



SDA : インターシステムズの 臨床データ形式

Version 2023.1
2024-01-02

SDA : インターシステムズの臨床データ形式
InterSystems 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

このドキュメントの概要	1
1 SDA について	3
2 SDA ドキュメント	5
2.1 SDA ドキュメントの基本的な XML 構造	5
2.1.1 SDA の Patient	6
2.1.2 SDA の Encounter	7
2.1.3 SDA クラスの詳細	9
2.2 SDA オブジェクトからの SDA ストリームの作成	9
2.3 ストリームレット・マッチング	10
3 SDA のカスタマイズ	13
3.1 拡張クラスを使用した SDA のカスタマイズ	13
3.1.1 Extension クラスへのプロパティの追加	14
3.1.2 クラスのリコンパイル	15
3.1.3 拡張プロパティへのデータの入力	15
3.1.4 ストリームレット・クラスのカスタマイズ	17
3.1.5 カスタム SDA プロパティの使用	20
3.2 カスタム SDA コンテナの作成による SDA のカスタマイズ	20
3.2.1 カスタム SDA データ・クラスの作成	21
3.2.2 カスタム SDA ストリームレット・クラスの作成	24
3.2.3 カスタム SDA コンテナ・クラスの作成	25
3.2.4 構成レジストリへのカスタム SDA コンテナの登録	26
3.3 名前と値のペアを使用した SDA の拡張	26
3.3.1 カスタムの SDA の名前と値のペアの作成	26
3.3.2 カスタム SDA オブジェクトの作成	26

テーブル一覧

テーブル 2-1: HS.SDA3.Patient のプロパティ	7
テーブル 2-2: HS.SDA3.Encounter のプロパティ	8
テーブル 3-1: 変換から除外されるコード・テーブル	15
テーブル 3-2: HS.Types.PatientInfo から継承する、または HS.Types.PatientInfo を使用して継承するクラス	17
テーブル 3-3: 変換から除外されるコード・テーブル	21
テーブル 3-4: 変換から除外されるコード・テーブル	23

このドキュメントの概要

このドキュメントでは、インターシステムズの臨床データ形式である SDA と、以下の製品の場合にこれをカスタマイズする方法について紹介します。

- ・ InterSystems IRIS for Health™
- ・ HealthShare Health Connect™ (Health Connect)
- ・ HealthShare Unified Care Record™ (Unified Care Record)

このドキュメントは、以下の章で構成されます。

- ・ [SDA について](#)
- ・ [SDA ドキュメント](#)
- ・ [SDA のカスタマイズ](#)

詳細は、以下のドキュメントを参照してください。

- ・ “[IHE のユース・ケース](#)” では、IHE の通信の設定方法について説明します。
- ・ “[SDA と CDA の相互運用](#)” では、CDA のドキュメント、変換、およびアノテーションの操作方法について説明します。
- ・ “[InterSystems IRIS for Health および Health Connect のレジストリ・ガイド](#)” では、これらの製品に必要なサービス・レジストリを設定および管理する方法について説明します。
- ・ “[Unified Care Record Registries](#)” では、HealthShare Unified Care Record を管理する方法について説明します。
- ・ “[インターシステムズ製品における FHIR のサポート](#)” では、インターシステムズ製品における FHIR のサポートについて説明します。
- ・ “[インターシステムズのサポート対象プラットフォーム](#)” には、インターシステムズの医療製品がサポートされているプラットフォームがリストされています。

1

SDA について

SDA (Summary Document Architecture) は、患者データを表すために使用されるインターシステムズの形式です。SDA は、医療データを 1 つの形式から別の形式に変換する場合の中間形式です。例えば、患者データを CDA から FHIR[®] に変換する必要がある場合、インターシステムズ製品では、XSLT を使用してデータを CDA から SDA に変換した後、DTL を使用してそのデータを SDA から FHIR に変換します。InterSystems IRIS for Health および HealthShare Health Connect で SDA を使用してデータを変換する方法の詳細は、“インターシステムズの医療製品のデータ変換”を参照してください。

SDA 形式は、必要に応じてカスタマイズすることも、拡張することもできます。

2

SDA ドキュメント

SDA 形式に含まれる情報は、HS.SDA3.Patient、HS.SDA3.Encounter などのクラスのインスタンスで構成されるオブジェクトによって表されます。これらのクラスは XML 対応です。

この章では、以下について説明します。

- ・ [SDA ドキュメントの構造](#)
- ・ [Patient SDA オブジェクト](#)
- ・ [Encounter SDA オブジェクト](#)
- ・ [SDA クラスの詳細の参照先](#)
- ・ [SDA オブジェクトのコレクションから SDA ストリームを作成する方法](#)

SDA のカスタマイズおよび拡張に使用できるオプションは、別の章“[SDA のカスタマイズ](#)”で説明します。

注釈 このドキュメントでは、SDA は HS.SDA3 クラスを指します。HS.SDA クラスはインスタンスにも存在しますが、これらは後方互換性のためだけに提供されています。

2.1 SDA ドキュメントの基本的な XML 構造

SDA ドキュメントの主要セクションは以下のようになっています。

XML

```
<Container>
  <Patient/>
  <Encounters/>
  <AdvanceDirectives/>
  <Alerts/>
  <Allergies/>
  <Appointments/>
  <Problems/>
  <Diagnoses/>
  <Documents/>
  <LabOrders/>
  <RadOrders/>
  <OtherOrders/>
  <Medications/>
  <Vaccinations/>
  <Observations/>
  <PhysicalExams/>
  <Procedures/>
  <FamilyHistories/>
  <IllnessHistories/>
  <SocialHistories/>
```

```

<CustomObjects/>
<Referrals/>
<ClinicalRelationships/>
<ProgramMemberships/>
<MemberEnrollments/>
<MedicalClaims/>
<GenomicsOrders/>
<CarePlans/>
<HealthConcerns/>
<Goals/>
<SocialDeterminants/>
</Container>

```

SDA ドキュメントには以下の規則が適用されます。

1. <Patient> は 1 つだけです。
2. <Patient> は必須で、最初に記述するセクションである必要があります (<Action>、<EventDescription>、<SendingFacility> などの <Container> のプロパティの後に記述します。具体的な詳細は、[クラス・リファレンス](#)を参照してください)。
3. <Patient> 以外のセクションはすべてオプションです。
4. <Patient> 以外のすべてのセクションには、複数のエントリを含めることができます。以下に例を示します。

XML

```

<Procedures>
  <Procedure>
    ...
  </Procedure>
  <Procedure>
    ...
  </Procedure>
</Procedures>

```

5. <Encounters> セクション (存在する場合) は、<Patient> セクションの直後に記述する必要があります。
6. その他のセクションはすべて任意の順番で記述できます。
7. いずれのエントリも、オプションで Encounter 番号を参照できます。この場合、<Encounters> を含め、そこに <Encounter> とその <EncounterNumber> を含める必要があります。
8. 一部のセクションでは、<ActionCode> または <ActionScope> を含めて、永続性に関して何らかのアクションを実行するよう指定することもできます。HealthShare Unified Care Record では、これらの要素によって、SDA の保存時にアクションを実行するようシステムに指示しますが、InterSystems IRIS for Health では、永続化された SDA に対してアクションを実行するようユーザが指示できます。例えば、<ActionCode> の D は、Unified Care Record から、一致するエントリを削除します。SDA の各セクションに関する具体的な詳細は、[クラスリファレンス](#)を参照してください。

2.1.1 SDA の Patient

HS.SDA3.Patient クラスは、患者を表します。このクラスには、以下のような情報を格納するプロパティが含まれます。

- ・ 名前、住所のリスト、性別、配偶者の有無、人種、宗教などの基本情報
- ・ 患者番号

以下の表は、HS.SDA3.Patient のプロパティを示しています。これらのプロパティには、単純なもの (文字列など) もあれば、自身のプロパティを含む複合オブジェクトもあります。また、複合オブジェクトのリストもあります。

テーブル 2-1: HS.SDA3.Patient のプロパティ

単純なプロパティ	複合オブジェクト	複合オブジェクトのリスト
ActionCode*	BirthGender	Addresses
BirthOrder	BirthPlace	Aliases
BirthTime	Citizenship	Organizations
BlankNameReason	ContactInfo	OtherLanguages
Comments	CreatedBy	PatientNumbers
CommunicationPreference	DeathDeclaredBy	PriorPatientNumbers
CreatedOn	EnteredAt	Providers
DeathLocation	EnteredBy	Races
DeathTime	EthnicGroup	SupportContacts
EnteredOn	Extension	(CustomPairs)
ImmunizationRegistryStatus	FamilyDoctor	
ImmunizationRegistryStatusEffectiveDate	Gender	
InactiveMRNs	MaritalStatus	
IsDead	MothersFullName	
IsProtected	Name	
MothersMaidenName	Occupation	
MPIID	PrimaryLanguage	
ProtectedEffectiveDate	PublicityCode	
PublicityEffectiveDate	Race	
UpdatedOn	Religion	

*<Patient> SDA セクションには、<ActionCode> を含めることができます。<ActionCode> の D は、HealthShare Unified Care Record に、MRN が一致する患者の削除を指示します。<ActionCode> の R は、Unified Care Record に、MRN に基づいて既存の患者をこの患者に置換するよう指示します。具体的な詳細は、[クラス・リファレンス](#)を参照してください。

2.1.2 SDA の Encounter

Encounter には、特定の医療インシデントに関連する医療情報すべてが含まれます。HS.SDA3.Encounter クラスは、1 人の患者の受診を表します。オーダー、手順、検査などを表す SDA の他のセクションは Encounter 番号を参照できるため、まとめて関連付けることができます。

以下の表は、HS.SDA3.Encounter のプロパティを示しています。これらのプロパティには、単純なもの（文字列など）もあれば、複合オブジェクト（自身のプロパティを含む）もあります。また、複合オブジェクトのリストもあります。

テーブル 2-2: HS.SDA3.Encounter のプロパティ

単純なプロパティ	複合オブジェクト	複合オブジェクトのリスト
ActionCode*	AdmissionSource	AttendingClinicians
AccountNumber	AdmissionType	ConsultingClinicians
AssignedBed	AdmitReason	Guarantors
AssignedRoom	AdmittingClinician	HealthFunds
AssignedWard	DiagnosisRelatedGroup	RecommendationsProvided
EmergencyAdmitDateTime	DischargeLocation	Specialties
EncounterMRN	EncounterCodedType	(CustomPairs)
EncounterMRNAA	Extension	
EncounterNumber	HealthCareFacility	
EncounterType	EnteredAt	
EndTime	EnteredBy	
EnteredOn	Priority	
ExpectedAdmitTime	PublicityCode	
ExpectedDischargeTime	ReferringClinician	
ExpectedLOAReturnTime	SeparationMode	
ExternalId		
FromTime		
PreAdmissionNumber		
PriorBed		
PriorRoom		
PriorWard		
PriorVisitNumber		
ToTime		
TransferredFromED		
UpdatedOn		
VisitDescription		

*Encounter には、HealthShare Unified Care Record にアクションの実行を指示する〈ActionCode〉を含めることもできます。可能な値には以下が含まれます。

- ・ D = 削除
- ・ E = 空の場合は削除
- ・ C = すべてクリア
- ・ R = 置換

Encounter の〈ActionCode〉の具体的な詳細は、[クラス・リファレンス](#)を参照してください。

2.1.3 SDA クラスの詳細

SDA 形式の詳細は、HS.SDA3.Container および HS.SDA3 パッケージ内のその他のクラスのクラス・リファレンスを参照してください。

SDA クラスのクラス・リファレンスにアクセスするには、以下の操作を行います。

1. HealthShare キューブの“クラスリファレンス”リンクをクリックするか、
host:port/csp/documatic/%25CSP.Documatic.cls に移動します。host はインストール環境が動作中のマシンの名前で、port はウェブサーバ・ポートです。例：
`http://localhost:57772/csp/documatic/%25CSP.Documatic.cls`
2. 必要に応じてログインします。
3. 左側のペインの上部にあるドロップダウン・リストを使用して、**HSLIB** ネームスペースに変更します。
4. 見やすくするには、[パーセントクラス] チェック・ボックスのチェックを外します。
5. パッケージ名の階層を HS までドリル・ダウンし、続いて SDA3、さらに目的のクラスまでドリル・ダウンします。

任意の SDA クラスの XML ドキュメント構造を表示することもできます。

1. 管理ポータルで Edge Gateway のネームスペースを選択して、[ホーム]→[相互運用性]→[相互運用]→[XML]→[XML スキーマ構造] に移動します。
2. [HS.SDA3] スキーマで、構造を表示するクラスの名前を選択します。

Tip ヒン HS.SDA3 XML スキーマをナビゲートするには、**Container** を選択し、そこからドリル・ダウンするのが最も簡単です。

“[CDA および SDA アノテーション](#)” 機能を使用して、SDA クラスの構造を表示することもできます。

2.2 SDA オブジェクトからの SDA ストリームの作成

注釈 ここでは、あらゆるサイズの SDA に対応した手法について説明します。患者レコード全体を 1 つの SDA オブジェクト (HealthShare Unified Care Record のコンテキストでは数千もの Encounter が含まれる可能性があります) としてインスタンス化するのではなく、ここで説明する手法では、個々のセクションのみをインスタンス化して、SDA ストリームを手動で構築します。SDA が小さいことが確実である場合は、SDA 全体を 1 つのオブジェクトとしてインスタンス化して、ストリームを直接記述することができます。

複数の SDA オブジェクトがあり、それらをまとめて 1 つの SDA ストリームを生成する場合は、以下の手法を使用します。

1. 新規ストリームをインスタンス化します。
2. <Container> 開始タグをストリームに記述します。
3. Patient オブジェクトは最初に記述する必要があり、SDA オブジェクトの ToQuickXML() メソッドを使用して、XML としてストリームに記述します。
4. 以降のオブジェクトはタイプ別 (Encounter、LabOrder など) にグループにする必要があり、SDA オブジェクトの StartXMLList() メソッドを使用して、そのコレクションの開始タグを記述します。例：<Encounters>。<Encounters> を含める場合、<Patient> の直後に配置する必要があることに注意してください。
5. ToQuickXML() メソッドを使用して、そのタイプの各オブジェクトをストリームに記述します。

6. 特定のタイプのコレクションの記述が完了したら、EndXMLList() メソッドを使用して、コレクションの終了タグを記述します。
7. その他のコレクションに対して上記の手順を繰り返します。
8. 完了したら、</Container> 終了タグを記述します。

2.3 ストリームレット・マッチング

各ストリームレット・クラスには、そのクラスの MATCHINGS パラメータで定義された、それぞれ異なる一致条件があります。このパラメータは `HS.SDA3.Streamlet.Abstract` から継承されます。ストリームレット・マッチング・ロジックにより、HealthShare はストリームレットが新規なのか、既存データの更新なのかを判断できます。2 つのストリームレットを比較してマッチングと重複除去を行うかどうかは、EncounterNumber プロパティによって決まります。

- ・ 2 つのストリームレットの EncounterNumber が異なる場合や、一方のストリームレットには EncounterNumber の値があるのに他方にはない場合、ストリームレットは比較の対象になりません。
- ・ 両方のストリームレットの EncounterNumber が同じ場合、ストリームレットは比較の対象になります。
- ・ いずれのストリームレットにも EncounterNumber の値がない場合、ストリームレットは比較の対象になります。

適切なストリームレットがある場合、マッチング・プロセスは 3 つのフェーズで実行されます。

1. まず、Edge Gateway でストリームレットが取り込まれると、そのデータが同じ MRN を持つ同じタイプのストリームレットと比較されます。複数の Edge Gateway が存在するシステムでは、比較は Access Gateway の集約キャッシュによって処理されます。このプロセスの詳細は、`HS.SDA3.Streamlet.Abstract.Aggregate()` メソッドのクラス・リファレンスを参照してください。
2. ストリームレットに外部 ID が含まれる場合は、この外部 ID を使用して、その ID を持つ同じタイプの他のストリームレットと照合します。
3. 上記のどのチェックでも一致が見つからない場合、HealthShare は各ストリームレット固有の MATCHINGS のキーと値のペアを、同じストリームレット・タイプの他のレコードと照合します。

どのチェックでも一致が見つからない場合、ストリームレットは新規と認識されます。いずれかのチェックで一致が見つかった場合、ストリームレットは既存のストリームレットの更新と認識され、既存のストリームレットと共に集約されます。これを重複除去と呼びます。後述する“[SDA のカスタマイズ](#)”に示すように、MATCHINGS パラメータは変更できます。

MATCHINGS パラメータは、`MatchList1 || MatchList2` という形式を取ります。各 MatchList 項目は `MatchType/MatchProp1,MatchProp2,.../NullProp1,NullProp2,...` という形式を取ります。

これらの要素は、以下のように定義されます。

- ・ MatchType は、マッチングするタイプの任意の名前です。
- ・ MatchProp は、マッチングに使用する SDA ストリームレット・クラスのプロパティの名前です。複雑なケースでは、一時プロパティまたは計算プロパティにすることができます。MatchProp 要素が 2 つ以上ある場合は、そのすべてに対してマッチングを実行する必要があります。
- ・ NullProp はマッチングに使用するプロパティの名前です (オプション)。空の NullProp は、すべてにマッチングすると見なされます。NullProp 要素が 2 つ以上ある場合は、そのすべての組み合わせに対してマッチングが実行されます。
- ・ MatchProp または NullProp の後のアスタリスクは、その要素が、Code および SDA Coding Standard に対してマッチングされる CodeTableDetail であることを示します。

架空のカスタム Allergy ストリームレットの以下のコードを考えてみましょう。

```
Parameter MATCHINGS = "ALG/Allergy*/FromTime,AllergyCategory* ||  
ZATC/ATCCode*/FromTime";
```

この場合、Unified Care Record は、以下の条件に従って順番にマッチングを試みます。

1. Allergy に対してマッチングします。これは CodeTableDetail です。オプションで、FromTime または AllergyCategory に対してマッチングします。いずれかが存在する場合、これも CodeTableDetail です。
2. ATCCode に対してマッチングします。これは CodeTableDetail です。オプションで、FromTime が存在する場合は、これに対してマッチングします。

3

SDA のカスタマイズ

ほとんどの場合、システムを経由するデータはすべて SDA 構造で十分処理できます。しかし、さらにデータを取得する必要がある場合は、簡単に SDA を拡張できます。

SDA を拡張する方法としてお勧めの方法は 2 つあり、ニーズに合わせて使用します。

- ・ [提供されている拡張クラスを使用して既存の SDA オブジェクトを拡張する](#)。この方法はすべての製品で使用できます。各 SDA クラスには対応する拡張クラスがあり、既定ではプロパティは指定されていません。その拡張クラスでプロパティを宣言することによって、SDA の機能を拡張できます。
- ・ [カスタム SDA コンテナおよび対応するカスタム・ストリームレットを作成する](#)。この方法は、HealthShare Unified Care Record でのみ使用できます。SDA に新たにセクションを追加するカスタム SDA コンテナを定義します。この定義には、カスタム SDA ストリームレットが含まれます。この方法は、Unified Care Record を実行していて、新たなデータ型を処理するために新規のストリームレット・タイプを作成する場合に選択してください。

すべての製品で使用できる方法がもう 1 つあります。[カスタムの名前と値のペアを既存の SDA セクションまたは新規のカスタム SDA セクションに追加する](#)方法です。取得するデータが単純なペア（質問と回答のセットが含まれるアンケートなど）として構造化されている場合にこの方法を使用できます。ただし、この方法には以下のようなさまざまな制限事項があります。

- ・ 0:* のカーディナリティを持つ複合オブジェクトまたはデータ・コレクションを保存する場合は複雑になります。
- ・ ストリームレットのシリアル・プロパティにカスタム・データを格納することはできません。
- ・ このデータを Health Insight のデータベースに伝播するのは簡単ではありません。
- ・ カスタム・データに対するデータ型の妥当性検証を簡単には実装できません。

これらの制限事項があるため、名前と値のペアの使用はお勧めできません。この方法がご自身の状況に適していると思われる場合は、このガイドの“[名前と値のペアを使用した SDA の拡張](#)”を確認し、必要に応じてインターシステムズのサポート窓口にお問い合わせください。

3.1 拡張クラスを使用した SDA のカスタマイズ

各 SDA データ・クラス（以下に説明している例外はあります）には、対応する拡張クラスとそれを参照するプロパティがあります。例えば、クラス HS.SDA3.Allergy には、タイプが HS.Local.SDA3.AllergyExtension のプロパティ **Extension** があります。既定では、これらの拡張クラスにプロパティは指定されていません。これらのクラスにプロパティを追加することで SDA をカスタマイズできます。

すべての拡張クラスは、インストール時またはアップグレード時に自動的に作成された HSCUSTOM データベースに配置されています。

拡張クラスを使用するには、以下の操作を行います。

1. 必要に応じて拡張クラスにプロパティを追加します。
2. クラスをリコンパイルします。
3. 拡張プロパティにデータを入力します。

HealthShare Unified Care Record を実行している場合は、SDA 拡張クラスに関連付けられたストリームレット・クラスをカスタマイズすることもできます。

拡張クラスにプロパティを追加した後は、それらのプロパティをコードで使用したり、Unified Care Record では Clinical Viewer で表示したりできます。

重要 Health Insight を分析に使用している場合は、SDA 拡張を Health Insight に伝播して、Health Insight で **HSAA.Local.<SDAType>Extension** クラスを変更する必要もあります。“Health Insight Installation and Configuration Guide” の “[SDA Extensions and Health Insight](#)” を参照してください。

拡張クラスに関する一般的な注意事項

すべてのコード・テーブル拡張クラスは HS.SDA3.CodeTableExtension を拡張します。その他の拡張クラスはすべて HS.SDA3.DataType を拡張します。

以下のクラスには、拡張クラスが関連付けられていません。

- ・ HS.SDA3.SuperClass および HS.SDA3.CodeTableDetail。これらはスーパークラスとしてのみ使用されるからです。
- ・ HS.SDA3.Container。この章で後述するように、このクラスの拡張には別のメカニズムが使用されます。
- ・ HS.SDA3.CustomObject。
- ・ HS.SDA3.ClinicalRelationship および HS.SDA3.ProgramMembership。これらは、レジストリ・データを格納するクラスで、ストリームレット・アーキテクチャには含まれていないからです。

Patient クラスは、すべての患者データの主な親クラスであるため、特に重要です。Patient SDA オブジェクトには、対応するレジストリ・クラス HS.Registry.Patient および関連付けられているメッセージ・クラスがあります。これらのクラスそれぞれに、HS.Local.SDA3.PatientExtension クラスへの参照である、**Extension** と呼ばれるプロパティが含まれています。このため、作成する拡張プロパティはすべて、これらの関連付けられたクラスおよび Patient クラス自体で使用可能になります。

3.1.1 Extension クラスへのプロパティの追加

SDA をカスタマイズするには、HSCUSTOM ネームスペースの **HS.Local** パッケージ内の拡張クラスにプロパティを追加します。

IDE で、該当するネームスペースに切り替えて、目的の拡張クラスを開きます。必要に応じて、新規プロパティを追加します。

以下のタイプのプロパティを追加できます。

- ・ **%String**
- ・ 任意の **HS.SDA3.<DataType>** (Boolean や Numeric など)。
- ・ 任意の既存の **HS.SDA3** シリアル・クラス。
- ・ 作成した任意のカスタム・シリアル・クラス (**HS.SDA3.DataType** を拡張する必要があります)。HS.Local や HS.SDA3 など、すぐに使用可能な HealthShare パッケージにカスタム・シリアル・クラスを定義しないでください。これにより、クラス名が SDA3 の将来の標準クラス名と競合しないようにします。例えば、カスタム・シリアル・クラスには、**HS.SDA3.MySDASection** ではなく **ZUser.HS.MySDASection** のような名前を付ける必要があります。HSCUSTOM にカスタム・パッケージ・マッピングを追加できます。

- 新規作成した任意のコード・テーブル (HS.SDA3.CodeTableTranslated を拡張する必要があります)。以下のコード・テーブルは変換できません。

テーブル 3-1: 変換から除外されるコード・テーブル

HealthCareFacility	CareProviderType	Country
Organization	City	County
User	State	Trust
CareProvider	Zip	

以下の例では、**Sneeziness** プロパティおよび **FlowerType** プロパティが HS.Local.SDA3.AllergyExtension クラスに追加されています。**Sneeziness** は文字列で、**FlowerType** はカスタム・タイプです。

Class Definition

```
Class HS.Local.SDA3.AllergyExtension Extends HS.SDA3.DataType
{
  Parameter STREAMLETCLASS = "HS.SDA3.Streamlet.Allergy";
  Property Sneeziness As %String;
  Property FlowerType As My.Local.Type.FT;
}
```

3.1.2 クラスのリコンパイル

変更を加えた後、関連するクラスをリコンパイルする必要があります。効率性を最大限にするために、HS.SDA3 パッケージ全体をコンパイルすることをお勧めします。

追加のインスタンスがある場合は、新規のコード (関連付けられたマッピングを含む) が必ず各インスタンスに配置されるようにします。

注釈 ミラーに配置する場合は、以下の操作を行います。

1. HSCUSTOM で行ったコード変更を最初にバックアップ・メンバに適用します。
2. フェイルオーバーを実行します。
3. 変更を新しいバックアップ・メンバに適用します。

3.1.3 拡張プロパティへのデータの入力

これで拡張プロパティの準備が整ったので、次はソース・システムからの受信データがそれらの拡張プロパティに入力されるようにする必要があります。

これは、ニーズとシステム設定に応じて複数の方法で実行できます。

- HealthShare Unified Care Record を実行している場合は、パイプラインを作成できます。詳細は、“Configuring Unified Care Record for Data Feeds” の “[Creating Pipelines and Inbound Processes](#)” の章を参照してください。
- InterSystems IRIS for Health または Health Connect を実行している場合は、DTL 変換を作成できます。詳細は、“DTL 変換の開発”を参照してください。DTL を作成する場合に目的の拡張プロパティが必ず使用可能になるようにするには、HSCUSTOM から XML スキーマをエクスポートして、それを目的のネームスペースにインポートする必要があります。
 1. ターミナルから、HSCUSTOM ネームスペースで以下を実行します。

```
do ##class(HS.SDA3.Container).ExportXMLSchema()
```

エクスポート・ファイルに使用する目的のファイル名を入力するよう求められます。必要に応じて、フル・パスとファイル名を .xsd 拡張子を付けて指定します。

2. 管理ポータルから、DTL を作成しようとしているネームスペースで、[Interoperability]→[相互運用]→[XML]→[XML スキーマ構造] の順に選択します。
 3. [インポート] ボタンを選択して、エクスポート・ファイルに移動します。エクスポートされたスキーマが目的のネームスペースにインポートされます。
 4. これで、新しい DTL 変換を開いて拡張プロパティを使用できるようになりました。拡張クラスのプロパティが 1 つの場合、そのカスタム・プロパティは DTL ダイアグラムの **拡張** プロパティの下には表示されません。例えば、拡張クラスに BloodType と Sneeziness という 2 つのプロパティがある場合は、DTL ダイアグラムの **拡張** プロパティを展開して 2 つのカスタム・プロパティを表示できます。ただし、拡張クラスに BloodType という 1 つのプロパティしかない場合、DTL ダイアグラムの **拡張** プロパティの下に **BloodType** は表示されません。BloodType プロパティは親拡張プロパティに “ロールアップ” されます。
- ・ 任意の製品を実行していて、HL7 メッセージを SDA に変換するために HS.Gateway.HL7.HL7ToSDA3 クラスを使用している場合は、以下の操作を行います。
1. HS.Gateway.HL7.HL7ToSDA3 を拡張する新規クラスを作成します。
 2. 新規クラスで、追加した SDA 拡張を処理するためのコールバック・メソッドを実装します。このコールバック・メソッドには、On<StreamletName>() という形式 (例えば、OnAllergy()) で名前が付けられます。
 3. HS.Gateway.HL7.InboundProcess オペレーションを編集して、その **HL7ToSDA3Class** プロパティを、新規クラスを参照するように変更します。

以下に、SDA 拡張にデータを入力する具体的なケースと考慮事項を示します。

HL7™ FHIR® DSTU2 とカスタマイズされた SDA との間の変換

新しい拡張を FHIR DSTU2 で使用するために、拡張を作成した SDA クラスごとに、対応する DTL のカスタム・コピーを作成する必要があります。FHIR/SDA DSTU2 DTL をカスタマイズする方法の詳細は、[“インターシステムズ製品における FHIR のサポート”](#) を参照してください。

認識できない SDA 拡張

異なる拡張のセットによって環境間で共有される未加工の SDA に対応するには、すべての SDA3 クラスで XML 処理指示文 XMLIGNOREINVALIDTAG = 1 を使用します。認識できない **Extension** プロパティを持つ SDA ストリームをシステムがオブジェクトに読み込もうとした場合、認識できないプロパティは無視されます。**TraceOperations** が有効になっている場合、これが発生すると警告トレースがログに記録されます。

Patient 拡張

HS.Local.SDA3.PatientExtension に格納されたデータを使用可能にし、HealthShare 導入環境全体で一貫して使用されるようにするために、HS.Types.PatientInfo クラスでは、タイプが HS.Local.SDA3.PatientExtension の **Extension** という名前のプロパティがサポートされます。これにより、HS.Types.PatientInfo のすべてのサブクラスでも拡張が継承されます。これらのクラスを以下の表に示します。また、HS.Registry.Patient には、タイプが HS.Local.SDA3.PatientExtension の **Extension** プロパティが関連付けられており、HS.Hub.MPI.Manager の AddUpdateHub() メソッドのコードによって Patient の拡張クラスが処理されて HS_Registry.Patient テーブルに格納されます。

重要 HS.Registry.Patient などの永続クラスは、HS.Local.SDA3.PatientExtension を変更した後に再コンパイルする必要があります。

テーブル 3-2: HS.Types.PatientInfo から継承する、または HS.Types.PatientInfo を使用して継承するクラス

HS.Audit.Criteria	HS.Message.PatientSearchRequest
HS.Gateway.Access.QueryProcess	HS.Message.QueueForFetchRequest
HS.Hub.HSWS.WebServices.Containers.Patient	HS.Message.RemovePatientRequest
HS.Hub.MPI.FetchStreamlet	HS.MPI.Initiate.Operations
HS.Hub.MPI.Manager	HS.MPI.Native.PatientRecord
HS.Message.AddPatientRequest	HS.MPI.SecondaryMPI
HS.Message.AddUpdateHubRequest	HS.MPI.SureScripts.Operations
HS.Message.FindAutoLinkMatchRequest	HS.Types.PatientInfo
HS.Message.GetCompositeRecordResponse	HS.Types.PatientSerial
HS.Message.MedicationHistoryRequest	HS.UI.ClinicianPortal
HS.Message.PatientBatchFetchRequestAsync	HS.UI.PatientSearch
HS.Message.PatientMPIMatch	HS.UI.PatientSearchUtil
HS.Message.PatientSearchMatch	

3.1.4 ストリームレット・クラスのカスタマイズ

重要

このセクションは、HealthShare Unified Care Record にのみ適用されます。

ストリームレット・クラスのカスタマイズは、高度なタスクです。ここに示された制限事項を注意深く確認し、必要に応じてインターシステムズのカスタマ・サポート窓口にお問い合わせください。

前のセクションで説明したように、SDA オブジェクトにプロパティを追加できることに加えて、関連付けられたストリームレット・クラスの動作の一部をカスタマイズすることもできます。このセクションでは、そのプロセスと、留意する必要がある重要な制限事項について説明します。

以下の場合に、ストリームレット・クラスのカスタマイズを選択できます。

- ・ 関連付けられている SDA クラスの照合ロジックを変更する場合
- ・ 関連付けられている SDA クラスの検証ロジックを変更する場合
- ・ 変換済みコードとして扱われるフィールドと、そのように扱われないフィールドを変更する場合
- ・ コールバック・メソッドの動作をオーバーライドする場合

3.1.4.1 手順

一般的な手順は以下のとおりです。

1. HSCUSTOM ネームスペースで、目的の SDA オブジェクトに対応するストリームレット・クラスを拡張します。クラス・パッケージ名または独自に作成したサブパッケージ名を使用する必要があります。カスタム・クラスを HS.Local に保

存する場合は、先頭が Z になっているサブパッケージ名 (HS.Local.ZMyPackage など) を使用する必要があります。HSCUSTOM にカスタム・パッケージ・マッピングを追加できます。

2. SDA 拡張クラスの STREAMLETCLASS パラメータを適切に調整します。例えば、HS.Local.SDA3.AllergyExtension には以下が指定されます。

```
Parameter STREAMLETCLASS = "HS.SDA3.Streamlet.Allergy";
```

このストリームレット・クラスを拡張してクラス MyPackage.SDA3.Streamlet.ZAllergy を新規作成する場合、このパラメータを以下のように変更できます。

```
Parameter STREAMLETCLASS = "MyPackage.SDA3.Streamlet.ZAllergy";
```

3. 必要に応じて、コールバック・メソッドをカスタマイズするか、メソッドを新たに追加するか、または一時プロパティを新たに追加します。操作を開始する前に、以下に示している制限事項を確認してください。

以下の例では、新規の **Sneeziness** プロパティに対するカスタムの照合および検証が含まれる新しい Allergy ストリームレットを定義しています。

Class Definition

```
Class ZHS.SDA3.Streamlet.ZAllergy Extends HS.SDA3.Streamlet.Allergy
{
  /// Adding a fallback match
  /// Previously, it matched based on the Allergy code table,
  /// with nullable matches on FromTime and AllergyCategory code table
  /// If that doesn't match, we want it to also try using the ATCCCode code table
  /// (with nullable match on FromTime)
  Parameter MATCHINGS = "ALG/Allergy*/FromTime,AllergyCategory*|| ZATC/ATCCCode*/FromTime";
  /// Adding some validation for our extension property
  Method OnValidate() As %Status
  {
    // We use IsDefined to avoid instantiating (swizzling) a null serial property, such as Extension
    // If we know that we'll always have data under Extension, this isn't needed
    If ..SDA.IsDefined("Extension") {
      Set tSneez=..SDA.Extension.Sneeziness
      If tSneez?'1.N1" Tissue".E, tSneez?'1.N1" Hankie".E {
        Quit $$$ERROR($$$GeneralError, "Sneeziness must be in terms of Tissues or Hankies.")
      }
    }
  }
  Quit ##super()
}
```

3.1.4.2 制限事項

ストリームレット・クラスをカスタマイズする際には、以下の制限事項に留意してください。

- ・ 名前の先頭が “HS” のパッケージに、カスタマイズされたストリームレット・クラスを作成することはできません。ただし、名前の先頭が Z になっている HS.Local のサブパッケージの場合 (HS.Local.ZMyPackage など) はこの限りではありません。
- ・ ストリームレットのカスタマイズは、メモリ内でのみ適用され、そのストリームレットの格納には影響しません。ストリームレットを格納する場合、格納時にそのタイプは常に **HS.SDA3.Streamlet.<type>** になります。
- ・ カスタム・ストリームレット・クラスで拡張できるメソッドは、以下に示したイベント・コールバック・メソッドだけです。その他のイベント・ハンドラはすべてスーパークラス内で **FINAL** とマークされるため、変更することはできません。
 - OnInactivate()
 - OnMatchActionScope()
 - OnBeforeMatch()
 - OnValidate()
 - OnBeforeSave()
 - OnAfterSave()

- ・ OnDeleteSQL コールバック・メソッドおよび OnDeleteHandler コールバック・メソッドをカスタマイズすることはできません。コールバック以外のメソッドもカスタマイズすることはできません。
- ・ その他のコールバック・メソッド (OnXXX) をカスタマイズする際には、必ず `##super` を呼び出す必要があります。
- ・ 将来、競合が発生しないように、新規に作成するメソッドやプロパティの先頭を文字 Z にすることをお勧めします。
- ・ カスタマイズできるのは、DATEPROPERTY、MATCHINGS、TRANSLATIONS、および ACTIONSCOPES の各パラメータだけです。これら以外のパラメータをカスタマイズすることはできません。
- ・ MATCHINGS パラメータを変更する場合、新規作成する照合タイプはすべて、先頭を文字 Z にする必要があります。既存の照合タイプを変更することはできません。既存の照合タイプを削除することはできます。
- ・ `Extension.<property>` を指定することによって、DATEPROPERTY パラメータおよび MATCHINGS パラメータから拡張クラスのプロパティにアクセスできるようになります。
- ・ MATCHINGS パラメータを変更する場合、以下の“照合の再計算”で説明している追加手順を実行する必要があります。
- ・ 以下のストリームレット・メタデータは、追加することも変更することもできません。
 - 永続プロパティ
 - リレーションシップ
 - インデックス
- ・ Patient ストリームレットには、OnAggregateExtensionImpl() と呼ばれるコールバック・メソッドが含まれています。これは、拡張クラス内の患者データを集約する方法を制御します。既定では、このメソッドは、集約対象の患者レコードの中で最適と判断された患者レコードから拡張データを取得します。必要に応じて、カスタマイズした集約動作を提供するようこのメソッドをオーバーライドできます。
- ・ ストリームレット・クラスで使用可能な **Stash** という名前の一時多次元プロパティがあります。これを使用して、`OnBeforeSave()` メソッドから `OnAfterSave()` メソッドにデータを渡すことができます。

ストリームレット・クラスをカスタマイズした後は、前述したように HS.SDA3 パッケージを再コンパイルする必要があります。

3.1.4.3 照合の再計算

カスタム・ストリームレットに合わせて MATCHINGS パラメータを変更した場合、データを再評価して新規の照合を確認し、必要な場合はそれらの照合を調整する必要があります。以下で説明するように、これを実行できるように 3 つのユーティリティが用意されています。

これらの各ユーティリティは、ストリームレット・タイプを引数として取ります。標準ストリームレットまたは標準ストリームレットの拡張の場合、これは `Allergy` などの標準タイプです。カスタム・ストリームレットの場合は、`HSCustom.SDA3.Streamlet.ZAllergy` のような完全クラス名を使用する必要があります。

1. まず、`RecalculateMatches` を使用して既存の照合を再計算します。例えば、以下のようになります。

ObjectScript

```
Do ##class(HS.SDA3.Streamlet.Utility.RecalculateMatches).Start("HSCustom.SDA3.Streamlet.ZAllergy")
```

2. 次に、`FindMatches` を使用して照合の重複をすべて特定します。例えば、以下のようになります。

ObjectScript

```
Do ##class(HS.SDA3.Streamlet.Utility.FindMatches).Start("HSCustom.SDA3.Streamlet.ZAllergy")
```

`^HS.SDA3.Streamlet.MatchGroups` グローバルのコンテンツを表示して、このユーティリティが重複を検出したかどうかを確認できます。

- FindMatches で照合の重複を特定した場合は、ReconcileMatches ユーティリティを実行して、このような照合をすべて調整できます。

ObjectScript

```
Do
##class(HS.SDA3.Streamlet.Utility.ReconcileMatches).ReconcileMatches("HSCustom.SDA3.Streamlet.ZAllergy")
```

ReconcileMatches() メソッドは、最新のストリームレットを受け取り、それに一致するその他すべてを削除済みとマークすることで、これを実行します。一致のグループから受け取るストリームレットの判別に別の条件を適用する場合は、ReconcileMatches() メソッドを拡張するか、これを目的とした独自のカスタム・メソッドを作成できます。

FindMatches および RecalculateMatches はオプションで 2 つ目のパラメータを取ることができます。このパラメータはブーリアン値であり、処理をメソッドの前回の実行から再開するか、最初からやり直すかを指定します。例えば、以下のような場合です。

ObjectScript

```
Do ##class(HS.SDA3.Streamlet.Utility.FindMatches).Start("HSCustom.SDA3.Streamlet.ZAllergy", 1)
```

このパラメータが 0 に設定されているか省略されている場合、メソッドは指定されたタイプのすべてのストリームレット ID を処理します。このパラメータが 1 に設定されている場合、メソッドは、前回正常に処理されたストリームレット ID から再開します。この ID はグローバル ^ISC.HS.Streamlet.Loader("Last") に格納されています。

3.1.5 カスタム SDA プロパティの使用

ObjectScript コードでカスタム・プロパティを使用できるようになりました。カスタム拡張クラスおよびそのプロパティは、他のクラスと同じように動作します。

例えば、**Sneeziness** プロパティを新たに Allergy 拡張クラスに追加した場合、これを以下のように使用できます。

ObjectScript

```
Set tAllergy=##class(HS.SDA3.Allergy).%New()
Set tAllergy.Extension.Sneeziness="3 Tissues"
```

Clinical Viewer で新規のプロパティを表示することもできます。詳細な手順は、“Customizing the Clinical User Interfaces” ガイドの “[Adding SDA Fields to the Clinical Viewer](#)” を参照してください。

カスタム SDA 拡張は、HealthShare Patient Index と Health Insight からアクセスして使用できます。詳細および追加の構成については、製品のドキュメントを参照してください。

3.2 カスタム SDA コンテナの作成による SDA のカスタマイズ

重要 この機能は、HealthShare Unified Care Record でのみ使用できます。

カスタム SDA コンテナを使用すると、複合プロパティおよび照合キーを格納するカスタム SDA セクションを作成できます。カスタム SDA コンテナを作成するには、以下の手順を正しい順序で実行する必要があります。

- カスタム SDA セクションごとに、カスタム SDA データ・クラスを定義します。
- カスタム SDA データ・クラスごとに、データを格納するための SDA ストリームレット・クラスを定義します。
- カスタム・セクションそれぞれをプロパティとして格納するカスタム SDA コンテナ・クラスを定義します。
- 構成レジストリにカスタム SDA コンテナを登録します。

これらの手順を完了したら、SDA 処理でカスタム SDA コンテナが受け取られ、格納され、集約されるようになります。カスタム SDA データは、集約キャッシュで使用可能になり、Clinical Viewer でカスタム表示に使用したり、カスタム・レポートに含めたりすることができます。

3.2.1 カスタム SDA データ・クラスの作成

カスタム SDA コンテナを作成する最初の手順では、カスタム SDA セクションごとにカスタム SDA データ・クラスを定義します。

1. HS.SDA3.SuperClass を拡張するクラスを新規作成します。

いずれかのカスタム SDA セクションにカスタム SDA データ・クラスを定義して、Health Insight でデータを解釈して保存できるようにする場合は、先に進む前に次のセクションを確認してください。

2. クラスに **User.ZMySection** のような名前を付けます。将来の標準 SDA3 クラス名と競合しないように、短いクラス名に接頭語 “Z” を付けることをお勧めします。

クラスは SDA プロパティの標準セットを継承します。

ActionCode	EncounterNumber	EnteredOn	ToTime
ActionScope	EnteredAt	ExternalId	UpdatedOn
CustomPairs	EnteredBy	FromTime	

3. 独自の SDA プロパティを追加できます。文字列以外のデータ型を使用する場合は、Blob、Boolean、Numeric、TimeStamp などの既存の SDA3 データ型クラスを使用します。これにより、二重引用符のペアが削除をトリガする削除メカニズムに対応できるようになります。

二重引用符の削除は、コレクション内の要素には適用されません。そのような要素を削除するには、問題のストリームレットに `ActionCode = 'R'` を割り当てて、ストリームレットの置換をトリガし、既存のすべてのコレクションをクリアする必要があります。

以下のタイプのプロパティを追加できます。

- ・ **%String**
- ・ 任意の **HS.SDA3.<DataType>** (Boolean や Numeric など)。
- ・ 任意の既存の **HS.SDA3** シリアル・クラス。
- ・ 作成した任意のカスタム・シリアル・クラス (**HS.SDA3.DataType** を拡張する必要があります)。HS.Local や HS.SDA3 など、すぐに使用可能な HealthShare パッケージにカスタム・シリアル・クラスを定義しないでください。これにより、クラス名が SDA3 の将来の標準クラス名と競合しないようにします。例えば、カスタム・シリアル・クラスには、**HS.SDA3.MySDASection** ではなく **ZUser.HS.MySDASection** のような名前を付ける必要があります。
- ・ 新規作成した任意のコード・テーブル (**HS.SDA3.CodeTableTranslated** を拡張する必要があります)。以下のコード・テーブルは変換できません。

テーブル 3-3: 変換から除外されるコード・テーブル

HealthCareFacility	CareProviderType	Country
Organization	City	County
User	State	Trust
CareProvider	Zip	

4. SDA クラスの標準の命名規則は、“Medication” のような単数形です。この場合、コンテナには、以下の XML に示すように “Medications” のような標準の複数形のリストが格納されます。

XML

```
<Medications>
  <Medication>
  </Medication>
  <Medication>
  </Medication>
</Medications>
```

カスタム・セクションの末尾に “s” を追加しても (Diagnosis/Diagnoses など) 機能しない場合は、SDA データ・クラスに StartXMLList() メソッドおよび EndXMLList() メソッドを実装し、コンテナで開始および終了のコレクション・タグを出力します。

Class Member

```
ClassMethod StartXMLList()
{
  Quit "<Diagnoses>"
}
```

Class Member

```
ClassMethod EndXMLList()
{
  Quit "</Diagnoses>"
}
```

5. クラスをコンパイルします。

注釈 <PROTECT> エラーが発生する場合、HSLIB データベースを一時的に書き込み可能にし、クラスを再コンパイルしてから、HSLIB を読み取り専用に戻します。

以下の例は、移植手術に関する情報のカスタム・データ・クラスを示しています。

Class Definition

```
Class User.ZTransplant Extends HS.SDA3.SuperClass
{
  Property OrganType As %String;

  Property TransplantPhysician As HS.SDA3.CodeTableDetail.CareProvider;

  Storage Default
  {
    <Data name="SuperClassState">
      <Subscript>"SuperClass"</Subscript>
      <Value name="1"><Value>ActionCode</Value></Value>
      <Value name="2"><Value>ActionScope</Value></Value>
      <Value name="3"><Value>EnteredBy</Value></Value>
      <Value name="4"><Value>EnteredAt</Value></Value>
      <Value name="5"><Value>EnteredOn</Value></Value>
      <Value name="6"><Value>ExternalId</Value></Value>
      <Value name="7"><Value>EncounterNumber</Value></Value>
      <Value name="8"><Value>FromTime</Value></Value>
      <Value name="9"><Value>ToTime</Value></Value>
      <Value name="10"><Value>Deleted</Value></Value>
      <Value name="11"><Value>UpdatedOn</Value></Value>
      <Value name="12"><Value>CustomPairs</Value></Value>
      <Value name="13"><Value>OrganType</Value></Value>
      <Value name="14"><Value>TransplantPhysician</Value></Value>
    </Data>
    <SequenceNumber>5</SequenceNumber>
    <Type>%Library.SerialState</Type>
  }
}
```

3.2.1.1 Health Insight での類似のカスタム・データ・クラスの作成

いずれかのカスタム SDA セクションにカスタム SDA データ・クラスを定義して、Health Insight でデータを解釈して保存できるようにする場合は、Health Insight で類似のカスタム・データ・クラスを作成する必要があります。

この場合、カスタム SDA ストリームレット・クラスとカスタム SDA データ・クラスの両方が HSCUSTOM ネームスペースに存在し、アナリティクス・ネームスペースにパッケージ・マッピングされる必要があります。これらのクラスは、アナリティクス・ネームスペースでコンパイルする必要もあります。

Health Insight でカスタム SDA データ・クラスを作成するには、以下の操作を実行します。

1. **HSAA.IndexCommonData** を拡張したクラスを新規作成します。クラス名に HSAA パッケージの名前を使用しないようにしてください。
2. クラスに **User.ZMyHISSection** のような名前を付けます。将来の標準 SDA3 クラス名と競合しないように、短いクラス名に接頭語“Z”を付けることをお勧めします。Health Insight のカスタム・データ・クラスがカスタム SDA データ・クラスと同じ名前でないことを確認します。

クラスは SDA プロパティの標準セットを継承します。

ActionCode	EncounterNumber	EnteredOn	ToTime
ActionScope	EnteredAt	ExternalId	UpdatedOn
CustomPairs	EnteredBy	FromTime	

3. 独自の SDA プロパティを追加できます。文字列以外のデータタイプを使用する場合は、既存の **HSAA.Internal.Boolean**、**HSAA.Internal.Numeric**、または **HSAA.TimeStamp** のデータタイプ・クラスを使用します。

以下のタイプのプロパティを追加できます。

- ・ %String。
- ・ 任意の **HSAA.Internal<DataType>** (Boolean や Numeric など)。
- ・ 任意の既存の HSAA シリアル・クラス。
- ・ 作成した任意のカスタム・シリアル・クラス (**HSAA.Internal.DataType** を拡張する必要があります)。HSAA など、すぐに使用可能な Health Insight パッケージにカスタム・シリアル・クラスを定義しないでください。これにより、クラス名が Health Insight の将来の標準クラス名と競合しないようにします。例えば、カスタム・シリアル・クラスには、**HSAA.MySDASection** ではなく **ZUser.HSAA.MySDASection** のような名前を付ける必要があります。
- ・ 新規作成した任意のコード・テーブル (**HSAA.Internal.Interface.CodeTableTranslated** を拡張する必要があります)。以下のコード・テーブルは変換できません。

テーブル 3-4: 変換から除外されるコード・テーブル

HealthCareFacility	CareProviderType	Country
Organization	City	County
User	State	Trust
CareProvider	Zip	

4. SDA カスタム・データ・クラスが **HS.SDA3.CodeTableDetail** クラスを使用する場合は、類似の **HSAA.Interface.CodeTableDetail** クラスを使用します。これが存在しない場合は、類似の **HSAA.Internal.Interface.CodeTableDetail** クラスを使用します。

- Health Insight でタグや他のフィールドを使用する場合は、他の HSAA クラスを拡張するのではなく、そのタグやフィールドをクラスに追加します。
- クラスをコンパイルします。

カスタム Health Insight データ・クラスをコンパイルしたら、登録します。カスタム Health Insight データ・クラスの登録方法については、“[Health Insight インストールおよび設定ガイド](#)”の“[カスタム・コンテナ・クラスの登録](#)”を参照してください。そのセクションで説明する手順に従います。ただし、[SDA ソース・クラス] フィールドにはカスタム SDA データ・クラス名を入力し、[Health Insight クラス名] フィールドには類似の Health Insight カスタム・データ・クラスを入力します。

3.2.2 カスタム SDA ストリームレット・クラスの作成

カスタム SDA コンテナ作成の 2 つ目の手順では、カスタム・データ・クラスごとに、データを格納するためのカスタム SDA ストリームレット・クラスを定義します。

- HS.SDA3.Streamlet.Abstract と %Persistent の両方を拡張するクラスを新規作成します。
- クラスに **User.Streamlet.ZMySection** のような名前を付けます。将来の標準 SDA3 クラス名と競合しないように、短いクラス名に接頭語“Z”を付けることをお勧めします。

前述のクラス名は、<PackageName>.streamlet.<SDADataClassname> というストリームレット・クラスの標準の命名規則を示しています。この命名規則に従わない場合は、SDA データ・クラスの HS.SDA3.SuperClass の GetStreamletClass() メソッドを実装して、ストリームレット・クラスの名前を出力する必要があります。既定のコードは次のとおりです。

```
// for a standard SDA class, we get the streamlet class from the extension class,
// to let the extension override it.  Classes that don't allow extension
// can override this method
If pType["."] Quit $Parameter("HS.Local.SDA3."_pType_"Extension","STREAMLETCLASS")
// For custom ones, if we have a parameter, use it
If $Parameter("STREAMLETCLASS")="" Quit $Parameter(",STREAMLETCLASS")
//The code below is for backwards compatibility - it predates the parameter
//and can be overridden if need be, but really one should just use the parameter
Quit $P(pType,".",1,$L(pType,".")-1)_.Streamlet."_$P(pType,".",1,$L(pType,"."))
```

- 以下のパラメータの値を入力します。

パラメータ	説明	必須かどうか
INFOTYPE	事前定義の情報タイプまたはカスタムの情報タイプ。	必須
SDACCLASS	SDA データ・クラスの名前。	必須
DATEPROPERTY	日付でのフィルタ処理に使用するプロパティの名前。指定されていない場合、既定値の ToTime に設定されます。	オプション
MATCHINGS	HealthShare Unified Care Record で対象のセクションに対する照合を実行する方法を指定します。Health Connect には適用されません。 照合の詳細は、HS.SDA3.Streamlet.Abstract のクラス・リファレンスを参照してください。 例については、いずれかの SDA3 ストリームレット・クラスのクラス・リファレンスを参照してください。	必須
ACTIONSCOPES	このストリームレット・クラスに、限定された一連の有効な ACTIONSCOPES 値が設定されている場合、このパラメータはそれらの値のコンマ区切りリスト（先頭と末尾にもコンマあり）である必要があります。HealthShare Unified Care Record 専用で、Health Connect には適用されません。	オプション

4. オプションで、HealthShare Unified Care Record に実装するコールバックの以下のコードを入力します (Health Connect および InterSystems IRIS for Health には適用されません)。
 - ・ OnInactivate()
 - ・ OnMatchActionScope()
 - ・ OnBeforeMatch()
 - ・ OnValidate()
 - ・ OnBeforeSave() – 一時多次元プロパティ **Stash** を使用して、OnBeforeSave() から OnAfterSave() にデータを渡すことができます：
 - ・ OnAfterSave()
5. クラスをコンパイルします。

以下の例は、移植手術に関する情報のカスタム・ストリームレット・クラスを示しています。

Class Definition

```
Class User.Streamlet.ZTransplant Extends (HS.SDA3.Streamlet.Abstract, %Persistent)
{
  Parameter INFOTYPE = "PRC";
  Parameter SDACCLASS = "User.ZTransplant";
  Parameter DATEPROPERTY = "EnteredOn";
  Parameter MATCHINGS = "PRC/EnteredOn";

  Storage Default
  {
    <ExtentSize>100000</ExtentSize>
    <SequenceNumber>5</SequenceNumber>
    <Type>%Library.Storage</Type>
  }
}
```

3.2.3 カスタム SDA コンテナ・クラスの作成

カスタム SDA コンテナ作成の 3 つ目の手順では、カスタム SDA セクションを格納するカスタム SDA コンテナ・クラスを定義します。

1. HS.SDA3.Container を拡張するクラスを新規作成します。
2. コンテナ・クラスに **User.ZMyContainer** のような名前を付けます。将来の標準 SDA3 クラス名と競合しないように、短いクラス名に接頭語 “Z” を付けることをお勧めします。
3. 新規の SDA データ・セクションごとにプロパティを追加します。プロパティに **ZMySections** (複数形) という名前を付けるか、またはデータ・クラスに StartXMLList() メソッドおよび EndXMLList() メソッドを実装している場合はその値をプロパティ名として使用します。プロパティは、**User.ZMySection** (単数形) のリストである必要があります。
4. クラスをコンパイルします。

注釈 <PROTECT> エラーが発生する場合、HSLIB データベースを一時的に書き込み可能にし、クラスを再コンパイルしてから、HSLIB を読み取り専用に戻します。

以下の例は、移植手術に関するセクションを格納するカスタム・コンテナ・クラスを示しています。

Class Definition

```
Class User.ZMyContainer Extends HS.SDA3.Container
{
Parameter XMLNAME = "Container";
Property ZTransplants As list Of User.ZTransplant;
}
```

3.2.4 構成レジストリへのカスタム SDA コンテナの登録

カスタム SDA コンテナ作成の最後の手順では、構成レジストリにカスタム・コンテナを登録します。

1. 構成レジストリに、以下のようなエントリを作成します。
 - ・ キーが \CustomSDA3Container である
 - ・ 値が、接尾語 “.cls” の付かないコンテナ・クラスの名称である

3.3 名前と値のペアを使用した SDA の拡張

以下のいずれかの方法を使用して、SDA を拡張できます。

3.3.1 カスタムの SDA の名前と値のペアの作成

SDA の各セクションには、<CustomPairs> 要素が含まれています。既存の SDA セクションに、追加データ項目を取得する名前と値のペアを追加するには、該当するセクションに <CustomPairs> 要素が含まれる SDA ドキュメントを登録するだけです。名前と値の各ペアは、<NVPair> 要素に記述されます。<CustomPairs><NVPair><Name> プロパティにデータの説明を入力し、<CustomPairs><NVPair><Value> プロパティにデータを格納します。

以下の例は、SDA のアレルギーに関するセクションに一連の治療を追加する方法を示しています。

XML

```
<Container>
...
  <Allergies>
    <Allergy>
      ...
      <CustomPairs>
        <NVPair>
          <Name>Treatment</Name>
          <Value>Oral Corticosteroids</Value>
        </NVPair>
        <NVPair>
          <Name>Treatment</Name>
          <Value>Injected Steroids</Value>
        </NVPair>
      </CustomPairs>
    </Allergy>
  </Allergies>
  ...
</Container>
```

3.3.2 カスタム SDA オブジェクトの作成

SDA には、他の SDA セクションに関連しないデータの格納に使用できる <CustomObjects> セクションを設定できます。このセクションは、1 つ以上の <CustomObject> 要素で構成されます。

各 <CustomObject> 要素に、以下の要素を 1 つ以上含める必要があります。

- ・ <CustomType> – エントリのタイプを指定します。
- ・ <CustomPairs> – データを格納する名前と値のペアのリストです。[前述のセクション](#)を参照してください。

各 <CustomObject> には以下の要素を含めることもできます。

- ・ <ActionCode><ActionScope> – 実行するアクションを示した、HealthShare Unified Care Record 製品に対する指示。具体的な詳細は、[クラス・リファレンス](#)を参照してください。
- ・ <CustomMatchKey> – データベースに既に存在するオブジェクトと、対象タイプのエントリとを照合する方法を示した、HealthShare Unified Care Record 製品に対する指示。通常は、カスタム・ペア・エントリを連結したのになります (例：<NVPair3>|<NVPair1>)。各値は照合で必須になり、指定された順序で評価されます。
- ・ <ExternalID> – 外部システムに対して意味を持つ可能性がある識別子。指定されている場合、プライマリ照合キーとして使用されます。
- ・ <EnteredOn>、<EnteredAt>、<EnteredBy> – ソースの詳細。
- ・ <FromTime>、<ToTime> – データが有効な期間に関する詳細。

詳細は、HS.SDA3.CustomObject の[クラス・リファレンス](#)を参照してください。

以下の例では、2 件の理学療法の訪問ケアに関するデータが格納されています。

XML

```
<Container>
...
<Patient>
...
</Patient>
...
<CustomObjects>
  <CustomObject>
    <CustomType>HomeCareEvent</CustomType>
    <EnteredOn>2012-02-05T13:00:00Z</EnteredOn>
    <CustomPairs>
      <NVPair>
        <Name>EventType</Name>
        <Value>Physiotherapy</Value>
      </NVPair>
      <NVPair>
        <Name>Comment</Name>
        <Value>Minimal progress. Assistance required for many tasks.</Value>
      </NVPair>
    </CustomPairs>
  </CustomObject>
  <CustomObject>
    <CustomType>HomeCareEvent</CustomType>
    <EnteredOn>2012-02-01T12:55:00Z</EnteredOn>
    <CustomPairs>
      <NVPair>
        <Name>EventType</Name>
        <Value>Physiotherapy</Value>
      </NVPair>
      <NVPair>
        <Name>Comment</Name>
        <Value>Home evaluation. Bathroom rails and detachable shower head required.</Value>
      </NVPair>
    </CustomPairs>
  </CustomObject>
</CustomObjects>
...
</Container>
```

