プロダクション内での FTP アダプタの使用法

Version 2024.1
2024-06-06

InterSystems Corporation　1 Memorial Drive　Cambridge MA 02142　www.intersystems.com

プロダクション内での FTP アダプタの使用法

InterSystems IRIS Data Platform　Version 2024.1　2024-06-06

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

目次

1

FTP 受信アダプタの使用法

このページでは、FTP 受信アダプタ (EnsLib.FTP.InboundAdapter) の使用方法について説明します。

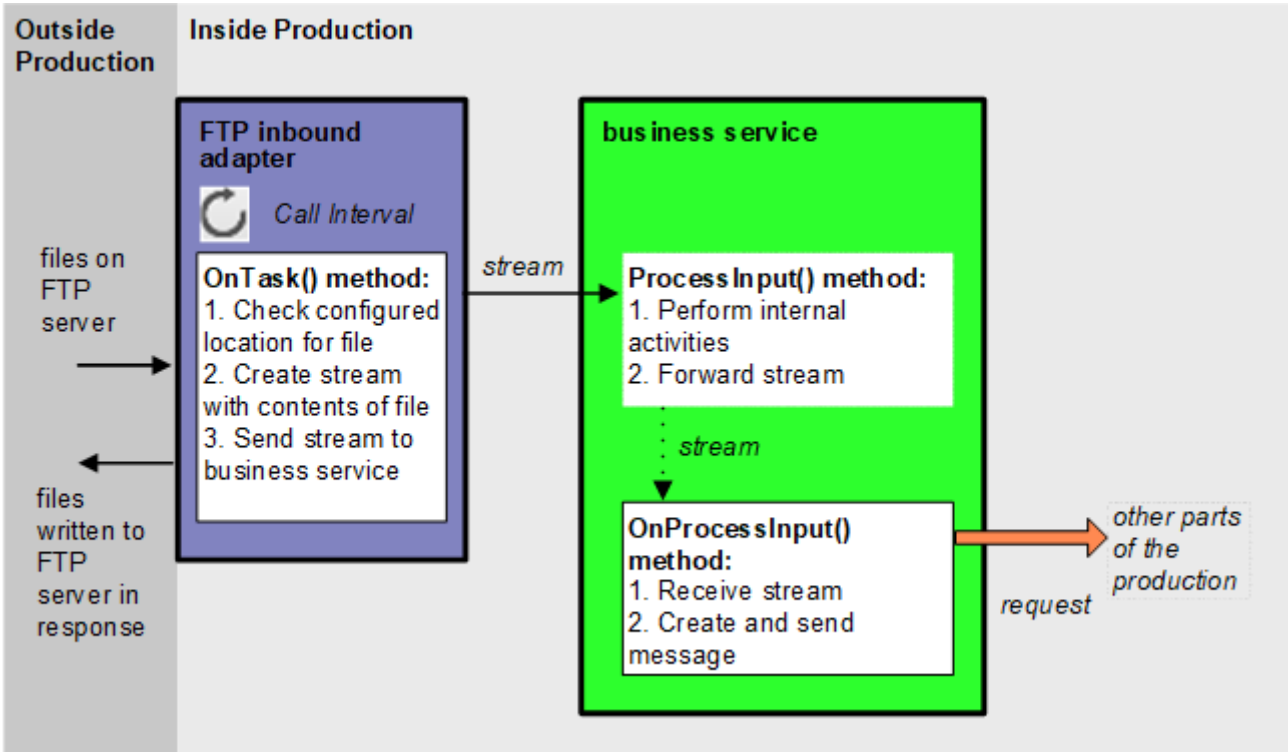
Tipヒント InterSystems IRIS® データ・プラットフォームでは、このアダプタを使用する特殊なビジネス・サービス・クラスとユーザのニーズに適したビジネス・サービス・クラスの 1 つも提供されます。そのため、プログラミングの必要がありません。"相互運用プロダクションの概要" の “接続オプション” を参照してください。

1.1 全般的な動作

EnsLib.FTP.InboundAdapter により、InterSystems IRIS は FTP プロトコル経由でファイルを受信できます。このアダプタは、構成された場所から FTP 入力を受信したり、入力を読み取ったり、入力をストリームとして関連するビジネス・サービスに送信したりします。ユーザが作成および構成するビジネス・サービスは、このストリームを使用してプロダクションの他の部分と通信します。

入力に対する通知または応答が FTP サーバに必要な場合、この応答の作成とデータ・ソースへの返送も、ビジネス・サービスが EnsLib.FTP.InboundAdapter を経由して行います。アダプタは応答を評価したり補足したりしませんが、応答を受け取った場合はこれを渡します。

下の図は、受信メッセージの全体フローを示しています (応答は含まれていません)。



詳細は、以下のとおりです。

1. アダプタは、構成されたデータ・ソースからの入力を検出するたびに、ビジネス・サービス・クラスの内部 ProcessInput() メソッドを呼び出して、ストリームを入力引数として渡します。
2. 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。
3. 次に、ProcessInput() メソッドがカスタムの OnProcessInput() メソッドを呼び出し、ストリーム・オブジェクトを入力として渡します。このメソッドの要件については、"OnProcessInput() メソッドの実装" で説明します。

   データ・ソースが何らかの確認または応答を必要としている場合は、ビジネス・サービスの OnProcessInput() メソッドがそれを作成します。受信アダプタは、単にこの応答を外部データ・ソースに返すだけです。

応答メッセージは、同じパスを逆向きにたどります。

1.2 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、該当するメッセージ・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。
- クラス内の ADAPTER パラメータは EnsLib.FTP.InboundAdapter と一致する必要があります。
- 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"、および "ビジネス・サービス・クラスの例" を参照してください。

1.3 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 %CharacterStream,pOutput As %RegisteredObject) As %Status
```

または、以下のようになります。

```
Method OnProcessInput(pInput As %BinaryStream,pOutput As %RegisteredObject) As %Status
```

説明：

- 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、予想されるストリームの内容に応じて、%CharacterStream と %BinaryStream のどちらかの型にすることができます。アダプタ設定 ([文字セット]) を使用して、入力ストリームが文字かバイナリかを指定します。設定のリファレンスを参照してください。
- 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. 入力ストリーム (pInput) を検査して、その使用方法を決定します。

   この入力の型は、Charset アダプタ設定の値により異なります。

   - Charset 設定にバイナリ値がある場合、pInput は %BinaryStream 型となり、バイトが含まれます。
   - バイナリ値がない場合は、pInput は %CharacterStream 型となり、文字が含まれます。

     Charset の詳細は、設定のリファレンス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. 要求メッセージの場合は、必要に応じて、入力内の値を使用してその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。

1.3.1 アダプタ・メソッドの呼び出し

ビジネス・サービス内では、以下のアダプタのインスタンス・メソッドを呼び出すことができます。

Connect()

```
Method Connect(pTimeout As %Numeric = 30,
               pInbound As %Boolean = 0) As %Status
```

FTP サーバに接続してログインし、ディレクトリと転送モードを設定します。

Disconnect()

```
Method Disconnect(pInbound As %Boolean = 0) As %Status
```

FTP サーバから切断します。

ParseFilename()

```
Method ParseFilename(pFilenameLine As %String,
                     Output pTimestamp As %String,
                     Output pSize As %String) As %Boolean
```

TestConnection()

```
Method TestConnection()
```

接続されていてもソケットを失っているとアダプタが判断した場合に、接続状態を示すプロパティを修正します。

以下のメソッドも使用できます。各メソッドは、ユーザが管理ポータルで調整可能なアダプタ設定に対応しています。ユーザが **[適用]** をクリックして [設定] の値の変更内容を受け入れるたびに、対応の Setting`Set` メソッドが実行します。いずれかの設定を変更した後に、これらのメソッドを使用して以下のように調整することができます。各設定の詳細な説明は、設定のリファレンスを参照してください。

ArchivePathSet()

```
Method ArchivePathSet(pInVal As %String) As %Status
```

ArchivePath は、アダプタがファイルの処理を完了した後に、各ファイルのコピーを格納するディレクトリです。

CharsetSet()

```
Method CharsetSet(cset As %String) As %Status
```

[文字セット] は、入力ファイルの文字セットです。

ConnectedSet()

```
Method ConnectedSet(pValue As %Boolean) As %Status
```

[Connected] は、FTP サーバへのアダプタの接続を追跡する内部プロパティです。

CredentialsSet()

```
Method CredentialsSet(pInVal As %String) As %Status
```

[認証情報] は、FTP サーバへの接続を承認できるプロダクション認証情報エントリです。プロダクション認証情報の作成方法は、"プロダクションの構成" を参照してください。

FilePathSet()

```
Method FilePathSet(path As %String) As %Status
```

[FilePath] は、FTP サーバ上のファイルの検索先ディレクトリです。

FTPPortSet()

```
Method FTPPortSet(port As %String) As %Status
```

[FTPPort] は、接続先の FTP サーバ上の TCP ポートです。

FTPServerSet()

```
Method FTPServerSet(server As %String) As %Status
```

[FTPServer] は、接続先の FTP サーバです。ここには IP アドレスか、ドメイン・ホスト・コントローラが名前を解決できるのであればサーバ名を指定できます。

SSLConfigSet()

```
Method SSLConfigSet(sslcfg As %String) As %Status
```

[SSL構成] は、この接続の認証に使用する TLS 構成のエントリです。

1.4 ビジネス・サービス・クラスの例

EnsLib.FTP.InboundAdapter を参照するビジネス・サービス・クラスのコードの例を以下に示します。

Class Definition

```
Class EnsLib.FTP.PassthroughService Extends Ens.BusinessService
{

Parameter ADAPTER = "EnsLib.FTP.InboundAdapter";

/// Configuration item(s) to which to send file stream messages
Property TargetConfigNames As %String(MAXLEN = 1000);

Parameter SETTINGS = "TargetConfigNames";

/// Wrap the input stream object in a StreamContainer message object and send
/// it. If the adapter has a value for ArchivePath, send async; otherwise send
/// synchronously to ensure that we don't return to the Adapter and let it
/// delete the file before the target Config Item is finished processing it.

Method OnProcessInput(pInput As %Stream.Object,
                      pOutput As %RegisteredObject) As %Status
{
  Set tSC=$$$OK, tSource=pInput.Attributes("Filename"),
                 tFileLocation=pInput.Attributes("FTPDir"),
                 pInput=##class(Ens.StreamContainer).%New(pInput)
  Set tWorkArchive=(""'=..Adapter.ArchivePath)

  For iTarget=1:1:$L(..TargetConfigNames, ",") {
    Set tOneTarget=$ZStrip($P(..TargetConfigNames,",",iTarget),"<>W")
    Continue:""=tOneTarget
    $$$sysTRACE("Sending input Stream ...")

    If tWorkArchive {
      Set tSC1=..SendRequestAsync(tOneTarget,pInput)
      Set:$$$ISERR(tSC1) tSC=$$$ADDSC(tSC,tSC1)
    } Else {
      #; If not archiving send Sync to avoid Adapter deleting file before
      #; Operation gets it
      Set tSC1=..SendRequestSync(tOneTarget,pInput)
      Set:$$$ISERR(tSC1) tSC=$$$ADDSC(tSC,tSC1)
      }
    }
    Quit tSC
  }
}
```

この例では、以下の 2 つの変数を設定して、受信ストリーム pInput に関するメタデータを取得します。

- tSource は、受信ストリームの **Attributes** プロパティの Filename 添え字に格納されている元のファイル名を取得します。

- tFileLocation は、同じプロパティの FTPDir 添え字に格納されている元の完全なファイル・パスを取得します。

1.5 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・サービスを構成します。設定の詳細は、設定のリファレンスを参照してください。
3. ビジネス・サービスを有効化します。
4. プロダクションを実行します。

2

FTP 送信アダプタの使用法

このページでは、FTP 送信アダプタ (EnsLib.FTP.OutboundAdapter) の使用方法について説明します。

Tipヒント InterSystems IRIS® データ・プラットフォームでは、このアダプタを使用する特殊なビジネス・サービス・クラスとユーザのニーズに適したビジネス・サービス・クラスの 1 つも提供されます。そのため、プログラミングの必要がありません。"相互運用プロダクションの概要" の "接続オプション" を参照してください。

2.1 全般的な動作

EnsLib.FTP.OutboundAdapter を使用すれば、プロダクションから FTP プロトコル経由でファイルを送信できます。このアダプタを使用するために、アダプタを使用するビジネス・オペレーションを作成して構成します。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。

2.2 送信アダプタを使用するビジネス・オペレーションの作成

EnsLib.FTP.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、そのクラス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、該当するメッセージ・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.FTP.OutboundAdapter と一致する必要があります。
- 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。後で、メッセージが処理されるときに、別のバックグラウンド・ジョブがこ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、メソッド・コードは、ビジネス・オペレーションのプロパティおよび Adapter プロパティのメソッドを参照します。

  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。

  メッセージ・ハンドラ・メソッドの定義方法は、"メッセージ・ハンドラ・メソッドの作成" を参照してください。

- 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。

2.3 メッセージ・ハンドラ・メソッドの作成

EnsLib.FTP.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最大のタスクは、このアダプタで使用されるメッセージ・ハンドラ、つまり、プロダクション・メッセージを受信して FTP 経由でやり取りするメソッドの作成です。

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. EnsLib.FTP.OutboundAdapter 内のメソッドの 1 つを呼び出すときは、Adapter プロパティを使用してアダプタ・オブジェクトを指定します。下の例は、ビジネス・オペレーション・メソッドから PutStream() メソッドを呼び出します。

   ```
   Quit ..Adapter.PutStream(pFilename,..%TempStream)
   ```

   同様の構文を使用して、"ビジネス・オペレーションからのアダプタ・メソッドの呼び出し" に記載されたメソッドのいずれかを呼び出すことができます。
3. 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。

4. 適切なステータスを返します。この手順は必須です。

2.3.1 ビジネス・オペレーションからのアダプタ・メソッドの呼び出し

ビジネス・オペレーション・クラスが EnsLib.FTP.OutboundAdapter を参照するときに、以下のアダプタ・メソッドが使用可能になります。前のトピックで説明した Adapter プロパティを使用して、ビジネス・オペレーション・クラス内のメソッドからこれらのアダプタ・メソッドを呼び出せます。

Connect()

```
Method Connect(pTimeout As %Numeric = 30,
               pInbound As %Boolean = 0) As %Status
```

FTP サーバに接続してログインし、ディレクトリと転送モードを設定します。

FTP 送信アダプタには再試行サイクルがあり、以下の場合に実行して接続を再確立します。

- FTP サーバへの TCP 接続の確立を試行している間に、TCP ソケットの失敗またはタイムアウトが発生した場合。
- ローカル・クライアントがネットワーク・コードで応答した場合 (タイムアウト)。
- ローカル・クライアントが再試行可能コードで応答した場合 (タイムアウトのコード 529 などの FTP コード 52x または 4xx)。

CreateTimestamp()

```
ClassMethod CreateTimestamp(pFilename As %String = "",
                            pSpec As %String = "")
                            As %String
```

基準として pFilename 文字列を使用し、pSpec で指定されているタイム・スタンプ指定子を組み込んで、結果の文字列を返します。デフォルトのタイム・スタンプ指定子は %f_%Q です。指定子のそれぞれの意味は次のとおりです。

- %f は、データ・ソースの名前 (この場合は入力ファイル名) を表します。
- _ は、リテラルのアンダースコア文字で、出力ファイル名に表示されます。
- %Q は、ODBC 形式の日付と時刻を表します。

書式コード %f に具体的な値を使用すると、その値に使用している |、?、¥、/、:、[、]、<、>、&、,、;、NUL、BEL、TAB、CR、LF の各文字はすべて削除され、空白は下線 (_)、スラッシュ (/) はハイフン (-)、コロン (:) はドット (.) にそれぞれ置き換えられます。

タイム・スタンプ規則の詳細は、"ファイル名に関するタイム・スタンプ指定" を参照してください。

Delete()

```
Method Delete(pFilename As %String) As %Status
```

指定されたファイルを FTP サーバから削除します。サーバ、ユーザ名、パスワード、およびファイル・パスは、このアダプタの設定としてすでに構成されています。FTP オペレーションの成功を示すステータス値を返します。

Disconnect()

```
Method Disconnect(pInbound As %Boolean = 0) As %Status
```

FTP サーバから切断します。

GetStream()

```
Method GetStream(pFilename As %String,
                 ByRef pStream As %Stream.Object = $$$NULLOREF) As %Status
```

指定されたファイルを FTP サーバから取得して、ストリームとして返します。サーバ、ユーザ名、パスワード、ファイル・パス、および転送モード (文字セット) は、このアダプタの設定としてすでに構成されています。このメソッドは、FTP オペレーションの成功を示すステータス値を返します。呼び出し元がストリームを提供する場合、このストリームは転送に適したタイプ (文字またはバイナリ) である必要があります。何も提供されていない場合は、このメソッドはストリームを作成します。

NameList()

```
Method NameList(Output pFileList As %ListOfDataTypes) As %Status
```

FTP サーバ上のファイルのリストを取得します。サーバ、ユーザ名、パスワード、およびファイル・パスは、このアダプタの設定としてすでに構成されています。ファイル名は %ListOfDataTypes オブジェクト内に返されます。このメソッドは、FTP オペレーションの成功を示すステータス値を返します。

PutStream()

```
Method PutStream(pFilename As %String,
                 pStream As %Stream.Object) As %Status
```

FTP サーバにストリームを指定されたファイルとして保存します。サーバ、ユーザ名、パスワード、ファイル・パス、および転送モード (文字セット) は、このアダプタの設定としてすでに構成されています。このメソッドは、FTP オペレーションの成功を示すステータス値を返します。

Rename()

```
Method Rename(pFilename As %String,
                 pNewFilename As %String,
                 pNewPath As %String = "") As %Status
```

FTP サーバ上のファイルの名前を変更します。サーバ、ユーザ名、パスワード、およびファイル・パスは、このアダプタの設定としてすでに構成されています。このメソッドは、FTP オペレーションの成功を示すステータス値を返します。

TestConnection()

```
Method TestConnection()
```

接続されていてもソケットを失っているとアダプタが判断した場合に、接続状態を示すプロパティを修正します。

以下のメソッドも使用できます。各メソッドは、ユーザが管理ポータルで調整可能なアダプタ設定に対応しています。ユーザが設定を変更するたびに、対応する Setting`Set` メソッドが実行されます。いずれかの設定を変更した後に、これらのメソッドを使用して以下のように調整することができます。各設定の詳細な説明は、設定のリファレンスを参照してください。

CharsetSet()

```
Method CharsetSet(cset As %String) As %Status
```

[**文字セット**] は、入力ファイルの文字セットです。

ConnectedSet()

```
Method ConnectedSet(pValue As %Boolean) As %Status
```

[Connected] は、FTP サーバへのアダプタの接続を追跡する内部プロパティです。

CredentialsSet()

```
Method CredentialsSet(pInVal As %String) As %Status
```

[認証情報] は、FTP サーバへの接続を承認できるプロダクション認証情報エントリです。プロダクション認証情報の作成方法は、"プロダクションの構成" を参照してください。

FilePathSet()

```
Method FilePathSet(path As %String) As %Status
```

[FilePath] は、FTP サーバ上のファイルの検索先ディレクトリです。

FTPPortSet()

```
Method FTPPortSet(port As %String) As %Status
```

[FTPPort] は、接続先の FTP サーバ上の TCP ポートです。

FTPServerSet()

```
Method FTPServerSet(server As %String) As %Status
```

[FTPServer] は、接続先の FTP サーバです。ここには IP アドレスか、ドメイン・ホスト・コントローラが名前を解決できるのであればサーバ名を指定できます。

SSLConfigSet()

```
Method SSLConfigSet(sslcfg As %String) As %Status
```

[SSL構成] は、この接続の認証に使用する TLS 構成のエントリです。

2.4 ビジネス・オペレーション・クラスの例

EnsLib.FTP.OutboundAdapter を参照するビジネス・オペレーション・クラスのコードの例を以下に示します。

Class Definition

```
Class EnsLib.FTP.PassthroughOperation Extends Ens.BusinessOperation
{
Parameter ADAPTER = "EnsLib.FTP.OutboundAdapter";

/// Name of file to output the document(s) to. May include timestamp
/// specifiers. The %f specifier if present will be replaced with the
/// name of the document's original source filename (stripped of
/// characters illegal in filenames).  See the method
/// Ens.Util.File.CreateTimestamp() for documentation of timestamping options.

Property Filename As %String(MAXLEN = 1000);

Parameter SETTINGS As %String = "Filename";

Method OnMessage(pRequest As Ens.StreamContainer,
                 Output pResponse As %Persistent) As %Status
  {
  Set tFilename=
  ..Adapter.CreateTimestamp(##class(%File).GetFilename(
    pRequest.Stream.Attributes("Filename")),..Filename)
```

```
  Quit ..Adapter.PutStream(tFilename, pRequest.Stream)
  }
}
```

2.5 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. 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・サービスを構成します。設定の詳細は、設定のリファレンスを参照してください。
3. ビジネス・オペレーションを有効化します。
4. プロダクションを実行します。

FTP アダプタ設定

このページでは、FTP の受信アダプタおよび送信アダプタに関するリファレンス情報を提供します。"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。

FTP 受信アダプタに関する設定

FTP 受信アダプタ EnsLib.FTP.InboundAdapter の設定に関する参照情報を提供します。

概要

受信 FTP アダプタには以下の設定があります。

グループ	設定
基本設定	[外部レジストリ ID]、[ファイルパス]、[サーバから削除]、[ファイル仕様のデリミタ]、[ファイルスペック]、[アーカイブパス]、[呼び出し間隔]、[FTPサーバ]、[FTPポート]、[認証情報]
接続設定	[プロトコル]、[SSL構成]、[SSL でサーバIDを確認する]、[SSL セッション再開を使用]、[UsePASV]、[接続中を維持]、[接続タイムアウト]
FTP 設定	コマンド変換テーブル
SFTP 設定	[SFTPAuthenticationMethods]、[SFTP公開鍵ファイル]、[SFTP秘密鍵ファイル]、[SFTP Server Character Set]、[SFTP Local Character Set]、[SFTP パスフレーズ資格情報]、[SFTPInteractiveDTL]
追加設定	[ファイルストリーム使用]、[サーバリストスタイル]、[サブディレクトリレベル]、[文字セット]、[タイムスタンプ追加]、[完了を確認]、[ファイル・アクセス・タイムアウト]、[セマフォ仕様]

残りの設定はすべてのビジネス・サービスに共通しています。詳細は、"すべてのビジネス・サービスに含まれる設定" を参照してください。

[タイムスタンプ追加]

[アーカイブパス] と **[ワークパス]** のディレクトリ内のファイル名にタイム・スタンプを付加します。これにより、同じファイル名の繰り返し処理で発生する可能性のある名前の競合を避けることができます。

- この値が空か 0 の場合は、タイム・スタンプが付加されません。
- この設定が 1 の場合は、標準テンプレート '%f_%Q' が付加されます。
- その他の可能性のある値は、"ファイル名に関するタイム・スタンプ指定" を参照してください。

[アーカイブパス]

FTP サーバから受信した各ファイルのコピーを保存するための InterSystems IRIS® サーバ上のディレクトリの完全パス名です。このディレクトリは存在するディレクトリであること、また、ローカル InterSystems IRIS マシンのファイル・システムからアクセス可能なディレクトリであることが必要です。

指定しない場合、InterSystems IRIS は一時保管場所にファイルのローカル・コピーを保管し、処理の完了後に削除します。

注釈　システムからメッセージ本文をパージすると、メッセージ本文で参照していた Archive Path 内のファイルはすべて削除されます。詳細は、"プロダクション・データのパージ" を参照してください。

呼び出し間隔

アダプタの秒単位のポーリング間隔。これは、アダプタが指定された場所で入力ファイルをチェックする時間間隔です。

ポーリング時にアダプタがファイルを検出すると、アダプタはファイルをストリーム・オブジェクトにリンクし、ストリーム・オブジェクトを関連ビジネス・サービスに渡します。一度に複数のファイルを検出すると、アダプタはファイルが検出されなくなるまで、個々のファイルのビジネス・サービスに対して 1 つの要求を送信します。

ビジネス・サービスが各ファイルを同期で処理した場合、ファイルは順次処理されます。ビジネス・サービスが各ファイルをビジネス・プロセスまたはビジネス・オペレーションに非同期で送信した場合、ファイルは同時処理される場合があります。

有効なファイルをすべて処理し終わると、アダプタはポーリング間隔が経過してから、再度ファイルをチェックします。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。

入力間の **[呼び出し間隔]** 期間だけアダプタが遅延するように、ビジネス・サービス内にコールバックを実装できます。詳細は、"プロダクションの開発" の "ビジネス・サービスの定義" を参照してください。

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

Charset

入力ファイルの文字セットを指定します。InterSystems IRIS は、自動的に、文字をこの文字エンコーディングから変換します。この設定値は大文字と小文字が区別されません。`Binary` は、バイナリ・ファイル、新規行文字と改行文字が異なるデータ、または、HL7 バージョン 2 や EDI メッセージのように変更しないまま残す必要のあるデータに対して使用します。テキスト・ドキュメントを転送するときは、他の設定が便利な場合があります。選択肢は以下のとおりです。

- `Binary` ― バイナリ転送 (FTP アダプタのデフォルト)
- `Ascii` ― 文字エンコーディング変換を伴わない Ascii モード FTP 転送
- `Default` ― ローカル InterSystems IRIS サーバのデフォルトの文字エンコード
- `Latin1` ― ISO Latin1 8 ビット・エンコード
- `ISO-8859-1` ― ISO Latin1 8 ビット・エンコード
- `UTF-8` ― Unicode 8 ビット・エンコード
- `UCS2` ― Unicode 16 ビット・エンコード
- `UCS2-BE` ― Unicode 16 ビット・エンコード (ビッグ・エンディアン)
- InterSystems IRIS に NLS (各国言語サポート) をインストールするための、国際文字エンコード規格に基づくその他のエイリアス。
- @TranslationTable、ここで、TranslationTable は、変換テーブルの名前です (この場合は、特定の変換テーブルが使用されます)。

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。

[深刻なエラー]

レコード・マップ・サービスにおいて、個々のレコードでの検証エラーなどのエラーが発生した場合に、システムによるメッセージの処理を停止するかどうかを指定します。アダプタを構成する際は、以下のいずれかのオプションを選択します。

- `Any` ― 既定値。InterSystems IRIS で個々のレコードの保存中にエラーが発生した場合、メッセージの処理を停止します。
- `ParseOnly` ― InterSystems IRIS で 1 つのレコードの保存中にエラーが発生した場合、エラーをログに記録し、そのレコードをスキップしてから、メッセージの解析を続行します。このログには、無効なレコードのストリーム内の位置が含まれます。また、**[エラー時に警告]** が有効になっている場合は、システムによってアラートが生成されます。

[ヘッダ・カウント]

レコード・マップ・サービスにおいて、受信ドキュメントで先頭行としてサービスが無視する行数を指定します。先頭行を無視することで、サービスは、列ヘッダが付いたレポートおよびコンマ区切り値 (CSV) ファイルを解析できるようになります。

[完了を確認]

可能な場合、各ファイルを完全に受信したことを確認します。この設定は、ダウンロードが開始したときにファイルがサーバ上で完全に使用可能になっていないようなケースに対処します。アダプタを構成する際は、以下のいずれかのオプションを選択します。

- [ ] ー ファイルのダウンロードの試行ごとに FTP ディレクトリのリスト要求を実行する必要がないため、小さいファイルのパフォーマンスが最大化されます。
- [ ] ー これがデフォルトです。[ ] は、サーバ・ディレクトリ・リスト内でレポートされるファイル・サイズが増加しない間は、このファイルのデータを読み続けることを意味します。テキスト・モードでは、行フィード文字の挿入または削除によって、ダウンロードに使用されるファイル位置が破損する可能性があるため、このオプションは **[文字セット]** が `Binary` のときにだけ信頼できます。
- [ ] ー サーバからファイルの名前を変更する許可があるまで、ファイルのデータ読み取りを試行し続けます。このオプションが機能するための条件は、FTP サーバがこのアダプタで構成されている**認証情報**を利用して、ダウンロード・ディレクトリに対する名前変更特権を InterSystems IRIS に付与すること、かつ FTP サーバがそのファイルの名前変更の特権を有するようにそのファイル自体のファイル権限が設定されていることです。これらの条件が満たされない場合は、名前変更の試行は常に失敗し、ダウンロードは決して完了しません。
- [ ] ー FTP サーバまたはソース・アプリケーションの動きが遅いときは、[ ] オプションだけでは十分ではありません。サーバが 2 秒おきに 2 回連続して同じファイル・サイズをレポートした場合、InterSystems IRIS はダウンロードが完了したと見なします。したがって、サーバが名前変更をサポートしている場合は [ ] 設定をお勧めします。

プログラムを通じて ConfirmComplete プロパティを操作している場合は、各オプションには 0 ([なし]) ～ 3 ([サイズと名前変更]) のいずれかの整数値が割り当てられます。デフォルトは [サイズ] (整数値 1) です。

[コマンド変換テーブル]

FTP プロトコルの場合、コマンド・チャンネル、特にファイル名/パス名で使用する変換テーブル。通常、FTP サーバで UTF8 をサポートしている場合、アダプタはファイル名/パス名にこれを使用し、サーバで UTF8 をサポートしていない場合、アダプタは RAW モードを使用して送信されたバイトを読み取るだけであるようなケースでは、これは指定しないでください。 サーバは RAW モードでファイル名リストを提供でき、UTF8 をサポートできます。この場合、[コマンド変換テーブル] を RAW に設定し、検出された UTF8 をオーバーライドする必要がある可能性があります。リストに示された値は、内部テーブル名です。

[接続タイムアウト]

FTP サーバへの接続試行を待機する秒数です。デフォルトは 5 秒です。

Credentials

FTP サーバへの接続を承認できるプロダクション認証情報エントリを識別します。プロダクション認証情報の作成方法は、"プロダクションの構成" を参照してください。

[サーバから削除]

真または偽。真の場合、処理が正常終了した後に FTP サーバからファイルを削除します。偽の場合、アダプタは他の何かによって FTP サーバからファイルが削除されるまで、処理済みのファイルを無視します。デフォルトは真です。

この設定が有効な場合に、ファイルの削除が失敗すると、以下のようになります。

- 原因が接続以外の場合、警告が発せられ (例えば、ファイルが読み取り専用である、権限の問題がある、ファイルは既に削除されているなど)、ファイルの削除はスキップされます。
- 接続の問題が原因の場合、再接続と削除が再試行されます。2 回目も削除が失敗した場合は、以下のようになります。

- 原因が接続以外の場合、警告が表示され（例えば、ファイルが読み取り専用である、権限の問題がある、ファイルは既に削除されているなど）、ファイルの削除はスキップされます。
- 接続の問題が原因の場合、ファイルの削除が遅延されます。

後で InterSystems IRIS が FTP サービスに再接続した際に **[サーバから削除]** 設定がまだ有効な場合、このサービスは遅延されている削除でループされ、ファイルの削除が試行されます。削除がまた失敗した場合は、以下のようになります。

- 原因が接続以外の場合、警告が発せられ、ファイルの削除はスキップされます。遅延レコードは削除されます。
- 接続の問題が原因の場合、遅延レコードは維持され、後で接続が確立されたときに再処理されます。

また、遅延されている削除は、システムが即時の正常な削除を処理する場合と同じように、処理の直後であるかのように動作します（日付と時刻の変更の確認なし）。

[外部レジストリ ID]

外部サービス・レジストリ・エントリの ID。外部レジストリを使用しない場合は空白のままにします。レジストリ・エンドポイントで FTPServer プロパティ、FTPPort プロパティ、FilePath プロパティ、および SSLConfig プロパティが設定されます。詳細は、"ESB としてのプロダクションの使用" の "ESB のサービスおよびオペレーションの構成" を参照してください。

[ファイルアクセスタイムアウト]

システムがファイル受領の完了を確認する前に、ソース・アプリケーションからの情報を待機する時間（秒単位）。詳細は、"[完了を確認]" を参照してください。

10 進数値を指定すると、最も近い整数に値が切り上げられます。既定値は 2 です。

File Path

FTP サーバ上でのファイルの検索先ディレクトリの完全パス名です。このディレクトリは実在する必要があり、指定された [認証情報] を使用してアクセスできる必要があります。SubdirectoryLevel が 0 より大きい場合、この設定を空白にすることはできず、検索が開始される最上位ディレクトリを指定する必要があります。

ファイルスペック

FTP サーバの複数ファイル指定から取得するファイルについて、ファイル名またはワイルドカードのファイル指定を、区切り文字で区切って入力できます。[ファイル仕様のデリミタ] 設定で、使用される区切り文字を入力する必要があります。FileSpec を 1 行で入力します。この長さは、最大 2000 文字です。ワイルドカード仕様では、FTP サーバ・マシンのオペレーティング・システムに適した規則を適用します。

[ファイル仕様のデリミタ]

これが空白ではない場合、ファイル仕様設定を複数のファイル名/ワイルドカード検索に分割するための区切り文字として使用されます。

[FTPポート]

接続先の FTP サーバ上の TCP ポートを指定します。デフォルト値は 21 です。

[FTPサーバ]

接続先の FTP サーバを指定します。ここには IP アドレスか、ドメイン・ホスト・コントローラが名前を解決できるのであればサーバ名を指定できます。

[プロトコル]

FTP (ファイル転送プロトコル) と SFTP (SSH ファイル転送プロトコル) のどちらを使用するかを示します。プロトコルが FTP である場合は、**[SSL構成]** 設定を使用して FTP over TLS を構成できます。プロトコルが SFTP である場合は、以下のとおりです。

- [UsePASV] 設定と [ServerListStyle] 設定は無視されます。
- **[FTPポート]** 設定は通常、22 に設定する必要があります。
- **[認証情報]** 設定の値を指定する必要があります。
- [SFTPPublicKeyFile] 設定と [SFTPPrivateKeyFile] 設定の値を指定した場合、アダプタは鍵ペア認証を試行します。アダプタはこの認証を試行する際に、**[認証情報]** 設定で指定されたユーザ名とパスワードも使用し、**[認証情報]** のパスワードを秘密鍵のパスフレーズとして使用します。KeyFile の設定を使用しない場合、アダプタは、**[認証情報]** 設定に基づくユーザ名/パスワード認証のみを試行します。

この設定は、FTP と SFTP のどちらを使用するかを指定します。空白のままの場合は、**[SSL構成]** 設定が [!SFTP] に設定されていなければ、FTP が使用されます。これが設定されている場合は、**[プロトコル]** 設定に関係なく、SFTP が使用されます。

セマフォ仕様

セマフォ仕様では、セマフォとして使用する 2 番目のファイルを作成することで、データ・ファイルの作成完了と読み取り準備の完了を示すことができるようになります。受信 FTP アダプタは、セマフォ・ファイルが存在するまで待機した後、[完了を確認] の要件で指定されたその他の条件をチェックしてから、データ・ファイルを処理します。これにより、データ・ファイルを作成するアプリケーションでは、データ・ファイルの完了までアダプタがそのファイルの処理をしないように待機させることができます。アダプタは、セマフォ・ファイルが存在することのみをテストし、セマフォ・ファイルの内容を読み取ることはありません。

セマフォ仕様が空の文字列の場合、アダプタはセマフォ・ファイルを待機せずに、[完了を確認] の要件で指定された条件が成立すると即座にデータ・ファイルを処理します。セマフォ・ファイルを使用して、アダプタがデータ・ファイルをいつ処理するかを制御する場合は、[完了を確認] フィールドを [なし] に設定することを検討してください。

セマフォ仕様では、データ・ファイルごとに個別のセマフォ・ファイルを指定できます。また、1 つのセマフォ・ファイルで複数のデータ・ファイルを制御することもできます。セマフォ・ファイルとデータ・ファイルのペアにワイルドカードを使用することができます。さらに一連のパターン・マッチング・セマフォ・ファイルをデータ・ファイルに指定することができます。アダプタは、常にデータ・ファイルと同じディレクトリで一致するセマフォ・ファイルを検索します。アダプタがサブディレクトリのデータ・ファイルを検索する場合、セマフォ・ファイルは対応するデータ・ファイルと同じサブディレクトリのレベルに存在する必要があります。

セマフォ仕様を使用する標準的な形式は次のとおりです。

```
[DataFileSpec=] SemaphoreFileSpec [;[DataFileSpec=] SemaphoreFileSpec]...
```

例えば、次のセマフォ仕様があるとします。

```
ABC*.TXT=ABC*.SEM
```

`ABCTest.SEM` セマフォ・ファイルは、アダプタが `ABCTest.TXT` ファイルをいつ処理するかを制御し、`ABCdata.SEM` セマフォ・ファイルは、アダプタが `ABCdata.txt` ファイルをいつ処理するかを制御することを意味します。

注釈　セマフォ仕様では、*(アスタリスク) はドット以外のすべての文字に対応します。ファイル仕様では、アスタリスクはドットを含むすべての文字に対応します。

1 つのセマフォ・ファイルで複数のデータ・ファイルを制御できます。例えば、次のセマフォ仕様があるとします。

```
*.DAT=DATA.SEM
```

DATA.SEM セマフォ・ファイルは、同じディレクトリ内のすべての *.DAT ファイルについてアダプタがいつ処理するかを制御します。アダプタがデータ・ファイルと対応するセマフォ・ファイルを検索するとき、ポーリングの間隔ですべてのデータ・ファイルをループします。上のセマフォ仕様の場合、ABC.DAT ファイルの DATA.SEM から検索を開始し、検出できない場合は、別のファイルのセマフォ・ファイルの検索を続けます。このプロセスで XYZ.DAT の一致を検索していたときに DATA.SEM が作成された場合には、この対応するセマフォ・ファイルが検出されます。ただし、アダプタは XYZ.DAT の処理を次のポーリングまで延期します。これは前のデータ・ファイル ABC.DAT が同じセマフォ・ファイルを待機していたためです。

複数のペアを指定する場合は、それらを ; (セミコロン) で区切ります。例えば、次のセマフォ仕様があるとします。

```
*.TXT=*.SEM;*.DAT=*.READY
```

セマフォ・ファイル MyData.SEM は、アダプタが MyData.TXT をいつ処理するかを制御しますが、セマフォ・ファイル MyData.READY は、アダプタが MyFile.DAT をいつ処理するかを制御します。

アダプタは、セマフォ仕様を左から右に読み取って、各データ・ファイルに対応するセマフォ・ファイルを検出します。対応するセマフォ・ファイルを特定すると、そのファイルのセマフォ仕様の読み取りを停止します。例えば、次のセマフォ仕様があるとします。

```
VIData.DAT=Special.SEM; *.DAT=*.SEM
```

アダプタは、VIData.DAT を処理する前にセマフォ・ファイル Special.SEM を検索しますが、VIData.SEM は VIData.DAT のセマフォ・ファイルと見なされません。stuff.SEM は、stuff.DAT のセマフォ・ファイルと見なされます。これは stuff.DAT が以前の仕様と一致しなかったためです。したがって、同じファイルと一致する複数の仕様を含める場合は、より限定的な仕様を指定してから、より一般化された仕様を指定する必要があります。

データ・ファイルのターゲット・パターンは大小文字が区別され、セマフォ・パターンの大小文字の区別はオペレーティング・システムに依存します。つまり、*.TXT=*.SEM は、大文字の .TXT で終わるターゲット・ファイルのみに適用されますが、オペレーティング・システムは *.SEM と *.sem を区別できない場合があります。オペレーティング・システムで大小文字が区別されない場合、アダプタは、大小文字の任意の組み合わせで終わる *.SEM と *.sem のセマフォ・ファイルを同等に処理しますが、それらは名前が *.TXT のデータ・ファイルのみに対してセマフォとして使用されます。セマフォ・ファイルでは大小文字を区別できませんが、データ・ファイルでは区別が可能です。

1 つのファイル仕様のみを指定し、= (等号) 記号を省略すると、アダプタがこれを全データ・ファイルのセマフォ仕様として扱います。例えば、次のセマフォ仕様があるとします。

```
*.SEM
```

これは、1 つのワイルドカードを = (等号) 記号の左側に指定した場合と等価です。

```
*=*.SEM
```

この場合、セマフォ・ファイル MyFile.SEM はデータ・ファイル MyFile.txt を制御し、セマフォ・ファイル BigData.SEM はデータ・ファイル BigData.DAT を制御します。

セマフォ仕様でワイルドカードが使用されていない場合、その仕様はセマフォ・ファイルの完全な fileSpec です。例えば、次のセマフォ仕様があるとします。

```
*.DAT=DataDone.SEM
```

この場合、DataDone.SEM セマフォ・ファイルは、アダプタが .DAT ファイル拡張子を持つデータ・ファイルをいつ読み取るかを制御します。

セマフォ仕様が指定され、データ・ファイルがいずれのパターンとも一致しない場合、対応するセマフォ・ファイルは存在せず、アダプタはこのデータ・ファイルを処理しません。この状況は、セマフォ仕様で最後のデータ・ファイルとして * を指定することで回避できます。例えば、次のセマフォ仕様があるとします。

```
*.DAT=*.SEM; *.DOC=*.READY; *=SEM.LAST
```

SEM.LAST は、.DAT や .DOC で終了しないすべてのファイルのセマフォ・ファイルです。

FileSpec が * でアダプタが構成されている場合、そのアダプタは、通常はディレクトリ内にあるすべてのファイルをデータ・ファイルと見なします。ただし、アダプタにセマフォ仕様も存在し、特定のファイルをセマフォ・ファイルとして認識する場合、そのファイルがデータ・ファイルとして扱われることはありません。

アダプタは、すべてのデータ・ファイルを特定のポーリング・サイクルで処理し終わると、対応する全セマフォ・ファイルを削除します。

SFTP 認証メソッド

サポートされている AuthenticationMethods は以下のとおりです。

- Empty ー 定義されていれば公開/秘密鍵を使用し、定義されていなければ認証情報にあるユーザ名とパスワードを使用します。
- p ー 認証情報にあるユーザ名とパスワードを使用します。
- k ー 公開/秘密鍵を使用します。
- i ー インタラクティブ (チャレンジ/レスポンス) を使用します。

複数のフラグを指定する場合は、必要な順序で結合します。例えば kp では、公開/秘密鍵認証が最初で、ユーザ名とパスワードがその後に続きます。

SFTP インタラクティブ DTL

指定されている場合、これはインタラクティブ認証 (キーボード・インタラクティブまたはチャレンジ/レスポンス認証とも呼ばれます) の処理に使用される DTL となります。これは、チャレンジ/レスポンス認証の回答の配列を作成する DTL にできます。詳細は "Ens.SSH.InteractiveAuth.DTL" を参照してください。認証情報のパスワードのみを返す場合は空白にします。

SFTP パスフレーズ認証情報

これを使用して、別の認証情報エントリを指定し、そのパスワードを鍵認証パスフレーズとして使用することができます。これが空白の場合、認証情報設定が使用されます。この個別設定により、公開鍵とパスワードの両方の認証が可能になります。

SFTPPrivateKeyFile

SSH 秘密鍵を含むファイルへのファイル・パス。秘密鍵は PEM でエンコードする必要があります。AutheticateWithKeyPair と共に使用する場合は、秘密鍵と公開鍵の両方が必要です。"OpenSSH と PEM でエンコードされたキー" を参照してください。

SFTPPublicKeyFile

SSH 公開鍵を含むファイルへのファイル・パス。公開鍵は OpenSSH 形式である必要があります。AutheticateWithKeyPair と共に使用する場合は、秘密鍵と公開鍵の両方が必要です。"OpenSSH と PEM でエンコードされたキー" を参照してください。

SFTP Server Character Set

リモート・システムでファイル名のエンコードに使用される文字セット。既定値は UTF8 です。このプロパティを "" (空の文字列) に設定すると、ファイル名の文字セット変換は不要であることが示されます。選択した文字セットにより、該当する SSH セッションを表す %Net.SSH.Session オブジェクトの RemoteCharset プロパティが設定されます。

SFTP Local Character Set

ローカル・システムでファイル名のエンコードに使用される文字セット。Windows システムの場合、既定値は "" (空の文字列) です。Unix システムの場合、既定値は UTF8 です。このプロパティを "" (空の文字列) に設定すると、ファイル名

の文字セット変換は不要であることが示されます。選択した文字セットにより、該当する SSH セッションを表す %Net.SSH.Session オブジェクトの LocalCharset プロパティが設定されます。

[サーバリストスタイル]

FTP サーバ上のオペレーティング・システムを指定します。これにより、FTP サーバが返すリスト形式のタイプが決定します。以下のいずれかを選択します。

- UNIX (デフォルト)
- MSDOS

[SSL構成]

この接続の認証に使用する既存の TLS 構成の名前。アダプタから通信が開始されるため、クライアント TLS 構成を選択します。

TLS 構成を作成して管理するには、管理ポータルを使用します。インターシステムズの "TLS ガイド" を参照してください。[SSL/TLS構成を編集] ページの最初のフィールドは [構成名] です。この文字列は [SSL構成] 設定の値として使用します。

[プロトコル] 設定を使用して、SFTP を使用できます。

SSL チェック・サーバ ID

この設定は、TLS を介して FTP サーバに接続する場合に、証明書のサーバ名を、そのサーバへの接続に使用する DNS 名と一致させる必要があるかどうかを指定します。この一致は、RFC 2818 のセクション 3.1 に規定されているルールに基づきます。

SSL セッション再開を使用

この設定は、データ・チャンネルの SSL 接続を確立する際に、そのコマンド・チャンネルのセッション・パラメータを再利用するかどうかを指定します。セッション・パラメータを再利用する機能では、OpenSSL v1.1.x+ が必要です。

接続を維持

値がゼロの場合は、各入力イベントの後、直ちに切断します。値がデフォルトの -1 の場合、アイドル・タイムでも永久的に接続されたままとなります。アダプタは起動時にはアイドルと見なされるため、StayConnected の値が -1 に設定されている場合にのみ自動接続されます。通常のケースで有効な StayConnected の値は 0 と -1 のみです。ただし、StayConnected は正の数値を取ることもできます。StayConnected の時間が CallInterval よりも長い場合、アダプタは常に接続された状態のままとなります。StayConnectedの時間が CallInterval よりも短い場合、アダプタは、CallInterval ごとに切断と再接続を行います。

[サブディレクトリレベル]

ファイルを検索するディレクトリ下のサブディレクトリの階層数。値が 0 より大きい場合、[FilePath] 設定で、検索が開始される最上位ディレクトリを指定する必要があります。

[ファイルストリームを使用]

アダプタで受信したデータのファイル・ストリームを使用するかどうかを指定します。これが偽の場合は、アダプタでグローバル・ストリームが使用されます。この設定に関係なく、[アーカイブパス] または [アーカイブIO] が設定されている場合は、ファイル・ストリームが使用されます。

[UsePASV]

パッシブ FTP モードを使用します。このモードでは、サーバがデータ・ポート・アドレスを返し、クライアントがそれに接続します。制御 TCP 接続とデータ TCP 接続の両方がクライアントから開始されるため、ほとんどのファイアウォールでパッシブ・モード FTP を受け入れやすくなります。

FTP 送信アダプタに関する設定

FTP 送信アダプタ EnsLib.FTP.OutboundAdapter の設定に関する参照情報を提供します。

概要

送信 FTP アダプタには以下の設定があります。

グループ	設定
基本設定	[外部レジストリ ID]、[FTPサーバ]、[FTPポート]、[認証情報]、[ファイルパス]、[ファイル名]
SFTP 設定	[SFTP AppendMode]、[SFTP 認証メソッド]、[SFTP公開鍵ファイル]、[SFTP秘密鍵ファイル]、[SFTP Server Character Set]、[SFTP Local Character Set]、[SFTP パスフレーズ資格情報]、[SFTPInteractiveDTL]、[SFTPSetFileAccessMode]
接続設定	[プロトコル]、[SSL構成]、[SSL でサーバIDを確認する]、[SSL セッション再開を使用]、[UsePASV]、[接続中を維持]、[接続タイムアウト]
追加設定	[上書き]、[文字セット]、[TempFileName]

残りの設定はすべてのビジネス・オペレーションに共通しています。詳細は、"すべてのビジネス・オペレーションに含まれる設定" を参照してください。

Charset

出力ファイルに必要な文字セットを指定します。InterSystems IRIS® は、自動的に、文字をこの文字エンコーディングに変換します。"文字セット" を参照してください。

[接続タイムアウト]

FTP サーバへの接続試行を待機する秒数です。デフォルトは 5 秒です。

Credentials

FTP サーバへの接続を承認できるプロダクション認証情報エントリを識別します。"認証情報" を参照してください。

ファイル名

ドキュメントの出力先のファイルの名前。タイム・スタンプ指定子を含めることができます。%f 指定子が存在する場合、これはドキュメントの元のソース・ファイル名 (ターゲットのファイル名で不正な文字を削除したもの) に置換されます。タイム・スタンプのオプションのドキュメントについては、メソッド Ens.Util.File.CreateTimestamp() を参照してください。

File Path

FTP サーバ上でのファイルの書き込み先ディレクトリの完全パス名です。このディレクトリは実在する必要があり、指定された [認証情報] を使用してアクセスできる必要があります。

[FTPポート]

接続先の FTP サーバ上の TCP ポートを指定します。デフォルト値は 21 です。

[FTPサーバ]

接続先の FTP サーバを指定します。ここには IP アドレスか、ドメイン・ホスト・コントローラが名前を解決できるのであればサーバ名を指定できます。

Overwrite

真または偽。真のときに、出力ファイルがある場合は、これを上書きします。偽のときに、出力ファイルがある場合は、これに追加します。デフォルトは真です。

[プロトコル]

FTP (ファイル転送プロトコル) と SFTP (SSH ファイル転送プロトコル) のどちらを使用するかを示します。プロトコルが FTP である場合は、**[SSL構成]** 設定を使用して FTP over TLS を構成できます。プロトコルが SFTP である場合は、以下のとおりです。

- [UsePASV] 設定と [ServerListStyle] 設定は無視されます。
- **[FTPポート]** 設定は通常、22 に設定する必要があります。
- **[認証情報]** 設定の値を指定する必要があります。
- [SFTPPublicKeyFile] 設定と [SFTPPrivateKeyFile] 設定の値を指定した場合、アダプタは鍵ペア認証を試行します。アダプタはこの認証を試行する際に、**[認証情報]** 設定で指定されたユーザ名とパスワードも使用し、**[認証情報]** のパスワードを秘密鍵のパスフレーズとして使用します。KeyFile の設定を使用しない場合、アダプタは、**[認証情報]** 設定に基づくユーザ名/パスワード認証のみを試行します。

この設定は、FTP と SFTP のどちらを使用するかを指定します。空白のままの場合は、**[SSL構成]** 設定が **[!SFTP]** に設定されていなければ、FTP が使用されます。これが設定されている場合は、**[プロトコル]** 設定に関係なく、SFTP が使用されます。

SFTP Append モード

SFTP サーバの append (追加) を使用するか (サーバ・モード)、クライアントで append (追加) をエミュレートするか (クライアント・モード) を制御します。一部の SFTP サーバは、append (追加) アクセスをサポートしていません。append (追加) をサポートしていないサーバを使用する場合は、**[クライアント・モード]** を選択し、FTP 送信アダプタによって SFTP サーバからのファイルを読み取り、データをファイルに append (追加) し、さらにファイルを SFTP サーバに返送して既存ファイルを上書きします。この設定は、[SSL構成] が **[!SFTP]** に設定され、[上書き] が偽に設定されている場合にのみ効力があります。

append (追加) をエミュレートする影響として、アダプタがファイルを読み取ってから追加されたデータをファイルに書き込む間に、他のプロセスが SFTP サーバの同じファイルを変更した場合、他のプロセスによる更新が消失してしまいます。

注釈　SFTP サーバが append アクセスをサポートしているかどうかわからない場合は、サーバ append テストを使用することができます。これは、**[SFTP Append モード]** ポップアップ・ヘルプの文中にあるリンクによってアクセスできます。

SFTPPrivateKeyFile

SSH 秘密鍵証明書を含むファイルへのファイル・パス秘密鍵は PEM でエンコードする必要があります。AutheticateWithKeyPair と共に使用する場合は、秘密鍵と公開鍵の両方が必要です。"OpenSSH と PEM でエンコードされたキー" を参照してください。

SFTPPublicKeyFile

SSH 公開鍵証明書を含むファイルへのファイル・パス公開鍵は OpenSSH 形式である必要があります。AutheticateWithKeyPair と共に使用する場合は、秘密鍵と公開鍵の両方が必要です。"OpenSSH と PEM でエンコードされたキー" を参照してください。

SFTPSetFileAccessMode

SFTP ファイル・アクセス・モードは、転送時にリモート・システム上のファイルに割り当てるアクセス権を指定します。0600 のように 8 進表現で、または u+rw,g+r のように記号で指定できます。デフォルト値は 0600 です。8 進で指定する場合、4 桁にする必要があります。記号で "すべて" を指定するには、'ugo' を使用します。

SetPermissions 呼び出しでエラーが発生した場合 (ターゲット・ファイルが削除されている場合など)、そのエラーは警告として記録され、これを使用して Put の失敗を示すことはありません。

[SSL構成]

この接続の認証に使用する既存のクライアント TLS 構成の名前。アダプタから通信が開始されるため、サーバ TLS 構成ではなく、クライアント TLS 構成を選択します。"FTP 受信アダプタに関する設定" の "SSL構成" を参照してください。

以前のバージョンとの互換性を確保するため、!SFTP の値により、プロトコルが SFTP であることが示されます。ただし、[プロトコル] 設定を使用して、SFTP を使用することをお勧めします。

SSL チェック・サーバ ID

この設定は、TLS を介して FTP サーバに接続する場合に、証明書のサーバ名を、そのサーバへの接続に使用する DNS 名と一致させる必要があるかどうかを指定します。この一致は、RFC 2818 のセクション 3.1 に規定されているルールに基づきます。

SSL セッション再開を使用

この設定は、データ・チャンネルの SSL 接続を確立する際に、そのコマンド・チャンネルのセッション・パラメータを再利用するかどうかを指定します。セッション・パラメータを再利用する機能では、OpenSSL v1.1.x+ が必要です。

接続を維持

正の値の場合は、アダプタはオペレーションの完了後に、ここで指定された秒数だけリモート・システムとの接続を維持します。値がゼロの場合は、オペレーションが完了するたびに直ちに切断します。値がデフォルトの -1 の場合、アイドル・タイムでも常時接続になります。アダプタは起動時にはアイドルと見なされるため、この値が -1 に設定されている場合にのみ自動接続します。

TempFileName

ドキュメントの出力先の一時ファイルの名前。この名前のファイルの FTP サーバへのアップロードが完了すると、このファイルの名前は [ファイル名] 設定で指定された名前に変更されます。この設定が空白の場合、一時ファイルは使用されません。タイム・スタンプ指定子を含めることができます。%f 指定子が存在する場合、これはドキュメントの元のソース・ファイル名 (ターゲットのファイル名で不正な文字を削除したもの) に置換されます。タイム・スタンプのオプションのドキュメントについては、メソッド Ens.Util.File.CreateTimestamp() を参照してください。

[UsePASV]

パッシブ FTP モードを使用します。このモードでは、サーバがデータ・ポート・アドレスを返し、クライアントがそれに接続します。制御 TCP 接続とデータ TCP 接続の両方がクライアントから開始されるため、ほとんどのファイアウォールでパッシブ・モード FTP を受け入れやすくなります。