



プロダクション内での X12 ド キュメントのルーティング

Version 2024.1
2024-06-06

プロダクション内での X12 ドキュメントのルーティング

InterSystems IRIS Data Platform Version 2024.1 2024-06-06

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

1 X12 の概要	1
1.1 X12 エンベロープの構造	1
1.2 X12 と HIPAA	3
1.3 X12 スキーマ配布ファイル	3
1.4 InterSystems IRIS による X12 ドキュメントのサポート	4
2 X12 スキーマと利用可能なツール	5
2.1 X12 スキーマ構造ページの使用法	5
2.1.1 InterSystems IRIS への X12 スキーマのロード	5
2.1.2 ドキュメント構造の表示	6
2.1.3 セグメント構造の表示	8
2.1.4 複合構造の表示	10
2.1.5 異なるカテゴリの選択	10
2.2 X12 ドキュメント・ビューワ・ページの使用法	11
2.2.1 オプションの選択	11
2.2.2 ドキュメントの解析	12
2.2.3 変換のテスト	16
2.3 X12 クラス	16
2.4 カスタム X12 スキーマの作成	17
2.5 X12 検索テーブルの定義	18
3 プロダクションの構成	19
3.1 新しい X12 プロダクションの作成	19
3.2 X12 ビジネス・サービスの追加	20
3.3 X12 ビジネス・プロセスの追加	20
3.4 X12 ルーティング・ルールへの追加	21
3.5 X12 ビジネス・オペレーションへの追加	21
4 X12 データ変換の作成	23
4.1 一般的な X12 DTL 使用事例	23
4.2 X12 DTL を作成する手順	24
4.2.1 ソースおよびターゲット・クラスの作成	24
4.2.2 インターチェンジ・コントロール・ヘッダのコピー	25
4.2.3 機能グループのコピー	25
4.2.4 トランザクション・セットのコピー	26
4.2.5 機能グループおよびトランザクション・セットのループの完了	26
4.3 バッチ全体用のデータ変換	27
4.3.1 各インターチェンジ内のグループのループ処理	27
4.3.2 各グループ内のトランザクション・セットのループ処理	28
4.4 X12 データ変換のテスト	28
5 X12 インターチェンジの処理	29
5.1 X12 バッチ・ハンドリング	29
5.1.1 バッチ・ドキュメントの受信	29
5.1.2 バッチ・ドキュメントの送信	29
5.1.3 バッチ・モード	30
5.2 バッチ全体向けのビジネス・プロセスの構成	30
5.2.1 バッチ全体用のルーティング・ルール	30
5.3 他のバッチ設定向けのビジネス・プロセスの構成	31

5.3.1 他のバッチ設定用のルーティング・ルール	31
5.3.2 他のバッチ設定用のデータ変換	32
X12 の設定	33
X12 ビジネス・サービスの設定	34
X12 ビジネス・プロセスの設定	40
X12 ビジネス・オペレーションの設定	43

1

X12 の概要

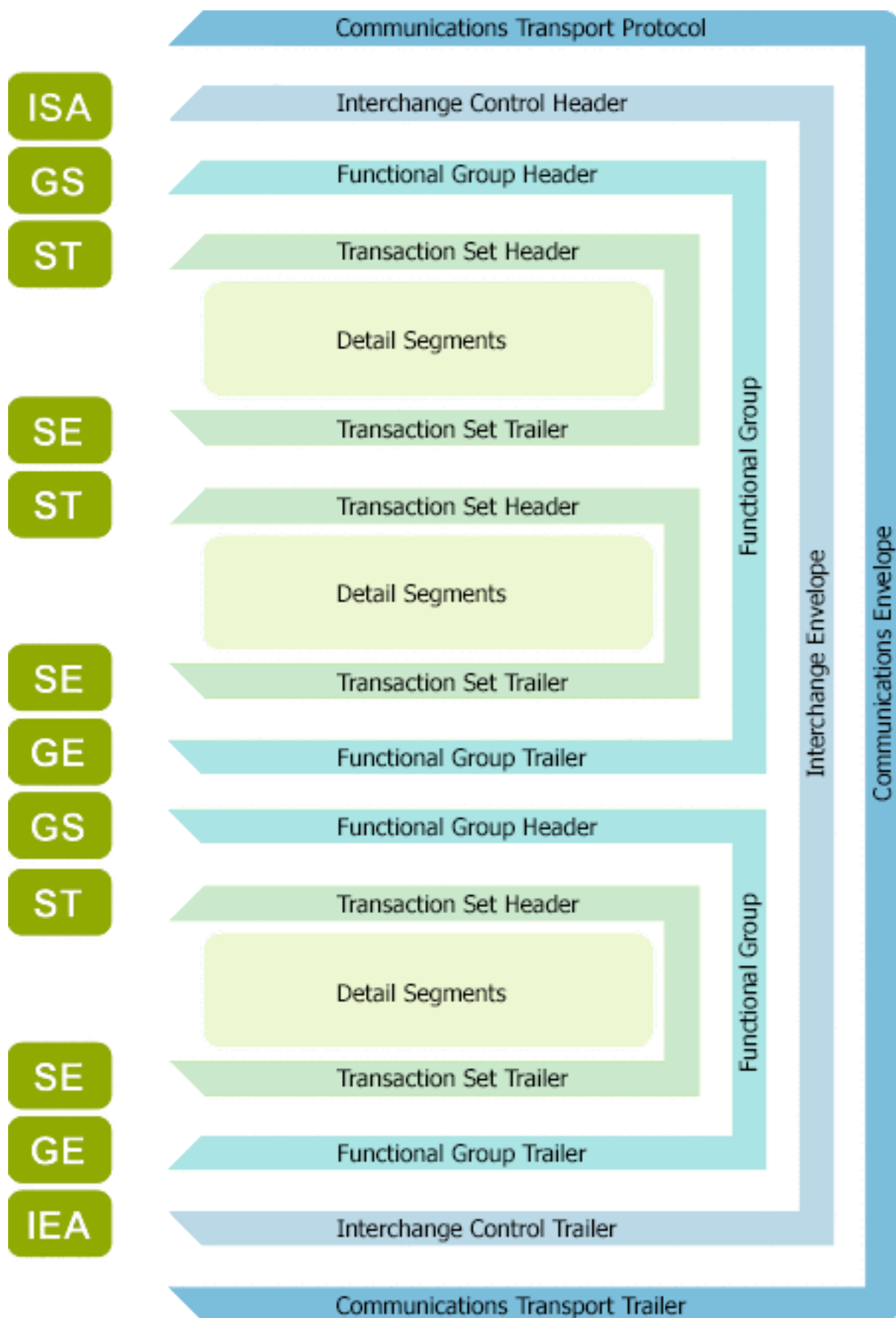
ここでは、X12 標準と InterSystems IRIS[®] データ・プラットフォームによる X12 のサポートについて簡単に紹介します。

米国規格協会 (ANSI) は、世界規模での使用を目的とした電子データ交換標準とそれに関連するドキュメントや製品を構築およびサポートする業界間フォーラムとして、Accredited Standards Committee (ASC) X12 を設立しました。したがって、X12 は Electronic Data Interchange (EDI) に関する ANSI 標準です。これには、300 以上のドキュメント・タイプが定義されています。どの X12 ドキュメント・タイプも InterSystems IRIS のサポート対象に含まれますが、InterSystems では、Health Insurance Portability and Accountability Act (HIPAA : 医療保険の積算と責任に関する法律) に関するドキュメントに主な重点を置いてきました。

1.1 X12 エンベロープの構造

X12 エンベロープ構造のルールにより、データの整合性および情報交換の効率性が確保されます。実際の X12 メッセージ構造のプライマリ・レベルは階層的で、上位から下位の順で、以下のようにになっています。

- ・ インターチェンジ・エンベロープ
- ・ 機能グループ
- ・ トランザクション・セット



1.2 X12 と HIPAA

HIPAA の第 2 編、管理の簡略化に関する規定 (AS : Administrative Simplification) では、医療情報の電子的なトランザクションに関する国家標準の構築と、医療サービス提供者、健康保険プラン、および雇用者に関する国家的な ID の構築が義務付けられています。AS 規定では、また、医療データのセキュリティとプライバシー保護についても定められています。これらの標準の目的は、米国の医療システムで電子データ交換の普及を促進することにより、その医療システムの効果と効率を向上させることです。

X12 ドキュメントの中で、HIPAA を明確にサポートしているものは以下のとおりです。

- ・ 270_X092: Eligibility, Coverage or Benefit Inquiry
- ・ 271_X092: Eligibility, Coverage or Benefit Information
- ・ 276_X093: Healthcare Claim Status Request
- ・ 277_X093: Healthcare Claim Status Notification
- ・ 278_X094Request: Healthcare Service Review (Request)
- ・ 278_X094Response: Healthcare Service Review (Response)
- ・ 820_X061: Payment Order/Remittance Advice
- ・ 834_X095: Benefit Enrollment and Maintenance
- ・ 835_X091: Healthcare Claim Payment/Advice
- ・ 837_X096: Healthcare Claim: Institutional
- ・ 837_X097: Healthcare Claim: Dental
- ・ 837_X098: Healthcare Claim: Professional

InterSystems IRIS には、HIPAA 4010 と HIPAA 5010 のスキーマがあらかじめインストールされています。HIPAA 5010 では、277CA、820hix、837apD、837apI、837apP、999 などの医療保険取引所に関する追加のドキュメントが提供されます。

1.3 X12 スキーマ配布ファイル

X12 標準では、ドキュメントの構文とセマンティクスを詳細に説明しています。この説明は、公式の発行元である Washington Publishing Company の以下の URL から PDF ファイル形式で入手できます。

<http://www.wpc-edi.com>

有益であっても、ドキュメントはコンピュータで読み取り可能ではありません。コンピュータで読み取り可能な X12 スキーマは、SEF または XSD ファイル形式で入手できます。SEF ファイルは EDI ドキュメントの構文とセマンティクスの大半をエンコードします。例えば、フィールドのサイズと種類や使用可能なあらゆるコード値を指定するだけでなく、ドキュメント内のフィールド間の依存関係やセグメント間の依存関係も定義します。SEF 標準は、EDI 仕様 (X12 以外も) をエンコードし、パブリック・ドメインに入れることを目的として、Foresight Corporation によって開発されました。次の URL から標準形式の印刷バージョンを入手できます。

<http://www.edidev.com/articles/sef16.pdf>

この情報をよく調べると、EDI の適用範囲と複雑さを理解するのに役立ちます。多くの組織がさまざまな EDI 標準をエンコードし、一般に公開してきました。また、米国国防総省も、収集システムに関する SEF ファイルを多数発行しています。

XSD ファイル内の情報は、SEF ファイル内の情報とほとんど同じですが、主な省略がいくつかあります。

- ・ XSD ファイルには、セグメントの位置も欠落しています。これは、実装ガイドにも存在する概念ですが、InterSystems IRIS のドキュメントの処理には影響ありません。
- ・ さらに、XSD ファイルには、セグメントの順序も欠落しています。これは、検証スタイル・スキーマの表現でセグメントの特定の出現を識別するため、実際に必要となります。ただし、順序値は実装ガイドには記載されておらず、これらはデフォルトで 1 ずつ増加する (このパターンに従わない場合にのみ SEF ファイルに表示される) ため、各セグメント間で順序を 1 増加させるというデフォルトに従うことにより、ドキュメントを適切に処理することが可能になり、スキーマが不正確に表現されることはありません。
- ・ 最後に、XSD ファイルには、関係条件は含まれません。

1.4 InterSystems IRIS による X12 ドキュメントのサポート

InterSystems IRIS では、処理を高速化するために、X12 ドキュメントは仮想ドキュメントと呼ばれる InterSystems IRIS オブジェクトのタイプとして保管されます。具体的には、X12 ドキュメントは `EnsLib.EDI.X12.Document` として保管され、この X12 ドキュメントの各セグメントは `EnsLib.EDI.X12.Segment` として保管されます。X12 インターチェンジも `EnsLib.EDI.X12.Document` として保管され、この中に格納されている他の X12 ドキュメントへのポインタが付随します。トランザクション・セットのセグメントと要素にアクセスするには、仮想プロパティ・パスを使用します。例えば、指定されたトランザクション・セットの識別子コードにアクセスするには、“`ST:TransactionSetIdentifierCode`” という仮想プロパティ・パスを使用します。詳細は、“[ドキュメントの解析](#)” を参照してください。

InterSystems IRIS で提供されてい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。

2

X12 スキーマと利用可能なツール

ここでは、X12 スキーマと X12 ドキュメントの操作に使用可能な InterSystems IRIS® ツールの概要を示します。

2.1 X12 スキーマ構造ページの使用法

[Interoperability]→[相互運用]→[ASC X12]→[ASC X12 スキーマ構造] ページを使用すると、X12 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。

スキーマ構造を決定するために、InterSystems IRIS は、この X12 ドキュメントの構造を定義するためにインポートされた SEF ファイルの .SETS、.SEGS、.COMS、.ELMS、および .CODES セクションから詳細を抽出します。ドキュメントの説明はこの X12 スキーマを定義する SEF ファイルの .INI セクションから抽出されます。Healthcare Claim Status Request は 1 つの例です。

2.1.1 InterSystems IRIS への X12 スキーマのロード

[スキーマ構造] ページから X12 スキーマを InterSystems IRIS にロードするには、[インポート] をクリックして目的のファイルを選択します。SEF ファイルまたは XSD ファイルのいずれかをロードできます。

インポートする SEF ファイル (X12 スキーマ) の検索場所に関する提案は、“[標準交換フォーマット \(SEF : Standard Exchange Format\) ファイル](#)”を参照してください。

2.1.1.1 プログラムによる SEF ファイルのロード

X12 スキーマ・ファイルをプログラムによってロードするには、以下の操作を行います。

1. ターミナル・セッションを開始します。
2. 相互運用対応ネームスペースに変更し、以下のコマンドを発行します。

```
Do ##class(EnsLib.EDI.SEF.Compiler).Import(filename)
```

filename は、スキーマ・ファイルの完全パス名です。

このコマンドにより、スキーマ・ファイルからデータがインポートされ、このデータを InterSystems IRIS 内でスキーマ定義として使用できるようになります。

3. InterSystems IRIS では、SEF ファイルの .INI セクションの先頭行の最初の部分を使用して、新しいスキーマ・カテゴリの名前を作成します。例えば、837_X097.SEF ファイルに以下の行があるとしています。

```
.INI 837_X098,,004 010 X098,X,X12-4010,Healthcare Claim: Professional
```

抽出されたスキーマ・カテゴリは、以下のような名前になります。

```
837_X098
```

スキーマ名前付け規則に従い、スキーマ・ファイルを編集してカスタマイズする場合は、まず、カテゴリ名を提供するスキーマ・ファイルのテキストを変更し、カスタマイズするバージョンと InterSystems IRIS にインポートする他のスキーマ・ファイルを区別できるようにすることをお勧めします。

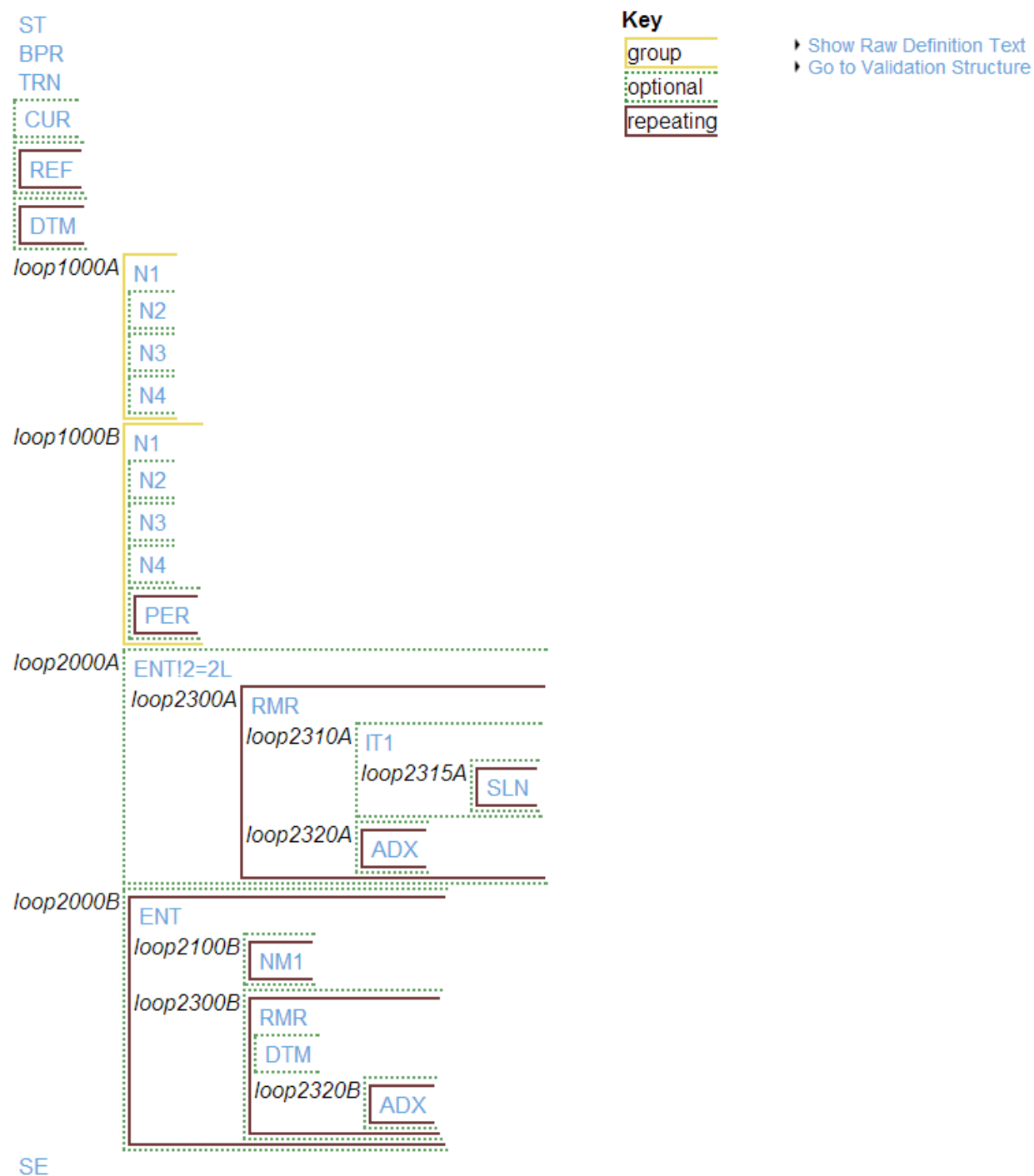
4. スキーマ・ファイルには、構文エラーが含まれている場合があります。その場合、InterSystems IRIS からエラー・メッセージが表示され、スキーマ・ファイル内のエラーの場所が特定されます。

2.1.2 ドキュメント構造の表示

ドキュメント構造の内部編成を表示するには、[Interoperability]→[相互運用]→[ASC X12]→[ASC X12 スキーマ構造] ページの [DocType 構造] タブで、そのドキュメント構造の名前を選択します。InterSystems IRIS では、以下に説明する表現上の規則に基づいてドキュメントのセグメント構造が表示されます。以下の例は、このページの HIPAA_4010:820 ドキュメント構造を示しています。[Interoperability]→[相互運用]→[ASC X12]→[ASC X12 スキーマ構造] ページの左側で [HIPAA_4010] を選択し、右側で [820] を選択します。[X12 ドキュメント構造] ページが表示されます。

Schema Category : [HIPAA_4010](#)Document Structure (TransactionSet) : **820**Implementation Guide : **X061A1**

Payment Order/Remittance Advice



2.1.2.1 レイアウト

- セグメントは青色テキストで表示され、関連するセグメント構造ページにリンクしています。

- ・ ループは黒色テキストで表示されます。
- ・ ドキュメント構造を構成するセグメントは、左から右および上から下へ順番に表示されます。

注釈 X12 標準に従って、ドキュメント内で、セグメントやサブグループ・タイプが、次のセグメントやサブグループ・タイプの前であれば任意の順序で、許可された回数繰り返し表示される場合があります。例えば、スキーマで 8 つの REF セグメントを CRC セグメントの前に表示する必要があると指定している場合、その REF セグメントは任意の順序で表示されます。ただし、それらの間にその他のセグメント・タイプを点在させることはできません。

- ・ 各ドキュメント・セグメントの名前が表示されます (BPR、NM1、DTM など)。この名前は、X12 ドキュメント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。
- ・ 緑色の点線で囲まれているセグメントは、オプションのセグメントです。
- ・ 茶色の実線で囲まれているセグメントは、何度か繰り返される可能性のあるセグメント (存在する場合) です。
- ・ 黄色の実線で囲まれているセグメントは、グループの一部であるセグメントです。
- ・ セグメントは、繰り返しとオプションの両方である場合もあります (上記のセグメント NM1 を参照してください)。

2.1.2.2 未加工の定義

未加工のテキスト形式でドキュメント構造を表示するには、[未加工の定義テキストの表示] をクリックします。HIPAA_4010:820 ドキュメント構造の未加工定義は次のとおりです。

Raw Definition

```
ST~BPR~TRN~[~CUR~]~[~{~REF~}~]~[~{~DTM~}~]~(1000A~N1~[~N2~]~[~N3~]~[~N4~]~[~{~PER~}~]~[2000A~ENT!2=2L~{2300A~RMR~[2310A~IT~AD~X~}~]~[2000B~{~ENT~[2100B~{~NM1~}~]~[2300B~{~RMR~[~DTM~
```

注釈 X12 内のループにはカスタム名を割り当てることができます。上記の HIPAA_4010:820 の未加工定義例では、ループ・タイトルは左側の括弧記号 “[{ (” の直後に記述されているのに対して、セグメント名はチルダ “~” によってこれらの括弧記号と区切られていることに注目してください。カスタム・ループ名を使用するには、スキーマを InterSystems IRIS にインポートする前に、スキーマ・ファイル内でデフォルトのループ名の代わりにカスタム名を入力します。

2.1.2.3 レガシー・ドキュメント構造

[検証構造に移動] をクリックすることで、スキーマの旧ドキュメント構造ビューを表示できます。レガシー・ドキュメント構造では、デフォルトのドキュメント構造ビューでは得られない情報も提供されます。例えばレガシー・ドキュメント構造では、コード・テーブル、許可されているループ反復数、および構文の注意事項を提供できます。

2.1.2.4 実装ガイド

このコードは、<http://www.wpc-edi.com/> で提供されている関連する ASC X12 **実装ガイド** を識別します。この識別子は、すべてのドキュメント構造およびスキーマにわたって一意です。

2.1.3 セグメント構造の表示

ドキュメント・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、そのドキュメント・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 InterSystems IRIS により表示されます。これは [スキーマ・セグメント構造] ページです。

例えば、HIPAA_4010:276 ドキュメント構造内の PER セグメントをクリックすると、InterSystems IRIS では次のページが表示されます。

Schema Category: [HIPAA_4010](#)
 Document Structure (Transaction Set): [276](#)
 Segment Structure: **PER**

Path you followed to get to this segment structure: **loop2000A().loop2100A().PER**

HIPAA_4010:PER Administrative Communications Contact

Elem	Description	Property Name	Data Type	Required	Length	Max Repeats	Alternate Description
1	Contact Function Code	ContactFunctionCode	(ID)	O	2	1	
2	Name	Name	(AN)	O	1-60	1	
3	Communication Number Qualifier	CommunicationNumberQualifier	(ID)	O	2	1	
4	Communication Number	CommunicationNumber	(AN)	O	1-80	1	
5	Communication Number Qualifier	CommunicationNumberQualifier2	(ID)	O	2	1	
6	Communication Number	CommunicationNumber2	(AN)	O	1-80	1	
7	Communication Number Qualifier	CommunicationNumberQualifier3	(ID)	O	2	1	
8	Communication Number	CommunicationNumber3	(AN)	O	1-80	1	
9	Contact Inquiry Reference	ContactInquiryReference	(AN)	O	1-20	1	

列は以下のとおりです。

- ・ **【要素】** – セグメント内の当該要素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】** – 当該要素の簡単な説明。
- ・ **【データ型】** – 当該要素のデータ型を表す 1 文字または 2 文字の略称。詳細は、次のテーブルを参照してください。

SYMBOL	TYPE
Nn	Numeric
R	Decimal
ID	Identifier
AN	String
DT	Date
TM	Time
B	Binary

- ・ **【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。
- ・ **【長さ】** – 当該要素に含めることができる文字数。1 つの数字のみが表示されている場合は、その数字は文字の最大数を表します。2 つの数字がハイフンで区切られて表示されている場合は、それらの数字は該当する要素に含めることができる文字数の範囲を表します (最少文字数-最大文字数)。
- ・ **【最大繰り返し回数】** – 当該要素を繰り返すことができる最大回数。
- ・ **【代替説明】** – 当該要素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:elem という形式で InterSystems IRIS の仮想プロパティ・パスを作成できます。以下は、HIPAA_4010:276 ドキュメント構造内の **PER** セグメントにある単純な elem 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。

```
loop2000A().loop2100A().PER:ContactFunctionCode  
loop2000A().loop2100A(1).PER:ContactFunctionCode  
loop2000A().loop2100A(2).PER:ContactFunctionCode  
loop2000A().loop2100A(x).PER:ContactFunctionCode  
loop2000A().loop2100A().PER:Name
```

2.1.4 複合構造の表示

[複合構造] 列で名前を選択すると、そのデータ構造内のすべての要素が表示されます。これは [複合構造] ページです。ここでの列の値は、[前の節](#)の列の値と同じです。

2.1.5 異なるカテゴリの選択

X12 標準の特性として、ドキュメント構造は X12 のバージョンによって異なる場合があります、構造の名前と番号が同じであっても同様です。例えば、X12 HIPAA_4010 と X12 HIPAA_5010 の両方で 277 というドキュメント構造が定義されていますが、これらの定義には、セグメントとセグメントの繰り返しでいくつかの差異があります。InterSystems IRIS では、HIPAA_4010:277 と HIPAA_5010:277 というドキュメント構造定義が用意されています。[X12 ドキュメント構造] ページでは、これら 2 つの定義の違いを簡単に確認できます。次の 2 つの図ではこれらの定義の違いを示しています。

2.2 X12 ドキュメント・ビューワ・ページの使用法

InterSystems IRIS では、X12 用の [ドキュメントビューワ] ページが用意されています。このページを使用して、X12 ドキュメントを表示、変換、およびエクスポートできます (プロダクション・メッセージ・アーカイブ内のドキュメントまたは外部ファイル)。

このページにアクセスするには、[Interoperability]→[相互運用]→[ASC X12]→[ASC X12 ドキュメント・ビューワ] を選択します。

2.2.1 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・ドキュメント数] に、表示するドキュメント数を入力します。
 - ・ [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。
 - ・ [スキーマのカテゴリまたはバージョンを使用] – 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] – 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] – 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] – 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ドキュメントには適用されません)。
 - ・ [なし] – 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として表示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。

2.2.2 ドキュメントの解析

ドキュメントを解析するには、前述のオプションを設定して、[OK] をクリックします。ドキュメント・ビューワの画面右側に以下が表示されます。

- ・ 以下の基本情報が含まれたドキュメントのサマリ。
 - 適用されているデータ変換 (該当する場合)
 - ドキュメント ID
 - DocType
 - DocType の説明 (ある場合)
 - セグメントの数
 - 親ドキュメントのドキュメント ID (該当する場合)
- ・ ドキュメント構造内のセグメントごとに 1 行が含まれたドキュメント・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 (ISA や DN1 など)
 - 要素のコンテンツと区切り文字 (ドキュメントに含まれているもの)

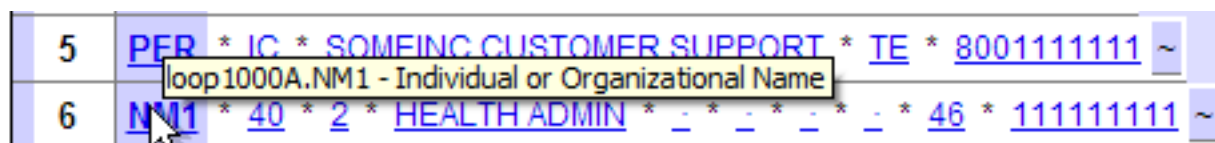
当該ドキュメントが選択されたスキーマと一致する場合は、セグメントと要素は関連する構造ページへのリンクとして表示されます。

注釈 同じトランザクション・セット識別子コード (ST:1) を使用する複数の X12 DocType 構造の多重度のために、InterSystems IRIS では、ドキュメント・タイプを一意に識別するために、追加の参照識別番号 ST:3、GS:8、または REF:2 を確認し、最初に検出された実装ガイドのバージョン番号を使用します。例えば、HIPAA_4010:837P と HIPAA_4010:837D について見てみましょう。これらのドキュメントのトランザクション・セット識別子はどちらも 837 ですが、それぞれのドキュメントの参照識別番号は異なります。“[実装ガイド](#)”を参照してください。

2.2.2.1 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。

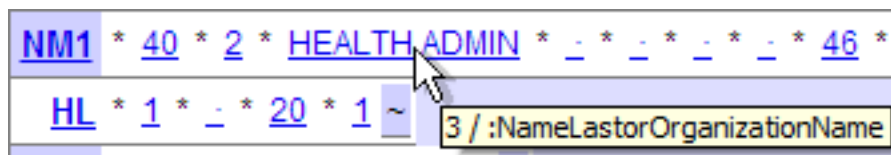
- ・ 仮想プロパティ・パスで使用するセグメント・アドレス
- ・ このセグメントの説明的な名前



2.2.2.2 要素アドレスの表示

要素アドレスを表示するには、ドキュメント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 element アドレス (数値)
- ・ 仮想プロパティ・パスで使用する element アドレス (名前)

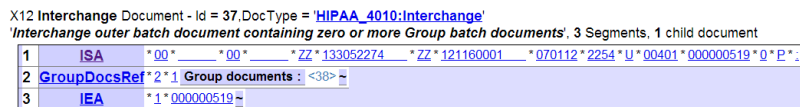


2.2.2.3 X12 インターチェンジの表示

グループ内とインターチェンジ内でネストされているトランザクション・セットを表示する際は、[Interoperability]→[相互運用]→[ASC X12]→[ASC X12 ドキュメント・ビューワ] ページを使用して、ドキュメント構造を 1 レベルずつ確認できます。

次の画面は、X12 ドキュメント・ビューワを使用して 4010:Interchange ドキュメントを表示した結果を示しています。

ドキュメント・ビューワは、グループ・ドキュメントに <38> という識別子を割り当てています。



X12 インターチェンジ・ドキュメント画面でグループ・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グループ・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 [ドキュメントビューワ] ウィンドウは、元のブラウザ・ウィンドウで開かれたままになります。

次の画面は、グループ・ドキュメント <38> へのリンクをクリックした結果を示しています。

注釈 単一のインターチェンジ内に 10 を超えるグループがある場合や、単一のグループ内に 10 を超えるトランザクション・セットがある場合は、ドキュメント・ビューワには最初の 5 つのドキュメントと最後の 5 つの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、[その他] をクリックします。新しいブラウザ・ウィンドウが開き、ドキュメントが表示されます。

X12 Group Document - Id = 38, DocType = 'HIPAA_4010:Group'
 'Group batch document containing TransactionSet documents of a given type', 3 Segments, 10 child documents, parent document : <37>

1	GS	* HC * 133052274 * 121160001 * 20070112 * 2254 * 000000519 * X * 004010X097A1 ~
2	TransactionSetDocsRef	* 2 * 10 TransactionSet documents : <39> <40> <41> <42> <43> <44> <45> <46> <47> <48> ~
3	GE	* 10 * 000000519 ~

X12 TransactionSet Document Id = 39
 Type Name = '837'

次の画面は、トランザクション・セット<39>をクリックした結果を示しています。グループまたはインターチェンジに戻るには、それぞれのドキュメント ID 番号のリンクをクリックします。

X12 837 Document - Id = 39, DocType = 'HIPAA_4010:837D'

'Health Care Claim: Dental', 32 Segments, parent documents : <38>: <37>

1	ST	* 837 * 0000000001 ~
2	BHT	* 0019 * 00 * 0000000001 * 20070112 * 2254 * CH ~
3	REF	* 87 * 004010X097A1 ~
4	NM1	* 41 * 2 * SOMEINC * _ * _ * _ * _ * 46 * 111111111 ~
5	PER	* IC * SOMEINC CUSTOMER SUPPORT * TE * 8001111111 ~
6	NM1	* 40 * 2 * HEALTH ADMIN * _ * _ * _ * _ * 46 * 111111111 ~
7	HL	* 1 * _ * 20 * 1 ~
8	PRV	* BI * ZZ * 204E00000X ~
9	NM1	* 85 * 1 * Xxxxx * Xxxxx * _ * _ * _ * XX * 1111111111 ~
10	N3	* 1111 SOME ST ~
11	N4	* SOME CITY * CA * 11111 ~
12	REF	* EI * 1111111111 ~
13	REF	* G5 * 0001 ~
14	REF	* 1E * 1111111111 ~
15	REF	* LU * 1111111111 ~
16	HL	* 2 * 1 * 22 * 0 ~
17	SBR	* P * 18 * GRGRGRGRG * _ * _ * _ * 6 * _ * _ * CI ~
18	NM1	* IL * 1 * Xxxxxx * Xxxxx * X * _ * _ * MI * 111111 ~
19	N3	* 111 SOME AVE ~
20	N4	* SOME CITY * CA * 11111 ~
21	DMG	* D8 * 11111111 * M ~
22	NM1	* PR * 2 * HEALTH ADMIN * _ * _ * _ * _ * PI * 11111 ~
23	REF	* FY * NOCD ~
24	CLM	* 11111111 * 55 * _ * _ * _ * 11 : _ : 1 * Y * C * Y * Y ~
25	REF	* D9 * 1111111111 ~
26	NM1	* 82 * 1 * Xxxxx * Xxxxx * _ * _ * _ * XX * 11111111111 ~
27	PRV	* PE * ZZ * 204E00000X ~
28	REF	* EI * 1111111111 ~
29	LX	* 1 ~
30	SV3	* AD : D0150 * 55 * 11 * _ * _ * _ * 1 ~
31	DTP	* 472 * D8 * 17760704 ~
32	SE	* 32 * 0000000001 ~

バッチ・ドキュメント階層でドキュメントの確認を終了したら、最上位レベルの親ドキュメントが元の [ドキュメントビューワ] ウィンドウに残っている状態になるまで、すべてのポップアップ・ブラウザ・ウィンドウを閉じることができます。このウィンドウから、管理ポータル別のアクティビティに戻ることができます。

2.2.3 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] にチェックを付けます。
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。

2.3 X12 クラス

この節では、X12 ドキュメントを操作するために提供されているクラスを参考として列挙します。

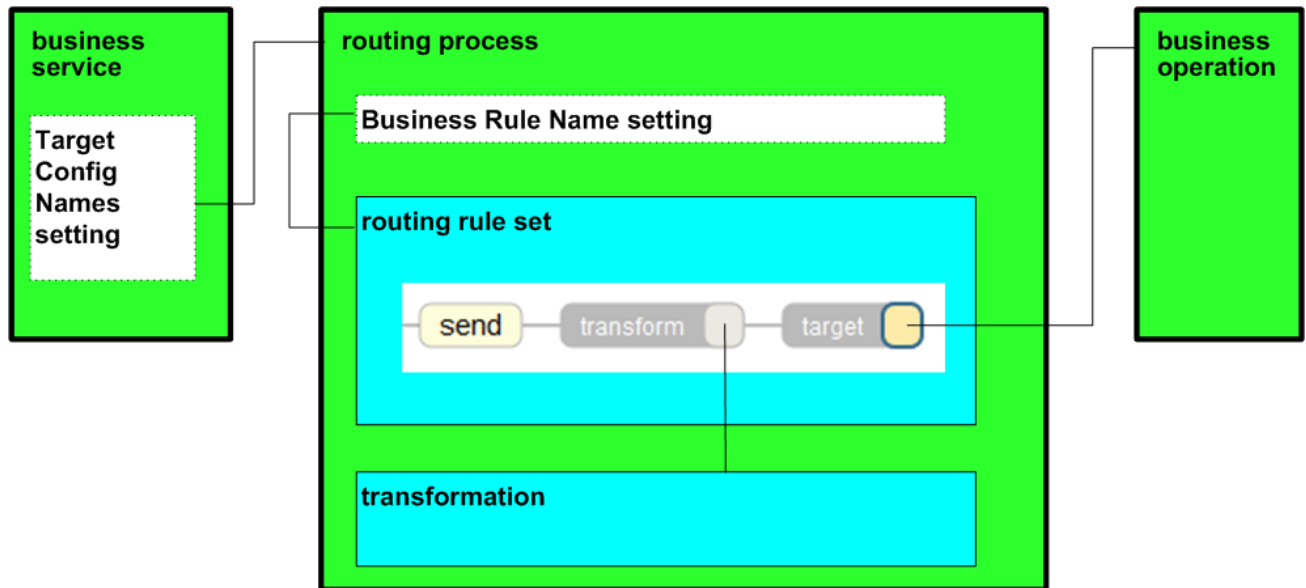
項目	クラス	メモ
X12 ビジネス・サービス*	<ul style="list-style-type: none"> EnsLib.EDI.X12.Service.FTPService EnsLib.EDI.X12.Service.FileService EnsLib.EDI.X12.Service.TCPService EnsLib.EDI.X12.Service.SOAPService 	クラス名が示しているように、これらの X12 ビジネス・サービス・クラスのそれぞれで別々のアダプタが使用されます。
X12 ビジネス・プロセス*	EnsLib.MsgRouter.VDocRoutingEngine	このクラスは標準の仮想ドキュメント・ビジネス・プロセスです。
X12 ビジネス・オペレーション*	<ul style="list-style-type: none"> EnsLib.EDI.X12.Operation.FTPOperation EnsLib.EDI.X12.Operation.FileOperation EnsLib.EDI.X12.Operation.TCPOperation EnsLib.EDI.X12.Operation.SOAPOperation 	クラス名が示しているように、これらの X12 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.EDI.Document (ビジネス・ホスト・クラスで自動的に使用されます)	X12 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.EDI.X12.SearchTable	X12 ドキュメント専用の検索テーブル・クラスです。

項目	クラス	メモ
検証とバッチ処理	<ul style="list-style-type: none"> EnsLib.EDI.X12.Util.Validator Ens.X12.FunctionSet 	これら 2 つのクラスを使用して、X12 バッチおよびドキュメントの検証と処理を行います。これには、応答の生成とバッチの二分割（検証に合格した子を含むバッチと検証に失敗した子を含むバッチ）が含まれます。

*X12 ドキュメントを操作するようにプロダクションを構成すると、管理ポータルで自動的に適切なビジネス・ホスト・クラスが選択されます。

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関する情報は、[設定のリファレンス](#)を参照してください。

2.4 カスタム X12 スキーマの作成

カスタム X12 スキーマは以下のいずれかの方法で作成できます。

- XML エディタを使用して .x12 ファイルを作成する。
- X12 スキーマ管理ポータル・ページを使用して既存のスキーマをインポートする。[Interoperability] > [相互運用] > [ASC X12] > [ASC X12 スキーマ構造] を選択して、[X12 スキーマ] ページを開きます。
- IDE で XML ファイルを作成し、そのファイルに拡張子 .x12 を付ける。

以上で、ファイルを編集して、<Category> ブロックに XML 文を追加したり、このブロックから XML 文を削除する準備ができました。編集の基本手順は以下のとおりです。

- <SegmentStructure> 要素を使用してカスタム・セグメントを定義します。
- カスタム・セグメントを含むカスタムの <MessageStructure> 要素を定義します。
- カスタム・メッセージ構造を含むカスタムの <MessageType> 要素を定義します。

4. 管理ポータルから新しいカテゴリ定義を表示します。“[ポータル・ツール](#)”を参照してください。

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。

2.5 X12 検索テーブルの定義

X12 検索テーブル・クラスの `EnsLib.EDI.X12.SearchTable` は、自動的に X12 ドキュメント ID のインデックスを作成し、`Identifier` という名前を付けます。

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。

すべてのタスクを、対象のプロダクションが含まれているネームスペースで実行する必要があります。予約パッケージ名は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。“プロダクション内での仮想ドキュメントの使用法”の“[検証ロジックの上書き](#)”も参照してください。

3

プロダクションの構成

ここでは、X12 プロダクションを作成して構成するプロセスを説明します。

すべてのタスクを、対象のプロダクションが含まれているネームスペースで実行する必要があります。ルール・セットと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。

“プロダクション内での仮想ドキュメントの使用法”の“[検証ロジックの上書き](#)”も参照してください。

3.1 新しい X12 プロダクションの作成

X12 コンポーネントを既存のプロダクションに追加できます。ただし、X12 を処理するための新しいプロダクションを明示的に作成する場合は、以下の手順を実行してください。

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・バーで現在のネームスペースの名前をクリックし、表示されるリストで該当するネームスペースをクリックします。
2. [Interoperability] をクリックします。
3. [構成する] をクリックします。
4. [プロダクション] をクリックし、[進む] をクリックします。
最後にアクセスしたプロダクションが、InterSystems IRIS® によって [プロダクション構成] ページに表示されます。
5. [プロダクション設定] メニューで [アクション] タブをクリックします。
6. [新規作成] をクリックして、プロダクション・ウィザードを呼び出します。
7. [パッケージ名]、[プロダクション名]、および [説明] に入力します。
8. [汎用] プロダクション・タイプを選択して、[OK] をクリックします。

InterSystems IRIS では空のプロダクションが作成され、このプロダクションから、ビジネス・サービス、ビジネス・プロセス、ビジネス・オペレーションなどのコンポーネントを追加できます。詳細は、後続の節を参照してください。

注釈 プロダクションの作成時に、あるコンポーネントを構成しているときに、まだ作成していない別のコンポーネント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。ルールについては、“[構成名](#)”を参照してください。

3.2 X12 ビジネス・サービスの追加

プロダクションが X12ドキュメントを受信する元となるアプリケーションごとまたはソースごとに、1 つの X12 ビジネス・サービスを追加します。

X12 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. ビジネス・サービス・ウィザードに[通常どおり](#)アクセスします。“[プロダクションの構成](#)”を参照してください。
2. [X12 入力] タブをクリックします。
3. [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
 - ・ File
 - ・ FTP
 - ・ SOAP
4. [X12 サービス名] フィールドに、このビジネス・サービスの名前を入力します。
5. [X12 サービスターゲット] 領域で、以下のいずれかのオプションを選択します。
 - ・ [新規ルータ作成] – InterSystems IRIS によってビジネス・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、このビジネス・プロセスの詳細を編集できます。
 - ・ [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。
 - ・ [今はなし] – このビジネス・サービスのターゲットを指定しないでください。このオプションを選択した場合は、後で必ずターゲットを指定してください。
6. [OK] をクリックします。

3.3 X12 ビジネス・プロセスの追加

X12 ビジネス・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. ビジネス・プロセス・ウィザードに[通常どおり](#)アクセスします。“[プロダクションの構成](#)”を参照してください。
2. [X12 ルータ] タブをクリックします。ルータ・クラスのデフォルト値は `EnsLib.EDI.X12.MsgRouter.RoutingEngine` です。
3. [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。
後で、[ルーティング・ルール](#)を編集して、ルーティング・ルールにロジックを追加する必要があります。
4. [X12 ビジネス・プロセス名] に、このビジネス・プロセスの名前を入力します。
5. [OK] をクリックします。

6. お使いの [X12 ビジネス・サービス](#)がこの新しい X12 ビジネス・プロセスに接続されていることを確認します。このプロセスを接続するには、以下の操作を行います。
 - ・ 対象の X12 ビジネス・サービスを選択します。
 - ・ 画面の右側のメニューで、**[設定]** タブをクリックして **[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 X12 ビジネス・プロセスの名前を入力します。
7. 必要に応じて、ビジネス・プロセスの追加の設定を構成します。詳細は、[“X12 ビジネス・プロセスの設定”](#) を参照してください。

3.4 X12 ルーティング・ルールの追加

ビジネス・ルールの定義に関する一般情報は、[“ビジネス・ルールの開発”](#) を参照してください。

X12 ルーティング・ルールを作成する際は、以下の操作を行います。

- ・ **[一般]** タブで、**[ルールタイプ]** に **[仮想ドキュメント・メッセージ・ルーティング・ルール]** を指定する必要があります。この選択によって以下のオプションが設定されます。
 - － **[ルール・アシスト・クラス]** は `EnsLib.MsgRouter.VDocRuleAssist` に設定されます。
 - － **[コンテキスト・クラス]** は `EnsLib.MsgRouter.VDocRouting Engine` に設定されます。
- ・ ルールの **[制約]** で、**[メッセージ・クラス]** に `EnsLib.EDI.X12.Document` を指定します。

これ以外のすべての点については、両タイプのルール・セットの構造と構文は同じです。

3.5 X12 ビジネス・オペレーションの追加

プロダクションからファイルまたはアプリケーションに X12 メッセージを送信するには、X12 ビジネス・オペレーションを追加する必要があります。出力先ごとに 1 つずつの X12 ビジネス・オペレーションを追加します。

不正なメッセージを処理するためのビジネス・オペレーションを追加することもできます（背景については、[“プロダクション内での仮想ドキュメントの使用法”](#) の [“仮想ドキュメント用のビジネス・プロセス”](#) を参照してください）。

X12 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. ビジネス・オペレーション・ウィザードに [通常どおり](#) アクセスします。[“プロダクションの構成”](#) を参照してください。
2. **[X12 出力]** タブをクリックします。
3. **[出力タイプ]** リストから以下のいずれかをクリックします。
 - ・ TCP
 - ・ File
 - ・ FTP
 - ・ SOAP
4. **[X12 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。
5. **[OK]** をクリックします。

6. このビジネス・オペレーションが関連するビジネス・サービスまたはビジネス・プロセスに接続されていることを確認します。
 - ・ ルーティング・ルールの場合は、X12 ビジネス・オペレーションの名前をルーティング・ルール・セットの[ターゲット] フィールドに入力します。
 - ・ 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・インターフェースを使用する設計を採用している場合は、X12 ビジネス・オペレーションの名前を X12 ビジネス・サービスの [ターゲット構成名] フィールドに入力します。
7. 必要に応じて、ビジネス・オペレーションの追加の設定を構成します。詳細は、“X12 ビジネス・オペレーションの設定” を参照してください。

X12 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 “接続オプション” も参照してください。

4

X12 データ変換の作成

ルーティング・ルールに、1 つ以上のデータ変換が必要な場合があります。データ変換言語 (DTL) データ変換の定義に関する一般情報は、“[DTL 変換の開発](#)”を参照してください。

DTL データ変換をプロダクションに統合するには、[ルーティング・ルール](#)の[変換]フィールドに、そのデータ変換の完全なパッケージ名とクラス名を入力します。

X12 ドキュメント用の DTL データ変換を作成する場合は、次の手順に従います。

- ・ [変換] タブで、[ソースクラス] と [ターゲットクラス] の両方で `EnsLib.EDI.X12.Document` を指定する必要があります。
- ・ [ソースドキュメントタイプ] には、ビジネス・サービスによって割り当てられたスキーマ・カテゴリ名を指定する必要があります。
- ・ [ターゲットドキュメントタイプ] には、ターゲット・スキーマ・カテゴリの名前を指定する必要があります。これは、InterSystems IRIS にロードしたスキーマ・カテゴリ名と同じにする必要があります。

4.1 一般的な X12 DTL 使用事例

DTL エディタにより、以下のように、一般的な X12 使用事例でのデータ変換の作成が容易になります。

- ・ 複数の機能グループを持つことができます。
- ・ 通常、1 つのグループでは、1 タイプのトランザクション・セットのみが使用されます。

一般に、各 DTL には、入力メッセージ用に 1 つのスキーマ、出力メッセージ用に 1 つのスキーマがあります。これは、単一のスキーマで表されているメッセージには機能しますが、X12 では 3 つの異なるレベルに対して以下のように異なるスキーマが存在します。

- ・ インターチェンジ・エンベロープのスキーマ
- ・ 機能グループのスキーマ
- ・ トランザクション・セットのスキーマ

DTL では、レベルごとに 1 つのサブ変換を用意することでこれを処理しますが、DTL X12 サポートにより、階層内の各レベルにスキーマを指定できます。複数タイプのトランザクション・セットを持つメッセージなど、より複雑な X12 メッセージを扱う場合は、サブ変換を使用してこれを処理する必要があります。

4.2 X12 DTL を作成する手順

このセクションでは、ソース X12 ドキュメントの内容をターゲットの X12 ドキュメントにコピーするデータ変換を作成する方法について説明します。例では、ソースとターゲットの両方のドキュメントに HIPAA 5010 834 ドキュメント構造を使用します。他の形式にも同様の手順を使用できます。

“[X12 エンベロープの構造](#)” で説明している X12 メッセージの階層構造により、[変換] タブの [作成] フィールドの値を単に [コピー] に設定することはできません。代わりに、メッセージ構造の各レベルをコピーするデータ変換を構築します。Ens.Rule.FunctionSet のサブクラスであるクラス Ens.X12.FunctionSet は、このタスクに役立つ以下のクラス・メソッドを提供します。

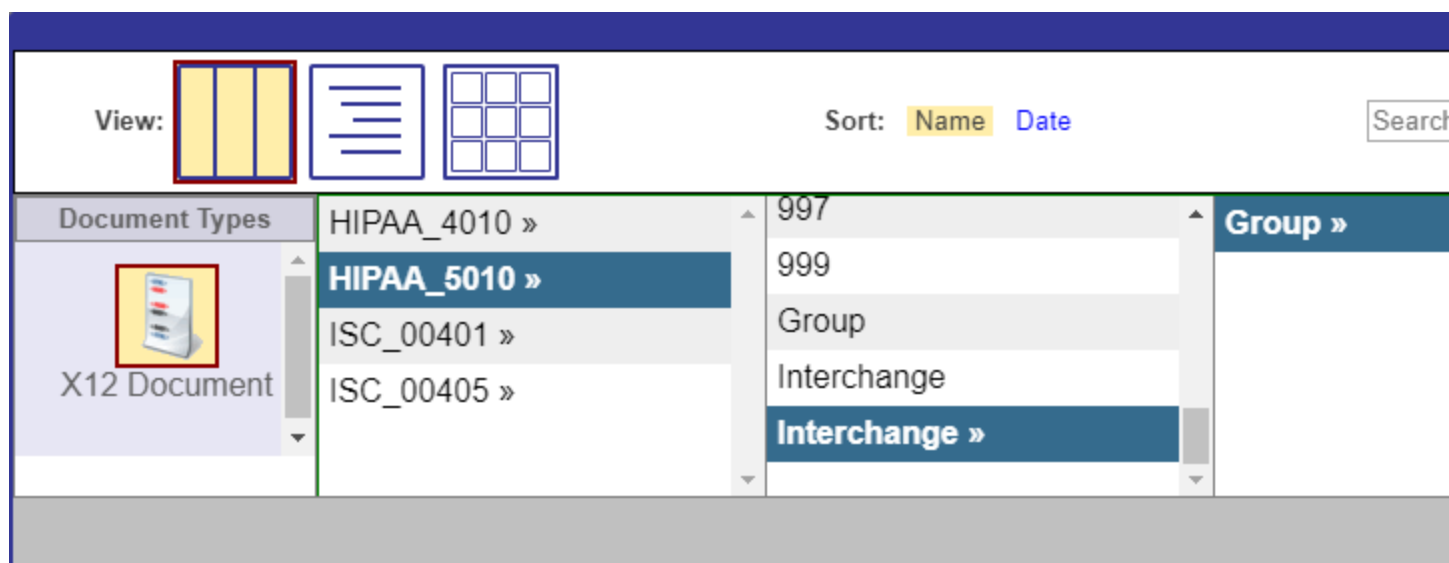
- ・ X12NewFunctionalGroup()
- ・ X12NewTransactionSet()
- ・ X12CompleteTrailerAndReturnClear()
- ・ X12SaveDocumentAndReturnClear()

4.2.1 ソースおよびターゲット・クラスの作成

まず、適切なドキュメント・タイプのソースおよびターゲットのデータ変換クラスを作成します。

1. [データ変換ビルダ] ページで [新規] をクリックしてデータ変換ウィザードを開きます。
2. [パッケージ] フィールドで新しいクラスが属するパッケージの名前を入力するか、矢印をクリックして現在のネームスペースでパッケージを選択します。
3. [名前] フィールドに、新しいデータ変換クラスの名前を入力します。
4. [説明] フィールドに、データ変換の説明を入力します。
5. [ソースタイプ] と [ターゲットタイプ] の両方で [X12] を選択します。このアクションにより、[ソースクラス] と [ターゲットクラス] に自動的に EnsLib.EDI.X12.Document が入力されます。
6. [ソース・ドキュメント・タイプ] を選択するには、虫めがねアイコンをクリックします。
 - ・ 表示されるダイアログ・ボックスで、HIPAA_5010 などの適切なスキーマ・カテゴリを選択します。
 - ・ 右の列で、[Interchange>>] を選択します。
 - ・ その右の列で、[グループ>>] を選択します。これは機能グループのスキーマです。
 - ・ その右の列で、ドキュメント・タイプを表す名前 (この例では 834) を選択します。これはトランザクション・セットのスキーマです。
7. これらのアクションにより、[ソース・ドキュメント・タイプ] として HIPAA_5010:Interchange:Group:834 が指定され、デフォルトで同じタイプが [ターゲット・ドキュメント・タイプ] に指定されます。
8. [OK] をクリックすると、新しいソースとターゲットのクラスが作成されます。

以下の図は、[ソース・ドキュメント・タイプ] の選択手順を示したものです。



4.2.2 インターチェンジ・コントロール・ヘッダのコピー

ソースからターゲットへ、インターチェンジ・コントロール・ヘッダ (ISA) の内容をコピーできます。“DTL 変換の開発”の“ソース・プロパティからターゲット・プロパティへの値のコピー”で説明しているドラッグ・アンド・ドロップ手法を使用して、[set] アクションを作成できます。これらのアクションを [group] アクションにまとめることもできます。

4.2.3 機能グループのコピー

1. [アクション追加] をクリックして [group] を選択して、新しい group アクションを作成します。
2. ダイアグラムで [FunctionalGroups()] ノードを選択します。
3. [アクション追加] をクリックして [for each] を選択して、新しい for each アクションを作成します。[FunctionalGroups()] を選択していれば、[プロパティ] フィールドのデフォルト値が指定されます。
4. [アクション] タブにある新しい for each アクションの [プロパティ] フィールドにはデフォルト値 `source.FunctionalGroups()` が入力されています。これは、for each にソース・クラス内の機能グループを繰り返すよう指示します。
5. [キー] フィールドには、ループを繰り返すのに使用される変数が含まれます。このデフォルト値は `k1` です。これを覚えやすい変数名 (機能グループ・キーの場合 `fgKey` など) に変更します。
6. [アクション追加] をクリックして [set] を選択して、for each ループで新しい set アクションを作成します。[プロパティ] フィールドで `target.FunctionalGroups(fgKey)` を追加します。
7. [値] フィールドを編集するには、虫めがねアイコンをクリックして [値編集] を開きます。
8. [値編集] ダイアログ・ボックスで、[関数] フィールドの右の矢印をクリックし、ドロップ・ダウン・リストを開きます。関数 `X12NewFunctionalGroup()` を選択します。
9. [InterchangeParent] フィールドで `target` と入力し、[DocType] フィールドで “Group” と入力して [OK] をクリックします。
10. このアクションにより、[値] フィールドに `##class(Ens.X12.FunctionSet).X12NewFunctionalGroup(target, "Group")` が追加されます。set アクション全体で `X12NewFunctionalGroup()` メソッドを使用し、ターゲット・クラスに機能グループを作成します。

次の手順ではトランザクション・セットをコピーします。

4.2.4 トランザクション・セットのコピー

1. 前のセクションの for each アクション内で、[アクション追加] をクリックし、[group] を選択することにより、新しい group アクションを作成します。
2. ダイアグラムで [TransactionSets()] ノードを選択します。
3. [アクション追加] をクリックして [for each] を選択して、新しい for each アクションを作成します。[TransactionSets()] を選択していれば、[プロパティ] フィールドのデフォルト値が指定されます。
4. [アクション] タブにある新しい for each アクションの [プロパティ] フィールドには値 `source.FunctionalGroups(fgKey).TransactionSets()` が入力されています。これは、for each にソース・クラス内のトランザクション・セットを繰り返すよう指示します。
5. [キー] フィールドには、ループを繰り返すのに使用される変数が含まれます。このデフォルト値は k2 です。これを覚えやすい変数名 (トランザクション・セット・キーの場合 tsKey など) に変更します。
6. [アクション追加] をクリックして [set] を選択して、新しい set アクションを作成します。[プロパティ] フィールドで `target.FunctionalGroups(fgKey).TransactionSets(tsKey)` を追加します。
7. [値] フィールドを編集するには、虫めがねアイコンをクリックして [値編集] を開きます。
8. [値編集] ダイアログ・ボックスで、[関数] フィールドの右の矢印をクリックし、ドロップ・ダウン・リストを開きます。関数 `X12NewTransactionSet()` を選択します。
9. [GroupParent] フィールドで `target.FunctionalGroups(fgKey)` と入力し、[DocType] フィールドで "834" と入力して [OK] をクリックします。
10. このアクションにより、[値] フィールドに

```
##class(Ens.X12.FunctionSet).X12NewTransactionSet(target.FunctionalGroups(fgKey),"834")
```

 が追加されます。set アクション全体で `X12NewTransactionSet()` メソッドを使用し、ターゲット・クラスにトランザクション・セットを作成します。

4.2.5 機能グループおよびトランザクション・セットのループの完了

前の 2 つのセクションでは、ソース内で機能グループを繰り返し、ターゲット内で対応する機能グループを作成するループと、各機能グループで必要なトランザクション・セットを作成する 2 つ目の入れ子のループの 2 つのループを作成しました。ここで、各トランザクション・セットとそれらを含む各機能グループに、トレーラ・セグメントを作成する必要があります。

トランザクション・セットのトレーラ・セグメントを作成するには、以下の手順に従います。

1. トランザクション・セットを作成するループの末尾の直前の位置で、[アクション追加] をクリックし、[set] を選択することにより、新しい set アクションを作成します。[プロパティ] フィールドで `target.FunctionalGroups(fgKey).TransactionSets(tsKey)` を追加します。
2. [値] フィールドを編集するには、虫めがねアイコンをクリックして [値編集] を開きます。
3. [値編集] ダイアログ・ボックスで、[関数] フィールドの右の矢印をクリックし、ドロップ・ダウン・リストを開きます。関数 `X12CompleteTrailerAndReturnClear()` を選択します。
4. [X12Document] フィールドで `target.FunctionalGroups(fgKey).TransactionSets(tsKey)` と入力し、[保存] フィールドで 1 と入力して [OK] をクリックします。
5. このアクションにより、[値] フィールドに

```
##class(Ens.X12.FunctionSet).X12CompleteTrailerAndReturnClear(target.FunctionalGroups(fgKey).TransactionSets(tsKey),1)
```

 が追加されます。set アクション全体で `X12CompleteTrailerAndReturnClear()` メソッドを使用し、現在のトランザクション・セットにトランザクション・セット・トレーラを作成します。

機能グループのトレーラ・セグメントを作成するには、以下の手順に従います。

1. 機能グループを作成するループの末尾の直前の位置で、[アクション追加] をクリックし、[set] を選択することにより、新しい set アクションを作成します。[プロパティ] フィールドで `target.FunctionalGroups(fgKey)` を追加します。
2. [値] フィールドを編集するには、虫めがねアイコンをクリックして [値編集] を開きます。
3. [値編集] ダイアログ・ボックスで、[関数] フィールドの右の矢印をクリックし、ドロップ・ダウン・リストを開きます。関数 `X12CompleteTrailerAndReturnClear()` を選択します。
4. [X12Document] フィールドで `target.FunctionalGroups(fgKey)` と入力し、[保存] フィールドで 1 と入力して [OK] をクリックします。
5. このアクションにより、[値] フィールドに

```
##class(Ens.X12.FunctionSet).X12CompleteTrailerAndReturnClear(target.FunctionalGroups(fgKey),1))
```

 が追加されます。set アクション全体で `X12CompleteTrailerAndReturnClear()` メソッドを使用し、現在の機能グループに機能グループ・トレーラを作成します。

最後に、IEA セグメントを作成します。

1. 機能グループを作成するループの末尾の直後の位置で、[アクション追加] をクリックし、[set] を選択することにより、新しい set アクションを作成します。[プロパティ] フィールドで `temp` を追加します。
2. [値] フィールドを編集するには、虫めがねアイコンをクリックして [値編集] を開きます。
3. [値編集] ダイアログ・ボックスで、[関数] フィールドの右の矢印をクリックし、ドロップ・ダウン・リストを開きます。関数 `X12CompleteTrailerAndReturnClear()` を選択します。
4. [X12Document] フィールドで `target` と入力し、[保存] フィールドで 0 と入力して [OK] をクリックします。
5. このアクションにより、[値] フィールドに

```
##class(Ens.X12.FunctionSet).X12CompleteTrailerAndReturnClear(target,0))
```

 が追加されます。set アクション全体で `X12CompleteTrailerAndReturnClear()` メソッドを使用し、IEA セグメントを作成します。

4.3 バッチ全体用のデータ変換

バッチ全体ドキュメントをグループ・レベルまたはトランザクション・セット・レベルで変換するには、各グループを反復処理すると共に、各グループ内の各トランザクション・セットを反復処理する必要があります。このためには、以下の手順を実行します。

4.3.1 各インターチェンジ内のグループのループ処理

1. [アクション追加] をクリックして [for each] を選択して、新しい for each アクションを作成します。
2. [プロパティ] フィールドに `[source].[GroupDocsRef]` と入力します。
3. [キー] フィールドに `[Group]` と入力します。Group は、実際のグループ・レベルのドキュメント・オブジェクトです。
4. 各グループにサブ変換を追加します。サブ変換に渡される値は、Group (for each からのキー) であり、プロパティは新しい Group (これは GroupOut という名前の可能性があります) になります。
5. メインの変換のソースとターゲットは、インターチェンジ・ドキュメントです。このサブ変換では、グループ・ドキュメントがソースおよびターゲットであるため、DTL のドラッグ・アンド・ドロップ機能により、グループ・セグメントを容易に操作でき、また個々のトランザクション・セットを容易にループできます。

4.3.2 各グループ内のトランザクション・セットのループ処理

1. グループ・サブ変換内で、[アクション追加] をクリックして [for each] を選択して、新しい for each アクションを作成します。
2. 次に、各トランザクション・セットで別のサブ変換を実行します。for each のキーであったトランザクション・セット・オブジェクトを渡します。返されるプロパティもトランザクション・セットです。このサブ変換には、ソースとターゲットのトランザクション・セットのドキュメント・タイプが含まれます (HIPAA_5010:834 または HIPAA_5010:837P など)。
3. バッチにさまざまなドキュメント・タイプのトランザクション・セットが存在する可能性がある場合、ドキュメント・タイプに条件を追加して、どのトランザクション・セット・サブ変換を実行するかを特定できます。このシナリオは、前のセクションで説明した手法ではなく、この手法を使用する理由にもなり得ます。

各グループ変換およびインターチェンジ変換で、サブ変換の後、ForEach のクローズ前に、返された子オブジェクトを親ターゲット・ドキュメントに追加する必要があります。これを行うにはカスタム関数が必要です。このカスタム関数は、以下のようなものにできます。

```
/// Adds a child X12 document <var>pChild</var>, (a Transaction Set or a Group)
as the child of a Group or Interchange

/// <var>pParent</var>'s DocsRef segment. <var>pDocsRefSeg</var>
is the segment name for the DocsRef segment

/// ("TransactionSetDocsRef" for a Group or "GroupDocsRef" for an Interchange)
Returns a status code.

ClassMethod AddChildToDocsRef(pChild As EnsLib.EDI.X12.Document,
  pParent As EnsLib.EDI.X12.Document, pDocsRefSeg As %String) As %Status
{
  Set tSC = $$$OK
  Set tSC = pParent.BuildMap() Quit:$$$ISERR(tSC) tSC
  Set index = pParent.GetSegmentIndex(pDocsRefSeg,.tSC) Quit:$$$ISERR(tSC) tSC
  Set tSC = pParent.AddChild(index,pChild) Quit:$$$ISERR(tSC) tSC
  Set tSC = pParent.SetValueAt(1+pParent.GetValueAt((index+1)_"":1),(index+1)_"":1)
  Quit:$$$ISERR(tSC) tSC
  Set tSC = pParent.%Save()
  Quit tSC
}
```

4.4 X12 データ変換のテスト

DTL エディタの [ツール] タブは、データ変換を迅速にテストできる [テスト] ボタンを備えています。[テスト] をクリックし、X12 テキストを [入力メッセージ] フィールドに貼り付け、[変換のテスト] ダイアログの [テスト] ボタンをクリックします。

“利用可能なツール”の説明に従って、[X12ドキュメント・ビューワ・ページ](#)を使用して変換をテストすることもできます。ドキュメント・ビューワでは、さまざまな X12 データ・ソースに対して DTL を実行できます。

注釈 データ変換には、[バッチハンドリング] の設定を必ず反映させてください。

5

X12 インターチェンジの処理

X12 ドキュメントがインターチェンジ内で送信される際は、InterSystems IRIS® によって受信ドキュメントが処理される方法を選択する必要があります。特に、グループ・レベルまたはトランザクション・セット・レベルでドキュメントにアクセスする必要がある場合は、このことが当てはまります。ここでは、X12 インターチェンジの処理に関するさまざまなシナリオを説明します。

5.1 X12 バッチ・ハンドリング

[バッチハンドリング] 設定では、InterSystems IRIS によるドキュメントの送受信方法を指定します。ここで選択した内容は、必要とされるビジネス・プロセス構成内容に影響を与えます。

5.1.1 バッチ・ドキュメントの受信

X12 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。選択項目は、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] – インターチェンジ内のすべてのドキュメントを単一セッションの一部として転送します。この転送内容には、バッチとグループのヘッダ・セグメントとトレーラ・セグメントが含まれた最終親ドキュメント・オブジェクトも含まれます。
- ・ [] – インターチェンジ内の各ドキュメントをそれぞれに固有のセッションで転送した後、バッチとグループのヘッダ・セグメントとトレーラ・セグメントが含まれた最終親ドキュメント・オブジェクトを転送します。
- ・ [] – バッチの各子ドキュメントを専用のセッションで転送します。親バッチ・ドキュメント・オブジェクトは転送しません。

注釈 [バッチ全体] を選択した場合は、個別のトランザクション・セットのすべての変換ではインターチェンジ・レベルとグループ・レベルが考慮される必要があります。後出の “[バッチ全体向けのビジネス・プロセスの構成](#)” を参照してください。

5.1.2 バッチ・ドキュメントの送信

送信側では、X12 [ファイル](#) および [FTP](#) ビジネス・オペレーションには [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。

ビジネス・サービスで [バッチハンドリング] = [個別] が使用されている場合など、末尾のセグメントが正しく記述されない状況では、ファイルおよび FTP ビジネス・オペレーションには [自動バッチ完了タイムアウト] 設定を使用することで、末尾のセグメントが書き込まれるようにすることができます。

5.1.3 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、X12 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。

[バッチハンドリング]	[親セグメント自動バッチ]	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現在の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。出力側のビジネス・オペレーションは同じです。
[]		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。出力側のビジネス・オペレーションは同じです。

5.2 バッチ全体向けのビジネス・プロセスの構成

ビジネス・サービスが [バッチ全体] モードで受信 X12 ドキュメントを処理するように設定した場合は、それに応じて適切にビジネス・プロセスを構成する必要があります。

5.2.1 バッチ全体用のルーティング・ルール

[バッチ全体] モードでドキュメントを適切にルーティングするには、ルーティング・ルールがインターチェンジを受け付けるように設定します。必要に応じて、関連するデータ変換に送信します。

5.2.1.1 新しいルーティング・ルールを作成

1. 適切なネームスペース内にいることを確認してから、[\[ルール・エディタ\]](#) ページに移動します。
2. [\[新規作成\]](#) をクリックします。
3. “[X12 ルーティング・ルールを追加](#)” の手順に従います。

5.2.1.2 制約の編集

1. [\[制約\]](#) ルール項目をダブルクリックします。
2. [\[ソース\]](#) で、お使いのビジネス・サービスを指定します。
3. [\[メッセージ・クラス\]](#) を `EnsLib.EDI.X12.Document` に設定します。
4. [\[スキーマ・カテゴリ\]](#) をお使いのプロダクションに適したものに設定します。
5. [\[ドキュメント名\]](#) オプションから [] を選択します。

詳細は、“[ルール制約エディタの使用](#)” を参照してください。

5.2.1.3 条件の定義

お使いのプロダクションに適した任意の条件文を選択します。当該アクションが常に適用されるようにするには、[\[条件\]](#) ルール項目を選択して “1” と入力します。

5.2.1.4 Send アクションの追加

1. [\[タイミング\]](#) ルール項目をクリックします。
2. [\[送信\]](#) をクリックして send アクションを追加します。
3. [\[変換\]](#) ルール項目をダブルクリックして、適切なデータ変換を選択します。
4. [\[ターゲット\]](#) ルール項目をダブルクリックして、プロダクションから適切なビジネス・オペレーションを選択します。

詳細は、“[send アクションの変換とターゲットの選択](#)” を参照してください。

5.3 他のバッチ設定向けのビジネス・プロセスの構成

単一セッション・モード、複数セッション・モード、または個別モードで、プロダクション内の X12 ドキュメントを変換するには、プロダクションで検出される X12 ドキュメントのタイプごとに別個のデータ変換を作成する必要があります。詳細は、“[DTL 変換の開発](#)” を参照してください。

5.3.1 他のバッチ設定用のルーティング・ルール

単一セッション・モード、複数セッション・モード、または個別モードでドキュメントを適切にルーティングするには、プロダクションで検出される X12 ドキュメントのタイプごとに一意のルーティング・ルールが含まれたルーティング・ルール・セットを作成します。

ルール・セット内に新しいルールを作成するには、以下の手順を実行します。

1. [\[ruleSet\]](#) をクリックします。
2. [\[ルール\]](#) をクリックしてルールを追加します。

3. “[バッチ全体用のルーティング・ルール](#)” の手順に従います。
4. 制約を編集する際は、必ず [ドキュメント名] を適切な X12 ドキュメント・タイプに設定してください。
5. データ変換を追加する際は、必ず適切な X12 ドキュメント・タイプ向けに記述された変換を使用してください。

5.3.2 他のバッチ設定用のデータ変換

単一セッション・モード、複数セッション・モード、または個別モードで、プロダクション内の X12 ドキュメントを変換するには、プロダクションで検出される X12 ドキュメントのタイプごとに別個のデータ変換を作成する必要があります。詳細は、“[X12 データ変換の作成](#)” および “[DTL 変換の開発](#)” を参照してください。

X12 の設定

このセクションでは、X12 ビジネス・ホストの参照情報を提供します。

ビジネス・プロセス (EnsLib.MsgRouter.VDocRoutingEngine) の設定に関する情報は、“プロダクション内での仮想ドキュメントの用法” の “[仮想ドキュメント・ルーティング・プロセスの設定](#)” を参照してください。

X12 ビジネス・サービスの設定

X12 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

X12 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	[ターゲット構成名] 、 [Docスキーマ・カテゴリ] 、 [バッチハンドリング]	“プロダクション内での仮想ドキュメントの使用法”の “ビジネス・サービスに関する設定”
応答確認	[リプライターゲット構成名] (ファイル)、 [検証モード] 、 [検証] 、 [SNIP レベル] 、 [バッチエラーアクション] 、 [バッチリプライタイプ] 、 [NACK ERR追加] 、 [リプライモード] 、 [不正メッセージ・ハンドラ]	このトピック内の節
接続設定	[新規ライン許容]	このトピック内の節
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 “ビジネス・サービスに関する設定”
	[リプライターゲット構成名] (FTP)	このトピック内の節
	[ローカルアプリケーションID] 、 [デフォルト文字エンコード]	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、以下の各ドキュメント内の設定のリファレンスのセクションを参照してください。

- 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 FTP アダプタの使用法”の[“FTP 受信アダプタの設定”](#)

EnsLib.EDI.X12.Adapter.TCPInboundAdapter の[\[接続毎のジョブ\]](#)が偽 (False) に設定されます。X12 の場合はこれが一般的です。

- “プロダクション内での SOAP サービスおよび Web クライアントの追加”の[“SOAP 受信アダプタに関する設定”](#)

NACK ERR追加

エラー・ステータス応答メッセージの生成時に、TA1 & 997/999 AKx セグメントに追加エラー・テキスト・フィールドを追加します。追加しない場合、確認応答メッセージに内部のエラー状態情報は組み込まれません。

不正メッセージ・ハンドラ

ドキュメントの検証が失敗すると、この設定で指定されている[\[不正メッセージ・ハンドラ\]](#)にそのドキュメントが送信されます。そのドキュメントがバッチの一部の場合、[バッチ・エラー・アクション](#)によって、バッチの処理方法が決められます。この設定は、[\[検証モード\]](#)、[\[検証\]](#)、[\[SNIP レベル\]](#)、および[\[バッチエラーアクション\]](#)の組み合わせで機能します。

[“不正メッセージ・ハンドラの定義”](#)を参照してください。

バッチエラーアクション

バッチ Interchange ドキュメントで検証エラーが検出された場合のサービスの対処方法を指定します。このオプションは以下のとおりです。

[]

バッチ内のドキュメントにエラーが検出された場合は、バッチの残りの部分も検証され、バッチ全体が拒否されます。

さらに、[バッチリプライタイプ] で許可されている場合は、応答ドキュメントには、すべてのエラーが列挙され、適宜、各機能グループ応答トレイラー (AK9) セグメントやトランザクション・セット応答トレイラー (AK5) セグメントに A (許可) または R (拒否) のいずれかの確認応答コードが表示されます。

バッチやバッチの一部がターゲットのビジネス・ホストに転送されることはありません。

[]

バッチ内のドキュメントにエラーが検出された場合は、バッチ全体が即座に拒否されます。他にエラーがないか確認されたり、バッチの残りの部分が解析されることはありません。

さらに、[バッチリプライタイプ] で許可されている場合は、応答ドキュメントには、エラーが列挙され、適宜、検証された各機能グループ応答トレイラー (AK9) セグメントやトランザクション・セット応答トレイラー (AK5) セグメントに A (許可) または R (拒否) のいずれかの確認応答コードが表示されます。トランザクション・セット応答ヘッダ (AK2) セグメントが、R が記された AK5 セグメントの後に表示されることはありません。R はエラーが発生したトランザクション・セットに相当します。

バッチやバッチの一部がターゲットのビジネス・ホストに転送されることはありません。

[]

検証に合格したバッチ内のドキュメントが、ターゲットのビジネス・ホストに転送されます。

エラーを含むトランザクション・セットはターゲット・ビジネス・ホストに送信されず、無効なトランザクション・セットは送信対象の機能グループから削除されます。ある機能グループのすべてのトランザクション・セットにエラーが含まれている場合、その機能グループは送信されません。同様に、ある interchange のすべての機能グループのすべてのトランザクション・セットにエラーが含まれている場合、その interchange は送信されません。

さらに、[バッチリプライタイプ] で許可されている場合は、応答ドキュメントには、エラーが列挙され、適宜、各機能グループ応答トレイラー (AK9) セグメントには A (許可)、P (一部許可)、または R (拒否) のいずれかの確認応答コードが、各トランザクション・セット応答トレイラー (AK5) セグメントには A または R のいずれかの確認応答コードが表示されます。

[] がデフォルト値です。

[]

バッチ内のすべてのドキュメントが転送されます。

さらに、[バッチリプライタイプ] で許可されている場合は、応答ドキュメントには、すべてのエラーが列挙され、適宜、各機能グループ応答トレイラー (AK9) セグメントやトランザクション・セット応答トレイラー (AK5) セグメントに A (許可) または E (受け入れられましたが、エラーが記載されました) のいずれかの確認応答コードが表示されます。

注釈 [リプライモード] が Application で、[バッチハンドリング] が Individual でない場合は、InterSystems IRIS® により、エラー検出時にバッチ全体が拒否される前に、バッチ内のドキュメントの一部が転送される可能性があります。

検証エラーのためにドキュメントが転送されない場合、そのドキュメントは削除されます。さらに、[バッチハンドリング] の値が Individual である場合、永続化された親ドキュメントは、対応するどの子ドキュメントも転送されないと削除されます。

バッチハンドリング

X12トランザクション・セットのドキュメントは、多くの場合 Interchange というバッチ・ドキュメントにパッケージ化されます。これには、機能グループというネストされたサブバッチが含まれています。[バッチハンドリング] 設定では、受信したドキュメント・バッチの InterSystems IRIS での処理方法について指定します。この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、親ドキュメントのヘッダ・セグメントとトレーラ・セグメントを表すオブジェクトが含まれています。これがデフォルトです。
- ・ [] – バッチ内の各ドキュメントを個別のセッションで転送します。各セッションには、バッチのヘッダ・セグメントとトレーラ・セグメントを表すオブジェクトが含まれています。
- ・ [] – 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。親バッチ・ドキュメントのヘッダ・セグメントとトレーラ・セグメントを表すオブジェクトは転送されません。

バッチリプライタイプ

受信した Interchange バッチに対して作成するバッチ応答のタイプを指定します。以下のテーブルに可能な選択肢を列挙します。

値	意味
[なし]	バッチ応答を生成しません。エラーが発生しても、送信元への即時通知応答を作成しません。
[すべて]	Interchange で受信されたトランザクション・セットごとに、応答通知を含む応答 Interchange を生成します。
[すべて + TA1]	Interchange 全体の承認またはエラー・ステータスを示す TA1 セグメントおよび Interchange で受信された各トランザクション・セットの応答通知を含む、応答 Interchange を生成します。
[すべて + ISA14TA1]	インターチェンジ・コントロール・ヘッダ (ISA) セグメントの確認応答要求 (ISA:14) フィールドに 1 が表示される場合、またはインターチェンジ・エンベロープにエラーが表示される場合にのみ、TA1 セグメントを含む応答 Interchange と各トランザクション・セットの応答を生成します。または、各トランザクション・セットに対する応答を含む実装の確認応答 (999) ドキュメントを生成します。
[エラー]	エラーが検出されたかどうかにかかわらず、応答 Interchange を生成します。エラーが検出されなかった場合は、空の応答 Interchange を生成します。エラーが検出された場合は、エラーが検出されたトランザクション・セットに対してのみ、応答通知を含む Interchange を生成します。 選択項目が指定されていない場合は、これがデフォルトの設定です。
[OnlyIfErrors]	エラーが検出された場合は、エラーが検出されたトランザクション・セットに対してのみ、応答通知を含む応答 Interchange を生成します。
[成功]	エラーが検出されたかどうかにかかわらず、応答 Interchange を生成します。エラーがすべてのトランザクション・セットで検出された場合は、空の応答 Interchange を生成します。それ以外の場合は、エラーが検出されていない (正常な) トランザクション・セットに対してのみ、応答通知を含む応答 Interchange を生成します。

値	意味
[TA1]	受信された Interchange 全体の承認またはエラー・ステータスを示す TA1 セグメントのみを含む応答 Interchange を生成します。
[OnlyIfErrorTA1]	エラーが検出された場合は、受信された Interchange 全体に対するエラー・ステータスを示す TA1 セグメントのみを含む応答 Interchange を生成します。
[ISA14-TA1]	受信 ISA ヘッダ・セグメントの ISA:14 フィールドが 1 に設定されている場合は、TA1 セグメントのみを含む応答 Interchange を生成します。1 に設定されていない場合は、何も返しません。
[ISA14-OnlyIfErrorTA1]	エラーが検出されたうえ、受信 ISA ヘッダ・セグメントの ISA:14 フィールドが 1 に設定されている場合は、エラーの TA1 セグメントのみを含む応答 Interchange を生成します。それ以外の場合は、何も返しません。
[バイト]	1 文字のコードで構成した応答を生成します。Interchange 全体が承認された場合は「A」、1 つまたは複数のエラーにより拒否された場合は「R」です。

TA1 セグメントに関連するすべてのオプションは、多くの場合は応答 Interchange の唯一の本文セグメントとして、TA1 セグメントを強制的に生成するのに使用されます。この規則は、受信 Interchange 全体のエラーの有無を表すのに使用されます。ただし、TA1 セグメントでのみレポート可能なエラーが受信 ISA や IEA で検出されると、構成済みの設定で TA1 を強制的に表示しない場合でも、TA1 が生成されます。

デフォルト文字エンコーディング

入力データの文字セットを指定します。InterSystems IRIS は、自動的に、文字をこの文字エンコーディングから変換します。サポートされる値は UTF-8 または Latinn ファミリのメンバです。値の Native は、InterSystems IRIS サーバのネイティブ・エンコーディングを使用することを意味します。

@ (アット・マーク) 文字をこのフィールドの先頭に配置すると、このフィールドは論理文字エンコードではなく、内部 NLS 変換テーブルを指定します。

デフォルトはアダプタによって異なります。

文字セットおよび変換テーブルの詳細は、“変換テーブル”を参照してください。

デフォルトの区切り文字と終端文字

X12 ドキュメントは通常、1 つ以上の機能グループを含み、そのそれぞれに 1 つ以上のトランザクション・セットが含まれる、インターチェンジで構成されるバッチで提供されます。インターチェンジ・ヘッダ (ISA) セグメントには、ドキュメントの解析に使用する区切り文字に関する情報が含まれます。

X12 インターチェンジの解析時、ヘッダ内で見つかった値に区切り文字が設定され、ドキュメントの残りの部分がそれらの値を使用して解析されます。インターチェンジ・ヘッダがない場合、唯一の明らかな区切り文字は要素区切り文字 (セグメント名の直後の文字) であり、セグメント終端文字が該当する場合があります ([CRLF] が続く場合)。この場合、システム・デフォルトの区切り文字値が区切り文字として使用され、これらに基づいてドキュメントが解析されます。システム・デフォルトの区切り文字は以下のとおりです。

- ・ \$C(30) – 繰り返し区切り文字
- ・ : – コンポーネント要素の区切り文字
- ・ ~ – セグメント終端文字

[追加の設定] 領域に表示される以下の設定を使用して、インターチェンジ・ヘッダがない場合にこれらの区切り文字に使用する異なるデフォルト値を渡すことができます。

- ・ デフォルトの繰り返し区切り文字
- ・ デフォルトの要素コンポーネント区切り文字

・ デフォルトのセグメント終端文字

EnsLib.EDI.X12.Document メソッドの ImportFromFile、ImportFromDevice、ImportFromLibraryStream、ImportFromString、および ImportFromIOStream にもこれらの値のパラメータがあります。

ソース・ドキュメントに無効な区切り文字または重複している区切り文字が含まれる場合、**[追加の設定]** 領域で対応する区切り文字が定義されているか、対応するシステムのデフォルト区切り文字であれば、これが使用されます。

ローカル・アプリケーションID

このビジネス・サービスを通じて X12 ドキュメントを受け取る機能とアプリケーションを表す、コロンで区切られた LocalID:Qualifier コード。これらは、応答ドキュメントのヘッダを作成するために使用されます。@ (アット・マーク) 文字は、受信ドキュメントから対応するフィールドを使用することを表します。ID にリテラルの @ 記号を含める必要がある場合は、\@ のようにバック・スラッシュでエスケープします。

デフォルト値は、以下です。

EnsembleX12Service:03

リプライモード

X12 応答ドキュメント (TA1 や 997 など) の発行方法を指定します。オプションは以下のとおりです。

- ・ [] – 応答を送信しません。
- ・ [] – Interchange の受信後直ちに、ビジネス・サービスから応答を送信します。これがデフォルトです。
- ・ [] – ターゲットの構成項目からの応答を待機します。応答を受信したら、送信元に応答を渡します。検証に失敗したり、その他のエラーが発生した場合は、**[BatchReplyType]** に選択されているオプションに従って、即座に応答を生成します。

注釈 受信 ISA セグメントに、X12 標準に照らして長すぎたり短すぎたりするフィールドが含まれている場合は、応答ドキュメントにコピーされるときに、標準に適合するように、そのフィールドに対して埋め込みまたは切り捨てが行われます。

リプライターゲット構成名

(**[ファイル]** および **[FTP]** のみ) ビジネス・サービスが実行した検証に基づいて構成した応答をどこに送信するかを指定します。通常、リストには、1 つの項目が含まれますが、それ以上の項目数の場合があります。リストには、ビジネス・プロセスまたはビジネス・オペレーション、あるいはその両方の組み合わせを含めることができます。

[ターゲット構成名] と比較してください。

SNIP レベル

[Validation Mode] が **[SNIP]** の場合にのみ関係します。

- ・ SNIP レベル 1 – セグメントが有効であること、セグメント順序が有効であること、要素属性が有効であること、数値データ要素に数値があること、メッセージが X12 ルールに準拠していること。
- ・ SNIP レベル 2 – HIPAA 要件 (必要な要素が存在すること、使用されていないとマークされた要素が使用されていないこと、値がコード・テーブルに従っていることなど) を満たしていること

新規ライン許容

真または偽。真 (True) の場合は、読みやすくするために、新しい行がセグメント終端文字の後に (または終端文字の代わりに) ファイルに挿入された場合でも、ビジネス・サービスは受信した X12 ファイルをエラーなしで処理します。偽 (False) の場合は、追加されたこれらの新しい行によってファイル解析中にエラーが発生します。デフォルトは真です。

検証モード

以下の 2 種類の X12 検証が利用可能です。

- ・ SNIP レベル検証 – Workgroup for Electronic Data Exchange (WEDI) の Strategic National Implementation Process (SNIP) で開発された標準に従って X12 メッセージを検証します。
- ・ フラグ・ベースの検証。

SNIP 検証では、以下を検証できます。

- ・ SNIP レベル 1 – セグメントが有効であること、セグメント順序が有効であること、要素属性が有効であること、数値データ要素に数値があること、メッセージが X12 ルールに準拠していること。
- ・ SNIP レベル 2 – HIPAA 要件 (必要な要素が存在すること、使用されていませんとマークされた要素が使用されていないこと、値がコード・テーブルに従っていることなど) を満たしていること

フラグ・ベースの検証では、以下を検証できます。

- ・ 条件付きの必須フィールドなど、必要なフィールドが存在し、すべてのフィールドがスキーマで許可されていること。
- ・ 条件付きで除外されるフィールドが存在しないこと。
- ・ セグメント内のフィールド数、およびそれらがスキーマで許容されているだけ繰り返されているかどうか。
- ・ フィールドおよびコンポーネントのデータ型が正しいこと。
- ・ フィールド値が指定されたコード・テーブルに従っていること。
- ・ フィールドおよびコンポーネントの値が長さ制限に従っていること。

検証

詳細は、以下のセクションの “[検証](#)” を参照してください。

X12 ビジネス・プロセスの設定

X12 ビジネス・プロセスの設定に関する参照情報を提供します。

概要

X12 ビジネス・プロセスには以下の設定があります。

グループ	設定	参照先
基本設定	[検証]	このトピック内の節
	[ビジネス・ルール名]	“プロダクション内での仮想ドキュメントの使用法”の “ルーティング・プロセスに関する設定”
追加設定	[不正メッセージに警告] 、 [不正メッセージ・ハンドラ] 、 [Forward Generated Response To Target] 、 [レスポンス・ターゲット構成名] 、 [レスポンス From] 、 [レスポンス・タイムアウト] 、 [強制同期送信] 、 [変換エラー時のアクション] 、 [妥当性検証エラーで動作]	“プロダクション内での仮想ドキュメントの使用法”の “ルーティング・プロセスに関する設定”
開発とデバッグ	ルールのロギング	“プロダクション内での仮想ドキュメントの使用法”の “ルーティング・プロセスに関する設定”

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。“[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。

[検証]

X12 ドキュメント・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、InterSystems IRIS® によってイベント・ログにその失敗がレポートされ、X12 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)。メッセージが指定された検証に失敗したが、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、その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、X12 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ます、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、X12 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、X12 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。

X12 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
- ・ セグメント内のフィールドおよび複合構造内のコンポーネントがスキーマに適合しているかどうか

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 X12 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。

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。

以下のテーブルは、X12 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
d	ドキュメントの DocType プロパティを調べ、値が指定されているかどうかを確認します。
m	ドキュメントのセグメント構造が適切に構築されており、ドキュメントの DocType プロパティで識別されたスキーマを使用して解析できることを確認します。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。
dm	d と m の両方が適用されます。これは、ビジネス・プロセス (ルータ) のデフォルトです。
s	セグメント構造 (セグメント内のフィールドの数と繰り返し) を強制します。
c	複合構造 (コンポーネントの数) を強制します。
x	セグメントやフィールドに関係条件を適用します。関係条件は、他のフィールドやコンポーネントの存在に基づいて、どのフィールドやコンポーネントが必要か、または除外されるのかを示します。少なくとも、セグメントレベルの検証用には s を追加で指定する必要があります。また、コンポーネントレベルの検証用には c を指定することもできます。 注釈 関係条件は、XSD ファイルではなく、SEF ファイルに含まれています。詳細は、“ X12 スキーマ配布ファイル ” を参照してください。
r	必要な要素の存在を強制します。s も指定されている場合は、必要なフィールドがセグメント内に存在しているかどうかをテストします。c も指定されている場合は、必要なコンポーネントが複合構造に指定されているかどうかをテストします。s も c も指定されていない場合、r には何の効果もありません。
u	[使用されていません] とマークされる要素が存在するかどうかをテストします。s も指定されている場合はフィールドの存在をテストし、c も指定されている場合はコンポーネントの存在をテストします。これらの要素がどちらも存在しない場合、検証は失敗します。唯一使用可能なスキーマが新しいスタイルのスキーマである場合、これは従来のスキーマにのみ格納される情報が必要なため、テストは実行できません。
l	要素の長さをテストします。s も指定されている場合はフィールドの長さをテストし、c も指定されている場合はコンポーネントの長さをテストします。
t	要素のデータ型が正しいかどうかをテストします。s も指定されている場合はフィールドのデータ型をテストし、c も指定されている場合はコンポーネントのデータ型をテストします。

フラグ	ルーティング・プロセス
v	要素の値がコード・テーブルで利用できるかどうかをテストします。s も指定されている場合はフィールドのコード・テーブルに対してテストし、c も指定されている場合はコンポーネントのコード・テーブルに対してテストします。唯一使用可能なスキーマが新しいスタイルのスキーマである場合、これは従来のスキーマにのみ格納される情報が必要なため、テストは実行できません。
n	dmscr1t と同じです。新しいスタイルのスキーマのみを使用して実行できるすべての検証です。
a	dmscrultv と同じで、すべての検証です。
e	エラーが発生しても、ドキュメント全体を通して検証を継続します。e が指定されていない場合、最初のエラーが発生した後、検証は停止します。
(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。これは、ビジネス・サービスとビジネス・オペレーションのデフォルトです。

X12 ビジネス・オペレーションの設定

X12 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

X12 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
追加設定	[テーブルクラス検索]、[デフォルト文字エンコーディング]	“プロダクション内での仮想ドキュメントの使用法”の“ ビジネス・オペレーションに関する設定 ”
	[親セグメント自動バッチ]、[自動バッチ完了タイムアウト]、[切断中は失敗としない]、[セパレータ]、[検証]、[リプライコードアクション]、[再試行タイムアウト]	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。

- ・ [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 送信アダプタの設定](#)”

EnsLib.X12.Adapter.TCPOutboundAdapter には、X12 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。

- [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。
- [準備完了] は偽に設定されています。これは、ソケットから返される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。
- [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。
- ・ “プロダクション内での SOAP サービスおよび Web クライアントの追加”の“[SOAP 送信アダプタに関する設定](#)”

親セグメント自動バッチ

([File] および [FTP] のみ) 真の場合は、バッチの親ドキュメントを含むドキュメントを書き込む際に、バッチ・ヘッダ・セグメント、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・ドキュメント・オブジェクトやファイル名の変更によってトリガされた場合は、バッチ・トレーラ・セグメントを出力します。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。X12 のデフォルトは真です。

自動バッチ完了タイムアウト

([ファイル] および [FTP] のみ) この設定に値を指定することにより、InterSystems IRIS は、完全なバッチ・ドキュメントの作成に必要な末尾のセグメントがまだないバッチ出力に、末尾のセグメントを追加できます。この値には秒単位の時間を指定し、その間ドキュメントが変更されずに維持されると、末尾のセグメントが追加されます。また、この設定により、同じファイルに書き出されるバッチ間に必ず末尾のセグメントが追加されるようになります。これは主に、サービスがトランザクション・セットをその親グループとは異なるビジネス・オペレーションにルーティングしているにもかかわらず、[バッチハンドリング] = [個別] を使用しており、[バッチハンドリング] = [複数のセッションのバッチ] または [バッチハンドリング] = [1

つのセッションのバッチ] の場合に、インターチェンジがトレーラの追加を妨げるようなプロダクションに効果があります。この設定は、[バッチハンドリング] = [バッチ全体] の場合には効果がありません。

0 はタイムアウトが発生しないことを意味します。この設定は、[親セグメント自動バッチ] が [真] の場合にのみ関係します。

デフォルト文字エンコーディング

出力データに適した文字セットを指定します。InterSystems IRIS® は、自動的に、文字をこの文字エンコーディングに変換します。X12 出力の場合は、デフォルトが Latin1 です。“X12 ビジネス・サービスの設定” 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。

再試行タイムアウト

再試行を実行できる秒数。この時間を経過すると、ビジネス・オペレーションは再試行を中断して、エラー・コードを返します。X12 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、すべての X12 ドキュメントがスキップされないようにします。

ファイル名

([File] および [FTP] のみ) ファイル名を出力します。この設定には、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] を空白にした場合のデフォルト値は %f_%Q で、この指定子の意味は次のとおりです。

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

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

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、“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。

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

([TCP] のみ) アダプタ設定の [準備完了] が真 (True) の場合に、この設定を使用して、各種確認応答ドキュメント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、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

ここで、code は、確認応答ドキュメント TA1:4、AK5:1、または AK9:1 の各フィールドに表示されるリテラル値を表しています。予測される code の値を以下に示します。

コード	意味
A	受け入れられました。
E	受け入れられましたが、エラーが記載されました。

コード	意味
R	拒否されました。
M	拒否されました。Message Authentication Code (MAC) が失敗しました。
W	拒否されました。検証テストに失敗しました。
X	拒否されました。コンテンツの複合化に失敗しました。
~	チルダ文字は、TA1、AK5、または AK9 セグメントを含まない応答に相当します。
_	下線文字は、フィールドの値が空の応答に相当します。空または空白コードの値は _ と同じです。
*	アスタリスク文字は、他の文字に相当しない任意の値に相当します (デフォルト = S)。
I?	応答 ControlId が元のドキュメントの ControlId に一致しない場合に相当します。

action に対して以下の値が単独または組み合わせて使用され、文字列を形成します。他に指定されていない場合は、S がデフォルトのアクションです。ただし、A は除きます。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	ドキュメントを正常に完了したとして処理します。コード A のデフォルトのアクションは C です。
W	警告をログに記録しますが、ドキュメントを正常に完了したとして処理します。
R	構成済みの [RetryInterval および FailureTimeout] の設定に従ってドキュメントを再試行します。別のアクションが指定されていない場合は、最終的には失敗します。
S	ドキュメントを中断し、エラーをログに記録し、次のドキュメントに進みます。S は、コード A 以外のすべてのコードのデフォルトのアクションです。コード A のデフォルトのアクションは C です。
D	ビジネス・オペレーションを無効にし、エラーをログに記録し、送信ドキュメントをビジネス・オペレーションのキューの前にリストアします。
F	エラーにより失敗し、次のドキュメントに進みます。

この設定の文字列のデフォルト値は、以下のとおりです。

A=C, *=S, ~=S, I?=W

これは、以下のような意味になります。

- ・ A=C – アクションが受け入れられると、ドキュメントを正常に完了したとして処理します。
- ・ I?=W – 応答 ControlId が元のドキュメントの ControlId と一致しない場合は、警告をログに記録しますが、ドキュメントは正常に完了したとして処理します。
- ・ *=S, ~=S – TA1、AK5、または AK9 セグメントを含まない応答によってドキュメントが中断された場合など、その他すべての場合には、エラーをログに記録し、次のドキュメントに進みます。

セパレータ

後述のように、X12 区切り文字に左から右の順に割り当てられる区切り文字の列。

X12 ドキュメントでは、未加工のコンテンツを編成するために特殊文字を使用します。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。[セパレータ] の値が空白ではない場合は、位置 1 ~ 3 (左から順に) は以下のとおり解釈されます。

1. データ要素区切り文字 (ES)

2. コンポーネント区切り文字 (CS)
3. データ要素の繰り返し区切り文字 (RS)

位置 1 ～ 3 のデフォルト値は以下のとおりです。

1. * (アスタリスク)
2. : (コロン)
3. ^ (キャレット、SOAP ベースの操作用)、\a (レコード区切り文字、その他すべての操作用)

[セパレータ] には、InterSystems IRIS で X12 区切り文字に左から右へ割り当てる 3 文字 (ES、CS、RS) の文字列を指定する必要があります。これらの文字については、前出のリストで説明しています。

位置 4 ～ 6 の文字は、デフォルトのセグメント終端文字、~ (チルダ) を上書きします。以下のリテラルを使用して、位置 4 ～ 6 には 0 ～ 3 文字を指定できます。

- ・ \r : 復帰改行 (ASCII 13)
- ・ \n : 改行 (ASCII 10)
- ・ \a : 配列のレコード区切り文字 (ASCII 30)

位置 4 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、出力ドキュメントで使用する区切り文字を 2 個以下にする場合は、位置 1 ～ 3 に \x を使用できます。位置 1 ～ 3 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 4 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。

[セパレータ] の文字列が空白の場合、デフォルトで、X12 用の現在のデフォルトの区切り文字とセグメント終端文字、および復帰改行 (ASCII 13) と改行 (ASCII 10) が使用されます。以下に例を示します。

*~a~r~n

検証

空白以外の文字列により、送信ドキュメントの基本的な検証がトリガされます。[検証] フィールドが空白のままの場合は、送信ドキュメントの検証は実行されません。