ ミラーリングされないデータ – プライマリ・フェイルオーバー・メンバのミラーリングされるデータベースにおけるデータのみが、バックアップ・フェイルオーバー・メンバまたは非同期メンバ上で複製され、同期を保ちます。したがって、バックアップまたは DR 非同期が効率的にプライマリから引き継ぎを行うために必要なファイルはすべて (例えば、SQL ゲートウェイおよび Web サーバの構成に関連するものを含む)、必要に応じて、それらのメンバに手動でコピーして更新する必要があります。

注釈 ミラーリングされたデータベースのファイル・ストリームは、デフォルトではデータベース・ディレクトリの **stream** サブディレクトリにあり、ミラーリングされません (ファイル・ストリームの詳細は、“[ストリームを使用した作業](#)”を参照してください)。

- ICMP – ミラーリングはメンバへのアクセス可否の検出をインターネット制御メッセージ・プロトコル (ICMP) に依存しているため、ミラー・メンバとして構成されているすべてのシステムで、ICMP を無効にしないでください。
- ネットワーク – 2 つのフェイルオーバー・メンバ間で、広帯域かつ低遅延の、信頼性の高いネットワークを使用することをお勧めします。可能であれば、2 つのフェイルオーバー・メンバ用にプライベート・サブネットを作成し、このプライベート・ネットワークにデータ・チャンネル・トラフィックと制御チャンネル・トラフィックを専用でルーティングできるようにすることをお勧めします。ネットワークが遅いと、プライマリ・フェイルオーバー・メンバとバックアップ・フェイルオーバー・メンバの両方のパフォーマンスに影響する可能性があり、フェイルオーバー時にバックアップ・フェイルオーバー・メンバがプライマリを引き継ぐときに直接的な影響を及ぼす可能性があります。ネットワークにおける要件および構成に関する詳細は、“[ネットワーク構成に関する考慮事項](#)” および “[ネットワーク遅延に関する考慮事項](#)” を参照してください。また、VIP 使用時の重要なネットワーク要件や考慮事項の詳細は、“[ミラー仮想 IP \(VIP\) の構成](#)” を参照してください。
- ディスク・サブシステム – バックアップ・フェイルオーバー・メンバをプライマリ・システムと同じ状態に保つためには、両方のフェイルオーバー・メンバのディスク・サブシステムが同等である必要があります。例えば、第 1 のフェイルオーバー・メンバにストレージ・アレイを構成する場合は、第 2 のフェイルオーバー・メンバにも同様のストレージ・アレイを構成することをお勧めします。また、ネットワーク接続ストレージ (NAS) が一方または両方のシステムで使用されている場合は、ディスク I/O とミラー・データからのネットワーク負荷に対して別々のネットワーク・リンクを構成し、ネットワークのオーバーフローの可能性を最小限に抑えることを強くお勧めします。
- ジャーナリングのパフォーマンスおよびジャーナルのストレージ – ジャーナル/ディジャーナルはミラー同期の中核を成す機能なので、フェイルオーバー・メンバのジャーナリングのパフォーマンスを監視して最適化するのは重要です。特に、すべてのミラー・メンバで共有メモリ・ヒープ (gmheap) のサイズを大きくすることをお勧めします。パフォーマンスと復元可能性の両方を高めるために、プライマリおよび代替ジャーナル・ディレクトリは、それぞれ異なるストレージ・デバイスに配置し、データベースやライト・イメージ・ジャーナル (WIJ) が使用しているストレージ・デバイスとも異なるストレージ・デバイスに配置することをお勧めします。詳細は、“[ジャーナリングの最善の使用法](#)”と“[ジャーナル・ファイルのリストア](#)”を参照してください。
- 仮想化 – 仮想化環境でのミラーリングの使用により、その両方の利点を併せ持つ高可用性ソリューションが提供されますが、重要な推奨事項が適用されます。詳細は、“[仮想環境でのミラーリング](#)”を参照してください。
- タスクのスケジューリング – タスク・マネージャを使用してミラー・メンバのタスクを作成する場合、タスクを実行できるメンバを、プライマリのみ、プライマリ以外の任意のメンバ、もしくは任意のミラー・メンバのどれにするかを指定する必要があります。複数のミラー・メンバ上で実行されることが意図されているタスクは、各メンバ上で個別に作成するか、または 1 つのメンバのタスク・マネージャからエクスポートして他のメンバにインポートする必要があります。タスクの作成、インポート、エクスポートの詳細は、“[タスク・マネージャの使用](#)”を参照してください。
- 開始 – プライマリ・フェイルオーバー・メンバでは、コードを既存の `%ZSTART` ルーチンから `ZMIRROR` ルーチンに移動して、ミラーを初期化しない限りコードが実行されないようにすることができます。詳細は、“[ZMIRROR ルーチンの使用法](#)”を参照してください。

4.3 アービターのインストール

自動フェイルオーバーができる限り広範な停止シナリオに対処できるように、“[自動フェイルオーバーのメカニズム](#)”の説明に従って、各ミラーにアービターを構成することをお勧めします。“[ミラーの可用性を最適化するためのアービターの配置](#)”の説明にあるように、推奨されるアービターのネットワーク上の場所は、フェイルオーバー・メンバの位置によって異なります。1 つのシステムが複数のミラーそれぞれに適切な場所にあれば、そのシステムを複数のミラーのアービター

として構成できます。そのように構成するには、システムがアービターとして機能する対象の各ミラーを作成または編集するときに、“[ミラーの作成](#)”の説明にあるようにホストとポート番号を指定します。

アービターとして動作するには、システムで ISCAgent プロセスが稼働している必要があります。ISCAgent は InterSystems IRIS と共にインストールされるため、InterSystems IRIS のインスタンスを 1 つ以上ホストするシステムは、この要件を満たしていて、それ以上の準備を必要とせずにアービターとして構成できます。ただし、ミラーのフェイルオーバー・メンバまたは DR 非同期メンバを 1 つ以上ホストしているシステムは、そのミラーのアービターとして構成してはいけません。

InterSystems IRIS インスタンスをホストしていないシステムを、アービターとして機能するように準備するには、次のいずれかの方法を使用します。

- ・ キットを使用して ISCAgent をインストールします。

そのようなシステムの準備を整えるには、目的のアービター・システムに適した ISCAgent インストール・キットをインターシステムズからダウンロードして、以下に示すように ISCAgent をインストールします。

- Windows システムの場合は、単にインストール・ファイル（例えば、**ISCAgent-2018.1.0.540.0-win_x64.exe**）を実行します。ログと自動インストールのコマンドライン・オプションを使用してインストールすることもできます。コマンドライン・オプションの使用の詳細は、“[自動インストール手順](#)”を参照してください。
- UNIX®、Linux、および macOS システムの場合は、単一ファイルのインストール・キットをアンパックして（必要な場合）、インストール・キットの最上位レベル **/ISCAgent** で **agentinstall** を実行します。以下に例を示します。

```
[root@arbiterhost home]# gunzip ISCAgent-2020.1.0.540.0-lnxrhx64.tar.gz
[root@arbiterhost home]# tar -xf ISCAgent-2020.1.0.540.0-lnxrhx64.tar
[root@arbiterhost home]# ./ISCAgent/agentinstall
```

ISC_PACKAGE_MODE を **unattended** に設定することで、このインストールを自動インストールとして実行できます。以下に例を示します。

```
[root@arbiterhost home]# ISC_PACKAGE_MODE="unattended" ./ISCAgent/agentinstall
```

- ・ コンテナで ISCAgent を導入します。

アービターとして機能するように、コンテナ化された ISCAgent を任意の Linux システムに導入します。インターシステムズが提供している **arbiter** イメージの取得および使用の詳細は、“[InterSystems IRIS コンテナを使用したミラーリング](#)”を参照してください。

重要 システムの起動時に、アービター・システムで ISCAgent プロセスが起動するよう構成されていることを確認してください。詳細は、“[ISCAgent の開始および停止](#)”を参照してください。

ポートの設定など、ISCAgent のその他のオプションの詳細は、“[ISCAgent のカスタマイズ](#)”を参照してください。

注釈 InterSystems IRIS がサポートされるすべてのプラットフォームに向けた ISCAgent インストール・キットがあります。“[インターシステムズのサポート対象プラットフォーム](#)”を参照してください。

アービターとして機能する ISCAgent は、構成されているミラーのメンバと同じ InterSystems IRIS バージョンでなくてもかまいません。ただし、確実に ISCAgent の最新バージョンを使用するために、ミラーのアップグレード時にアービターをアップグレードすることをお勧めします。

4.4 ISCAgent の起動

ISCAgent プロセスがホスト・システムで実行されていない限り、InterSystems IRIS インスタンスをフェイルオーバーまたは DR 非同期メンバとしてミラーに追加することはできません。ISCAgent は、システムの起動時に自動的に起動するよう構成されている必要があります。詳細は、“[ISCAgent の開始および停止](#)”を参照してください。

4.5 TLS セキュリティを使用したミラー通信の保護

ミラー内でセキュリティを実現するために、相互での通信時に TLS を使用するようにミラーのメンバを構成できます。ミラーの作成時に TLS を使用する必要がある場合は、すべてのメンバ間のすべての通信に、すべてのメンバが TLS を使用する必要があります。

TLS セキュリティを使用したミラーの作成に関する詳細は、“[ミラーの作成](#)”を参照してください。既存のミラーへの TLS セキュリティの追加に関する詳細は、“[フェイルオーバー・メンバの編集または削除](#)”を参照してください。

インターシステムズのシニア・サポート・スペシャリストが TLS セキュリティを伴ったミラーの作成手順を 1 つにまとめた包括的なガイドがあります。InterSystems Developer Community の“[Creating SSL-Enabled Mirror on InterSystems IRIS Using Public Key Infrastructure \(PKI\)](#)”を参照してください。

重要 ミラーリングでは TLS を使用することを強くお勧めします。ミラーで TLS を無効にしないことを強くお勧めします。

インスタンスでジャーナルの暗号化が有効になっていて、そのインスタンスをミラーのプライマリ・フェイルオーバー・メンバに設定する場合、TLS を使用するようにミラーを構成する必要があります。

ミラー・メンバのミラー通信に TLS の使用には、ミラー・メンバ・インスタンスをホストするシステムで適切な TLS 設定が要求されます。詳細は、“[ミラーリングで TLS を使用するための InterSystems IRIS の構成](#)”を参照してください。

また、ミラーリングにおける暗号化ジャーナル・ファイルの使用には準備が必要です。ジャーナル暗号化の詳細は“[ミラー内ジャーナル暗号の有効化](#)”および“[暗号化ガイド](#)”を参照してください。

4.6 ^MIRROR ルーチンの使用法

ミラーリング構成、管理、およびステータスの操作のほとんどは管理ポータルで実行できます。また、%SYS ネームスペースで実行する ^MIRROR ルーチンでも実行できます。ただし、バックアップ・フェイルオーバー・メンバを強制的にプライマリ・フェイルオーバー・メンバにする操作 (“[自動フェイルオーバーなしのプライマリ計画外停止](#)”を参照) など、操作によっては、^MIRROR ルーチンでのみ実行可能なものもあります。このページに掲載する手順では、使用可能である場合の管理ポータルの操作を説明しますが、同等の操作を備える ^MIRROR オプションについても常に注記します。

4.7 ミラーの作成

ミラーの作成にあたっては、プライマリ・フェイルオーバー・メンバ、通常は (必須ではない) バックアップ・フェイルオーバー・メンバ、および (オプションで) 1 つ以上の非同期メンバを構成します。ミラーを作成した後、データベースをミラーに追加できます。

重要 InterSystems IRIS インスタンスをフェイルオーバー・メンバまたは非同期メンバとしてミラーに追加する前に、“ISC Agent の開始および停止”の説明に従って、ISC Agent プロセスが開始されていることを確認する必要があります。

バックアップ・メンバおよび非同期メンバを追加する手順では、推奨されているように TLS を使用するようミラーを構成する場合に、追加の手順が必要です (“TLS セキュリティを使用したミラー通信の保護”を参照)。この場合、新しいメンバをミラーに参加させる前に、プライマリでそれぞれのメンバを承認する必要があります。

ミラーを作成および構成するには、以下の手順を使用します。

1. ミラーを作成し、第 1 のフェイルオーバー・メンバを構成する
2. 第 2 のフェイルオーバー・メンバを構成する
3. 第 2 のフェイルオーバー・メンバを承認する (TLS ミラーのみ)
4. フェイルオーバー・メンバのステータスをミラー・モニタで確認する
5. 非同期ミラー・メンバを構成する
6. 新しい非同期メンバを承認する (TLS ミラーのみ)

ミラーを作成して、フェイルオーバー・メンバおよび任意で 1 つ以上の非同期メンバを構成してから、“ミラーへのデータベースの追加”の手順を使用して、データベースをミラーに追加します。

重要 タスク・マネージャ (“タスク・マネージャの使用”を参照してください) を使用してシステム・タスクをインスタンスに追加すると、以下のように、[ミラーのためにどのようにタスクを実行するか] の設定によって、タスクが実行されるミラー メンバが決定されます。

- ・ プライマリ・フェイルオーバー・メンバでのみ実行
- ・ バックアップ・フェイルオーバー・メンバおよび非同期メンバでのみ実行 (プライマリを除くすべてのメンバ)
- ・ すべてのミラー・メンバで実行 (プライマリ、バックアップ、および非同期)

インスタンスがミラー・メンバでない場合、この設定は無視されます。ただし、ミラー・メンバにおいて、この設定がユーザ定義タスクで指定されていない場合、タスクは実行されません。また、インスタンスをミラーに追加しても、設定は自動更新されません。

したがって、以下のいずれかを実行する必要があります。

- ・ インスタンスが (まだ) ミラー内に存在しない場合でも、タスクの作成時、必ず [ミラーにおけるタスクの実行方法] を設定する。
- ・ インスタンスがミラーに追加されている場合は、必ずすべてのユーザ定義タスクを確認し、[ミラーにおけるタスクの実行方法] を設定する。

4.7.1 ミラーを作成し、第 1 のフェイルオーバー・メンバを構成する

次の手順では、ミラーを作成し、第 1 のフェイルオーバー・メンバを構成する方法について説明します。

1. 第 1 のフェイルオーバー・メンバで、管理ポータル [ミラーの作成] ページ ([システム管理] → [構成] → [ミラー設定] → [ミラーの作成]) に移動し、[ミラーの作成] をクリックします。このオプションがアクティブでない場合、ミラーリングが有効になっていません。[ミラーサービスを有効にする] をクリックし、[サービス有効] チェック・ボックスにチェックを付けて、[保存] をクリックしてから、[ミラーの作成] オプションを選択します。
2. [ミラーの作成] ページで、[ミラー情報] セクションに次の情報を入力します。
 - a. [ミラー名] – ミラーの名前を入力します。

注釈 有効な名前は、1 ～ 15 文字の英数字とする必要があります。小文字は自動的に大文字に置き換えられます。

- b. **[SSL/TLS が必要]** – このチェック・ボックスにチェックを付ける、または外すことにより、ミラー内のすべての通信に対して (推奨どおりに) TLS セキュリティを必要とするかどうかを指定します。**[SSL/TLS が必要]** を選択するときに、インスタンスでミラーリングに対する有効な TLS 構成を済ませていない場合は、手順を完了する前に、**[SSL/TLS の構成]** リンクをクリックして、このメンバに対して必要な TLS 構成を作成する必要があります。(TLS 構成の作成方法の詳細は、“[ミラー用 TLS 構成の作成および編集](#)” を参照してください。)[ミラーの作成] の手順をキャンセルして、**[TLS 構成]** ページに移動することもできます (**[システム管理]** → **[セキュリティ]** → **[SSL/TLS 構成]**)。ミラーリングに対する有効な TLS 構成が行われているインスタンスの場合、そのリンクは **[SSL/TLS 編集]** になりますが、**[SSL/TLS が必要]** を選択している場合は、そのリンクを使用する必要はありません (その構成を変更する場合を除く)。
- c. **[アービターを使用]** – “[自動フェイルオーバーのメカニズム](#)” で説明したように、できる限り広範な停止シナリオに対して自動フェイルオーバーを有効にするために (推奨どおりに) アービターを構成するかどうかを指定します。**[アービターを使用]** を選択した場合、アービターとして構成するシステムのホスト名または IP アドレス、およびその ISCAgent プロセスで使用するポート (既定では 2188) を指定する必要があります。アービターに関する詳細は、“[ミラーの可用性を最適化するためのアービターの配置](#)” および “[アービターのインストール](#)” を参照してください。
- d. **[仮想 IP を使用]** – チェック・ボックスにチェックを付ける、または外すことにより、仮想 IP アドレスを使用するかどうかを指定します。**[仮想 IP を使用]** を選択した場合、IP アドレス、クラスレスのドメイン間ルーティング (CIDR) マスク、およびネットワーク・インタフェースの入力を求められます。

重要 VIP 構成前の要求事項や重要考慮事項の詳細は、“[ミラー仮想 IP \(VIP\) の構成](#)” を参照してください。

- e. **[フェイルオーバー・メンバの圧縮モード]**、**[非同期メンバの圧縮モード]** – プライマリからバックアップ・メンバおよび非同期メンバにジャーナル・データを転送する前に、データを圧縮するかどうかとそれぞれに使用する圧縮タイプを指定します。詳細は、“[ミラー・トラフィックの圧縮](#)” を参照してください。既定の設定は、どちらも **[システム選択]** です。この設定では、フェイルオーバー・メンバ間の応答時間と、プライマリと非同期との間のネットワーク使用率が最適化されます。

3. **[ミラー・フェイルオーバー情報]** セクションに次の情報を入力します。

- ・ **[ミラー・メンバ名]** – このシステムで構成しているフェイルオーバー・メンバの名前 (既定値は、システムのホスト名と InterSystems IRIS インスタンス名の組み合わせです)。ミラー・メンバ名には、以下の後に続くスペース、タブ、および句読点文字を含めることはできません。

: [] # ; / * = ^ ~ ,

保存する前にアルファベット文字は大文字に変換されます。ミラー・メンバの名前の最大長は 32 文字です。

- ・ **[スーパーサーバ・アドレス]** – このフェイルオーバー・メンバとの通信に外部システムが使用できる IP アドレスまたはホスト名を入力します。通常は既定値を受け入れることができます。スーパーサーバ・アドレスの詳細は、“[ミラー・メンバのネットワーク・アドレス](#)” および “[ミラー・メンバのネットワーク・アドレスの更新](#)” を参照してください。
- ・ **[エージェント・ポート]** – このフェイルオーバー・メンバの ISCAgent のポート番号。プロンプトで提示される、インストールしたエージェントのポートを受け入れます。エージェント・ポートの詳細は、“[ISCAgent の構成](#)” を参照してください。

4. **[詳細設定]** をクリックして、以下のように追加のミラー設定を表示および編集します。

- ・ **サービス品質のタイムアウト (msec)** – フェイルオーバー・メンバが、アクションを実行する前に、他のフェイルオーバー・メンバからの応答を待機する最大時間 (ミリ秒単位)。さらに、フェイルオーバー・メンバの応答をアー

ビターが待機する場合にも適用されます。サービス品質 (QoS) タイムアウトに関する詳細は、“[サービス品質 \(QoS\) タイムアウトの構成](#)” を参照してください。

- ・ **[パラレル・デジャーナリングを許可する]** – 並列デジャーナリングを有効にする対象をフェイルオーバー・メンバーおよび DR 非同期にする (既定) か、レポート非同期を含むすべてのメンバーにするか、またはフェイルオーバー・メンバーのみにするかを指定する設定を変更します。並列デジャーナリングではミラーのスループットが向上しますが、複数のデータベースが関与するクエリまたはレポートの結果の一貫性が損なわれる可能性が若干高まる場合があります。詳細は、“[並列デジャーナリングの構成](#)” を参照してください。
- ・ **[ミラー・プライベート・アドレス]** – このフェイルオーバー・メンバーとの通信に別のフェイルオーバー・メンバーが使用できる IP アドレスまたはホスト名を入力します。“[ミラー・メンバーのネットワーク・アドレス](#)” と “[ミラー・メンバーのネットワーク・アドレスの更新](#)” を参照してください。
- ・ **[エージェント・アドレス]** – このメンバーの ISCAgent に別のミラー・メンバーが最初に通信しようとするアドレスを入力します。“[ミラー・メンバーのネットワーク・アドレス](#)” と “[ミラー・メンバーのネットワーク・アドレスの更新](#)” を参照してください。

5. **[保存]** をクリックします。

注釈 ミラーの作成には、`^MIRROR` ルーチンも使用できます (“[^MIRROR ルーチンの使用法](#)” を参照)。既存のミラー構成なしで、InterSystems IRIS インスタンスの `^MIRROR` を実行する際に、ミラーリングがまだ有効になっていない場合は、**[ミラーサービスを有効にする]** オプションを使用できます。ミラーが有効になると、**[ミラーの作成]** オプションを使用でき、ミラー作成およびプライマリ・フェイルオーバー・メンバーとしてのインスタンス構成の代替手段になります。また、`SYS.Mirror.CreateNewMirrorSet()` ミラーリング API メソッドもこの目的に使用できます。

4.7.2 第 2 のフェイルオーバー・メンバーを構成する

次の手順に従って、ミラーに第 2 のフェイルオーバー・メンバーを構成します。

1. 第 2 のフェイルオーバー・メンバーで、**[フェイルオーバーとしてミラーに参加]** ページ (**[システム管理]** → **[構成]** → **[ミラー設定]** → **[フェイルオーバーとして参加]**) に移動します。**[フェイルオーバーとして参加]** オプションが使用できない場合、最初に **[ミラーサービスを有効にする]** をクリックし、**[サービス有効]** チェック・ボックスにチェックを付けて、**[保存]** をクリックしてから、**[フェイルオーバーとして参加]** オプションを選択します。
2. **[フェイルオーバーとしてミラーに参加]** ページで、**[ミラー情報]** セクションに、[第 1 のフェイルオーバー・メンバーを構成したときに指定したミラー名](#)を入力します。
3. **[他方のミラー・フェイルオーバー・メンバーの情報]** セクションに次の情報を入力します。
 - ・ **[その他のシステムのエージェント・アドレス]** – 第 1 のフェイルオーバー・メンバーを構成したときに指定した **[スーパーサーバー・アドレス]** を入力します。
 - ・ **[エージェント・ポート]** – 第 1 のフェイルオーバー・メンバーを構成したときに指定した ISCAgent のポートを入力します。
 - ・ **[InterSystems IRIS インスタンス名]** – 第 1 のフェイルオーバー・メンバーとして構成されている InterSystems IRIS インスタンスの名前を入力します。
4. **[次へ]** をクリックして、ミラーおよび第 1 のフェイルオーバー・メンバーに関する情報を取得および表示します。**[ミラー・フェイルオーバー・メンバー情報]** セクションには、次の情報が入力されています。
 - ・ **[ミラー・メンバー名]** – このシステムで構成しているフェイルオーバー・メンバーの名前を指定します (既定値は、システムのホスト名と InterSystems IRIS インスタンス名の組み合わせです)。ミラー・メンバー名には、以下の後に続くスペース、タブ、および句読点文字を含めることはできません。

: [] # ; / * = ^ ~ ,

保存する前にアルファベット文字は大文字に変換されます。ミラー・メンバーの名前の最大長は 32 文字です。

- ・ **[スーパーサーバ・アドレス]** – このフェイルオーバー・メンバとの通信に外部システムが使用できる IP アドレスまたはホスト名を入力します。詳細は、[“ミラー・メンバのネットワーク・アドレス”](#)と[“ミラー・メンバのネットワーク・アドレスの更新”](#)を参照してください。
- ・ **[エージェント・ポート]** – このフェイルオーバー・メンバの ISCAgent のポート番号を入力します。プロンプトで提示される、インストールしたエージェントのポートを受け入れます。エージェント・ポートの詳細は、[“ISCAgent の構成”](#)を参照してください。
- ・ **[仮想 IP 用ネットワーク・インタフェース]** – 第 1 のフェイルオーバー・メンバを構成したときに指定したネットワーク・インタフェースが表示されます。この設定を第 2 のフェイルオーバー・メンバで変更することはできません。
- ・ **[SSL/TLS の要件]** – 第 1 のフェイルオーバー・メンバを構成したときに指定した設定が表示されます。この設定を第 2 のフェイルオーバー・メンバで変更することはできません。

ミラーが TLS を要求している場合に、インスタンスでミラーリングに対する有効な TLS 構成を済ませていないときには、手順を完了する前に、**[SSL/TLS の構成]** リンクをクリックして、このメンバに対して必要な TLS 構成を作成する必要があります。(TLS 構成の作成方法の詳細は、[“ミラー用 TLS 構成の作成および編集”](#)を参照してください。)[フェイルオーバーとして参加]の手順をキャンセルして、管理ポータルの [TLS 構成] ページに移動することもできます (**[システム管理]**→**[セキュリティ]**→**[SSL/TLS 構成]**)。

ミラーリングに対する有効な TLS 構成が行われているインスタンスの場合、そのリンクは **[SSL/TLS 編集]** になりますが、TLS が要求される場合は、そのリンクを使用する必要はありません(その構成を変更する場合を除く)。

- ・ **[ミラー・プライベート・アドレス]** – このフェイルオーバー・メンバとの通信に別のフェイルオーバー・メンバが使用できる IP アドレスまたはホスト名を入力します。[“ミラー・メンバのネットワーク・アドレス”](#)と[“ミラー・メンバのネットワーク・アドレスの更新”](#)を参照してください。
 - ・ **[エージェント・アドレス]** – このメンバの ISCAgent に別のミラー・メンバが最初に通信しようとするアドレスを入力します。[“ミラー・メンバのネットワーク・アドレス”](#)と[“ミラー・メンバのネットワーク・アドレスの更新”](#)を参照してください。
5. **[詳細設定]** をクリックすると、第 1 のフェイルオーバー・メンバを構成したときに指定した **[サービス品質タイムアウト (ミリ秒)]** の設定が表示されます。この設定を第 2 のフェイルオーバー・メンバで変更することはできません。
 6. **[保存]** をクリックします。

TLS を使用するようにミラーを構成した場合は、以下のセクションで説明するように、第 1 のフェイルオーバー・メンバで **第 2 のフェイルオーバー・メンバを承認**することで、第 2 のフェイルオーバー・メンバをミラーに追加するプロセスを完了する必要があります。

注釈 第 2 のフェイルオーバー・メンバの構成には、`^MIRROR` ルーチンも使用できます ([“`^MIRROR` ルーチンの使用法”](#)を参照)。既存のミラーリング構成なしで、InterSystems IRIS インスタンスの `^MIRROR` を実行する際に、ミラーリングがまだ有効になっていない場合は、**[ミラーサービスを有効にする]** オプションを使用できます。ミラーリングが有効になると、**[フェイルオーバー・メンバとしてミラーに参加]** オプションを使用でき、バックアップ・フェイルオーバー・メンバの構成とミラーへの追加の両方についての代替手段になります。また、`SYS.Mirror.JoinMirrorAsFailoverMember()` ミラーリング API メソッドもこの目的に使用できます。

4.7.3 第 2 のフェイルオーバー・メンバまたは非同期メンバを承認する (TLS ミラーのみ)

TLS を使用するようにミラーを構成した場合は、**第 2 のフェイルオーバー・メンバの構成**後または**非同期メンバの構成**後に、追加の手順が必要になります。ミラーを作成してから第 1 のフェイルオーバー・メンバを構成したシステムでは、次の手順に従って、新しいミラー・メンバを承認する必要があります。

1. **[ミラーの編集]** ページ (**[システム管理]**→**[構成]**→**[ミラー設定]**→**[ミラーの編集]**) に移動します。

- ページの下部にある、**[保留中の新しいメンバ]** 領域には、ミラーに追加されているメンバがリスト表示されます。承認するメンバを選択し、**[承認]** をクリックして確定します。(第 2 のフェイルオーバー・メンバを追加する際に、そのメンバの TLS 証明書が自動的に検証されます)。
- [ミラーの編集]** ページの **[ミラーメンバ情報]** セクションに示される情報には、追加したメンバが含まれています。(このリストに表示されるアドレスの詳細は、[“ミラー・メンバのネットワーク・アドレス”](#) を参照してください。)

注釈 また、第 1 のフェイルオーバー・メンバの `MIRROR` ルーチンの **[ミラー構成]** メニューにある **[保留中の新しいメンバの承認/拒否]** オプションも、TLS を要求するように構成したミラーで、新しいフェイルオーバー・メンバまたは非同期メンバを承認するために使用できます。

`SYS.Mirror.AddFailoverMember()` ミラーリング API メソッドは、TLS を要求するように構成したミラーで、第 2 のフェイルオーバー・メンバを承認するために使用できます。また、`Config.MapMirrors.Create()` API メソッドは、承認されるメンバ (フェイルオーバーまたはバックアップ) を作成するために使用できます。`SYS.Mirror.VerifyMirrorSSLCertificates()` は、ミラー・メンバの TLS 証明書を検証するために使用できます。

TLS を要求するミラーのメンバに対する X.509 DN 更新を承認する (メンバの証明書が置き換えられた場合など) 方法の詳細は、[“X.509 DN 更新の承認 \(TLS のみ\)”](#) を参照してください。

4.7.4 フェイルオーバー・メンバのステータスをミラー・モニタで確認する

[“ミラーの監視”](#) で説明しているように、ミラー内のフェイルオーバー・メンバに関する情報を、ミラーにおけるその現行ステータス (ロール) を含め、ミラー・モニタを使用して確認できます。以下の手順に従い、ミラー・モニタを使用して、ミラーおよびそのフェイルオーバー・メンバが本来の意図どおりに設定されていることを確認します。

- 構成した第 1 のフェイルオーバー・メンバで、**[ミラー・モニタ]** ページを表示します (**[システム処理]** → **[ミラー・モニタ]**)。
- [ミラー・フェイルオーバー・メンバ情報]** 領域に、2 つのフェイルオーバー・メンバのミラー・メンバ名およびネットワーク・アドレスがリストされます。
- [ミラー・メンバ・ステータス]** 領域の **[ステータス]** 列に、構成した第 1 のフェイルオーバー・メンバが **[プライマリ]** として表示され、第 2 のフェイルオーバー・メンバが **[バックアップ]** として表示されます。[“ミラー・メンバのジャーナル転送およびデジャーナリングのステータス”](#) で説明しているように、バックアップの **[ジャーナル転送]** ステータスが **[アクティブ]** で、**[デジャーナリング]** ステータスが **[キャッチアップ済み]** になっているはずです。
- アービターが構成されている場合、**[アービター接続ステータス]** 領域に、アービターのネットワーク・アドレスとエージェント・ポート番号が表示されます。**[フェイルオーバー・モード]** が **[アービター制御]** で、**[接続ステータス]** が **[両方のフェイルオーバー・メンバがアービターに接続されている]** になっているはずですが、そのようになっていない場合、アービターが正しくインストールされていないか、アービターの ISCAgent プロセスが実行されていないか、または入力したネットワーク・アドレスかポート番号が正しくない可能性があります。ネットワーク上の問題により、一方または両方のフェイルオーバー・メンバがアービターと通信できなくなっている場合も、**[フェイルオーバー・モード]** が **[エージェント制御]** になる可能性があります。

バックアップ・フェイルオーバー・メンバのミラー・モニタにも同じ情報が表示されます。

4.7.5 非同期ミラー・メンバを構成する

構成する各非同期メンバに対して、以下の手順を使用します。フェイルオーバーのペアを使用するミラーには、レポート非同期メンバまたは災害復旧 (DR) 非同期メンバを最大 14 含めることができます。単一の InterSystems IRIS インスタンスは、最大で 10 個のミラーのレポート非同期メンバになれます。ただし、インスタンスは 1 つのミラーでのみ、DR 非同期になれます。読み取り専用または読み書き可能ないずれかのレポート非同期としてインスタンスを構成すれば、そのいずれかのタイプのレポート非同期メンバとしてのみ、他のミラーに追加することができます (ただし、[“非同期メンバのミラー構成の編集”](#) の説明のとおり、レポート非同期メンバのタイプをそのメンバの属するすべてのミラーに対して変更することはできません)。

注釈 レポート非同期メンバとしてインスタンスをミラーに追加する手順は、ここで説明する **[非同期として参加]** オプションを使用する場合も、“**非同期メンバのミラー構成の編集**” で説明する **[非同期構成の編集]** ページの **[ミラーに参加]** ボタンを使用する場合も同じです。ただし、DR 非同期が所属できるのは 1 つのミラーにのみであるため、**[非同期構成の編集]** ページの **[ミラーに参加]** ボタンは、レポート非同期メンバにのみ使用できます。

1. **[非同期としてミラーに参加]** ページ (**[システム管理]**→**[構成]**→**[ミラー設定]**→**[非同期として参加]**) に移動します。**[非同期として参加]** オプションを選択できない場合は、**[ミラーサービスを有効にする]** を選択して、サービスを有効化します。
2. **[非同期としてミラーに参加]** ページにて、**[ミラー名]** プロンプトで、**ミラーの作成**時に指定したミラー名を入力します。
3. プライマリまたはバックアップ・フェイルオーバー・メンバを選択し、**[ミラー・フェイルオーバー・メンバの情報]** セクションで、以下のプロンプトそれぞれに、選択したメンバの情報を入力します。
 - a. **[フェイルオーバー・システムのエージェント・アドレス]** – 選択したフェイルオーバー・メンバを構成したときに指定した **[スーパーサーバ・アドレス]** を入力します。
 - b. **[エージェント・ポート]** – 選択したフェイルオーバー・メンバに指定した ISCAgent のポートを入力します。
 - c. **[InterSystems IRIS インスタンス名]** – 選択したフェイルオーバー・メンバとして構成されている InterSystems IRIS インスタンスの名前を入力します。
4. **[次へ]** をクリックして、フェイルオーバー・メンバの情報を確認した後、**[非同期メンバ情報]** セクションに移動します。このセクションで、次の情報を入力します。

- a. **[非同期メンバ名]** – 対象のシステムで構成している非同期メンバの名前を指定します (既定値は、システムのホスト名と InterSystems IRIS インスタンス名の組み合わせです)。ミラー・メンバの名前には、英数文字、下線文字、およびハイフンが使用できます。

注釈 ミラー・メンバ名を変更することはできないので、この名前はレポート非同期メンバが将来、他のミラーに参加する際にも使用されることになります。

- b. **[非同期メンバ・アドレス]** – この非同期メンバと通信するために外部システムが使用できる IP アドレスまたはホスト名を入力します。

注釈 入力した **[非同期メンバ・アドレス]** は、非同期メンバのスーパーサーバ・アドレスおよびミラー・プライベート・アドレスになります (“**ミラー・メンバのネットワーク・アドレス**” を参照)。例えば、DR 非同期のミラー・プライベート・アドレスをミラー・プライベート・ネットワークに配置し、そのスーパーサーバ・アドレスを外部ネットワークに置いたままにするなど、これらのアドレスを異なったものにするには、ミラーに追加後、非同期のアドレスをプライマリで更新できます。詳細は、“**ミラー・メンバのネットワーク・アドレスの更新**” を参照してください。

- c. **[エージェント・アドレス]** – このメンバの ISCAgent に別のミラー・メンバが最初に通信しようとするアドレスを入力します。“**ミラー・メンバのネットワーク・アドレス**” と “**ミラー・メンバのネットワーク・アドレスの更新**” を参照してください。
- d. **[非同期メンバのシステム・タイプ]** – ドロップダウン・リストから以下のタイプのいずれかを選択します。単一の InterSystems IRIS インスタンスは、複数のミラーのレポート非同期メンバになります。ただしインスタンスは 1 つのミラーでのみ、DR 非同期になります。

・ **[災害復旧 (DR)]** – 単一ミラー内のミラーリングされるデータベースすべてを読み取り専用コピーとして維持するシステムに対するオプションです。これにより、いずれかのフェイルオーバー・メンバの障害時に、DR 非同期メンバがフェイルオーバー・メンバに昇格することができます。DR 非同期の昇格に関する詳細は、“**DR 非同期メンバのフェイルオーバー・メンバへの昇格**” を参照してください。

重要 VIP を使用するようにミラーが構成されている場合、災害復旧の非同期メンバには、フェイルオーバー・メンバへの TCP/IP 直接接続性を備える必要となります。詳細は、“**ミラー仮想 IP (VIP) の構成**” を参照してください。

- ・ **【読み取り専用のレポート】** – このオプションを使用して、1 つ以上のミラーからのミラーリングされるデータベースの読み取り専用コピー（すなわちミラーリングされるデータベースのサブセット）を維持し、データの変更/追加の必要性のないデータ・マイニングおよび企業レポートを利用します。
 - ・ **【読み書き可能なレポート】** – このオプションを使用して、1 つ以上のミラーからのミラーリングされるデータベースの読み書き可能コピー（すなわちミラーリングされるデータベースのサブセット）を維持します。それらのデータベースは、解析中にデータの変更や追加が可能であることが必要なレポート/ビジネス・インテリジェンス業務のデータ・ソースとして利用されます。
- e. **【SSL/TLS の構成】** – ミラーが TLS を要求するときに、インスタンスでミラーリングに対する有効な TLS 構成を済ませていない場合は、エラー・メッセージと、このリンクが含まれるようになります。手順を完了する前に、このリンクをクリックして、このメンバに対して必要な TLS 構成を作成する必要があります。（TLS 構成の作成方法の詳細は、“[ミラー用 TLS 構成の作成および編集](#)”を参照してください。）[非同期として参加] の手順をキャンセルして、管理ポータルの [TLS 構成] ページに移動することもできます（[システム管理]→[セキュリティ]→[SSL/TLS 構成]）。
- f. **【SSL/TLS 編集】** – ミラーリングに対する有効な TLS 構成が行われているインスタンスの場合、このリンクが [SSL/TLS の構成] の代わりに表示されます。必要な場合は、このリンクを使用して、既存の TLS 構成を編集できます。インスタンスの X.509 識別名も表示されます。

5. **【保存】** をクリックします。

TLS を使用するようにミラーを構成した場合は、“[第 2 のフェイルオーバー・メンバまたは非同期メンバを承認する \(TLS ミラーのみ\)](#)” で説明するように、第 1 のフェイルオーバー・メンバで非同期メンバを承認することで、非同期メンバをミラーに追加するプロセスを完了する必要があります。

注釈 非同期ミラー・メンバの構成には、`MIRROR` ルーチンも使用できます（“[MIRROR ルーチンの使用法](#)”を参照）。ミラーリングが有効化された InterSystems IRIS インスタンスで `MIRROR` ルーチンを実行すると、[ミラー構成] メニューの [非同期メンバとしてミラーに参加]（または [非同期メンバとして他のミラーに参加]）オプションが選択でき、非同期メンバを構成してミラーに追加するための代替手段になります。また、`SYS.Mirror.JoinMirrorAsAsyncMember()` ミラーリング API メソッドを使用して、非同期メンバを構成することもできます。

このセクションで説明する手順を使用して、インスタンスを非同期メンバとして 1 つのミラーに追加した後に、[非同期の編集] ページ（“[非同期メンバのミラー構成の編集](#)”を参照）の [ミラーに参加] ボタンを使用してインスタンスを他のミラーも追加できますが、同じタイプの非同期に限られます。

4.8 ミラーへのデータベースの追加

ミラーに追加できるのは、現在のプライマリ・フェイルオーバー・メンバのローカル・データベースのみです。最初はプライマリ上、その次にバックアップ上、さらにその次に任意の非同期メンバ上で追加します。すべてのミラーリングされるデータベースはジャーナリングされる必要があります。

ミラーリングされるデータベースの同一セットをプライマリおよびバックアップ・フェイルオーバー・メンバの両方、ならびにあらゆる DR 非同期メンバに追加する必要があります。いずれのミラーリングされるデータベースをレポート非同期メンバに追加するかは、レポートのニーズによって異なります。ミラーリングされるデータベースに関連付けられるネームスペースやグローバル/ルーチン/パッケージのマッピングは、すべてのミラー・メンバで同じである必要があります。メンバにはそのデータベースが存在するすべての非同期メンバも含まれます。バックアップ・フェイルオーバー・メンバのミラーリングされるデータベースをマウントおよびキャッチアップして（“[ミラーリングされるデータベースの有効化とキャッチアップ](#)”を参照）、フェイルオーバー時にプライマリを引き継げるようにする必要があります。DR 非同期メンバのミラーリングされるデータベースをマウントおよびキャッチアップして、フェイルオーバー・メンバへの昇格に適するようにする必要があります。

ミラーリングされるデータベースの作成手順（つまり、データが含まれない新規データベースの追加）は、ミラーへの既存データベース追加の手順とは異なります。ミラーリングされるデータベースとして作成されたデータベースでのグローバル操作は、始めからミラー・ジャーナル・ファイルに記録されるので、ミラーはミラー・メンバ全体のデータベースと同期する必要があるデータすべてに対するアクセス権を備えています。しかし、ミラーへ追加される前の既存データベースのグローバル操作は非ミラー・ジャーナル・ファイルに含まれるので、ミラーにはアクセス権がありません。この理由から、既存のデータベースは、ミラーに追加した後にプライマリ・フェイルオーバー・メンバにバックアップし、データベースを置くことになるバックアップ・フェイルオーバー・メンバおよび非同期メンバにリストアする必要があります。これを実行した後は、データベースをアクティブにしてキャッチアップし、プライマリと同じ最新状態にする必要があります。

- ・ [ミラーリングされたデータベースの考慮事項](#)
- ・ [ミラーリングされるデータベースを作成する](#)
- ・ [ミラーへ既存データベースを追加する](#)
- ・ [ミラーリングされたデータベースのアクティブ化/キャッチアップ](#)

4.8.1 ミラーリングされたデータベースの考慮事項

ミラーリング・データベースを作成および追加する際は、以下の点に注意してください。

- ・ ミラーリングできるのは、**IRIS.DAT** ファイル内のデータのみです。外部データ（すなわち、ファイル・システムに保存されているデータ）は、InterSystems IRIS ではミラーリングできません（詳細は、["ミラー構成のガイドライン"](#) を参照してください）。
 - ・ システム・データベース (**IRISSYS**、**IRISLIB**、**IRISLOCALDATA**、**IRISTEMP**、**IRISAUDIT**、および **ENSLIB**) はミラーリングできません。
 - ・ InterSystems IRIS for Health™ または HealthShare® Health Connect を実行している場合：
 - **HSLIB** をミラーリングしないでください。
 - **HSSYS** はミラーリングする必要があります。
 - **HSCUSTOM** をミラーリングして、カスタム・コードを同期された状態に保つことができます。
- 重要** **HSCUSTOM** をミラーリングする場合は、アップグレードを実行する際、アップグレード中は一時的にミラーからこのデータベースを削除して、後でミラーに戻すようにしてください。そうしないと、アップグレード・プロセス後に、**HSCUSTOM** を更新できなくなります。
- ・ ミラーリングされたデータベースにはさらに多くのデータベース・ディレクトリ情報が格納されるので、InterSystems IRIS インスタンス内に構成可能なデータベースの最大数は減ります。詳細は、["データベースの構成"](#) を参照してください。
 - ・ ミラーは、バックアップまたは非同期のミラーされたデータベースの以下のプロパティを、プライマリのデータベースのプロパティと自動的にかつ継続的に同期します。
 - **最大サイズ**
 - **拡張サイズ**
 - **リソース名**
 - **照合**

例えば、プライマリで、ミラーリングされたデータベースの **[最大サイズ]** が増やされた場合、必要に応じて、プライマリと一致するようにその他のメンバ上のミラーリングされたデータベースも自動的に増やされます。非同期で **[最大サイズ]** が減らされた場合、同期によってプライマリ上の値もその値まで自動的に増やされます。データベースのプロパティがプライマリまたは別のミラー・メンバで変更され、そのメンバが切断されている場合、そのメンバがミラーに再

接続したときに自動的に同期されます。これらのデータベースのプロパティの自動同期には、次の 2 つの例外があります。

- 非同期の **【最大サイズ】** プロパティと **【拡張サイズ】** プロパティの値は、同期によって増加することがありますが、減少することはありません。例えば、プライマリ上のデータベースの **【最大サイズ】** が減らされた場合、バックアップのこのプロパティ値は減らされますが、ミラーに属している非同期のこのプロパティ値は減らされません。非同期のデータベースの **【最大サイズ】** がプライマリのものより増やされた場合、この値が同期によってプライマリの値にまで減らされることはありません。
- データベースの **【リソース名】** プロパティは、フェイルオーバー・データベースとしてマークされている指定のデータベースを持つミラー・メンバでプライマリと同期します。実際には、読み書き可能レポート非同期メンバを除くすべてのミラー・メンバに **【リソース名】** が同期することになります。

重要 InterSystems IRIS for Health™ または HealthShare® Health Connect を実行している場合は、“[インターシステムズの医療製品のミラーリングに関する考慮事項](#)” で追加のデータベースの考慮事項を参照してください。

ミラーリングされるデータベース・プロパティの詳細は、“[ミラーされているローカル・データベース・プロパティの編集](#)” を参照してください。

4.8.2 ミラーリングされるデータベースを作成する

ミラーリングされるデータベースを作成するには、以下の手順に従います。

注釈 “DATABASE ルーチンを使用して、ミラーリングされるデータベースを作成することもできます。”[“DATABASE ルーチンを使用したミラーリングされるデータベースの作成”](#) を参照してください。

1. 現在のプライマリ・フェイルオーバー・メンバで、管理ポータルの **【ローカルデータベース】** ページ (**【システム管理】** → **【構成】** → **【システム構成】** → **【ローカルデータベース】**) に移動し、**【新規データベース作成】** ボタンをクリックします。
2. “[ローカル・データベースの作成](#)” の手順に従います。2 番目のパネルで、**【ミラーリングされるデータベース】** に対して **【はい】** を選択して、ミラー内のデータベースの名前を入力します。既定はユーザが提供したローカル・データベース名となります(ミラー・データベース名の先頭文字は、英字またはアンダースコアである必要があり、残りの部分は、英字、ハイフン、およびアンダースコアである必要があります)。ミラー・データベース名では大文字と小文字が区別されないため、2 つの名前を大文字と小文字の違いだけで区別することはできません。既にミラーに含まれているミラー・データベース名を入力した場合、新規データベースをミラーに追加できないので、削除する必要があります(以前のバージョンの InterSystems IRIS で作成しミラーリングされるデータベースの名前は、大文字と小文字の混在状態または小文字で保存されている可能性があります、重複する大文字の名前が付いたデータベースの追加は、これまでどおりに防止されます)。

複数のミラーに属している非同期メンバでは、そのデータベースが属することになるミラーも選択する必要があります。

注釈 **【ミラーされたデータベース】** に対して **【はい】** を選択した場合、**【ジャーナルグローバル】** は自動で **【はい】** に固定されます。

3. 手順によってデータベースが作成されたことを確認し、そのデータベースをプライマリのミラーに追加します。
4. バックアップ・フェイルオーバー・メンバ、およびミラーリングされるデータベースを追加する各非同期メンバで、前述の 3 つのステップに従います。それぞれの他のメンバのミラーリングされるデータベース名として、プライマリの正しいミラー・データベース名を入力するよう注意してください(ローカル・データベース名と一致する必要はありません)。

注釈 プライマリ上のミラーリングされるデータベースとして作成されていないが作成後にミラーに追加した非プライマリ・メンバ上のミラーに新しいデータベースを追加しようとすると、これについて知らせるエラー・メッセージが表示され、操作は完了できません。

重要

ミラーリングされたデータベースの最初のミラー・ジャーナル・ファイルがプライマリから削除されると、別のメンバでデータベースをミラーリングされたデータベースとして作成できなくなります。その代わり、“[ミラーへ既存データベースを追加する](#)”の説明のとおり、プライマリにバックアップを作成し、バックアップまたは非同期でリストアする必要があります。このため、プライマリでデータベースを作成した後は、できるだけすぐにバックアップと非同期メンバでデータベースを作成することをお勧めします。(プライマリでミラー・ジャーナル・ファイルが削除された場合の詳細は、“[ジャーナル・ファイルの削除](#)”を参照してください。)

4.8.3 ミラーへ既存データベースを追加する

以下の手順を使用して、1 つ以上の既存データベースをミラーに追加します。

注釈 SYS.Mirror.AddDatabase() ミラーリング API メソッドは、既存のデータベースをミラーに追加するための代替手段を提供します。

1. 現在のプライマリ・フェイルオーバー・メンバで、管理ポータルの[ローカルデータベース] ページ([システム管理]→[構成]→[システム構成]→[ローカルデータベース])に移動し、[ミラーに追加] ボタンをクリックします。
2. リストのデータベース(システム・データベース以外はまだミラーに存在しない)から、追加するものを選択して、[追加]をクリックします。ミラー内の各データベースの名前を入力する必要があります。既定はユーザが提供したローカル・データベース名となります(ミラー・データベース名の先頭文字は、英字またはアンダースコアである必要があり、残りの部分は、英字、ハイフン、およびアンダースコアである必要があります)。ミラー・データベース名では大文字と小文字が区別されないため、2 つの名前を大文字と小文字の違いだけで区別することはできません。既にミラーに含まれているミラー・データベース名を入力した場合、操作は失敗します(以前のバージョンの InterSystems IRIS で作成しミラーリングされるデータベースの名前は、大文字と小文字の混在状態または小文字で保存されている可能性があります、重複する大文字の名前が付いたデータベースの追加は、これまでどおりに防止されます)。

タスクをバックグラウンドで実行するには、[バックグラウンドで追加を実行]を選択します。5 つ以上のデータベースを選択する場合、タスクはバックグラウンドで自動的に実行されます。手順により、プライマリ上で選択したデータベースがミラーに追加されたことを確認します。

また、個々のデータベースをミラーに追加するには、名前をクリックによりプロパティを編集してから、[ミラー <mirror-name> に追加] リンクをクリックし、その後、[追加] をクリックして、手順を確認することでも行えます(データベースでジャーナリングが有効になっていない場合、このリンクの代わりに[ミラーリングのためにデータベースをジャーナリングする必要があります]と表示されるので、[グローバルジャーナル状態] ドロップダウン・リストから[はい]を選択します)。また、`^MIRROR` ルーチンの [ミラー管理] メニューの [ミラーリングされるデータベースの追加] オプションでも、個々のデータベースを追加できます。どちらの場合も、ローカル名と同じ既定のミラー・データベース名をそのまま使用することも、別の名前を入力することもできます。

注釈 バックアップ・フェイルオーバー・メンバまたは非同期メンバにプライマリ・フェイルオーバー・メンバとは異なるエンディアンがある場合は、“[メンバのエンディアンに関する考慮事項](#)”での手順に従って、データベースをプライマリに追加してから、バックアップまたは非同期メンバに追加する必要があります。以下の手順では、データベースをバックアップまたは非同期メンバ上で追加します。

3. データベースがミラーに追加されたら、プライマリ・フェイルオーバー・メンバ上でバックアップを行います。各種バックアップ技術および関連するリストア手順の詳細は、“[バックアップの方法](#)”、“[バックアップからのリストア](#)”、および“[ミラーリングされたデータベースの考慮事項](#)”を確認してください。

重要

コピーしているデータベースをプライマリで暗号化する場合、暗号化に使用したキーをバックアップ (および非同期 (存在する場合)) で有効化するか、宛先システムで有効化されたキーを使用するようデータベースを変換する必要があります (これについては、“[新しいキーを使用するように暗号化データベースを変換する](#)” で説明しています)。

バックアップからのミラーリングされるデータベースのリストア後 (以下のステップを参照)、そのデータベースが確実に同期するには、プライマリ・フェイルオーバー・メンバ上でのバックアップ時点よりジャーナル・ファイルが使用可能かつオンラインである必要があります。例えば、関連ジャーナル・ファイルが削除されてしまった場合、より最新のバックアップを作成およびリストアする必要があります。ミラー・ジャーナル・ファイルのリストアについての一般事項は“[ミラー・ジャーナル・ファイルのリストア](#)”を参照してください。ミラー・ジャーナル・ファイルの削除については“[ジャーナル・ファイルの削除](#)”を参照してください。

4. バックアップ・フェイルオーバー・メンバおよび接続された各非同期メンバでは、以下の手順を実行します。
 - a. プライマリ・フェイルオーバー・メンバ上で追加したばかりのミラーリングされるデータベースと同じローカル名とデータベース・ディレクトリを持つローカル・データベースがまだ存在しない場合は、それを作成します。
 - b. プライマリ上のミラーリングされるデータベースで作成したバックアップをリストアして、既存のデータベースを上書きします。この手順は、以下のように使用しているリストア方法によって異なります。
 - ・ オンライン・バックアップ・リストア (DBREST ルーチン) – このルーチンは、バックアップおよび非同期メンバのミラーリングされるデータベースを自動的に認識、有効化、およびキャッチアップします。詳細は、“[ミラーリングされるデータベースの考慮事項](#)”を参照してください。

注釈

ミラーリングされるデータベースが非プライマリ・メンバ上でリストアされる場合、自動同期プロセスを開始するためのデータがまだ送信されていない場合があります。必要なデータが 60 秒以内に到着しない場合、プロセスはとりあえず開始されます。データが必要となる前に到着しない場合、それらのデータベースがキャッチアップしないことがあります。ただしその場合は、問題のあるデータベースに関するメッセージが `messages.log` ファイルにロギングされます (データベース作成中は、このプロセスに影響されるデータベースは 1 つのみとなります。ただし、これは複数のデータベースが関係するその他の状況のキャッチアップにも適用されます)。

- ・ 外部バックアップ・リストアまたはコールド (オフライン) バックアップ・リストア – これらのどちらの方法でも、ミラーリングされるデータベースを、バックアップ・フェイルオーバー・メンバまたは非同期メンバ上でリストアし、かつマウントした後に、手動で有効化し、キャッチアップする必要があります。これについては、直後の“[ミラーリングされるデータベースの有効化とキャッチアップ](#)”で説明しています。

前述の手順の代わりとして、既存のデータベースをプライマリ上のミラーに追加した後で、データベースのバックアップとリストアを実行するのではなく、データベースの `IRIS.DAT` ファイルをプライマリからバックアップ・メンバと非同期メンバにコピーすることもできます。この操作には、以下の手順を使用します。

1. バックアップ・メンバまたは各非同期メンバ上にプレースホルダのターゲット・データベースが存在することを確認します。
2. 両方のフェイルオーバー・メンバと各非同期メンバで、ソース・データベースとターゲット・データベースがマウントされていないことを確認します (“[ローカル・データベースの管理](#)”を参照してください)。
3. ミラーリングされる `IRIS.DAT` ファイルをプライマリ・フェイルオーバー・メンバから、バックアップおよび各非同期メンバ上のプレースホルダ・ターゲット・データベースのデータベース・ディレクトリにコピーして、既存の `IRIS.DAT` ファイルを上書きします。

注釈

コピーしているデータベースをプライマリで暗号化する場合、暗号化に使用したキーをバックアップ (および非同期 (存在する場合)) で有効化するか、宛先システムで有効化されたキーを使用するようデータベースを変換する必要があります (これについては、“[新しいキーを使用するように暗号化データベースを変換する](#)”で説明しています)。

- すべてのメンバにデータベースをマウントします。
- バックアップ・フェイルオーバー・メンバおよび非同期メンバ上のミラーリングされるデータベースを有効化してキャッチアップします。詳細は、“[ミラーリングされるデータベースの有効化とキャッチアップ](#)”を参照してください。

注釈 既存のミラーリングされるデータベースを非同期メンバに追加するときに、完全にキャッチアップされているものと仮定すると、プライマリではなく、バックアップ・フェイルオーバー・メンバまたは別の非同期メンバ上でデータベースをバックアップする（または、それらのメンバから **IRIS.DAT** ファイルをコピーする）ことができます。バックアップをリストアする非同期とは異なるデータ・センタにプライマリがある場合など、これは便利な場合があります。ただし、データベースの整合性に高度な信頼性がない限り、メンバをソースとして使用しないでください。

4.8.4 ミラーリングされるデータベースの有効化とキャッチアップ

ミラー・モニタを使用して、バックアップ・フェイルオーバー・メンバおよび非同期メンバ上のミラーリングされるデータベースの有効化またはキャッチアップ、あるいはその両方を実行できます。

“[ミラーへ既存データベースを追加する](#)”でも記載していますが、データを含んでいてミラーリングされる新規追加データベースは、**DBREST** ルーチンの使用を通してプライマリと自動的に同期して、プライマリ・フェイルオーバー・メンバからバックアップをリストアできます。その他の手段を使用する場合、バックアップ・フェイルオーバー・メンバおよび非同期メンバ上で有効化およびキャッチアップする必要があります。

ミラーリングされるデータベースを有効化およびキャッチアップするには、バックアップ・フェイルオーバー・メンバおよび非同期メンバで次の手順を実行します。

- [ミラー・モニタ] ページ ([システム処理]→[ミラー・モニタ]) に移動します。
- 必要に応じて、非同期メンバ上で、実行するデータベースを含むミラーの [詳細] リンクをクリックします。
- “[ミラー・モニタの使用法](#)”で説明しているように、[ミラーリングされるデータベース] リストに、各データベースのステータスが表示されます。ここに表示される可能性のあるステータスの中で特に、[要キャッチアップ] はキャッチアップ処理が必要であることを示し、[要有効化] は有効化とキャッチアップの両方の処理が必要であることを示し、[キャッチアップ実行中] はデータベースで現在キャッチアップ処理が実行中であることを示します。
- 1 つのデータベースに対してのみ処理を実行する場合、[有効化] リンクまたは [キャッチアップ] リンクを選択します。または、アクションの対象として該当するすべてのデータベースのリストから複数のデータベースを選択できるダイアログを開いて、選択したすべてのデータベースに一度にアクションを適用する場合、[アクションを選択してください] ドロップダウンから [有効化] または [キャッチアップ] を選択して、[進む] をクリックします。これを実行する場合、[有効化] および [キャッチアップ] タスクはバックグラウンドで実行されます。[キャッチアップ] を選択すると、[要有効化] ステータスのデータベースと [要キャッチアップ] ステータスのデータベースが表示されます。[有効化] と [キャッチアップ] の両方が、選択したすべての [要有効化] データベースに適用されます。

[ミラーリングされるデータベース] リストを使用して、1 つ以上のミラーリングされたデータベースをマウントまたはディスマウントするか、またはミラーから 1 つ以上のデータベースを削除することもできます。これについては“[ミラーリングされるデータベースをミラーから削除する](#)”で説明します。

注釈 データベース内のエラーが原因でミラーリングされたデータベースをキャッチアップできない場合、影響を受けたデータベースのホスト・メンバがプライマリになっても、そのデータベースはアクティブにはなりません。”[自動フェイルオーバーのルール](#)”で説明しているように、データベースが[開始時にマウントが必要]とマークされている場合は、そのメンバがプライマリになることはなくなります。

^MIRROR ルーチンの[ミラー管理]メニューの[ミラーリングされるデータベースの有効化/キャッチアップ]オプション、および SYS.Mirror.ActivateMirroredDatabase() と SYS.Mirror.CatchupDB() ミラーリング API メソッドは、ミラーリングされるデータベースを有効化/キャッチアップするための代替手段となります。

[ミラーされたデータベース] リスト、管理ポータル の [データベース] ページ (“システム管理ガイド” の “[ローカル・データベースの管理](#)” の章を参照)、または ^DATABASE ルーチン (“[コマンド行セキュリティ管理ユーティリティ](#)” を参照) を使用して、ミラーリングされたデータベースをマウントする場合、マウント操作後にデータベースをキャッチアップするかどうかを選択できます。

並列デジャーナリング (“[並列デジャーナリングの構成](#)” を参照) は、有効になっていて、使用可能なリソースによってサポートされている場合、ミラーリングされるデータベースをキャッチアップするときに使用されます。

4.9 ミラーの削除

ミラーを永久に削除して、プライマリ上のミラーリングされたデータベースをミラーリングされていない状態に戻すには、以下の手順を実行します。

1. “[非同期メンバの編集または削除](#)” の手順 2 の説明に従って、ミラーの非同期メンバを 1 つずつ削除します。この際、提供されているオプションを使用して、[ミラーリングされたデータベースからのミラー属性の削除](#)、およびインスタンスの [ミラー TLS 構成](#) の削除も行います。インスタンスを再起動し、必要に応じてデータベースをバックアップしたら、それらを [削除](#) または [再配置](#) して、稼働中のミラーリングされていないデータベースの場所 (前のプライマリ上) に関して混乱が生じないことを確認します。
2. “[フェイルオーバー・メンバの編集または削除](#)” の手順 2 の説明に従って、現在のバックアップ・フェイルオーバー・メンバを削除します。この際、提供されているオプションを使用して、[ミラーリングされたデータベースからのミラー属性の削除](#)、およびインスタンスの [ミラー TLS 構成](#) の削除も行います。インスタンスを再起動し、必要に応じてデータベースをバックアップしたら、それらを [削除](#) または [再配置](#) して、稼働中のミラーリングされていないデータベースの場所 (前のプライマリ上) に関して混乱が生じないことを確認します。
3. “[フェイルオーバー・メンバの編集または削除](#)” の手順 3 の説明に従って、プライマリ・フェイルオーバー・メンバを削除します。この際、提供されているオプションを使用して、[ミラーリングされたデータベースからのミラー属性の削除](#)、およびインスタンスの [ミラー TLS 構成](#) の削除も行います。以前のミラーリングされたデータベースをそのインスタンス上の稼働データベースとして保持しない場合は、2 回目に [ミラーの構成を削除] を使用した後、必要に応じてバックアップを行ってからそれらを [削除](#) または [再配置](#) します。

重要 分散キャッシュ・クラスタ・データ・サーバを含むミラーを削除して、それをミラーリングされた状態からミラーリングされていない状態に変更する場合は、すべてのアプリケーション・サーバ上でリモート・データ・サーバとしてそれを削除し、[ミラー接続] チェック・ボックスのチェックを外した状態で再度それを追加する必要があります。同様に、ミラーを作成して、データ・サーバをミラーリングされていない状態からミラーリングされた状態に変更する場合、同じことを行う必要がありますが、今度は [ミラー接続] チェック・ボックスのチェックを付けた状態で再度それを追加します。詳細は、“[ミラーへのアプリケーション・サーバ接続の構成](#)” を参照してください。

4.10 ミラー・メンバの編集または削除

以下の手順は、ミラーをまとめて削除する処理も含め、ミラー・メンバでミラー構成を編集または削除する方法と、ミラー構成を削除しない場合にミラーからデータベースを削除する方法を示しています。

- ・ [レポート非同期ミラー・メンバの FailoverDB フラグのクリア](#)
- ・ [ミラー・メンバ削除時のミラーリングされるデータベースの属性の削除](#)
- ・ [非同期メンバの編集または削除](#)
- ・ [フェイルオーバー・メンバの編集または削除](#)
- ・ [ミラーリングされるデータベースのミラーからの削除](#)

注釈 ^①MIRROR ルーチンの **[ミラーの構成]** メニューにあるいくつかのオプションでは、ミラー構成の編集のための代替手段を用意しています。使用可能な具体的なオプションは、ルーチンがフェイルオーバー・メンバまたは非同期メンバのいずれにより使用されるかによって異なります。

4.10.1 レポート非同期ミラー・メンバの FailoverDB フラグのクリア

“[非同期ミラー・メンバ](#)” で説明しているように、非同期メンバには以下の 3 種類があります。

- ・ 災害復旧 (DR) プライマリ上のミラーリングされるデータベースのすべての読み取り専用コピーを維持します。フェイルオーバー・メンバに昇格する条件を満たしています (詳細は、“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)” を参照)。
- ・ 読み取り専用のレポートミラーリングされるデータベースの読み取り専用コピーを維持します。フェイルオーバー・メンバに昇格する条件を満たしていません。
- ・ 読み書き可能なレポートミラーリングされるデータベースの読み書き可能なコピーを維持します。フェイルオーバー・メンバに昇格する条件を満たしていません。

ミラーリングされるデータベースが DR または読み取り専用のレポート非同期に追加されると、そのデータベースは読み取り専用としてマウントされ、プライマリでデータベースを作成したときに設定される **[FailoverDB]** フラグは、以下の目的で非同期のコピーに設定されたままになります。

- ・ データベースが読み取り専用の状態を維持するようにして、プライマリ上のデータベースの正確なコピーであることを保証します (デジャーナリングがキャッチアップされることが前提)。
- ・ DR 非同期メンバがフェイルオーバー・メンバに昇格したときに、データベースがミラー内でプライマリ・コピーになれることを示します。DR 非同期メンバは、ミラー内のすべてのデータベースを含んでいて、そのすべてのデータベースに **[FailoverDB]** フラグが設定されている場合にのみ昇格できます。

その一方で、ミラーリングされるデータベースが、読み書き可能なレポート非同期に追加されるときには、データベースを読み書き可能でマウントできるように、**[FailoverDB]** フラグがクリアされます。ミラーリングされるデータベースは、**[FailoverDB]** フラグがクリアされていると、ミラーのプライマリ・コピーとして使用できなくなります。

DR 非同期の場合、**[FailoverDB]** フラグはクリアできません。ただし、レポート非同期では、このフラグを手動でクリアできます。

読み取り専用のレポート非同期の場合、**[FailoverDB]** フラグをクリアすると、データベースは読み書き可能に変更されますが、一般に、これは推奨されません。そのため、非同期のタイプを **[災害復旧 (DR)]** から **[読み取り専用のレポート]** に変更する場合を含めて (“[非同期メンバの編集または削除](#)” を参照)、ほとんどの場合、読み取り専用レポート非同期のすべてのデータベースでは、**[FailoverDB]** フラグを設定したままにできます。

非同期メンバのタイプを [災害復旧 (DR)] または [読み取り専用のレポート] から [読み書き可能なレポート] に変更する場合は、すべての [FailoverDB] フラグをクリアするオプションが使用可能です。ミラーリングされるデータベースの [FailoverDB] フラグは読み取り専用を維持する必要があるため、一般に、このオプションを使用することになります。ただし、読み書き可能なレポート非同期で、1 つ以上のミラーリングされるデータベースを読み取り専用のままにする必要がある場合は、[ミラーリングされるデータベース] リストにある個別の [フラグのクリア] リンクを使用して、一部のデータベースを読み書き可能にし、それ以外のものを読み取り専用のままにすることができます。

タイプを変更した後の非同期メンバに追加したデータベースは、前述したように、メンバの新しいタイプに従ってマウントされ、フラグが設定されます。[FailoverDB フラグのクリア] ボタンを使用すると、どちらのタイプのレポート非同期でも、すべてのデータベースから、いつでもフラグをクリアできます。

[FailoverDB] フラグを手動で設定することはできません。このフラグは、ミラーリングされるデータベースが DR または読み取り専用のレポート非同期に追加されるときにのみ設定されます。

4.10.2 ミラー・メンバ削除時のミラーされるデータベースの属性の削除

メンバをミラーから削除するときには、そのミラーに属するミラーリングされるデータベースからミラー属性を削除するためのオプションが常に提供されます。その結果は、以下のとおりです。

- ミラー属性を保持しておいて、後で InterSystems IRIS インスタンスをそのミラーにリストアする場合、データベースはミラーに自動的に追加されますが、キャッチアップして同期するにはそれらのデータベースを有効化する必要があります (“ミラーリングされるデータベースの有効化とキャッチアップ” を参照してください)。

ただし、ミラー属性を維持すると、以下のいずれかを先に実行しない限り、そのデータベースは削除できなくなります。

- メンバを削除した同じミラーにメンバを復元する (メンバがプライマリ・フェイルオーバー・メンバであった場合、ミラーは存在しなくなっているため、この方法は選択できません)。そうすると、1 つ以上のデータベースをミラーから削除できるようになり (“ミラーリングされるデータベースのミラーからの削除” を参照)、必要な場合は消去します。
 - ^MIRROR ルーチンの [1 つ以上のミラーリングされるデータベースの削除] オプションを使用して (“^MIRROR ルーチンの使用法” を参照)、1 つ以上のデータベースからミラー属性を削除し、必要に応じてデータベースを削除する。
- ミラー属性を削除すると、そのデータベースは永久にミラーリングされなくなり、ローカル・データベースのように使用できるようになります。インスタンスがミラー・メンバとして再び参加した後にミラーに戻す場合、最初にデータベースを既存データベースとしてミラーに追加するための手順を使用する必要があります。

バックアップまたは非同期メンバのミラーから個別のデータベースを削除すると、ミラーリングされるデータベースの属性は、自動的に削除されます。

4.10.3 非同期メンバの編集または削除

- [非同期構成の編集] ページ ([システム管理]→[構成]→[ミラー設定]→[非同期の編集]) に移動します。
- [ミラー構成を削除する] ボタンを使用して、DR 非同期をミラーから削除するか、レポート非同期をすべての所属先ミラーから削除して、インスタンスのミラー構成全体を削除します。(単一ミラーからレポート非同期を削除するには、この手順の後半で述べる [ミラーから脱退] リンクを使用します)。

メンバ上のミラーリングされるデータベースからミラー属性を削除するためのオプションが選択できます。この選択の判断に関する詳細は、“ミラー・メンバ削除時のミラーリングされるデータベースの属性の削除” を参照してください。

また、インスタンスの TLS 構成を削除するというオプションを選択することもできます (“TLS セキュリティを使用したミラー通信の保護” を参照)。

[ミラー構成を削除する] ボタンを使用して、インスタンスのミラー構成全体を削除した後は、InterSystems IRIS を再起動する必要があります。

注釈 `^MIRROR` ルーチン (“[^MIRROR ルーチンの使用法](#)” を参照) の [ミラー構成] メニューにある [ミラー構成を削除する] オプションおよび `SYS.Mirror.RemoveMirrorConfiguration()` API 呼び出しでは、非同期メンバのミラー構成全体を削除するための代替手段が提供されます。

3. [ミラーに参加] ボタンを使用して、レポート非同期メンバを他のミラーに追加します (最大で 10 個まで所属できます)。非同期メンバを最初のミラーに追加するための手順は、“[非同期ミラー・メンバを構成する](#)” で説明したものと同じですが、メンバ名と非同期タイプ (読み取り専用または読み書き可能) の変更は不可能です。このボタンは、DR 非同期メンバには使用できません。別のミラーに参加するには、まず、後述の手順で説明するように、[非同期メンバのシステム・タイプ] を変更しておく必要があります。
4. “[レポート非同期ミラー・メンバの FailoverDB フラグのクリア](#)” の説明にあるように、[FailoverDB フラグのクリア] ボタンを使用すると、読み取り専用のレポート非同期にあるすべてのミラーリングされるデータベースの [FailoverDB] フラグをクリアできます。または、非同期システム・タイプを [災害復旧 (DR)] から [読み書き可能] または [読み取り専用のレポート] に変更した後で、このボタンを使用するとフラグをクリアできます。
5. [ミラー・メンバ情報] セクションの以下の設定は、編集集中の非同期メンバに対して変更可能ですが、ミラー・メンバ名は例外です。1 つ以上変更した後に、[保存] をクリックします。
 - ・ [ミラー・メンバ名] – 非同期メンバが最初のミラーへ参加したときに、与えられる名前。変更はできません。
 - ・ [非同期メンバのシステム・タイプ] – ドロップダウン・リストを使用して、非同期メンバのタイプを変更できます。以下の条件が適用されます。
 - [災害復旧 (DR)] から [読み書き可能なレポート] に変更すると、すべてのミラーリングされるデータベースに対して、[FailoverDB] フラグをクリアするよう促されます。これについては、“[レポート非同期ミラー・メンバの FailoverDB フラグのクリア](#)” で説明しています。
 - [読み書き可能なレポート] から [読み取り専用のレポート] (またはその逆) へ変更した場合、変更はレポート非同期メンバが属するすべてのミラーに対して施されます。
 - [読み書き可能] または [読み取り専用のレポート] から [災害復旧 (DR)] に変更することはできません。ただし、以下の条件がすべて当てはまる場合は変更できます。
 - ・ ジャーナル暗号化を使用している場合、非同期はフェイルオーバー・メンバと同じジャーナル暗号化キーを使用していること (“[ミラー内ジャーナル暗号の有効化](#)” を参照)。
 - ・ [FailoverDB] フラグが、ミラーリングされるデータベースのすべてに設定されていること (一度クリアすると、このフラグはリセットできません。これに対処するには、[FailoverDB] が設定されている別のメンバから取ったデータベースのコピーを代わりに使用できます)。
 - ・ メンバが別のミラーに属していないこと。
 - ・ ISCAgent を実行していること (“[ISCAgent の開始および停止](#)” を参照)。

デジャーナル・フィルタが非同期に対して設定されている場合 (“[レポート非同期に対するデジャーナル・フィルタの使用](#)” を参照)、そのフィルタは、[非同期メンバのシステム・タイプ] を [災害復旧 (DR)] に変更するときに削除されます。

重要

レポート非同期を DR 非同期に変換する前に、昇格が必要な災害が発生した場合にそのメンバがフェイルオーバー・メンバになる準備が整っていることを確認してください(“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”を参照)。このときに確認する事項は次のとおりです。

- ・ 対象のメンバに、ミラーリングされたデータベースがすべて含まれている。
- ・ その他すべてのメンバから対象のメンバに接続することができる(“[ミラーリング通信](#)”および“[ミラーリング・アーキテクチャとネットワーク構成のサンプル](#)”で説明しています)。
- ・ 対象のメンバに、プライマリとして動作するために必要なリソースが含まれている。
- ・ **[ミラー・ジャーナル・ファイルの保有]** (レポート非同期のみ) – ミラー・ジャーナル・ファイルがデジャーナルされた直後にミラー・ジャーナル・ファイルを削除するか、またはインスタンスのローカル削除ポリシーに従ってミラー・ジャーナル・ファイルを削除するか。この設定は、レポート非同期にのみ使用できます。ミラー・ジャーナル・ファイルを削除する方法の詳細は、“[ミラー・ジャーナル・ファイルの削除](#)”を参照してください。
- ・ **[SSL/TLS構成]** – TLS が要求される場合は(“[TLS セキュリティを使用したミラー通信の保護](#)”を参照)、X.509 識別名 (DN) と共に **[SSL を検証]** ボタンが表示されます。このボタンを使用すると、編集中の非同期と通信可能な、現在のすべてのミラー・メンバの TLS 証明書を検証できます。有効でない証明書があった場合、情報メッセージが表示されます(証明書は、`^Mirror` ルーチンを使用して検証することもできます)。

ミラーで TLS を使用していないときに、ミラーに TLS を追加する場合は、TLS を構成するための **[SSL/TLS]** リンクを使用できます(“[フェイルオーバー・メンバの編集または削除](#)”を参照)。

注釈 SYS.Mirror.UpdateMirrorSSL() ミラーリング API メソッドおよび `^SECURITY` ルーチンを使用して、ミラー・メンバの TLS 設定を更新することもできます。

6. **[この非同期メンバが属するミラー]** リストにより、非同期メンバとしてインスタンスが属するすべてのミラーが表示されます。各エントリには、変更のための 3 つのリンクがあります。

- ・ **[ミラー名] – [名前]** 列に表示されるミラーの名前をクリックすると、**[ミラーの編集]** ダイアログが開き、ミラーのすべてのメンバのインスタンス・ディレクトリおよびネットワーク・アドレスが表示されます(“[ミラー・メンバのネットワーク・アドレス](#)”を参照)。

現時点で非同期がミラーに接続されている場合、非同期のスーパーサーバ・ポートを除いて、表示されているネットワーク情報は一切変更できません。非同期メンバが切断されているときに、プライマリのネットワーク情報を更新した場合、ここでプライマリの情報を更新できるので、必要に応じて非同期を再接続できます。ミラー・メンバのネットワーク・アドレスの更新に関する重要な情報は、“[ミラー・メンバのネットワーク・アドレスの更新](#)”を参照してください。

- ・ **[ミラーから脱退]** – クリックしたリンク先のミラーから、およびそのミラーのみから非同期メンバを削除します。(DR 非同期の場合、これは所属ミラーのみになります。)

非同期メンバ上のミラーリングされるデータベースからミラー属性を削除するためのオプションが提供されます。このときの判断に関する詳細は、“[ミラーリングされるデータベースの属性の保持または削除](#)”を参照してください。

注釈 `^MIRROR` ルーチン ("[^MIRROR ルーチンの使用法](#)" を参照) の [ミラー構成] メニューにある [ミラーからこのメンバを削除する] オプションおよび `SYS.Mirror.RemoveOneMirrorSet()` API 呼び出しでは、非同期メンバをミラーから削除するための代替手段が提供されます。フェイルオーバー・メンバ上で [ミラーの編集] ページの [他のミラー・メンバを削除する] ボタンを使用して、ミラーから非同期メンバを削除することもできます。

あらゆる非同期メンバで、ミラーリングを一時停止できます (1 つのレポート非同期が複数ミラーに所属している場合は、ミラーごとに一時停止できます)。詳細は、"[バックアップおよび非同期メンバのミラーリングの停止](#)" を参照してください。

- ・ [デジャーナル・フィルタの編集] (レポート非同期のみ) – 非同期に対するデジャーナル・フィルタを設定または削除できます。詳細は、"[レポート非同期に対するデジャーナル・フィルタの使用](#)" を参照してください。

7. [ミラーリングされるデータベース] リストにより、非同期メンバのミラーリングされるデータベースがすべて表示されます。インスタンスが DR 非同期メンバの場合は、インスタンスにミラーのフェイルオーバー・メンバ上のミラーリングされるデータベースをすべて含める必要があります。また、それぞれに [FailoverDB] フラグが設定される必要があります。
8. TLS を使用するミラーで、[保留中の DN 更新を承認] を選択し (表示されている場合)、プライマリからの保留中の DN 更新を承認して、非同期がプライマリとの通信を継続できるようにします。DN 更新の承認の詳細は、"[X.509 DN 更新の承認 \(TLS のみ\)](#)" を参照してください。

4.10.4 フェイルオーバー・メンバの編集または削除

1. [ミラーの編集] ページ ([システム管理]→[構成]→[ミラー設定]→[ミラーの編集]) に移動します。
2. バックアップ・フェイルオーバー・メンバで [ミラー構成を削除する] ボタンを使用すると、ミラーから削除され、InterSystems IRIS インスタンスのミラー構成全体が削除されます。

ミラーからフェイルオーバー・メンバを削除するときには、そのメンバ上のミラーリングされるデータベースからミラー属性を削除するオプションが選択できます。この選択の判断に関する詳細は、"[ミラー・メンバ削除時のミラーリングされるデータベースの属性の削除](#)" を参照してください。これは、プライマリ・フェイルオーバー・メンバを削除することで、ミラーを永久に削除する場合は特に重要です。

バックアップでは、インスタンスの TLS 構成を削除するオプションを選択することもできます ("[TLS セキュリティを使用したミラー通信の保護](#)" を参照)。

プライマリで [他のミラー・メンバを削除する] ボタンを使用して、ミラーからバックアップまたは非同期を削除することもできます。バックアップで [他のミラー・メンバを削除する] ボタンを使用すると、ミラーから非同期を削除できます。

[ミラー構成を削除する] ボタンまたは [他のミラー・メンバを削除する] ボタンを使用して、非同期またはバックアップ・メンバのミラー構成全体を削除した場合は、InterSystems IRIS を再起動する必要があります。

注釈 `^MIRROR` ルーチン ("[^MIRROR ルーチンの使用法](#)" を参照) の [ミラー構成] メニューにある [ミラー構成を削除する] オプションおよび `SYS.Mirror.RemoveMirrorConfiguration()` API 呼び出しでは、フェイルオーバー・メンバのミラー構成全体を削除するための代替手段が提供されます。

バックアップ・フェイルオーバー・メンバのミラーリングは一時的に停止できます。詳細は、"[バックアップおよび非同期メンバのミラーリングの停止](#)" を参照してください。

3. ミラーからプライマリ・フェイルオーバー・メンバを削除して、ミラー全体を削除するには、以下の手順を使用します (プライマリがミラー内に残された最後のメンバである場合にのみ実行できます)。
 - a. [ミラーの編集] ページの [ミラー構成を削除する] ボタンを使用します。ダイアログが表示され、インスタンスの [JoinMirror] フラグをクリアできます。
 - b. フラグをクリアした後で、インスタンスを再起動します。

- c. [ミラーの編集] ページに移動して、[ミラー構成を削除する] ボタンを再び使用します。

重要 分散キャッシュ・クラスタ・データ・サーバを含むミラーの削除に関する重要な情報は、“[ミラーの削除](#)”を参照してください。

4. [ミラー情報] セクションで [ミラー名] を編集することはできません。それ以外の設定は、プライマリ・フェイルオーバー・メンバでのみ変更できます。

- ・ [SSL/TLS 使用] – ミラー作成時に TLS セキュリティを選択しなかった場合 (“[TLS セキュリティを使用したミラー通信の保護](#)”を参照)、以下の手順に従ってミラーに TLS セキュリティを追加できます。
 - a. プライマリ、バックアップ、およびすべての非同期 (存在する場合) を含む各ミラー・メンバで、ミラーを編集し、[SSL/TLS 使用] チェック・ボックスの右にある [SSL/TLS の構成] リンクをクリックし、“[ミラー用 TLS 構成の作成および編集](#)” セクションの説明に従って、メンバに対してミラーの TLS 構成を作成してください。(リンクが [SSL/TLS を設定] ではなく [SSL/TLS を編集] である場合、構成は既に存在するため、そのメンバに対しては作成する必要はありません。)
 - b. プライマリでミラーを編集し、[SSL を検証] ボタンをクリックします。これにより、編集中のフェイルオーバー・メンバが通信できる現在のすべてのミラー・メンバの証明書を検証できます。(証明書は、`MIRROR ルーチン`を使用して検証することもできます)。無効な証明書がある場合、通知メッセージが表示されます。構成を確認し、必要に応じて証明書を置き換えてください。それ以外の場合は、次のステップに進みます。
 - c. [SSL/TLS を使用] チェック・ボックスにチェックを付けて、[保存] をクリックします。
 - d. “[第 2 のフェイルオーバー・メンバまたは非同期メンバを承認する \(TLS ミラーのみ\)](#)” に説明されているとおり、バックアップおよび非同期のメンバを承認します。

注釈 この手順で TLS セキュリティを追加する際、ミラーが稼働している必要はありません。

SYS.Mirror.UpdateMirrorSSL() ミラーリング API メソッドおよび `SECURITY ルーチン`を使用して、ミラー・メンバの TLS 設定を更新することもできます。

- ・ **アービターを使用** – ミラー作成時にアービターを構成しなかった場合は、この設定をプライマリで選択し、アービターとして構成するシステムのホスト名または IP アドレス、およびその ISCAgent プロセスで使用するポート (既定では 2188) を入力することで、アービターを構成できます。アービターに関する詳細は、“[自動フェイルオーバーのメカニズム](#)”、“[ミラーの可用性を最適化するためのアービターの配置](#)”、および“[アービターのインストール](#)”を参照してください。
- ・ **仮想 IP を使用** – このチェック・ボックスにチェックを付ける、または外すことにより、仮想 IP アドレスを使用するかどうかを変更します。[仮想 IP を使用] を選択した場合、IP アドレス、クラスレスのドメイン間ルーティング (CIDR) マスク、およびネットワーク・インタフェースを与える必要があります (提供済みの場合は変更可能)。

重要 VIP 構成前の要求事項や重要考慮事項の詳細は、“[ミラー仮想 IP \(VIP\) の構成](#)”を参照してください。

- ・ **サービス品質のタイムアウト (msec)** – フェイルオーバー・メンバが、アクションを実行する前に、他のフェイルオーバー・メンバからの応答を待機する最大時間 (ミリ秒単位)。さらに、フェイルオーバー・メンバの応答をアービターが待機する場合にも適用されます。詳細は、“[サービス品質 \(QoS\) タイムアウトの構成](#)”を参照してください。この設定は、プライマリ・フェイルオーバー・メンバでのみ変更できます。
- ・ **[フェイルオーバー・メンバの圧縮モード]、[非同期メンバの圧縮モード]** – プライマリからバックアップ・メンバおよび非同期メンバそれぞれにジャーナル・データを転送する前にデータを圧縮するかどうかと、それぞれに使用する圧縮タイプを指定する設定を変更します。詳細は、“[ミラー・トラフィックの圧縮](#)”を参照してください。一方または両方の圧縮設定を変更すると、その影響を受けるすべてのメンバ (バックアップや非同期) のミラー接続が再起動され、新しい圧縮モードがすぐに使用できるようになります。

- ・ **[並列デジャーナリングを許可する]** – 並列デジャーナリングを有効にする対象をフェイルオーバー・メンバおよび DR 非同期にする (既定) か、レポート非同期を含むすべてのメンバにするか、またはフェイルオーバー・メンバのみにするかを指定する設定を変更します。並列デジャーナリングではミラーのスループットが向上しますが、複数のデータベースが関与するクエリまたはレポートの結果の一貫性が損なわれる可能性が若干高まる場合があります。詳細は、“[並列デジャーナリングの構成](#)”を参照してください。

5. **[ミラー・メンバ情報]** セクションには、各ミラーメンバのメンバ名とタイプ、インスタンス・ディレクトリ、および[ネットワーク・アドレス](#)が表示されます。プライマリで、メンバ名をクリックして、そのメンバのネットワーク情報を更新します (メンバのスーパーサーバ・ポートを除きます。このポートは、ローカルに更新する必要があります。詳細は、“[ミラー・メンバのネットワーク・アドレスの更新](#)”を参照してください)。

現時点でバックアップがミラーに接続されている場合、ネットワーク情報は一切変更できません (バックアップのスーパーサーバ・ポートを除く)。バックアップを切断して、プライマリのネットワーク情報を変更した場合は、ここでプライマリの情報を更新できるので、必要に応じてバックアップが再接続できます。ミラー・メンバのネットワーク・アドレスの更新に関する重要な情報は、“[ミラー・メンバのネットワーク・アドレスの更新](#)”を参照してください。

6. TLS を使用するミラー内のプライマリで、**[保留中の新しいメンバの承認/拒否]** リンクを選択し (表示されている場合)、新しいメンバを承認して、ミラーに参加できるようにします。または、**[保留中の DN 更新の承認/拒否]** リンクを選択し (表示されている場合)、その他のメンバの DN 更新を承認して、ミラーの通信を継続できるようにします。バックアップで、**[保留中の DN 更新の承認]** を選択し (表示されている場合)、プライマリからの保留中の DN 更新を承認して、バックアップがプライマリとの通信を継続できるようにします。DN 更新の承認の詳細は、“[X.509 DN 更新の承認 \(TLS のみ\)](#)”を参照してください。
7. **[保存]** をクリックします。

注釈 プライマリの **[ミラーの編集]** ページにある **[新規非同期メンバの追加]** ボタンは、他のインターシステムズ製品で使用するために予約されています。今回のバージョンの InterSystems IRIS では、このボタンを使用しないでください。

4.10.5 ミラーリングされるデータベースをミラーから削除する

ミラー・モニタを使用して、データベースをミラーから削除することにより、データベースをミラーリングされる状態から、ミラーリングされないローカル用に変換できます (ミラー・モニタに関する詳細は、“[ミラーの監視](#)”を参照してください)。

注釈 あるいは、ミラーリングされるデータベースをミラーから複数削除するには、`^MIRROR` ルーチン (“[MIRROR ルーチンの使用法](#)”を参照) の **[ミラー管理]** メイン・メニュー・リストから **[ミラーリングされるデータベースの削除]** オプションを選択するか、`SYS.Mirror.RemoveMirroredDatabase()` API 呼び出しを使用できます。

非同期メンバ上でミラーからデータベースを削除しても、フェイルオーバー・メンバは影響を受けません。データベースは機能しているミラーの一部として残ります。ただし、データベースをフェイルオーバー・メンバから削除した場合、データベースがミラーリングされる他のフェイルオーバー・メンバや非同期メンバからもデータベースを削除する必要があります。データベース全体をミラーから削除するには、プライマリ・フェイルオーバー・メンバ、バックアップ・フェイルオーバー・メンバ、任意の非同期メンバの順でデータベースの削除を行います。

重要 プライマリでミラーからデータベースを削除すると、データベースは完全に削除されます。プライマリで、ミラーリングされるデータベースを削除して、後でミラーに戻すには、最初に[既存のデータベースをミラーに追加する](#)ために使用した手順が必要になります。

データベースをミラーから削除するには、いずれかのフェイルオーバー・システムで次の手順を実行します。

1. プライマリ・フェイルオーバー・メンバの **[ミラー・モニタ]** ページ (**[システム処理]**→**[ミラー・モニタ]**) に移動します。
2. **[ミラーリングされるデータベース]** リスト内で、ミラーから削除するデータベースの行の **[削除]** をクリックします。

一度に複数のデータベースを削除する場合は、[アクションを選択してください]ドロップダウンから[削除]を選択し、[検索]をクリックしてダイアログを開き、そこで複数のミラーリングされたデータベースを選択してすべてを一度に削除できます。

4.11 ミラーでのマネージド・キー暗号化の使用法

“[暗号化ガイド](#)”に説明されているように、個々の InterSystems IRIS データベースを暗号化して保護することができます。任意の InterSystems IRIS インスタンスでジャーナル・ファイルの暗号化を有効にすることもできます。以下のセクションでは、ミラーでこれらの機能を使用する方法を説明します。

- ・ [ミラーリングされたデータベースの暗号化](#)
- ・ [ミラー内ジャーナル暗号の有効化](#)

4.11.1 ミラーリングされたデータベースの暗号化

ミラー・メンバ上のデータベース暗号化には、いずれのシステムでも準備が必要ですが、データベース暗号化についてはミラーリングに関する特定の要件はありません。ただし、最高度のセキュリティを実施するには、すべてのミラー・メンバ上のミラーリングされたデータベースを暗号化することをお勧めします。この理由から、別のメンバを暗号化せずにプライマリ上で暗号化されているミラーリング・データベースをそのメンバに追加する場合、セキュリティ警告がメッセージ・ログに送信されます。

“[ミラー構成のガイドライン](#)”で説明しているように、フェイルオーバー・メンバの平等性のベスト・プラクティスに基づいて、所定のデータベースは通常、両方のフェイルオーバー・メンバおよび DR 非同期メンバ（フェイルオーバーに昇格する可能性のあるメンバ）と同じ暗号化キーを使用して暗号化されます。

少なくとも 1 つの暗号化キーが有効化されると、新規作成するいずれのデータベースでも暗号化することを選択できます。したがって、“[ミラーリングされるデータベースを作成する](#)”で示している手順を使用する場合は、各ミラー・メンバでデータベースを作成する際に暗号化オプションを選択します。“[ミラーへ既存データベースを追加する](#)”で説明しているように既存のデータベースをプライマリ上のミラーに追加する場合に、そのデータベースが暗号化されているときは、追加先の各メンバで暗号化に使用されたキーを有効にするか、またはデータベースの追加後に各ミラー・メンバ上でデータベースを新しいキーに変換する必要があります。後者を実行する手順については、“[新しいキーを使用するように暗号化データベースを変換する](#)”を参照してください。（この手順を使用して、1 つ以上の既存の暗号化されたミラーリング済みデータベースを新しい暗号化キーに切り替えることや、データベースから暗号化を削除することもできます）。

既存のミラーのフェイルオーバー・メンバで 1 つ以上の暗号化されていないミラーリング済みデータベースを暗号化するには、以下の手順を使用します。

1. “[キー・ファイル内のキーの管理](#)”の説明に従って、両方のフェイルオーバー・メンバで使用するために暗号化キーをロードして有効にします。
2. バックアップでは、以下を実行します。
 - a. “[バックアップおよび非同期メンバのミラーリングの停止](#)”の説明に従って、ミラーリングを停止します。
 - b. “[暗号化されていないデータベースを暗号化データベースに変換する](#)”の説明に従って、各データベースを暗号化します。
 - c. ミラーリングを開始します。
 - d. “[ミラーリングされるデータベースのステータス](#)”の説明に従って、[ミラー・モニタ] ページ ([システム処理]→[ミラー・モニタ])に進み、ミラーリングされたすべてのデータベースのステータスが[デジャーナリング]になるまで待機します。

- iris stop コマンドを使用して、プライマリを適切にシャットダウンし ("[InterSystems IRIS インスタンスの制御](#)" を参照)、ミラーがフェイルオーバーして、バックアップが新しいプライマリになるようにします。
- プライマリを再起動します。プライマリがバックアップになると、前の手順で元のバックアップについて説明した手順に従って、同じキーを使用して同じデータベースを暗号化します。
- 現在のバックアップをシャットダウンして、元のプライマリがもう一度プライマリになるようにします。
- 元のバックアップを再起動して、これがもう一度バックアップになるようにします。

非同期メンバで、ミラーリングされたデータベースを暗号化するには、前の手順でバックアップについて説明した手順に従って、ミラーリングを停止し、データベースを暗号化し、ミラーリングを開始します。ベスト・プラクティスは、フェイルオーバーに昇格する可能性のある DR 非同期上のフェイルオーバー・メンバで使用されたキーを使用することです。

4.11.2 ミラー内ジャーナル暗号の有効化

ミラー・メンバでジャーナル暗号化を有効にする場合、次の 3 つの重要な考慮事項に留意してください。

- ミラーが TLS セキュリティを要求しない場合、フェイルオーバー・メンバおよび DR 非同期でジャーナル・ファイルの暗号化を有効にすることはできません。
- ジャーナル暗号化がプライマリで有効化されている場合は、ミラーに属しているレポート非同期でも有効化されている必要があります。さらに、バックアップ非同期および DR 非同期も同様にジャーナル暗号化を有効化することが最善策になります。これにより、フェイルオーバーまたは [DR 昇格](#) の発生時に、ジャーナル暗号化の強制実施が継続されます。
- フェイルオーバー・メンバ間または DR 非同同期間のジャーナル暗号化には、あるメンバで有効化されるジャーナル暗号化に使用する暗号化キー（ただし、必ずしもジャーナル暗号化に使用されている必要はありません）が別のメンバで、受信したジャーナル・ファイルの解読に（必要に応じて）使用するために必要になります。具体的には以下の処理を行います。
 - ジャーナル暗号化がプライマリで有効化されている場合、プライマリでジャーナル暗号化に使用されているキーは、バックアップおよびすべての DR 非同期にロードされていて有効化されている必要があります。（プライマリのジャーナル暗号化キーがアクティブ化されていないレポート非同期が [DR 非同期に変更されると](#)、警告が生成されます。その非同期は一時的にミラーに接続された状態を維持できますが、次の再接続ができなくなります。キーがアクティブ化されている場合を除いて、これが必要になります。）
 - バックアップまたは DR 非同期でジャーナル暗号化がアクティブ化されている場合は、そのメンバでジャーナル暗号化に使用されているキーはプライマリにロードされていてアクティブ化されている必要があります。

繰り返しになりますが、フェイルオーバーまたは昇格の準備をする際の最善策として、フェイルオーバー・メンバであるメンバ（プライマリ、バックアップ）またはフェイルオーバー・メンバになる可能性のあるメンバ（DR 非同期）にジャーナル暗号化キーが指定されている場合、そのようなメンバのすべて（複数の DR 非同期を含む）に、このキーをロードしておく必要があります。

注釈 レポート非同期のみでジャーナル暗号化を有効にする場合、ミラーで TLS セキュリティを要求する必要はなく、唯一の暗号化キー要件は、有効化する各レポート非同期でジャーナル暗号化用のキーを選択することです。

以下は、フェイルオーバー・メンバ A（現在のプライマリ）と B（現在のバックアップ）、DR 非同期 D、およびレポート非同期 R で構成されるミラーでジャーナル暗号化を有効にする手順を説明しています。

- 現時点でミラーに TLS セキュリティが必要ない場合は ("[TLS セキュリティを使用したミラー通信の保護](#)" を参照)、“[フェイルオーバー・メンバの編集または削除](#)” で説明している手順を使用して、そのようにミラーを構成します。
- A、B、および D の暗号化ジャーナル・ファイルに使用する 1 つまたは複数の暗号化キーを選択します。これらは、別々のものにすることができます（必要な場合）。

3. A、B、D、およびオプションで R を DR 非同期に変換できる場合は、それぞれで次の手順を実行します。
 - a. A、B、D (およびオプションで R) で、ジャーナル・ファイルの暗号化に使用するすべてのキーをロードして、まだ有効にしていない場合は有効にします。
 - b. “[インスタンスの既定のデータベース暗号化キーまたはジャーナル暗号化キーの指定](#)” の説明に従って、インスタンスでジャーナル暗号化に使用する目的のキーを選択します。
4. 前の手順でまだ実行していない場合は、R でジャーナル暗号化キーをロードし、有効にして、選択します。
5. “[暗号化の起動設定の構成](#)” の説明に従って、A、B、D、および R で、この順にジャーナル暗号化を有効にします。
 インスタンスでジャーナル暗号化を有効にすると、インスタンスの再起動か次のジャーナル切り替えのいずれか先に生じた方の後に暗号化が開始します。ミラー・メンバを再起動せずに変更がすぐに生じるようにするには、“[ジャーナル・ファイルの切り替え](#)” で説明しているように、各メンバでジャーナル・ファイルを手動で切り替えます。

注釈 JOURNAL ルーチン (“[ジャーナリングの概要](#)” を参照) には、ジャーナル暗号化を有効化/無効化する際に、管理ポータルの代わりに使用できるオプションが含まれます。このオプションを使用して暗号化を有効化すると、インスタンスは直ちに暗号化ジャーナル・ファイルに切り替わり、暗号化の起動設定が [インタラクティブ] に設定されます。

インスタンスでジャーナル暗号化キーを切り替えるには、新しいキーをロードして、有効にし、選択します。この新しいキーは、インスタンスの再起動か次のジャーナル切り替えのいずれか先に生じた方の後に暗号化に使用されます。

ジャーナル暗号化を有効にした後で、ミラーに DR 非同期を追加するときに、A、B、および D で使用しているジャーナル暗号化キー (1 つまたは複数) が、追加前の新しい DR 非同期で有効化されていることを確認してください。

4.12 ミラーへのアプリケーション・サーバ接続の構成

“[ミラーの自動導入方法](#)” に説明されているいずれかの方法を使用して、ミラーリングされたデータ・サーバと共に分散キャッシュ・クラスタを導入する場合、必要な構成はすべて自動化されます。管理ポータルを使用してクラスタを導入する場合は、各アプリケーション・サーバに追加するときにデータ・サーバがミラーであることを示す必要があります。いずれかの方法によって、データ・サーバがミラー接続として構成されている場合、各アプリケーション・サーバは、ミラーについて更新された情報をプライマリから定期的に収集し、フェイルオーバーを自動的に検出し、必要に応じて新しいプライマリに接続をリダイレクトします。

自動導入方法を使用した、ミラーリングされたデータ・サーバの構成の詳細は、“[ミラーの自動導入方法](#)” にリストされているドキュメントを参照してください。分散キャッシュ・クラスタ内のデータ・サーバとしてミラーを手動で構成するには、次の手順を実行します。

1. “[データ・サーバの準備](#)” の説明に従って、フェイルオーバー・メンバと DR 非同期メンバの両方をデータ・サーバとして準備します。これらのインスタンスはすべて、同じ [アプリケーションサーバの最大数] 設定で構成する必要があります。
2. 各アプリケーション・サーバで、以下を実行します。
 - ・ “[アプリケーション・サーバの構成](#)” の説明に従って、データ・サーバを追加します。このとき、必ず [ミラー接続] チェック・ボックスにチェックを付けてください。また、[ホスト DNS 名または IP アドレス] に、ミラーの仮想 IP アドレス (VIP) ではなく (ある場合でも)、現在のプライマリ・フェイルオーバー・メンバの DNS 名または IP アドレスを入力してください。
 - ・ “[アプリケーション・サーバの構成](#)” の説明に従って、データ・サーバ上の 1 つ以上のリモート・データベースにマッピングされた 1 つ以上のネームスペースを作成します。ミラーリングされているデータベース (:mirror:mirror_name:mirror_DB_name としてリストされているデータベース) とミラーリングされていないデータベース (:ds:DB_name としてリストされているデータベース) の両方を選択できます。ミラーのフェイルオーバー

の発生時には、ミラーリングされているデータベースにのみアプリケーション・サーバがアクセス可能な状態が維持されます。データ・サーバがフェイルオーバー・メンバの場合、ミラーリングされているデータベースは読み取り/書き込みとして追加され、ミラーリングされていないデータベースは、ジャーナリングされている場合は読み取り専用、ジャーナリングされていない場合は読み取り/書き込みとして追加されます。データ・サーバが DR 非同期メンバの場合、すべてのデータベースが読み取り専用として追加されます。

注釈 ミラーリングされるデータベースの「:mirror:mirror_name:mirror_DB_name:」というパス形式は、暗示的なネームスペース拡張グローバル参照に使用することもできます（“[拡張グローバル参照](#)”を参照してください）。

重要 フェイルオーバー・ミラー・メンバは、前記したようにミラー接続として構成されていない ECP 接続を受け入れません。一方、ミラー・メンバではないデータ・サーバは、ミラー接続として構成されている ECP 接続を受け入れません。つまり、データ・サーバのミラー・ステータスを変更する場合、以下を実行する必要があります。

- ・ 既存のミラーリングされていないデータ・サーバがミラーの[フェイルオーバー・メンバとして構成されている](#)場合、前述の手順に従って、すべてのアプリケーション・サーバ上でリモート・データ・サーバとしてのデータ・サーバを削除し、**[ミラー接続]** チェック・ボックスにチェックを付けた状態で[再び追加](#)する必要があります。
- ・ データ・サーバを含むミラーが[削除](#)され、前のプライマリがミラーリングされていないインスタンスに変換された場合、すべてのアプリケーション・サーバ上でリモート・データ・サーバとしてのデータ・サーバを削除し、**[ミラー接続]** チェック・ボックスのチェックを外した状態で[再び追加](#)する必要があります。

ミラーに接続するようアプリケーション・サーバを構成した後に、現在のプライマリを適切にシャットダウンすることによってリダイレクト・テストを実施し、アプリケーション・サーバが目的のミラー・メンバに接続していることを確認してください。

アプリケーション・サーバの **ECPServer** 定義の **Address** プロパティおよび **Port** プロパティによって指定されたミラー・メンバに接続を制限している間には、データ・サーバをミラー接続として識別することもできます。つまり、アプリケーション・サーバは、指定されたメンバがプライマリではない場合でも接続をリダイレクトしません。このように接続を構成すると、以下のルールが適用されます。

- ・ 指定されたメンバがプライマリの場合、アプリケーション・サーバからの接続を通常として受け入れます。メンバがフェイルオーバー・メンバだがプライマリではない場合（前のプライマリが再起動されてバックアップになったときなど）、これがプライマリになったときに接続を受け入れます。
- ・ 指定されたメンバが前のプライマリで、再起動されて再びプライマリになった場合、アプリケーション・サーバのこのメンバへの接続はリカバリされます。メンバがフェイルオーバー・メンバだがプライマリではない場合、これがプライマリになるまで接続を受け入れません。
- ・ 指定されたメンバが DR 非同期の場合、接続を受け入れ、ミラーリングされるデータベースへの読み取り専用アクセス権をアプリケーション・サーバに付与します（また、アプリケーション・サーバでリモート・データベースとして構成されているその他すべてのデータベースにも付与します）。

指定されたミラー・メンバに接続を制限することは、他のメンバに接続をリダイレクトすることが望ましくないとき（接続が高遅延 ECP 接続になるなど）に、ある特別な構成で役に立ちます。2 つの使用例を以下に示します。

- ・ ミラー・プライマリがデータ・センタ A (DCA) にあり、バックアップまたは DR 非同期がリモート・データ・センタ (DCB) にあるとします。各メンバにアプリケーション・サーバが構成された独自のバンクがあるとします。ネットワーク・ロード・バランサが接続を正しいデータ・センタに接続します。ただし、プライマリが使用できなくなり、DCB のメンバがフェイルオーバーまたは昇格によってプライマリになった場合、DCA のアプリケーション・サーバが DCB のメンバに接続することを望みません。DCA と DCB の間で高遅延接続が発生することになるからです。この状況では、DCA のアプリケーション・サーバで、フェイルオーバー発生時に DCB にリダイレクトしないようにミラー接続をプライマリに制限し、DCA のメンバが再びプライマリになるときにデータ・センタ間の接続をリカバリできるようにします。

- ・ プライマリとバックアップが DCA にあり、DR 非同期が災害発生時に使用する独自のアプリケーション・サーバと共にリモート DCB にあるとします。DCA のアプリケーション・サーバでは、プライマリは標準のミラー接続で構成されます。これは、フェイルオーバー時に DCA 内で接続をリダイレクトするためです。ただし、DCB のアプリケーション・サーバでは、ミラー接続は DR 非同期にリダイレクトされます。このようにすれば、災害リカバリ準備の一環としてまたは実際の災害時のカット・オーバーの前に読み取り専用ベースでミラー接続をテストできます。DR 非同期が昇格された後、DCA のアプリケーション・サーバは接続を DCB の新しいプライマリにリダイレクトしますが（ネットワーク・レベルで妨げられていない場合）、まだ降格されていない場合には、降格してこの動作を妨げることができます。

管理ポータルを使用してアプリケーション・サーバの接続を指定ミラー・メンバに制限できません。代わりに、以下の手順を実行します。

1. このセクションで説明した手順を使用してフェイルオーバー・メンバおよび DR 非同期をデータ・サーバとして準備し、各アプリケーション・サーバでデータ・サーバへの接続を構成します（この手順をまだ実行していない場合）。
2. `Config.ECPSTServer` クラスを使用してアプリケーション・サーバの `MirrorConnection` プロパティを変更し、その値として `-1` を指定します。アプリケーション・サーバ・インスタンスの `iris.cpf` ファイルを編集することもできます。ファイルの `[ECPSTServers]` セクションで、3 番目のパラメータを `0` から `-1` に変更します。詳細は、“[ECPSTServers](#)” を参照してください。

いずれかの方法で `MirrorConnection` プロパティを変更したら、管理ポータルを使用して [[ミラー接続](#)] チェック・ボックスの設定を変更しないでください。

4.13 ミラー仮想 IP (VIP) の構成

“[ミラー仮想 IP \(VIP\) の計画](#)” で説明しているように、フェイルオーバーで継続的なアクセスを確保するために、外部アプリケーションが単一のアドレスを使用してミラーと対話できるようにするミラー仮想アドレスを構成することができます。

[ミラー VIP を使用するよう InterSystems IRIS を構成](#)してからミラー VIP を構成した後に、現在のプライマリを適切にシャットダウンすることによってフェイルオーバー・テストを実施し（“[計画的停止の手順](#)” に記載）、どちらのフェイルオーバー・メンバがプライマリであるかに関係なく、アプリケーションが引き続きミラーに接続できることを確認してください。

重要 Linux プラットフォームでミラー VIP を構成する前に、該当するパッケージ（例えば、[Debian iputils-arping パッケージ](#)）をインストールすることによって `arping` コマンドを使用可能にしておいてください。

1 つ以上のミラー・メンバが、UNIX® または Linux システム上の非 root InterSystems IRIS インスタンスの場合、“[InterSystems IRIS 非 root インストール](#)” で説明されているように、ミラー VIP は使用できません。

注釈 VIP 使用時に DR 非同期をプライマリに昇格する処理に関する重要な情報は、“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)” を参照してください。

4.13.1 ミラー VIP の InterSystems IRIS 構成

現在どちらのフェイルオーバー・メンバがプライマリとなっているのかに関係なく、管理ポータルとスタジオが、確実にミラーにシームレスにアクセスできるようにするために、同じスーパーサーバ・ポート番号を使用するように、フェイルオーバー・メンバを構成することをお勧めします。

ミラーリングされたデータ・サーバと組み合わせた分散キャッシュ・クラスタ内のアプリケーション・サーバは、ミラーの VIP を使用しません。データ・サーバとしてミラーを追加する場合は（“[ミラーへのアプリケーション・サーバ接続の構成](#)” を参照）、ミラーの仮想 IP アドレス (VIP) を入力するのではなく、現在のプライマリ・フェイルオーバー・メンバの DNS 名または IP アドレスを入力します。アプリケーション・サーバは、指定のホストからミラーに関する更新情報を定期的に収集しているので、フェイルオーバーを自動的に検出し、新しいプライマリ・フェイルオーバー・メンバに切り替えます。この理由

から、フェイルオーバー・メンバとすべての DR 非同期メンバは、データ・サーバとして、同じ [アプリケーションサーバの最大数] 設定で準備する必要があります。詳細な分散キャッシュに関する考慮事項については、“[ミラーへのアプリケーション・サーバ接続の構成](#)”を参照してください。

1 つまたは両方のフェイルオーバー・メンバをライセンス・サーバとして構成する場合 (“[InterSystems IRIS ライセンスの管理](#)”で説明しています)、[ホスト名/IP アドレス]として構成するシステムの実際のホスト名または IP アドレスを指定します。VIP アドレスは入力しないでください。

4.13.2 ミラー VIP の構成

ミラー VIP を構成するには、次の情報を入力する必要があります。

- ・ ミラー VIP として使用可能な IP アドレス。VIP を予約して、その他のシステムが使用できないようにしておくことが重要です。例えば、動的ホスト構成プロトコル (DHCP) のネットワーク構成では、この VIP を予約し DNS テーブルから削除して、ネットワークに参加しているホストに動的に割り当てられないようにする必要があります。
- ・ クラスレスのドメイン間ルーティング (CIDR) 表記での指定が必要な適正ネットワーク・マスク。CIDR 表記の形式は、ip_address/CIDR_mask です。ip_address は、システムのベース IP アドレスです。CIDR_mask は、次のようにプラットフォームによって異なります。
 - macOS - /32 である必要があります。
 - 他のすべてのプラットフォーム - ベース・インタフェースに割り当てられた IP アドレスのマスクと一致する必要があります。以下はその例です。

```
bash-2.05b# uname -a
AIX apis 3 5 00C0B33E4C00
bash-2.05b# ifconfig en1
en1:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,
GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),PSEG,LARGESEND,CHAIN>
inet 10.0.2.11 netmask 0xffffffff00 broadcast 10.0.2.255
tcp_sendspace 131072 tcp_recvspace 65536 rfc1323
```

この例では、[en1] インタフェースは、ベース・アドレスが 10.0.2.11 でネットマスクが 0xffffffff00 となっており、ネットマスクは /24 に変換されます。したがって、[en1] インタフェースに 10.0.2.100 を VIP として割り当てるには、ネットワーク・マスクを 10.0.2.100/24 (CIDR 表記) と指定します。

- ・ 各フェイルオーバー・メンバで使用可能なネットワーク・インタフェース 2 つのシステムで選択されたインタフェースは、VIP と同じサブネットとする必要があります。

ネットワーク・インタフェースを選択するときには、そのインタフェースを正しく動作させるために、以下のプラットフォーム固有の規則に従う必要があります。

- IBM AIX®, Linux (Red Hat, SuSE, Ubuntu)、Apple macOS、および Windows - VIP を構成するときに、既存の物理ネットワーク・インタフェースを使用できるように設定する必要があります。これらのプラットフォームでは、IP アドレスのエイリアスは、IP アドレス (すなわち VIP) を既存の物理ネットワーク・インタフェースにバインドするために使用されます。また、このプラットフォームでは、単一の物理ネットワーク・インタフェースで複数の IP アドレスをホストできます。

4.14 ISCAgent の構成

ISCAgent の実行は、各ミラー・メンバ上の構成可能な専用ポート (既定値は 2188) で安全に行われます。このエージェントがネットワーク接続を受信してミラーリングされたインスタンスにつながると、そのインスタンスの irisuxagent を実行し

て、ミラー・メンバの管理に必要な特権にエスカレートします。ミラーが [TLS を要求する構成](#) の場合、何らかのアクションを実行する前に、受信接続が認証されます。

1 つ以上のミラーに属する複数の InterSystems IRIS インスタンスが、1 つのシステムでホストされている場合、それらのインスタンスは、1 つの ISCAgent を共有します。

ここでは、以下の方法で ISCAgent を管理するための情報を提供します。

- ・ [ISCAgent の開始および停止](#)
- ・ [ISCAgent のカスタマイズ](#)

4.14.1 ISCAgent の開始および停止

InterSystems IRIS のインストール時またはアップグレード時にインストールされる ISCAgent は既定により、ユーザ **[ISCAgent]** として、かつ **[ISCAgent]** グループのメンバとして動作します。InterSystems IRIS インスタンスへのアクセスを提供する `irisuxagent` ユーティリティの実行に必要なグループ特権 ("**ISCAgent**" を参照してください) を取得するために、ISCAgent は、システム起動時に自動で開始されるようにするか、root 特権を持つユーザが開始する必要があります。ISCAgent は、自身に必要なユーザ特権およびグループ特権を割り当てると、すべての root 特権を破棄します。

ISCAgent の構成により、各フェイルオーバー・メンバおよび DR 非同期ミラー・メンバでのシステム起動時に、ISCAgent を自動で開始させる必要があります。システム管理者が初期化プロセスに追加できるプラットフォーム固有の制御スクリプトが提供されています。これについて、以下の節で説明します(システム起動の構成手順の詳細は、オペレーティング・システムのドキュメントを参照してください)。

- ・ [UNIX® および macOS システムでの ISCAgent の開始](#)
- ・ [Linux システムでの ISCAgent の開始](#)
- ・ [UNIX®/Linux および macOS システムでの非 root インスタンス用 ISCAgent の開始](#)
- ・ [Microsoft Windows システムでの ISCAgent の開始](#)

4.14.1.1 UNIX® および macOS システムでの ISCAgent の開始

UNIX® および macOS のプラットフォームでは、オペレーティング・システムに応じて、次の場所にインストールされる ISCAgent の開始および停止スクリプトを実行します。

- ・ IBM AIX® : `/etc/rc.d/init.d/ISCAgent`
- ・ macOS : `/Library/StartupItems/ISCAgent/ISCAgent`

例えば、IBM AIX® プラットフォームで ISCAgent を開始する場合は、root としてコマンド `/etc/rc.d/init.d/ISCAgent start` を実行します。このサービスを停止するには、コマンド `/etc/rc.d/init.d/ISCAgent stop` を実行します。

UNIX®/Linux のプラットフォームでの ISCAgent に関するその他の考慮事項としては、以下のものがあります。

- ・ 前述したように、ISCAgent は、各フェイルオーバー・メンバまたは DR 非同期メンバで、システム起動時に自動的に開始される必要があります。また、ユーザがこのエージェントを開始、停止、または再開するのが便利なときもあります。これは以下の方法で実行できます。
 - － 前の手順で説明したように、root ユーザが直接的に実行します。
 - － InterSystems IRIS インスタンスの開始と停止が可能なユーザが、InterSystems IRIS インスタンスの `/bin` ディレクトリにある `agentctrl` 実行ファイルを使用します。例えば、エージェントを開始するには、以下のコマンドを実行します。

```
/iris/bin$ ./agentctrl start
```

このコマンドは、引数 **[停止]** および **[再開始]** を取ります。

- InterSystems IRIS では、ISCAgent のステータスを追跡するために `iscagent.status` ファイルを使用します。このファイルは、IRISSYS 環境変数で指定されるディレクトリ内にあります（“インストール・ディレクトリ”を参照してください）。IRISSYS が定義されていない場合は、`/var/run` にあります。エージェントは、このファイルに対する排他的ロックを取得できる必要があるため、`iscagent.status` ファイルが `/var/run` に配置されていて、`/var/run` が NFS でマウントされたファイルシステムにある場合は、その NFS 構成が `fcntl` ファイル・ロックをサポートしている必要があります。
- 前述したように、ISCAgent は `irisuxagent` を使用して、InterSystems IRIS インスタンスの管理に必要な特権を取得します。既定により、エージェントは `irisuxagent` の実行に必要な特権（[ISCAgent] ユーザまたは [ISCAgent] グループ）を所有しており、標準的な構成においては変更が不要です。

ただし、ユーザ・システムのセキュリティ構成により、ユーザ・サイトのインスタンスには、`irisuxagent` を実行するにあたって、ミラーリングされるインスタンスの `/bin` ディレクトリに移動するための追加特権が必要になる場合があります。この場合、ISCAgent の特権がコマンドの実行に十分なものであるかを確認する必要があります。そのためには、以下の手順に従って、エージェントの特権を変更できます。

- `/etc/iscagent/iscagent.conf` ファイルを作成します。ファイルが既に存在する場合は、それを編集します（例えば、以前にファイルを作成して、ISCAgent のポート番号やインタフェースをカスタマイズした場合）。
- グループ特権を追加するには、以下の行を追加して、`irisuxagent` の実行が必要なグループを 1 つ以上指定します。

```
privileges.group=iscagent,<groupname>[,<groupname>[,...]]
```

通常は、グループ特権の追加で十分となります。しかし、一部の構成では、別のユーザとして ISCAgent を実行することが必要になる場合もあります。これは、`/etc/iscagent/iscagent.conf` で、以下のように実行することもできます。

```
privileges.user= <username>
```

注釈 `irisuxagent` には、[ISCAgent] グループ特権が必要なので、[ISCAgent] はグループ・リストに残る必要があります。

- ISCAgent はオペレーティング・システムのシステム・エラー・ログ（Linux では `/var/log/messages` など）にメッセージを書き込みます。

4.14.1.2 Linux システムでの ISCAgent の開始

`systemd` をサポートしている Linux システム（SUSE Linux Enterprise Server 12 SP1 以降など）の場合は、`/etc/systemd/system/ISCAgent.service` ファイルがインストールされています。これにより、`systemd` を使用した ISCAgent の管理がサポートされるようになります。このようなシステムでは、ISCAgent の開始、停止およびステータスの表示に、以下のコマンドを使用できます。

```
systemctl start ISCAgent.service
systemctl stop ISCAgent.service
systemctl status ISCAgent.service
```

`systemd` をサポートするシステムの場合、システムのブート時に ISCAgent を開始するかどうかを制御するには、以下のコマンドを使用します。

```
sudo systemctl enable ISCAgent.service
sudo systemctl disable ISCAgent.service
```

既定では、`systemd` サービスは無効化されています。このサービスが無効化されている場合でも、`systemctl` を使用すると、必要に応じてサービスを開始および停止できます。

`ISCAgent.service` ファイルは、InterSystems IRIS レジストリ・ファイルと共有サポート・ファイルの場所を IRISSYS 環境変数から読み込みません（“インストール・ディレクトリ”を参照してください）。代わりに、その場所として `/usr/local/etc/irissys`

が指定されてインストールされます。**ISCAgent.service** を編集して、別のレジストリ・ディレクトリを指定することもできます (必要な場合)。

どの Linux システムでも、ISCAgent の開始/停止スクリプトは、**/etc/init.d/ISCAgent** にインストールされます。このスクリプトについては、“[UNIX® および macOS システムでの ISCAgent の開始](#)” を参照してください。systemd がサポートされていない場合は、上記のセクションで説明している ISCAgent の開始および停止コマンドを使用します。

“[UNIX® および macOS システムでの ISCAgent の開始](#)” に記載されたそれ以外の情報は、systemd をサポートする Linux システムにも適用されます。

重要 systemd をサポートする Linux システムでは、systemctl コマンドと **/etc/init.d/ISCAgent** スクリプトのどちらを使用することも可能ですが、どちらか一方の方法に決めて、それを切り替えることなく排他的に使用する必要があります。ISCAgent を停止する際には、常に、それを開始した方法を使用する必要があります。

こういった Linux システムで InterSystems IRIS をアップグレードすると、実行されている ISCAgent は、systemd を使用して自動的に再起動されます。ISCAgent の管理に **/etc/init.d/ISCAgent** スクリプトを使用している場合は、このエージェントが自動的に再起動されないように、エージェントを停止してからアップグレードを実行し、アップグレードの後に、このスクリプトを使用してエージェントを再起動します。

/etc/init.d/ISCAgent スクリプトの使用から systemctl コマンドの使用に変更したら、初めて systemctl でエージェントを開始する前に root として以下を実行します。

1. 以下のコマンドを実行します。

```
systemctl status ISCAgent
```

2. このコマンドの出力に以下の警告が含まれている場合：

```
Warning: Unit file changed on disk, 'systemctl daemon-reload' recommended.
```

以下のコマンドを実行します。

```
systemctl daemon-reload
```

3. 上記のコマンドが完了したら、systemctl status ISCAgent を再実行し、警告が表示されないことを確認します。

4.14.1.3 UNIX®/Linux および macOS システムでの非 root インスタンス用 ISCAgent の開始

通常、InterSystems IRIS は root としてインストールされますが、UNIX®/Linux および macOS システムでは、それとは別のユーザがインスタンスをインストールして実行することが可能です。非 root インストールの詳細は、“[InterSystems IRIS 非 root インストール](#)” を参照してください。

非 root インスタンス用の ISCAgent は、インストール・ユーザが **ISCAgentUser** スクリプト (IRISSYS 環境変数で定義されたディレクトリにある) を実行すると、バックグラウンドで開始されます。以下に例を示します。

```
nohup <IRISSYS_directory>/ISCAgentUser &
```

システム起動時に、ISCAgent が自動的に開始するように構成することはできないかもしれませんが、それが実現可能ならば、そちらを選択してください。ミラーに 2 つのフェイルオーバー・メンバが含まれている場合のベスト・プラクティスは、システムの起動後に、InterSystems IRIS を開始するつもりがない場合でも、できるだけ早くエージェントを起動することです。これは、“[両方のフェイルオーバー・メンバの計画外停止](#)” の説明にあるような特定の状況で、復旧の支援になります。

4.14.1.4 Microsoft Windows システムでの ISCAgent の開始

Microsoft Windows システムでは、次の方法で ISCAgent プロセスを開始します。

1. Microsoft Windows の [コントロール・パネル] で [システムとセキュリティ] メニューをクリックして開きます。
2. [システムとセキュリティ] で、[管理ツール] メニューをダブルクリックし、表示されるサブメニューから [サービス] を選択します。
3. [サービス] で [ISCAgent] をダブルクリックして、[ISCAgent のプロパティ] ウィンドウを表示します。
4. [全般] タブの [スタートアップの種類] ドロップダウン・リストから [自動] を選択します。
5. [全般] タブで [開始] をクリックして、ISCAgent を開始するか、[停止] をクリックして停止します。

4.14.2 ISCAgent のカスタマイズ

ISCAgent の以下の属性をカスタマイズできます。

- ・ ポート番号

このページで前述のとおり、既定の ISCAgent ポートは 2188 です。通常はこれで十分ですが、このポート番号は必要に応じて変更できます。

- ・ インタフェース

ISCAgent は使用可能なインタフェースすべての既定 (もしくは構成された) ポートに結合します。通常はこれで十分ですが、必要に応じて、特定のアドレスに対応するインタフェースに結合するよう ISCAgent を変更することができます。

- ・ SYSLOG 深刻度レベル

既定で、ISCAgent は、すべてのログ・メッセージを SYSLOG と呼ばれる InterSystems IRIS システム・エラー・ログに送信します (“InterSystems IRIS システム・エラー・ログ” を参照してください)。必要に応じて、最小の深刻度設定を構成できます。そうすることで、設定した深刻度より低いメッセージがシステム・エラー・ログに渡されることがなくなります。

ISCAgent をカスタマイズする手順は以下のとおりです。

1. **iscagent.conf** ファイルを作成します。ファイルが既に存在する場合は、そのファイルを編集します。
 - ・ UNIX/Linux/macOS : **/etc/iscagent/iscagent.conf**
 - ・ Windows : **windir¥system32¥iscagent.conf** (windir はシステムのルート・ディレクトリ)
2. ポートをカスタマイズするには、以下の行を追加して、<port> を目的のポート番号に置き換えます。

```
application_server.port=<port>
```

3. インタフェースをカスタマイズするには、以下の行を追加して、<ip_address> を目的のインタフェースが対応するアドレスに置き換えます。

```
application_server.interface_address=<ip_address>
```

使用可能なすべてのインタフェースに明示的に結合するには (既定)、IP アドレスとして * を指定します。

4. SYSLOG 深刻度レベルをカスタマイズするには、以下の行を追加して、<severity> を目的の最小深刻度レベルに置き換えます。1 = 警告、2 = 重大、および 3 = 致命的です。

```
logging.minimum_severity=<severity>
```

4.15 サービス品質 (QoS) タイムアウトの構成

[サービス品質タイムアウト] (QoS タイムアウト) は、フェイルオーバー・メンバとアービターの動作を管理する際に重要な役割を果たすものであり、ミリ秒単位の時間の範囲を定義します。ミラー・メンバは、この時間の範囲内で別のミラー・メンバからの応答を待機し、その範囲を超えると対応策を実行します。QoS タイムアウト自体は、最大待機時間を表します。最小待機時間は、その半分になります。QoS タイムアウトの値を大きくすると、ホストやネットワークが停止と見なされるまでの無応答時間の許容範囲が広がります。QoS の値を小さくすると、ミラーは迅速に停止に対応できるようになります。具体的には、QoS タイムアウトは以下の状況に作用します。

- ・ バックアップ・フェイルオーバー・メンバが、プライマリからのデータに対して QoS タイムアウトで定義した範囲内に受信確認を行わないと、プライマリはバックアップとの接続を切断して、バックアップが停止したと見なしてアクションを取ります。
- ・ バックアップが、プライマリからのメッセージを QoS タイムアウトで定義した範囲内に受信できなかった場合、バックアップは接続を切断して、プライマリが停止したと見なしてアクションを取ります。
- ・ アービターが、フェイルオーバー・メンバからの応答を QoS タイムアウトで定義した範囲内に受信できなかった場合、アービターは、そのフェイルオーバー・メンバとの接続が失われていると見なします。
- ・ フェイルオーバー・メンバのホストで実行される処理により、QoS タイムアウトで定義した範囲内の期間、ホストが完全に無反応になると、不要なフェイルオーバーやアラートが発生することがあります。これは、バックアップや移行を伴う処理を実行する仮想化プラットフォームで特に問題になります。詳細は、“[仮想環境でのミラーリング](#)”を参照してください。

フェイルオーバー・メンバとアービターの動作に関して QoS タイムアウトが果たす役割についての完全な詳細は、“[自動フェイルオーバーのメカニズム](#)”を参照してください。

既定の QoS は 8 秒 (8000 ミリ秒) です。これは、一部のハードウェア構成で発生する数秒間の断続的な無応答状態を考慮に入れたものです。一般に、専用のローカル・ネットワークに接続された物理 (仮想化されていない) ホストへの配置では、停止に対するより迅速な対応が必要な場合に、この設定を短くすることができます。

[サービス品質タイムアウト] の設定は、[\[ミラーの作成\]](#) ページ、またはプライマリ・フェイルオーバー・メンバの [\[ミラーの作成\]](#) ページで調整できます。

注釈 QoS タイムアウトは、`MIRROR` ルーチン (“[MIRROR ルーチンの使用法](#)”を参照) の [\[ミラー構成\]](#) メニューにある [\[サービス品質タイムアウト・パラメータの調整\]](#) オプションを使用して調整することもできます。

4.16 並列デジャーナリングの構成

“[ミラー同期](#)”で説明したように、バックアップ・フェイルオーバー・メンバおよびミラーの非同期メンバのミラーリングされるデータベースは、デジャーナリングによってプライマリとの同期状態が維持されます。これは、プライマリで実行され、プライマリのジャーナル・ファイルに記録されるデータベース更新を他のメンバの対応するデータベースに適用する動作です。十分な処理能力があり、十分なメモリ・リソースを利用できる場合、1 回のデジャーナリング・オペレーションで最大 16 件のデジャーナリング・ジョブが更新を並列実行できます (“[並列デジャーナリングのシステム要件](#)”を参照してください)。この機能は、並列デジャーナリングと呼ばれ、特にデータベースの更新頻度が高いミラーのスループットを向上させます。並列デジャーナリングはジャーナル・リストア中にも使用されます。詳細は、“[JRNRESTO を使用したジャーナル・ファイルからのグローバルのリストア](#)”を参照してください。

並列デジャーナリングはミラーのフェイルオーバー・メンバでは常に有効になっているため、必要なリソースが利用可能な場合は使用されます。既定では、DR 非同期メンバでも有効になっています。レポート非同期についても有効にすることができます (つまり、すべてのメンバについて有効にできます)。また、フェイルオーバー・メンバのみに制限することもできます。そのためには、第 1 のフェイルオーバー・メンバを構成するときに [\[並列デジャーナリングを許可する\]](#) 設定を

変更 (“ミラーを作成し、第 1 のフェイルオーバー・メンバを構成する”を参照)、またはプライマリでミラーを編集 (“フェイルオーバー・メンバの編集または削除”を参照)します。並列デジャーナリングが有効になっている場合 (また利用可能なリソースが対応している場合)、1 回の操作で複数のデータベースをキャッチアップするときに使用されます。“ミラーリングされるデータベースの有効化とキャッチアップ”を参照してください。

レポート非同期に対して並列デジャーナリングを有効にすると、パフォーマンスの観点では有利ですが、クエリまたはレポートで予期しない結果になる可能性が高まる場合があります。これは、別個のデジャーナリング・ジョブによって更新されるデータベースまたはデータベース内のグローバルが、デジャーナリングの順序において若干異なる可能性があるためです。例えば、データベース A に 11:45:30 にプライマリで実行される更新が含まれているときに、データベース B は 11:45:28 からの更新に対してのみ準備できているような場合です。この場合、同一のトークンによって、あるグローバルは前者の時間に従って更新され、同じデータベース内の別のグローバルは後者の時間に従って更新される可能性があります。ただし、並列デジャーナリングが原因の不確実性は、デジャーナリング処理において変化するデータに対してレポートまたはクエリを実行するときに常に発生する不確実性と同様です。したがって、インターシステムズは、並列デジャーナリングが有効になっている、ミラーリングされるデータベースに対して実行されるほとんどのレポート・アプリケーションでは、その影響を無視できると考えています。データベース内の 1 つのグローバルに対するすべての更新は常に 1 つのデジャーナリング・ジョブによって適用され、これらの更新は順番に行われます。

4.17 ^ZMIRROR ルーチンの使用法

ユーザ定義の ^ZMIRROR ルーチンにより、特定のミラーリング・イベント用の (フェイルオーバー・メンバがプライマリになるなど)、独自のカスタムで構成固有のロジックおよびメカニズムを実装できます。

^ZMIRROR ルーチンには、次のエントリ・ポイントが含まれています。値を省略した場合、これらすべては、適切な既定値が指定されます。

- ・ `$$CanNodeStartToBecomePrimary^ZMIRROR()` – インスタンスで以下のことと判断された場合、このプロシージャが呼び出されます。
 - 他方のフェイルオーバー・メンバが現在プライマリとして動作しておらず、手動操作なしではプライマリになれない。
 - ローカル・メンバはプライマリになることが可能で、引き継ぎのプロセスを開始しようとしている。

`CanNodeStartToBecomePrimary` は、起動時またはバックアップとしての接続時のいずれかに、フェイルオーバー・メンバが自動的にプライマリになることをブロックするロジックのエントリ・ポイントを提供し、フェイルオーバーを手動制御できるようにします。これはほとんどの ^ZMIRROR ルーチンには組み込まれていません。

`CanNodeStartToBecomePrimary` が 1 を返すと、ローカル・インスタンスがプライマリ・フェイルオーバー・メンバとして完全初期化され、プライマリになるプロセスを続行できます。つまり、ミラーリングされるデータベースすべてが読み書き可能となり、ECP セッションが回復またはロールバックされていて、以前のプライマリからのローカル・トランザクション (存在する場合) がロールバックされています。ユーザのログインは許可されておらず、スーパーサーバ接続は阻止され、かつ ECP が復旧途中にあるので、新規の処理は実行されていません。

このエントリ・ポイントが 0 (False) を返す場合、インスタンスは再試行ループに入り、`CanNodeStartToBecomePrimary` の呼び出しを 30 秒ごとに、以下のことが生じるまで続けます。

- `CanNodeStartToBecomePrimary` が 1 (True) を返し、ローカル・メンバがプライマリになるプロセスを続行する。
 - インスタンスで、他方のフェイルオーバー・メンバがプライマリになったこと (手動操作による必要があります) が検出され、この時点でローカル・メンバがバックアップになる。
- ・ `$$CheckBecomePrimaryOK^ZMIRROR()` – このプロシージャは `CanNodeStartToBecomePrimary` が 1 (True) を返した後に呼び出されます。

CheckBecomePrimaryOK が存在し、1 を返すと、ミラーはローカル・メンバをプライマリとして動作を再開します。この時点で、任意のローカル・プロセスを開始でき、またユーザのアプリケーション環境を整えるために必要な初期化ができます。ただし、CheckBecomePrimaryOK を実行するプロセスのみが、1 を返した後に、ミラーリングされるデータベースに実際に書き込み、その時点でミラーリングされるデータベースが一般の使用のために更新されることに注意してください。

CanNodeStartToBecomePrimary と同様に、CheckBecomePrimaryOK が 0 (False) を返すと、インスタンスはプライマリになるプロセスを中止して、CheckBecomePrimaryOK の呼び出しを 30 秒ごとに、以下のことが生じるまで再試行します。

- エントリ・ポイントが 1 を返し、ミラーがローカル・メンバをプライマリとして動作を再開する。
- インスタンスで、他方のフェイルオーバー・メンバがプライマリになったこと（手動操作による必要があります）が検出され、この時点でローカル・メンバがバックアップになる。

通常、CheckBecomePrimaryOK は正常に実行されます。ノードがプライマリ・メンバにならない“事例がよく起こる”場合は、CheckBecomePrimaryOK ではなく、CanNodeStartToBecomePrimary を使用して処理する必要があります。

プライマリの既存の ^%ZSTART ルーチンから ^ZMIRROR にコードを移動することで、ミラーを初期化しない限りコードが実行されないようにする場合は、CheckBecomePrimaryOK が最適な位置となります。ただし、job コマンドを使用してその他のジョブを開始する場合、そのジョブは \$SYSTEM.Mirror.IsPrimary() が true を返すまで待機します。これは CheckBecomePrimaryOK が 1 を返した後に行われます。または、代わりに \$\$NotifyBecomePrimary^ZMIRROR() でジョブを開始できます。

注釈 CheckBecomePrimaryOK が False を返す場合、ECP セッションはリセットされます。ノードのプライマリ化が成功すると、ECP クライアントは再接続され、ECP トランザクションが（保持ではなく）ロールバックされます。クライアント・ジョブは、TRollback コマンドが明示的に実行されるまで <NETWORK> エラーを受け取ります（“ECP Rollback Only の保証”を参照してください）。

- ・ \$\$NotifyBecomePrimary^ZMIRROR() – このプロシージャは、プライマリ・フェイルオーバー・メンバになるプロセスの最後に情報提供を目的として呼び出されます（つまり、ユーザが許可された後、かつ ECP セッションが（存在する場合）アクティブになった後）。このエントリ・ポイントは値を返しませんが、コードを組み込むことにより、必要に応じた通知の生成やアプリケーション・ログインの有効化ができます。
- ・ \$\$NotifyBecomePrimaryFailed^ZMIRROR() – このプロシージャは、以下の場合に情報提供を目的として呼び出されます。
 - フェイルオーバー・メンバが起動したが、プライマリまたはバックアップ・メンバになれなかった。
 - バックアップがプライマリの失敗を検出し、バックアップがプライマリになろうとしたが、なれなかった。

このエントリ・ポイントは、各問題の発生に対して 1 回だけ呼び出されます。メンバがプライマリになるか、プライマリが検出されるまで再度呼び出されることはありません。

4.18 医療製品向けのミラーリングの構成

InterSystems IRIS for Health™ および HealthShare® Health Connect でミラーリングを設定および管理する際は、いくつかの特別な考慮事項があります。その他の HealthShare 製品には、独自のミラーリングに関するドキュメントがあることに注意してください。以下の考慮事項は、これらの製品には必ずしも適用されません。

InterSystems IRIS for Health または HealthShare Health Connect システムにミラーリングを適用する場合は、Foundation ネームスペースの設定時に [Mirror Database] オプションを選択する必要があります。”InterSystems IRIS for Health イ

インストール・ガイド”または“HealthShare Health Connect インストール・ガイド”の“インストーラ・ウィザードの使用”を参照してください。

システムでミラー VIP を構成し、この VIP をネットワーク・ホスト名として設定する必要があります。管理ポータル内でネットワーク・ホスト名を設定するには、[Health]→[インストーラ・ウィザード]→[ネットワーク・ホスト名の構成]に移動し、VIP を新しいネットワーク・ホスト名として入力して、[保存]を選択します。

最後に、HSSYS ネームスペース内の HS_Services 相互運用認証情報のパスワードが HS_Services のパスワードと一致することを確認する必要があります。管理ポータル内で相互運用認証情報を更新するには、HSSYS ネームスペースに切り替え、[相互運用性]→[構成]→[認証情報]に移動し、テーブルから [HS_Services] 認証情報を選択し、新しい [パスワード] を入力して、[保存]を選択します。

データベースをミラーリングする際は、以下の点に留意してください。

- ・ HSLIB をミラーリングしないでください。
- ・ HSSYS はミラーリングする必要があります。
- ・ HSCustom をミラーリングして、カスタム・コードを同期された状態に保つことができます。

5

ミラーリングの管理

ここでは、InterSystems IRIS のミラーに関し、運用上の管理と維持に関連するトピックについて説明します。

5.1 ミラーの監視

以下の 2 つの方法のいずれかを使用して、既存のミラーの動作を監視できます。

- ・ 管理ポータル の [\[ミラーモニタ\]](#) ページ
- ・ ^MIRROR ルーチンの [\[ステータス・モニタ\]](#) オプション

どちらの方法でも、[ミラーおよびそのメンバの動作状態](#)、[受信ジャーナルの転送速度](#)、および[ミラーリングされるデータベースの状態](#)に関する情報が表示されます。また、ミラー・モニタでは、ミラーリングされるデータベースに対して複数の処理を実行できます。

“[ミラーリング通信プロセスの監視](#)” で、ミラー・メンバ上で実行されるミラー通信プロセスについて説明します。

注釈 管理ポータル・ホーム・ページのメッセージ・ペインには、ミラー・モニタへのリンクを含め、基本的なミラー・メンバの情報も表示されます (“[管理ポータル・メッセージ・ペイン](#)” を参照してください)。

データベースのマウントとディスマウントやミラーに対するデータベースの追加と削除など、データベースおよびミラーに関連する多くの操作は、メッセージ・ログに記録されます (“[ログ・ファイルの監視](#)” を参照してください)。

5.1.1 ミラー・モニタの使用法

ミラー・モニタを表示するには、ミラー・メンバで [\[システム処理\]](#) → [\[ミラー・モニタ\]](#) ページに移動します。

フェイルオーバー・メンバでは、ミラー・モニタに以下のボタンとセクションが含まれています。

- ・ [\[ミラージャーナルファイル参照\]](#) ボタンは、メンバのミラー・ジャーナル・ファイルまたは非ミラー・ジャーナル・ファイルを表示および検索します。詳細は、“[ジャーナル・ファイルの表示](#)” を参照してください。
- ・ [\[このメンバのミラーリングを停止\]](#) ボタン (バックアップのみ) は、“[バックアップおよび非同期メンバのミラーリングの停止](#)” で説明されているとおり、バックアップのミラーリングを一時的に停止します。
- ・ DR 非同期へのバックアップの降格に使用する 2 つのボタンのいずれか。このとき、ミラーには 1 つのフェイルオーバー・メンバだけが残されます。詳細は、“[DR 非同期へのバックアップの降格](#)” を参照してください。
- ・ [\[ミラー・フェイルオーバー・メンバ情報\]](#) には、各フェイルオーバー・メンバのメンバ名、スーパーサーバ・アドレス、およびミラー・プライベート・アドレスが表示されます (これらのアドレスの詳細は、“[ミラー・メンバのネットワーク・アドレス](#)” を参照してください)。

- ・ [アービター接続ステータス] には、アービターが構成されている場合にそのアドレス (ホスト名とポート)、現在のフェイルオーバー・モード、およびメンバのアービター接続のステータスが次のように表示されます。
 - [両方のフェイルオーバー・メンバがアービターに接続されている]
 - [このメンバのみがアービターに接続されている]
 - [このメンバはアービターに接続されていない] (接続が失われているか、アービターが構成されていない場合)

アービターおよびこの接続情報の意味に関する詳細は、“[自動フェイルオーバーのメカニズム](#)”を参照してください。

注釈 フェイルオーバー・メンバがアービターとの通信を失うと、深刻度 2 のメッセージがメッセージ・ログに送信されます。そのメンバがアービターとの接続の再構築に失敗すると、別の深刻度 2 のメッセージがログに記録されます。

- ・ [ミラー・メンバ・ステータス] には、“[ミラー・メンバのジャーナル転送およびデジャーナリングのステータス](#)”で説明しているように、各ミラー・メンバのメンバ・タイプとステータス、ジャーナル転送ステータス、およびデジャーナリング・ステータスが表示されます。このテーブルには、メンバの X.509 DN が構成されている場合はそれらも表示されます。
- ・ [ミラーリングされるデータベース] には、“[ミラーリングされるデータベースのステータス](#)”で説明しているように、現在のメンバ上のミラーリングされるデータベースそれぞれに関する情報が、その名前、ディレクトリ、ステータス、およびデジャーナリング・ステータスも含めて表示されます。[ミラーリングされるデータベース]を使用して、1 つ以上のデータベースに対して複数の処理を実行することもできます。

注釈 [ミラーリングされるデータベース] リストには、現在のメンバのミラーに含まれているデータベースのみが挙げられます。例えば、異なるレポート非同期メンバでリストされるデータベースは異なる場合があります。ミラーリングされるデータベースのセットが異なる場合があります。

非同期メンバでは、ミラー・モニタには、以下のボタンとセクションが含まれています。

- ・ [ジャーナル・ファイルの表示] ボタン (DR 非同期のみ) は、メンバのミラー・ジャーナル・ファイルまたは非ミラー・ジャーナル・ファイルを表示および検索します。詳細は、“[ジャーナル・ファイルの表示](#)”を参照してください。
- ・ [このメンバのミラーリングを停止] ボタン (DR 非同期のみ) は、“[バックアップおよび非同期メンバのミラーリングの停止](#)”で説明されているとおり、非同期のミラーリングを一時的に停止します。
- ・ [フェイルオーバー・メンバに昇格] ボタン (DR 非同期のみ) は、DR 非同期をフェイルオーバー・メンバに昇格します。この操作とその使用法については、“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”を参照してください。
- ・ ボタンの下部のセクションには、メンバのメンバ名、非同期タイプ、および X.509 DN (構成されている場合) が表示されます。
- ・ [この非同期メンバの所属先のミラー] には、“[ミラー・メンバのジャーナル転送およびデジャーナリングのステータス](#)”で説明しているように、レポート非同期メンバの所属先の各ミラーに関する情報、およびそのミラー内でのメンバのステータス、ジャーナル転送ステータス、およびデジャーナリング・ステータスが表示されます。各行には、そのミラーのメンバに関する情報を表示する[詳細]リンクと、そのミラーの非同期メンバによるミラーリングを停止させる[このメンバのミラーリングを停止]リンクが含まれています。詳細は、“[バックアップおよび非同期メンバのミラーリングの停止](#)”を参照してください。
- ・ [ミラー・メンバ・ステータス] には、現在の非同期メンバを含めて、[この非同期メンバの所属先のミラー]で選択したミラーのすべてのメンバのタイプ、ステータス、ジャーナル転送遅延、およびデジャーナリング遅延が表示されます。

非同期メンバが単一ミラーに属している場合 (すべての DR 非同期に当てはまります)、既定でそのミラーがこのセクションに表示されます。メンバが複数のミラーに属している場合、このセクションおよびその下の [ミラーデータベース] MIRROR セクションは、[この非同期メンバの所属先のミラー] セクションに示されるいずれかのミラーの [詳細] リンクをクリックするまで表示されません。

- ・ [ミラーデータベース] MIRROR セクションは、フェイルオーバー・メンバの場合の説明と同様で、また同じ処理を実行できます。レポート非同期の場合は、[この非同期メンバの所属先のミラー] で選択し [ミラー・メンバ・ステータス] に表示されるミラーに含まれるデータベースが表示されます。

5.1.1.1 ミラー・メンバのジャーナル転送およびデジャーナリングのステータス

“ミラーの監視” で説明しているように、InterSystems IRIS インスタンスがミラーに属している場合、そのメンバ・タイプとステータス、ジャーナル転送ステータス、およびデジャーナリング・ステータスは、ミラー・モニタおよび ^MIRROR ルーチンの [ステータス・モニタ] オプションによって表示されます。

次のテーブルは、表示される可能性のあるタイプおよびステータスを示しています。最初のテーブルは、特定のメンバ・タイプに固有のステータスを示し、2 番目のテーブルのステータスは、すべてのメンバ・タイプに適用されます。

タイプ	ステータス	説明
フェイルオーバー	プライマリ	現在のプライマリ。
フェイルオーバー	バックアップ	バックアップとしてプライマリに接続
フェイルオーバー	障害中	バックアップとの接続が切断されたために、プライマリが障害になった状態。プライマリが一時的または永久的な障害状態になりうるさまざまな状況の詳細は、“ 自動フェイルオーバーのメカニズム ” を参照してください。
災害復旧	Connected	非同期としてプライマリに接続
読み取り専用レポート	接続しました	(同上)
読み書き可能なレポート	接続しました	(同上)
不確定	初期化されていない	メンバが初期化されていません (ミラー構成がまだロードされていません)。

ステータス (任意のタイプ)	説明
移行	初期化やその他の操作が完了したときの、その後すぐに変わる遷移状態。この状態では、メンバの状態を検索するプロセスがすぐに再検索を実行します。 動作しているプライマリがない場合、フェイルオーバー・メンバは、プライマリになる過程でジャーナル・ファイルの取得と適用を続けている間、このステータスをレポートできます。プライマリである別のフェイルオーバー・メンバがある場合、ステータスは代わりに同期中になります。
同期	停止または切断された後の起動または再接続。バックアップになるまたは接続される前に、データベースをジャーナルの状態と同期させるために、ジャーナル・ファイルを取得し、適用します。
待機中	プライマリやバックアップになる、または接続されるなどのアクションを完了できない状態。無限に再試行されますが、ユーザの介入が必要な場合があります。詳細は、メッセージ・ログを参照してください。
停止	ユーザにより、メンバのミラーリングが無期限に 停止 され、自動的に開始されません。詳細は、メッセージ・ログを参照してください。
クラッシュ	不測の事態によりミラー実行不可。詳細は、メッセージ・ログを参照してください。
Error	メンバのステータスをフェッチ中に予期しないエラーが発生しました。
ダウン	他のメンバに表示された、ダウンしたまたはアクセスできないメンバ。

注釈 ミラー・メンバの [タイプ] と [ステータス] は、`$SYSTEM.Mirror.GetMemberType()` メソッドおよび `$SYSTEM.Mirror.GetMemberStatus()` メソッドを使用して取得することもできます。上記にリストされていない [タイプ] と [ステータス] の一部の組み合わせは、以下の呼び出しによって報告されます。

- ・ [メンバではない] および [初期化されていない] – インスタンスがミラー・メンバ以外として構成されています。
- ・ [読み取り専用] または [読み取り/書き込みレポート] および M/N [ステータス] – インスタンスがいくつかのミラーの非同期メンバです。mirrorname 引数を指定して特定のミラーの [ステータス] を取得してください。

バックアップおよび非同期ミラー・メンバの場合、[ジャーナル転送] は、プライマリからの最新ジャーナル・データがミラー・メンバにあるかどうかを示します。データがない場合、ジャーナル転送の遅延の度合いを示します。一方、[デジャーナリング] は、プライマリから受信したジャーナル・データがすべてデジャーナル (そのメンバのミラーリングされるデータベースに適用) されているかどうかを示します。デジャーナルされていない場合、デジャーナリングの遅延の度合いを示します。以下のテーブルは、ミラー・モニタと `^MIRROR` で表示される可能性のある、これらのフィールドのステータスを示しています (プライマリの場合、これらのフィールドは常に [N/A] です)。

ジャーナル転送	説明
アクティブ (バックアップのみ)	バックアップは、プライマリから最新のジャーナル・データを受信しており、プライマリと同期されています。([アクティブ] バックアップ・ステータスに関する詳細は、“ バックアップ・ステータスと自動フェイルオーバー ” を参照してください。)[デジャーナリング] ステータスが [キャッチアップ済み] でない場合でも、バックアップは [アクティブ] になることがあります。必要なジャーナル・ファイルをバックアップがすべて有している限り、プライマリとの接続を失った後でもデジャーナルできます。
キャッチアップ済み	バックアップの場合、バックアップは最新のジャーナル・データをプライマリから受信していますが、プライマリがジャーナル・データの受信確認を待機していないという点で完全には同期されていないことを意味します。多くの場合、このステータスは、バックアップがミラーに再接続しているときなど、一時的なものです。 非同期の場合は、プライマリから最新のジャーナル・データを受信していて、プライマリと同期されていることを意味します。
time 遅延	メンバは明示されている時間だけプライマリから遅延しています。time は、メンバが最後に受信したジャーナル・ブロックのタイムスタンプから現在の時刻までに経過した時間を表します。
time に切断	メンバは、明示されている時刻にプライマリから切断されました。

デジャーナリング	説明
キャッチアップ済み	プライマリから受信したジャーナル・データすべてがデジャーナルされています (メンバのミラーリングされるデータベースに適用されています)。
time 遅延	プライマリから受信したジャーナル・データの一部がまだデジャーナルされていません。time は、最後にデジャーナルされたジャーナル・ブロックのタイムスタンプと、プライマリから最後に受信したジャーナル・ブロックのタイムスタンプとの間で経過した時間を表します。
time に切断	メンバは、明示されている時刻にプライマリから切断されました。
警告: データベースの一部に対処が必要です。	1 つ以上のミラーリングされるデータベースが正常な状態ではありません。データベースを確認する必要があります。
警告: デジャーナリングが停止しました。	オペレータまたはエラーの発生によりデジャーナリングが停止しています。“ データベース・デジャーナリングの管理 ” を参照してください。

記載のとおり、[ジャーナル転送] フィールドの [アクティブ] は、バックアップがプライマリからジャーナル・データをすべて受信しており、プライマリと同期されていることを示しています。このため、バックアップは、フェイルオーバーの発生時に、追加ジャーナル・データを取得するためにプライマリの ISCAgent と通信する必要なく、プライマリから引き継ぐことができます。

[アクティブ] なバックアップ・フェイルオーバー・メンバの場合の [デジャーナリング] フィールドの [キャッチアップ済み]、および非同期メンバの場合の [デジャーナリング] フィールドと [ジャーナル転送] フィールドの [キャッチアップ済み] は、そのメンバがプライマリから最新のジャーナル・データを受信して、そのデータに含まれる最新のグローバル処理を適用したことを表します。そのメンバがキャッチアップされていない場合、最新ジャーナル・データが生成された時点からの経過時間または最新処理をプライマリで記述した時点からの経過時間が代わりに表示されます。

5.1.1.2 受信ジャーナルの転送速度

以下に示すバックアップおよび非同期のミラー・メンバのステータス・リストは、プライマリから到着したジャーナル・データ
の速度です。この速度は、[ミラー・モニタ]の前の更新以降のもので、[このメンバの受信ジャーナルの転送速度]の下
に表示されます。

[ミラー・モニタ] ページを最初にロードしたときには、この領域に [--- (更新時に表示)] というテキストが表示されます。
次にページが更新されたとき、表示される情報は、以下に示すように、受信ジャーナル・データが圧縮されているかどう
かによって異なります (“ミラー・トラフィックの圧縮” を参照)。

- ・ ジャーナル・データが圧縮されていない場合、受信ジャーナル・データの速度は、キロバイト (KB) 毎秒で示されま
す。以下に例を示します。

42345 KB/s (22s 間隔)

- ・ 受信ジャーナル・データが圧縮されている場合は、受信圧縮データの速度、受信ジャーナル (非圧縮) データの速
度、および圧縮データ速度に対する非圧縮データ速度の比率が表示に含まれます。以下に例を示します。

14288 KB/s ネットワーク; 39687 KB/s ジャーナル; 比率 2.78:1 (143s 間隔)

5.1.1.3 ミラーリングされるデータベースのステータス

重要 バックアップおよび DR 非同期メンバの場合は、[ミラー・モニタ] ページの [見つからないミラーリングさ
れるデータベースのレポート] に、プライマリに存在していても、現在のメンバには存在していないミラー
リングされるデータベースについての警告が表示されます。これは、バックアップとして非常に重要なこ
とです。DR 非同期がバックアップに昇格したときに、ミラーリングされるデータベースの完全なセットが
揃っていないと、プライマリが停止したときに、その役割を正常に引き継ぐことができなくなります。見つ
からないデータベースごとに、完全なミラー・データベース名がリストされます。見つからないデータベー
スが存在しない場合、[見つからないミラーリングされるデータベースのレポート] は表示されません。

すべてのメンバに関して、[ミラー・モニタ] ページの [ミラーリングされるデータベース] リストには、リストされている各デー
タベースについて、次のいずれかのステータスが表示されます。

ステータス	説明
正常 (プライマリのみ)	ミラーリングされるデータベースは書き込み可能 (読み取り専用データベースではない場合) で、グローバル更新がジャーナルされています。
デジャーナリング (バックアップおよび非同期)	データベースはアクティブ化され、キャッチアップされており、ミラーがデータベースにジャーナル・データを適用しています。
要キャッチアップ	データベースはアクティブ化されていますが、まだキャッチアップされていません。ユーザ開始のキャッチアップ操作が必要です。
要有効化	データベースはまだアクティブ化されていません。ユーザ開始のアクティブ化操作とキャッチアップ操作が必要です。
キャッチアップ実行中	データベースでユーザ開始のキャッチアップ操作が実行されています。
デジャーナリング停止	オペレータまたはエラーによってデジャーナリングが停止されています。“ バックアップおよび非同期メンバのミラーリングの停止 ” および “ データベース・デジャーナリングの管理 ” を参照してください。
データベース・ディスマウント	データベースがディスマウントされています。
独立 (バックアップおよび非同期)	ミラーリングされるデータベースがプライマリに存在しません。
廃止	ミラーリングされるデータベースは廃止されているため、ミラーから削除する必要があります。

プライマリでは、データベースのステータスが [正常] の場合、[次にデジャーナルするレコード] 列には、[N/A] が表示されます。それ以外の場合、この列には以下の情報が表示されます。

- ・ [時刻] は、データベースに適用される次のジャーナル・レコードの先頭のタイムスタンプです。または、これがプライマリの現在のジャーナル位置に一致する場合は [現在] と表示されます。
- ・ [ファイル名] は、次に適用されるジャーナル・レコードが含まれるミラー・ジャーナル・ファイルの名前です。
- ・ [オフセット] は、次に適用されるジャーナル・レコードの先頭の、ジャーナル・ファイル内における位置です。

データベースのステータスおよびそれに関連する操作 (有効化およびキャッチアップ) については、“[ミラーリングされるデータベースの有効化とキャッチアップ](#)” で説明しています。これらの操作は、リストの下にあるドロップダウンで選択できます。このドロップダウンを使用して、ディスマウントされているデータベースをマウントできます (ただし、マウントされているデータベースのディスマウントには使用できません)。[\[削除\]](#) リンクを使用して、またはドロップダウンから [\[削除\]](#) を選択して、リストされているデータベースをミラーから削除することができます。詳細は、“[ミラーリングされるデータベースをミラーから削除する](#)” を参照してください。

5.1.2 ^MIRROR ステータス・モニタの使用法

^MIRROR ルーチンには、文字ベースのミラー・ステータス・モニタが用意されています。^MIRROR の [ステータス・モニタ] オプションでは、タイプ、ステータス、ジャーナル転送遅延、およびデジャーナル遅延を含む、ミラー・メンバのステータスを表示できます (“[ミラー・メンバのジャーナル転送およびデジャーナリングのステータス](#)” を参照)。このモニタは任意のミラー・メンバで実行できます。ただし、これをフェイルオーバー・メンバで実行すると、アービター構成に関する情報およびすべての接続非同期メンバに関する情報が表示されますが、これらの情報は、非同期メンバで実行する場合には表示されません。

このステータス・モニタを起動するには、ターミナル・ウィンドウを開き、%SYS ネームスペースで `MIRROR ルーチン` を実行し（“[MIRROR ルーチンの使用法](#)”を参照）、[ミラー・ステータス] メニューから [ステータス・モニタ] を選択します。フェイルオーバー・メンバで実行した場合のモニタからの出力のサンプルを以下に示します。

```
Status of Mirror MIR25FEB at 17:17:53 on 02/27/2018
```

Member Name+Type	Status	Journal Transfer	Dejournaling
MIR25FEB_A Failover	Primary	N/A	N/A
MIR25FEB_B Failover	Backup	Active	Caught up
MIR25FEB_C Disaster Recovery	Connected	Caught up	Caught up
MIR25FEB_D Read-Only Reporting	Connected	Caught up	Caught up

```
Arbiter Connection Status:
  Arbiter Address: 127.0.0.1|2188
  Failover Mode: Arbiter Controlled
  Connection Status: Both failover members are connected to the arbiter

Press RETURN to refresh, D to toggle database display, Q to quit,
or specify new refresh interval <60>
```

非同期メンバでステータス・モニタを実行すると、フェイルオーバー・メンバおよび対象の非同期メンバのみがリストされ、非同期メンバの [デジャーナリング](#)・ステータス (running または stopped) も表示されます。以下に例を示します。

```
Status of Mirror MIR25FEB at 17:17:53 on 02/27/2018
```

Member Name+Type	Status	Journal Transfer	Dejournaling
MIR25FEB_A Failover	Primary	N/A	N/A
MIR25FEB_B Failover	Backup	Active	Caught up
MIR25FEB_C Disaster Recovery	Connected	Caught up	Caught up

```
Dejournal Status: running (process id: 12256)

Press RETURN to refresh, D to toggle database display, Q to quit,
or specify new refresh interval <60>
```

既定では、ミラーリングされるデータベースの情報は表示されません。“[ミラー・モニタの使用法](#)”で説明しているように、プロンプトで **a** を入力すると、名前、ディレクトリ、ステータス、および次にデジャーナルされるレコードを含め、ミラー内の各データベースに関する情報が表示されます。

```
Mirror Databases:
```

Name	Directory path	Status	Last Record Dejournalled
MIR25FEB_DB1	C:\InterSystems\20182209FEB25A\Mgr\MIR25FEB_DB1\	Active	
Current, c:	\intersystems\20182209feb25a\mgr\journal\MIRROR-MIR25FEB-20180227.001,40233316		
MIR25FEB_DB2	C:\InterSystems\20182209FEB25A\Mgr\MIR25FEB_DB2\	Active	
Current, c:	\intersystems\20182209feb25a\mgr\journal\MIRROR-MIR25FEB-20180227.001,40233316		

5.1.3 ミラーリング通信プロセスの監視

ミラー通信および同期化を行う各システム (プライマリ・フェイルオーバー・メンバおよびバックアップ・フェイルオーバー・メンバと、接続されている各非同期メンバ) 上で実行されるプロセスがあります。

詳細は、次のトピックを参照してください。

- ・ [プライマリ・フェイルオーバー・メンバ上でのミラーリング・プロセス](#)
- ・ [バックアップ・フェイルオーバー・メンバおよび非同期メンバ上でのミラーリング・プロセス](#)

5.1.3.1 プライマリ・フェイルオーバー・メンバ上でのミラーリング・プロセス

プライマリ・フェイルオーバー・メンバ上でシステム・ステータス・ルーチン(%SS)を実行すると、次のテーブルにリストされたプロセスが表示されます。

注釈 ここでは、CPU、Glob、および Pr の各列を %SS の出力から意図的に省略しています。

テーブル 5-1: プライマリ・フェイルオーバー・メンバ上でのミラーリング・プロセス

デバイス	ネームスペース	ルーチン	ユーザおよび場所
/dev/null	%SYS	MIRRORMGR	Mirror Master
MDB2	%SYS	MIRRORCOMM	Mirror Primary*
192.168.1.1	%SYS	MIRRORCOMM	Mirror Svr:Rd*

プロセスは次のように定義されます。

- ・ **Mirror Master** : システム起動時に開始されるこのプロセスでは、さまざまなミラー制御タスクおよびミラー管理タスクが実行されます。
- ・ **Mirror Primary** : これは、発信データ・チャンネルであり、単方向チャンネルです。接続されているシステム (バックアップ・フェイルオーバー・メンバまたは非同期メンバ) ごとに 1 つのジョブがあります。
- ・ **Mirror Svr:Rd*** : これは、受信応答チャンネルで、単方向チャンネルです。接続されているシステム (バックアップ・フェイルオーバー・メンバまたは非同期メンバ) ごとに 1 つのジョブがあります。

接続されている各非同期メンバは、プライマリ・フェイルオーバー・メンバにおいて、**Mirror Master**、**Mirror Primary**、および **Mirror Svr:Rd*** の各プロセスの新しいセットになります。

5.1.3.2 バックアップ・フェイルオーバー・メンバおよび非同期メンバ上でのミラーリング・プロセス

バックアップ・フェイルオーバー/非同期メンバ上でシステム・ステータス・ルーチン(%SS)を実行すると、次のテーブルにリストされたプロセスが表示されます。

テーブル 5-2: バックアップ・フェイルオーバー・メンバおよび非同期メンバ上でのミラーリング・プロセス

デバイス	ネームスペース	ルーチン	ユーザおよび場所
/dev/null	%SYS	MIRRORMGR	Mirror Master
/dev/null	%SYS	MIRRORMGR	Mirror Dejour
/dev/null	%SYS	MIRRORMGR	Mirror Prefet*
/dev/null	%SYS	MIRRORMGR	Mirror Prefet*
MDB1	%SYS	MIRRORMGR	Mirror Backup
/dev/null	%SYS	MIRRORMGR	Mirror JrnRead

この表で識別されるプロセスは、接続された各非同期メンバでも表示されます。

- ・ **Mirror Master** : システム起動時に開始されるこのプロセスでは、さまざまなミラー制御タスクおよびミラー管理タスクが実行されます。

- ・ **Mirror JrnRead** (ミラー・ジャーナル読み取り) : このプロセスは、バックアップで生成されメモリに書き込まれたジャーナル・データを読み取り、それらの変更をデジャーナル・ジョブによってデジャーナルするためにキューイングします。
- ・ **Mirror Dejour** (ミラー・デジャーナル) : これは、バックアップ・フェイルオーバー・メンバでのデジャーナル・ジョブであり、受信したジャーナル・データからミラーリングされるデータベースに set および kill を発行します。
- ・ **Mirror Prefetch** (ミラー事前フェッチ) : これらのプロセスは、デジャーナル・ジョブが実際にディスク・ブロックを使用しようとする前に、デジャーナル・ジョブに必要なディスク・ブロックをメモリに事前フェッチします。これは、デジャーナル・プロセスを高速化するために実行されます。通常、システムには複数のミラー事前フェッチ・ジョブが構成されます。
- ・ **Mirror Backup** (ミラー・バックアップ) : このプロセスは、プライマリから受信したジャーナル・レコードをバックアップのミラー・ジャーナル・ファイルに書き込み、応答をプライマリに送信する双方向チャンネルとなります。

5.2 ミラー・メンバのネットワーク・アドレスの更新

1 つ以上のミラー・メンバ (プライマリを含む) の 1 つ以上のネットワーク・アドレスを更新する必要がある場合 (“フェイルオーバー・メンバの編集または削除” を参照)、この情報は、一般にプライマリで変更されます。変更を保存すると、プライマリは接続されているすべてのミラー・メンバに、その変更を伝播します (接続されていないメンバには、再接続されたときに変更を伝播します)。接続されているバックアップまたは非同期メンバでは、いずれのミラー・メンバのネットワーク・アドレスも変更できません。ミラー・メンバは、このような情報をすべてプライマリから受信する必要があるためです。この一般的なケースには、以下に示すように、いくつかの例外があります。

- ・ InterSystems IRIS インスタンスのスーパーサーバ・ポートは、汎用構成の一部であるため、ローカルに変更する必要があります。そのため、ミラー・メンバのスーパーサーバ・ポートは、常にメンバ自体で更新されるミラー・ネットワーク情報になります。プライマリのスーパーサーバ・ポートを変更する場合はプライマリの [ミラーの編集] ページに移動して、バックアップの場合はバックアップの [ミラーの編集] ページに移動します。その他についても同様になります。

注釈 [ネットワーク・アドレスを編集] ダイアログで、ローカル・メンバのスーパーサーバ・ポートに対応する [ポートを編集] リンクをクリックすると、管理ポータル [メモリと開始設定] ページが含まれているダイアログが表示され、ポート番号を変更できます。ただし、ミラー・メンバのスーパーサーバ・ポートを変更するために、このページに直接移動しないでください。[ミラーの編集] ページまたは [非同期構成の編集] ページと [ネットワーク・アドレスの編集] ダイアログを必ず使用して、この変更を実施してください。

- ・ フェイルオーバー・メンバまたは非同期メンバが切断されているときに、プライマリのネットワーク・アドレスが変更された場合、まず、すべてのミラーのネットワーク・アドレスが現在のプライマリ上で正しいことを確認してから、切断されているメンバでプライマリのネットワーク・アドレスを更新する必要があります (“フェイルオーバー・メンバの編集または削除” または “非同期メンバの編集または削除” を参照)。プライマリのネットワーク情報の更新後に、メンバがミラーに再接続できるようにするために、切断されているそのメンバの再起動が必要な場合があります。
- ・ どちらのフェイルオーバー・メンバもプライマリとしては動作していないなどの場合に、いずれかのフェイルオーバー・メンバがプライマリになれるようにするために、そのフェイルオーバー・メンバ上でネットワーク・アドレスを更新する必要があります。プライマリになると、接続時に他のメンバにこれらのアドレスが伝播されます。ネットワーク・アドレスの更新後に、メンバがプライマリになれるようにするために、そのメンバの再起動が必要な場合があります。

注釈 “非同期ミラー・メンバを構成する” で説明しているように、非同期メンバをミラーに追加するときに指定する [非同期メンバ・アドレス] が、その非同期のスーパーサーバ・アドレスおよびミラー・プライベート・アドレスになります (“ミラー・メンバのネットワーク・アドレス” を参照)。例えば、DR 非同期のミラー・プライベート・アドレスをミラー・プライベート・ネットワークに配置し、そのスーパーサーバ・アドレスは外部ネットワークに置いたままにするなど、ここで説明しているように、ミラーに非同期を追加した後で、その非同期のアドレスを変更できます。

5.3 ネットワーク・アドレス検証エラーの解決

ミラー・メンバが起動すると、ミラーの初期化処理によって、プライマリでミラーに構成されているミラー・プライベート・アドレスまたはスーパーサーバ・アドレス ("[ミラー・メンバのネットワーク・アドレス](#)" を参照) でインスタンスにアクセスできることが確認されます。これが成功せず、検証が失敗すると、メンバはミラーに参加できなくなります。ネットワーク・アドレス検証エラーが生じる状況としては、次の 2 つがあります。

- ・ ローカル・インスタンスの構成済みミラー・プライベート・アドレスでアクセスしたインスタンスがローカル・インスタンスではない。ローカル・インスタンスは、例えばバックアップとリストアや VM のクローン化などを通して、別のホストからコピーされているため、この状況が生じる可能性は高いです。
- ・ ローカル・インスタンスに構成されているミラー・プライベート・アドレスで通信できるインスタンスがない。この状況が生じる原因は、次のいずれかの場合があります。
 - ミラー・メンバ・ホストのネットワーク構成が変更されている (IP アドレスの変更など)。この場合、"[ミラー・メンバのネットワーク・アドレスの更新](#)" で説明しているようにプライマリでメンバの構成済みアドレスを更新できます。その後、ローカル・インスタンスを再起動します。
 - 構成されているアドレスのホストがダウンしている。この場合、オンラインに戻ったときに、前の問題 (アドレスの不一致) が生じる可能性は高いです。

これらいずれかの検証エラーが生じると、メッセージ・ログに該当するメッセージが表示され、検証の問題の内容が [[ミラー・モニタ](#)] ページと [[ミラーの編集](#)] ページで説明されます。両方のページに、状況の解決に使用できる以下の 2 つのリンクが提示されます。

- ・ [[ローカル・ミラー構成の削除](#)] リンクを使用すると、"[ミラー構成の編集または削除](#)" で説明しているように、ローカルの (無効な) ミラー構成を削除できます。
- ・ [[新しいメンバとしてミラーに参加](#)] リンクを使用して、"[ミラーの作成](#)" で説明しているように、インスタンスを新しいメンバとしてミラーに追加します。これによって、無効なローカル・ミラー構成が、有効な構成に置き換えられます。

この種類の検証エラーが発生すると、これらのオプションのいずれかを選択するときに表示されるダイアログに、問題の説明情報と解決の関連手順も示されます。

もう 1 つのオプションは、当面何もしないことです。これを選択して問題を調査することも、当該ミラー・メンバに構成されているネットワーク・アドレスを更新することもできます。

注釈 ネットワーク・アドレス検証エラーは、`^Mirror` ルーチンを使用して解決することもできます ("[MIRROR ルーチンの使用法](#)" を参照)。

5.4 X.509 DN 更新の承認 (TLS のみ)

TLS を使用するようにミラーを構成する場合は、新しく追加した第 2 のフェイルオーバー・メンバと、第 1 フェイルオーバー・メンバの新しい各非同期メンバをミラーに参加させるには、まずそれらを承認する必要があります ("[第 2 のフェイルオーバー・メンバまたは非同期メンバを承認する \(TLS ミラーのみ\)](#)" を参照)。同様の理由から TLS を使用するミラーのメンバを X.509 証明書と DN で更新する場合、この更新は、以下のいずれかの方法で、他のメンバに伝達され、承認される必要があります。

- ・ プライマリの X.509 DN 更新は、更新が行われた時点で、プライマリに接続された他のミラー・メンバに自動的に伝達され、承認されます。

- ・ プライマリの X.509 DN 更新時に、バックアップまたは非同期メンバがプライマリに接続されていない場合、その更新はプライマリへの次の接続時に、そのメンバの **[保留中の DN 更新の承認]** リストに追加されます。メンバが引き続きミラーの一部にできるようにするには、**[ミラーの編集]** ページ (バックアップ) または **[非同期の構成を編集]** ページ (非同期) の **[保留中の DN 更新を承認]** リンクをクリックして、更新を承認する必要があります。バックアップまたは非同期メンバは、プライマリからの X.509 DN 更新を拒否できません。
- ・ バックアップまたは非同期の X.509 DN 更新は、メンバがプライマリに接続されたとき、または次回メンバがプライマリに接続したときに、プライマリの **[保留中の DN 更新の承認/拒否]** リストにすぐに表示されます。メンバが引き続きミラーの一部にできるようにするには、プライマリの **[ミラーの編集]** ページで **[保留中の DN 更新の承認/拒否]** リンクをクリックし、**[承認]** を選択することで、更新を承認する必要があります。

注釈 `^MIRROR` ルーチンの **[ミラー構成]** メニューにある **[保留中の DN 更新の承認/拒否]** オプション (プライマリの場合) または **[保留中の DN 更新を承認]** オプション (バックアップまたは非同期の場合) も、`SYS.Mirror.AuthorizePendingMembers()` API 呼び出しと同様に X.509 DN 更新の承認に使用できます。

5.5 DR 非同期メンバのフェイルオーバー・メンバへの昇格

災害復旧 (DR) 非同期ミラー・メンバはフェイルオーバー・メンバに昇格することができ、2 メンバ構成の場合は現在のフェイルオーバー・メンバに取って代わり、1 メンバ構成の場合は、現在のメンバに結合します。例えば、フェイルオーバー・メンバのいずれかが、計画的メンテナンスもしくは障害の結果により、長期間ダウンすることになる場合、一時的に DR 非同期を昇格させて、代替させることができます (**“昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え”** を参照)。実際の災害復旧の間、両フェイルオーバー・メンバに障害が発生してしまった場合、DR を昇格させることで、ある程度のデータ損失のリスクは許容しつつ、DR がプライマリ・フェイルオーバー・メンバとしてプロダクションを引き継ぐことが可能になります。詳細は、**“災害復旧時の昇格した DR 非同期への手動フェイルオーバー”** を参照してください。

DR 非同期がフェイルオーバー・メンバに昇格すると、可能な場合、フェイルオーバー・パートナーとして最新プライマリとペアになります。これを自動で行うことができない場合、フェイルオーバー・パートナー選択のオプションが用意されます。昇格の後、フェイルオーバー・メンバが起動時にするように、昇格したメンバはフェイルオーバー・パートナーの ISCAgent と通信してまず直近ジャーナル・データを取得した後、フェイルオーバー・パートナーがプライマリでない場合はプライマリとなり、フェイルオーバー・パートナーがプライマリである場合はバックアップとなります。昇格したメンバは、フェイルオーバー・パートナーとの通信による直近ジャーナル・データ取得ができない限り、自動的にプライマリとなることはできません。

DR 非同期をフェイルオーバー・メンバに昇格する際、留意する必要がある以下の重要な考慮事項があります。

- ・ DR 非同期の位置によっては、フェイルオーバー・パートナーとの間におけるネットワーク遅延が受け入れられないほど長くなる場合があります。フェイルオーバー・メンバ間における遅延要件に関する詳細は、**“ネットワーク遅延に関する考慮事項”** を参照してください。
- ・ DR 非同期がフェイルオーバー・メンバになるときは、フェイルオーバー・メンバの圧縮設定が適用されます。昇格前と同じ、非同期メンバの圧縮が適用されるわけではありません (この設定の詳細は、**“ミラー・トラフィックの圧縮”** を参照してください)。ネットワーク構成によっては、フェイルオーバー・メンバの圧縮設定に調整が必要になることがあります。最適なミラー機能についての詳細は、**“フェイルオーバー・メンバの編集または削除”** を参照してください。
- ・ ミラー・プライベート・ネットワークを、フェイルオーバー・メンバのミラー・プライベート・アドレスの接続に使用しているときは、**“ミラーリング・アーキテクチャとネットワーク構成のサンプル”** で説明しているように、このネットワークに接続されていない DR 非同期の昇格は、プライマリとして機能することを目的とし、ほかに動作中のフェイルオーバー・メンバがない場合にのみ実行する必要があります。プライマリが動作中だがプライマリのミラー・プライベート・アドレスにアクセスできないときに DR 非同期を昇格する場合、バックアップになることはできません。ただし、プライマリのエージェントからジャーナル・データを取得できるため、プライマリがシャットダウンされたときには最新ジャーナル・データを使用してプライマリになることができます。

- ミラー VIP が使用中であり、昇格された DR 非同期が VIP のサブネット上にない場合にこの DR がプライマリになるときは、何らかの代替方法を使用してユーザの接続を昇格された DR にリダイレクトする必要があります。例えば、VIP ではなく、DR 非同期の IP をポイントするように DNS 名を手動で更新するか、“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)” で説明したメカニズムのいずれかを構成します。

ただし、障害復旧の状況によっては、昇格された DR 非同期が既存のフェイルオーバー・メンバのエージェントに通信できない場合があります。このような場合は、フェイルオーバー・パートナーなしで DR を昇格するオプションがあります。このセクションの “Promotion With Partner Selected by User” の説明を参照してください。つまり、DR は、既に受け取っているジャーナル・データと、その他の接続されているミラー・メンバで使用可能な最新のジャーナル・データ (存在する場合) のみで、プライマリに昇格できるということです。この場合、新しいプライマリがミラーにより生成されたジャーナル・データすべてを所有しているとは限らないので、アプリケーション・データが一部損失することがあります。このように昇格された DR 非同期がプライマリとして機能している間、以前のフェイルオーバー・パートナーを再起動する場合、その再構築が必要ことがあります。詳細は、“[ミラー・メンバの再構築](#)” を参照してください。このセクションで後述する DR の昇格手順で、詳細を必ず確認してください。

注釈 プライマリの InterSystems IRIS インスタンスが、アービター制御モードでバックアップとアービターの両方から遮断されたことにより無限の障害状態にある場合、“[自動フェイルオーバーのメカニズムの詳細](#)” で説明しているように、DR 非同期をフェイルオーバー・メンバに昇格できません。

自動選択されたパートナーでの昇格

可能である場合、昇格した DR 非同期のフェイルオーバー・パートナーは、以下のように自動で選択されます。

- 動作中のプライマリ・フェイルオーバー・メンバがある場合、そのプライマリがフェイルオーバー・パートナーとして自動で選択されます。昇格したメンバはそのパートナーより直近ジャーナル・データを取得して、バックアップとなります。InterSystems IRIS が現在のバックアップ上で動作している場合、メンバは直ちに DR 非同期に降格されます。動作していない場合、InterSystems IRIS の再起動時に、メンバは DR 非同期に降格されます。
- InterSystems IRIS がいずれのフェイルオーバー・メンバ上でも動作していないが、両フェイルオーバー・メンバ (1 つのみの場合は 1 つ) 上の ISCAgents に通信可能である場合、直近のプライマリがフェイルオーバー・パートナーとして自動選択され、昇格したメンバはそのパートナーより直近ジャーナル・データを取得して、プライマリとなります。InterSystems IRIS が以前のプライマリで再起動した場合、そのプライマリは自動的にバックアップとなります。InterSystems IRIS が以前のバックアップで再起動した場合、そのバックアップは自動的に DR 非同期となります。

ユーザが選択したパートナーでの昇格

InterSystems IRIS がいずれのフェイルオーバー・メンバでも動作しておらず、少なくとも 1 つの ISCAgent に通信不可能である場合、昇格プロシージャが通信不可能なエージェントを通知して、フェイルオーバー・パートナーの選択オプションを提示します。データ損失の可能性を避けるため、エージェントに通信不可能である場合でも、最後にプライマリだったフェイルオーバー・メンバを選択する必要があります。ユーザの選択および ISCAgent の利用可能性により、以下のように結果が異なります。

- ユーザの選択したパートナーのエージェントに通信可能である場合、昇格した DR 非同期がそのエージェントより直近ジャーナル・データを取得した後、プライマリとなります。InterSystems IRIS がそのパートナーで再起動されると、そのパートナーは自動的にバックアップとなります。
- ユーザの選択したパートナーのエージェントが通信不可能である場合、そのパートナーのエージェントに通信して、直近ジャーナル・データを取得できるまで、昇格した DR 非同期はプライマリにはなりません。パートナーのエージェントが使用可能になる前であればいつでも、直近のジャーナル・データを取得せずに、昇格したメンバを強制的にプライマリ化することができます (“[災害復旧時の昇格した DR 非同期への手動フェイルオーバー](#)” を参照)。ただし、結果として、アプリケーション・データが一部損失するおそれがあります。
- フェイルオーバー・パートナーを選択しない場合、昇格される DR 非同期は、プライマリになる前に、接続されている非同期ミラー・メンバのすべてから最新の使用可能なジャーナル・データを取得しようとします。昇格される DR 非同期よりも新しいジャーナル・データを保持している接続されたメンバが存在しないことがあるため、一部のアプリケーション・データが失われる可能性があります。

これを選択するときに、昇格される DR 非同期でフェイルオーバーなしの状態を設定するオプションがあります。このオプションを選択すると、接続されている別のメンバからのジャーナル・データの取得を含め、プライマリになるための準備は行われますが、フェイルオーバーなしを取り消すまでプライマリになることはありません。これにより、必要とする追加の検証を実行して、可能であれば追加のメンバをオンラインにして、昇格される DR 非同期をプライマリにする前に、より新しい潜在的なジャーナル・データを使用できるようにします。

注釈 ジャーナル・データを確認するためにミラー・メンバとの通信を試行したときの成功および失敗のメッセージと、最新データが識別された場合にデータ取得を試行したときの成功および失敗のメッセージは、メッセージ・ログに格納されます。

注意 DR 非同期が昇格されたときに ISCAgent がダウンしていた旧フェイルオーバー・メンバでは、InterSystems IRIS インスタンスの構成パラメータ・ファイルの [MirrorMember] セクションで ValidatedMember=0 を設定するまで、InterSystems IRIS を再起動しないでください。これについては、以下の DR の昇格手順で説明します。

フェイルオーバー・パートナーが自動で選択されない場合、以下のルールが適用されます。

- ・ パートナーとして選択されない旧フェイルオーバー・メンバはいずれも、InterSystems IRIS の再起動時に DR 非同期メンバとなります。
- ・ InterSystems IRIS インスタンスの再起動前のできるだけ早い段階において、DR 非同期の昇格時に自身のエージェントが通信不可能であった任意の旧フェイルオーバー・メンバ上で、InterSystems IRIS インスタンスの構成パラメータ・ファイルの [MirrorMember] セクションで ValidatedMember=0 を設定する必要があります (“[MirrorMember]” を参照してください)。これにより、InterSystems IRIS インスタンスに対して、以前のロールでのミラーへの再接続ではなく、昇格した DR 非同期からのミラー内での新規ロール取得が指示されます。^MIRROR ルーチンにより、この変更が必要なフェイルオーバー・メンバがリストされます。

注意 直近ジャーナル・データの取得なしで、昇格した DR 非同期がプライマリになるか強制的にプライマリになった場合、グローバル更新処理の一部が失われることがあり、他のミラー・メンバの再構築が必要になることがあります (“[ミラー・メンバの再構築](#)” に記載のとおり)。しかし、一部の災害復旧シナリオでは、直近ジャーナル・データの取得なしで、DR 非同期がプライマリに昇格するための代替手段がないことがあります。昇格プロセスの機能について不明な点がある場合、[インターシステムズのサポート窓口](#)に問い合わせることをお勧めします。

DR 非同期メンバをフェイルオーバー・メンバに昇格させるには、以下の手順を実行します。

1. フェイルオーバー・メンバに昇格させる DR 非同期メンバで、[システム処理]→[ミラー・モニタ] ページに移動して、ミラー・モニタを表示します。
2. ページ上部の [フェイルオーバー・メンバに昇格] ボタンをクリックします。
3. その結果表示されるダイアログ・ボックスの指示に従ってください。最も単純な例では、昇格処理を続行するかの確認のみがこの指示に含まれますが、このセクションで前述したとおり、それにはフェイルオーバー・パートナーを選択するかしないかが含まれる場合があります。
4. [ミラーに VIP が構成されている](#)場合、昇格された DR 非同期は、プライマリになるときに (手動フェイルオーバーにより、またはバックアップとして動作中にプライマリが後で停止したことにより) VIP を取得できるように、VIP のサブネット上にネットワーク・インタフェースを有している必要があります。
 - ・ この DR 非同期が VIP のサブネット上にインタフェースを 1 つだけ有している場合、プロセスによってこのインタフェースが自動的に選択されます。
 - ・ この DR 非同期が VIP のサブネット上に複数のインタフェースを有している場合、プロセスからインタフェースの選択を求められます。

- ・ この DR 非同期が VIP のサブネット上にインタフェースを有していない場合、昇格プロシージャからこの事実が警告され、続行する前に確認を求められます。このプロシージャを使用して続行し、DR 非同期を昇格する場合、ユーザとアプリケーションが新しいプライマリに接続できるように、例えば、VIP ではなく DR 非同期の IP をポイントするように DNS 名を更新するなど、手動ステップを実行する必要があります。

5. DR 非同期の昇格時に、旧フェイルオーバー・メンバのエージェントが有効である場合、InterSystems IRIS インスタンスの構成パラメータ・ファイルの [MirrorMember] セクションで ValidatedMember=0 が自動設定されます (“[MirrorMember]” を参照してください)。これにより、InterSystems IRIS インスタンスに対して、以前のロールでのミラーへの再接続ではなく、昇格した DR 非同期からのミラー内での新規ロール取得が指示されます。

昇格時に以前のフェイルオーバー・メンバのエージェントと通信できない場合、この変更は自動では行われません。したがって、InterSystems IRIS インスタンスが再起動する前のできるだけ早い段階において、昇格時にエージェントが通信不可能であった任意の旧フェイルオーバー・メンバで、InterSystems IRIS インスタンスの構成パラメータ・ファイルの編集により、手動で ValidatedMember=0 を設定する必要があります。この指示により、この変更が必要な以前のフェイルオーバー・メンバがリストされます。

注意 DR 非同期の昇格時に自身のエージェントがダウンしたミラー・メンバ上で、最初に ValidatedMember=0 を設定せずに InterSystems IRIS を再起動すると、両方のフェイルオーバー・メンバが同時にプライマリとして動作してしまうことがあります。

注釈 SYS.Mirror.Promote()、SYS.Mirror.PromoteWithPartner()、SYS.Mirror.PromoteWithNoPartner()、および SYS.Mirror.PromoteWithSelectedPartner() ミラーリング API メソッドを使用して、DR 非同期をフェイルオーバー・メンバに昇格することもできます。

5.6 DR 非同期へのバックアップの降格

DR 非同期をフェイルオーバー・メンバに昇格することに加えて、その反対も実行できます。現在のプライマリではないフェイルオーバー・メンバを DR 非同期に降格して、ミラーに 1 つのフェイルオーバー・ミラーを残すことができます。これは、フェイルオーバー・メンバがミラーの構成に対する一時的な変更に対処しないようにする必要がある計画的停止の状況で有用です。次に、例を示します。

- ・ メンテナンスのためにバックアップ・フェイルオーバー・メンバとそのホスト・システムをシャットダウンしているときに、(何らかの理由で) プライマリ上の InterSystems IRIS インスタンスを再起動すると、このインスタンスはバックアップ・インスタンスまたはその ISCAgent に通信できないため、最新のプライマリであるかどうかを判断する方法がないことにより、再起動後にプライマリになることができません。ただし、“バックアップ・フェイルオーバー・メンバのメンテナンス”で説明しているように、シャットダウンする前にバックアップを DR 非同期に降格しておく、このリスクが回避されます。プライマリは、現在バックアップはないことと、そのためにプライマリが再起動後にプライマリになれることを認識しているからです。再起動後に、降格していた DR 非同期をバックアップに昇格できます (“DR 非同期メンバのフェイルオーバー・メンバへの昇格”を参照)。
- ・ “昇格した DR 非同期への計画的フェイルオーバー”で説明しているように、DR 非同期に意図的にフェイルオーバーすることで災害復旧機能をテストしているときに、プライマリ・インスタンスをシャットダウンしてフェイルオーバーをトリガする場合、プライマリ・インスタンスを自動的にバックアップにすることなく (実際の災害時には利用できない可能性が高いため)、再起動して同期状態を維持できます。この場合、再起動前に (ISCAgent を使用して) DR 非同期に降格しておき、後で準備が整った時点でフェイルオーバー・メンバに昇格することができます。

フェイルオーバー・メンバを降格するには、“ミラー・モニタの使用法”で説明しているように、いずれかのフェイルオーバー・メンバで [ミラー・モニタ] ページ ([ホーム]→[システム処理]→[ミラー・モニタ]) に移動します。次に、以下の手順を実行します。

- ・ バックアップで、[DR メンバに降格] ボタンを使用して、バックアップを DR 非同期に降格します(この方法は、前述の例の最初で使用します)。
- ・ プライマリで、[他のメンバを降格] ボタンを使用して、バックアップを DR 非同期に降格します(この方法は、前述の例の 2 つ目で使用します)。現在のメンバがプライマリで、バックアップ・インスタンスまたはその ISCAgent にアクセス可能な場合にのみ、降格に成功します。

注釈 “フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避” で説明しているように、フェイルオーバーが設定されていない場合に、現在のプライマリを降格することはできません。

^MIRROR ルーチンの [ミラー管理] メニューの [非同期 DR メンバへのバックアップ・メンバの降格] オプション、および SYS.Mirror.Demote() と SYS.Mirror.DemotePartner() ミラーリング API メソッドも、バックアップを DR 非同期に降格する手段として使用できます。

5.7 ミラー・メンバの再構築

状況によっては、停止または障害の発生後に、特に手動手順を使用してミラーを稼働状態に戻した場合、メンバのミラーリングされるデータベースがミラーと同期しなくなっていることがあります。例えば、プライマリの停止後に自動的に引き継がなかったバックアップを、最新のジャーナル・データなしに強制的にプライマリにすると (“バックアップがアクティブでない場合の手動フェイルオーバー” を参照)、前のプライマリ上の 1 つ以上のミラーリングされるデータベースが、新しいプライマリのデータベースと一致しない場合があります。

ミラーがその不一致を調整できる場合もありますが、調整できない場合もあります。ミラー・メンバのデータの不一致が調整不可能なミラー・メンバを再起動して、ミラーに再参加させようとする、そのプロセスは停止し、次のような深刻度 2 のメッセージがメッセージ・ログに書き込まれます。

```
This member has detected that its data is inconsistent with the mirror MIRRORNAME. If the primary is running and has the correct mirrored data, this member, including its mirrored databases, must be rebuilt.
```

このメッセージの前に、不一致の詳細を示した深刻度 1 のメッセージが書き込まれます。

このメッセージがメッセージ・ログに表示された場合は、以下のステップを実行します。

1. 機能しているミラーが有しているデータのバージョンが目的のものであること、また不一致を報告しているメンバを再構築する必要があることを確認します。これは、例えば、すべての最新ジャーナル・データなしに手動で別のメンバをプライマリにすることを選択した後、前のプライマリを再起動しているときにこのメッセージが表示された場合に当てはまることが多いです。この場合、以下のステップを使用して、不一致メンバを再構築します。

代わりに、不一致を報告しているメンバには目的のバージョンのデータがあると結論付けた場合、この手順を採用して、他のメンバを再構築できます。

使用するデータのバージョンが不明であったり、不一致メンバの再構築が適切であるかどうかわからない場合、最善策を判断できるように、[インターシステムズのサポート窓口](#)までお問い合わせください。

2. 機能しているミラー内の 1 つのメンバの、ミラーリングされるデータベースをバックアップします。以下のことを確信している場合は、ミラー内の 1 つのメンバで作成した既存のバックアップを使用することもできます。
 - ・ バックアップは、データの不一致をまねいた停止または障害が発生する前に作成された。
 - ・ 現在のプライマリには、バックアップが作成された時点のすべてのジャーナル・ファイルが含まれている。
3. “ミラー構成の編集または削除” の説明に従って、不一致メンバをミラーから削除し、ミラーリングされるデータベースの、ミラーリングされる DB 属性を保持します。

4. “第2のフェイルオーバー・メンバを構成する”または“非同期ミラー・メンバを構成する”の説明に従い、適切な手順を使用してメンバをミラーに追加します。
5. “ミラーへ既存データベースを追加する”の説明に従って、作成または選択したバックアップから、ミラーリングされるデータベースをメンバにリストアします。

5.8 バックアップおよび非同期メンバのミラーリングの停止

バックアップまたは非同期メンバのミラーリングは、一時的に停止できます。例えば、メンテナンスや再構成を行うための短い時間や、プライマリのデータベース・メンテナンスを行っている間は、プライマリのバックアップ・メンバのミラーリングを停止することが必要になる場合があります。また、ネットワークの使用率を下げるために、レポート非同期メンバのミラーリングを一時的に停止することがあるかもしれません。そのためには、以下を実行します。

1. ミラーリングを停止するメンバの [システム処理]→[ミラー・モニタ] ページに移動します。
2. そのメンバが、バックアップ・フェイルオーバー・メンバの場合は、[このメンバのミラーリングを停止] ボタンをクリックします。
3. そのメンバが非同期の場合は、ミラーリングを停止する非同期メンバのミラーの行にある [このメンバのミラーリングを停止] リンクをクリックします。(1つのミラーのミラーリングを停止しても、レポート非同期が属しているその他のミラーには影響しません。)

この処理には、数秒かかります。[ミラー・モニタ]を更新すると、[このメンバのミラーリングを停止]は[このメンバのミラーリングを開始]に置き換えられています。これは、ミラーリングの再開に使用できます。

重要 あるメンバでミラーリングを停止すると、前述したように、ミラーリングは明示的に再開するまで停止したままになります。ミラーの再初期化やメンバの再開始によって、そのメンバのミラーリングが開始されることはありません。

注釈 また、ミラーリングの `SYS.Mirror.StopMirror()` および `SYS.Mirror.StartMirror()` API メソッド、あるいは `MIRROR` ルーチン (“[MIRROR ルーチンの使用法](#)”を参照)を使用して、これらのタスクを実行することもできます。

5.9 データベース・デジャーナリングの管理

“ミラー同期”で説明されているように、デジャーナリングとは、プライマリ・フェイルオーバー・メンバのジャーナル・データを他のミラーメンバのミラーリングされるデータベースに適用することにより、ミラーリングされるデータベースを同期させるプロセスのことです。デジャーナリングは、ルーチン・ミラー処理中の自動プロセスですが、ある状況下では、`MIRROR` ルーチン (“[MIRROR ルーチンの使用法](#)”を参照)で用意されているオプションを使用したデジャーナリングの管理が必要になる場合があります。バックアップ・フェイルオーバー・メンバ、DR 非同期メンバ、およびレポート非同期メンバのそれぞれの目的の違いにより、デジャーナリングとデジャーナリング管理でも違いがあります。特に、意図的なものであれ、エラーによるものであれ、デジャーナリングにおける中断の違いがあります。さらに、レポート非同期が属する1つ以上のミラーに向けたデジャーナリングに、ユーザ定義のフィルタを適用することもできます。

- ・ [バックアップまたは DR 非同期でのデジャーナリング管理](#)
- ・ [レポート非同期でのデジャーナリング管理](#)
- ・ [レポート非同期に対するデジャーナル・フィルタの使用](#)

注釈 1 つまたはすべてのミラーされたデータベースのデジャーナリングが一時停止している場合でも、すべてのタイプのミラー・メンバはジャーナル・データの受信を続けます。

デジャーナリングの管理には、SYS.Mirror.AsyncDejournalStatus()、SYS.Mirror.AsyncDejournalStart()、SYS.Mirror.AsyncDejournalStop()、および SYS.Mirror.DejournalPauseDatabase() ミラーリング API メソッドを使用することもできます。

5.9.1 バックアップまたは DR 非同期でのデジャーナリング管理

バックアップ・フェイルオーバー・メンバおよび DR 非同期メンバ上のミラーリングされるデータベースは常にできる限りキャッチアップされて、それぞれプライマリとして引き継ぐ可能性や災害復旧での使用の可能性に備える必要があるため、エラーによるデジャーナリングの一時停止は、影響を受けたミラーリングされるデータベースに対してのみ行われ、それ以外に対してはデジャーナリングは続行されます。

例えば、バックアップまたは DR 非同期メンバで <FILEFULL> などのデータベース書き込みエラーが発生した場合、書き込みエラーが発生したデータベースのデジャーナリングは自動的に一時停止されますが、他のミラーリングされるデータベースのデジャーナリングは続行されます。エラーのデータベースをディスマウントし、エラーを修正したら、データベースを再マウントします。さらに、**MIRROR ルーチンの [ミラー管理]** メニューにより **[ミラーリングされるデータベースの有効化/キャッチアップ]** オプションを選択してデジャーナリングを再開するか、管理ポータルの使用によりデータベースのキャッチアップを再開します (**“ミラーリングされるデータベースの有効化とキャッチアップ”** を参照)。

DR 非同期では、**MIRROR ルーチンの [ミラー管理]** メニューにある **[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを使用して、メンバ上のミラーリングされるデータベースすべてに対してデジャーナリングを一時停止するオプションもあります。(このオプションは、バックアップ・メンバでは無効になっています。)これは、デジャーナリング・エラーの後や、メンテナンスを目的として使用できます。例えば、デジャーナリング・エラーによって、1 つのデータベースに対してのみデジャーナリングが一時停止した場合に、ミラー内のすべてのデータベースに対してデジャーナリングを一時停止するときは、以下を実行できます。

1. **MIRROR ルーチンの [ミラー管理]** メニューから **[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを選択して、すべてのデータベースに対してデジャーナリングを一時停止します。
2. 問題の発生しているデータベースをディスマウントして、エラーを修正し、データベースを再マウントします。
3. **MIRROR ルーチンの [ミラー管理]** メニューから **[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを選択して、すべてのデータベースに対するデジャーナリングを再開します。(このオプションによって、エラーが生じたデータベースが自動的に有効化され、ミラー内の最新のデータベースと同じポイントまでキャッチアップされます。)

注釈 **[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを使用して、DR 非同期メンバ上のデジャーナリングを一時停止している場合、再度このオプションを使用して再開するまで、デジャーナリングは再開しません。

5.9.2 レポート非同期でのデジャーナリング管理

“非同期ミラー・メンバ” で説明するように、レポート非同期メンバは複数のミラーに属することができます。それら各ミラーに対して、データベースの使われ方により、データベース・デジャーナリングを継続的に実施することも、定期的にも実施することもできます。例えば、指定されたミラーに対して、深夜から午前 4 時にかけてのデジャーナリングを行い、残りの時間帯には安定的なレポート生成のためデータベースに静的な状態を維持させることができます。

また、メンテナンスまたはデジャーナリング中のエラー発生のため、データベースをディスマウントする場合に、各種ミラーに対して異なる動作をさせることもできます。例えば、あるミラーでは、デジャーナリング一時停止中のデータベースが、ミラー内の他のデータベースから後れを取らないようにすることが最も重要なため、ミラー全体でデジャーナリングを一時停止することが望ましくなります。また別のミラーでは、ミラー内のデータベースができるだけ最新を保つことが最も重要なため、関与するデータベースのみを一時停止させることが最も重要となります。

レポート非同期上の 1 つ以上のミラーに対するデジャーナリングを、1 回限りの処理として、または定期的に一時停止する場合、`MIRROR ルーチン`の **[ミラー管理]** メニューから **[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを選択して、目的のミラーですべてのデータベースに対してデジャーナリングを一時停止できます。デジャーナリングを再開するには、**[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを再度使用します。(このオプションは、バックアップ・メンバでは使用できません。)

バックアップや DR 非同期メンバとは異なり、レポート非同期メンバ上のデータベースのデジャーナリング中にエラーが生じた場合、そのミラー内のデータベースすべてに対して、デジャーナリングが自動的に一時停止します。ユーザの要望や方針に応じて、以下のいずれかができます。

- ・ エラーが発生したデータベースをディスマウントし、`MIRROR ルーチン`の **[ミラー管理]** メニューから **[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを選択して、ミラー内のその他すべてのデータベースに対してデジャーナリングを再開し、エラーを修正して、データベースをマウントします。その後、`MIRROR ルーチン`の **[ミラー管理]** メニューから **[ミラーリングされるデータベースの有効化/キャッチアップ]** オプションを選択して、そのデータベースに対するデジャーナリングを再開するか、または、管理ポータルの使用により、データベースのキャッチアップを再開します (**“ミラーリングされるデータベースの有効化とキャッチアップ”** を参照)。
- ・ エラーを修正し、データベースを再マウントする間は、ミラー全体に対してデジャーナリングを一時停止したままにし、その後、**[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを使用してミラー全体に対してデジャーナリングを再開します (このオプションによって、エラーが発生したデータベースが自動的に有効化され、ミラー内の最新のデータベースと同じポイントまでキャッチアップされます)。

レポート非同期メンバ上のミラーリングされるデータベースにメンテナンスを実行する場合、単にデータベースをディスマウントして、メンテナンスの後にデータベースをマウントし、**[ミラーリングされるデータベースの有効化/キャッチアップ]** オプションまたは管理ポータルを使用して、データベースをキャッチアップできます。(メンテナンスにこのようなデータベースが複数関係する場合は、**“ミラーリングされるデータベースの有効化とキャッチアップ”** の説明に従い、ミラー・モニタを使用して、一度にそれらのデータベースすべてに対して処理を実行します。これは、データベースを個々にキャッチアップするよりも、効率的で、かかる時間の短縮になります。)

注釈 エラーにより、レポート非同期メンバ上のミラーに対して、デジャーナリングが一時的に停止している場合、メンバはプライマリとの接続が次に再構築されたときに、そのミラーに対するデジャーナリングの再開を試みます。**[非同期メンバ上でのミラー・デジャーナリングの管理]** オプションを使用して、非同期メンバ上のミラーのデジャーナリングを一時停止している場合、再度このオプションを使用して再開するまで、そのミラーでのデジャーナリングは再開しません。

5.9.3 レポート非同期に対するデジャーナル・フィルタの使用

レポート非同期に限り、特定のミラーにユーザ定義のデジャーナル・フィルタを設定できます。これにより、ジャーナル・レコードごとにユーザ独自のコードを実行して、そのミラー内の読み書き可能なデータベースに適用するレコードを決定できます。フィルタをいったん定義しておけば、任意の数のミラーに設定することができます。また、フィルタは、いつでも設定、変更、および削除できます。

重要 この機能は、きわめて特殊なケースのみを想定しています。その他の使用方法には、十分な注意が必要です。ミラー・メンバに複製するグローバルを制御する場合は、ミラーリングされないデータベースへのグローバルのマッピングにより、さらに簡単に軽量のソリューションが得られます。アプリケーション・データベースへの更新を監視する場合、アプリケーション・レベルで構築されたソリューションは、一般に高い柔軟性があります。

デジャーナル・フィルタにより、レポート非同期は、プライマリから受信したジャーナル・ファイルに含まれる一部のレコードのデジャーナリングをスキップできるようになります。ただし、これは、読み取り書き込み可能なデータベース (最初に読み書き可能なレポート非同期のミラーに追加されたデータベース、または読み取り専用としてミラーに追加されたときから **[FailoverDB]** フラグがクリアされているデータベース) にのみ当てはまることです。(レポート非同期のミラーリングされるデータベースのマウント・ステータスおよび **[FailoverDB]** フラグの詳細は、**“レポート非同期ミラー・メンバの FailoverDB フラグのクリア”** を参照してください。) **FailoverDB** フラグがデータベースに設定されている場合は、データベースが読み

取り専用としてマウントされたことを意味します。それでもデジャーナル・フィルタのコードは実行されますが、フィルタコードの返す内容にかかわらず、そのデータベースには、常にすべてのレコードがデジャーナルされます。

重要 デジャーナル・フィルタを設定すると、フィルタが設定されたミラーのデジャーナリングが遅くなります。この影響は、フィルタの内容によっては重大なものになることがあります。

デジャーナル・フィルタを作成するには、スーパークラス `SYS.MirrorDejournal` を拡張して、ミラー・デジャーナル・フィルタ・クラスを作成します。クラス名は、Z または z で始める必要があります。これにより、InterSystems IRIS のアップグレード時にそのクラスが保持されます。

レポート非同期のミラーにデジャーナル・フィルタを設定するには、[非同期構成の編集] ページ ([システム管理]→[構成]→[ミラー設定]→[非同期の編集]) に移動して、[この非同期メンバの所属先のミラー] リストに示された目的のミラーの横にある [デジャーナル・フィルタの編集] リンクをクリックし、ミラー・デジャーナル・フィルタ・クラスの名前を入力して、[保存] をクリックします。フィルタを削除する場合は、同じ手順を実行しますが、入力ボックスをクリアしてから、[保存] をクリックします。ミラーのジャーナル・フィルタを追加、変更、または削除すると、そのミラーに対するデジャーナリングが自動的に再開され、フィルタが適用できるようになります。ただし、ミラー・デジャーナル・フィルタ・クラスに変更を加えてリコンパイルする場合は、`^MIRROR` ルーチンの [ミラー管理] メニューにある [非同期メンバ上でのミラー・デジャーナリングの管理] オプションを使用して、フィルタを設定したすべてのミラーのデジャーナリングを手動で停止および再開する必要があります。

5.10 一般的なミラーリングに関する考慮事項

このセクションでは、ミラーリングに関する考慮事項、推奨事項、および最善の方法のガイドラインを示します。このセクションには、以下のサブセクションがあります。

- ・ [ミラーの API](#)
- ・ [プライマリ・フェイルオーバー・メンバの外部バックアップ](#)
- ・ [ミラー・メンバでの InterSystems IRIS のアップグレード](#)

5.10.1 ミラーの API

`SYS.Mirror` クラスには、管理ポータルおよび `^MIRROR` ルーチン ("[^MIRROR ルーチンの使用法](#)" を参照) で使用可能なミラー処理をプログラムで呼び出すためのメソッドと、多数のクエリが用意されています。例えば、`SYS.Mirror.CreateNewMirrorSet()` メソッドを使用することで、[ミラーを作成し、第 1 のフェイルオーバー・メンバを構成](#)することができます。その一方で、`SYS.Mirror.MemberStatusList()` クエリは、ミラー・メンバのリストと、それぞれのジャーナル遅延ステータスを返します。これらのメソッドの詳細は、`SYS.Mirror` のクラス・ドキュメントを参照してください。

バックアップを実行するために外部スクリプトを使用する場合は、`$SYSTEM.Mirror` クラスを使用すると、システムがミラーの一部であるかどうかを検証し、一部である場合はそのロールが何かを検証できます。

```
$System.Mirror.IsMember()
$System.Mirror.IsPrimary()
$System.Mirror.IsBackup()
$System.Mirror.IsAsyncMember()
$System.Mirror.MirrorName()
```

`$SYSTEM.Mirror.IsMember()` は、このシステムがフェイルオーバー・メンバである場合は 1 を返し、非同期ミラー・メンバである場合は 2 を返し、ミラー・メンバではない場合は 0 を返します。`$SYSTEM.Mirror.IsPrimary()` は、このシステムがプライマリ・フェイルオーバー・メンバである場合は 1 を返し、そうでない場合は 0 を返します。`$SYSTEM.Mirror.IsBackup()` は、このシステムがバックアップ・フェイルオーバー・メンバである場合は 1 を返し、そうでない場合は 0 を返します。`$SYSTEM.Mirror.IsAsyncMember()` は、このシステムが非同期メンバである場合は 1 を返し、そうでない場合は 0 を返

します。`$SYSTEM.Mirror.MirrorName()` は、このインスタンスがフェイルオーバー・ミラー・メンバとして構成されている場合はミラー名を返し、そうでない場合は `NULL` を返します。

また、`$SYSTEM.Mirror.GetMemberType()` および `$SYSTEM.Mirror.GetMemberStatus()` を使用して、InterSystems IRIS の現行インスタンスのミラー・メンバシップ (存在する場合) およびそのロールにおけるステータスについての情報を取得できます。詳細は、“[ミラー・メンバのジャーナル転送およびデジャーナリングのステータス](#)” を参照してください。

5.10.2 プライマリ・フェイルオーバー・メンバの外部バックアップ

`Backup.General.ExternalFreeze()` メソッドの使用により、プライマリ・フェイルオーバー・メンバのデータベースへの書き込みをフリーズして、“[バックアップとリストア](#)” の説明どおりに、外部バックアップを実行できるようにするには、`Backup.General.ExternalFreeze()` の `ExternalFreezeTimeOut` パラメータの指定より長く、外部フリーズが更新を中断しないようにします。指定時間より長い中断が起こると、ミラーがバックアップ・フェイルオーバー・メンバをフェイルオーバーし、その結果、進行中のバックアップ操作が終了されてしまいます。

5.10.3 ミラー・メンバでの InterSystems IRIS のアップグレード

ミラー・メンバで InterSystems IRIS をアップグレードするときのオプションおよび考慮事項を確認するには、“[ミラーリングを使用した最小ダウンタイムのアップグレード](#)” を参照してください。

重要 リリース 2023.3 では、以前のリリースと比べてより新しい、よりコンパクトなバージョンのジャーナル・ファイル形式が使用されており、このことは、ミラーリングされたシステム上でアップグレードを実行する際に必要な順序に影響を及ぼします。特に、2023.3 以前のリリースからアップグレードする場合は、プライマリをアップグレードする前にバックアップ・メンバをアップグレードするというベスト・プラクティスに従う必要があります。

5.11 ミラーリングにおけるデータベースの考慮事項

このセクションでは、ミラーリングされるデータベースを構成および管理する際に考慮する点について説明します。

- ・ [InterSystems IRIS インスタンスの互換性](#)
- ・ [メンバのエンディアンに関する考慮事項](#)
- ・ [`DATABASE ルーチンを使用したミラーリングされるデータベースの作成](#)
- ・ [`DATABASE ルーチンを使用した既存のミラーリングされるデータベースの再作成](#)
- ・ [ミラーリングされるデータベースのマウント/ディスマウント](#)
- ・ [ミラーリングされないシステムへのミラーリングされるデータベースのコピー](#)

5.11.1 InterSystems IRIS インスタンスの互換性

ミラーの各 InterSystems IRIS インスタンスは以下のようにいくつかの方法で互換性がある必要があります。

1. ミラー内のすべての InterSystems IRIS インスタンスで以下のことが必要です。
 - ・ 同一の文字幅を使用します (8 ビットまたは Unicode、“[文字幅設定](#)” を参照)。
 - ・ 同じロケールを使用します (“[管理ポータル](#)の [NLS 設定ページの使用法](#)” を参照)。

注釈 これらの要件に対する唯一の例外が、ISO 8859 Latin-1 文字セットに基づいたロケールを使用する 8 ビット・インスタンスは、対応する幅の文字ロケールを使用する Unicode インスタンスと互換性があることです。例えば、**enu8** ロケールを使用する 8 ビット・プライマリ・インスタンスは、**enuw** ロケールを使用する Unicode バックアップ・インスタンスと互換性があります。しかし、**heb8** ロケールを使用する 8 ビット・プライマリ・インスタンスは、**hebw** ロケールを使用する Unicode バックアップ・インスタンスと互換性がありません。これらのロケールは ISO 8859 Latin-1 に基づいていないためです。

2. 各フェイルオーバー・メンバで、同じデータベースのブロック・サイズが有効にされている必要があります（“[ラージ・ブロック・サイズに関する考慮事項](#)” 参照してください）。また、フェイルオーバー・メンバで有効になっているサイズを非同期メンバで有効にする必要があります。プライマリに追加されるミラーリングされるデータベースのブロック・サイズが別のメンバで有効になっていないと、そのデータベースはそのメンバのミラーに追加できません。
3. フェイルオーバー・メンバおよびすべての DR 非同期メンバは、同じ InterSystems IRIS バージョンでなければなりません。ただし、“インストール・ガイド” の “InterSystems IRIS のアップグレード” の章の “[ミラーリングを使用した最小ダウンタイムのアップグレード](#)” に記載されているアップグレード手順のいずれかを行っている間のみ、異なるバージョンでもかまいません。アップグレード対象のメンバがプライマリになったら、このアップグレードが完了するまで、他方のフェイルオーバー・メンバおよびすべての DR 非同期メンバを使用できません（特に、これらをプライマリにすることはできません）。

ミラーリングでは、レポート非同期メンバがフェイルオーバー・メンバと同じ InterSystems IRIS バージョンである必要はありませんが、アプリケーションの機能でこれが必要な場合があります。

5.11.2 メンバのエンディアンに関する考慮事項

- ・ 分散キャッシュ・クラスタ内のミラーリングされたデータ・サーバのメンバ（例えば、ECP データ・サーバとして機能しているミラー・セット）では、混合エンディアンはサポートされません。フェイルオーバー・メンバが一時的に混合している場合（例えば、エンディアン間でサーバを移行している場合）、ECP セッションはフェイルオーバー後に回復しません。
- ・ ミラーリングされるデータベースの作成時または既存のデータベースのミラーへの追加時に、バックアップ・フェイルオーバー・メンバまたは非同期メンバのエンディアンがプライマリ・フェイルオーバー・メンバのエンディアンと異なる場合は、“[ミラーへ既存データベースを追加する](#)” で説明しているバックアップおよびリストアの手順は使用できません。その代わりに、データベースの **IRIS.DAT** ファイルをコピーする操作を含めて、そのセクションの手順を使用する必要があります。それに加えて、その手順を使用するときには、すべての非プライマリ・メンバに **IRIS.DAT** ファイルをコピーした後で、それらのメンバにデータベースをマウントする前に以下の手順を挿入します。
 - バックアップ・フェイルオーバー・メンバおよび各非同期メンバ上で、**IRIS.DAT** ファイルのコピーを変換します。“[cvendian を使用したバイト・オーダー変換](#)” を参照してください。

5.11.3 ^DATABASE ルーチンを使用したミラーリングされるデータベースの作成

ミラーリングされるデータベース（ミラーリング・データベース）をミラー・メンバに作成するには、**^DATABASE** ルーチンを使用します（“[^DATABASE](#)” を参照）。他のミラー・メンバに新規ミラーリング・データベースを作成する前に、プライマリ・メンバにそれを作成する必要があります。ミラーリングされるデータベースを作成するには：

1. **^DATABASE** ルーチンを実行し、**[1) データベースの作成]** オプションを選択します。
2. **[データベースディレクトリ ?]** プロンプトでディレクトリ・パスを入力します。
3. **[既定のデータベースプロパティを変更?]** プロンプトで **[はい]** を入力します。
4. **[フィールド番号を変えますか?]** プロンプトで **3** (**[ミラーDB名:]**) を入力し、**[ミラーDB名は?]** プロンプトでミラーリングされるデータベースのミラー名を入力します。

注釈 作成しているミラーリング・データベースのメンバが複数ミラーのメンバで、かつ既定のリストのミラーと異なるミラーでミラーリング・データベースを作成している場合、**[フィールド番号を変えますか?]** プロンプトで (**[ミラーセット名:]**) を入力し、リストから適切なミラー名を選択します。ルーチンを実行しているメンバが唯一のミラーのメンバである場合、このフィールドは変更できません。

5. ユーザのデータベースの必要に応じて、他のフィールドを変更し、変更が終了したら、**[フィールド番号を変えますか?]** プロンプトで、オプションを指定せずに **Enter** を押します。
6. **[構成中データベースのデータセット名:]** プロンプトで、データベースのデータセット名を入力します。これは管理ポータルで表示される名前です。
7. ミラーリングされるデータベースが作成されるまで、残りのプロンプトに返答を入力してください。

バックアップおよび非同期メンバにミラーリングされるデータベースを作成する場合、プライマリ・メンバに作成したデータベースに対するキャッチアップが自動的に行われます。

注釈 ^DATABASE ルーチンの使用によって、既存のミラーリングされないデータベースをミラーに追加することはできません。必要なプロシージャの詳細は、[“ミラーへのデータベースの追加”](#) を参照してください。

5.11.4 ^DATABASE ルーチンを使用した既存のミラーリングされるデータベースの再作成

^DATABASE ルーチンの **[10) データベースの再作成]** オプションを使用すると、データベースの名前やサイズを変更することなく、既存のデータベースのデータをクリアできます(ルーチンについては、[“^DATABASE”](#) を参照)。このオプションは、ミラーリングされるデータベースに使用できますが、そのデータベースを認識するすべてのミラー・メンバで使用する必要があり、新しいミラーリングされるデータベースの作成に **[データベースの作成]** オプションを使用した順序 (最初にプライマリ、次にバックアップ、その次にミラーに含まれるデータベースの非同期) で使用する必要があります。

注意 **[10) データベースの再作成]** オプションを使用してプライマリにデータベースを再作成する場合、バックアップとミラーの DR 非同期にも、この操作を繰り返す必要があります。このようにしないと、フェイルオーバー時または災害復旧時に、そのデータベースは破棄されることがあります。再作成操作は、レポート非同期にも繰り返すことを強くお勧めします。

5.11.5 ミラーリングされるデータベースのマウント/ディスマウント

ミラーリングされるデータベースは、どちらのフェイルオーバー・メンバ上でもマウントおよびディスマウントできます。ただし、バックアップ・フェイルオーバー・メンバ上でディスマウントされた場合、そのデータベースは、再マウントされた後でミラーリングによって自動的にデータベースのキャッチアップが試みられるまで “古い” 状態のままになります。必要なジャーナル・ファイルがプライマリ・フェイルオーバー・メンバ上で使用可能な場合は、自動アップデートが正常に実行されます。ただし、プライマリ・メンバ上の必要なジャーナル・ファイルのいずれかが削除されている場合は、プライマリ・メンバ上の最新のバックアップからデータベースをリストアする必要があります。

5.11.6 ミラーリングされないシステムへのミラーリングされるデータベースのコピー

以下のようにして、ミラーリングされるデータベースを、ミラーリングされないシステムにコピーし、そのシステム上での読み書き可能としてマウントできます。

1. プライマリまたはバックアップ・フェイルオーバー・メンバ上のミラーリングされたデータベースをバックアップし、[“ミラーへ既存データベースを追加する”](#) で説明されている手順で、ミラーリングされていないシステムにバックアップ内容をリストアします (外部バックアップ・リストアまたはコールド・バックアップ・リストアに続くデータベースの手動での有効化とキャッチアップの手順は省略します)。リストアしても、データベースは引き続きミラーリングされているとマークされており、読み取り専用です。

2. ミラーリングされないシステム上では、`MIRROR ルーチン` (“[MIRROR ルーチンの使用法](#)” を参照) を使用して、[1 つ以上のミラーリングされるデータベースの削除]を選択し、以下の手順を実行することで、ミラーからデータベースを削除します。この手順に従うと、データベースは読み取り/書き込みモードでマウントされます。

5.12 ミラーリングにおけるプロダクションの考慮事項

このセクションでは、InterSystems IRIS プロダクションに適用するその他の考慮事項について説明します。内容は以下のとおりです。

- ・ [InterSystems IRIS によるミラーリングされるデータを使用した相互運用対応ネームスペースの処理方法](#)
- ・ [InterSystems IRIS プロダクションの推奨ミラーリング構成](#)
- ・ [ミラーリング環境におけるプロダクションの自動開始の機能](#)

5.12.1 InterSystems IRIS によるミラーリングされるデータを使用した相互運用対応ネームスペースの処理方法

InterSystems IRIS では、相互運用対応ネームスペースのマッピングを検査し、ミラーリングされるデータベースからのマッピングがネームスペースに含まれているか判断し、以下のような結果を導きます。

- ・ 相互運用対応ネームスペースを含むミラー・メンバを起動またはアップグレードすると、プロダクションはプライマリのみで起動されます。
- ・ InterSystems IRIS のアップグレード時に、特定のタスクにはデータベースへの書き込みアクセス権が必要です。これらのタスクはプライマリ・ミラー・メンバでのみ実行されます。
- ・ フェイルオーバーが発生して、メンバがプライマリ・ミラー・メンバになると、アップグレード時にはスキップされていた（その時点ではプライマリではなかったため）タスクがすべて、プロダクションの起動前に実行されます。

5.12.2 InterSystems IRIS プロダクションの推奨ミラーリング構成

ミラーリングには高可用性ソリューションとなることが意図されており、よって、ミラー・インスタンスのいずれにおいても無関係なアクティビティを最小限とすることが必要です。すなわち、ミラーリングされるあらゆるインスタンスのデータベースすべてをミラーリングする必要があります。

ユーザには、いずれのノードにおいてもデータのミラーリングなしで実行される、“より危険の少ない”プロダクションが選ばれてしまうことがあります。しかし、そのような構成では処理が複雑になり、維持が困難になる可能性があります。よって、そのような構成は避けて、代わりにすべてのデータベースをミラーリングすることを強く推奨します。

5.12.3 ミラーリング環境におけるプロダクションの自動開始の機能

ミラー・システムの起動時に（この時点では、まだいずれのメンバもプライマリ・フェイルオーバー・メンバになっていません）、以下のようになります。

1. InterSystems IRIS では、ミラーリングされるデータにアクセスするプロダクションは、`Ens.AutoStart` で指定されている場合でも起動されることはありません。メンバがプライマリ・インスタンスになると、その時点でこれらのプロダクションは起動されます。
2. InterSystems IRIS により、ミラーリングされるデータにアクセスしないネームスペースがインスタンス上にあるかどうかは判別されます。前述のとおり、ミラー・メンバにはミラーリングされるプロダクションのみをインストールすることをお勧めしています。それでも、ミラーリングされないデータベースと共にプロダクションがインストールされている場合、

InterSystems IRIS によりそのプロダクションは `Ens.AutoStart` で指定して起動されます(このロジックによって、ミラーリングされないネームスペースがミラー・メンバ上にインストールされている場合、InterSystems IRIS 起動時に起動されます)。

後で、メンバがプライマリ・フェイルオーバー・メンバになったとき、InterSystems IRIS は、ミラーリングされるデータを参照するネームスペースを検索し、それらのネームスペースでプロダクションを起動できるようにします。インターシステムズの推奨設定に従っていれば、インスタンスがプライマリ・ミラー・メンバになる前に、ミラーリングされるデータにアクセスするプロダクションは実行しません。InterSystems IRIS ではまず、起動前にプロダクションが実行中でないかどうかを確認されます。具体的には以下の処理が実行されます。

1. InterSystems IRIS により、ネームスペースで `Ensemble` ユーザとして実行されているジョブを数えることによって、プロダクションが既に実行中でないかどうかは判別されます。そのようなジョブが 3 個以上ある場合、これはプロダクションが既に実行中であることを示しており、InterSystems IRIS によりメッセージ・ログに警告が記録され、そのプロダクションの起動は試行されません。
2. 予想どおりにプロダクションが実行中でない場合、InterSystems IRIS により、プロダクションは `Ens.AutoStart` で指定して自動的に起動されます。

プロダクションの起動と停止の詳細は、“プロダクションの開始と停止”を参照してください。

5.13 インターシステムズの医療製品のミラーリングに関する考慮事項

InterSystems IRIS for Health™ および HealthShare® Health Connect でミラーリングを設定および管理する際は、いくつかの特別な考慮事項があります。その他の HealthShare 製品には、独自のミラーリングに関するドキュメントがあることに注意してください。以下の考慮事項は、これらの製品には必ずしも適用されません。

IRIS for Health または Health Connect のシステムをミラーリングする場合は、バックアップ・メンバの HSSYS ネームスペースでミラー・タスクを必ず手動でスケジュールする必要があります。このタスクをスケジュールすることで、Mirror Monitor Agent が確実に実行されます。このエージェントは、バックアップ・ミラー・インスタンス上で実行され、IRISSYS に保存されたミラーリングされていない項目をプライマリとバックアップとの間で同期します。このミラー・タスクをスケジュールするには、バックアップ・メンバのターミナルを開き、以下のように入力します。

```
do ##class(HS.Util.Mirror.Task).Schedule("HSSYS")
```

IRIS for Health では、HS_Services のユーザ名も各ミラー・メンバに対して有効にする必要があります。

6

ミラー停止の手順

計画的メンテナンスまたは不測の問題により、ミラー内の一方または両方のフェイルオーバー・メンバ上で InterSystems IRIS インスタンスが使用できなくなる場合があります。フェイルオーバー・メンバの InterSystems IRIS インスタンスが使用できないときに、その ISCAgent は引き続き使用できることも (ホスト・システムが引き続き動作している場合)、使用できなくなることも (ホスト・システムがダウンしている場合) あります。このセクションでは、インスタンスの停止や一方または両方のフェイルオーバー・メンバの全体的な停止を含め、さまざまな計画的および計画外停止シナリオの対処手順について説明します。

“[自動フェイルオーバーのメカニズム](#)” に記載されているように、プライマリ・フェイルオーバー・メンバからバックアップ・フェイルオーバー・メンバへの安全で正常なフェイルオーバーには、以下の 2 つの要件があります。

- ・ プライマリのインスタンスが実際にダウンしており、一時的なネットワークの問題によって遮断されているのではないことの確認。
- ・ プライマリの障害発生時に、バックアップがアクティブであったか (“[ミラー同期](#)” を参照)、または手動でキャッチアップされていた (“[自動フェイルオーバーなしのプライマリ計画外停止](#)” を参照) ため、バックアップにはプライマリからの直近のジャーナル・データがあることの確認。

このドキュメントを使用するにあたって、“[自動フェイルオーバーのルール](#)” を参照しながら、自動フェイルオーバーを制御するルールを確認することもできます。

バックアップ・フェイルオーバー・メンバがアクティブであるかどうかや DR 非同期がキャッチアップされているかどうかを確認するためのミラー・モニタ使用についての詳細は、“[ミラー・メンバのジャーナル転送およびデジャーナリングのステータス](#)” および “[ミラーの監視](#)” を参照してください。

ここでは、以下の手順について説明します。

- ・ 計画的停止の手順
 - [バックアップ・フェイルオーバー・メンバのメンテナンス](#)
 - [プライマリ・フェイルオーバー・メンバのメンテナンス](#)
 - [フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避](#)
 - [ミラー内 InterSystems IRIS インスタンスのアップグレード](#)
- ・ 計画外停止の手順
 - [バックアップ・フェイルオーバー・メンバの計画外停止](#)
 - [自動フェイルオーバーによるプライマリ・フェイルオーバー・メンバの計画外停止](#)
 - [自動フェイルオーバーが発生しない場合のプライマリ・フェイルオーバー・メンバの計画外停止](#)
 - [プライマリ・フェイルオーバー・メンバの計画外分離](#)

- 両方のフェイルオーバー・メンバの計画外停止
- ・ 災害復旧の手順
 - 災害復旧時の昇格した DR 非同期への手動フェイルオーバー
 - 昇格した DR 非同期への計画的フェイルオーバー
 - 昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え

6.1 計画的停止の手順

計画的メンテナンスを実施するには、フェイルオーバー・メンバのいずれかで InterSystems IRIS インスタンスをシャットダウンするか、もしくはそのインスタンスをホストしているシステム全体をシャットダウンする必要がある場合があります。これを実行する状況としては、以下のようなものがあります。

- ・ [バックアップ・フェイルオーバー・メンバのメンテナンス](#)
- ・ [プライマリ・フェイルオーバー・メンバのメンテナンス](#)
- ・ [フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避](#)

このセクションでは、適切なシャットダウンとは、iris stop コマンドの使用を表します。iris コマンドの詳細は、“[InterSystems IRIS インスタンスの制御](#)”を参照してください。

注釈 iris stop コマンドのほかに、**SYS.Mirror** API および `^MIRROR` ルーチンも手動でのフェイルオーバーのトリガに使用できます。

自動フェイルオーバーをトリガせずにプライマリをシャットダウンする操作に関する詳細は、“[フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避](#)”を参照してください。

計画的または計画外のフェイルオーバー・メンバの停止により、どのバックアップ・フェイルオーバー・メンバも使用できない場合、必要に応じて、DR 非同期メンバをフェイルオーバー・メンバに昇格させて、プライマリ障害時に発生するデータベース・アクセスの中断やデータ損失の可能性から保護できます。DR 非同期メンバのバックアップ・フェイルオーバー・メンバへの一時的昇格に関する詳細は、“[昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え](#)”を参照してください。

6.1.1 バックアップ・フェイルオーバー・メンバのメンテナンス

バックアップ・フェイルオーバー・メンバの InterSystems IRIS インスタンスを停止する必要がある場合、バックアップのインスタンスで適切なシャットダウンを実行できます。これによって、プライマリの機能が影響を受けることはありません。バックアップ・インスタンスが再起動すると、このインスタンスは自動的にバックアップとして再びミラーに加わります。

ただし、バックアップのホストがシャットダウンされていて、そのためにバックアップの ISCAgent に通信できないときに、プライマリの InterSystems IRIS インスタンスが（何らかの理由で）再起動された場合、そのプライマリは、再起動後にプライマリになることはできません。自身が最新プライマリだったかどうかを判断できないからです。バックアップのホスト・システムをシャットダウンする必要がある場合は、以下の手順を使用して、このリスクをなくすことができます。

1. “[DR 非同期へのバックアップの降格](#)”で説明しているように、バックアップ上で、バックアップを DR 非同期に降格します。
2. 前のバックアップ・インスタンスとそのホスト・システムをシャットダウンし、メンテナンス作業を完了して、そのメンバを DR 非同期として再起動します。

3. “[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”の説明に従って、以前のバックアップを DR 非同期からフェイルオーバー・メンバに昇格し、元のロールにリストアします。

バックアップが降格された後でプライマリを再起動すると、それは自動的にプライマリになります（プライマリの状態を維持します）。

バックアップをシャットダウン前に降格せず、バックアップのエージェントが利用できない間にプライマリの InterSystems IRIS インスタンスを再起動する必要がある場合、“[両方のフェイルオーバー・メンバの計画外停止](#)”の手順に従います。

6.1.2 プライマリ・フェイルオーバー・メンバのメンテナンス

プライマリ・フェイルオーバー・メンバの InterSystems IRIS インスタンス、またはホスト・システムを停止する必要がある場合、最初にバックアップへの適切なフェイルオーバーを行うことができます。バックアップがアクティブなときに（“[ミラー同期](#)”を参照してください）、プライマリの InterSystems IRIS インスタンスで適切なシャットダウンを実行します。自動フェイルオーバーがトリガされ、バックアップはプライマリとして引き継ぎをします。

メンテナンスが完了したら、旧プライマリの InterSystems IRIS インスタンス、またはホスト・システムを再起動します。InterSystems IRIS インスタンスが再起動すると、このインスタンスは自動的にバックアップとしてミラーに加わります。旧プライマリを元のロールに戻す場合、この手順（バックアップの InterSystems IRIS インスタンスでの適切なシャットダウン実行によりフェイルオーバーをトリガした後、このインスタンスを再起動する）を繰り返すことができます。

6.1.3 フェイルオーバー・メンバのメンテナンス時の不要なフェイルオーバーの回避

バックアップ・メンバによるプライマリとしての引き継ぎが生じることなしに、プライマリ・フェイルオーバー・メンバを適切にシャットダウンする必要がある場合があります。例えば、プライマリがきわめて短時間停止する場合や、プライマリに障害が発生したときにバックアップが引き継がないようにする場合などです。これは、以下の 3 つの方法のいずれかで行うことができます。

- ・ “[バックアップ・フェイルオーバー・メンバのメンテナンス](#)”の説明に従って、バックアップ・フェイルオーバー・メンバを降格します。
- ・ `iris stop /nofailover` コマンドを使用して、プライマリの InterSystems IRIS インスタンスを適切にシャットダウンします。/nofailover 引数は、フェイルオーバーのトリガを避けるための予防措置として使用します。
- ・ プライマリまたはバックアップの [\[ミラー・モニタ\]](#) ページの上部で、[\[フェイルオーバーなしの設定\]](#) をクリックして、フェイルオーバーなしを設定します。フェイルオーバーなしを設定すると、ボタンの表示が [\[フェイルオーバーなしのクリア\]](#) になり、`MIRROR` ルーチンの [\[ミラー・ステータス\]](#) メニューの [\[ステータス・モニタ\]](#) オプションに、この設定状態が示されます（[\[ステータス・モニタ\]](#) オプションに関する詳細は、“[ミラーの監視](#)”を参照してください）。

どちらかのフェイルオーバー・メンバで [\[フェイルオーバーなしのクリア\]](#) をクリックして、フェイルオーバーなしの状態をクリアし、フェイルオーバーを有効にします。フェイルオーバーなしの状態は、プライマリを再起動すると自動的にクリアされます。

6.1.4 ミラー内 InterSystems IRIS インスタンスのアップグレード

ミラー全体で InterSystems IRIS をアップグレードするには、“[インストール・ガイド](#)”の“[InterSystems IRIS のアップグレード](#)”の章にある“[ミラーリングを使用した最小ダウンタイムのアップグレード](#)”を参照してください。

ミラー全体で InterSystems IRIS をアップグレードするには、“[ミラーリングを使用した最小ダウンタイムのアップグレード](#)”で説明している手順を参照してください。

ミラー全体で InterSystems IRIS をアップグレードするには、“[ミラーリングを使用した最小ダウンタイムのアップグレード](#)”で説明している手順を参照してください。

6.2 計画外停止の手順

フェイルオーバー・メンバに予期せず障害が発生した場合、適切な手順は、障害が発生した InterSystems IRIS インスタンス、ミラーが入っていたフェイルオーバー・モード（“[自動フェイルオーバーのメカニズムの詳細](#)”を参照してください）、他方のフェイルオーバー・メンバのインスタンスのステータス、両方のフェイルオーバー・メンバの ISCAgent の可用性、およびミラーの設定によって異なります。

- ・ [バックアップ・フェイルオーバー・メンバの計画外停止](#)
- ・ [自動フェイルオーバーによるプライマリ・フェイルオーバー・メンバの計画外停止](#)
- ・ [自動フェイルオーバーが発生しない場合のプライマリ・フェイルオーバー・メンバの計画外停止](#)
- ・ [プライマリ・フェイルオーバー・メンバの計画外分離](#)
- ・ [両方のフェイルオーバー・メンバの計画外停止](#)

このセクションを使用するにあたって、プライマリが使用不可になった場合のバックアップの動作の詳細を説明している“[さまざまな停止シナリオに対するミラーの対応](#)”を参照することができます。

6.2.1 バックアップ・フェイルオーバー・メンバの計画外停止

バックアップ・フェイルオーバー・メンバの InterSystems IRIS インスタンスまたはそのホスト・システムに障害が発生したとき、プライマリは通常の動作を続けますが、アプリケーションによっては、短い間一時停止するものがあります（詳細は、“[バックアップ停止の影響](#)”を参照してください）。

バックアップの計画外停止が発生した場合は、障害の原因となった状態を修復した後に、バックアップの InterSystems IRIS インスタンスまたはホスト・システムを再起動します。バックアップの InterSystems IRIS インスタンスが再起動すると、このインスタンスは自動的にバックアップとしてミラーに加わります。

注釈 バックアップがエージェント制御モードに入り（“[自動フェイルオーバーのルール](#)”を参照してください）、バックアップの ISCAgent に通信できない場合、プライマリの InterSystems IRIS インスタンスは再起動後にプライマリになることはできません。自身が最新プライマリだったかどうかを判断できないからです。したがって、バックアップ・ホスト・システムがダウンしているときに、何らかの理由でプライマリの InterSystems IRIS インスタンスを再起動する必要がある場合は、“[バックアップ・フェイルオーバー・メンバのメンテナンス](#)”で説明されている手順を実行してそれを行う必要があります。

6.2.2 自動フェイルオーバーによるプライマリ・フェイルオーバー・メンバの計画外停止

“[自動フェイルオーバーのルール](#)”で説明しているように、プライマリの InterSystems IRIS インスタンスが使用不可になると、バックアップは以下の場合にプライマリとして自動的に引き継ぐことができます。

- ・ バックアップがアクティブで、さらに以下の場合。
 - － 引き継ぎを要求するプライマリから通信を受信した。
 - － プライマリとの通信を失ったという情報をアービターから受信した。
 - － アービターが使用不可になっているか、アービターが構成されていない場合、プライマリの ISCAgent に通信して、プライマリのインスタンスがダウンまたはハングしていることを確認した。
- ・ バックアップがアクティブではないが、プライマリの ISCAgent に通信して、プライマリのインスタンスがダウンまたはハングしていることを確認し、プライマリの最新ジャーナル・データを ISCAgent から取得することができる場合。

自動フェイルオーバーを実行することができる状況の詳細は、“[プライマリ停止シナリオに対応した自動フェイルオーバー](#)”を参照してください。

計画外のプライマリ停止後にバックアップが自動的に引き継いだ場合は、停止の原因となった状態を修復してから、旧プライマリの InterSystems IRIS インスタンスまたはホスト・システムを再起動します。InterSystems IRIS インスタンスが再起動すると、このインスタンスは自動的にバックアップとしてミラーに加わります。旧プライマリをその元のロールに戻すには、“[プライマリ・フェイルオーバー・メンバのメンテナンス](#)”の説明に従って、バックアップの InterSystems IRIS インスタンスで適切なシャットダウンを実行してフェイルオーバーをトリガし、それから再起動します。

6.2.3 自動フェイルオーバーが発生しない場合のプライマリ・フェイルオーバー・メンバの計画外停止

“[自動フェイルオーバーのルール](#)”で説明しているように、プライマリのホスト・システムが、その ISCAgent も含めて使用不可になり、以下のいずれかが当てはまる場合、バックアップの InterSystems IRIS インスタンスは、応答しないプライマリのインスタンスから自動的に引き継ぐことはできません。

- ・ バックアップはアクティブではなかった。
- ・ バックアップは、エラーにより引き継ぐことができない。
- ・ アービターが構成されていないことにより、またはプライマリの InterSystems IRIS インスタンスおよびその ISCAgent との通信を失う前かそれと同時にアービターとの通信も失ったことにより、バックアップはプライマリがダウンしていることを確認できない。

このシナリオでは、3 つの状況が考えられます。それぞれの状況を可能なソリューションと共に以下に示します。

1. プライマリのホスト・システムに障害が発生しましたが、再起動できます。以下のいずれかを実行できます。
 - ・ プライマリの InterSystems IRIS インスタンスを再起動せずに、プライマリのホスト・システムを再起動します。プライマリの ISCAgent が使用できるようになると、バックアップは必要に応じてそのエージェントから最新のジャーナル・データを取得し、プライマリになります。
 - ・ プライマリの InterSystems IRIS インスタンスを含めて、プライマリのホスト・システムを再起動します。フェイルオーバー・メンバは、一方がプライマリになり、他方がバックアップになるまで、ネゴシエートします。
2. プライマリのホスト・システムに障害が発生し、再起動できません。[バックアップによる引き継ぎを手動で強制](#)することができます。このための手順は、バックアップがプライマリとの通信を失ったときにアクティブだったかどうかによって異なります。以下のセクションで説明しているように、何らかのデータ損失のリスクがあります。
3. プライマリのホスト・システムは稼働していますが、そのネットワークはアービターからもバックアップからも切り離されています。“[プライマリ・フェイルオーバー・メンバの計画外分離](#)”を参照してください。

6.2.3.1 フェイルオーバー・メンバを手動で強制的にプライマリにする処理

フェイルオーバー・メンバがプライマリになることができない場合はそれを強制することができますが、直前のプライマリのジャーナル・データが、強制の対象メンバのものより最新の可能性がある状況では、この強制を実行した場合にデータ損失のリスクがあります。以下の手順は、このリスクを見極めて、それに対処する方法を示しています。直近ジャーナル・データを有していることを確認できていないときに、メンバを強制的にプライマリにすると、他のミラー・メンバがミラーに再参加できなくなるため、再構築が必要になる場合があります（“[ミラー・メンバの再構築](#)”に記載のとおり）。

注意 処理を進める前に、プライマリがダウンしており、この手順の間もダウンしたままであることを確認してください。これを確認できない場合は、元のプライマリが再度使用可能になって、両方のメンバが同時にプライマリとして動作することになるというリスクを回避するために、この手順は中止することをお勧めします。この手順が適切であるかどうかについて不明な点がある場合は、[インターシステムズのサポート窓口](#)までご相談ください。

6.2.3.2 手動フェイルオーバーの前にバックアップがアクティブであったかどうかを判別する処理

InterSystems IRIS A と InterSystems IRIS B の 2 つのフェイルオーバー・メンバがあるとします。プライマリ (InterSystems IRIS A) との通信が失われた時点でバックアップ (InterSystems IRIS B) がアクティブだったこと、またそれにより InterSystems IRIS B が InterSystems IRIS A からの最新ジャーナル・データを有していることが、`MIRROR ルーチン`によって確認できると、1 つの手順のみを使用してフェイルオーバーを手動で実行できます。プライマリ上の障害によって接続が失われた場合、これによってデータ損失のリスクがもたらされることはありません。ただし、複数の障害が発生した場合、接続が途切れた後しばらくの間プライマリが動作を続けていたために、アクティブなバックアップがプライマリからの最新ジャーナル・データをすべて有しているとは限らない可能性があります。

以下の手順を使用して、バックアップがアクティブであったかどうかを判別します。

1. InterSystems IRIS A 上の InterSystems IRIS インスタンスとその ISCAgent の両方が実際にダウンしていることを確認します (そしてこの手動フェイルオーバーの手順全体を通して、ダウンしたままであることを確認します)。
2. InterSystems IRIS B 上で、ターミナルの %SYS ネームスペースにある `MIRROR ルーチン` ("[`MIRROR` ルーチンの使用法](#)") を参照してください) を実行します。
3. メイン・メニューから **[ミラー管理]** を選択して、次のサブメニューを表示します。

```

1) Add mirrored database(s)
2) Remove mirrored database(s)
3) Activate or Catchup mirrored database(s)
4) Change No Failover State
5) Try to make this the primary
6) Connect to Mirror
7) Stop mirroring on this member
8) Modify Database Size Field(s)
9) Force this node to become the primary
10) Promote Async DR member to Failover member
11) Demote Backup member to Async DR member
12) Mark an inactive database as caught up
13) Manage mirror dejournaling on async member (disabled)
14) Pause dejournaling for database(s)

```

4. **[このノードを強制的にプライマリにする]** オプションを選択します。通信が途絶えた時点でバックアップがアクティブであった場合は、以下のようなメッセージが表示されます。

```

This instance was an active backup member the last time it was
connected so if the primary has not done any work since that time,
this instance can take over without having to rebuild the mirror
when the primary reconnects. If the primary has done any work
beyond this point (file #98),
C:\InterSystems\MyIRIS\mgr\journal\MIRROR-GFS-20180815.009
then the consequence of forcing this instance to become the primary is
that some operations may be lost and the other mirror member may need
to be rebuilt from a backup of this node before it can join as
a backup node again.
Do you want to continue? <No>

```

プライマリのジャーナル・ファイルにアクセスできる場合、続行前に、示されているファイルが最新のものであることを確認できます。

プライマリとの通信が途絶えた時点でバックアップがアクティブでなかった場合は、以下のようなメッセージが表示されます。

```

Warning, this action can result in forcing this node to become
the primary when it does not have all of the journal data which
has been generated in the mirror. The consequence of this is that
some operations may be lost and the other mirror member may need
to be rebuilt from a backup of this node before it can join as
a backup node again.
Do you want to continue? <No>

```


6.2.3.3 アクティブなバックアップに対する手動フェイルオーバー

^MIRROR ルーチンの [このノードを強制的にプライマリにする] オプションによって、バックアップからプライマリへの接続が失われたときにバックアップがアクティブであったことが確認された場合は、以下のように手動フェイルオーバーの手順を実行します。

1. 手順を続行するには、Do you want to continue? プロンプトで、y と入力します。[このノードを強制的にプライマリにする] オプションは、ミラー・メンバがプライマリ化するために 60 秒間待ちます。60 秒以内に処理が正常完了しない場合、^MIRROR は、処理が成功しなかった可能性があることを報告して、かつメッセージ・ログの確認により、処理に失敗したのか、まだ処理中であるのかを判断するよう指示します。
2. ^MIRROR ルーチンによって、バックアップがプライマリになったことが確認されたら、InterSystems IRIS A を再起動できるときにそのようにします。InterSystems IRIS A は、InterSystems IRIS インスタンス再起動時にバックアップとしてミラーに参加します。

6.2.3.4 バックアップがアクティブでない場合の手動フェイルオーバー

^MIRROR ルーチンにより、プライマリ (InterSystems IRIS A) との通信を失った時点でバックアップ (InterSystems IRIS B) がアクティブであったことを確認できなかった場合でも、以下の手順を使用して手動フェイルオーバー・プロセスを続行できますが、続行すると、何らかのデータ損失のリスクがあります。InterSystems IRIS A の最新ミラー・ジャーナル・ファイルへアクセスできる場合、この手順で説明しているように、手動フェイルオーバー前に、それらのファイルを InterSystems IRIS A から InterSystems IRIS B にコピーすることによって、このリスクを最小限に抑えることができます。

1. プライマリのミラー・ジャーナル・ファイルにアクセスできる場合は、最新のファイルを InterSystems IRIS B にコピーします。このとき、InterSystems IRIS B での最新のジャーナル・ファイルから始め、InterSystems IRIS A にあるそれ以降のファイルを含めます。例えば、**MIRROR-MIRRORA-20180220.001** が InterSystems IRIS B での最新のファイルである場合、**MIRROR-MIRRORA-20180220.001** とそれ以降のファイルを InterSystems IRIS A からコピーします。ファイルの権限と所有権を確認し、必要に応じて、それらを既存のジャーナルファイルと一致するよう変更します。
2. データ損失のリスクを受け入れる場合は、プロンプトで y と入力し、続行することを確定します。これでバックアップがプライマリになります。[このノードを強制的にプライマリにする] オプションは、ミラー・メンバがプライマリ化するために 60 秒間待ちます。60 秒以内に処理が正常完了しない場合、^MIRROR は、処理が成功しなかった可能性があることを報告して、かつメッセージ・ログの確認により、処理に失敗したのか、まだ処理中であるのかを判断するよう指示します。
3. ^MIRROR ルーチンによって、バックアップがプライマリになったことが確認されたら、InterSystems IRIS A を再起動できるときにそのようにします。
 - ・ InterSystems IRIS インスタンスが再起動して、InterSystems IRIS A がバックアップとしてミラーに加わった場合、これ以上のステップは不要となります。障害メンバ上にはあるが、現在のプライマリ上にはないジャーナル・データは破棄されています。
 - ・ InterSystems IRIS のインスタンスが再起動したとき、InterSystems IRIS A がミラーに参加できない場合、“[ミラー・メンバの再構築](#)” で説明しているデータ不一致を表すメッセージ・ログ・メッセージが示すように、InterSystems IRIS A 上の最新データベース変更は、InterSystems IRIS B が強制的にプライマリになったときに InterSystems IRIS B に存在する最新ジャーナル・データよりも新しいものになります。これを解決するには、そのセクションの説明に従って、InterSystems IRIS A を再構築します。

6.2.4 プライマリ・フェイルオーバー・メンバの計画外分離

[自動フェイルオーバーのメカニズム](#)で説明しているように、バックアップとの通信とアービターとの通信を両方とも同時に失った場合、プライマリは無限の障害状態に入り、プライマリとしては動作できなくなります。通常、このような状態が生じた場合は、バックアップが引き継いで、プライマリになります。プライマリからバックアップへの接続がリストアされると、バックアップはプライマリを強制的にダウンさせます。または、接続をリストアする前に、ユーザ自身がプライマリを強制的にダウンさせることができます。

ただし、1 つのネットワーク・イベント (または一連のネットワーク・イベント) が原因で、フェイルオーバー・メンバとアービターがすべて同時に (またはほとんど同時に) お互いの通信を失った場合、プライマリがなくなる可能性があります。これは、バックアップは引き継ぐことができず、プライマリはプライマリとしては動作できなくなるからです。この状況は、“[自動フェイルオーバーのメカニズムの詳細](#)” セクションの “[アービター・モードで失われた接続に対するミラーの対応](#)” テーブルで最後のシナリオとして示されています。プライマリが切り離されて、エラーによりバックアップが引き継ぐことができない場合、同様の状況が生じる可能性があります。

このような状況が生じた場合、以下のオプションがあります。

- ・ フェイルオーバー・メンバ間の接続をリストアします。前のバックアップが前のプライマリに通信すると、これらのメンバはネゴシエートして、一方がプライマリになり、他方がバックアップになります。
- ・ 接続をリストアせずに、プライマリでターミナル・ウィンドウを開くことができる場合は、このウィンドウを開いて、プライマリで `^MIRROR` ルーチン (“[^MIRROR ルーチンの使用法](#)” を参照してください) を実行します。このルーチンによって、プライマリのインスタンスが無限の障害状態にあることが確認され、以下の 2 つのオプションが提供されます。
 - － もう一方のフェイルオーバー・メンバがダウンしており (ユーザがシャットダウンした場合がある)、そのメンバがプライマリになることはなく、プライマリ上の最新ミラー・ジャーナル・ファイルより新しいミラー・ジャーナル・ファイルがそのメンバで作成されていないことが確認できたら、そのメンバに強制的にプライマリとして動作を再開させることができます。そのように実行して、プライマリとバックアップの間の接続をリストアすると、そのバックアップはバックアップとして動作を再開します。
 - － これらの状態を確認できない場合は、プライマリをシャットダウンできます。その後、“[自動フェイルオーバーが発生しない場合のプライマリ・フェイルオーバー・メンバの計画外停止](#)” で説明しているいずれかの手順を使用して、バックアップへのフェイルオーバーを手動で実行できます。
- ・ プライマリでターミナル・ウィンドウを開くことができないが、もう一方のフェイルオーバー・メンバがダウンしており、そのメンバがプライマリになることはなく、プライマリ上の最新ミラー・ジャーナル・ファイルより新しいミラー・ジャーナル・ファイルがそのメンバで作成されていないことを確認できる場合は、プライマリの InterSystems IRIS インスタンスを再起動し、`^MIRROR` ルーチンの [[このノードを強制的にプライマリにする](#)] オプションを使用して、それを強制的にプライマリにすることができます。または、これらの状態を確認できない場合、プライマリの InterSystems IRIS インスタンスがダウンしていて、このままダウンし続けることを確認してから、“[自動フェイルオーバーが発生しない場合のプライマリ・フェイルオーバー・メンバの計画外停止](#)” で説明しているいずれかの手順を使用して、バックアップへのフェイルオーバーを手動で実行できます。

注意 ここに挙げた状態を確認することなく、プライマリにプライマリとして強制的に動作を再開させる場合は、データ損失や、両方のフェイルオーバー・メンバが同時にプライマリとして動作するリスクがあります。この手順が適切であるかどうかについて不明な点がある場合は、[インターシステムズのサポート窓口](#)までご相談ください。

6.2.5 両方のフェイルオーバー・メンバの計画外停止

同じイベントまたは異なるイベントが原因で、両方のフェイルオーバー・メンバに予期せず障害が発生した場合、適切な手順は、可用性の要件の制限内でフェイルオーバー・メンバのいずれかまたは両方を再起動できるかどうかによって異なります。ミラーの機能不全の状態での許容時間が長くなればなるほど、選択できるオプションの数は多くなる傾向があります。

- ・ 両方のエージェントと少なくとも 1 つの InterSystems IRIS インスタンスを再起動できる場合、フェイルオーバー・メンバはお互いにネゴシエートして、プライマリとして動作する方のフェイルオーバー・メンバを自動的に選択し、データ損失のリスクなしにミラーを稼働状態に戻します。
- ・ どちらのフェイルオーバー・メンバが直前のプライマリだったのかを確信しており、それを再起動できる場合に、それがもう一方のフェイルオーバー・メンバの InterSystems IRIS インスタンスまたはエージェントと通信できない (ダウンしているため) ときは、それは自動的にプライマリになりませんが、`^MIRROR` ルーチンの [[このノードを強制的にプ](#)

イマリにする] オプションを使用して、データ損失のリスクなく、それを手動で強制的にプライマリにすることができます (“[自動フェイルオーバーを使用しないプライマリ・フェイルオーバー・メンバの計画外停止](#)” に記載のとおり)。

- ・ どちらか1つのフェイルオーバー・メンバのみを再起動できるが、それが直前のプライマリだったかどうかわからない場合は、**MIRROR** ルーチンの [**このノードを強制的にプライマリにする**] オプションを使用して、データ損失のリスクなく、それを手動で強制的にプライマリにすることができます。

注意 アクティブでなかったバックアップを強制的にプライマリにすると、グローバル更新処理の一部が失われる可能性があり、他のミラー・メンバの再構築が必要になる場合があります (“[ミラー・メンバの再構築](#)” に記載のとおり)。この手順が適切であるかどうかについて不明な点がある場合は、[インターシステムズのサポート窓口](#)までご相談ください。

- ・ いずれのフェイルオーバー・メンバも再起動できない場合は、“[災害復旧の手順](#)”に進んでください。

6.3 災害復旧の手順

“**非同期ミラー・メンバ**”の説明のとおり、災害復旧 (DR) 非同期メンバはミラーリングされるデータベースの読み取り専用コピーを維持して、必要に応じて DR 非同期メンバをフェイルオーバー・メンバへ昇格できるようにします。DR 非同期メンバの昇格手順については、“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”で説明しています。このセクションでは、DR 非同期の昇格が使用できる3つのシナリオについて説明します。

- ・ [災害復旧時の昇格した DR 非同期への手動フェイルオーバー](#)
- ・ [昇格した DR 非同期への計画的フェイルオーバー](#)
- ・ [昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え](#)

この手順では、**InterSystems IRIS A** は元のプライマリ・フェイルオーバー・メンバ、**InterSystems IRIS B** は元のバックアップ・フェイルオーバー・メンバ、**InterSystems IRIS C** は昇格する DR 非同期メンバとします。

6.3.1 災害復旧時の昇格した DR 非同期への手動フェイルオーバー

ミラーから機能しているフェイルオーバー・メンバがなくなった場合、昇格した DR 非同期メンバに手動でフェイルオーバーすることができます。以下の手順では、これをオプションとするシナリオを扱います。

- ・ [追加ジャーナル・データなしの DR 昇格および手動フェイルオーバー](#)
- ・ [プライマリの ISCAgent のジャーナル・データありの DR 昇格および手動フェイルオーバー](#)
- ・ [ジャーナル・ファイルのジャーナル・データありの DR 昇格および手動フェイルオーバー](#)

注意 プライマリ・フェイルオーバー・メンバの **InterSystems IRIS** インスタンスが実際にダウンしていることを確認できない、かつインスタンスが使用できるようになる可能性がある場合、他のミラー・メンバへの手動フェイルオーバーは避けてください。手動フェイルオーバーを行った後に元のプライマリが使用できるようになると、両方のフェイルオーバー・メンバは同時にプライマリとして動作します。

注釈 プライマリの **InterSystems IRIS** インスタンスが、アービター制御モードでバックアップとアービターの両方から遮断されたことにより無限の障害状態にある場合、“[自動フェイルオーバーのメカニズムの詳細](#)”で説明するように、DR 非同期をフェイルオーバー・メンバに昇格できません。

6.3.1.1 追加ジャーナル・データなしの DR 昇格および手動フェイルオーバー

実際の災害復旧のシナリオで、両方のフェイルオーバー・メンバのホスト・システムがダウンして、それらのジャーナル・ファイルへのアクセスができなくなった場合は、旧プライマリから直近のジャーナル・データを取得せずに、DR 非同期メンバをプライマリに昇格させることができます。このとき、一部のデータが損失する可能性があります。フェイルオーバー・メンバのホスト・システムにアクセス可能な場合、“[プライマリの ISCAgent のジャーナル・データありの DR 昇格および手動フェイルオーバー](#)” または “DR” で示したいいずれかの手順を代用します。これらにより、昇格された DR 非同期は、プライマリになる前に最新ジャーナル・データを取得できるため、データ損失のリスクを最小限に抑えることができます。

ミラー VIP に参加していない DR 非同期をプライマリに昇格させた場合、このセクションで示している手順を実行する前に、ユーザとアプリケーションを新しいプライマリにリダイレクトするために必要なあらゆる変更を加える必要があります (“[フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト](#)” を参照)。

注釈 [\[開始時にマウントが必要\]](#)とマークされているすべてのミラーリングされたデータベース (“[ローカル・データベースのプロパティの編集](#)” を参照してください) がマウント、アクティブ化、およびキャッチアップされており、プライマリ化時にすぐに使用できるようになっていない限り、昇格した DR 非同期はプライマリ化を試行しません。

注意 前のプライマリの直近ジャーナル・データなしの DR 非同期のプライマリ昇格では、グローバル更新処理の一部が失われる可能性があり、他のミラー・メンバの再構築が必要になる場合があります (“[ミラー・メンバの再構築](#)” に記載のとおり)。この手順が適切であるかどうかについて不明な点がある場合は、[インターシステムズのサポート窓口](#)までご相談ください。

直近ジャーナル・データの取得なしで、DR 非同期メンバ (InterSystems IRIS C) をプライマリに昇格させるには、以下のことを実行します。

1. フェイルオーバー・パートナーを選択せずに、InterSystems IRIS C をフェイルオーバー・メンバに昇格させます。InterSystems IRIS C は、追加ジャーナル・データなしでプライマリになります。
2. 旧フェイルオーバー・メンバ (InterSystems IRIS A および InterSystems IRIS B) のホスト・システムが動作状態になったら、InterSystems IRIS 再起動前のできるだけ早い段階において、各メンバ上で InterSystems IRIS インスタンスの構成パラメータ・ファイルの [MirrorMember] セクションで ValidatedMember=0 を設定します (“[\[MirrorMember\]](#)” を参照してください)。これにより、InterSystems IRIS インスタンスに対して、以前のロールでの再接続ではなく、昇格した DR 非同期からのミラー内での新規ロール取得が指示されます。昇格指示では、この変更は必須であることが示されます。

注意 ValidatedMember=0 を設定しなければ、2 つのミラー・メンバが同時にプライマリとして動作してしまうおそれがあります。

3. 各旧フェイルオーバー・メンバ上で InterSystems IRIS を再起動します。
 - a. InterSystems IRIS が再起動して、そのメンバが DR 非同期としてミラーに加わった場合、これ以上のステップは不要となります。障害メンバ上にはあるが、現在のプライマリ上にはないジャーナル・データは破棄されています。
 - b. InterSystems IRIS の再起動時にメンバがミラーに参加できない場合、“[ミラー・メンバの再構築](#)” で説明しているデータ不一致を表すメッセージ・ログ・メッセージが示すように、メンバの最新データベース変更は、InterSystems IRIS C がプライマリになったときに InterSystems IRIS C に存在する最新ジャーナル・データよりも新しいものになります。これを解決するには、そのセクションの説明に従って、InterSystems IRIS A を再構築します。
4. InterSystems IRIS A および InterSystems IRIS B が再びミラーに加わったら、“[昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え](#)” に記載されている手順を使用して、メンバすべてを元のロールに戻すことができます。InterSystems IRIS A または InterSystems IRIS B のいずれかがバックアップとして再起動した場合、バックアップにフェイルオーバーするためにバックアップがアクティブになっているときに、InterSystems IRIS C の適切なシャットダウンから開始します。InterSystems IRIS A および InterSystems IRIS B の両方が DR 非同期として再起動した場合、いずれかをバックアップに昇格させた後に、InterSystems IRIS C で適切なシャットダウンを実行します。

他方の旧フェイルオーバー・メンバをバックアップに昇格させてから、InterSystems IRIS C を DR 非同期として再起動します。

6.3.1.2 プライマリの ISCAgent のジャーナル・データありの DR 昇格および手動フェイルオーバー

InterSystems IRIS A のホスト・システムは動作しているが、InterSystems IRIS インスタンスは動作しておらず、かつ再起動不可能である場合、以下の手順を使用して、InterSystems IRIS A の ISCAgent を通じて昇格した InterSystems IRIS C を InterSystems IRIS A の直近ジャーナル・データで更新できます。

1. InterSystems IRIS C を昇格させて、InterSystems IRIS A をフェイルオーバー・パートナーとして選択します。InterSystems IRIS C がフェイルオーバー・メンバに昇格され、InterSystems IRIS A のエージェントから直近ジャーナル・データを取得した後、プライマリとなります。
2. InterSystems IRIS A の InterSystems IRIS インスタンスを再起動すると、バックアップとして再びミラーに再び参加します。
3. InterSystems IRIS A がミラーに再び参加し、アクティブになった後に、“昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え”に記載されている手順を使用して、メンバすべてを元のロールに戻すことができます。その際、InterSystems IRIS C の適切なシャットダウンから始めて、InterSystems IRIS B の構成パラメータ・ファイルの [MirrorMember] セクションで ValidatedMember=0 を設定して (“[MirrorMember]” を参照)、InterSystems IRIS B を DR 非同期として再起動してバックアップに昇格させた後、InterSystems IRIS C を DR 非同期として再起動します。

注釈 InterSystems IRIS A のホスト・システムはダウンしているが、InterSystems IRIS B のホスト・システムはその InterSystems IRIS インスタンスが実行されていないにもかかわらず稼働している場合、“アクティブなバックアップに対する手動フェイルオーバー”の説明に従って、InterSystems IRIS B で `MIRROR` ルーチンを実行し、InterSystems IRIS B が障害発生時にアクティブなバックアップであったかどうかを判別します。その場合、前述の手順を使用しますが、昇格の際に InterSystems IRIS B をフェイルオーバー・パートナーとして選択して、InterSystems IRIS C が InterSystems IRIS B の ISCAgent から最新ジャーナル・データを取得できるようにします。

6.3.1.3 ジャーナル・ファイルのジャーナル・データありの DR 昇格および手動フェイルオーバー

InterSystems IRIS A と InterSystems IRIS B の両方のホスト・システムがダウンしているが、InterSystems IRIS A のジャーナル・ファイルにはアクセスできる場合、または InterSystems IRIS B のジャーナル・ファイルおよびメッセージ・ログが使用可能な場合、以下の手順を使用して、昇格前にプライマリからの最新ジャーナル・データで InterSystems IRIS C を更新できます。

1. 以下のように、InterSystems IRIS A または InterSystems IRIS B からの最新ジャーナル・ファイルで InterSystems IRIS C を更新します。
 - ・ InterSystems IRIS A のジャーナル・ファイルが使用可能な場合、InterSystems IRIS A から最新ミラー・ジャーナル・ファイルを InterSystems IRIS C にコピーします。このとき、InterSystems IRIS C の最新ジャーナル・ファイルから始め、InterSystems IRIS A にあるそれ以降のファイルを含めます。例えば、**MIRROR-MIRRORA-20180220.001** が InterSystems IRIS C で最新ファイルである場合、**MIRROR-MIRRORA-20180220.001** とそれ以降のファイルを InterSystems IRIS A からコピーします。
 - ・ InterSystems IRIS A のジャーナル・ファイルは使用できないが、InterSystems IRIS B のジャーナル・ファイルおよびメッセージ・ログは使用可能な場合は、以下の操作を行います。
 - a. 以下のように、InterSystems IRIS B がキャッチアップされている可能性が高いことを確認します。
 1. InterSystems IRIS A およびそのエージェントが使用不可になったと同時に InterSystems IRIS B が InterSystems IRIS A から切断されたことを確認します。InterSystems IRIS B が切断された時間は、その **messages.log** ファイルで、以下に類似したメッセージを検索することによって確認できます (“[管理ポータルを使用した InterSystems IRIS の監視](#)” を参照してください)。

MirrorClient: Primary AckDaemon failed to answer status request

2. 切断時に InterSystems IRIS B がアクティブなバックアップであったことを、その **messages.log** ファイルで以下に類似したメッセージを検索することによって確認します。

Failed to contact agent on former primary, can't take over

注意 **messages.log** ファイル内の以下のようなメッセージは、InterSystems IRIS B が切断時にアクティブではなかったことを示しています。

nonactive Backup is down

昇格された DR 非同期がキャッチアップされていることを確認できないときに、その DR 非同期を強制的にプライマリにすると、ミラーによって生成されたジャーナル・データがすべてない状態でプライマリになるおそれがあります。その結果、グローバル更新処理の一部が失われる可能性があり、他のミラー・メンバのバックアップからの再構築が必要になる場合があります。この手順が適切であるかどうかについて不明な点がある場合は、[インターシステムズのサポート窓口](#)までご相談ください。

- b. InterSystems IRIS B がアクティブであったことを確認できた場合、InterSystems IRIS B から最新ミラー・ジャーナル・ファイルを InterSystems IRIS C にコピーします。このとき、InterSystems IRIS C の最新ジャーナル・ファイルから始め、InterSystems IRIS B にあるそれ以降のファイルを含めます。例えば、**MIRROR-MIRRORA-20180220.001** が InterSystems IRIS C で最新ファイルである場合、**MIRROR-MIRRORA-20180220.001** とそれ以降のファイルを InterSystems IRIS C からコピーします。ファイルの権限と所有権を確認し、必要に応じて、それらを既存のジャーナル・ファイルと一致するよう変更します。
2. フェイルオーバー・パートナーを選択せずに、InterSystems IRIS C をフェイルオーバー・メンバに昇格させます。InterSystems IRIS C がプライマリになります。
3. InterSystems IRIS A および InterSystems IRIS B の問題が修正されたら、InterSystems IRIS の再起動前のできるだけ早い段階において、各メンバ上で InterSystems IRIS インスタンスの構成パラメータ・ファイルの **[MirrorMember]** セクションで **ValidatedMember=0** を設定します ("**[MirrorMember]**" を参照してください)。昇格指示では、この変更は必須であることが示されます。これを実行したら、InterSystems IRIS A (直前にプライマリであったメンバ) から始めて、各メンバ上の InterSystems IRIS を再起動します。
 - a. InterSystems IRIS が再起動して、メンバがバックアップまたは DR 非同期としてミラーに参加した場合、これ以上のステップは不要となります。障害メンバ上にはあるが、現在のプライマリ上にはないジャーナル・データは破棄されています。
 - b. InterSystems IRIS インスタンスの再起動時にそのメンバがミラーに参加できない場合、"**ミラー・メンバの再構築**" で説明している、データの不一致を表すメッセージ・ログ・メッセージが示すように、そのメンバ上の最新データベース変更は、InterSystems IRIS C がプライマリになったときに InterSystems IRIS C に存在する最新ジャーナル・データよりも新しいものになります。これを解決するには、そのセクションの説明に従って、そのメンバを再構築します。
4. ほとんどの場合、DR 非同期システムは、プライマリ・フェイルオーバー・メンバにとって、適切な永続的ホストではありません。InterSystems IRIS A および InterSystems IRIS B が再びミラーに参加した後に、"**昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え**" に記載されている手順を使用して、メンバすべてを元のロールに戻します。InterSystems IRIS A または InterSystems IRIS B のいずれかがバックアップとして再起動した場合、バックアップにフェイルオーバーするためにバックアップがアクティブになっているときに、InterSystems IRIS C の適切なシャットダウンから開始します。InterSystems IRIS A および InterSystems IRIS B の両方が DR 非同期として再起動した場合、いずれかをバックアップに昇格させた後に、InterSystems IRIS C で適切なシャットダウンを実行します。他方の旧フェイルオーバー・メンバをバックアップに昇格させてから、InterSystems IRIS C を DR 非同期として再起動します。

6.3.2 昇格した DR 非同期への計画的フェイルオーバー

DR 非同期を 1 つ以上ミラーに追加して、災害復旧機能を装備させている場合、各 DR 非同期への計画的フェイルオーバーによって、この機能を定期的にテストすることをお勧めします。このテストを実行するか、または他の何らかの理由で DR 非同期にフェイルオーバーを実行する場合（フェイルオーバー・メンバを含むデータ・センタ内の計画的な停電など）、以下の手順を使用します。

1. InterSystems IRIS C をフェイルオーバー・メンバに昇格させます。InterSystems IRIS A が使用できるため、フェイルオーバー・パートナーの選択を要求されることはありません。InterSystems IRIS C はバックアップとなり、InterSystems IRIS B（存在する場合）は DR 非同期に降格されます。

注釈 ミラーに開始するフェイルオーバー・メンバが 1 つだけしかない場合でも、手順は同じです。フェイルオーバー・パートナーの選択を要求されることはなく、InterSystems IRIS C がバックアップとなるので、すぐにミラーは 2 つのフェイルオーバー・メンバを持つことになります。

2. InterSystems IRIS C がアクティブになったら（“バックアップ・ステータスと自動フェイルオーバー”を参照してください）、InterSystems IRIS A 上で適切なシャットダウンを実行します。自動フェイルオーバーがトリガされ、InterSystems IRIS C はプライマリとして引き継ぐことができます。

3. InterSystems IRIS C で適宜テストをした後、InterSystems IRIS A を再起動すると、自動的にバックアップとしてミラーへ参加します。

または、実際の災害時にはプライマリは利用できない可能性が高いため、プライマリを自動的にバックアップにせずに再起動して同期状態を維持したい場合は、再起動前に（その ISCAgent を使用して）DR 非同期に降格してから、後で準備が整った時点でプライマリをフェイルオーバー・メンバに昇格することができます。この方法の詳細は、“DR 非同期へのバックアップの降格”を参照してください。

4. InterSystems IRIS A がバックアップとしてアクティブになったら、InterSystems IRIS C で適切なシャットダウンを実行して、InterSystems IRIS A へのフェイルオーバーを行います。
5. InterSystems IRIS B（存在する場合）をフェイルオーバー・メンバに昇格させます。これはバックアップとなります。
6. InterSystems IRIS C の InterSystems IRIS インスタンスを再起動すると、自動的に DR 非同期として元のロールでミラーに参加します。

“ミラーリング・アーキテクチャとネットワーク構成のサンプル”で説明しているように、フェイルオーバー・メンバのミラー・プライベート・アドレスへのネットワーク・アクセス権がない DR 非同期の昇格は、プライマリとして機能することを目的とし、ほかに動作中のフェイルオーバー・メンバがない場合にのみ実行する必要があります。従って、この場合、前述の手順は当てはまりません。代わりに、次の手順を実行します。

1. InterSystems IRIS B（存在する場合）で、適切なシャットダウンを実行し、InterSystems IRIS A のみがフェイルオーバー・メンバ（プライマリ）として機能している状況にします。
2. InterSystems IRIS C がキャッチアップされたら（“ミラー・メンバのジャーナル転送およびデジャーナリングのステータス”を参照してください）、InterSystems IRIS A で適切なシャットダウンを実行します。
3. “プライマリの ISCAgent のジャーナル・データありの DR 昇格および手動フェイルオーバー”の説明に従って、InterSystems IRIS C をプライマリに昇格します。新しいプライマリが前のプライマリの ISCAgent と通信して、この手順の間に最新ジャーナル・データを有していることを確認します。
4. InterSystems IRIS C で適宜テストを実行した後、シャットダウンします。
5. InterSystems IRIS A を再起動します。これは、自動的にプライマリになります。
6. InterSystems IRIS B を再起動します（存在する場合）。InterSystems IRIS C の昇格によって、DR 非同期として参加します。
7. InterSystems IRIS B をバックアップに昇格します。

8. InterSystems IRIS C を再起動すると、自動的に元のロールである DR 非同期としてミラーに参加します。

注釈 このセクションの両方の手順で、InterSystems IRIS B が存在しない場合、つまりミラーがプライマリと非同期のみによって構成されている場合、再起動時に InterSystems IRIS C がバックアップになります。[“バックアップ・フェイルオーバー・メンバのメンテナンス”](#)の説明に従って、これを DR 非同期に降格します。

6.3.3 昇格した DR 非同期によるフェイルオーバー・メンバの一時的置き換え

“計画的停止の手順”および“計画外停止の手順”で説明した手順の一部は、単一フェイルオーバー・メンバでのミラーの一時操作に関係します。常に動作中のバックアップ・フェイルオーバー・メンバを維持する必要はありませんが、これにより、プライマリの障害発生時におけるデータベース・アクセスの中断やデータ損失の可能性からの防御が確実にになります。そのため、計画的または計画外のフェイルオーバー・メンバ停止により、プライマリしか使用できない場合、DR 非同期メンバをバックアップ・フェイルオーバー・メンバへ一時的昇格することを検討するのが妥当です。ただし、これを実行する前に、以下の点を考慮します。

- DR 非同期がフェイルオーバー・メンバから遠く離れた別のデータ・センタにある場合、それらの間には相当なネットワーク遅延が存在することがあります。DR メンバが昇格して、アクティブなフェイルオーバー・メンバになると、この往復遅延はプライマリとバックアップ間の同期データ・レプリケーションの重要な要素となり（[“ミラー同期”](#)を参照）、ミラーにアクセスするアプリケーションのパフォーマンスに悪影響を与える可能性があります（[“ネットワーク遅延に関する考慮事項”](#)を参照）。
- “[ミラーリング・アーキテクチャとネットワーク構成のサンプル](#)”で説明しているように、DR 非同期にフェイルオーバー・メンバのミラー・プライベート・アドレスへのネットワーク・アクセス権がない場合、その DR 非同期は、ここで示した手順では使用できません。その DR 非同期の昇格は、プライマリとして機能することを目的とし、ほかに動作中のフェイルオーバー・メンバがない場合にのみ実行する必要があるためです。
- ミラーでユーザおよびアプリケーションの自動リダイレクトに VIP が使用されており（[“フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト”](#)を参照）、別のサブネット上にあるため DR 非同期がミラー VIP を取得できない場合、通常、これらの手順を使用できません。

注釈 このオプションを使用する前に、“[DR 非同期メンバのフェイルオーバー・メンバへの昇格](#)”において、フェイルオーバー・パートナー選択の情報と、昇格時にエージェントに通信できないフェイルオーバー・メンバ上で ValidatedMember=0 を設定する要件について確認します。

現在のバックアップ・フェイルオーバー・メンバである InterSystems IRIS B で、計画的メンテナンスを実施する必要がある場合（[“バックアップ・フェイルオーバー・メンバのメンテナンス”](#)を参照してください）、次の操作を実行できます。

- キャッチアップされる DR 非同期である InterSystems IRIS C を昇格させます（[“ミラー・メンバのジャーナル転送およびデジャーナリングのステータス”](#)を参照してください）。InterSystems IRIS C は自動的にバックアップとなり、InterSystems IRIS B は DR 非同期に降格されます。
- InterSystems IRIS B の InterSystems IRIS インスタンスまたはホスト・システムをシャットダウンして、計画的メンテナンスを完了します。
- InterSystems IRIS B を再起動すると、これが DR 非同期としてミラーに加わります。
- InterSystems IRIS B がキャッチアップされたら、これをフェイルオーバー・メンバに昇格させ、バックアップとして元のロールに戻します。InterSystems IRIS C は、自動的に元のロールである DR 非同期に降格されます。

現在のプライマリ・フェイルオーバー・メンバである InterSystems IRIS A で、計画的メンテナンスを実施する必要がある場合（[“プライマリ・フェイルオーバー・メンバのメンテナンス”](#)を参照してください）、次の操作を実行できます。

- InterSystems IRIS B がアクティブになったら（[“ミラー同期”](#)を参照してください）、InterSystems IRIS A 上で適切なシャットダウンを実行します。自動フェイルオーバーがトリガされ、InterSystems IRIS B はプライマリとして引き継ぐことができます。

2. キャッチアップされる DR 非同期メンバである InterSystems IRIS C を昇格させます。InterSystems IRIS C は自動的にバックアップとなります。
3. InterSystems IRIS A の計画的メンテナンスを完了し、必要に応じて、ホスト・システムのシャットダウンおよび再起動を行います。
4. InterSystems IRIS A の InterSystems IRIS インスタンスを再起動すると、DR 非同期としてミラーに参加します。
5. InterSystems IRIS A がキャッチアップされたら、これをフェイルオーバー・メンバに昇格させます。これがバックアップになり、InterSystems IRIS C は自動的に降格され、元のロールに戻ります。
6. InterSystems IRIS A がアクティブになったら、InterSystems IRIS B 上で適切なシャットダウンを実行します。自動フェイルオーバーがトリガされ、InterSystems IRIS A は元のロールに戻されます。
7. InterSystems IRIS B の InterSystems IRIS インスタンスを再起動すると、元のロールでミラーに参加します。

InterSystems IRIS B の計画外停止があった場合、または **InterSystems IRIS A の計画外停止**による InterSystems IRIS B への自動/手動フェイルオーバーを行った場合（“[計画外停止の手順](#)”を参照してください）、次の操作を実行できます。

1. キャッチアップされる DR 非同期メンバである InterSystems IRIS C を昇格させます。InterSystems IRIS C は自動的にバックアップとなります。
2. 障害が発生したフェイルオーバー・メンバを再起動します。DR 非同期の昇格時に障害の発生したメンバの ISCAgent が通信不可能であった場合、InterSystems IRIS インスタンス再起動前のできるだけ早い段階において、InterSystems IRIS インスタンスの構成パラメータ・ファイルの [MirrorMember] セクションで ValidatedMember=0 を設定する必要があります（“[\[MirrorMember\]](#)”を参照してください）。昇格指示では、この変更は必須であることが示されます。旧フェイルオーバー・メンバの InterSystems IRIS インスタンスを再起動すると、DR 非同期としてミラーに加わります。
3. 再起動したフェイルオーバー・メンバがキャッチアップされたら、これをフェイルオーバー・メンバに昇格させます。このメンバはバックアップとなり、InterSystems IRIS C は自動的に元のロールである DR 非同期に降格されます。
4. フェイルオーバー・メンバ間で現在のロールを交換する場合、バックアップがアクティブになるとき、現在のプライマリで適切なシャットダウンを実行し、自動フェイルオーバーをトリガします。別のフェイルオーバー・メンバを再起動します。これはバックアップとしてミラーに加わります。

