



インストール・ガイド

Version 2023.1
2024-01-02

インストール・ガイド

InterSystems IRIS Data Platform Version 2023.1 2024-01-02

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働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 InterSystems IRIS のインストール準備	1
1.1 インストール計画の考慮事項	1
1.1.1 インストール・ディレクトリ	2
1.1.2 セットアップ・タイプ	2
1.1.3 文字幅設定	4
1.1.4 ポート番号	4
1.1.5 セキュリティ設定	4
1.1.6 VS Code - ObjectScript 開発環境のインストール	4
1.2 メモリの計画および管理	5
1.2.1 ディスク容量の最小要件	5
1.2.2 追加のメモリ・リソース	5
1.3 プラットフォーム固有の準備	5
1.3.1 サポートされるプラットフォームとコンポーネント	5
1.3.2 ユーザ・プロセスの最大数に関する推奨事項	6
1.3.3 AIX® プラットフォームの問題	6
1.3.4 Red Hat Linux プラットフォームの問題	10
1.3.5 SUSE Linux プラットフォームの問題	12
1.3.6 Ubuntu プラットフォームの問題	13
2 Microsoft Windows への InterSystems IRIS のインストール	15
2.1 インストール前	15
2.1.1 Windows ユーザ・アカウント	15
2.1.2 構成方法の選択	16
2.2 InterSystems IRIS のインストール手順	16
2.2.1 開発コンポーネントまたはサーバ・コンポーネントのみのインストール	17
2.2.2 クライアント・コンポーネントのみのインストール	18
2.2.3 Web ゲートウェイのみのインストール	19
2.2.4 カスタム・インストールの実行	20
2.3 自動インストール手順	21
2.3.1 自動インストールの実行	21
2.3.2 自動アップグレードまたは再インストールの実行	22
2.3.3 自動削除の実行	23
2.3.4 コマンド行のリファレンス	23
2.4 インストール後のタスク	30
2.5 特別な考慮事項	31
2.5.1 複数の InterSystems IRIS インスタンスの構成	31
2.5.2 InterSystems IRIS の言語の変更	31
2.5.3 InterSystems IRIS の再インストールまたはアンインストール	32
2.5.4 書き込みキャッシュ・バッファ	32
3 InterSystems IRIS の UNIX®, Linux, および macOS へのインストール	33
3.1 インストール前	33
3.1.1 インストール・キットの解凍	33
3.1.2 所有者およびグループの決定	34
3.1.3 必要な依存関係のインストール	34
3.1.4 構成方法の選択	35
3.2 InterSystems IRIS のインストール	35
3.2.1 InterSystems IRIS の標準インストール手順	35

3.2.2 クライアントのみの InterSystems IRIS のインストール手順	39
3.2.3 非 root ユーザとしてのインストール	40
3.3 InterSystems IRIS の自動インストール	41
3.3.1 自動インストール・パラメータ	42
3.3.2 自動インストール・パッケージ	47
3.3.3 自動インストールの例	48
3.4 インストール後のタスク	48
3.4.1 InterSystems IRIS の起動	49
3.4.2 開発環境のインストール	49
3.5 特別な考慮事項	50
3.5.1 複数の InterSystems IRIS インスタンス	50
3.5.2 InterSystems IRIS のアンインストール	50
3.5.3 macOS での多数の同時プロセスの調整	51
4 以前のバージョンからのアップグレード	53
4.1 互換性の目標	53
4.2 アップグレードの前に	53
4.3 インスタンスのアップグレード	55
4.3.1 Windows でのアップグレード手順	55
4.3.2 UNIX®, Linux、および macOS でのアップグレード手順	56
4.4 ミラーのアップグレード	57
4.4.1 ミラーのアップグレード手順の選択	58
4.4.2 アップグレード時のミラー・メンバの追加	59
4.4.3 ミラーのアップグレードの用語	59
4.4.4 ミラーのアップグレード手順	60
4.5 ECP 構成のアップグレード	64
4.6 アップグレード後のタスク	65
4.6.1 メンテナンス・リリースのアップグレード後のタスク	65
4.6.2 メジャー・バージョンのインストール後のタスク	65
付録A: インストール・マニフェストの作成および使用	69
A.1 インストール・マニフェストの概要	69
A.2 インストール・マニフェストの作成	69
A.2.1 マニフェスト・クラス定義	69
A.2.2 マニフェストの一般的なオプション	70
A.2.3 <Manifest> タグ内の変数	71
A.2.4 <Manifest> タグのリスト	72
A.3 マニフェストの使用	83
A.3.1 手動で実行する場合	84
A.3.2 インストール時に実行する場合	84
A.4 インストール・マニフェストの例	85
A.4.1 空のテンプレート	85
A.4.2 ネームスペースの作成	85
A.4.3 アップグレード後のコンパイル	86
付録B: InterSystems IRIS ディストリビューションへの UNIX® インストール・パッケージの追加	89
B.1 チュートリアル	89
B.2 parameters.isc ファイルの内容	92

テーブル一覧

テーブル 1-1: 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	3
テーブル 2-1: コマンド行のプロパティ	24
テーブル 2-2: カスタム・インストール可能な機能	29
テーブル 3-1: 自動インストール・パラメータ	42
テーブル II-1: InterSystems IRIS のインストール・パラメータ・ファイルの変数	93

1

InterSystems IRIS のインストール準備

InterSystems IRIS® は複数の導入方法をサポートしており、さまざまなプラットフォーム上で実行できます。

“インストール・ガイド”では、インストール・キットから InterSystems IRIS をインストールするプロセスについて説明します。InterSystems IRIS は、“[コンテナ内でのインターシステムズ製品の実行](#)”で説明されているように、コンテナにも導入でき、複数の自動導入方法が用意されています。これについては、“InterSystems IRIS の基礎：IDE の接続”の“[InterSystems IRIS の導入](#)”を参照してください。これらのいずれかの方法を使用する場合は、後述するドキュメントを参照してください。

インストールの前に、InterSystems IRIS がお使いのサーバまたは開発プラットフォームをサポートしていることを確認してください。各リリースでサポートされるテクノロジーが記載されている現在のリリース用のドキュメント“InterSystems IRIS Data Platform サポート対象プラットフォーム”を確認してください。

インストールの前に、自由に使用できるシステム・リソースについて検討し、開発および実行するアプリケーションにそれらのリソースが十分であることを確認し、それらを管理して、システムのパフォーマンスを最適化する必要があります。このプロセスの詳細は、“[システム・リソースの計画と管理](#)”を参照してください。

1.1 インストール計画の考慮事項

InterSystems IRIS の新規インストールは、何とおりかの方法で構成できます。インストールを実行すると、インストールする構成を決定するために、以下の情報を入力するよう求められます。

- ・ [インストール・ディレクトリ](#)
- ・ [セットアップ・タイプ](#)
- ・ [文字幅設定](#)
- ・ [ポート番号 \(オプション\)](#)
- ・ [セキュリティ設定](#)

以下のセクションは、InterSystems IRIS をインストールする前または後に実行できます。

- ・ [VS Code - ObjectScript 開発環境のインストール](#)

1.1.1 インストール・ディレクトリ

インストール・ディレクトリは、InterSystems IRIS インスタンスをインストールするディレクトリです。このドキュメントでは、このディレクトリを `install-dir` と呼びます。InterSystems IRIS のインストール後にインストール・ディレクトリを変更することはできません。

`install-dir` として指定できるディレクトリには、いくつかの制限があります。ディレクトリは、シンボリック・リンクを含まない、完全に解決された物理パスである必要があります。ディレクトリ名には、US ASCII 文字セットの文字のみを使用でき、キャレット (^) は使用できません。また、InterSystems IRIS を以下のディレクトリにインストールすることはできません。

- ・ UNC (ローカルでない) パスのディレクトリ。
- ・ ドライブのルート・レベル (`C:\` など) にあるディレクトリ。
- ・ `%Program Files` ディレクトリ下の任意の場所にあるディレクトリ。

インストール時に指定しない場合、`install-dir` には既定値が使用されます。以下の表に示すように、この既定値は、プラットフォーム、インストール・タイプ、およびユーザの選択によって異なります。

プラットフォーム	インストール・タイプ	既定のディレクトリ
Windows	手動	インストール・ユーザが他に指定しない限り、 <code>C:\%InterSystems%\Iris</code> (複数インスタンスが存在する場合は、 <code>IrisN</code>)。
	自動	<code>INSTALLDIR</code> プロパティで他に指定されない限り、 <code>C:\%InterSystems%\Iris</code> (複数インスタンスが存在する場合は、 <code>IrisN</code>)。
UNIX®, Linux、macOS	手動	既定値なし。インストール・ユーザが指定する必要があります。 /home ディレクトリ、そのサブディレクトリ、または /usr/local/etc/irissys ディレクトリは選択しないでください。
	自動	既定値なし。 <code>ISC_PACKAGE_INSTALLDIR</code> パラメータが必要です。

1.1.2 セットアップ・タイプ

インストール時に、インストールする InterSystems IRIS コンポーネントを選択できます。オプションは以下のとおりです。

- ・ **[開発]** – InterSystems IRIS データベース・エンジン (User データベース、言語ゲートウェイ、サーバ監視ツール)、スタジオ (Windows のみ)、すべてのサポート対象言語バインディング、および xDBC (ODBC と JDBC) ドライバを含めます。このインスタンスをクライアントのタスクとサーバのタスクの両方を実行するために使用する場合は、このオプションを選択します。
- ・ **[サーバ]** – InterSystems IRIS データベース・エンジン (User データベース、言語ゲートウェイ、サーバ監視ツール) および Web ゲートウェイを含めます。InterSystems IRIS クライアントからアクセス可能な InterSystems IRIS データベース・サーバとしてこのインスタンスを使用する場合は、このオプションを選択します。
- ・ **[カスタム]** – インストールまたはアンインストールする特定のコンポーネントを指定できます。特定の InterSystems IRIS コンポーネントをインストールまたは削除する場合、このオプションを選択します。

Windows では、以下の 2 つの追加セットアップ・タイプから選択できます。

- ・ **[クライアント]** – スタジオ、ODBC ドライバ、JDBC ドライバ、および InterSystems IRIS アプリケーション開発コンポーネントを含めます。このコンピュータまたは別のコンピュータで、InterSystems IRIS データベース・サーバのクライアントとしてこのインスタンスを使用する場合は、このオプションを選択します。

- ・ [Web サーバ] – Web ゲートウェイ (IIS、Apache 2.0、Apache 2.2) を含めます。InterSystems IRIS の機能のうち、Web ゲートウェイに必要なもののみをインストールする場合は、このオプションを選択します。

以下のテーブルでは、それぞれのセットアップ・タイプでインストールされるコンポーネントを示しています。カスタム・インストールを実行する場合は、グループから個別のコンポーネントのみを含めることができます。

テーブル 1-1: 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

コンポーネント・グループ	コンポーネント	開発	サーバ	クライアント	Web
InterSystems IRIS データベース・エンジン (InterSystems IRIS サーバ)	サーバ監視ツール User データベース 言語ゲートウェイ エージェント・サービス (ISCAgent) Apache FOP (Formatting Objects Processor) InterSystems IntegratedML	X	X		
InterSystems IRIS ランチャー (Windows のみ)		X	X	X	
スタジオ (Windows のみ)		X		X	
xDBC	ODBC ドライバ Java Database Connectivity	X		X	
InterSystems IRIS アプリケーション開発	InterSystems IRIS 用の Java バインディング InterSystems IRIS 共有ライブラリ・サポート InterSystems IRIS 用の .NET バインディング スレッド・サーバ・ライブラリ その他のサンプル	X		X	
Web ゲートウェイ	CSP for IIS CSP for Apache 2.0.x CSP for Apache 2.2.x		X		X

1.1.2.1 既存の CSP ゲートウェイ

Web ゲートウェイをインストールする際に、お使いのシステムに CSP ゲートウェイが既にインストールされている場合は、インストーラにより CSP ゲートウェイが Web ゲートウェイに自動的にアップグレードされます。ただし、インストーラは **CSP.ini** ファイルの新しいコピーを作成します。CSP ゲートウェイの **CSP.ini** ファイルを保持するには、以下の手順に従います。

1. install-dir/CSP/bin にある **CSP.ini** ファイルのバックアップを作成します。

- Web ゲートウェイ・インストーラを実行します。
- CSP.ini のバックアップをリストアします。

注釈 Web ゲートウェイをインストールしても、古い接続は閉じられません。これらの古い接続を削除するには、[\[システムステータス\]](#) ページで手動で閉じる必要があります。

1.1.3 文字幅設定

インストールには、8 ビットまたは Unicode (16 ビット) の文字幅を選択する必要があります。8 ビット・インスタンスは 16 ビット形式のデータを処理できませんが、Unicode インスタンスは 8 ビット・データと 16 ビット・データの両方を処理できます。

インスタンスで、Unicode 文字のみを使用する言語 (日本語など) のデータを格納および処理する必要がある場合は、Unicode を選択してください。インスタンスで使用するロケールに設定されている基本文字セットが、Latin-1 文字セット ISO8859-1 に基づいている場合は、8 ビットを選択できますが、以下の点に留意してください。

- 8 ビットを選択した場合、インスタンスによって格納されるデータは、同じ文字セットに基づく 8 ビット・ロケールにのみ移植できます。
- Unicode を選択した場合、インスタンスによって格納されるデータは 8 ビット・インスタンスに移植できません。

1.1.4 ポート番号

標準インストールでは、InterSystems IRIS インスタンスに対して以下のポート番号が設定されます。

- スーパーサーバのポート番号 – 1972。取得されている場合は、51773 またはこれ以降の使用可能な最初のポート番号。
- Web サーバのポート番号 – 52773、またはこれ以降の使用可能な最初のポート番号

新規インストール時に**カスタム**・インストールを実行することで、別のポート番号を割り当てることができます (“[セットアップ・タイプ](#)” を参照)。65535 よりも大きなポート番号は入力できません。

InterSystems IRIS のインストール済みインスタンスでのポート番号の設定は、“システム管理ガイド” の “InterSystems IRIS 複数インスタンスの使用法” の章にある “[ポート番号の設定](#)” を参照してください。

1.1.5 セキュリティ設定

InterSystems IRIS インストールには、**[最小]**、**[通常]**、または **[ロック・ダウン]** の 3 つの初期セキュリティ構成があります。**[最小]** は、InterSystems IRIS インストールにのみ使用できます。**[通常]** または **[ロック・ダウン]** を選択した場合、インストーラにより、パスワードやアカウント情報などの追加情報の入力を求められます。

詳細は、“[インターシステムズの初期セキュリティ設定](#)” を参照してください。

1.1.6 VS Code – ObjectScript 開発環境のインストール

Visual Studio Code (VS Code) は、Microsoft によって開発された、Windows、Linux、および macOS 向けの無料のソース・コード・エディタです。Visual Studio Code の InterSystems ObjectScript 拡張機能により、VS Code を使用して InterSystems IRIS サーバに接続し、ObjectScript でコードを開発できます。VS Code – ObjectScript の代わりになるのが**スタジオ**で、これは Windows ベースのオペレーティング・システムに InterSystems IRIS と一緒にインストールされます。

VS Code – ObjectScript のダウンロードと使用の詳細は、<https://intersystems-community.github.io/vscode-objectscript/> を参照してください。

1.2 メモリの計画および管理

メモリの管理は、InterSystems IRIS 導入環境のパフォーマンスを最大限に高めるための非常に重要な要素です。以下の、計画上の考慮事項を確認してください。

- ・ [ディスク容量の最小要件](#)
- ・ [追加のメモリ・リソース](#)

1.2.1 ディスク容量の最小要件

インストールを実行するには、コンピュータまたはネットワーク上でインストール・キットを利用できる必要があります。InterSystems IRIS をインストールする前に、1.5 GB 以上のディスクの空き容量があることをお勧めします。インストール前に、インストール手順によってディスクの空き容量が十分にあることが確認されます。

1.2.2 追加のメモリ・リソース

InterSystems IRIS とアプリケーションのパフォーマンスと可用性を最大限に高めるには、システム・メモリを正しくサイジングすることが重要です。InterSystems IRIS のメモリ使用量については、“[メモリの計画](#)”を参照してください。ここにはシステムに十分なメモリがあるかどうかを判断できるように、初期メモリ要件を計算するためのガイドラインが示されています。

インストール後は、InterSystems IRIS がどのようにメモリを割り当てるかを監視することが、パフォーマンスの重要な要因になります。InterSystems IRIS がメモリを効率的に割り当てていない場合は、ユーザがプロセス、ルーチン・キャッシュ、データベース・キャッシュ、および一般ヒープのメモリ設定を調整できます。これらのメモリ関連タスクを実行する方法は、“[メモリ使用量の確認と監視](#)” および “[メモリと開始設定](#)” を参照してください。

1.3 プラットフォーム固有の準備

以下のセクションでは、プラットフォーム固有の考慮事項を示します。目的のインストール・プラットフォームに該当するセクションを参照してください。

- ・ [サポートされるプラットフォームとコンポーネント](#)
- ・ [ユーザ・プロセスの最大数に関する推奨事項](#)
- ・ [AIX® プラットフォームの問題](#)
- ・ [Red Hat Linux プラットフォームの問題](#)
- ・ [SUSE Linux プラットフォームの問題](#)
- ・ [Ubuntu プラットフォームの問題](#)

1.3.1 サポートされるプラットフォームとコンポーネント

サポート対象オペレーティング・システムおよび Web サーバなど、InterSystems IRIS のリリース・バージョンでサポートされるテクノロジーについては、リリースのオンライン・ドキュメント “[インターシステムズのサポート対象プラットフォーム](#)” の “[サポート対象テクノロジー](#)” の章を参照してください。

最適なジャーナル・パフォーマンスを実現し、システムがクラッシュしたときにジャーナル・データの整合性を確保するために、インターシステムズはジャーナル・ファイルに適用するさまざまなファイル・システムとマウント・オプションを推奨し

ています。プラットフォームごとの詳細は、“InterSystems IRIS データ整合性ガイド”の“ジャーナリング”の章にある“[UNIX® ファイル・システムの推奨事項](#)”のセクションを参照してください。

注釈 管理ポータルが確実に機能するようにするために、InterSystems IRIS では、インスタンスごとに CSP ページ用のプライベート Web サーバとプライベート Web ゲートウェイがインストールされます。プライベート Web サーバでは、以下のことが可能になります。

- ・ 管理ポータルをすぐに使用できます。
- ・ 開発環境に、すぐに使用できるテスト機能が提供されます。

プライベート Web サーバでは、この他の用途はサポートされていません。プライベート Web サーバを、HTTP ベースのアプリケーション (CSP や SOAP over HTTP/HTTPS など) の導入に使用しないでください。代わりに、サポートされているいずれかの Web サーバをインストールおよび導入してください。

プライベート Web サーバの構成は、アップグレードしても維持されます。[WebServer](#) CPF パラメータを使用して、プライベート Web サーバを無効にできます。

Windows では、プライベート Web サーバの Windows サービス名は“Web Server for instance name”です。この Web サーバは、install-dir¥httpd ディレクトリにインストールされます。対応する InterSystems IRIS インスタンスをアンインストールすると、その Web サーバもアンインストールされます。

詳細は、“[プライベート Web サーバの使用または置換](#)”を参照してください。

1.3.2 ユーザ・プロセスの最大数に関する推奨事項

指定のユーザのすべての InterSystems IRIS プロセスおよびその他の既定のプロセスを実行するうえで十分な最大プロセス数を maximum user processes に設定します。

1.3.3 AIX® プラットフォームの問題

一部の AIX® パラメータの既定の設定は、パフォーマンスに悪影響を及ぼす可能性があります。設定および推奨事項については以下を参照してください。

- ・ [I/O ペーシング・パラメータ](#)
- ・ [ファイル・システムのマウント・オプション](#)
- ・ [メモリ管理パラメータ](#)
- ・ [AIX® の調整可能パラメータ](#)
- ・ [必須の C/C++ ランタイム・ライブラリ](#)
- ・ [InterSystems IRIS 共有ライブラリ・サポートに対する共有ライブラリ環境変数](#)
- ・ [Raw Ethernet の使用法](#)

1.3.3.1 I/O ペーシング・パラメータ

AIX® は、InterSystems IRIS ライト・デーモンの動作を阻害する可能性のある I/O ペーシング・アルゴリズムを実装しています。AIX® 5.2 および AIX® 5.3 では、HACMP クラスタを使用すると I/O ペーシングが自動的に有効になります。一方、AIX® 6.1 以降では、すべてのシステムで I/O ペーシングが有効となり、既定の最高水準点は以前のバージョンよりも大きくなっています。

ライト・デーモンの速度が遅い場合やライト・デーモンが停止する場合は、この最高水準点の調整が必要なことがあります。詳細は、IBM AIX ドキュメントの“[ディスク入出力ペーシング](#)”を参照してください。

重要

IBM AIX® 6.1 以降では、最高水準点の調整は不要です。

システムへの影響について疑問点がある場合は、設定を変更する前に [インターシステムズのサポート窓口](#) または AIX® サプライヤまでお問い合わせください。これらの推奨事項は、JFS と Enhanced JFS (JFS2) 両方のファイル・システムに適用されます。

1.3.3.2 ファイル・システムのマウント・オプション

さまざまなマウント・オプションを使用することで作業負荷のパフォーマンスを向上させられる可能性があります。

注釈 ファイル・システムによるキャッシュが効果を発揮する、InterSystems IRIS ではない負荷 (例えば、オペレーティング・システム・レベルのバックアップやファイル・コピー) は、cio マウント・オプションにより速度が低下します。

ハード・シャットダウンやシステム・クラッシュの後の **IRIS.WIJ** ファイルを使用したリカバリ速度を向上させるには、**IRIS.WIJ** ファイルを扱うファイル・システムで、ファイル・システム・バッファリングを伴うマウント・オプション (例えば、rw) を使用することをお勧めします。

mount オプションの詳細は、IBM AIX ドキュメントの "[mount コマンド](#)" を参照してください。

1.3.3.3 メモリ管理パラメータ

ファイル・システムの数およびファイル・システムに対するアクティビティの量によって、JFS または JFS2 のメモリ構造が制限され、それらのメモリ構造を使用する I/O 処理に遅延が生じる可能性があります。

こうした状況を監視するには、vmstat -vs コマンドを実行し、2 分経過してから、もう一度 vmstat -vs コマンドを実行します。次のような結果が出力されます。

```
# vmstat -vs
1310720 memory pages
1217707 lruable pages
144217 free pages
1 memory pools
106158 pinned pages
80.0 maxpin percentage
20.0 minperm percentage
80.0 maxperm percentage
62.8 numperm percentage
764830 file pages
0.0 compressed percentage
0 compressed pages
32.1 numclient percentage
80.0 maxclient percentage
392036 client pages
0 remote pageouts scheduled
0 pending disk I/Os blocked with no pbuf
5060 paging space I/Os blocked with no psbuf
5512714 filesystem I/Os blocked with no fsbuf
194775 client filesystem I/Os blocked with no fsbuf
0 external pager filesystem I/Os blocked with no fsbuf
```

以下のパラメータが増加している場合は、InterSystems IRIS のパフォーマンスを向上するため、それらの値を大きくしてください。

- ・ pending disk I/Os blocked with no pbuf
- ・ paging space I/Os blocked with no psbuf
- ・ filesystem I/Os blocked with no fsbuf
- ・ client filesystem I/Os blocked with no fsbuf
- ・ external pager filesystem I/Os blocked with no fsbuf

これらのパラメータを既定値より大きくする場合は、以下の手順に従ってください。

1. 現在の値を 50% 大きくします。

2. vmstat 出力を確認します。
3. 2 分間隔で、vmstat を 2 回実行します。
4. フィールドがまだ増大している場合は、もう一度、パラメータ値を同じ割合だけ大きくして、上記の手順に従います。vmstat を 2 回実行したとき、フィールドが増大しなくなるまで繰り返してください。

重要 I/O パターンは時間の経過と共に変化する可能性があるため、現在値と再起動値の両方を変更し、vmstat 出力を定期的 (時間単位、日単位、または週単位) に確認します。

詳細は、IBM の以下の Web ページを参照してください。

- ・ vmstat によって報告される各フィールドの詳細は、IBM AIX ドキュメントの “[vmstat コマンド](#)” を参照してください。
- ・ これらのパラメータ値を大きくする方法については、IBM AIX ドキュメントの “[VMM ページ置換のチューニング](#)” を参照してください。
- ・ I/O 調整可能パラメータの管理の詳細は、IBM AIX ドキュメントの “[ioo コマンド](#)” を参照してください。

1.3.3.4 AIX® の調整可能パラメータ

AIX® プロセス間通信の調整可能なパラメータ

以下のテーブルは、IBM pSeries AIX® 5.2 オペレーティング・システムの調整可能なパラメータの一覧です。以下に一覧表示されているパラメータは調整の必要がありません。これらのパラメータは、必要に応じて、カーネルによって動的に調整されます。詳細は、AIX ドキュメントの “[プロセス間通信のチューナブル・パラメーター](#)” を参照してください。

Parameter (パラメータ)	目的	動的値
msgmax	最大メッセージ・サイズを指定します	最大値 4 MB
msgmnb	キューでの最大バイト数を指定します	最大値 4 MB
msgmni	メッセージ・キュー ID の最大数を指定します	最大値 4096
msgmnmx	キューごとの最大メッセージ数を指定します	最大値 524288
semaem	終了の調整最大値を指定します	最大値 16384
semmni	セマフォ ID の最大数を指定します	最大値 4096
semmsl	ID ごとの最大セマフォ数を指定します	最大値 65535
semopm	semop() 呼び出しごとの最大操作数を指定します	最大値 1024
semume	プロセスごとの最大 undo エントリ数を指定します	最大値 1024
semvmx	セマフォの最大値を指定します	最大値 32767
shmmax	共有メモリ・セグメントの最大サイズを指定します	32 ビット・プロセスでは最大値 256 MB、64 ビットでは最大値 0x80000000u
shmmni	共有メモリ・セグメントの最小サイズを指定します	最小値 1

Parameter (パラメータ)	目的	動的値
shmmni	共有メモリ ID の最大数を指定します	最大値 4096

maxuproc

maxuproc は、root 以外の単独ユーザが開始可能なプロセスの最大数を指定する調整可能パラメータで、このサブセクションの説明に従って調整できます。

このパラメータの設定が低すぎると、プロセスの開始を試みるユーザがさらに増えた場合に、CSP ページの消失、バックグラウンド・タスクの失敗など、オペレーティング・システムのさまざまなコンポーネントで障害が起こる可能性があります。したがって、maxuproc パラメータは、root ユーザ以外（インタラクティブ・ユーザ、Web サーバ・プロセスなど、プロセスを開始可能なものすべて）による開始が見込まれるプロセスの最大数よりも大きく設定する必要があります。

注釈 この値を過度に高く設定しないでください。この値には不必要に新規プロセスを作成し続ける暴走アプリケーションからサーバを保護する目的があります。しかし、設定が低すぎると不明な問題の原因になります。

maxuproc を予想される最大プロセス数の倍にして、エラーを起こさないだけの余裕をとりつつ、暴走プロセスから保護することをお勧めします。例えば、システムに 1000 人のインタラクティブ・ユーザがいて、頻繁に 500 個のバックグラウンド・プロセスを実行する場合、この値は最低でも 3000 を選択するとよいでしょう。

maxuproc 値は、コマンド行もしくは smit/smitty 管理者用ユーティリティのいずれかより、以下のように双方共に root ユーザとして検証および変更ができます。

- ・ コマンド行より、現在の設定を表示します。

```
# lsattr -E -l sys0 -a maxuproc
```

次に、以下のように値を変更します。

```
# chdev -l sys0 -a maxuproc=NNNNNN
```

ここで、NNNNNN は新規の値です。

- ・ 管理者用ユーティリティ smit (または smitty) で、[システム環境]→[オペレーティングシステム特性の変更/表示]→[ユーザあたりの許容プロセス最大数]を選択します。

maxuproc の値を増やすと、変更は瞬時に有効となります。maxuproc の値を減らすと、変更は次のシステムの再起動まで有効になりません。どちらの場合も、変更はシステムを再起動しても存続します。

1.3.3.5 必須の C/C++ ランタイム・ライブラリ

InterSystems IRIS をインストールする前に、必須の C/C++ ランタイム・ライブラリが IBM AIX® システムにインストールされていることを確認する必要があります。

InterSystems IRIS for AIX は、IBM XL C/C++ for AIX 16.1.0 コンパイラを使用してコンパイルされます。InterSystems IRIS のインストール先のシステムに、既にインストールされているランタイムに対応するバージョンがない場合は、インストールする必要があります。ランタイムは、IBM Support Web サイトの ["IBM XL C/C++ Runtime for AIX 16.1"](#) からダウンロードできます。

1.3.3.6 InterSystems IRIS 共有ライブラリ・サポートに対する共有ライブラリ環境変数

InterSystems IRIS 共有ライブラリ・サポートには、インストールされたすべての C リンカを参照するバッチ・ファイルが含まれています。

標準 UNIX® C ライブラリまたは独自の C ライブラリを LIBPATH 環境変数で定義した環境が必要です。

このような定義がない環境では、標準 UNIX® C ライブラリのパスである /usr/lib と /lib を LIBPATH に追加します。

1.3.3.7 Raw Ethernet の使用法

Raw Ethernet を使用するには、IBM AIX® マシンに DLPI (データ・リンク・プロバイダ・インタフェース) パッケージがインストールされている必要があります。マシンに DLPI パッケージがない場合、IBM プロバイダから取得し、以下の手順に従って DLPI デバイスを作成してください。

1. root としてログインします。
2. `/etc/pse.conf` ファイルの PSE ドライバ・セクションで、DLPI ドライバを参照している 4 行をアンコメントします。
3. ファイルを保存します。
4. コンピュータを再起動します。

DLPI デバイスがインストールされていない場合、`%SYSTEM.INetInfo` クラスの `EthernetAddress()` メソッドは、イーサネット・デバイスの情報ではなく、NULL 文字列を返します。

1.3.4 Red Hat Linux プラットフォームの問題

このトピックには、以下の調整に関する情報が含まれます。

- ・ [共有メモリ制限](#)
- ・ [ロックインされたメモリ](#)
- ・ [多数の同時プロセスの調整](#)
- ・ [ダーティ・ページのクリーンアップ](#)
- ・ [Kerberos の使用](#)

1.3.4.1 共有メモリ制限

Linux プラットフォームでの、既定の共有メモリ制限 (`shmmax`) は、32 MB です。この値は InterSystems IRIS では小さすぎますが、`proc` ファイル・システムで変更することができます。その場合、再起動は必要ありません。新しいメモリ制限は、Red Hat Linux システムを再起動するまで維持されます。

例えば 128 MB で設定したい場合は、以下のコマンドを入力します。

```
$ echo 134217728 >/proc/sys/kernel/shmmax
```

上記のコマンドは、起動スクリプトに入力することができます。

また、`sysctl(8)` が利用できる場合はこれを利用して、パラメータを恒久的に設定することもできます。以下のような行を `/etc/sysctl.conf` に追加します。

```
kernel.shmmax = 134217728
```

このファイルは通常、スタートアップ時に処理されますが、`sysctl` は後で明示的に呼び出すこともできます。

重要 1 台のマシン上で 複数の InterSystems IRIS インスタンスを実行する場合、`msgmni` パラメータの設定値が小さすぎる可能性があります。この値には、システムで同時に実行する InterSystems IRIS インスタンス数の 3 倍に相当する値を設定してください。

他のパラメータは、InterSystems IRIS アプリケーションに対して十分にサイズ調整されています。他のパラメータの値を表示するには、`/usr/src/linux/include/asm-xxx/shmparam.h` と `/usr/src/linux/include/linux/sem.h` の 2 つのファイルを調べてください。

詳細は、“Red Hat Enterprise Linux 4: リファレンス・ガイド” の “[proc ファイル・システム](#)” の章を参照してください。

1.3.4.2 ロックインされたメモリ

Linux プラットフォームでは、共有メモリがヒュージ・ページに割り当てられている場合、ヒュージ・ページはメモリ内で自動的にロックされます（それ以上の操作は必要ありません）。ヒュージ・ページの割り当ての詳細は、この章の“[Linux でのヒュージ・ページの構成](#)”を参照してください。

ヒュージ・ページを使用しない場合、共有メモリ・セグメントをメモリにロックしてページングを防ぐように InterSystems IRIS を構成できます。これについては、“構成パラメータ・ファイル・リファレンス”の“memlock”エントリの [LockSharedMemory](#) セクションで説明されています。

それ以外の場合は、メモリ内にロックされる可能性のある最大サイズを増やす必要があります。既定値は 32 KB です。現在の値を表示するには、ulimit コマンドを使用します。

例えば、現在の制限値をすべて表示するには、以下のようにします。

```
bash$ ulimit -a
core file size (blocks, -c) unlimited
data seg size (KBytes, -d) unlimited
file size (blocks, -f) unlimited
pending signals (-i) 1024
max locked memory (KBytes, -l) 32 <----- THIS ONE
max memory size (KBytes, -m) unlimited
open files (-n) 1024
pipe size (512 bytes, -p) 8
POSIX message queues (bytes, -q) 819200
stack size (KBytes, -s) 10240
cpu time (seconds, -t) unlimited
max user processes (-u) 49000
virtual memory (KBytes, -v) unlimited
file locks (-x) unlimited
```

max-locked memory のみ表示するには、-l オプションを使用します。

```
bash$ ulimit -l
32
```

権限を持っていれば、ulimit コマンドを使用して直接値を変更できますが、`/etc/security/limits.conf` ファイルで memlock パラメータを更新することをお勧めします。memlock の制限値が小さすぎる場合、Linux から ENOMEM () エラーが表示されますが、これでは原因が明らかになりません。実際にメモリは割り当てられています。エラーを起こしているのはロックなのです。

詳細は、“構成パラメータ・ファイル・リファレンス”の“[memlock](#)”を参照してください。

1.3.4.3 多数の同時プロセスの調整

多数のプロセスまたは Telnet ログインを必要とするシステムを実行する場合は、以下の調整を行う必要があります。

1. `/etc/xinetd.d/telnet` ファイルに、以下の行を追加します。

```
instances = unlimited
```

2. `/etc/xinetd.conf` ファイルで、インスタンスの設定を以下のように追加または変更します。

```
instances = unlimited
```

3. これらの変更が完了したら、以下のコマンドを使用して xinetd を再起動します。

```
# service xinetd restart
```

4. 既定の pty (擬似ターミナル接続) 制限は 4096 です。それでも足りない場合は、`/etc/sysctl.conf` ファイルに最大 pty 行を追加するか変更します。以下はその例です。

```
kernel.pty.max=10000
```

1.3.4.4 ダーティ・ページのクリーンアップ

(例えば 8 GB 以上の) 大規模なメモリ・システムでは、フラット・ファイルを何度も書き込む場合 (例えば InterSystems IRIS のバックアップやファイル・コピーなど)、`proc/sys/vm/` に格納されている以下のパラメータを調整することにより、パフォーマンスを向上させることができます。

- **dirty_background_ratio** – `pdflush` が書き込みを開始するまでに、ダーティ・ページで満たすことのできるアクティブの最大割合。このパラメータは 5 に設定することをお勧めします。
- **dirty_ratio** – より多くの書き込みが許可される代わりに、タイム・スライス間のダーティ・バッファを書き込むプロセスが強制されるまでに、ダーティ・ページで満たすことのできる合計メモリの最大割合。このパラメータは 10 に設定することをお勧めします。

これらの変数は、`/etc/sysctl.conf` ファイルに以下を加えることで設定できます。

```
vm.dirty_background_ratio=5
vm.dirty_ratio=10
```

これらの変更は、Linux `pdflush` デーモンに対して、大量の更新バーストでストレージが溢れる可能性のある大量の更新をキューに入れるより頻繁に、ダーティ・ページへの書き出しを強制するものです。

1.3.4.5 Kerberos の使用

Red Hat Linux プラットフォームで Kerberos を使用するには、`krb5-libs` パッケージに加えて `krb5-devel` パッケージもインストールする必要があります。`krb5-devel` をインストールすると、Kerberos を使用するために必要なシンボリック・リンクが設定されます。このパッケージは、開発環境だけでなく、実稼働環境にも必要です。これらのコンポーネントの詳細は、[Red Hat Network](#) Web サイトを参照してください。

1.3.5 SUSE Linux プラットフォームの問題

このトピックには、以下の調整に関する情報が含まれます。

- [共有メモリ制限](#)
- [ロックインされたメモリ](#)
- [Kerberos の使用](#)

1.3.5.1 共有メモリ制限

SUSE Linux プラットフォームでの、既定の共有メモリ制限 (`shhmax` と `shmall`) は InterSystems IRIS にとって小さすぎます。この制限は、再起動しなくても `proc` ファイル システム内で変更できます。

InterSystems IRIS はデータベース・バッファ、グローバル・バッファ、ルーチン・バッファ、およびライセンス使用に共有メモリを使用します。InterSystems IRIS 専用にマシンを使用している場合は、共有メモリをメモリの総容量の約半分に設定することをお勧めします。詳細は、“システム・リソースの計画と管理” の “[メモリの計画](#)”、および “システム管理ガイド” の “[メモリと開始設定](#)” と “[ライセンスの機能範囲と使用の決定](#)” を参照してください。

注釈 共有メモリ制限を変更する推奨事項は、SUSE Linux 64 ビット・システムには適用されません。

例えば 512 MB で設定したい場合は、以下のコマンドを入力します。

```
#sets shmall and shmmax shared memory
echo 536870912 >/proc/sys/kernel/shmall      #Sets shmall to 512 MB
echo 536870912 >/proc/sys/kernel/shmmax      #Sets shmmax to 512 MB
```

上記のコマンドは、スタートアップ時のスクリプトで入力できます。SUSE Linux 製品のドキュメントでは、`/etc/init.d/boot.local` スクリプト・ファイル内にコマンドを入れることが推奨されています。

また、`/etc/profile.local` というファイルを変更することで、システム・メモリ・ユーザ制限に対する設定を変更できます。以下のような行を追加します。

```
#sets user limits (ulimit) for system memory resources
ulimit -v 512000      #set virtual (swap) memory to 512 MB
ulimit -m 512000      #set physical memory to 512 MB
```

この同じファイルで、以下のような行を追加することで、PATH パラメータと CLASSPATH パラメータの値を永続的に変更できます。

```
#sets env values PATH and CLASSPATH
export PATH=$PATH:/usr/iris/bin:/path/to/j2sdk/bin:/.
export CLASSPATH=
$CLASSPATH:/iris/dev/java/lib/JDK18/interSystems-jdbc-3.0.0.jar.
```

重要 システム・アップグレード時に変更が失われるリスクを回避するために、`/etc/profile` ファイルは変更しないでください。

1.3.5.2 ロックインされたメモリ

Linux プラットフォームでは、共有メモリがヒュージ・ページに割り当てられている場合、それらはメモリ内で自動的にロックされます（それ以上の操作は必要ありません）。ヒュージ・ページの割り当ての詳細は、“[Linux でのヒュージ・ページの構成](#)”を参照してください。

ヒュージ・ページを使用しない場合、共有メモリ・セグメントをメモリにロックしてページングを防ぐように InterSystems IRIS を構成できます。これについては、“構成パラメータ・ファイル・リファレンス”の“memlock”エントリの [LockSharedMemory](#) セクションで説明されています。

それ以外の場合は、メモリ内にロックされる可能性のある最大サイズを増やす必要があります。手順については、この章の“Red Hat Linux プラットフォームの問題”にある“[ロックインされたメモリ](#)”のセクションを参照してください。

1.3.5.3 Kerberos の使用

SUSE Linux プラットフォームで Kerberos を使用するには、`krb5-libs` パッケージに加えて `krb5-devel` パッケージもインストールする必要があります。`krb5-devel` をインストールすると、Kerberos を使用するために必要なシンボリック・リンクが設定されます。このパッケージは、開発環境だけでなく、実稼働環境にも必要です。これらのコンポーネントの詳細は、[SUSE documentation](#) の Web サイトを参照してください。

1.3.6 Ubuntu プラットフォームの問題

このトピックには、以下の調整に関する情報が含まれます。

- ・ [セマフォの削除の設定](#)

1.3.6.1 セマフォの削除の設定

状況によっては、インスタンス所有者が SSH などを使用して Ubuntu ホストに接続すると、OS によってインスタンスのセマフォが削除される場合があります。これを防止するには、`/etc/systemd/logind.conf` ファイルを編集して、`RemoveIPC=yes` (既定値) を `RemoveIPC=no` に変更します。

Ubuntu の新規バージョンに更新すると、`RemoveIPC` が既定値の `yes` に戻ることがあります。Ubuntu を更新したら、不要なセマフォの削除を防止するために、必ず `RemoveIPC` を変更してください。

2

Microsoft Windows への InterSystems IRIS のインストール

この章では、Microsoft Windows システムに InterSystems IRIS® Data Platform をインストールする方法を説明します。ここでは、ユーザが Windows のディレクトリ構造、ユーティリティおよびコマンドに精通していることを前提としています。

注釈 同一マシン上で複数のインストールを同時に実行することはできません。

2.1 インストール前

インストールを開始する前に、このドキュメントの“[InterSystems IRIS のインストールの準備](#)”の章に記載された、Windows に該当するすべての情報を確認しておいてください。さらに、以下のトピックを確認してください。

- ・ [Windows ユーザ・アカウント](#)
- ・ [構成方法の選択](#)

2.1.1 Windows ユーザ・アカウント

InterSystems IRIS のインストール時に、InterSystems IRIS サービス、InterSystems IRIS Controller for <instance-name> を実行する Windows アカウントを選択する必要があります。これには、以下の 2 つのオプションがあります。

- ・ 既定の SYSTEM アカウント (Windows ローカル・システム・アカウント)。これは、**最小セキュリティ・インストール**で使用されます。
- ・ 定義済みの Windows ユーザ・アカウント。

多くのインストールでは、既定の SYSTEM アカウントで InterSystems サービスを実行することが適切ですが、ファイルのアクセス許可やネットワーク・セキュリティのアクセスに関連する問題が発生する場合があります。InterSystems IRIS インスタンスのこれらの領域において潜在的な問題が予測される場合 (ネットワーク構成やセキュリティ設定に起因する問題など)、必要な特権やアクセス権を持っている InterSystems サービスの Windows アカウントを指定します (ドメイン管理者のアカウントなど)。

インストール後にサービス・アカウントを変更する手順は、“[InterSystems IRIS インスタンスへのアクセスの管理](#)”を参照してください。

重要 Kerberos を使用する場合は、InterSystems IRIS をインストールする前に、Windows アカウントを構成する必要があります。“[Kerberos を使用したセキュリティ環境の準備](#)” で説明したように、この目的のために特別に設定したアカウントを使用することをお勧めします。

注釈 定義済みの Windows ユーザ・アカウントで InterSystems IRIS を実行する場合は、そのアカウントに Windows の“メモリ内のページのロック”(SELockMemory) 特権を付与してください。詳細は、“InterSystems IRIS のインストール準備”の章の“[Windows でのラージ・ページの構成](#)”を参照してください。

2.1.2 構成方法の選択

特定の InterSystems IRIS 構成を記述するインストール・マニフェストを定義して、インストール・ファイルの実行時にそれを読み出すことができます。これは、自動導入を検討する場合に便利です。インストールの完了後に、設定を調整する必要が少なくなるためです。

インストール・マニフェストの詳細は、“[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。

2.2 InterSystems IRIS のインストール手順

このセクションでは、Windows コンピュータに InterSystems IRIS をインストールする手順について説明します。最初のステップの後の手順は、インストールするコンポーネントによって異なります。その違いについては、主要な手順の下にあるサブセクションで説明します。

InterSystems IRIS のインストールを実行するには、以下の手順で行います。

1. インストール・キットがユーザのコンピュータまたはネットワーク上で使用可能であることを確認します。
2. インストール・ファイルを実行します。その方法として、Windows エクスプローラでインストール・ファイルをダブルクリックするか、または以下のようにしてコマンド行でインストール・ファイルを実行します。

```
C:\Users\Public\Downloads\IRIS-2018.1.0.508.0-win_x64.exe
```

注釈 既定では、新しくインストールした InterSystems IRIS インスタンスがインストール直後に起動し、InterSystems IRIS ランチャーがシステム・トレイに配置されます。このように動作しないようにするには、コマンド行で **ISCSTARTIRIS** および **ISCSTARTLAUNCHER** をそれぞれ 0 に設定します。以下に例を示します。

```
C:\Users\Public\Downloads\IRIS-2018.1.0.508.0-win_x64.exe ISCSTARTIRIS=0 ISCSTARTLAUNCHER=0
```

この章の“自動インストール手順”のセクションの“[コマンド行のプロパティ](#)”のテーブルには、InterSystems IRIS Windows インストーラのすべてのプロパティの説明が記載されています。

3. InterSystems IRIS の設定は、Windows の **[Unicode 対応でないプログラムの言語]** の設定によって指定された言語で開始されます。セットアップでは、**[次へ]** をクリックすると次のダイアログ・ボックスに進み、**[戻る]** をクリックすると前のダイアログ・ボックスに戻り、**[キャンセル]** をクリックするとインストールが停止します。
4. システムに既存の InterSystems IRIS インスタンスがインストールされている場合は、**[インスタンス選択]** ダイアログ・ボックスにインストール・ディレクトリがリストされます。**[新規インスタンス]** を選択して新しい InterSystems IRIS インスタンスをインストールします。

注釈 既存のインスタンスを選択して、そのインスタンスを再インストールまたはアップグレードします。

5. コンピュータに新規の InterSystems IRIS インスタンスをインストールする場合は、**[使用許諾契約]** ダイアログ・ボックスが表示されます。**[使用許諾契約書に同意します]** をクリックして、ライセンス契約に同意します。

6. **[InterSystems IRIS インスタンス名]** ダイアログ・ボックスでは、インストールする新規インスタンスに名前を割り当てることができます。既定の名前は **IRIS** です。別のインスタンスが存在する場合は **IRIS_n** になります。n は、この新しいインスタンスを含めた InterSystems IRIS インスタンスの数です。既定の名前を受け入れるか、英数字、アンダースコア、ダッシュのみを使用して別の名前を入力します。インスタンスのインストール後に、インスタンス名を変更することはできません。
7. **[目的のフォルダ]** ダイアログ・ボックスでは、新規インスタンスの InterSystems IRIS ソフトウェアをインストールするディレクトリを選択できます。**[変更]** をクリックすると、別のディレクトリを選択または作成できます。指定したディレクトリが存在しない場合は自動的に作成されます。

インストール・ディレクトリを選択するための要件は、“InterSystems IRIS のインストール準備” の章の “**インストール・ディレクトリ**” を参照してください。
8. **[セットアップタイプ]** ダイアログ・ボックスでは、InterSystems IRIS の使用目的を以下から指定できます。このドキュメントの “InterSystems IRIS のインストール準備” の章の “**セットアップ・タイプ**” のセクションを参照して、必要な InterSystems IRIS のコンポーネントを決定します。

注釈 Web ゲートウェイをインストールする際に、お使いのシステムに CSP ゲートウェイが既にインストールされている場合は、インストーラにより CSP ゲートウェイが Web ゲートウェイに自動的にアップグレードされます。詳細は、“**既存の CSP ゲートウェイ**” を参照してください。

インストール手順の次のステップは、選択する**セットアップ・タイプ**によって異なります。InterSystems IRIS のインストールを終了するには、選択した**セットアップ・タイプ**に対応するセクションのステップを実行します。

- ・ **開発** – **開発コンポーネントまたはサーバ・コンポーネントのみのインストール**
- ・ **サーバ** – **開発コンポーネントまたはサーバ・コンポーネントのみのインストール**
- ・ **クライアント** – **クライアント・コンポーネントのみのインストール**
- ・ **Web サーバ** – **Web ゲートウェイのみのインストール**
- ・ **カスタム** – **カスタム・インストールの実行**

2.2.1 開発コンポーネントまたはサーバ・コンポーネントのみのインストール

必要に応じて、開発システムまたはサーバ・システムで必要な InterSystems IRIS のコンポーネントのみをインストールできます。

開発またはサーバのインストールを実行するには、まず、“**InterSystems IRIS のインストール手順**” の各ステップを完了してから、以下を実行します。

1. **[セットアップタイプ]** ダイアログ・ボックスで **[開発]** または **[サーバ]** を選択します。
2. **[Unicodeサポートのインストール]** ダイアログ・ボックスで、インストールの **8 ビットまたは Unicode 文字のサポート** を選択できます (既定値は、オペレーティング・システムのロケールによって決まります)。
3. **[初期セキュリティ設定]** ダイアログ・ボックスでは、インターシステムズの初期セキュリティ設定による制限を指定できます (**[最小]**、**[通常]**、または **[ロック・ダウン]**)。これらのセキュリティ設定の詳細は、“**インターシステムズの初期セキュリティ設定**” を参照してください。

[最小] は、InterSystems IRIS インストールにのみ使用できます。**[最小]** を選択する場合は、**[インストール準備完了]** 手順に進みます。

重要 初期セキュリティ設定を **[最小]** に選択したうえで、InterSystems IRIS からネットワーク経由で共有ドライブや共有プリンタにアクセスできるようにする場合は、InterSystems サービスを実行する Windows ユーザ・アカウントを、必要なアクセス特権を持つアカウントに手動で変更する必要があります。その手順については、“**InterSystems IRIS インスタンスへのアクセスの管理**” を参照してください。

4. **[InterSystemsサービスの認証情報入力]** ダイアログで、Windows InterSystems サービスを実行する資格情報を選択します。

既定は、ローカルの既定の SYSTEM アカウントです。代わりに、定義済みの（既存の）Windows ユーザ・アカウントおよびパスワードを指定することもできます。この場合、インストーラは以下を確認します。

- ・ そのアカウントがドメインに存在するかどうか。
- ・ パスワードが正しいかどうか。

注釈 既定の SYSTEM アカウントが常に適切であるとは限らず、ファイルのアクセス許可やネットワーク・セキュリティのアクセスに関連する問題を引き起こす場合があります。また、Kerberos を使用する場合は、InterSystems サービスを実行するために設定した定義済みのアカウントを入力する必要があります。Windows InterSystems IRIS サービスを実行するアカウントを決定するには、この章の **“Windows ユーザ・アカウント”** のセクションを参照してください。

5. **[InterSystems IRIS ユーザ設定]** ダイアログでは、ダイアログ・ボックスにリストされている事前定義の InterSystems IRIS ユーザ・アカウントの初期パスワードを入力できます。パスワードは、**“初期のユーザ・セキュリティ設定”** の一覧に示した条件を満たしている必要があります。**[次へ]** をクリックすると、CSPSystem の事前定義アカウントの初期パスワードを入力するよう求められます。

前の手順で Windows アカウントを指定した場合、インストール時に Windows アカウントと同じ名前の **“特権ユーザ・アカウント”** が作成され、そのユーザに、InterSystems IRIS の管理に必要なサービスへのアクセス権が付与されます。これらのすべての事前定義アカウントの詳細は、**“事前定義のユーザ・アカウント”** を参照してください。

6. **[インストール準備完了]** ダイアログ・ボックスでは、インストール名、インストール・タイプ、インストール・ディレクトリ、およびライセンス・キーの状況を確認できます。

また、**[ライセンス]** ボタンをクリックして、InterSystems IRIS ライセンス・キーを選択することもできます。キーが有効の場合、インストール中にライセンスは自動的に有効化されて、ライセンス・キーはインスタンスの `install-dir/mgr` ディレクトリに `iris.key` としてコピーされるので、これ以上の有効化手順は不要です。キーを選択しない場合、インストールに続いて InterSystems IRIS ライセンス・キーを有効化できます。ライセンス、ライセンス・キーおよび有効化についての詳細は、**“システム管理ガイド”** の **“InterSystems IRIS ライセンスの管理”** の章にある **“ライセンス・キーの有効化”** を参照してください。

[インストール] をクリックして次に進みます。選択したディレクトリに InterSystems IRIS がインストールされます。

7. **[InterSystems IRIS インストールの完了]** ダイアログ・ボックスに、インストールが正常に完了したことが示されます。**[はじめに]** ページを表示するかどうかを選択し、**[完了]** をクリックします。

既定では、インストールが完了すると、システムが自動的に起動し、Windows ツール・バーのシステム・トレイ領域に InterSystems IRIS ランチャーのアイコンが表示されます。ランチャーをクリックすると、InterSystems IRIS メニューが表示されます。また、Windows の **[プログラム]** メニューにも **InterSystems IRIS** 項目が表示されます。

“インストール後のタスク” のセクションに進んで、インストール・プロセスを完了します。

2.2.2 クライアント・コンポーネントのみのインストール

必要に応じて、InterSystems IRIS の機能のうち、クライアント・マシンで必要なもののみをインストールできます。

クライアントのインストールを実行するには、まず、**“InterSystems IRIS のインストール手順”** の各ステップを完了してから、以下を実行します。

1. **[セットアップタイプ]** ダイアログ・ボックスで **[クライアント]** を選択して、**[次へ]** をクリックします。
2. **[インストール準備完了]** ダイアログ・ボックスでは、インストール名、インストール・タイプ、およびソフトウェア・ファイルのインストール先ディレクトリを確認できます。

[インストール] をクリックして次に進みます。選択したディレクトリに InterSystems IRIS がインストールされます。

3. **[InterSystems IRIS インストールの完了]** ダイアログ・ボックスに、インストールが正常に完了したことが示されます。
[完了] をクリックします。

InterSystems IRIS をクライアントにインストールすると、Windows ツール・バーのシステム・トレイ領域に InterSystems IRIS ランチャーのアイコンが表示されます。ただし、InterSystems IRIS サーバが動作していないので、淡色表示になっています。

重要 クライアントを使用するには、このクライアントの優先サーバを指定する必要があります。この手順の詳細は、“システム管理ガイド”の“リモート・サーバへの接続”の章にある“[リモート・サーバ接続の定義](#)”のセクションを参照してください。

“[インストール後のタスク](#)”のセクションに進んで、インストール・プロセスを完了します。

2.2.3 Web ゲートウェイのみのインストール

必要に応じて、InterSystems IRIS の機能のうち、Web ゲートウェイに必要なもののみをインストールできます。

Web ゲートウェイのインストールを実行するには、まず、“[InterSystems IRIS のインストール手順](#)”の各ステップを完了してから、以下を実行します。

1. **[セットアップタイプ]** ダイアログ・ボックスで **[Web サーバ]** を選択して、**[次へ]** をクリックします。
2. **[インストール準備完了]** ダイアログ・ボックスでは、インストール名、インストール・タイプ、およびソフトウェア・ファイルのインストール先ディレクトリを確認できます。
[インストール] をクリックして次に進みます。選択したディレクトリに InterSystems IRIS がインストールされます。
3. **[InterSystems IRIS インストールの完了]** ダイアログ・ボックスに、インストールが正常に完了したことが示されます。
[完了] をクリックします。

Web サーバが実行している場合、Web サーバを再起動するかどうかを尋ねるダイアログ・ボックスが表示されます。**[はい]** をクリックすると、Web サーバが再起動されます。**[いいえ]** をクリックすると、Web サーバは再起動されません。この場合、Web サーバは、手動で再起動するかシステムを再起動するまで起動しません。

インストーラにより Internet Information Services (IIS) Web サーバがシステムにインストールされていることが検出されると、Web サーバは Web ゲートウェイ用に構成されます。また、IIS 用のチェック・ボックスが表示されます。このチェック・ボックスにチェックを付けると、Web ゲートウェイ IIS モジュールが **C:\inetpub\wwwroot\CSPGateway** にインストールされます。

Web ゲートウェイでは、インスタンス用に構成されたサーバをポイントする次のアプリケーション・パスが構成されます。

```

・ /
・ /csp
・ /<instancename> (既定では、/IRIS)

```

インストールが完了した後、Web ゲートウェイ・アプリケーションから手動で構成を変更できます。詳細は、“[Web ゲートウェイ・ガイド](#)”を参照してください。

注釈 このインストーラは、InterSystems IRIS および CSP で使用する Apache Web サーバを自動的に構成できません。必要な手動構成手順の詳細は、“[Web ゲートウェイ・ガイド](#)”を参照してください。

“[インストール後のタスク](#)”のセクションに進んで、インストール・プロセスを完了します。

2.2.4 カスタム・インストールの実行

InterSystems IRIS インストール・プログラムを利用すると、特定の InterSystems IRIS コンポーネントのみを選択してシステムにインストールできます。例えば、Web ゲートウェイのみをインストールできます。ただし、選択によっては、他のコンポーネントのインストールが必要な場合もあります。

InterSystems IRIS のカスタム・インストールを実行するには、まず、“[InterSystems IRIS のインストール手順](#)”の各ステップを完了してから、以下を実行します。

1. **[セットアップタイプ]** ダイアログ・ボックスで **[カスタム]** を選択します。
2. **[カスタム・セットアップ]** ダイアログ・ボックスで、インストールするコンポーネントを選択します。一部のコンポーネントにはサブ項目があります。これらを表示するには、**[+]** アイコンを選択します。

コンポーネントを選択して簡単な説明を表示するか、“[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](#)” テーブルで使用可能なコンポーネントのリストを表示します。

注釈 インストールされているコンポーネント・グループまたはコンポーネントについて **[X]** メニュー項目を選択すると、そのコンポーネントを削除できます。

3. 必要に応じて **[容量]** をクリックすると、選択したコンポーネントをインストールするための十分な容量がディスクにあるかどうかを確認できます。
4. **[Unicode サポートのインストール]** ダイアログ・ボックスで、インストールの **8 ビット** または **Unicode 文字のサポート** を選択できます (既定値は、オペレーティング・システムのロケールによって決まります)。
5. **[ポート番号の入力]** ダイアログ・ボックスでは、InterSystems IRIS が割り当てたポート番号を変更できます。詳細は、このドキュメントの “InterSystems IRIS のインストール準備” の章の “[ポート番号](#)” のセクションを参照してください。
6. **[初期セキュリティ設定]** ダイアログ・ボックスでは、インターシステムズの初期セキュリティ設定による制限を指定できます (**[最小]**、**[通常]**、または **[ロック・ダウン]**)。これらのセキュリティ設定の詳細は、“[インターシステムズの初期セキュリティ設定](#)” を参照してください。

[最小] は、InterSystems IRIS インストールにのみ使用できます。**[最小]** を選択する場合は、**[インストール準備完了]** 手順に進みます。

重要 初期セキュリティ設定を **[最小]** に選択したうえで、InterSystems IRIS からネットワーク経由で共有ドライブや共有プリンタにアクセスできるようにする場合は、InterSystems サービスを実行する Windows ユーザ・アカウントを手動で変更して、サーバ・マシンに対するローカル管理者特権を持つ既存のアカウントを選択するか、同様の特権を持つ新しいアカウントを作成する必要があります。その手順については、“[InterSystems IRIS インスタンスへのアクセスの管理](#)” を参照してください。

7. **[InterSystems サービスの認証情報入力]** ダイアログで、Windows InterSystems サービスを実行する資格情報を選択します。

既定は、ローカルの既定の SYSTEM アカウントです。代わりに、定義済みの (既存の) Windows ユーザ・アカウントおよびパスワードを指定することもできます。この場合、インストーラは以下を確認します。

- ・ そのアカウントがドメインに存在するかどうか。
- ・ パスワードが正しいかどうか。

注釈 既定の SYSTEM アカウントが常に適切であるとは限らず、ファイルのアクセス許可やネットワーク・セキュリティのアクセスに関連する問題を引き起こす場合があります。また、Kerberos を使用する場合は、InterSystems サービスを実行するために設定した定義済みのアカウントを入力する必要があります。Windows InterSystems IRIS サービスを実行するアカウントを決定するには、この章の “[Windows ユーザ・アカウント](#)” のセクションを参照してください。

8. **[InterSystems IRIS ユーザ設定]** ダイアログでは、ダイアログ・ボックスにリストされている事前定義の InterSystems IRIS ユーザ・アカウントの初期パスワードを入力できます。パスワードは、“[初期のユーザ・セキュリティ設定](#)”の一覧に示した条件を満たしている必要があります。**[次へ]**をクリックすると、CSPSystem の事前定義アカウントの初期パスワードを入力するよう求められます。

前の手順で Windows アカウントを指定した場合、インストール時に Windows アカウントと同じ名前の“特権ユーザ・アカウント”が作成され、そのユーザに、InterSystems IRIS の管理に必要なサービスへのアクセス権が付与されます。これらのすべての事前定義アカウントの詳細は、“[事前定義のユーザ・アカウント](#)”を参照してください。

9. **[インストール準備完了]** ダイアログ・ボックスでは、インストール名、インストール・タイプ、インストール・ディレクトリ、およびライセンス・キーの状況を確認できます。

また、**[ライセンス]** ボタンをクリックして、InterSystems IRIS ライセンス・キーを選択することもできます。キーが有効の場合、インストール中にライセンスは自動的に有効化されて、ライセンス・キーはインスタンスの `install-dir/mgr` ディレクトリに `iris.key` としてコピーされるので、これ以上の有効化手順は不要です。キーを選択しない場合、インストールに続いて InterSystems IRIS ライセンス・キーを有効化できます。ライセンス、ライセンス・キーおよび有効化についての詳細は、“システム管理ガイド”の“InterSystems IRIS ライセンスの管理”の章にある“[ライセンス・キーの有効化](#)”を参照してください。

[インストール] をクリックして次に進みます。選択したディレクトリに InterSystems IRIS がインストールされます。

10. **[InterSystems IRIS インストールの完了]** ダイアログ・ボックスに、インストールが正常に完了したことが示されます。**[はじめに]** ページを表示するかどうかを選択し、**[完了]** をクリックします。

“[インストール後のタスク](#)”のセクションに進んで、インストール・プロセスを完了します。

2.3 自動インストール手順

Windows 用 InterSystems IRIS インストーラを使用して、ユーザのコンピュータに InterSystems IRIS のインスタンスを自動でインストール、アップグレード、再インストール（修復）、および削除（アンインストール）することができます。通常のインストール操作の場合、ユーザがダイアログ・ボックスに必要な情報を入力し、この情報からインストールを実行します。ただし、自動操作ではユーザの入力が求められません。その代わりに、コマンド行で InterSystems IRIS インストール・ファイルに渡されるプロパティから入力を取得します。これらのプロパティの詳細は、“[コマンド行のリファレンス](#)”のセクションを参照してください。

このセクションでは、以下のトピックについて説明します。

- ・ [自動インストールの実行](#)
- ・ [自動アップグレードまたは再インストールの実行](#)
- ・ [自動削除の実行](#)
- ・ [コマンド行のリファレンス](#)

注釈 自動インストール、アップグレード、再インストール、およびアンインストール中には、メッセージは表示されません。

2.3.1 自動インストールの実行

InterSystems IRIS の新しいインスタンスの自動インストールを起動するには、以下のコマンドを使用します。

```
<path>\<installer>.exe /instance <instancename> /q{b|n} <properties>
```

以下を指定する必要があります。

変数	説明
<path>	InterSystems IRIS インストール・ファイルのパス。
<installer>.exe	InterSystems IRIS インストール・ファイルの名前。
<instancename>	新しい InterSystems IRIS インスタンスの名前。省略した場合、既定値は IRIS ですが、マシンに 1 つ以上のインスタンスが既にインストールされている場合は、別の値を指定する必要があります。
/qb または /qn	インストール中に進捗バーを表示するか (/qb)、または完全サイレント・インストールを実行するか (/qn)。
<properties>	インストーラに渡すプロパティ(“ コマンド行のプロパティ ”のテーブルを参照)。

例えば、64ビット Windows システム上の **C:\InterSystems\MyIris** というインストール・ディレクトリに既定のインスタンス名で InterSystems IRIS インスタンスをインストールするには、以下を指定します。

```
C:\downloads\IRIS-2018.1.0.508.0-win_x64.exe /qn INSTALLDIR=C:\InterSystems\MyIris
```

インスタンス名 **IrisA** で InterSystems IRIS のインスタンスをインストールするには、以下を指定します。

```
C:\downloads\IRIS-2018.1.0.508.0-win_x64.exe /instance IrisA /qn
```

ADDLOCAL プロパティを使用して、機能のサブセットをカスタム・インストールできます。詳細は、“[コマンド行のプロパティ](#)”のテーブルの ADDLOCAL を参照してください。例えば、64 ビット Windows システムにランチャーのみをインストールするには (既定のインスタンス名とディレクトリをそのまま使用)、以下を指定します。

```
C:\downloads\IRIS-2018.1.0.508.0-win_x64.exe /qn ADDLOCAL=cube
```

注釈 既定では、自動インストールでは Web ゲートウェイはインストールされません。これは、ADDLOCAL プロパティを使用して指定する必要があります。

インストールが終了したら、この章の“[インストール後のタスク](#)”のセクションに進みます。

2.3.2 自動アップグレードまたは再インストールの実行

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
```

2.3.3 自動削除の実行

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
```

2.3.3.1 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。

2.3.4 コマンド行のリファレンス

“**コマンド行のプロパティ**” のテーブルでは、コマンド行インタフェースを使用して変更できる InterSystems IRIS 固有の自動インストールのプロパティについて説明します。プロパティ名は大文字で指定する必要がありますが、引数では大文字/小文字の区別はありません。それぞれのプロパティは 1 つ以上の空白で区切る必要があり、プロパティは **PROPERTYNAME=argument** という形式で指定できます。指定の順序は問われません。以下に例を示します。

```
... ISCSTARTIRIS=0 WEBSERVERPORT=57779 INITIALSECURITY=Normal
```

注釈 以下のテーブルでは、REINSTALL プロパティと REMOVE プロパティが、インストールされたインスタンスと共に使用されます。詳細は、それぞれ “**自動アップグレードまたは再インストールの実行**” および “**自動削除の実行**” を参照してください。

テーブル 2-1: コマンド行のプロパティ

プロパティ名	説明
ADDLOCAL	<p>このプロパティを使用して、このテーブルの後の例で説明しているように、featurenames のコンマ区切りリストをグループ名と共に指定することで、InterSystems IRIS の新規インスタンスを機能のサブセットと共にカスタム・インストールしたり、オプションのデータベースを除外したりします (“カスタム・インストール可能な機能” テーブルを参照)。</p> <p>注釈 ADDLOCAL プロパティが存在しない場合、または ADDLOCAL=ALL が指定されている場合、すべての機能がインストールされます。</p> <p>インストールされたインスタンスと共に使用される、REINSTALL プロパティも参照してください。</p>
CSPSKIPIISCONFIG	<p>オプションで、このプロパティを使用して IIS CSP バイナリ・ファイルをインストールします。値 1 を指定すると、IIS Web サーバ構成に変更を加えることなくファイルがインストールされます。値 0 を指定すると、仮想ディレクトリ /csp の有無にかかわらず、IIS Web サーバ構成が更新されます。</p>
CSPSKIPAPACHE20CONFIG	<p>オプションで、このプロパティを使用して Apache 2.0 CSP バイナリ・ファイルをインストールします。値 1 を指定すると、IIS Web サーバ構成に変更を加えることなくファイルがインストールされます。値 0 を指定すると、仮想ディレクトリ /csp の有無にかかわらず、IIS Web サーバ構成が更新されます。</p>
CSPSKIPAPACHE22CONFIG	<p>オプションで、このプロパティを使用して Apache 2.2 CSP バイナリ・ファイルをインストールします。値 1 を指定すると、IIS Web サーバ構成に変更を加えることなくファイルがインストールされます。値 0 を指定すると、仮想ディレクトリ /csp の有無にかかわらず、IIS Web サーバ構成が更新されます。</p>
CSPSYSTEMUSERPASSWORD	<p>セキュリティ・レベルが Normal または LockedDown (このテーブルの INITIALSECURITY プロパティを参照) の場合、必要に応じてこのプロパティを使用して、CSPSystem の事前定義されたユーザのパスワードを指定します。このプロパティを省略した場合、IRISUSERPASSWORD の値が使用されます。</p> <p>注釈 初期セキュリティ・レベルが None の場合は、このプロパティを使用しないでください。</p>
INITIALSECURITY	<p>必要に応じてこのプロパティを使用して、インストールされるインスタンスで使用されるセキュリティ・レベルを指定します。None、Normal、または LockedDown を指定してください。</p> <p>注釈 既定の None を受け入れるには、このプロパティを省略します。</p> <p>また、このテーブルの IRISUSERPASSWORD プロパティ、CSPSYSTEMUSERPASSWORD プロパティ、および SERVICECREDENTIALS プロパティも参照してください。</p>

プロパティ名	説明
INSTALLDIR	<p>必要に応じてこのプロパティを使用して、インスタンスをインストールするディレクトリを指定します。</p> <p>注釈 このプロパティを省略すると、既定のインストール・ディレクトリは、<code>C:\InterSystems\IRISn</code> になります。ここで、<code>n</code> は <code>{empty}</code>、1、2、...127 になります。</p>
INSTALLERMANIFEST	<p>付録 “インストール・マニフェストの作成および使用” の “マニフェストの使用” で説明されているようにインストール・マニフェストと共にインストールする場合、このプロパティを使用して、インストール・マニフェスト（つまり、エクスポートされるマニフェスト・クラス）の場所を指定する必要があります。</p>
INSTALLERMANIFESTLOGFILE	<p>付録 “インストール・マニフェストの作成および使用” の “マニフェストの使用” で説明されているように、インストール・マニフェストと共にインストールする場合、%Installer によるメッセージの出力先をこのプロパティで指定します。</p>
INSTALLERMANIFESTLOGLEVEL	<p>インストール・マニフェストでインストールする場合、必要に応じてこのプロパティを使用して、インストール・マニフェスト・クラスの <code>setup()</code> メソッドの ログ・レベル を指定します。デフォルトのログのレベルは 1 です。</p>

プロパティ名	説明
INSTALLERMANIFESTPARAMS	<p>付録 “インストール・マニフェストの作成および使用” の “マニフェストの使用” で説明されているように、インストール・マニフェストと共にインストールする場合、このプロパティを使用して、インストール・マニフェスト・クラスの setup() メソッドの最初の引数に渡される名前と値の組み合わせ (name=value) を指定します。このプロパティを使用して、構成パラメータ・ファイル (iris.cpf) を変更して、マニフェストの実行前に変更を有効化することができます。以下のパラメータを指定できます。</p> <ul style="list-style-type: none"> • bbsiz • globals4kb、globals8kb、globals16kb、globals32kb、globals64kb • gmheap • LibPath • locksiz • MaxServerConn • Path • routines • ZFSize、ZFString <p>以下に例を示します。</p> <pre>INSTALLERMANIFESTPARAMS="bbsiz=512000,globals4kb=20, globals8kb=30,globals16kb=40,globals32kb=50, globals64kb=100,routines=40,gmheap=10000, LibPath=c:\libpath\,locksiz=2179648,MaxServerConn=5, Path=c:\lib\,ZFSize=2000,ZFString=3000"</pre> <p>64KB ブロック・サイズのデータベースを作成するマニフェストを実行する前に、100 MB (64KB バッファ) をインストールおよび有効化する場合には、以下が役立ちます。</p> <pre>INSTALLERMANIFESTPARAMS="globals64kb=100"</pre>
IRISSERVICEDOMAIN	<p>サービスの資格情報が UserDefined に定義されている場合に必要になります。このテーブルの SERVICECREDENTIALS プロパティを参照してください。このプロパティを使用して、IRISSERVICEUSER によって指定された Windows の InterSystems サービス・ログイン・アカウントのドメインを指定します。</p> <p>注釈 サービスの資格情報が LocalSystem と指定されている場合は、このプロパティを使用しないでください。</p>
IRISSERVICEPASSWORD	<p>サービスの資格情報が UserDefined に定義されている場合に必要になります。このテーブルの SERVICECREDENTIALS プロパティを参照してください。このプロパティを使用して、IRISSERVICEUSER によって指定された Windows の InterSystems サービス・アカウントのパスワードを指定します。</p> <p>注釈 サービスの資格情報が LocalSystem と指定されている場合は、このプロパティを使用しないでください。</p>

プロパティ名	説明
IRISSERVICEUSER	<p>サービスの資格情報が UserDefined に定義されている場合に必要になります。このテーブルの SERVICECREDENTIALS プロパティを参照してください。このプロパティを使用して、Windows の InterSystems サービスを実行するアカウントのユーザ名を指定します。</p> <p>注釈 サービスの資格情報が LocalSystem と指定されている場合は、このプロパティを使用しないでください。</p>
IRISUSERPASSWORD	<p>セキュリティ・レベルが Normal または LockedDown の場合に必要になります。このテーブルの INITIALSECURITY プロパティを参照してください。このプロパティを使用して、事前定義の InterSystems IRIS アカウント (SYSTEM、Admin、SuperUser) のパスワードを指定します。また、SERVICECREDENTIALS が UserDefined に指定されている場合には、IRISSERVICEUSER によって指定されたユーザ名を持つアカウントのパスワードも指定します。</p> <p>注釈 初期セキュリティ・レベルが None の場合は、このプロパティを使用しないでください。</p>
ISCSTARTIRIS	<p>インストール後に InterSystems IRIS が起動しないようにしたい場合は、オプションでこのプロパティを 0 に設定できます。既定は 1 です (InterSystems IRIS が起動します)。</p>
ISCSTARTLAUNCHER	<p>必要に応じてこのプロパティを 0 に設定すると、システム・トレイに InterSystems IRIS ランチャーが追加されなくなります。既定は 1 です (ランチャーが追加されます)。</p>
REINSTALL	<p>このプロパティを使用して、インストールされた InterSystems IRIS のインスタンスを再インストール (修復) したり、インストールされた InterSystems IRIS のインスタンスのカスタム・インストールされた機能を変更したりします (“カスタム・インストール可能な機能” テーブルを参照)。</p> <ul style="list-style-type: none"> ・ インスタンスに現在インストールされている機能 (カスタム・インストールされた機能のサブセットの場合も、すべての機能の場合も) を再インストールするには、ALL と指定します。 ・ 現在インストールされている機能のサブセットとは異なる InterSystems IRIS 機能のサブセットを再インストールするには、featurenames のコンマ区切りリストをグループ名と共に指定します (このテーブルに続く例を参照)。 <p>ADDLOCAL プロパティ (新規インスタンスと共に使用する場合) および REMOVE プロパティ (インストールされたインスタンスをアンインストールする場合) も参照してください。</p>

プロパティ名	説明
REMOVE	<p>このプロパティを使用して、InterSystems IRIS のインスタンス、またはインストールされた InterSystems IRIS のインスタンスにカスタム・インストールされた機能のサブセットをアンインストール（削除）します（“カスタム・インストール可能な機能” テーブルを参照）。</p> <ul style="list-style-type: none"> InterSystems IRIS のインスタンスを削除するには、ALL と指定します。 InterSystems IRIS 機能のサブセットを削除するには、featurename のコンマ区切りリストをグループ名と共に指定します（このテーブルに続く例を参照）。 <p>このテーブルにある ADDLOCAL（新規インスタンスの場合）および REMOVE プロパティも参照してください。</p>
SERVICECREDENTIALS	<p>セキュリティ・レベルが Normal または LockedDown（このテーブルの INITIALSECURITY プロパティを参照）の場合、必要に応じてこのプロパティを使用して、Windows InterSystems サービスを実行する資格情報を指定します。LocalSystem（既定のローカル・システム・アカウント用）または UserDefined（既存の Windows ユーザ・アカウント）のいずれかを指定します。プロパティを指定しない場合、既定の LocalSystem が使用されます。</p> <p>注釈 初期セキュリティ・レベルが None の場合は、このプロパティを使用しないでください。</p> <p>InterSystems サービス・アカウントの重要な情報は、“InterSystems IRIS インスタンスへのアクセスの管理” を参照してください。</p> <p>このプロパティに UserDefined を指定した場合、IRISSERVICEDOMAIN プロパティ、IRISSERVICEPASSWORD プロパティ、および IRISSERVICEUSER プロパティも指定する必要があります。</p>
SKIPUPGRADECHECK	<p>インスタンスをアップグレードする場合に、アップグレード前のシステム・チェックをバイパスするには、このプロパティを 1 に設定します。既定値は 0 です。</p> <p>注釈 一般的に、アップグレード前のチェックは有効なままにすることをお勧めします。</p>
SUPERSERVERPORT	<p>必要に応じてこのプロパティを使用して、インストールされるインスタンスで使用されるスーパーサーバ・ポートを指定します。</p> <p>注釈 既定では、このポートは 1972 に設定されます（可能な場合）。それ以外の場合、ポートは 51773、またはこれ以降の使用可能な最初のポート番号に設定されます。</p>

プロパティ名	説明
UNICODE	<p>必要に応じてこのプロパティを使用して、インストールされるインスタンスでサポートされるのが 8 ビット文字か 16 ビット Unicode 文字かを指定します。8 ビット文字の場合は 0 を指定し、16 ビット文字の場合は 1 を指定します。</p> <p>このプロパティを省略する場合、中国語、韓国語、日本語を除くすべての言語に、既定で 8 ビットが指定されます。中国語、韓国語、日本語のシステムには、既定で 16 ビットが指定されます。</p>
WEBSERVERPORT	<p>必要に応じてこのプロパティを使用して、インストールされるインスタンスで使用される Web サーバ・ポートを指定します。</p> <p>注釈 既定では、このポートは自動的に検出されます。最初は 52773 で、インストールされた InterSystems IRIS のインスタンスごとに 1 ずつ増加します。</p>

“カスタム・インストール可能な機能” のテーブルには、コンポーネント・グループ/コンポーネント名とそれぞれに関連付けられた featurename が示されています。“ALL” (使用可能なすべての機能を指定) または機能名のコンマ区切りリスト (空白なし、個々の機能を指定) を指定できます。

ADDLOCAL、REINSTALL、および REMOVE のプロパティでコンポーネントを指定するには、コンポーネント・グループの featurename の後に、インストールするそのグループからの各特定コンポーネントの featurename を付けて指定します。例えば、USER データベースのみをインストールする場合、コマンド行で以下を指定します。

```
ADDLOCAL=server,server_user
```

コンポーネント・グループを指定する場合は、関連付けられたコンポーネントを少なくとも 1 つ指定することも必要となります。コンポーネント・グループにコンポーネントがリストされていない場合、そのコンポーネント・グループは無視され、コンポーネントはインストールされません。例えば、次のように指定したとします。

```
ADDLOCAL=documentation,documentation_pdf,server,development,callin
```

この場合、server コンポーネント・グループは無視され、サーバ・コンポーネントはインストールされません。([スタジオ] および [キューブ] グループにはコンポーネントがないため、これらのグループにはこの要件は適用されません。)

テーブル 2-2: カスタム・インストール可能な機能

コンポーネント・グループ (機能名)	コンポーネント (機能名)
開発 (development)	<p>コールイン (callin)</p> <p>コールイン、スレッド (callin_threaded)</p> <p>スレッド・サーバ・ライブラリ (server_threaded)</p> <p>その他のサンプル (other_samples)</p> <p>その他の開発ライブラリ (development_other)</p>
ドキュメント (documentation)	<p>PDF ドキュメント (documentation_pdf)</p> <p>オンライン・ドキュメント (documentation_online)</p>
InterSystems IntegratedML (integratedml)	

コンポーネント・グループ (機能名)	コンポーネント (機能名)
ランチャー (cube)	
サーバ (server)	User データベース (server_user) SQL ゲートウェイ (sqlgateway) Apache Formatting Objects Processor (fop) サーバ監視ツール (server_monitoring) エージェント・サービス (agent_service)
スタジオ (studio)	
xDBC (sqltools)	ODBC (odbc) JDBC (jdbc)
Web ゲートウェイ (cspgateway)	IIS (cspiis) Apache 2.0 (cspapache20) Apache 2.2 (cspapache22)

2.4 インストール後のタスク

以下の重要なインストール後のタスクを確認してください。

- InterSystems IRIS ランチャーからアクセスできる管理ポータルを使用して、InterSystems IRIS インスタンスを管理できます。この管理ツールの詳細は、“システム管理ガイド”の“[管理ポータルの使用](#)”の章を参照してください。
- Windows InterSystems サービスはインスタンス名を含んでおり、サーバを起動すると自動的に開始されます。つまり、InterSystems IRIS インスタンスは、自動開始 (システムの起動時に開始) するように自動的に構成されます。
このインスタンスが自動的に開始されないように構成するには、管理ポータルの [メモリと開始設定] ページ (ホーム・ページから [システム管理] → [構成] → [システム構成] → [メモリと開始設定]) で [システム起動時に InterSystems IRIS を開始する] 設定を変更します。
- リモートで他の InterSystems IRIS のインスタンスに接続する場合は、“システム管理ガイド”の“リモート・サーバへの接続”の章にある“[リモート・サーバ接続の定義](#)”のセクションに記載されている手順に従います。
- Windows の IIS Web サーバを使用している場合は、ファイルの種類を手動でマップする必要があります。“Web ゲートウェイ・ガイド”の“[InterSystems IRIS ファイル拡張子のマッピング](#)”を参照してください。
- “システム管理ガイド”の“InterSystems IRIS の構成”の章にある“[メモリと開始設定](#)”のセクションの説明に従って、InterSystems IRIS で使用されるシステム・メモリを割り当てます。
- インターシステムズ製品は、インターシステムズ製ではないツールと共に実行したり、そのようなツールとやり取りしたりすることが頻繁にあります。このようなやり取りでもたらされる可能性がある影響に関する重要な情報については、“システム管理ガイド”の付録“[インターシステムズ製品と連係して動作するようにサードパーティ・ソフトウェアを構成する方法](#)”を参照してください。
- 別のデータベースから新しくインストールした InterSystems IRIS インスタンスにデータを移行する必要がある場合は、“データの移行”を参照してください。
- 必要に応じて、“[特別な考慮事項](#)”に記載されているタスクも追加で実行してください。

2.5 特別な考慮事項

以下では、ライセンス、プラットフォーム固有の問題、インストールの種類に関する特定の問題やタスクについて説明します。

- ・ [複数の InterSystems IRIS インスタンスの構成](#)
- ・ [InterSystems IRIS の言語の変更](#)
- ・ [InterSystems IRIS の再インストールまたはアンインストール](#)
- ・ [書き込みキャッシュ・バッファ](#)

2.5.1 複数の InterSystems IRIS インスタンスの構成

1 台の Windows マシンに、複数のインスタンスをインストールして、同時に実行できます。それには、InterSystems IRIS の各インスタンスに一意的な名前を付け、別々のディレクトリにインストールし、それぞれに異なるポート番号を割り当てます。

詳細は、“システム管理ガイド”の“[複数の InterSystems IRIS インスタンス](#)”のセクションを参照してください。

2.5.2 InterSystems IRIS の言語の変更

InterSystems IRIS をインストールすると、サポート対象言語固有のユーティリティ DLL は、すべて `install-dir\bin` ディレクトリにインストールされます。各 DLL には、ローカライズされた文字列とメッセージが含まれます。

DLL 名の形式は `UTILaaa.DLL` で、aaa には、以下の言語を表す 3 文字のコードが入ります。

コード	Language
CHS	中国語（簡体字）
DEU	ドイツ語（標準）
ENU	英語（アメリカ）
ESP	スペイン語（スペイン）
FRA	フランス語
ITA	イタリア語（標準）
JPN	日本語
KOR	韓国語
NLD	オランダ語（標準）
PTB	ポルトガル語（ブラジル）
RUS	ロシア語

InterSystems IRIS インストールのロケール変更の詳細は、“InterSystems IRIS システム管理ガイド”の“InterSystems IRIS の構成”の章の“[管理ポータル の NLS 設定 ページ の 使用法](#)”を参照してください。

注釈 変更は 8 ビット・ロケール間または Unicode ロケール間でのみ可能で、8 ビットから Unicode、または Unicode から 8 ビットに変更することはできません。詳細は、“インターシステムズ・クラスリファレンス”の %SYS.NLS のエントリを参照してください。

2.5.3 InterSystems IRIS の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。

“[InterSystems IRIS のインストール手順](#)”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 (“[カスタム・インストールの実行](#)”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。

2.5.4 書き込みキャッシュ・バッファ

特定の InterSystems IRIS 機能では、**[Windowsによる書き込みキャッシュバッファのフラッシュ]** を使用します。これは、既定で有効化されています。このオプションが適切に有効化されていて、InterSystems IRIS がこれらの機能を最大限に活用できることを確認するには、次の手順を実行します。

1. **[コントロール パネル]** から **[デバイス マネージャー]** を開きます。
2. **[ディスクドライブ]** セクションでストレージ・デバイスを選択します。
3. **[ポリシー]** タブをクリックします。
4. **[デバイスで Windows による書き込みキャッシュ バッファのフラッシュをオフにする]** が選択されていないことを確認します。

3

InterSystems IRIS の UNIX®、Linux、および macOS へのインストール

この章では、UNIX® システム、Linux システム、または macOS システムに InterSystems IRIS® Data Platform をインストールする方法を説明します。この章は、ユーザが UNIX®、Linux、および macOS のディレクトリ構造、ユーティリティおよびコマンドに精通していることを前提に記述されています。

注釈 同一マシン上で複数のインストールを同時に実行することはできません。

3.1 インストール前

インストールを開始する前に、このドキュメントの“[InterSystems IRIS のインストール準備](#)”の章に記載された、お使いのプラットフォームに該当するすべての情報を確認しておいてください。さらに、以下のトピックを確認してください。

- ・ [インストール・キットの解凍](#)
- ・ [所有者およびグループの決定](#)
- ・ [構成方法の選択](#)
- ・ [Linux でのヒュージ・ページの構成](#)
- ・ [必要な依存関係のインストール](#)
- ・ [構成方法の選択](#)

3.1.1 インストール・キットの解凍

インストール・キットが .tar ファイルの形式 (iris-2019.3.0.710.0-lnxrhx64.tar.gz など) の場合、以下の例に示すようにファイルを一時ディレクトリに解凍し、許可の問題を回避します。

```
# mkdir /tmp/iriskit
# chmod og+rx /tmp/iriskit
# umask 022
# gunzip -c /download/iris-2019.3.0.710.0-lnxrhx64.tar.gz | ( cd /tmp/iriskit ; tar xf - )
```

インストール・ファイルは、.tar ファイルと同じ名前のディレクトリに解凍されます (例えば、/tmp/iriskit/iris-2019.3.0.710.0-lnxrhx64)。

重要 /home ディレクトリ (またはこのディレクトリのサブディレクトリ) にファイルを解凍したり、InterSystems IRIS インストールをこれらのディレクトリから実行したりしないでください。また、一時ディレクトリのパス名にスペースを含めることはできません。

注釈 従来の tar コマンドは、長いパス名に遭遇すると通知なしで失敗する場合がありますため、GNU tar を使用してこのファイルを untar することをお勧めします。tar コマンドが GNU tar かどうかを判定するには、tar --version を実行します。

3.1.2 所有者およびグループの決定

インストール・プロセスでは、以下のユーザとグループの情報の入力を求められます。

- ・ インスタンスの所有者
- ・ InterSystems IRIS スーパーサーバの実効ユーザとそのジョブ
- ・ InterSystems IRIS プロセスの実効グループ
- ・ インスタンスの開始および停止を許可されたグループ

これらのカテゴリの詳細は、“システム管理ガイド”の“UNIX®, Linux、および macOS での InterSystems IRIS 使用法”の章にある“UNIX® ユーザとグループの識別”のセクションを参照してください。

重要 InterSystems IRIS はインストール対象のファイルに対するユーザやグループなどの許可を設定する必要があります。これを行うために、InterSystems IRIS はインストール・プロセスの umask を 022 に設定します。インストールが完了するまで、umask を変更しないでください。

インスタンスの所有者として指定するユーザ・アカウントと、インスタンスの開始および停止を許可されたグループとして指定するグループの両方が、インストールを開始する前に存在している必要があります。これらのプロンプトの 1 つで指定したエントリが存在しない場合、そのプロンプトが再表示されるため、指定しようとしているユーザおよびグループがインストール開始前に存在することを確認してください。

使用しているオペレーティング・システムに useradd ユーティリティおよび groupadd ユーティリティが含まれている場合は、指定したエントリが存在しなければ、InterSystems IRIS スーパーサーバ用の実効ユーザのアカウントおよび InterSystems IRIS プロセスの実効グループが作成されます。ただし、これらのユーティリティがなく、指定したエントリが存在しない場合、該当のプロンプトが再表示されます。システムにこれらのユーティリティがあるかどうか不明な場合は、インストールを開始する前に、指定しようとしているユーザおよびグループが存在するかどうかを確認します。

注釈 オペレーティング・システムで Network Information Services (NIS) やその他のネットワーク・ベースのユーザまたはグループのデータベースを使用している場合は、groupadd ユーティリティと useradd ユーティリティによりローカル・ユーザおよびグループが作成され、ネットワーク・データベースに既に存在するエントリと競合する可能性があります。この問題を回避するには、ユーティリティに InterSystems IRIS 実効グループおよび実効ユーザを作成させるのではなく、インストールを開始する前に適切な管理ツールを使用して、これらをネットワーク・データベースに作成することをお勧めします。

プロセスの所有権を表示するために UNIX® のオペレーティング・システムで使用するツールでは、実際の所有権と実効所有権との対応が必ずしも表示できるとは限りません。InterSystems IRIS による許可の割り当ての詳細は、“システム管理ガイド”の“UNIX® のユーザ、グループおよび権限”の章を参照してください。

3.1.3 必要な依存関係のインストール

インストール・プロセスの一環として、InterSystems IRIS インストーラはシステムに必要な依存関係があるかどうかを確認します。システムにいずれかの依存関係がない場合、インストールは失敗し、インストーラを再実行する前にインストールすべき依存関係を指定するメッセージが表示されます。

prechecker オプション (例えば、`# irisinstall --prechecker`) を指定して **irisinstall** を実行することで、インストールとは別に要件チェッカーを実行できます。これは、InterSystems IRIS の依存関係を変更またはアップグレードして、変更後も要件が満たされることを確認する場合に便利です。

要件チェッカーは常に、インスタンスの起動中に実行されます。要件が満たされない場合、起動は失敗します。

注釈 macOS では、バージョン 2022.1 以前の InterSystems IRIS バージョンのインストーラに要件チェッカーはありませんが、必要な依存関係は存在します。

macOS 上で InterSystems IRIS をインストールまたは実行する際は、Homebrew を介して `openssl@1.1` をインストールする必要があります (<https://formulae.brew.sh/formula/openssl@1.1>)。

macOS 上の InterSystems IRIS 内で組み込み Python を使用するには、`python@3.9` もインストールする必要があります (<https://formulae.brew.sh/formula/python@3.9>)。

3.1.4 構成方法の選択

インストールする前に InterSystems IRIS を構成する方法はいくつかあります。この方法は、自動導入を検討する場合に特に便利です。インストールの完了後に、設定を調整する必要が少なくなるためです。

InterSystems IRIS をインストールする前に、以下のいずれかの方法を使用するかどうかを検討します。

- ・ **構成マージ**では、マージ・ファイルで InterSystems IRIS 構成を定義できます。マージ・ファイルは必要な設定を指定します。ユーザ、ネームスペース、データベースの作成などの管理アクションを実行することもできます。マージ・ファイルは、インストール中またはインストール後に適用できます。
- ・ **インストール・マニフェスト**は特定の InterSystems IRIS 構成を記述し、`irisinstall` または `irisinstall_silent` コマンドで 사용할 수 있습니다。
- ・ “InterSystems IRIS ディストリビューションへの **Unix® インストール・パッケージの追加**” では、InterSystems IRIS ディストリビューションに新規パッケージを追加する方法について説明しています。

3.2 InterSystems IRIS のインストール

このセクションでは、UNIX® プラットフォームに InterSystems IRIS をインストールする手順について説明します。InterSystems IRIS の手動インストールを実行するには、以下のセクションで説明するように 2 つの方法があります。

- ・ **InterSystems IRIS の標準インストール手順** – `irisinstall` を使用して、基本の InterSystems IRIS インストールを実行します。さまざまなインストール・オプションを選択可能で、InterSystems IRIS サーバと共にインストールする項目をインタラクティブに選択できます。
- ・ **クライアントのみの InterSystems IRIS のインストール手順** – InterSystems IRIS サーバをインストールせずに、`irisinstall_client` を使用して適切なクライアント・コンポーネントをインストールします。

一般に、InterSystems IRIS インストールは root 特権を使用して実行します。非 root ユーザとしてインストールを実行することもできますが、この場合、“**非 root ユーザとしてのインストール**” に説明されている制限があります。

3.2.1 InterSystems IRIS の標準インストール手順

InterSystems IRIS の標準インストールは、複数のモジュール・パッケージ・スクリプトから構成されます。これらのスクリプトでは、それまでの手順に対する入力、システム環境、および既存のインスタンスをアップグレードしているかどうかなどの条件に応じて、情報の入力が求められます。インストールの第 1 段階では、このインストールについて収集された情報がすべてパラメータ・ファイルに保存されます。その後、実際のインストールが開始される前に、インストールの詳細を確認します。最終段階では、インスタンスのセットアップなど、正常終了したインストールに付随する操作を実行します。

インストール・スクリプト `irisinstall` は、以下を実行します。

- ・ InterSystems IRIS システム・マネージャ・データベースをインストールします。
- ・ インストール・モードで InterSystems IRIS を起動します。
- ・ InterSystems IRIS システム・マネージャ・グローバルとルーチンをインストールします。
- ・ InterSystems IRIS を終了し、既定の構成ファイル (`iris.cpf`) を使用して再起動します。アップグレード・インストールの場合は、更新後の構成ファイルを使用して再起動されます。

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、インストールは失敗します。

InterSystems IRIS サーバの標準インストールを実行するには、以下のインストール手順に従います。

1. ユーザ ID `root` としてログインします。InterSystems IRIS の標準インストールは、`root` 特権を持つユーザが実行する必要があります。`root` 以外のアカウントでログインしているときでも、`su` (スーパーユーザ) コマンドで `root` に変更できます。

注釈 `root` が使用できない場合は、非 `root` ユーザとして InterSystems IRIS の非標準の制限付きインストールを実行できます。インストール手順を続行する前に、以下の [“InterSystems IRIS の非 root ユーザとしてのインストール”](#) のセクションで詳細を参照してください。

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。

```
# /<pathname>/irisinstall
```

ここで、`pathname` はインストール・キットの場所で、この章の [“インストール・キットの解凍”](#) のセクションで説明されているように、通常はキットの抽出先の一時ディレクトリです。

3. インストール・スクリプトは、ご使用のシステム・タイプを識別し、インストール・タイプあるいは配布メディアに対する検証を行います。ご使用のシステムが、例えば 32 ビットと 64 ビットなど、複数のタイプをサポートしている場合、あるいは、インストール・スクリプトがシステム・タイプを識別できない場合は、別の質問が表示されます。これには、インストール・キット名の末尾の文字列の形式で“プラットフォーム名”を尋ねる質問などが含まれます。目的のシステム・タイプが配布メディア上のものと適合しない場合、インストールが停止します。この問題を解決するには、[インターシステムズのサポート窓口](#)にお問い合わせください。
4. スクリプトにより、ホスト上の既存の InterSystems IRIS インスタンスが一覧表示されます。
5. **[インスタンス名の入力]** プロンプトで、英数字、アンダースコア、ダッシュのみを使用してインスタンス名を入力します。入力した名前を持つインスタンスが既に存在する場合、現在のインスタンスを更新するかどうかを尋ねられます。入力した名前を持つインスタンスが存在しない場合は、新しいインスタンスを作成するかどうか、作成するのであればインストール・ディレクトリを指定するように指示されます。指定したディレクトリが存在しない場合は、ディレクトリを作成するかどうかを尋ねられます。これらの質問に対する既定の回答は `Yes` です。“InterSystems IRIS のインストール準備”の章の [“インストール・ディレクトリ”](#) のセクションで、インストール・ディレクトリの選択に関する重要な情報を必ず確認してください。

注釈 インストール・キットと同じ InterSystems IRIS バージョンの既存のインスタンスを選択する場合、インストールはアップグレードと見なされます。また、次のステップで説明されているように、Custom 選択を使用して、インストールされたクライアント・コンポーネントと特定の設定を変更できます。

InterSystems IRIS レジストリ・ディレクトリ (`/usr/local/etc/irissys`) が常に InterSystems IRIS インストール・ディレクトリと共に作成されます。

6. 次に、以下の選択肢からインストール・タイプを選択します。

```
Select installation type.
  1) Development - Install IRIS server and all language bindings
  2) Server only - Install IRIS server
  3) Custom - Choose components to install
Setup type <1>?
```

注釈 Web ゲートウェイをインストールする際に、お使いのシステムに CSP ゲートウェイが既にインストールされている場合は、インストーラにより CSP ゲートウェイが Web ゲートウェイに自動的にアップグレードされます。詳細は、“[既存の CSP ゲートウェイ](#)”を参照してください。

このドキュメントの“InterSystems IRIS のインストール準備”の章の“[セットアップ・タイプ](#)”のセクションを参照して、適切なインストール・タイプを決定します。新しいインスタンスの既定は、InterSystems IRIS の完全なインストールを含む Development です。既存のインスタンスをアップグレードしている場合、既定は元々インスタンスをインストールするために使用されたオプションです。

注釈 Custom オプションを選択する場合は、“[InterSystems IRIS のカスタム・インストール](#)”で追加のインストール手順を参照してください。

7. 次に、InterSystems IRIS を [8 ビット・サポート](#)または [Unicode 文字のサポート](#)のどちらでインストールするかを指定します。アップグレード時に、8 ビットから Unicode に変換することはできますが、逆には変換できません。
8. 新規インスタンスの場合、次にセキュリティ設定を指定する必要があります。(アップグレード時には、セキュリティ設定を変更できません。)

最初に、インターシステムズの初期セキュリティ設定の制限を指定します。[最小](1)、[通常](2)、[ロック・ダウン](3)から選択できます。これらのセキュリティ設定の詳細は、“[インターシステムズの初期セキュリティ設定](#)”を参照してください。

[最小]は、InterSystems IRIS インストールにのみ使用できます。これを選択した場合、次の手順はスキップできます。インスタンスの所有者はインストーラによって root として設定されます。

重要 カスタム・インストールでなければ選択できないセキュリティ設定もあります。詳細は、“[InterSystems IRIS カスタム・インストール](#)”を参照してください。

9. 2 または 3 を入力した場合は、その他の情報を入力するよう求められます。
 - a. インスタンス所有者 – InterSystems IRIS プロセスの実行に使用するアカウントのユーザ名を入力します。このアカウントに関する詳細は、“[所有者およびグループの決定](#)”を参照してください。InterSystems IRIS のインストールが完了すると、インスタンスの所有者は変更できなくなります。
 - b. インスタンス所有者のパスワード – 前のプロンプトで入力したユーザ名のパスワードを入力し、確認のために再度入力します。このユーザに対して、%All ロールを持つ InterSystems IRIS 特権ユーザ・アカウントが作成されます。

このパスワードは、InterSystems IRIS 特権ユーザ・アカウントだけでなく、_SYSTEM、Admin、および SuperUser の事前定義の各ユーザ・アカウントにも使用されます。これらの事前定義ユーザの詳細は、“[事前定義のユーザ・アカウント](#)”を参照してください。
 - c. CSPSystem 事前定義のユーザのパスワード。

注釈 パスワードは、“[初期のユーザ・セキュリティ設定](#)”のテーブルに示した条件を満たしている必要があります。この手順で入力したパスワードには、スペース、タブ、またはバックスラッシュの各文字は使用できません。インストーラはそのようなパスワードを受け付けません

10. インストールのこの時点で、InterSystems IRIS を起動および停止できるグループを指定するように求められます。これらの特権を持つことができるのは 1 グループのみで、このマシンで有効なグループでなければなりません。詳細は、“[所有者およびグループの決定](#)”を参照してください。既存のグループ名またはグループ ID 番号を入力します。InterSystems IRIS は、処理を続行する前にそのグループが実際に存在するかを確認します。

11. 最後に、新しいインスタンスの場合、またはアップグレード時に既存のインスタンスの **mgr** ディレクトリにある **iris.key** ファイルをスクリプトが検出しない場合、ライセンス・キー・ファイルの入力を求めるプロンプトが表示されます。有効なキーを指定する場合、インストール中にライセンスは自動的に有効化されて、ライセンス・キーはインスタンスの **mgr** ディレクトリにコピーされるので、これ以上の有効化手順は不要です。ライセンス・キーを指定しない場合、インストールに続いて、ライセンス・キーを有効化できます。ライセンス、ライセンス・キーおよび有効化についての詳細は、“システム管理ガイド”の“InterSystems IRIS ライセンスの管理”の章にある“[ライセンス・キーの有効化](#)”を参照してください。

注釈 macOS では、**irisdb** のネットワーク接続に関するプロンプトが表示されることがあります。表示された場合は、**[許可]** を選択します。

12. インストール・オプションを確認し、Enter キーを押してインストールを続行します。**[Yes]** を選択すると、ファイルのコピーが開始されます。

インストールが完了すると、InterSystems IRIS システムを管理する管理ポータル URL が示されます。詳細は、“システム管理ガイド”の“[管理ポータルの使用](#)”の章を参照してください。

“[インストール後のタスク](#)”のセクションに進んで、インストール・プロセスを完了します。

3.2.1.1 InterSystems IRIS のカスタム・インストール

カスタム・インストールを選択した場合、いくつかのコンポーネントの[インストール手順](#)に関する追加の質問に回答する必要があります。既定値は角かっこで囲まれ、これに続いて疑問符(?)が表示されます。既定値をそのまま使用するには、Enter を押します。

- ・ [SuperServer port number] と [WebServer port number] – ポート番号が自動的に割り当てられるようにするか、または独自のポート番号を入力できます。アップグレード・インストールではこの選択肢は表示されず、元のインスタンスのポート番号がそのまま使用されます。詳細は、このドキュメントの“InterSystems IRIS のインストール準備”の章の“[ポート番号](#)”のセクションを参照してください。
- ・ [Web ゲートウェイおよび外部 Web サーバ] – サポートされている外部 Web サーバを使用するように Web ゲートウェイを構成できます。InterSystems IRIS のインストールが完了したら、**[Yes]** を選択します。次に、Web サーバの構成に関する追加の質問に回答して Web ゲートウェイを構成します。
- ・ [追加のセキュリティ・オプション] – 初期のセキュリティ設定として [通常] または [ロック・ダウン] を選択した場合は、追加のセキュリティ・オプションを構成できます (最小のインストールでは、既定の ID が使用されています)。
 - [InterSystems IRIS の実効グループ] – InterSystems IRIS 内部の実効グループ ID。また、インストール環境にあるすべてのファイルと実行可能ファイルに対する特権をすべて保持しています。最大限のセキュリティ対策として、このグループには実際のユーザは含めないでください。(既定値は **irisusr** です。)
 - [InterSystems IRIS スーパーサーバの実効ユーザ] – スーパーサーバとジョブ・サーバによって開始されたプロセスの実効ユーザ ID です。ここでも最大限のセキュリティを実現するために、実際のユーザにはこのユーザ ID を使用しないようにします。(既定値は **irisusr** です。)

追加情報は、“[所有者およびグループの決定](#)”を参照してください。

- ・ IntegratedML – InterSystems IRIS IntegratedML なしでインストールすることもできます。IntegratedML の詳細は、“[IntegratedML の使用](#)”を参照してください。
- ・ [クライアント・コンポーネント] – 1 つのクライアント・コンポーネントをインストールできます。このクライアント・コンポーネントは、一部のプラットフォームではサポートされない可能性があります。

```
Client component selection
[*]      1) IRIS shared library support (callin)
          +) Select all

Enter the number of each component you wish to install.
Enter the number of an already selected component to deselect it.
Multiple selections can be separated by spaces.
Enter a blank line to continue.
```

すべてのコンポーネントを選択するには、プラス記号 (+) を使用します。

注釈 UNIX® InterSystems IRIS キットでは常にクライアント・コンポーネントがインストールされます。

選択リスト内の特定のクライアント・コンポーネントに関する詳細は、以下のガイドを参照してください。

- ・ [InterSystems ODBC ドライバの使用法](#)
- ・ [コールイン API の使用法](#)

3.2.2 クライアントのみの InterSystems IRIS のインストール手順

InterSystems IRIS ディストリビューションには、InterSystems IRIS のクライアントのみバージョンをインストールするためのスクリプトが用意されています。インストール手順は極めてシンプルです。root としてインストールする必要はありません。このインストールからのファイルは、インストールしているユーザのユーザ許可とグループ許可を持ちます。InterSystems IRIS クライアントのインストールを実行するには、以下の手順に従って操作します。

1. インストール・ファイルの最上位にある `irisinstall_client` スクリプトを実行し、インストール手順を開始します。

```
# /<pathname>/irisinstall
```

ここで、pathname はインストール・キットの場所で、この章の“[インストール・キットの解凍](#)”のセクションで説明されているように、通常はキットの抽出先の一時ディレクトリです。

2. インストール・スクリプトは、ご使用のシステム・タイプを識別し、インストール・タイプあるいは配布メディアに対する検証を行います。ご使用のシステムが、例えば 32 ビットと 64 ビットなど、複数のタイプをサポートしている場合、あるいは、インストール・スクリプトがシステム・タイプを識別できない場合は、別の質問が表示されます。目的のシステム・タイプが配布メディア上のものと適合しない場合、インストールが停止します。この問題を解決するには、[インターシステムズのサポート窓口](#)にお問い合わせください。
3. [クライアント・コンポーネントのインストール先ディレクトリの入力] のプロンプトが表示されたら、インストール・ディレクトリを指定します。指定したディレクトリが存在しない場合は、ディレクトリを作成するかどうかを尋ねられます。既定の回答は Yes です。“InterSystems IRIS のインストール準備”の章の“[インストール・ディレクトリ](#)”のセクションで、インストール・ディレクトリの選択に関する重要な情報を必ず確認してください。Enter キーを押して、インストールを続行します。

注釈 InterSystems IRIS レジストリ・ディレクトリ (`/usr/local/etc/irissys`) が常に InterSystems IRIS インストール・ディレクトリと共に作成されます。

4. 使用可能なクライアント・コンポーネント・オプションから選択します。同じマシンに InterSystems IRIS サーバを必要とするコンポーネントは、このリストには表示されません。例を以下に示します。

```
Client component selection

[*]      1) ODBC client
[*]      2) C++ binding
[*]      3) C++ SDK
          +) Select all

Enter the number of each component you wish to install.
Enter the number of an already selected component to deselect it.
Multiple selections can be separated by spaces.
Enter a blank line to continue.
```

すべてのコンポーネントを選択するには、プラス記号 (+) を使用します。

共有ライブラリ・サポートや Light C++ バインディングはサーバのインストールを必要とするため、クライアントのみコンポーネントのリストには含まれません。

サーバ環境でクライアント・コンポーネントを更新するには、このスクリプトは使用できません。代わりに、`irisinstall` スクリプトを使用してください。

選択リスト内の具体的なクライアント・コンポーネントに関する詳細は、“InterSystems IRIS 言語バインディング” セットの以下のガイドを参照してください。

- ・ [InterSystems ODBC ドライバの使用法](#)

“インストール後のタスク” のセクションに進んで、インストール・プロセスを完了します。

3.2.3 非 root ユーザとしてのインストール

プロダクション環境に InterSystems IRIS をインストールする場合、root 特権を使用することが推奨されます。root 特権なしで InterSystems IRIS インストールを実行することは可能ですが、このインストールにはいくつかの制限があります。以下のセクションで、これらの制限について、および標準の InterSystems IRIS インストールとの違いについて説明します。

- ・ [InterSystems IRIS インストールで root を使用する理由](#)
- ・ [非 root インストールの制限事項](#)
- ・ [非 root インストールの相違点](#)

重要 root 特権は、InterSystems IRIS をインストールするときのみ使用する必要があります。インストールが完了したら、すべてのユーザは非 root 特権を使用して InterSystems IRIS とやり取りする必要があります。

3.2.3.1 InterSystems IRIS インストールで root を使用する理由

通常、InterSystems IRIS は root を使用してインストールされ、非 root 特権を使用して操作されます。いくつかの機能には root アクセス権が必要ですが、ほとんどのプロセスはインストール時に指定したユーザまたはグループとして実行されます。これらのユーザとグループの目的、および root をどのように使用するかについては、“[UNIX のユーザ、グループおよび権限](#)” で説明します。

root 特権を使用する InterSystems IRIS プロセスには次のようなものがあります。

- ・ オペレーティング・システムでネットワーク設定を変更する有効なユーザ ID (UID) として root を使用する、[仮想 IP](#) プロセス。
- ・ 有効な UID としてインスタンス所有者を使用し、実際の UID として root を使用する制御プロセス。実際の UID は、開始時に[ラージ・ページ](#)を取得し、他の InterSystems IRIS プロセスと通信するために使用します。
- ・ 有効な UID として root を使用する開始実行可能ファイル。

InterSystems IRIS を root としてインストールすると、root 特権を持つユーザのみがファイル構造を変更または置換できるため、セキュリティも強化されます。

3.2.3.2 非 root インストールの制限事項

InterSystems IRIS の非 root インストールはサポートされていますが、この方法でインストールされたインスタンスでは、使用できない機能がいくつかあります。

- ・ ヒュージ・ページは使用できません。

- 外部 Web サーバを使用するように Web ゲートウェイを構成することはできません。
- ミラー仮想 IP は使用できません。

注釈 ネットワーク・ロード・バランサや Web ゲートウェイを使用するなど、ネットワーク・トラフィックをルーティングする別の方法については、“[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](#)”を参照してください。

- インストール時に InterSystems IRIS を開始および停止できるインスタンス所有者およびグループを指定するオプションはありません (“[所有者およびグループの決定](#)”を参照)。
- グループ・アクセスは存在しません。レジストリを含むすべてのインスタンス・ファイルはインストールするユーザの所有となり、このユーザによってのみ読み込み、書き込み、および実行が可能となります。

例えば、標準インスタンスが以下であるとしてします。

```
-rws--x--- 5 root develop 43282 Aug 28 07:52 irismgr
-r-x--s--x 1 <nonroot-user> irisusr 23058 Aug 28 07:52 irisuxsession
```

非 root インスタンスは以下ようになります。

```
-rwx----- 5 <installing-user> develop 43282 Aug 28 07:52 irismgr
-r-x----- 1 <installing-user> develop 23058 Aug 28 07:52 irisuxsession
```

レジストリは IRISYS で指定するディレクトリに配置され、そのレジストリに非 root インスタンスがあります (iris の実行可能ファイルも、このディレクトリにあります)。この非 root レジストリには、非 root のインスタンスのみが存在します。非 root レジストリから root によりインストールされたインスタンスへのアクセスはいずれも失敗します。反対に、非 root のインスタンスを root レジストリに定義することはできますが、所有者以外のユーザによるこのインスタンスへのアクセスはいずれも失敗します。

注釈 レジストリは、NFS ディレクトリではなく、インスタンスがインストールされたマシンのローカル・ディレクトリに配置することをお勧めします。標準の場所 `/usr/local/etc` がそのようなディレクトリとなります。

3.2.3.3 非 root インストールの相違点

上述した機能の制限のほかに、InterSystems IRIS の root インストールと非 root インストールにはいくつかの明確な違いがあります。

- IRISYS 環境変数は、インストールするユーザによって書き込み可能な既存ディレクトリとして定義する必要があり、インストール中およびすべてのインスタンス操作中に存在している必要があります。
- ISCAgent は、IRISYS で指定されたディレクトリにインストールされます。

注釈 非 root インスタンスの ISCAgent の開始の詳細は、“[高可用性ガイド](#)”の“[ミラーリング](#)”の章にある“[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](#)”を参照してください。

- インストールしているユーザのアカウントのみが、InterSystems IRIS インスタンスにアクセスして操作できます。
- InterSystems IRIS の実行可能ファイルおよびプロセスはすべて、インストールするユーザとして実行されます。

3.3 InterSystems IRIS の自動インストール

`irisinstall_silent` スクリプトを使用すると、InterSystems IRIS インスタンスの自動インストールを各自のシステムで実行できます。標準のインストール操作では、必要となる仕様や選択肢をプロンプトでのユーザ応答形式で取得するのに対し、

自動操作では、`irisinstall_silent` コマンド行で指定された構成パラメータやパッケージからこの情報を取得します。各指定パッケージは InterSystems IRIS コンポーネントを表しています。各コンポーネントのインストール・スクリプトは、`irisinstall_silent` スクリプトを含むディレクトリの下に `packages` ディレクトリにあります。

コマンド行 `irisinstall_silent` の一般的な形式は、以下のように、環境変数の設定によるインストール・パラメータの定義の後にコマンド自体が続きます。

```
sudo ISC_PACKAGE_INSTANCENAME="<instancename>"
ISC_PACKAGE_INSTALLDIR="<installdir>"
ISC_PACKAGE_PLATFORM="<platform>" ISC_PACKAGE_UNICODE="Y" | "N"
ISC_PACKAGE_INITIAL_SECURITY="Minimal" | "Normal" | "Locked Down"
ISC_PACKAGE_MGRUSER="<instanceowner>" ISC_PACKAGE_MGRGROUP="<group>"
ISC_PACKAGE_USER_PASSWORD="<pwd>" ISC_PACKAGE_CSPSYSTEM_PASSWORD="<pwd>"
ISC_PACKAGE_IRISUSER="<user>" ISC_PACKAGE_IRISGROUP="<group>"
ISC_PACKAGE_CLIENT_COMPONENTS="<component1> <component2> ..."
ISC_PACKAGE_STARTIRIS="Y" | "N"
./irisinstall_silent [<pkg> ...]
```

注釈 ルート権限なしで InterSystems IRIS をインストールする場合、詳細は、[“非 root ユーザとしてのインストール”](#) のセクションを参照してください。

このセクションでは、以下のトピックについて説明します。

- ・ [自動インストール・パラメータ](#)
- ・ [自動インストール・パッケージ](#)
- ・ [自動インストールの例](#)

3.3.1 自動インストール・パラメータ

以下のテーブルでは、自動インストールの `irisinstall_silent` スクリプトで使用するパラメータについて説明しています。

テーブル 3-1: 自動インストール・パラメータ

パラメータ	説明
<code>ISC_CPF_MERGE_FILE="<i><location></i>"</code>	<p>構成のマージを実行する際に、構成マージ・ファイルの場所を指定します。</p> <p>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。</p>
<code>ISC_PACKAGE_INSTANCENAME="<i><instancename></i>"</code> (必須)	<p>インストールまたはアップグレードするインスタンスの名前を指定します。インスタンスが存在しない場合は、新規インストールとなります。インスタンスが存在する場合は、アップグレードとなります。以下に例を示します。</p> <pre>ISC_PACKAGE_INSTANCENAME="MyIris"</pre> <p>注釈 新規インストールの場合は <code>ISC_PACKAGE_INSTALLDIR</code> パラメータが必要です。</p>

パラメータ	説明
ISC_PACKAGE_INSTALLDIR=" <i><installdir></i> " (新規インスタンスで必要)	<p>新しいインスタンスをインストールするインストール・ディレクトリを指定します。以下に例を示します。</p> <p>ISC_PACKAGE_INSTALLDIR="/opt/MyIris"</p> <p>指定されたディレクトリが存在しない場合、インストールによりそのディレクトリの作成が試行されます。インストールをアップグレードする場合、このパラメータは無視されます。</p> <p>注釈 インストール・ディレクトリの選択の詳細は、このドキュメントの“インストール・ディレクトリ”のセクションを参照してください。</p>
ISC_PACKAGE_INSTALL_INTEGRATEDML="Y" "N"	<p>InterSystems IntegratedML をインストールするかどうかを指定します。</p> <p>IntegratedML の詳細は、“IntegratedML の使用”を参照してください。</p> <p>注釈 既定では、Y に設定されます。</p>
ISC_PACKAGE_UNICODE="Y" "N" (オプション)	<p>UNICODE インストールかどうかを指定します。有効値は、Y または N です。詳細は、“文字幅設定”を参照してください。</p>
ISC_PACKAGE_INITIAL_SECURITY="Minimal" "Normal" "LockedDown" (オプション)	<p>インストールの初期セキュリティ設定を指定します。有効値は、“Minimal”、“Normal”、または“LockedDown”です。</p> <p>注釈 “Minimal” は、InterSystems IRIS インストールにのみ使用できます。</p> <p>“Normal”、または“LockedDown”に設定する場合、ISC_PACKAGE_USER_PASSWORD が必要です。</p>
ISC_PACKAGE_MGRUSER=" <i><user></i> " (オプション)	<p>インストールの所有者のログイン名を指定します。以下はその例です。</p> <p>ISC_PACKAGE_MGRUSER="jcsmith"</p> <p>注釈 既定では、インスタンスをインストールするユーザのユーザ名に設定されます。</p> <p>セキュリティ・レベルが“Minimal”の場合、このパラメータは無視され、ISC_PACKAGE_MGRUSER は“root”に設定されます。</p>

パラメータ	説明
ISC_PACKAGE_MGRGROUP=" <i><group></i> " (オプション)	<p>インスタンスの開始および停止を許可されるグループを指定します。以下はその例です。</p> <p>ISC_PACKAGE_INITIAL_MGRGROUP="irisusr"</p> <p>注釈 既定では、インスタンスをインストールするユーザのグループが設定されます。</p>
ISC_PACKAGE_USER_PASSWORD=" <i><password></i> " (安全なインスタンスのインストールの場合、必須)	<p>通常のセキュリティまたはロック・ダウン・セキュリティでインスタンスに必要なパスワードを指定します。</p> <p>注釈 セキュリティ・レベルが "Minimal" の場合、このパラメータは無視されます。</p> <p>セキュリティ・レベルが "Normal"、または "LockedDown" で、このパラメータを指定しない場合、インストールは失敗し、エラーがスローされます。</p>
ISC_PACKAGE_CSPSYSTEM_PASSWORD=" <i><password></i> "	<p>CSPSystem ユーザのパスワードを指定します。</p> <p>注釈 セキュリティ・レベルが "Minimal" の場合、このパラメータは無視されます。</p> <p>セキュリティ・レベルが "Normal" または "LockedDown" で、このパラメータを指定しない場合、ISC_PACKAGE_USER_PASSWORD の値が使用されます。</p>
ISC_PACKAGE_IRISUSER=" <i><user></i> " (オプション)	<p>InterSystems IRIS スーパーサーバの実効ユーザを指定します。</p> <p>注釈 既定では、irisusr に設定されます。</p> <p>セキュリティ・レベルが "Minimal" の場合、このパラメータは無視され、既定の設定となります。</p>
ISC_PACKAGE_IRISGROUP=" <i><group></i> " (オプション)	<p>InterSystems IRIS プロセスの実効ユーザを指定します。</p> <p>注釈 既定では、irisusr に設定されます。</p> <p>セキュリティ・レベルが "Minimal" の場合、このパラメータは無視され、既定の設定となります。</p>

パラメータ	説明
ISC_PACKAGE_CLIENT_COMPONENTS=" <i><component1></i> <i><component2></i> ..." (オプション)	<p>client_components パッケージからインストールされるクライアント・バインディングを指定します (“自動インストール・パッケージ” を参照)。</p> <p>注釈 既定では、すべてのクライアント・バインディングがインストールされます。指定するコンポーネント (バインディング) はスペースで区切る必要があります。使用可能なコンポーネントの一覧は package/client_components/manifest.isc にあります。インストールでは指定したコンポーネントを検証して、個々のシステムに存在しない、またはサポートされていないコンポーネントを削除します。</p>
ISC_PACKAGE_WEB_CONFIGURE="Y" "N" (オプション)	<p>外部 Web サーバ用に Web ゲートウェイを構成するかどうかを指定します。</p> <p>注釈 既定では、これは N に設定されます (外部 Web サーバ用にゲートウェイを構成しません)。</p>
ISC_PACKAGE_WEB_SERVERTYPE="Apache" "SunOne" "None" (オプション)	<p>Web ゲートウェイで使用する既存の Web サーバのタイプ。例 : ISC_PACKAGE_WEB_SERVERTYPE="Apache"</p> <p>注釈 既定では、これは None に設定されます。</p>
ISC_PACKAGE_WEB_APACHE_VERSION=2.2 2.4 (オプション、ISC_PACKAGE_WEB_SERVERTYPE="Apache" と共に使用)	<p>Apache Web サーバのバージョン。</p> <p>注釈 既定では、バージョンは自動検出されません。</p>
ISC_PACKAGE_WEB_APACHE_USER=" <i><username></i> " (オプション、ISC_PACKAGE_WEB_SERVERTYPE="Apache" と共に使用)	<p>Apache Web サーバのユーザ名。</p> <p>注釈 既定では、ユーザ名は自動検出されません。</p>
ISC_PACKAGE_WEB_APACHE_CONF=" <i><path_to_httpd.conf></i> " (オプション、ISC_PACKAGE_WEB_SERVERTYPE="Apache" と共に使用)	<p>Apache Web サーバの構成ファイルの場所 (例 : /etc/httpd/conf/httpd.conf)</p> <p>注釈 既定では、インストールにより、複数ある標準の場所のいずれかのファイルの自動検出が試行されます。 ISC_PACKAGE_WEB_SERVERTYPE="Apache" および httpd.conf の場所が特定されない場合、インストールはエラーで終了します。</p>

パラメータ	説明
ISC_PACKAGE_WEB_SUNONE_DIR=" <i><sunone_installation_dir></i> " (オプション、ISC_PACKAGE_WEB_SERVERTYPE="SunOne" と共に使用)	<p>SunOne Web サーバのインストール・ディレクトリ。以下に例を示します。</p> <pre>/usr/netscape/server4/httpd-production</pre> <p>注釈 既定では、インストール・ディレクトリは自動検出されます。</p>
ISC_PACKAGE_WEB_SUNONE_NUMBER=" <i><number></i> " (オプション、ISC_PACKAGE_WEB_SERVERTYPE="SunOne" と共に使用)	<p>構成する SunOne Web サーバの数 (複数のサーバがインストールされている場合)。</p> <p>注釈 既定値は 1 です。</p>
ISC_PACKAGE_WEB_GATEWAY_DIR=" <i><web_gateway_directory></i> " (オプション、Web ゲートウェイの新規インストールでのみ)	<p>Web ゲートウェイ・ファイルが保存されるディレクトリ。</p> <p>注釈 既定のディレクトリは /opt/webgateway_ です。</p>
ISC_PACKAGE_STARTIRIS="Y" "N" (オプション)	<p>インストール後に、インストールした InterSystems IRIS インスタンスを開始するかどうかを指定します。</p> <p>注釈 既定では、Y に設定されていて、インスタンスを開始します。</p>
ISC_INSTALLER_MANIFEST=" <i><location></i> "	<p>インストール・マニフェストでインストールする際、エクスポートされたマニフェスト・クラスの場所を指定します。</p> <p>注釈 詳細は、付録 “インストール・マニフェストの作成および使用” の “マニフェストの使用” を参照してください。</p>
ISC_INSTALLER_PARAMETERS=" <i><var>=<value>, <var>=<value> ...</i> "	<p>インストール・マニフェストでインストールする際、変数の名前と値の組み合わせを指定します。</p>
ISC_INSTALLER_LOGFILE=" <i><filename></i> "	<p>インストール・マニフェストでインストールする際、ログ・ファイル名を指定します。</p>
ISC_INSTALLER_LOGLEVEL=" <i><level></i> "	<p>インストール・マニフェストでインストールする際、-1 (“なし”) から 3 (“詳細”) までの数値でログ・レベルを指定します。</p>
ISC_PACKAGE_SUPERSEVER_PORT=" <i><port_number></i> " (オプション)	<p>インストールされるインスタンスで使用されるスーパーサーバ・ポートを指定します。</p> <p>注釈 既定では、このポートは 1972 に設定されます (可能な場合)。それ以外の場合、ポートは 51773、またはこれ以降の使用可能な最初のポート番号に設定されます。</p>

パラメータ	説明
ISC_PACKAGE_USER_DATABASE=Y N	オプションのユーザ・データベースをインストールするかどうかを指定します。既定では、Y に設定されます。
ISC_PACKAGE_WEBSERVER_PORT=" <port_number> " (オプション)	<p>インストールされるインスタンスで使用される Web サーバ・ポートを指定します。</p> <p>注釈 既定では、このポートは自動的に検出されます。最初は 52773 で、インストールされた InterSystems IRIS のインスタンスごとに 1 ずつ増加します。</p>

3.3.2 自動インストール・パッケージ

各コンポーネントのインストール・スクリプトは、`irisinstall_silent` スクリプトを含んでいるディレクトリの配下の `packages` ディレクトリに含まれています。各パッケージは、それぞれのディレクトリ内に配置され、各パッケージのディレクトリには、そのディレクトリ内パッケージの前提条件となるパッケージが定義された `manifest.isc` ファイルが含まれています。

`standard_install` パッケージは、すべてのパッケージがインストールされるサーバ・インストールの開始ポイントです。カスタム・パッケージを定義するには、以下のように `standard_install` パッケージの `manifest.isc` ファイルをテンプレートとして使用します。

1. `standard_install` ディレクトリを新しいディレクトリにコピーします。

例えば、`custom_install` という名前のディレクトリにコピーします。新しいディレクトリにコピーされた `manifest.isc` ファイルの初期の状態は次のようになります。

```
#This is the target for a standard (non-client-only) install
package: standard_install
prerequisite: install_mode
prerequisite: database_server
prerequisite: databases
prerequisite: gadget
prerequisite: fop
prerequisite: renderserver
prerequisite: printserver
prerequisite: exceleporter
prerequisite: callin_components
prerequisite: client_components
prerequisite: addenda
prerequisite: install_confirmation
prerequisite: copyright
```

2. 新しいディレクトリで、以下のように `manifest.isc` ファイルを変更します。

- ・ パッケージ・キーをディレクトリ名の値に設定します (必須)。
- ・ カスタム・インストールの前提条件ファイルを追加または削除します。

例えば、以下の `manifest.isc` ファイルでは、パッケージ・キーの値がディレクトリ名 (`custom_install`) に一致するように変更されています。


```
#This is the target for a custom (non-client-only) install
package: custom_install
prerequisite: install_mode
prerequisite: database_server
prerequisite: gadget
prerequisite: fop
prerequisite: renderserver
prerequisite: printserver
prerequisite: excelexporter
prerequisite: callin_components
prerequisite: client_components
prerequisite: addenda
prerequisite: install_confirmation
prerequisite: copyright
```

次に、自動インストールを実行するときに新しいカスタム・パッケージを指定できます。例えば、`sudo ISC_PACKAGE_INSTANCENAME="MyIris" ./irisinstall_silent custom_install` のように指定します。

注釈 独自の UNIX® インストール・パッケージを作成して、それらを InterSystems IRIS ディストリビューションに追加する方法の詳細は、付録 [“InterSystems IRIS ディストリビューションへの UNIX® インストール・パッケージの追加”](#) を参照してください。

3.3.3 自動インストールの例

以下の例は、`irisinstall_silent` スクリプトを使用して、InterSystems IRIS を UNIX® プラットフォームに自動インストールする方法について示しています。

例 1

この例では、すべてのパッケージが最低限のセキュリティでインストールされます。

```
sudo ISC_PACKAGE_INSTANCENAME="MyIris" ISC_PACKAGE_INSTALLDIR="/opt/MyIris1" ./irisinstall_silent
```

MyIris インスタンスが既に存在する場合は、これがアップグレードされます。存在しない場合は、`/opt/MyIris1` ディレクトリにインストールされます。

例 2

この例では、**MyIris** という名のインスタンスが存在しない場合に、インストールが中止され、エラーがスローされます。

```
sudo ISC_PACKAGE_INSTANCENAME="MyIris" ./irisinstall_silent
```

例 3

この例では、`database_server` パッケージ、`odbc` パッケージおよび `odbc` クライアント・バインディングのみが最低限のセキュリティでインストールされます。

```
sudo ISC_PACKAGE_INSTANCENAME="MyIris" ISC_PACKAGE_INSTALLDIR="/opt/MyIris2"
ISC_PACKAGE_CLIENT_COMPONENTS="odbc" ./irisinstall_silent database_server odbc
```

3.4 インストール後のタスク

インストール手順の実行が完了したら、以下のタスクを実行します。

- ・ 以下のセクションの説明に従って、[InterSystems IRIS を起動](#)します。
- ・ スタジオを使用した開発を計画している場合は、[“開発用の InterSystems IRIS クライアントの Windows へのインストール”](#) のセクションを参照してください。
- ・ “システム管理ガイド” の “InterSystems IRIS の構成” の章にある [“メモリと開始設定”](#) のセクションの説明に従って、InterSystems IRIS で使用されるシステム・メモリを割り当てます。

- ・ インターシステムズ製品は、インターシステムズ製ではないツールと共に実行したり、そのようなツールとやり取りしたりすることが頻繁にあります。このようなやり取りでもたらされる可能性がある影響に関する重要な情報については、“システム管理ガイド”の付録“[インターシステムズ製品と連係して動作するようにサードパーティ・ソフトウェアを構成する方法](#)”を参照してください。
- ・ 別のデータベースから新しくインストールした InterSystems IRIS インスタンスにデータを移行する必要がある場合は、“データの移行”を参照してください。
- ・ 必要に応じて、“[特別な考慮事項](#)”に記載されているタスクも追加で実行してください。

3.4.1 InterSystems IRIS の起動

ほとんどのインストールでは、インストールが完了すると InterSystems IRIS は自動的に起動します。InterSystems IRIS を起動する必要がある場合は、最初にオペレーティング・システムにログインし、iris コマンドを使用して InterSystems IRIS を起動します。

```
iris start <instname>
```

instname は、インストール時に指定したインスタンス名です。Red Hat Linux では、起動コマンドは以下のとおりです。

```
service <instname> start
```

InterSystems IRIS の起動および停止には、iris コマンドを使用します。この詳細は、“システム管理ガイド”の“[InterSystems IRIS インスタンスの制御](#)”のセクションを参照してください。

注釈 mgr サブディレクトリへのパスを構成するすべての要素に対する許可を、(少なくとも) irisusr グループが読み取りアクセスできるように設定する必要があります。このような許可を設定しないと、インスタンスは完全には起動せず、「`manager_subdirectory (errno 2)`」というメッセージが `messages.log` に記録されます。

InterSystems IRIS が起動したら、iris terminal コマンドを使用して InterSystems IRIS セッションを開始します。“システム管理ガイド”の“InterSystems IRIS の複数インスタンスの使用法”の章にある“[InterSystems IRIS インスタンスの接続](#)”を参照してください。

```
iris terminal <instname> [parameters]
```

instname は、インストール時に指定したインスタンス名です。

詳細は、“システム管理ガイド”の“[UNIX®、Linux、および macOS での InterSystems IRIS 使用法](#)”の章を参照してください。

3.4.2 開発環境のインストール

InterSystems IRIS ではプライベート Apache Web サーバがインストールされるため、管理ポータルにはアクセスできますが、UNIX® 上の開発環境は含まれません。したがって、Windows クライアント上で実行する必要がある[スタジオ](#)か、Visual Studio Code の InterSystems ObjectScript 拡張機能を選択する必要があります。VS Code - ObjectScript の詳細は、<https://intersystems-community.github.io/vscode-objectscript/> を参照してください。

Windows クライアントをインストールするには、このドキュメントの“Microsoft Windows への InterSystems IRIS のインストール”の章の“[クライアント・コンポーネントのみのインストール](#)”のセクションの手順に従います。Windows クライアントがインストールされたら、以下のタスクを実行して、スタジオをインスタンスに接続します。

1. Windows クライアントで、InterSystems IRIS [ランチャー](#)を開きます。
2. [優先接続サーバ]を指して[追加/編集]をクリックし、インストールした InterSystems IRIS インスタンスへの[リモート・サーバ接続を追加](#)します。この接続に適切なポート番号を指定するように注意してください。

3. [リモート システム アクセス]、[スタジオ] の順にポイントし、前の手順で入力した該当する接続名をクリックします。

3.5 特別な考慮事項

このセクションでは、以下のトピックについて説明します。

- ・ [複数の InterSystems IRIS インスタンス](#)
- ・ [InterSystems IRIS のアンインストール](#)
- ・ [macOS での多数の同時プロセスの調整](#)

3.5.1 複数の InterSystems IRIS インスタンス

1 台のマシンに、複数のインスタンスをインストールして、同時に実行できます。それには、InterSystems IRIS の各インスタンスに一意の名前を付け、別々のディレクトリにインストールし、それぞれに異なるポート番号を割り当てます。

詳細は、“システム管理ガイド”の“InterSystems IRIS 複数インスタンスの使用法”の章にある“[複数の InterSystems IRIS インスタンスの構成](#)”のセクションを参照してください。

3.5.2 InterSystems IRIS 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、保持する必要の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。

3.5.3 macOS での多数の同時プロセスの調整

多数のプロセスまたは Telnet ログインを必要とするシステムを実行する場合は、以下の調整を行う必要があります。

1. リモート接続 – pty (擬似ターミナル) 接続の数はシステム全体で 128 までに制限されています。アプリケーションがユーザ・アクセスを Telnet などの pty 使用接続に依存している場合は、この点に注意してください。
2. プロセス数 – 多数のプロセスの実行が必要な場合は、pty 制限に該当しなくても、プロセス数自体に制限があります。
 - ・ システム全体でのプロセス制限 – kern.maxproc および kern.maxprocperuid パラメータの既定値はそれぞれ 532 と 100 に設定されています。これらのパラメータは、次のコマンドを使用して変更できます。

```
administrator$ sudo sysctl -w kern.maxproc=2500
kern.maxproc: 2065 -> 2500
administrator$ sudo sysctl -w kern.maxprocperuid=2500
kern.maxprocperuid: 2000 -> 2500

administrator$ sysctl -a | grep maxproc
kern.maxproc = 2500
kern.maxprocperuid = 2500
```

ただし、2500 が変更不能な絶対上限であることに注意してください。

4

以前のバージョンからのアップグレード

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 Cache® 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。

4.1 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。

4.2 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、“InterSystems IRIS Data Platform 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ます。
- 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。

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
```

4.3 インスタンスのアップグレード

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。

実行する手順は、お使いのオペレーティング・システムによって異なります。

- [Windows でのアップグレード手順](#)
- [UNIX®, Linux、および macOS でのアップグレード手順](#)

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。

4.3.1 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。代わりに、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、このドキュメントの“[Microsoft Windows への InterSystems IRIS のインストール](#)”の章の“[自動アップグレードまたは再インストールの実行](#)”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. 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。

5. アップグレードが完了したら、[完了] をクリックします。
6. `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.

4.3.2 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“InterSystems IRIS 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** ファイルの形式である場合は、まずファイルを解凍する必要があります (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“インストール・キットの解凍”のセクションを参照してください)。
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. 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。

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.
```

4.4 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 사용되는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。

4.4.1 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な要素は、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に、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。

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に、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に、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は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。

4.4.2 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。

4.4.3 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。

4.4.4 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。

4.4.4.1 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。

4.4.4.2 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、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)。

4.4.4.3 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。

4.4.4.4 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。

4.5 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なルーチン・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。

4.6 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。

4.6.1 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。

4.6.2 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。
- ・ 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、このドキュメントの“Microsoft Windows への InterSystems IRIS のインストール”の章にある“[自動インストールの実行](#)”のセクションの説明に従って、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。

4.6.2.1 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。

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")
```

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")
```

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

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()
```


A

インストール・マニフェストの作成および使用

このトピックでは、特定の InterSystems IRIS® Data Platform 構成を記したインストール・マニフェストを作成し、これを使用して InterSystems IRIS インスタンスを構成するコードを生成するための **%Installer** ユーティリティの使用法について説明します。

A.1 インストール・マニフェストの概要

%Installer ユーティリティによりインストール・マニフェストを定義します。これは段階的なインストール・プロセスではなく、特定の InterSystems IRIS 構成を記述および構成するものです。これには、通常インストール中（スーパーサーバ・ポート、オペレーティング・システムなど）に指定した情報を含む変数を使用して、目的の構成を記述した XData ブロックを含むクラスを作成します。また、このクラスにインスタンス構成用のコードを生成するための XData ブロックを使用するメソッドを組み込みます。この付録では、[インストール・マニフェストの例](#)を提供します。この例をコピーして貼り付けて、作業を開始することができます。

マニフェストを定義すると、インストール中、またはターミナル・セッションやコードからマニフェストを呼び出すことができます。マニフェストは **%SYS** ネームスペースで実行する必要があります。

A.2 インストール・マニフェストの作成

このセクションでは、インストール・マニフェストの作成に必要な情報を以下の内容に分けて説明します。

- ・ [マニフェスト・クラス定義](#)
- ・ [マニフェストの一般的なオプション](#)
- ・ [〈Manifest〉 タグ内の変数](#)
- ・ [〈Manifest〉 タグのリスト](#)

A.2.1 マニフェスト・クラス定義

インストール・マニフェストを定義するクラスを作成するには、以下の条件を満たすクラスを作成します（または、以下の[“インストール・マニフェストの例”](#)のセクションの空白のテンプレートから開始します）。

- ・ このクラスには、**%occInclude** インクルード・ファイルを含める必要があります。

- このクラスには、インストールの詳細を指定する XData ブロックを含める必要があります。XData ブロックの名前は、後で引数として使用されます。XData ブロックのルート要素は `<Manifest>` とする必要があります。詳細は、“[<Manifest> タグのリスト](#)”を参照してください。

スタジオでは、XData ブロックの名前の後に `[XMLNamespace = INSTALLER]` を追加すると、XData ブロックの入力に応じてサポートが提供されます。

- このクラスは、名前 XData ブロックを参照する `setup()` メソッドを定義する必要があります (“[インストール・マニフェストの例](#)”のセクションを参照)。

A.2.2 マニフェストの一般的なオプション

このセクションでは、インストール・マニフェストを使用して実行するいくつかの一般的なタスクについて説明します。以下のオプションについて説明します。

- [ネームスペースの定義](#)
- [ユーザとパスワードの追加](#)
- [マニフェスト・ログへの書き込み](#)
- [アップグレード後のタスクの実行](#)

A.2.2.1 ネームスペースの定義

ネームスペースを定義するには、`<Manifest>` タグ内に任意の数の `<Namespace>` タグを追加します。

ネームスペースのデータベースまたはマッピングを定義するには、`<Namespace>` タグの中に `<Configuration>` タグを入れます。ネームスペースに含まれるデータベースごとに、`<Configuration>` タグ内に `<Database>` タグを追加します。マッピングを定義するには、該当する `<Database>` タグの下に、`<GlobalMapping>` タグ、`<RoutineMapping>` タグ、および `<ClassMapping>` タグを追加します。`</Configuration>` タグは、定義されたマッピングを有効化します。

さらに、`<Import>` タグを使用して、`<Namespace>` タグ内にグローバル、ルーチン、およびクラスをロードできます。`<Invoke>` タグでクラス・メソッドを起動してルーチンを実行し、インポートされているグローバルにアクセスすることもできます。

`<Var>` タグで変数を定義できます。マニフェスト内の変数を参照するには、`${var_name}` 構文を使用します。詳細は、以下の“[<Manifest> タグ内の変数](#)”を参照してください。

A.2.2.2 ユーザとパスワードの追加

インストールしたインスタンスにユーザ (ロールとパスワードを含む) を追加する方法は複数あります。

- “[<Manifest> タグのリスト](#)”の説明に従って、インストール・マニフェストに `<User>` タグを含めます。

`<User>` タグの `PasswordVar` パラメータはユーザのパスワードを格納した変数を指定します。例えば、`PasswordVar="Pwd"` を定義すると、変数 `Pwd` の値がユーザのパスワードに指定されます。この変数にデータを移入する方法は多様ですが、移入方法は担当者次第です。InterSystems IRIS または Web サービスの別のインスタンスに対してメソッドをリモートで呼び出すことも検討できますが、この方法の問題点は、InterSystems IRIS がインストールされているサーバでインターネット・アクセスが必要な場合があることです。インストールしているインスタンスに対して使用するメソッドをインポートするか、ユーザとパスワード (マニフェストに渡すことができます) の入力を求めるインストールにクライアント側フォームを追加する方法も考えられます。

- インストール完了後に管理ポータルでユーザを編集します (“[ユーザ](#)”を参照)。
- InterSystems IRIS のステージング・インスタンスで `Security.Users` クラスを使用します (以下を参照)。

1. Security.Users.Export() メソッドを使用して、ユーザ情報をエクスポートします。
2. ユーザ情報をインポートするには、(%SYS ネームスペースの) マニフェスト・クラスの先頭で以下を追加します。

```
<Invoke Class="Security.Users" Method="Import" CheckStatus="true">
  <Arg Value="PathToExportedUserInformation"/>
</Invoke>
```

PathToExportedUserInformation は、Security.Users.Export() メソッドで指定された出力ファイルの場所です。

A.2.2.3 マニフェスト・ログへの書き込み

次の形式でクラスに `<Log>` タグを組み込むことで、マニフェスト・ログに追加するメッセージを定義できます。

```
<Log Level="<level>" Text="<text>" />
```

ログのレベルは -1 から 3 までの範囲になる必要があります。-1 は “none” (なし)、3 は “verbose” (詳細) です。setup() メソッドで指定されたログのレベルが `<Log>` タグの **Level** プロパティの値以上の場合にメッセージがログに記録されます。テキストは 32,000 文字に制限されます。

setup() メソッドの 2 番目の引数によりレベルを設定します (詳細は、この付録で後述の “[マニフェストの使用](#)” を参照)。これはインストールからマニフェストへ渡すことはできないため、次のようにクラスに設定する必要があります。

```
ClassMethod setup(ByRef pVars, pLogLevel As %Integer = 3,
  pInstaller As %Installer.Installer,
  pLogger As %Installer.AbstractLogger)
  As %Status [ CodeMode = objectgenerator, Internal ]
```

メッセージの表示位置を setup() メソッドの 4 番目の引数 (%Installer.AbstractLogger) により指定できます。例えば、オペレーティング・システム・ファイルを参照する %Installer.FileLogger オブジェクトをインスタンス化する場合、すべての %Installer およびログ・メッセージはそのファイルに書き込まれます。(この引数がないと、setup() が直接呼び出された場合には、すべてのメッセージがプライマリ・デバイスに書き込まれ、インストーラによって実行された場合には無視されます。)

A.2.2.4 アップグレード後のタスクの実行

InterSystems IRIS のインスタンスをアップグレードする際、インストール・マニフェストを使用して、コードのリコンパイルなど、必要な[アップグレード後のタスク](#)を自動的に実行できます。そのためには、`<Invoke>` タグを使用して、このドキュメントの “InterSystems IRIS のアップグレード” の章の “[ネームスペースのコンパイル方法](#)” のセクションに記載されているクラス・メソッドを呼び出します。

アップグレード時に使用できるマニフェストの例は、この付録の “[インストール・マニフェストの例](#)” のセクションを参照してください。

A.2.3 <Manifest> タグ内の変数

一部のタグは、マニフェストが実行されると展開される式 (文字列) を含めることができます。展開できる式には 3 種類あります (下記参照)。

`#{<var_name>}` – 変数

変数の値に展開されます。このセクションで説明する事前定義変数のほかに、`<Var>` タグで追加の変数を指定できます。

`#{#<param_name>}` – クラス・パラメータ

マニフェスト・クラスの指定されたパラメータの値に展開します。

#{<ObjectScript_expression>} – ObjectScript 式

指定された InterSystems IRIS Object Script 式に展開します（正しく引用符を付ける必要があります）。

注釈 パラメータの式はコンパイル時に展開されます。つまり、変数および ObjectScript 式の内側で入れ子にできます。また、変数の式は ObjectScript 式の前に展開されるため、後者の内側で入れ子にできます。

以下のテーブルに、マニフェストで使用できる事前定義変数を示します。

変数名	説明
SourceDir	（インストーラ実行中にのみ使用可能）インストール (setup_irisdb.exe または irisinstall) の実行元となるディレクトリ。
ISCUUpgrade	（インストーラ実行中にのみ使用可能）新規インストールまたはアップグレードであるかを示します。この変数は 0（新規インストール）または 1（アップグレード）のいずれかになります。
CFGDIR	“INSTALLDIR” を参照してください。
CFGNAME	インスタンス名。
CPUCOUNT	オペレーティング・システムの CPU の数。
CSPDIR	CSP ディレクトリ。
HOSTNAME	ホスト・サーバの名前。
HTTPPORT	Web サーバのポート。
INSTALLDIR	InterSystems IRIS がインストールされるディレクトリ。
MGRDIR	管理者用 (mgr) ディレクトリ。
PLATFORM	オペレーティング・システム。
PORT	InterSystems IRIS スーパーサーバ・ポート。
PROCESSOR	プロセッサ・チップ。
VERSION	InterSystems IRIS のバージョン番号。

A.2.4 <Manifest> タグのリスト

コード生成に関するすべての情報は、最も外側の XML タグ (<Manifest>) 内に含まれます。<Manifest> 内の各タグは、InterSystems IRIS の特定の構成を記述します。このセクションでは、これらのタグとその関数のリストを示します。ここでは、すべてのタグと属性は記載されていません。完全なリストについては、%Installer クラスリファレンスのドキュメントを参照してください。属性に既定値がある場合は、各属性の横に角括弧で囲んでリストされています。

重要 NULL 引数はマニフェスト・クラスのタグで渡すことはできません。例えば、<Arg/> タグは空の文字列を渡すので、<Arg=" "/> と同等となります。NULL ではありません。

- ・ <Arg>
- ・ <ClassMapping>
- ・ <Compile>
- ・ <Configuration>
- ・ <CopyClass>
- ・ <CopyDir>

- ・ [〈CopyFile〉](#)
- ・ [〈Credential〉](#)
- ・ [〈Database〉](#)
- ・ [〈Default〉](#)
- ・ [〈Else〉](#)
- ・ [〈Error〉](#)
- ・ [〈ForEach〉](#)
- ・ [〈GlobalMapping〉](#)
- ・ [〈If〉](#)
- ・ [〈IfDef〉](#)
- ・ [〈IfNotDef〉](#)
- ・ [〈Import〉](#)
- ・ [〈Invoke〉](#)
- ・ [〈LoadPage〉](#)
- ・ [〈Log〉](#)
- ・ [〈Manifest〉](#)
- ・ [〈Namespace〉](#)
- ・ [〈Production〉](#)
- ・ [〈Resource〉](#)
- ・ [〈Role〉](#)
- ・ [〈RoutineMapping〉](#)
- ・ [〈Setting〉](#)
- ・ [〈SystemSetting〉](#)
- ・ [〈User〉](#)
- ・ [〈Var〉](#)

<Arg>

親タグ : [〈Invoke〉](#)、[〈Error〉](#)

[〈Invoke〉](#) または [〈Error〉](#) から呼び出されるメソッドに引数を渡します。

- ・ **Value**—引数の値。

```
<Error Status="$$$NamespaceDoesNotExist">  
  <Arg Value="{NAMESPACE}" />  
</>
```

<ClassMapping>

親タグ : [〈Configuration〉](#)

データベースからこの [〈Configuration〉](#) 項目を含むネームスペースへのクラス・マッピングを作成します。

- ・ **Package**—マップするパッケージ。
- ・ **From**—マッピングに使用するソース・データベース。

```
<ClassMapping Package="MYAPP"
  From="MYDB" />
```

<Compile>

親タグ : [<Namespace>](#)

%SYSTEM.OBJ.Compile(Class, Flags) を呼び出して、指定されたクラス名をコンパイルします。

- ・ **Class**—コンパイルするクラスの名前。
- ・ **Flags**—コンパイル・フラグ [ck]。
- ・ **IgnoreErrors**—エラー時に続行するかどうかを指定します [0]。

```
<Compile
  Class="MyPackage.MyClass"
  Flags="ck"
  IgnoreErrors=0 />
```

<Configuration>

親タグ : [<Namespace>](#)

子タグ : [<ClassMapping>](#)、[<Database>](#)、[<GlobalMapping>](#)、[<RoutineMapping>](#)

<Namespace> 内で構成タグの親タグとして記述する必要があります。終了タグ (</Configuration>) は、ネームスペース内のデータベースのマッピングを有効化して、.cpf ファイルを更新します。

プロパティはありません。

```
<Configuration>
  <Database> . . . />
  <ClassMapping> . . . />
/>
```

<CopyClass>

親タグ : [<Namespace>](#)

ソース・クラス定義をターゲットにコピーまたは移動します。

- ・ **Src**—ソース・クラス。
- ・ **Target**—ターゲット・クラス。
- ・ **Replace**—ターゲット・クラスを上書きするかどうかを指定します [0]。

```
<CopyClass
  Src="MyPackage.MyClass"
  Target="NewPackage.NewClass"
  Replace="0" />
```

<CopyDir>

親タグ : [<Manifest>](#)

ソース・ディレクトリをターゲットにコピーします。

- ・ **Src**—ソース・ディレクトリ。
- ・ **Target**—ターゲット・ディレクトリ。

- ・ **IgnoreErrors**—エラー時に続行するかどうかを指定します [0]。

```
<CopyDir
  Src="{MGRDIR}"
  Target="F:\MyTargetDir"
  IgnoreErrors="0"/>
```

<CopyFile>

親タグ : [Manifest](#)

ソース・ファイルをターゲットにコピーします。

- ・ **Src**—ソース・ファイル。
- ・ **Target**—ターゲット・ファイル。
- ・ **IgnoreErrors**—エラー時に続行するかどうかを指定します [0]。

```
<CopyFile
  Src="{MGRDIR}\messages.log"
  Target="F:\${INSTANCE}_log"
  IgnoreErrors="0"/>
```

<CSPApplication>

親タグ : [Namespace](#)

Security.Applications クラス内で定義されている 1 つ以上の Web [アプリケーション](#)を定義します (これらの各フィールドの詳細は、[“アプリケーションの作成および編集”](#) も参照してください)。

- ・ **AuthenticationMethods**—有効な認証方法。(サポートされている認証方法と、対応する提供値については、“Security.Applications” の “AuthEnabled” プロパティを参照してください)。例えば、一般的に使用される値として、4=Kerberos、32=password、64=unauthenticated があります。
- ・ **AutoCompile**—(CSP 設定で) 自動コンパイルするかどうかを指定します[1]。
- ・ **CSPZENEnabled**—CSP/ZEN を有効にするかどうかを指定します [1]。
- ・ **ChangePasswordPage**—パスワード変更ページへのパス。
- ・ **CookiePath**—セッション cookie パス。
- ・ **CustomErrorPage**—カスタム・エラー・ページへのパス。
- ・ **DefaultSuperclass**—デフォルト・スーパークラス。
- ・ **DefaultTimeout**—セッション・タイムアウト。
- ・ **Description**—説明。
- ・ **Directory**—CSP ファイルへのパス。
- ・ **EventClass**—イベント・クラス名。
- ・ **Grant**—システムにログインするときに割り当てられるロールのリスト。
- ・ **GroupById**—ID でグループ化するかどうかを指定します。
- ・ **InboundWebServicesEnabled**—着信 Web サービスを有効にするかどうかを指定します[1]。
- ・ **IsNamespaceDefault**—ネームスペースの既定のアプリケーションかどうかを指定します[0]。
- ・ **LockCSPName**—CSP名をロックするかどうかを指定します[1]。
- ・ **LoginClass**—ログイン・ページへのパス。

- ・ **PackageName**—パッケージ名。
- ・ **PermittedClasses**—許可されるクラス。
- ・ **Recurse**—繰り返す (サブディレクトリを提供する) かどうかを指定します[0]。
- ・ **Resource**—Web アプリケーションにアクセスするために必要なリソース。
- ・ **ServeFiles**—ファイルを提供するかどうかを指定します。[1]。
 - 0—提供しない
 - 1—常に提供する
 - 2—常に提供し、キャッシュする
 - 3—CSP セキュリティを使用する
- ・ **ServeFilesTimeout**—静的ファイルをキャッシュする時間を秒単位で指定します。
- ・ **TwoFactorEnabled**—2 ステップ認証を有効にするかどうかを指定します[0]。
- ・ **Url**—Web アプリケーションの名前。
- ・ **UseSessionCookie**—セッションに cookie を使用するかどうかを指定します。

```
<CSPApplication
  Url="/csp/foo/bar"
  Description=""
  Directory="C:\InterSystems\IRIS\CSP\Democode1"
  Resource=""
  Grant="%DB_%DEFAULT"
  Recurse="1"
  LoginClass=""
  CookiePath="/csp/demo1"
  AuthenticationMethods="64" />
```

<Credential>

親タグ : [<Production>](#)

アクセス資格情報を作成またはオーバーライドします。

- ・ **Name**—アクセス資格情報の名前。
- ・ **Username**—ユーザ名。
- ・ **Password**—ユーザ・パスワード。
- ・ **Overwrite**—アカウントが既に存在する場合は上書きします。

```
<Credential
  Name="Admin"
  Username="administrator"
  Password="123jUgT540!f3B$#"
  Overwrite="0" />
```

<Database>

親タグ : [<Configuration>](#)

ネームスペース内でデータベースを定義します。

以下のプロパティで制御するデータベース設定の詳細は、“システム管理ガイド”の“InterSystems IRIS の構成”の章にある“[データベースの構成](#)”を参照してください。

- ・ **BlockSize**—データベースのブロック・サイズ (4096、8192、16384、32768、65536)。

- ・ **ClusterMountMode**—起動時にクラスタの一部としてデータベースをマウントするかどうかを指定します。
- ・ **Collation**—データベース内に作成される、グローバルの既定の照合。
- ・ **Create**—新しいデータベースを作成するかどうかを指定します (yes/no/overwrite) [yes]
- ・ **Dir**—データベース・ディレクトリ。
- ・ **Encrypted**—データベースを暗号化するかどうかを指定します。
- ・ **EncryptionKeyID**—暗号化キーの ID。
- ・ **InitialSize**—データベースの初期サイズを MB 単位で指定します。
- ・ **ExpansionSize**—必要な場合にデータベースを拡張するサイズを MB 単位で指定します。
- ・ **MaximumSize**—データベースを拡張できる最大サイズ。
- ・ **MountAtStartup**—インストールされたインスタンスを起動するときにマウントするかどうかを指定します。
- ・ **MountRequired**—インスタンスを起動するたびにデータベースをマウントする必要があるかどうかを指定します。
- ・ **Name**—データベース名。
- ・ **PublicPermissions**—リソースを作成する必要がある場合、リソースに割り当てられる許可の値。リソースが既に存在する場合は無視されます。読み取り専用または読み書き可能です。
- ・ **Resource**—データベースへのアクセスを制御するリソース。
- ・ **StreamLocation**—データベースに関連付けられているストリームが格納されるディレクトリ。

```
<Database Name="{DBNAME}"
  Dir="{MGRDIR}/{DBNAME}"
  Create="yes"
  Resource="{DBRESOURCE}"
  Blocksize="8192"
  ClusterMountMode="0"
  Collation="5"
  Encrypted="0"
  EncryptionKeyID=
  ExpansionSize="0"
  InitialSize="1"
  MaximumSize="0"
  MountAtStartup="0"
  MountRequired="0"
  StreamLocation=
  PublicPermissions=""/>
<Database Name="MYAPP"
  Dir="{MYAPPPDIR}/db"
  Create="no"
  Resource="{MYAPPRESOURCE}"
  Blocksize="8192"
  ClusterMountMode="0"
  Collation="5"
  Encrypted="0"
  EncryptionKeyID=
  ExpansionSize="0"
  InitialSize="1"
  MaximumSize="0"
  MountAtStartup="0"
  MountRequired="0"
  StreamLocation=
  PublicPermissions=""/>
```

<Default>

親タグ : <Manifest>

変数値がまだ設定されていない場合に、変数値を設定します。

- ・ **Name**—変数名。

- ・ **Value**—変数値。
- ・ **Dir**—フォルダまたはファイルのパスを示す変数値。

```
<Default Name="blksiz"
  Value="8192" />
```

<Else>

親タグ : [<If>](#)

先行する <If> 文で定義される条件文が false の場合に実行されます。

プロパティはありません。

```
<If Condition='#
{##class(%File).Exists(INSTALL_"mgr\iris.key")}'
'>
<Else/>
<CopyFile Src="C:\\InterSystems\\key_files\\iris300.key" Target="{MGRDIR}\\iris.key"
IgnoreErrors="0"/>
</If>
```

<Error>

親タグ : [<Manifest>](#)

子タグ : [<Arg>](#)

エラーを生成します。

- ・ **Status** : エラー・コード。
- ・ **Source** : エラーの発生源。

```
<Error Status="$$$NamespaceDoesNotExist" Source=>
  <Arg Value="{NAMESPACE}" />
</Error>
```

<ForEach>

親タグ : [<Manifest>](#)

指定されたインデックス・キーの繰り返しに使用する値を定義します。

- ・ **Index**—変数名。
- ・ **Values**—変数値のリスト。

```
<ForEach
  Index="TargetNameSpace"
  Values="%SYS,User">
  <!--Code for each iteration of TargetNameSpace-->
</ForEach>
```

<GlobalMapping>

親タグ : [<Configuration>](#)

グローバルを現在のネームスペースにマップします。

- ・ **Global** : グローバル名。
- ・ **From** : グローバルのソース・データベース。

- ・ **Collation** : グローバル照合 [IRIS Standard]。

```
<GlobalMapping Global="MyAppData.*"
  From="MYAPP" Collation="30"/>
<GlobalMapping Global="cspRule"
  From="MYAPP"/>
```

<If>

親タグ : [<Manifest>](#)、[<Namespace>](#)

子タグ : [<Else>](#)

条件付きアクションを指定します。

- ・ **Condition** : 条件文。

```
<If Condition='$L("${NAMESPACE}")=0'>
  <Error Status="$$$NamespaceDoesNotExist" Source=>
    <Arg Value="${NAMESPACE}"/>
  </Error>
</If>
```

<IfDef>

親タグ : [<Manifest>](#)、[<Namespace>](#)

変数が設定されている場合の条件付きアクションを指定します。

- ・ **Var** : 変数名。

```
<IfDef Var="DBCreateName">
  <Database Name="${DBNAME}"
    Dir="${MGRDIR}/${DBNAME}"
    Create="yes"
    ...
</IfDef>
```

<IfNotDef>

親タグ : [<Manifest>](#)、[<Namespace>](#)

変数が設定されていない場合の条件付きアクションを指定します。

- ・ **Var** : 変数名。

<Import>

親タグ : [<Namespace>](#)

%SYSTEM.OBJ.ImportDir(File,Flags,Recurse) または %SYSTEM.OBJ.Load(File,Flags) を呼び出してファイルをインポートします。

- ・ **File**—インポートするファイルまたはフォルダ。
- ・ **Flags**—コンパイル・フラグ [ck]。
- ・ **IgnoreErrors**—エラー時に続行するかどうかを指定します [0]。
- ・ **Recurse**—再帰的にインポートするかどうかを指定します [0]。

<Invoke>

親タグ : [<Namespace>](#)

子タグ : [<Arg>](#)

クラス・メソッドを呼び出して、実行結果を変数の値として返します。

- ・ **Class**—クラス名。
- ・ **Method**—メソッド名。
- ・ **CheckStatus**—返されたステータスをチェックするかどうかを指定します。
- ・ **Return**—結果の書き込み先の変数の名前。

```
<Invoke Class="SECURITY.SSLConfigs" Method="GetCertificate" CheckStatus="1" Return="CertContents">
  <Arg Value="iris.cer" />
</Invoke>
```

<LoadPage>

親タグ : [<Namespace>](#)

%SYSTEM.CSP.LoadPage(PageURL,Flags) または %SYSTEM.CSP.LoadPageDir(DirURL,Flags) を呼び出して CSP ページをロードします。

- ・ **Name**—CSP ページの URL。
- ・ **Dir**—CSP ページを含むディレクトリの URL。
- ・ **Flags**—コンパイル・フラグ [ck]。
- ・ **IgnoreErrors**—エラー時に続行するかどうかを指定します [0]。

<Log>

親タグ : [<Manifest>](#)

setup() メソッドで指定されたログのレベルが **level** プロパティの値以上の場合に、setup() メソッドで指定されたログにメッセージを記録します。

- ・ **Level**—ログのレベル。-1 (none) から 3 (verbose) までの範囲です。
- ・ **Text**—ログ・メッセージ (最大 32,000 文字の文字列)。

詳細は、“[マニフェスト・ログへの書き込み](#)” を参照してください。

<Manifest>

親タグ : なし (ルート・タグ。他のすべてのタグを含みます)

子タグ : [<CopyDir>](#)、[<CopyFile>](#)、[<Default>](#)、[<Else>](#)、[<Error>](#)、[<ForEach>](#)、[<If>](#)、[<IfDef>](#)、[<IfNotDef>](#)、[<Log>](#)、[<Namespace>](#)、[<Resource>](#)、[<Role>](#)、[<SystemSetting>](#)、[<User>](#)、[<Var>](#)

プロパティはありません。

```
<Manifest>
  <Namespace ... >
    <Configuration>
      <Database .../>
      <Database .../>
    </Configuration>
  </Namespace>
</Manifest>
```

<Namespace>

親タグ : [<Manifest>](#)

子タグ : [<Compile>](#)、[<Configuration>](#)、[<CopyClass>](#)、[<CSPApplication>](#)、[<Else>](#)、[<If>](#)、[<IfDef>](#)、[<IfNotDef>](#)、[<Import>](#)、[<Invoke>](#)、[<LoadPage>](#)、[<Production>](#)

ネームスペースを定義します。

- ・ **Name**—ネームスペースの名前。
- ・ **Create**—新しいネームスペースを作成するかどうかを指定します (yes/no/overwrite) [yes]
- ・ **Code**—コード用データベース。
- ・ **Data**—データ用データベース。
- ・ **Ensemble**—相互運用対応のネームスペースかどうかを指定します [0]。

(相互運用 Web アプリケーションには他のプロパティを適用可能)

```
<Namespace Name="${NAMESPACE}"
  Create="yes"
  Code="${NAMESPACE}"
  Data="${NAMESPACE}">
  <Configuration>
    <Database Name="${NAMESPACE}" . . . />
  </Configuration>
</Namespace>
```

<Production>

親タグ : [<Namespace>](#)

子タグ : [<Credential>](#)、[<Setting>](#)

プロダクションを定義します。

- ・ **Name**—プロダクション名。
- ・ **AutoStart**—プロダクションを自動的に起動するかどうかを指定します [0]。

```
<Production Name="${NAMESPACE}"
  AutoStart="1" />
```

<Resource>

親タグ : [<Manifest>](#)

リソースを定義します。

- ・ **Name**—リソース名。
- ・ **Description**—リソースの説明。
- ・ **Permission**—パブリック許可。

```
<Resource
  Name="%accounting_user"
  Description="Accounting"
  Permission="RW" />
```

<Role>

親タグ : [<Manifest>](#)

ロールを定義します。

- ・ **Name**—ロール名。
- ・ **Description**—ロールの説明 (コンマは使用できません)。

- ・ **Resources**—ロールで保持されている特権 (リソースと特権のペア)。
- ・ **RolesGranted**—指定されたロールで付与されるロール。

```
<Role
  Name="%DB_USER"
  Description="Database user"
  Resources="MyResource:RW,MyResource1:RWU"
  RolesGranted= />
```

<RoutineMapping>

親タグ : [<Configuration>](#)

ルーチン・マッピングを定義します。

- ・ **Routines** : ルーチン名。
- ・ **Type** : ルーチン・タイプ (MAC、INT、INC、OBJ、ALL)。
- ・ **From** : ルーチンのソース・データベース。

```
<RoutineMapping Routines="MyRoutine"
  Type="ALL" From="{NAMESPACE}" />
```

<Setting>

親タグ : [<Production>](#)

Ens.Production.ApplySettings() メソッドを呼び出して、プロダクション内のアイテムを構成します。

- ・ **Item**—アイテム名。
- ・ **Target**—設定タイプ (Item、Host、Adapter)。
- ・ **Setting**—設定名。
- ・ **Value**—設定の構成値。

```
<Production Name="Demo.ComplexMap.SemesterProduction">
  <Setting Item="Semester_Data_FileService"
    Target="Item"
    Setting="PoolSize"
    Value="1"/>
  <Setting Item="Semester_Data_FileService"
    Target="Host"
    Setting="ComplexMap"
    Value="Demo.ComplexMap.Semester.SemesterData"/>
  <Setting Item="Semester_Data_FileService"
    Target="Adapter"
    Setting="FilePath"
    Value="C:\Practice\in\" />
</Production>
```

<SystemSetting>

親タグ : [<Manifest>](#)

Config.CommonSingleMethods から Modify() メソッドを継承する任意の Config クラスのプロパティの値を設定します。

- ・ **Name**—Config クラスの Class.property。
- ・ **Value**—プロパティに割り当てる値。

<User>

親タグ：<Manifest>

ユーザを定義します。**PasswordVar** が含まれる場合、指定された変数名でユーザのパスワードを渡す必要があります。

- ・ **Username**—ユーザ名。
- ・ **PasswordVar**—ユーザ・パスワードが含まれる変数の名前（以下の <Var> を参照）。
- ・ **Roles**—ユーザを割り当てるロールのリスト。
- ・ **Fullname**—ユーザのフルネーム。
- ・ **Namespace**—ユーザの実行開始ネームスペース。
- ・ **Routine**—ユーザの実行開始ルーチン。
- ・ **ExpirationDate**—それ以降のユーザ・ログインが無効になる日付。
- ・ **ChangePassword**—次回ログイン時にユーザにパスワードの変更を要求するかどうかを指定します。
- ・ **Enabled**—ユーザが有効かどうかを指定します。
- ・ **Comment**—オプションコメント。

```
<User
  Username="Clerk1"
  PasswordVar="clerk1pw"
  Roles="Dataentry"
  Fullname="Data Entry Clerk"
  Namespace=
  Routine=
  ExpirationDate=
  ChangePassword=
  Enabled=
  Comment=" " />
```

<Var>

親タグ：<Manifest>

マニフェストで可以使用変数を定義および設定します。

- ・ **Name**—変数名。
- ・ **Value**—変数に割り当てる値。

```
<Var Name="Namespace" Value="MUSIC" />
<Var Name="GlobalDatabase" Value="{Namespace}G" />
<Var Name="RoutineDatabase" Value="{Namespace}R" />
<Var Name="AppDir" Value="C:\MyApp\${CFGNAME}" />
<Var Name="GlobalDatabaseDir" Value="{AppDir}\${GlobalDatabase}" />
<Var Name="RoutineDatabaseDir" Value="{AppDir}\${RoutineDatabase}" />
<Var Name="Resource" Value="%DB_${Namespace}" />
<Var Name="Role" Value="{Namespace}" />
<Var Name="CSPResource" Value="CSP_${Namespace}" />
```

A.3 マニフェストの使用

マニフェストは、手動で実行することも、インストールの一部として実行することもできます。

A.3.1 手動で実行する場合

%SYS ネームスペースで、ターミナルに以下のコマンドを入力します。

```
%SYS>do ##class(MyPackage.MyInstaller).setup()
```

変数の配列を参照として setup() メソッドに渡すことができます。各添え字は**変数名**として使用され、ノードは値として使用されます。以下に例を示します。

```
%SYS>set vars("SourceDir")="c:\myinstaller"
%SYS>set vars("Updated")="Yes"
%SYS>do ##class(MyPackage.MyInstaller).setup(.vars,3)
```

この例の 2 番目の引数 (3) は、ログ・レベルです。

A.3.2 インストール時に実行する場合

マニフェスト・クラスを DefaultInstallerClass.xml として InterSystems IRIS インストール (.msi、setup_irisdb.exe、または irisinstall) が実行されている場所と同じディレクトリにエクスポートします。これは %SYS にインポートされてコンパイルされ、setup() メソッドが実行されます。

注釈 インストールを実行するユーザが、マニフェスト・ログ・ファイル・ディレクトリへの書き込み許可を持っていることを確認します。ログ・ファイルに書き込めない場合、インストールは続行されません。

また、単一ファイル・キットを使用してインストールしている場合、想定どおりに動作するようにするには、ファイルを .exe ファイルから抽出する必要があります。

エクスポートする方法では、引数を setup() メソッドに直接渡せないことに注意してください。以下のセクションでは、[Windows](#) で、または [UNIX®](#)、[Linux](#)、および [macOS](#) で引数を渡す方法について説明します。

A.3.2.1 Windows

.msi インストール・パッケージを変更して、変数の名前と値のペアを setup() メソッドに渡すことができます。

また、以下の例に示すようにインストーラでコマンド行引数を使用して、エクスポートされるマニフェスト・クラスの場合、変数、ログ・ファイル名、およびログのレベルを渡すこともできます。

```
setup.exe INSTALLERMANIFEST="c:\MyStuff\MyInstaller.xml"
INSTALLERMANIFESTPARAMS="SourceDir=c:\mysourcedir,Updated=Yes"
INSTALLERMANIFESTLOGFILE="installer_log" INSTALLERMANIFESTLOGLEVEL="2"
```

注釈 INSTALLERMANIFESTPARAMS 引数を使用して渡される変数名には、アルファベット文字と数字 (A-Za-z0-9)、およびアンダースコア () のみを含めることができますが、先頭をアンダースコアにすることはできません。

詳細は、このドキュメントの “Microsoft Windows への InterSystems IRIS のインストール” の章の “[コマンド行のリファレンス](#)” のセクションを参照してください。

A.3.2.2 UNIX®, Linux、および macOS

UNIX®, Linux、および macOS システムでは、以下の例に示すように、irisinstall または irisinstall_silent のいずれかを実行する前に環境変数を設定して、エクスポートされるマニフェスト・クラスの場合、変数、ログ・ファイル名、およびログのレベルを定義できます。

```
ISC_INSTALLER_MANIFEST="/MyStuff/MyInstaller.xml"
ISC_INSTALLER_PARAMETERS="SourceDir=/mysourcedir,Updated=Yes"
ISC_INSTALLER_LOGFILE="installer_log"
ISC_INSTALLER_LOGLEVEL="2"
./irisinstall
```

詳細は、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”の章の“[自動インストール・パラメータ](#)”のセクションを参照してください。

A.4 インストール・マニフェストの例

このセクションでは、以下を行うインストール・マニフェストの例を示します。

- ・ [空のテンプレート](#)
- ・ [ネームスペースの作成](#)
- ・ [アップグレード後のコンパイル](#)

A.4.1 空のテンプレート

以下に、基本的なマニフェスト・テンプレートを示します。

Class Definition

```
Include %occInclude

/// Simple example of installation instructions (Manifest)
Class MyInstallerPackage.SimpleManifest
{

XData SimpleManifest [ XMLNamespace = INSTALLER ]
{
<Manifest>
  <Log Level="3" Text="Start manifest" />
  <Namespace Name="">
    <Import File="" />
    <Invoke Class="" Method="" CheckStatus="" Return="">
      <Arg Value="" />
    </Invoke>
  </Namespace>
  <CopyFile Src="" Target="" IgnoreErrors="" />
  <CopyDir Src="" Target="" IgnoreErrors="" />
  <Log Level="3" Text="End manifest" />
</Manifest>
}

/// This is a method generator whose code is generated by XGL.
ClassMethod setup(ByRef pVars, pLogLevel As %Integer = 3,
  pInstaller As %Installer.Installer,
  pLogger As %Installer.AbstractLogger)
  As %Status [ CodeMode = objectgenerator, Internal ]
{
  #; Let our XGL document generate code for this method.
  Quit ##class(%Installer.Manifest).%Generate(%compiledclass, %code, "SimpleManifest")
}
}
```

A.4.2 ネームスペースの作成

この例のマニフェストでは、ネームスペース (MyNamespace) と 3 つのデータベースを作成します。MyDataDB データベースと MyRoutinesDB データベースはそれぞれグローバルとルーチンの既定のデータベースで、MyMappingDB データベースは別のグローバル・データベースです。<GlobalMapping> タグは、MyNamespace ネームスペースにある t* グローバルに対して MyMappingDB へのマッピングを作成します。これは、既定のマッピングをオーバーライドします。

Class Definition

```

Include %occInclude

/// Simple example of installation instructions (Manifest)
Class MyInstallerPackage.SimpleManifest
{

XData SimpleManifest [ XMLNamespace = INSTALLER ]
{
<Manifest>
  <Namespace Name="MyNamespace" Create="yes"
    Code="MyRoutinesDB" Data="MyDataDB">
    <Configuration>
      <Database Name="MyRoutinesDB" Create="yes"
        Dir="C:\MyInstallerDir\MyRoutinesDB"/>
      <Database Name="MyDataDB" Create="yes"
        Dir="C:\MyInstallerDir\MyDataDB"/>
      <Database Name="MyMappingDB" Create="yes"
        Dir="C:\MyInstallerDir\MyMappingDB"/>
      <GlobalMapping Global="t*" From="MyMappingDB"/>
    </Configuration>
  </Namespace>
</Manifest>
}

/// This is a method generator whose code is generated by XGL.
ClassMethod setup(ByRef pVars, pLogLevel As %Integer = 3,
  pInstaller As %Installer.Installer,
  pLogger As %Installer.AbstractLogger)
  As %Status [ CodeMode = objectgenerator, Internal ]
{
  #; Let our XGL document generate code for this method.
  Quit ##class(%Installer.Manifest).%Generate(%compiledclass,
    %code, "SimpleManifest")
}
}

```

A.4.3 アップグレード後のコンパイル

この例のマニフェストでは **APPTEST** という名前のネームスペース内のコードをリコンパイルし、アップグレード後に実行します。以下のタスクを実行します。

1. マニフェスト・インストール・プロセスの開始を示すメッセージをマニフェスト・ログに書き込みます。
2. 現在のネームスペースを **APPTEST** に設定します。これには、アップグレード後にリコンパイルする必要のあるコードが含まれます。
3. メソッド %SYSTEM.OBJ.CompileAll() を呼び出して、ネームスペース内のクラスをコンパイルします。
4. メソッド %Routine.CompileAll() を呼び出して、ネームスペース内のカスタム・ルーチンをコンパイルします。
5. マニフェスト・インストール・プロセスの終了を示すメッセージをマニフェスト・ログに書き込みます。

Class Definition

```

Include %occInclude

/// Simple example of installation instructions (Manifest)
Class MyInstallerPackage.SimpleManifest
{

XData SimpleManifest [ XMLNamespace = INSTALLER ]
{
<Manifest>
  <Log Level="3" Text="The Installer Manifest is running."/>
  <Namespace Name="APPTEST">
    <Invoke Class="%SYSTEM.OBJ" Method="CompileAll" CheckStatus="1">
      <Arg Value="u"/>
    </Invoke>
    <Invoke Class="%Routine" Method="CompileAll" CheckStatus="1"/>
  </Namespace>
  <Log Level="3" Text="The Installer Manifest has completed."/>
</Manifest>
}

```

```
}  
  
/// This is a method generator whose code is generated by XGL.  
ClassMethod setup(ByRef pVars, pLogLevel As %Integer = 3,  
    pInstaller As %Installer.Installer,  
    pLogger As %Installer.AbstractLogger)  
    As %Status [ CodeMode = objectgenerator, Internal ]  
{  
    #; Let our XGL document generate code for this method.  
    Quit ##class(%Installer.Manifest).%Generate(%compiledclass, %code, "SimpleManifest")  
}  
  
}
```


B

InterSystems IRIS ディストリビューションへの UNIX® インストール・パッケージの追加

この付録では、既存の InterSystems IRIS® Data Platform ディストリビューションに新しい UNIX® インストール・パッケージを追加する方法を説明します。この追加方法はチュートリアル形式で示し、InterSystems IRIS インスタンス・ディレクトリに追加ファイルをコピーする単純なパッケージを作成する場合について解説します。

注釈 インストール・パッケージは UNIX® シェル・スクリプトを使用して実装するため、これよりはるかに複雑な操作を実行するパッケージも作成できます。

B.1 チュートリアル

Foo9000 という名前のイメージング・デバイスに接続するインターシステムズのコールアウト共有ライブラリ (“コールアウト・ゲートウェイの使用法” の “InterSystems コールアウト・ライブラリの作成” の章を参照) を記述したとします。このライブラリを **libfoo9000.so** としてコンパイルします。このライブラリが InterSystems IRIS と共にインストールされるようにしたいとします。また、インストール時に、このライブラリの接続先デバイス (Foo9000) のネットワーク・サーバ名を入力するようにユーザに求めるプロンプトが表示されるようにする必要もあります。この情報は InterSystems IRIS インスタンスの `install-dir¥mgr` ディレクトリ内の構成ファイルに保存されます。

既存の InterSystems IRIS キットから開始します。

```
~/kit:>ls
irisinstall  cplatname docs  lgpl.txt NOTICE
copyright.pdf dist  kitlist LICENSE package
```

また、コンパイルしたライブラリ (**libfoo9000.so**) を使用します。

```
~/lib:>ls
libfoo9000.so
```

まず、ライブラリを保存するキット内の場所を選択し、その場所にライブラリをコピーする必要があります。規約により、プラットフォーム別のライブラリは `dist/package/platform` ディレクトリ (`~/kit/dist/foo9000/lnxsusex64` など) に保存します。

```
~/kit:>cd dist
~/kit/dist:>mkdir foo9000
~/kit/dist:>cd foo9000
~/kit/dist/foo9000:>mkdir lnxsusex64
~/kit/dist/foo9000:>cd lnxsusex64
~/kit/dist/foo9000/lnxsusex64:>cp ~/lib/libfoo9000.so .
```


次に、インストール・パッケージのディレクトリを作成し、パッケージについて記述する `manifest.isc` ファイルをこのディレクトリに追加します。最も単純な形式の `manifest.isc` ファイルでは、パッケージ名のみを記述します。この名前はパッケージ・ディレクトリの名前 (`foo9000`) と一致させる必要があります。

```
~/kit/package:>mkdir foo9000
~/kit/package:>cd foo9000
~/kit/package/foo9000:>emacs manifest.isc
package: foo9000
```

パッケージ内が空だと処理は何も実行されませんが、このチュートリアルでは、次の処理を実行する必要があります。

1. Foo9000 をホストするサーバ名の入力をユーザに求めるプロンプトを表示する。
2. `install-dir¥mgr` ディレクトリ内の構成ファイルにこの情報を保存する。
3. インスタンス・バイナリ・ディレクトリにライブラリ (`libfoo9000.so`) をコピーする。

パッケージ・インストーラは複数のフェーズで処理を実行します。そのうち重要なフェーズは次のフェーズです。

- ・ “parameters” フェーズ
- ・ “install” フェーズ

注釈 パッケージには、フェーズごとに、それぞれのフェーズの名前を持つ Bourne シェル・スクリプトを格納することができます。パッケージ・インストーラは、フェーズ内の適切な時点で、各パッケージのスクリプトを実行します。パッケージのスクリプトで所定のフェーズの実行が正常に終了すると、最終コマンドによって明示的または暗黙的にエラー・コード 0 が返されます。正常に終了しなかった場合は、0 以外のエラー・コードが返されます。

“parameters” フェーズでは、パッケージのインストールに必要な情報を (通常ユーザに入力を求めることによって) 収集します。このフェーズでは、システムへの永続的な変更は行いません。通常、“parameters” フェーズ後に、ユーザがインストールの取り消しを必要に応じて選択できるようにします。取り消しが選択された場合、システムがインストール・プロセスによる影響を何も受けていない状態になっていなければなりません。

“install” フェーズでは、システムを変更します。無人インストールや自動インストールが実行される場合があるため、install フェーズ中はユーザに情報の入力を求めないようにします。

パッケージによっては、ユーザからの情報の入力が不要、つまり “parameters” スクリプトが不要な場合があります。特定のフェーズのスクリプトがパッケージに含まれていない場合、そのフェーズ中、そのパッケージの処理は何も実行されません。

まず、`foo9000` パッケージ用の “parameters” スクリプトとして、以下のようなスクリプトを作成したとします。

```
~/kit/package/foo9000:>emacs parameters
#!/bin/sh
echo "Please enter host name of the Foo9000 imaging server: "
read host
echo "Host $host entered."
```

このスクリプトを実行すると、以下のように、ホスト名の入力を求めるプロンプトが実際に表示されます。この情報は `host` 変数に記録されます。

```
~/kit/package/foo9000:>sh parameters
Please enter host name of the Foo9000 imaging server:
cupertino
Host cupertino entered.
```

しかし、`host` 値を取得した後、この値をどう処理したらいいでしょうか。スクリプトの実行が終了すると、この情報は失われるため、“install” フェーズでこの情報を構成ファイルに書き込む必要が生じたときに利用できません。

注釈 “parameters” フェーズではユーザのシステムに何の影響も及ぼさないようにするため、この時点ではまだ構成ファイルを作成しません。

パッケージ・インストーラは、複数のフェーズや複数のパッケージで情報を共有できるようにする便利な関数のペア Import と Export を備えています。通常のシェル・スクリプトのメカニズムによって、これらの関数を `parameters.include` ファイルに組み込んで使用できます。

```
#!/bin/sh
. parameters.include
echo "Please enter host name of the Foo9000 imaging server: "
read host
echo "Host $host entered."
Export foo9000.host $host
```

Export 関数は、エクスポートする(通常、スクリプトのローカル変数から)パラメータ変数の名前とその値を取ります。Import 関数ではこの逆になります。最初の引数には、あらかじめエクスポートされている値のインポート先のローカル変数を指定し、2 番目の引数には、エクスポート先のパラメータ変数の名前を指定します。

注釈 規約により、パラメータ関数の名前は、package name.local variable name 形式の名前 (`foo9000.host` など) にします。

これで、インストール・プロセスに必要なすべての Foo9000 情報が “parameters” スクリプトで収集されるようになったため、次は “install” スクリプトを作成します。

```
~/kit/package/foo9000:>emacs install
#!/bin/sh
. parameters.include
Import host foo9000.host
echo host=$host > ???/mgr/foo9000.cfg
cp ???/dist/foo9000/???/libfoo9000.so ???/bin
```

以下の詳細を指定する必要があります(上記スクリプトの ??? の部分)。

- ・ インストールを作成するインスタンス・ディレクトリの場所。
- ・ インストール元のキットの場所。
- ・ インストールするプラットフォームの種類。

これらの質問は “parameters” スクリプトに組み込むこともできますが、ユーザはこれらの情報をインストールの初期段階で既に入力しているため、混乱を招く可能性があります。代わりに、必要な情報を提供する他のパッケージからパラメータ変数をインポートします。これが可能なのは、`irisinstall`、`irisinstall_client` または `irisinstall_silent` スクリプトを使用したインストールが成功すると(“[InterSystems IRIS のインストール](#)” を参照)、これらの変数と値を含んだ `parameters.isc` ファイルがインストール・ディレクトリに作成されるためです。`parameters.isc` ファイルの変数の一覧は、この付録の最後にある “[InterSystems IRIS インストール・パラメータ・ファイルの変数](#)” のテーブルに掲載しています。

注釈 セキュリティ上の理由により、`parameters.isc` ファイルには root ユーザのみがアクセスできます。

特定のパッケージのパラメータ変数を使用するためには、現在のパッケージ (`foo9000`) が別のパッケージに依存すること、したがって各フェーズで別のパッケージの後に処理する必要があることを、パッケージ・インストーラに認識させる必要があります。それには、以下に示すように、現在のパッケージの `manifest.isc` ファイルに “prerequisite” の各値を追加します。

```
~/kit/package/foo9000:>emacs manifest.isc
package: foo9000
prerequisite: server_location
prerequisite: legacy_dist
prerequisite: platform_selection
```

これで、指定したパッケージからパラメータ変数をインポートし、現在のパッケージのインストール・スクリプトの実行に使用できるようになります。

```
~/kit/package/foo9000:>emacs install
#!/bin/sh
. parameters.include
Import host foo9000.host
Import tgtmdir "server_location.target_dir"
Import srcdir "legacy_dist.source_dir"
Import platform_family "platform_selection.platform_family"
echo host=$host > $tgtmdir/mgr/foo9000.cfg
cp $srcdir/dist/foo9000/$platform_family/libfoo9000.so $tgtmdir/bin
```

パッケージ **foo9000** はほぼ完成しました。最後の作業として、関連する既存のパッケージの prerequisite リストにこのパッケージを追加します。それにより、パッケージ・インストーラは、その関連パッケージのインストールを完了するために、今回作成したパッケージを処理するようになります。この場合、InterSystems IRIS サーバのインストール時に必ずこのライブラリのインストールと構成が行われるようにする必要がありますため、「database_server」パッケージの **manifest.isc** ファイルで、このパッケージの prerequisite リストに新しいパッケージを追加します。

```
~/kit/package/database_server:>emacs manifest.isc
package: database_server
prerequisite: legacy_dist
prerequisite: platform_selection
prerequisite: server
prerequisite: server_location
prerequisite: upgrade
prerequisite: available_disk_space
prerequisite: posix_tools
...
prerequisite: isql
prerequisite: zlib
prerequisite: udp
prerequisite: bi
prerequisite: foo9000
```

上のリストから明らかなように、サーバ・インストールの作成には多くのパッケージが必要ですが、これで、**irisinstall** を実行すると、パッケージ **foo9000** の構成とインストールが実行されるようになりました。

```
~/kit:>sudo ./irisinstall
Your system type is 'SuSE Linux Enterprise Server 10 (x64)'.
Currently defined instances:
IRIS instance 'TRWTURBO'
directory: /home/testUser/TRWTURBO
versionid: 2018.1.0.508.0
conf file: iris.cpf (SuperServer port = 1972, WebServer = 57785)
status: crashed, last used Sat Sep 22 08:37:32 2018
Enter instance name: TRWPACK1
Do you want to create IRIS instance 'TRWPACK1' ? Y
...
Please enter host name of the Foo9000 imaging server:
cupertino
Host cupertino entered.
...
Do you want to proceed with the installation ? Y
...
Installation completed successfully
~/TRWPACK1/bin:>ls libfoo*
libfoo9000.so
~/TRWPACK1/mgr:>cat foo9000.cfg
host=cupertino
```

B.2 parameters.isc ファイルの内容

以下の表は、**parameters.isc** ファイルに記載されている変数とその説明、およびサンプル値、または有効な値の一覧です。

テーブル II-1: InterSystems IRIS のインストール・パラメータ・ファイルの変数

変数名	説明 ()
dist.source_dir	インストール媒体のソース・ディレクトリ。 /iriskit
legacy_dist.source_dir	従来の目的では、インストール媒体のソース・ディレクトリ。 /iriskit
product_info.version	インターシステムズ社製品のバージョン番号。 2018.1.0.100.0
product_info.name	インターシステムズ社製品の名前。 InterSystems IRIS
platform_selection.platform	インストール先プラットフォームに対するインターシステムズ社の略称。 lnxrhx64
platform_selection.platform_family	インストール先プラットフォーム・ファミリーに対するインターシステムズ社の略称。 lnxrhx64
platform_selection.endianness	プラットフォームのエンディアン・バイト・オーダー。 (big/little)
platform_selection.os	プラットフォームのオペレーティング・システム。uname コマンドの値。 Linux
posix_tools.user_add	Portable Operating System Interface (POSIX) 準拠のユーザ追加ツール。 /usr/sbin/useradd
posix_tools.group_add	POSIX 準拠のグループ追加ツール。 /usr/sbin/groupadd
posix_tools.grep	POSIX 準拠の grep ユーティリティ。 grep
posix_tools.id	POSIX 準拠の id ユーティリティ。 id
posix_tools.ps_opt	プロセスのリスト表示ですべて展開するオプション。 -ef
posix_tools.gzip	Gnu 準拠の zip ユーティリティ。 gzip

変数名	説明 ()
posix_tools.shared_ext	共有ライブラリ・ファイルの拡張子。 so
posix_tools.symbolic_copy	POSIX 準拠のシンボル・コピー・コマンド。 cp -Rfp
posix_tools.tr	POSIX 準拠の変換ユーティリティ。 tr
posix_tools.shared_ext1	共有ライブラリ・ファイルの代替の拡張子。 so
posix_tools.permission	選択されたファイルに適用される POSIX 準拠の特権。 755
posix_tools.dir_permission	選択されたディレクトリに適用される POSIX 準拠の特権。 775
server_location.target_dir	サーバ・インストールのターゲット・ディレクトリ。 /test/IRIS
server_location.is_server_install	これがサーバ・インストールであるかどうかを示す。 (N/Y)
server_location.is_nonroot_install	これが非 root インストールであるかどうかを示す。 (N/Y)
server_location.instance_name	インスタンス名。 IRIS
server_location.is_new_install	これが新規インストールであるかどうかを示す。 (N=アップグレード/Y=新規)
server_location.is_new_directory	新規ディレクトリを作成するかどうかを示す。 (N/Y)
server_location.registry_dir	InterSystems IRIS レジストリ・ディレクトリの場所 (ローカルの ファイル・システム上に配置する必要がある)。 /usr/local/etc/irissys
server_location.iris	インストール中に、iris が存在するディレクトリ。 /iriskit/dist/lnxrhx64/bin/shared/iris
server_location.is_dr_mirror	このインスタンスが災害復旧 (DR) ミラー・メンバかどうかを示す。 (N/Y)

変数名	説明 ()
postinstall*	パラメータ・ファイル・フェーズの後で実行するパッケージを指定。 upgrade
install_mode.setup_type	インストールのタイプ。 (Development/Server/Custom)
unicode_selection.binary_type	インストールするバイナリのタイプ。 (unicode/eightbit)
unicode_selection.install_unicode	この製品の Unicode 版をインストールするかどうかを示す。 (N/Y)
security_settings.iris_user	InterSystems IRIS スーパーサーバの実効ユーザ irisusr
security_settings.iris_group	InterSystems IRIS の実効グループ irisusr
security_settings.manager_user	インスタンスの所有者。 root
security_settings.manager_group	インスタンスの開始および停止を許可されたグループ。 develop
security_settings.dbencrypted	起動時に暗号化キーを有効にするかどうかを示す。 (0/1)
security_settings.dbenckeyfile	暗号化キーのパス。 このパラメータは空白でもかまいません。
security_settings.dbenckeyuser	キーを有効にできる管理者の名前。 このパラメータは空白でもかまいません。
security_settings.dbenckeypassword	キー管理者のパスワード。パラメータ・ファイルが保存される前にクリアされます。 このパラメータは空白でもかまいません。
security_settings.personal_database	パーソナル・データベース機能を使用するかどうかを示す。 (N/Y)
security_settings.initial_level	初期のセキュリティ設定。 (Minimal/Normal/LockedDown)
security_settings.already_secured	5.1 以前のインスタンスからアップグレードしている場合に、セキュリティ設定が必要かどうかを示す。 (N/Y)

変数名	説明 ()
security_settings.password	irisinstall から実行している場合は、パラメータ・ファイルの格納前にパスワード・フィールドがクリアされる。
installer.manifest	DefaultInstallerClass.xml (エクスポートされた %Installer クラス) の場所。以下はその例です。 /home/user/Downloads/DefaultInstallerClass.xml
installer.manifest_parameters	インストーラ・マニフェストのパラメータの場所。 SourceDir=/home/user/Downloads
installer.manifest_loglevel	マニフェストのログ・レベルを指定する。 (-1/0/1/2/3)
installer.manifest_logfile	ログ・ファイル名を指定する。 /manifests/IRIS-installManifestLog.txt
port_selection.superserver_port	スーパーサーバ・ポート番号。 1972
port_selection.webserver_port	Web サーバのポート番号。 52773
port_selection.jdbcgateway_port	Java Database Connectivity (JDBC) ゲートウェイ・ポート番号。 62972
csp_gateway.configure	外部 Web サーバ用に Web ゲートウェイを構成するかどうかを示す。 (N/Y)
csp_gateway.web_server_type	Web ゲートウェイで使用する既存の Web サーバのタイプ。 (Apache/SunOne/None)
csp_gateway.apache_version	Apache Web サーバのバージョン
csp_gateway.apache_user	Apache Web サーバのユーザ名
csp_gateway.apache_conf_file	Apache Web サーバ構成ファイルの場所。 /etc/httpd/conf/httpd.conf
csp_gateway.apache_pid_file	Apache Web サーバ・デーモンのプロセス ID を記録するファイル。 /usr/local/apache/logs/httpd.pid
csp_gateway.apache_use32bit	Apache Web サーバで 32 ビット・アーキテクチャを使用するかどうかを示す。 Y/N

変数名	説明 ()
csp_gateway.sunone_server	Web ゲートウェイで使用する Sun ONE サーバの場所。 /usr/netscape/server4/httpd-production
csp_gateway.directory	Web ゲートウェイ・ファイルが保存されるディレクトリ。
license_key.enter_key	インストール中にキーをインストールするかどうかを示す。 N/Y
license_key.license_file	enter_key の値が Y の場合、キー・ファイル情報の場所。
agent.user_account	ISCAgent のユーザ名。 iscagent
agent.user_group	ISCAgent のグループ名。 iscagent
agent.install	ISCAgent がインストールされているかどうかを示す。 (N/Y)
client_location.target_dir	クライアントのみのインストールのターゲット・ディレクトリ。 test/IRIS
client_location.is_client_install	これがクライアント・インストールであるかどうかを示す。 (N/Y)
install*	database_server
postinstall*	database_server
japanese_docs.install	日本語ドキュメント・ソースをインストールするかどうかを示す。 (N/Y)
install*	インストールするコンポーネント名。

* このインストール変数は、インストールするコンポーネント 1 つにつき 1 回ずつ、パラメータ・ファイルに登場します。カスタム・インストール、またはクライアントのみのインストールでは、条件に応じて、以下のいずれか、またはすべてが生成されます。

- ・ dev_kit
- ・ odbc
- ・ cpp_binding
- ・ cpp_sdk
- ・ engine_link_libraries
- ・ light_cpp_binding
- ・ addenda
- ・ install_confirmation
- ・ copyright

