



# HealthShare Health Connect リリース・ノート

Version 2023.1  
2024-01-02

HealthShare Health Connect リリース・ノート  
HealthShare 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動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

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

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

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

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

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

# 目次

1 HealthShare Health Connect 2023.1 の新機能	1
1.1 分析と AI の強化	1
1.1.1 列指向ストレージ	1
1.2 スピード、スケール、セキュリティの強化	1
1.2.1 外部テーブル	1
1.2.2 メモリ設定	2
1.2.3 プラットフォームのスケーラビリティ	2
1.3 プラットフォームの更新	2
2 HealthShare Health Connect 2022.3 の新機能	3
2.1 開発者エクスペリエンスの強化	3
2.1.1 SQL の開発	3
2.2 分析と AI の強化	3
2.3 スピード、スケール、セキュリティの強化	4
2.4 プラットフォームの更新	4
3 HealthShare Health Connect 2022.2 の新機能	5
3.1 分析と AI の強化	5
3.1.1 列指向ストレージ (試験的機能)	5
3.2 スピード、スケール、セキュリティの強化	5
3.2.1 SQL の機能強化	5
3.3 プラットフォームの更新	7
4 Health Connect 2022.1 の新機能	9
4.1 相互運用性の向上	9
4.1.1 Kafka メッセージングのサポート	9
4.1.2 Embedded Python	9
4.1.3 Python での相互運用プロダクション	10
4.1.4 Visual Studio Code ObjectScript Extension Pack の更新	10
4.1.5 相互運用プロダクション内での最小限のコードでの SQL クエリの実行	11
4.2 クラウドと操作の向上	11
4.2.1 クラウド・コネクタ	17
4.2.2 IKO の機能強化	11
4.3 スピード、スケール、セキュリティの強化	11
4.3.1 適応型の SQL オプティマイザ	11
4.3.2 ストレージの節約	12
4.3.3 TLS 1.3 のサポート (OpenSSL 1.1.1)	12
4.3.4 新しい ^TRACE ユーティリティ	12
4.4 その他の機能強化と効率の向上	13
5 Health Connect 2021.2 の新機能	15
5.1 相互運用性の向上	15
5.1.1 組み込み Python	15
5.1.2 Python での相互運用プロダクション	16
5.1.3 Visual Studio Code ObjectScript Extension Pack の更新	16
5.1.4 相互運用プロダクション内での最小限のコードでの SQL クエリの実行	17
5.2 クラウドと操作の向上	17
5.2.1 クラウド・コネクタ	17
5.2.2 IKO の機能強化	17

5.3 スピード、スケール、セキュリティの強化 .....	17
5.3.1 適応型の SQL オプティマイザ .....	17
5.3.2 ストレージの節約 .....	18
5.3.3 TLS 1.3 のサポート (OpenSSL 1.1.1) .....	18
5.3.4 新しい ^TRACE ユーティリティ .....	18
5.4 Health Connect の最初の継続配布リリース .....	19
5.5 その他の機能強化と効率の向上 .....	19
6 Health Connect 2021.1 の新機能 .....	21
6.1 医療相互運用性の向上 .....	21
6.1.1 FHIR® および IHE プロファイルの機能強化 .....	21
6.1.2 FHIR R4 データ変換の機能強化 .....	22
6.1.3 FHIR リポジトリおよび FHIR 相互運用アダプタ .....	22
6.1.4 クライアント側の FHIR 操作の API .....	22
6.1.5 FHIR サーバの新しい構成 UI .....	22
6.1.6 IHE RMU プロファイルのサポート .....	22
6.1.7 IHE コネクタソンの更新 .....	22
6.1.8 HL7 の生産性ツールの機能強化 .....	22
6.2 相互運用性を向上させる機能強化 .....	23
6.3 操作の機能強化 .....	23
7 Health Connect 2020.1 の新機能 .....	25
7.1 医療用途の相互運用性 .....	26
7.1.1 FHIR R4 のサポート .....	26
7.1.2 HL7 の生産性ツール .....	26
7.2 API 管理 .....	26
7.2.1 InterSystems API Manager .....	27
7.2.2 Open API/Swagger 仕様優先の REST 開発 .....	27
7.3 Caché、Ensemble、Health Connect からのインプレース変換 .....	27
7.4 管理ポータルの新たな外観 .....	28
7.5 SQL の機能強化 .....	28
7.5.1 ユニバーサル・クエリ・キャッシュ .....	28
7.6 相互運用プロダクションの機能強化 .....	29
7.6.1 Java および .NET でプロダクション・コンポーネントをコーディングするための新しい PEX フレームワーク .....	29
7.6.2 相互運用プロダクションでポートの使用を監視するポート・オーソリティ .....	29
7.6.3 X12 検証の機能強化 .....	29
7.6.4 強化された X12 の DTL サポート .....	30
7.6.5 XSD ファイルからの X12 スキーマのインポート .....	30
7.6.6 MQTT アダプタ .....	30
7.7 インフラストラクチャおよびクラウド導入の改善 .....	30
7.8 新規の自動構成カスタマイズ .....	31
7.9 分析の機能強化 .....	31
7.9.1 選択的キューブ構築 .....	31
7.9.2 PowerBI コネクタ .....	31
7.9.3 ピボット・テーブル・プレビュー .....	32
7.10 データベースのパフォーマンスとスケーラビリティの向上 .....	32
7.11 その他の機能強化と効率の向上 .....	32
8 Health Connect 2019.1 の新機能 .....	35
8.1 Health Connect 2019.1 の新機能 .....	35
8.1.1 FHIR STU3 のサポート .....	35

8.1.2 Java ビジネス・ホスト .....	35
8.1.3 Managed File Transfer (MFT) .....	36
8.1.4 Health Connect のコンテナ化 .....	36
8.1.5 2019.1.1 メンテナンス・リリースの新機能 .....	36
8.2 Health Connect 2019.1 の他の変更 .....	37
8.2.1 機能強化 .....	37
8.2.2 クラスの削除と非推奨 .....	40
8.2.3 メソッドの置換 .....	41
8.2.4 その他の非推奨 .....	41
8.3 Health Connect 2019.1 の修正 .....	42
8.3.1 FHIR DSTU2 および STU3 の並べ替え .....	42
8.3.2 不明なリソース・タイプに対する HTTP 応答 .....	42
8.3.3 MHD Find Document Manifest 要求の正しい URL .....	42
8.3.4 XDS.b メッセージの新しい最大 ID 長 .....	43
8.3.5 AA コード解決時の識別子タイプの除外 .....	43
8.3.6 DSUB トランザクションのタイム・ゾーン・オフセット .....	43
8.3.7 BirthTime に関する C-CDA v2.1 エクスポートの修正 .....	43
8.3.8 C-CDA v2.1 インポートでの名前修飾子 BR の Birth への変換 .....	43
8.4 Health Connect 15.03 から Health Connect 2019.1 へのアップグレード .....	43
9 このリリースへのアップグレード .....	45
9.1 コンテナのアップグレード .....	45
10 非推奨の機能とサポート中止の機能 .....	47
10.1 サポート中止のテクノロジーと機能 .....	47
10.2 非推奨になった Spark コネクタ .....	47



# 1

## HealthShare Health Connect 2023.1 の新機能

このページでは、HealthShare® Health Connect の 2023.1 リリースの新機能と機能強化について説明します。このリリースは、拡張メンテナンス (EM) リリースです。これらの機能の一部は、2022.1 (前回の EM リリース) 以降の継続配布 (CD) リリースでも利用可能です。

### 1.1 分析と AI の強化

#### 1.1.1 列指向ストレージ

列指向ストレージは、InterSystems SQL テーブルの新しいストレージ・オプションです。列指向ストレージは、InterSystems IRIS での従来の行クエリよりも桁違いに速い分析クエリを提供します。こうしたクエリでは通常、非常に大きいテーブルのデータを集計し、1 つ以上の列に対してフィルタ処理およびグループ化を行います。行ごと (一度に少数の行に対してトランザクションを行う場合にはこれが最適です) ではなく列ごとにテーブル・データを並べることで、こうしたクエリに必要な I/O 量を大幅に減らすことができます。また、SIMD (Single Instruction Multiple Data) という最新のチップセットレベルの最適化を活用し、ベクトル化されたクエリ処理の一部としてさらにパフォーマンスを向上させることができます。

**注釈** この機能は、HealthShare Health Connect 2022.2 で試験的機能として初めて利用できるようになりました。2023.1 では、実稼働環境での使用が完全にサポートされるようになりました。ただし、シャード・テーブルでの列指向ストレージの使用は例外です。この組み合わせに対するサポートは、将来のリリースで提供される予定です。この機能の試験バージョンを使用していたお客様は、2023.1 にアップグレードした後、すべての列指向のテーブル・データを再ロードする必要があります。

詳細は、“SQL テーブルのストレージ・レイアウトの選択” を参照してください。

### 1.2 スピード、スケール、セキュリティの強化

#### 1.2.1 外部テーブル

このリリースでは、HealthShare Health Connect で外部データを活用するための新機能を導入しています。InterSystems IRIS で作成するすべての SQL クエリに対して、外部テーブルは通常の InterSystems テーブルとして提供されます。ただし、そのデータは Health Connect サーバ内に物理的には格納されません。データはリモート・ファイル、サードパーティ・データベース (オンプレミスまたは DBaaS)、または ECP 接続が現実的でない別個の Health Connect サーバに存

在します。つまり、これらのテーブルのデータは、Health Connect インスタンスによって管理されるのではなく、Health Connect インスタンスに投影されます。

Health Connect 2023.1 では、CSV ファイルおよび JDBC データ・ソースからのデータの投影がサポートされています。JDBC オプションは、既存の SQL ゲートウェイ・インフラストラクチャを活用して接続の詳細と資格情報を管理しており、ファイル・ソースを使用するための構文や機能は既存の LOAD DATA コマンドと完全に一致しています。

**重要**            外部テーブルは、Health Connect 2023.1 では試験的機能として利用できます。つまり、実稼働環境ではサポートされません。ただし、この機能は十分にテストされており、お客様に大きな価値をもたらすことができると考えています。

インターシステムズでは、この新機能を実際の環境で使用したお客様からのフィードバックをお待ちしています。ご自分の体験を共有したい場合、または質問がある場合は、Developer Community を参照するか、インターシステムズのサポート窓口 (WRC) までお問い合わせください。

詳細は、“外部テーブル” を参照してください。

## 1.2.2 メモリ設定

Health Connect の新規インストールでは、シャード・メモリおよびロック・テーブルのサイズ設定に、よりスマートな既定値を使用するようになりました。この新しい既定値により、構成されたグローバル・バッファ・サイズに基づいてベスト・プラクティスの構成が適用され (ユーザによって設定されていない場合は、使用可能なシステム・メモリが考慮されます)、ほとんどのワークロードでうまく機能します。以前と同様、ユーザは引き続き特定の値でこれらの既定値をオーバーライドできます。既存の設定は影響を受けません。

## 1.2.3 プラットフォームのスケラビリティ

このリリースには、大規模なプロダクションの導入で非常に厳しいワークロードに対応できるようにする、スケラビリティの機能強化が数多く含まれています。これらの機能強化には、デジャーナリング時のジャーナル・ファイルの非同期読み取りや、リソースの使用を最適化して非常に負荷が高い状態での競合を制限するエンタープライズ・キャッシュ・プロトコル (ECP) のインフラストラクチャの変更などがあります。

# 1.3 プラットフォームの更新

このリリースでは、以下の新しいサーバ・プラットフォームに対するサポートが追加されています。

- ・ macOS 13 (Ventura)



# 2

## HealthShare Health Connect 2022.3 の新機能

このページでは、HealthShare® Health Connect の 2022.3 リリースの新機能と機能強化について説明します。このリリースは、継続配布 (CD) リリースです。

### 2.1 開発者エクスペリエンスの強化

#### 2.1.1 SQL の開発

InterSystems SQL では、クエリ・プランに機械で判読可能な形式が提供されるようになりました。この新しいオプションを使用して、`$SYSTEM.SQL.Explain()` メソッドは、プラン自体とアクセスするテーブルおよびインデックスに関する詳細を含む、クエリ・プランの JSON ベースのレンダリングを生成します。以前の XML ベースの形式では、簡単な英語のフレーズを使用してさまざまなステップを記述していました。新しい形式は、より徹底的な分析またはクエリ・プランのグラフィカルなレンダリングを実行するツールで使いやすいものになっています。

このリリースでは、SQL 文の実行時パラメータをサンプリングするオプトイン機能を導入しています。文インデックスでは既に、各文の詳細な実行時統計やクエリ・プランなど、一連の豊富なメタデータが記録されており、これには通常、クエリ・キャッシュ・コードをパラメータ化するプレースホルダに代入されるリテラルが含まれています。これで、文インデックスを、これらのパラメータの実際の実行時値のサンプリングで拡張できるようになりました。そして、これらを正規化された文のテキストと組み合わせ、例えば新しいハードウェア環境をベンチマーキングしたり、スキーマのさまざまなインデックスのセットを試すなど、他の導入に対して実行できる典型的なワークロードを構築したりすることができます。

InterSystems SQL では、アプリケーション環境を設定および破棄するためにスクリプトに含める、`CREATE SCHEMA` および `DROP SCHEMA` コマンドがサポートされるようになりました。

### 2.2 分析と AI の強化

このリリースでは、分析と AI に対して以下の機能強化が行われています。

- ・ InterSystems Reports の更新バージョン (Logi Report 19.2) の主な改善点は以下のとおりです。
  - ブックマーク機能 – パラメータとフィルタを Web レポートに保存します。
  - Report サーバで利用可能な Report Studio – サーバから直接レポートを追加編集できます。
  - 詳細は、<https://devnet.logianalytics.com/hc/en-us/articles/9898557594903-Logi-Report-v19-2-Release-Notes> を参照してください。

- ・ Adaptive Analytics の更新バージョン (AtScale 2022.3) の主な改善点は以下のとおりです。
  - Microsoft Excel でのタイムライン機能のサポート
  - AtScale Semantic Layer をデータ・カタログ・ベンダーに公開する Data Catalog API
  - 詳細は、<https://documentation.atscale.com/2022.3.1/release-notes/new-features-and-improvements> (ログインが必要) を参照してください。

## 2.3 スピード、スケール、セキュリティの強化

このリリースでは、シャード・クラスタに完全な弾力性を提供しています。DBA は API メソッドを呼び出して、削除するシャードをマークできるようになりました。このメソッドは、指定したシャードからクラスタ内の他のデータ・ノードにデータをオフロードし、すべてのデータ・バケットが他のシャードに移動したら、ノードを自動的に切断します。このプロセスは、オンライン再分散と同じメカニズムを利用します。つまり、ユーザは引き続きシャード・テーブルを照会し、データの移動中にそのデータを取り込むことができます。

## 2.4 プラットフォームの更新

このリリースでは、以下の新しいサーバ・プラットフォームに対するサポートが追加されています。

- ・ Oracle Linux 9
- ・ SUSE 15 SP4

# 3

## HealthShare Health Connect 2022.2 の新機能

このページでは、HealthShare® Health Connect の 2022.2 リリースの新機能と機能強化について説明します。このリリースは、継続配布 (CD) リリースです。

### 3.1 分析と AI の強化

#### 3.1.1 列指向ストレージ (試験的機能)

列指向ストレージは、IRIS の従来の行指向ストレージと比べて桁違いに速い分析クエリを提供する、IRIS SQL テーブルの新しいストレージ・オプションです。こうしたクエリでは通常、非常に大きいテーブルのデータを集計し、1 つ以上の列をフィルタ処理およびグループ化します。行ごと (一度に少数の行に対してトランザクションを行う場合にはこれが最適です) ではなく列ごとにテーブル・データを並べることで、こうしたクエリの実行に必要な I/O 量を大幅に減らすことができます。また、SIMD (Single Instruction Multiple Data) という最新のチップセットレベルの最適化を活用して、さらにパフォーマンスを向上させることができます。つまり、正味のパフォーマンスの向上は、実行しているチップセットにも左右されません。

**重要** 列指向ストレージは、リリース 2022.2 では試験的機能として利用できます。つまり、実稼働環境ではサポートされません。この機能は十分にテストされており、分析クエリに対しては非常に大きなパフォーマンス上のメリットがあることが確認されています。ただし、パフォーマンスはクエリの構造、実際のデータ、実行しているプラットフォーム (チップセット) によって変化します。

インターシステムズでは、お客様がテスト環境で実際に使用しているシステムで見られる、パフォーマンス上のメリットとスペースの節約の可能性に関するフィードバックをお待ちしています。ご自分の体験を共有したい場合、または質問がある場合は、インターシステムズのサポート窓口 (WRC) までお問い合わせください。

詳細は、“SQL Table Storage Layout” を参照してください。

### 3.2 スピード、スケール、セキュリティの強化

#### 3.2.1 SQL の機能強化

2022.2 リリースには、SQL の機能強化が数多く含まれています。

### 3.2.1.1 SQL プロセス・ビュー

SQL プロセス・ビューは、システム上で現在実行されているすべての SQL 文を参照するために、システム管理ポータルで、SQL にアクセス可能なシンプルなビューと対応するページを提供します。これにより、管理者は文インデックスで取得した情報に基づいて、異常に長い時間実行されている可能性のあるクエリを迅速に識別し、この実行時間を同じクエリの履歴実行時統計と直ちに比較できます。

この変更は、同じ内部インフラストラクチャを利用して対応するメトリックを公開する、弊社の [System Alerting and Monitoring \(SAM\)](#) 製品と完全に統合されています。

### 3.2.1.2 シャード・クラスタ全体での負荷を分散

このリリースで、シャード・クラスタでのオブジェクト・データ・モデルに対して完全にサポートされるようになりました。以前より、個別のオブジェクトスタイルのアクセスは既にサポートされており、そのようなコード（本質的にプロシージャ）は呼び出されたノードで実行されていました。このリリースでは、シャード・クラスタのすべてのノードでプロシージャ・コードを呼び出すための、2 つの使いやすい API メソッドを導入しています。

Broadcast() メソッドは、各データまたは計算ノードでルーチンまたはメソッドを 1 回呼び出すメソッドで、クラスタ全体で複雑な ObjectScript コードを開始するために使用できます（フィードから新しいデータを取り出してローカルに挿入するためなど）。別個の Map() メソッドは、そのインスタンスが物理的に格納されているデータ・ノードで、シャード・クラスタのインスタンスごとにルーチンまたはメソッドを 1 回呼び出します。これらのメソッドは、単一サーバ上のプロセス全体に作業を分散する作業キュー・マネージャと同じシングニチャ・スタイルに従います。

(%SYSTEM.ShardWorkMgr 内の) これらの 2 つのメソッドにより、シャード・クラスタを利用するのに必要な柔軟性が開発者に提供されます。

### 3.2.1.3 ラテラル結合のサポート

このリリースでは、ラテラル結合のサポートが導入されています。これは、テーブルまたはサブクエリを結合する方法で、結合されたそれぞれのストリームが別個に評価されることはありません。典型的なユース・ケースとして、サブクエリに、結合されている別のテーブルへの参照、または結合されている別のテーブルの列値を入力として取るテーブル値関数が含まれる場合があります。LATERAL JOIN は、標準の ANSI SQL 構文です。

### 3.2.1.4 その他

- スキーマ管理の向上 — このリリースでは、テーブルやビュー用の CREATE IF NOT EXISTS、およびプロシージャや関数などのコード・アーティファクト用の CREATE OR REPLACE など、数多くの便利な拡張機能を IRIS DDL に導入しています。また、DDL エクスポート・ユーティリティは、DDL として表現できるより多くのクラス定義機能をカバーするようになりました。%SYSTEM.SQL.Schema の ExportDDL() メソッドを参照してください。
- リスト・コレクションの投影 — 以前に Array of コレクション・プロパティが投影されていたのと同じような方法で、List of スタイル・コレクション・プロパティを子テーブルに投影できるようになりました。つまり、物理ストレージ・オプションと SQL プロジェクションは完全に独立しています。
- パフォーマンスの向上 — SQL クエリの実行時に、JOIN、GROUP BY、ORDER BY およびその他の構文を使用することは、クエリ・プランのある部分で一時的なデータ構造（内部的に“tempfile”と呼ばれます）が構築された後、クエリ・プランの別の部分によってそれが読み取られることを意味します。この変更は、内部構文を使用してこれらの一時的なデータ構造を格納し、利用可能なメモリを適切に活用して、全体的なクエリ・パフォーマンスを向上させる方法に関するいくつかの変更をまとめています。これらの変更では、プロセス・プライベート・メモリの新しい既定値を利用します（新しい既定の bbsiz 設定により、既定で無制限になります）。

## 3.3 プラットフォームの更新

RedHat Enterprise Linux の更新 : 2022.2 リリースでは、RHEL 9 のサポートが追加され、RHEL 7 のサポートは終了しました。

Ubuntu の更新 : 2022.2 リリースでは、Ubuntu 22.04 のサポートが追加され、Ubuntu 18.04 のサポートは終了しました。



# 4

## Health Connect 2022.1 の新機能

このページでは、HealthShare® Health Connect の 2022.1 リリースの新機能と機能強化について説明します。このリリースは、拡張メンテナンス (EM) リリースです。これらの機能の一部は、2021.1 (前回の EM リリース) 以降の継続配布 (CD) リリースでも利用可能です。

### 4.1 相互運用性の向上

#### 4.1.1 Kafka メッセージングのサポート

このリリースでは、Apache Kafka <https://kafka.apache.org/> がサポートされます。Apache Kafka はオープンソースの分散型イベント・ストリーミング・プラットフォームであり、高パフォーマンスのデータ・パイプライン、ストリーミング分析、データ統合、およびミッションクリティカル・アプリケーションに使用されます。Kafka を相互運用プロダクションで使用したり (“Kafka メッセージングの使用法” を参照)、プロダクションの外部で共通のメッセージング API を使用したりできます。

#### 4.1.2 Embedded Python

Health Connect 2022.1 では、カーネルに完全に統合された Python が導入され、Python は ObjectScript と完全に対等の機能を持つようになりました。クラス・メソッドの定義をはじめ、ObjectScript で可能な操作はほぼすべて Python でも実行できるようになりました。さらに、Python と ObjectScript を混在させることもでき、Python のコードを一切記述することなく Python ライブラリを ObjectScript から直接呼び出すことも可能です。Python では、数千もの高品質な事前構築済みライブラリを利用できるので、開発スピードを速め、メンテナンス・コストを削減できます。Python 開発者であれば、ObjectScript に精通していなくても、新しい言語を学ぶことなく開発を始めることができます。(初導入 2021.2)

あらゆる Health Connect オブジェクトを Embedded Python で作成してアクセスすることができます。

- Embedded Python で実装したオブジェクトは、ObjectScript で実装したオブジェクトと同じように扱われます。
- Embedded Python では、Python オブジェクトから ObjectScript オブジェクトにも、ObjectScript オブジェクトから Python オブジェクトにも完全に直接アクセスできます。
- Embedded Python でグローバルに完全にアクセスすることができます。その際、グローバルは通常の Python オブジェクトとしてアクセスされます。Health Connect の永続化を使用してオブジェクトをデータベースに保存すると、明示的に削除しない限り、そのオブジェクトを今後のセッションで使用できます。

Embedded Python は Health Connect Python SDK を拡張します。この SDK には、クライアント・ライブラリと外部 Python ゲートウェイが含まれます。

Embedded Python の概要は、“組み込み Python の概要” を参照してください。

注釈 Embedded Python は、お使いのマシンにインストールされているどのバージョンの Python でも動作するように設計されています。

Microsoft Windows を実行している場合に、Python がまだインストールされていないときは、Health Connect インストール・キットによってインストールされます。

UNIX または Linux の多くのフレーバーには、Python がインストール済みの状態で付属します。Python をインストールする必要がある場合は、パッケージ・マネージャによってオペレーティング・システムに対して推奨されるバージョンを使用してください。以下に例を示します。

- ・ macOS:Homebrew (<https://formulae.brew.sh/formula/python@3.9>) を使用して Python 3.9 をインストール
- ・ Ubuntu:apt-get install python3
- ・ Red Hat Enterprise Linux または Oracle Linux:yum install python3
- ・ SUSE:zypper install python3

“Failed to load python” というエラーが表示される場合は、Python がインストールされていないか、予期されていないバージョンの Python がシステムにインストールされていることを意味します。上記のいずれかの方法で Python をインストールまたは再インストールしてください。

UNIX ベースのシステムでは、Python パッケージを pip3 コマンドでインストールできます。pip3 がまだインストールされていない場合は、システムのパッケージ・マネージャを使用して python3-pip パッケージをインストールしてください。

### 4.1.3 Python での相互運用プロダクション

このリリースでは、Production EXtension (PEX) フレームワークを使用して、相互運用プロダクションを Python で開発できます。このため、Python、Java、.NET、または ObjectScript でのプロダクションの開発を選択できます。異なる言語で開発された複数のプロダクション・コンポーネントを簡単に組み合わせることができます。他のプロダクション・コンポーネントが異なる言語で開発されていても、自身が精通した言語で開発できます。Python を PEX と共に使用すると、新しいプロトコル・アダプタの作成、複雑な分析や計算の実行、永続メッセージングや長期実行型のビジネス・プロセスの作成を行うことができます。詳細は、“外部言語によるプロダクションの開発” を参照してください。(初導入 2021.2)

### 4.1.4 Visual Studio Code ObjectScript Extension Pack の更新

Visual Studio Code ObjectScript Extension Pack は [Visual Studio Code のダウンロード・ページ](#) から入手可能で、コードの開発を迅速化・容易化する以下の機能強化が含まれます。

- ・ ドキュメントの統合 – インライン・ドキュメントにカーソルを合わせて、クラス階層を参照したり、カスタム・クラスのドキュメントをプレビューしたりできます。
- ・ サーバ側のソースの改善 – さまざまなクライアント側の Web アプリケーション・ワークフローの検索とサポート。
- ・ デバッグ – オブジェクトのプロパティの検査と信頼性の向上。

この Extension Pack には、ObjectScript 拡張機能と Language Server 拡張機能が含まれます。詳細は、[VSCode ObjectScript Extension のドキュメント](#) (<https://intersystems-community.github.io/vscode-objectscript/>) を参照してください。(初導入 2021.2)



## 4.1.5 相互運用プロダクション内での最小限のコードでの SQL クエリの実行

このリリースでは、新しい SQL ビジネス・サービスとビジネス・オペレーションにより、SQL クエリをプロダクション内で簡単に実行できます。詳細は、“SQL ビジネス・サービスの使用法” および “SQL ビジネス・オペレーションの使用法” を参照してください。(初導入 2021.2)

# 4.2 クラウドと操作の向上

## 4.2.1 クラウド・コネクタ

このリリースにはクラウド・コネクタが含まれており、これによって Amazon Web Services 内で Health Connect アプリケーションを簡単に管理したり、コネクタを使用してサービスに簡単にアクセスしたりできます。このリリースには以下のアダプタが含まれます。

- ・ S3 (Amazon Simple Storage Service) 用の受信アダプタと送信アダプタ (初導入 2021.2)
- ・ Cloudwatch (Amazon の監視サービス) 用の送信アダプタ (初導入 2021.2)
- ・ SNS (Amazon Simple Notification Service) によるメッセージング用の送信アダプタ (初導入 2021.2)
- ・ Healthlake (Amazon の医療データ・サービス) 用の送信アダプタ (初導入 2021.2)

## 4.2.2 IKO の機能強化

このリリースでは、以下の新しい InterSystems Kubernetes Operator (IKO) 機能により、Kubernetes 内での Health Connect の導入と管理が容易になります。

- ・ IKO を使用すると、InterSystems System Alerting and Monitoring (SAM) および InterSystems API Manager (IAM) を Health Connect クラスターで導入・管理できます。これにより、システムの管理と拡張が容易になります。(初導入 2021.2)
- ・ IKO では、Health Connect および InterSystems Web ゲートウェイのロック・ダウン・コンテナを導入できます。(初導入 2021.2)
- ・ IKO では、InterSystems Web ゲートウェイ・コンテナを Nginx および Apache の Web サーバと共に導入できます。(初導入 2021.2)
- ・ IKO では、導入環境に一時ボリュームと永続ボリュームを含めることができます。(初導入 2021.2)

# 4.3 スピード、スケール、セキュリティの強化

## 4.3.1 適応型の SQL オプティマイザ

Health Connect SQL オプティマイザは、テーブル統計を利用して、ユーザが送信した文それぞれに最適なクエリ・プランを導出します。また、効率的なクエリ・キャッシュを使用して、生成されたコードを再利用します。このような文にパラメータが含まれる場合、実行時に送信される値により、代替クエリ・プランを使用して実行を高速化する機会が得られることがあります。このリリースで導入された新しい実行時プラン選択 (RTPC) インフラストラクチャにより、Health Connect SQL はこのような機会を確実に効率的に利用します。RTPC は、異常値が使われているかどうかスキャンし、より詳細なテ

ブル統計に基づいて値域条件の選択性を効率的に予想します。この結果、現実のデータセットの多くでクエリ・プランの適応性が向上し、実行時間と I/O が大幅に削減されます。(初導入 2021.2)

さらに Health Connect は、フル・スキャンまたは行ベースのスキャンではなく、ブロック・レベルのサンプリングを使用して、オプティマイザが使用するテーブル統計を収集するようになりました。この効率的なアルゴリズムにより、数十億行が含まれる最大級のテーブルでも、数秒で統計を収集できます (例えば、TUNE TABLE コマンドを使用)。また、Health Connect SQL は、適切なクエリ・プランを保証するものがテーブルにない場合、テーブル統計をその場で収集するようになります。(初導入 2021.2)

### 4.3.2 ストレージの節約

このリリースでは、ストリームとジャーナルの圧縮により、Health Connect 導入環境に必要なストレージを大幅に削減できます。

- ・ ストリームの圧縮 – グローバル・ベースのすべてのストリーム・クラスに対して既定でオンになりました。アプリケーションの変更は必要ありません。既存のデータは読み取り可能なまま残り、次回書き込み時に圧縮されます。現実のデータで試したところ、圧縮率は、短いテキストでは 30%、XML などのドキュメント・タイプでは 80% 以上と、幅があることがわかりました。(初導入 2021.2)
- ・ ジャーナルの圧縮 – ジャーナルが切り替わるとすぐに、非アクティブなジャーナル・ファイルを圧縮します。ロールバックとロール・フォワードは圧縮形式から直接実行されます。これにより、Health Connect のデータ整合性戦略において不可欠なこの部分のストレージ要件が大きく減ります。詳細は、“ジャーナリングの最善の使用法”を参照してください。(初導入 2021.2)

### 4.3.3 TLS 1.3 のサポート (OpenSSL 1.1.1)

このバージョンでは、Health Connect に OpenSSL 1.1.1 のサポートが組み込まれ、TLS 1.3 が完全にサポートされます。TLS 1.3 では、暗号化の遅延が半分に短縮されるなど、ユーザは特にパフォーマンスの高速化を体感できます。これは、ハンドシェイク・プロセスからラウンドトリップ全体を排除することによって実現しています。(初導入 2021.2)

Health Connect 2021.2 より、UNIX では OpenSSL ライブラリを出荷しなくなり、この機能は OS に依存して提供するようになります。この変更には利点が 1 つあり、OpenSSL ライブラリを更新する場合に Health Connect を新規インストールする必要がなくなり、通常の OS の更新で実行できるようになります。この新しいアプローチの詳細は、“[自身の InterSystems IRIS のインスタンスでサポートされている TLS のバージョン](#)”を参照してください。インターシステムズ製品は、OS で提供される OpenSSL ライブラリへのアクセスが必要になるため、インストール時およびインスタンスの起動時にチェックを実行するようになります。この呼び出しは手動で実行することもできます。詳細は、“[必要な依存関係のインストール](#)”を参照してください。

OpenSSL ライブラリの出荷停止という変更を受けて、キットの調整も必要になりました。すべてのキットは OpenSSL のメジャー・バージョンに固有です (OpenSSL 1.1.1 はメジャー・バージョンであり、マイナー・バージョンは OpenSSL 1.1.1f のようにメジャー・バージョンに続く文字で示されます)。

Windows では、このキットで OpenSSL ライブラリはインストールされません。

**注釈** UNIX システムに OpenSSL の正しいバージョンがインストールされていない場合、インストールは正常に完了しません。OpenSSL をインストールしてから、Health Connect を再インストールする必要があります。例えば、macOS では、Homebrew を使用して OpenSSL をインストールできます。<https://formulae.brew.sh/formula/openssl@1.1> を参照してください。

### 4.3.4 新しい ^TRACE ユーティリティ

このリリースでは、1 つ以上のプロセスからの未加工のイベントをトレースする新しいツールが導入されます。%SYS.MONLBL や PERFMON などの既存のユーティリティもほぼ同じイベント・タイプを追跡しますが、こうしたユーティリティは、特定のタイプの分析に合わせた形式のレポートを即座に生成します。新しい ^TRACE ツールでは、これらのイベントをより汎用

的なファイル形式でキャプチャし、キャプチャした情報を、コマンド行インタフェースまたは API を介して対話形式で操作および要約できます。サポートされるイベント・タイプには、グローバル SET と KILL、物理書き込み、ネットワーク要求、キャッシュのヒットと読み取り、さまざまなジャーナル・イベントなどがありますが、これらに限定されません。これらのイベントでキャプチャされる情報としては、ルーチン行とコール・スタックに加え、完全なグローバル参照（該当する場合）もあります。このため、1 つのインタフェースで幅広いパフォーマンス分析タスクを実行することができます。（初導入 2021.2）

## 4.4 その他の機能強化と効率の向上

各リリースで、インターシステムズは多くの効果的な改善とマイナーな機能強化を行っています。このリリースには以下が含まれます。

- ・ 一般的な堅牢性とミラー・フェイルオーバー時の復元可能性を確保するために、DataMove が機能強化されました。
- ・ JDBC、.NET クライアント、Python、IRISNative などの外部クライアントでの Compact Double のサポート。（初導入 2021.2）
- ・ このリリースでは、セキュリティ・テーブルに埋め込みのバージョン番号が含まれるようになったため、実行可能なインポートに、より詳細にアクセスできます。セキュリティ・テーブルをバージョン 2021.1 からエクスポートして、このバージョンにインポートできます。詳細は、“SECURITY” を参照してください。（初導入 2021.2）
- ・ このリリースにより、Log4j ライブラリがバージョン 2.17.0 に更新されます。（初導入 2021.2）
- ・ このリリースにより、Node.js ライブラリがバージョン 14 に更新されます。（初導入 2021.2）



# 5

## Health Connect 2021.2 の新機能

このトピックでは、HealthShare® Health Connect の 2021.2 リリースの新機能と機能強化について説明します。2021.1 バージョンにはなかった 2021.2 の新機能を取り上げます。これは、Health Connect の最初の[継続配布 \(CD\) リリース](#)です。

### 5.1 相互運用性の向上

#### 5.1.1 組み込み Python

Health Connect 2021.2 では、カーネルに完全に統合された Python が導入され、Python は ObjectScript と完全に対等の機能を持つようになりました。クラス・メソッドの定義をはじめ、ObjectScript で可能な操作はほぼすべて Python でも実行できるようになりました。さらに、Python と ObjectScript を混在させることもでき、Python のコードを一切記述することなく Python ライブラリを ObjectScript から直接呼び出すことも可能です。Python では、数千もの高品質な事前構築済みライブラリを利用できるので、開発スピードを速め、メンテナンス・コストを削減できます。Python 開発者であれば、ObjectScript に精通していなくても、新しい言語を学ぶことなく開発を始めることができます。

あらゆる Health Connect オブジェクトを Embedded Python で作成してアクセスすることができます。

- Embedded Python で実装したオブジェクトは、ObjectScript で実装したオブジェクトと同じように扱われます。
- Embedded Python では、Python オブジェクトから ObjectScript オブジェクトにも、ObjectScript オブジェクトから Python オブジェクトにも完全に直接アクセスできます。
- Embedded Python でグローバルに完全にアクセスすることができます。その際、グローバルは通常の Python オブジェクトとしてアクセスされます。Health Connect の永続化を使用してオブジェクトをデータベースに保存すると、明示的に削除しない限り、そのオブジェクトを今後のセッションで使用できます。

Embedded Python は Health Connect Python SDK を拡張します。この SDK には、クライアント・ライブラリと外部 Python ゲートウェイが含まれます。

Embedded Python の概要は、“組み込み Python の概要”を参照してください。

注釈 Embedded Python は、お使いのマシンにインストールされているどのバージョンの Python でも動作するように設計されています。

Microsoft Windows を実行している場合に、Python がまだインストールされていないときは、InterSystems IRIS インストール・キットによってインストールされます。

UNIX または Linux の多くのフレーバーには、Python がインストール済みの状態で付属します。Python をインストールする必要がある場合は、パッケージ・マネージャによってオペレーティング・システムに対して推奨されるバージョンを使用してください。以下に例を示します。

- ・ macOS:Homebrew (<https://formulae.brew.sh/formula/python@3.9>) を使用して Python 3.9 をインストール
- ・ Ubuntu:`apt-get install python3`
- ・ Red Hat Enterprise Linux または Oracle Linux:`yum install python3`
- ・ SUSE:`zypper install python3`

“Failed to load python” というエラーが表示される場合は、Python がインストールされていないか、予期されていないバージョンの Python がシステムにインストールされていることを意味します。上記のいずれかの方法で Python をインストールまたは再インストールしてください。

UNIX ベースのシステムでは、Python パッケージを `pip3` コマンドでインストールできます。pip3 がまだインストールされていない場合は、システムのパッケージ・マネージャを使用して `python3-pip` パッケージをインストールしてください。

## 5.1.2 Python での相互運用プロダクション

このリリースでは、Production EXtension (PEX) フレームワークを使用して、相互運用プロダクションを Python で開発できます。このため、Python、Java、.NET、または ObjectScript でのプロダクションの開発を選択できます。異なる言語で開発された複数のプロダクション・コンポーネントを簡単に組み合わせることができます。他のプロダクション・コンポーネントが異なる言語で開発されていても、自身が精通した言語で開発できます。Python を PEX と共に使用すると、新しいプロトコル・アダプタの作成、複雑な分析や計算の実行、永続メッセージングや長期実行型のビジネス・プロセスの作成を行うことができます。詳細は、“外部言語によるプロダクションの開発” を参照してください。

## 5.1.3 Visual Studio Code ObjectScript Extension Pack の更新

Visual Studio Code ObjectScript Extension Pack は [Visual Studio Code のダウンロード・ページ](#) から入手可能で、コードの開発を迅速化・容易化する以下の機能強化が含まれます。

- ・ ドキュメントの統合 – インライン・ドキュメントにカーソルを合わせて、クラス階層を参照したり、カスタム・クラスのドキュメントをプレビューしたりできます。
- ・ サーバ側のソースの改善 – さまざまなクライアント側の Web アプリケーション・ワークフローの検索とサポート。
- ・ デバッグ – オブジェクトのプロパティの検査と信頼性の向上。

この Extension Pack には、ObjectScript 拡張機能と Language Server 拡張機能が含まれます。詳細は、[VSCode ObjectScript Extension のドキュメント](#) (<https://intersystems-community.github.io/vscode-objectscript/>) を参照してください。

## 5.1.4 相互運用プロダクション内での最小限のコードでの SQL クエリの実行

このリリースでは、新しい SQL ビジネス・サービスとビジネス・オペレーションにより、SQL クエリをプロダクション内で簡単に実行できます。詳細は、“SQL ビジネス・サービスの使用法” および “SQL ビジネス・オペレーションの使用法” を参照してください。

# 5.2 クラウドと操作の向上

## 5.2.1 クラウド・コネクタ

このリリースにはクラウド・コネクタが含まれており、これによって Amazon Web Services 内で Health Connect アプリケーションを簡単に管理したり、コネクタを使用してサービスに簡単にアクセスしたりできます。このリリースには以下のアダプタが含まれます。

- ・ S3 (Amazon Simple Storage Service) 用の受信アダプタと送信アダプタ
- ・ Cloudwatch (Amazon の監視サービス) 用の送信アダプタ
- ・ SNS (Amazon Simple Notification Service) によるメッセージング用の送信アダプタ
- ・ Healthlake (Amazon の医療データ・サービス) 用の送信アダプタ

## 5.2.2 IKO の機能強化

このリリースでは、以下の新しい InterSystems Kubernetes Operator (IKO) 機能により、Kubernetes 内での Health Connect の導入と管理が容易になります。

- ・ IKO を使用すると、InterSystems System Alerting and Monitoring (SAM) および InterSystems API Manager (IAM) を Health Connect クラスターで導入・管理できます。これにより、システムの管理と拡張が容易になります。
- ・ IKO では、Health Connect および InterSystems Web ゲートウェイのロック・ダウン・コンテナを導入できます。
- ・ IKO では、InterSystems Web ゲートウェイ・コンテナを Nginx および Apache の Web サーバと共に導入できます。
- ・ IKO では、導入環境に一時ボリュームと永続ボリュームを含めることができます。

# 5.3 スピード、スケール、セキュリティの強化

## 5.3.1 適応型の SQL オプティマイザ

Health Connect SQL オプティマイザは、テーブル統計を利用して、ユーザが送信した文それぞれに最適なクエリ・プランを導出します。また、効率的なクエリ・キャッシュを使用して、生成されたコードを再利用します。このような文にパラメータが含まれる場合、実行時に送信される値により、代替クエリ・プランを使用して実行を高速化する機会が得られることがあります。このリリースで導入された新しい実行時プラン選択 (RTPC) インフラストラクチャにより、Health Connect SQL はこのような機会を確実に効率的に利用します。RTPC は、異常値が使われているかどうかスキャンし、より詳細なテーブル統計に基づいて値域条件の選択性を効率的に予想します。この結果、現実のデータセットの多くでクエリ・プランの適応性が向上し、実行時間と I/O が大幅に削減されます。



さらに Health Connect は、フル・スキャンまたは行ベースのスキャンではなく、ブロック・レベルのサンプリングを使用して、オプティマイザが使用するテーブル統計を収集するようになりました。この効率的なアルゴリズムにより、数十億行が含まれる最大級のテーブルでも、数秒で統計を収集できます (例えば、TUNE TABLE コマンドを使用)。また、Health Connect SQL は、適切なクエリ・プランを保証するものがテーブルにない場合、テーブル統計をその場で収集するようになります。

### 5.3.2 ストレージの節約

このリリースでは、ストリームとジャーナルの圧縮により、Health Connect 導入環境に必要なストレージを大幅に削減できます。

- ・ ストリームの圧縮 – グローバル・ベースのすべてのストリーム・クラスに対して既定でオンになりました。アプリケーションの変更は必要ありません。既存のデータは読み取り可能なまま残り、次回書き込み時に圧縮されます。現実のデータで試したところ、圧縮率は、短いテキストでは 30%、XML などのドキュメント・タイプでは 80% 以上と、幅があることがわかりました。
- ・ ジャーナルの圧縮 – ジャーナルが切り替わるとすぐに、非アクティブなジャーナル・ファイルを圧縮します。ロールバックとロール・フォワードは圧縮形式から直接実行されます。これにより、Health Connect のデータ整合性戦略において不可欠なこの部分のストレージ要件が大きく減ります。詳細は、“ジャーナリングの最善の使用法”を参照してください。

### 5.3.3 TLS 1.3 のサポート (OpenSSL 1.1.1)

このバージョンでは、Health Connect に OpenSSL 1.1.1 のサポートが組み込まれ、TLS 1.3 が完全にサポートされます。TLS 1.3 では、暗号化の遅延が半分に短縮されるなど、ユーザは特にパフォーマンスの高速化を体感できます。これは、ハンドシェイク・プロセスからラウンドトリップ全体を排除することによって実現しています。

Health Connect 2021.2 より、UNIX では OpenSSL ライブラリを出荷しなくなり、この機能は OS に依存して提供するようになります。この変更には利点が 1 つあり、OpenSSL ライブラリを更新する場合に Health Connect を新規インストールする必要がなくなり、通常の OS の更新で実行できるようになります。この新しいアプローチの詳細は、“[自身の InterSystems IRIS のインスタンスでサポートされている TLS のバージョン](#)”を参照してください。インターシステムズ製品は、OS で提供される OpenSSL ライブラリへのアクセスが必要になるため、インストール時およびインスタンスの起動時にチェックを実行するようになります。この呼び出しは手動で実行することもできます。詳細は、“[必要な依存関係のインストール](#)”を参照してください。

OpenSSL ライブラリの出荷停止という変更を受けて、キットの調整も必要になりました。すべてのキットは OpenSSL のメジャー・バージョンに固有です (OpenSSL 1.1.1 はメジャー・バージョンであり、マイナー・バージョンは OpenSSL 1.1.1f のようにメジャー・バージョンに続く文字で示されます)。

Windows では、このキットで OpenSSL ライブラリはインストールされません。

**注釈** UNIX システムに OpenSSL の正しいバージョンがインストールされていない場合、インストールは正常に完了しません。OpenSSL をインストールしてから、Health Connect を再インストールする必要があります。例えば、macOS では、Homebrew を使用して OpenSSL をインストールできます。<https://formulae.brew.sh/formula/openssl@1.1> を参照してください。

### 5.3.4 新しい ^TRACE ユーティリティ

このリリースでは、1 つ以上のプロセスからの未加工のイベントをトレースする新しいツールが導入されます。%SYS.MONLBL や PERFMON などの既存のユーティリティもほぼ同じイベント・タイプを追跡しますが、こうしたユーティリティは、特定のタイプの分析に合わせた形式のレポートを即座に生成します。新しい ^TRACE ツールでは、これらのイベントをより汎用的なファイル形式でキャプチャし、キャプチャした情報を、コマンド行インタフェースまたは API を介して対話形式で操作および要約できます。サポートされるイベント・タイプには、グローバル SET と KILL、物理書き込み、ネットワーク要求、キャッシュのヒットと読み取り、さまざまなジャーナル・イベントなどがありますが、これらに限定されません。これらのイベン



トでキャプチャされる情報としては、ルーチン行とコール・スタックに加え、完全なグローバル参照（該当する場合）もあります。このため、1 つのインタフェースで幅広いパフォーマンス分析タスクを実行することができます。

## 5.4 Health Connect の最初の継続配布リリース

Health Connect 2021.2 は、Health Connect の最初の継続配布リリースです。Health Connect のリリースには以下の 2 つのストリームがあります。

- ・ 継続配布リリース—これらのリリースは、新機能へのアクセスを提供します。継続的に更新されるアプリケーションの開発・導入に最適であり、新機能を即座に利用できます。
- ・ 拡張メンテナンス・リリース—これらのリリースは、継続配布リリースほど頻繁ではありませんが、安定性が向上したメンテナンス・リリースを提供します。新機能に早期にアクセスできることよりも、メンテナンス・リリースで修正を簡単に入手できることの方が重要な大規模エンタープライズ・アプリケーションに最適です。

CD リリースと EM リリースはどちらもすべてのサポート対象プラットフォームで提供され（“サポート対象プラットフォーム・ガイド”を参照）、UNIX、Windows、クラウド・プラットフォーム、OCCI に準拠したコンテナ形式も含まれます。以前の CD リリースは、クラウド・プラットフォームと OCCI に準拠したコンテナ形式でのみ提供されていました。

インターシステムズでは、完全にサポートされるリリースを提供する以外にも、新機能をいち早く確認したい開発者向けにプレリリース・ソフトウェアへのアクセスも提供しています。

## 5.5 その他の機能強化と効率の向上

各リリースで、インターシステムズは多くの効果的な改善とマイナーな機能強化を行っています。このリリースには以下が含まれます。

- ・ JDBC、.NET クライアント、Python、IRISNative などの外部クライアントでの Compact Double のサポート。
- ・ このリリースでは、セキュリティ・テーブルに埋め込みのバージョン番号が含まれるようになったため、実行可能なインポートに、より詳細にアクセスできます。セキュリティ・テーブルをバージョン 2021.1 からエクスポートして、このバージョンにインポートできます。詳細は、“SECURITY”を参照してください。
- ・ このリリースにより、Log4j ライブラリがバージョン 2.17.0 に更新されます。
- ・ このリリースにより、Node.js ライブラリがバージョン 14 に更新されます。



# 6

## Health Connect 2021.1 の新機能

このトピックでは、HealthShare® Health Connect の 2021.1 リリースの新機能と機能強化について説明します。2020.1.0 バージョンにはなかった 2021.1 の新機能を取り上げます。

### 6.1 医療相互運用性の向上

#### 6.1.1 FHIR® および IHE プロファイルの機能強化

InterSystems HealthShare Health Connect 2021.1 では、InterSystems HealthShare Health Connect の FHIR® サーバ・フレームワークの機能を大幅に強化しました。この機能強化により、FHIR パッケージをロードし、それを使用して FHIR サーバ・エンドポイントを構成できるなど、FHIR プロファイルを統合し、利用するためのユーザフレンドリな標準メカニズムが提供されます。InterSystems HealthShare Health Connect は、外部に公開された FHIR プロファイルと共にカスタム・プロファイルもサポートしています。また、US Core プロファイル (US Core Implementation Guide v3.1.0) にもすぐに対応できます。詳細は、“[FHIR のプロファイルと適応](#)”を参照してください。

HealthShare Health Connect 2021.1 では、XML の Xpath に似た FHIR 向けのナビゲーションおよび抽出言語である、FHIRPath をサポートするようになりました。FHIRPath は、スキーマレベルの条件および検索パラメータ・パスなどを表現するために FHIR 仕様で使用されます。InterSystems HealthShare Health Connect には、FHIR データに対して FHIRPath 式を解析および評価するための API が含まれており、公式の FHIRPath 仕様で定義されている各種関数や演算のサブセットをサポートしています。

このリリースでは、InterSystems HealthShare Health Connect の FHIR® サーバ・フレームワークの機能を強化することにより、FHIR パッケージをロードし、それを使用して FHIR サーバ・エンドポイントを構成できるなど、FHIR プロファイルを統合し、利用するためのユーザフレンドリな標準メカニズムが提供されます。InterSystems HealthShare Health Connect は、外部に公開された FHIR プロファイルと共にカスタム・プロファイルもサポートしています。また、US Core プロファイル (US Core Implementation Guide v3.1.0) にもすぐに対応できます。詳細は、“[FHIR のプロファイルと適応](#)”を参照してください。

RMD (Remove Metadata and Documents) は、患者のケア・レコードで検出する必要がなくなったメタデータとドキュメントを、それぞれ XDS.b Document Registry と XDS.b Document Repository から削除するプロセスを定義する新しい IHE プロファイルです。InterSystems HealthShare Health Connect では、現在、この IHE プロファイルの Document Registry アクターと Document Repository アクターをサポートしています。

## 6.1.2 FHIR R4 データ変換の機能強化

InterSystems HealthShare Health Connect のこのリリースでは、FHIR R4 と SDA 間の双方向のデータ変換を提供します。これには、変換 API メソッドに加え、データ変換を実行するための基本のビジネス・プロセス・コンポーネントも含まれています。詳細は、“[SDA-FHIR 変換](#)”を参照してください。

## 6.1.3 FHIR リポジトリおよび FHIR 相互運用アダプタ

一部のソリューションでは、要求を内部リポジトリにルーティングする FHIR サーバが必要ですが、要件は FHIR 要求を受信して外部 FHIR サーバに転送することだけで、ペイロードを保存する必要がない場合もあります。このリリースでは、以下の 2 つの異なる要件に対処する 2 つの重要な機能を提供します。

- ・ FHIR リポジトリ – [FHIR リソース・リポジトリ](#)は、FHIR サーバの既定のストレージ・ストラテジで、他の開発タスクなしに、完全に機能する FHIR® サーバをインストールすることができます。FHIR リソース・リポジトリは、このリリースの Health Connect で利用可能なアドオン・コンポーネントです。
- ・ FHIR 相互運用アダプタ – [FHIR 相互運用アダプタ](#)は、特殊なビジネス・ホストを使用して、プロダクションで FHIR 要求を処理する新しい相互運用 REST エンドポイントを作成します。Health Connect ライセンスに FHIR リソース・リポジトリが含まれていない場合は、FHIR 相互運用アダプタにより、プロダクションで FHIR 要求を受信する簡単な手段が提供されます。

以前のリリースでも FHIR 要求を処理できましたが、これらの新機能によって簡単に処理できるようになり、プロダクションでカスタム・コードを開発する必要性が減少しました。

## 6.1.4 クライアント側の FHIR 操作の API

FHIR 要求/応答メッセージを送受信するための新しい API。これにより、プロダクションでクライアント側の FHIR 操作が可能になります。詳細は、“[FHIR クライアント](#)”を参照してください。

## 6.1.5 FHIR サーバの新しい構成 UI

新しい UI ページが追加され、管理ポータルから、FHIR サーバを直接作成および構成できるようになりました。このページにアクセスするには、[Health] → [FHIR 構成] → [サーバ構成] に移動します。

## 6.1.6 IHE RMU プロファイルのサポート

RMU (Restricted Metadata Update) は、制御された方法で、コミュニティ境界内で、またコミュニティ境界を越えて、ドキュメント共有メタデータを変更するメカニズムを提供する新しい IHE プロファイルです。InterSystems HealthShare Health Connect では、現在、このプロファイルの Update Initiator アクターと Update Responder アクターの両方をサポートしています。

## 6.1.7 IHE コネクタソンの更新

InterSystems HealthShare Health Connect のこのリリースには、2020年の北米の IHE コネクタソンのすべてのソフトウェアの更新とテスト結果が含まれています。これには、2020 年 1 月より前に承認された公式の IHE 変更提案に関連する各種修正の統合が含まれます。

## 6.1.8 HL7 の生産性ツールの機能強化

HL7 移行ツールでの eGate サポート。変換ロジックを eGate インタフェース・エンジンから InterSystems HealthShare Health Connect に移行します。詳細は、“[HL7 移行ツール](#)”を参照してください。

## 6.2 相互運用性を向上させる機能強化

InterSystems HealthShare Health Connect 2021.1 では、[InterSystems API Manager \(IAM\)](#) 2.3 を導入できます。これには、最新の API 中心の環境において、この重要なコンポーネントの対象範囲を拡大する数多くの機能強化が含まれています。

その他にも、以下のような相互運用性の機能強化があります。

- ・ 新しい SOAP ビジネス・サービスとビジネス・オペレーション、**EnsLib.EDI.X12.Service** および **EnsLib.EDI.X12.Operation**。これにより、SOAP を使用して X12 メッセージの送受信が可能になります。
- ・ X12 エラー処理の改善。
- ・ このリリースでは、新しい“foreach”アクションをサポートするようになりました。これは、セグメント化された仮想ドキュメント (ASTM、EDIFACT および X12) に使用されるルーティング・ルール内で使用できます。foreach アクションはルール・タイプ“Segmented Virtual Document Message Routing Rule”内でサポートされます。また、仮想ドキュメントでの繰り返しセグメントのループが可能で、入れ子のループもサポートしています。これにより、開発者は、繰り返しグループ内のセグメントの位置に関係なく、特定の条件に適合するルールを構築できます。詳細は、“アクションについて”を参照してください。
- ・ [Proof Key for Code Exchange \(PKCE\)](#) を OAuth 認証で使えるようになりました。PKCE により、パブリック・クライアントから OAuth 交換を安全に実行し、承認コードが傍受される脅威を軽減できます。PKCE は OAuth のクライアントとサーバの両方でサポートされます。

## 6.3 操作の機能強化

このリリースでは、クラウドとオンプレミスの両方において、導入および操作エクスペリエンスに対する以下の機能強化が行われました。

- ・ InterSystems Kubernetes Operator (IKO) では、Health Connect 固有の知識とベスト・プラクティスが、ダイナミック・クラスタのプロビジョニングと操作のための、簡単に使用できる自動化されたツールにパッケージ化されています。2021.1 以降、IKO は、InterSystems System Alerting & Monitoring (SAM) の導入もサポートしています。
- ・ InterSystems Cloud Manager (ICM) は、InterSystems API Manager と SAM の導入をサポートするようになりました。
- ・ このリリースには、シャード・クラスタに対する非同期ミラーリングのサポートが含まれています。ユーザは、既存のクラスタにミラーリング (同期または非同期) を構成したり、災害復旧シナリオで別のデータ・センターにある一連の非同期ミラー・メンバにクラスタ全体をフェイルオーバーできるようにしました。詳細は、“スケーラビリティ・ガイド”の対応するセクションを参照してください。
- ・ InterSystems SQL 構文が、SQL プロンプトからデータベースを管理、構成するための新たなコマンド・セットにより拡張されました。これにより、JDBC または ODBC アクセスしか持たないユーザが、システム管理ポータルまたは ObjectScript ターミナル・プロンプトにアクセスすることなく、ほとんどの管理タスクを実行できるようになります。また、インデックスの構築や凍結プランの管理などの共通のタスクも含まれています。詳細は、“BUILD INDEX”、“プランを凍結”、“キャッシュ処理 QUERIES”、“CREATE INDEX”、および“SET OPTION”の新しいオプションを参照してください。
- ・ システム管理ポータルから作業キューを管理できるようになりました。
- ・ 新たに利用可能になった IRIS ロック・ダウン・コンテナは、セキュリティに関するベスト・プラクティスを多数実装し、セキュリティが強化されたコンテナ・イメージであるため、お客様は、複雑な環境に機密性の高いアプリケーションを安心して導入することができます。Web ゲートウェイ・コンテナのユーザにとって、既定の構成が向上することは喜ばしいものです。

- ・ 2021.1 以降、Health Connect は、ARM プラットフォームで、完全キットとしても、付属のコンテナとしても利用できるようになりました。これにより、お客様はそのアプリケーションを、物理的な環境かクラウドかを問わず、コスト効率の高いハードウェア・プラットフォームに導入することができます。詳細は、“サポート対象プラットフォーム” ガイドを参照してください。
- ・ このリリースでは、InterSystems Reports (Health Connect の新規レポート機能) の導入が簡素化されました。より緊密な統合の一環として、InterSystems Reports では、現 Health Connect と同じユーザ・アカウントを使用して、レポートを管理、構築、実行するようになりました。また、InterSystems Reports のすべての構成データと管理データは、設定スクリプトが使用されている場合、Health Connect を使用します。Health Connect Report サーバの初期構成を行ってオンプレミスで導入するためのスクリプトと、Reports サーバを Docker に導入するための Docker Compose ファイルは両方とも、このリリースに含まれています。
- ・ このリリースでは、データベース・キャッシュ・サイズが構成されていない、新たにインストールされたシステムのパフォーマンスが向上しました。ほとんどの場合、最高のシステム・パフォーマンスを実現するためには、慎重にキャッシュ・サイズを構成し、ヒュージ・ページとラージ・ページを構成する必要があります。ライブ・プロダクション・システム、負荷が重いシステム、および複数のインスタンスを持つシステムにとって、キャッシュ・サイズの構成は特に重要です。詳細は、“[グローバル・バッファおよびルーチン・バッファのメモリ使用量の変更](#)” を参照してください。
- ・ このリリースでは、ユーザが 'root' の場合または管理特権を持っている場合は、コマンド行を記録しないことで、コマンド行の履歴のセキュリティを向上させています。通常、コマンド行の履歴は ~/.iris\_history に書き込まれます (~ は \$HOME (ユーザのホーム・ディレクトリ) の値に展開されます)。現在のセッションで最初のコマンドの前にスクロールすると、ログのコマンドの履歴が使用されます。ユーザが 'root' の場合、コマンドの履歴はログに書き込まれないか、前のセッションから読み取られるため、スーパーユーザとして実行されたコマンドは公開されません。

# 7

## Health Connect 2020.1 の新機能

このページでは、HealthShare® Health Connect の 2020.1 リリースの新機能と機能強化について説明します。2019.1.0 バージョンにはなかった 2020.1 の新機能を取り上げます。これらの機能のうちいくつかは、2019.1 メンテナンス・リリース、または 2019.2、2019.3、2019.4 の継続配布リリースで初めて導入されました。これらの機能については、説明の中で示しています。以下のセクションでは、2020.1 リリースとその新機能および機能強化について説明します。

- ・ [医療用途の相互運用性](#)
  - － [FHIR R4 ベースの標準のサポート](#)
  - － [HL7 の生産性ツール](#)
- ・ [API 管理](#)
  - － [InterSystems API Manager](#)
  - － [Open API/Swagger 仕様優先の REST 開発](#)
- ・ [Cache と Ensemble からのインプレース変換](#)
- ・ [管理ポータルの新たな外観](#)
- ・ [SQL の機能強化](#)
  - － [ユニバーサル・クエリ・キャッシュ](#)
- ・ [相互運用プロダクションの機能強化](#)
  - － [Java および .NET でプロダクション・コンポーネントをコーディングするための新しい PEX フレームワーク](#)
  - － [相互運用プロダクションでポートの使用を監視するポート・オーソリティ](#)
  - － [X12 検証の機能強化](#)
  - － [強化された X12 の DTL サポート](#)
  - － [XSD ファイルからの X12 スキーマのインポート](#)
  - － [MQTT アダプタ](#)
- ・ [インフラストラクチャおよびクラウド導入の改善](#)
- ・ [新規の自動構成カスタマイズ](#)
- ・ [分析の機能強化](#)
  - － [選択的キューブ構築](#)



- [PowerBI コネクタ](#)
- [ピボット・テーブル・プレビュー](#)
- ・ [データベースのパフォーマンスとスケーラビリティの向上](#)
- ・ [その他の機能強化と効率の向上](#)

## 7.1 医療用途の相互運用性

このリリースには、医療用途の相互運用性に関する次の 2 つの新機能が含まれます。

- ・ [FHIR R4 のサポート](#)
- ・ [HL7 の生産性ツール](#)

### 7.1.1 FHIR R4 のサポート

新しい FHIR サーバ・アーキテクチャは、パフォーマンスの向上を実現し、すべての FHIR R4 ベース・リソースに対する FHIR RESTful API サポートを提供します。FHIR R4 と SDA 間の変換は、今後のリリースでサポートされる予定です。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。

### 7.1.2 HL7 の生産性ツール

このリリースには、HL7 インタフェースの構築プロセス、またはこれらを別のインタフェース・エンジンから移行するプロセスをスピードアップすることを目的に設計された、3 つの新しいツールが含まれます。

- ・ [プロダクション・ジェネレータ](#) - CSV ファイルの情報に基づいて、新しいインタフェース (ビジネス・サービス、ルータ、ルール、変換、ビジネス・オペレーション) のインフラストラクチャを生成します。バルク更新にも使用できます。詳細は、["プロダクション・ジェネレータ"](#) を参照してください。
- ・ [メッセージ・アナライザ](#) - HL7 メッセージをスキャンして、HL7 スキーマに準拠しているかどうかを確認します。メッセージが逸脱している場合、このツールを使用して、フィールド、データ構造およびコード・テーブル・レベルでカスタム・スキーマに変更を加えることができます。詳細は、["メッセージ・アナライザ"](#) を参照してください。
- ・ [移行ツール](#) - Cloverleaf インタフェース・エンジンから Health Connect に変換ロジックを移行します。今後のリリースでは、追加のインタフェース・エンジンもサポートする予定です。詳細は、["HL7 移行ツール"](#) を参照してください。

## 7.2 API 管理

このリリースには、次の 2 つの新しい API 管理機能が含まれます。

- ・ [InterSystems API Manager](#)
- ・ [Open API/Swagger 仕様優先の REST 開発](#)



## 7.2.1 InterSystems API Manager

このリリースには、Web ベースの API との間のトラフィックの監視と制御を可能にする InterSystems API Manager (IAM) が含まれています。API Manager は、メンテナンス・リリース 2019.1.1 と継続配布リリース 2019.2 でリリースされました (2019.2 の初期バージョンには API Manager は含まれません)。

サービス指向アプリケーション層を構築している場合、使用している API の数が急速に増えていると感じることがよくあります。環境の分散が大きいくほど API トラフィックの適切な管理と監視の重要性が増します。API Manager により、一元化されたゲートウェイを介してすべてのトラフィックを簡単にルーティングしたり、API 要求を適切なターゲット・ノードに転送したりすることができます。これにより、以下が可能になります。

- ・ すべての API トラフィックを一元的に監視します。
- ・ 使用している API と API を提供するサーバのリストを計画、文書化、および更新します。
- ・ 問題が深刻になる前に特定します。
- ・ スループットを制限し、許可されるペイロード・サイズを構成し、IP アドレスとドメインをホワイトリストとブラックリストに登録し、エンドポイントをすばやくメンテナンス・モードにすることで、API トラフィックを制御します。
- ・ カスタマイズ可能な専用の開発者ポータルでインタラクティブな API ドキュメントを提供し、社内外の開発者の研修を行います。
- ・ API を一元的にセキュリティ保護します。

API Manager には相互運用性、信頼性、直感的操作性、拡張性が備わっています。すべての構成をシンプルな Web ベースのユーザ・インタフェースを使用して実行できますが、API 呼び出しを使用して API Manager を構成することもできるため、リモート導入を容易に実行できます。

API Manager は、独自のコンテナでリリースされます。API Manager は複数ノードのクラスタとして構成できますが、単一ノードでも毎秒何万もの要求の負荷を処理できます。

詳細は、“InterSystems API Manager” を参照してください。

## 7.2.2 Open API/Swagger 仕様優先の REST 開発

このリリースでは、API 管理サービスを強化し、OpenAPI 2.0 仕様から REST サービスの ObjectScript コードを生成できるようにしました。生成されたこのコードが、受信した REST 呼び出しを処理するため、ユーザはサービスで実行される特定機能を実行するカスタム・コードを記述するだけで済みます。OpenAPI 2.0 仕様で既に定義されているサービスを実装する場合、作業が大幅に削減されます。既存の OpenAPI 2.0 仕様がない場合でも、REST API の定義に必要なカスタム・コードを記述するよりは、新しい仕様を作成する方がはるかに簡単です。また、この仕様はドキュメントも提供しており、サービスのクライアント・コードを開発するすべてのユーザを支援します。詳細は、“REST サービスの作成” を参照してください。(初リリース 2019.2)

## 7.3 Caché、Ensemble、Health Connect からのインプレース変換

Health Connect のこのリリースでは、Caché、Ensemble、Ensemble ベースの Health Connect の既存のインスタンスを Health Connect に変換できます。変換プロセスでは、アプリケーション・コード、構成スクリプト、およびその他のプロシージャに対して多少の変更が必要になる場合がありますが、ほとんどの場合は比較的簡単なものです。他のメジャー・アップグレードの場合と同様、任意のプロダクション・ビジネス・サービス、プロセス、およびオペレーションを含むカスタム・コードを、実際の実稼働環境へ配置する前に、テスト環境で十分にテストする必要があります。

以下の変換パスがサポートされています。

- ・ Health Connect (HSAP) 15.03x から Health Connect

- ・ Ensemble 2016.2 (またはそれ以上) から Health Connect
- ・ Caché 2016.2 (またはそれ以上) から Health Connect

インプレース変換を実行する前に、“InterSystems IRIS インプレース変換ガイド”と“InterSystems IRIS 導入ガイド”を読んで、Caché または Ensemble と Health Connect の違いに関するバックグラウンド情報を把握しておくことが重要です。これらのドキュメントは、インターシステムズのサポート窓口の[ドキュメント配布ページ](#)からダウンロードできます。

## 7.4 管理ポータルの新たな外観

このリリースから、管理ポータルの新しい、よりモダンな外観がスタートします。この最初のフェーズでは、メニューとボタンの外観が新しくなりましたが、機能の変更はありません。この新しい実装は、将来のストリーミングおよびユーザ・インタフェース向上の基礎となります。(初リリース 2019.2)

## 7.5 SQL の機能強化

どのリリースでもそうですが、Health Connect には、基盤となるソフトウェアの進歩や、業界標準および顧客のワークロードに対する継続的なベンチマーキングに基づく、SQL エンジンに対する多くの機能強化が含まれています。2019.1 リリースと比較して、高負荷時のクエリ・スループットが目に見えて向上する可能性があります。ベンチマーキングを拡張して特定の新しいユース・ケースを追加する機会がある場合は、それをインターシステムズと共有することをお勧めします。

新しいメジャー・バージョンにアップグレードすると、既存のクエリ・プランは自動的に凍結されます。これにより、ソフトウェアのメジャー・アップグレードによって既存のクエリのパフォーマンスが低下することがなくなります。パフォーマンスが重要なクエリの場合、パフォーマンスの改善が達成できたかどうかをテストする必要があります。詳細は、“InterSystems SQL 最適化ガイド”の“ソフトウェア・バージョンのアップグレードによるプランの自動凍結”を参照してください。

- ・ [ユニバーサル・クエリ・キャッシュ](#)
- ・ より多くのタイプのクエリと DML を並列化し (自動的に)、CPU 処理能力をより効率的に使用できるように、並列化エンジンを改善しました。(初リリース 2019.4)
- ・ → 構文を使用して、シャード・クエリで暗黙結合を使用できるようになりました。(初リリース 2019.4)
- ・ システム管理ポータルの SQL エクスプローラ・ページから発行されたクエリは、バックグラウンドで実行されます。これにより、クエリのキャンセルが可能になり、Web 要求のタイムアウトを回避できますが、フォアグラウンドの実行に依存して現在のデバイスに書き込む特定の従来のストア・プロシージャが、このログ情報を正しく表示しなくなる可能性があります。(初リリース 2019.3)

### 7.5.1 ユニバーサル・クエリ・キャッシュ

このリリースでは、すべてのクエリ (埋め込みクエリおよびクラス・クエリを含む) をキャッシュ・クエリとして保存できる、ユニバーサル・クエリ・キャッシュが導入されました。以前は、埋め込み SQL を使用する場合、現在のテーブル統計または新たに利用可能になったインデックスを取得するには、アプリケーション・コードをリコンパイルする必要がありました。現在は、すべてのクエリ・プランが 1 つのキャッシュで管理されており、適宜削除できます (クエリ、テーブル、ネームスペースごとに)。これにより、複数のサイトに導入されている場合に、アプリケーションが実際のデータ特性に適応する能力が大幅に向上しました。

また、すべてのクエリ・タイプが均等に、生成済みクエリ・コードに実装された効率的なデータ・アクセスを利用できるようになりました。

## 7.6 相互運用プロダクションの機能強化

### 7.6.1 Java および .NET でプロダクション・コンポーネントをコーディングするための新しい PEX フレームワーク

このリリースには、相互運用プロダクションの開発時に実装言語を選択できる Production EXtension (PEX) フレームワークが含まれます。このリリースでは、Java および .NET を使用して、ビジネス・サービス、プロセス、オペレーションに加え、インバウンド・アダプタおよびアウトバウンド・アダプタも開発できます。以前のリリースでは、ビジネス・サービスとオペレーションのみを、Java でのみコーディング可能で、管理ポータルで特別なコード・ジェネレータ・ウィザードを使用する必要がありました。PEX フレームワークは、Java および .NET コードを相互運用プロダクション・コンポーネントに接続する柔軟なプログラミングを提供します。PEX を使用することで、最小限の ObjectScript コーディングまたは ObjectScript コーディングなしで、Java および .NET コードを接続できます。PEX パッケージには、以下のクラスが含まれます。

- ・ EnsLib.PEX.BusinessService
- ・ EnsLib.PEX.BusinessProcess
- ・ EnsLib.PEX.BusinessOperation
- ・ EnsLib.PEX.InboundAdapter
- ・ EnsLib.PEX.OutboundAdapter
- ・ EnsLib.PEX.Message

詳細は、“PEX : Java および .NET によるプロダクションの開発”を参照してください。

### 7.6.2 相互運用プロダクションでポートの使用を監視するポート・オーソリティ

ポート・オーソリティ・ユーティリティでは、インターチェンジ・システムでのポートの使用状況を監視できます。ポート・オーソリティは、複数のプロダクションとインスタンスでビジネス・サービスとビジネス・オペレーションを調べ、各システムでどのポートが使用されているかを判断します。新しいサービスおよびオペレーション用に空いているポートを特定して、特定ユーザのためにポートを予約できます。詳細は、“ポートの使用の管理”を参照してください。(初リリース 2019.3)

### 7.6.3 X12 検証の機能強化

このリリースでは、次の 2 種類の拡張された X12 検証を提供します。

- ・ SNIP レベル 1 およびレベル 2 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。
- ・ X12 要素検証 – (初リリース 2019.1.1 および 2019.2)

以前のリリースでは、SNIP 検証は使用できず、セグメント構造全体を検証できるだけでした。セグメントのコンテンツを検証するメカニズムはありませんでした。

SNIP 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

X12 要素検証では、以下を検証できます。

- ・ 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。
- ・ 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可されているとおりに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が長さ制限に従っていること。

詳細は、“X12 ビジネス・プロセスの設定” の X12 の検証を参照してください。

## 7.6.4 強化された X12 の DTL サポート

このリリースでは、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・セットのスキーマを含め、X12 バッチ全体のデータ変換を定義できます。これにより、1 つのデータ変換を使用して X12 バッチ・メッセージを処理できます。サブ変換を使用する必要はありません。このリリースでは、ユーザ・インタフェースも改善されています。また、繰り返し要素の処理を容易にする便利な関数も提供されています。詳細は、“X12 データ変換の作成” を参照してください。

## 7.6.5 XSD ファイルからの X12 スキーマのインポート

以前のバージョンでは、X12 スキーマをインポートできるのは SEF ファイルまたはインターシステムズ専用の XML 形式からのみでした。このリリースでは、新しい XSD スキーマ・ファイルからも X12 スキーマをインポートできます。詳細は、“InterSystems IRIS への X12 スキーマのロード” を参照してください。

## 7.6.6 MQTT アダプタ

このリリースには、IoT アプリケーションで使用されることが多い MQTT (Message Queuing Telemetry Transport) プロトコルをサポートする MQTT アダプタが含まれます。詳細は、“プロダクション内での MQTT アダプタの使用法” を参照してください。

# 7.7 インフラストラクチャおよびクラウド導入の改善

このリリースには、インフラストラクチャおよびクラウド導入に対する、以下のような改善が含まれます。

- ・ Tencent Cloud のサポート – InterSystems Cloud Manager (ICM) では、Health Connect に基づき、Tencent Cloud で実行されるアプリケーションの、エンドツーエンドのクラウド・プロビジョニングと導入が可能になりました。(初リリース 2019.4)
- ・ バインド・マウントに加えて、Docker 名前付きボリュームがサポートされます。(初リリース 2019.4)
- ・ InterSystems Cloud Manager (ICM) での弾力的な拡張/縮小のサポート – 既存の構成を拡張/縮小して、より多いノードまたは少ないノードで再プロビジョニングおよび再導入できるようになりました。詳細は、“InterSystems Cloud Manager ガイド” の “インフラストラクチャの再プロビジョニング” と “サービスの再導入” を参照してください。(初リリース 2019.2。ノードレベル・アーキテクチャでの DATA ノードのスケール・アウトおよび COMPUTE ノードのスケール・イン/アウトは初リリース 2019.4)
- ・ 独自のコンテナをパッケージ化する際のユーザ・エクスペリエンスが向上しました。(初リリース 2019.3)
- ・ InterSystems Cloud Manager (ICM) での、ノードレベルのシャーディングのサポート。(初リリース 2019.3)
- ・ コンテナは、root 以外の既定のユーザを使用します。これはコンテナのベスト・プラクティスで、セキュリティが向上します。(初リリース 2019.3)

- ・ ICM による、プライベート・ネットワーク上での作成および導入のサポート。このネットワークで、要塞サーバはプライベート・ネットワークをパブリック・ネットワークに接続し、サービス拒否攻撃に対する向上したセキュリティ保護を提供します。(初リリース 2019.3)
- ・ セキュリティ保護された RPC 通信でのサービス検出のサポート。(初リリース 2019.3)
- ・ ICM によるマルチリージョン導入のサポート。リージョン全体が機能を停止した場合でも、高可用性を提供できます。(初リリース 2019.3)
- ・ ICM をアップグレードして、導入されたシステムの情報を保持する機能。(初リリース 2019.3)
- ・ コンテナレス・モード – コンテナレス・モードを使用した Google Cloud Platform でのシャード構成の導入、およびコンテナレス・モードを使用した Ubuntu または SUSE ノードでの Web ゲートウェイの導入は、以前は制限されていましたが、現在は実行できるようになりました。(初リリース 2019.2)

## 7.8 新規の自動構成カスタマイズ

新しい Health Connect 構成機能では、開始する前に、Health Connect インスタンスの構成パラメータ・ファイル (CPF) をカスタマイズでき、開始時にカスタム構成が自動的に実装されます。この機能により、自動化が大幅に簡素化され、Health Connect での Kubernetes などの構成管理ツールの使用がサポートされます。この機能は、このバージョンの ICM にも含まれています。自動構成カスタマイズは、今後のバージョンで拡張される重要な新機能です。(初リリース 2019.4)

## 7.9 分析の機能強化

このリリースでは、分析に対して以下の機能強化が行われています。

- ・ [選択的キューブ構築](#)
- ・ [PowerBI コネクタ](#)
- ・ [ピボット・テーブル・プレビュー](#)

### 7.9.1 選択的キューブ構築

このリリースでは、選択的キューブ構築機能を提供します。これは、個別に構築するメジャーとディメンジョンを選択できる、InterSystems IRIS Business Intelligence の機能です。キューブ全体を使用中止にしなくても、変更を加えて、選択的に再構築することができます。また、何を再構築するかを把握できるように、ユーザ・インタフェースにより、追加または変更されたディメンジョンまたはメジャーに自動的にフラグが立てられます。

### 7.9.2 PowerBI コネクタ

InterSystems ユーザは、Microsoft Power BI を使用して Health Connect に保存されている表形式のデータとキューブ・データにアクセスできるようになりました。これにより、Power BI が提供するデータ視覚化機能と、Health Connect が提供する高性能のデータ管理およびクエリ機能を組み合わせることができます。コネクタは ODBC を利用しますが、ユーザは InterSystems 2019.2 以降に接続する場合、Power BI から直接 Health Connect BI キューブにアクセスすることもできます。2019 年 4 月リリース以降、コネクタは Power BI の一部として出荷されます。詳細は、“Power BI 向け InterSystems IRIS コネクタ”を参照してください。



### 7.9.3 ピボット・テーブル・プレビュー

このリリースには、Analytics ピボット・テーブル・プレビューが含まれます。これは、切り捨てられたデータ・セットに基づいて代表的なピボット・テーブルを表示するアナライザの新しいモードです。これにより、完全な結果セットを分析するよりもずっと速く、ピボット・テーブルをプレビューできます。プレビュー・モードでは、結果セットが完全でないことを示す、**[すべて表示]** ボタンも表示されます。**[すべて表示]** ボタンを選択すると、プレビュー・モードが自動的にオフになります。(初リリース 2019.2)

## 7.10 データベースのパフォーマンスとスケーラビリティの向上

このリリースでは、データベース・エンジンが大幅に最適化されました。これは、非常に大規模なシステムでは特に重要で、非常に高い負荷を扱うためにシステムを拡張する能力が大幅に向上しました。

このリリースで変更された効率化の 1 つにより、グローバル内のデータ走査時の効率性が向上しています。頻繁にアクセスされるが、めったに変更されないメモリ内のデータベース・ブロックに対して、ブロック内のノードの検索を高速化するために、システムでノード・テーブルと呼ばれる最適化構造が自動的に構築されることがあります。これにより、特に、ノードへのアクセスがまばらまたはランダムに分散している場合、あるいは逆方向の照合順(逆方向の \$order/\$query を含む) でノードにアクセスするパターンの場合、グローバル・アクセスが高速化されます。このためのメモリはデータベース・キャッシュ自体から取得され、通常は 1% 未満とごくわずかです。

## 7.11 その他の機能強化と効率の向上

各リリースで、インターシステムズは多くの効果的な改善とマイナーな機能強化を行っています。このリリースでは、次のような改善が行われています。

- ・ ジャーナル・パフォーマンスの向上。
- ・ ミラーリングされている環境の容易な構成。
- ・ Apache Spark の新しいバージョン 2.3 と 2.4 のサポート。
- ・ Web ゲートウェイでの既存の WebSocket サーバのサポートに加えて、WebSocket クライアントのサポート。
- ・ プロダクションのソース・コントロール プロダクションをエンティティとしてチェックインおよびチェックアウトするためのソース・コントロール・フックが追加され、変更の追跡および構成管理が簡素化されました。(初リリース 2019.4)
- ・ ペネトレーション・テストをサポートするためのホワイトリスト 独自のセキュリティ・ペネトレーション・テストを実施するユーザは、CSP、Zen、REST に関連する誤検出を削減または排除できます。(初リリース 2019.4)
- ・ .NET のサポートを .NET Core 2.1 にアップグレード。(初リリース 2019.3)
- ・ ODBC データベース・アクセスの効率の向上。(初リリース 2019.3)
- ・ ログ・メッセージへのアクセスを向上させる構造化ログ。(初リリース 2019.3)
- ・ アラートを取得するための API の改善。(初リリース 2019.3)
- ・ コンパイラが長すぎるグローバル名をテストし、エラーを報告。以前は、これらのグローバル名は通知なしで切り捨てられていました。関連する互換性の問題については、"[Class Compiler Validates Global Name Length Limit](#)" を参照してください。(初リリース 2019.3)

メンテナンス・リリース 2019.1.1、継続配布リリース 2019.3、および後続のリリースには、ジャーナリングとミラーリングに関連する問題を修正した、JournalingGroup2019 として識別される一連の変更が含まれます。これらの問題に関連する変

更は、SML2776、SML2781、SML2782、SML2783、SML2785、JO2990、JO3117、JO3137、JO3140、JO3141、RJF391、RJF392、HYY2362、HYY2364、および HYY2373 です。





# 8

## Health Connect 2019.1 の新機能

Health Connect 2019.1 は、InterSystems IRIS を利用する Health Connect の最初のリリースです。このページでは、その機能、修正、およびアップグレードのチェックリスト項目について説明します。

### 8.1 Health Connect 2019.1 の新機能

Health Connect 2019.1 の主要な新機能は以下のとおりです。

- ・ [FHIR STU3 のサポート](#)
- ・ [Java ビジネス・ホスト](#)
- ・ [Managed File Transfer \(MFT\)](#)
- ・ [Health Connect のコンテナ化](#)
- ・ [2019.1.1 メンテナンス・リリースの新機能](#)

#### 8.1.1 FHIR STU3 のサポート

HealthShare Health Connect のこのリリースでは、FHIR STU3 のサポートが追加されます。この新機能により、Health Connect は FHIR DSTU2 と STU3 の両方をサポートできるようになり、FHIR を含む顧客ユース・ケースへの対応範囲が広がります。FHIR STU3 のサポートは、現在 FHIR DSTU2 の対象であるすべての FHIR 関連コンポーネントと機能に適用されます。これには、FHIR サーバとクライアントのコンポーネント、FHIR のメッセージとオブジェクト・モデル、SDA との間のデータ変換、FHIR ベースの IHE プロファイルなどがあります。SDA から FHIR STU3 への変換、および FHIR STU3 から SDA への変換を実行する DTL クラスをカスタマイズするための標準メカニズムが用意されています。こうした変換は、組み込みのビジネス・プロセスを使用するか、変換 API を呼び出すことによって実行できます。詳細は、["HealthShare Health Connect の FHIR のサポート"](#) を参照してください。

#### 8.1.2 Java ビジネス・ホスト

この機能により、ユーザは、ObjectScript をコーディングすることなくビジネス・サービスとビジネス・オペレーションを完全に Java で作成できます。ユーザが新しいプロトコルを追加したり、既存の Java ライブラリを使用して複雑な操作を実行したりできるようになりました。詳細は、["Developing Productions with Java Business Services and Operations"](#) を参照してください。

### 8.1.3 Managed File Transfer (MFT)

Managed File Transfer (MFT) は、プロダクション内からサードパーティのファイル転送サービスを直接使用できる新しい統合オプションです。この機能は、Box、DropBox、および Accellion kiteworks をサポートできるビジネス・ホストを提供します。ユーザはプロダクションでこれらのビジネス・ホストを構成して、エンドユーザ・アカウントからファイルを取得するなど、さまざまなファイル転送操作を実行できます。

### 8.1.4 Health Connect のコンテナ化

HealthShare Health Connect で Docker コンテナへの導入がサポートされるようになりました。これにより、Google、Amazon、Azure などのパブリック・クラウド・プラットフォームへの導入に適した、プラットフォームに依存しない完全に移植可能なランタイム・ソリューションとして Health Connect を導入できます。詳細は、“機能紹介：Docker コンテナ内のインターシステムズ製品”を参照してください。

### 8.1.5 2019.1.1 メンテナンス・リリースの新機能

ここでは、Health Connect 2019.1.1 メンテナンス・リリースおよび今後のメンテナンス・リリースでのみ利用可能な新機能について説明します。リリース 2019.1.0 を実行している場合、これらの機能はありません。

#### 8.1.5.1 InterSystems API Manager

このリリースには、Web ベースの API との間のトラフィックの監視と制御を可能にする InterSystems API Manager (IAM) が含まれています。

サービス指向アプリケーション層を構築している場合、使用している API の数が急速に増えていると感じることがよくあります。環境の分散が大きいほど API トラフィックの適切な管理と監視の重要性が増します。API Manager により、一元化されたゲートウェイを介してすべてのトラフィックを簡単にルーティングしたり、API 要求を適切なターゲット・ノードに転送したりすることができます。これにより、以下が可能になります。

- ・ すべての API トラフィックを一元的に監視します。
- ・ 使用している API と API を提供するサーバのリストを計画、文書化、および更新します。
- ・ 問題が深刻になる前に特定します。
- ・ スループットを制限し、許可されるペイロード・サイズを構成し、IP アドレスとドメインをホワイトリストとブラックリストに登録し、エンドポイントをすばやくメンテナンス・モードにすることで、API トラフィックを制御します。
- ・ カスタマイズ可能な専用の開発者ポータルでインタラクティブな API ドキュメントを提供し、社内外の開発者の研修を行います。
- ・ API を一元的にセキュリティ保護します。

API Manager には相互運用性、信頼性、直感的操作性、拡張性が備わっています。すべての構成をシンプルな Web ベースのユーザ・インタフェースを使用して実行できますが、API 呼び出しを使用して API Manager を構成することもできるため、リモート導入を容易に実行できます。

API Manager は、独自のコンテナでリリースされます。API Manager は複数ノードのクラスタとして構成できますが、単一ノードでも毎秒何万もの要求の負荷を処理できます。

注釈 詳細は、“InterSystems API Manager”を参照してください。

API Manager は、Docker コンテナのディストリビューションでのみ利用できます。API Manager は、UNIX、Windows、クラウド・プラットフォーム、Docker コンテナなど、任意の [InterSystems IRIS サポート対象プラットフォーム](#)にインストールされた Health Connect システムで使用できます。

### 8.1.5.2 相互運用プロダクションでの X12 要素検証

このリリースでは、拡張された X12 検証を提供します。以前のリリースでは、必須セグメントが正しい順序で並んでいること、および禁止されているセグメントが存在しないことしか検証できず、セグメントのコンテンツを検証するメカニズムはありませんでした。この機能強化では、以下を検証できます。

- ・ 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。
- ・ 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とおりに繰り返されていること。
- ・ 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が長さ制限に従っていること。

詳細は、“プロダクション内での X12 ドキュメントのルーティング” の“検証” を参照してください。

## 8.2 Health Connect 2019.1 の他の変更

Health Connect 2019.1 の追加の変更には、以下が含まれます。

- ・ [機能強化](#)
- ・ [クラスの削除と非推奨](#)
- ・ [メソッドの置換](#)
- ・ [その他の非推奨](#)

### 8.2.1 機能強化

Health Connect 2019.1 の機能強化には、以下が含まれます。

- ・ [医療用途の相互運用性の機能強化](#)
- ・ [システムの相互運用性の機能強化](#)

#### 8.2.1.1 医療用途の相互運用性の機能強化

- ・ HL7 スキーマ・エディタの機能強化 – HL7 スキーマ・エディタでは、ユーザフレンドリなグラフィカル・ユーザ・インタフェースを提供するようになり、ユーザはコンポーネントをドラッグ・アンド・ドロップするだけで HL7 スキーマを編集することができます。詳細は、“プロダクション内での HL7 バージョン 2 メッセージのルーティング” の“新しいメッセージ・タイプと構造タイプの定義” を参照してください。
- ・ HL7 スキーマのソース・コントロール – 管理ポータルをサードパーティのソース・コントロール・システムと統合して、HL7 スキーマをバージョン管理下に置くことができます。詳細は、“プロダクション内での HL7 バージョン 2 メッセージのルーティング” の“HL7 スキーマと HL7 メッセージの概要” を参照してください。
- ・ FHIR REST ハンドラによるアクセス・トークン対象ユーザの検証 – FHIR DSTU2 および STU3 用の FHIR REST ハンドラ・クラス (HS.FHIR.vDSTU2.REST.Handler および HS.FHIR.vSTU3.REST.Handler) に、指定されたアクセス・トークンの対象ユーザを現在の CSP アプリケーション・エンドポイントに対して検証するためのサポートが含まれるようになりました。この更新は、アクセス・トークンの対象ユーザの検証を強く推奨する OAuth 2.0 仕様に従ったものです。
- ・ IHE コネクタソンの更新 – Health Connect のこのバージョンは、2018 年の北米およびヨーロッパの IHE コネクタソンの変更とテスト結果をすべて反映して更新されています。これには、2018 年 4 月より前に承認された IHE 変更

提案に関連するすべての修正の統合が含まれます。サポートされる IHE プロファイルの完全なリストは、<http://www.intersystems.com/ihe> を参照してください。

- ・ リポジトリ OID の XDS.b レジストリ検証設定 – [ValidateRepositoryOID] という名前の新しいビジネス・オペレーション設定が IHE XDS.b Registry Operation コンポーネント (HS.HC.IHE.XDSb.Registry.Operations) に追加されました。このチェック・ボックスは、リポジトリ OID を検証する必要があるかどうかを示します。指定しない場合、ビジネス・オペレーションは、構成レジストリの “¥IHE¥XDSb¥Registry¥ValidateRepositoryOID” エントリからこの情報を取得する以前の動作に既定で設定されます。
- ・ XDS.b レジストリのキュー・パフォーマンスの向上 – 大規模なデータセットに対するレジストリのキュー・パフォーマンスを向上させるため、XDS.b レジストリの既存の実装に複数の更新が行われています。
- ・ XDS.b レジストリの Reference ID オプション – Health Connect のこのリリースでは、XDS.b レジストリの既存の実装に Reference ID オプションが追加され、Document Registry が Document Consumer からの FindDocumentsByReferenceId クエリに応答できるようになります。
- ・ DSUB Notification Broker の機能強化 – DSUB Notification Broker が強化されて SubmissionSets の購読がサポートされ、“ihe:SubmissionSetMetadata” を TopicExpression として使用できるようになりました。さらに、DocumentEntry 購読で Notification Broker の既存サポートが拡張され、\$XSDDocumentEntryReferenceIdList フィルタ・パラメータが含まれるようになりました。
- ・ XCPD の Shared/National Patient Identifier Query and Feed オプション – Health Connect のこのリリースでは、Shared/National Patient Identifier Query and Feed という新しい操作モードを追加することにより、IHE XCPD (Cross-Community Patient Discovery) の既存の実装が拡張されます。このモードでは、Cross Gateway Patient Discovery (ITI-55) クエリで、患者基本情報の代わりに国単位の患者識別子を使用できます。これは XCPD の Initiating Gateway と Responding Gateway の両方に適用可能です。構成を容易にするため、[NationalPatientIdentifier] という新しいホストレベルの設定 (チェック・ボックス) が HS.IHE.XCPD.InitiatingGateway.Process および HS.IHE.XCPD.RespondingGateway.Process に追加されました。さらに、国単位の患者識別子进行处理する際に使用する割り当て機関を指定するため、“¥IHE¥NationalPatientIdentifierAA” という新しい構成レジストリ・キーが追加されました。
- ・ PIXv3 Patient Identity Feed メッセージの新しいデータ・フィールド – この機能強化は、Patient Identity Feed HL7 V3 (ITI-44) メッセージに追加の基本情報データを含めることができるようにすることにより、PIXv3 の既存の実装を拡張します。このメッセージの対応する内部メッセージは HS.Message.AddUpdateHubRequest です。この変更は、PIX Manager および Patient Identity Source の両方に適用されます。より具体的には、追加のデータ・フィールド Birth Name (Family, Given, Prefix)、Aliases、MothersName、FathersName、および SpousesName がサポートされるようになりました。
- ・ IHE サービス用のファイルベースの WSDL – この更新により、Health Connect でサポートされるさまざまな IHE サービス用のファイルベースの WSDL が導入されます。新しい WSDL ファイル (HS.IHE.PIXv3.Manager.Services.CLS.wsdl、HS.IHE.XCA.InitiatingGateway.Services.CLS.wsdl など) は <Install Dir>/dev/wsdls にあり、これらの WSDL から参照される適用可能な XSD スキーマはすべて /wsdls/schema にローカルに保存されるようになりました。以前は、IHE サービス用の WSDL は個々の Web サービスから生成されて提供されており、XSD への参照は intersystems.com に外部的に存在していました。最近の製品レベルでのセキュリティ更新のため、特定のセキュリティ設定を変更せずに古いメカニズムで WSDL にアクセスすることが困難になりました。
- ・ XUA の機能強化 – この機能強化により、Health Connect で XUA を設定する際に、顧客は詳細に制御と構成を行うことができます。より具体的には、新しいクラス HS.HC.IHE.XUA.Creator および HS.HC.IHE.XUA.Processor が導入されました。これらのクラスは、ユーザ/臨床医レジストリが必要ないという点で、その前身とは異なります。XUA の以前の実装は、HealthShare のユーザ/臨床医レジストリに密接に結び付いていたため、別のユーザ検証方法をサポートするのは困難でした。

### 8.2.1.2 システムの相互運用性の機能強化

Health Connect 2019.1 には、プロダクションの構成とトラブルシューティングを迅速化する以下のシステムの相互運用性に関する新機能が含まれています。

- ・ インタフェース・マップ – ユーザは、プロダクション内でメッセージが取り得るすべてのルートを検索および表示できます。詳細は、“プロダクションの監視”の“インタフェース・マップの表示”を参照してください。
- ・ インタフェース参照の検索 – ユーザはプロダクション・コンポーネントが他のプロダクション・コンポーネントによって参照されている場所を検索できます。詳細は、“プロダクションの監視”の“インタフェース参照の検索”を参照してください。
- ・ データ変換テストの機能強化 – データ変換のテストで [変換のテスト] ダイアログへの未加工テキストの入力を可能にすることにより、データ変換エディタでレコード・マップのユニット・テストを実行できるようになりました。また、aux、コンテキスト、およびプロセス・システム・オブジェクトがインスタンス化された状態でデータ変換が呼び出された場合と同様に、これらのオブジェクトの値を入力できるようになりました。詳細は、“DTL 変換の開発”の“変換テスト・ページの使用法”を参照してください。
- ・ DTL エディタの機能強化 – データ変換エディタの使いやすさが switch/case アクションの追加、アクションのグループ化機能、グループの折りたたみ/展開機能、およびデータ変換へのコメントの追加機能により向上しています。詳細は、“DTL 変換の開発”の“Switch アクションの追加”、“アクションのグループの操作”、および“Comment アクションの追加”を参照してください。
- ・ ルーティング・ルール・ユニット・テスト – この機能強化により、ユニット・テスト機能がルール・エディタに導入されました。これにより、ユーザはビジネス・ルールを通じてメッセージを提供し、プロダクション全体を通じてメッセージを実行することなく、ルールの実行結果を表示できます。詳細は、“ビジネス・ルールの開発”の“ルーティング・ルール・テスト”を参照してください。
- ・ 複数のメッセージのローカル・コンピュータへのダウンロード – ユーザはメッセージ・ビューワでメッセージを複数選択し、それらをローカル・コンピュータにダウンロードできます。詳細は、“プロダクションの監視”の“メッセージのエクスポート”を参照してください。
- ・ イベント・ログのローカル・コンピュータへのダウンロード – ユーザは、イベント・ログをローカル・コンピュータにダウンロードできます。以前は、イベント・ログはサーバにしかダウンロードできませんでした。詳細は、“プロダクションの監視”の“イベント・ログ・ページの概要”を参照してください。
- ・ ルール・エディタの機能強化 – コメントをビジネス・ルールに追加する機能や、所定のデータ変換 (DTL) がビジネス・ルールで使用されている場合にルール・エディタから直接 DTL を表示および編集する機能により、ルール・エディタの使いやすさが向上しています。ルール・エディタから DTL を開く方法の詳細は、“ビジネス・ルールの開発”の“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。
- ・ [キュー待機警告] の変更 – [キュー待機警告] の設定では、アラートがトリガされるまでにメッセージがビジネス・ホストのキュー内に留まることができる、またはアクティブなメッセージであることができる時間を指定できるようになりました。以前は、この設定はキュー内のメッセージのみに適用され、アクティブなメッセージには適用されませんでした。詳細は、“プロダクションの構成”の“キュー待機警告”の設定を参照してください。
- ・ システムのデフォルト設定へのアクセスの制限 – 管理者は、ユーザがシステムのデフォルト設定を作成、編集、または削除できるようにするかどうかを制御できます。詳細は、“プロダクションの管理”の“システムのデフォルト設定のセキュリティ”を参照してください。
- ・ プロダクションのローカル・コンピュータへのエクスポート – ユーザはプロダクションをローカル・コンピュータにエクスポートできます。以前は、プロダクションはサーバにしかエクスポートできませんでした。詳細は、“プロダクションの構成”の“プロダクションのエクスポート”を参照してください。
- ・ ローカル・コンピュータからのプロダクションの導入 – ユーザはローカル・コンピュータからプロダクションを導入できます。以前は、プロダクションはサーバからしか導入できませんでした。詳細は、“プロダクションの開発”の“ターゲット・システムでのプロダクションの配置”を参照してください。
- ・ [プロダクション構成] ページからのナビゲーションの強化 – 関連項目を別のウィンドウですばやく開くために、[プロダクション構成] ウィンドウのタブにリンクが追加されました。[キュー] タブで、メッセージ ID をクリックすると、メッセージのビジュアル・トレースを表示するウィンドウが開きます。[メッセージ] タブで、セッション ID をクリックすると、メッセージのビジュアル・トレースを表示するウィンドウが開きます。[ジョブ] タブで、メッセージ ID をクリックすると、メッセージのビジュアル・トレースを表示するウィンドウが開きます。また、ジョブ ID をクリックすると、ジョブのプロセス詳細を表示するウィンドウが開きます。



- ・ ビジネス・ホスト・ウィザードの機能強化 – ユーザの生産性を強化するために、ビジネス・ホストの作成に使用されるウィザードにオプションが追加されました。ユーザは、ビジネス・ホスト・ウィザードを使用して、フィールドが空白のときに、システム既定値を自動的に割り当てることができます。また、自動生成されたルーティング・ルールにパッケージ接頭語を定義できます。ビジネス・ホスト・ウィザードの新たなオプションの詳細は、“プロダクションの構成”の“ウィザード・オプション”を参照してください。
- ・ データベースのカスタムの場所のサポート – この機能強化により、ユーザは Foundation プロダクションをインストールする際に、データベースのカスタムの場所を指定できるようになります。以前は、<Install Dir>/mgr にある既定の場所とは異なるデータベースの場所を指定する場合、正式にサポートされている方法はありませんでした。管理ポータル [インストーラ・ウィザード] に [データベースの代替場所] という新しいフィールドが追加されました。このオプションのフィールドは、新しい Foundation プロダクションの作成中のみユーザに表示されます。さらに、代替場所をサポートするため、対応する変更がインストール・スクリプトに加えられ、“DBInstallDirectory”という新しい変数が定義されています。この機能強化では、既存のプロダクションのデータベースの場所の移動はサポートされていないことに注意してください。

## 8.2.2 クラスの削除と非推奨

以下のクラスは Health Connect 2019.1 で削除されたか、非推奨になりました。

Health Connect 15.03 のクラス	Health Connect 2019.1 でのステータス
HS.Audit.ConsolidationServices	削除。代わりに HS.HC.Audit.ConsolidationServices を使用してください。
HS.FHIR.MHD.*	非推奨。代わりに HS.FHIR.vDSTU2.MHD.* または HS.FHIR.vSTU3.MHD.* を使用してください。
HS.FHIR.PDQm.*	非推奨。代わりに HS.FHIR.vDSTU2.PDQm.* または HS.FHIR.vSTU3.PDQm.* を使用してください。
HS.FHIR.PIXm.*	非推奨。代わりに HS.FHIR.vDSTU2.PIXm.* または HS.FHIR.vSTU3.PIXm.* を使用してください。
HS.FHIR.Model.*	非推奨。代わりに HS.FHIR.vDSTU2.Model.* または HS.FHIR.vSTU3.Model.* を使用してください。
HS.FHIR.Utills.DateConversion	非推奨。
HS.FHIR.Utills.SearchTableBuilder	削除。代わりに HS.FHIR.vDSTU2.SearchTableBuilder または HS.FHIR.vSTU3.SearchTableBuilder を使用してください。
HS.FHIR.REST.Handler	非推奨。HS.FHIR.vDSTU2.REST.Handler または HS.FHIR.vSTU3.REST.Handler を使用してください。
HS.FHIR.Operation.Process	削除。このクラスをカスタマイズしている場合は、カスタマイズを HS.FHIR.vDSTU2.Repository.OperationProcessor または HS.FHIR.vSTU3.Repository.OperationProcessor に移動してください。
HS.FHIR.Gateway.*	削除。
HS.FHIR.Repository.*	HS.FHIR.Repository.Operations を除くすべてのクラスが削除されました。HS.FHIR.Repository.Operations は HS.FHIR.vDSTU2.Repository.Operations の拡張です。

Health Connect 15.03 のクラス	Health Connect 2019.1 でのステータス
HS.IHE.DSUB.Publisher.Process	削除。代わりに HS.HC.IHE.DSUB.Publisher.Process を使用してください。
HS.IHE.XDR.Recipient.CommonProcess	削除。代わりに HS.HC.IHE.XDR.Recipient.CommonProcess を使用してください。
HS.IHE.XDSb.Consumer.Operations	削除。代わりに HS.HC.IHE.XDSb.Consumer.Operations を使用してください。
HS.IHE.XDSb.Registry.Operations	削除。代わりに HS.HC.IHE.XDSb.Registry.Operations を使用してください。
HS.Message.ECRUpdateRequest	削除。以前に HS.Message.ECRUpdateRequest を HS.FHIR.FromSDA.DTL.Transaction.Process の入力メッセージ・タイプとして使用していた場合、代わりに HS.Message.XMLMessage または Ens.StreamContainer を使用するように入力を変更してください。
HS.MPI.Manager	削除。代わりに HS.HC.MPI.Manager を使用してください。
HS.Test.UI.FHIR.LoggedOut	削除。
HS.Test.UI.FHIR.ServerSelect	削除。
HS.Test.UI.FHIR.ServerSelectOAuth2	削除。
HS.UI.AssigningAuthorities	削除。代わりに HS.HC.UI.AssigningAuthorities を使用してください。
HS.UI.Home	削除。代わりに HS.HC.UI.Home を使用してください。
HS.Util.Installer.Kit.FHIR.*	削除。代わりに HS.HC.Util.Installer.Kit.FHIR.* を使用してください。
HS.UI.Installer.Welcome	削除。代わりに HS.HC.UI.Installer.Welcome を使用してください。
HS.Util.Trace.Helper	削除。HS.HC.Util.Trace.Helper を使用してください。

## 8.2.3 メソッドの置換

Health Connect のこのリリースでは、旧リリースにおける以下の FHIR 関連のインストール・メソッドが削除および置換されています。

削除されたメソッド	新しい置換メソッド
HS.Util.Installer.InstallFHIRServer()	HS.HC.Util.Installer.FHIR.Install()
HS.Util.Installer.InstallOAuth2()	HS.HC.Util.Installer.OAuth2.ConfigureSampleOAuth()

## 8.2.4 その他の非推奨

HealthShare Core 13 で初めて導入された QRDA (Quality Reporting Document Architecture) のサポートはこのリリースで非推奨になり、この機能に関連するコードと XSLT は Health Connect の次期メジャー・リリースですべて削除される予定です。

## 8.3 Health Connect 2019.1 の修正

このバージョンの重要な修正には以下が含まれます。

- ・ FHIR DSTU2 および STU3 の並べ替え
- ・ 不明なリソース・タイプに対する HTTP 応答
- ・ MHD Find Document Manifest 要求の正しい URL
- ・ XDS.b メッセージの新しい最大 ID 長
- ・ AA コード解決時の識別子タイプの除外
- ・ DSUB トランザクションのタイム・ゾーン・オフセット
- ・ BirthTime に関する C-CDA v2.1 エクスポートの修正
- ・ C-CDA v2.1 インポートでの名前修飾子 BR の Birth への変換

“Health Connect Maintenance Release Changes (2019.1.4)” も参照してください。

### 8.3.1 FHIR DSTU2 および STU3 の並べ替え

Health Connect のこのリリースでは、FHIR サーバの以前の実装で見つかったリソースの並べ替えの問題が修正され、FHIR DSTU2 と STU3 の両方に対して “\_sort” 検索パラメータが正しく実装されます。FHIR STU3 では、検索 URL の \_sort パラメータの動作がいくつか変更されます。DSTU2 では、\_sort パラメータを繰り返すことで複数の \_sort 値を送信できました。STU3 では、\_sort パラメータは 1 つの要求につき 1 つだけにする必要があり、値は検索パラメータのコンマ区切りリストになります。さらに DSTU2 では、各 \_sort パラメータに修飾子 “asc” または “desc” を付けて並べ替え順序を指定でき、修飾子を付けなければ “asc” を指定したときと同じ処理になっていました。STU3 では、\_sort パラメータに修飾子はありません。\_sort リストの値の先頭に “-” を付けると降順を示し、付けると昇順が想定されます。

### 8.3.2 不明なリソース・タイプに対する HTTP 応答

不明なリソース・タイプに対する要求への応答方法について FHIR DSTU2 および STU3 の仕様により厳格に従うよう、FHIR サーバ API の動作が修正されました。FHIR のさまざまな対話処理のために、以下の応答動作が実装されました。

- ・ read : HTTP ステータス「404 未検出」と OperationOutcome
- ・ vread : HTTP ステータス「404 未検出」と OperationOutcome
- ・ update : HTTP ステータス「404 未検出」のみ
- ・ delete : HTTP ステータス「204 コンテンツなし」
- ・ history : HTTP ステータス「404 未検出」のみ
- ・ create : HTTP ステータス「404 未検出」のみ
- ・ search : HTTP ステータス「404 未検出」と OperationOutcome

### 8.3.3 MHD Find Document Manifest 要求の正しい URL

MHD Document Responder の以前の実装では、Find Document Manifest 要求への応答で fullURL 要素に相対 URL の値が入力されていました。この問題は Find Document Manifest でのみ確認されており、これが修正されました。



### 8.3.4 XDS.b メッセージの新しい最大 ID 長

以前は、Health Connect では XDS.b メッセージの ID 値が内部的に 73 文字に制限されていました。このため、例えば非常に長い ID を持つ XDS.b Provide and Register 要求を受信した場合、エラーが発生する可能性があります。IHE では長さの制限は直接は仕様化されていないため、ebRIM の仕様に基づいて、この ID 長制限が 256 文字に増やされました。

### 8.3.5 AA コード解決時の識別子タイプの除外

[ExcludeIdentifierTypes] という新しいホストレベルの設定 (テキスト・ボックス) が PIX および PIXv3 の Consumer プロセスと Manager プロセス (HS.IHE.PIX.Consumer.Process、HS.IHE.PIX.Manager.Process、HS.IHE.PIXv3.Consumer.Process、および HS.IHE.PIXv3.Manager.Process) に追加されました。この新しい設定は、PIX Consumer または Manager が HS\_AssignAuth.Config テーブルに対してクエリを実行して、指定された AA コードの割り当て機関 (AA) 識別子タイプを取得する場合に除外する識別子タイプを指定します。以前は、AA コードに関連付けられている AA 識別子タイプは 1 つのみであることが想定されていました。この新しい設定によって AA コードの解決時に考慮する識別子タイプを正確に制御することで、この間違っただけの想定が解決されています。DL (Driver License) と DN (Doctor Number) は既定で除外されます。

### 8.3.6 DSUB トランザクションのタイム・ゾーン・オフセット

DSUB トランザクションの購読時刻は UTC またはオフセットであることが予期されていますが、DSUB の以前の実装では、時刻は常にローカルであると誤って想定されており、指定したタイム・ゾーン・オフセットがすべて無視されていました。これが修正されました。購読で UTC と UTC +/- の両方を適切に処理できるようになり、内部処理では常に UTC が使用されます。

### 8.3.7 BirthTime に関する C-CDA v2.1 エクスポートの修正

SDA から C-CDA v2.1 のエクスポートが、playingEntity ノード下に記述されている予期しない stdc:birthTime 要素を削除するよう修正されました。以前は、この無関係の要素のためにスキーマ検証の問題が発生していました。

### 8.3.8 C-CDA v2.1 インポートでの名前修飾子 BR の Birth への変換

以前は、C-CDA v2.1 ドキュメントを SDA にインポートする際に "BR" を "Birth" に変換するためのロジックが、ベースとなる C-CDA v2.1 インポート変換にありませんでした。"BR" は、本姓を表すために CDA で使用される名前修飾子コードで、HealthShare では内部的に "Birth" として表されます。この問題が修正されました。

## 8.4 Health Connect 15.03 から Health Connect 2019.1 へのアップグレード

Health Connect 2019.1 は、InterSystems IRIS を利用する Health Connect の最初のリリースです。基盤のテクノロジーがこのように変更されているため、旧バージョンの Health Connect からアップグレードする場合は特別なアップグレード手順が必要です。Health Connect 2019.1 へのアップグレードを検討している場合は、[InterSystems WRC Documents](#) で入手できる "InterSystems IRIS インプレース変換ガイド" を参照してください。



# 9

## このリリースへのアップグレード

このセクションでは、このメジャー・リリースへのアップグレードに適用する具体的な手順について説明します。これらの手順は、“インストール・ガイド”の“以前のバージョンからのアップグレード”への補足情報です。

前回のメジャー・リリースよりも前のバージョンからこのリリースへアップグレードする場合は、[前回のメジャー・リリースのアップグレード手順](#)のほか、これまでのアップグレードでインストールしていない他のすべてのメジャー・リリースのアップグレード手順も参照する必要があります。

Health Connect 15.03 からアップグレードする場合は、特別なアップグレード手順が必要です。詳細は、[InterSystems WRC Documents](#) ページで入手できる“InterSystems IRIS インプレース変換ガイド”を参照してください。

### 9.1 コンテナのアップグレード

コンテナ化されたアプリケーションはホスト環境から分離されているので、永続データを書き込みません。コンテナが削除され、新しいコンテナに置き換えられるときに、アプリケーションがコンテナ内で書き込んだ内容が失われるためです。このため、コンテナ化されたアプリケーションの導入で重要な点は、コンテナの外部にデータを保管し、他のコンテナや後続のコンテナがそのデータを使用できるようにすることです。

永続的な %SYS 機能を使用すると、インスタンス固有のデータ（ユーザ定義、監査レコード、ログ・ファイル、ジャーナル・ファイル、WIJ ファイルなど）の永続ストレージが可能になり、コンテナ内での Health Connect の実行時に、単一インスタンスを時間の経過と共に複数のコンテナで順番に実行できるようになります。例えば、永続的な %SYS を使用して Health Connect コンテナを実行すると、元のコンテナを停止して、元のコンテナで作成されたインスタンス固有のデータを使用する新しいコンテナを実行することによって、インスタンスをアップグレードすることができます。アップグレードの詳細は、“InterSystems IRIS コンテナのアップグレード”を参照してください。永続的な %SYS の詳細は、“永続インスタンス・データを保存するための永続的な %SYS”を参照してください。

**重要** このリリースでは、ディストリビューション・コンテナに非 root の既定ユーザがあります。これにより、コンテナのセキュリティが向上します。2019.2 以前のインスタンスの永続的な %SYS を使用している場合、Health Connect 2022.1 を実行する前に、ホストの永続ディレクトリにあるいくつかのファイルの所有権を変更する必要があります。ファイル所有権の変更に関する手順は、担当のインターシステムズ・セールス・エンジニアまたは[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。このような変更を行わない場合、Health Connect の起動時にコンテナでエラーが発生します。



# 10

## 非推奨の機能とサポート中止の機能

### 10.1 サポート中止のテクノロジーと機能

インターシステムズでは、より新しいオプションやより優れたオプションが利用可能になった場合、テクノロジーの開発を停止します。ただし、これらの機能に対する製品サポートは、[最小サポート・バージョン](#)以上の製品に対するサポートと同様に続行されます。

非推奨とは、インターシステムズが積極的に開発しなくなった、より優れたオプションが存在する機能またはテクノロジーを示します。新規開発では、非推奨のアイテムは使用しないでください。非推奨の指定は、顧客が機能またはテクノロジーの使用を排除するよう計画すべきことを示します。インターシステムズは、非推奨の製品機能をサポートするためのスタッフの専門知識を維持しています。例として、Zen や Zen Reports などがあります。

サポート中止とは、既存のアプリケーションのものであっても、もはや使用できなくなった機能またはテクノロジーを示します。インターシステムズは、このようなテクノロジーを使用し続けることは、顧客にとってリスクであると考えます。この理由の一部を以下に示します。

- ・ 使用が減少し、顧客が少数になった。
- ・ 機能が、最新のテクノロジーまたはセキュリティ手法と互換しなくなった。
- ・ 機能またはテクノロジーと最新の製品実装との間に互換性がなくなったことで、アプリケーションの保守に多大なコストがかかる。
- ・ 機能またはテクノロジーが、サードパーティのサポート中止コンテンツに依存している。

例として、DCP (ECP に取って代わられた分散キャッシュ・プロトコル)、WebLink および Caché Direct (Visual M/VISM) があります。

### 10.2 非推奨になった Spark コネクタ

このリリースから、Spark コネクタが削除されています。Spark には JDBC コネクタが組み込まれました。Spark コネクタを使用する場合は、Spark JDBC コネクタを使用するようにコードを変更する必要があります。

