



# InterSystems Supply Chain Orchestrator の概要

Version 2024.1  
2024-06-03

## InterSystems Supply Chain Orchestrator の概要

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

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](mailto:support@InterSystems.com)

# 目次

|  |    |
|--|----|
| 1 InterSystems Supply Chain Orchestrator の概要 | 1  |
| 1.1 Supply Chain Orchestrator の目的            | 2  |
| 1.2 機能                                       | 3  |
| 1.3 アーキテクチャの概要                               | 3  |
| 1.4 利用可能なオンライン学習                             | 5  |
| 2 インストールとアップグレード                             | 7  |
| 2.1 新規インストール                                 | 7  |
| 2.1.1 InterSystems IRIS のデプロイ                | 7  |
| 2.1.2 Supply Chain Framework の追加             | 7  |
| 2.1.3 アプリケーションのデプロイ                          | 8  |
| 2.2 InterSystems Supply Chain Framework の内容  | 9  |
| 2.3 アップグレード                                  | 9  |
| 2.4 ドキュメント・リンクの再構成                           | 10 |
| 2.5 関連項目                                     | 10 |
| 3 サプライ・チェーン・ソリューションの概要                       | 11 |
| 3.1 前提条件                                     | 11 |
| 3.2 実装の基本要件                                  | 11 |
| 3.3 プロダクションの要件                               | 12 |
| 3.4 安全なカスタマイズのためのガイドライン                      | 13 |
| 3.5 関連項目                                     | 13 |
| 4 問題の管理                                      | 15 |
| 4.1 問題分析の例                                   | 15 |
| 4.2 問題のライフ・サイクル                              | 16 |
| 4.3 KPI からの問題の生成                             | 17 |
| 4.4 問題管理 API の概要                             | 18 |
| 4.5 関連項目                                     | 18 |
| 5 サプライ・チェーン・データ・モデル                          | 21 |
| 5.1 サプライ・チェーン・データ・オブジェクト                     | 21 |
| 5.1.1 サプライ・チェーンのマスタおよび参照データ                  | 21 |
| 5.1.2 アウトバウンドおよび製造データ                        | 21 |
| 5.1.3 インバウンドのサプライ・チェーン・データ                   | 22 |
| 5.2 データ・モデル・クラスの概要                           | 22 |
| 5.3 データ・モデルのカスタマイズ                           | 22 |
| 5.3.1 コーディングによるカスタマイズ                        | 22 |
| 5.3.2 API によるカスタマイズ                          | 22 |
| 5.4 関連項目                                     | 23 |
| 6 サプライ・チェーン・データ・モデル API                      | 25 |
| 6.1 サプライ・チェーン・データ・オブジェクトのリスト                 | 25 |
| 6.2 オブジェクト定義の取得                              | 26 |
| 6.3 カスタム属性の追加                                | 26 |
| 6.4 関連項目                                     | 27 |
| 7 データ API の概要                                | 29 |
| 7.1 API の URL パターン                           | 29 |
| 7.2 検索 API                                   | 30 |

|   |    |
|---|----|
| 7.3 結果の並べ替え .....   | 31 |
| 7.4 既定の並べ替え .....   | 31 |
| 7.5 結果のページ付け .....  | 32 |
| 7.6 日付および日付/時刻の形式 .....                                   | 32 |
| 7.7 関連項目 .....  | 33 |
| 8 InterSystems Supply Chain Orchestrator の分析キューブ .....    | 35 |
| 8.1 利用可能なデータ .....  | 35 |
| 8.1.1 注文に関する洞察 .....                                      | 35 |
| 8.1.2 出荷に関する洞察 .....                                      | 36 |
| 8.1.3 在庫に関する洞察 .....                                      | 36 |
| 8.2 利用可能なキューブ .....                                       | 36 |
| 8.3 キューブの使用開始 .....                                       | 37 |
| 8.4 関連項目 .....  | 37 |
| 9 Supply Chain Orchestrator 向け BPL チュートリアル .....          | 39 |
| 9.1 一般的な構造 .....  | 39 |
| 9.2 コンテキストの初期化 .....                                      | 40 |
| 9.3 分析値の指定 .....  | 42 |
| 9.4 分析のプロパティ .....  | 42 |
| 9.5 分析結果の問題への保存 .....                                     | 43 |
| 9.6 ビジネス・プロセスのテスト .....                                   | 43 |
| 9.7 関連項目 .....  | 44 |
| 10 ワークフロー・シナリオの追加 .....                                   | 47 |
| 10.1 概要 .....   | 47 |
| 10.2 シナリオを作成する手順 .....                                    | 48 |
| 10.3 シナリオのプロパティ .....                                     | 50 |
| 10.4 関連項目 .....   | 51 |
| 11 InterSystems Supply Chain Orchestrator の KPI の構成 ..... | 53 |
| 11.1 KPI の目的 .....  | 53 |
| 11.2 構成プロセス .....   | 54 |
| 11.3 KPI ビジネス・ロジック .....                                  | 54 |
| 11.4 関連項目 .....   | 55 |
| 12 KPI の構成例 .....   | 57 |
| 12.1 要件の収集 .....  | 57 |
| 12.2 要件マッピング .....  | 57 |
| 12.3 最終結果 .....   | 60 |
| 12.4 関連項目 .....   | 61 |
| 13 KPI API の概要 .....                                      | 63 |
| 13.1 KPI の定義 .....  | 63 |
| 13.2 KPI 値 .....  | 65 |
| 13.3 KPI の詳細リスト .....                                     | 65 |
| 13.4 KPI のフィルタリング .....                                   | 66 |
| 13.5 関連項目 .....   | 66 |
| 14 API リファレンス .....                                       | 67 |
| 14.1 利用可能な API .....                                      | 67 |
| 14.2 関連する API .....                                       | 67 |
| 14.3 関連項目 .....   | 67 |

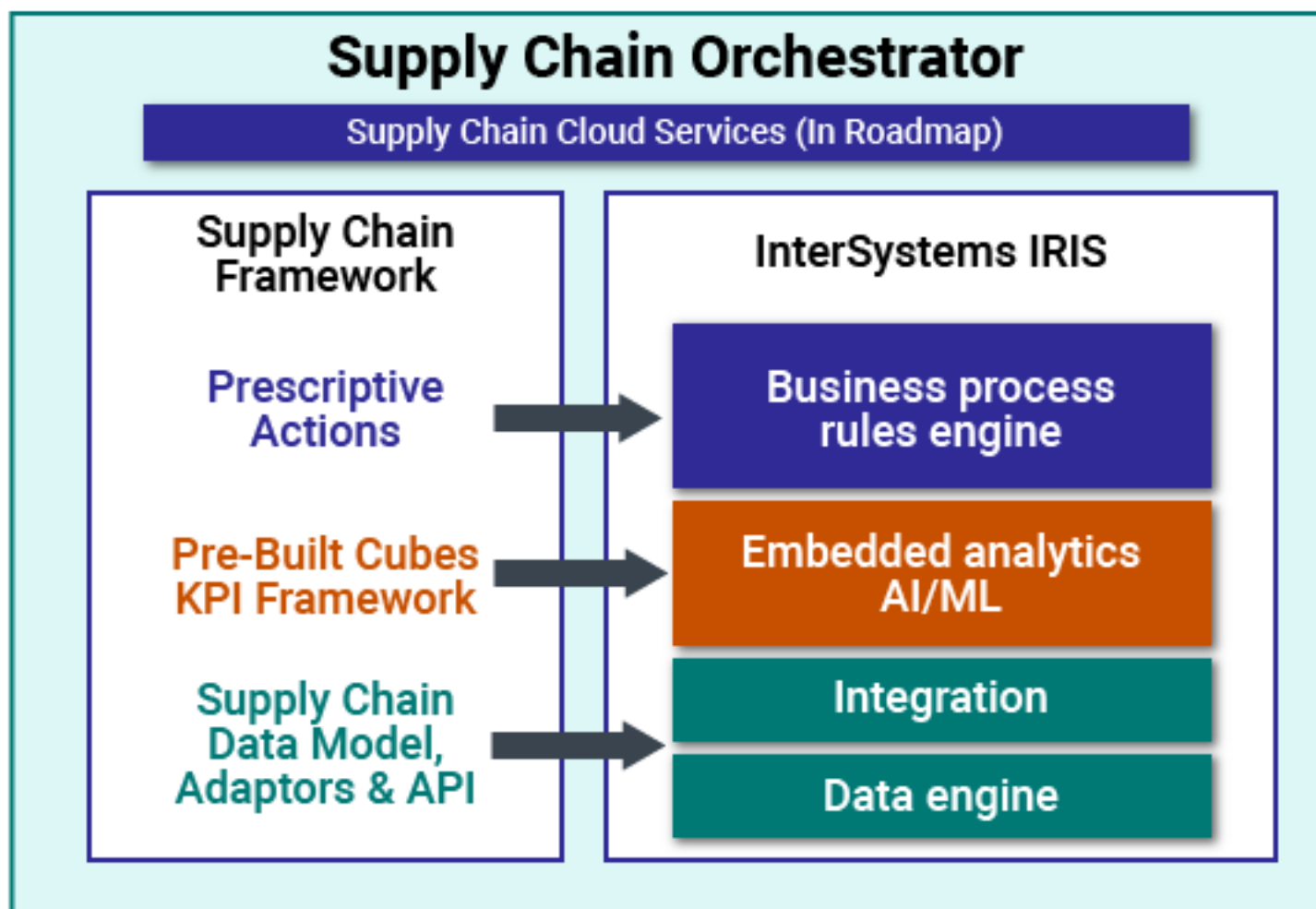
# 1

## InterSystems Supply Chain Orchestrator の概要

InterSystems Supply Chain Orchestrator™ は、インテリジェントなデータ・プラットフォームをサプライ・チェーン固有の一連のクラウド・サービスと共に提供し、サプライ・チェーンのパフォーマンスと俊敏性を変革する、エンドツーエンドの可視性と比類のない予測的・規範的な機能を実現します。

Supply Chain Orchestrator の中核を成す部分は InterSystems IRIS for Supply Chain と呼ばれ、InterSystems IRIS® と、別途リリースされる InterSystems Supply Chain Framework で構成されます。以下の図に示すように、Supply Chain Framework は InterSystems IRIS の主要な部分を拡張します。

この中核部分に加えて、一連のクラウド・サービスが予定されています。お客様やパートナーは、このエコシステムと相互運用する独自のクラウド・サービスを作成し、販売することができます。



## 1.1 Supply Chain Orchestrator の目的

Supply Chain Orchestrator (特にその中核部分である IRIS for Supply Chain) は、以下の目的を満たすよう設計されています。

- ・ サプライ・チェーン全体の KPI や問題管理など、ほとんどのサプライ・チェーンの問題に必要な基本的なインフラストラクチャを提供する。
- ・ サプライ・チェーン・データ・モデルや関連する分析キューブなど、一般的なサプライ・チェーン機能を提供する。
- ・ サプライ・チェーン機能にアクセスするための一連の API を提供する。これは、UI 開発などで外部からも容易に利用することができます。
- ・ フレームワークの導入を容易にする一連のユーティリティを提供する。例えば、ビジネス・プロセス設計を簡略化してノーコード/ローコード開発を実現するためのメッセージング機能やヘルパー・クラスが用意されています。

Supply Chain Orchestrator は、以下のとおりではありません。

- ・ TMS、WMS、OMS などの既存のサプライ・チェーン運用アプリケーションを置き換えるサプライ・チェーン・ソリューション。ただし、本製品は、これらのシステムからのデータやプロセスの接続を容易にし、データの可視性のギャップを埋め、サプライ・チェーン全体に対するリスクの影響を最小限に抑えるためのインテリジェントなアクションを提供します。

- ・ 任意のユース・ケースのためのビジネス・プロセスやビジネス・ルール、またはサプライ・チェーンの任意の部分の KPI など、特定のユース・ケース向けに事前定義されたソリューション・セット。本製品は、最小限のコーディング作業またはコーディング作業なしで、独自のユース・ケース固有のソリューションを定義するためのシンプルなメカニズムを提供します。
- ・ サプライ・チェーンの実務担当者が直接操作できるエンドユーザ・アプリケーション。本フレームワークのソリューションや成果は、コントロール・タワーの UI、BI ツール、その他のサプライ・チェーン・アプリケーションなど、他のアプリケーションに公開されることが期待されます。

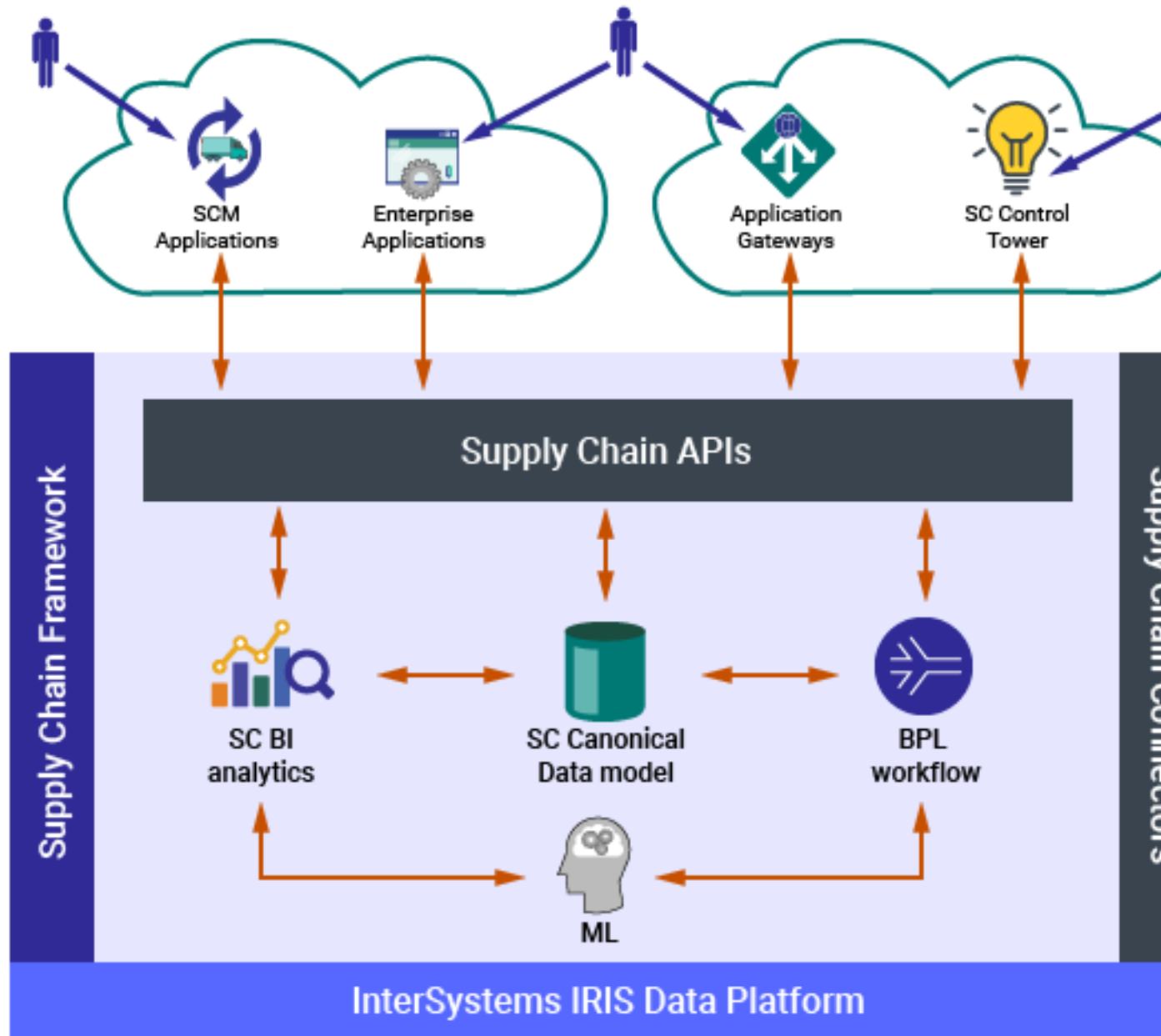
## 1.2 機能

Supply Chain Orchestrator (特にその中核部分である IRIS for Supply Chain) は、以下の機能を提供します。

- ・ [拡張可能なサプライ・チェーン・データ・モデル](#)。InterSystems IRIS データ・プラットフォームに実装されるキャノニク形式のサプライ・チェーン・データ・モデルで、ObjectScript コードまたは API を使用して拡張/カスタマイズできます。
- ・ データ・モデルの検出とライブ・ドキュメンテーションのための [データ・モデル API](#)。例えば、すべてのサプライ・チェーン・データ・オブジェクトをリストしたり、サプライ・チェーン・データ・オブジェクトの詳細な定義を取得することができます。
- ・ [データ・アクセス API](#)。CRUD 操作と、並べ替えやページ付けに対応した高度な検索機能の両方をサポートします。
- ・ [分析キューブ](#)。サプライ・チェーン・データ・モデルに基づいて事前構築された Business Intelligence キューブです。カスタム属性を含めるようにするなど、組み込みのキューブは必要に応じて拡張できます。
- ・ [KPI フレームワークと関連 API](#)。KPI 管理 (KPI 構成とライフ・サイクル管理) および KPI 値とリストへのアクセスをサポートします。現在の KPI フレームワークは事前定義された [分析キューブ](#)に基づいています。
- ・ [問題のライフ・サイクル管理](#)。問題の特定、問題の分析、およびアクションの推奨を担当するコンポーネントです。

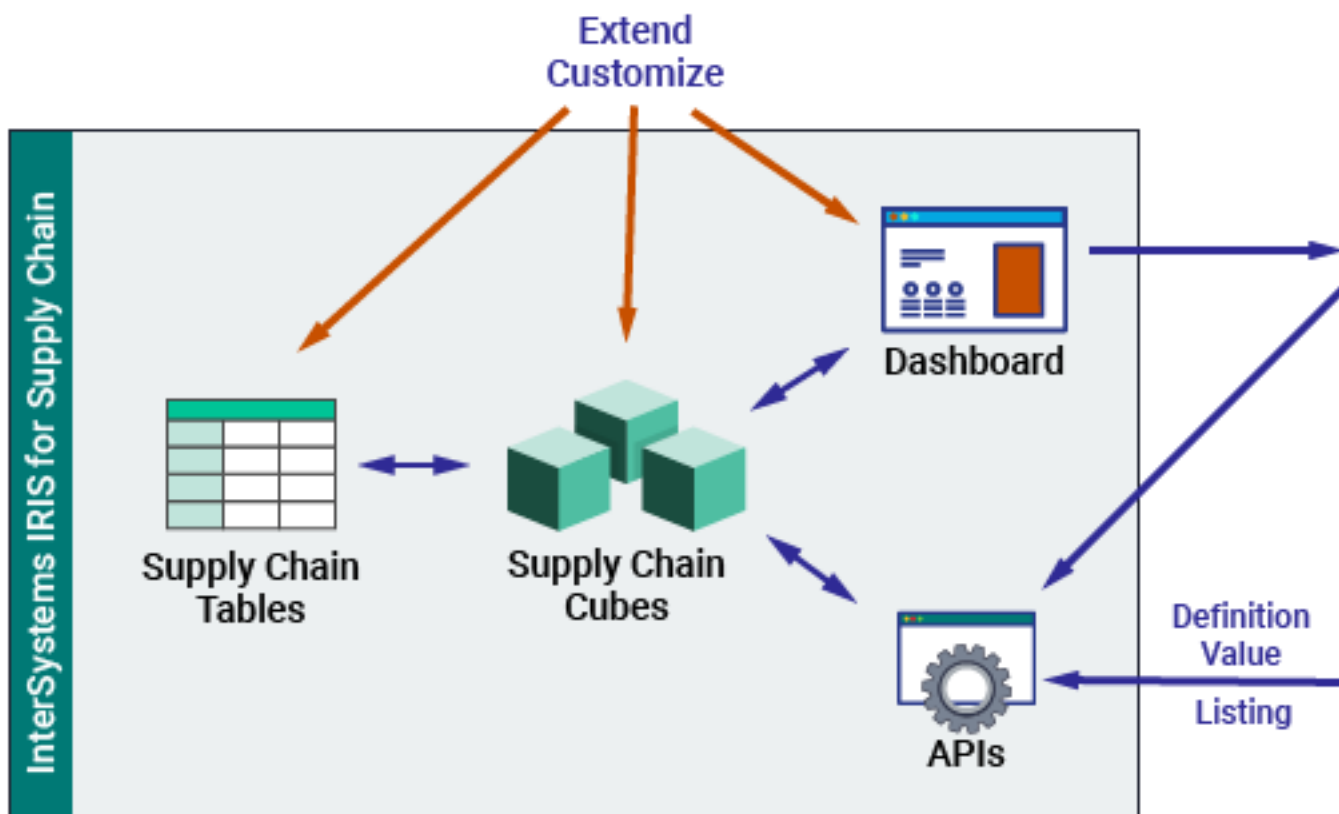
## 1.3 アーキテクチャの概要

上記のように、Supply Chain Orchestrator の中核となるのは、InterSystems IRIS for Supply Chain (InterSystems IRIS® サーバに InterSystems Supply Chain Framework を追加したもの) です。以下の図にその詳細を示します。



以下の図は、分析のサポートをより詳細に示しています。





## 1.4 利用可能なオンライン学習

ドキュメント Web サイトに加えて、オンライン学習パス "[Building Smart Real-Time Supply Chain Applications](#)" を参照してください。ここでは、このプラットフォームのユース・ケースやアーキテクチャを調べたり、それをインストールして実装を開始する方法を学習することができます。また、KPI を作成して、問題解決のためのビジネス・ロジックを使用する方法についても学ぶことができます。



# 2

## インストールとアップグレード

InterSystems Supply Chain Orchestrator™ 上に構築されたサプライ・チェーン・アプリケーションまたはソリューションは、以下のコンポーネントで構成されます。

- ・ コア InterSystems IRIS® データ・プラットフォーム
- ・ InterSystems Package Manager (IPM) を介して提供される InterSystems Supply Chain Framework 機能  
これは InterSystems IRIS と共に、Supply Chain Orchestrator の中核部分を形成します。
- ・ アプリケーションまたはソリューションの実装コード

インストールとアップグレードのプロセスは、インストールまたは更新されるコンポーネントによって異なります。このページでは、さまざまなシナリオに対して詳細な手順を説明します。

オンライン学習パス “[Building Smart Real-Time Supply Chain Applications](#)” も参照してください。

### 2.1 新規インストール

新規インスタンスをデプロイするには、以下の 3 つのステップを実行します。

1. InterSystems IRIS データ・プラットフォームをデプロイします。
2. InterSystems Supply Chain Framework をデプロイし、Supply Chain Orchestrator の中核部分を作成します。
3. アプリケーション・コードをデプロイします。

#### 2.1.1 InterSystems IRIS のデプロイ

InterSystems IRIS をデプロイするには、コンテナ内、Kubernetes 上、インストール・キットを使用するなど、さまざまな方法があります。それぞれの方法の詳細は、“[導入](#)” を参照してください。

#### 2.1.2 Supply Chain Framework の追加

InterSystems IRIS サーバをデプロイしたら、以下の手順を実行します。

1. Supply Chain Orchestrator で使用するよう予約される新しいネームスペースを作成します。このドキュメントでは、ネームスペース名として **SC** を使用していますが、別の名前を使用することもできます。

注釈     ネームスペースが相互運用対応かつ[分析対応](#)であることを確認してください。分析対応にすることで、Business Intelligence アナライザが使用可能になるため、キューブを探索できます。

- ターミナル・アプリケーションを起動し、ターミナルで残りのステップを完了します。  
Unix® の場合は、ターミナル・アプリケーションが利用できないため、ObjectScript シェルを起動します。
- 新たに作成したネームスペースに移動します。
- IPM がインストールされているかどうかを確認します。そのためには、コマンド `zpm` を入力します。IPM がインストールされていない場合、このコマンドはエラーになります。IPM がインストールされている場合、ターミナル・プロンプトは以下のようにになります。

#### Terminal

```
zpm:SC>
```

ここで、コロンの後の部分が現在作業中のネームスペースです。

- InterSystems Package Manager (IPM) がインストールされていない場合、以下のようにインストールします。

```
set r = ##class(%Net.HttpRequest).%New()
set r.Server="pm.community.intersystems.com",
set r.SSLConfiguration="ISC.FeatureTracker.SSL.Config"
do r.Get("/packages/zpm/latest/installer"),
do $system.OBJ.LoadStream(r.HttpResponse.Data,"c")
```

Tip ヒン 便利のように、前述のすべての行は次の 1 行で表されます。

```
ト
set r =
##class(%Net.HttpRequest).%New(),r.Server="pm.community.intersystems.com",r.SSLConfiguration="ISC.FeatureTracker.SSL.Config"
d r.Get("/packages/zpm/latest/installer"),$system.OBJ.LoadStream(r.HttpResponse.Data,"c")
```

その後、IPM を起動します。そのためには、コマンド `zpm` を入力します。

- まだ IPM トークンがない場合は、InterSystems アカウントの資格情報を使用して [Pm.InterSystems.com](https://pm.intersystems.com) にログインし、トークンを入手します。
- IPM のプロンプトで、以下のコマンドを使用して IPM リポジトリにログインします。

```
repo -n registry -r -url https://pm.intersystems.com/ -token YOUR_IPM_TOKEN
```

YOUR\_IPM\_TOKEN は前の手順で取得した IPM トークンです。

- 引き続き IPM プロンプトで、InterSystems Supply Chain Framework をインストールします。最新のフレームワークをインストールするには、以下のようになります。

```
install isc-supply-chain
```

または、以下を実行すると、特定のバージョンのフレームワーク (この例ではバージョン 1.0.0) をインストールできます。

```
install isc-supply-chain 1.0.0
```

- インストールが完了したら、`q` を入力して、IPM プロンプトを終了します。

## 2.1.3 アプリケーションのデプロイ

アプリケーション・コードは、DevOps ツールを使用してデプロイできます。アプリケーション・コードには以下が含まれる場合があります (以下に限定されるわけではありません)。

- サプライ・チェーンのデータ・モデルの拡張
- カスタムの分析キューブの設計

- ・ ビジネス・プロセス
- ・ 任意の DTL 実装
- ・ カスタム API
- ・ その他

## 2.2 InterSystems Supply Chain Framework の内容

InterSystems Supply Chain Framework をネームスペースにインストールすると、そのネームスペースに以下の新しいパッケージが表示されます。

- ・ **SC** パッケージ。
  - **SC.Data** サブパッケージには、変更可能なデータ・モデル・クラスが含まれます。
  - **SC.Core.BP** サブパッケージには、問題を管理するためのビジネス・プロセスを作成する際に参照するクラスが含まれます。
  - **SC.Core.Tasks** サブパッケージには、タスク・マネージャでスケジュールされるタスクが含まれます。

このパッケージ内のその他のコードは内部でのみ使用されます。

- ・ **datamodelAPI**、**scbi**、および **utils** パッケージ。これらは内部での使用のみです。

インストール後、タスク・マネージャのスケジュールには新しいタスク (具体的には、**SC.Core.Tasks** サブパッケージ内のタスク) も表示されます。

["安全なカスタマイズのためのガイドライン"](#) も参照してください。

## 2.3 アップグレード

前述のように、InterSystems IRIS、InterSystems Supply Chain Framework、およびアプリケーション・コードの 3 つのデプロイ可能なコンポーネントが関与します。これらの各コンポーネントは、一緒にアップグレードすることも、個別にアップグレードすることもできます。デプロイ可能なコンポーネントを複数同時にアップグレードする必要がある場合は、以下の順序で行ってください。

1. コア InterSystems IRIS データ・プラットフォームをアップグレードします。

InterSystems IRIS のアップグレード手順を参照し、指示に従って **%SYS** ネームスペースをリコンパイルしてください。

2. InterSystems Supply Chain Framework をアップグレードします。

InterSystems Supply Chain Framework をアップグレードするには、最初のデプロイと同じコマンドを使用します。

```
zpm:SC> install isc-supply-chain
```

オプションとして、特定のバージョンのフレームワーク (この例ではバージョン 1.1.0) にアップグレードするには、以下のコマンドを使用します。

```
zpm:SC> install isc-supply-chain 1.1.0
```

3. アプリケーション・コードをアップグレードします。これを実行するにはクライアントの DevOps ツールを使用します。

## 2.4 ドキュメント・リンクの再構成

基礎となる製品は InterSystems IRIS であり、ドキュメント・リンクはその製品に固有であるため、Supply Chain Orchestrator のドキュメントにリダイレクトするよう Web サーバを更新する必要があります。

プライベート Web サーバを使用している場合、install-dir¥httpd¥conf¥httpd-doc.conf ファイルを更新します。これは最初、以下のように表示されます。

```
Redirect /csp/docbook/ http://docs.intersystems.com/iris20231/csp/docbook/
```

このファイルを以下のように (この特定のリリースのコンテンツに) 変更します。

```
Redirect /csp/docbook/ http://docs.intersystems.com/supplychain20231/csp/docbook/
```

または、以下のように (時間の経過に伴って変更される最新のコンテンツに) 変更します。

```
Redirect /csp/docbook/ http://docs.intersystems.com/supplychainlatest/csp/docbook/
```

別の Web サーバを使用している場合、上記の手順をそのサーバに合わせて適宜変更してください。

## 2.5 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [安全なカスタマイズのためのガイドライン](#)
- ・ [Building Smart Real-Time Supply Chain Applications](#)

# 3

## サプライ・チェーン・ソリューションの概要

このページでは、InterSystems Supply Chain Orchestrator™ で構築されたソリューションに必要なものをまとめます。ソリューションは、以下を容易に構築できるよう設計されています。

- ・ サプライ・チェーン全体にわたる可視性を提供する KPI。
- ・ 問題管理：問題を生成、分析（必要に応じてユーザ入力を伴う）、および解決（外部 API の呼び出しを伴う場合もある）します。

InterSystems IRIS® の基盤によって、ソリューションでは、データのロード、データ変換、独自の API の提供、追加の分析オプションのサポートなど、さらに多くのことが可能になります。

### 3.1 前提条件

前提条件として、Supply Chain Orchestrator のコア・システムを以下のように設定する必要があります。

1. InterSystems IRIS インスタンスをデプロイします。
2. Supply Chain Orchestrator で使用するよう予約された InterSystems ネームスペース (supply chain namespace) を作成します。このドキュメントでは、ネームスペース名として **SC** を使用していますが、別の名前を使用することもできます。
3. InterSystems Supply Chain Framework をそのインスタンスにインストールします。

“[インストールとアップグレード](#)” を参照してください。

### 3.2 実装の基本要件

[コア・システム](#)を設定したら、以下の手順を実行する必要があります（特定の順序で実行する必要はありません）。

- ・ 必要に応じて、[サプライ・チェーン・テーブル](#)を拡張します。
- ・ サプライ・チェーン・テーブルにデータをロードします。このためには、製品により提供される [API](#) を使用できます。また、これは以下に示すように[サプライ・チェーン・ネームスペース](#)内で作成する[プロダクション](#)内でも実行できます。
- ・ タスク・マネージャを使用して、以下のタスクをスケジュールに追加し、ビジネス・ニーズに適したタイミングでこれらを実行します。
  - `SC.Core.Tasks.AnalyzeAllNewIssues`

- SC.Core.Tasks.BuildCubes
- SC.Core.Tasks.ConsolidatedInventoryTask
- SC.Core.Tasks.PredictInventory
- SC.Core.Tasks.SynchCube
- SC.Core.Tasks.SynchIssueCube
- SC.Core.Tasks.UpdateKPIIssue

これらのタスクはすべて、[サプライ・チェーン・ネームスペース](#)で実行する必要があります。

- ・ 以下に示すように、[サプライ・チェーン・ネームスペース](#)に[プロダクション](#)を作成します。上にリストした問題に関連したタスクと連携して、[プロダクション](#)は[問題管理](#)を担います。
- ・ Business Intelligence アナライザを使用して、製品により提供される[分析キューブ](#)を理解します。これらのキューブの目的は、[サプライ・チェーン・データ](#)を分析し、KPI を作成できるようにすることです。
- ・ 必要に応じて、[分析キューブ](#)をコピーしてそのコピーを変更するか、独自の分析キューブを最初から[作成](#)します。これは、事前定義されたキューブで必要なデータが定義されない場合にのみ必要になります。他に、[サブジェクト領域](#)を定義して、データのサブセットへのアクセスを提供することもできます。サブジェクト領域はキューブとまったく同じように使用できます。
- ・ 1 つ以上の KPI を定義します。このタスクは主に、要件を調べ、[分析キューブ](#)を使用して必要な値を取得し、単純な仕様を作成し、API を使用して KPI をシステムに追加することで構成されます。KPI により問題を生成できますが、必ずしもそうしなければならないわけではありません。

## 3.3 プロダクションの要件

Supply Chain Orchestrator の中核となる機能は[問題の管理](#)です。これには通常カスタムの[ビジネス・ロジック](#)が含まれ、このロジックには、自動化、サードパーティ API の呼び出し、および InterSystems のワークフロー・テクノロジーを使用したユーザの関与 (必要に応じて) を含めることができます。カスタムのビジネス・ロジックは、管理ポータル内で作成、構成、および管理する[プロダクション](#)により提供されます。このプロダクションの要件は以下のとおりです。

- ・ プロダクションは、[サプライ・チェーン・ネームスペース](#)内に存在する必要があります。
- ・ プロダクションには、すぐに使用可能なビジネス・サービス SC.Core.BP.Service.SingleIssueBS を含める必要があります。
- ・ 実行する必要がある問題分析のタイプごとに、1 つのビジネス・プロセスを含める必要があります。通常問題のタイプごとに独自の分析が必要です。例えば、出荷の遅れは 1 つの問題のタイプです。出荷の遅れは通常どれも同じように分析され、そのロジックは他のタイプの問題には適用されません。

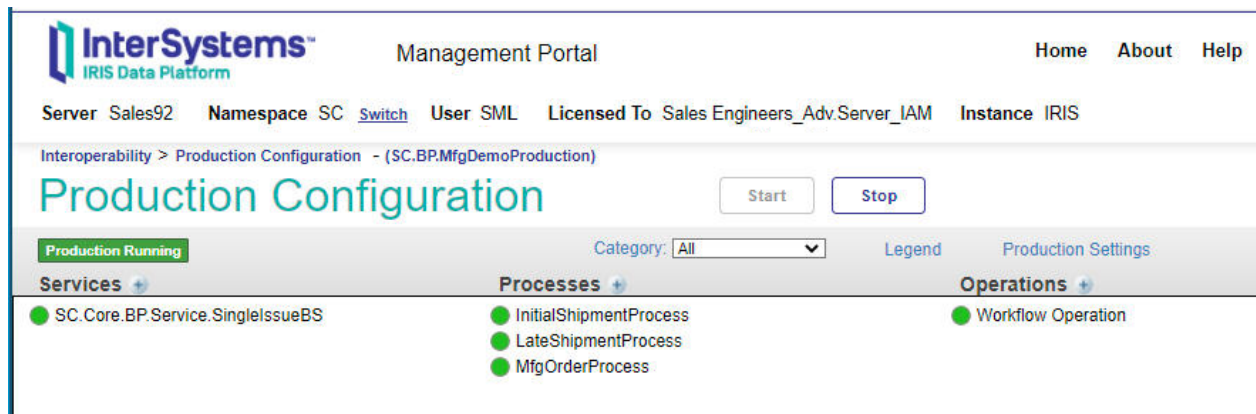
KPI から自動的に生成された問題の場合は、通常各 KPI 定義に 1 つのビジネス・プロセスがあります。

これらのビジネス・プロセスを作成するには、管理ポータル内でグラフィカル・エディタを使用します。これにより、コーディングなしに複雑なロジックを作成できます。詳細は、[チュートリアル](#)を参照してください。

- ・ ワークフローを使用する必要がある場合、プロダクションには、すぐに使用できるビジネス・オペレーション **EnsLib.Workflow.Operation** を含める必要があります。このビジネス・ホストの構成名をメモしてください。ビジネス・プロセス内ではこれを使用する必要があります。該当する場合、ビジネス・プロセスで <call> 要素を使用してこのビジネス・オペレーションを呼び出すことができます。

以下に例を示します。





また、プロダクションを使用して一部またはすべてのデータをサプライ・チェーン・テーブルにロードすることもできます。

## 3.4 安全なカスタマイズのためのガイドライン

アプリケーションを作成する際、以下のガイドラインに留意して、追加またはカスタマイズした内容が InterSystems IRIS やアドオンの InterSystems Supply Chain Framework に対するアップグレード後も存続するようにしてください。

- ・ サプライ・チェーンのデータ・モデルを表す **SC.Data** パッケージ内のクラスを編集できます。その場合、これらのクラスの **Storage** セクションに対して手動で変更を行わないようにしてください。クラスを変更する際に InterSystems IRIS が更新を行うようにしてください。クラス・コンパイラはストレージ定義を安全に拡張して、データへのアクセスを維持します。
- ・ フレームワークにより提供される他のクラスは変更しないでください。例えば、パッケージ **SC.Core.\*** の下にあるものは何も変更しないでください。

**SC.Data** 内のクラスを除き、すべてのクラスはアップグレード時に置換される可能性があります。

- ・ **フレームワークにより提供されるパッケージ**にはクラスを追加しないでください。
- ・ 相互運用対応のネームスペースに適用される標準の InterSystems IRIS 命名規則に従って、他のパッケージ内に独自のクラスを作成できます。

## 3.5 関連項目

- ・ [インストールとアップグレード](#)
- ・ [Building Smart Real-Time Supply Chain Applications](#)
- ・ [タスク・マネージャの使用 \(一般\)](#)
- ・ [プロダクションの正式な概要 \(一般\)](#)
- ・ [プロダクション内のワークフローの概要 \(一般\)](#)

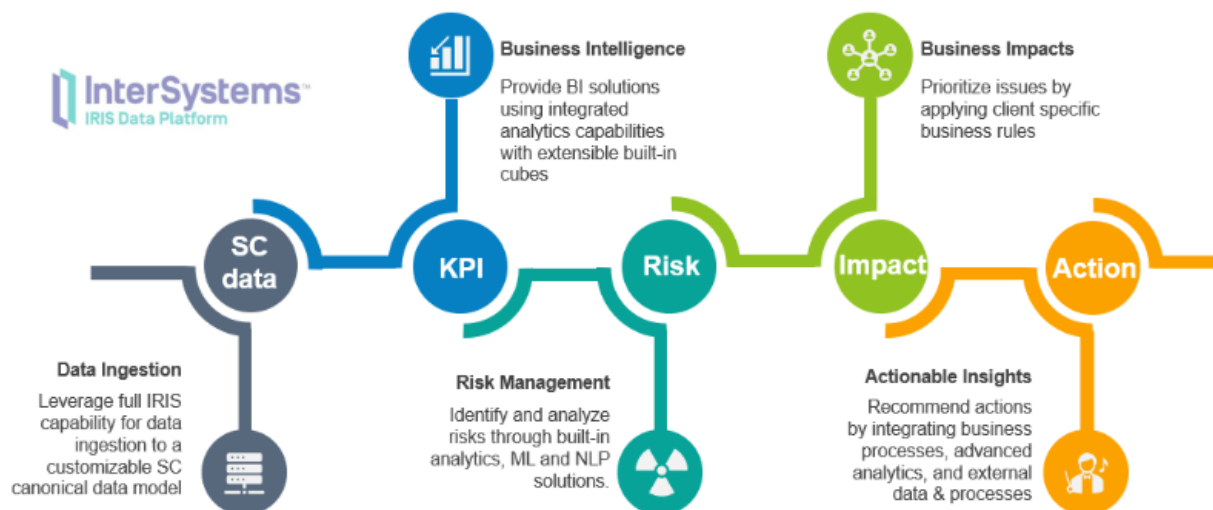


# 4

## 問題の管理

InterSystems Supply Chain Orchestrator™ では、問題とは、サプライ・チェーンに影響を与える可能性のあるリスク、懸念、事柄を取得するために使用される InterSystems IRIS® オブジェクトです。ほとんどの問題では、影響を最小限に抑えるためにある種のアクションが必要になる場合もあります。問題の例としては、出荷の遅れ、在庫切れ、人材不足などがあります。問題は、注文、出荷、在庫など、サプライ・チェーン・オブジェクトの特定のインスタンスにリンクできますが、リレーションシップは常に必要なわけではありません。問題は、カテゴリ、関連するデータ・オブジェクト、期間、ステータスなどにより追跡できます。

以下の図は、Supply Chain Orchestrator でサプライ・チェーン・データがどのように処理されるのかと、使用される関連テクノロジーを示しています。図内のリスクは、問題オブジェクトを使用してモデル化され、関連する影響とアクションも、問題オブジェクトにリンクされています（つまり、問題 API は、関連付けられた影響とアクションのデータが存在する場合、それらも返します）。



### 4.1 問題分析の例

問題管理は、Supply Chain Orchestrator の中核を成します。問題管理には、問題分析と解決が含まれます。以下は、デモ UI に示されている問題解決の例です。

The screenshot displays the 'InterSystems IRIS for Supply Chain' interface. On the left, a sidebar contains navigation icons. The main content area shows details for a 'Sales Order: d60924c1-ec18-4de8-963d-c569d75fb201'. The order is from 'Customer: China Unicom' and is 'Ship to: LOC-CUST-12, 21 Jin Rong Jie Gong Yu G Zuo, Beijing, CHN'. The order value is \$372,000, and the status is 'Closed'. Key dates include 'Order Placed Date: 8/17/2022, 12:00:00 AM', 'Requested Ship Date: 8/19/2022, 12:00:00 AM', 'Requested Delivery Date: 8/26/2022, 12:00:00 AM', 'Committed Ship Date: 8/18/2022, 12:00:00 AM', and 'Committed Delivery Date: 8/28/2022, 12:00:00 AM'.

The 'Issue' section on the right provides a detailed analysis:

- Issue:** Sales orders which were shipped late compared to the committed ship date
- Analysis:** 10/17/2022, 5:50:35 AM (with an 'Update Analysis' link)
- Severity:** 2 **Urgency:** 3
- Root Causes:** Order expected to be late due to a delayed maintenance caused by a late shipment of a part needed.
- Impact Analysis:** Serious customer satisfaction concerns, and a penalty of \$50K.
- Recommendations:**
  - ☐ **Option A** RECOMMENDED  
Move the sales order to Hamburg plant  
Impact: The order will be fulfilled on time, with extra \$1500 shipping cost. Extra work load on Hamburg plant will require an early maintenance window.
  - ☐ **Option B**  
Wait for the part, complete the maintenance, and use expedited shipping once done.  
Impact: \$2000 extra shipping cost. Order expected to be 2 days late.
  - ☐ **Option C**  
Leverage spare parts in Munich plant and redirect current shipment to Munich.  
Impact: Current sales order will be done on time, but this will put \$250,000 revenue at risk at Munich plant due to the delayed maintenance.

An 'Accept' button is located at the bottom of the recommendations section.

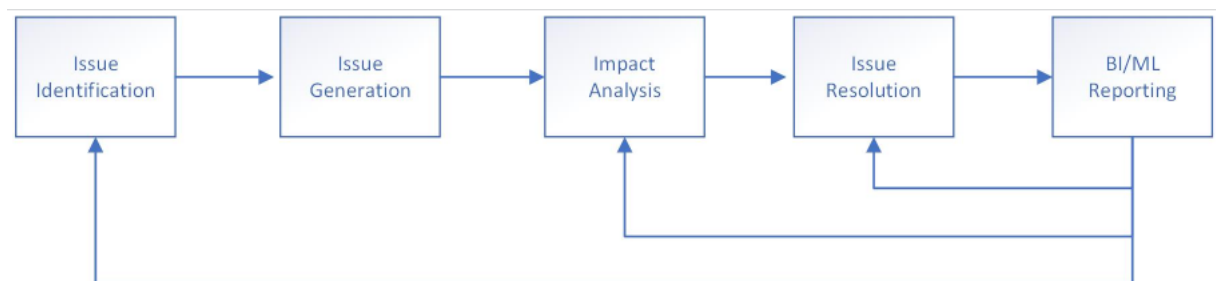
上記の例に示すように、問題分析では以下の側面を扱うことができますが、これらに限定されるわけではありません。

- ・ 根本原因の分析
- ・ 影響の分析
- ・ 重大度と緊急度による問題の評価
- ・ 問題の解決/推奨
- ・ 通知の送信や ERP システムでの発注など、解決に関連したアクション

ビジネス・ロジック向けに設計されたグラフィカル UI を使用して、InterSystems IRIS [ビジネス・プロセス](#)に分析を実装できます。

## 4.2 問題のライフ・サイクル

以下の図は、Supply Chain Orchestrator における問題のライフ・サイクルを示しています。



問題の特定および生成には、さまざまな方法があります。一般的な方法の 1 つとして、[KPI](#)を使用した問題の自動生成があります。つまり、KPI の条件を満たすあらゆるレコードについて問題が生成されます。例えば、販売注文の納期遅延の KPI は、納期が遅れる各注文に対して問題を生成できます。

KPI は問題を生成するための 1 つの方法にすぎません。問題を生成するための他の方法としては、以下のものがあります。

- ・ Supply Chain Orchestrator の BPL/DTL プロセス経由。任意のデータ統合またはビジネス・プロセスを使用して、特定の条件について問題を生成できます。
- ・ Smart Data Service (SDS) を使用。例えば、SDS を実装して、外部イベントをサブスクライブし、現在のサプライ・チェーンに対する影響を判断できます。そのような影響は、問題の形で捕捉され、InterSystems IRIS で保持できます。
- ・ 外部システムから送信。サプライ・チェーンのリスクは、Supply Chain Orchestrator の外部で特定できます。例えば、製造アプリケーションが工場における特定の製品の部品の不足を特定し、問題解決と報告のため、API を介してその情報を問題として Supply Chain Orchestrator に送信することができます。

各問題には、影響分析と問題解決ロジックのために、ビジネス・プロセスを関連付けることができます。ビジネス上の影響には、重大度と緊急度のレベルの設定、推定される時間と金銭的な影響、および問題によるその他の影響が含まれます。

同じビジネス・プロセスを、解決ロジックの実装にも使用できます。問題は、プロセスを通じて自動的に解決できる場合がありますが、ビジネス・ユーザがオプションを確認して最終決定を行うことが必要な場合があります (その場合 InterSystems IRIS の相互運用ワークフローが使用されます)。

## 4.3 KPI からの問題の生成

KPI により問題を生成できます。KPI の問題の生成を有効にするには、KPI の定義で以下を指定します。

```
"issueKpi": true,
```

この値が true に設定されている場合、以下のオプションの属性も KPI 定義に追加できます。

- ・ `defaultIssueSeverity` は、この KPI から生成されたすべての問題に既定の重大度レベルを指定します。
- ・ `analysisService` は、問題の分析および解決ロジックの実行のための BPL 名を指定します。

問題生成が有効なすべての KPI で問題を更新するために、自動的に実行される以下のシステム・タスクがあります。

- ・ **SC.Core.Tasks.UpdateKPIIssues** は KPI で必要に応じて問題を生成します。具体的には、このタスクは問題生成が有効な KPI ごとに以下を実行します。
  1. KPI 条件をトリガしたレコード (注文の遅れなど) を特定します。
  2. この KPI と組み合わせて、これらのレコードに対応する問題がデータベースにあるかどうかを確認します。問題がまだ存在しない場合、タスクは KPI 定義に含まれる情報 (既定でこの種の問題を分析する [ビジネス・プロセス](#) の名前など) を使用して問題を作成します。新規の問題のステータスは `open` です。
  3. データベースにこの KPI に関連付けられた追加のレコード (現在は条件をトリガしていないが以前はトリガしていたレコード) が含まれているかどうかを確認します。含まれる場合、タスクはこれらの問題を更新して閉じます。具体的には、そのステータスを `closed` に設定し、`resolutionType` を `noLongerValid` に設定します。
- ・ **SC.Core.Tasks.AnalyzeAllNewIssues** は問題分析を開始します。このタスクは、閉じられていない問題で、まだ問題分析レコードがないものをすべて探します。このような問題に対して、タスクは新しい問題分析を作成します。プロダクションのビジュアル・トレースでは、**SC.Core.BP.Service.SingleIssueSS** が問題分析の要求メッセージを受信しているのがわかります。その後、**SC.Core.BP.Service.SingleIssueSS** は問題によって指定されたビジネス・プロセスにメッセージを送信します。

ビジネス・プロセスにワークフローが含まれる場合、ビジュアル・トレースには該当するこれらのコンポーネントも表示されます。

KPI の問題更新ロジックを実行する [API 呼び出し](#)もあります。

## 4.4 問題管理 API の概要

Supply Chain Orchestrator は、問題を処理するためのオプションを含む [データ API](#) を提供します。

### 問題の作成

```
POST {{IRIS-SERVER}}/api/scbi/v1/issues
```

問題の値は JSON 形式で要求の本文に追加されます。

### 問題の取得

```
GET {{IRIS-SERVER}}/api/scbi/v1/issues
```

任意の問題属性に基づいた追加のクエリ・パラメータを HTTP パラメータの形式で追加できます (status=open など)。

### KPI の問題の更新

```
POST {{IRIS-SERVER}}/api/scbi/v1/kpiissues
```

指定された KPI の問題を強制的に更新します。これは、問題生成が有効な KPI に適用されます。この場合、以下のように JSON 本文データで KPI 情報を指定します。

```
{
  "kpi": "SalesOrderLateShipVsCommitted"
}
```

### 問題の新規分析の強制

```
POST {{IRIS-SERVER}}/api/scbi/v1/analyzeissue/[ISSUE-ID]
```

このオプションは、プロセスが問題に関連付けられている場合に適用されます。この場合、以下の JSON 本文データを含めることで使用するプロセスを指定します。

```
{
  "processName": "YourProcessName"
}
```

## 4.5 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [Building Smart Real-Time Supply Chain Applications](#)
- ・ [プロダクションの正式な概要 \(一般\)](#)
- ・ [Building BPL Business Processes \(一般\)](#)

- ・ [Supply Chain Orchestrator 向け BPL チュートリアル](#)
- ・ [ワークフロー・シナリオの追加](#)
- ・ [API リファレンス](#)





# 5

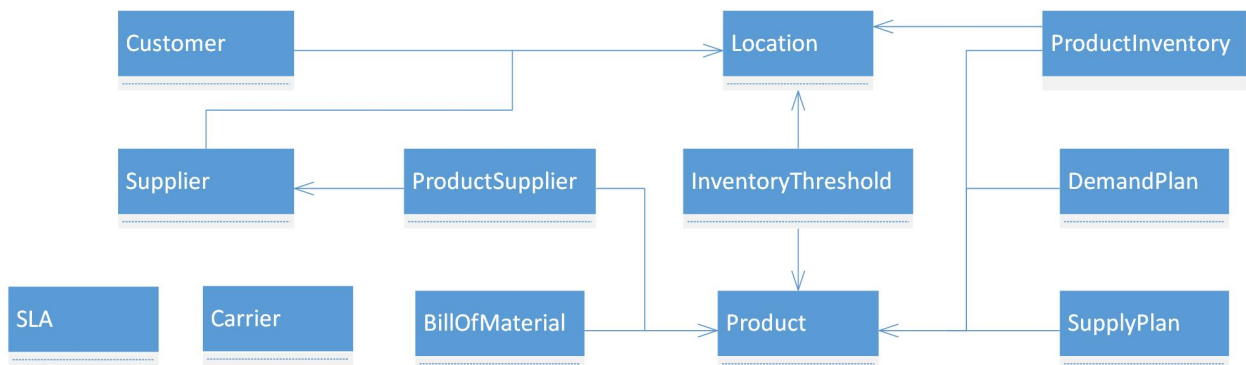
## サプライ・チェーン・データ・モデル

このページでは、InterSystems Supply Chain Orchestrator™ で提供されるデータ・モデルについて説明します。

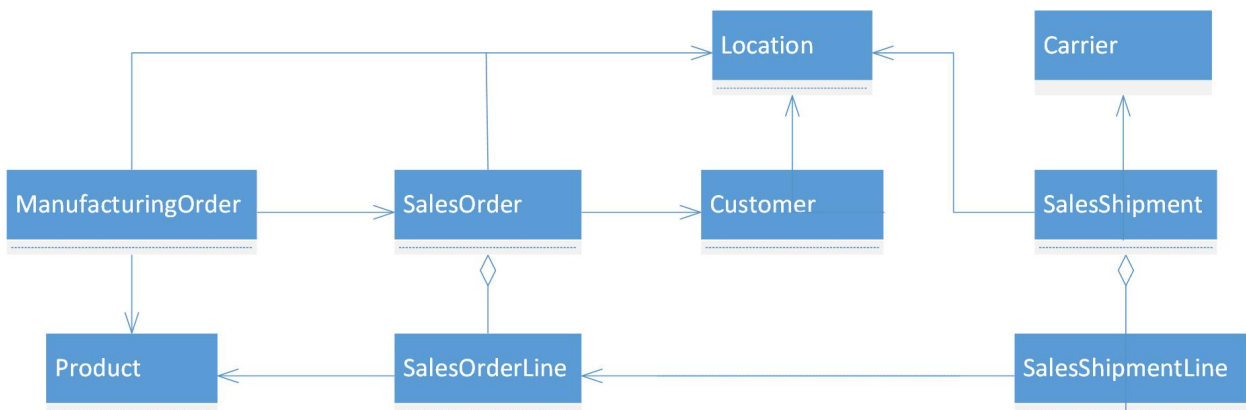
### 5.1 サプライ・チェーン・データ・オブジェクト

以下の図は、現在のモデルでサポートされるデータ・オブジェクトと、そのリレーションシップを示しています。

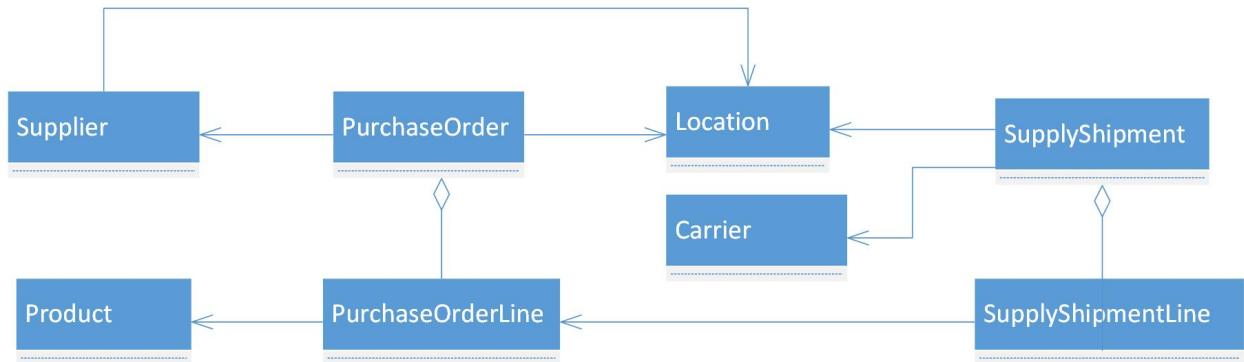
#### 5.1.1 サプライ・チェーンのマスタおよび参照データ



#### 5.1.2 アウトバウンドおよび製造データ



### 5.1.3 インバウンドのサプライ・チェーン・データ



## 5.2 データ・モデル・クラスの概要

モデルでサポートされるすべてのサプライ・チェーン・オブジェクトは、パッケージ **SC.Data** 内にあります。これらのクラスは、任意のクライアントのビジネス・ニーズに合わせて**カスタマイズ**できます。これらのデータ・モデル・クラスでは、パッケージ **SC.Core.Data** で提供されるコア実装を利用しますが、クライアントはコア・パッケージに含まれるものを変更することはできません。データ・モデルの利用（データ統合や UI 開発など）を支援するため、データ・モデルのライブ・ドキュメンテーションとして [API](#) が提供されています。

## 5.3 データ・モデルのカスタマイズ

技術スキルに応じて、さまざまな方法でデータ・モデルを拡張できます。

### 5.3.1 コーディングによるカスタマイズ

すべてのサプライ・チェーン・データ・オブジェクトには、パッケージ **SC.Data** 内に、インターシステムズのクラス定義があります。これらは、すべてのサプライ・チェーン・アプリケーションとサービスに使用されるクラスです。これらのクラスを変更して、既存のデータ・モデルに新しい属性を追加できます。標準のモデル属性はすべて親クラスで定義されているため、以下の例に示すように、これらのクラスは、そのままではほとんど空の状態です。これらのクラスに新しいプロパティやインデックスを追加して、カスタマイズできます。

```

///This is a custom implementation of the Customer class
Class SC.Data.Customer Extends SC.Core.Data.Customer
{
    Property MyNewProp as %String;
}

```

### 5.3.2 API によるカスタマイズ

クラス・プログラミングの知識のないユーザや、IRIS サーバに直接アクセスせずにただ変更を加える必要のあるユーザには、任意のサプライ・チェーン・オブジェクトに新しいカスタム属性を追加するために使用できる [API 呼び出し](#)があります。そのような API 呼び出しを行うことで、対応するオブジェクト定義クラスが更新され（手動でコーディングされたかのように）、データベース・テーブルが自動的に調整され、属性が InterSystems IRIS® プラットフォーム上（分析キューブなど）やビジネス・プロセス内のどこでも使用できるようになります。

## 5.4 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [サプライ・チェーン・データ・モデル API](#)
- ・ [サプライ・チェーン・データ API](#)
- ・ [API リファレンス](#)



# 6

## サプライ・チェーン・データ・モデル API

このページでは、InterSystems Supply Chain Orchestrator™ で提供されるデータ・モデル API について説明します。Supply Chain Orchestrator には、モデルの検出とカスタマイズに使用できる、以下の手段を提供する[データ・モデル API](#) が含まれます。

- ・ すべてのサプライ・チェーン・データ・オブジェクトをリスト
- ・ 任意のオブジェクトの詳細 (属性、データ型、サイズ制限など) を取得
- ・ 既存のサプライ・チェーン・オブジェクトにカスタム属性を追加することでデータ・モデルを拡張

### 6.1 サプライ・チェーン・データ・オブジェクトのリスト

この API 呼び出しを使用して、データ・モデル内のすべてのサプライ・チェーン・オブジェクトを検索します。

```
GET {{IRIS-SERVER}}/api/scdata/v1/objects
```

応答は以下の例のようになります。

```
[
  {
    "objectName": "BOM",
    "className": "SC.Data.BOM",
    "description": "Object for bill of material. This object is commonly
                    used in manufacturing to specify the parts required
                    to make or assemble a product."
  },
  {
    "objectName": "Carrier",
    "className": "SC.Data.Carrier",
    "description": "This object is used to capture shipment
                    carrier information."
  },
  ...
]
```

各オブジェクトの詳細を検索するには、この API 応答で返された `objectName` 値を、次の API 呼び出しで指定します。

## 6.2 オブジェクト定義の取得

オブジェクトの詳細を取得するには、まず上記の API 呼び出しを使用してオブジェクト名を取得します。次に、以下の API 呼び出しを使用します。

```
GET {{IRIS-SERVER}}/api/scdata/v1/objects/[ObjectName]
```

例えば、以下に示すのは、Customer オブジェクトに対して返される応答の一部です。

```
{
  "objectName": "Customer",
  "className": "SC.Data.Customer",
  "objectName": "This object is used to capture the master
    data for a customer.",
  "attributes": [
    {
      "name" : "uid",
      "description" : "unique ID of a customer",
      "dataType" : "String",
      "required" : 1,
      "maxLength" : "256",
      "isCustom" : 0
    },
    {
      "name" : "name",
      "description" : "official name of the customer",
      "dataType" : "String",
      "required" : 0,
      "maxLength" : "256",
      "isCustom" : 0
    },
    ... ]
}
```

## 6.3 カスタム属性の追加

サプライチェーン・モデル内のオブジェクトに新しい属性を追加するには、以下の API 呼び出しを使用します。

```
POST {{IRIS-SERVER}}/api/scdata/v1/attributes/[ObjectName]
```

この際に、以下のように API 本文に属性定義の JSON を含めます。

```
{
  "name" : "customProperty",
  "description" : "Custom attribute added for testing",
  "dataType" : "String",
  "maxlength" : 120,
  "required" : 1
}
```

この API によって属性が追加されると、データ・アクセス API を使用して、属性にデータをロードしたり、新規作成した属性に基づいてレコードを検索することができますようになります。追加の手順や待機は必要ありません。

**重要** この API では、インデックスや検証規則の追加など、クラスを直接変更することによって実行される、より高度なカスタマイズはサポートされません。また、カスタム属性を更新または削除する API は存在しないため、この API は注意して使用してください。

## 6.4 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [サプライ・チェーン・データ・モデル](#)
- ・ [サプライ・チェーン・データ API](#)
- ・ [API リファレンス](#)





# 7

## データ API の概要

InterSystems Supply Chain Orchestrator™ は、データの作成、更新、削除、および取得のための [API](#) を提供します。ここでは、この API の概要を説明します。

### 7.1 API の URL パターン

すべての API 呼び出しは以下の同じ URL パターンに従います。

```
GET {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/OBJECT_PATH?parameters
POST {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/OBJECT_PATH
GET/PUT/DELETE {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/OBJECT_PATH/uid
```

以下はその説明です。

- `{{IRIS-SERVER}}` の部分は、InterSystems IRIS® インスタンスのサーバ情報です。ローカルで導入されたサーバでは、`http://localhost:52773` のようになります。
- `{{DATAMODEL-PATH}}` の部分は、API ベース URL です。例えば、`/api/scdata/v1` のようになります。
- `OBJECT_PATH` の部分は、小文字の複数形のオブジェクト名です。例えば、`salesorders` や `customers` のようになります。すべてのオブジェクト値については、[以下の表](#)を参照してください。
- 最初の URL パターンは、オブジェクトのデータを取得するか、一連のパラメータを使用してオブジェクトを検索します。任意のオブジェクトの属性を[検索条件](#)で使用できます。
- 2 番目の URL パターンは、新しいオブジェクト・レコードを作成します。要求の本文には、新しいオブジェクトの JSON 文字列を含める必要があります。
- 3 番目の URL パターンは、`uid` (外部の主キー) によりオブジェクト・レコードを取得、更新、または削除します。

以下に例を示します。

- ステータスが `Open` または `PartialShip` であるすべての販売注文の検索：

```
GET {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/salesorders?orderStatus=Open,PartialShip
```

- 新規顧客の作成：

```
POST {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/customers
```

この際に、要求の本文に以下の JSON を含めます。

```
{
  "uid" : "CUST-TEST-101",
  "name" : "Google",
  "type" : "HighTech",
  "contact" : "Ming",
  "url" : "https://google.com"
}
```

- uid 値が SUP-SHIP-1001 である供給出荷レコードの取得：

```
GET {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/supplyshipments/SUP-SHIP-1001
```

- 位置レコードの更新：

```
PUT {{IRIS-SERVER}}/{{DATAMODEL-PATH}}/locations/LOC-PLANT-002
```

この際に、要求の本文に新しい位置の JSON データを含めます。

## 7.2 検索 API

検索は、1 つまたは複数の URL パラメータを使用した GET API でサポートされます。各パラメータは、1 つの検索条件にマップされます。複数のパラメータが使用される場合、パラメータ条件は論理 AND 演算を使用して結合されます。例えば、以下はすべての Lenovo ラップトップ製品を検索します。

```
GET /products?brand=Lenovo&category=laptop
```

各パラメータ名はプライマリ・オブジェクトの属性名と正確に一致する必要があります。場合によっては、セカンダリ・オブジェクト(含まれているオブジェクトや参照オブジェクト)の属性も使用できます。例えば、一部の製品属性を注文検索で使用できます。

検索パラメータ値は、以下に示すように、単一値、値のリスト、または値の範囲にすることができます。

- 単一値。これは最も単純な検索条件で、parameterName=value の形式を使用します。パラメータが文字列または文字列のリストである場合、値のマッチングで大文字と小文字は区別されません。そのため、brand=lenovo と brand=Lenovo は同じです。
- 値のリスト。リスト内の任意の値との一致を必要とする条件の場合、値のリストをコンマ区切りの文字列として指定できます。例えば、Lenovo ブランドか Dell ブランドの製品を見つけるには、brand=Lenovo,Dell を使用できます。すべてのデータ型、文字列、日付、および数字に対してリストを使用できます。
- 値の範囲。日付および数値(通貨を含む)の場合、検索パラメータで範囲を使用できます。範囲は、.. (2 つのドット) で区切られた 2 つの値で指定します。min..max の形式を取り、両方の値が範囲に含まれます。例えば、price=100..200 は、100 (100 を含む) ~ 200 (200 を含む) の範囲の価格のすべてのレコードを検索します。2 つの範囲の境界値は常に必須ではなく、片方がない場合、境界がないことを意味します。例えば、price=..200 は、価格が 200 以下であるレコードを意味し、price=100.. は価格が 100 以上であるレコードを意味します。
- NULL 値。どのデータ型でも、属性が設定されていない(つまり NULL 値である)レコードと、何らかの値が設定された属性(つまり NULL ではない)を持つレコードを検索することができます。そのような場合、任意の属性に対して 2 つの特殊な文字列値 NULL および NOTNULL を使用できます。例えば、attr1=NULL&attr2=NOTNULL のようになります。

例えば、次の呼び出しでは、Lenovo または Dell のブランドで、500 ~ 1000 の価格範囲のすべてのラップトップ・コンピュータを検索します。

```
GET /products?category=laptop&brand=Lenovo,Dell&price=500..1000
```

## 7.3 結果の並べ替え

API 呼び出しによって複数のレコードが返される場合、並べ替えパラメータ `sortBy` を指定することで応答データの順序を定義できます。このパラメータの値は、返されるプライマリ・オブジェクトの属性名のコンマ区切りリストである必要があります。

例えば、次の API 呼び出しは、応答で注文リストをどのように並べ替えるかを定義します。

```
GET /salesorders?sortBy=customer,orderValue
```

これは、最初は顧客順で、次に注文額順で並べ替えが実行されるよう定義します。

既定では、並べ替えは昇順で行われます。順序を降順に変更するには、以下の例に示すように属性名の前に `-` 記号を使用します。この例では、最初に顧客によって昇順で並べ替え、その後 `orderValue` によって降順で並べ替えます。

```
GET /salesorders?sortBy=customer,-orderValue
```

## 7.4 既定の並べ替え

要求の URL で並べ替えパラメータが指定されない場合、返されるプライマリ・オブジェクトに基づいて、既定の並べ替えが適用されます。以下の表は、各オブジェクト・タイプの既定の並べ替えパラメータを示しています。

| オブジェクト             | URL パス              | 既定の並べ替え属性                 |
|--------------------|---------------------|---------------------------|
| Carrier            | carriers            | name                      |
| Customer           | customers           | name                      |
| Supplier           | suppliers           | name                      |
| Product            | products            | name                      |
| Location           | locations           | locationName              |
| BillOfMaterial     | billofmaterials     | productId, parentId       |
| InventoryThreshold | inventorythresholds | siteLocationId, productId |
| Milestone          | milestones          | ID                        |
| ProductInventory   | productinventories  | siteLocationId, productId |
| ProductSupplier    | productsuppliers    | productId                 |
| SupplyPlan         | supplyplans         | locationId, productId     |
| DemandPlan         | demandplans         | locationId, productId     |
| SalesOrder         | salesorders         | -orderPlacedDate          |
| SalesOrderLine     | salesorderlines     | orderId, lineNumber       |
| SalesShipment      | salesshipments      | -actualShipDate           |
| SalesShipmentLine  | salesshipmentlines  | shipmentId, lineNumber    |

| オブジェクト             | URL パス              | 既定の並べ替え属性             |
|--------------------|---------------------|-----------------------|
| PurchaseOrder      | purchaseorders      | -orderPlacedDate      |
| PurchaseOrderLine  | purchaseorderlines  | orderId、lineNumber    |
| SupplyShipment     | supplyshipments     | -actualShipDate       |
| SupplyShipmentLine | supplyshipmentlines | shipmentId、lineNumber |
| ManufacturingOrder | manufacturingorders | -orderEntryDate       |

## 7.5 結果のページ付け

既定では、複数のレコードを返す API 呼び出しは、指定された条件と一致する最初の 100 件のレコードを返します。ページ付けパラメータを使用して、追加のレコードまたはより大きなレコード・セットを取得できます。ページ付けパラメータにより、例えばリソース（メモリ、ネットワーク帯域幅など）を過剰に消費したり、パフォーマンスの問題を引き起こすことなく、Web UI でテーブルを生成するために必要な制御が可能になります。

ページ付けパラメータは、以下のとおりです。

- ・ **pageSize**. このパラメータは、API 呼び出しで返されるレコードの最大数を定義します。返されるレコードの実際数は、この値以下になります。既定値は 100 で、許可される最大値は 1000 です。
- ・ **pageIndex**. このパラメータは、応答で返されるページ・インデックス (0 から始まる) を指定します。既定値は 0 です。

例えば、以下の API 呼び出しは、最初の 200 件の注文をスキップし、次の 100 件の注文を、注文額で並べ替えて返します。

```
GET /salesorders?sortBy=orderValue&pageSize=100&pageIndex=2
```

応答でページ付けを使用して部分的なレコードのみが返される場合、応答内で以下の HTTP ヘッダ・パラメータが生成されます。これらの値を使用して、UI のページ付けロジックを制御します。

- ・ **pageSize**. 要求パス・パラメータで指定されている場合は同じ値です。要求で明示的に指定されていない場合は、既定値が返されます。
- ・ **pageIndex**. 要求パス・パラメータで指定されている場合は同じ値です。要求で明示的に指定されていない場合は、既定値が返されます。
- ・ **returnCount**. 現在の応答で返されるレコード数。この値は **pageSize** 以下です。

## 7.6 日付および日付/時刻の形式

日付または日付/時刻の属性については、メッセージの JSON 本文内で指定するか、または HTTP パラメータ内で指定するかに関係なく、ISO 8601 形式で値を指定する必要があります。

以下は、その例です。日付属性の場合：

2021-02-28

日付/時刻属性の場合（以下の 2 つの値は同等です）：

```
2021-12-15T13:23:15-05:00  
2021-12-15T18:23:15Z
```

タイム・ゾーンの情報が指定されていない場合、UTC 時刻を使用するものと見なされます。例えば、2021-05-24T08:30:00 は 2021-05-24T08:30:00Z と同様に扱われます。

## 7.7 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [サプライ・チェーン・データ・モデル](#)
- ・ [API リファレンス](#)



# 8

## InterSystems Supply Chain Orchestrator の分析キューブ

このページでは、InterSystems Supply Chain Orchestrator™ で提供される [Business Intelligence](#) キューブについて説明します。これらのキューブは、以下のような複数の目的で使用できます。

- ・ [KPI](#) の定義
- ・ InterSystems IRIS® 内で提供される Business Intelligence ツール (Business Intelligence [アナライザ](#)など) による簡単な探索
- ・ [Business Intelligence REST API](#) を介してアクセス可能なピボット・テーブルの作成

これらのキューブは、[サプライ・チェーン・ネームスペース](#)で利用できます。データを[サプライ・チェーン・テーブル](#)にロードし、自動化された[システム・タスク](#)が分析キューブを構築すると、キューブが使用可能になります。

Tip ヒン アナライザが使用できない場合は、ネームスペースが[分析対応](#)であることを確認してください。  
ト

### 8.1 利用可能なデータ

このセクションでは、分析キューブで利用可能なデータについてまとめます。

#### 8.1.1 注文に関する洞察

Supply Chain Orchestrator は、すべての販売注文についての洞察を得るために必要な、以下のようなデータとツールを提供します。

- ・ 注文数、総注文収益、注文あたりの平均収益
- ・ 注文調達のパフォーマンス (時間通り、完全納入など)
- ・ 顧客別 (地域/国/企業)
- ・ 製品別 (カテゴリ/ファミリー/ブランド/製品)
- ・ 時間別 (年/月/日)
- ・ ステータス別

同様に、供給/購入注文についても以下を提供します。

- ・ 注文数、総注文収益、注文あたりの平均収益
- ・ サプライヤのパフォーマンス
- ・ サプライヤ別 (地域/国/企業)
- ・ 製品別 (カテゴリ/ファミリー/ブランド/製品)
- ・ 時間別 (年/月/日)
- ・ ステータス別

## 8.1.2 出荷に関する洞察

Supply Chain Orchestrator では、インバウンドとアウトバウンドの両方について、出荷に関する以下の情報も提供します。

- ・ 出荷数、出荷額
- ・ 輸送中の荷物の ETA および関連するステータス
- ・ 配達された荷物のパフォーマンス
- ・ 顧客/サプライヤ別
- ・ 運送業者、輸送手段、ルート別など
- ・ 出発地、目的地別
- ・ 時間別 (年/月/日)

## 8.1.3 在庫に関する洞察

Supply Chain Orchestrator では、在庫に関する以下の情報も提供します。

- ・ 在庫レベル：在庫切れ、在庫切れ間近、余剰在庫
- ・ 計画と実績
- ・ 製品別 (カテゴリ/ブランドなど)
- ・ 場所別

## 8.2 利用可能なキューブ

Supply Chain Orchestrator では、以下のキューブを提供します。

- ・ SalesOrder キューブ。このキューブは、販売注文関連情報のプライマリ・ソースです。
  - － メジャー：注文数、総収益、平均注文額
  - － デイメンジョン：顧客 (国 → 名前)、販売地域、注文ステータス、発注日 (年 → 月 → 日)、出荷ステータス、配送ステータス。
- ・ SalesOrderLine キューブ。このキューブは SalesOrder キューブを補完するもので、製品別の収益など、主に製品に関連する側面に対応します。
  - － メジャー：合計数、合計額、合計ユニット数



- デイメンジョン：製品カテゴリ(カテゴリ→製品)、製品ブランド(ブランド→製品)、出荷ステータス、配送ステータス
- ・ SalesShipment キューブ。このキューブは、販売注文の出荷についての洞察を提供します。
- ・ PurchaseOrder キューブ。購入または供給注文の情報のソースです。
- ・ PurchaseOrderLine キューブ。このキューブは PurchaseOrder キューブを補完するもので、製品別のコストや数量など、主に製品に関連する側面に対応します。
- ・ SupplyShipment キューブ。供給出荷情報のキューブです。
- ・ Inventory キューブ。在庫関連情報のキューブです。

## 8.3 キューブの使用開始

データを[サプライ・チェーン・テーブル](#)にロードし、自動化された[システム・タスク](#)によって分析キューブが構築されると、[アナライザ](#)を使用してデータを表示できるようになります。キューブについて理解を深めるには、“[サブジェクト領域入門](#)”を参照してください。

## 8.4 関連項目

- ・ [Business Intelligence の概要](#) (一般)
- ・ [アナライザの概要](#) (一般)
- ・ [サブジェクト領域入門](#) (一般)



# 9

## Supply Chain Orchestrator 向け BPL チュートリアル

InterSystems Supply Chain Orchestrator™ では、ユーザが作成および構成する[相互運用プロダクション](#)内のビジネス・プロセスにより[問題](#)分析と解決ロジックが提供されます。プロダクションにはビジネス・プロセスを含める必要があり、それぞれのビジネス・プロセスが特定の種類の問題を分析します。ビジネス・プロセス・ロジックを含むビジネス・プロセスを作成するグラフィカル・エディタを使用して、管理ポータルで各ビジネス・プロセスを作成できます。

このチュートリアルでは、ビジネス・プロセスを製品により提供される残りの問題解決ロジックに接続するために必要な、ビジネス・プロセス設計/構成のステップについて説明します。[ワークフロー・シナリオ](#)の作成については別途取り上げます。

オンライン学習パス "[Building Smart Real-Time Supply Chain Applications](#)" も参照してください。これには、BPL ビジネス・プロセスの作成に関する情報が含まれています。基本的な情報については、[このオンライン・トレーニング](#)と製品ドキュメントを参照してください。

### 9.1 一般的な構造

問題分析の一般的なビジネス・プロセスには、全体的に以下のようなシーケンスがあります。

1. [コンテキストを初期化](#)します。このステップでは、コンテキスト・オブジェクト内の問題と関係するビジネス・オブジェクトを追跡します。
2. 分析を実行します。この分析内で、以下を実行します。
  - ・ [分析値を指定](#)します。具体的には、問題の重大度、緊急度、根本原因、および影響分析です。
  - ・ ワークフローを使用する場合、<call> 要素を使用して、適切な時点でワークフロー・オペレーションを呼び出します。
  - ・ ワークフローを使用している場合、アドオン・ツールを使用して、[ワークフロー・ユーザのための推奨事項を作成](#)します。ビジネス・プロセスで問題解決のためのオプションを提供する場合は、オプションの詳細にアクセスするためにいくつかの追加のステップが必要です。["ワークフロー・シナリオの追加"](#) を参照してください。

分析は必然的にユース・ケースに依存するため、チュートリアルではその一部のみを示します。

3. 問題に[分析結果を保存](#)します。そうすることで、API と UI を介して結果にアクセスできるようになります。

## 9.2 コンテキストの初期化

Supply Chain Orchestrator で使用するためにビジネス・プロセスを初期化するには、以下の手順を実行します。

- ・ [コンテキスト] タブで、以下の値を指定します。
  - [リクエスト・クラス] - SC.Core.BP.Message.IssueAnalysisRequest
  - [レスポンス・クラス] - Ens.Response
  - [コンテキスト・クラス] - SC.Core.BP.IssueContext (またはこのクラスのサブクラス)
- ・ BPL の最初のステップの 1 つとして、<assign> 要素を使用して、以下のように context.issueId を設定します。

The screenshot shows the configuration interface for an **<assign>** activity in the BPL editor. The interface has four tabs: General, Context, Activity, and Preferences. The **Context** tab is selected. The activity is titled **<assign>** with the description "Assigns a value to a property." and a link to "View documentation".

Configuration fields include:

- Name:** set issue to context
- Caption for shape:** (empty)
- Position:** x=200, y=250. There is a **Disabled** checkbox which is unchecked.
- Annotation:** (empty text area)
- Action:** set (selected from a dropdown)
- Property:** context.issueId
- Value:** request.issueId
- Key:** "" (empty text box)

A note at the bottom states: "For collection properties, this string specifies the member".

- ・ 多くの場合、問題は注文や出荷など、ビジネス・オブジェクトまたはトランザクションのインスタンスと結び付けられています。そのオブジェクトについての情報を BPL プロセスで利用できるようにするには、<code> 要素を追加して、必要な値を取得し、それを context オブジェクトのプロパティに割り当てます。これを BPL の最初のステップの 1 つとして実行してください。

インスタンスの影響を受けるビジネス・オブジェクトまたはトランザクションを取得するには、以下に示すように構文 context.getImpactedObject() を使用します。getImpactedObject() メソッドは、選択したコンテキスト・クラス (SC.Core.BP.IssueContext) により提供されます。以下に例を示します。

| General  | Context | Activity | Preferences |
|--|---------|----------|-------------|
| <b>&lt;code&gt;</b><br>Executes one or more lines of code.<br><a href="#">View documentation</a>   |         |          |             |
| Name<br><input type="text" value="Initialize impacted business object"/>   |         |          |             |
| Caption for shape<br>X Y<br><input type="text" value="200"/> <input type="text" value="350"/> <input type="checkbox"/> Disabled<br>Position of shape |         |          |             |
| Annotation<br><input type="text"/>   |         |          |             |
| Code<br><pre>set context.supplyshipment =context.getImpactedObject()</pre>   |         |          |             |

- ・ BPL の最初のステップの 1 つとして、<assign> 要素を使用して、以下のように `context.analysis.processName` を `request.targetProcess` に設定します。

| General  | Context | Activity | Preferences |
|--|---------|----------|-------------|
| <b>&lt;assign&gt;</b><br>Assigns a value to a property.<br><a href="#">View documentation</a>  |         |          |             |
| Name<br><input type="text" value="set process name"/>  |         |          |             |
| Caption for shape<br>X Y<br><input type="text" value="200"/> <input type="text" value="450"/> <input type="checkbox"/> Disabled<br>Position of shape |         |          |             |
| Annotation<br><input type="text"/>   |         |          |             |
| Action<br><input type="text" value="set"/>   |         |          |             |
| Property<br><input type="text" value="context.analysis.processName"/>  |         |          |             |
| Value<br><input type="text" value="request.targetProcess"/>  |         |          |             |
| Key<br><input type="text" value=""/>   |         |          |             |
| <small>For collection properties, this string specifies the member collection that is the target of this assignment.</small>                         |         |          |             |

このステップにより、ビジネス・プロセス名がコンテキストに保存され、分析結果と共に問題に保存し直すことができます。この操作は、異なるビジネス・プロセスを同じ問題の分析に使用できるようにするために必要になります。

## 9.3 分析値の指定

コンテキストを設定したら、重大度、緊急度、根本原因、影響などを判断する実際のビジネス・ロジックを含むステップを追加します。このチュートリアルでは、これらの詳細は説明しません。これらの値を決定したら、`context.analysis` オブジェクトに保存します。プロパティの名前の詳細は、“[分析のプロパティ](#)”を参照してください。

分析値を指定するには、複数の `<assign>` 要素を使用します。各要素が 1 つのプロパティを設定します。以下に例を示します。

## 9.4 分析のプロパティ

分析オブジェクトは、クラス `SC.Core.Data.Internal.IssueAnalysis` により定義されます。これには以下のプロパティがあり、以下の説明に従って設定できます。

- **severity** — 数値スケールで、問題の重大度を指定します。
- **urgency** — 数値スケールで、問題の緊急度を指定します。
- **rootCauseAnalysis** — わかりやすい用語で、問題の根本原因を説明します。
- **impactAnalysis** — わかりやすい用語で、分析結果を説明します。

データ型と長さの制限については、ローカルのクラス・リファレンスまたは任意の IDE で “`SC.Core.Data.Internal.IssueAnalysis`” を参照してください。

## 9.5 分析結果の問題への保存

ビジネス・ロジックが完了したら、分析の詳細を問題に保存する `<code>` 要素を BPL に含める必要があります。

- ・ ビジネス・プロセスにワークフローが含まれない場合、以下の構文を使用して分析を問題に保存します。

```
do context.saveAnalysisToIssue()
```

以下に例を示します。

The screenshot shows a configuration window with tabs: General, Context, Activity, and Preferences. The 'Context' tab is active. It contains a section for a code element with the following fields:

- <code>**: Executes one or more lines of code. [View documentation](#)
- Name**: Save analysis to issue
- Caption for shape**: (empty)
- x**: 200
- y**: 450
- Disabled**: ☐
- Position of shape**: (label)
- Annotation**: (empty text area)
- Code**: do context.saveAnalysisToIssue()

- ・ ビジネス・プロセスでワークフローを使用する場合は、`<code>` 要素には代わりに以下の 2 行のコードが必要です。

```
set context.analysis.workflowId = $listget(process.%MasterPendingResponses.GetAt(1),1)
do context.saveWorkflowAnalysisToIssue()
```

この追加の手順により、ワークフロー ID が保存されます。この ID は、現在のワークフローが閉じられる前に新たな分析が実行されるためにワークフローをクリアする必要がある場合に使用できます。

## 9.6 ビジネス・プロセスのテスト

問題分析のビジネス・プロセスを構成したら、API を介して問題の分析を手動で実行できます。

[分析 API](#) を使用することもできます。これには、指定されたビジネス・プロセスを使用して問題分析を実行し、その後分析結果を取得する方法が含まれています。問題分析を実行するには、以下の API 呼び出しを使用します。

```
POST {{IRIS-SERVER}}/scbi/v1/runissueanalysis/ISSUE_ID
```

この際に、本文に以下の JSON メッセージを含めます。

```
{ "processName": "Your business process name" }
```

この呼び出しが正常に返されたら、以下の API 呼び出しを使用して分析の詳細を取得できます。

```
GET: {{IRIS-SERVER}}/scbi/v1/issues/ISSUE_ID
```

応答の例を以下に示します。

```
{
  "ID": "877",
  "recordCreatedTime": "2022-10-15T14:32:55.472Z",
  "lastUpdatedTime": "2022-10-17T09:50:35.180Z",
  "description": "Sales orders which were shipped late compared to the committed ship date",
  "triggerType": "KPI",
  "triggerObjectId": "SalesOrderLateShipVsCommitted",
  "impactedObjectType": "SalesOrder",
  "impactedObjectId": "d60924c1-ec18-4de8-963d-c569d75fb201",
  "severity": 2,
  "urgency": 3,
  "status": "workflow",
  "latestAnalysis": {
    "recordCreatedTime": "2022-10-17T09:50:35.174Z",
    "lastUpdatedTime": "2022-10-17T09:50:35.179Z",
    "issueId": 877,
    "runSequence": 3,
    "processName": "MfgOrderProcess",
    "severity": 2,
    "urgency": 3,
    "rootCauseAnalysis": "Order expected to be late due to a delayed maintenance
      caused by a late shipment of a part needed. ",
    "impactAnalysis": "Serious customer satisfaction concerns, and a penalty of $50K.",
    "workflowId": "54",
    "status": "workflow",
    "resolution": "none",
    "scenarios": [
      {
        "optionNumber": "A",
        "optionName": "Option A",
        "description": "Move the sales order to Hamburg plant",
        "costImpact": 1500,
        "timeImpact": 36,
        "supportingData": "Impact: The order will be fulfilled on time, with
          extra $1500 shipping cost. Extra work load on Hamburg plant will
          require an early maintenance window.",
        "feasibility": 1,
        "recommended": 1
      },
      {
        "optionNumber": "B",
        "optionName": "Option B",
        "description": "Wait for the part, complete the maintenance, and
          use expedited shipping once done.",
        "costImpact": 500,
        "timeImpact": 0,
        "supportingData": "Impact: $2000 extra shipping cost. Order expected
          to be 2 days late.",
        "feasibility": 1,
        "recommended": 0
      },
      {
        "optionNumber": "C",
        "optionName": "Option C",
        "description": "Leverage spare parts in Munich plant and redirect
          current shipment to Munich.",
        "costImpact": 12000,
        "timeImpact": 0,
        "supportingData": "Impact: Current sales order will be done on time,
          but this will put $250,000 revenue at risk at Munich plant due to
          the delayed maintenance.",
        "feasibility": 1,
        "recommended": 0
      }
    ]
  }
}
```

## 9.7 関連項目

- ・ [サプライチェーン・ソリューションの概要](#)
- ・ [Building Smart Real-Time Supply Chain Applications](#)
- ・ [プロダクションの正式な概要 \(一般\)](#)



- ・ [Building BPL Business Processes \(一般\)](#)
- ・ [問題の管理](#)
- ・ [ワークフロー・シナリオの追加](#)



# 10

## ワークフロー・シナリオの追加

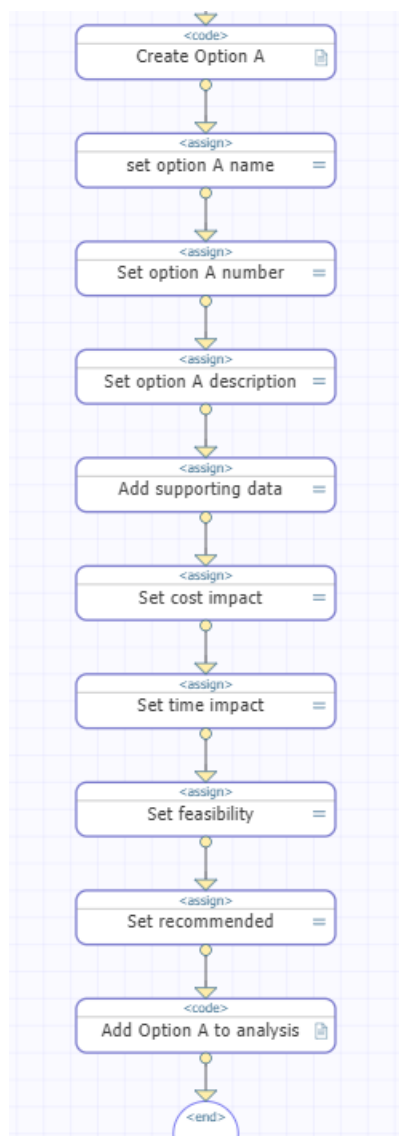
サプライ・チェーンの問題を調査する BPL ビジネス・プロセス内にワークフローを含めることができますが、場合によってはユーザーにさまざまなシナリオを提示する必要があります。InterSystems Supply Chain Orchestrator™ では、これを簡単に実行できます。

このページでは、ワークフロー内にシナリオ・オブジェクトを作成および設定する方法について説明します。

各シナリオには名前、説明、時間やコスト面での影響など、さまざまなプロパティがあります。BPL 内でカスタム・プロパティを定義する必要はなく、既に必要なプロパティを備えたデータ・オブジェクト `SC.Core.Data.Internal.ActionScenario` を使用することができます。

### 10.1 概要

概要として、サンプルの BPL ダイアグラムの関連する部分のスクリーンショットを以下に示します。



## 10.2 シナリオを作成する手順

BPL ビジネス・プロセス内で、以下の手順を使用して各シナリオをビジネス・プロセスに追加します。

1. <code> 要素を追加して、シナリオ・オブジェクトを作成します。具体的には、この要素に以下のように **Code** を指定します。

```
do context.createNewScenario()
```

以下に例を示します。

General Context **Activity** Preferences

**<code>**  
Executes one or more lines of code.  
[View documentation](#)

Name  
Create Option A

Caption for shape

X Y  
200 250 ☐ Disabled

Position of shape

Annotation

Code  
do context.createNewScenario()

この手順は新しいプロパティをコンテキスト・オブジェクトに追加するため、`context.newScenario` のプロパティを設定できるようになります。

- 一連の `<assign>` 要素を含めます。それぞれの要素が 1 つのシナリオ・プロパティの値を指定します。例えば、以下のように `optionName` プロパティを割り当てます。

General Context **Activity** Preferences

**<assign>**  
Assigns a value to a property.  
[View documentation](#)

Name  
set option A name

Caption for shape

X Y  
200 350 ☐ Disabled

Position of shape

Annotation

Action  
set

Property  
context.newScenario.optionName

Value  
"Option A"

Key  
""

- すべてのプロパティを指定したら、`<code>` 要素を追加してシナリオ・オブジェクトを作成します。具体的には、この要素に以下のように `Code` を指定します。

```
do context.addScenarioToAnalysis()
```

以下に例を示します。

The screenshot shows a configuration window with four tabs: General, Context, Activity, and Preferences. The 'Activity' tab is selected. It contains a code editor area with the following text:

```
<code>
Executes one or more lines of code.
View documentation

Name
Add Option A to analysis
Caption for shape
X: 200 Y: 1150 [ ] Disabled
Position of shape

Annotation

Code
do context.addScenarioToAnalysis()
```

上記の手順が完了したら、別のシナリオに対して同じ一連の手順を繰り返すことができます。

これらの手順は、残りのワークフローを実装するために通常の InterSystems IRIS® ガイドラインを補足するものであることに注意してください。

## 10.3 シナリオのプロパティ

シナリオ・オブジェクトは、クラス `SC.Core.Data.Internal.ActionScenario` により定義されます。これには以下のプロパティがあり、以下の説明に従って設定できます。

- ・ **optionNumber** — ユーザへの提示に使用する、単一の文字や単一の数字など、オプションの非常に短い識別子。例は、“[問題分析の例](#)”を参照してください。
- ・ **optionName** — オプションのわかりやすい短い名前。
- ・ **description** — オプションのわかりやすい説明。
- ・ **supportingData** — 分析で使用される裏付けデータ（在庫の位置、労働力の有無など）。
- ・ **feasibility** — 数値スケールで、シナリオの実現可能性を示します。
- ・ **recommended** — これが推奨されるシナリオであるかどうかを示します。
- ・ **costImpact** — オプション/シナリオの財務的影響（数値）。
- ・ **timeImpact** — シナリオの時間的影響。

データ型と長さの制限については、ローカルのクラス・リファレンスまたは任意の IDE で “`SC.Core.Data.Internal.ActionScenario`” を参照してください。

## 10.4 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [プロダクションの正式な概要 \(一般\)](#)
- ・ [Building BPL Business Processes \(一般\)](#)
- ・ [Supply Chain Orchestrator 向け BPL チュートリアル](#)
- ・ [問題の管理](#)





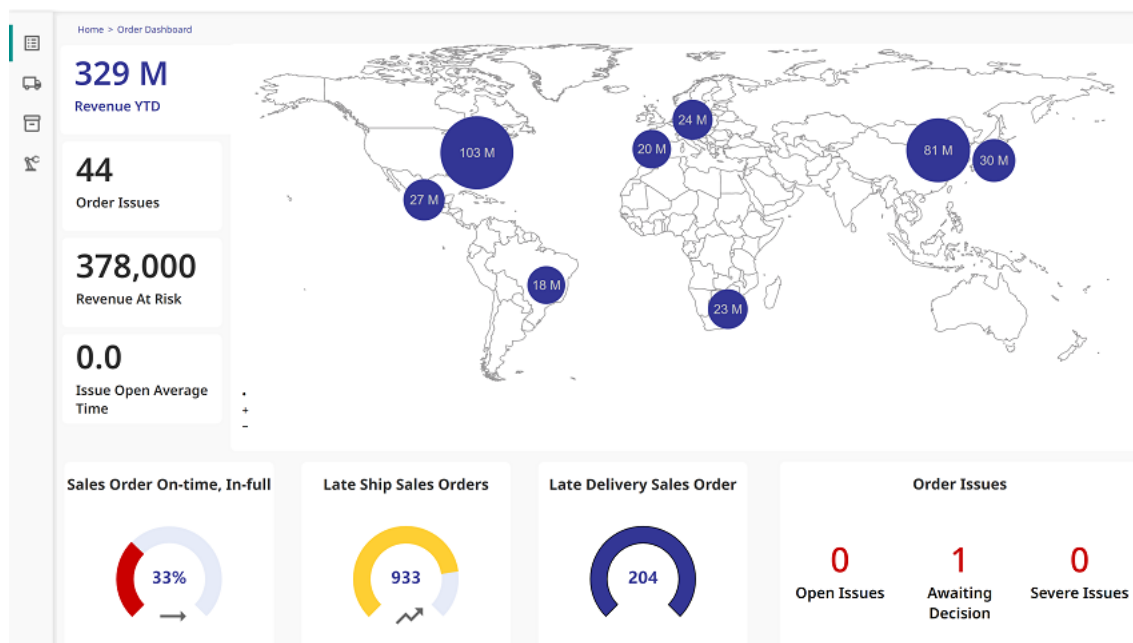
# 11

## InterSystems Supply Chain Orchestrator の KPI の構成

このページでは、主要業績評価指数 (KPI) の定義を構成する方法について説明します。

### 11.1 KPI の目的

KPI とは、一連の目標、目的、同業他社などに照らして企業のパフォーマンスを測定するための定量化可能な指標です。KPI は、関係者が企業の業績やサプライ・チェーン全体でどのようなリスクが存在しているかを理解できるようにするために、サプライ・チェーンで広く使用されています。以下のスクリーンショットは、いくつかのサプライ・チェーンの KPI を示しています。画像内の各数値が KPI です。



Supply Chain Orchestrator™ では、KPI は自動的に問題<sup>問題</sup>を生成するという、もう 1 つの重要な役割も果たします。これは、KPI で修正が必要な条件が特定された場合に適しています。

## 11.2 構成プロセス

KPI を定義するには、以下の一般的なプロセスを使用します。

1. 特定の条件を使用して、[ビジネス用語](#)で KPI ロジックを概念的に定義します。通常これは、この KPI を使用してどのように業績を測定するかについてビジネス・ユーザから収集する情報です。
2. “[KPI の構成例](#)” に示すように、KPI ロジックを Supply Chain Orchestrator のデータ・モデルと分析キューブにマップします。
  - ・ データ・モデルは必要なビジネス・エンティティに対応しているか、または対応するデータ・モデル・オブジェクトに必須の属性すべてがあるか確認します。

そうでない場合は、必要に応じてデータ・モデルを拡張します。詳細は、[データ・モデルのドキュメント](#)を参照してください。

  - ・ 関連するキューブが KPI の要件を満たしているか確認します。このプロセスには、Business Intelligence [アナライザ](#)を使用します (アナライザが使用できない場合は、ネームスペースが[分析対応](#)であることを確認してください)。

KPI にカスタム属性が必要な場合は、そのキューブの独自のコピーを作成し、その属性をディメンジョンまたはメジャーとして含めることが必要な場合もあります。また、KPI に新しいオブジェクトが作成される場合、そのオブジェクトに新しいキューブが必要です。これらのトピックの詳細は、“[InterSystems IRIS Business Intelligence のモデルの定義](#)” および関連する[トレーニング資料](#)を参照してください。
3. KPI 仕様を KPI 定義の JSON 構造にマップします。“[KPI の構成例](#)” を参照してください。
4. [API](#) を使用して KPI 定義を作成し、作成された KPI 定義を別の API 呼び出しで検証します。
5. [KPI 値 API](#) または [KPI リスト API](#) を介して、アプリケーションで新規に作成した KPI 定義の使用を開始します。

## 11.3 KPI ビジネス・ロジック

KPI ビジネス・ロジックの情報を収集する際は、以下を考慮してください。

- ・ KPI の測定対象。目標は、トランザクションの数を測定することでしょうか。それとも、影響を受ける収益、または特定のタスクの平均時間を測定することでしょうか。メトリックが定義されたら、それがキューブのメジャーとして表されていることを確認します。キューブのメジャーがマップされると、メジャーには 2 つのバリエーションが考えられます。
  - 未加工のデータ。KPI 値は、カウント、収益、平均時間など、キューブのメジャーが定義する値です。
  - パーセンテージ。KPI 値は、2 つの数値の割合で定義されるパーセンテージです。この場合、ビジネス・ロジックは分子と分母の両方に必要です。例えば、分子は遅延することが予想される出荷数で、分母はすべての輸送中の出荷品のようになります。
- ・ KPI が測定する主なビジネス・エンティティ。KPI では、販売注文に基づいて注文の出荷遅延を測定したり、供給出荷に基づいて出荷遅延を測定することができます。そのようなビジネス・エンティティは、Supply Chain Orchestrator データ・モデル内のオブジェクトにマップする必要があります。
- ・ KPI 条件。KPI 条件は、対象の条件を定義するビジネス・ロジックを指します。例えば、出荷遅延 KPI の KPI 条件は、要求された到着時間よりも遅い到着が予想される輸送中の出荷品のようになります。チュートリアルでは、そのような条件を KPI 定義の技術用語で捉える方法を示します。以下に追加の考慮事項を示します。
  - KPI 条件は空にできます。この場合、KPI ですべてのレコードが考慮されます。例えば、総注文利益はすべての注文からの収益の合計であるため、KPI 条件は空にできます。

- KPI メジャーのタイプがパーセンテージである場合、分子用と分母用の 2 つの条件セットが必要です。
- ・ KPI データをさらに処理する方法。KPI はさまざまな方法で細分化できます。具体的には、KPI を細分化するためのディメンジョンを構成できます。例えば、出荷遅延 KPI の、サプライヤ、運送業者、出荷先所在地ごとの内訳を確認するとします。この場合、サプライヤ、運送業者、および出荷先所在地の KPI ディメンジョンを定義できます。各 KPI ディメンジョンには対応するキューブのディメンジョンが必要ですが、すべてのキューブ・ディメンジョンを KPI ディメンジョンにマップする必要はありません。すなわち、KPI ディメンジョンはキューブ・ディメンジョンのサブセットです。
- ・ KPI のしきい値。測定された側面がユーザのビジネスにとってどれほど良い/悪いかを示すために、しきい値を定義できます。しきい値を使用して、通知をトリガしたり、UI の視覚表現 (緑/黄/赤の配色など) を設計したりできます。各 KPI で、監視しきい値 (緑～黄色) と警告しきい値 (黄～赤色) の 2 つのしきい値がサポートされます。
- ・ KPI についてのその他の説明的な情報。各 KPI に一意の名前 (API でキーとして使用)、ラベル (UI で表示される名前)、説明、およびステータス (アクティブであるかどうか) を指定できます。

“[KPI の構成例](#)” を参照してください。

KPI がサプライ・チェーンのリスクや懸念事項の捕捉を意図している場合、KPI 定義を利用して自動的に[問題](#)を生成し、それをプラットフォームで追跡できます。例えば、納期遅延 KPI を使用して、納期遅延 KPI により定義される条件を満たす出荷に対して自動的に問題を作成できます。そのような場合、KPI 条件で追加の一連のパラメータを考慮する必要があります。

- ・ 問題生成のインジケータ
- ・ 既定の問題の緊急度レベル
- ・ 問題分析のためのビジネス・プロセス名

## 11.4 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [問題の管理](#)
- ・ [KPI の構成例](#)
- ・ [API リファレンス](#)



# 12

## KPI の構成例

このページでは、InterSystems Supply Chain Orchestrator™ で KPI を定義および構成するプロセスの例を示します。

この例では、提供される分析キューブを使用した KPI 構成のみを取り上げます。その他の方法を使用した KPI の実装は、将来追加される予定です。

オンライン学習パス “Building Smart Real-Time Supply Chain Applications” も参照してください。これには、KPI の作成に関する情報が含まれています。

### 12.1 要件の収集

この例では、早期納品が予想される（サプライヤからの）出荷を捉える KPI に焦点を当てています。以下の表は、この KPI により必要となるロジックについての詳細情報を示しています。

| KPI の機能      | 要件の詳細   |
|--------------|---|
| 名前           | ExpectedEarlyDeliverySupplyShipment (名前にはスペースなし)          |
| ラベル          | Supply Shipment with Expected Early Delivery              |
| 説明           | Supply shipments which are expected to be delivered early |
| ビジネス・エンティティ  | サプライヤからの出荷  |
| KPI メジャー     | 出荷数   |
| KPI メジャー・タイプ | パーセンテージ   |
| KPI 条件 (分子)  | 輸送中、かつ ETA が要求された納期よりも 4 時間以上早い出荷品                        |
| KPI 条件 (分母)  | 輸送中のすべての出荷品   |
| KPI ディメンジョン  | ユーザはサプライヤ、運送業者、および出荷先所在地による KPI の内訳を希望                    |
| 問題のインジケータ    | 問題は KPI 条件を満たす出荷に対し、既定の重大度レベル 2 で生成                       |

### 12.2 要件マッピング

上記の表に収集されたビジネス要件を使用して、以下のマッピングを実行できます。

1. 対象のデータ・モデル・オブジェクトを指定します。既存のサプライ・チェーン・データ・モデルは、オブジェクト SupplyShipment を持つ供給出荷エンティティをサポートします。SupplyShipment オブジェクトには、ETA (estimatedTimeOfArrival) および要求された納期 (requestedTimeOfArrival) に必要な属性もあるため、データ・モデルのカスタマイズは必要ありません。サプライ・チェーン・データ・モデルの各オブジェクトのデータ・オブジェクトと属性の詳細は、[データ・モデルの API ドキュメント](#)を参照してください。このマッピングの結果により、KPI 定義の以下の部分が得られます。


```
{
  "baseObject": "SupplyShipment"
}
```

2. この KPI に使用するキューブを指定します。KPI ロジックのほとんどはキューブの設計とキューブのマッピングで実装されます。このステップのために、InterSystems IRIS® は、非技術系ユーザでも利用できる、以下の手法をサポートする強力なツールを提供しています。

- ・ 分析キューブの理解。KPI ロジックは、事前定義された[分析キューブ](#)に基づいています。具体的には、キューブ・クエリを使用して、対象のレコードとメジャーのセットを特定します。事前構築されたキューブにより、販売注文、供給出荷、在庫といった一般的なサプライ・チェーン・エンティティについての情報に容易にアクセスできます。
- ・ キューブのマッピング。一般的に、プライマリ・ベース・オブジェクトが KPI ベース・エンティティのデータ・オブジェクトに一致するキューブを使用する必要があります。この例では、すぐに使用可能なキューブ SupplyShipmentCube を使用しています。このキューブのベース・オブジェクト SupplyShipment が、サプライヤからの出荷のビジネス・エンティティに一致するためです。このマッピングの結果により、KPI 定義の以下の部分が得られます。

```
{
  "deepseeKpiSpec":
  {
    "cube": "SupplyShipmentCube"
  }
}
```

3. この KPI に使用するメジャーを指定します。前述のように、KPI メジャーはキューブのメジャーである必要があります。この例の KPI では、出荷数を KPI メジャーとして使用します。既定では、すべてのキューブにカウントのメジャーがあります。KPI 定義で使用するメジャーの正確な名前を見つけるには、以下の手順を実行します。

- a. Business Intelligence [アナライザ](#)でキューブを開きます。
- b. 左側の領域 (キューブの内容を表示) から、右側の **[メジャー]** ボックスにメジャーをドラッグ・アンド・ドロップします。
- c. ツールバーの **[クエリを表示]** ボタン  をクリックします。これにより [MDX クエリ] ウィンドウが表示され、以下の例のように単純なクエリが表示されます。

```
SELECT [Measures].[%COUNT] ON 0 FROM [SUPPLYSHIPMENTCUBE]
```

この場合、メジャーの名前は %COUNT です。

- d. メジャー名は、KPI 定義内の kpiMeasure 属性の値として使用します。この結果が、KPI 定義の以下の部分です。

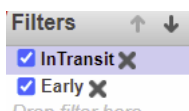
```
{
  "deepseeKpiSpec":
  {
    "kpiMeasure": "%COUNT"
  }
}
```


4. KPI が識別する条件 (多くの場合、複数の特性を伴う) を説明します。最終的に、KPI 条件は一連の MDX [メンバ式](#) として表現されます。この式はシステムにより自動的に論理 AND で結合されます。配送ステータスや注文ステータスなどの一般的な条件については、すぐに使えるキューブ設計により、適切なディメンジョンが提供されます。[アナライザ](#)を使用してこれらの式を取得できます。

- a. アナライザ内の以前の作業内容をクリアします (1 つの方法としては、**[新規]** ボタンを押すことが挙げられます)。
- b. 左側でキューブのディメンジョンを展開し、そこから **[フィルタ]** ボックスにドラッグ・アンド・ドロップします。

この例では、SupplyShipmentCube にはディメンジョン Actual Time of Arrival が含まれ、このディメンジョンにはメンバ Delivered および InTransit が含まれます。InTransit をドラッグして、**[フィルタ]** ボックスにドロップします。同じキューブには、ディメンジョン Estimated vs Actual Time of Arrival も含まれています。これには、メンバ Early、OnTime、および Late を含む status レベルがあります。Early をドラッグして、**[フィルタ]** ボックスにドロップします。

すると、**[フィルタ]** ボックスは以下のようになります。



- c. ツールバーの **[クエリを表示]** ボタン  をクリックします。これにより、[MDX クエリ] ウィンドウが表示されます。この場合、以下のようなクエリが表示されます (意図的に改行を入れてあります)。

```
SELECT FROM [SUPPLYSHIPMENTCUBE] %FILTER
NONEMPTYCROSSJOIN([estimatedVsRequestedDelivery].[H1].[status].&[Early],[actualTimeOfArrival].[H1].[value].&[<null>])
```

この場合、ピボット・テーブルには使用する 2 つの MDX メンバのクロス結合が表示されます。

- ・ [estimatedVsRequestedDelivery].[H1].[status].&[Early]
- ・ [actualTimeOfArrival].[H1].[value].&[<null>]

これらは、NONEMPTYCROSSJOIN 関数内でコンマで区切られていることに注意してください。

- d. 以下の例に示すように、これらのメンバ式を kpiConditions リスト内の項目として使用します。


```
{
  "deepseeKpiSpec": {
    "kpiConditions": [
      "[estimatedVsRequestedDelivery].[H1].[status].&[Early]",
      "[actualTimeOfArrival].[H1].[value].&[<null>]"
    ]
  }
}
```

5. KPI の値タイプが percentage の場合、基本条件 (分母用) も記述します。このステップは、以下の例に示すように baseConditions プロパティを指定する点を除き、前のステップと同じです。

```
{
  "deepseeKpiSpec": {
    "baseConditions": [
      "[actualTimeOfArrival].[H1].[value].&[<null>]"
    ]
  }
}
```

6. KPI ディメンジョンを特定します。これらは、KPI 値の考えられる内訳です。各 KPI ディメンジョンは、MDX [レベル式](#) として表現されます。[アナライザ](#)を使用してこれらの式を取得できます。

- a. アナライザ内の以前の作業内容をクリアします (1 つの方法としては、**[新規]** ボタンを押すことが挙げられます)。

- b. 左側の領域から、右側の [行] ボックスにディメンジョン (またはディメンジョン内のレベル) をドラッグ・アンド・ドロップします。例えば、carrier をドラッグ・アンド・ドロップします。
- c. ツールバーの [クエリを表示] ボタン  をクリックします。これにより、[MDX クエリ] ウィンドウが表示されます。この場合、以下のようなクエリが表示されます (意図的に改行を入れてあります)。

```
SELECT NON EMPTY [carrier].[H1].[name].Members ON 1 FROM [SUPPLYSHIPMENTCUBE]
```

この場合、レベル式は [carrier].[H1].[name] です。

Members は、メンバを表示する MDX 関数で、構文のその部分は必要ありません。

- d. 以下に示すように、KPI 定義に含まれる kpiDimensions 配列のオブジェクト内で、このレベル式を使用します。

```
{
  "deepseeKpiSpec": {
    "kpiDimensions": [
      {
        "name": "carrier",
        "label": "Carrier",
        "cubeDimension": "[carrier].[H1].[name]"
      }
    ]
  }
}
```

このオブジェクトでは、cubeDimension 属性は見つかったレベル式である必要があり、name 属性は API で使用するこの KPI ディメンジョンのローカル名、label 属性は表示のみを目的としています。

- e. 必要に応じてこのステップを繰り返し、他の KPI ディメンジョンを取得します。
7. その他の KPI 構成データ。このチュートリアルでは、KPI 構成の難しい部分について説明しました。残りの部分は、追加のマッピングは不要で、KPI 名や説明などほとんどの情報はビジネス要件のセクションで収集されているため、かなり簡単はずです。

## 12.3 最終結果

最終的な KPI 定義は以下のようになります。

```
{
  "name": "ExpectedEarlyDeliverySupplyShipment",
  "label": "Supply Shipment with Expected Early Delivery",
  "description": "Supply shipments which is expected to be delivered early",
  "baseObject": "SupplyShipment",
  "status": "Active",
  "watchingThreshold": 5,
  "warningThreshold": 10,
  "issueKpi": true,
  "defaultIssueSeverity": 2,
  "analysisService": "Early Arrival Process",
  "type": "DeepSee",
  "deepseeKpiSpec": {
    "namespace": "SC",
    "cube": "SupplyShipmentCube",
    "kpiMeasure": "%COUNT",
    "valueType": "percent",
    "kpiConditions": [
      "[actualTimeOfArrival].[H1].[value].&[<null>]",
      "[estimatedVsRequestedDelivery].[H1].[status].&[Early]"
    ],
    "baseConditions": [
      "[actualTimeOfArrival].[H1].[value].&[<null>]"
    ],
    "kpiDimensions": [
```



```

    {
      "name": "carrier",
      "label": "Carrier",
      "cubeDimension": "[carrier].[H1].[name]"
    },
    {
      "name": "supplier",
      "label": "Supplier",
      "cubeDimension": "[supplier].[H1].[name]"
    },
    {
      "name": "toCountry",
      "label": "Ship to country",
      "cubeDimension": "[shipToLocation].[H1].[country]"
    }
  ]
}

```

以下の API を使用して JSON ペイロードを送信し、KPI を作成できます。

POST {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/definitions

KPI を定義したら、以下のように API 呼び出しにより KPI 値をクエリできます。

GET {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/values/ExpectedEarlyDeliverySupplyShipment

## 12.4 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [Building Smart Real-Time Supply Chain Applications](#) (KPI の作成に関する情報を含む)
- ・ [InterSystems Supply Chain Orchestrator の KPI の構成](#)
- ・ [API リファレンス](#)



# 13

## KPI API の概要

InterSystems Supply Chain Orchestrator™ では、[初期構成の手順](#)を含む、サプライ・チェーン KPI のための[分析 API](#)を提供します。このページでは、KPI に適用される API 呼び出しについて説明します。

### 13.1 KPI の定義

すべての定義済みの KPI を見つけるには、以下の API を使用します (パラメータは不要です)。

```
GET {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/definitions
```

新しい KPI を定義するには、以下の API 呼び出しを使用します。

```
POST {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/definitions
```

この際に、メッセージ本文に以下の JSON を含めます。

```
{
  "name": "KPI name",
  "description": "KPI description",
  "type": "DeepSee",
  "baseObject": "SC base object for the KPI, such as SalesOrder",
  "status": "Status value, such as Active/Inactive",
  "watchingThreshold": xx,
  "warningThreshold": yy,
  "deepseeKpiSpec": {
    "namespace": "IRISnamespace",
    "cube": "KpiCubeName",
    "kpiMeasure": "cubeMeasure",
    "valueType": "raw/percentage",
    "kpiConditions": [
      "MDX condition 1",
      "MDX condition 2"
    ],
    "kpiDimensions": [
      {
        "name": "dim1",
        "cubeDimension": "MDX dimension expression"
      },
      {
        "name": "dim2",
        "cubeDimension": "MDX dimension expression"
      }
    ]
  }
}
```

一般的な KPI 定義属性は以下のとおりです。

| 属性                | 必須  | 説明  |
|-------------------|-----|---|
| name              | はい  | KPI 名。スペースやその他の文字を含まないアルファベット文字列で、同じデプロイにおいて一意である必要があります。                 |
| description       | いいえ | KPI の説明   |
| type              | いいえ | KPI 定義のタイプ。現在の 1 つのタイプ (DeepSee) のみをサポートしています。将来的にはさらに多くのタイプがサポートされる予定です。 |
| baseObject        | はい  | KPI のベースとなるサプライ・チェーン・オブジェクト (SalesOrder や SupplyShipment など)。             |
| status            | はい  | KPI 定義のステータス (Active、Inactive など)。  |
| watchingThreshold | いいえ | 監視レベルのしきい値  |
| warningThreshold  | いいえ | 警告レベルのしきい値  |
| resolutionService | いいえ | この KPI の問題解決を実行するビジネス・サービスのホスト名 (サプライ・チェーン・プロダクション内)                      |

以下に、KPI 仕様の詳細な説明を示します。

| 属性             | 必須  | 説明   |
|----------------|-----|--|
| cube           | はい  | キューブ名  |
| valueType      | いいえ | 値のタイプ。raw または percentage である必要があります。既定値は raw です。これは対応するキューブのメジャーで指定されるのと同じ値を意味します。   |
| kpiMeasure     | いいえ | KPI の MDX 形式のキューブのメジャー。指定されない場合、キューブの対応するカウント・メジャーが使用されます。   |
| kpiConditions  | いいえ | MDX フィルタ値のリスト。KPI 条件を定義するために使用されます。値が指定されないと、KPI は総注文数や総収益などのメジャー合計を取得します。   |
| baseConditions | いいえ | これは、KPI 値のタイプが percentage である場合に使用されます。この一連の MDX フィルタ値により、KPI の分母が決定されます。それに対して KPI の分子は、この一連の条件を kpiConditions で定義された一連の条件と組み合わせることで決定されます。 |
| kpiDimensions  | いいえ | この属性は、KPI の範囲で使える一連のディメンジョンを追跡するために使用されます。各ディメンジョンの名前は、KPI のフィルタリングまたはブレイクダウンに使用されます。  |

KPI に関連した API 呼び出しには、以下のようなものもあります。

- 特定の KPI の定義の取得

```
GET {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/definitions/{KPI_NAME}
```

- KPI 定義の更新。この API 呼び出しは、新規 KPI の作成に使用するのと同じ形式の本文で使します。

```
PUT {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/definitions/{KPI_NAME}
```

- ・ KPI 定義の削除

```
DELETE {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/definitions/{KPI_NAME}
```

## 13.2 KPI 値

ディメンジョンを拡張せずに KPI 値を取得するには、以下の API 呼び出しを使用します。

```
GET {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/values/{KPI_NAME}
```

応答は以下のようになります。

```
{
  "kpiName": "kpiName",
  "values": [
    {
      "label": "kpi",
      "value": xxx
    }
  ]
}
```

ディメンジョンごとにブレイクダウンして KPI 値を取得するには、以下を使用します。

```
GET {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/values/{KPI_NAME}?expandDimension={dimName}
```

応答は以下のようになります。

```
{
  "kpiName": "kpiName",
  "expandDimension": "dimName",
  "values": [
    {
      "label": "dim-value-label1",
      "value": x1
    },
    {
      "label": "dim-value-label2",
      "value": x2
    },
    ...
  ]
}
```

この API 呼び出しを UI で使用して、地域ごとの出荷の遅れなど、KPI の内訳を確認するためのグラフを表示できます。

## 13.3 KPI の詳細リスト

KPI に関連したソース・レコードを取得するには、以下の API 呼び出しを使用できます。

```
GET {{IRIS-SERVER-URL}}/api/scbi/v1/kpi/listings/{KPI_NAME}
```

応答はソース・レコードのリストです。例えば、出荷遅延注文の KPI リスト要求は、出荷が遅れた注文のリストを返します。この API 呼び出しは、通常のページ付けパラメータと並べ替えパラメータを受け入れます。

## 13.4 KPI のフィルタリング

定義済みの KPI については、API 呼び出しに追加のフィルタを適用して、KPI 値またはリストを取得できます。KPI フィルタは REST パラメータ `kpiFilter` で、`(dimName1,value1),(dimName1,value2),...` の形式の値を持ちます。必要に応じて任意の数のフィルタ名と値のペアを指定できますが、特定のディメンジョンに対しては、1 つのフィルタで指定できる名前と値のペアは 1 つのみとなります。`kpiFilter` で使用されるディメンジョンは、KPI 定義の `kpiDimensions` の下で定義する必要があります。別の制限として、ディメンジョンが `expandDimension` で使用される場合、同じディメンジョンをフィルタで使用することはできません。それを除いては、`expandDimension` は同じ API 呼び出しで `kpiFilter` と共に使用して、KPI 値を取得できます。

以下に、要求でフィルタが定義された KPI 要求の例を示します。

```
GET
{{IRIS-SERVER-URL}}/api/scbi/v1/kpi/values/OrderLateShip?kpiFilter=(productFamily,iPhone),(region,EMEA)
```

これは、地域 EMEA における製品ファミリ iPhone の出荷遅延注文の KPI 値を取得します。

## 13.5 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)
- ・ [問題の管理](#)
- ・ [InterSystems Supply Chain Orchestrator の KPI の構成](#)
- ・ [API リファレンス](#)

# 14

## API リファレンス

このページでは、InterSystems Supply Chain Orchestrator™ でソリューションを作成する際に関連する API へのリンクを提供します。

### 14.1 利用可能な API

InterSystems Supply Chain Orchestrator には、以下の 3 つの API があります。

- ・ [データ・モデル API](#) – サプライ・チェーン・モデルを参照し、データ・モデルを拡張する場合に使用します。  
“[サプライ・チェーン・データ・モデル API](#)” も参照してください。
- ・ [データ API](#) – サプライ・チェーン・データの作成、更新、削除、および問題の作成に使用します。  
“[データ API の概要](#)” も参照してください。
- ・ [分析 API](#) – KPI の構成、KPI 値およびデータへのアクセス、問題分析の実行、問題のクローズ、ワークフローのクローズに使用します。  
“[KPI API の概要](#)” および “[問題管理 API の概要](#)” も参照してください。

### 14.2 関連する API

基礎となる InterSystems IRIS® データ・プラットフォームは、さらに以下の関連する Business Intelligence API を提供します。

- ・ [Business Intelligence REST API](#) – 分析キューブ (またはサブジェクト領域)、ピボット・テーブル、および詳細リストにアクセスするために使用します。
- ・ [DeepSee.js](#) – フロントエンド・ライブラリと組み合わせて使用することで、分析データのインタラクティブな表示を作成できます。

### 14.3 関連項目

- ・ [サプライ・チェーン・ソリューションの概要](#)

- ・ [InterSystems Supply Chain Orchestrator の分析キューブ](#)