



# InterSystems IRIS の基礎： IDE の接続

Version 2024.1  
2024-06-03

InterSystems IRIS の基礎 : IDE の接続

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

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

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

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼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](mailto:support@InterSystems.com)

# 目次

InterSystems IRIS の基礎 : IDE の接続.....	1
1 InterSystems IRIS の導入 .....	1
1.1 ライセンス付与されたインスタンスの導入 .....	1
1.2 無料の評価版インスタンスの導入 .....	2
2 InterSystems IRIS 接続情報 .....	3
2.1 InterSystems Kubernetes Operator .....	4
2.2 コンテナに導入された InterSystems IRIS .....	4
2.3 キットからインストールされた InterSystems IRIS .....	5
2.4 InterSystems IRIS Community Edition .....	5
2.5 InterSystems Learning Labs .....	5
3 ObjectScript IDE .....	6
3.1 Visual Studio Code .....	6
3.2 スタジオ .....	6
4 Java IDE .....	6
4.1 Visual Studio Code .....	6
4.2 Eclipse .....	7
4.3 IntelliJ .....	8
4.4 NetBeans .....	8
5 .NET IDE .....	9
5.1 Visual Studio .....	9
5.2 Visual Studio Code .....	9
6 Python IDE .....	10
7 JavaScript IDE .....	11



# InterSystems IRIS の基礎：IDE の接続

このドキュメントでは、GitHub からダウンロードしたテンプレート・ファイルを使用して、ObjectScript、Java、.NET、Python、または JavaScript/Node.js 向けの統合開発環境 (IDE) を InterSystems IRIS® のインスタンスにすばやく接続し、正常に接続されていることを確認する方法について説明します。

そのほかに、次の 2 つのセクションでは以下の内容について説明します。

- ・ [InterSystems IRIS の導入](#)

InterSystems IRIS のライセンス付与されたインスタンスと無料の評価版インスタンスの両方について導入オプションを紹介します。操作するインスタンスがまだない場合は、このセクションを確認してください。

- ・ [InterSystems IRIS 接続情報](#)

IDE を InterSystems IRIS のライセンス付与されたインスタンスまたは無料インスタンスに接続するため、およびその他さまざまな方法でそのインスタンスを操作するために必要な詳細情報を示します。また、インスタンスに関する情報を取得する方法について説明します。InterSystems IRIS に接続する IDE のカテゴリに関係なく、操作対象のインスタンスのタイプについて説明しているセクションを確認してください。

## 1 InterSystems IRIS の導入

このセクションでは、[ライセンス付与されたインスタンス](#)と[無料の評価版インスタンス](#)の両方について説明します。

### 1.1 ライセンス付与されたインスタンスの導入

インターシステムズから 1 つ以上のライセンスを購入した場合は、以下のいずれかの方法を使用して、ライセンス付与された InterSystems IRIS インスタンスを導入できます。

- ・ [InterSystems Kubernetes Operator](#)
- ・ [InterSystems IRIS コンテナ・イメージ](#)
- ・ [InterSystems IRIS インストール・キット](#)

コンテナ・イメージまたはインストール・キットから導入する場合、[構成マージ機能](#)を使用して自動導入をサポートできます。

#### 1.1.1 InterSystems Kubernetes Operator

[Kubernetes](#) は、コンテナ化されたワークロードとサービスの導入、拡張、および管理を自動化するためのオープンソースのオーケストレーション・エンジンです。導入するコンテナ化されたサービスと、そのサービスを管理するポリシーを定義すると、Kubernetes は、必要なリソースを可能な限り効率的な方法で透過的に提供します。また、導入が指定値から外れた場合は導入を修復またはリストアするほか、拡張を自動またはオンデマンドで行います。InterSystems Kubernetes Operator (IKO) は、IrisCluster カスタム・リソースで Kubernetes API を拡張します。このリソースは、InterSystems IRIS のシャード・クラスタ、分散キャッシュ・クラスタ、またはスタンドアロン・インスタンスとして、すべて任意でミラーリングした状態で、Kubernetes プラットフォームに導入できます。IKO は、InterSystems IRIS 固有のクラスタ管理機能も Kubernetes に追加して、クラスタにノードを追加するなどのタスクの自動化を可能にします。このようなタスクは、自動化されなければ、インスタンスを直接操作して手動で行わなければなりません。IKO の使用方法の詳細は、“[InterSystems Kubernetes Operator の使用](#)” を参照してください。

## 1.1.2 InterSystems IRIS コンテナ・イメージ

インターシステムズのコンテナ・イメージでは、ユーザ独自のツールと方法を使用して、Linux ホスト(クラウド、仮想、または物理) 上のコンテナに InterSystems IRIS を導入できます。インスタンス固有のデータの永続ストレージによって、コンテナ化された InterSystems IRIS のアップグレードが簡単になるため、いつでも既存の構成を最新バージョンの InterSystems IRIS に、トラブルやダウンタイムを最小限に抑えながら簡単に移行できます。

**注釈** インターシステムズのコンテナ・イメージは Open Container Initiative (OCI) の仕様に準拠しているため、Linux ベースのオペレーティング・システム上の任意の OCI 準拠ランタイム・エンジンでサポートされます。このドキュメントの以降の部分では、Docker ランタイム・エンジンの使用を想定しています。

InterSystems IRIS イメージの詳細な使用ガイドは、“[コンテナ内でのインターシステムズ製品の実行](#)” を参照してください。InterSystems IRIS イメージを取得する方法の詳細は、“コンテナ内でのインターシステムズ製品の実行” の “[InterSystems IRIS イメージのダウンロード](#)” を参照してください。

Google Cloud Platform、Microsoft Azure、または Amazon Web Services のパブリック・クラウド・プラットフォームにインストールされた InterSystems IRIS コンテナ・イメージと Docker を使用して、BYOL (ライセンス持ち込み) クラウド・ノードをプロビジョニングしてから、自分のライセンスを追加して、このイメージから InterSystems IRIS コンテナを実行できます (“コンテナ内でのインターシステムズ製品の実行” を参照)。詳細は、“InterSystems IRIS の導入と操作” の “[クラウド・ノードへの InterSystems IRIS BYOL の導入](#)” を参照してください (これらのドキュメントは、無料の [InterSystems IRIS Community Edition イメージ](#)でも使用できます)。

## 1.1.3 InterSystems IRIS インストール・キット

[インターシステムズのサポート窓口](#)のダウンロード・エリアからお使いのプラットフォーム向けの最新のインストール・キットを入手し、ローカル・マシンまたはネットワーク上の別のマシンに InterSystems IRIS の開発インスタンスをインストールしてライセンスを取得できます。インストール手順は、“[インストール・ガイド](#)” を参照してください。

### 1.1.4 構成マージを使用した導入

構成マージ機能を使用すると、同じイメージから導入した InterSystems IRIS コンテナや、同じキットからインストールしたローカル・インスタンスの構成を変更できます。構成マージ機能は、以下の方法で使用できます。

- ・ 構成マージを使用した導入

Linux および UNIX® システムでは、[構成マージを導入で使用](#)して、インスタンスの構成パラメータ・ファイル (CPF) を更新できます。CPF にはインスタンスの構成設定が含まれており、これらの設定は開始時に毎回、インスタンス導入後の最初の設定を含めて CPF から読み取られます。導入時に構成マージを適用すると、インスタンスと共に提供された既定の CPF が実質的に独自の更新バージョンに置き換えられます。この機能を、既存のインスタンスの再起動時に使用することもできます。

- ・ 実行中のインスタンスで構成マージを使用

Linux/UNIX® と Windows の両方のシステムで、[iris merge コマンド](#)を使用して実行中のインスタンスを再構成し、その CPF をマージ・ファイルのコンテンツで更新できます。

[InterSystems Kubernetes Operator](#) には、構成マージ機能が組み込まれています。

構成マージの使用法の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)” を参照してください。

## 1.2 無料の評価版インスタンスの導入

まだ InterSystems IRIS ユーザではない場合は、無料の評価版インスタンスを導入することで、InterSystems IRIS のさまざまな機能の詳細を確認し、お好みの言語とツールおよび独自のコードを使用してそれらの機能を試してみることができます。これを行うには、以下のような方法があります。

- ・ パブリック・クラウド・ノードで、Google Cloud Platform、Microsoft Azure、または Amazon Web Services のアカウントを作成するか、いずれかにログインして、プラットフォームのマーケットプレースのページに進み、InterSystems IRIS を検索します。
- ・ 任意のシステムのコンテナで、InterSystems Container Registry (ICR) または Docker Hub から Community Edition イメージをプルします。

### 1.2.1 InterSystems IRIS Community Edition

InterSystems IRIS Community Edition には 13 か月分の無料の組み込みライセンスが付属しています (これにはいくつかの機能制限があります)。Community Edition は以下の 2 つの方法で導入できます。

- ・ Google Cloud Platform、Microsoft Azure、または Amazon Web Services のパブリック・クラウド・プラットフォーム上で稼働している Community Edition コンテナをホストするクラウド・ノードをプロビジョニングします。
- ・ InterSystems Container Registry (ICR) または Docker Hub から Community Edition イメージをプルし、それを使用して任意のシステムにコンテナを導入します。

クラウドまたは独自のシステムへの InterSystems IRIS Community Edition の導入および使用の手順は、“InterSystems IRIS の導入と操作” を参照してください。

### 1.2.2 InterSystems Learning Labs

InterSystems Learning Labs Web ページからは、30 日間無料ライセンスで利用できる、トレーニング、開発、テスト用の InterSystems IRIS のラボ・インスタンスをすばやく簡単に起動できます。InterSystems Learning Labs インスタンスには、統合 IDE および操作を試すことができるサンプルが豊富に含まれ、ユーザ自身の IDE を接続できます。

InterSystems Learning Labs インスタンスを起動するには、[learning.intersystems.com](https://learning.intersystems.com) にログインしている必要があります。まだアカウントがない場合は簡単に作成できます (Community Edition と同様に、Learning Labs インスタンスにもいくつかの機能制限があります)。

## 2 InterSystems IRIS 接続情報

IDE を InterSystems IRIS インスタンスに接続するには、まずインスタンスを選択してから、以下の必要な接続情報を指定する必要があります。

- ・ ホスト ID : インスタンスのホストの IP アドレスまたはホスト名
- ・ インスタンスのスーパーサーバ・ポート番号
- ・ インスタンスの有効な資格情報
- ・ インスタンス上の既存のネームスペースの名前

インスタンスの Web ベースの管理ポータル URL、およびインスタンスで InterSystems ターミナルを開く方法 (インスタンス名が必要になります) を確認することも有用です。

こういった接続情報をどのように指定するかは、操作するインスタンスを以下のいずれの方法で導入するかによって異なります。

- ・ InterSystems Kubernetes Operator
- ・ コンテナに導入された InterSystems IRIS
- ・ キットからインストールされた InterSystems IRIS
- ・ InterSystems IRIS Community Edition

## 2.1 InterSystems Kubernetes Operator

Kubernetes の導入に、必要に応じて外部 IP アドレスを使用して、ポッドとポッドに含まれるコンテナをネットワークに公開するサービスを含めることができます。IKO で導入する InterSystems IRIS トポロジに関係なく、プライマリ・データ・ノード (シャード・クラスタのデータ・ノード 1、分散キャッシュ・クラスタのデータ・サーバ、またはスタンドアロン・インスタンス) を公開するためのサービスが常に作成されます。このサービスを使用して、プライマリ・データ・ノード上の InterSystems IRIS インスタンスのスーパーサーバ・ポート (1972) または Web サーバ・ポート (80) に接続する方法の詳細は、“InterSystems Kubernetes Operator の使用” の [“IrisCluster への接続”](#) を参照してください。

IDE 接続または管理ポータルに必要な認証情報には、[事前定義のユーザ・アカウント](#)の 1 つと configSource フィールドで指定した構成マージ・ファイル内の PasswordHash パラメータを使用して設定した既定のパスワード、あるいはインスタンスに以前追加したアカウントとパスワードを使用します。

プライマリ・データ・ノード・インスタンスのターミナルを開くには、以下のコマンドを使用します。

```
kubect1 exec -it IrisCluster_name-data-0 iris terminal IRIS
```

導入がミラーリングされている場合は、代わりにポッド名 IrisCluster\_name-data-0-0 を使用して、現在のプライマリでターミナルを開きます。インターシステムズのどのコンテナでも、インスタンス名は常に **IRIS** です。InterSystems IRIS コンテナの外部から docker exec を使用して発行されたコマンド、および結果的に kubect1 exec は、[コンテナ内部で irisowner として実行](#)され、認証を必要としないため、ターミナルにログインする必要はありません。

すべての InterSystems IRIS インスタンスに存在する **USER** ネームスペースには常に IDE を接続できますが、管理ポータルを使用して以前作成した別のネームスペースに接続することもできます。

## 2.2 コンテナに導入された InterSystems IRIS

“コンテナ内でのインターシステムズ製品の実行” の [“Web ゲートウェイ・コンテナを使用した Web アクセス”](#) で詳細に説明しているように、InterSystems IRIS 管理ポータルは組み込みの Web アプリケーションであるため、これをブラウザにロードするには、Web サーバおよびインターシステムズの Web ゲートウェイが必要です。ただし、複数の InterSystems IRIS コンテナとやり取りするよう構成されている単一の Web ゲートウェイ・インスタンスでは、すべてのインスタンスに共通するアプリケーションの要求を特定のインスタンスに転送することはできません。

管理ポータルからコンテナ化された InterSystems IRIS インスタンスにアクセスできるようにする簡単な方法の 1 つとして、各 InterSystems IRIS コンテナに専用の Web ゲートウェイ・コンテナ (Web サーバも含む) を実行する方法があります。専用の Web ゲートウェイ・コンテナは、ペアになっている InterSystems IRIS コンテナとのみやり取りするよう構成されます。これは、Web サーバ層が必要な導入の場合、さらに Web ゲートウェイ・コンテナを追加する必要があります。専用の Web ゲートウェイ・コンテナおよび Web サーバ・ノードの Web ゲートウェイ・コンテナを実行する方法、および管理ポータルからコンテナ化された InterSystems IRIS インスタンスにアクセスできるようにするその他の方法に関する詳細と手順は、[“Web ゲートウェイ・コンテナの実行オプション”](#) を参照してください。

**重要** InterSystems IRIS Community Edition コンテナは前述の説明には当てはまりません。Community Edition インスタンスの管理ポータルにアクセスする方法の詳細は、“InterSystems IRIS の導入と操作” の [“管理ポータルを使用した操作”](#) を参照してください。

IDE 接続または管理ポータルに必要な認証情報には、[事前定義のユーザ・アカウント](#)の 1 つとコンテナの作成時に[設定した既定のパスワード](#)、あるいはインスタンスに以前追加したアカウントとパスワードを使用します。

コンテナ内のインスタンスのターミナルを開くには、コンテナ内で docker exec コマンドを使用して [iris terminal](#) コマンドを実行するか、コンテナ内で docker exec を使用してシェルを開き、そのシェルから iris terminal を実行できます。使用例は、“InterSystems IRIS の導入と操作” の [“InterSystems ターミナルを使用した操作”](#) を参照してください。インターシステムズのどのコンテナでも、インスタンス名は常に **IRIS** です。InterSystems IRIS コンテナの外部から docker exec を



使用して発行したコマンドは、**コンテナ内部で `irisowner` として実行**され、認証を必要としないため、ターミナルにログインする必要はありません。

すべての InterSystems IRIS インスタンスに存在する **USER** ネームスペースには常に IDE を接続できますが、管理ポータルを使用して以前作成した別のネームスペースに接続することもできます。

## 2.3 キットからインストールされた InterSystems IRIS

**キットからインストールする** InterSystems IRIS インスタンスの場合、ホスト ID はインスタンスが実行されているシステムのホスト名と IP アドレスです。ローカルにインストールする場合は **localhost** を使用できます。

InterSystems IRIS のインストールでは、スーパーサーバのポートは既定で 1972 に設定されます。ただし、1 つのシステムに InterSystems IRIS のインスタンスを複数インストールしている場合は、インスタンスのスーパーサーバのポート番号は変わります。すべてのインスタンスのポート番号を表示するには、オペレーティング・システムのコマンド行で **iris list** コマンドを使用します。

管理ポータルを開くには、すべてのシステムで、“Web ゲートウェイ・ガイド”の“**Web サーバを使用した管理ポータルおよび他の組み込みの Web アプリケーションへのアクセス**”に記載されている URL を使用できます。Windows システムの場合、**Web ゲートウェイおよび Web サーバが正しく構成**されていれば、システムトレイにある InterSystems IRIS のアイコンをクリックして**【管理ポータル】**を選択することで、管理ポータルを開くことができます。

インストールしたインスタンスのターミナルを開くには、次の操作を行います。

- ・ Windows システムの場合、InterSystems IRIS **ランチャー・メニュー**の**【ターミナル】**オプションを選択できます。
- ・ すべてのシステムで、オペレーティング・システムのコマンド行で、引数としてインスタンス名を指定して **iris terminal** コマンドを使用できます。インスタンス名は、インスタンスをインストールした際に設定され、変更することはできません。インスタンス名を表示するには、**iris list** コマンドを使用します。

すべての目的で、資格情報には、**事前定義のユーザ・アカウント**の 1 つと既定のパスワード **SYS** (これらのアカウントの 1 つにログインすると、直ちにパスワードを変更するよう求められます)、またはインスタンスに以前追加したアカウントとパスワードを使用します。

すべての InterSystems IRIS インスタンスに存在する **USER** ネームスペースには常に IDE を接続できますが、管理ポータルを使用して以前作成した別のネームスペースに接続することもできます。

## 2.4 InterSystems IRIS Community Edition

Community Edition インスタンスで管理ポータルにアクセスする方法の詳細は (この点に関しては、他のタイプのインスタンスと異なります)、“InterSystems IRIS の導入と操作”の“**管理ポータルを使用した操作**”を参照してください。

任意の Community Edition インスタンスのターミナルを開く方法の詳細は、“InterSystems IRIS の導入と操作”の“**InterSystems ターミナルを使用した操作**”を参照してください。

IDE 接続および管理ポータルに必要な認証情報には、**事前定義のユーザ・アカウント**の 1 つと、**クラウド・ノードに初めて接続した際のパスワードの変更**で指定した新しい既定のパスワード、または既定パスワード **SYS** (独自のシステムにいるか、最初にノードに接続せずにクラウド・インスタンスにログインした場合)を使用します。**SYS**を使用する場合、使用しているアカウントのパスワードをすぐに変更するよう求められます。インスタンスで以前作成したアカウントを使用することもできます。

すべての InterSystems IRIS インスタンスに存在する **USER** ネームスペースには常に IDE を接続できますが、管理ポータルを使用して以前作成した別のネームスペースに接続することもできます。

## 2.5 InterSystems Learning Labs

**InterSystems Learning Labs** の構成に含まれるコンテナ化された InterSystems IRIS インスタンスへの接続はすべて Learning Labs の起動時に設定され、以下の必要な接続情報はすべて起動ページに表示されます。

- ・ 統合 IDE の URL。
- ・ すべての目的で使用できる一連の資格情報。
- ・ 管理ポータル の URL。
- ・ Web サーバのポート番号 (常に 80)。
- ・ **[外部接続]** の下に表示されるサーバの IP アドレスおよびスーパーサーバのポート番号 (**[InterSystems IP]** および **[InterSystems ポート]** というラベルが付いている)。これらを使用して IDE (InterSystems スタジオや Object Script 拡張機能付きの Visual Studio Code など) をインスタンスに接続できます。

統合 IDE でターミナルのオプションは **[InterSystems]** メニューから使用できる (**管理ポータル** のオプションも同様) ので、常にインスタンス名に気を配る必要はありません。

## 3 ObjectScript IDE

ここでは、InterSystems IRIS インスタンスでの ObjectScript コードの開発に使用できる IDE (ObjectScript 拡張機能付きの [Visual Studio Code](#) (VS Code) および [スタジオ](#)) について説明します。

### 3.1 Visual Studio Code

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。VS Code は、[.NET IDE](#)、[Python IDE](#)、または [JavaScript IDE](#) としてすぐに InterSystems IRIS で使用できますが、提供されている拡張機能を使用すると、VS Code を利用して ObjectScript でコードを開発することも可能です。このコードは、クライアント・システムに保存されてバージョン管理され、InterSystems IRIS サーバでコンパイル、実行、およびデバッグされます。VS Code は、InterSystems IRIS インスタンスが実行されているプラットフォームに関係なく、ホストの DNS 名か IP アドレス、インスタンス名、およびインスタンスの Web サーバのポート番号を使用して、あらゆる InterSystems IRIS インスタンスに接続できます。インストールと使用の手順は、"[Use Visual Studio Code as a Development Environment for InterSystems Applications](#)" を参照してください。

### 3.2 スタジオ

InterSystems IRIS アプリケーション開発者はスタジオ IDE を使用することもできます。スタジオ IDE は、Windows システム上で動作するクライアント・アプリケーションです。詳細は、"[スタジオの使用法](#)" を参照してください。スタジオは、インスタンスが実行されているプラットフォームに関係なく、ホストの DNS 名か IP アドレス、インスタンス名、およびインスタンスのスーパーサーバのポート番号 (既定で 1972 ですが、システムに複数のインスタンスがインストールされている場合は異なる可能性があります) を使用して、あらゆる InterSystems IRIS インスタンスに接続できます。

## 4 Java IDE

ここでは、[Visual Studio Code](#) (Coding Pack for Java または Java Extension Pack に含まれる)、[Eclipse](#)、[IntelliJ](#)、および [NetBeans](#) を使用する場合は接続手順を説明します。これらは、InterSystems Native SDK、および [JDBC](#) を使用して InterSystems IRIS を操作する Java コードの開発に使用できます。

### 4.1 Visual Studio Code

Visual Studio Code を InterSystems IRIS に接続するには、以下の手順を使用します (必要に応じて GitHub にログインします)。

1. Web ブラウザで、<https://github.com/intersystems/Samples-java-helloworld> をダウンロードまたは複製します。
2. Visual Studio Code で、[ファイル]→[フォルダーを開く] を使用して、GitHub から最近ダウンロードまたは複製したフォルダを開きます。
3. `src/main/java.com.intersystems.samples.helloworld.java` を選択して開きます。
4. メイン・メソッドの上部にある変数宣言でインスタンスのユーザ名、パスワード、IP アドレス、およびスーパーサーバのポートを構成します（“[InterSystems IRIS 接続情報](#)”を参照）。これらの値は、`jdbc:IRIS://ipAddress:webserverPort/namespace`（例：`jdbc:IRIS://12.345.678.910:1972/USER`）の形式の InterSystems JDBC 接続文字列の作成に使用します。
5. [表示]→[端末] を使用して端末を開き、以下のコマンドを実行します。

```
javac -cp ".:intersystems-jdbc-3.2.0.jar" HelloWorld.java
java -cp ".:intersystems-jdbc-3.2.0.jar" HelloWorld
```
6. [出力] ペインに、以下のメッセージが表示されます。“Hello World! You have successfully connected to InterSystems IRIS via JDBC.”

## 4.2 Eclipse

Eclipse を InterSystems IRIS に接続するには、以下の手順を使用します（必要に応じて、各ダイアログの次のパネルに進むには、[次へ] をクリックします）。

まず、サンプル・パッケージをインポートします。

1. [ファイル]→[インポート]→[Git]→[Git からプロジェクト] を選択します。
2. [URI の複製] を選択します。<https://github.com/intersystems/Samples-java-helloworld> をコピーして、[URI] フィールドに貼り付けます。
3. [master] ブランチを選択し、ローカル・ストレージを構成して、[Import existing Eclipse projects] を選択します。
4. [終了] をクリックしてインポートを確定します。

次に、サンプル・コードを更新して、実行します。

1. [samples-java-helloworld]→[src]→[main.java.com.intersystems.samples]→[HelloWorld.java] を開き、以下のように変更します。
  - a. 上部にあるパッケージ宣言を `[package main.java.com.intersystems.samples;]` に変更します。
  - b. メイン・メソッドの上部にある変数宣言でインスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します（“[InterSystems IRIS 接続情報](#)”を参照）。これらの値は、`jdbc:IRIS://ipAddress:superserverPort/namespace`（例：`jdbc:IRIS://12.345.678.910:1972/USER`）の形式の InterSystems JDBC 接続文字列の作成に使用します。
2. [実行]→[実行] を選択して、コードを実行します。
3. [コンソール] タブに、以下のメッセージが表示されます。“Hello World! You have successfully connected to InterSystems IRIS via JDBC.”

成功しなかった場合は、以下の手順を実行して、Java 実行環境が適切に設定されていることを確認します。

1. [パッケージ・エクスプローラー] ペインで [samples-java-helloworld] プロジェクトを右クリックして、[ビルド・パス]→[ビルド・パスの構成...] を選択します。
2. [ライブラリー] タブで、[ライブラリーの追加]→[JRE システム・ライブラリ]→[実行環境] を選択し、[JRE-1.1 (jre 1.8.0\_172)] などの実行環境を選択します。[完了]、[適用して閉じる] を順にクリックします。

## 4.3 IntelliJ

IntelliJ を InterSystems IRIS に接続するには、以下の手順を使用します（必要に応じて、各ダイアログの次のパネルに進むには、**[Next]** をクリックします）。

まず、サンプル・プロジェクトを作成します。

1. **[VCS]**→**[Checkout from Version Control]**→**[Git]** を選択します。
2. **[Clone Repository]** ダイアログで以下の操作を行います。
  - a. <https://github.com/intersystems/Samples-java-helloworld> をコピーして、**[Clone URI]** フィールドに貼り付けます。
  - b. **[Directory]** フィールドに、ローカル GitHub リポジトリの作成場所へのパスを入力します。
3. **[Clone]** をクリックし、このソースに基づいてプロジェクトを作成することを確認するポップアップで **[Yes]** を選択します。

次に、サンプル・コードを更新して、実行します。

1. `src/main/java.com.intersystems.samples.helloworld.java` を開き、メイン・メソッドの上部にある変数宣言でインスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します（“[InterSystems IRIS 接続情報](#)”を参照）。これらの値は、`jdbc:IRIS://ipAddress:superserverPort/namespace`（例：`jdbc:IRIS://12.345.678.910:1972/USER`）の形式の InterSystems JDBC 接続文字列の作成に使用します。
2. **[Project]** ペインで、**[HelloWorld.java]** を右クリックして、**[Run ‘HelloWorld.main()’]** を選択します。
3. **[出力]** ペインに、以下のメッセージが表示されます。“Hello World! You have successfully connected to InterSystems IRIS via JDBC.”

成功しなかった場合は、以下の手順を実行して、Java 実行環境が適切に設定されていることを確認します。

1. **[Build Path]**→**[Build Project]** を選択します。
2. 構成を編集し、メイン・クラスに `[com.intersystems.samples.HelloWorld]` を選択して、新しいアプリケーション構成を追加します。
3. エラー “Error:java: invalid source release: 9” が表示されたら、以下の場所でプロジェクト SDK とプロジェクト言語レベルを 1.8 に変更します。
  - ・ **[File]**→**[Project Structure]**→**[Project Settings]**
  - ・ **[File]**→**[Project Structure]**→**[Module Settings]**→**[Sources: Language Level]** タブ
  - ・ **[File]**→**[Project Structure]**→**[Module Settings]**→**[Dependencies: Module SDK]** タブ

## 4.4 NetBeans

NetBeans を InterSystems IRIS に接続するには、以下の手順を使用します（必要に応じて、各ダイアログの次のパネルに進むには、**[次]** をクリックします）。

まず、サンプル・プロジェクトを作成します。

1. **[チーム]**→**[Git]**→**[クローン]** を選択します。
2. <https://github.com/intersystems/Samples-java-helloworld> をコピーして、**[リポジトリURL]** フィールドに貼り付けます。

3. フェッチする分岐として **[Master]** を選択し、ローカル GitHub リポジトリの作成場所を選択して、**[終了]** をクリックします。

次に、サンプル・コードを更新して、実行します。

1. **HelloWorld.java** を開き、メイン・メソッドの上部にある変数宣言でインスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します（“[InterSystems IRIS 接続情報](#)”を参照）。これらの値は、`jdbc:IRIS://ipAddress:superserverPort/namespace`（例：`jdbc:IRIS://12.345.678.910:1972/USER`）の形式の InterSystems JDBC 接続文字列の作成に使用します。
2. **[プロジェクト]** ペインで、**[依存性]** フォルダを開き、**[intersystems-jdbc-3.0.0.jar]** を右クリックして、**[アーティファクトを手動でインストール]** を選択します。最近複製したフォルダに移動して、**[intersystems-jdbc-3.0.0.jar]** を選択し、**[ローカルにインストール]** をクリックします。
3. **[プロジェクト]** ペインで **HelloWorld.java** を右クリックして、**[ファイルの実行]** を選択します。
4. **[出力]** ペインに、以下のメッセージが表示されます。“Hello World! You have successfully connected to InterSystems IRIS via JDBC.”

## 5 .NET IDE

このセクションでは、Microsoft の [Visual Studio](#) と [Visual Studio Code](#) の接続手順を説明します。これらの IDE は、InterSystems Native SDK、[ADO.NET Managed Provider for .NET](#)、および [InterSystems XEP](#) を通して InterSystems IRIS を操作する .NET コードの開発に使用できます。

### 5.1 Visual Studio

Visual Studio を InterSystems IRIS に接続するには、以下の手順を使用します（必要に応じて GitHub にログインします）。

まず、サンプル・ソリューションを作成します。

1. **[表示]→[チーム エクスプローラー]** を選択します。
2. **[チーム エクスプローラー - 接続]** ペインで、**[ローカル Git リポジトリ]→[複製]** を選択し、<https://github.com/intersystems/Samples-dotnet-helloworld> をコピーして **[URL]** ボックスに貼り付け、**[複製]** をクリックします。

次に、サンプル・コードを更新して、実行します。

1. **HelloWorld.sln** をダブルクリックしてソリューション内のファイルを表示し、**helloworld.cs** を開きます。
2. 変数宣言を使用して、インスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します（“[InterSystems IRIS 接続情報](#)”を参照）。
3. F5 キーを押して、プログラムを実行します。
4. コマンド・ウィンドウに、以下のメッセージが表示されます。“Hello World! You have successfully connected to InterSystems IRIS.Press any key to continue.”

### 5.2 Visual Studio Code

ObjectScript 拡張機能を利用すると、InterSystems IRIS で Visual Studio Code を使用して ObjectScript コードを開発できます。詳細は、“[ObjectScript IDE](#)”を参照してください。

Visual Studio Code を InterSystems IRIS に接続するには、以下の手順を使用します (必要に応じて GitHub にログインします)。

1. Web ブラウザで、<https://github.com/intersystems/Samples-dotnet-helloworld> をダウンロードまたは複製します。
2. Visual Studio Code で、[ファイル]→[フォルダーを開く] を使用して、GitHub から最近ダウンロードまたは複製したフォルダを開きます。
3. [samples-dotnet-helloworld] を展開し、[helloworld.cs] を選択します。
4. 変数宣言を使用して、インスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します ("InterSystems IRIS 接続情報" を参照)。
5. [表示]→[端末] を使用して端末を開き、以下のコマンドを実行します。

```
csc /reference:InterSystems.Data.IRISClient.dll helloworld.cs
.\helloworld.exe
```

6. [出力] ペインに、以下のメッセージが表示されます。"Hello World! You have successfully connected to InterSystems IRIS.Press any key to continue."

## 6 Python IDE

ここでは、Visual Studio Code、PyCharm、Spyder、IDLE/IdleX、Vim などの Python に焦点を合わせた IDE を InterSystems IRIS に接続する手順を説明します。

1. Web ブラウザで、<https://github.com/intersystems/Samples-python-helloworld> をダウンロードまたは複製します。
2. IDE で、GitHub から最近ダウンロードまたは複製したフォルダを開き、**hello\_world.py** を開きます。
3. 変数宣言を使用して、インスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します ("InterSystems IRIS 接続情報" を参照)。
4. IDE の端末で、以下の手順を実行します。
  - a. 以下のいずれかのコマンドを使用して、InterSystems Native SDK for Python をインストールします。
    - ・ Microsoft Windows :
 

```
pip install wheel/irisnative-1.0.0-cp34-cp35-cp36-cp37-none-win_amd64.whl
```
    - ・ UNIX®/Linux :
 

```
pip install wheel/irisnative-1.0.0-cp34-abi3-linux_x86_64.whl
```
    - ・ Apple macOS :
 

```
pip install wheel/irisnative-1.0.0-cp34-abi3-macosx_10_13_x86_64.macosx_10_14_x86_64.whl
```
  - b. コマンド `python hello_world.py` を使用してコードを実行します。
5. [出力] ペインに、以下のメッセージが表示されます。"Hello World! You have successfully connected to InterSystems IRIS."

**注釈** Python 2 は 2020 年に終了されるため、Python コード・サンプルおよびサポートされる wheel 形式のファイルは Python 3 向けに設計されています。Python 2 向けのバージョンは、InterSystems 教育サービスから入手できます。



## 7 JavaScript IDE

ここでは、Visual Studio Code、Vim、Webstorm などの、JavaScript および Node.js 向けの IDE を InterSystems IRIS に接続する手順を説明します。

1. Web ブラウザで、<https://github.com/intersystems/Samples-nodejs-helloworld> をダウンロードまたは複製します。
2. IDE で、GitHub から最近ダウンロードまたは複製したフォルダを開き、**hello\_world.js** を開きます。
3. 変数宣言を使用して、インスタンスのユーザ名、パスワード、IP アドレス、およびポートを構成します（“[InterSystems IRIS 接続情報](#)”を参照）。
4. IDE の端末で、以下の手順を実行します。
  - a. コマンド `npm install --save intersystems-iris-native` を使用して、InterSystems Native SDK for Node.js をインストールします。
  - b. コマンド `node hello_world.js` を使用してコードを実行します。
5. [出力] ペインに、以下のメッセージが表示されます。“Hello World! You have successfully connected to InterSystems IRIS.”

