



相互運用プロダクションでの Java Message Service (JMS) の使用（従来の実装）

Version 2024.1

2024-06-06

相互運用プロダクションでの Java Message Service (JMS) の使用 (従来の実装)

InterSystems 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 JMS の概要	1
1.1 JMS メッセージ	1
1.2 Jar ファイル	1
2 JMS ビジネス・サービスおよびオペレーションの構成と使用	3
3 アダプタを使用したカスタム JMS サービスおよびオペレーションの作成	5

1

JMS の概要

重要 このページでは、JMS メッセージングの従来の実装について説明します。新しい実装では、この実装を使用しないでください。今後のリリースで、この従来の実装は廃止される予定です。代わりに、[PEX](#) フレームワークを使用してインターシステムズが実装している JMSPEX 相互運用アダプタ ([受信](#)と[送信](#)) を使用します。または、JMS メッセージング API を使用します。

Java Message Service (JMS) は、複数のシステム間の通信を提供するための Java メッセージング・フレームワークです。このフレームワークで、JMS プロバイダは、JMS クライアントが送信したメッセージのキューを管理します。一般的な JMS メッセージには以下のような経路があります。

1. JMS クライアントが JMS プロバイダにメッセージを送信する。
2. JMS プロバイダが別の JMS クライアントにメッセージを送信する。

相互運用プロダクションにより、インターシステムズ製品は、JMS メッセージを送受信する JMS クライアントになることができます。インターシステムズの JMS クライアントは、`EnsLib.JMS.Operation` ビジネス・ホストを使用してメッセージを JMS プロバイダに送信し、`EnsLib.JMS.Service` ビジネス・ホストを使用して JMS プロバイダからのメッセージを受信します。ObjectScript に精通している上級ユーザは、このような組み込みコンポーネントを使用するのではなく、独自の [カスタム JMS ビジネス・ホスト](#) を作成することもできます。

内部的には、JMS ビジネス・ホストは InterSystems 外部サーバを利用して Java に接続します。

1.1 JMS メッセージ

JMS クライアントの相互運用プロダクションにおいて、JMS メッセージは `EnsLib.JMS.Message` オブジェクトです。メッセージ・オブジェクトの `text` プロパティは、メッセージ・コンテンツが含まれます。メッセージ・オブジェクトの `type` プロパティは、`TextMessage` や `BytesMessage` などのメッセージ・タイプを指定します。`EnsLib.JMS.Message` クラスは、メッセージのプロパティの設定および取得のためのメソッドも提供します。

1.2 Jar ファイル

JMS 機能のための jar ファイルは `install-dir¥dev¥java¥lib¥JDK18¥intersystems-enslib-jms-3.0.0.jar` で入手できます。

以下のクライアント開発用 jar ファイルも提供されています。

- ・ `install-dir¥dev¥java¥jms¥wljmsclient.jar`

- ・ `install-dir¥dev¥java¥jms¥wlthint3client.jar`

2

JMS ビジネス・サービスおよびオペレーションの構成と使用

重要

このページでは、JMS メッセージングの従来の実装について説明します。新しい実装では、この実装を使用しないでください。今後のリリースで、この従来の実装は廃止される予定です。代わりに、[PEX](#) フレームワークを使用してインターシステムズが実装している JMSPEX 相互運用アダプタ ([受信](#)と[送信](#)) を使用します。または、JMS メッセージング API を使用します。

インターシステムズ製品は、組み込みのビジネス・ホストを相互運用プロダクションに追加することにより、JMS クライアントとして構成できます。

インターシステムズ製品が JMS メッセージを受信できるようにするには、新しいビジネス・サービスを相互運用プロダクションに追加して、このビジネス・サービスの **[サービスクラス]** を `EnsLib.JMS.Service` として定義します。このビジネス・サービスは、すべての応答を無視します。

インターシステムズ製品が JMS メッセージを送信できるようにするには、新しいビジネス・オペレーションを相互運用プロダクションに追加して、このビジネス・オペレーションの **[オペレーション・クラス]** を `EnsLib.JMS.Operation` として定義します。このビジネス・オペレーションは、JMS メッセージを送ってきたビジネス・ホストに `EnsLib.JMS.Response` オブジェクトを返します。

これらのビジネス・ホストをプロダクションに追加したら、**[設定]** タブで以下の設定を構成します。

- ・ **[JMSCredentials]** – JMS サーバのユーザ名とパスワードに対して定義される認証情報。JMS のユーザ名とパスワードのための認証情報の作成の詳細は、[“認証情報の定義”](#) を参照してください。
- ・ **[JavaGatewayHost]** および **[JavaGatewayPort]** – JMS サポートを有効にするためにプロダクションが使用している InterSystems 外部サーバの IP アドレスおよびポート。外部サーバは、Java ゲートウェイとも呼ばれます。
`EnsLib.JavaGateway.Service` ビジネス・ホストをプロダクションに追加した場合は、その **[基本設定]** の下の IP アドレスとポートを使用します。
- ・ **[JMSServer]** – JMS サーバの URL。
- ・ **[JMSFactory]** – `QueueConnectionFactory` の名前。
- ・ **[JMSQueue]** – JMS キューの名前。
- ・ **[JMSClientID]** – アクティブな接続の JMS サーバ・リストに表示される名前。

3

アダプタを使用したカスタム JMS サービスおよびオペレーションの作成

重要 このページでは、JMS メッセージングの従来の実装について説明します。新しい実装では、この実装を使用しないでください。今後のリリースで、この従来の実装は廃止される予定です。代わりに、[PEX](#) フレームワークを使用してインターシステムズが実装している JMSPEX 相互運用アダプタ ([受信](#)と[送信](#)) を使用します。または、JMS メッセージング API を使用します。

カスタムの JMS ビジネス・サービスおよびビジネス・オペレーションを作成するには、カスタムの ObjectScript コードの記述が必要となるため、組み込みの JMS サービスおよびオペレーションより多くの開発リソースが必要ですが、Java ゲートウェイ・プロキシ・オブジェクトに直接アクセスできるため、より優れたパフォーマンスを提供します。

カスタムの JMS ビジネス・サービスを開発するには、以下のようにします。

- ・ `EnsLib.JMS.InboundAdapter` をアダプタとして使用して、カスタムの `BusinessService` クラスを実装します。
- ・ `OnProcessInput()` メソッドを、以下のシグニチャでオーバーライドします。

```
Method OnProcessInput(pMessage As %Net.Remote.Object, Output pOutput As %RegisteredObject) As %Status
```
- ・ `pMessage` は、クラス `com.intersystems.enslib.jms.Message` の Java メッセージ・オブジェクトのゲートウェイ・プロキシ・オブジェクトです。Java メッセージ・オブジェクトのプロパティおよびメソッドには、ゲートウェイ・プロキシ・インタフェースを使用してアクセスできます。`pMessage` オブジェクトには、JMS プロバイダから受信したメッセージが含まれます。

カスタムの JMS ビジネス・オペレーションを開発するには、以下のようにします。

- ・ `EnsLib.JMS.OutboundAdapter` をアダプタとして使用して、カスタムの `BusinessOperation` クラスを実装します。
- ・ `OnMessage()` メソッドをオーバーライドするか、メッセージ・マップを実装します。メッセージ・マップの詳細は、“[プロダクションの開発](#)” の “[メッセージ・マップの定義](#)” を参照してください。
- ・ `..Adapter.GetNewMessage(tMessage)` を呼び出して、プロダクション内の別のホストからビジネス・オペレーションに送信したメッセージを取得します。
`tMessage` には、クラス `%Net.Remote.Object` があります。
- ・ `tMessage` は、クラス `com.intersystems.enslib.jms.Message` の Java メッセージ・オブジェクトのゲートウェイ・プロキシ・オブジェクトです。Java メッセージ・オブジェクトのプロパティおよびメソッドには、ゲートウェイ・プロキシ

シ・インタフェースを使用してアクセスできます。Java クラス `com.intersystems.enslib.jms.Message` に実装されているプロパティとメソッドにより、`tMessage` にアクセスします。

- ・ `..Adapter.SendMessage(tMessage)` を呼び出すことにより、JMS プロバイダにメッセージを送信します。

カスタムの JMS ビジネス・サービスおよび JMS ビジネス・オペレーションを開発したら、組み込みの JMS ビジネス・ホストで行ったのと同様にこれらをプロダクションに追加します。