



# プロダクション内での HL7 バージョン 2 メッセージのルー ティング

Version 2023.1  
2024-01-02

プロダクション内での HL7 バージョン 2 メッセージのルーティング  
InterSystems IRIS Data Platform Version 2023.1 2024-01-02  
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 概要	1
1.1 インターシステムズにおける HL7 バージョン 2 のサポート	1
1.2 HL7 バージョン 2 ルーティング・プロダクション	1
1.3 HL7 の実装ツール	3
2 HL7 スキーマと利用可能なツール	5
2.1 HL7 スキーマと HL7 メッセージの概要	5
2.2 HL7 スキーマ構造ページの使用方法	6
2.2.1 ドキュメント・タイプのリストの表示	6
2.2.2 メッセージ構造の表示	6
2.2.3 セグメント構造の表示	7
2.2.4 データ構造の表示	8
2.2.5 コード・テーブルの表示	9
2.3 カスタム・スキーマ・エディタの使用方法	10
2.3.1 新しいカスタム・スキーマの作成	11
2.3.2 新しいセグメントの定義	12
2.3.3 新しいメッセージ・タイプと構造タイプの定義	14
2.3.4 データ構造とコード・テーブルの編集	17
2.4 HL7 メッセージ・ビューワ・ページの使用方法	17
2.4.1 オプションの選択	17
2.4.2 メッセージの解析	18
2.4.3 変換のテスト	20
2.5 バッチ・メッセージの表示	21
2.6 HL7 クラス	22
2.6.1 HL7 メッセージ・クラスの詳細	24
3 プロダクションの構成	25
3.1 新しい HL7 ルーティング・プロダクションの作成	25
3.2 HL7 ビジネス・サービスの追加	26
3.2.1 HL7 ビジネス・サービスの作成	26
3.2.2 HL7 ビジネス・サービスの統合と構成	27
3.3 HL7 ルーティング・プロセスの追加	27
3.3.1 HL7 ルーティング・プロセスの作成	27
3.3.2 HL7 ルーティング・プロセスの統合と構成	28
3.4 HL7 シーケンス・マネージャの追加	28
3.4.1 HL7 シーケンス・マネージャの作成	28
3.4.2 HL7 シーケンス・マネージャの統合と構成	29
3.4.3 プログラムによる HL7 シーケンス・データへのアクセス	29
3.5 HL7 ビジネス・オペレーションの追加	30
3.5.1 HL7 ビジネス・オペレーションの作成	30
3.5.2 HL7 ビジネス・オペレーションの統合と構成	31
3.6 携帯電話アラートと電子メール・アラート	31
4 HL7 用のルーティング・ルール・セットの定義	33
5 HL7 用の DTL データ変換の定義	37
5.1 長いセグメント・フィールドの変換	39
5.2 NULL マッピング・コード	40
6 HL7 検索テーブルの定義	43

6.1 デフォルトでインデックスが作成されるプロパティ .....	43
6.2 例 .....	44
7 HL7 の重要なシナリオ .....	47
7.1 HL7 確認応答 (ACK) モード .....	47
7.2 HL7 二重確認応答シーケンス .....	49
7.2.1 受信メッセージの二重 ACK シーケンス .....	49
7.2.2 送信メッセージの二重 ACK シーケンス .....	50
7.2.3 二重 ACK シーケンスの構成 .....	51
7.3 HL7 バッチ・メッセージ .....	52
7.3.1 サポートされているバッチ形式 .....	52
7.3.2 受信バッチ・ドキュメントの処理 .....	53
7.3.3 バッチ・メッセージの送信 .....	54
7.3.4 バッチ・モード .....	54
7.3.5 カスタム送信バッチ処理 .....	54
参照 .....	57
HL7 ビジネス・サービスに関する設定 .....	58
HL7 ルーティング・プロセスに関する設定 .....	65
HL7 シーケンス・マネージャに関する設定 .....	70
HL7 ビジネス・オペレーションに関する設定 .....	74
HL7 のエスケープ・シーケンス .....	78

# 1

## 概要

このドキュメントでは、インターシステムズ製品をルーティング・エンジンとして使用してアプリケーション間で HL7 バージョン 2 メッセージをルーティングする方法について説明します。

### 1.1 インターシステムズにおける HL7 バージョン 2 のサポート

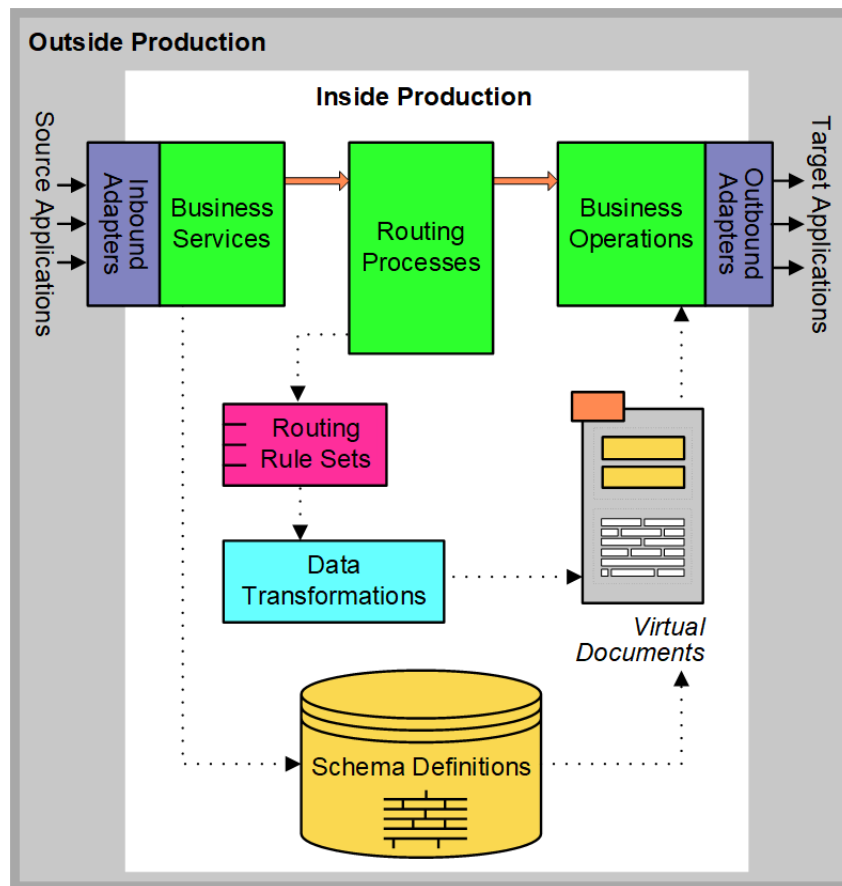
インターシステムズ製品は HL7 バージョン 2 メッセージを仮想ドキュメントとしてサポートしています。仮想ドキュメントはメッセージの一種で、インターシステムズ製品では部分的にしか解釈されません。この種のメッセージは、標準のプロダクション・メッセージ・ヘッダと標準のメッセージ・プロパティ (ID、Priority、SessionId など) で構成されています。しかし、メッセージ内のデータはメッセージ・プロパティとして提供されず、代わりに、処理速度を向上させるために内部使用のグローバルに直接格納されます。

インターシステムズが提供する各種のツールでは、仮想ドキュメント内の値にアクセスして、データ変換、ビジネス・ルール、およびメッセージの検索とフィルタリングに使用できます。基礎的な情報は、[“プロダクション内での仮想ドキュメントの使用法”](#)を参照してください。

HL7 セグメント名はすべて大文字にする必要があります。

### 1.2 HL7 バージョン 2 ルーティング・プロダクション

次の図は、HL7 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プロダクション内における HL7 バージョン 2 メッセージのフローを示しています。構成項目によって参照されるがそれ自体は構成項目ではない要素が表示されています。これらの要素には、ルーティング・ルール・セット、データ変換、仮想ドキュメント、スキーマ定義などがあります。このドキュメントに記載されている手順に従って、これらの項目を作成します。



プロダクション内における HL7 バージョン 2 メッセージのフローは次のとおりです。

1. HL7 ビジネス・サービスは、メッセージを受け取るように構成されている特定のソース・アプリケーションから受信メッセージを受け取ります。
2. ビジネス・サービスは、そのメッセージを特定の HL7 ルーティング・プロセスに渡します。これは、HL7 ビジネス・サービスからの受信メッセージを特定の HL7 ビジネス・オペレーション経由でプロダクションの外部に配信するように準備するビジネス・プロセスです。
3. ルーティング・プロセスでは、必要とする HL7 スキーマ定義に対してメッセージを検証することができます。これは、標準の HL7 スキーマである場合も、カスタム・スキーマである場合もあります。

(表示なし) 検証に失敗すると、HL7 ルーティング・プロセスは、メッセージを構成済みの不正メッセージ・ハンドラに渡します。これは、検証に失敗した受信 HL7 メッセージを破棄する HL7 ビジネス・オペレーションです。この場合、メッセージは通常、ファイルに保存されます。エラーをイベント・ログに記録したり、オペレータに警告したりすることもあります。

4. HL7 ルーティング・プロセスは、ルーティング・ルール・セットをメッセージに適用します。ルーティング・ルール・セットは 1 つまたは複数のターゲット・ビジネス・オペレーションを選択し、ターゲット・アプリケーション用にメッセージを準備するためにデータ変換が必要であれば適用します。
5. 通常、ターゲット用にメッセージを準備するには、何らかのデータ変換が必要です。ルーティング・ルール・セットでは、カスタム・コーディングされた変換を呼び出すこともできますが、変換は一般的に、データ変換言語 (DTL) を使用して作成されます。DTL は、ユーティリティ関数を呼び出すことも、より複雑な計算を実行する独自のクラス・メソッドを呼び出すこともできます。
6. 送信メッセージの準備ができたなら、ルーティング・プロセスはメッセージを HL7 ビジネス・オペレーションに渡します。ビジネス・オペレーションは、HL7 メッセージをターゲット・アプリケーションに送信するために必要なアドレスおよびフレーミング情報を提供します。

(表示なし) デフォルトでは、プロダクションを通過した HL7 メッセージはすべて、必要な期間、プロダクションのメッセージ・ウェアハウスに保持されます。メッセージ・ウェアハウス内にある間、[HL7 メッセージ・ビューワ](#)、メッセージ・ブラウザ、ビジュアル・トレースなどの管理ポータル機能を使用するか、SQL クエリを発行して、HL7 メッセージのコンテンツを追跡したり、参照したりすることができます。古いメッセージを自動的にパージするか、管理者の判断でパージするようにプロダクションを構成できます。

## 1.3 HL7 の実装ツール

HL7 プロダクションとカスタム・スキーマの実装を迅速化するために設計されたツールの詳細は、“[HL7 の生産性ツール](#)”を参照してください。





# 2

## HL7 スキーマと利用可能なツール

この章では、HL7 バージョン 2 スキーマと HL7 バージョン 2 ドキュメントの操作に使用可能なインターシステムズ製ツールの概要を示します。

### 2.1 HL7 スキーマと HL7 メッセージの概要

インターシステムズ製品では、スキーマを使用して解析しなくても、HL7 メッセージを処理して渡すことができますが、スキーマをメッセージに関連付けると、次のことが可能になります。

- ・ メッセージを解析し、以下でフィールド値にアクセスできます。
  - データ変換
  - ルーティング・ルール
  - カスタム ObjectScript コード
- ・ メッセージがスキーマに適合しているかどうかを検証できます。

各 HL7 メッセージは、MSH セグメントの `MessageType` フィールド (MSH:9) で指定されたメッセージ・タイプによって識別されます。一部のメッセージ・タイプは同じメッセージ構造を共有しています。例えば、HL7 バージョン 2.3.1 では、患者を事前登録する ADT\_A05 メッセージの構造は、ADT\_A01 入院メッセージと同じです。スキーマでは、ADT\_A05 メッセージは構造タイプ ADT\_A01 を使用するように指定されています。

インターシステムズ製品で HL7 メッセージを解析するには、次の 2 つの情報が必要です。

- ・ スキーマ・カテゴリ - 2.3.1 や 2.7 など、HL7 のバージョン番号です。インターシステムズ製品で定義したカスタム・スキーマのカテゴリである場合もあります。プロダクションは、ビジネス・サービスの [メッセージ・スキーマ・カテゴリ] 設定、またはデータ変換の設定からスキーマ・カテゴリを取得します。HL7 メッセージでは、MSH セグメントの `VersionID` フィールド (MSH:12) にスキーマ・バージョン番号が含まれますが、多くのアプリケーションではこのフィールドが一貫した方法で設定されないため、インターシステムズではこの値を使用しません。
- ・ 構造タイプ - プロダクションは MSH:9 フィールドからメッセージ・タイプを取得した後、スキーマ定義をチェックして、そのメッセージの構造タイプを取得します。

インターシステムズでは、MSH:9.3 サブフィールドを使用してメッセージ・タイプを修飾することもあります。HL7 メッセージでは、MSH:9.3 サブフィールドは、1) メッセージ・タイプの修飾子として使用されるか、2) 構造タイプを指定するために使用されます。MSH:9.3 がメッセージ・タイプを修飾する場合 (通常は数字として修飾する)、インターシステムズ製品はその値をメッセージ・タイプの一部として組み込みます。MSH:9.3 が構造タイプ (ADT\_A01 など) を指定する場合、インターシステムズでは、メッセージ・タイプを特定する際にも **Name** プロパティを設定する際にもそ

の値を無視します。インターシステムズ製品はスキーマから構造タイプを取得するため、構造タイプを特定するために MSH:9.3 サブフィールドは必要ありません。

ビジネス・サービスまたはデータ変換によって、HL7 メッセージを格納する `EnsLib.HL7.Message` オブジェクトが作成される際、スキーマ・カテゴリと構造タイプが結合され、次の構文を使用して **DocType** プロパティに格納されます。

```
category:structureType
```

例えば、カテゴリ 2.3.1 の有効な DocType 値には、2.3.1:ACK、2.3.1:ADT\_A17、2.3.1:BAR\_P01、2.3.1:PEX\_P07 などがあります。メッセージ・タイプは、構造タイプとは異なる場合があり、**Name** プロパティに格納されます。

ObjectScript コードで `EnsLib.HL7.Message` オブジェクトを作成する場合は、MSH:9 フィールドの値に基づいて **DocType** プロパティと **Name** プロパティを設定する必要があります。

HL7 標準では、末尾の Z セグメントなど、ローカル拡張が許可されています。これらのセグメントは、ベース・スキーマ・カテゴリでは定義されていません。データ変換、ルーティング・ルール、または ObjectScript でカスタム Z セグメント内のフィールドにアクセスするには、拡張されたメッセージを指定するカスタム・スキーマ・カテゴリを定義する必要があります。カスタム・スキーマの定義の詳細は、“[カスタム・スキーマ・エディタの使用法](#)”を参照してください。

カスタム HL7 スキーマをソース・コントロールに配置できます。ソース・コントロールを有効にすると、[HL7 スキーマ・メッセージ・タイプ] ページなど、カスタム・スキーマの操作に使用する他のページではなく、メインの [HL7 スキーマ] ページからチェック・アウトやチェック・インなどのソース・コントロール・オプションを使用できるようになります。他のページを使用してスキーマを編集する前に、[HL7 スキーマ] ページからスキーマ全体をチェック・アウトする必要があります。関連するすべての HL7 スキーマ・ページからソース・コントロール・アクティビティを収集するメインの [HL7 スキーマ] ページから [ソース・コントロール出力] ウィンドウを使用できます。ソース・コントロールの設定については、“[InterSystems IRIS とソース・コントロール・システムの統合](#)”を参照してください。

## 2.2 HL7 スキーマ構造ページの使用法

[HL7 スキーマ] ページを使用すると、HL7 バージョン 2 スキーマ仕様のインポートと表示を行うことができます。このページを表示するには、ホーム・ページから [Interoperability] → [相互運用] → [HL7 v2.x] → [HL7 v2.x スキーマ構造] を選択します。このページの使用に関する一般情報は、“[プロダクション内での仮想ドキュメントの使用法](#)”の“[スキーマ構造ページの使用法](#)”を参照してください。

[HL7 スキーマ] ページには、[メッセージ・タイプ] という追加タブがあります。このタブでは、2 つのメッセージ構造が要求/応答ペアとして識別されます。

カスタム・スキーマ・カテゴリの作成と編集については、“[カスタム・スキーマ・エディタの使用法](#)”を参照してください。

### 2.2.1 ドキュメント・タイプのリストの表示

カテゴリ内のすべてのドキュメント・タイプ構造をリストするには、最初にカテゴリを選択した後、[DocType 構造] タブをクリックします。


### 2.2.2 メッセージ構造の表示

メッセージ構造の内部編成を表示するには、[HL7 スキーマ] ページ ([Interoperability] → [相互運用] → [HL7 v2.x] → [HL7 v2.x スキーマ構造] を選択) の [DocType 構造] タブで、そのメッセージ構造の名前をクリックします。次に示す表現上の規則と名前付け規約を使用して、メッセージのセグメント構造が [構造] セクションに表示されます。

- ・ メッセージ構造を構成するセグメントは、上から下へ順番に表示されます。
- ・ セグメント名はすべて大文字にする必要があります。

- 各メッセージ・セグメントの 3 文字の名前が表示されます (MSH、NTE、PID など)。この名前は、HL7 メッセージ構造内のこの場所に存在するセグメントのタイプを示します。選択を囲むセグメント、繰り返されるセグメント、または他のセグメントのグループを含むセグメントでそれぞれの名前には、特別な文字が追加されます。
- 緑色の点線で囲まれているセグメント、グループ、またはフィールドはオプションです。
- 繰り返すことができるセグメントの名前には括弧が追加されます。例えば、PID セグメントを繰り返すことができる場合、PID( ) が表示されます。
- 他のセグメントの選択を囲むセグメントは、セグメントの和集合と見なされます。このような和集合セグメントの名前には、“union” という単語が追加されます。和集合に含まれるセグメントの 1 つのみが、メッセージ構造内のこの場所に存在できます。
- セグメントのグループを含むセグメントの名前には、“grp” という文字が追加されます。グループを展開したり折りたたんだりするには、グループの名前の横にある矢印アイコンを使用します。

セグメント名をダブルクリックすると、そのセグメントの構造が別のウィンドウで開きます。

未加工のテキスト形式でメッセージ構造を表示するには、 をクリックします。

## 2.2.3 セグメント構造の表示

メッセージ・セグメントの構造を表示するには、[前の節](#)に示されている例と類似する任意のページで、そのメッセージ・セグメントの名前をクリックします。そのセグメント内のすべてのフィールドをリストするテーブルがインターシステムズ製品により表示されます。これは [HL7 スキーマ・セグメント構造] ページです。

例えば、2.3:ADT\_A01 メッセージ構造内の [PR1] セグメントをクリックすると、次のページが表示されます。

Schema Category: [2.3](#)  
 Message Structure: [ADT\\_A01](#)  
 Segment Structure: **PR1**

Path you followed to get to this Segment Structure: **PR1grp().PR1**

### 2.3:PR1 Procedures

Field	Description	Property Name	Data Structure	Symbol	Repeat Count	Minimum Length	Maximum Length	Required	Repeating	Code Table	Alternate Description
1	Set ID - Procedure	<a href="#">SetIDProcedure</a>	2.3:SI	! (exactly one required)			4	R	0		
2	Procedure Coding Method	<a href="#">ProcedureCodingMethod</a>	2.3:IS	! (exactly one required)			2	R	0	<a href="#">2.3:89</a>	
3	Procedure Code	<a href="#">ProcedureCode</a>	2.3:CE				80	O	0	<a href="#">2.3:88</a>	
4	Procedure Description	<a href="#">ProcedureDescription</a>	2.3:ST				40	O	0		
5	Procedure Date/Time	<a href="#">ProcedureDateTime</a>	2.3:TS				26		0		
6	Procedure Type	<a href="#">ProcedureType</a>	2.3:ID	! (exactly one required)			2	R	0	<a href="#">2.3:230</a>	
7	Procedure Minutes	<a href="#">ProcedureMinutes</a>	2.3:NМ				4	O	0		
8	Anesthesiologist	<a href="#">Anesthesiologist</a>	2.3:XCN	* (zero or more)			120	O	1	<a href="#">2.3:10</a>	
9	Anesthesia Code	<a href="#">AnesthesiaCode</a>	2.3:IS				2	O	0	<a href="#">2.3:19</a>	
10	Anesthesia Minutes	<a href="#">AnesthesiaMinutes</a>	2.3:NМ				4	O	0		
11	Surgeon	<a href="#">Surgeon</a>	2.3:XCN	* (zero or more)			120	O	1	<a href="#">2.3:10</a>	
12	Procedure Practitioner	<a href="#">ProcedurePractitioner</a>	2.3:XCN	* (zero or more)			230	O	1	<a href="#">2.3:10</a>	
13	Consent Code	<a href="#">ConsentCode</a>	2.3:CE				60	O	0	<a href="#">2.3:59</a>	
14	Procedure Priority	<a href="#">ProcedurePriority</a>	2.3:NМ				2	O	0		
15	Associated Diagnosis Code	<a href="#">AssociatedDiagnosisCode</a>	2.3:CE				80	O	0		

列は以下のとおりです。

- ・ **【フィールド】** – セグメント内のフィールドにアクセスするために使用する番号 (番号を選ぶ場合)。
- ・ **【説明】** – フィールドの簡単な説明。
- ・ **【プロパティ名】** – セグメント内のフィールドにアクセスするために使用する名前 (名前を選ぶ場合)。
- ・ **【データ構造】** – データ構造を使用する複雑なフィールド値については、segment:field 仮想プロパティ・パスを入力する前に、さらに構文の詳細が必要です。これを取得できますが、そのためには、この列で名前をクリックします。
- ・ **【記号】** – フィールドの構文規則を示します。この列の文字は、そのフィールドがメッセージ・セグメント内に存在する、存在しない、またはメッセージ・セグメント内で繰り返されると考えることができるかどうかを示します。可能な値は以下のとおりです。

記号	意味
!	(1 回のみ) フィールドは必須で、1 回のみ出現する必要があります。
?	(0 または 1 回) フィールドはオプションですが、出現する場合は、1 回のみ出現することができます。
+	(1 回以上) フィールドは 1 回以上繰り返すことができます。
*	(0 回以上) フィールドは 0 回以上繰り返すことができます。
&	特定の条件においてのみ、フィールドは存在することも、繰り返すこともできます。
n*	(0 ~ n 回) フィールドは最大 n 回繰り返されます。

- ・ **【繰り返し回数】** – フィールドを繰り返すことができる最大回数 (繰り返す場合、および上限がある場合)。
- ・ **【最小長】** – フィールド内の最小文字数。フィールドの繰り返しごとに、この文字数が含まれている必要があります。
- ・ **【最大長】** – フィールド内の最大文字数。フィールドの繰り返しごとに、この文字数を含めることができます。
- ・ **【必須】** – 必須の場合は R、オプションの場合は O が表示されます。
- ・ **【繰り返し】** – 真の場合は 1、偽の場合は 0 が表示されます。
- ・ **【コードテーブル】** – エントリをクリックすると、そのフィールドに入力できる有効なコードを参照します。
- ・ **【代替説明】** – フィールドの追加の詳しい説明。

この情報、特に**【プロパティ名】**列を使用して、segment:field という形式で仮想プロパティ・パスを作成できます。以下は、2.3:ADT\_A01 メッセージ構造内の PR1 セグメントにある単純な field 値を含む仮想プロパティ・パスの例です。() ショートカット構文は、繰り返しフィールドで使用可能なすべてのインスタンスを示します。(1) は、最初のインスタンスを示します。

```
PR1grp().PR1:ProcedureType
PR1grp().PR1:ProcedureCode()
PR1grp().PR1:ProcedureCode(1)
PR1grp().PR1:ProcedureCode(x)
PR1grp().PR1:ProcedurePriority
```

## 2.2.4 データ構造の表示

**【データ構造】**列で名前をクリックすると、そのデータ構造内のすべてのフィールドが表示されます。これは**【HL7データ構造】**ページです。最も有用な列は次のとおりです。

- ・ **【コンポーネント】**列には、セグメント内のフィールドにアクセスするために使用できる番号がリストされます (番号を選ぶ場合)。

- ・ **[プロパティ名]** 列には、セグメント内のフィールドにアクセスするために使用できる名前がリストされます (名前を選ぶ場合)。
- ・ **[データ構造]** 列のエントリ (存在する場合) をクリックすると、詳細がドリル・ダウンされます。
- ・ **[コードテーブル]** 列のエントリ (存在する場合) をクリックすると、そのフィールドに入力できる有効なコードが表示されます。

上のセグメント構造ページで 2.3:XCN という **[データ構造]** 項目をクリックすると、次に示すサンプル・ページが表示されます。このページは、カテゴリ 2.3 のデータ構造 XCN が“拡張複合 ID 番号と名前”を表し、14 個のフィールドで構成されることを示しています。単純な値もあれば、データ構造やコードもあります。

Schema Category: [2.3](#)  
 Message Structure: [ADT\\_A01](#)  
 Segment Structure: [PR1](#)  
 Field Number: **8**  
 Data Structure: **XCN**

Path you followed to get to this Data Structure: **PR1grp().PR1:Anesthesiologist()**

2.3:XCN Extended Composite ID number and name (2.8.46)

Component	Description	Property Name	Data Structure	Minimum Length	Maximum Length	Required	Code Table	Alternate Description
1	ID number (ST)	IDnumberST	2.3:ST					
2	family name	familyname	2.3:ST					
3	given name	givenname	2.3:ST					
4	middle initial or name	middleinitialorname	2.3:ST					
5	suffix (e.g., JR or III)	suffix	2.3:ST					
6	prefix (e.g., DR)	prefix	2.3:ST					
7	degree (e.g., MD)	degree	2.3:ST					
8	source table	sourcetable	2.3:ID				2.3:297	
9	assigning authority	assigningauthority	2.3:HD					
10	name type	nametype	2.3:ID				2.3:200	
11	identifier check digit	identifiercheckdigit	2.3:ST					
12	code identifying the check digit scheme employed	codeidentifyingthecheckdigit	2.3:ID				2.3:61	
13	identifier type code	identifiertypecode	2.3:IS				2.3:203	
14	assigning facility ID	assigningfacilityID	2.3:HD					

この情報を指定すると、メッセージ構造 2.3:ADT\_A01 内の複雑な PR1grp().PR1:Surgeon フィールドの仮想プロパティ・パスを次のように作成できます。

```
PR1grp().PR1:Surgeon.familyname
PR1grp().PR1:Surgeon.degree
```

## 2.2.5 コード・テーブルの表示

**[コードテーブル]** 列で名前をクリックすると、そのフィールドの有効なコードと説明のリストが表示されます。これは **[HL7 コード・テーブル]** ページです。[前の節](#)に示されているデータ構造ページで 2.3:200 という **[コードテーブル]** 項目をクリックすると、次に示すサンプル・ページが表示されます。

Schema Category: 2.3

Code Table: 200

Description: Name Type

Type: 2 (HL7)

Code	Meaning
L	Legal Name
O	Other
M	Maiden Name
A	Alias Name
C	Adopted Name
D	Display Name

上の例は、カテゴリ 2.3 のコード・テーブル 200 が“名前タイプ”を表し、値 L、O、M、A、C、または D を持つことができることを示しています。

これは、DocType が 2.3:ADT\_A01 の HL7 メッセージがある場合、そのメッセージには PR1grp().PR1:Anesthesiologist.nametype というパスのオプションの仮想プロパティがあり、L、O、M、A、C、または D のいずれかの値を指定できることを意味します。

注釈 詳細は、“プロダクション内での仮想ドキュメントの使用法”の[構文ガイド](#)を参照してください。

## 2.3 カスタム・スキーマ・エディタの使用法

カスタム・スキーマ・エディタを使用すると、新しいカスタム HL7 スキーマを作成したり、既存のカスタム HL7 スキーマを編集したりすることができます。カスタム・スキーマには通常、標準スキーマまたは別のカスタム・スキーマであるベース・スキーマがあります。インターシステムズ製品でカスタム・スキーマを使用してメッセージを解析する際、メッセージ・タイプ、セグメント、または他の要素がカスタム・スキーマで定義されていない場合、ベース・スキーマの定義が使用されます。そのため、カスタム・スキーマでは、ベース・スキーマに存在しない要素やベース・スキーマのものとは異なる定義を必要とする要素を定義するだけでかまいません。標準スキーマを編集することはできません。

カスタム・スキーマを定義する最も一般的な理由は、末尾の Z セグメントがある HL7 メッセージを解析できることです。インターシステムズ製品では、スキーマで定義されていない末尾の Z セグメントがあるメッセージを処理できますが、次のことを行うには、カスタム・スキーマを定義する必要があります。

- ・ ルーティング・ルール、データ変換、または ObjectScript コードで末尾の Z セグメントのフィールド・パスにアクセスします。
- ・ 末尾の Z セグメントを検証します。

現在、標準スキーマを使用しているプロダクションがあり、データ変換またはルーティング・ルールで末尾の Z セグメントのフィールド・パスにアクセスする必要がある場合は、次の作業が必要です。

1. 管理ポータルのカスタム・スキーマ・エディタを使用して、新しい HL7 スキーマを作成します。カスタム・スキーマの名前を入力し、ベース・スキーマを指定します。“[新しいカスタム・スキーマの作成](#)”を参照してください。



2. メッセージに含めることができる Z セグメントを定義します。ベース・スキーマの既存のセグメントと同様のフィールドが Z セグメントにある場合、ベースから定義をコピーし、必要に応じて変更することができます。それ以外の場合は、新しいセグメントを作成できます。フィールドの追加、フィールドの削除、フィールドの順序の変更が可能です。“[新しいセグメントの定義](#)”を参照してください。
3. 末尾の Z セグメントを含む各メッセージ・タイプについて、基になるスキーマからコピーしたメッセージ・タイプと構造タイプをカスタム・スキーマに作成します。構造タイプの最後に Z セグメントを追加します。“[新しいメッセージ・タイプと構造タイプの定義](#)”を参照してください。
4. ベース・スキーマの代わりに新しいカスタム・スキーマを使用するようにプロダクションのビジネス・サービスを変更します。
5. 末尾の Z セグメントがある新しいメッセージをプロダクションのビジネス・サービスに渡して、プロダクションをテストします。メッセージ・ビューでメッセージを表示したときに、Z セグメントがスキーマで定義されていれば、青で表示されます。認識できないセグメントは黒で表示されます。

詳細な手順については、この後の各節で説明します。

### 2.3.1 新しいカスタム・スキーマの作成

管理ポータルからカスタム・スキーマ・エディタを起動するには、ホーム・ページから [Interoperability]→[相互運用]→[HL7 v2.x]→[HL7 v2.x スキーマ構造] を選択します。

新しい HL7 スキーマを作成するには、[新規作成] をクリックします。カスタム・スキーマ・エディタで、ベース・スキーマ、スキーマ名、およびスキーマの説明（オプション）を選択します。例えば、バージョン 2.5 の標準スキーマ・カテゴリに基づいてカスタム・スキーマを定義する場合、次のように入力できます。

カスタム・スキーマを作成すると、カスタム・スキーマ・エディタに空のスキーマが表示され、メッセージ・タイプ、構造タイプ、セグメント構造、データ構造、およびコード・テーブルを定義できます。ベース・スキーマのものとは定義が異なる要素やベース・スキーマで定義されていない要素を定義するだけでかまいません。

カスタム HL7 スキーマ・エディタには次のタブがあります。

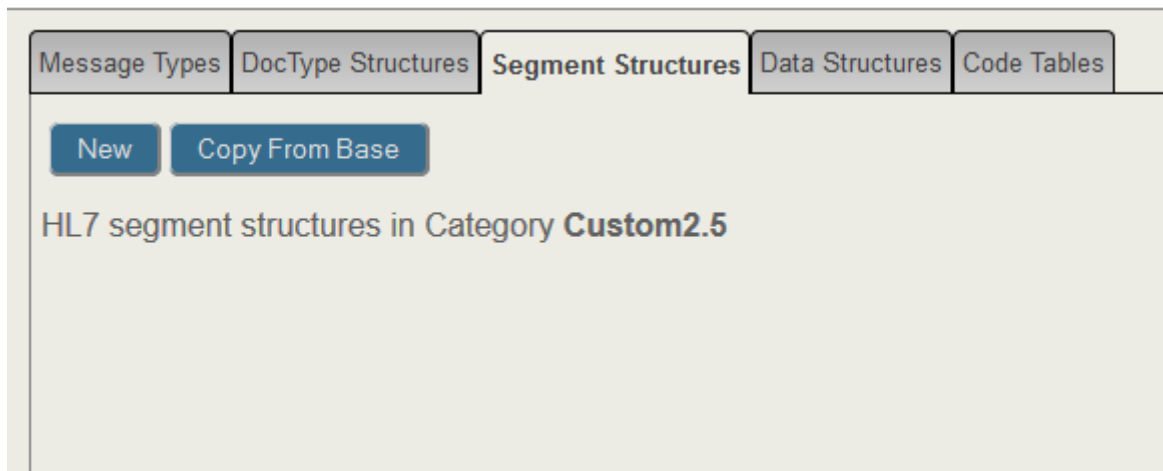
- ・ **メッセージ・タイプ**
- ・ DocType 構造
- ・ セグメント構造
- ・ データ構造
- ・ コードテーブル

これらの各タブで、[ベースからコピー]を選択して、既存のベース定義から要素をコピーできます。これにより、共通する定義を再入力することなく、既存の定義の拡張であるメッセージ・タイプやその他の要素を作成することができます。

## 2.3.2 新しいセグメントの定義

新しいセグメントを定義するには、左側のパネルで新しいカスタム・スキーマを選択した後、[セグメント構造]タブを選択します。カスタム・スキーマの場合、[セグメント構造]タブには[新規作成]ボタンと[ベースからコピー]ボタンがあり、カスタム・スキーマで現在定義されているセグメントのリストが表示されます。このスキーマにはセグメントが定義されていないため、セグメント構造のリストは空です。標準スキーマを表示している場合、新しいセグメントを追加することはできず、ボタンは表示されません。





別のセグメントからフィールドをコピーせずに新しいセグメントを定義する場合は、[新規作成] ボタンをクリックすると、空のセグメントが作成されます。セグメントの名前を指定し、[フィールドの追加] ボタンをクリックします。空のフィールドが作成され、フォームに入力できます。次に、例を示します。

Custom Schema Category Custom2.5  
Base Schema Category 2.5

**New Segment Structure Name**  
ZZR

**Description**

**Segment Fields**

Field	Description	Property Name	Data Structure	Symbol	Count	Min Length	Max Length	Required	Repeating	Code Table	Alternative Description
1								O	No		

**Add Field**

ベース・スキーマの既存のセグメントとよく似たセグメントを作成する場合は、[ベースからコピー] を選択する必要があります。これにより、指定したセグメントと同じフィールドを持つ新しいセグメントが作成されます。カスタム・セグメント構造ウィザードには、コピーされたフィールドが表示され、それらのフィールドの後には新しいフィールドを作成するための [新規作成] ボタンがあります。例えば、次のセグメントは PID セグメントからコピーされたものです。

Custom Schema Category Custom2.5  
Base Schema Category 2.5

**Segment Structure to Copy**  
PID (2.5)

**New Segment Structure Name**  
ZPI

**Description**  
Patient Identification

**Segment Fields**

Field	Description	Property Name	Data Structure	Symbol	Count	Min Length	Max Length	Required	Repeating	Code Table	Alternative Description
1	Set ID - PID	SetIDPID	SI				4	O	No		
2	Patient ID	PatientID	CX				20	B	No		
3	Patient Identifier List	PatientIdentifierList	CX	+			250	R	Yes		
4	Alternate Patient ID - PID	AlternatePatientIDPID	CX	*			20	B	Yes		
5	Patient Name	PatientName	XP	+			250	R	Yes		

新しいセグメントとしてまたはベース・セグメントのコピーとしてセグメントを作成したら、次のようにフィールドを追加または更新できます。

- ・ **[フィールドの追加]** ボタンをクリックして、セグメントの最後にフィールドを追加します。
- ・ セグメント・フィールドを定義するフォームのテキスト・ボックスを更新します。**[プロパティ名]** テキスト・ボックスを編集することはできません。プロパティ名は、**[説明]** フィールドの値からスペースや特殊記号を削除して設定されます。**[OK]** をクリックしてウィザードを終了すると、プロパティ名が設定されます。
- ・ 上下の矢印をクリックして、フィールドの順序を変更します。
- ・ 赤の X をクリックして、フィールドをセグメントから削除します。

フィールドの入力が完了したら、**[OK]** をクリックしてセグメントを保存します。

カスタム・スキーマの保存済みセグメントを編集できますが、そのためには、セグメント名をクリックした後、**[編集]** をクリックします。

Z セグメントを定義したら、Z セグメントを含むメッセージ・タイプと構造タイプを定義する必要があります。

### 2.3.3 新しいメッセージ・タイプと構造タイプの定義

メッセージ・タイプはメッセージを識別し、HL7 MSH:9 フィールドの値と一致します。メッセージ・タイプを定義する際には、送信メッセージ構造タイプ (メッセージ・タイプと同じである場合がある) と返りタイプを指定します。ただし、メッセージに含めることができるセグメントは、メッセージ・タイプではなく構造タイプで指定します。メッセージ・タイプを作成するときに、オプションで構造タイプを同時に作成することもできます。

ベース・スキーマで定義されているメッセージ・タイプに Z セグメントを追加するには、メッセージ・タイプと構造をカスタム・スキーマにコピーした後、末尾の Z セグメントを構造タイプに追加します。例えば、バージョン 2.5 をベース・スキーマとするカスタム・スキーマの ORU\_R01 メッセージに ZPI セグメントを追加するには、以下の操作を行います。

1. 左側のパネルでカスタム・スキーマを選択し、**[メッセージ・タイプ]** タブを選択して、**[ベースからコピー]** をクリックします。
2. **[コピーするメッセージ・タイプ]** プルダウンで ORU\_R01 メッセージ・タイプを選択します。コピーしたメッセージ・タイプと同じ名前が新しいメッセージ・タイプ名として入力され、ベースの定義と一致するように **[メッセージ構造]** と **[返されるメッセージ・タイプ]** が設定されます。チェック・ボックスにチェックを付けると、送信メッセージ構造がカスタム・スキーマでまだ定義されていない場合、自動的に作成されます。このメッセージ構造は、ベース・スキーマから構造をコピーすることによって作成されます。HL7 標準の一部のバージョンには、2 つのメッセージ構造を使用できるメッセージ・タイプが含まれています。そのタイプで使用可能な追加の構造を定義するには、**[代替メッセージ構造]** を使用します。

**Custom Message Type Wizard**  
Create a custom message type definition

Custom Schema Category Custom2.5  
Base Schema Category 2.5

Message Type to Copy: ORU\_R01 (2.5)

New Message Type Name: ORU\_R01

Message Type Description: Unsolicited transmission of an observation message - ORU Subscription

Message Structure: ORU\_R01  
☒ Create this message structure in the Custom Schema if it does not exist there already

Alternate Message Structure:   
☒ Create this message structure in the Custom Schema if it does not exist there already

Returning Message Type: ACK\_R01 (2.5)

Use this to create a custom type.  
For help, hover over the field in the list.

[OK] をクリックすると、ORU\_R01 メッセージ・タイプと ORU\_R01 構造タイプがカスタム・スキーマで定義されます。

3. [DocType 構造] タブ、ORU\_R01 構造タイプの順にクリックします。カスタム・スキーマ・エディタに構造タイプのグラフィカル表現が表示されます。
4. メッセージ構造に含まれるセグメントの追加、削除、または変更を開始するには、[編集] ボタンをクリックします。[編集] ボタンと [利用可能なセグメント] セクションが表示されるのは、カスタム・スキーマの一部である構造を表示している場合のみです。

例えば、カスタム・メッセージ構造ウィザードを使用して ORU\_R01 構造タイプのコピーを編集している場合、ウィザードは次のように表示されます。

Message Structure Name	ORU_R01	Edit	Delete
Custom Schema Category	Custom2.5		
Base Schema Category	2.5		
Description (optional)	Unsolicited transmission of an observation message - ORU Subscription (Response)		

### Available Segments

+

ABS (2.5)

ACC (2.5)

ADD (2.5)

AFF (2.5)

AIG (2.5)

AIL (2.5)

### Structure

⇌

MSH


SFT()

▶ PIDgrpgrp()

DSC

ベース・スキーマのメッセージ定義を拡張する場合は、ベース・スキーマで指定されているものと同じセグメント名と構造名を使用する必要があります。

**[利用可能なセグメント]** セクションから **[構造]** セクションにセグメントをドラッグ・アンド・ドロップして、メッセージ構造内の

正しい位置にセグメントを配置することができます。カスタム・セグメントの追加に焦点を絞るには、フィルタ・ボタン  をクリックします。カスタム・セグメントのみが **[利用可能なセグメント]** セクションに表示されます。

セグメントを **[構造]** セクションに移動した後、セグメントをクリックして、セグメントを **[オプション]** または **[Repeating]** として定義することができます。

また、全体としてオプションまたは繰り返しであるセグメントのグループを作成することもできます。グループを作成するには、以下の手順に従います。

1. グループの最上位にするセグメントをクリックして、**[グループ]** を選択します。グループの最上位レベルが作成されます。その名前は選択したセグメントと同じですが、grp が追加されます。
2. 最上位レベルのグループ名をクリックして、**[オプション]** または **[Repeating]** を選択します。グループ全体を **[オプション]** または **[Repeating]** として指定しない限り、セグメントはグループとして保存されません。
3. ドラッグ・アンド・ドロップを使用してセグメントをグループに追加します。グループに含まれるセグメントが 1 つだけの場合、グループは保存されません。

セグメントのセットを作成することもできます。この場合、メッセージの構造に、セット内の任意のセグメントを 1 つ含めることができます。このセットを作成するには、セットの一部にするセグメントをクリックして、**[Choice]** を選択します。セットの最上位レベルが作成されます。その名前は選択したセグメントと同じですが、union が追加されます。このセットの一部である他のセグメントをドラッグ・アンド・ドロップすることができます。和集合に含まれるセグメントが 1 つだけの場合、和集合は保存されません。

**[構造]** セクションにおけるセグメントの表現上の規則と名前付け規約の詳細は、“[メッセージ構造の表示](#)” を参照してください。

未加工のテキスト形式でメッセージ構造を表示するには、 をクリックします。編集モードでは、未加工のテキストを編集してメッセージ構造を変更できます。

**注釈** カスタム・スキーマでメッセージ構造を定義すると、同じ構造を共有するすべてのメッセージ・タイプでその定義が使用されます。例えば、末尾の ZPI セグメントを ORU\_R30 構造に追加した場合、ORU\_R30、ORU\_R31、および ORU\_R32 の各メッセージ・タイプはすべて、同じ ORU\_R30 構造を共有するため、これらのメッセージ・タイプで末尾の Z セグメントを使用できます。これらのメッセージ・タイプをカスタム・スキーマに含める必要はありません。ベース・スキーマの定義によって、カスタム・スキーマの構造タイプが使用されます。

### 2.3.4 データ構造とコード・テーブルの編集

データ構造は、単純なデータ型ではなく、構造化された値を持つフィールドを指定するためのメカニズムを提供します。コード・テーブルは、フィールドで使用可能な値のセットを定義するためのメカニズムを提供します。通常、データ構造とコード・テーブルは HL7 標準の本文で定義され、カスタム拡張として定義されることはありません。まれなケースとして、カスタム・スキーマでデータ構造とコード・テーブルを定義する必要がある場合は、カスタム HL7 スキーマ・エディタを使用して定義することができます。データ構造を編集するウィザードは、セグメントを編集するウィザードとよく似ています。コード・テーブルを編集するウィザードでは、コード・テーブルのコードと説明を定義できます。コードによって、フィールドで使用可能な値が指定されます。

## 2.4 HL7 メッセージ・ビューワ・ページの使用法

インターシステムズでは、HL7 のメッセージ・ビューワ・ページを提供しています。このページを使用して、HL7 メッセージ (プロダクション・メッセージ・アーカイブ内のメッセージまたは外部ファイル) を表示、変換、およびエクスポートできます。

このページにアクセスするには、[Interoperability]→[相互運用]→[HL7 v2.x]→[HL7 v2.x メッセージ・ビューワ] を選択します。

### 2.4.1 オプションの選択

表示するドキュメントを指定するには、以下の操作を行います。

1. [ドキュメント・ソース] で、[ファイル]、[メッセージ・ヘッダ ID]、または [メッセージ・ボディ ID] を選択します。
2. 表示するドキュメントを指定します。
  - ・ [ファイル] を選択した場合は、[参照] を使用してファイルを選択します。[ファイル内のドキュメント番号] に、表示するドキュメントの番号を入力します。
  - ・ [メッセージ・ヘッダ ID] または [メッセージ・ボディ ID] を選択した場合は、表示するメッセージ・ヘッダまたはメッセージ・ボディの ID を入力します。
3. ドキュメントの解析方法を指定します。そのためには、[ドキュメント構造またはスキーマ] で以下のオプションのいずれかを選択します。
  - ・ [ビジネス・サービスから受け取る] – ビジネス・サービスによって割り当てられるスキーマを使用します。これを選択する場合は、ドロップダウン・リストからビジネス・サービスを選択します。  
このオプションでは、特定のビジネス・サービスがこのドキュメントに割り当てる DocType を決定できます。
  - ・ [特定のスキーマ・カテゴリ/バージョンを使用] – ドロップダウン・リストからドキュメント・カテゴリを選択します。

- ・ **[特定の DocType を使用]**ードキュメント構造 (<MessageStructure>) の名前を category:structure の形式で入力します。パーサでは、このドキュメント構造が使用されます。
- ・ **[コンテンツの宣言されたバージョン:名前を使用]**ードキュメント内で宣言されているドキュメント・タイプに関連付けられたドキュメント構造を使用します。
- ・ **[オブジェクトの保存された DocType を使用]**ードキュメント本文オブジェクト内で宣言されている DocType を使用します (このオプションは、ファイルからロードされた保存済みのドキュメントには適用されません)。
- ・ **[なし]**ードキュメントの解析にどの DocType も使用しません。代わりに、未加工のセグメントをいずれもリンクに変換せずにそのまま表示します。

このオプションでは、特定のデータ・ソースからのドキュメントの解釈をさまざまなスキーマ・カテゴリ・タイプとして試みることによって、そのソースからのドキュメントの処理時にどの DocType を使用するのが適切か決定できます。これを行う理由はさまざまです。例えば、外部アプリケーションを更新する際に、このアプリケーションから送信されるドキュメントの実際のバージョンが変更されているものの、このドキュメントで送信されるタイプ宣言が更新されていない場合があります。また、ドキュメントでカスタム・ドキュメント構造を使用する場合に、スキーマ・ベースとして使用する組み込みカテゴリを決定する場合にも役立ちます。

4. 必要に応じて、**[ドキュメントを変換]**をクリックして、変換の詳細を指定します。“[変換のテスト](#)”を参照してください。
5. **[OK]**をクリックします。

## 2.4.2 メッセージの解析

前述の手順を完了すると、メッセージ・ビューワの画面右側に以下が表示されます。

- ・ ドキュメントに関する次の基本情報を含む要約レポート。
  - 適用されているデータ変換 (該当する場合)
  - メッセージ ID
  - DocType
  - DocType カテゴリ
  - DocType の説明 (ある場合)
  - セグメントの数
  - 子ドキュメントと親ドキュメントの数 (該当する場合)
- ・ メッセージ構造内のセグメントごとに 1 行ずつあるメッセージ・データ。各行の内容は以下のとおりです。
  - セグメント番号
  - セグメント名 (PID や NTE など)
  - フィールドのコンテンツと区切り文字 (メッセージに含まれているもの)

選択したスキーマとメッセージが一致する場合は、次のように、セグメントと要素が青で表示されます。セグメントまたはフィールドをクリックすると、関連する構造ページにリンクします。

HL7 ORM\_001 Message - Id = , DocType = '2.3.1:ORM\_001', DocType Category = '2.3.1'  
'Pharmacy/treatment order message - Order message (also RDE, RDS, RGV, RAS)', 8 Segm

1	MSH ^~\& EPIC_EC CCF PHARMACY 2623735 20050126150624 11111 ORM^O01
2	PID _ 16284718^_^-^_^-^AM Z1907 _ JONES^IVAN^S^A^_^-^_^-^ _ 19490518 M
3	ORC NW 244674^EPC _ _ _ _ _^-^_^-^_^-^200501260000^200503272359^Norm^_ _
4	RXO 0045--04-52-2^PRILOSEC 325 MG PO TABS^NDC 60 _ _^-^_ _ _ 2 tabs po q4-6
5	NTE 1 _ _ _
6	ORC NW 244674^EPC _ _ _ _ _ _^-^_^-^_^-^200501260000^200503272359^Norm^_ _
7	RXO 0045--04-52-2^WELLBUTRIN 325 MG PO TABS^NDC 60 _ _^-^_ _ _ 2 tabs po q
8	NTE 1 _ _ _

#### 2.4.2.1 セグメント・アドレスの表示

セグメント・アドレスを表示するには、濃色の列に示されているセグメント名にカーソルを合わせます。ツールヒントに以下の情報が表示されます。

- ・ 仮想プロパティ・パスで使用するセグメント・アドレス
- ・ このセグメントの説明的な名前

[illegible]

#### 2.4.2.2 フィールド・アドレスの表示

フィールド・アドレスを表示するには、メッセージ構造内のフィールドにカーソルを合わせます。ツールヒントに以下の情報が表示されます。

- ・ 仮想プロパティ・パスで使用する field アドレス (数値)
- ・ 仮想プロパティ・パスで使用する field アドレス (名前)
- ・ フィールドの構文規則を示す文字。文字の先頭には次の記号を使用できます。



記号	意味
!	(1 回のみ) フィールドは必須で、1 回のみ出現する必要があります。
?	(0 または 1 回) フィールドはオプションですが、出現する場合は、1 回のみ出現することができます。
+	(1 回以上) フィールドは 1 回以上繰り返すことができます。
*	(0 回以上) フィールドは 0 回以上繰り返すことができます。
&	特定の条件においてのみ、フィールドは存在することも、繰り返すこともできます。
n*	(0 ~ n 回) フィールドは最大 n 回繰り返されます。
(m)	m は、フィールド内の最大文字数です。フィールドの繰り返しごとに、この文字数を含めることができます。

<sup>A</sup>WELLBUTRIN 325 MG PO TABS<sup>A</sup> NDC | 60 | \_ | .  
1.2.1 / RequestedGiveCode.text.1 / ?(100)

### 2.4.2.3 バッチ・メッセージ

フィールドが山かっこ (< >) で囲まれている場合、サブドキュメントへのリンクです。クリックすると、そのドキュメントの要約レポートとメッセージ・データが表示されます。

この章で後述する“[バッチ・メッセージの表示](#)”も参照してください。

## 2.4.3 変換のテスト

変換をテストするには、以下の操作を行います。

1. [ドキュメントを変換] をクリックします。
2. [データ変換の選択] で、データ変換を選択します。
3. [表示オプションの選択] で、以下のいずれかを選択します。
  - ・ [変換結果のみ] – 変換後のドキュメントのみ表示します。
  - ・ [元のメッセージと結果を一緒に表示] – 元のドキュメントと変換後のドキュメントの両方を表示します。
4. 次に、以下のいずれかまたは両方の操作を行います。
  - ・ [OK] をクリックして、変換後のドキュメントを表示します。
  - ・ [結果をファイルに保存] をクリックして、変換後のドキュメントをファイルに保存します。この場合は、パスとファイル名も指定します。

デフォルトのディレクトリはアクティブなネームスペースの管理ディレクトリです。例えば、InterSystems IRIS を C:¥MyIRIS ディレクトリにインストールし、現在のネームスペースが MyNamespace である場合、ファイルは C:¥MyIRIS¥Mgr¥MyNamespace¥filename として保存されます。



## 2.5 バッチ・メッセージの表示

メッセージが 1 つの HL7 メッセージではなく、バッチ形式の HL7 メッセージのグループである場合、[HL7 ドキュメント] ページでは、異なる方法でメッセージが処理されます。具体的には、バッチ・メッセージ構造を 1 レベルずつ確認できます。

次に示す画面は、FHS セグメントから始まるバッチ・メッセージを表示するように要求したときの結果です。バッチ・メッセージが解析され、3 つのセグメント (FHS、FTS、およびその間の子ドキュメントのブロック) があることがわかりました。ブロックには 2 つの子ドキュメントが含まれており、それぞれ BHS で始まり、BTS で終わります。このメッセージは 2 レベルのバッチ・メッセージです。

メッセージ・ビューワでは、子ドキュメントに識別子 <2> と <33> が割り当てられています。最上位レベルの親ドキュメントが表示され、リンク (<2> と <33>) を使用して 2 つの子ドキュメントが表されています。画面は次のとおりです。

This is HL7 Message #1 found in file 'C:\EnsProjectFiles\HL7\_practice\HL7safe\XYZ30Batch2'  
DocType '2.3.1:FHS' based on 'Category / doc.Name'

HL7 FHS Message - Id = 1, DocType = '2.3.1:FHS', MessageCategory = '2.3.1'  
'batch message file', 3 Segments, 2 child documents

1	FHS	: 1:18 : 1
2	BHSDocsRef : 2 : 2	BHS documents : <2> <33>
3	FTS	

HL7 バッチ・メッセージ画面で子ドキュメントのリンクをクリックすると、新しいブラウザ・ウィンドウが開いてその子ドキュメントが表示されます。最上位レベルの親を表示する [メッセージ・ビューワ] ウィンドウは、元のブラウザ・ウィンドウで開いたままです。

次の画面は、前の画面で子ドキュメントのリンク <2> をクリックしたときの結果です。この例は 2 レベルのバッチ・メッセージなので、子ドキュメント <2> には、独自の子ドキュメント <3> から <32> があります。

この例では、メッセージ・ビューワの便利なナビゲーション機能を紹介します。10 個を超える子ドキュメントがバッチ・メッセージにある場合、メッセージ・ビューワには最初の 5 つと最後の 5 つの子ドキュメントへのリンクが表示されます。リストの間はテキスト・フィールドになっていて、ここには最初の番号から最後の番号までの範囲の ID 番号を入力することができます。番号を入力したら、[その他] をクリックします。新しいブラウザ・ウィンドウが開いて、その子ドキュメントが表示されます。

This is the HL7 Message object with Id 2  
DocType " based on 'HL7 Message stored DocType property'

HL7 BHS Message - Id = 2, DocType = "", MessageCategory = "", 3 Segments, 57 child documents, parent document : <1>

1	BHS	: 1:18 : 1
2	MessageDocsRef : 2 : 30	Message documents : <3> <4> <5> <6> <7> 8 <56> <57> <58> <59> <60> Other
3	BTS	

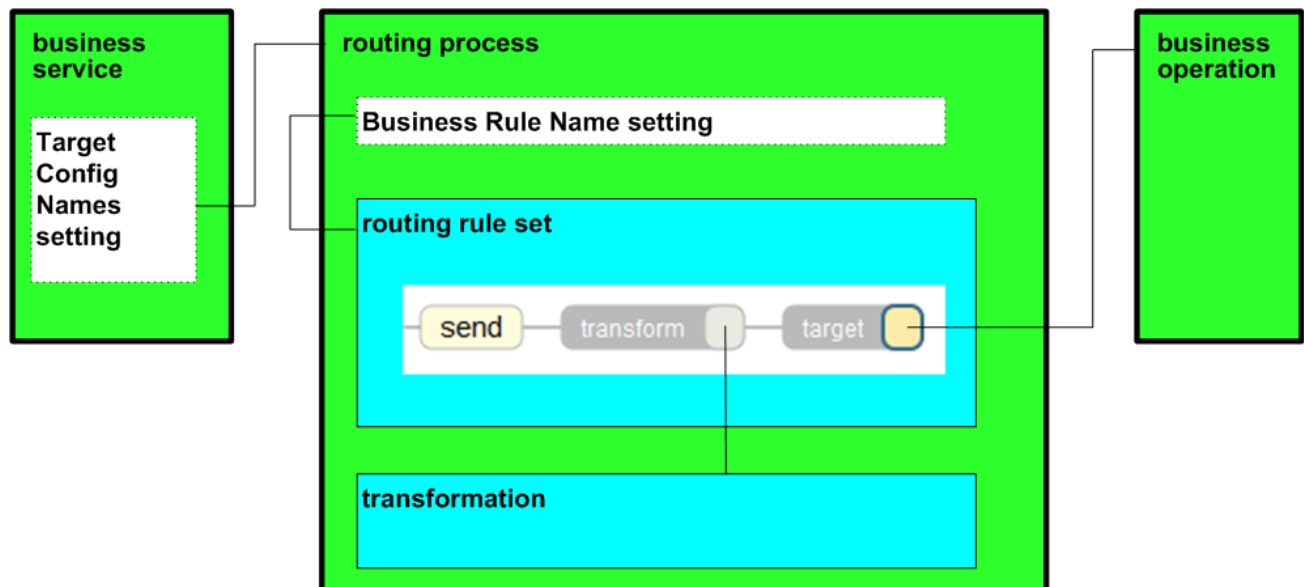
次の図は、前の図で子ドキュメントのリンク <6> をクリックしたときの結果です。これはバッチ・メッセージ階層の最下位レベルなので、次の図に示されているメッセージ <6> は、MSH セグメントから始まる通常の HL7 バージョン 2 メッセージです。



項目	クラス	メモ
ビジネス・サービス	<ul style="list-style-type: none"> <li>EnsLib.HL7.Service.FileService</li> <li>EnsLib.HL7.Service.FTPService</li> <li>EnsLib.HL7.Service.HTTPService</li> <li>EnsLib.HL7.Service.SOAPService</li> <li>EnsLib.HL7.Service.TCPService</li> </ul>	クラス名が示しているように、これらの HL7 ビジネス・サービス・クラスのそれぞれで別々のアダプタが使用されます。HL7 HTTP サービスでは CSP ポートを使用することも、専用ポートを使用することもできます。
ビジネス・プロセス	EnsLib.HL7.MsgRouter.RoutingEngine	このクラスは、標準の仮想ドキュメント・ルーティング・プロセスの特殊なバージョンです。
ビジネス・オペレーション	<ul style="list-style-type: none"> <li>EnsLib.HL7.Operation.FileOperation</li> <li>EnsLib.HL7.Operation.FTPOperation</li> <li>EnsLib.HL7.Operation.TCPOperation</li> </ul>	クラス名が示しているように、これらの HL7 ビジネス・オペレーション・クラスのそれぞれで別々のアダプタが使用されます。
メッセージ	EnsLib.HL7.Message	HL7ドキュメントを仮想ドキュメントとして転送するための特殊なメッセージ・クラスです。
検索テーブル	EnsLib.HL7.SearchTable	HL7ドキュメント専用の検索テーブル・クラスです。

これらのクラスのサブクラスを作成して使用することもできます。

ビジネス・ホスト・クラスには構成可能なターゲットが含まれています。それらのいくつかを下の図に示します。



その他の構成可能なターゲットについては、“[参照](#)”を参照してください。

## 2.6.1 HL7 メッセージ・クラスの詳細

インターシステムズ製品には、HL7 バージョン 2 仮想ドキュメントの組み込みクラスが用意されています。そのクラスは **EnsLib.HL7.Message** です。仮想ドキュメント・メッセージ・クラスの基本情報は、“プロダクション内での仮想ドキュメントの使用法”の“[仮想ドキュメント・クラス](#)”を参照してください。基本的なプロパティとメソッドに加えて、**EnsLib.HL7.Message** には次のプロパティがあります。

### TypeCategory

**TypeCategory** プロパティには HL7 カテゴリ名が格納されます。通常、HL7 データを受け取る HL7 ビジネス・サービスは HL7 メッセージをインスタンス化し、そのメッセージに **TypeCategory** 値を割り当てます。プロダクションは、受信メッセージ・データの MSH セグメントで宣言されているメッセージ・タイプとこの **TypeCategory** を組み合わせます。この組み合わせによって、HL7 スキーマ定義内の **<MessageType>** が識別されます。この **<MessageType>** には、他の **DocType** が割り当てられていない場合に HL7 メッセージで DocType として使用される **<MessageStructure>** が関連付けられています。

### Name

**Name** プロパティは、外部データ・ソースから MSH セグメントで提供された HL7 メッセージ構造名 (ADT\_A08 や ORM\_O01 など) を格納する読み取り専用の文字列です。**Name** は、医療アプリケーションが送信していると認識している HL7 メッセージ構造を確認する際に役立つことがあります。ただし、これは実際のメッセージ・コンテンツとは異なる場合があります。

# 3

## プロダクションの構成

この章では、HL7 ルーティング・インタフェースを組み込むようにプロダクションを構成する方法について説明します。また、プロダクションがまだない場合に新しい HL7 ルーティング・プロダクションを作成する方法についても説明します。

この章では、**[プロダクション構成]** ページで実行するタスクについて説明します。次の章で追加のタスクについて説明します。

### 3.1 新しい HL7 ルーティング・プロダクションの作成

新しい HL7 ルーティング・プロダクションを作成できますが、そのためには、以下の操作を行います。

1. 管理ポータルで、該当するネームスペースに切り替えます。  
そのためには、タイトル・バーの **[切り替え]** を選択し、対象のネームスペースをクリックして、**[OK]** をクリックします。
2. **[Interoperability]** → **[リスト]** → **[プロダクション]** の順に選択します。
3. **[新規作成]** を選択して、プロダクション・ウィザードを呼び出します。
4. **[パッケージ名]**、**[プロダクション名]**、および **[説明]** を入力します。
5. **[HL7メッセージ]** プロダクション・タイプを選択し、**[OK]** を選択します。

初期のプロダクションのインタフェースは 1 つで、その要素は次のとおりです。

- ・ **HL7FileService** — デフォルト設定が適用された、無効な HL7 ファイル・サービス
- ・ **MsgRouter** — 空のルーティング・ルール・セットを含む HL7 ルーティング・プロセス
- ・ **HL7FileOperation** — デフォルト設定が適用された、無効な HL7 ファイル・オペレーション

HL7 ルーティング・プロダクションを作成する際には、このようなインタフェースを多数作成して構成します。まず、これらの初期要素を有効にし、コピーして名前を変更し、ニーズに合わせて変更できます。インタフェースの作成時に、ある項目を構成しているときに、まだ作成していない別の項目の名前を入力する必要があることがよくあります。混乱を回避するには、明確な名前付け規約が不可欠です。推奨事項については、“プロダクション作成のベスト・プラクティス”の“**名前付け規約**”を参照してください。規則については、“プロダクションの構成”の“**構成名**”を参照してください。

インタフェース要素に加えて、初期のプロダクションには、HL7 メッセージをルーティングするのではなく、サポート機能をプロダクションに提供する次のような要素も用意されています。

- ・ **不正なメッセージ・ハンドラ** — 検証に失敗したメッセージの組み込みの宛先

- ・ **Ens.Alert** – 企業のスケジュールや手順に合わせ、アラートのソース（つまり、問題がある要素）とさまざまな条件（時刻など）に応じて異なる電子メール・ビジネス・オペレーションに**アラート・メッセージをルーティング**するように構成できるルーティング・ルール・セットを含むルーティング・プロセス
- ・ **PagerAlert** と **EmailAlert** – テキスト・メッセージ（アラートなど）を携帯電話または電子メール・アドレスに送信するように構成できる電子メール・ビジネス・オペレーション

## 3.2 HL7 ビジネス・サービスの追加

プロダクションがインターシステムズ製品外部からの HL7 メッセージを受信するようにするには、HL7 ビジネス・サービスをプロダクション構成に追加する必要があります。HL7 ビジネス・サービスをプロダクションに追加するには、HL7 ビジネス・サービスを作成してプロダクションに統合し、必要に応じて構成する必要があります。以降の項で詳しく説明します。

### 3.2.1 HL7 ビジネス・サービスの作成

HL7 ビジネス・サービスをプロダクションに追加するには、以下の操作を行います。

1. 管理ポータル の **[プロダクション構成]** ページ（ホーム・ページから **[Interoperability]**→**[構成する]**→**[プロダクション]** を選択）でプロダクションを表示します。
2. **[サービス]** 列で、**[追加]** ボタン（プラス記号）をクリックします。
3. **[HL7入力]** タブをクリックします。
4. **[入力タイプ]** リストから以下のいずれかをクリックします。
  - ・ TCP
  - ・ ファイル
  - ・ FTP
  - ・ HTTP
  - ・ SOAP

これにより、ビジネス・サービスのホスト・クラスが選択されます。

5. **[HL7 サービス名]** に、このビジネス・サービスの名前を入力します。この名前は、ビジネス・サービス間で一意である必要があります。ピリオドやスペースを使用しないでください。  
デフォルトは、このサービスの基になるクラスの名前です。
6. **[HL7 サービス・ターゲット]** で、以下のいずれかを選択します。
  - ・ **[新規ルーター作成]** – ルーティング・プロセスがプロダクションに追加され、ビジネス・サービスがそのプロセスをターゲットとして使用するよう構成されます。後で詳細を編集できます。
  - ・ **[今はなし]** – このビジネス・サービスのターゲットを指定しません。
  - ・ **[リストから選択]** – この場合、ドロップダウン・リストから既存のビジネス・ホストも選択します。
7. **[OK]** をクリックします。

ここで使用する HL7 ビジネス・サービスに加えて、インターシステムズは、**EnsLib.File.PassthroughService** と **EnsLib.FTP.PassthroughService** という 2 つの単純なビジネス・サービスを提供しています。プロダクション内でファイルを転送するだけでよく、ファイルを解析したり、ファイルの形式を HL7 として設定したりする必要がない場合は、いずれかを選択できます。

HL7 メッセージでないデータをプロダクションが受信するようにする場合は、“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。“相互運用プロダクションの概要”の“[接続オプション](#)”も参照してください。

### 3.2.2 HL7 ビジネス・サービスの統合と構成

新しい HL7 ビジネス・サービスをプロダクションに統合するには、メッセージの転送先となるルーティング・プロセスまたはビジネス・オペレーションにそのビジネス・サービスを関連付ける必要があります。さらに、非標準のメッセージ構造をビジネス・サービスが受信するようにする場合は、それらのメッセージを解析して検証するためのカスタム HL7 スキーマ定義を作成する必要があります。これを行うには、以下を実行します。

1. HL7 ビジネス・サービスに必要な **[ターゲット構成名]** 項目または **[メッセージ・スキーマ・カテゴリ]** 項目を作成するための手順を完了します。これらの項目は次のいずれかにすることができます。
  - ・ [HL7 ルーティング・プロセス](#) (ルーティング・インタフェースの場合)。[次の節](#)を参照してください。
  - ・ [HL7 ビジネス・オペレーション](#) (設計上、このインタフェースのルーティング・プロセスを省略し、受信ビジネス・サービスから送信ビジネス・オペレーションにメッセージを渡すだけの場合)。この章で後述する“[HL7 ビジネス・オペレーションの追加](#)”を参照してください。
  - ・ 受信 HL7 メッセージを解析するためのカスタム HL7 スキーマ定義。カスタム・スキーマ・カテゴリの作成に関する情報は、“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](#)”を参照してください。
2. **[プロダクション構成]** ページのダイアグラムに戻ります。新しい HL7 ビジネス・サービスを選択します。**[ターゲット構成名]** フィールドと **[メッセージ・スキーマ・カテゴリ]** フィールドがこれまで空白だった場合は、ここで構成し、**[適用]** をクリックします。
3. 必要に応じて、ビジネス・サービスのその他の設定を構成します。詳細は、“[参照](#)”の“[HL7 ビジネス・サービスに関する設定](#)”を参照してください。

## 3.3 HL7 ルーティング・プロセスの追加

HL7 ルーティング・プロセスをプロダクションに追加するには、HL7 ルーティング・プロセスを作成してプロダクションに統合し、必要に応じて構成する必要があります。以降の項で詳しく説明します。

### 3.3.1 HL7 ルーティング・プロセスの作成

HL7 ルーティング・プロセスをプロダクションに追加するには、以下の操作を行います。

1. 管理ポータル**[プロダクション構成]** ページ (ホーム・ページから **[Interoperability]**→**[構成する]**→**[プロダクション]** を選択) でプロダクションを表示します。
2. **[プロセス]** 列で、**[追加]** ボタン (プラス記号) をクリックします。
3. ビジネス・プロセス・オプションとして **[HL7メッセージ・ルータ]** をクリックします。ルータ・クラスは、デフォルトで `EnsLib.HL7.MsgRouter.RoutingEngine` に設定されます。
4. **[HL7 ルータ名]** に、このビジネス・プロセスの名前を入力します。この名前は、ビジネス・プロセス間で一意である必要があります。ピリオドやスペースを使用しないでください。  
デフォルトは、このプロセスの基になるクラスの名前です。
5. **[ルーティング・ルール名]** で、以下のいずれかの操作を行います。
  - ・ **[ルーティング・ルール名]** ドロップダウン・リストから既存のルーティング・ルールを選択します。



- ・ [ルールの自動作成] を選択して、[ルーティング・ルール名] にルール名を入力します。この場合、ウィザードによって、プロダクションと同じパッケージにルーティング・ルール・クラスが作成されます。

6. [OK] をクリックします。

### 3.3.2 HL7 ルーティング・プロセスの統合と構成

新しい HL7 ルーティング・プロセスをプロダクションに統合するには、その受信メッセージを受け取るビジネス・サービスと、それらのメッセージに基づいてそのアクションを決定するルーティング・ルール・セットにそのルーティング・プロセスを関連付ける必要があります。これを行うには、以下を実行します。

1. [HL7 ビジネス・サービス](#)を選択します。画面の右側にあるメニューで、[設定] タブをクリックし、[基本設定] メニューを開きます。[ターゲット構成名] フィールドに、新しい HL7 ルーティング・プロセスの名前を入力します。
2. [ルーティング・ルール・セット](#)を作成します。構成ダイアグラムでルーティング・プロセスを選択します。[ビジネス・ルール名] フィールドに、新しいルーティング・ルール・セットの完全名を入力します。
3. 必要に応じて、ルーティング・プロセスのその他の設定を構成します。詳細は、“[参照](#)” の “[HL7 ルーティング・プロセスに関する設定](#)” を参照してください。

## 3.4 HL7 シーケンス・マネージャの追加

HL7 メッセージは、複数のプロセッサで処理される場合は特に、さまざまな理由から順序が正しくなくなることがあります。場合によっては、HL7 メッセージが正しい順序で処理されるようにするのが望ましいことがあります。そのような場合は、プロダクションの該当する部分に HL7 シーケンス・マネージャを追加できます。

HL7 シーケンス・マネージャは、受信 HL7 メッセージを（場合によっては複数のソースから）受け取った後、メッセージの MSH:13 SequenceNumbers フィールドで指定された順序でメッセージをターゲット構成項目に転送するビジネス・プロセスです。

シーケンス・マネージャは、重複するメッセージ、およびメッセージ間の時間差を検出できます。また、連続するメッセージ間の時間差が大きく、問題を示している状況も特定します。感度のレベルを調整できますが、そのためには、構成設定を使用します。

HL7 メッセージ・ルーティング・プロダクションで使用する HL7 シーケンス・マネージャをビルドするには、[HL7 シーケンス・マネージャを作成して構成](#)した後、プロダクションに[統合](#)する必要があります。このトピックでそれぞれの手順について説明します。

**重要** HL7 シーケンス・マネージャは HL7 準拠の格納転送アプリケーションであり、HL7 標準の第 2 章、第 2.10.1 節で定義されている HL7 シーケンス番号プロトコルをサポートしていません。

### 3.4.1 HL7 シーケンス・マネージャの作成

HL7 シーケンス・マネージャをプロダクションに追加するには、以下の操作を行います。

1. 管理ポータルの [プロダクション構成] ページ (ホーム・ページから [Interoperability]→[構成する]→[プロダクション] を選択) でプロダクションを表示します。
2. [プロセス] 列で、[追加] ボタン (プラス記号) をクリックします。
3. [プロセス・クラス] リストから `EnsLib.HL7.SequenceManager` を選択します。
4. [名前] に、このビジネス・プロセスの名前を入力します。この名前は、ビジネス・プロセス間で一意である必要があります。ピリオドやスペースを使用しないでください。



デフォルトは、このプロセスの基になるクラスの名前です。

5. [OK] をクリックします。

必要に応じて、複数のシーケンス・マネージャをプロダクションに追加できます。

### 3.4.2 HL7 シーケンス・マネージャの統合と構成

新しい HL7 シーケンス・マネージャをプロダクションに統合するには、その受信メッセージを受け取るビジネス・サービスと、シーケンス・マネージャがメッセージを正しい順序に編成してから送信するターゲットにそのシーケンス・マネージャを関連付ける必要があります。これを行うには、以下を実行します。

1. [HL7 ビジネス・サービス](#)を選択します。[ターゲット構成名] フィールドに、新しい HL7 シーケンス・マネージャの名前を入力します。
2. HL7 シーケンス・マネージャの構成に戻ります。成功したメッセージの [出力ターゲット構成名] のリストを指定します。
3. 重複するメッセージをシーケンス・マネージャがチェックするようにする場合は、[メッセージ重複チェック有効] を真に設定します。

トラブルシューティングを目的として重複するメッセージを保存する場合は、それらを受け取る [HL7 ビジネス・オペレーション](#)を作成します。[重複したメッセージターゲット] フィールドに、新しい HL7 ビジネス・オペレーションの名前を入力します。

4. 順序が正しくないメッセージをシーケンス・マネージャがチェックするようにする場合は、[シーケンス番号チェックを実施 On] を [ ] または [ ] に設定し、[ラジギャップサイズ] と [メッセージウェイトタイムアウト] の値を設定して順序チェックの詳細を構成します。それ以外の場合は、[シーケンス番号チェックを実施 On] を [ ] に設定します。

トラブルシューティングを目的として順序が正しくないメッセージを保存する場合、それらを受け取る [HL7 ビジネス・オペレーション](#)を作成します。[メッセージターゲットのシーケンス範囲外] フィールドに、新しい HL7 ビジネス・オペレーションの名前を入力します。

5. シーケンス・マネージャがインターシステムズ製品外にメッセージを送信する前に変換するようにする場合は、[出力変換を実施 On] を [ ] または [Receiver] に設定し、[出力ファシリティアプリケーション] の値を設定します。それ以外の場合は、[出力変換を実施 On] を [ ] に設定します。
6. シーケンス・マネージャがチェックや変換を行わずに、単に [出力ターゲット構成名] に送信する [パススルーメッセージタイプ] を指定します。
7. 必要に応じて、シーケンス・マネージャのその他の設定を構成します。詳細は、“参照” の “[HL7 シーケンス・マネージャに関する設定](#)” を参照してください。

### 3.4.3 プログラムによる HL7 シーケンス・データへのアクセス

SQL を使用してシーケンス・マネージャの実行時データにアクセスできます。そのためには、テーブル `EnsLib_HL7.SM.RuntimeData.Thread` に対してクエリを実行します。このテーブルには、次の文字列フィールドがあります。

#### Application

HL7 メッセージから取得した、送信アプリケーションまたは受信アプリケーションの名前。

#### Facility

HL7 メッセージから取得した、送信施設または受信施設の名前。

#### Thread

次のいずれかの文字列。

- ・ main
- ・ resend

#### Type

次のいずれかの文字列。

- ・ Sender
- ・ Receiver

#### NextSequenceNumber

指定された施設、アプリケーション、スレッド、およびタイプにおけるシーケンス内の次の番号を識別します。

SQL クエリの例は次のようになります。

#### SQL

```
SELECT Application,Thread,Type,NextSequenceNumber FROM EnsLib_HL7.SM.RuntimeData.Thread WHERE Facility = 'mine'
```

## 3.5 HL7 ビジネス・オペレーションの追加

プロダクションから HL7 メッセージを送信するには、HL7 ビジネス・オペレーションを追加する必要があります。HL7 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、HL7 ビジネス・オペレーションを作成してプロダクションに統合し、必要に応じて構成する必要があります。以降の項で詳しく説明します。

### 3.5.1 HL7 ビジネス・オペレーションの作成

HL7 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、以下の操作を行います。

1. 管理ポータルでの **[プロダクション構成]** ページ (ホーム・ページから **[Interoperability]**→**[構成する]**→**[プロダクション]** を選択) でプロダクションを表示します。
2. **[オペレーション]** 列で、**[追加]** ボタン (プラス記号) をクリックします。
3. **[HL7出力]** をクリックします。
4. **[出カタイプ]** リストから以下のいずれかをクリックします。
  - ・ TCP
  - ・ ファイル
  - ・ FTP
  - ・ HTTP
  - ・ SOAP

この選択により、このビジネス・オペレーションのホスト・クラスが決まります。

5. **[オペレーション名]** に、このビジネス・オペレーションの名前を入力します。この名前は、ビジネス・オペレーション間で一意である必要があります。ピリオドやスペースを使用しないでください。

デフォルトは、このオペレーションの基になるクラスの名前です。

6. **[OK]** をクリックします。

ここで使用する HL7 ビジネス・オペレーションに加えて、インターシステムズは、**EnsLib.File.PassthroughOperation** と **EnsLib.FTP.PassthroughOperation** という 2 つの単純なビジネス・オペレーションを提供しています。プロダクション内でファイルを転送するだけでよく、ファイルを解析したり、ファイルの形式を HL7 として設定したりする必要がない場合は、いずれかを選択できます。

HL7 メッセージでないデータをプロダクションから送信するようにする場合は、“プロダクションの開発”の“[ビジネス・オペレーションの定義](#)”を参照してください。“相互運用プロダクションの概要”の“[接続オブション](#)”も参照してください。

### 3.5.2 HL7 ビジネス・オペレーションの統合と構成

新しい HL7 ビジネス・オペレーションをプロダクションに統合するには、メッセージを受信するターゲットとしてそのビジネス・オペレーションを指定するだけです。そのためには、HL7 ビジネス・オペレーションにメッセージを送信する項目を構成し、HL7 ビジネス・オペレーションの構成済みの名前を該当するフィールドに入力する必要があります。

- ・ ルーティング・インタフェースの場合は、[ルーティング・ルール・セット](#)の**[ターゲット]**フィールドに名前を入力します。
- ・ 受信ビジネス・サービスから送信ビジネス・オペレーションにメッセージを転送するだけのパススルー・インタフェースを使用する設計を採用している場合は、[HL7 ビジネス・サービス](#)の**[ターゲット構成名]**フィールドに名前を入力します。
- ・ 必要に応じて、ビジネス・オペレーションのその他の設定を構成します。詳細は、“[参照](#)”の“[HL7 ビジネス・オペレーションに関する設定](#)”を参照してください。

## 3.6 携帯電話アラートと電子メール・アラート

アラートを使用すると、プロダクションの実行中にユーザのデバイスに通知を送信できます。その目的は、システム管理者またはサービス技術者に問題の存在を警告することです。アラートは、電子メール、携帯電話、またはその他のメカニズムにより送信されます。詳細は、“プロダクションの構成”の“[アラートの構成](#)”を参照してください。

初期の HL7 ルーティング・プロダクションには、アラートをサポートするために次の要素が用意されています。

- ・ Ens.Alert – 初期のプロダクションには組み込みの Ens.Alert 要素が用意されています。この Ens.Alert は、AlertRule という関連するルーティング・ルール・セットを含むルーティング・プロセスです。

アラート・メッセージのソース（つまり、問題がある要素）とさまざまな条件（時刻など）に応じて異なるビジネス・オペレーションにアラート・メッセージをルーティングするように AlertRule を構成できます。

AlertRule を構成するには、[HL7 ルーティング・ルール・セット](#)の場合と同様の手順に従います。ただし、次の点異なります。

- **[メッセージ・クラス]** で、Ens.AlertRequest を選択します。
- **[ターゲット]** フィールドで、電子メール・ビジネス・オペレーションの構成済みの名前を選択または入力します。
- ・ PagerAlert と EmailAlert – テキスト・メッセージ（アラートの本文など）を携帯電話または電子メール・アドレスに送信するように構成できる通常の電子メール・オペレーションです。初期状態では、PagerAlert と EmailAlert は名前以外は同じです。アラートの優先度が異なるという事実を強調するために両方が用意されています。通常、優先度の高いアラートは携帯電話に送信され、優先度の低いアラートは電子メールに送信されます。

これらの初期のオペレーションをコピーする任意の数の宛先を構成できます。実行時に、AlertRule ルーティング・ルール・セットによって、使用する適切な宛先が決定されます。

# 4

## HL7 用のルーティング・ルール・セットの定義

HL7 インタフェースのルーティング・ルール・セットを作成する際の目的は、ソース・メッセージ内で見つかったセグメントに基づいてメッセージをどのように処理するかをプロダクションに指示することです。見つかったセグメントが重要である場合もあれば、それらのセグメント内で見つかった値が重要である場合もあります。

通常のルール・セットでは、各ルールは、そのルール・セットを呼び出したビジネス・プロセスに値を返します。ルーティング・ルール・セットでは、ルールは通常、HL7 メッセージを宛先に送信します。HL7 メッセージを送信する前に、変換することもあります。

ルール・セットを作成する場合は、予約パッケージ名を使用しないでください。“プロダクションの開発”の“[予約パッケージ名](#)”を参照してください。

ルール・エディタの使用については、“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。

1. ビジネス・ルール・ウィザードを使用して新しい HL7 ルーティング・プロセスを作成すると、新しいルーティング・プロセスと共に新しい空のルーティング・ルール・セットが作成されます。その情報テーブルには次の値が表示されます。

- ・ **[パッケージ名]** – プロダクション・クラスを格納するパッケージ。例えば、ウィザードを使用して、`TestRule.MyTest` というプロダクションにルーティング・プロセスを追加した場合、関連付けられたルーティング・ルールの **[パッケージ名]** は次のようになります。

`TestRule`

- ・ **[ルール名]** – ウィザードで選択した簡単な **[ルーティング・ルール名]**。例えば、次のようになります。

`MyRule`

**[パッケージ名]**と**[ルール名]**の組み合わせによって、ネームスペース内でルールが一意に識別されます。ルール定義の完全名は、次のように、**[パッケージ名]**と**[ルール名]**をドット(.)でつないだものになります。

`TestRule.MyRule`

**HL7 ルーティング・プロセス**を構成する際には、**[ルール名]**ではなく、この完全名が **[BusinessRuleName]** フィールドで使用する正しい値となります。ビジネス・プロセス・ウィザードでは自動的にこのように設定されます。

- ・ **[ルーチン・エンジン・クラス]** – ウィザードからのデフォルトの **[ルーター・クラス]**。変更しないでください。

`EnsLib.HL7.MsgRouter.RoutingEngine`

2. 独自のルール・レポートを目的として、次の追加フィールドを入力することもできます。

- ・ **[レポートグループ]** – レポートを目的としてルールをグループ化するために使用する値
- ・ **[レポート名]** – ルール・レポート・グループの表示値
- ・ **[簡単な説明]** – ルール定義の簡単な説明 (オプション)

3. ルーティング・ルールの制約を入力する方法の詳細は、“ビジネス・ルールの開発”の“[ルール制約エディタの使用](#)”を参照してください。
4. 新しいルールを作成したら、[条件]と[アクション]を追加できます。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](#)”の章を参照してください。各ルールに[条件]を追加する際には、次のヒントに留意してください。

- ・ [条件] では、AND 演算子、OR 演算子の順に評価されます。
- ・ [条件] では、HL7 メッセージ・オブジェクトのプロパティを参照できます。[条件] 内では、次の例のように、特殊変数 Document は HL7 メッセージ・オブジェクトを表します。HL7 バッチ・ドキュメントについては、特殊変数 Document.Parent を使用して、親メッセージ・オブジェクトを表すことができます。

```
Document.Name
Document.Parent.DocType
Document.{PIDgrp.PV1grp.PV1:18}
Document.{PIDgrp.PID:PatientName.familylastname}
Document.{ORCgrp(1).OBRuniongrp.OBRunion.OBR:4.3}
```

注釈 過去のリリースとの下位互換性を維持するため、特殊変数 HL7 と HL7.Parent が Document と Document.Parent の代替としてサポートされています。

ドットによって、Document 変数とプロパティ名が区切られます。この名前には次のものがあります。

- クラス・プロパティのいずれか (DocType、TypeCategory、BuildMapStatus、または Name)
- 次のいずれかの規則を使用して参照される仮想プロパティ
  - ・ [中かっこ](#)

```
{segmentPath:field}
```
  - ・ [角かっこ](#)

```
[segmentName:field]
```
  - ・ [丸かっこ](#)

```
(multi-valued-property-path)
```
  - ・ [山かっこ](#)

```
<context|expression>
```

詳細は、“プロダクション内での仮想ドキュメントの使用法”の[構文ガイド](#)の節を参照してください。ただし、手動でプロパティを入力する必要はありません。“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](#)”の説明に従ってプロパティを参照できます。

5. 次のいくつかのトピックで説明する手順を完了して、ルーティング・ルール・セットに必要な [ソース] 項目、[ターゲット] 項目、または [変換] 項目を作成します。これらの項目は次のいずれかにすることができます。
  - ・ メッセージをルールにルーティングする [HL7 ビジネス・サービス](#)
  - ・ メッセージを送信する前に変換する [DTL データ変換](#)
  - ・ メッセージを外部アプリケーションにルーティングする [HL7 ビジネス・オペレーション](#)
  - ・ [HL7 ルーティング・プロセス](#) ([ソース] にすることも、[ターゲット] にすることも可能)

項目を作成したら、メッセージ・ルーティング・ルール・エディタに戻って、ルール定義の該当フィールドに追加します。

6. **【プロダクション構成】** ページのダイアグラムに戻ります。対応する [HL7 ルーティング・プロセス](#) を選択します。[BusinessRuleName] フィールドに、新しいルーティング・ルール・セットの完全名を入力します。





# 5

## HL7 用の DTL データ変換の定義

各インタフェースでは、いくつかのデータ変換が必要になることがあります。変換を作成する場合は、予約パッケージ名を使用しないでください。“[プロダクションの開発](#)”の“[予約パッケージ名](#)”を参照してください。

重要 データ変換内の HL7 エスケープ・シーケンスを手動で変更しないでください。これらは自動的に処理されます。

データ変換を作成するには、[データ変換ビルダ] ページを使用します（ホーム・ページから [Interoperability] → [ビルド] → [データ変換] を選択します）。このページの使用に関する一般情報は、“[DTL 変換の開発](#)”を参照してください。

以下の図は、[データ変換ビルダ] ページを示しています。

#	Action	Condition	Property	Value	Key / Transform
1	set		target.{MSH}	source.{MSH}	-
2	set		target.{MSH.9.1}	"ADT"	-
3	set		target.{MSH.9.2}	"A01"	-

次のヒントに留意してください。

- “[DTL 変換の開発](#)”を手元に置いてください。このドキュメントでは、各種の DTL アクションを追加する方法を説明しています。

- データ変換クラスの **create** オプションに指定する値を明確に理解してください。create には、次のいずれかの値を指定できます。
  - new** – データ変換内の要素を実行する前に、ターゲット・タイプの新しいオブジェクトを作成します。ターゲット・オブジェクトに明示的に割り当てていないソース・セグメントは無視されます。これがデフォルトです。
  - copy** – 変換内の要素を実行する前に、ターゲット・オブジェクトとして使用するソース・オブジェクトのコピーを作成します。
  - existing** – データ変換の呼び出し側から指定された既存のオブジェクトをターゲット・オブジェクトとして使用します。

ソースの完全コピーであるターゲット・オブジェクトを作成する場合、次のようなアクションを使用しないでください。

```
<assign property='target' value='source' />
```

代わりに、データ変換クラスで **create='copy'** 属性を使用します。

- データ変換クラスによって、次の属性の正しい **スキーマ・カテゴリ** が特定されることを確認します。
  - sourceDocType 属性
  - targetDocType 属性

**スキーマ・カテゴリ** は、ソース・オブジェクトとターゲット・オブジェクトで同じ場合もあれば、異なる場合もあります。

- そのデータ変換内のすべての式および code アクションに使用するスクリプト language を **[変換]** タブで指定していることを確認します。ObjectScript がデフォルトの言語です。
  - assign アクションを使用して、ドラッグ・アンド・ドロップ操作によってソース・メッセージからターゲット・メッセージに HL7 セグメントを割り当てます。
- 場合によっては、複数の手法を組み合わせで使用します。まず、ドラッグしてコード行を生成した後、テキストを編集してコードを微調整することができます。
- データ変換では、次のような HL7 メッセージ・オブジェクトのプロパティを参照できます。
    - クラス・プロパティ **DocType**、**TypeCategory**、**BuildMapStatus**、および **Name**。
    - “プロダクション内での仮想ドキュメントの使用法” で **構文ガイド** の節の “**中かっこ {} の構文**” に記載されている仮想ドキュメント・プロパティ。

データ変換クラスのコード内では、次の例のように、特殊変数 **source** と **target** はそれぞれの HL7 メッセージ・オブジェクトを表します。

```
source.Name
target.DocType
source.{PIDgrp.PV1grp.PV1:18}
target.{PIDgrp.PID:PatientName.familylastname}
source.{ORCgrp(1).OBRuniongrp.OBRunion.OBR:4.3}
```

- 任意のリテラル文字列値を割り当てるか、条件付きの割り当てを行います。“DTL 変換の開発” の “**構文ルール**” の章を参照してください。

**注釈** 文字列リテラルに XML 予約文字を含めることはできません。HL7 で使用される区切り文字を含めることもできません。

また、HL7 の NULL マッピング・コード "" は特殊な処理を必要とします。次の例は、ソース・メッセージで NULL マッピング・コードを確認し、それをターゲット・メッセージで空白の文字列に置き換えます。

```
<if condition='source.{PV1:7().4}=""'>
<true>
<assign property='target.{PV1:7().4}' value='' />
</true>
</if>
```

詳細は、“追加の手順”の章の“[NULL マッピング・コード](#)”を参照してください。

- ・ 単純な計算については、DTL データ変換で次のことが可能です。
  - ObjectScript ユーティリティ関数の呼び出し
  - DTL [code](#) アクションでの ObjectScript 式の使用
  - DTL [sql](#) アクションの呼び出し

より複雑な計算については、独自のクラス・メソッドを記述し、[code](#) アクションから、または別の DTL 要素内の値文字列からそれらのメソッドを呼び出すことができます。

- ・ 変換と各アクションの説明を追加します。
- ・ データ変換をコンパイルすると、同時に保存も行われます。
- ・ DTL データ変換をプロダクションで使用するには、[ルーティング・ルール・セット](#)の[変換]フィールドにそのパッケージとクラスの完全名を入力するだけです。

## 5.1 長いセグメント・フィールドの変換

DTL 変換で使用する ObjectScript メソッド GetValueAt は、3.6 MB で HL7 セグメント・フィールドを切り捨てます。したがって、3.6 MB よりも長いフィールドを変換する場合、DTL エディタでは左から右へのドラッグ操作は使用できません。例えば、OBX:5 フィールドが 3.6 MB を超える場合、切り捨てられることになるため、DTL エディタを使用してこのソース・フィールドをターゲットにドラッグすることはできません。同様に、3.6 MB より長いフィールドを変換する場合、カスタム・コードで GetValueAt を呼び出すことはできません。

3.6 MB よりも長いフィールドを変換するには、Code アクションを使用して、カスタム・コードを変換に追加する必要があります。このカスタム・コードは、EnsLib.HL7.Segment クラスの GetFieldStreamRaw()、GetFieldStreamUnescaped()、または GetFieldStreamBase64() のいずれかのメソッドを含み、ソースのフィールド値をストリームに読み込む必要があります。

これらの Get メソッドは、ストリーム出力引数、フィールドの VDoc パス、および pRemainder 出力引数の 3 つの引数を取ります。pRemainder 引数には、抽出されたフィールドの後に続くすべてのフィールドの値が入ります。例を以下に示します。

```
/// Segment:  OBX|1|2|3|4|5|6|7
do GetFieldStreamRaw(.stream, "OBX:5", .rem)
/// rem contains: |6|7
```

ストリームにフィールド値が入ると、カスタム・コードは EnsLib.HL7.Segment のメソッド StoreFieldStreamRaw()、StoreFieldStreamUnescaped()、または StoreFieldStreamBase64() のいずれか 1 つを使用して、ターゲットに値を格納する必要があります。

これらの Store メソッドは、フィールドに格納するストリーム、ストリームを格納するフィールドの VDoc パス、および pRemainder の 3 つの引数を受け入れます。pRemainder 引数が指定されていない場合は、格納されるフィールドの後に続くターゲット・フィールドはすべて削除されます。例を以下に示します。

```
/// Before: OBX|1|2|3|4|5|6|7
do StoreFieldStreamRaw(stream, "OBX:5")
/// After: OBX|1|2|3|4|<stream>
/// |6|7 are gone because a remainder was not specified.
```

pRemainder を指定すると、格納されるフィールドに続くすべてのフィールドが pRemainder の指定内容に置き換えられます。

```
/// Before: OBX|1|2|3|4|5|6|7
do StoreFieldStreamRaw(stream, "OBX:5", "|six|seven")
/// After: OBX|1|2|3|4|<stream>|six|seven
```

**重要** Store メソッドが呼び出されると、ターゲット・セグメントは変更できなくなるため、これがセグメントに対する最後の変更になります。

以下の例は、ソースからフィールドを抽出し、ターゲットに格納する方法を示しています。この例では Store メソッドを呼び出す前に、3.6 MB を超えるフィールドに続くターゲット・フィールドに対してカスタム・コードが編集を加えたことを前提としています。Store メソッドの呼び出し後には、このセグメントは変更できなくなるため、この手法が必要です。サンプル・コードはソースから残りの部分を取得せず、これをターゲットに格納します。これは、ターゲットの長いフィールドの後のフィールドに対して既に行われた編集を元に戻すためです。したがって、ストリーム・フィールドはソースから取得しますが、残りの部分はターゲットから取得し、その両方をターゲットに格納します。

```
/// previous code makes edits to fields that come after OBX:5 in the target
do source.GetFieldStreamRaw(.stream, "OBX:5")
do target.GetFieldStreamRaw(.dummy, "OBX:5", .rem)
do target.StoreFieldStreamRaw(stream, "OBX:5", rem)
///Segment is now immutable
```

このようにして、OBX:5 の後続のターゲット・フィールドに対して行われた編集が元に戻されたり、削除されるのを防ぎます。

## 5.2 NULL マッピング・コード

NULL マッピング規則を使用する HL7 アプリケーションがあります。この規則に従って、このフィールドにデータがある場合、2 つの連続する二重引用符 (" ") で構成されるフィールドをソース・アプリケーションが送信して、アプリケーションから削除することを示すことができます。

多くのターゲット・アプリケーションはこのような指示を想定しておらず、これに対応するように設計されていません。このような場合、二重引用符がターゲット・アプリケーションで実際の患者データとして保存されると、アプリケーション・ユーザの画面に二重引用符が表示されます。これは厄介で誤解を招く可能性があります。

ソース・アプリケーションが NULL マッピング規則を使用している場合、HL7 データ変換によって HL7 フィールドの NULL マッピング・エントリをチェックし、それらを空の文字列に置き換えるか、ターゲット・アプリケーションに適したデータを入力することができます。

次の <if> 文は、最も単純な例を表しています。この例は、ソースで NULL マッピングをチェックし、それをターゲットで空の文字列に置き換えます。<if> condition は、引用符で囲まれた 2 つの引用符で構成される文字列を使用して、NULL

マッピング・コード "" をテストします。この結果、二重引用符が合計 6 つになります。condition 値全体を囲む一重引用符は含みません (慎重に数えてください)。

```
<if condition='source.{PV1:7().4}=""'>
<true>
<assign property='target.{PV1:7().4}' value='' />
</true>
</if>
```

上の例では、<assign> value は、2 つの連続する二重引用符を使用し、value 全体を一重引用符で囲んで空の文字列を示しています。

次の構文も同様に有効です。

```
<if condition='source.{PV1:7().4}=""'>
<true>
<assign property='target.{PV1:7().4}' value='' />
</true>
</if>
```

NULL マッピングの処理について、より高度な目標を達成することもできます。次の例は、{PV1:3} に実際に値があるかどうかに基づいて、別のアクションを実行します。フィールドに NULL マッピング・コードが含まれている場合、<true> 要素が実行されます。それ以外の場合は、<false> 要素が実行されます。

## XML

```
<if condition='source.{PV1:3}=""'>
<true>
<assign property='target.{PV1:3.1}' value='source.{PV1:PatientType}' />
<assign property='target.{PV1:3.2}' value='source.{PV1:PatientType}' />
</true>
<false>
<code>
// Dr Chart pulls subfields as follows:
// 1 location, 2 desc, 3 room, 4 bed, 5 wing, 6 floor

</code>
<assign property='target.{PV1:3.1}' value='source.{PV1:3.1}' />
<assign property='target.{PV1:3.2}' value='source.{PV1:3.1}' />
<assign property='target.{PV1:3.3}' value='source.{PV1:3.2}' />
<assign property='target.{PV1:3.4.1}' value='source.{PV1:3.3}' />
<assign property='target.{PV1:3.5}' value='source.{PV1:3.1}' />
</false>
</if>
```



# 6

## HL7 検索テーブルの定義

HL7 検索テーブル・クラスの `EnsLib.HL7.SearchTable` は、一般的な HL7 プロパティのインデックスを自動的に作成します。“[デフォルトでインデックスが作成されるプロパティ](#)”を参照してください。

検索する項目を増やす場合は、サブクラスを作成できます。サブクラスは、`Identifier` プロパティに加えて、検索テーブルを機能させるインフラストラクチャを継承します。詳細は、“[プロダクション内での仮想ドキュメントの使用法](#)”の“[検索テーブル・クラスの定義](#)”を参照してください。

HL7 については、インターシステムズは `PropType` の追加の値をサポートしています。“[プロダクション内での仮想ドキュメントの使用法](#)”に列挙されたタイプのほかに、`DateTime:HL7` を使用できます。

検索テーブルを作成するときに、予約パッケージ名を使用しないでください。“[プロダクションの開発](#)”の“[予約パッケージ名](#)”を参照してください。

### 6.1 デフォルトでインデックスが作成されるプロパティ

`EnsLib.HL7.SearchTable` を検索テーブル・クラスとして選択すると、HL7 メッセージで次の仮想プロパティを検索できます。

選択するプロパティ	参照する値
MSHTypeName	<p>メッセージ構造名。この文字列を作成するために、インターシステムズでは HL7 メッセージ内の次の値が連結されます。</p> <ul style="list-style-type: none"><li>MSH メッセージ・ヘッダ・セグメント フィールド 9 (メッセージ・タイプ) サブフィールド 1 (メッセージ・タイプ : ADT、ORM など)</li><li>リテラル文字 _</li><li>MSH メッセージ・ヘッダ・セグメント フィールド 9 (メッセージ・タイプ) サブフィールド 2 (トリガ・イベント : A01、A12、O01_2 など)</li></ul> <p>結果は、ADT_A01、ADT_A12、ORM_O01_2 といった形式のメッセージ構造名です。</p>

選択するプロパティ	参照する値
MSHControlID	<p>メッセージの一意の識別子番号。プロダクションは、この値を次の値から取得します。</p> <ul style="list-style-type: none"> <li>MSH メッセージ・ヘッダ・セグメント フィールド 10 (メッセージ制御 ID)</li> </ul> <p>インターシステムズでは、この値は大文字/小文字を区別する文字列として解釈されます。</p>
PatientID	<p>メッセージの患者識別子。これは、HL7 標準の改訂に伴って位置が変わったフィールドです。そのため、インターシステムズでは、次の場所すべてでこの値が検索されます。そのようにすることで、メッセージが準拠する HL7 スキーマ・カテゴリに関係なく、患者識別子を見つけることができます。</p> <ul style="list-style-type: none"> <li>PID 患者識別子セグメント フィールド 2 (患者外部識別子) サブフィールド 1 (患者識別子)</li> <li>PID 患者識別子セグメント フィールド 3 (患者識別子リスト)、リスト内のすべてのエントリ サブフィールド 1 (患者識別子)</li> <li>PID 患者識別子セグメント フィールド 4 (患者識別子リスト)、リスト内のすべてのエントリ サブフィールド 1 (患者識別子)</li> </ul>
PatientName	<p>PID 患者識別子セグメント フィールド 5 (患者名)</p>
PatientAcct	<p>PID 患者識別子セグメント フィールド 18 (患者アカウント番号) サブフィールド 1 (ID)</p>

## 6.2 例

次の例は、`{}` 構文を使用する 1 つの仮想プロパティ・パスで構成されています。この `<Item>` 要素は、HL7 メッセージのセグメント 1、フィールド 10 の値を参照します。

### XML

```
<Item DocType=" "
  PropName="MSHControlID"
  PropType="String:CaseSensitive"
  StoreNulls="true" >
  {1:10}
</Item>
```



次に示す、より複雑な <Item> 要素は、ObjectScript の \_ 演算子を使用して 3 つの文字列を連結します。左から右へ、次のように連結されます。

- ・ セグメント 1、フィールド 4 の値
- ・ リテラルの - 文字
- ・ セグメント 1、フィールド 3 の値

## XML

```
<Item DocType=" "
      PropName="SendingFacilApp" >
      {1:4}_"-_{1:3}
</Item>
```

次の <Item> の例では、使用可能な構文オプションのほとんど（連結、仮想プロパティ、リテラルのハイフン文字 (-)、および ObjectScript 文字列関数 \$PIECE) を使用しています。

## Class Member

```
XData SearchSpec [ XMLNamespace="http://www.intersystems.com/EnsSearchTable" ]
{
<Items>
  <Item DocType="Mater:ORM_001 "
        PropName="RelationKey" >
    $P(
      {ORCgrp(1).OBRuniongrp.OBRunion.OBR:UniversalServiceID.text}, "-", 1, 2
    )_"-_{MSH:12}
  </Item>
</Items>}
```

次の検索テーブル・クラスのサンプルでは、有効な <Item> エントリの例をいくつか示します。このクラスは、HL7 検索テーブルの規定に従って、**EnsLib.HL7.SearchTable** から継承されます。<Item> エントリの各グループの上にあるコメントには、その一連のエントリの目的が記載されています。{} や [] の構文の詳細は、“プロダクション内での仮想ドキュメントの使用法”で [構文ガイド](#) の節を参照してください。

## Class Definition

```
Class Demo.HL7.MsgRouter.SearchTable Extends EnsLib.HL7.SearchTable
{
XData SearchSpec [ XMLNamespace="http://www.intersystems.com/EnsSearchTable" ]
{
  <Items>
    <!-- Items that do not depend on DocType, indexing any HL7 message -->
    <Item DocType=" " PropName="SendingFacilApp" >{1:4}_"-_{1:3}</Item>
    <Item DocType=" " PropName="RecvingFacilApp" >{1:6}_"-_{1:5}</Item>
    <Item DocType=" " PropName="MSHDateTime" PropType="DateTime:HL7" >{1:7}</Item>

    <!-- Get fields from named segments found in any HL7 message -->
    <Item DocType=" " PropName="PatientName" >[PID:5]</Item>
    <Item DocType=" " PropName="InsuranceCo" >[IN1:4]</Item>

    <!-- Get patient name from any HL7 message declared type ADT_A05 -->
    <Item DocType="ADT_A05" PropName="PatientName" >{3:5}</Item>

    <!-- Get specific field from specific segment when the -->
    <!-- HL7 message is assigned a specific DocType. Only in this -->
    <!-- case can you use names for segments, instead of numbers. -->
    <Item DocType="Demo.HL7.MsgRouter.Schema:ORM_001 " PropName="ServiceId" >
      {ORCgrp().OBRuniongrp.OBRunion.OBR:UniversalServiceID.text}
    </Item>
    <Item DocType="2.3.1:ORU_R01 " PropName="ServiceId" >
      {PIDgrpgrp().ORCgrp(1).OBR:UniversalServiceID.text}
    </Item>
  </Items>
}
}
```



# 7

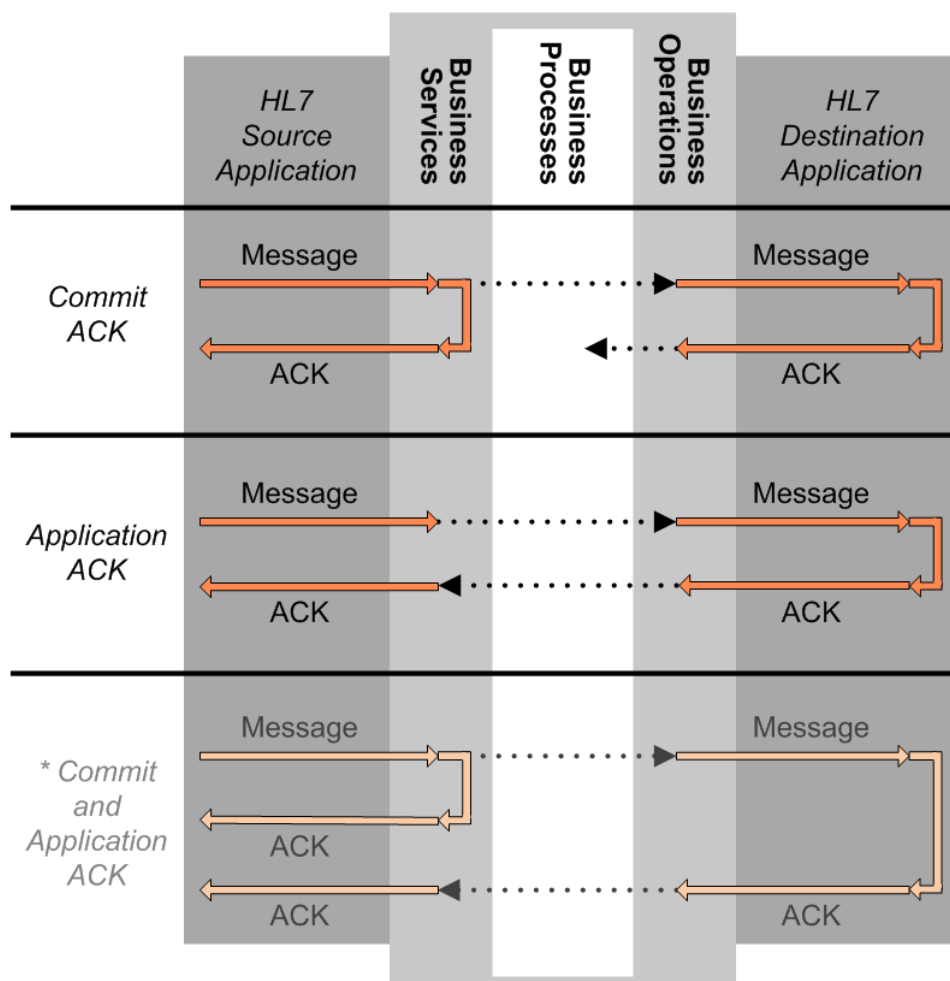
## HL7 の重要なシナリオ

この章では、構成設定の選択に影響するいくつかの重要な HL7 シナリオについて説明します。

### 7.1 HL7 確認応答 (ACK) モード

HL7 確認応答 (ACK) メッセージは、宛先が HL7 メッセージを受け取ったという通知です。否定 ACK (NACK) メッセージは、宛先が送信を認識しているものの、メッセージを受け取らなかったという通知です。

次の図は、ACK メッセージと NACK メッセージの送信に関する HL7 メッセージの規則にプロダクションが準拠する仕組みを示しています。



コミット ACK – ビジネス・サービスは、ソース・アプリケーションから受け取ったデータを保存するトランザクションをコミットするとすぐに、そのソースに ACK を返します。何らかの理由でこれが不可能な場合は、NACK を送信します。ビジネス・オペレーションは、ターゲット・アプリケーションからの ACK または NACK を解釈するように設定できますが、それらのメッセージをソースに返すことはありません。

アプリケーション ACK – ビジネス・サービスは、ビジネス・オペレーションを経由してターゲット・アプリケーションから返されるまで、ACK または NACK をソース・アプリケーションに送信しません。ビジネス・サービスは、ビジネス・オペレーションから受け取った ACK または NACK を返します。

\* コミットおよびアプリケーション ACK – 3 番目のオプションは、MSH によって決定するように **[Ackモード]** を設定し、ACK を必要とする値が MSH セグメントのフィールド 15 と 16 の両方に含まれているというまれな状況でのみ発生します。このような状況は回避することをお勧めします。

これらの確認応答規則を使用して、次の 3 種類の主な ACK メッセージ・コンテンツのいずれかを送信します。

- ・ 承認 – メッセージは到着し、受け取られました。
- ・ 拒否 – メッセージは到着しましたが、拒否されました。
- ・ エラー – メッセージは正常に到着しませんでした。再試行してください。

ビジネス・サービスの次の構成設定を使用して、ACK 処理を制御します。

- ・ [Ackターゲット構成名](#)
- ・ [Ackモード](#)
- ・ [Ackコミットコード使用](#)

- ・ インバウンドACKを無視
- ・ NACK ERR追加
- ・ NACKエラーコード

ビジネス・オペレーションの次の構成設定を使用して、ACK 処理を制御します。

- ・ リプライコードアクション
- ・ 準備完了

## 7.2 HL7 二重確認応答シーケンス

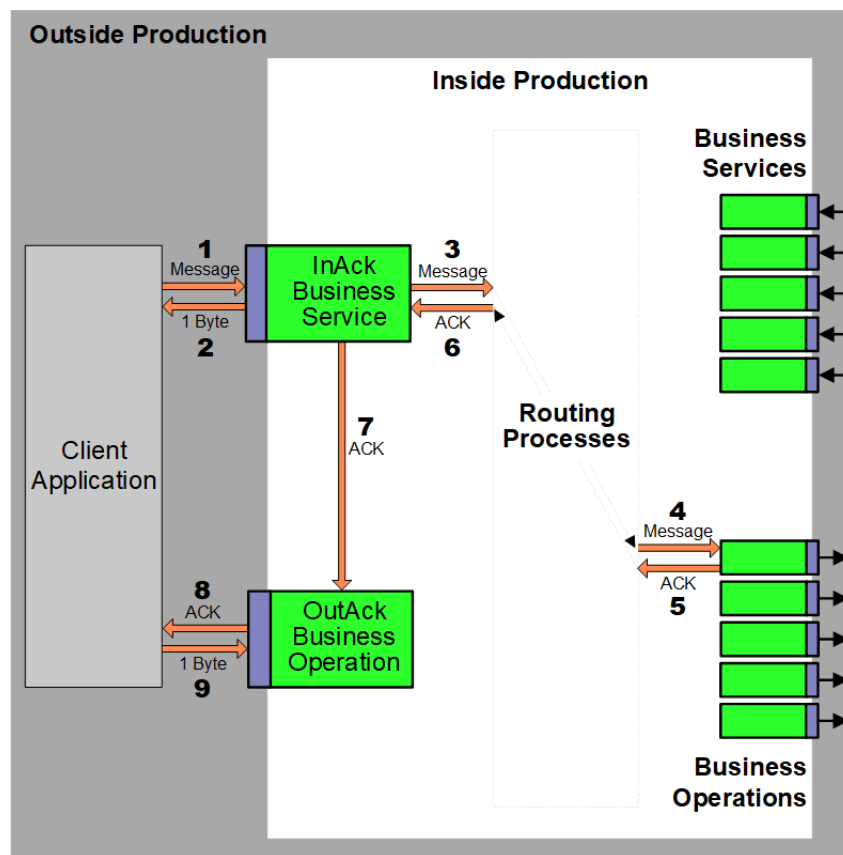
一部のシステムでは、インターシステムズ製品からの二重確認応答シーケンスが必要です。1 バイトの即時 ACK の後に、完全な ACK メッセージを送信する必要があります。このようなシステムの 1 つに、二重チャンネルの iSoft iCM アプリケーションがあります。二重確認応答シーケンスを必要とする iCM のようなクライアント・システムが構成に含まれている場合は、必要な ACK をインターシステムズ製品が提供できるようにビジネス・サービスとビジネス・オペレーションのペアを設定する必要があります。

インターシステムズ製品には、TCP と HTTP を介した二重確認応答シーケンスを定義するために使用できる特殊なビジネス・ホスト・クラスが用意されています。

- ・ **EnsLib.HL7.Service.TCPAckInService** は、ペアになっている HL7 TCP ビジネス・オペレーションに代わって ACK を受け取る特殊な HL7 ビジネス・サービスです。ACK の送信は、パートナーが代わりに行います。
- ・ **EnsLib.HL7.Operation.TCPAckOutOperation** は、ペアになっている HL7 TCP ビジネス・サービスに代わって ACK を送信する特殊な HL7 TCP ビジネス・オペレーションです。ACK の収集は、パートナーが代わりに行います。これらの構成項目はそれぞれ、パートナー項目のために行う作業に加えて、通常の役割も果たします。
- ・ **EnsLib.HL7.Service.HTTPAckInService** と **EnsLib.HL7.Operation.HTTPAckOutOperation** を使用することもできます。

### 7.2.1 受信メッセージの二重 ACK シーケンス

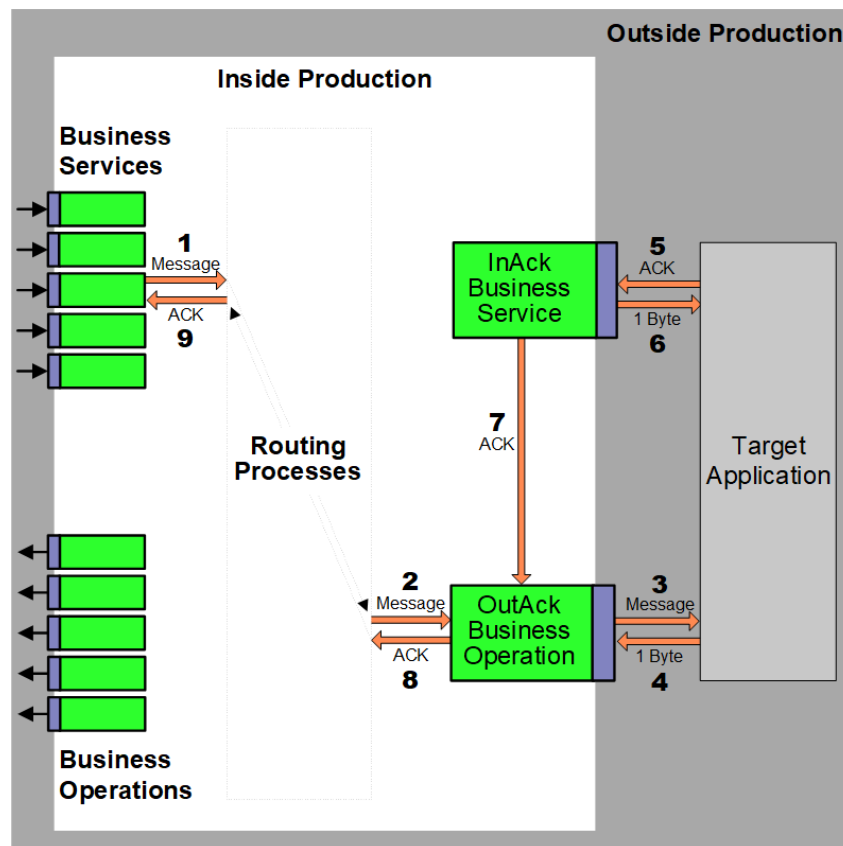
インターシステムズ製品に入ってくるメッセージの場合、二重確認応答シーケンスは次の図のように動作します。



1. クライアント・アプリケーションがインターシステムズ製品にメッセージを送信します。
2. 受信 ACK ビジネス・サービスが 1 バイトの即時 ACK をクライアント・アプリケーションに送信します。
3. 受信 ACK ビジネス・サービスがルーティング・プロセスにメッセージを送信します。
4. ルーティング・プロセスがビジネス・オペレーションを介してターゲットにメッセージをルーティングします。
5. ターゲット・アプリケーションがビジネス・オペレーションに ACK メッセージを返します。
6. プロダクションが受信 ACK ビジネス・サービスに ACK を中継します。
7. ペアになっているビジネス・オペレーションにビジネス・サービスが ACK を中継します。
8. ビジネス・オペレーションがクライアント・アプリケーションに ACK を中継します。
9. クライアント・アプリケーションが 1 バイトの ACK を返すことにより、ACK メッセージの受信を通知します。

## 7.2.2 送信メッセージの二重 ACK シーケンス

インターシステムズ製品から出ていくメッセージの場合、二重確認応答シーケンスは次の図のように動作します。



1. ビジネス・サービスがルーティング・プロセスにメッセージを送信します。
2. ルーティング・プロセスが送信 ACK ビジネス・オペレーションにメッセージをルーティングします。
3. 送信 ACK ビジネス・オペレーションがターゲット・アプリケーションにメッセージを中継します。
4. ターゲット・アプリケーションが 1 バイトの ACK を返すことにより、メッセージの受信を通知します。
5. ターゲット・アプリケーションが受信 ACK ビジネス・サービスに ACK メッセージを返します。
6. ビジネス・サービスが 1 バイトの即時 ACK をターゲット・アプリケーションに送信します。
7. ペアになっているビジネス・オペレーションにビジネス・サービスが ACK を中継します。
8. ビジネス・オペレーションがビジネス・サービスに ACK メッセージを中継して返します。
9. ビジネス・サービスが元のメッセージに対する ACK を受け取ります。

## 7.2.3 二重 ACK シーケンスの構成

二重確認応答機能を使用するようにルーティング・プロダクションを構成するには、以下の操作を行います。

1. ビジネス・サービスをプロダクションに追加します。

EnsLib.HL7.Service.TCPAckInService または EnsLib.HL7.Service.HTTPAckInService をビジネス・サービス・クラスとして選択します。これらは、ビジネス・サービス・ウィザードから使用可能な標準の [HL7入力] オプションには含まれていませんが、[その他] オプションを選択し、該当するクラスを探して選択することができます。

2. ビジネス・オペレーションをプロダクションに追加します。

EnsLib.HL7.Operation.TCPAckOutOperation または EnsLib.HL7.Service.HTTPAckOutOperation をビジネス・オペレーション・クラスとして選択します。これらは、ビジネス・オペレーション・ウィザードから使用可能な標準の [HL7

**出力** オプションには含まれていませんが、**[その他]** オプションを選択し、該当するクラスを探して選択することができます。

### 3. ビジネス・サービスについて次の設定を構成します。

- ・ **[即時 Byte ACK]** を真に設定します。これにより、ビジネス・サービスは、**[Ackモード]** の設定に従って完全な ACK メッセージを転送するだけでなく、その TCP 接続または HTTP 接続で 1 バイトの即時 ACK を返します。
- ・ **[パートナーオペレーション]** として、手順 2 で追加したビジネス・オペレーションを選択します。**[パートナーオペレーション]** の値を指定すると、ACK フィードバック・ループが生成されないように、ビジネス・サービスは、直接受け取る受信 ACK メッセージを無視します。

このビジネス・オペレーションが存在し、基盤のクラスがそれぞれ `EnsLib.HL7.Operation.TCPAckOutOperation` または `EnsLib.HL7.Operation.HTTPAckOutOperation` である必要があります。

### 4. ビジネス・オペレーションについて次の設定を構成します。

- ・ ビジネス・オペレーションが送信した通常の送信メッセージに対応する ACK をパートナー・ビジネス・サービスから受け取るまでビジネス・オペレーションが待機する秒数に **[パートナーACKタイムアウト]** を設定します。デフォルトは 600 秒 (10 分) です。

## 7.3 HL7 バッチ・メッセージ

インターシステムズ製品は、HL7 の入れ子になった子ドキュメント (バッチ形式) をサポートしています。子ドキュメントはそれぞれ、それ自体で 1 つの仮想ドキュメントです。ここでは、2 つの設定によって制御される詳細について説明します。以下のセクションで構成されます。

- ・ [サポートされているバッチ形式](#)
- ・ [受信バッチ・ドキュメントの処理](#)
- ・ [バッチ・メッセージの送信](#)
- ・ [バッチ・モード](#)
- ・ [カスタム送信バッチ処理](#)

### 7.3.1 サポートされているバッチ形式

インターシステムズ製品では、次の HL7 バッチ形式がサポートされています。

- ・ BHS MSH ...MSH ...BTS

BHS はバッチ・ヘッダ・セグメントとして、BTS はバッチ・トレーラ・セグメントとして認識されます。このコンテナ内では、各 MSH メッセージ・ヘッダ・セグメントは子ドキュメントの先頭として認識されます。

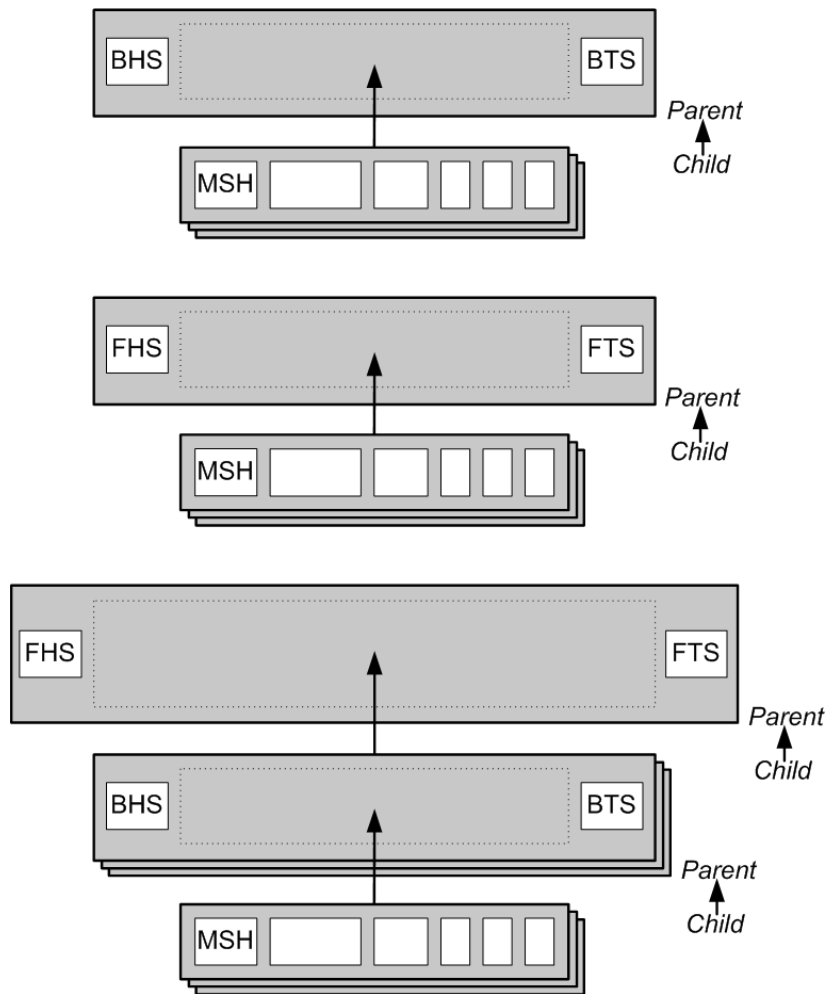
- ・ FHS MSH ...MSH ...FTS

FHS はバッチ・ヘッダ・セグメントとして、FTS はバッチ・トレーラ・セグメントとして認識されます。このコンテナ内では、各 MSH メッセージ・ヘッダ・セグメントは子ドキュメントの先頭として認識されます。

- ・ FHS BHS MSH ...MSH ...BTS BHS MSH ...MSH ...BTS FTS

FHS と BHS が共にメッセージを開始する場合、FHS は第 1 レベルの親ドキュメントとして、各 BHS は子ドキュメントの先頭として認識されます。これにより、BHS は第 2 レベルの親になり、各 MSH セグメントに続くコンテンツはその子ドキュメントになります。





### 7.3.2 受信バッチ・ドキュメントの処理

HL7 ビジネス・サービスの [\[バッチハンドリング\]](#) 構成設定では、受信バッチ・ドキュメントを処理する方法を指定できます。オプションは以下のとおりです。

- ・ [ ] – 子ドキュメントを個別に処理しません。累積してバッチ全体を 1 つの複合ドキュメントとして送信します。
- ・ [1] – バッチ内のすべての子ドキュメントを 1 つのセッションでまとめて転送します。セッションには、バッチ・ヘッダとバッチ・トレーラを表すオブジェクトが含まれます。[\[バッチハンドリング\]](#) の値が指定されていない場合、[1] がデフォルトです。
- ・ [ ] – バッチ内の各子ドキュメントを、一意のセッション ID を持つ独自のセッションで転送します。
- ・ [ ] – バッチ内の各子ドキュメントを独自のセッションで転送します。バッチ・ヘッダとバッチ・トレーラを表すオブジェクトは転送しません。

受信したメッセージに応答する際のデフォルトの動作は、各子ドキュメントの ACK メッセージを含むバッチ・ドキュメントとして確認応答を送信者に送信するというものです。この動作は、ほとんどの状況で有効です。ただし、HL7 ビジネス・サービスには、構成設定として公開されていない NoBatchReply というプロパティもあります。そのデフォルト値は 0 (偽) で、デフォルトの動作になります。次の文を含むようにビジネス・サービスの OnInit() メソッドを編集するとします。

```
Set ..NoBatchReply = 1
```

この場合、バッチ応答は禁止されます。各メッセージは、ラップされていない別個の ACK を取得します。また、次のコードを挿入することにより、サブクラスでプロパティの定義をオーバーライドして、値を設定することもできます。

```
Property NoBatchReply As %Boolean [ InitialExpression = 1 ];
```

### 7.3.3 バッチ・メッセージの送信

送信側では、HL7 [ファイル](#) および [FTP](#) ビジネス・オペレーションに [\[親セグメント自動バッチ\]](#) 構成設定があります。[\[親セグメント自動バッチ\]](#) が偽 (デフォルト) の場合、ビジネス・オペレーションは子ドキュメントを出力しますが、バッチ・ヘッダおよびトレーラは出力しません。[\[親セグメント自動バッチ\]](#) が真の場合、バッチの親を持つメッセージを出力する際、ビジネス・オペレーションはバッチ・ヘッダ、子ドキュメントの順に出力し、最後のバッチ・ヘッダ・メッセージまたはファイル名の変更によりトリガされると、バッチ・トレーラを出力します。

### 7.3.4 バッチ・モード

[\[バッチハンドリング\]](#) と [\[親セグメント自動バッチ\]](#) の組み合わせにより、HL7 バッチ・ドキュメント向けのオペレーションの以下のモードが有効になります。

<a href="#">[バッチハンドリング]</a>	<a href="#">[親セグメント自動バッチ]</a>	結果
[       ]	(いずれか)	ビジネス・サービスは親ドキュメントのみを送信します。子ドキュメントはすべて参照されますが、個々に送信されることはありません。オペレーションでは、親ドキュメントを受信すると、一度にバッチ全体を出力します。
[ 1       ] または [       ]		サービスでは、各子ドキュメントを受信および解析すると、これを送信します。すべての子ドキュメントが送信されると親ドキュメントを送信します。ビジネス・オペレーションは、最初に子ドキュメントを受信すると親ヘッダを出力し、親ドキュメント・オブジェクトを受信すると、トレーラで終了します。トレーラ・セグメントには自動的に正しい子カウント値が含まれます。
[ 1       ] または [       ]		これにより重複した出力が得られます。ビジネス・オペレーションは、各子ドキュメントを個々に送信した後、同じ子ドキュメントを含む親ドキュメントを送信するので、子ドキュメントが 2 回送信されます。
[       ]		ビジネス・サービスでは、バッチ内の各子ドキュメントを個別のセッションで転送します。バッチ・ヘッダおよびトレーラを表すオブジェクトは転送されません。送信側のビジネス・オペレーションの動作も同じです。

### 7.3.5 カスタム送信バッチ処理

出力側でバッチ・ドキュメントを特別に処理できるように、ルーティング・プロセスにカスタム・コードを追加することもできます。次の 2 つの方法があります。

- ・ ルーティング・プロセス・コードで新しい親ドキュメントと子ドキュメントを作成し、それらをリンクした後、それぞれの子をビジネス・オペレーションに送信します。ビジネス・オペレーションの [\[親セグメント自動バッチ\]](#) が真に設定されている必要があります。ビジネス・オペレーションは、最初に子ドキュメントを受信すると親ヘッダを出力し、親ドキュメント・オブジェクトを受信すると、トレーラで終了します。トレーラ・セグメントには自動的に正しい子カウント値が含まれます。

- ・ ルーティング・プロセス・コードで新しい親ドキュメントと子ドキュメントを作成しリンクしますが、親オブジェクトのみをビジネス・オペレーション経由で送信します。



# 参照

## HL7 ビジネス・サービスに関する設定

HL7 ビジネス・サービスの設定に関する参照情報を提供します。

### 概要

HL7 ビジネス・サービスには次のような設定があります。

グループ	設定	参照先
基本設定	<a href="#">[ターゲット構成名]</a>	“プロダクション内での仮想ドキュメントの使用法”の <a href="#">“ビジネス・サービスに関する設定”</a>
基本設定	<a href="#">[Ackターゲット構成名]</a> 、 <a href="#">[メッセージ・スキーマ・カテゴリ]</a>	このトピックの節
接続設定	<a href="#">[構成]</a>	このトピックの節
追加設定	<a href="#">[テーブルクラス検索]</a>	“プロダクション内での仮想ドキュメントの使用法”の <a href="#">“ビジネス・サービスに関する設定”</a>
追加設定	Override Segment Terminator、ローカル・ファンリティ・アプリケーション、Ackモード、Ackコミットコード使用、インバウンドACKを無視、NACK ERR追加、NACKエラーコード、バッチハンドリング、デフォルト文字エンコーディング、DocTypeResolution、応答の保存	このトピックの節

残りの設定は、すべてのビジネス・サービスに共通のものであるか、アダプタのタイプによって異なるものであるかのどちらかです。詳細は、以下を参照してください。

- ・ “プロダクションの構成”の[“すべてのビジネス・サービスに含まれる設定”](#)
- ・ “プロダクション内でのファイル・アダプタの使用法”の[“ファイル受信アダプタに関する設定”](#)
- ・ “プロダクション内での HTTP アダプタの使用法”の[“HTTP 受信アダプタの設定”](#)
- ・ “プロダクション内での FTP アダプタの使用法”の[“FTP 受信アダプタに関する設定”](#)

EnsLib.HL7.Adapter.TCPInboundAdapter という特殊用途のアダプタの違いは、[\[JobPerConnection\]](#) が偽に設定されている点です。これは通常、HL7 に適しています。

HL7 に関する最も重要な設定は次のとおりです。

- ・ [\[プール・サイズ\]](#) – デフォルト値 1 では、FIFO (先入れ先出し) 処理をサポートできます。多くの場合、複数の患者基本情報の更新を順番に受信する必要があります。例えば、多くのアプリケーションでは、オーダー・メッセージを処理する前に ADT 登録メッセージを受信する必要があります。また、結果メッセージの前にオーダー・メッセージを受信する必要があります。
- ・ [\[カテゴリ\]](#) – このテキスト・ラベルにより、構成ダイアグラム内で構成項目をソートできます。
- ・ [\[タイムスタンプ追加\]](#) – (ファイルのみ) [\[アーカイブパス\]](#) のファイル名にタイム・スタンプを追加します。
- ・ [\[アーカイブパス\]](#) – (ファイルおよび FTP のみ) HL7 メッセージをアーカイブする場所を指定します。
- ・ [\[呼び出し間隔\]](#) – 新しい入力を確認するまで待機する秒数。デフォルトは 5 秒です。最小値は 0.1 秒です。

## Ackモード

受信した HL7 メッセージへの応答として HL7 確認応答メッセージを発行する際の形式と規則を設定できます。ビジネス・サービスについては、次のテーブルに示されている値のいずれかを指定することができます。

Ackモード	意味
[なし]	ACK を返しません。
[即時]	受信メッセージを受け取るとすぐに、コミット ACK 応答メッセージを返します。何も指定されていない場合は、これがデフォルトです。
[アプリケーション]	メッセージが検証に合格した場合、ターゲット・アプリケーションからの ACK 応答メッセージを待機し、ACK が到着すると、その ACK を返します。 呼び出し側が応答を要求しており、プロダクション・ルーティング・エンジンがそのターゲットからの応答を転送して返すように構成されていない場合、ACK または NACK オブジェクトを作成して呼び出し側に返します。検証に失敗し、[Ackモード] が [ ] である場合、ターゲット・アプリケーションと通信しません。代わりに、即時検証 NACK を呼び出し側に送信します。
[MSH により決定]	MSH ヘッダ・フィールド 15 および 16 で要求されたように ACK 応答メッセージを返します。どちらのフィールドにも、次の 4 つの制御コードのいずれかが含まれます。 <ul style="list-style-type: none"> <li>AL – 常時</li> <li>NE – 返さない</li> <li>ER – エラーまたは拒否が発生した場合のみ</li> <li>SU – 正常に完了した場合のみ</li> </ul> MSH 15 (AcceptAcknowledgmentType) はコミット ACK を制御し、MSH 16 (ApplicationAcknowledgmentType) はアプリケーション ACK を制御します。受信メッセージの MSH セグメントでどのように設定されているかに応じて、一方または両方が発生することもあるれば、いずれも発生しないこともあります。
[バイト]*	受信メッセージを受け取るとすぐに、ACK メッセージの代わりに、シングル・バイトの ACK コードを返します。ASCII 6 は OK を意味し、ASCII 21 はエラーを意味します。このオプションは、組み込みの HL7 ビジネス・サービス (TCP、ファイル、HTTP など) で使用することはできませんが、[Ackモード] の設定をオーバーライドせずに、EnsLib.HL7.Service.Standard のサブクラスとなるカスタム・ビジネス・サービスを記述すると使用できます。

\* ビジネス・オペレーションでは自動的に、シングル・バイトの ASCII 6 を AA コミット・コードの HL7 ACK として、ASCII 21 を AE コミット・コードの HL7 ACK としてそれぞれ処理します。

ACK 処理の全般的な説明は、このドキュメントで前述した“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## Ackターゲット構成名

(ファイルおよび FTP のみ) TCP ビジネス・サービスとは異なり、ファイルおよび FTP ビジネス・サービスには、HL7 確認応答メッセージ (ACK または NACK) を送信するための永続的な接続がありません。そのため、ファイルおよび FTP ビジネス・サービスのデフォルトの [Ackモード] は [ ] になっています。通常は、この設定が適切です。ただし、ファイルまたは FTP ビジネス・サービスから ACK を送信する場合は、[Ackターゲット構成名] 設定を使用して、ACK メッセージを受け取るルーティング・プロセスまたはビジネス・オペレーションを指定します。

詳細は、“[Ackモード](#)”を参照してください。ACK 処理の全般的な説明は、このドキュメントで前述した“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## NACK ERR追加

真の場合、NACK メッセージを生成する際に、インターシステムズのエラー・コードとエラー・テキストを含む ERR セグメントを追加します。それ以外の場合は、内部エラー状態情報を NACK メッセージに埋め込みません。

ACK 処理の全般的な説明は、このドキュメントで前述した“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## バッチハンドリング

受信したメッセージ・バッチの処理方法です。オプションは以下のとおりです。

- ・ [ ] – メッセージ・ドキュメントを個別に処理しません。累積してバッチ全体を 1 つの複合ドキュメントとして送信します。
- ・ [1] – バッチ内のすべてのメッセージを 1 つのセッションでまとめて転送します。セッションには、バッチ・ヘッダ・セグメントとバッチ・トレーラ・セグメントを表すオブジェクトが含まれます。これがデフォルトです。
- ・ [ ] – バッチ内の各メッセージを独自のセッションで転送します。各セッションには、バッチ・ヘッダ・セグメントとバッチ・トレーラ・セグメントを表すオブジェクトが含まれます。
- ・ [ ] – バッチ内の各子メッセージを独自のセッションで転送します。バッチ・ヘッダ・セグメントとバッチ・トレーラ・セグメントを表すオブジェクトは転送しません。

バッチ処理の詳細は、“[HL7 バッチ・メッセージ](#)”を参照してください。

## デフォルト文字エンコーディング

受信 HL7 メッセージの文字エンコーディングです。文字は、自動的にこのエンコーディングから変換されます。

サポートされるエンコーディング値は、UTF-8、Latin1、および InterSystems IRIS サーバにインストールされているその他の NLS 定義です。値 `Native` は、InterSystems IRIS サーバのデフォルト・エンコーディングを使用することを意味します。インターシステムズの変換テーブルを直接使用することもできます。そのためには、値 `@tablename` を使用します。tablename はテーブルの名前です。

デフォルトでは、受信 HL7 メッセージの MSH:18 (文字セット) フィールドが空でない場合、この設定の代わりにその値が使用されます。MSH:18 を無視し、代わりにこの設定を使用するようインターシステムズ製品に強制するには、設定値の先頭に ! (感嘆符) を配置します。例えば、!UTF-8 のように指定します。

デフォルト値はアダプタによって異なります。

インターシステムズ製品の文字変換に関する背景情報は、“サーバ側プログラミングの入門ガイド”の“ローカライズのサポート”を参照してください。

## DocTypeResolution

MSH:9 のメッセージ・タイプに基づいて DocType を解決する方法を指定します。以下のいずれかを選択します。

- ・ [ ] – 有効な [\[メッセージ・スキーマ・カテゴリ\]](#) の値と、対応するスキーマ・カテゴリの MSH:9 メッセージ・タイプ値について検出されたメッセージ構造名を組み合わせます。これがデフォルトです。
- ・ [9.3] – [標準]と同様ですが、MSH:9 に 3 つ以上の成分がある場合、追加成分を無視します。一部のスキーマには 3 つの部分から成るタイプ名が含まれるため、標準の動作では、下位成分がない場合にタイプ名の一部として成分 3 が使用されます。
- ・ [9.3] – [標準]と同様ですが、MSH:9 に 3 つ以上の成分がある場合、該当するスキーマ・カテゴリ内のドキュメント構造のリテラル名として追加成分を使用します。選択したスキーマ・カテゴリに構造が存在しない MSH:9.3 値を持つメッセージが到着する可能性があるため、慎重に使用してください。



- ・ [ ] – 有効な [メッセージ・スキーマ・カテゴリ] の値と、メッセージ構造の名前として解釈されたリテラル MSH:9 メッセージ・タイプ値を組み合わせます。すべてのメッセージ・タイプに対応する構造定義があるカスタム・スキーマのみで使用してください。

## Override Segment Terminator

(FTP のみ) セグメント終端文字として使用する ASCII 制御文字のコンマ区切りリスト。値は 10 進形式、または先頭に x を付けた 16 進形式にすることができます。例えば、改行文字をセグメント終端文字として指定するには、「10」または「x0A」を入力します。既定値はキャリッジ・リターンであり、その 10 進値は 13、16 進値は x0D です。

## 構成

HL7 ビジネス・サービスが受信 HL7 メッセージ・パケットを解釈する方法を制御します。使用する値がわからない場合は、HL7 ビジネス・サービスのデフォルトの [ ] フレーミングをそのまま使用してください。

次のテーブルは、この設定の有効な値のリストです。

フレーミング・タイプ	受信/送信	意味
フレキシブル	受信	受信したデータの内容からフレーミング・スタイルを決定します。
Flexible!	受信	受信した最初のメッセージのデータの内容からフレーミング・スタイルを決定します。また、後続のメッセージはこれと同じフレーミング・スタイルである必要があります。
なし	両方	フレーミングなし。文字列 MSH で始まる各行が新しいメッセージの先頭です。
MLLP	両方	Minimal Lower Level Protocol – ASCII 11 接頭語、および ASCII 28 とその後の ASCII 13 で構成される接尾語を付けて各 HL7 メッセージをフレーミングします。
MLLP[nn]/[mm]	両方	非標準の ASCII 値を使用する MLLP。nn で指定した ASCII 文字値で構成される接頭語を付けて各 HL7 メッセージをフレーミングします。また、mm で指定した ASCII 文字値とその後の ASCII 13 (キャリッジ・リターン文字) で構成される接尾語も付けます。
AsciiLF	両方	ASCII 10 (改行文字) を使用してメッセージをフレーミングし、各メッセージと後続のメッセージを区切ります。
AsciiCR	両方	追加の ASCII 13 (キャリッジ・リターン文字) を使用してメッセージをフレーミングし、各メッセージと後続のメッセージを区切ります。
Ascii[nn]	両方	接尾語を付けてメッセージをフレーミングし、各メッセージと後続のメッセージを区切ります。この接尾語は、nn で指定した ASCII 文字値で構成されます。
Ascii[nn]/[mm]	両方	各メッセージの前に接頭語を付けてメッセージをフレーミングします。この接頭語は、nn で指定した ASCII 文字値で構成されます。また、mm で指定した ASCII 文字値で構成される接尾語も付けます。ただし、末尾の ASCII 13 は含まれません。
LLP	両方	(廃止) Lower Level Protocol – 冗長チェックサム・ブロックで各 HL7 メッセージをフレーミングします。
MsgEnvelope	送信	メッセージの Envelope プロパティをそのまま使用します (存在する場合)。文字列 <!--HL72MSG--> がエンベロープに存在する場合、メッセージ・コンテンツに置き換えられます。それ以外の場合は、エンベロープ・テキストの後にメッセージが続きます。

フレーミング・タイプ	受信/送信	意味
MLLPMsgEnvelope	送信	[MsgEnvelope] と同じですが、さらにエンベロープ内で MLLP 接頭語と接尾語でメッセージを囲みます。

フレーミング・タイプが [MLLP] である場合、終了フレーミングより前にメッセージ内に出現する追加のキャリッジ・リターン (ASCII 13) が自動的に検出されます。これは、空白行がメッセージを区切るために使用されていないことを示すものであるため、空白行はメッセージ・コンテンツの一部であり、無視してもかまわないと見なされます。

HL7 標準では、セグメント終端文字はキャリッジ・リターン (CR) です。ただし、キャリッジ・リターン/改行 (CRLF) 文字も許可されています。

複数の文字を指定できます。例えば、メッセージの先頭に \$Char(2)、メッセージの末尾に \$Char(3,4) など、非標準のフレーミングが HL7 メッセージに必要な場合、[Ascii[nm]/[mm]] フレーミング・オプションを次のように使用できます。

Ascii2/3,4

注釈     **【構成】** フィールドに ASCII 値を入力する際には、数値として指定する必要があります。例えば、小文字 **x** を Ascii 'x' としてではなく Ascii120 として入力します。

## インバウンドACKを無視

真の場合、ACK フィードバック・ループが生成されないように、ビジネス・サービスは受信 ACK メッセージを無視します。

ACK 処理の全般的な説明は、このドキュメントで前述した “[HL7 確認応答 \(ACK\) モード](#)” を参照してください。

## ローカル・ファシリティ・アプリケーション

このビジネス・サービスを介して HL7 メッセージを受け取る施設とアプリケーションを表す、コロンの区切られた LocalFacility:LocalApplication コード。このビジネス・サービスが独自の ACK を作成する場合、**[ローカル・ファシリティ・アプリケーション]** は ACK メッセージに SendingFacility:SendingApplication コードを提供します。それ以外の場合、この設定は無視されます。

## メッセージ・スキーマ・カテゴリ

完全な DocType 仕様を生成するために受信メッセージ・タイプに適用するカテゴリ。ドキュメント・タイプの Name (MSH:9) と組み合わせて、指定された HL7 スキーマ・カテゴリの MessageTypes セクションにある MessageStructure/DocType を検索するために使用される MessageType 仕様を生成します。

この設定では、複数のタイプ名をカンマ区切りリストで指定することもできます。それぞれのタイプ名を含む HL7 メッセージに適用する DocTypeCategory の値または完全な DocType の値も、このリストでタイプ名の後に = で続けて指定できます。タイプ名の一部分を指定してその末尾にアスタリスク (\*) を付けると、そのエントリで始まるすべてのタイプと一致します。

HL7 メッセージ・オブジェクトの DocType プロパティ

例 : MessageSchemaCategory='2.3.1, ADT\_\*=2.5, BAR\_P10=2.4, ORM\_O01\_6=2.4:RDE\_O01'

DocType 割り当ては、**[検証]** または **[テーブルクラス検索]** のインデックス付けに必要な場合があることに注意してください。

HL7 ビジネス・サービスのカスタム・スキーマ定義をまだ準備していない場合、この時点ではこのフィールドを空白にしてもかまいません。ただし、ルーティング・プロセスの検証も無効にする場合を除き、永久に空白にしないでください。空白のままにすると、検証エラーが自動的に発生します。“[検証](#)” を参照してください。

## NACKエラーコード

受信メッセージの処理中にエラーが発生した場合にこのサービスが生成する NACK メッセージの MSA:1 のエラー・コードを制御します。既定は ContentE です。これは、メッセージ・コンテンツに関するエラーの場合はコード E を返し、メッセージの処理中に検出されたシステム・エラーの場合はコード R を返します。

システム・エラーが解決されてリモート・クライアントが後で再試行した場合にエラーが発生しないことがプロダクションで前提になりますが、メッセージ・コンテンツや検証のエラーは、ソースにおける修正が必要であり、同じ形で再試行しても効果がないと想定されています。そのため、この区別は重要です。一部のクライアント・システムでは異なるエラー動作が想定される場合や必要とされる場合があるため、インターシステムズ製品には、その他に 3 つの動作が用意されています。次のテーブルでは、4 つのオプションについて説明します。

コード	意味
ContentE	メッセージ・コンテンツ内のエラーを報告する場合は MSA エラー・コード E を使用し、(再試行可能な) システム・エラーが原因で拒否する場合はコード R を使用します。
ContentR*	コンテンツ・エラーの場合は R、システム・エラーの場合は E を返します。
AIIE	コンテンツ・エラーとシステム・エラーすべてについて E を返します。
AIIR	コンテンツ・エラーとシステム・エラーすべてについて R を返します。

ACK 処理の全般的な説明は、このドキュメントで前述した“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## 応答の保存

リモート・システムに返された応答メッセージのコピーを保存するかどうかを指定します。また、オプションで、構成された検索テーブル・クラスがある場合にそれを使用してメッセージのインデックスを作成するかどうかを指定します。以下のいずれかを選択します。

- ・ **[なし]** – 応答メッセージの保存もインデックス作成も行いません。
- ・ **[NotOKs]** – 単純な OK ACK メッセージでない応答のみを保存します。例えば、エラー NACK およびクエリ応答を保存します。
- ・ **[すべて]** – リモート・システムに送信されたすべての応答メッセージのコピーを保存します。
- ・ **[IndexNotOKs]** – 単純な OK ACK メッセージでない応答を保存し、かつ、構成された検索テーブルを使用してインデックスを作成します。このクラスで IndexReplies、SaveOKACKs、または IndexACKs パラメータをオーバーライドしていない限り、これがデフォルトの動作です。IndexReplies、SaveOKACKs、および IndexACKs は非推奨になりました。
- ・ **[IndexAll]** – すべての応答メッセージのコピーを保存し、かつ、構成された検索テーブルを使用してインデックスを作成します。

## Ackコミットコード使用

真または偽。真の場合、HL7 メッセージ・バージョン 2.3 以上の ACK メッセージを作成する際に、ビジネス・サービスは、いずれかの拡張モード ACK コミット・コードを MSA セグメントの AcknowledgmentCode フィールドに配置します。

HL7 ビジネス・サービスには、**[メッセージ・スキーマ・カテゴリ]** が 2.3 以上の場合に適用される **[Ackコミットコード使用]** 設定があります。これは真または偽に設定できます。真の場合、HL7 メッセージ・バージョン 2.3 以上の ACK メッセージを作成する際に、ビジネス・サービスは、いずれかの拡張モード ACK コミット・コードを MSA セグメントの AcknowledgmentCode フィールドに配置します。このコードは、次に示す 2 文字のシーケンスのいずれかです。

コード	元のモードにおける意味	拡張モードにおける意味
AA	アプリケーション承認	アプリケーション確認応答：承認

コード	元のモードにおける意味	拡張モードにおける意味
AE	アプリケーション・エラー	アプリケーション確認応答：エラー
AR	アプリケーション拒否	アプリケーション確認応答：拒否
CA	—	承認確認応答：コミット承認
CE	—	承認確認応答：コミット・エラー
CR	—	承認確認応答：コミット拒否

ACK 処理の全般的な説明は、このドキュメントで前述した“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

# HL7 ルーティング・プロセスに関する設定

HL7 ルーティング・プロセスの設定に関する参照情報を提供します。

## 概要

HL7 ルーティング・プロセスには次のような設定があります。

グループ	設定	参照先
基本設定	<a href="#">[ビジネス・ルール名]</a>	“プロダクション内での仮想ドキュメントの使用法”の“ <a href="#">ルーティング・プロセスに関する設定</a> ”
基本設定	<a href="#">[検証]</a>	このトピックの節
追加設定	<a href="#">[変換エラー時に作動]</a> 、 <a href="#">[検証エラー時に作動]</a> 、 <a href="#">[不正メッセージ警告]</a> 、 <a href="#">[不正なメッセージ・ハンドラ]</a> 、 <a href="#">[レスポンス From]</a> 、 <a href="#">[応答タイムアウト]</a> 、 <a href="#">[強制同期送信]</a>	“プロダクション内での仮想ドキュメントの使用法”の“ <a href="#">ルーティング・プロセスに関する設定</a> ”
追加設定	<a href="#">[ローカル・ファシリティ・アプリケーション]</a> 、 <a href="#">[ACK タイプ]</a> 、 <a href="#">[NACKコード]</a> 、 <a href="#">[NACK ERR追加]</a> 、 <a href="#">[応答ターゲット構成名]</a>	このトピックの節
開発とデバッグ	<a href="#">[ルールのロギング]</a>	“プロダクション内での仮想ドキュメントの使用法”の“ <a href="#">ルーティング・プロセスに関する設定</a> ”

残りの設定はすべてのビジネス・プロセスに共通のものです。“プロダクションの構成”の“[すべてのビジネス・プロセスに含まれる設定](#)”を参照してください。

## ACK タイプ

ACK または NACK 応答メッセージをローカルで作成する場合、ACK タイプ (AA または CA) を決定します。A がデフォルト・コードです。詳細は、“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## NACK ERR追加

真の場合、NACK メッセージを生成する際に、インターシステムズのエラー・コードとエラー・テキストを含む ERR セグメントを追加します。それ以外の場合は、内部エラー状態情報を NACK メッセージに埋め込みません。

## ローカル・ファシリティ・アプリケーション

このルーティング・プロセスを介して HL7 メッセージを受け取る施設とアプリケーションを表す、コロンで区切られた LocalFacility:LocalApplication コード。このルーティング・プロセスが独自の ACK または NACK メッセージを作成する場合、[\[ローカル・ファシリティ・アプリケーション\]](#) はメッセージに SendingFacility:SendingApplication コードを提供します。それ以外の場合、この設定は無視されます。

## NACKコード

エラーを報告するために NACK 応答メッセージをローカルで作成する場合、NACK コード・タイプ (AE または AR) を決定します。E がデフォルト・コードです。詳細は、“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## 応答ターゲット構成名

プロダクション内の構成項目のカンマ区切りリスト。指定されている場合、このリストは呼び出し側のほかに応答を転送する宛先を示します。空の場合は、応答は呼び出し側のみに返されます。この設定は、[レスポンス From] フィールドに値がある場合のみ有効です。

## 検証

HL7 メッセージ・ルータの [検証] 設定は、ルータによる受信メッセージの検証方法を制御します。受信メッセージが指定された検証に失敗した場合、イベント・ログにその失敗がレポートされ、HL7 ルーティング・プロセスはそのメッセージを不正メッセージ・ハンドラの上に渡します ([不正なメッセージ・ハンドラ] 設定を参照してください)。メッセージが指定検証に失敗したが、不正メッセージ・ハンドラがない場合、エラーがログに記録されますが、メッセージはどのターゲットにも送信されません。メッセージが検証に合格した場合、HL7 ルーティング・プロセスはルーティング・ルールで指定されたターゲットにそのメッセージを送信します。

ルーティング・ルールとデータ変換を使用して、メッセージごとにターゲット・システムで受理可能かどうかを確認することができ、その結果として検証の使用を避けることができる、というのが理想的です。これにより、すべてのメッセージが適切なターゲットで処理されます。検証を有効にした場合、ルーティング・ルールの前に検証テストが適用されます。検証に失敗したメッセージは、ルーティング・ルールに従ってターゲットに送信されず、不正なメッセージ・ハンドラの上に送信されます。ただし、HL7 メッセージ検証がメッセージのフィルタリング方法として適している環境も存在します。例えば、以下の状況では、HL7 検証を使用することが正しい選択です。

- ・ インタフェースを開発またはデバッグしており、システムで処理が必要なメッセージに類似する形式を判定する必要がある場合。
- ・ ターゲット・アプリケーションで、仕様と相違があるメッセージを処理できず、ルーティング・ルールおよび変換でその相違を解決できない場合。
- ・ 規制やその他のビジネスの要件があり、メッセージがその仕様に従っている場合。

HL7 検証はルーティング・プロセスにオーバーヘッドを付加します。このオーバーヘッドが大きな影響を与えるために、プロダクションで処理可能なメッセージの最大量が減少する可能性があります。

**Validation** プロパティでは、以下を制御するフラグを指定できます。

- ・ メッセージが有効なドキュメント・タイプを持つかどうか
- ・ メッセージ構造が検証されるかどうか
- ・ スキーマに指定されていない末尾の Z セグメントが検証されるかどうか
- ・ どのフィールドに検証を適用する必要があるか
- ・ 最初のエラーをレポートした後に検証を停止するか、または引き続きメッセージの残りのセグメントおよびフィールドの検証を行うか

**Validation** プロパティ値に空の文字列を指定した場合、メッセージ・ルータは検証を省略し、すべてのメッセージをルーティングします。管理ポータルで新しい HL7 ルーティング・プロセスを作成すると、[検証] 設定が空の文字列に初期化されます。

以前のリリースとの下位互換性を維持するため、[検証] を 1 に設定することは dm-z に設定することと同様に扱われ、[検証] を 0 に設定することは空の文字列に設定することと同様に扱われます。

メッセージ検証シーケンスは以下のとおりです。

1. メッセージ・オブジェクトに **DocType** プロパティ値があることを検証します。
2. セグメントの順序を検証します。
3. セグメント内のフィールドを検証します。



このシーケンスのいずれかの時点でメッセージが検証に失敗し、-x フラグが指定されていない場合、HL7 ルーティング・プロセスはメッセージの残りのセグメントを検証しません。

注釈 指定された **[検証]** フラグによっては、検証に合格したメッセージでも、厳密にはスキーマ定義に適合しない場合があります。

## DocType プロパティの検証

ビジネス・サービスは、HL7 メッセージの MSH:9 フィールドで指定されたサービスとドキュメント名に対して定義された **[メッセージ・スキーマ・カテゴリ]** に基づいて、DocType プロパティ値を設定します。Validation プロパティ値に d が含まれている場合、メッセージ・ルータは DocType プロパティに値があるかどうかをテストします。値がない場合、そのメッセージは不正メッセージになります。

以下のいずれかの状況では、ビジネス・サービスは DocType プロパティに値を割り当てません。

- ・ ビジネス・サービスに対して **[メッセージ・スキーマ・カテゴリ]** が指定されていない。
- ・ HL7 メッセージ内のドキュメント名が、指定されたスキーマ (カスタム・スキーマの場合は指定されたスキーマまたはベース・スキーマ) で定義されていない。

## セグメントの順序の検証

この節では、ルーティング・エンジンによるセグメントの検証方法について説明します。

Validation プロパティの値が dm、dmz、または dm-z である場合、ルーティング・プロセスの検証では、以下のテストによって、メッセージを検証するか、またはメッセージが不正であると宣言します。

1. ビジネス・サービスがメッセージに割り当てた DocType 値で指定されるメッセージ構造にメッセージのセグメントが準拠している場合、m 検証に適合します。例えば、メッセージの DocType 値が 2.3.1 で、メッセージ内の MSH:9 の値が ORU^R01 の場合、メッセージは 2.3.1 カテゴリの ORU\_R01 メッセージ構造に対して検証されます。2.3.1 ORU\_R01 と比較した際、メッセージ・データに必要なセグメントがない場合、または未定義の追加セグメントがある場合、それは不正メッセージとなります。HL7 では、末尾の Z セグメントとして、標準メッセージに独自のローカルなバリエーションを追加できます。メッセージの末尾の Z セグメントは、以下のルールに従って検証されます。
  - a. スキーマで、末尾の Z セグメントが指定されているか、“Zxx” ワイルドカード・セグメントが指定されている場合、末尾の Z セグメントは検証に合格します。
  - b. -z フラグが指定されている場合、メッセージの末尾の Z セグメントは検証に合格します。
  - c. -z フラグが指定されておらず、スキーマに末尾の Z セグメントが定義されていない場合、末尾の Z セグメントは検証に失敗します。
  - d. スキーマで末尾の Z セグメントが要求され、メッセージに含まれていない場合、そのメッセージは検証に失敗します。
2. DocType がカスタム・スキーマを参照する場合、メッセージがカスタム定義に対して照合されることを除き、テスト 1 と同じルールが適用されます。各カスタム・スキーマには、ベース・カテゴリ (例えば base="2.3.1") が定義されます。カスタム・スキーマで明示的に定義されていないメッセージが到着すると、ベース・カテゴリが使用されます。

MSH:9 フィールドに非標準値が含まれる可能性がある場合は、カスタム・スキーマを定義して、この非標準値を持つメッセージが検証に失敗することを回避する必要があります。例えば、MSH:9 が ORU^R01 ではなく ORU^Z22 と同じである場合、構造 ORU\_R01 に対してメッセージ名 ORU^Z22 を照合する必要があることを指定するには、カスタム・スキーマが必要となります。

カスタム・スキーマ・カテゴリの作成に関する情報は、“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](#)”を参照してください。

## フィールドの検証

f フラグは、すべてのフィールド検証を有効にします。f を指定すると、メッセージ・ルータは以下のフィールド検証を実行します。

- ・ メッセージのフィールドとコンポーネントの順序がスキーマに適合しているかどうかをテストします。
- ・ 必須のフィールド、コンポーネント、およびサブコンポーネントが存在することをテストします。
- ・ メッセージのフィールド・サイズがスキーマに適合しているかどうかをテストします。
- ・ フィールドのデータ値がスキーマで定義されたデータ型に適合するかどうかをテストします。
- ・ フィールドのデータ値がコード・テーブルに適合するかどうかをテストします。ただし、コード・テーブルに ... の値が含まれているか、コード・テーブルが空の場合は任意の値が許可されます。

## 検証フラグの指定

それぞれのフラグの前には、- (ハイフン記号) を付けてフラグの意味を無効化することができます。例えば、x フラグは、最初のエラーが検出された後にルーティング・プロセスが検証を停止する必要があることを指定し、-x フラグは、メッセージ全体を検証してすべてのエラーが検出されるまでルーティング・プロセスが検証を続行する必要があることを指定します。いくつかの検証フラグは、一連のフラグを指定するのと同等となります。例えば、s フラグは、jpw フラグを指定するのと同等です。無効化フラグを指定することで、連続するフラグの 1 つを解除できます。例えば、s-p は jw と等価です。フラグを無効化するには、フラグ文字の直前にハイフンを入力する必要があります。例えば、z フラグと x フラグの両方を無効化する場合、-z-x を指定する必要があります。

次のテーブルは、HL7 検証フラグを示しており、それぞれが指定されたときに、ルーティング・プロセスがメッセージをどのように検証するかを説明しています。

フラグ	ルーティング・プロセス
a	フィールド配列の繰り返しの制限値を強制します。
b	コード・テーブルの ... を解釈して、任意のフィールド値が検証に合格することを可能にします。
c	すべてのコンポーネントおよびサブコンポーネントの検証を実行します。gijopw と等価です。
d	DocType プロパティを定義するように要求します。他のフラグはすべて、指定された検証を実行するために DocType を定義する必要があります。
e	すべての検証を実行します。abdgijlmnoprtuwy と等価です。
f	セグメント内ですべてのフィールドの検証を実行します。abdgijlmnoprtuwy と等価です。
g	セグメント内でデータ構造を強制します。
i	コンポーネント・サイズの制限を強制します。
j	必須のサブコンポーネントが存在することを確認します。これらは、スキーマ内のオプション設定によって適用されます。
l	フィールド・サイズの制限を強制します。
m	スキーマ定義内のすべての必須セグメントがメッセージ内に含まれており、そのメッセージにスキーマで許可されていない間違ったセグメントが含まれていないことを確認します。
n	セグメントに正しい数のフィールドが含まれていることを確認します。
o	必須のコンポーネントが存在することを確認します。これらは、スキーマ内のオプション設定によって適用されます。



フラグ	ルーティング・プロセス
p	コンポーネント・レベルのデータ構造を強制します。
r	必須フィールドが存在することを確認します。
s	すべてのサブコンポーネント・レベルの検証を実行します。jpw と等価です。
t	コード・テーブルを強制します。このフラグは b フラグおよび u フラグで変更できます。
u	許容可能な値がリストされていないコード・テーブルを無視します。これらの空のコード・テーブルに対して任意の値を許可します。
w	サブコンポーネント・サイズの制限を強制します。
x	最初のエラーが検出されたら検証を停止します。これがデフォルトです。-x は、メッセージ全体を検証し、該当するステータスのすべてのエラーを返すことを指定します。
y	数値など、フィールドのデータ型を強制します。
z	Z セグメントをスキーマ内で明示的に指定するか、またはスキーマ内でワイルドカード zxx によって指定することを要求します。-z は、末尾の Z セグメントを、スキーマに指定されていない場合でも許可します。
(空の文字列)	検証を省略し、すべてのメッセージをルーティングします。

## HL7 シーケンス・マネージャに関する設定

HL7 シーケンス・マネージャの設定に関する参照情報を提供します。

### 概要

HL7 シーケンス・マネージャには次のような設定があります。

グループ	設定
追加設定	[メッセージ重複チェック有効]、[シーケンス番号チェックを実施 On]、[出力変換を実施 On]、[出力シーケンス番号インデックスフィールド]、[出力ターゲット構成名]、[メッセージウェイトタイムアウト]、[パススルーメッセージタイプ]、[順序外メッセージ通知タイプ]、[メッセージターゲットのシーケンス範囲外]、[重複メッセージ通知タイプ]、[重複したメッセージターゲット]、[出力ファシリティアプリケーション]、[メッセージ再送可能タイムウインドウ]、[ラージギャップサイズ]、[内部再送信のチェックを省略]

残りの設定はすべてのビジネス・プロセスに共通のものです。“プロダクションの構成”の“[すべてのビジネス・プロセスに含まれる設定](#)”を参照してください。

### 内部再送信のチェックを省略

この設定が真の場合、シーケンス・マネージャは内部再送信メッセージのチェックを省略します。

### 重複メッセージ通知タイプ

重複メッセージ通知のメッセージ・タイプを指定します。[WorkflowRequest] または [OriginalMessage] を指定します。

### 重複したメッセージターゲット

プロダクション内の項目（通常はビジネス・オペレーション）の構成済みの名前。指定した場合、シーケンス・マネージャが受信した重複メッセージを送信する場所となります。

### メッセージ重複チェック有効

真の場合、シーケンス・マネージャは、受信メッセージが以前に受け取ったメッセージと重複しているかどうかをチェックします。そのために、各メッセージの MSH セグメントで次の 3 つのフィールドを確認します。

- ・ MSH:3 SendingApplication
- ・ MSH:4 SendingFacility
- ・ MSH:10 MessageControlld

シーケンス・マネージャは、重複メッセージおよび順序外メッセージをチェックすることも、重複メッセージまたは順序外メッセージをチェックすることもできます。また、どちらもチェックしないようにすることもできます。そのため、[メッセージ重複チェック有効] は [シーケンス番号チェックを実施 On] から完全に独立しており、これらの設定の一方または両方を真に設定することができます。

### ラージギャップサイズ

メッセージのシーケンス内の重大なギャップを示す数値。デフォルトは 100 です。

遅延メッセージは、シーケンス内で前に受信したメッセージよりもシーケンス番号が大きいものの、正しく連続していないメッセージです。例えば、メッセージ 102 がメッセージ 101 の後に到着した場合、102 は遅延メッセージではありません。

連続する次のメッセージです。ただし、メッセージ 110 がメッセージ 101 の後に到着した場合、110 は遅延メッセージです。

遅延メッセージがあり、前のメッセージと後続のメッセージの間のシーケンス番号のギャップが、構成された[ラージギャップサイズ]より小さい場合、これはスモール・ギャップです。これよりも大きいギャップはラージ・ギャップです。

- ・ スモール・ギャップの場合、シーケンス・マネージャは、[メッセージウェイトタイムアウト]で指定された秒数の間待機して、ギャップを埋めるメッセージが到着するかどうかを確認します。待機期間内にメッセージが到着した場合、シーケンス・マネージャは、それらのメッセージを適切な順序に並べます。メッセージが到着せず、[メッセージウェイトタイムアウト]を超えた場合、シーケンス・マネージャは待機を停止し、現在のシーケンス内のメッセージに関する情報を送信します。
- ・ ラージ・ギャップがある場合、シーケンス・マネージャは一切待機しません。即座にアラートを送信した後、現在のシーケンス内のメッセージに関する情報を送信します。

## メッセージ再送可能タイムウインドウ

シーケンス・マネージャが重複メッセージを重複していると見なす時間ウインドウ。[メッセージ再送可能タイムウインドウ]を指定するのは、重複メッセージが誤りでないことがあるためです。場合によっては、メッセージ・シーケンスを意図的に再送信した結果であることもあります。

メッセージが到着し、シーケンス・マネージャが、そのメッセージを以前に受信したメッセージの重複として検出したとします。この場合は以下ようになります。

- ・ シーケンス・マネージャがそのメッセージを最初に確認してから[メッセージ再送可能タイムウインドウ]の秒数を超えている場合、新しく到着したメッセージは、意図的に再送信されたコピーとして解釈されます。シーケンス・マネージャは、メッセージのメイン・シーケンスの追跡とは別に、この再送信シーケンス内のメッセージの追跡を開始します。

シーケンス・マネージャが 1001、1002、1003、1004 などの番号を含むシーケンスを処理しているときに、同時にシーケンス番号 1、2、および 3 のメッセージの受信を開始したとします。[メッセージ再送可能タイムウインドウ]の秒数よりも前に、小さいシーケンス番号を持つ古いメッセージを最初に受信していた場合、シーケンス・マネージャは、それらのメッセージを順序外メッセージとしてではなく、再送信シーケンス内のメッセージとして解釈します。メイン・シーケンスと再送信シーケンスの追跡を同時に開始します。その間、2 つのシーケンスを別々に保持し、2 と 1001 など、関連しないシーケンス番号間のギャップを適宜無視します。

- ・ シーケンス・マネージャがそのメッセージを最初に確認してから[メッセージ再送可能タイムウインドウ]の秒数に達していない場合、新しく到着したメッセージは真の重複として解釈され、シーケンス・マネージャはそのメッセージを適宜処理します。“メッセージ重複チェック有効”を参照してください。

[メッセージ再送可能タイムウインドウ]のデフォルトは 300 秒 (5 分) です。

## メッセージウェイトタイムアウト

ラージ・ギャップ、スモール・ギャップ、および遅延メッセージの定義は、[ラージギャップサイズ]設定を参照してください。

スモール・ギャップの場合、シーケンス・マネージャは、[メッセージウェイトタイムアウト]で指定された秒数の間待機して、前のメッセージと遅延メッセージの間のギャップを埋めるメッセージが到着するかどうかを確認します。メッセージが到着せず、[メッセージウェイトタイムアウト]を超えた場合、シーケンス・マネージャは待機を停止し、現在のシーケンス内のメッセージに関する情報を送信します。

[メッセージウェイトタイムアウト]のデフォルトは 60 です。

## 順序外メッセージ通知タイプ

順序外メッセージ通知のメッセージ・タイプを指定します。[WorkflowRequest] または [OriginalMessage] を指定します。

## メッセージターゲットのシーケンス範囲外

プロダクション内の項目（通常はビジネス・オペレーション）の構成済みの名前。指定した場合、シーケンス・マネージャが順序外メッセージ通知を送信する場所となります。

## 出力ファシリティアプリケーション

[出力変換を実施 On] が [ ] または [ ] に設定されているときにメッセージ変換で使用する施設とアプリケーションを指定します。

[出力ファシリティアプリケーション] の値の形式は次のとおりです。

Facility:Application

## 出力シーケンス番号インデックスフィールド

出力シーケンス番号がインクリメントされるフィールドを制御します。以下の値のいずれかを指定します。

- ・ [ ] – インデックス・フィールドは MSH:3 SendingApplication と MSH:4 SendingFacility です。
- ・ [ ] – インデックス・フィールドは MSH:5 ReceivingApplication と MSH:6 ReceivingFacility です。
- ・ [ ] – この設定の値は、[出力変換を実施 On] の設定によって制御されます。[出力変換を実施 On] が [ ] である場合、[出力シーケンス番号インデックスフィールド] は [ ] です。それ以外の場合、[出力シーケンス番号インデックスフィールド] は [ ] です。

## 出力ターゲット構成名

プロダクション内の構成項目のカンマ区切りリスト。指定した場合、このリストは、テストに合格したメッセージをシーケンス・マネージャが送信する宛先を示します。

## パススルーメッセージタイプ

重複チェック、順序チェック、または出力変換から除外されるメッセージ・タイプのカンマ区切りリスト。シーケンス・マネージャは常に、これらのメッセージを変更せずに渡します。デフォルトのリストは次のとおりです。

QBP\_Q21,QBP\_Q22,RSP\_K21,RSP\_K22,ACK

## 出力変換を実施 On

メッセージを送信する前に変換するフィールドを指定します。出力変換は、次のリストに記載されているように施設、アプリケーション、およびシーケンス番号の新しいフィールドを提供し、構成されているターゲットにメッセージを送信します。[出力変換を実施 On] に指定できる値は次のとおりです。

- ・ [ ] – これがデフォルトです。シーケンス・マネージャは、[出力ファシリティアプリケーション] 設定から Facility と Application をコピーし、新しいシーケンス番号を割り当てます。その結果、送信メッセージで次のフィールドが変更されます。
  - MSH:3 SendingApplication
  - MSH:4 SendingFacility
  - MSH:13 SequenceNumber
- ・ [ ] – シーケンス・マネージャは、[出力ファシリティアプリケーション] 設定から Facility と Application をコピーし、新しいシーケンス番号を割り当てます。その結果、送信メッセージで次のフィールドが変更されます。
  - MSH:5 ReceivingApplication
  - MSH:6 ReceivingFacility

- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは、メッセージを送信する前に変換を実行しません。

## シーケンス番号チェックを実施 On

順序が正しくない受信メッセージがないかどうかをチェックするフィールドを指定します。順序が正しくない受信メッセージがあれば、シーケンス・マネージャはメッセージを並べ替え、構成されているターゲットに送信します。

可能な値は以下のとおりです。

・ [ ] – これがデフォルトです。シーケンス・マネージャは次のフィールドをチェックします。

- MSH:3 SendingApplication
- MSH:4 SendingFacility
- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは次のフィールドをチェックします。

- MSH:4 SendingFacility
- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは次のフィールドをチェックします。

- MSH:3 SendingApplication
- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは次のフィールドをチェックします。

- MSH:5 ReceivingApplication
- MSH:6 ReceivingFacility
- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは次のフィールドをチェックします。

- MSH:6 ReceivingFacility
- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは次のフィールドをチェックします。

- MSH:5 ReceivingApplication
- MSH:13 SequenceNumber

・ [ ] – シーケンス・マネージャは、順序外メッセージをチェックしません。

シーケンス・マネージャは、重複メッセージおよび順序外メッセージをチェックすることも、重複メッセージまたは順序外メッセージをチェックすることもできます。また、どちらもチェックしないようにすることもできます。そのため、**[メッセージ重複チェック有効]** は **[シーケンス番号チェックを実施 On]** から完全に独立しており、これらの設定の一方または両方を真に設定することができます。

## HL7 ビジネス・オペレーションに関する設定

HL7 ビジネス・オペレーションの設定に関する参照情報を提供します。

### 概要

HL7 ビジネス・オペレーションには次のような設定があります。

グループ	設定	参照先
基本設定	[ファイル名]	このトピックの節
接続設定	[準備完了]、[構成]	このトピックの節
追加設定	[テーブルクラス検索]	“プロダクション内での仮想ドキュメントの使用法”の“ <a href="#">ビジネス・オペレーションに関する設定</a> ”
追加設定	[親セグメント自動バッチ]、[セパレータ]、[デフォルト文字エンコーディング]、[リプライコードアクション]、[切断中は失敗としない]、[再試行間隔]	このトピックの節

残りの設定は、すべてのビジネス・オペレーションに共通のものであるか、アダプタのタイプによって異なるものであるかのどちらかです。詳細は、以下を参照してください。

- ・ “プロダクションの構成”の“[すべてのビジネス・オペレーションに含まれる設定](#)”
- ・ “プロダクション内でのファイル・アダプタの使用法”の“[ファイル送信アダプタに関する設定](#)”
- ・ “プロダクション内での FTP アダプタの使用法”の“[FTP 送信アダプタに関する設定](#)”
- ・ “プロダクション内での HTTP アダプタの使用法”の“[HTTP 送信アダプタの設定](#)”
- ・ “プロダクション内での TCP アダプタの使用法”の“[TCP 送信アダプタに関する設定](#)”

EnsLib.HL7.Adapter.TCPOutboundAdapter には、HL7 用に適切に構成されている以下の設定が含まれています。

- － [準備完了] は真に設定されています。これは、ソケットからの応答メッセージを読み取るまで待機してから返すことを意味します。
- － [接続タイムアウト] の通常のデフォルト値は 5 秒ですが、最大値は 30,000 秒となります。
- － [応答タイムアウト] のデフォルト値は、通常の 15 秒ではなく 30 秒で、最大値は 30,000 秒となります。

HL7 ビジネス・オペレーションに関する最も重要な設定は次のとおりです。

- ・ [プール・サイズ] – [プール・サイズ] のデフォルト値 1 では、FIFO (先入れ先出し) 処理をサポートできます。受信アプリケーションで正しいデータを確保するには、FIFO 処理が重要です。複数の患者基本情報の更新を順番に受信する必要があります。そうしないと、古い不適切なデータが受信アプリケーションに表示される可能性があります。さらに、多くのアプリケーションでは、オーダー・メッセージを処理する前に ADT 登録メッセージを受信する必要があります。また、結果メッセージの前にオーダー・メッセージを受信する必要があります。
- ・ [再試行タイムアウト] – 再試行を実行できる秒数。HL7 ビジネス・オペレーションは、“タイムアウトが発生しないように” この値を自動的に -1 に設定して、HL7 メッセージがスキップされることがないようにします。
- ・ [カテゴリ] – このテキスト・ラベルにより、構成ダイアグラム内で構成項目をソートできます。
- ・ [再接続試行] – (TCP のみ) 接続を停止し、接続を再試行する回数です。値が 0 (ゼロ) の場合は、切断しません。デフォルトは 5 です。



- ・ [\[接続を維持\]](#) 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、**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。

## 親セグメント自動バッチ

(ファイルおよび FTP のみ) 真の場合、バッチの親を持つメッセージを書き込む際に、バッチ・ヘッダ、子ドキュメントの順に出力します。さらに、最後のバッチ・ヘッダ・メッセージまたはファイル名の変更によってトリガされると、バッチ・トレーラを出力します。偽の場合、ヘッダおよびトレーラは省略され、子ドキュメントのみが出力されます。デフォルトは偽です。

バッチ処理の詳細は、“[HL7 バッチ・メッセージ](#)”を参照してください。

## デフォルト文字エンコーディング

送信 HL7 メッセージに適用する文字エンコーディングです。文字は、自動的にこのエンコーディングに変換されます。ビジネス・サービスの“[デフォルト文字エンコーディング](#)”を参照してください。

## ファイル名

(ファイルおよび FTP のみ) ターゲット・ファイル名。[\[ファイル・パス\]](#) アダプタ設定はこのファイルのパスを決定しますが、[\[ファイル名\]](#) は名前を決定します。[\[ファイル名\]](#) には、タイム・スタンプ指定子を含めることができます。[\[ファイル名\]](#) を空白にしておくと、デフォルトで、%f\_%Q というタイム・スタンプ指定子が使用されます。

- ・ %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。

デフォルトの %f\_%Q の代わりに使用できる各種のコードを含む、タイム・スタンプの規則の詳細は、“[プロダクションの構成](#)”の“[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。

## 構成

HL7 ビジネス・オペレーションが送信 HL7 メッセージ・パケットを作成する方法を制御します。フレーミング・オプションのリストは、“[HL7 ビジネス・サービスに関する設定](#)”の“[構成](#)”を参照してください。使用する値がわからない場合は、HL7 ビジネス・オペレーションのデフォルトの [\[MLLP\]](#) フレーミングをそのまま使用してください。

## 準備完了

(TCP ビジネス・オペレーションのみ) 真の場合、ビジネス・オペレーションは、ソケットからの ACK またはその他の応答メッセージを読み取るまで待機してから戻ります。また、定義されている [\[リプライコードアクション\]](#) を適用します。

ACK 処理の全般的な説明は、このドキュメントで前述した“[HL7 確認応答 \(ACK\) モード](#)”を参照してください。

## 切断中は失敗としない

(TCP のみ) 真の場合、TCP サーバから切断されている間、[\[再試行タイムアウト\]](#) までの秒数のカウントを中断します。[\[再試行タイムアウト\]](#) が -1 または [\[接続を維持\]](#) が 0 の場合は、この設定は適用されません。

## リプライコードアクション

さまざまなタイプの ACK 応答メッセージを受信した際にビジネス・オペレーションが実行するアクションを指定するように、コードとアクションのペアのカンマ区切りリストを指定できます。リストの形式は以下のとおりです。

```
code=action,code=action, ... code=action
```

code (: で始まる) は、応答メッセージの MSA:1 (確認応答コード) フィールドで見つかったリテラル値、または次に示す特殊なコード値のいずれかを表します。

コード	意味
:?A	AA 値または CA 値 (承認) に相当します。
:?E	AE 値または CE 値 (エラー) に相当します。
:?R	AR 値または CR 値 (拒否) に相当します。
:_	空の MSA:1 フィールドを含む応答に相当します。空または空白のコード値は _ (アンダースコア) と同じです。
:*	他の値と一致しない任意の MSA:1 値に相当します (既定値 = S)。
:~	MSA セグメントを含まない応答に相当します。
:I?	応答の MSA:2 ControlId が元のメッセージの ControlId と一致しない場合に相当します。
:T?	応答の MSH:9 タイプ名が、元のメッセージについてスキーマで宣言されている応答タイプと一致しない場合に相当します。

“プロダクションの構成” の参照セクションの “[リプライコードアクション](#)” に記載されている標準コードを使用することもできます。

action には次の値を単独で使用することも、値を組み合わせて文字列を形成することもできます。他に指定されていない場合は、F がデフォルトの action です。ただし、:?A のデフォルトのアクションは C です。

アクション	意味
C	メッセージを正常に完了したものとして処理します。
W	警告をログに記録しますが、メッセージを正常に完了したものとして処理します。
R	構成済みの RetryInterval と FailureTimeout に従ってメッセージを再試行します。別のアクションが指定されていない場合、最終的には失敗します。
S	メッセージを中断し、エラーをログに記録して、次のメッセージに進みます。
D	オペレーションを無効にし、エラーをログに記録して、オペレーションのキューの先頭に送信メッセージをリストアします。
F	エラーにより失敗し、次のメッセージに進みます。

code=action, code=action, ... code=action

[リプライコードアクション] のデフォルト値は次のとおりです。

:?R=RF, :?E=S, :~=S, :?A=C, :\*=S, :I?=W, :T?=C

この既定値は、確認応答コードが AR または CR のメッセージを再試行するよう指定します。コードが AE または CE のメッセージについては、現在のメッセージを中断し、エラーをログに記録して、次のメッセージに進みます。また、コードが AA または CA のメッセージを正常に完了したものとして処理し、リストされている他の応答コードと MSA:1 フィールドの値が一致しないメッセージを中断します。

**注釈** 再試行のデフォルトは ?R=RF です。ほとんどの場合、これはエラーの直後に失敗しますが、エラーが “エラー #5005 : ファイルを開くことができません。” である場合、オペレーションは、FailureTimeout に達するまで再試行を続行します。多くの場合、このエラーの原因は一時的な問題であり、再試行することで問題が解決しますが、ディレクトリが正しくない場合などは、オペレーションが繰り返し失敗することもあります。

バッチ ACK メッセージについては、バッチ内で見つかった最初の子 ACK に基づいて、適切な [リプライコードアクション] が決定されます。

ACK 処理の全般的な説明は、このドキュメントで前述した “[HL7 確認応答 \(ACK\) モード](#)” を参照してください。



## 再試行間隔

インターシステムズ製品外部の宛先への接続試行を待機する秒数。

## セパレータ

送信メッセージで使用する HL7 区切り文字。このフィールドを空白のままにした場合、デフォルトは次のとおりです。

|^~\&

## 基本

HL7 メッセージでは、未加工のコンテンツを編成するために特殊文字が使用されます。これらの文字は、医療アプリケーションによって異なる場合があります。そのため、HL7 標準では、各 HL7 メッセージで区切り文字として使用されている 5 つの具体的な文字を MSH セグメントの先頭に左から右へ次の順にリストするよう規定されています。

1. フィールド区切り文字 (FS)
2. コンポーネント区切り文字 (CS)
3. 繰り返し区切り文字 (RS)
4. エスケープ文字 (ESC)
5. サブコンポーネント区切り文字 (SS)

6 番目の文字であるセグメント終端文字は MSH では指定しません。特に指定がない場合、セグメント終端文字は、HL7 標準に従ってキャリッジ・リターン (CR) です。ただし、キャリッジ・リターン/改行 (CRLF) 文字もデフォルトで許可されています。

## 詳細

[セパレータ] には、前出のリストで説明したように、HL7 区切り文字に左から右へ FS、CS、RS、ESC、SS の順に割り当てられる文字列を指定する必要があります。

[セパレータ] 文字列の位置 1 ～ 5 以外に、追加の文字を指定して、デフォルトのセグメント終端文字であるキャリッジ・リターン (ASCII 13) をオーバーライドすることができます。位置 5 の後に、キャリッジ・リターン (ASCII 13) の場合は \r、改行 (ASCII 10) の場合は \n を使用します。

位置 6 以降にセグメント終端文字を指定する必要があるものの、出力メッセージで使用する区切り文字が 5 つより少ない場合は、位置 1 ～ 5 に \x を使用できます。位置 1 ～ 5 に \x で指定されている区切り文字は、使用されません。 \x の目的は、単に区切り文字の列挙の長さを延長し、位置 6 が最初のセグメント終端文字として正しく解釈されるようにすることです。

## HL7 のエスケープ・シーケンス

HL7 のエスケープ・シーケンスに関する参照情報を提供します。

### 詳細

HL7 メッセージのデータ・コンテンツ内に区切り文字を表示する必要がある場合、HL7 には、区切り文字を置き換えるエスケープ・シーケンスが用意されています。

この機能を必要とする可能性が最も高い文字は & (アンパサンド) 文字です。多くの場合、この文字は HL7 区切り文字として使用されます。同時に、& は HL7 ドキュメント・データにも出現することがあります。雇用主の正式名 (Parker & Sons) に使用されていることもあれば、治療指示 (XR CHEST PA&LAT) で省略表現として使用されていることもあります。

そのため、HL7 データ・ストリーム内では、次の例の \T\ のように、& 文字がエスケープ・シーケンスに置き換えられていることがよくあります。

```
Parker \T\ Sons
XR CHEST PA\T\LAT
```

次のテーブルは、標準の HL7 エスケープ・シーケンスとその意味を示したリストです。HL7 エスケープ・シーケンスの先頭と末尾には、MSH セグメントで定義されたエスケープ文字が使用されます。HL7 エスケープ・シーケンスでは大文字と小文字が区別され、テーブルに示されているように、特定の文字 F R S T E に加えて .br、X0A、および X0D が使用されます。ここに示されている例では、\ (円記号) がエスケープ文字です。

文字シーケンス	HL7 データ内における意味	例
\.br\*	キャリッジ・リターン	
\F\	フィールド区切り文字	
\R\	繰り返し区切り文字	~
\S\	コンポーネント区切り文字	^
\T\	サブコンポーネント区切り文字	&
\E\	エスケープ文字	\
\X0A\	改行	
\X0D\	キャリッジ・リターン	

\* エスケープ解除のみに使用されます。

DTL を使用してリテラル文字列をターゲット・フィールドに <assign> するときに、その文字列に HL7 区切り文字を区切り文字としてまたはリテラル文字として含める必要がある場合は、特殊な処理が必要です。手順は次のとおりです。

MSH セグメントにリストされているように ^、\、および & が区切り文字であるソース・メッセージがあるとして。

```
MSH|^~\&
```

ここで、特定の医療ソースから到着したすべての ADT\_A01 メッセージについて、DTL データ変換でリテラル値を InsuranceCompanyName フィールドに割り当てる必要があるとします。正しい値については、Pierre DuRhône & Cie という名前の保険会社をはじめ、すべての関係者の合意を得ています。この値は、HL7 データ・ストリームでは次のように表示されます。

```
Pierre DuRho^ne & Cie
```

この場合、次のように、データ内の ^ と & を置き換える HL7 エスケープ・シーケンスが必要です。

## XML

```
<assign
  property='target.{PIDgrp.IN1grp(1).IN1:InsuranceCompanyName.organizationname}'
  value='"Pierre DuRho\S\ne \T\ Cie"' />
```

これ以外に区切り文字について配慮する必要があるのは、インターシステムズでの HL7 仮想ドキュメントとの間でデータを移行するときに、HL7 から HL7 へのデータ変換を使用しない場合のみです。例えば、XML から HL7 に変換する場合や、変換をまったく使用しない場合は、区切り文字を明示的に処理する必要があります。**EnsLib.HL7.Segment** クラスには、HL7 メッセージをエスケープ解除およびエスケープするために DTL データ変換で自動的に行われる処理を手動で実行するために呼び出すことができる明示的な `Escape()` メソッドと `Unescape()` メソッドが用意されています。`replace()` メソッドもあります。

