



InterSystems IRIS とソース・コントロール・システムの統合

Version 2024.1
2024-06-03

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

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

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

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

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

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

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

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

目次

InterSystems IRIS とソース・コントロール・システムの統合.....	1
1 概要	1
2 InterSystems IRIS ドキュメント	1
2.1 ドキュメントとファイルを管理するツール	1
2.2 内部名と外部名をマップする方法の指定	2
3 ソース・コントロール・クラスの作成と有効化	2
3.1 スタジオの拡張	2
3.2 ソース・コントロール・クラスの作成	3
3.3 ソース・コントロール・クラスの有効化	3
4 ソース・コントロール・システムへのアクセス	4
4.1 例 1	4
4.2 例 2	4
5 関連項目	5

InterSystems IRIS とソース・コントロール・システムの統合

このページでは、サードパーティのソース・コントロール・システムを接続することによって InterSystems IRIS® データ・プラットフォームのコードをソース・コントロールに配置する方法を説明します。

1 概要

InterSystems IRIS 開発プロジェクトをソース・コントロールに配置するには、以下の手順を実行します。

- ・ コード・ユニットを XML ファイルとしてファイル・システムにエクスポートします。ソース・コントロールのコンテキストでは、各コード・ユニットはドキュメントです。
- ・ XML ファイルをソース・コントロールに配置します。
- ・ XML ファイルを、内容と読み取り/書き込み状態の両方について、インターシステムズと同期された状態に保ちます。
- ・ InterSystems IRIS 内からソース・コントロールの機能を実行できることを確認します。
- ・ ドキュメントがチェック・アウトされたかどうか、およびどのユーザがチェック・アウトしたかについて、InterSystems IRIS とソース・コントロール・システムを同期された状態に保ちます。

2 InterSystems IRIS ドキュメント

ドキュメントには、クラス定義、ルーチン、およびデータ変換が含まれます。InterSystems IRIS では、最後にコンパイルされてから変更があったかどうかなどの情報が記録されます。ソース・コントロール・システムは、各ドキュメントを個別のユニットとして扱います。

InterSystems IRIS とソース・コントロール・システムでは、一度に 1 つのネームスペース内で作業します。

2.1 ドキュメントとファイルを管理するツール

InterSystems IRIS には、ドキュメントと外部ファイルを管理するために、以下のツールが用意されています。

- ・ `%Studio.Extension.Base` クラスは、基本的なドキュメント管理のためのメソッドを提供します。`%Studio.SourceControl.Base` は、`%Studio.Extension.Base` のサブクラスで、ソース・コントロール・メニューも提供します。“[ソース・コントロール・クラスの作成と有効化](#)”を参照してください。
- ・ `$system.OBJ.Export` 関数は、InterSystems IRIS ドキュメントを外部のドキュメント・システムにある XML ファイルにエクスポートします。この XML ファイルには、ドキュメントの再構築に十分な情報が含まれています。
- ・ `$system.OBJ.Load` 関数は、外部 XML ファイルをロードし、対応する InterSystems IRIS ドキュメントが存在する場合はそれを上書きします。
- ・ `%RoutineMgr.TS` クラス・メソッドは、InterSystems IRIS ドキュメントのタイムスタンプとコンパイル日時を返します。

2.2 内部名と外部名をマップする方法の指定

各ドキュメントには 2 つの名前があります。

- ・ 内部名。例えば、スタジオの [開く] ダイアログ・ボックスで使用する名前です。
- ・ 外部名。ファイル・システム内でのファイルの完全修飾パス。サポートされる InterSystems IRIS プラットフォーム間でファイル・パスの構成方法が異なるため、有効な既定値を提供することができません。

内部名と外部名との間に双方向のマッピングを作成することは、ソース・コントロール・システムを実装するうえで最も難しい部分であると言えます。このマッピングはユーザ固有のものであり、注意深く考慮する必要があります。類似する項目をグループ化できます。例えば、サンプルでは以下のディレクトリ構造が使用されます。

- ・ クラス・ファイルは、クラスのパッケージ階層に対応するサブディレクトリを含む **cls** サブディレクトリ内にあります。
- ・ .INT ルーチンは **int** サブディレクトリ内にあります。
- ・ .MAC ルーチンは **mac** サブディレクトリ内にあります。

例を以下に示します。

- ・ 内部名 — MyApp.Addresses.HomeAddress
- ・ 外部名 — C:\sources\cls\MyApp\Addresses\HomeAddress.xml

多数のルーチンがある場合は、機能別などで、ルーチンをサブディレクトリにグループ化できます。

3 ソース・コントロール・クラスの作成と有効化

このセクションでは、ソース・コントロール・クラスを作成して有効化するための基本的な要件について説明します。

3.1 スタジオの拡張

%Studio.Extension.Base または %Studio.SourceControl.Base を使用して、ソース・コード・コントロール・メニューをスタジオに追加できます。最大で 2 つのメニューと、そのそれぞれに 19 個のメニュー項目を追加できます。

注釈 スタジオのソース・コントロール・メニューを使用するには、ユーザに %Developer ロールが必要です。

%Studio.Extension.Base クラスでは、以下のメソッドが提供されます。これらのメソッドではすべて InterSystems IRIS ドキュメントの内部名が使用されます。

- ・ 空の Login メソッドと Logout メソッド。必要に応じて実装できます。変数 \$username によって、現在のユーザが記録されます (Login メソッドでは、Username 引数が下位互換性のために提供されているので、代わりに、変数 \$username の使用をお勧めします)。
- ・ 指定した InterSystems IRIS ドキュメントの状態を示す基本的なメソッド：GetStatus および IsInSourceControl。必要に応じて、これらのメソッドを実装します。
- ・ コールバック・メソッド。ユーザが InterSystems IRIS ドキュメントでアクションを実行するときに実行されます。これには、OnBeforeLoad、OnAfterLoad、OnBeforeCompile、OnAfterCompile、ItemIconState のメソッドが含まれます。

注釈 スタジオのクラスのコンパイルでは複数のプロセスを使用できます。したがって、%Studio.Extension.Base のプロパティを使用して、MenuItem から OnBeforeCompile に情報を渡さないでください。その代わりに、一時グローバルを使用してください。

%Studio.SourceControl.Base は %Studio.Extension.Base のサブクラスで、以下の追加要素を提供します。

- ・ Menu という名前の XDATA ブロック。InterSystems IRIS の追加メニュー [ソースコントロール] を定義します。既定では、このメニューには、メニュー項目 [チェックイン]、[チェックアウト]、[チェックアウト取消]、[最新を取得]、および [ソース・コントロールに追加] が含まれています。また、この XDATA ブロックは、スタジオのコンテキスト・メニューの追加メニュー項目も定義します。
これらすべてのメニュー項目は、このクラスで定義されているメソッドも呼び出します。
- ・ CheckIn()、CheckOut()、UndoCheckOut()、GetLatest()、および AddToSourceControl() という名前のメソッドは、既定では何も実行しません。

InterSystems IRIS を拡張するには、これらのクラスのいずれかを拡張する新しいクラスを定義します。“ソース・コントロール・クラスの有効化”で説明するように、管理ポータルには、指定したネームスペースで現在アクティブな拡張クラスを示す方法が用意されています。指定したネームスペースで拡張クラスがアクティブな場合、およびそのクラスが XDATA メニュー・ブロックを定義する場合は、これらのメニュー項目が InterSystems IRIS に追加されます。

3.2 ソース・コントロール・クラスの作成

ソース・コントロール・クラスを作成するには、以下を実行します。

1. %Studio.Extension.Base または %Studio.SourceControl.Base のサブクラスを作成します。
2. %Studio.Extension.Base から開始する場合、サブクラスで Menu という名前の XDATA ブロックを作成します(これを開始するには、%Studio.SourceControl.Base からコピーして貼り付けます)。
3. 必要に応じて、このクラスにメソッド AddToSourceControl()、CheckIn()、CheckOut() などを実装します。これらのメソッドは、通常、少なくとも以下を実行します。
 - ・ 必要に応じて、InterSystems IRIS ドキュメントを XML からインポート、または XML にエクスポートします。
 - ・ XML ファイルで機能するように、ソース・コントロール・ソフトウェアの適切な関数またはメソッドを呼び出します。
 - ・ 指定したファイルの状態について InterSystems IRIS の内部情報を更新します。
 - ・ InterSystems IRIS ドキュメントが編集可能であるかどうかを制御します。

詳細は、ソース・コントロール・システムにより異なります。

4. ソース・コントロール・クラスの GetStatus() メソッドを実装します。これは必須です。既定の実装が適切でない場合は、IsInSourceControl() メソッドの実装も必要な場合があります。

3.3 ソース・コントロール・クラスの有効化

指定したネームスペースに対してソース・コントロール・クラスを有効にするには、以下を実行します。

1. 拡張クラスが存在する場合、管理ポータルを使用して、指定したネームスペースで InterSystems IRIS が使用する拡張クラスを指定します。使用するクラスを指定するには、以下の操作を実行します。
 - a. [システム管理]→[構成]→[追加設定]→[ソースコントロール] を選択します。
 - b. ページの左側で、この設定に適用するネームスペースを選択します。
 - c. 使用する拡張クラスの名前を選択して、[OK] を選択します。
このリストには、選択されたネームスペースにある、%Studio.Extension.Base のコンパイル済みサブクラスがすべて含まれます。
2. スタジオが現在開いている場合は、いったん閉じてから再び開きます。または、別のネームスペースに切り替えてから元に切り替え直します。

管理ポータルの [プロダクション構成] ページでも、ソース・コントロールがサポートされます。”ソース・コントロール設定の構成”を参照してください。

4 ソース・コントロール・システムへのアクセス

ソース・コントロール・システムの API には、ファイルのチェック・アウトなど、ソース・コントロールの機能を実行するためのメソッドまたは関数が用意されています。ソース・コントロール・クラスは、この API の適切な呼び出しを行う必要があります。また、InterSystems IRIS サーバは、API 自体を定義する共有ライブラリまたは他のファイルの場所を特定する必要があります。

また、InterSystems IRIS が InterSystems IRIS サーバ上でソース・コントロール・コマンドを実行することにも注意する必要があります。つまり、XML ファイルは InterSystems IRIS サーバ上にあり、ファイル・マッピングが、そのサーバ上で使用するオペレーティング・システムで機能する必要があります。

4.1 例 1

以下のフラグメントでは、VSS の API 用のラップ・メソッドを生成しました。次に、ソース・コントロール・メソッド内に以下のようなコードを含めることができます。

```
do ..VSSFile.CheckIn(..VSSFile.LocalSpec,Description)
```

詳細は、ソース・コントロール・ソフトウェア、その API、および要件によって異なります。

4.2 例 2

以下のフラグメントでは、Windows コマンド行インタフェースを使用してファイルをチェック・アウトしています。この例では、ソース・コントロール・システムは Perforce です。

```
/// Check this routine/class out of source control.
Method CheckOut(IntName As %String, Description As %String) As %Status
{
    Set file=..ExternalName(IntName)
    If file="" Quit $$$OK
    ///...
    Set cmd="p4 edit ""_file_""

    #; execute the actual command
    Set sc=..RunCmd(cmd)
    If $$$ISERR(sc) Quit sc

    #; If the file still does not exist or
    #; if it is not writable then checkout failed
    If '##class(%File).Exists(file)||(#class(%File).ReadOnly(file)) {
        Quit $$$ERROR($$$GeneralError,
            "Failure: '"_IntName_"' not writeable in file sys")
    }

    #; make sure we have latest version
    Set sc=..OnBeforeLoad(IntName)
    If $$$ISERR(sc) Quit sc

    ///...
    Quit sc
}
```

この例では、RunCmd() は、指定したコマンドを実行して一部の汎用エラーのチェックを行う別のメソッドです(RunCmd() は、\$ZF(-1) インタフェースから OS コマンドを発行します)。

また、この CheckOut() メソッドは、InterSystems IRIS ドキュメントと外部 XML ドキュメントを同期させる OnBeforeLoad() メソッドを呼び出します。

5 関連項目

- ・ [Visual Studio Code の概要](#)
- ・ [スタジオの使用法](#)

