



プロダクション内での IBM WebSphere MQ アダプタの使 用法

Version 2024.1

2024-06-06

プロダクション内での IBM WebSphere MQ アダプタの使用法
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

1 IBM WebSphere MQ アダプタについて	1
1.1 前提条件	1
1.2 IBM WebSphere MQ 受信アダプタ	1
1.3 IBM WebSphere MQ 送信アダプタ	2
1.4 MQ アダプタとイベント・ログ	2
2 IBM WebSphere MQ 受信アダプタの使用法	3
2.1 全般的な動作	3
2.2 受信アダプタを使用するビジネス・サービスの作成	4
2.3 OnProcessInput() メソッドの実装	5
2.4 受信メッセージの使用法	5
2.5 例	5
2.6 ビジネス・サービスの追加と構成	6
3 IBM WebSphere MQ 送信アダプタの使用法	7
3.1 全般的な動作	7
3.2 アダプタを使用するビジネス・オペレーションの作成	7
3.3 メッセージ・ハンドラ・メソッドの作成	8
3.3.1 使用可能なメソッド	9
3.4 例	9
3.5 ビジネス・オペレーションの追加と構成	10
4 トラブルシューティング	11
5 IBM WebSphere MQ アダプタに関する設定	13
5.1 概要	13
5.2 キュー・マネージャ、チャネル、およびキューの指定	13
5.3 使用するログ・ファイルの指定	14
5.4 メッセージの文字セットの指定	14
5.5 メッセージをチェックする頻度の指定	14
5.6 例	15

1

IBM WebSphere MQ アダプタについて

InterSystems IRIS® IBM WebSphere MQ 受信アダプタおよび送信アダプタを使用することによって、プロダクションは、InterSystems IRIS IBM WebSphere MQ のメッセージ・キューとの間でメッセージの送受信を行うことができるようになります。ここでは、これらのアダプタについて簡単に説明します。

IBM WebSphere MQ 製品についてある程度の知識があり、この製品の正式なドキュメントにアクセスできるものとします。

1.1 前提条件

InterSystems IRIS IBM WebSphere MQ の受信アダプタと送信アダプタを使用するために、“IBM WebSphere MQ メッセージの送受信”で説明されているように IBM WebSphere MQ にアクセスできることを確認します。

アダプタは、InterSystems IRIS によってすべての適切なプラットフォームに自動的にインストールされる、ダイナミックリンク・ライブラリを使用します (Windows の場合は **MQInterface.dll** です。プラットフォームによってファイル拡張子は異なります。)同様に、InterSystems IRIS ダイナミックリンク・ライブラリは、IBM WebSphere MQ ダイナミックリンク・ライブラリ (IBM WebSphere MQ クライアント) を必要とします。

1.2 IBM WebSphere MQ 受信アダプタ

通常、受信 IBM WebSphere MQ アダプタ (**EnsLib.MQSeries.InboundAdapter**) は、指定されたチャネルを使用して指定されたキュー・マネージャにアクセスし、指定されたキューからメッセージを取得し、それらのメッセージを (**EnsLib.MQSeries.Message** のインスタンスとして) 関連するビジネス・サービスに送信します。

このアダプタには、以下のような項目の指定に使用する実行時設定が用意されています。

- ・ 使用するキュー・マネージャ。
- ・ 使用するチャネルの指定形式。この指定形式には、使用するチャネルの名前、チャネルが使用するトランスポート、IBM WebSphere MQ サーバを実行するサーバ名 (または IP アドレス)、およびポートが含まれます。
- ・ チェックするメッセージ・キュー。
- ・ アダプタが新規入力をチェックする頻度を制御する、ポーリング間隔。

1.3 IBM WebSphere MQ 送信アダプタ

IBM WebSphere MQ 送信アダプタ (`EnsLib.MQSeries.OutboundAdapter`) の設定によって、以下を指定します。

- ・ 使用するキュー・マネージャ。
- ・ 使用するチャネルの指定形式。この指定形式には、使用するチャネルの名前、チャネルが使用するトランスポート、IBM WebSphere MQ サーバを実行するサーバ名 (または IP アドレス)、およびポートが含まれます。
- ・ メッセージの送信先のメッセージ・キュー。

指定されたキューにメッセージを送信するためのメソッドを提供します。

1.4 MQ アダプタとイベント・ログ

他の多くのプロダクション・アダプタと異なり、IBM WebSphere MQ 受信アダプタおよび送信アダプタは、外部の InterSystems IRIS に送信される、または外部の InterSystems IRIS から取得されるメッセージの処理時にイベント・ログを記録しません。ただし、通常どおりに、メッセージ・ブラウザを使用して、受信トラフィックおよび送信トラフィックを確認することができます。

2

IBM WebSphere MQ 受信アダプタの使用法

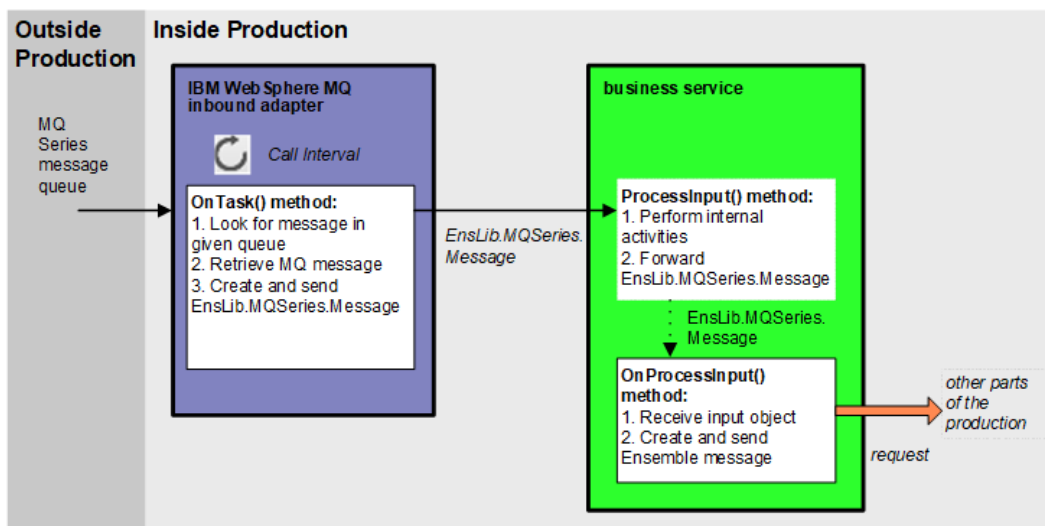
ここでは、InterSystems IRIS® IBM WebSphere MQ 受信アダプタ (`EnsLib.MQSeries.InboundAdapter`) のデフォルトの動作、およびプロダクションでのこのアダプタの使用法について説明します。

2.1 全般的な動作

まず、アダプタに対して指定する詳細について理解しておく役立ちます。`EnsLib.MQSeries.InboundAdapter` クラスには、以下のような項目の指定に使用する実行時設定が用意されています。

- ・ 使用するキュー・マネージャ。
- ・ 使用するチャネルの指定形式。この指定形式には、使用するチャネルの名前、チャネルが使用するトランスポート、IBM WebSphere MQ サーバを実行するサーバ名 (または IP アドレス)、およびポートが含まれます。
- ・ チェックするメッセージ・キュー。
- ・ アダプタが新規入力をチェックする頻度を制御する、ポーリング間隔。

通常、受信 IBM WebSphere MQ アダプタ (`EnsLib.MQSeries.InboundAdapter`) は、指定されたキューを定期的に確認し、メッセージがあれば取得し、それらのメッセージを (`EnsLib.MQSeries.Message` のインスタンスとして) 関連するビジネス・サービスに送信します。ユーザが作成および構成するビジネス・サービスでは、このメッセージを使用してプロダクションの他の部分と通信します。以下の図は、全体的なフローを示しています。



さらに具体的に説明します。

1. アダプタが初期化されると、指定されたチャネルを使用して、指定されたキュー・マネージャおよびキューに接続されます。
2. アダプタは定期的に OnTask() メソッドを実行します。このメソッドは、(メッセージが利用可能な場合に) キューからメッセージを取得します。ポーリング間隔は **CallInterval** 設定によって決定されます。
3. アダプタがメッセージを取得すると、以下を実行します。
 - a. アダプタは **EnsLib.MQSeries.Message** クラスのインスタンスを作成し、メッセージ・データをインスタンスに入れます。
 - b. アダプタは、関連するビジネス・サービス・クラスの内部 **ProcessInput** メソッドを呼び出し、**EnsLib.MQSeries.Message** インスタンスを入力として渡します。
4. ビジネス・サービス・クラスの内部 **ProcessInput()** メソッドが実行されます。このメソッドは、すべてのビジネス・サービスが必要とする内部情報の保持など、基本的なプロダクション・タスクを実行します。ビジネス・サービス・クラスが継承するこのメソッドは、カスタマイズや上書きを行いません。
5. **ProcessInput()** メソッドがカスタムの **OnProcessInput()** メソッドを呼び出し、**EnsLib.MQSeries.Message** インスタンスを入力として渡します。このメソッドの要件については、この後の“[OnProcessInput メソッドの実装](#)”で説明します。

応答メッセージは、同じパスを逆向きにたどります。

2.2 受信アダプタを使用するビジネス・サービスの作成

このアダプタをプロダクションで使用するには、ここに記載されているように新しいビジネス・サービス・クラスを作成します。後で、[それをプロダクションに追加して、構成します](#)。存在しなければ、適切なメッセージ・クラスを作成する必要もあります。“プロダクションの開発”の“[メッセージの定義](#)”を参照してください。

ビジネス・サービス・クラスの基本要件を以下に列挙します。

- ・ ビジネス・サービス・クラスは **Ens.BusinessService** を拡張するものでなければなりません。
- ・ クラスの ADAPTER パラメータは **EnsLib.MQSeries.InboundAdapter** である必要があります。
- ・ このクラスは **OnProcessInput()** メソッドを実装します。これについては“[OnProcessInput メソッドの実装](#)”で説明します。
- ・ その他のオプションと一般情報は、“プロダクションの開発”の“[ビジネス・サービス・クラスの定義](#)”を参照してください。

以下の例は、必要となる一般的な構造を示しています。

Class Definition

```
Class EMQS.Service Extends Ens.BusinessService
{
  Parameter ADAPTER = "EnsLib.MQSeries.InboundAdapter";

  Method OnProcessInput(pInput As EnsLib.MQSeries.Message,
                        pOutput As %RegisteredObject) As %Status
  {
    set tsc=$$$OK
    //your code here
    Quit tsc
  }
}
```


2.3 OnProcessInput() メソッドの実装

カスタム・ビジネス・サービス・クラスにおいて、OnProcessInput() メソッドは以下のシグニチャを持つ必要があります。

```
Method OnProcessInput(pInput As EnsLib.MQSeries.Message,  
                     pOutput As %RegisteredObject) As %Status
```

ここで、pInput は、アダプタがこのビジネス・サービスに送信するメッセージ・オブジェクトです。これは **EnsLib.MQSeries.Message** のインスタンスです。また、pOutput は、メソッド・シグニチャに必要な汎用出力引数です。

OnProcessInput() メソッドは、以下の一部またはすべてを実行する必要があります。

1. MQ メッセージ (**EnsLib.MQSeries.Message**) を調べて、それをどのように使用するかを決定します。後述の“[受信メッセージの使用法](#)”を参照してください。
2. ビジネス・サービスから送信されることになる要求メッセージのインスタンスを作成します。
メッセージ・クラスの作成方法は、“[プロダクションの開発](#)”の“[メッセージの定義](#)”を参照してください。
3. 要求メッセージに対し、MQ メッセージの値を使用して適切にプロパティを設定します。
4. ビジネス・サービスの適切なメソッドを呼び出して、要求をプロダクション内の宛先に送信します。具体的には、SendRequestSync()、SendRequestAsync()、または (あまり一般的ではない) SendDeferredResponse() を呼び出します。詳細は、“[プロダクションの開発](#)”の“[要求メッセージの送信](#)”を参照してください。
これらの各メソッドは、ステータス (具体的には、%Status のインスタンス) を返します。
5. 必ず出力引数 (pOutput) を設定します。通常、受信した応答メッセージと同じように設定します。この手順は必須です。
6. 適切なステータスを返します。この手順は必須です。

このトピックの最後の“[例](#)”を参照してください。

2.4 受信メッセージの使用法

アダプタは、**EnsLib.MQSeries.Message** のインスタンスをビジネス・サービスに送信します。このオブジェクトには、キューから取得したメッセージが含まれます。以下の 3 つのプロパティがあります。

- ・ **Body** — メッセージの本文
- ・ **MessageID** — メッセージ ID
- ・ **BodySize** — メッセージの長さを示す整数

2.5 例

以下のビジネス・サービスは、メッセージを取得し、EMQS.MessageProcessor という名前のビジネス・ホストにそのメッセージを転送します。

Class Definition

```
Class EMQS.Service Extends Ens.BusinessService
{
    Parameter ADAPTER = "EnsLib.MQSeries.InboundAdapter";

    Method OnProcessInput(pInput As EnsLib.MQSeries.Message,
        pOutput As EMQS.InboundMsg) As %Status
    {
        //create production message to carry the retrieved MQ message
        Set inbound=##class(EMQS.InboundMsg).%New()
        Set inbound.Body=pInput.Body
        Set inbound.MessageId=pInput.MessageId

        //forward this to the message processor
        Set tsc=..SendRequestSync("EMQS.MessageProcessor",inbound,.response)
        Set pOutput=response
        Quit tsc
    }
}
```

メッセージ・クラス EMQS.InboundMsg は以下のとおりです。

Class Definition

```
Class EMQS.InboundMsg Extends Ens.Request
{
    Property Body As %String (MAXLEN="");
    Property MessageId As %String(MAXLEN = 128);
}
```

2.6 ビジネス・サービスの追加と構成

ビジネス・サービスをプロダクションに追加するには、管理ポータルを使用して以下の操作を行います。

1. ビジネス・サービス・クラスのインスタンスをプロダクションに追加します。
2. ビジネス・サービスを有効化します。
3. IBM WebSphere MQ キューにアクセスしてメッセージを取得するアダプタを構成します。具体的には、以下を行います。
 - ・ [アクセスするキュー・マネージャ、チャネル、およびキューを指定します](#)。チャネルの指定形式には、使用するチャネルの名前、チャネルが使用するトランスポート、IBM WebSphere MQ サーバを実行するサーバ名（または IP アドレス）、およびポートが含まれます。
 - ・ [使用するエラー・ログを指定します](#)。
 - ・ [メッセージの変換先の文字セットを指定します](#)。
 - ・ [アダプタがメッセージを検索する頻度を指定します](#)。

これらのトピックについては、“[IBM WebSphere MQ アダプタに関する設定](#)”の節で説明します。

4. プロダクションを実行します。

3

IBM WebSphere MQ 送信アダプタの使用法

ここでは、InterSystems IRIS® IBM WebSphere MQ 送信アダプタ (`EnsLib.MQSeries.OutboundAdapter`) の動作、およびプロダクションでのこのアダプタの使用法について説明します。

3.1 全般的な動作

プロダクション内で、送信アダプタは、ユーザが作成および構成するビジネス・オペレーションに関連付けられます。このビジネス・オペレーションはプロダクション内からメッセージを受信し、メッセージ・タイプを調べ、適切なメソッドを実行します。このメソッドは、通常、関連するアダプタのメソッドを実行します。

IBM WebSphere MQ 送信アダプタ (`EnsLib.MQSeries.OutboundAdapter`) の設定によって、以下を指定します。

3.2 アダプタを使用するビジネス・オペレーションの作成

`EnsLib.MQSeries.OutBoundAdapter` を使用するビジネス・オペレーションを作成するために、新しいビジネス・オペレーション・クラスを作成します。後で、[それをプロダクションに追加して、構成します](#)。

存在しなければ、適切なメッセージ・クラスを作成する必要もあります。“プロダクションの開発”の“[メッセージの定義](#)”を参照してください。

ビジネス・オペレーション・クラスの基本要件を以下に列挙します。

- ・ ビジネス・オペレーション・クラスは、`Ens.BusinessOperation` を拡張するものでなければなりません。
- ・ クラスの ADAPTER パラメータは `EnsLib.MQSeries.OutboundAdapter` である必要があります。
- ・ クラスの INVOCATION パラメータは、使用する呼び出しスタイルを指定する必要があります。以下のいずれかを使用します。
 - **Queue** は、メッセージが 1 つのバックグラウンド・ジョブ内で作成され、元のジョブが解放された段階でキューに配置されます。その後、メッセージが処理された段階で、別のバックグラウンド・ジョブがそのタスクに割り当てられます。これは最も一般的な設定です。
 - **InProc** は、メッセージが、作成されたジョブと同じジョブで生成、送信、および配信されることを意味します。このジョブは、メッセージが対象に配信されるまで送信者のプールに解放されません。これは特殊なケースのみに該当します。

- ・ クラスでは、少なくとも1つのエントリを含むメッセージ・マップを定義します。メッセージ・マップは、以下の構造を持つ XData ブロック・エントリです。

```
XData MessageMap
{
  <MapItems>
    <MapItem MessageType="messageclass">
      <Method>methodname</Method>
    </MapItem>
    ...
  </MapItems>
}
```

- ・ クラスでは、メッセージ・マップ内で名前が付けられたすべてのメソッドを定義します。これらのメソッドは、メッセージ・ハンドラと呼ばれます。各メッセージ・ハンドラは、以下のシグニチャを持っている必要があります。

```
Method Sample(pReq As RequestClass, Output pResp As ResponseClass) As %Status
```

ここで、Sample はメソッド名、RequestClass は要求メッセージ・クラスの名前、ResponseClass は応答メッセージ・クラスの名前です。通常、これらのメソッドは、ビジネス・オペレーションの **Adapter** プロパティのプロパティおよびメソッドを参照します。

- ・ その他のオプションと一般情報は、“プロダクションの開発”の“[ビジネス・オペレーション・クラスの定義](#)”を参照してください。

以下の例は、必要となる一般的な構造を示しています。

Class Definition

```
Class EMQS.NewOperation1 Extends Ens.BusinessOperation
{
  Parameter ADAPTER = "EnsLib.MQSeries.OutboundAdapter";

  Parameter INVOCATION = "Queue";

  Method SampleCall(pRequest As Ens.Request,
                    Output pResponse As Ens.Response) As %Status
  {
    Quit $$$ERROR($$$NotImplemented)
  }

  XData MessageMap
  {
    <MapItems>
      <MapItem MessageType="Ens.Request">
        <Method>SampleCall</Method>
      </MapItem>
    </MapItems>
  }
}
```

3.3 メッセージ・ハンドラ・メソッドの作成

EnsLib.MQSeries.OutboundAdapter で使用するビジネス・オペレーション・クラスを作成する場合の主なタスクは、通常、このアダプタで使用するメッセージ・ハンドラ、つまり、プロダクション・メッセージを受信して IBM WebSphere MQ サーバにメッセージを送信するメソッドの記述です。

各メッセージ・ハンドラ・メソッドは、以下のシグニチャを持っている必要があります。

```
Method Sample(pReq As RequestClass, Output pResp As ResponseClass) As %Status
```

ここで、Sample はメソッド名、RequestClass は要求メッセージ・クラスの名前、ResponseClass は応答メッセージ・クラスの名前です。

通常、このメソッドは以下の操作を実行します。

1. 受信要求メッセージを調べます。
2. アダプタの `SendMessage()` メソッドを呼び出し、IBM WebSphere MQ の構成されたキューにメッセージを送信します。
3. 応答を調べます。
4. 応答内の情報を使用して、応答メッセージ (**Ens.Response** またはサブクラスのインスタンス) を作成します。メソッドは出力としてこのメッセージを返します。
メッセージ・クラスの定義方法は、“プロダクションの開発” の “[メッセージの定義](#)” を参照してください。
5. 必ず出力引数 (`pOutput`) を設定します。通常、応答メッセージと同じように設定します。この手順は必須です。
6. 適切なステータスを返します。この手順は必須です。

3.3.1 使用可能なメソッド

このアダプタは、以下のメソッドを提供します。

SendMessage()

```
Method SendMessage(pBody) As %Status
```

IBM WebSphere MQ メッセージの構成されたキューにメッセージを送信します。pBody にはデータ型または文字ストリームを使用できますが、バイナリ・データやオブジェクトは使用できません。

3.4 例

以下のビジネス・オペレーションは、メッセージを IBM WebSphere MQ に送信します。

Class Definition

```
Class EMQS.Operation Extends Ens.BusinessOperation
{
    Parameter ADAPTER = "EnsLib.MQSeries.OutboundAdapter";
    Parameter INVOCATION = "Queue";

    Method Send(pRequest As OutboundMsg, Output pResponse As OutboundMsgResponse) As %Status
    {
        Set string=pRequest.Body

        //Get part of the message so that we can provide
        //some information in the response message
        Set snippet=$Extract(string,1,50)

        //send the message to the configured queue
        Set status=..Adapter.SendMessage(string)
        If $$$ISERR(status) {
            Do $System.Status.DisplayError(status)
            Quit $$$ERROR($$$$GeneralError,"Error sending message")
        }

        //create the response message
        Set pResponse=##class(EMQS.OutboundMsgResponse).%New()
        Set pResponse.Body="Message sent: "_snippet

        Quit status
    }
}

XData MessageMap
{
```

```
<MapItems>
  <MapItem MessageType="EMQS.OutboundMsg">
    <Method>Send</Method>
  </MapItem>
</MapItems>
}
```

3.5 ビジネス・オペレーションの追加と構成

ビジネス・オペレーションをプロダクションに追加するには、管理ポータルを使用して以下の操作を行います。

1. カスタム・ビジネス・オペレーション・クラスのインスタンスをプロダクションに追加します。
2. ビジネス・オペレーションを有効化します。
3. IBM WebSphere MQ キューにアクセスしてメッセージを送信するアダプタを構成します。具体的には、以下を行います。
 - ・ [アクセスするキュー・マネージャ、チャネル、およびキューを指定します](#)。チャネルの指定形式には、使用するチャネルの名前、チャネルが使用するトランスポート、IBM WebSphere MQ サーバを実行するサーバ名（または IP アドレス）、およびポートが含まれます。
 - ・ [使用するエラー・ログを指定します](#)。
 - ・ [メッセージを記述する文字セットを指定します](#)。

これらのトピックについては、“[IBM WebSphere MQ アダプタに関する設定](#)” の節で説明します。

4. プロダクションを実行します。

4

トラブルシューティング

IBM WebSphere MQ に対してプロダクション・アダプタを使用するときに問題が発生した場合は、まず、クライアントが正しくインストールされていること、およびサーバと通信できることを確認します。このようなテストを実行するには、IBM WebSphere MQ によって提供されるサンプル・プログラムを使用できます。実行ファイルは、IBM WebSphere MQ クライアントの **bin** ディレクトリにあります。

以下の手順は、Windows でこれらのサンプル・プログラムを使用する方法です。その他のオペレーティング・システムの場合、詳細が異なる場合があります。IBM のドキュメントを参照し、自分のクライアントに含まれるファイル名を確認してください。

1. MQSERVER という名前の環境変数を作成します。この値の形式は、channel_name/transport/server です。ここで、channel_name は使用するチャネルの名前、transport は使用するトランスポートを指定する文字列、server はサーバ名です。例えば、S_antigua/TCP/antigua です。
2. コマンドラインで、以下のコマンドを入力します。

```
amqsputc queue_name queue_manager_name
```

ここで、queue_name は使用するキューの名前、queue_manager_name はキュー・マネージャの名前です。以下に例を示します。

```
amqsputc testqueue QM_antigua
```

amqsputc コマンドが認識されない場合は、PATH 環境変数が IBM WebSphere MQ クライアントの **bin** ディレクトリを含むように更新されていることを確認します。

その他のエラーの場合は、IBM のドキュメントを参照してください。

3. 以下のような 2 行が表示されるはずです。

```
Sample AMQSPUT0 start  
target queue is testqueue
```

4. これでメッセージを送信できます。各メッセージを入力し、各メッセージの後に Enter キーを押します。以下に例を示します。

```
sample message 1  
sample message 2
```

5. メッセージの送信が完了したら、Enter キーを 2 度押します。以下のような行が表示されます。

```
Sample AMQSPUT0 end
```

6. このテストを完了するため、キューに送信したメッセージを取得します。コマンドラインで、以下のコマンドを入力します。

```
amqsgetc queue_name queue_manager_name
```

ここで、queue_name は使用するキューの名前、queue_manager_name はキュー・マネージャの名前です。以下に例を示します。

7. 以下のように、開始行に続いて、先に送信したメッセージが表示されます。

```
Sample AMQSGET0 start  
message <sample message 1>  
message <sample message 2>
```

8. このプログラム例は、他のメッセージの受信を短時間待機し、以下を表示します。

```
no more messages  
Sample AMQSGET0 end
```

テストに失敗した場合は、IBM のドキュメントを参照してください。問題の原因として、以下が考えられます。

- ・ セキュリティの問題
- ・ キューが正しく定義されていない
- ・ キュー・マネージャが開始されていない

5

IBM WebSphere MQ アダプタに関する設定

ここでは、IBM WebSphere MQ アダプタの参照情報を提供します。

“プロダクションの管理” の “[すべてのプロダクションに含まれる設定](#)” も参照してください。

5.1 概要

IBM WebSphere MQ の受信アダプタおよび送信アダプタの両方に対して、指定されたキューへの接続を記述する設定を指定します。詳細は両方のアダプタでほとんど同じです。

このドキュメントの最初の章にある “[前提条件](#)” も参照してください。

5.2 キュー・マネージャ、チャネル、およびキューの指定

以下の設定に対して値を指定し、目的のメッセージ・キュー、使用するキュー・マネージャ、および使用するチャネルを指定します。

QueueName

(必須) キューの名前を指定します。指定されたキュー・マネージャに対して有効なキューでなければなりません。また、このキューを使用する権限を持っている必要があります。

QueueManager

キュー・マネージャを指定します。これは、IBM WebSphere MQ サーバ上の有効なキュー・マネージャであり、それを使用する権限を持っている必要があります。

この設定を省略すると、システムは、IBM WebSphere MQ で構成されたデフォルトのキュー・マネージャを使用します。また、IBM WebSphere MQ でキューの名前によってキュー・マネージャが決定されるように構成されている場合、システムは、指定されたキューの名前に適切なキュー・マネージャを使用します。

Channel

以下の形式でチャネルを指定します。

```
"channel_name/transport/host_name(port)"
```

ここで、channel_name は使用するチャネルの名前、transport はチャネルが使用するトランスポート、host_name は IBM WebSphere MQ サーバを実行するサーバ名 (または IP アドレス)、port はこのチャネルが使用するポートです。

transport には、TCP、LU62、NETBIOS、SPX のいずれかを使用できます。

以下に例を示します。

```
"CHAN_1/TCP/rodan(1401)"
```

```
"CHAN_1/TCP/127.0.0.1(1401)"
```

この設定を省略すると、システムは、IBM WebSphere MQ で構成されたデフォルトのチャネル指定形式を使用します。また、システムでキューの名前によってチャネルが決定されるように構成されている場合、システムは、指定されたキューの名前に適切なチャネルを使用します。

5.3 使用するログ・ファイルの指定

以下の設定を使用して、書き込むログ・ファイルを指定します。

ErrorFile

エラー・メッセージを書き込むログ・ファイルを指定します。この設定を省略すると、ログは記録されません。

5.4 メッセージの文字セットの指定

メッセージ変換に使用する文字セットを指定できます。

CharSet

IBM WebSphere MQ で使用される整数の Coded Character Set ID (CCSID) を指定します。

- 受信アダプタの場合、これはメッセージの変換先の文字セットです。文字セットを指定しない場合、MQ システムでは、メッセージは MQ クライアントに対して指定されたデフォルトの文字セットを使用するものと見なされます。
- 送信アダプタの場合、この設定はメッセージが記述されている文字セットを指定し、変換は行われません。

文字セットおよび変換テーブルの詳細は、“変換テーブル”を参照してください。

5.5 メッセージをチェックする頻度の指定

これは受信アダプタにのみ適用されます。メッセージをチェックする頻度を指定するには、以下の設定を使用します。

CallInterval

アダプタが新しいメッセージをもう一度チェックするまで待機する秒数です。この秒数を超えると、プロダクション・フレームワークからのシャットダウン信号を確認します。

アダプタは、入力を検出すると、データを取得してビジネス・サービスに渡します。ビジネス・サービスはデータを処理し、アダプタは直ちに新規入力の待機を開始します。プロダクションが実行中であり、ビジネス・サービスが有効化され、アクティブになるようにスケジュールされている場合、このサイクルは常に継続されます。

`CallInterval` のデフォルト値は 5 秒です。最小値は 0.1 秒です。

ここに記載されていない設定については、“[プロダクションの構成](#)”を参照してください。

5.6 例

このドキュメントの例では、ポート 1414 を使用し、`antigua` というマシンで実行している IBM WebSphere MQ サーバを使用しました。この例を実行したユーザ ID には、`QM_antigua` というキュー・マネージャと `testqueue` というキューの使用が許可されていました。InterSystems IRIS は、`S_antigua` という名前の TCP チャネルを使用してサーバと通信しました。以下のテーブルに設定をリストします。

設定	値
Queue Manager	QM_antigua
Channel	S_antigua/TCP/antigua(1414)
Queue Name	testqueue
Error File	c:\mq-log.txt

