



プロダクション作成のベスト・プラクティス

Version 2024.1
2024-06-06

プロダクション作成のベスト・プラクティス

InterSystems IRIS Data Platform Version 2024.1 2024-06-06

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 プロダクション開発のベスト・プラクティス	1
1.1 プロジェクトの目標	1
1.2 プロジェクトの配布	1
1.3 ドキュメント	2
2 ルーティング・プロダクションの設計モデル	5
2.1 構成項目	5
2.2 アプリケーションの仕様	7
2.3 プロダクション・スプレッドシート	8
2.4 インタフェース設計	8
2.5 命名規則	10
2.5.1 ビジネス・サービス	12
2.5.2 ルーティング・プロセス	12
2.5.3 ルーティング・ルール・セット	13
2.5.4 ビジネス・オペレーション	13
2.5.5 データ変換	14
2.5.6 カスタム・スキーマ・カテゴリ	14
3 プロダクション要素へのインタフェース変換	15
3.1 プロダクションのバックアップ	16
3.2 インタフェースの説明	16
3.3 スキーマ・カテゴリの選択	16
3.4 ルーティング・ルール・セットの定義	17
3.5 データ変換の作成	17
3.6 ビジネス・オペレーションの追加	17
3.7 ルーティング・プロセスの作成または編集	17
3.8 ビジネス・サービスの追加	18
3.9 インタフェースのテスト	18
3.10 インタフェースの導入	19
4 プロダクションのデータベースの管理	21
4.1 メッセージ・データを含むグローバル	21
4.2 既定のデータベース用のグローバル	23
4.3 新しいシステムでのグローバルのマッピング	23
4.4 既存システムでのグローバルのマッピング	24

テーブル一覧

テーブル 4-1: InterSystems IRIS	21
テーブル 4-2: InterSystems IRIS for Health	22
テーブル 4-3: HealthConnect	22
テーブル 4-4: HealthShare	22

1

プロダクション開発のベスト・プラクティス

ここでは、プロダクション・プロジェクトに着手するための一般的な概要情報を取り上げます。また、開発作業の概要を説明し、InterSystems IRIS® データ・プラットフォームの情報の入手先、および企業統合について説明します。ここで提示する情報は、すべてのタイプのプロダクション・プロジェクトに関連します。

1.1 プロジェクトの目標

プロダクション開発プロジェクトの目的は、2 つ以上のシステムを接続することです。プロダクション とは、企業の特定の統合問題を解決するための、専門化されたソフトウェアとドキュメントのパッケージです。概要は、“[相互運用プロダクションの概要](#)” を参照してください。

この節では、アプリケーション開発者がソリューションを提供するために作成および構成する必要のあるソフトウェア要素の観点から InterSystems IRIS を説明します。次の節の“[プロジェクトの配布](#)” では、InterSystems IRIS 開発プロジェクトのシーケンスと成果について説明します。

1.2 プロジェクトの配布

InterSystems IRIS 製品のアーキテクチャは、企業でのさまざまな配布形態をサポートしています。

- ・ プロダクションは、企業の統合ソリューション全体として構成することも、企業に導入済みの既存のソリューションやソリューションの一部と合わせて使用することもできます。
- ・ プロダクションを使用すると、必要に応じてレガシ・システムを置換またはアップグレードしたり、レガシ・システムをまったく除去または変更することなく、新機能を追加したりすることができます。
- ・ InterSystems IRIS は、インクリメンタルな開発プロジェクトをサポートしているため、開発チームは、旧システムから新システムへの開発分を、企業の必要に応じた速さで実施できます。

多くの InterSystems IRIS 開発プロジェクトは、次のような順序のフェーズをたどります。

順序	フェーズ	目標	重点
1	仕様	プロダクションの要件の指定	プロダクションはどのような機能を備えているべきか。
2	設計	プロダクション・ソフトウェアの設計	各要素がどのように相互動作すべきか。
3	コーディング	プロダクション・ソフトウェアの構築	追加の要素が必要かどうか。
4	テスト	プロダクション・ソフトウェアのテスト	プロダクションは指定された要件を満たすかどうか。
5	導入	目的の場所へのソフトウェアのインストール	必要に応じて、テスト、設計、作成、および再作業の準備が整っているかどうか。
6	リリース	ソフトウェアおよびプロジェクト成果物の提供	システム管理チームに役立つものは何か。

1.3 ドキュメント

InterSystems IRIS ドキュメントを理解することは基本的なベスト・プラクティスです。以下のドキュメントおよび章の理解から始めることをお勧めします。それぞれに相互参照が記載されています。

- ・ “[相互運用プロダクションの概要](#)” では、用語と基本概念を紹介します。
- ・ “[プロダクションの開発](#)” では、開発タスクについて説明します。
- ・ “[プロダクションの構成](#)” では、プロダクションの作成に関連した構成タスクの実行方法について説明します。
- ・ このドキュメントの後半の章は、インタフェース・ルーティング・ソリューションの導入に適用されます。“[ルーティング・プロダクションの設計モデル](#)” と “[プロダクション要素へのインタフェース変換](#)” を参照してください。

また、プロダクションをサポートする InterSystems IRIS のプラットフォーム機能を理解することも重要です。“サーバ側プログラミングの入門ガイド” が入門書として役に立ちます。

有益な参考資料には以下のものがあります。

- ・ クラスの定義と使用
- ・ ObjectScript の使用法
- ・ InterSystems SQL ゲートウェイの使用法
- ・ Web サービスおよび Web クライアントの作成

他に有益な開発ガイドには以下のものがあります。

- ・ InterSystems SQL の使用法
- ・ InterSystems IRIS 多次元ストレージの使用法 (グローバル)
- ・ 高可用性ガイド

言語の参考資料には以下のものがあります。

- ・ ObjectScript リファレンス

システム・ユーティリティおよびセキュリティの詳細は、以下のドキュメントを参照してください。

- ・ システム管理ガイド

- ・ インターシステムズのセキュリティについて

2

ルーティング・プロダクションの設計モデル

ここでは、インターシステムズのユーザがインタフェース・ルーティング・ソリューションを構築するために使用した設計モデルについて説明します。このモデル自体を、ルーティング・プロダクションの開発のためのベスト・プラクティスのコレクションと見なすことができます。

ここでは、1つのアプローチのみを説明します。その他の実装計画でも同様に成功を収めることができます。各組織には、コードの記述およびプログラミング・プロジェクトの実施に関する独自の標準があります。ここでは、InterSystems IRIS® に特有で、入念に体系化されたアプローチによって利点を得られるルーティング・プロダクションの側面について簡単に説明します。

一般に、優れた設計にとって重要な点は、明確さと簡潔さです。簡潔さとは、部分の数の少なさではありません。簡潔さとは、モデル内の各部が明確な機能を持ち、直感的に探しやすいことを意味します。

2.1 構成項目

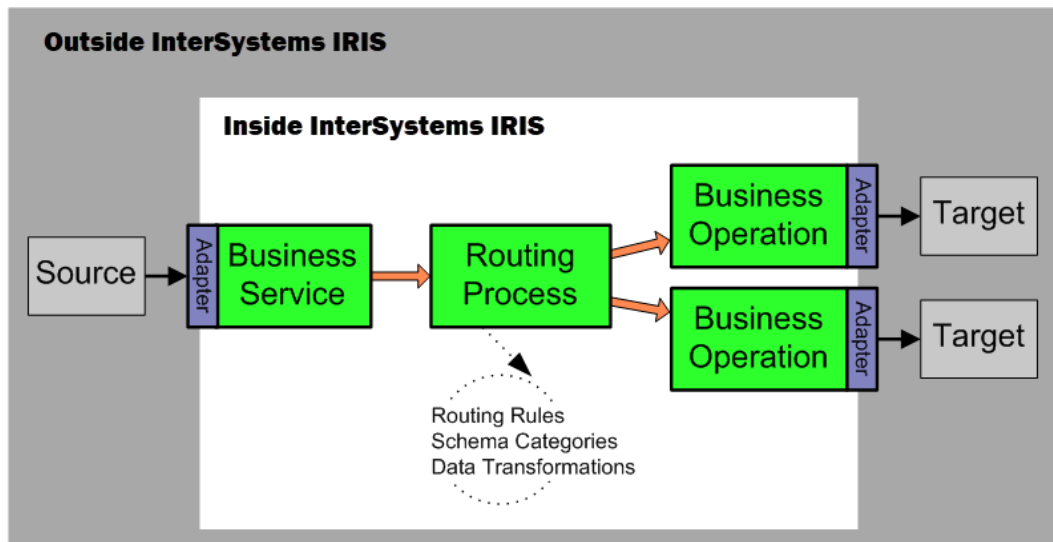
プロダクションは構成項目より構成されます。構成項目には以下の3タイプがあります。

- ・ ビジネス・サービスは、受信メッセージを受け付けます。
- ・ ビジネス・オペレーションは、送信メッセージを送信します。
- ・ ビジネス・プロセスは、その中間のすべての処理を指示します。ビジネス・プロセスにはいくつかの特殊なタイプがあります。

ルーティング・プロセスは、ルーティング・プロダクションの以下の主要なコンポーネントを呼び出すことにより、メッセージのルーティングおよび変換を行います。

- － ルーティング・ルールは、メッセージのコンテンツに基づいてメッセージを宛先に送ります。
- － スキーマ・カテゴリは、メッセージのコンテンツの検証およびアクセスの手段を提供します。
- － データ変換は、変更を適用し、宛先に向けてメッセージを準備します。

シーケンス・マネージャは、関連するメッセージが適切なシーケンスとタイミングでターゲットに到達するようにします。



以下の図は、ルーティング・プロセスのダイアグラムを拡大したもので、このプロセスがルーティング・ルールを使用してインタフェースでの情報のフローを指示する方法を示しています。以下の図には、上のダイアグラムの点線の円で示された部分のみのルーティング・ルールおよびデータ変換の詳細が示されています。

ルーティング・プロセスには、ルーティング・ルール・セットが関連付けられています。ルール・セットの定義方法、およびソース・アプリケーションから着信するメッセージのタイプにより、ルール・セットは実行するルールを特定します。複数のルールを順に実行することもできますが、説明を簡易化するため、以下の図にはルール・セットが 1 つのルールのみを選択する場合を示します。図のように、ルールはここからメッセージを削除することも、InterSystems IRIS 内の宛先に送信することもできます。この宛先がビジネス・オペレーションの場合、メッセージはプロダクションからターゲット・アプリケーションに送信されます。

これらのタスクをサポートする背景情報については、このドキュメントの後続の章および以下のドキュメントを参照してください。

- ・ “プロダクション内での仮想ドキュメントの使用法” の “[カスタム・スキーマ・カテゴリの作成](#)”
- ・ “プロダクション内での仮想ドキュメントの使用法” の “[メッセージ検証の制御](#)”
- ・ “プロダクション要素へのインタフェース変換” の章の “[スキーマ・カテゴリの選択](#)”

2.3 プロダクション・スプレッドシート

情報システムを編成するスプレッドシートをアプリケーションごとに管理するのは有用なことです。一般的なガイドラインとして、送受信データを提供する各アプリケーションの行を作成します。それぞれの行では、以下の列が有用です。

- ・ **[フィールド]** – インタフェース・エンジンまたはサーバでは、いくつかのアプリケーションからのメッセージをルーティング・プロダクションに送ることがよくあります。このような場合、エンジンまたはサーバの名前をここに記します。
- ・ **[アプリケーション]** – 特定のアプリケーション、システム内でのロール、およびこのアプリケーションの問題についての連絡先を簡単に指定します。InterSystems IRIS を使用していないがシステムの動作については大まかに理解している組織のメンバが理解できる説明であることが理想的です。
- ・ **[名前]** – このアプリケーションに対する 3 ～ 6 文字の一意の名前。
- ・ **[タイプ]** – アプリケーションが外部の通信に使用するプロトコル。
- ・ **[接続]** – IP アドレスやポート番号など、接続の詳細。
- ・ **[送信]** – このアプリケーションが情報システムに提供するメッセージ構造。

情報システムに入った後の受信メッセージの構造を考慮してください。宛先のシステムまたはその他の要因によって異なるルーティングまたは変換を行う必要があるでしょうか。その必要がある場合、必要な数だけこれをリストし、相違点に関する注釈を付けます。このようにして、スプレッドシートは、それぞれの場合で必要なルーティング・ルール、データ変換、およびカスタム・スキーマを作成する際のチェックリストとして機能します。

- ・ **[受信]** – このアプリケーションが情報システムから利用するメッセージ構造。
- ・ **[ACK]** – 確認応答の詳細。ACK および NACK メッセージを必要としますか。これらのメッセージを送信および受信するための個別のインタフェースがありますか。プロダクションで ACK および NACK を生成する必要がありますか。もしくは受信アプリケーションで生成しますか。

プロジェクトの開始時、最初のスプレッドシートに情報システム内のすべてのアプリケーションについて記述する必要はありません。新規インタフェースを導入するごとにスプレッドシートに追加することができます。

2.4 インタフェース設計

このトピックでは、インタフェース設計をモジュール式にして、開発の各段階においてプロダクションを容易に理解、拡張、および管理するための効果的な方法を説明します。

- ・ InterSystems IRIS にメッセージを送信するアプリケーションごとに 1 つのビジネス・サービスを提供する。これは、[プロダクション・スプレッドシート](#)の **[送信]** 列に 1 つ以上のエントリがあるアプリケーションです。

1 つのビジネス・サービスは、同じアプリケーションのすべてのメッセージ構造を受信できます。通常、これが適切な設計です。この目的でビジネス・サービスを設定する場合、一般的にはビジネス・サービスをそのアプリケーション・システムに常時接続します。

同じアプリケーションに対してさらにビジネス・サービスの接続が必要となる構成がある場合もあります。例えば、ソース・アプリケーションが既に異なる 2 つの TCP/IP ポートにメッセージを送信するよう設定されている場合、1 つのポートからのすべてのメッセージを受信するよう 1 つのビジネス・サービスを設定し、別のポート用にもう 1 つのビジネス・サービスを設定します。1 通信ソースあたり 1 ビジネス・サービスであるため、これは一般的にアプリケーションごとに 1 ビジネス・サービスというモデルと一貫しています。

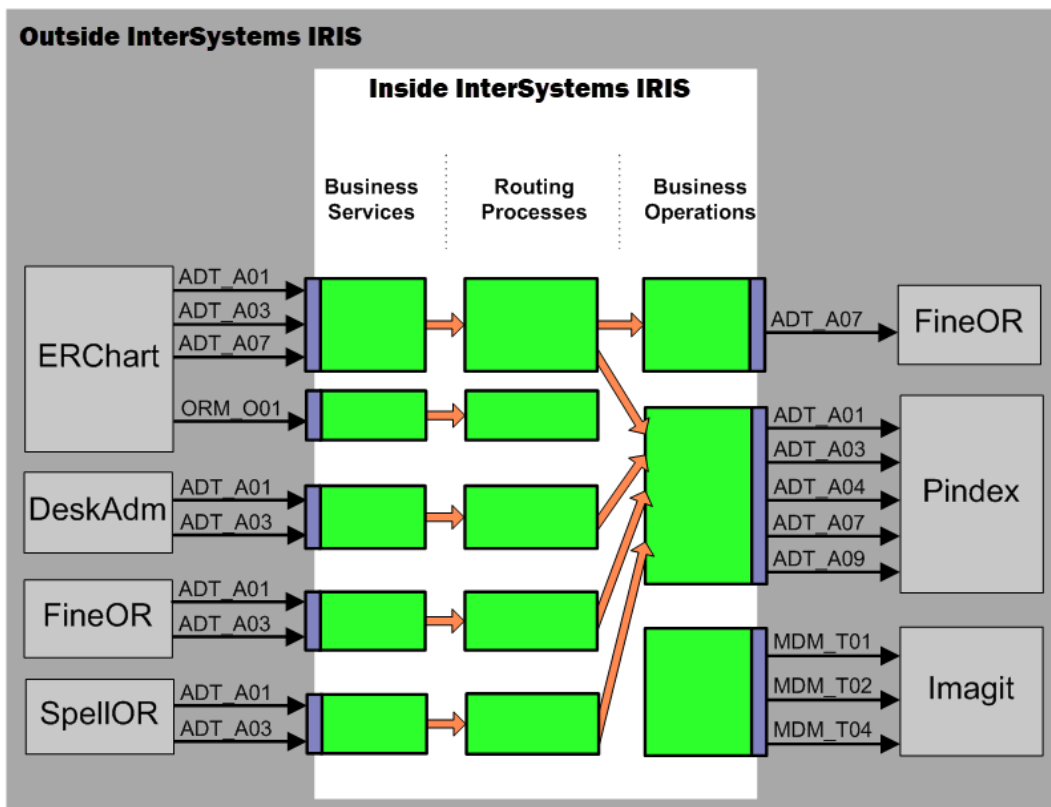
一方、同じ医療アプリケーションを使用する何百人ものユーザがデータを企業に送信している場合、これらすべてのメッセージを 1 つのビジネス・サービスにより（アプリケーションの単一のインスタンスに常時接続されているかどうかによらず）InterSystems IRIS にルーティングすると便利なこともあります。

- ・ ビジネス・サービスごとに 1 つのルーティング・プロセスを提供する。ルーティング・プロセスでは、メッセージ自体のコンテンツまたは InterSystems IRIS に保存された情報に基づいて、メッセージをルーティングすることができます。ルーティングが別のメッセージのコンテンツに依存する場合、またはルーティングの決定に外部アプリケーションの呼び出しを必要とする場合は、ルーティング・プロセスは別のクラスを呼び出す BPL ビジネス・プロセスである必要があります。ただし、ほとんどの場合は、組み込みのルーティング・ルール・エンジンに基づくルーティング・プロセスで十分です。
- ・ プロダクションからメッセージを受信するアプリケーションごとに 1 つのビジネス・オペレーションを提供する。これは、[プロダクション・スプレッドシート](#)の **[受信]** 列に 1 つ以上のエントリがあるアプリケーションです。
1 つのビジネス・オペレーションは、同一アプリケーションに対してすべて異なるメッセージ構造を送信できます。通常、これが適切な設計です。この目的でビジネス・オペレーションを設定する場合、一般的にはビジネス・オペレーションをそのアプリケーション・システムに常時接続します。ただし、ビジネス・サービスの場合、このモデルにバリエーションがある場合があります。

基本概念は、一貫したモジュール式设计です。つまり、一度に 1 つのインタフェースを開発し、インタフェースごとに 1 つのルーティング・プロセスを使用します。これは、単一の大規模なルーティング・プロセスがすべてのインタフェースのルーティング・エンジンとして機能するモデルとは対照的です。ルーティング・プロセスが 1 つより多い方が優れている理由はいくつかあります。

- ・ 1 つのインタフェースで必要なケースのみに対応すればよいため、それぞれのルーティング・ルール・セットの管理が単純かつ容易である。自己完結した、簡単な一連の問題を解決する作業を共有および再利用する方が容易です。
- ・ インタフェースの個別の開発、置換、および管理が容易になる。企業が特定のアプリケーションを削除またはアップグレードする必要がある場合、かわりが必要になるのは、このアプリケーションを処理するルーティング・プロセスのみです。インタフェースがダウンした場合、そのインタフェースを無効化、診断、修正、および再テストするだけで、これを再起動することができます。

これは、大規模なルーティング・プロセスで 1 つのインタフェースを追加、削除、または修正する必要があるたびにすべてのルールを再テストし、検証する必要性に比べ、非常に簡潔です。このような考慮事項は、一部のインタフェースを動作させる一方で、その他はまだ開発中である場合に特に重要となります。既に稼働中のルーティング・プロセスに追加するたびに、何百もの既存のルールを再テストし、検証する必要がある可能性があります。下の図は、複数のインタフェースを持つルーティング・プロダクションを示しています。



プロダクションの各構成項目には、果たすべき特定のロールがあります。各項目にそのロールを遂行させます。

- ・ ビジネス・サービスおよびビジネス・オペレーションは簡単にする。一般に、ビジネス・サービスまたはビジネス・オペレーションに対して独自のコードを記述する必要はありません。InterSystems IRIS が提供する組み込みクラスを選択してください。この選択により、自動的に正しいアダプタが指定されます。管理ポータルの設定を使用してこれらのクラスを構成します。
- ・ 複雑なアクティビティはすべてルーティング・プロセスの制御下に配置する。インタフェースが単純な場合、そのルーティング・プロセスを複雑にする必要はありません。しかし、複雑なアクティビティが存在する場合、それはビジネス・サービスやビジネス・オペレーションではなく、ルーティング・プロセスに所属させます。タスクを実行するため、ルーティング・プロセスは別のルーティング・プロセスに連鎖するか、ビジネス・ルール、ルーティング・ルール、データ変換、ビジネス・プロセス、またはカスタム・コードを呼び出すことができます。

2.5 命名規則

このトピックでは、命名規則の重要性について説明し、その例を示します。

一般に、プロダクションは、一度に 1 インタフェースずつ、付加的に開発します。項目は“その都度”命名し、命名規則をプロジェクトの進行に伴って増やしていくという考えもありますが、命名規則にはこのような付加的アプローチを取らないことをお勧めします。

HIS システム全体には数百ものインタフェースが関与する可能性があることに注意してください。これらのインタフェースがすべて含まれた場合のルーティング・プロダクションの管理タスクを予想してください。各コンポーネントの名前は、その目的を明確に示す必要があります。このようにすることで、開発者とシステム管理者は、機能に基づいてコンポーネントの名前を容易に予測することができます。

命名規則は一式すべてプロジェクトの開始時に確立し、その後はこれらに従うことをお勧めします。これによって、名前を修正するだけのためにプロダクション構成に戻る必要性が軽減されます。

機能別に明確にコンポーネントを特定できる規則であれば問題ありません。InterSystems ユーザが使用し、成功を収めている命名規則一式を以下に示します。

- ・ 構成項目名に関する規則は、“[構成名](#)”を参照してください。
- ・ 名前を作成するために組み合わせる個々のセグメントをリストします。具体的な内容は次のとおりです。
 - － キーワード：
 - ・ From (受信用)
 - ・ To (送信用)
 - ・ Router (ルーティング用)
 - ・ Rules (ルール・セット用)
 - ・ ベース・スキーマ・カテゴリの番号－2.1、2.2、2.3、2.3.1、2.4、2.5、2.5.1、2.6、2.7、または2.7.1－(スキーマ用)
 - － アプリケーションごとの短い名前
 - － メッセージ構造の一般的なクラスを特定するキーワード (ADT、ORM など)
 - － 完全なメッセージ構造名 (ADT_A01、ADT_A04、ORM_O01 など)
- ・ 名前の各セグメントは簡潔にする (3 ～ 6 文字)
- ・ 名前は大文字と小文字を区別する

プロダクションの要素ごとに、定義したセグメントから名前を組み立てます。

- ・ ビジネス・サービス：FromSourceAppName
- ・ ビジネス・オペレーション：ToTargetAppName
- ・ ルーティング・プロセス：SourceAppNameRouter
- ・ ルーティング・ルール・セット：SourceAppNameRules
- ・ データ変換：

SourceAppNameSourceMessageTypeToTargetAppNameTargetMessageType
- ・ カスタム・スキーマ・カテゴリ：SourceAppNameBaseSchemaNumber

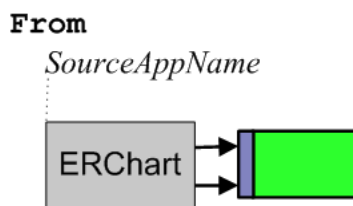
さらに複雑な設計の場合は以下ようになります。

- ・ ビジネス・サービス：FromSourceAppNameMessageTypes
- ・ ビジネス・オペレーション：ToTargetAppNameMessageTypes
- ・ ルーティング・プロセス：SourceAppNameMessageTypesRouter
- ・ ルーティング・ルール・セット：SourceAppNameMessageTypesRules

以下のトピックでは、これらの命名規則について説明し、例を示します。

2.5.1 ビジネス・サービス

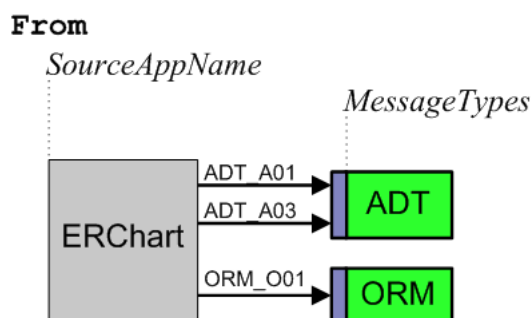
規則



例

FromERChart、FromFineOR、FromDeskAdm

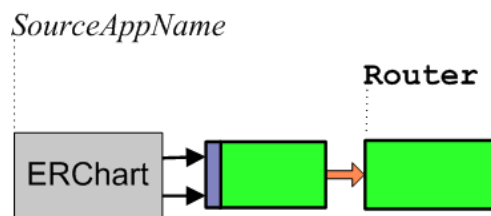
バリエーション



1つのアプリケーションが複数のビジネス・サービスを介してメッセージを InterSystems IRIS に送信するようインタフェースを編成する場合は、各ビジネス・サービスを通過する MessageTypes を分類するキーワードも含めます。一般的な MessageTypes キーワードでは、メッセージ構造の最初の 3 文字を使用します。この場合、ビジネス・サービス名の規則は、FromSourceAppNameMessageTypes となります。

2.5.2 ルーティング・プロセス

規則

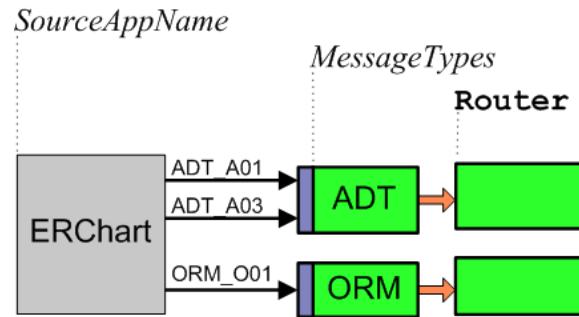


例

ERChartRouter、FineORRouter、DeskAdmRouter

1つのアプリケーションが複数のビジネス・サービスを介してメッセージを InterSystems IRIS に送信するようインタフェースを編成する場合は、各ビジネス・サービスを通過してルーティング・プロセスに渡される MessageTypes を分類するキーワードも含めます。この場合、ルーティング・プロセス名の形式は以下のとおりです。

バリエーション



例

ERChartADTRouter、ERChartORMRouter、ERChartOtherRouter

2.5.3 ルーティング・ルール・セット

規則

各ルーティング・プロセスには関連付けられたルール・セットがあり、これらのルールによってメッセージの宛先が特定されます。ルールはメッセージ・タイプ、メッセージのコンテンツ、時刻、その他の要素に基づき、メッセージをデータ変換またはビジネス・オペレーションに送ります。各ルール・セットは、そのルーティング・プロセスに適した名前で命名します。つまり、各 SourceAppNameRouter に対し、ルール・セットは SourceAppNameRules とします。各 SourceAppNameMessageTypesRouter に対し、ルール・セットは SourceAppNameMessageTypesRules とします。

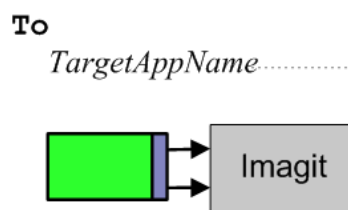
例

DeskAdmRules

DeskAdmORMRules

2.5.4 ビジネス・オペレーション

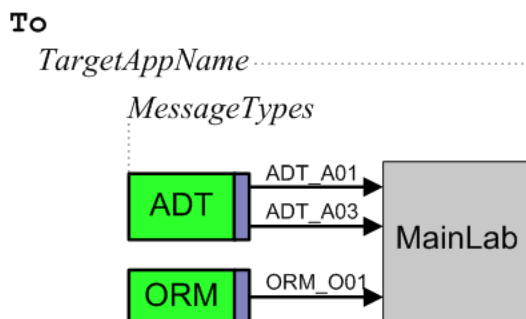
規則



例

ToImagit、ToFineOR、ToPindex

バリエーション



1 つのアプリケーションが複数のビジネス・オペレーションを介して InterSystems IRIS からメッセージを受信するようインタフェースを編成する場合は、各ビジネス・オペレーションを通過する MessageTypes を分類するキーワードも含めます。

2.5.5 データ変換

SourceMessageType と TargetMessageType は同じ名前になる場合がありますが、異なるスキーマ・カテゴリの同じメッセージ構造を表しています。

2.5.6 カスタム・スキーマ・カテゴリ

規則

ApplicationName の BaseSchemaNumber

ApplicationName は、送信アプリケーションまたは受信アプリケーションを表すことができます。

3

プロダクション要素へのインタフェース変換

ここでは、各アプリケーションに対し、受信側と送信側の両方のインタフェースを定義することが必要な状況を想定します。

InterSystems IRIS®をこのような構成にする場合、多くは、既存のインタフェース・エンジンを InterSystems IRIS と置き換えることが目的です。このためには、以下のようにする必要があります。

1. 前の章で説明したように、プロダクションをルーティング・エンジンとして使用します。
2. 以前のインタフェース・エンジンではなく InterSystems IRIS ルーティング・エンジンを使用するよう、既存の各インタフェースを変換します。
3. すべてのインタフェースが変換されたら、以前のインタフェース・エンジンを構成から削除します。

この変換アプローチでは、既存のインタフェース・エンジンをそのままにし、InterSystems IRIS を使用するようにインタフェースを 1 つずつ変換します。新しいインタフェースが InterSystems IRIS を使用して適切に機能することをテストし、確認する間、変換されていないインタフェースは、そのまま以前のエンジンを使用します。変換されたインタフェースが正しく機能したらすぐに、変換されていない次のインタフェースに取り掛かります。このような逐次変更を基にしたポリシーにより、変換の際、可能な限り中断されず、エラーのないサービスが保証されます。

関連するインタフェースのグループを同時に変換するアプローチを取ることもできます。それでもやはり、この作業は一度に 1 インタフェースの変換になります。

ここでは、1 つの既存のインタフェースを InterSystems IRIS を使用するように変換する方法を説明します。一般的な手順は以下のとおりです。各セクションで、以下の手順を 1 つずつ詳細に説明します。

1. [プロダクションのバックアップ](#)
2. [インタフェースの説明](#)
3. [スキーマ・カテゴリの選択](#)
4. [データ変換の作成](#)
5. [ルーティング・ルール・セットの定義](#)
6. [ビジネス・オペレーションの追加](#)
7. [ルーティング・プロセスの作成または編集](#)
8. [ビジネス・サービスの追加](#)
9. [インタフェースのテスト](#)
10. [インタフェースの導入](#)

この順序は、通常プロダクションについて説明する順序とは逆です（“ビジネス・サービスが受信メッセージを受け取り...”）。しかし、インタフェース変換の経験より、要素をトップダウンで開発し、ビジネス・サービスより始めてカスタム・スキーマへと作業を進めると、多くの場合、前の手順に戻り、上位項目を構成したさまざまな形式で下位項目の名前を入力または修正する必要があることは明らかです。

この手順では、ボトムアップのアプローチを取り、上位項目を構成するときに下位項目の名前がわかるように、下位項目を最初に作成することで後戻りを最小限に抑えます。ボトムアップの順序は、手順 1 および 2、手順 3、4、5、6、7、8、手順 9 および 10 となります。

単純なエラーを削減する確立された命名規則がある場合は特に、トップダウンの順序でこの手順が成功する場合があります。トップダウンの順序は、手順 1 および 2、手順 8、7、6、5、4、3、手順 9 および 10 となります。

3.1 プロダクションのバックアップ

フル・バックアップを実行するには、“プロダクションの開発”の“[プロダクションの配置](#)”を参照してください。

注釈 XML バックアップ・ファイルについて：UNIX® システムを使用する場合、バックアップ XML ファイルをバイナリ・モードで FTP 転送しないでください。通常の FTP では、このファイルを DOS から UNIX® に正しく変換しますが、バイナリ FTP では正しく変換しない場合があります。

3.2 インタフェースの説明

企業で使用されている各アプリケーションは、独自のアプリケーション仕様ドキュメントを持っています。仕様ドキュメントでは、アプリケーションが送信（または受信）できるメッセージのタイプ、およびこれらのイベントに対してアプリケーションが送信（または受信）するメッセージ・セグメントおよび各セグメント部分について説明しています。情報は非常に詳細です。各アプリケーションのこのようなドキュメントは、そのアプリケーション管理者に問い合わせて確認してください。

ソースおよびターゲット・アプリケーションにはそれぞれアプリケーションの仕様ドキュメントがあります。ただし、変換するインタフェースなど、これらのアプリケーション間の単一のインタフェースは、仕様ドキュメントにリストされている機能の小さなサブセットを使用します。インタフェースを説明する最も有効なソースは、事実上、置換対象のインタフェース定義です。

既存のインタフェース定義を開きます。この定義を表示し、テキスト・ファイルを開いて、ここにインタフェースの動作について簡単な説明を入力します。既存の定義（LOOP、CASE など）のコーディング規則の詳細を複製しないでください。以下のように簡単に表します。

- ・ ソースからの受信が予定されるメッセージ・セグメント。
- ・ ターゲットに送信する前の、インタフェースによるこれらの変更方法。

3.3 スキーマ・カテゴリの選択

この手順では、インタフェースの受信側と送信側で使用するスキーマを決定する必要があります。まずおおよその選択を行い、管理ポータルでそのスキーマに対するメッセージをテストすることによりこれを調整します。

各アプリケーションには、メッセージの送信時に使用したり、メッセージの受信時に予定されるスキーマを指定する、アプリケーションの仕様ドキュメントがあります。

3.4 ルーティング・ルール・セットの定義

ルーティング・ルール・セットを作成します。詳細は、“ビジネス・ルールの開発”の“[ルール・セットの作成および編集](#)”を参照してください。

3.5 データ変換の作成

DTL データ変換を作成します。詳細は、“[DTL 変換の開発](#)”を参照してください。

3.6 ビジネス・オペレーションの追加

インタフェースのビジネス・オペレーションは、ターゲット・アプリケーションへの送信メッセージ転送を制御します。ビジネス・オペレーションの追加方法は、“[ビジネス・ホストの追加](#)”を参照してください。

テスト用に、同じ構成名を持つ以下の 2 つのビジネス・オペレーションのあるプロダクションを構成すると便利です。

- ・ 1 つは、プロダクションが正常に動作しているときにインタフェース上でのメッセージの FTP または TCP 転送を制御する FTP または TCP のビジネス・オペレーションです。
- ・ もう 1 つは、インタフェースのテストまたはトラブルシューティングの際にメッセージをファイルに送信する File ビジネス・オペレーションです。

規則（項目は同じ構成名を持つ）で、これらの構成項目のうち、一度に有効にできるのは 1 つのみです。“テスト”環境（File オペレーション）にするのか、“ライブ”環境（TCP または FTP オペレーション）にするのかに応じて、どちらか一方を有効にします。

“ライブ”ビジネス・オペレーションおよび“テスト”ビジネス・オペレーションを作成する手順は以下のとおりです。

1. ターゲット・アプリケーションを調査し、メッセージに含まれることが想定される区切り文字を確認します。
2. “ライブ”（FTP または TCP）ビジネス・オペレーションを作成します。
3. 同様の手順を使用して、“テスト”（File）ビジネス・オペレーションを作成します。
“テスト”（File）オペレーションに“ライブ”オペレーションと同じ構成の**名前**を付与します。
4. **[有効にする]** フィールドを使用して、“ライブ”（FTP または TCP）または“テスト”（File）バージョンのビジネス・オペレーションを有効化および無効化します。一度にアクティブにできる同じ名前のビジネス・サービスは 1 つのみです。

詳細は、“[複数バージョンのビジネス・ホストの操作](#)”を参照してください。

3.7 ルーティング・プロセスの作成または編集

新しいインタフェースでルーティング・エンジンを操作できるようにするには、以下を行うルーティング・プロセスをプロダクションに追加する必要があります。

1. ソースからのデータの解釈方法を指示する（ルーティング・ルールを指定する）

2. 解釈されたデータの送信先を指示する (ビジネス・オペレーションを指定する)

新しいルーティング・プロセスを作成できます。

3.8 ビジネス・サービスの追加

インタフェースのビジネス・サービスは、ソース・アプリケーションからの受信メッセージを受け取ります。

テスト用に、同じ構成名を持つ以下の 2 つのビジネス・サービスのあるプロダクションを構成すると便利です。

- ・ 1 つは、プロダクションが正常に動作しているときに FTP または TCP を介してソース・アプリケーションからのメッセージを受け取る FTP または TCP のビジネス・サービスです。
- ・ もう 1 つは、インタフェースのテストまたはトラブルシューティングの際にファイルからのメッセージを受け取る File ビジネス・サービスです。

規則 (項目は同じ構成名を持つ) で、これらの構成項目のうち、一度に有効にできるのは 1 つのみです。“テスト” 環境 (File サービス) にするのか、“ライブ” 環境 (TCP または FTP サービス) にするのかに応じて、どちらか一方を有効にします。

“ライブ” ビジネス・サービスおよび“テスト” ビジネス・サービスを作成する手順は以下のとおりです。

1. “ライブ” (FTP または TCP) ビジネス・サービスを作成します。
2. 同様の手順を使用して、“テスト” (File) ビジネス・サービスを作成します。
“テスト” (File) サービスに“ライブ” サービスと同じ構成の**名前**を付与します。
3. **[有効にする]** フィールドを使用して、“ライブ” (FTP または TCP) または“テスト” (File) バージョンのビジネス・サービスを有効化および無効化します。一度にアクティブにできる同じ名前のビジネス・サービスは 1 つのみです。

詳細は、“[複数バージョンのビジネス・ホストの操作](#)”を参照してください。

3.9 インタフェースのテスト

一般には、“稼動”しているプロダクションの完全コピーである、別の“テスト”プロダクションを管理する必要があります。新しいインタフェースは“テスト”プロダクション内で開発します。これが完了すると、新しいインタフェースのコピーを“ライブ”プロダクションに移行することができます。

新しいインタフェースをテストするには、以下の手順を実行します。

1. ソース・アプリケーションからのいくつかのサンプル・メッセージをファイルに捕捉します。
2. “テスト”プロダクションで、File ビジネス・サービスと File ビジネス・オペレーションを有効にし、ファイルとしてメッセージを送信します。
3. 結果として作成された出力ファイルのメッセージ・データを調べ、ターゲット・アプリケーションの要件を満たしているかどうかを確認します。
4. 必要であれば、インタフェース要素を調整し、再テストします。
5. “テスト”プロダクション内で、ビジネス・サービスとビジネス・オペレーションの“テスト” (File) バージョンで無効を選択し、“ライブ” (FTP または TCP) バージョンを再度有効にします。

3.10 インタフェースの導入

テスト・プロダクションでインタフェースのテストを完了したら、新しいインタフェース要素を稼動中のプロダクションに追加します。これを行うには、以下を実行します。

1. “プロダクションの開発”の“[プロダクションのエクスポート](#)”で説明されているプロセスを使用して、ライブ・プロダクション全体をバックアップします。

注釈 XML バックアップ・ファイルについて：UNIX® システムを使用する場合、バックアップ XML ファイルをバイナリ・モードで FTP 転送しないでください。通常の FTP では、このファイルを DOS から UNIX® に正しく変換しますが、バイナリ FTP では正しく変換しない場合があります。

2. “プロダクションの開発”の“[プロダクションのエクスポート](#)”で説明されているプロセスを使用して、テスト・プロダクションの新規要素をエクスポートします。

例えば、以下の項目をエクスポートします。

- ・ インタフェースの新しいクラス：ビジネス・プロセス、データ変換、ビジネス・オペレーション、ビジネス・サービス、およびあらゆるユーティリティ・クラス
- ・ ビジネス・ルール
- ・ カスタム・スキーマ

指定した場所に、InterSystems IRIS により配置パッケージ・ファイルが作成されます。

3. 新規要素が含まれる配置パッケージ・ファイルをライブ・プロダクションに配置します。

詳細は、“プロダクションの開発”の“[ターゲット・システムでのプロダクションの配置](#)”を参照してください。

4. オプションで、新しいプロダクション要素のアラートを構成します。

- ・ ビジネス・サービスおよびビジネス・オペレーションには、[エラー時に警告]という構成設定があります。[エラー時に警告]を真に設定している場合、その項目でエラー状態が発生するとすぐ、アラートが自動的にトリガされます。アラートがトリガされた時点で、イベント・ログにメッセージが書き込まれます。また、電子メールや携帯電話でアラートをユーザに通知することもできます。[エラー時に警告]設定を偽に設定している場合、オプションは無効になります。
- ・ [警告猶予期間] (ビジネス・サービスの場合) と [警告再試行猶予期間] (ビジネス・オペレーションの場合) は、有効になっているアラートの頻度を適切に制限します。

注釈 プロダクション全体にアラートを設定するには、“プロダクションの監視”の“[アラートの監視](#)”を参照してください。

5. 新しいインタフェースが新しいメッセージをすべて処理していることを確認します。
6. 以前のインタフェース・テクノロジーを無効化またはクリーンアップします。
 - ・ 以前保留されていたすべての要求が満たされ、すべてのキューが空となっていることを確認します。
 - ・ 古いインタフェースを無効にします。“自動開始”オプションを無効にします。

4

プロダクションのデータベースの管理

プロダクションを開発するには、[相互運用対応ネームスペース](#)を使用するか作成することが必要になります。プロダクションでは、単一のデータベースに依存することもできますが、メッセージに関連するデータを保存するために別のデータベースを定義することが望ましい場合がよくあります。このデータは、トラフィックが通常より多い時間やページ・プロセスの実行がスケジュールされていない場合に、データベース・サイズを急増させる可能性があります。このプラクティスによって、プロダクションのコードと関連付けられているデータベースが、メッセージ・データを含むデータベースから分離されます。ここでは、このプロセスについて説明し、マップすべき[推奨グローバル](#)や、[新規](#)と[既存](#)の両方のシステムでのマップのガイドラインも提示します。

4.1 メッセージ・データを含むグローバル

以下の表は、個別のメッセージ・データベースにマップすべき推奨グローバルを示しています。使用している製品のタブを選択してください。また、メッセージ・データを含むカスタムのメッセージ・クラスがある場合は、同じメッセージ・データベースにこれらをマップすることを検討してください。

テーブル 4-1: InterSystems IRIS

<code>^Ens.MessageHeaderD</code>
<code>^Ens.MessageHeaderI[†]</code>
<code>^Ens.MessageBodyD</code>
<code>^Ens.MessageBodyI[†]</code>
<code>^Ens.MessageBodyS</code>
<code>^EnsHL7.Segment</code>
<code>^EnsLib.H.MessageD</code>
<code>^Ens.Util.LogD</code>
<code>^Ens.Util.LogI[†]</code>

テーブル 4-2: InterSystems IRIS for Health

^Ens.MessageHeaderD
^Ens.MessageHeaderI [†]
^Ens.MessageBodyD
^Ens.MessageBodyI [†]
^Ens.MessageBodyS
^EnsHL7.Segment
^EnsLib.H.MessageD
^Ens.Util.LogD
^Ens.Util.LogI [†]

テーブル 4-3: HealthConnect

^Ens.MessageHeaderD
^Ens.MessageHeaderI [†]
^Ens.MessageBodyD
^Ens.MessageBodyI [†]
^Ens.MessageBodyS
^EnsHL7.Segment
^EnsLib.H.MessageD
^Ens.Util.LogD
^Ens.Util.LogI [†]
^HS.Message.XMLMessageD
^HS.Message.XMLMessageS

テーブル 4-4: HealthShare

^Ens.MessageHeaderD
^Ens.MessageHeaderI [†]
^Ens.MessageBodyD
^Ens.MessageBodyI [†]
^Ens.MessageBodyS
^EnsHL7.Segment
^EnsLib.H.MessageD
^Ens.Util.LogD
^Ens.Util.LogI [†]

<code>^HS.Message.XMLMessageD</code>
<code>^HS.Message.XMLMessageS</code>
<code>^HS.IHE.Comm8F7E.MIMEAttachmentD</code>
<code>^HS.IHE.Comm8F7E.MIMEAttachmentS</code>
<code>^HS.IHE.Comm8F7E.MIMEAttachmentI[†]</code>

注釈 [†]クラスのインデックスを格納するグローバルを、既存のシステムで添え字レベルのマッピングを使用してマップすることはできません。詳細は、“[既存システムでのグローバルのマッピング](#)”を参照してください。

4.2 既定のデータベース用のグローバル

以下のグローバルは、メッセージ・データベースにマップせずに、相互運用対応ネームスペースの既定のデータベースに保持しておく必要があります。これらのグローバルは、相互運用が適切に機能するために不可欠です。

- ・ `^Ens.Runtime`
- ・ `^Ens.Queue`
- ・ `^Ens.JobLock`
- ・ `^Ens.JobStatus`
- ・ `^Ens.ActiveMessage`
- ・ `^Ens.JobRequest`
- ・ `^Ens.Suspended`
- ・ `^Ens.SuspendedAppData`
- ・ `^Ens.Alarm`
- ・ `^Ens.AppData`
- ・ `^Ens.Rule.Notification`

4.3 新しいシステムでのグローバルのマッピング

プロダクションの初期構成時に、メッセージに関連付けられているデータに個別のデータベースを設定する場合、関連グローバルのマッピング・プロセスは簡単です。新しいメッセージ・データベースを作成して、すべての[関連グローバル](#)を新たに作成したデータベースにマップするだけです。データベースへのグローバルのマッピングの詳細は、“[ネームスペースへのグローバル、ルーチン、およびパッケージ・マッピングの追加](#)”を参照してください。

4.4 既存システムでのグローバルのマッピング

既に使用しているプロダクションがある場合、新規データベースへの既存グローバルのマッピング・プロセスは、もう少し複雑です。この場合も、まず新しいメッセージ・データベースを作成します。最上位の添え字として増分 ID を持つグローバルの場合、添え字レベルのマッピングを実行する必要があります。グローバルの現在の最大の添え字より大きい添え字を選択します。これによって、そのグローバルが指定された添え字まで増加したときに、新しいデータベースにロール・オーバーできるようになります。添え字レベルのマッピングの詳細は、“グローバル・マッピングと添え字レベル・マッピング” を参照してください。

クラスのインデックスを格納するグローバルや、\$JOB の添え字を持つグローバルなど、最上位の添え字として増分 ID を持たないグローバルの場合は、DataMove を実行することで、新たに作成したデータベースに添え字をマップできます。