



InterSystems IRIS のアップグレードと変更

Version 2024.1
2024-06-03

InterSystems IRIS のアップグレードと変更
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 アップグレードの概要	1
1.1 互換性の目標	1
1.2 アップグレードの前に	1
2 Windows でのアップグレードまたは変更	5
2.1 Windows でのアップグレード手順	5
2.2 再インストールまたはアンインストール	6
2.3 自動アップグレードまたは再インストール	7
2.4 自動削除	7
2.4.1 特別な考慮事項	7
3 UNIX®, Linux、および macOS でのアップグレードまたは変更	9
3.1 UNIX®, Linux、および macOS でのアップグレード手順	9
3.2 UNIX®, Linux、および macOS でのアンインストール	10
4 ミラーのアップグレード	13
4.1 ミラーのアップグレード手順の選択	13
4.2 アップグレード時のミラー・メンバの追加	15
4.3 ミラーのアップグレードの用語	15
4.4 ミラーのアップグレード手順	16
4.4.1 メンテナンス・リリース・アップグレード	16
4.4.2 メジャー・バージョン・アップグレード (ミラーリング・データベースの変更あり)	17
4.4.3 メジャー・バージョン・アップグレード (ミラーリング・データベースの変更なし)	18
4.4.4 メジャー・バージョン・アップグレード (計画的ダウンタイムあり)	19
5 ECP 構成のアップグレード	21
6 アップグレード後のタスク	23
6.1 メンテナンス・リリースのアップグレード後のタスク	23
6.2 メジャー・バージョンのインストール後のタスク	24
6.2.1 ネームスペースのコンパイル方法	24

1

アップグレードの概要

このトピックは、前のバージョンの InterSystems IRIS®、InterSystems IRIS for Health™、または HealthShare® HealthConnect からアップグレードするお客様を対象としています。InterSystems Caché® からの移行に関心があるお客様は、"[InterSystems IRIS に移行すべき理由](#)" を参照してください。

重要 各アプリケーションは、お客様に導入してライブ・データの処理を開始する前に、アップグレードされた環境でくまなくテストすることをお勧めします。

1.1 互換性の目標

各リリースの目標は、既存のアプリケーションに対する変更を必要とせずに、新機能および強化機能にスムーズにアップグレードすることです。ただし、エラー修正、重要な機能強化、または該当する標準への変更により、この目標を達成できない場合もあります。このような場合、インターシステムズでは、アプリケーションの変更が必要な特定インスタンスすべてを開示して、新しいリリースへの移行に必要な工数をお客様が測定できるようにしています。

リリース固有の変更 ([以下にリンクされています](#)) について読んだ後、お客様のアプリケーションにはどれも影響を与えないという結論に至ることもあります。このような場合でも、アプリケーションがいかに堅牢に設計されているか、およびいかに適切に実装されているかに関係なく、お客様の判断を確認するとともに、アプリケーションがアップグレードの影響を受けていないことを示す品質保証テスト結果に代わるものではありません。

1.2 アップグレードの前に

すべてのプラットフォームで、以下のアップグレード・タスクが必要です。これらのタスクは、アップグレード手順を実行する前に行ってください。

注釈 メンテナンス・リリースにアップグレードする場合、アスタリスク(*)が付いているタスクはオプションです。ただし、すべてのタスクを実行すると、最適なパフォーマンスが保証されます。

1. "[Incompatibility History](#)" のドキュメントを読み、必要に応じて互換性の欠如について考慮する。
2. ご使用の製品のリリース・ノートのアップグレードに関する章を読み、必要に応じて互換性の欠如について考慮する。
 - ・ "[InterSystems IRIS リリース・ノート](#)" の "[このリリースへのアップグレード](#)"
 - ・ "[InterSystems IRIS for Health リリース・ノート](#)" の "[このリリースへのアップグレード](#)"

- ・ “HealthShare Health Connect リリース・ノート” の “このリリースへのアップグレード”

インストールの前に実行する必要がある具体的な推奨事項があれば、それに従ってください。

- 更新先の InterSystems IRIS バージョンについて、“インターシステムズのサポート対象プラットフォーム”を読む。ご使用のテクノロジーが InterSystems IRIS の新規バージョンでサポートされていることを確認してください。
- アプリケーションのすべてのソース・コードが利用可能であることを確認する*—メジャー・アップグレード後は、すべてのクラス・コードを InterSystems IRIS の新規バージョンでリコンパイルするか、既に新規バージョンでコンパイルされているクラス・コードを用意する必要があります。そうしないと、アップグレード後にアプリケーションを実行できなくなります。ルーチン・コードをすべてリコンパイルすることをお勧めします。

配置モードで実行されているクラスのコードがあるかどうか（アップグレード完了後に再び配置モードで配置できます）、およびクラスから生成されていないカスタム・ルーチンのコード（*.mac または *.int）があるかどうかを確認します。

- カスタムのクラス、ルーチン、およびグローバルを保存する*—アップグレード時に、IRISSYS データベースが変更されます。**%SYS** ネームスペース内の独自のクラス、ルーチン、およびグローバルがアップグレード・インストールの影響を受けないようにするには、これらの名前が “Z”、“z”、“%Z”、または “%z” で始まるようにします。**.int** ルーチンと **.obj** ルーチンはすべて（Z*、z*、%Z*、および %z* を除く）、アップグレード時に **%SYS** ネームスペースから削除されます。

同様に、アップグレード時に **IRISLIB**、**IRISTEMP**、および **ENSLIB** のデータベースは完全に置換されます。

- ユーザ・ファイルを保存する*—ユーザ・ファイルがアップグレード中に削除または置換されないよう、これらのファイルを `install-dir¥mgr¥User` ディレクトリに保存します。これは、アップグレード中に変更の影響を受けない唯一のディレクトリです。
- インストール・マニフェストを作成するかどうかを決定する*—サイレント・インストールを行う場合、マニフェストがあると便利です。アップグレード完了後すぐにアップグレード後のタスクを呼び出して実行できるため、インタラクティブな手順が必要なくなります。詳細は、“[インストール・マニフェストの作成および使用](#)”を参照してください。
- 起動時に実行するユーザ・コードをプリコンパイルする*—インスタンスの起動時にアプリケーションがユーザ・コードを呼び出す場合（%ZSTART、ZAUTHENTICATE、または他の方法を使用）、InterSystems IRIS の新規バージョンのインスタンスでユーザ・コードをプリコンパイルし、インストール・マニフェストを使用してアップグレード・プロセスの一部としてそのコードをインストールする必要があります。この作業を行わないと、インスタンスは起動できません。
- 更新されたライセンス・キーを取得する*—詳細は、“システム管理ガイド”の“[InterSystems IRIS ライセンスの管理](#)”の章を参照してください。
- システムの整合性をチェックする—既存のディレクトリに対してシステム整合性チェックを実行し、データベースが破損していないことを確認します。詳細は、“データ整合性ガイド”の“[データ整合性の概要](#)”の章を参照してください。
- システムをバックアップする—アップグレードの前に、オペレーティング・システムの通常の完全バックアップ手順を使用して、システムの完全なバックアップを実行することをお勧めします。アップグレードに失敗した場合、このバックアップからリストアしなければならないことがあります。バックアップの詳細は、“データ整合性ガイド”の“[バックアップとリストア](#)”の章を参照してください。
- すべてのプロダクションを停止し、プロダクションの自動開始を無効にする—システムでプロダクションが実行されている場合は、“プロダクションの管理”の“プロダクションの開始と停止”の章にある“プロダクションの停止”のセクションの手順に従ってください。

プロダクションを再起動する前にコードをリコンパイルできるように、自動開始を無効にすることが重要です。

重要 外部言語ゲートウェイ構成をすべて削除した場合、アップグレード・プロセス中に検証エラーが発生することがあります。これらの検証エラーは、影響を受けるバージョンからアップグレードする際に発生します。アップグレード先のバージョンには関係ありません。影響を受けるバージョンは、2021.1.0、2021.1.1、2021.1.2、2022.1.0、2022.1.1 などです。

これらのエラーを防ぐには、任意のポート番号を持つローカル・ゲートウェイを指すリモート・タイプの単一ゲートウェイ構成を追加します。例えば、[サーバ名 / IPアドレス] を 127.0.0.1、[ポート] を 1 に設定し、ForUpgrade という名前を付けることができます。この操作はアップグレード前の任意の時点で実行でき、通常のシステム・オペレーションに影響を与えません。この構成は、アップグレードの完了後に削除できます。

注釈 HealthShare Health Connect または InterSystems IRIS for Health のユーザの場合、Caché/Ensemble プラットフォームをベースとする Health Connect/HSAP から移行されたインスタンスでは、インストール・ディレクトリの **components.ini** ファイルに、この製品ではもう使用されていないデータベース **MPRLLIB** への参照が含まれることがあります。各行の先頭にセミコロンを挿入することで、アップグレードの前にこの参照をコメントアウトしてください。これによって、このデータベースは存在しないという、誤解を招くエラー・メッセージが **messages.log** に記録されることがなくなります。

例：

```
; [MPRLLIB]
; Version=15.032.9686
[HSLIB]
Version=2018.1.0
Compatibility_HSAALIB=15.0
Compatibility_HSPILIB=14.0
Compatibility_VIEWERLIB=17.0
```


2

Windows でのアップグレードまたは変更

ここでは、InterSystems IRIS、InterSystems IRIS for Health、または HealthShare HealthConnect の単一インスタンスをアップグレードする手順について説明します。この章の“[アップグレードの前に](#)”のセクションのタスクを実行してから、次に進んでください。ミラーまたは ECP 構成をアップグレードする場合は、この章の“[ミラーのアップグレード](#)”および“[ECP 構成のアップグレード](#)”のセクションの情報も確認してください。

アップグレード・プロセスの一部としてマニフェストを使用する場合は、標準のインストール手順を実行する前に、“[インストール・マニフェストの作成および使用](#)”の手順を参照してください。

注意 インスタンスのアップグレードを開始する前に、インスタンスを正常にシャットダウンすることが重要です。正常にシャットダウンされたことを確認するには、シャットダウンの終了後に `messages.log` ファイルを調べます。ログに以下のようなエントリが含まれる場合、シャットダウンは正常に行われています。

```
...
05/03/19-14:24:13:234 (5204) 0 Journal restore not required at next startup
05/03/19-14:24:13:234 (5204) 0 Transaction rollback not required at next startup
...
```

これらのエントリが記述されていない場合、インスタンスは正常にシャットダウンされていません。アップグレードを続行する前に、[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。

2.1 Windows でのアップグレード手順

InterSystems IRIS の新規バージョンへのアップグレードに必要なすべての機能は、インストーラによって提供されます。マニフェストを使用する場合に必要なサイレント・アップグレードを実行するには、“[自動アップグレードまたは再インストール](#)”の情報を確認してください。

アップグレードを実行するには、以下の手順に従います。

1. インストーラ・ファイル (`IRIS-win_x64.exe` など) をダブルクリックします。
2. インストーラにより、ホスト上にある既存の InterSystems IRIS インスタンスがすべて一覧表示されます。アップグレードするインスタンスの名前を選択し、[OK] をクリックします。
3. InterSystems IRIS ライセンス・キーの入力を求められたら、[ライセンス] をクリックして新しい `iris.key` ファイルを参照します。インストーラによって `install-dir/mgr` ディレクトリで新しいキー・ファイルが検出された場合、ライセンス・キーの入力を求めるプロンプトは表示されずに次に進みます。
4. ローカルの Microsoft インターネット・インフォメーション・サービス (IIS) Web サーバが検出された場合、[ローカル Web サーバ] ダイアログ・ボックスで、インストーラがこのインスタンスに [Web サーバを自動的に構成](#) するかどうかを選択できます。インストーラが Web サーバを自動構成しないようにする場合は、インストールの完了後に [手動](#) で構

成する必要があります。ローカル Web サーバが検出されなかった場合は、ダイアログ・ボックスで代わりに **[インストールを中断]** するか **[Web ゲートウェイなしでインストールを続行]** するかを尋ねられます。インストールを続行することを選択した場合は、インストールの完了後に **手動で Web サーバを構成** する必要があります。

重要 Microsoft IIS Web サーバは、アップグレード・プロセスで自動的に構成できるため、この Web サーバの使用をお勧めします。アップグレード・プロセスの開始前に、それがインストールされ、実行されていることを確認してください。ほとんどの場合、IIS Web サーバを手動で構成する必要はありません。

5. インストーラによってライセンス・キーが検証されたら、**[アップグレード]** をクリックします。

重要 インストールの進行中にインストールを中断しないでください。エラー・メッセージが表示されてアップグレードが失敗した場合は、問題を修正して、アップグレード・インストールを再開します。

6. アップグレードが完了したら、**[完了]** をクリックします。

7. `install-dir¥mgr` ディレクトリにある `messages.log`、`iboot.log`、および `ensinstall.log` を調べて、エラーがないことを確認します。致命的なエラーが見つかった場合、エラーを修正して、インストーラを再び実行します。

重要 オペレーティング・システムがラージ・メモリ・ページを使用するように構成されている場合は、起動メッセージをチェックして、これらの設定に従って共有メモリが割り当てられていることを確認します。以下のようなメッセージが表示された場合は、サーバを再起動して、メモリ不足の状態に陥るのを防いでください。

Failed to allocate 592MB shared memory using large pages. Switching to small pages.

2.2 再インストールまたはアンインストール

セットアップを実行してからインストーラと同じバージョンの InterSystems IRIS インスタンスを選択するか、または Windows のコントロール・パネルで **[プログラムと機能]** を選択してから InterSystems IRIS インスタンスを選択することで、インスタンスを変更またはアンインストールできます。

“[Windows での手動インストール](#)” で説明されているセットアップを実行してから、**[インスタンス選択]** ボックスでインストーラと同じバージョンの InterSystems IRIS インスタンスを選択するか、または **[プログラムと機能]** でインスタンスを選択してから **[変更]** または **[修復]** ボタンを使用すると、**[インスタンスのアップグレード]** `instancename` ダイアログ・ボックスが表示されます。

注釈 **[プログラムと機能]** で **[アンインストール]** ボタンを選択すると、すぐにアンインストール操作が開始されます。

[次] をクリックすると、**[プログラムの変更、修復、または削除]** ダイアログ・ボックスが表示されます。このダイアログで適切なオプションを選択して、インスタンスを変更、修復、またはアンインストールします。

- ・ **[変更]** を選択すると、**[カスタム・セットアップ]** ダイアログ・ボックスが表示されます (“[Windows でのカスタム・インストール](#)” を参照)。このダイアログ・ボックスを使用して、追加または削除するコンポーネント・グループまたはコンポーネントを選択できます。コンポーネントは “[セットアップ・タイプ別のインストールされるコンポーネント](#)” のテーブルに記載されています。
- ・ 消失または破損したファイルやレジストリ・エントリなどのインスタンスの問題を修復するには、**[修復]** を選択します。
- ・ インスタンスをアンインストールするには、**[削除]** を選択します。

重要

InterSystems IRIS のアンインストールには、InterSystems IRIS インストーラまたは Windows コントロール・パネルの **[プログラムと機能]** のみを使用してください。他のアンインストール・プログラムはサポートされていないため、使用すると予期しない結果が生じる可能性があります。

2.3 自動アップグレードまたは再インストール

新規インスタンスのインストールのほかに、インストール済みの既存のインスタンスに対して InterSystems IRIS インストーラを呼び出すこともできます。そのためには、/instance フラグを使用して、ターゲットの既存のインスタンスの名前を指定する必要があります。インストーラが実行するアクションは、インスタンスのバージョンとインストール・ファイルのバージョンとを比較した結果によって異なります。

- ・ インストール・ファイルが、ターゲットのインストール済みインスタンスと同じバージョンである場合、インストーラはインスタンスを再インストール（修復）します。
- ・ インストール・ファイルが、ターゲットのインストール済みインスタンスより後のバージョンである場合、インストーラはインスタンスを新規バージョンにアップグレードします。

例えば、インストール・ファイルより前のバージョンのインストール済みインスタンス **IRISB** の自動アップグレードを実行するには、以下を使用します。

```
C:\downloads\IRIS-2019.1.0.516.0-win_x64.exe /instance IRISB /qn
```

ターゲット・インスタンスを指定して REINSTALL プロパティ（“**コマンド行のプロパティ**” のテーブルを参照）を使用すると、“**カスタム・インストール可能な機能**” のテーブルにリストされている 1 つ以上の機能を再インストールできます。例えば、インストール済みインスタンス **IRISB** に対してスタジオを再インストールするには、以下のコマンドを使用します（インストール・ファイルと **IRISB** が同じバージョンであると想定しています）。

```
C:\downloads\IRIS-2018.1.0.508.0-win_x64.exe /instance IRISB /qn REINSTALL=studio
```

2.4 自動削除

自動削除を起動するには、以下のようにアンインストールするインスタンスと REMOVE=ALL プロパティを指定します。

```
<path>\<installer>.exe /instance <instancename> /q[b|n] REMOVE=ALL
```

また、REMOVE プロパティを使用して、“**カスタム・インストール可能な機能**” のテーブルに示されている特定の機能を削除することもできます。例えば、インスタンス **IrisC** から Apache 2.0 Web ゲートウェイを削除するには、以下のコマンドを使用します。

```
C:\downloads\IRIS-2018.1.0.508.0-win_x64.exe /instance IrisC /qn REMOVE=cspgateway,cspapache20
```

2.4.1 特別な考慮事項

元のインストール・パッケージにアクセスできない場合は、以下のように Windows® インストーラのコマンド行アプリケーション（msiexec）およびレジストリの情報を使用して、自動モードでアンインストールを実行できます。

```
msiexec /x {<product_guid>} /qn /l <logfile>
```

ここで、<product_guid> はインストールしたバージョンの **ProductCode** プロパティ値です。

レジストリの以下の場所から **ProductCode** プロパティ値を取得できます。

プロセッサ・タイプ	レジストリの場所
32 ビット	KEY_LOCAL_MACHINE\SOFTWARE\InterSystems\IRIS\Configurations\<instance>
64 ビット	KEY_LOCAL_MACHINE\SOFTWARE\Wow64\InterSystems\IRIS\Configurations\<instance>

ここで <instance> は、自動モードでアンインストールする InterSystems IRIS インスタンスの名前です。**ProductCode** プロパティ値は、以下のような行に表示されます。

```
ProductCode      REG_SZ      {80E3F658-2D74-4A81-92AD-FD16CD226154}
```

また、msiexec には “[コマンド行のプロパティ](#)” テーブル内の任意のプロパティを使用することもできます。msiexec の詳細は、TechNet のドキュメント “[Msiexec \(コマンドライン オプション\)](#)” ([https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759262\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc759262(v=ws.10))) を参照してください。

3

UNIX®、Linux、および macOS でのアップグレードまたは変更

ここでは、InterSystems IRIS、InterSystems IRIS for Health、または HealthShare HealthConnect の単一インスタンスをアップグレードする手順について説明します。この章の“[アップグレードの前に](#)”のセクションのタスクを実行してから、次に進んでください。ミラーまたは ECP 構成をアップグレードする場合は、この章の“[ミラーのアップグレード](#)”および“[ECP 構成のアップグレード](#)”のセクションの情報も確認してください。

アップグレード・プロセスの一部としてマニフェストを使用する場合は、標準のインストール手順を実行する前に、“[インストール・マニフェストの作成および使用](#)”の手順を参照してください。

注意 インスタンスのアップグレードを開始する前に、インスタンスを正常にシャットダウンすることが重要です。正常にシャットダウンされたことを確認するには、シャットダウンの終了後に `messages.log` ファイルを調べます。ログに以下のようなエントリが含まれる場合、シャットダウンは正常に行われています。

```
...
05/03/19-14:24:13:234 (5204) 0 Journal restore not required at next startup
05/03/19-14:24:13:234 (5204) 0 Transaction rollback not required at next startup
...
```

これらのエントリが記述されていない場合、インスタンスは正常にシャットダウンされていません。アップグレードを続行する前に、[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。

3.1 UNIX®、Linux、および macOS でのアップグレード手順

InterSystems IRIS の新規バージョンへのアップグレードに必要なすべての機能は、インストール・スクリプト `irisinstall` によって提供されます。代わりにサイレント・アップグレードを実行する場合は、このドキュメントの“InterSystems IRIS の UNIX®、Linux、および macOS へのインストール”の章にある“[UNIX®、Linux、および macOS での自動インストール](#)”の情報を確認してください。

注釈 `irisinstall` を実行しているユーザのプロファイルに `CDPATH` 変数の値が設定されている場合、アップグレードは失敗します。

アップグレードを実行するには、以下の手順に従います。

1. インストール・キットが `.tar` ファイル形式の場合は、まず解凍する必要があります (“[インストール・キットの解凍](#)”を参照)。

2. root 特権を持つユーザとして、インストール・ファイルの最上位にある `irisinstall` スクリプトを実行し、インストール手順を開始します。

```
# sudo sh /<pathname>/irisinstall
```

ここで、`<pathname>` はインストール・キットの場所で、通常はキットの抽出先の一時ディレクトリです。

3. スクリプトにより、ホスト上にある既存の InterSystems IRIS インスタンスがすべて一覧表示されます。
4. **[インスタンス名の入力]** プロンプトで、アップグレードする InterSystems IRIS インスタンスの名前を入力して、**Enter** キーを押します。
5. ライセンス・キー・ファイルの入力を求められたら、「`keypath`**¥iris.key**」と入力します。ここで、`keypath` は新しい **iris.key** ファイルの場所です。インストール・スクリプトによって `install-dir/mgr` ディレクトリで新しいライセンス・キーが検出された場合、ライセンス・キーの入力を求めるプロンプトは表示されずに次に進みます。
6. ローカル Web サーバが検出されると、その Web サーバを使用してインストールに接続するかどうかを尋ねられます。「`y`」と入力すると、Web サーバは**自動的に接続**されます。「`n`」と入力すると、Web サーバは自動的に接続されず、インストール完了後に**手動で構成**する必要があります。Web サーバが検出されない場合、中止するかどうかを尋ねるプロンプトが表示されます。インストールを続行することを選択した場合は、インストールの完了後に手動で Web サーバを構成する必要があります。

重要 Apache httpd Web サーバは、アップグレード・プロセスで自動的に構成できるため、この Web サーバの使用をお勧めします。アップグレード・プロセスの開始前に、それがインストールされ、実行されていることを確認してください。ほとんどの場合、Apache Web サーバを手動で構成する必要はありません。

7. アップグレード・オプションの概要が表示され、もう一度、アップグレードを確認するよう求められます。**Enter** キーを押して、アップグレードを続行します。

重要 インストールの進行中にインストールを中断しないでください。エラー・メッセージが表示されてアップグレードが失敗した場合は、問題を修正して、アップグレード・インストールを再開します。

8. `install-dir/mgr` ディレクトリにある `messages.log`、`iboot.log`、および `ensinstall.log` を調べて、エラーがないことを確認します。致命的なエラーが見つかった場合、エラーを修正して、インストール・スクリプトを再び実行します。

アップグレードが完了すると、インスタンスが再起動され、**インストール・マニフェスト**が実行されます。

重要 オペレーティング・システムがヒュージ・メモリ・ページを使用するように構成されている場合は、起動メッセージをチェックして、これらの設定に従って共有メモリが割り当てられていることを確認します。以下のようなメッセージが表示された場合は、サーバを再起動して、メモリ不足の状態に陥るのを防いでください。

```
Failed to allocate 1468MB shared memory using Huge Pages. Startup will retry with standard pages.
```

3.2 UNIX®, Linux、および macOS でのアンインストール

InterSystems IRIS を安全にアンインストールするには、以下の手順に従います。

1. `iris list` コマンドを使用して、対象のマシンにあるインスタンスをすべて一覧表示し、削除したい InterSystems IRIS インスタンスの名前を見つけます。

```
iris list
```

2. 目的のインスタンスが停止していることを確認します。停止していない場合、`iris stop` コマンドを使用して、このインスタンスを停止します。

```
iris stop <instname>
```

`instname` は、インストール時に指定したインスタンス名です。このインスタンスがハングアップしている場合は、`iris force` を使用して強制終了します。

```
iris force <instname>
```

3. `iris delete` コマンドを使用して、インスタンスを削除します。

```
iris delete <instname>
```

4. 以下のオペレーティング・システム・コマンドを使用して、インストール・ディレクトリを削除します。

```
rm -r <directory>
```

重要 この操作では、保持する必要があるファイルも削除されます。例えば、ライセンス・キー (`iris.key`)、構成ファイル (`iris.cpf`)、ユーザ・データベース・ファイル (`iris.dat`) などです。

このアンインストール手順では、ジャーナルや一時データベース・ファイルを含め、通常の InterSystems IRIS 処理中にインストールおよび作成されたファイルがすべて削除されます。

重要 SUSE Linux Enterprise Server 9 プラットフォームでは非同期スクリプトレットが使用されるので、アンインストール処理でファイルを削除する前に InterSystems IRIS が停止していることを保証できません。

4

ミラーのアップグレード

このセクションでは、InterSystems IRIS インスタンス、アプリケーション、またはその両方を InterSystems IRIS ミラーのメンバ上でアップグレードする手順を示します。

“InterSystems IRIS 高可用性ガイド” の [“ミラーリング”](#) の章に記載されているように、ミラーのすべてのフェイルオーバー・メンバと DR 非同期メンバが同じ InterSystems IRIS バージョンである必要がありますが、アップグレード中だけは異なるバージョンであってもかまいません。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同じバージョンが要求される場合があります。詳細は、“高可用性ガイド” の [“ミラーリング”](#) の章の [“InterSystems IRIS インスタンスの互換性”](#) を参照してください。

4 つのミラー・アップグレード・パスの中から選択できます。どの手順を実行する必要があるかは、以下の [“ミラーのアップグレード手順の選択”](#) のセクションの各要素を参照して決定してください。これら 4 つの手順は、以下の [“ミラーのアップグレード手順”](#) のセクションに記載されています。

ミラーのアップグレードに関する以下の 2 つのセクションも参照してください。

- ・ [“ミラーのアップグレードの用語”](#) では、ミラーのアップグレード手順で使用されるいくつかの用語を定義します。
- ・ [“アップグレード時のミラー・メンバの追加”](#) では、ミラーにメンバを追加すべきタイミングについて説明します。

重要

systemd をサポートする Linux システムでは、systemctl コマンドまたは `/etc/init.d/ISCAgent` スクリプトを使用して ISCAgent を管理できますが、どちらか一方の方法を選択して、その方法のみを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。詳細は、“InterSystems 高可用性ガイド” の [“ミラーリング”](#) の章の [“Linux システムでの ISCAgent の開始”](#) を参照してください。

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。

4.1 ミラーのアップグレード手順の選択

ミラーのアップグレードに適した手順を選択する際に考慮すべき主要要素は、以下の 2 つです。

- ・ InterSystems IRIS のインストール済みバージョンのメンテナンス・リリースにアップグレードするのか、または InterSystems IRIS の新規メジャー・バージョンにアップグレードするのか。
- ・ アップグレードにミラーリング・データベースへの変更が含まれるか。

メンテナンス・リリースにアップグレードする場合に、アプリケーション・アップグレードは行わないときは、2 番目の質問を考慮する必要はありません。常に、“[メンテナンス・リリース・アップグレード](#)” の簡単な手順を使用できます。この手順では、計画的フェイルオーバーの実行に要する時間のみ、ミラーは使用できない状態になります。

ただし、InterSystems IRIS のメジャー・バージョン間でアップグレードする場合、アプリケーション・アップグレードを実行する場合、またはその両方を行う場合は、アップグレードにミラーリング・データベースの変更が含まれる可能性があります。このような変更は、アプリケーション・アクセスが無効化されている間に、アップグレード手順の一部として、機能しているプライマリ・フェイルオーバー・メンバ上で行う必要があります。このため、アップグレードでは、ミラーリング・データベースの変更を行わない場合よりも長いダウンタイムが必要になります（その後、変更はバックアップ・メンバおよび非同期メンバ上でミラーによって複製されます）。

重要 メジャー・アップグレードの後に、インスタンス上のすべてのアプリケーション・ネームスペース内のクラスをすべてリコンパイルすることをお勧めします。一部のルーチンも、リコンパイルが必要です。実行するアプリケーション・アップグレードで、アプリケーション・データの変更が必要になることがあります。いずれの状況においても、アップグレードによって、ミラーリングされたデータベースが変更されることがあります。

メジャー・リリースにアップグレードする場合、アプリケーション・アップグレードを実行する場合、またはその両方を行う場合は、以下のいずれかの条件に該当するかどうかを判断する必要があります。

- ・ クラスとルーチンは、ミラーリングされたデータベースに保存されます。これには、アプリケーション・データも格納されています。クラスは、プライマリ上でリコンパイルする必要があります（アプリケーションがアップグレードされていない場合でも）。ルーチンについては、プライマリでのリコンパイルをお勧めします。
- ・ アプリケーション・アップグレードに伴って、ミラーリングされたデータベース内のデータをアップグレードする必要がある場合。これらの変更は、プライマリ上で実行する必要があります。

注釈 すべての相互運用プロダクションにはミラーリングされたアプリケーション・コードがあるため、相互運用プロダクションでミラーリングされている環境には、これらの条件が常に適用されます。

メジャー・アップグレードまたはアプリケーション・アップグレード、あるいはその両方がこれらのいずれかの条件を満たす場合は、“[メジャー・バージョン・アップグレード \(ミラーリングされたデータベースの変更あり\)](#)” の手順を使用します。この手順では、計画的フェイルオーバーを実行し、ミラーリングされたデータベースに必要な変更を加える時間のみ、ミラーは使用できない状態になります。

アップグレードに、ここに挙げられているミラーリングされたデータベースの変更が含まれない場合は、“[メジャー・バージョン・アップグレード \(ミラーリングされたデータベースの変更なし\)](#)” の手順を検討してください。この手順では、計画的フェイルオーバーの実行に要する時間のみ、ミラーは使用できない状態になります。

計画的メンテナンスの時間が十分にあり、ミラーのダウンタイムを最小限に抑えるかどうか重大な問題ではない場合は、代わりに“[メジャー・バージョン・アップグレード \(計画的ダウンタイムあり\)](#)” の簡単な手順を使用することができます。

メジャー・アップグレードの場合、ミラーの InterSystems IRIS インスタンス、アプリケーション・コード、またはその両方のアップグレードに、各手順の一般的な順序が適用されます。アップグレードする対象に合わせて、個々のステップを調整します。

注釈 状況によっては、特定の手順で追加のステップや詳細が必要になることがあります。これでもまだどの手順が適しているのかわからない場合や、記載されている特定の手順が適切かどうかかわからない場合は、[インターシステムズのサポート窓口 \(WRC\)](#) にお問い合わせください。

4.2 アップグレード時のミラー・メンバの追加

ミラーにメンバを追加する計画がある場合、このセクションで説明するいずれかのアップグレードの実行計画を立てるまでメンバの追加を延期できます。新規バージョンのメンバをアップグレード時に追加することで、後からアップグレードを行う必要がなくなります。直ちに処理を続行し、説明どおりにアップグレードを完了させるのであれば、アップグレード中いつでも、新規バージョンのメンバをミラーに追加することができます。ただし、制約事項が 1 つあり、新規バージョンのメンバがプライマリになったら、そのメンバは、他のすべてのメンバがアップグレードされるまでプライマリであり続ける必要があります。

新規ハードウェアにミラーを移行する場合は、アップグレード中に新規メンバを追加すると非常に効果的です。いずれかのフェイルオーバー・メンバをアップグレードしてからフェイルオーバーするのではなく、ターゲット・システムに新規バージョンの新規インスタンスをインストールし、そのインスタンスを DR 非同期メンバとしてミラーに追加し、バックアップに昇格させてからフェイルオーバーすることで、ミラー・プライマリを新規システムに移行することができます。この手法を繰り返して、残りのフェイルオーバー・メンバを移行することができます。この手法を説明するため、このセクションの手順には、既存のバックアップをアップグレードする代わりに、新規バージョンの新規バックアップを追加するステップが含まれます。

4.3 ミラーのアップグレードの用語

このセクションの手順では、以下の用語が使用されています。

クラスおよびルーチンのアップグレード

インスタンスのすべてのアプリケーション・ネームスペース内のクラスをすべてリコンパイルします。これは、InterSystems IRIS メジャー・バージョン・アップグレード (“[アップグレード後のタスク](#)” を参照) またはアプリケーション・アップグレード、あるいはその両方の後に行う必要があります。状況によっては、以下の 1 つ以上の操作が必要になることがあります。

- ・ 前述のように、クラスが存在するミラーリングされたデータベースにアプリケーション・データも含まれる場合、これらのクラスは、機能しているプライマリ・フェイルオーバー・メンバ上でローカルにリコンパイルする必要があります (アプリケーション・アップグレードでこれらに変更されるかどうかは関係ありません)。ルーチンが存在する場合は、これらをリコンパイルすることをお勧めします。
- ・ アプリケーション・データとは別個のミラーリングされないルーチン・データベースに格納されているクラスおよびルーチンは、現在のプライマリかどうかに関係なく、ミラー・メンバ上でリコンパイルできます。
- ・ 別個のミラーリングされないルーチン・データベースに格納されているクラスおよびルーチンは、“プリコンパイル” することもできます。それには、既にターゲット InterSystems IRIS リリースにアップグレードされているシステム上でデータベースのコピーをリコンパイルしてから、アップグレード後に各ミラー・メンバに配布します。

システム A

最初はプライマリ・フェイルオーバー・メンバであるミラー・メンバ。

システム B

最初はバックアップ・フェイルオーバー・メンバであるミラー・メンバ。

システム C

フェイルオーバー・メンバに昇格された、新規 InterSystems IRIS バージョンに新たに追加された DR 非同期メンバ。

フェイルオーバーなしの設定

MIRROR ルーチンを使用してフェイルオーバーなしの状態を設定し、フェイルオーバーが発生しないようにします。手順については、“InterSystems IRIS 高可用性ガイド”の“ミラーリング”の章の[“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。

適切なシャットダウンの実行

iris stop コマンドを使用して、インスタンスを正常にシャットダウンします (“InterSystems IRIS システム管理ガイド”の“InterSystems IRIS 複数インスタンスの使用法”の章の[“InterSystems IRIS インスタンスの制御”](#)を参照してください)。

フェイルオーバー・メンバへの昇格

“InterSystems IRIS 高可用性ガイド”の“ミラーリング”の章の[“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、DR 非同期ミラー・メンバをフェイルオーバー・メンバに昇格させる手順を実行します。

ミラー・モニタの表示

インスタンスの管理ポータルの [ミラー・モニタ] ページで、ミラーとそのメンバの状態を確認します。詳細は、“InterSystems IRIS 高可用性ガイド”の“ミラーリング”の章の[“ミラー・モニタの使用法”](#)を参照してください。

4.4 ミラーのアップグレード手順

4 つのミラー・アップグレード・パスの中から選択できます。どの手順を実行する必要があるかは、“[ミラーのアップグレード手順の選択](#)”のセクションの各要素を参照して決定してください。4 つの手順は、以下のとおりです。

- ・ [メンテナンス・リリース・アップグレード](#)
- ・ [メジャー・バージョン・アップグレード \(ミラーリング・データベースの変更あり\)](#)
- ・ [メジャー・バージョン・アップグレード \(ミラーリング・データベースの変更なし\)](#)
- ・ [メジャー・バージョン・アップグレード \(計画的ダウンタイムあり\)](#)

最初の 3 つの手順は、可能な限り短いアプリケーション・ダウンタイムで必要なアップグレードを実行できるよう設計されています。これらの手順は、アップグレードのさまざまな状況に対応するために異なります。最後の手順は、やや簡易な手順であり、ダウンタイムの最小化が優先事項でない場合に、任意のタイプのアップグレードに使用できます。

4.4.1 メンテナンス・リリース・アップグレード

InterSystems IRIS の新規メジャー・バージョンではなく InterSystems IRIS のメンテナンス・リリースにアップグレードし、アプリケーションに変更を加えない場合は、以下の手順を使用します。この手順では、計画的フェイルオーバーの実行に要する時間のみ、ミラーは使用できない状態になります。

1. バックアップが完全にアップグレードされるまでフェイルオーバーが発生しないようにするには、[フェイルオーバーなしを設定](#)します。
2. 以下の 2 つの操作のいずれかを実行します。

- ・ 既存のバックアップをアップグレードします。
 - a. バックアップ (B) の適切なシャットダウンを実行します。
 - b. バックアップ (B) を InterSystems IRIS の新規バージョンでアップグレードします。システム B がバックアップになります。
 - ・ 新規バージョンの新規バックアップを追加します。
 - a. システム C に InterSystems IRIS の新規バージョンをインストールします。
 - b. システム C を DR 非同期メンバとしてミラーに追加します。
 - c. システム C を昇格させてフェイルオーバー・メンバにします。システム C がバックアップになり、システム B は DR 非同期になります。
3. ミラー・モニタを表示して、バックアップ (B または C) がアクティブになっていることを確認します。
 4. フェイルオーバーなしをクリアします。
 5. プライマリ (A) の適切なシャットダウンを実行します。ミラーがフェイルオーバーし、バックアップ (B または C) がプライマリとして引き継ぎます。
 6. システム A を InterSystems IRIS の新規バージョンでアップグレードします。システム A がバックアップになります。
 7. システム C がプライマリになり、システム B が DR 非同期に降格された場合は、システム B をアップグレードします。

4.4.2 メジャー・バージョン・アップグレード (ミラーリング・データベースの変更あり)

InterSystems IRIS の新規メジャー・バージョンへのアップグレードまたはアプリケーション・アップグレード、あるいはその両方を実行する際、ミラーリング・データベースの変更が必要であると判断した場合 (“ミラーのアップグレード手順の選択” を参照)、以下の手順を使用します。この手順では、計画的フェイルオーバーを実行し、ミラーリング・データベースに必要な変更を加える時間のみに、ミラーは使用できない状態になります。

1. バックアップが完全にアップグレードされるまでフェイルオーバーが発生しないようにするには、フェイルオーバーなしを設定します。
2. 以下の 2 つの操作のいずれかを実行します。
 - ・ 既存のバックアップをアップグレードします。
 - a. バックアップ (B) の適切なシャットダウンを実行します。
 - b. バックアップ (B) を InterSystems IRIS の新規バージョンでアップグレードします。システム B がバックアップになります。
 - c. システム B で、ミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
 - ・ 新規バージョンの新規バックアップを追加します。
 - a. システム C に InterSystems IRIS の新規バージョンをインストールします。
 - b. システム C を DR 非同期メンバとしてミラーに追加します。
 - c. システム C を昇格させてフェイルオーバー・メンバにします。システム C がバックアップになり、システム B は DR 非同期になります。
3. ミラー・モニタを表示して、バックアップ (B または C) がアクティブになっていることを確認します。
4. サイトで確立されている手法を使用して、ミラーへの新規ユーザ・アクセスを禁止します。また、システム B がプライマリになったときに通常開始されるアプリケーション・ジョブを無効にする必要があります。

5. フェイルオーバーなしをクリアします。
6. プライマリ (A) の適切なシャットダウンを実行します。ミラーがフェイルオーバーし、バックアップ (B または C) がプライマリとして引き継ぎます。
7. 新規プライマリ (B または C) でミラーリングされたクラスとルーチンをアップグレードします。アプリケーション・アップグレードでミラーリング・データベースのさらなる変更が必要な場合、それらの変更を行います。
8. ミラーへのユーザ・アクセスをリストアします。
9. 完全にアップグレードされるまで、システム A がプライマリとして引き継がないようにするには、フェイルオーバーなしを設定します。
10. システム A を InterSystems IRIS の新規バージョンでアップグレードします。システム A がバックアップになります。
11. システム A で、ミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
12. フェイルオーバーなしをクリアします。
13. システム C がプライマリになり、システム B が DR 非同期に降格された場合は、システム B をアップグレードし、システム B でミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。

4.4.3 メジャー・バージョン・アップグレード (ミラーリング・データベースの変更なし)

InterSystems IRIS の新規メジャー・バージョンへのアップグレードまたはアプリケーション・アップグレード、あるいはその両方を実行する際、ミラーリング・データベースの変更が不要であると判断した場合 ("[ミラーのアップグレード手順の選択](#)" を参照)、以下の手順を使用できます。

別個のミラーリングされないルーチン・データベースを常に維持する静的アプリケーションの場合は、これが最も簡易な手順です。ただし、別個のルーチン・データベースが通常はミラーリングされる場合でも、アップグレード期間中にこれらのデータベースをミラーから削除することで、この手順を使用できます。

重要 ルーチン・データベースが通常はミラーリングされる場合は、ルーチン・データベースをミラーから削除する前に、アプリケーション・データが含まれていないことを確認してください。

ECP アプリケーション・サーバがミラーに接続されている場合、通常はミラーリングされるルーチン・データベースでこの手順を使用することはできません。

1. サイトで確立されている手順を使用して、コード・フリーズとアプリケーション構成のフリーズを実行してアップグレード中のアプリケーションの変更を禁止し、アップグレード中にそのルーチン・データベースが変更されないようにします。
2. バックアップが完全にアップグレードされるまでフェイルオーバーが発生しないようにするには、[フェイルオーバーなしを設定](#)します。
3. 以下の 2 つの操作のいずれかを実行します。
 - ・ 既存のバックアップをアップグレードします。
 - a. ルーチン・データベースがミラーリングされている場合は、[システム B](#) でミラーから削除します。
 - b. [ミラー・モニタ](#)を表示して、バックアップ (B) がアクティブになっていることを確認します。
 - c. バックアップ (B) の [適切なシャットダウン](#)を実行します。
 - d. バックアップ (B) を InterSystems IRIS の新規バージョンでアップグレードします。システム B がバックアップになります。
 - e. システム B で、[クラスおよびルーチンをアップグレード](#)します。
 - ・ 新規バージョンの新規バックアップを追加します。

- a. システム C に InterSystems IRIS の新規バージョンをインストールします。
 - b. システム C を DR 非同期メンバとしてミラーに追加します。
 - c. システム C を昇格させてフェイルオーバー・メンバにします。システム C がバックアップになり、システム B は DR 非同期になります。
 - d. ルーチン・データベースがミラーリングされている場合は、システム C と B でミラーから削除します。
 - e. ミラー・モニタを表示して、バックアップ (C) がアクティブになっていることを確認します。
4. フェイルオーバーなしをクリアします。
 5. プライマリ (A) の適切なシャットダウンを実行します。ミラーがフェイルオーバーし、バックアップ (B または C) がプライマリとして引き継ぎます。
 6. 完全にアップグレードされるまで、システム A がプライマリとして引き継がないようにするには、フェイルオーバーなしを設定します。
 7. システム A を InterSystems IRIS の新規バージョンでアップグレードします。システム A がバックアップになります。
 8. システム A で、ミラーリングされないクラスおよびルーチンをアップグレードします。
 9. システム C がプライマリになり、システム B が DR 非同期に降格された場合は、システム B をアップグレードし、システム B でミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
 10. バックアップ (B または C) でミラーから削除したミラーリングされているルーチン・データベースが新規プライマリになる前に、以下を行います。
 - a. システム A で、ミラーからルーチン・データベースを削除します。
 - b. 既存のデータベースをミラーに追加する手順を使用して、新規プライマリ (B または C) でミラーにデータベースを追加してから、バックアップ (A) および DR 非同期 (C がプライマリの場合は B) でデータベースをバックアップおよびリストアします ("InterSystems IRIS 高可用性ガイド" の "ミラーリング" の章の "ミラーへのデータベースの追加" を参照)。これらのバックアップは、新たに追加されたミラーリングされたデータベースの最初のバックアップであるため、保持します。障害が発生した場合、アップグレード前に作成された古いバックアップを使用して、これらのデータベースをリストアすることはできません。
 11. フェイルオーバーなしをクリアします。

4.4.4 メジャー・バージョン・アップグレード (計画的ダウンタイムあり)

InterSystems IRIS の新規メジャー・バージョンへのアップグレードまたはアプリケーション・アップグレード、あるいはその両方を実行する際、計画的メンテナンスの時間が十分にあり、ミラー・ダウンタイムを最小化する必要がない場合、プライマリでミラーリング・データベースの変更が必要かどうかにかかわらず、以下の簡易な手順を使用できます。

1. サイトで確立されている手法を使用して、ミラーへのすべてのユーザ・アクセスを禁止します。また、インスタンス起動時に通常開始されるアプリケーション・ジョブを無効にする必要があります。
2. バックアップ (B) の適切なシャットダウンを実行します。
3. プライマリ (A) の適切なシャットダウンを実行します。
4. システム A を InterSystems IRIS の新規バージョンでアップグレードします。システム A がプライマリになります。
5. システム A で、ミラーリングされたクラスとルーチンをアップグレードします。アプリケーション・アップグレードでミラーリング・データベースのさらなる変更が必要な場合、それらの変更を行います。
6. システム A で、ミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
7. 以下の 2 つの操作のいずれかを実行します。

- ・ 既存のバックアップをアップグレードします。
 - a. システム B を InterSystems IRIS の新規バージョンでアップグレードします。システム B がバックアップになります。
 - b. システム B で、ミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
 - ・ 新規バージョンの新規バックアップを追加します。
 - a. システム C に InterSystems IRIS の新規バージョンをインストールします。
 - b. システム C を DR 非同期メンバとしてミラーに追加します。
 - c. システム C を昇格させてフェイルオーバー・メンバにします。システム C がバックアップになります。
 - d. システム B を InterSystems IRIS の新規バージョンでアップグレードします。システム B が DR 非同期になります。
 - e. システム B で、ミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
8. システム C がプライマリになり、システム B が DR 非同期に降格された場合は、システム B をアップグレードし、システム B でミラーリングされないクラスおよびルーチンをアップグレードします (ある場合)。
 9. ミラーへのユーザ・アクセスをリストアします。

5

ECP 構成のアップグレード

一般に、ECP アプリケーション・サーバは、接続先のデータベース・サーバをアップグレードする前にアップグレードする必要があります。アプリケーション・サーバは、アップグレード後にリコンパイルするローカルのルーチン・データベース、またはデータベース・サーバ上の“プリコンパイル済み”のルーチン・データベース (ターゲット InterSystems IRIS バージョンが稼働中の別個のシステムで以前にリコンパイル済み) のいずれかにアクセスできる必要があります。

以下のいずれかが当てはまる場合、アップグレード済みのアプリケーション・サーバと未アップグレードのアプリケーション・サーバの両方、およびデータベース・サーバにユーザが接続した状態でローリング・アップグレードを使用すると、ダウンタイムを最小限に抑えることができます。

- ・ アプリケーションの変更がない場合。
- ・ アプリケーションの変更はあるが、新しいコードによって新しいデータ構造が生成されない場合。つまり、古いコードが、新しいコードによって生成されたデータを操作できる場合。

新しいアプリケーション・コードが古いデータで動作するものの、古いコードでは認識できない新しいデータ構造が生成される可能性がある場合は、以下の手順に従います。

1. アプリケーション・サーバをローリング・アップグレードします。ただし、アップグレード後にユーザがアプリケーション・サーバに接続できないようにします (アプリケーション・サーバの処理能力が段階的に低下します)。
2. 必要なだけのアプリケーション・サーバがアップグレードされたら、アップグレード済みアプリケーション・サーバへのユーザ・アクセスをリストアし、残りの (アップグレードされていない) アプリケーション・サーバとデータ・サーバ (必要な場合) へのユーザ接続をすべて終了して、有効化するまでユーザがこれらのシステムにアクセスできないようにします。
3. 残りのアプリケーション・サーバをアップグレードし、アップグレード後に各アプリケーション・サーバへのユーザ・アクセスをリストアします。
4. 必要に応じてデータ・サーバをアップグレードし、ユーザ・アクセスをリストアします (データ・サーバをアップグレードすると、アプリケーション・アクティビティが一時停止します。一時停止する期間は、アップグレードに伴うダウンタイムの長さによって異なります)。

ECP 構成のアップグレード方法に関する質問や懸念事項については、[インターシステムズ WRC \(インターシステムズのカスタマ・サポート窓口\)](#) までお問い合わせください。

6

アップグレード後のタスク

必要なアップグレード後のタスクは、InterSystems IRIS の新規メジャー・バージョンへのアップグレードか、インストール済みバージョンのメンテナンス・リリースへのアップグレードかによって異なります。また、以下の点についても注意してください。

- ・ アップグレード中にインスタンスを 8 ビットから Unicode に変更した場合、データベースは自動的に変換されません。ユーザが自分自身でこの変換作業を行う必要があります。詳細は、“サーバ側プログラミングの入門ガイド”の“ネームスペースとデータベース”の章の“データベースの移植性”のセクションを参照してください。
- ・ [OpenAPI 2.0 仕様](#) ファイルを再インポートする必要はありません。
- ・ Web サービス定義 (WSDL) ファイルを再インポートする必要はありません。
- ・ クエリ・キャッシュは常にアップグレードの際に削除されます。これらは、必要に応じてリコンパイルされ、キャッシュされます。
- ・ インストール・プロセスで Web サーバを自動構成しなかった場合は、[これを手動で接続](#)する必要があります。
- ・ Web サーバで 80 以外のポート番号を使用している場合は、IDE と接続できるようにするため、CPF の [WebServerPort](#) および [WebServerURLPrefix](#) パラメータを Web サーバのポート番号に変更する必要があります。
- ・ プライベート Web サーバ (2023.1 以前) が設定されているインスタンスからアップグレードした場合は、[プライベート Web サーバを無効化および削除](#)する必要があります。

6.1 メンテナンス・リリースのアップグレード後のタスク

アップグレードの完了後、システムでプロダクションを実行している場合は、“プロダクションの管理”の“プロダクションの開始と停止”の章にある“プロダクションの開始”のセクションの手順に従ってプロダクションを再開します。

通常、メンテナンス・リリース・アップグレードでは、外部ファイルやクライアントを変更したり、クラスやルーチンをリコンパイルしたりする必要はありません。ただし、クライアント上のスタジオ・バージョンは、接続先の InterSystems IRIS サーバ・バージョンと同じかそれ以降である必要があります。

リコンパイルする場合は、“[メジャー・バージョンのインストール後のタスク](#)”のセクションの情報を参照してください。

6.2 メジャー・バージョンのインストール後のタスク

InterSystems IRIS を新規メジャー・バージョンにアップグレードした後は、“リリース・ノート”の“[アップグレードの一般情報](#)”に示されているガイダンスに従うことが重要です。以下のタスクも実行する必要があります（この章の前の手順の一部として実行していない場合）。

- ・ クラスおよびルーチンのリコンパイル – 各ネームスペースに含まれるすべてのクラスとすべてのルーチンをリコンパイルすることをお勧めします。手順については、“[ネームスペースのコンパイル方法](#)”を参照してください。アップグレードするインスタンスに固有の依存関係およびその他のニーズを考慮して、そのための独自のツールや手順を使用できます。
- ・ %SYS クラスおよびルーチンのリコンパイル – すべてのクラスおよびルーチンをコンパイルする際、既定では %SYS ネームスペースはスキップされます。必ずこのネームスペースも再コンパイルしてください。手順については、“[ネームスペースのコンパイル方法](#)”を参照してください。“Z”、“z”、“%Z”、“%z”で始まるクラスとルーチンのみが、アップグレード時に保持されます。これらのクラスとルーチンはプリコンパイルできません。
- ・ プロキシ・クラスの再生成 – アップグレードしたインスタンスで生成されていたプロキシ・クラスはすべて、“[InterSystems IRIS 言語バインディング](#)”セットの該当するガイドの手順に従って再生成する必要があります。
この項目は Web サービスおよび Web クライアントには適用されません。Web サービス定義 (WSDL) ファイルを再インポートする必要はありません。
- ・ ブラウザ・キャッシュのクリア – ブラウザ・キャッシュに、インストール済みバージョンの InterSystems IRIS との互換性がなくなった JavaScript ファイルが含まれていると、エラーが発生することがあります。アップグレードの完了後、直ちにブラウザ・キャッシュをクリアしてください。

お使いの環境および使用するコンポーネントによっては、以下のタスクが必要になることがあります。

- ・ クエリ・プランの確認。新しいメジャー・バージョンにアップグレードするときに、既存のクエリ・プランは自動的に凍結されます。これにより、ソフトウェアのメジャー・アップグレードによって既存のクエリのパフォーマンスが低下することがなくなります。パフォーマンスが重要なクエリの場合、パフォーマンスの改善が達成できたかどうかをテストする必要があります。
- ・ プロダクションの再開 – システムでプロダクションを実行している場合は、“プロダクションの管理”の“プロダクションの開始と停止”の章にある“プロダクションの開始”のセクションの手順に従ってください。
- ・ スタジオ・クライアントのアップグレード – クライアント上のスタジオ・バージョンは、接続先の InterSystems IRIS サーバ・バージョンと同じかそれ以降である必要があります。
- ・ Web ゲートウェイのアップグレード – Web ゲートウェイがアップグレードしている InterSystems IRIS サーバとは別個のマシン上にある場合、その別個のマシン上の Web ゲートウェイもアップグレードする必要があります。これを行うには、この章の“[Windows でのアップグレード手順](#)”に従うか、“[Windows での自動インストール](#)”の説明に従って、ADDLOCAL プロパティを使用してサイレント・インストールを実行します。
- ・ Web サーバ・ファイルの更新 – アップグレード後、サイトで確立されている手法を使用して、これらを導入する必要があります。

6.2.1 ネームスペースのコンパイル方法

アップグレード後、各ネームスペース内のコードをコンパイルして、新規バージョンの InterSystems IRIS で実行されるようにする必要があります。コンパイラでエラーが検出された場合、コンパイラですべての依存関係を解決するために、1 回または複数回のリコンパイルが必要になることがあります。

追加の環境でアップグレードを実行する必要がある場合は、コードが正常にコンパイルされた後、更新済みのクラスまたはルーチンをエクスポートできます。今後アップグレードを行う際にこれらのクラスまたはルーチンをインポートすると、ダウンタイムを最小限に抑えて、迅速にコードを更新することができます。

注釈 アップグレード・プロセスの一部としてマニフェストを使用する場合は、マニフェストからコードをコンパイルすることができます。手順については、このドキュメントの付録 [“インストール・マニフェストの作成および使用”](#) の [“アップグレード後のタスクの実行”](#) のセクションを参照してください。

6.2.1.1 クラスのコンパイル

ターミナルからすべてのネームスペース内のクラスをコンパイルするには：

```
do $system.OBJ.CompileAllNamespaces("u")
```

ターミナルから 1 つのネームスペース内のクラスをコンパイルするには：

```
set $namespace = "<namespace>"
do $system.OBJ.CompileAll("u")
```

ターミナルから %SYS ネームスペース内のクラスをコンパイルするには：

```
set $namespace = "%SYS"
do $system.OBJ.CompileAll("u")
```

注釈 ネームスペースに、マップされたクラスが含まれる場合は、CompileAllNamespaces() または CompileAll() の呼び出しに /mapped 修飾子を含めます。

```
do $system.OBJ.CompileAllNamespaces("u /mapped")
do $system.OBJ.CompileAll("u /mapped")
```

6.2.1.2 クラス・コンパイラ・バージョン・ユーティリティ

ネームスペース内のクラスがどのクラス・コンパイラ・バージョンを使用してコンパイルされているかをお客様が判断するのを支援するため、インターシステムズでは 2 つの支援を提供しています。

- ・ メソッド – \$System.OBJ.CompileInfoClass(<classname>)

このメソッドは、この <classname> のコンパイルに使用されたクラス・コンパイラのバージョンとこのクラスがコンパイルされた日時を返します。

- ・ クエリ – \$System.OBJ.CompileInfo(<sortby>)

このクエリは、すべてのクラス、各クラスをコンパイルするのに使用されたコンパイラのバージョン、各クラスがコンパイルされた日時が含まれる、現在のネームスペースのレポートを生成します。最初の引数 <sortby> は、以下の値を取ることができます。

- 0 – クラスがコンパイルされた時間
- 1 – クラス名
- 2 – クラスがコンパイルされた InterSystems IRIS のバージョン

6.2.1.3 ルーチンのコンパイル

ターミナルからすべてのネームスペース内のルーチンをコンパイルするには：

```
do ##Class(%Routine).CompileAllNamespaces()
```

ターミナルから 1 つのネームスペース内のルーチンをコンパイルするには：

```
set $namespace = "<namespace>"
do ##Class(%Routine).CompileAll()
```

ターミナルから %SYS ネームスペース内のルーチンをコンパイルするには：

```
set $namespace = "%SYS"  
do ##Class(%Routine).CompileAll()
```