

**Sterling Selling and Fulfillment Foundation**



## **製品 概念ガイド**

**バージョン 9.1**



**Sterling Selling and Fulfillment Foundation**



## **製品 概念ガイド**

**バージョン 9.1**

**お願い**

本書および本書で紹介する製品をご使用になる前に、311 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Sterling Selling and Fulfillment Foundation バージョン 9.1、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

**原典：** Sterling Selling and Fulfillment Foundation  
Product Concepts Guide  
Release 9.1

**発行：** 日本アイ・ビー・エム株式会社

**担当：** トランスレーション・サービス・センター

第1刷 2012.4

© Copyright IBM Corporation 1999, 2011.

# 目次

## 第 1 章 IBM Sterling Selling and Fulfillment Foundation の概要 . . . . . 1

Sterling Selling and Fulfillment Foundation の概要 . . . . .	1
IBM Sterling Application Platform . . . . .	1
アプリケーション・コンソール Sterling Application Platform . . . . .	1
Sterling Selling and Fulfillment Foundation Sterling Application Platform . . . . .	1
Sterling Distributed Order Management . . . . .	2
Sterling Supply Collaboration . . . . .	2
Sterling Global Inventory Visibility . . . . .	3
Sterling Logistics Management . . . . .	3
Sterling Reverse Logistics . . . . .	3
カタログ管理 . . . . .	3
Sterling Warehouse Management System . . . . .	3
Sterling Selling and Fulfillment Foundation Sterling Application Platform プロダクト・スイートの概要 . . . . .	4
顧客フルフィルメント・スイート . . . . .	4
供給コラボレーション・スイート . . . . .	5
Service Parts Logistics スイート . . . . .	5
サプライ・チェーン・イベント管理スイート . . . . .	6
Sterling Selling and Fulfillment Foundation のドキュメンテーション . . . . .	6
コンテキスト依存ライブラリーとコンテキスト・ヘルプの関係 . . . . .	7

## 第 2 章 Sterling Application Platform のテクノロジーの概念 . . . . . 9

Sterling Application Platform とは . . . . .	9
Sterling Application Platform フレームワーク . . . . .	9
ユーザー・インターフェースの拡張性 . . . . .	9
アプリケーション・コンソール・ユーザー・インターフェース拡張 . . . . .	10
Applications Manager の拡張 . . . . .	11
データベースの拡張 . . . . .	12
セキュリティー管理 . . . . .	12
ログ・ファイルの機密情報のセキュリティー . . . . .	14
データ・アクセス・ポリシー . . . . .	14
アクセス・ポリシー・モード . . . . .	14
カタログ・アクセス・ポリシー . . . . .	14
価格設定アクセス・ポリシー . . . . .	15
ユーザー・アクセスのレベル . . . . .	15
パスワード・ポリシー . . . . .	18
パスワードの暗号化 (encryption) . . . . .	18
JBoss に対する JMS セキュリティー . . . . .	18
API のセキュリティー . . . . .	19
システム管理 . . . . .	19
キュー管理 . . . . .	20
警告の集約 . . . . .	20
印刷拡張性 . . . . .	22

モバイル・アプリケーションの拡張性 . . . . .	22
警告管理 . . . . .	23

## 第 3 章 参加者の管理 . . . . . 25

組織のモデル化 . . . . .	25
Sterling Selling and Fulfillment Foundation の組織のモデル化 . . . . .	27
エンタープライズ . . . . .	27
カタログ組織 (Catalog Organization) . . . . .	28
在庫組織 . . . . .	28
キャパシティー組織 (Capacity Organization) . . . . .	29
エンタープライズのオンボード化 . . . . .	29
顧客定義とベンダー定義 . . . . .	31
組織モデル化のためのガイドライン . . . . .	41
組織モデル化のためのインストール・レベル・ルール . . . . .	42
モデル例 1: 電機会社 . . . . .	43
モデル例 2: サード・パーティーの物流管理会社 . . . . .	47

## 第 4 章 プロセス・モデリングの概念 . . . 51

プロセス・モデリングの概念 . . . . .	51
基本伝票種別と伝票種別 (document type) . . . . .	51
プロセス・タイプ・パイプライン . . . . .	52
パイプラインの決定 . . . . .	54
リポジトリ . . . . .	54
トランザクション . . . . .	55
トランザクション依存関係 . . . . .	56
イベント . . . . .	60
ステータス . . . . .	60
条件 . . . . .	61
拡張 XML 条件 . . . . .	61
アクション . . . . .	64
サービス . . . . .	64
プロセス・モデリング・タスク . . . . .	65

## 第 5 章 カタログ管理概念 . . . . . 67

## 第 6 章 統合在庫管理 . . . . . 69

在庫 ID . . . . .	69
在庫タグ番号として使用されるロット番号 . . . . .	69
バッチと改訂のタグ番号の組み合わせ . . . . .	70
供給と需要 . . . . .	70
予約 . . . . .	71
セグメンテーション . . . . .	71
在庫状況モニター . . . . .	72
最適化 . . . . .	73
回転の速い在庫 . . . . .	73
需要と供給の有効期間 . . . . .	74
在庫集約 . . . . .	75
ハブ・レベル集約 . . . . .	75
エンタープライズ・レベル集約 . . . . .	76

ノード在庫との同期化	76
ノード需要との同期化	76
在庫コスト計算	77
在庫コスト計算方法 1: 平均コスト計算	77
在庫コスト計算方法 2: 先入れ先出し (FIFO)	107
ホット SKU	108
棚卸	108
在庫委託	110
バリエーションのあるアイテムの在庫状況の確認	111
単一オーダー・アイテムに対する複数のフルフィルメント・アイテム	113

## 第 7 章 オーダー管理 . . . . . 115

オーダー管理	115
オーダーの構成要素	115
分類	116
オーダー・パイプライン	116
オーダー・フルフィルメント・パイプライン	116
交渉パイプライン	120
見積フルフィルメント・パイプライン	123
マスター・オーダー・パイプライン	124
交換オーダー (exchange order)	124
バンドル	125
バンドルのモデル化	126
追加のバンドル機能	127
オーダーの作成	127
バンドル・コンポーネントのオーダー数量計算	128
在庫更新	129
サービス関連付け	129
オーダー明細のスケジュール属性の同期	130
オーダー変更	130
オーダーのスケジューリング	131
価格	131
バンドルおよび連鎖オーダー	132
派生したオーダー (derived order)	132
返品	132
多層オーダー	133
連鎖オーダー	133
派生したオーダー (derived order)	136
カスタム・オーダー	136
受注生産オーダー	137
特注生産オーダー	138
オーダー保留処理	139
オーダー明細の保留タイプ	140
マルチクライアント・オーダー	141
オーダー取り込み中の在庫予約	142
複数明細予約	144
予約パラメーター	144
オーダー取得時のキャパシティの割り当て	145
キャパシティの割り当てのパラメーター	145
キャパシティ予約	146
アイテム検証	147
拡張検証	147
確認済みオーダーの変更保留	148

## 第 8 章 案件管理 . . . . . 149

案件管理とは	149
--------	-----

## 第 9 章 オーダー約束とスケジューリング . . . . . 151

オーダー約束の概要	151
出荷する製品の約束	153
製品の在庫状況の検索	153
予定日付の計算	174
日付同期	186
ソーシング・モデルの影響	187
オーダーまたはオーダー明細の出荷のスケジューリング	190
製品出荷の通知	197
バックオーダーの取り扱い	200
サービス要請の約束	201
地域および地域スキーマ	201
サービス・スロット	203
サービス・アイテム	205
サービス・リソース	206
リソース・プール (Resource Pools)	208
配送サービスの約束	211
出荷済み製品とのその他の主な相違点	216
提供サービスの約束	216
オーダー明細の複雑な順序付け	219
アイテム・ベース割り当て	221
FIFO IBA	221
ユーザー構成の IBA	223
スケジュールの変更	224

## 第 10 章 付加価値サービス . . . . . 227

付加価値サービスの概要	227
付加価値サービスの使用	227
アイテム作成の延期	229
作業オーダーの作成	229
作業オーダー保留タイプ	230
サービス作業オーダーのタイプ	230
単一作業オーダーでの複数のサービス明細	231
サービス・リソース付きの作業オーダー	231
複数日の作業オーダー	232
提供サービス作業オーダー	232
配送サービス作業オーダー	233
作業オーダー・タイプの動的決定	233
サービス作業オーダーの作成	234
提供サービスまたは配送サービスの作業オーダーの決定	234
作業オーダー・ノードの決定	234
作業オーダーのプロバイダーの決定	235
作業オーダーのキャパシティー計算	235
サービス作業オーダーの製品予約	237
作業オーダーの確認	237
作業オーダーのキャンセル	238
予定ステータス	238
予定を入れるための手順	239
提供サービスおよび配送サービスのインボイスの作成	240
作業オーダー・パイプライン	241

作業オーダーの付加価値サービス・パイプライン	241
コンプライアンス・サービス	242
コンプライアンス・サービスの構成	243
コンプライアンス処理の最適化のための実行数量の使用	243

## 第 11 章 供給コラボレーション . . . . . 245

供給コラボレーションとは	245
購入オーダー	245
購入オーダー実行パイプライン	245
購入オーダーのステータス	246

## 第 12 章 支払システム . . . . . 249

支払システム	249
決済処理の概要	249
複数の支払方法	249
同期決済処理	250
非同期支払処理	250
アカウントを介した外部集金受け入れ可能	250
カスタム・ロジックを処理するために提供される外部プログラムおよび API	250
料金の集約	250
記録済みの前払い	250
返される要請済みの金額	251
遅延再承認 (delayed reauthorization)	251
請求トランザクション要求 ID	251
承認取り消し	251
事前決済のサポート	252
名前付き外部プログラム	252
ドラフト・オーダー (draft order) の決済処理	252
インボイス作成	253
返品支払処理および返金フルフィルメント	256
交換決済処理	257
手操作による介入	257
複数の制御レベル	258
支払ルール	258
支払タイプのグループ	258
支払タイプ	258
支払方法	259
機密情報のトークン化	260
請求シーケンス	260
返金シーケンス	261
支払ステータス	262
一般支払いプロセス	262
承認処理	262
決済プロセス	268
顧客アカウント (customer account) 支払いプロセス	269

承認処理	270
決済プロセス	271
決済処理のトランザクション	271
カスタム・トランザクションの使用	271
支払関連 API	271
支払い関連の外部プログラム	273
データベースの詳細	273

## 第 13 章 物流管理 . . . . . 281

物流管理とは	281
出荷の計画に関する考慮事項	281
条件	281
出荷計画戦略	282
出荷計画機能	284
経済的な出荷パラメーターの使用	284
ルーティングの決定	286
入庫コンプライアンスと出庫制約	289
バイヤーおよび入庫コンプライアンス	289
エンタープライズと出庫制約	289
競合条件の解決	290
出荷	291
出荷の作成	291
出荷のルーティング	292
ゾーン・スキップ	294
出荷パイプライン	296
出荷コンソール	297
バンドル出荷の作成	298
バンドルの出荷の確認	298
バンドル出荷のインボイス作成	298
集合・混載	299
集合・混載の作成	299
集合・混載実行パイプライン	299
入荷	300
入荷パイプライン	301
入荷コンソール	301
保留処理	302

## 第 14 章 返品物流 . . . . . 305

返品物流とは	305
返品オーダーの構成要素	306
返品物流パイプライン	306
返品物流パイプラインのステータス	306
デフォルト・パイプライン	307

## 特記事項 . . . . . 311

## 索引 . . . . . 315





---

# 第 1 章 IBM Sterling Selling and Fulfillment Foundation の概要

---

## Sterling Selling and Fulfillment Foundation の概要

ERP またはレガシー・アプリケーションを使用して拡張サプライ・チェーンを管理している企業は、ビジネス・パートナーとのコラボレーションを試みた時点で問題に直面します。多くの場合、使用するアプリケーションは自社の社内ビジネス・プロセスを自動化するために開発されていますが、この「拡張」環境で動作を管理するために設計されたプラットフォーム上では作成されていません。その結果、企業は、拡張サプライ・チェーンを管理するために、費用のかかる手動プロセスを使用するか、またはカスタム・アプリケーションを開発します。運用コストの 60% までは、会社外での決定により影響を受ける可能性があります。

IBM® Sterling Selling and Fulfillment Foundation IBM Sterling Application Platform は、この複雑な環境をさらに効率よく制御するために既存のエンタープライズ・アプリケーションの機能を有効に活用します。Sterling Selling and Fulfillment Foundation Sterling Application Platform は、既存システムの機能を拡張するか、または複数の社内アプリケーションを活用して、主要サプライ・チェーン・プロセスの統合管理と制御を提供できるようにします。Sterling Selling and Fulfillment Foundation Sterling Application Platform は、主要サプライ・チェーン・マネジメント・システムとしても使用されます。

---

## IBM Sterling Application Platform

Sterling Application Platform は、Sterling Selling and Fulfillment Foundation Sterling Application Platform の技術インフラストラクチャーを定義します。これには、基本レベルのカスタマイズが含まれており、Sterling Selling and Fulfillment Foundation Sterling Application Platform が他のシステムと相互運用できるようになります。Sterling Application Platform

---

## アプリケーション・コンソール Sterling Application Platform

アプリケーション・コンソール Sterling Application Platform 層は、XML/EDI、UI、およびワイヤレス表示、ポータルを作成によって、情報を可視化しています。これによって、エンタープライズ・サプライ・チェーンの大半に存在するアプリケーションと統合できるようになり、高効率な管理が可能になります。

---

## Sterling Selling and Fulfillment Foundation Sterling Application Platform

Sterling Selling and Fulfillment Foundation Sterling Application Platform は、拡張サプライ・チェーンに関するビジネス上の特定の問題に対応する複数の製品スイートに分かれています。顧客は、各自の固有のニーズに応じて、スイート・アプリケーションの一部またはすべてを使用します。

Sterling Selling and Fulfillment Foundation Sterling Application Platform は、ビジネス上の特定の問題に対応する、API、ビジネス・プロセス、およびユーザー・インターフェース (コンソール) のセットで定義されています。

Sterling Selling and Fulfillment Foundation Sterling Application Platform レイヤーは、次のアプリケーションとオプション・コンポーネントで構成されます。

- IBM Sterling Distributed Order Management
- IBM Sterling Supply Collaboration
- IBM Sterling Global Inventory Visibility
- IBM Sterling Logistics Management
- IBM Sterling Reverse Logistics
- カタログ管理
- IBM Sterling Warehouse Management System

## Sterling Distributed Order Management

Sterling Selling and Fulfillment Foundation Sterling Application Platform 分散オーダー管理アプリケーションは、すべてのタイプの顧客オーダー (customer order) (製品およびサービス) に高度に構成可能なオーダー管理機能を提供します。これは、すべてのチャネルからオーダーを集約、管理およびモニターし、拡張エンタープライズ間のフルフィルメント・プロセスを調整します。分散オーダー管理は、在庫状況をチェックし、すべての内部および外部のフルフィルメント・ロケーション間でルール・ベースの動的割り振りを提供します。さらに、クレジット、物流管理、設置などの重要なサード・パーティー・サービスを調整し、関連するすべての参加者間の共同実行を可能にします。また、単一オーダー・リポジトリーが提供され、顧客、チャネル、サプライヤー、および取引パートナーは、フルフィルメント・ライフサイクル全体にわたって、リアルタイムのオーダー情報にアクセスできます。分散オーダー管理は、単一インスタンスにおいて複数のオーダー・フルフィルメント・プロセスを処理する完全な柔軟性を提供し、イベント主導かつルール・ベースのオーダー調整によるオーダー・プロセスの動的バリエーションを処理します。

## Sterling Supply Collaboration

Sterling Supply Collaboration Sterling Application Platform アプリケーションは、複数の部門および複雑なサプライヤー・ネットワークを含む拡張エンタープライズ環境での計画済みオーダーおよび購入オーダーの集約、ルーティングおよびトラッキングを可能にします。供給コラボレーション・アプリケーションは、購入オーダーの割り当て方法、追跡方法、および管理方法に影響する、部門、サプライヤー、パートナーなどによるさまざまなビジネス・ルール (business rules) を調整します。また、役割ベースの可視性をすべての購入オーダー情報に対して可能にし、複数の異なる購入システムまたはロケーションから生成された購入オーダーに対する中央購入オーダー・リポジトリーとして作用します (購入システムまたはロケーションが外部の場合も同様)。これは、統合在庫管理 (Global Inventory Visibility) アプリケーションと密接にリンクして、インバウンド、購入中、または計画済み (スケジュール済み) の予定在庫の可視性を提供できます。

## Sterling Global Inventory Visibility

Sterling Global Inventory Visibility Sterling Application Platform アプリケーションは、複数のサイト、エンタープライズおよび参加者間で全体的な在庫を調整します。管理者は、内部および外部の出荷ノードで在庫を追跡できます。複数の要求（スケジュール、予定、見積、オーダーなど）を同期することによって、リアルタイムの在庫状況が提供され、タイプ（手持ちあり、インバウンド、購入中、スケジュール済みなど）が提供されます。このアプリケーションは、不足を特定し、販売オーダーの割り当て、購入の実行、または在庫の移動によって在庫残高を操作して在庫計画者が問題を解決できるようにします。データは、需給管理のために外部システム、顧客、サプライヤー、およびパートナーと共有できます。このアプリケーションは、拡張エンタープライズ環境のためにグローバル可視性、ATP/ATD、予約、および追跡を提供します。

## Sterling Logistics Management

Sterling Logistics Management Sterling Application Platform アプリケーションは、インバウンドまたはアウトバウンドの配達プロセスを管理および実行する機能を提供します。これは、商品の複雑な、マルチステップ、マルチレッグ、かつマルチモードの移動のための配送計画アカウンティング（マージン・トランジット、連続移動、レーン最適化、クロス・ドッキング（cross-docking）などの手法を含む）の実行を受諾して格納し、さらに管理も行います。これは、配達チェーンのすべてのパーティー間のすべてのアクティビティを調整し、イベントを予防的にモニターし、逸脱が発生した場合に参加者に通知します。出荷および配達のリコードは、オーダーと出荷の間の依存関係を管理するために元の販売オーダーまたは購入オーダーに結合されます。これは、実際の値と約束の値、SLA 測定基準分析、参加者のパフォーマンスなどを比較して、パフォーマンスの配達後調整を行えるようにします。

## Sterling Reverse Logistics

Sterling Reverse Logistics Sterling Application Platform アプリケーションは、条件ベースの返品処理（交換オーダー（exchange order）、改装および修理要請、返品処分など関連プロセスの実行および管理を含む）を提供します。連鎖オーダー機能があると、返品物流は、複数の返品または修理の要請を元の販売オーダーと結び付けて、修理ライフサイクルの追跡を提供できます。これは、返品ループを閉じ、適切なビジネス・ルール（business rules）に基づいて返品在庫追跡を適切なノードに戻します。また、返品受入、在庫処置を処理し、クレジット・プロセスを開始します。

## カタログ管理

Sterling Application Platform カatalog管理アプリケーションは、詳細な製品およびカタログ・データを、複数の部門、エンタープライズおよび参加者間で、集約して管理します。これは、共有およびコラボレーションをサポートする、マルチテナント管理ツールのような働きをします。これによって、分類、関連製品販売（cross-sell）、上位製品販売、代替品、その他の機能が可能になります。

## Sterling Warehouse Management System

Sterling Selling and Fulfillment Foundation Sterling Application Platform 倉庫管理アプリケーションは、大量ピック、梱包、個人別設定操作および出荷操作のためのスケラブル・ソリューションです。これは、分配の効率およびコスト効果を向上する

ように設計されており、輸送、作業場、および世界クラスの物流管理が含まれています。主要機能には、計画のツール、実行のツール、測定のツール、ならびにビジネス・ルール (business rule) ベースの操作が含まれています。

Sterling Warehouse Management System について詳しくは、「*Sterling Selling and Fulfillment Foundation: Warehouse Management System Concepts Guide*」を参照してください。

---

## Sterling Selling and Fulfillment Foundation Sterling Application Platform プロダクト・スイートの概要

Sterling Warehouse Management System Sterling Application Platform は、拡張サプライ・チェーンの具体的なビジネス問題を扱うプロダクト・スイートにグループ分けされます。顧客は、各自の固有のニーズに応じて、スイート・アプリケーションの一部またはすべてを使用します。これらのプロダクト・スイートには、以下のものがあります。

- 顧客フルフィルメント
- 供給コラボレーション
- サービス・パーツ物流管理
- サプライ・チェーン・イベント管理

IBM は、ターゲットに設定したビジネス問題または地域に対してプロダクト・スイートとしてデプロイされたそのアプリケーションを販売およびライセンス許諾します。それは、ビジネスが、最重要プロセスのいくつか (オーダーからキャッシュまで、購入から支払まで、およびアフター・マーケット・サービス管理) を管理するために Sterling Selling and Fulfillment Foundation Sterling Application Platform を使用する際に必要なプロダクトが 1 つのみということは、まずないためです。したがって、IBM では、今日のネットワーク化されたビジネス環境におけるこれらのプロセスの特定の課題に対処するための主要機能を実現するソリューション・エリアまたはプロダクト・スイートを売り込み、提示します。

### 顧客フルフィルメント・スイート

Sterling Selling and Fulfillment Foundation Sterling Application Platform の顧客フルフィルメント・プロダクト・スイートは、会社が顧客オーダー (customer order) の実現と関連付けられたすべてのサプライ・チェーン・プロセスを調整するのに役立ちます。複数の販売チャネル、複雑なフルフィルメント・プロセス、またはサード・パーティー・フルフィルメント・モデルを持つ会社には、このアプリケーションの機能は特に強力です。

顧客フルフィルメントは、複雑な拡張エンタープライズ環境における顧客オーダー (customer order) の管理と実現の要件に焦点を合わせています。この機能一式を使用すると、会社は、複雑なサプライ・チェーンを意識せずに、複数のチャネルまたは対話ポイントを通じて顧客に対して 1 つの面を提示することができます。これは、顧客オーダー (customer order) のライフサイクル全体を通じて、オーダー取り込み、約束、変更、割り当て、フルフィルメント、および配達など、すべてのフルフィルメント・アクティビティを調整します。Sterling Selling and Fulfillment

FoundationSterling Application Platform 顧客フルフィルメント・プロダクト・スイートには、以下のプロダクトが含まれています。

- Sterling Application Platform
- サプライ・チェーン・イベント管理
- Sterling Distributed Order Management
- Sterling Global Inventory Visibility
- Sterling Reverse Logistics
- Sterling Logistics Management

## 供給コラボレーション・スイート

Sterling Supply Collaboration Sterling Application Platform 製品スイートでは、会社は、複数の内部ビジネス・ユニットおよび部門間のインバウンド・サプライ・プロセスの調整が可能になります。これは、会社が所有または制御していない分配ロケーションの補充プロセスの制御の向上にも役立ちます。

供給コラボレーションは、複数のビジネス・ユニット、部門、サプライヤー、外部製造業者、および輸送業者間での直接材料の供給計画、購入オーダー、補充、在庫、およびインバウンド・フルフィルメントの調整と共同実行の要件を重点的に扱います。このスイートは、内部システムまたは部門間での供給要求に対する共同実行、サプライヤー間でのオーダーのソーシング、ならびに在庫およびインバウンド・フルフィルメントへの全体的な可視性の提供の中心点を提供します。Sterling Supply Collaboration Sterling Application Platform 製品スイートには、以下の製品が含まれています。

- Sterling Application Platform
- サプライ・チェーン・イベント管理
- Sterling Supply Collaboration
- Sterling Logistics Management
- Sterling Global Inventory Visibility

## Service Parts Logistics スイート

Sterling Selling and Fulfillment FoundationSterling Application Platform Service Logistics 製品スイートを使用することで、会社は、初めて製品を顧客に販売した後も、顧客の収益と顧客の維持を最大化できるように関連付けて、物流プロセスを効率化することができます。これらのスイートは、複雑なサービス部品ネットワーク全体にわたって、重要な実行機能およびイベント・ドリブンの例外管理機能を提供します。

Sterling Selling and Fulfillment FoundationSterling Application Platform Service Parts Logistics 製品スイートは、返品、スベア管理、および保守/サービス・オーダー管理をはじめとしたアフター・サービスの運用に焦点を当てています。このスイートは統合在庫管理 (Global Inventory Visibility) を利用して、サービス・センター、店頭在庫ロケーション、販売業者などの複雑なネットワーク全体にわたって在庫の可視性を管理および維持します。さらに、サード・パーティーによって配達されるサービスの調整、および完全な返品物流プロセス (返品、交換、整備、および処分を含

む) の実行にも焦点を当てています。Sterling Selling and Fulfillment Foundation Sterling Application Platform Service Parts Logistics 製品スイートには次の製品が含まれています。

- Sterling Application Platform
- Supply Chain Event Management
- Sterling Distributed Order Management
- Sterling Global Inventory Visibility
- Sterling Reverse Logistics
- Sterling Logistics Management

## サプライ・チェーン・イベント管理スイート

イベント管理 Sterling Application Platform 製品スイートは、サプライ・チェーン・イベント管理に特化したソリューションで、アプリケーション・プラットフォーム Sterling Application Platform (プロセス構成、イベント・エンジン、ステータス・モニタリング)、分析、包括的なサプライ・チェーンの可視性の例外コンソール (オーダー、在庫、出荷、返品など)、例外処理の各機能を利用します。

イベント管理 Sterling Application Platform 製品スイートは、完全に統合されたイベント構成、ステータスおよびイベントのモニタリング、さらにアラート処理機能を提供します。これは、フルフィルメント、在庫管理、購入などの重要なサプライ・チェーン・プロセス内のトランザクション・アクティビティを推進する条件、つまり「イベント」の設定およびモニタリングのための基盤となるメカニズムを提供します。Sterling Selling and Fulfillment Foundation Sterling Application Platform に関連してデプロイされた場合、事前定義され、ハードコーディング済みのアプリケーション手順内ではなく、イベント発生 (顧客の製品の製造を開始するための外回りの営業担当者からの「プル・シグナル」、契約に基づくスケジュール済みのリリースに関する合図など) に基づいて、イベント・エンジンがプロセスをモデル化し、管理できます。例外は、フル・トラッキング、自動エスカレーション、および完全な監査履歴を使用しての解決による構成可能な例外コンソールを使用した場合と同様、自動で処理できます。

イベント管理 Sterling Application Platform 製品スイートには、アプリケーション・プラットフォームおよびサプライ・チェーン・イベント管理ライセンス付き製品がターゲットのサプライ・チェーン・プロセスに必要なアプリケーション (オーダーの可視性のための分散オーダー管理、在庫の可視性のための統合在庫管理など) とともに含まれています。これは、実際には、重要なプロセスのモニタリングおよび通知のみならず「制御」も可能にすることによって、サプライ・チェーン・イベント管理を差別化するアプリケーションとのリンクです。

---

## Sterling Selling and Fulfillment Foundation のドキュメンテーション

Sterling Selling and Fulfillment Foundation をインストールまたはアップグレードする際に、製品ドキュメンテーションのオンライン・ドキュメンテーション・ライブラリーまたはローカル・ドキュメンテーション・ライブラリーのどちらを有効にするかを選択できます。ここで選択した内容は、`sandbox.cfg` の `ACTIVE_HELP_URL`

プロパティを使用することで、インストール後でも変更できます。方法については、「*Sterling Selling and Fulfillment Foundation* プロパティ・ガイド」を参照してください。

オンラインおよびローカル・ドキュメンテーション・ライブラリーではそれぞれ次の機能が提供されます。

- オンライン・ドキュメンテーション・ライブラリー - このライブラリーは IBM によってホストされ、以下を提供します。
  - HTML 形式および PDF 形式の *Sterling Selling and Fulfillment Foundation* ドキュメンテーション・セットへのオンライン・アクセス
  - ドキュメンテーション・セット全体の Google Mini 検索機能
  - 変更およびフィックスパックのドキュメンテーションの動的な更新
- ローカル・ドキュメンテーション・ライブラリー - このライブラリーはエンタープライズによってローカルにホストされ、以下を提供します。
  - HTML 形式の *Sterling Selling and Fulfillment Foundation* ドキュメンテーション・セットへのローカル・アクセス
  - ブック・ベースの索引および検索機能
  - 変更およびフィックスパックの更新はなし (製品のアップグレードによるものを除く)

## コンテキスト依存ライブラリーとコンテキスト・ヘルプの関係

オンラインおよびローカルのドキュメンテーション・ライブラリーは、ユーザー・インターフェースの「ヘルプ」ボタンからアクセスできるコンテキスト・ヘルプとは異なります。コンテキスト・ヘルプは、1 ページのヘルプ・ドキュメンテーションを提供する一方、オンラインおよびローカルのドキュメンテーション・ライブラリーは、ドキュメンテーション・セット全体を提供します。

コンテキスト・ヘルプの使用可能化およびドキュメンテーション・ライブラリーについて詳しくは、「*Sterling Selling and Fulfillment Foundation* インストール・ガイド」および「*Sterling Selling and Fulfillment Foundation* プロパティ・ガイド」を参照してください。





---

## 第 2 章 Sterling Application Platform のテクノロジーの概念

---

### Sterling Application Platform とは

Sterling Application Platform は、ビジネス・トランザクションのスムーズなフローをサポートし、可能にする技術的基盤であり、フレームワークです。

これは、Java、Enterprise Java Beans (EJB)、Java Messaging Services (JMS)、Java Management Extensions (JMX) および Extensible Markup Language (XML) などの相互運用性を可能にする最新のテクノロジーおよび標準を使用します。Sterling Selling and Fulfillment Foundation Sterling Application Platform は、複雑な環境でのデプロイメントおよび変化するビジネス・プロセスへの適応性に関して、最高の柔軟性を提供するように設計されています。また、会社が拡張されたエンタープライズ環境でビジネスを実行できるフレームワークが提供されます。

---

### Sterling Application Platform フレームワーク

Sterling Application Platform フレームワークは、拡張容易性および高可用性の要件に適合するエンタープライズ規模のアプリケーションをデプロイできるようにします。これには、アプリケーション・コンポーネントの有効なモニターおよび管理のためのツールが用意されています。重大なセキュリティーの機能、国際化対応とローカリゼーションの機能、およびシステム管理機能を使用して拡張エンタープライズ・ネットワークを管理するための機能が提供されます。

このアプリケーション・フレームワークは、また、Sterling Selling and Fulfillment Foundation Sterling Application Platform データ・モデルを拡張したり、お客様実装の固有のデータ・ニーズをサポートしたりできるようにします。

Sterling Selling and Fulfillment Foundation Sterling Application Platform の拡張は、ユーザー・インターフェースの表示中の要素の変更、印刷されるドキュメントの変更、またはデータベース表への列の追加によって行うことができます。

---

### ユーザー・インターフェースの拡張性

ユーザー・インターフェースの拡張性により、情報が機能する方法を変更せずにそのレンダリング方法、つまり表示方法を変更することができます。Sterling Selling and Fulfillment Foundation Sterling Application Platform ユーザー・インターフェースを構成する Sterling Application Platform アプリケーション・コンソールおよび Sterling Application Platform アプリケーション・マネージャーの外観をカスタマイズできます。

- アプリケーション・コンソール Sterling Application Platform - オーダー、アイテム在庫、および返品を作成、追跡、および表示する標準 GUI
- Sterling Application Platform アプリケーション・マネージャー - Sterling Selling and Fulfillment Foundation を構成するグラフィカル・ユーザー・インターフェース (graphical user interface)

これは、Sterling Application Platform アプリケーション・マネージャーを使用した構成変更と JSP ファイルに対する HTML コード変更の組み合わせによって実現されます。

また、印刷されるドキュメントの外観も拡張できます。

## アプリケーション・コンソール・ユーザー・インターフェース拡張

Sterling Selling and Fulfillment Foundation Sterling Application Platform Presentation Framework によって、標準のアプリケーション・コンソールのユーザー・インターフェースを拡張できます。

標準のアプリケーション Sterling Application Platform コンソールの、ユーザー・インターフェースは、Java Server Pages に埋め込まれた HTML を使用しています。この UI 層は、サービス定義フレームワークを介して API にアクセスします。これにより、公開された API のみを確実に使用できます。

サービス定義フレームワークのユーザー・インターフェース層で使用される XML 操作は、最小限に抑えられます。XML 出力の大量の操作が必要となる個所では、API に対する変更により、使いやすいユーザー・インターフェースで出力が可能になります。

Sterling Selling and Fulfillment Foundation Sterling Application Platform Presentation Framework によって、標準のアプリケーション Sterling Application Platform コンソールのユーザー・インターフェースを、以下の方法で拡張できます。

- Sterling Selling and Fulfillment Foundation Sterling Application Platform サインイン画面 (Sterling Selling and Fulfillment Foundation Sterling Application Platform の始動時にユーザーに最初に表示されるページ) のカスタマイズ - 以下のとおり、カスタマイズできます。
  - 画面上の文字列を表示する言語を決定するためにロケールを設定
  - ロケール固有のログインを構成 - 複数言語のユーザー・コミュニティーの場合、すべての可能な言語でログイン・ページを表示するのが適切です。
  - 企業のルック・アンド・フィールの設定 - サインイン画面の 2 次的な目的は、アプリケーション全体に通じる企業のルック・アンド・フィールを紹介することです。
  - 外部アプリケーションからサインイン - Sterling Selling and Fulfillment Foundation Sterling Application Platform を外部アプリケーションと統合する際、ユーザーを外部アプリケーションから Sterling Selling and Fulfillment Foundation Sterling Application Platform に自動的にログインさせることができます。
  - シングル・サインオンのサポート - ユーザーは、ドメイン・パスワードを使用して、Sterling Selling and Fulfillment Foundation Sterling Application Platform に透過的にログインできます。Sterling Selling and Fulfillment Foundation Sterling Application Platform は、サード・パーティーのシングル・サインオン・アプリケーションをサポートします。
- 企業ブランドの追加 - 以下のロケーションの画面上に表示されるブランド・ロゴを変更できます。
  - ログイン画面 (サインイン・ウィンドウ)

- メニュー・バー
- アバウト・ボックス
- テーマの定義 - テーマはアプリケーションのルック・アンド・フィールを定義します (フォントおよび色など)。
- ビューのカスタマイズ - 以下のタイプのビューの画面レイアウトおよび編成。
  - 検索ビュー
  - リスト・ビュー
  - 詳細ビュー
- ビジネス・エンティティのカスタマイズ
- ルックアップの追加 - ユーザーがデータを入力せず、オプションの分類から選択できるようにする、ルックアップを追加できます。
- グラフおよび円グラフの追加 - グラフおよび円グラフによって、データをグラフィカルに表示できます。
- メニュー構造のカスタマイズ - カスタマイズされた画面を作成するとき、ユーザーがメニュー構造またはナビゲーションのいずれかの方法で、その画面に必ずアクセスできるようにする必要があります。
- Sterling Selling and Fulfillment Foundation Sterling Application Platform の画像のカスタマイズ - Sterling Selling and Fulfillment Foundation Sterling Application Platform 全体を通じて、以下の場所に表示される画像をカスタマイズできます。
  - メニュー・バー
  - メニュー
  - アプリケーション・コンソール Sterling Application Platform
- 画面ナビゲーションのカスタマイズ - リンクまたはアクション・リソースを構成して、ユーザーのエンティティからエンティティへのナビゲーションをカスタマイズすることができます。
- カスタム・イベント・ハンドラーの作成 - コントロールまたは画面レベルでユーザー・インターフェース制御に対するカスタム・クライアント・サイド検証を作成しプラグインできます。
- カスタム伝票種別用の画面を作成
- カスタム・トランザクションの作成 - カスタムのオーダーまたは配達トランザクションを作成し、これらのトランザクションを含めてステータス変更を行うようにパイプラインを構成できます。
- XML バインディング - 開発者は、API に渡す必要のある入力を容易に形成し、API の出力を画面に取り込むことができます。

## Applications Manager の拡張

Sterling Application Platform Applications Manager ユーザー・インターフェースは、Swing で開発されました。ユーザー・インターフェース拡張の主要な目的は、あらゆるデータベース拡張機能をグラフィカル・ユーザー・インターフェース (graphical user interface) に統合できるようにすることです。

拡張には以下の変更が含まれます。

- ボタンおよびラベルの追加

- テキスト・フィールドおよびチェック・ボックスの追加
- 必須ではないコンポーネントの非表示
- 画面上に表示されるコンポーネントの再編成
- ユーザー・テーマの作成または変更

以下のタイプの画面を変更できます。

- 検索画面
- 詳細画面
- リスト画面 - Applications Manager 内のリスト画面で列を追加、削除および再編成できます。

ツリー構造しか含まない画面を拡張することはできません。

---

## データベースの拡張

データベース拡張により、追加のデータを取り込むために、Sterling Selling and Fulfillment Foundation Sterling Application Platform テーブルに列を追加できます。1つ以上の列をテーブルに追加することにより、特定の Sterling Selling and Fulfillment Foundation Sterling Application Platform テーブルを拡張できます。

また、特殊なオーダーに関する動的に変化するフィールドが存在する状況が発生する場合があります。Sterling Selling and Fulfillment Foundation Sterling Application Platform では、オーダーまたはオーダー明細関係の属性を追加することにより、システム上で取り込まれたオーダー・レベルおよび明細レベルの情報を拡張できます。これにより、会社固有のフラット・データおよび階層データを取り込むことができます。Sterling Selling and Fulfillment Foundation API により、データ・モデルに対するすべての変更が自動的に Sterling Selling and Fulfillment Foundation に組み込まれるように、そのようなデータに透過的にアクセスできるようになります。これらの属性は、デフォルトのユーザー・インターフェースでは表示されず、検索不能です。しかし、そのようなフィールドを含むようにユーザー・インターフェースをカスタマイズすることができます。

---

## セキュリティ管理

「セキュリティ管理」を使用することで、各ユーザーが自身のタスクの遂行に適した情報、および各自が属する組織によって提供されたりソースにのみアクセスできるようにすることができます。ユーザーは、許可されているリソースのみにアクセスを制限されます。

インストール・レベルのセキュリティについて詳しくは、「*Sterling Selling and Fulfillment Foundation* インストール・ガイド」を参照してください。

### ユーザー

ユーザー とは、組織内で果たす役割に応じて、ハブ管理者や顧客サービス担当者 (CSR) (Customer Service Representative (CSR)) などの特定のタスクが割り当てられた単一の個人です。各組織には固有のユーザーがいます。

## ユーザー・グループ

ユーザー・グループとは、類似するタスクを実行するユーザーの集まりです。例えば、顧客サービス担当者 (Customer Service Representatives) のグループは、まとめて CSR ユーザー・グループに配置することができます。ユーザーは複数のユーザー・グループに属することができます。ユーザー・グループにはアクセス権が割り当てられます。ユーザーは、自分が属する各ユーザー・グループのアクセス権すべてを保持します。

各組織には固有のユーザー・グループがあります。ユーザー・グループには、ユーザーが作成された対象の組織と同じ組織のユーザーのみを含めることができます。ただし、ハブ組織により作成されたユーザー・グループの場合を除きます。これには、あらゆる組織のユーザーを含めることができます。

## データ・セキュリティー・グループ

データ・セキュリティー・グループとは、特定の伝票種別 (document type)、エンタープライズ、および出荷ノードへのアクセスを制御するユーザー・グループです。データ・セキュリティー・グループに関連付けられていない場合、そのユーザーは、デフォルトのアクセス権を持っているとみなされます。データ・セキュリティー・グループを定義することで、エンタープライズ、伝票種別 (document type)、またはデフォルトのアクセス・リストのサブセットである出荷ノードへのアクセスをさらに制限できます。

データ・セキュリティー・グループに関連付けられていない場合、そのユーザーは、最も制限度が低いアクセスまたはデフォルト・アクセスを持っていると見なされます。

## チーム

チームは、共通のデータ・アクセス要件を持つユーザーの集合です。チームは、特定の伝票種別 (document type)、エンタープライズ、出荷ノード、および顧客にアクセスできます。チームに関連付けられていない場合、そのユーザーは、最も制限度が低いアクセス権またはデフォルト・アクセス権を持っているとみなされます。チームを定義することで、エンタープライズ、伝票種別 (document type)、またはデフォルトのアクセス・リストのサブセットである出荷ノードへのアクセスをさらに制限できます。チームと、それらを Sterling Selling and Fulfillment Foundation Sterling Application Platform でどのように使用できるかについて詳しくは、35 ページの『顧客管理』を参照してください。

## リソースおよびリソース・アクセス権

Sterling Selling and Fulfillment Foundation Sterling Application Platform は、スクリーン、関数、URL などを含む多くのリソースで構成されており、これらに対してアクセス権を認可したり取り消したりすることができます。

セキュリティー管理により、あらかじめ決定されたリソースに対するアクセス権限がユーザーに認可されます。ユーザーは、自分がメンバーであるユーザー・グループの少なくとも 1 つに許可されているリソースのみにアクセスを制限されます。

---

## ログ・ファイルの機密情報のセキュリティー

ログ・ファイルの機密情報をマスクする方法について詳しくは、「*Sterling Selling and Fulfillment Foundation* プロパティ・ガイド」を参照してください。

---

## データ・アクセス・ポリシー

固有のアクセス・ルールを使用する場合は、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

### アクセス・ポリシー・モード

Sterling Selling and Fulfillment Foundation アクセス・ルールは、以下のアクセス・ポリシー・モードを介して適用されます。

- エンタープライズ・ユーザー - このユーザーは、ビジネスを仲介するエンタープライズに属します。各エンタープライズには、さまざまな役割を割り当てられた複数の組織を含むことができます。
- バイヤー・ユーザー (Buyer User) - このユーザーは、エンタープライズまたはその他のセラー組織から製品を購入する組織に属します。
- セラー・ユーザー - このユーザーは、エンタープライズまたはその他のバイヤー組織に製品を供給する組織に属します。
- ノード・ユーザー - このユーザーは、物理ロケーション (例えば、製造工場、小規模倉庫、または倉庫など) を表す組織に属します。

ユーザー・アクセス・ポリシー・モードに加えて、Sterling Selling and Fulfillment Foundation には以下があります。

### カタログ・アクセス・ポリシー

カタログ・アクセス・ポリシーは、ユーザーが、アイテム、カテゴリおよび属性などの、カタログ・データを管理 している場合に適用されます。ユーザーがカタログ・データを表示 するだけの場合は、カタログ・アクセス・ポリシーは適用されません。

カタログ・アクセス・ポリシーは、ユーザーがエンタープライズ・ユーザーまたはカタログ組織 (catalog organization) ユーザーの場合にのみ、適用されます。

カタログ組織 (catalog organization) ユーザーは、ユーザー自身の組織またはユーザーの組織によって管理されている組織のデータを管理できます。ただし、カタログ組織 (catalog organization) ユーザーは、サブカタログ組織 (subcatalog organization) のデータを管理することはできません。同様に、サブカタログ組織 (subcatalog organization) ユーザーは、ユーザー自身の組織またはユーザーの組織によって管理されている組織のデータを管理できます。

例えば、アイテムがカタログ組織 (catalog organization) によって所有されている場合 (つまり、アイテムにサブカタログ組織 (subcatalog organization) がない場合)、そのアイテムはサブカタログ組織 (subcatalog organization) ユーザーによって管理することができません。しかし、アイテムがサブカタログ組織 (subcatalog organization) によって所有されている場合 (つまり、アイテムにサブカタログ組織 (subcatalog

organization) がある場合)、そのアイテムはサブカタログ組織 (subcatalog organization) ユーザーによってのみ管理することができます。

また、アイテム資格も、目的に基づいてアクセス・ポリシーに適用できます。アイテム資格の場合、ユーザーがアイテム資格を表示 するだけの場合でも、アクセス・ポリシーが適用されます。例えば、購入資格 (購入が目的) は、エンタープライズ・ユーザーによってのみ、表示および管理できます。販売資格 (販売が目的) は、カタログ組織 (catalog organization) ユーザーによってのみ、表示および管理できます。

## 価格設定アクセス・ポリシー

価格設定アクセス・ポリシーは、ユーザーがエンタープライズ・ユーザー、セラー、または価格設定組織のユーザーである場合にのみ適用されます。これらの役割を持つすべてのユーザーは、ユーザー自身の組織の価格設定データまたはユーザーの組織により管理される組織の価格設定データを管理できます。

ユーザーはユーザーの価格設定組織により定義されたすべての価格リストを表示することができますが、変更できるのはユーザー自身の価格設定組織の価格リストのみです。例えば、XYZ-MA ユーザーおよび XYZ-CA ユーザーが価格設定組織として XYZ-MA を使用している場合、XYZ-CA ユーザーは、XYZ-MA により定義されたすべての価格リストを表示することができますが、これらの価格リストを変更できるのは XYZ-MA ユーザーのみです。

アクセス・ポリシーは、トランザクション・データおよびマスター・データだけに適用されます。アクセス・ポリシーは、組織およびユーザーなどの構成データには適用されません。

## ユーザー・アクセスのレベル

ユーザーは前述のユーザーの役割のいずれかでアプリケーションにログインできます。例えば、バイヤーとしてログインして販売オーダーを表示したり、セラーとしてログインして購入オーダーを表示したりすることができます。社内のストアフロント管理者用にエンタープライズ・ユーザー・アクセス権限を構成したり、組織内の全ノードを表示する必要がある倉庫管理者用にノード・ユーザー・アクセス権限を構成したりすることができます。

特定のチームに割り当てられた全顧客へのアクセスや、特定のユーザーに直接割り当てられた顧客のみへのアクセスなど、それぞれのデータ・アクセス・モードで、ユーザー・アクセスのレベルを柔軟に提供できます。アクセス権限は、アクセス・モードに応じて下位のチームや下位の顧客に拡張することもできます。

16 ページの図 1 は、CSR を特定のアカウントに割り当てるのではなく、チーム全体をアカウントに割り当てる企業の例を示しています。チームについて詳しくは、13 ページの『チーム』を参照してください。

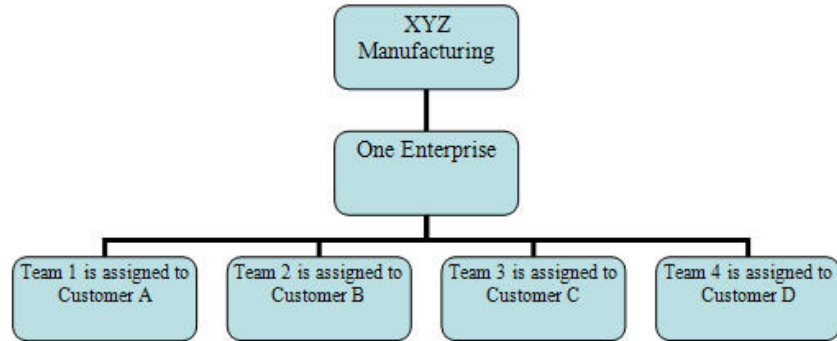


図1. チーム・アカウントのデータ・アクセス・シナリオ

図2 は、2つのエンタープライズ（1つは小売、もう1つは中小規模の企業）がある企業の例を示しています。これら2つのエンタープライズのそれぞれに個別のCSR チームがあります。これらのデータ・アクセス機能は混合して使用され、小売エンタープライズ E1 にはすべてのエンタープライズが表示され、もう一方のエンタープライズ E2 は、CSR を特定のアカウントに割り当てる必要があります。

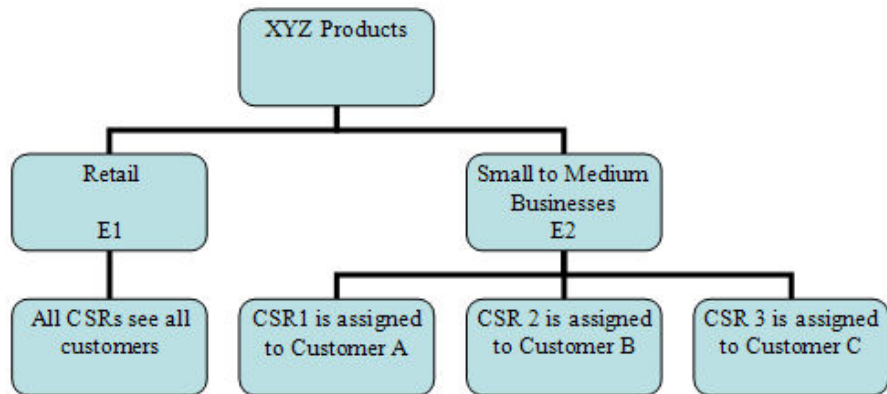


図2. 混合モードのデータ・アクセス・シナリオ

17 ページの図3 は、HIJ Telephone Corporation から電話機器を購入するソフト・ドリンク企業の組織構造を示しています。このパイヤー・シナリオで、SODA 飲料事業部 (3) は、事業部自体、およびその下位組織であるソフト・ドリンク (6)、フルーツ・ドリンク (7)、およびエネルギー・ドリンク (8) の電話機器を購入できます。また、データ・アクセス機能を使用して、SODA 飲料事業部 (3) が電話を購入できる組織から下位組織の1つであるエネルギー・ドリンク (8) を除外することもできます。



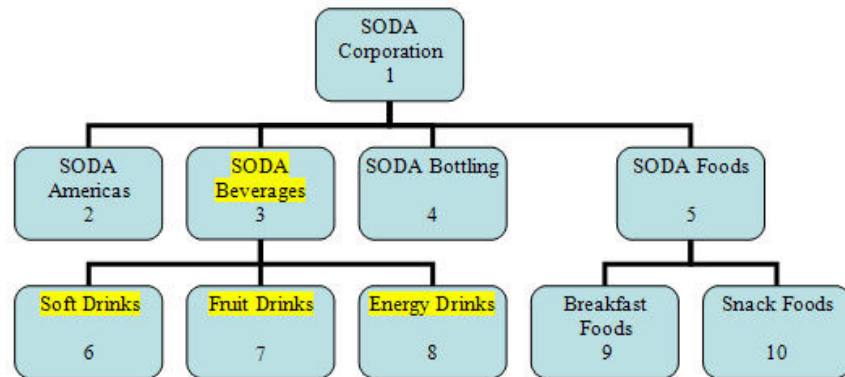


図3. バイヤーのデータ・アクセス・シナリオ

データ・アクセス・ポリシーはすべて、Applications Manager の Sterling Application Platform で構成できます。

図4 は、オフィス用品会社の顧客である XYZ Corporation の組織構造を示しています。東部、中部、西部の事業所は XYZ Corporation の子組織 (child organization) で、これらの事業所もオフィス用品会社の顧客です。このバイヤーのシナリオでは、バイヤー・ユーザー (Buyer User) 1 が東部および西部の事業所に、バイヤー・ユーザー (Buyer User) 2 が中部および西部の事業所に割り当てられています。バイヤー・ユーザー (buyer user) のアクセス・ポリシーでは、割り当てられた顧客のみに対するアクセスが提供されるように構成されています。このため、バイヤー 1 は、東部および中部の事業所の用品を購入でき、バイヤー・ユーザー (Buyer User) 2 は中部および西部の事業所の用品を購入できます。

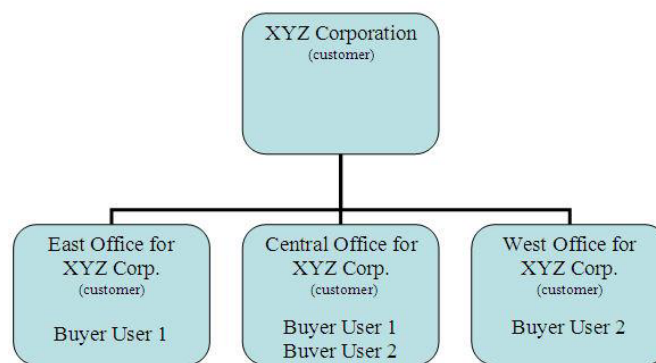


図4. バイヤー割り当てのデータ・アクセス・シナリオ

データ・アクセス・ポリシーはすべて、Applications Manager の Sterling Application Platform で構成できます。

---

## パスワード・ポリシー

パスワード・ポリシーでは、次の事項が広く管理されます。

- パスワード・ストレングス - これには、パスワードの長さ (最短および最長)、パスワード内の特殊文字、およびパスワードの再使用が含まれます。
- パスワードの生成 - これには、ユーザー登録時のパスワードの生成、パスワードの期限切れの頻度、失敗したログインの試行回数、パスワード・ポリシーに影響する可能性があるユーザーの役割が含まれます。
- パスワードのリセット - これには、E メールや SMS など、別のプロトコルを使用したパスワードのリセットが含まれます。

さらに、パスワード・ポリシーの構成によって、無効なログインの試行が反復された場合にユーザーのアクセスが拒否されます (ロックアウト)。

パスワード・ポリシーについては、「*Sterling Selling and Fulfillment Foundation: Password Policy Management*」を参照してください。

---

## パスワードの暗号化 (encryption)

Sterling Selling and Fulfillment Foundation データベースのパスワードは、ハッシュ・アルゴリズムで保護することによって潜在的なアタッカーから自動的に保護されます。

システムでは、最初にランダム文字のストリング (ソルト) をパスワードのそれぞれに追加することによって、ユーザー・パスワードを保護します。パスワードはソルトの追加後にハッシュされ、データベースで保管されます。システムでは、パスワードのそれぞれを、16 バイトのソルト (ランダム・データ) を追加した SHA-256 ハッシュで保管します。

YCPValidateChangedPasswordUE および YCPCheckPasswordsMatchUE の外部プログラムを実装して、ハッシュ化ロジックをカスタマイズすることもできます。

---

## JBoss に対する JMS セキュリティー

JBoss JMS キューを保護することによって、アプリケーション・サーバー上のキューに保管されたデータにセキュリティを提供します。JBoss に対して Sterling Selling and Fulfillment Foundation は、JMS キュー・ベースのセキュリティのみを提供します。保護された JMS キューへの接続時には、ユーザー名とパスワードを渡して認証を受ける必要があります。

JBoss に対する JMS セキュリティーは、Application Manager で使用可能な JBoss Default Messaging JMS キュー転送ノードをセットアップすることによって構成できます。JBoss Default Messaging JMS Queue ノードについて詳しくは、*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイドを参照してください。

さらに、JBoss アプリケーション・サーバー上で JMS キュー (JMS queue) を保護する必要もあります。

---

## API のセキュリティー

Sterling Selling and Fulfillment Foundation は、指定されたユーザー ID、証明書またはその両方の API 呼び出しについて、認証を提供します。これによって、API が呼び出されたときにアクセスできる API リソースを制御できます。

API を呼び出すときに、以下の 2 つのレベルのセキュリティーを提供できます。

- ユーザー ID、証明書、またはその両方による確認。ログイン API は、他のいずれかの API が呼び出される前に呼び出されます。ユーザー認証は、API 呼び出しの前に行われます。アクセスは、指定されたユーザーについて、API またはサービスのサブセットに制限することができます。
- 承認。これはユーザーがアクセスできるリソースを確認します。承認によって、そのユーザーは API およびサービスのサブセットに制限されます。この承認は、API の呼び出し元に関係なく、すべての API について、API レベルで行われます。この承認は、UI リソース・セキュリティーから完全に独立しています。例えば、ユーザーが UI を介して、ユーザーをリストする画面にアクセスしているとします。画面は、getUserList API を呼び出します。そのユーザーに対して画面が正常に機能するためには、getUserList API へのアクセス権限が必要です。

API セキュリティーの構成方法について詳しくは、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

---

## システム管理

Sterling Selling and Fulfillment Foundation Sterling Application Platform では、統合サーバーおよびエージェント・サーバーの管理、アプリケーション・サーバーのプロパティーの表示、データベース・キャッシュの有効化、API、外部プログラム、サービス、およびエージェントのトレース・ログ出力の増加を実行できます。

システム管理モジュールには、Sterling Selling and Fulfillment Foundation Sterling Application Platform を構成するさまざまなコンポーネントの管理およびモニターを行う機能が用意されています。システム管理コンソールは、実行中に Sterling Selling and Fulfillment Foundation Sterling Application Platform の完全なピクチャーを提供します。また、ヘルス・モニター・エージェントは、アプリケーション・サーバーの故障やエージェント・サーバーがタスクを処理しないなどの問題が発生したときに、システム管理者に警告できます。

さらに、アプリケーション・マネージャーが Visual Modeler アプリケーションからアイテムのモデル情報を取得できるようにするために、モデル、プロパティー、およびルールが格納されているロケーションを指定できます。

他に以下の機能があります。

- Sterling Selling and Fulfillment Foundation Sterling Application Platform で起こっていることをどんなときでも理解するためのリアルタイム統計情報の表示
- アプリケーション・サーバーの状況および応答時間のモニター
- API の応答時間のモニター

- ボトルネックが作成されていないことを確認するためのエージェントの状況および保留タスクの数のモニター
- メッセージが正しく処理されていることを確認するための JMS キュー (JMS queue) のモニター
- エージェントおよび統合サーバーのシャットダウン、中断、または再開
- データベース・キャッシュのクリア
- API、エージェント、外部プログラム、サービス、およびアプリケーション・コンソールのトレース
- 統計データの収集および維持

---

## キュー管理

「**キュー管理**」では、システムで警告が発生した際にユーザーに通知するように、キューを構成することができます。これらの警告は、E メール、ファクシミリなどのさまざまな形式で受け取ることができます。警告は、警告のタイプに応じて異なるキューに送信されます。

警告は、手操作による介入を必要とするトランザクションに関する、ユーザーまたはキューに送信されるメッセージです。

ユーザーに警告を配布するために、「**キュー**」を構成します。ユーザーをキューに割り当てることで、どのユーザーがそれぞれの警告タイプを受け取るかを決定します。さらに、警告の特定の条件が満たされた場合に出される警告の優先順位とアクションをセットアップすることもできます。

Sterling Selling and Fulfillment Foundation Sterling Application Platform では、警告の管理、警告の優先順位の設定、システムによる警告の自動的なユーザーへの割り当て、警告の 1 つ以上のタイプへの分類、警告を警告タイプ別に特定のキューに送付、警告へのメモの追加、コンテキストに依存した警告解決画面の指定などを行うことができます。

システムが自動的に警告をユーザーに割り当てるようにすることができます。これにより、誰かが出向いて手動で警告を割り当てる必要がなくなります。警告を 1 つ以上のタイプに分類して、これらのタイプを次の目的で使用できます。

- 警告にデフォルトの属性を設定する (デフォルトのフォロー・アップ時間や優先順位など)。
- コンテキストに依存した警告解決画面を指定する。

Sterling Application Platform Applications Manager の警告の構成 (configuring alerts) コンポーネントで、さまざまな警告タイプに対して、優先順位、キュー、有効な日数、フォロー・アップ時間、解決フォーム、およびユーザーを指定できます。警告サービスの構成については、「*Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド*」を参照してください。

## 警告の集約

警告は、異なる状況下で発行可能で、多くの場合、新しく出された警告の内容は、既存の警告と同じです。例えば、モニターは、オーダー条件に基づいて警告をトリ

ガーします。オーダーに一定期間変化がないと、もう一度警告が発行されます。警告の集約手法を使用して、同じ警告を大量に保有することなく、警告を集約できます。

サービス定義フレームワークの警告コンポーネントで、警告の集約を有効にできます。この集約は、API、サービス・ビルダーおよびエラー処理など、警告が発行される場所に関係なく、どこでも使用可能です。警告の集約の有効化について詳しくは、「*Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド*」を参照してください。

警告が発行され、集約が有効になっている場合、同じ情報を持つ警告が他にないか検索するため、データベースに対して照会が行われます。そのような警告が存在する場合、既存の警告で集約回数が加算され、新しいレコードはデータベースに挿入されません。

集約に使用される属性を識別するために、呼び出される API に対する入力の一部として集約テンプレートを渡すことができます。このテンプレートには、集約前の入力 XML およびデータベース内のものと同じでなければならぬすべての属性が含まれています。テンプレートが渡されない場合は、デフォルトのテンプレートが使用されます。これは、多くの受信ボックス属性、およびすべての拡張属性から構成されています。このデフォルトのテンプレートは、`<INSTALL_DIR>/repository/xapi/template/merged/resource/exceptionConsolidation.xml` にあり、API テンプレートと同じ方法で拡張できます。API テンプレートの拡張については、「*Sterling Selling and Fulfillment Foundation データベースの拡張*」を参照してください。

集約ウィンドウは、Day または Hour で警告を集約するように設定されています。集約テンプレートの日付属性をデータベース値と比較するとき、それらが等しいかどうかを判別するために、この集約ウィンドウが使用されます。集約ウィンドウは、個別の期間内でレコードを突き合わせます。以下に、例とともに概要を示します。

### 集約 (時間別)

例えば、集約ウィンドウが HOUR に設定され、最後に発生した警告のタイム・スタンプが 2005-02-01、06:24 AM とします。同じ警告が、2005-02-01、7:10 AM に発行されたとします。7:10 AM に発行されたこの新しい警告は、間隔が 1 時間未満ではあるけれども、6:24 AM に発行された警告と集約されません。これらの警告を集約するには、同じ時間内に発行されなければならないためです。

### 集約 (日別)

例えば、集約ウィンドウが DAY に設定され、最後に発生した警告が 2005-02-01、07:40 PM とします。新しい警告が 2005-02-02 1:30 AM に発行されても、この警告は前の警告と集約されません。これらは別の日に発行されているためです。

**注:** 集約テンプレートには、時間以外の属性しか含まれていないため、集約ウィンドウ (時間別または日別) は集約プロセスに影響を与えません。

集約ウィンドウは、拡張フィールドも含めて、集約テンプレートに含まれるすべてのタイム・スタンプ・フィールドに適用されます。

警告の集約における前提および制約は、以下のとおりです。

- 警告が集約された場合、集約回数および警告の最終発生時刻以外の警告属性フィールドは更新されません。
- 同じ集約テンプレートと一致する警告が複数ある場合は、正確に一致したとみなされて選択された警告は未定義になります。
- 子テーブルまたは拡張ハングオフ・テーブルでは、警告は集約できません。
- データに基づいた条件付き集約を有効にすることはできません。例えば、EnterpriseKey が DEFAULT である場合にのみ集約することは選択できません。
- 解決した警告は集約に使用されません。

Sterling Application Platform Applications Manager の警告サービスで、警告について有効な日数を指定できます。有効な日数は、経過するこの例外が自動的にクローズされる、アクティビティーのない日数を表します。ゼロの値は、例外が有効期限切れにならないことを意味します。

---

## 印刷拡張性

Sterling Warehouse Management System では、Loftware 印刷サーバー・システムおよび Loftware ラベル・マネージャー・システムを構成してカスタムのラベルおよび印刷物を作成できます。印刷サービスは、イベントまたはコンソール・ユーザー・インターフェースから起動されるサービス定義を作成することによってカスタマイズできます。Sterling Warehouse Management System に用意されている標準ラベルは、Loftware ラベル・マネージャーと統合することによってカスタマイズできます。

このようなカスタマイズの実行については、「*Sterling Selling and Fulfillment Foundation API* のカスタマイズ」を参照してください。

---

## モバイル・アプリケーションの拡張性

Sterling Application Platform のモバイル・アプリケーションの拡張性を使用すると、倉庫作業で使用されるモバイル・デバイスのカスタム・ユーザー・インターフェースの開発および表示が可能になります。

Sterling Selling and Fulfillment Foundation Sterling Application Platform モバイル・ユーザー・インターフェースのカスタマイズをデプロイおよびテストするための開発環境を用意する必要があります。モバイル・アプリケーション・インターフェースのカスタマイズ手順については、*Sterling Selling and Fulfillment Foundation: Customizing User Interfaces for Mobile Devices* を参照してください。

## 警告管理

Sterling Selling and Fulfillment Foundation Sterling Application Platform では、以下を行うことができます。

- 警告の構成
- 新しい例外タイプの作成
- 例外タイプに優先順位を設定
- 各例外を 1 つ以上のタイプに分類
- 例外タイプを特定の警告キューにルーティング
- 高優先順位警告のしきい値を指定
- 例外タイプに有効な日数およびフォローアップ時間を指定
- そのタイプの例外を集約するかどうかを指定

例外を 1 つ以上のタイプに分類して、それらタイプを使用して以下を行うことができます。

- 例外タイプのデフォルト属性 (デフォルトのフォローアップ時間や優先順位など) を指定
- 例外タイプのコンテキスト依存の解決画面を指定
- 例外タイプの構成画面を指定
- 例外タイプのリスト画面を指定

例外タイプに、追加構成を定義できます。例えば、PCA はクレジット限度額などの追加情報を収集したい場合に、それに関する必要な情報の収集に使用する構成フォームを指定できます。また、PCA は、特定の例外タイプに定義された警告のリストを提示する、リスト・フォームを指定することもできます。アプリケーション (PCA など)、およびアプリケーションがハブ・レベルでサポートする役割 (BUYER または SELLER など) に例外タイプを関連付けることもできます。

組織に対しては、1 つ以上の例外タイプおよび優先順位を構成できます。例外タイプにルーティング・ルールを構成することによって、例外を適切な警告キューにルーティングできます。また、組織に対して構成された例外タイプごとに 1 つ以上のルーティング・ルールを作成することもできます。組織レベルの例外タイプにルーティング・ルールを作成する際、警告の発行に使用するルーティング・ロジックを指定できます。使用されるルーティング・ロジックに基づいて、複数の警告を発行できます。ルーティング・ロジックによって、例外を割り当てる必要のあるキューが判別されます。ルーティング・ルールは、ルーティングのタイプを示します。指定された例外タイプについて、異なるタイプのルーティング・ルールまたはロジックを使用できます。

以下のルーティング・ロジックは、すぐに使用可能な状態で提供されています。

表 1. ルーティング・ロジック

ルーティング・タイプ	説明
部門キューを使用	部門のデフォルトのキューを使用します。
指定のキューを使用	ルーティング・ルールに指定されたキューを使用します。

表 1. ルーティング・ロジック (続き)

ルーティング・タイプ	説明
インターフェースを使用してキューを取得	Java クラスを使用して、キューを取得します。
役割にルーティング	指定した役割を持つ組織のルーティング・ルールを使用します。(バイヤー、運送会社、エンタープライズ、ハブ (HUB)、ノード、セラー)
カスタム役割にルーティング	カスタム組織属性を使用してキューを決定します。
例外タイプのキューへのルーティング	例外タイプで指定されたキューを使用します。

組織レベルの例外タイプにルーティング・ルールを作成する際、指定された E メール・アドレスに E メールを送信するかどうか指定することもできます。ルーティング・タイプ「部門キューの使用」を使用していて、そのタイプの E メールを送信する必要がある場合、および E メール ID が例外ルーティング・ルールに指定されていない場合、システムは E メール ID について部門の構成を確認します。例外が部門のデフォルト・キューに作成されている場合、E メールは、その部門に対して構成された E メール・アドレスに送信されます。テンプレートが構成されていない場合、デフォルトのテンプレートが使用されます。

警告の構成について詳しくは、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。



## 第 3 章 参加者の管理

### 組織のモデル化

組織は、企業内の会社、部署、コスト・センター、部門、販売単位、およびその他の組織単位を表します。一般的な組織は次のようになります。

- ビジネス・グループ (Business Group)
- 会社
- 法人組織
- セールス組織
- 購入組織
- 工場および倉庫

図 5は、一般的な組織モデルを表しています。

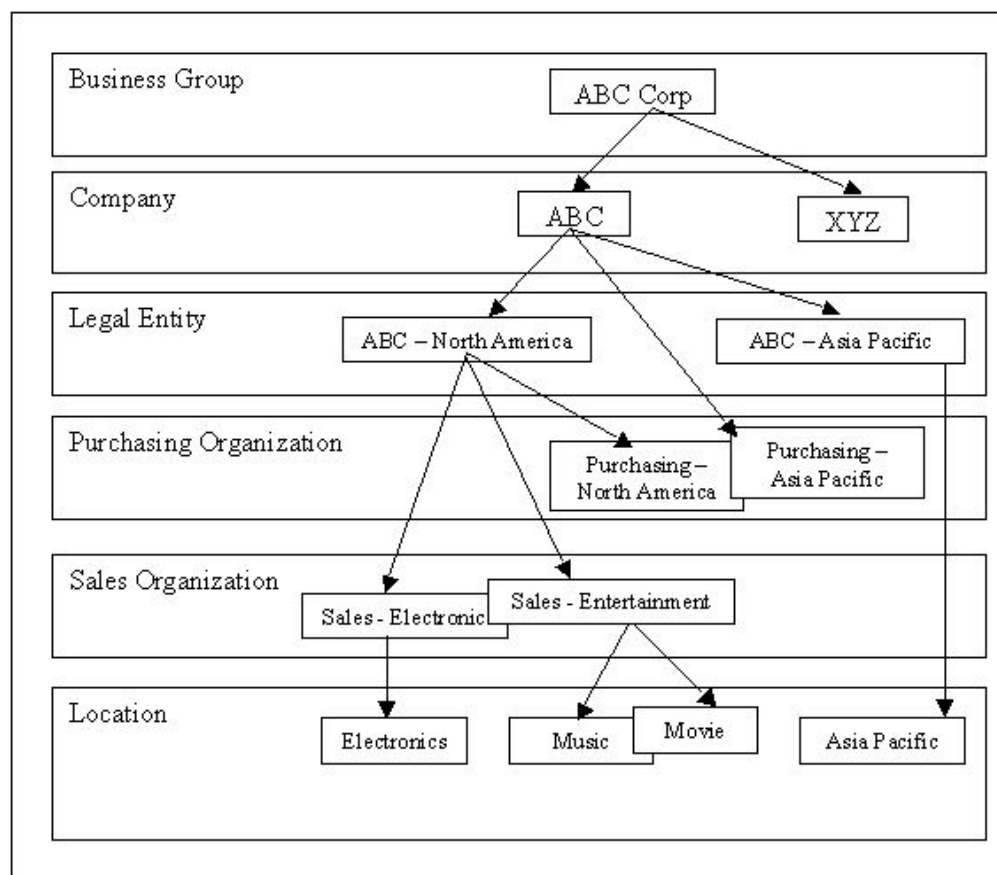


図 5. 一般的な組織モデル

## ビジネス・グループ (Business Group)

ビジネス・グループ (*business group*) は、一般に組織階層における最上位のレベルです。会計上の影響はありません。これは 1 つ以上の会社で構成されます。

## 会社

会社は一般にグローバルなブランド名を表し、当該の法的要件に従って個別の財務諸表が作成される組織単位です。会社には 1 つ以上の法人組織が含まれる場合があります。

## 法人組織

法人組織は、各政府によって事業単位として識別される組織単位であり、一般に企業が事業を行うすべての国で設立されます。この組織単位は一般に自己完結しており、外部レポートのための一連の勘定を作成できます。これには、関連するすべての取引の記録、貸借対照表や損益計算書などの財務諸表を補助するすべてのドキュメントの生成が含まれます。

## セールス組織

セールス組織は、製品とサービスの販売および流通を担当します。セールス組織は、次の特性を基準にして定義することができます。

- 販売チャンネル - 例えば、卸、小売、直売などです。
- 製品ライン - 例えば、電機製品、エンターテインメント、サービスなどです。
- 地域 - オーダーの出荷先ロケーションの地域です。例えば、東海岸、西海岸などです。
- 顧客 - (主要顧客)。
- 上記の 1 つ以上の組み合わせ。

## 購入組織

購入組織 (バイヤー組織とも呼ばれる) は、ベンダーに購入オーダーを発注することで、社内のロケーションの未加工材料または製品の補充を担当します。購入組織は、集中化することも、法人組織やセールス組織のそれぞれに関連付けることもできます。また、購入組織は製品ライン、地域、ベンダーを基準にしてモデル化することもできます。

## 工場および倉庫

工場と倉庫は、商品の製造または流通のための保管が行われる物理的なロケーションです。一般に、工場や倉庫は法人組織が所有します。企業は、自社の販売および購入組織にほとんどすべての組み合わせで関連付けられた多数の工場または倉庫を所有する場合があります。

倉庫は、単一のセールス組織の製品を保管するように指定されることも、セールス組織構造 (製品ベース、地域ベース、顧客ベース) に従って複数のセールス組織のために機能することもあります。1 つの購入組織が、特定工場の未加工材料購入や特定の倉庫での製品保管を担当することも、複数の購入組織が単一の工場や倉庫を使用することも (地域または製品基準の購入) あります。

サード・パーティー・ロジスティック (3PL) 会社、または一部の企業でまれに、特定のロケーションで複数の法人組織または会社の在庫が保管される場合があります。この場合は一般に、3PL 社が在庫保管ロケーションの所有者になります。

---

## Sterling Selling and Fulfillment Foundation の組織のモデル化

Sterling Selling and Fulfillment Foundation における「組織」は、会社、法人組織、ビジネス・グループ (business group)、セールス組織、購入組織、倉庫のいずれであるかを問わず、企業内の単位を表します。部署および部門も Sterling Selling and Fulfillment Foundation における組織として表されます。組織は、Sterling Selling and Fulfillment Foundation で階層構造として定義できます。組織階層は、Sterling Selling and Fulfillment Foundation の複数の場所で構成、およびその他のビジネス・ルール (business rules) の継承に使用されます。顧客とベンダーも複数レベルの組織として定義されます。31 ページの『顧客定義とベンダー定義』は、Sterling Selling and Fulfillment Foundation における顧客とベンダーのモデル化を示しています。

組織モデルの参加者を Sterling Selling and Fulfillment Foundation で定義する際は、組織の役割の割り当てだけでなく、いずれの組織も以下の責任を持たせるように指定できます。

- エンタープライズ
- カタログ組織 (Catalog Organization)
- 在庫組織
- キャパシティー組織 (Capacity Organization)

組織は、上記の責任のいずれか、またはすべてを同時に設定して定義することができます。

### エンタープライズ

エンタープライズは、Sterling Selling and Fulfillment Foundation 内のすべてのトランザクションを所有し、制御する組織を表します。Sterling Selling and Fulfillment Foundation 内のエンタープライズは、ドキュメント (販売オーダーなど) のフローを制御し、伝票の所有者と見なされます。1 つのオーダーのほとんどのビジネス・ルール (business rules) およびフルフィルメント・プロセスは、エンタープライズによって定義されます。販売オーダーでは、ほとんど場合、エンタープライズにセラー組織の役割も割り当てられます。同様に、購入オーダーでは、ほとんど場合、バイヤー組織の役割もエンタープライズに割り当てられます。場合によっては、より高いレベルの組織単位がそのすべての従属組織のビジネス・ルール (business rule) と文書フローを制御および実施する場合は、その組織単位にはエンタープライズの役割が割り当てられ、その従属組織にはセラーの役割とバイヤーの役割が割り当てられます。

ほとんどのビジネス・ルール (business rules) がエンタープライズによって制御される場合でも、価格設定ルールは、常に、販売シチュエーションと購入シチュエーションの両方においてセラー組織によって制御されます。

1 つの組織階層内に、複数の組織を 1 つのエンタープライズとして指定することができます。しかし、組織構造内のすべての組織が、エンタープライズとして定義さ

れているか、またはその主要エンタープライズとして指定されている組織をもつ必要があります。これは、Sterling Selling and Fulfillment Foundation が常にこの組織に適用される一連のルールを明確に識別できるようにするために必要です。（この主要エンタープライズ関係が確立されている場合でも、組織は、個々のトランザクション・ベースで別のエンタープライズと共に参加することができます）。

Sterling Selling and Fulfillment Foundation 内のすべてのトランザクション・ドキュメント（オーダー、出荷、集合・混載）は、指定のエンタープライズが必要です。

## カタログ組織 (Catalog Organization)

カタログ組織 (catalog organization) は、アイテム・マスターが定義される、組織階層内の組織レベルを表します。カタログ組織 (catalog organization) は、以下の機能を提供します。

- アイテム・マスターの定義。マルチレベル組織構造の場合でも、通常アイテム・マスターは、階層の非常に高いレベルに定義されます。これによって、他のすべての組織は 1 つの共通のアイテム・マスター定義を共有できるため、独自の定義を個別に作成する必要がなくなります。
- 2 つの別々の組織のアイテム・マスター定義を分離する仕組み。例えば、3PL シナリオ、および合併と買収 (M&A) によって取得したビジネス・グループ (business group) 下の会社があります。同じ製品 ID で、2 つのカタログ組織 (catalog organization) にまたがる完全に別の物理的な製品を表すことができます。
- 国際取引商品番号 (GTIN) を使用して、製品 ID を相互参照する仕組み。

カタログ組織 (catalog organization) は、1 つ以上のカタログを保有できます (マスター・カタログなど)。ただし、アイテムを定義できるのは、1 つのマスター・カタログ内のみです。マスター・カタログは、多くのシナリオでは、製品明細を表します。部門が製品のみに基づいて作成されている組織では、マスター・カタログは部門を表します。

通常、カタログ組織 (catalog organization) は、会社レベルで指定されます。

## 在庫組織

在庫組織は、すべての在庫情報が集約される組織階層内の組織レベルを表します。在庫組織は、以下の機能のいずれかを提供します。

- 製品の在庫 ID。異なる組織が、同じ在庫アイテム (inventory item) に対して異なる製品 ID を持つことがあります。在庫組織は、複数の組織間でそれらの製品 ID を単一の命名として論理的に結び付けるためのメカニズムを提供します。これにより、組織階層間で製品の一貫したグローバル・ビューが許可されるため、そのビジネスに関してより良い意思決定が可能となります。
- 単一の物理ロケーションが複数の組織間で共有されている場合、在庫の所有権を確立します。在庫の所有権を確立するのに、複数の論理ロケーションを作成する必要はありません。
- 在庫分離。同じ在庫組織の一部として定義されているすべての組織は、この在庫組織の一部である他のすべての組織の在庫を見ることができます。これにより、組織間での可視性は向上しますが、データ・セキュリティやその他の問題も生

じる可能性があります。在庫組織は、システム内に在庫定義の分離格納庫を作成します。この格納庫は、別の格納庫に属している組織からはアクセスできないため、在庫の完全分離を提供します。

すべての組織は、在庫組織として定義されるか、あるいはそれぞれの在庫組織として別の組織を指定する必要があります。組織について指定されている在庫組織は、同じカタログ組織 (catalog organization) を共有する必要があります。これにより、アイテムの在庫識別に到達するように該当する相互参照を確実に作成することができます。また、2 つの別個の物理的アイテムが、在庫モジュール内で同じ在庫 ID を持っており、そのために区別ができない状況が生じないようにもします。

また、セールス組織法人組織が、製品の調達元である物理的ロケーションと関連付けられている同じ法人組織でない場合、Sterling Selling and Fulfillment Foundation は、適切な在庫所有権移動が確実に行われるようにするために、購入オーダーを自動的に生成する (連鎖オーダーとして作成される) ことができます。これは、在庫組織が共通な場合でも行われます。物理的ロケーションが 3PL 組織によって所有され、実際の在庫所有権を示していない 3PL と同様に、これは望ましくない場合があります。これは、以下の 2 つの方法のいずれかにより避けることができます。

- そのような連鎖オーダーの作成を抑制するために、ノードを「3PL」ノードとして指定する。
- ノードの所有者組織に「連鎖オーダー不要」のフラグを立てる。

一般に、1 つの組織階層内の法人組織は、すべての補助組織のための在庫組織として指定されます。

## キャパシティー組織 (Capacity Organization)

キャパシティー組織 (Capacity Organization) は、すべてのキャパシティー情報が集約される、組織階層内の組織レベルを表します。キャパシティー組織 (capacity organization) は、以下の機能を提供します。

- オーダー約束機能用にサービス・スロットを定義。
- キャパシティーの分離。同じキャパシティー組織 (Capacity Organization) の一部であるすべての組織は、このキャパシティー組織 (Capacity Organization) の一部であるすべてのリソースのキャパシティーを見ることができます。これによって、リソース間の可視性が向上しますが、データ・セキュリティーおよびその他の問題が発生する可能性もあります。キャパシティー組織 (Capacity Organization) は、システム内にキャパシティー定義の別のサイロを作成します。このサイロには、異なるサイロに属する組織はアクセスできないため、キャパシティーを完全に分別することができます。

## エンタープライズのオンボード化

Sterling Application Platform フレームワークは、オンボードの新しいエンタープライズを既存の構成にする能力を提供します。これは、より新しい組織と既存の組織を分ける上でも役立つため、互いに妨げとなるさまざまな組織の出現が少なくなります。

エンタープライズのオンボード化の一部として、テンプレートや外部プログラムなど、構成可能なリソースをエンタープライズに基づいて分離することができます。また、既存のエンタープライズ・リソースに影響を及ぼすことなく、新しいエンタ

エンタープライズを追加することもできます。エンタープライズ・レベルで構成されたリソースは固有のリソース ID を使って識別されます。この ID は、対応するエンタープライズに属しているリソースを見つけのに使用することができます。

## エンタープライズの継承

エンタープライズのオンボード化フィーチャーの一部として、Sterling Application Platform は、エンタープライズが他のエンタープライズから構成を継承できるようにします。例えば、エンタープライズ B は、完全に新しい構成セットを定義する代わりに、既存のエンタープライズ A から構成を継承するよう選択することができます。

エンタープライズの継承について詳しくは、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイドプラットフォーム構成ガイド (*Platform Configuration Guide*)」の『継承の決定 (Determining Inheritance)』のセクションを参照してください。

## 拡張のパッケージ化

マルチエンタープライズ・セットアップにおいて、さまざまなレベルの拡張および構成を管理するのは煩わしいことです。例えば、エンタープライズのオンボード化の一部として、追加の属性を公開する場合は、追加の属性を含むようにデフォルトのテンプレートを変更する必要があります。この変更が、他のオンボード化されたエンタープライズに影響を及ぼす可能性があります。この複雑さを解決するために、Sterling Application Platform では、データベース拡張、テンプレート、UI リソースなどのエンタープライズに特有のリソースを拡張サービス・パッケージとして開発できます。このパッケージには、エンタープライズをオンボード化するためのコンポーネントがすべて含まれます。

エンタープライズ・レベルのリソースの開発およびパッケージ化について詳しくは、「*Sterling Selling and Fulfillment Foundation* カスタマイズ 基本」のセクション『エンタープライズ・レベルの拡張の構築および配備 (Building and Deploying Enterprise-Level Extensions)』を参照してください。

## マルチテナントおよびマルチスキーマ・アーキテクチャー

Sterling Selling and Fulfillment Foundation のマルチテナント・アーキテクチャーでは、アプリケーションの 1 回のデプロイメントで複数のエンタープライズ (テナント) をサポートし、データと構成を適宜分離できます。これにより、アプリケーション・サーバーおよびハードウェアへの投資を活用することができます。

マルチ・テナント・デプロイメントは、各種のプロセス・フロー、エンタープライズに特有の拡張、およびルールなど、固有のビジネス・ニーズをもつ複数のエンタープライズで構成されます。マルチテナンシーでは、データはトランザクション・スキーマと構成スキーマ (Configuration Schema) に分割されるため、各顧客がカスタマイズされた仮想アプリケーション・インスタンスを処理します。

マルチ・スキーマ・デプロイメントは、マルチ・テナント・デプロイメントのタイプの 1 つで、複数のトランザクション・データベース・スキーマを持っています。

マルチ・スキーマ・デプロイメントは、以下のスキーマを構成します。

- 構成 - 組織と同様に、システム・ルールおよびビジネス・ルール (business rules) を保管します。
- トランザクション - オーダー、出荷タスク、在庫に関連したデータ・トランザクションを保管します。
- マスター・データ - トランザクション・スキーマと共に常駐して、アイテムおよび顧客など、バッチ・フィードを通じて作成されたデータが含まれます。
- メタデータ - 構成情報のデータベース検索に必要なデータが含まれます。
- 統計 - エージェント統計および API 統計など、アプリケーションの統計を管理します。

マルチ・スキーマ・デプロイメントでは、一度に 1 つのエンタープライズをアップグレードすることができますが、他のエンタープライズはアップグレードの準備が整うまで同じバージョンのままになります。

Sterling Selling and Fulfillment Foundation のインストール時にマルチスキーマ・モードを有効にしたり、単一スキーマ・モードでインストールして、後でマルチスキーマ・モードを有効にしたりすることができます。リリース 9 にアップグレードしようとする場合は、アップグレード後にマルチスキーマ・モードを有効にすることができます。アップグレードの実行は、単一スキーマ・モードでのみ可能であるためです。

マルチ・テナント・デプロイメントとマルチ・スキーマ・デプロイメントのオプション、利点、および制限については、「*Sterling Selling and Fulfillment Foundation: Multi-Tenant Enterprise Guide*」を参照してください。この資料では、詳細なマルチスキーマの概念、およびマルチスキーマ・モードを有効にする方法についても説明しています。

## 顧客定義とベンダー定義

どのビジネス・グループ (business group) も、Sterling Selling and Fulfillment Foundation では、すべての従属組織が階層方式で定義された組織として表すことができます。この構造を見る参加者はいずれも、同じ一貫性のある見方をします。

Sterling Selling and Fulfillment Foundation は、顧客とベンダーを 2 つの参加者間の関係としてモデル化します。この関係では、組織 X は、組織 Y をその顧客としてモデル化する場合は、組織 Y に対する固有の明確な ID (組織 ID) を持ち、組織 Y をそのベンダーとしてモデル化する場合は、別の ID を持つことができます。これは、特に、複数のエンタープライズ・リソース・プランニング (ERP) (Enterprise Resource Planning (ERP)) システムがそれぞれ固有の ID を作成する組織と関係する場合に当てはまります。Sterling Selling and Fulfillment Foundation は、複数の役割を果たす単一の組織として参加者をモデル化するために強力な論理付け機能を提供する場合でも、ほとんどの他のサプライ・チェーン・システムにはまだそのような強力な機能がないことを認めます。

### 顧客および顧客連絡先の定義

顧客を定義する場合は、Sterling Selling and Fulfillment Foundation は、顧客が確実に組織としてモデル化されるようにします。Sterling Selling and Fulfillment Foundation では、顧客 ID と同じ組織 ID を使用するか、あるいは別の ID コード

を生成または入力する柔軟性が提供されます。顧客が作成された時点で自動的に組織を作成するために、顧客作成 API が有効になっています。

デフォルトの請求先住所、出荷住所、販売先住所、および支払方法は、Sterling Selling and Fulfillment Foundation オーダー・エントリー機能またはオーダー作成 API によって自動的にピックアップされます。これにより、Sterling Selling and Fulfillment Foundation で ERP をモデル化できるようになり、より良い顧客サービスと意思決定支援を可能にするために引き続き ERP システム間で組織を 1 つの共通組織として論理的に結びつけることができます。

オーダーでのデフォルトの住所および支払方法を顧客または顧客階層の顧客の親のどちらから引き出すかを決定するルールを設定できます。また、オーダー・エントリーの一部として有効な顧客連絡先情報が渡された場合は、オーダーの住所および支払方法のデフォルトを特定の顧客連絡先から取ることも可能です。顧客連絡先とは、B2C 関係にある特定の消費者または B2B 関係にある顧客エンタープライズ内の特定の連絡先です。

どのエンタープライズも、必要に応じて顧客エンティティを作成できます。顧客の役割は、以下のように指定されます。

- ビジネス - 企業間 (B2B) トランザクション。企業顧客 (business customers) は、以下のように表されます。
  - バイヤー組織 - 「請求先」情報を表します。
  - 受入ノード - 「出荷先」情報を表します。
- 消費者 - 企業消費者間 (B2C) トランザクション。消費者顧客は、「請求先」情報により示されます。

顧客ステータスは、顧客 (およびこの顧客の連絡先) がログインしてオーダーを発行できるかどうかを示します。

- アクティブ - 顧客およびこの顧客の連絡先はログインしてオーダーを発行できます。
- 非アクティブ - 顧客およびこの顧客の連絡先はログインできません。また、顧客は、新規カートを作成したり、既存のカートからオーダーを発行したりすることはできません。
- 保留中 - 顧客、顧客連絡先、およびこの顧客のオーダーは、オーダー確認の間、保留中になります。保留のタイプは構成可能です。

製品および価格設定資格の目的で、以下の属性で顧客をさらに定義することができます。

- 業種 - 顧客が属している業界を示します (教育、政府機関など)。
- 関係タイプ - 顧客が、販売店、ディストリビューター、委託製造業者 (Contract Manufacturer) などのいずれであるかを示します。
- 顧客レベル - 顧客のステータスを表す段階的な値 (ゴールド、プラチナ、シルバーなど) を示します。



## ベンダーの定義

顧客定義と同様に、エンタープライズは、ベンダーを定義できます。ベンダーを定義するとき、Sterling Selling and Fulfillment Foundation 組織に対して参照を作成する必要がありますが、ベンダー ID を Sterling Selling and Fulfillment Foundation 組織の ID と異なる ID にできます。

ベンダーは、セラー組織または出荷ノード、あるいはその両方にできます。ベンダーは、「出荷元」情報によって表されます。

## 従来の ERP システムでの顧客モデル化

製靴会社が 2 つの主要ブランド名の靴を製造している従来の ERP システムでは、各靴明細が、別個の会社 (SeeWok および Mockport) としてセットアップされています。これらの会社は両方とも、共通の顧客があり、それは M-T-Pockets という主要小売業者です。これらの靴明細は、ERP システムでは 2 つの異なる会社としてモデル化されるため、同じ顧客が、互いにリンクなしで 2 回モデル化されます。このモデルでは、主要小売業者 M-T-Pockets について、すべての顧客コンプライアンス (customer compliance) ルール (レーティング、ラベル付け、支払条件など) も 2 回定義する必要があります。図 6 は、このシナリオを示します。

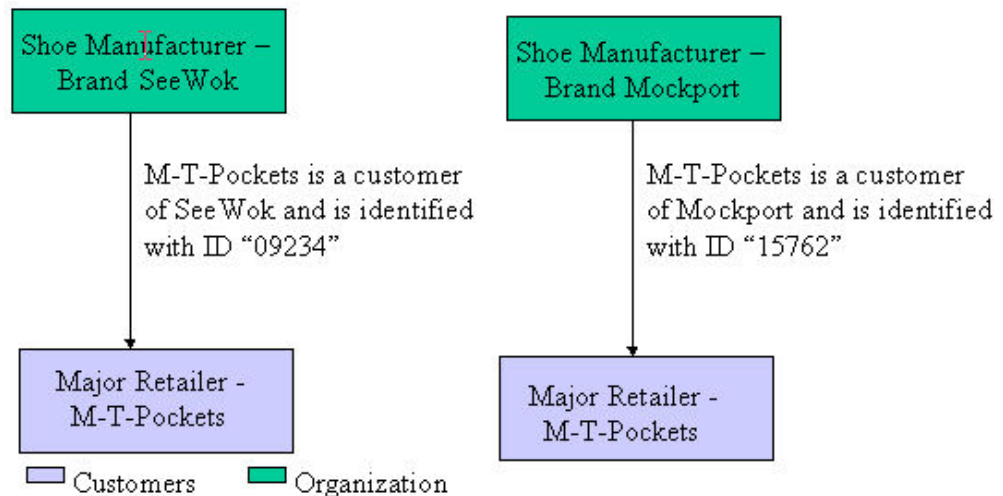


図 6. 従来の ERP システムの顧客モデル

## Sterling Selling and Fulfillment Foundation での顧客モデル化

Sterling Selling and Fulfillment Foundation は、顧客モデル化のための 2 つのオプションを提供します。それは、顧客を異なる顧客 ID を持つ 2 人の別個の顧客としてセットアップするか、あるいは同じ顧客 ID を持つ 2 人の別個の顧客としてセットアップすることです。どちらの場合も、両方の顧客の顧客レコードは、同じ組織レコードにリンクされます。34 ページの図 7 および 35 ページの図 8 は、この 2 つのオプションを示しています。

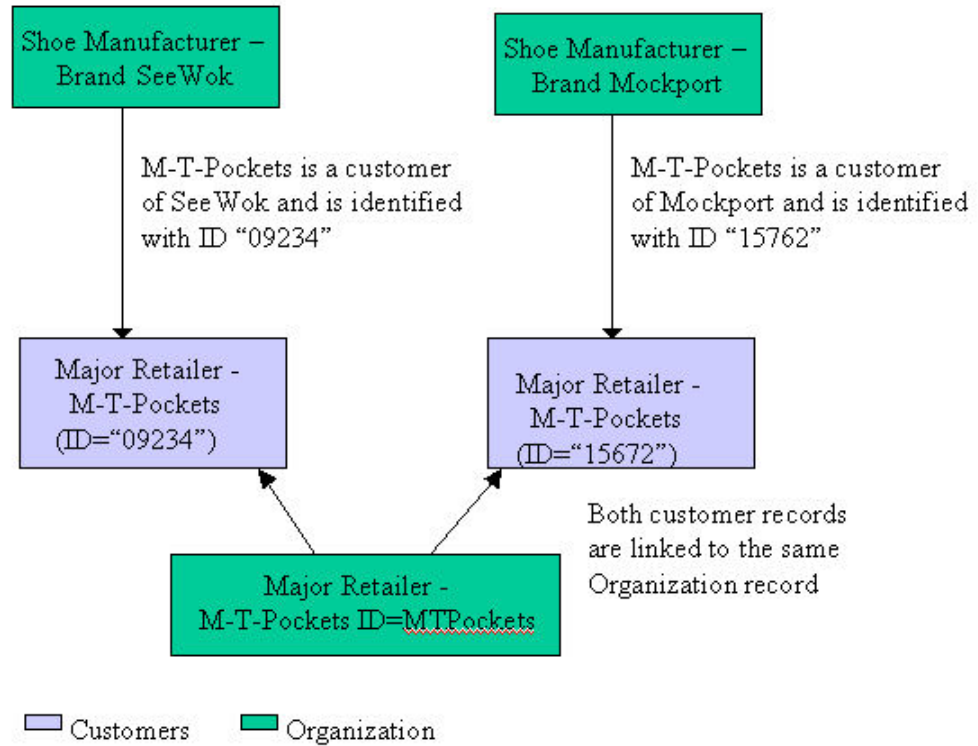


図7. Sterling Selling and Fulfillment Foundation 顧客モデル化オプション 1

このオプションの場合、組織「M-T-Pockets」について、主要小売店に対して定義されるすべての顧客コンプライアンス (customer compliance) ルール (ルーティング、ラベル付け、支払条件など) を、このモデルで「一度」だけ定義する必要があります。Sterling Selling and Fulfillment Foundation は、顧客が 2 つの異なる顧客としてモデル化されている場合でも、正しい構成を自動的にピックアップします。

このアプローチのもう 1 つの長所は、この、単一組織への論理的結び付けを、一定期間にわたって実行できる点です。当初、顧客を 2 つの異なる組織として表すことがあっても、論理的な結び付けが実行可能な場合は、マージして単一のエンティティに組み込むことができます。

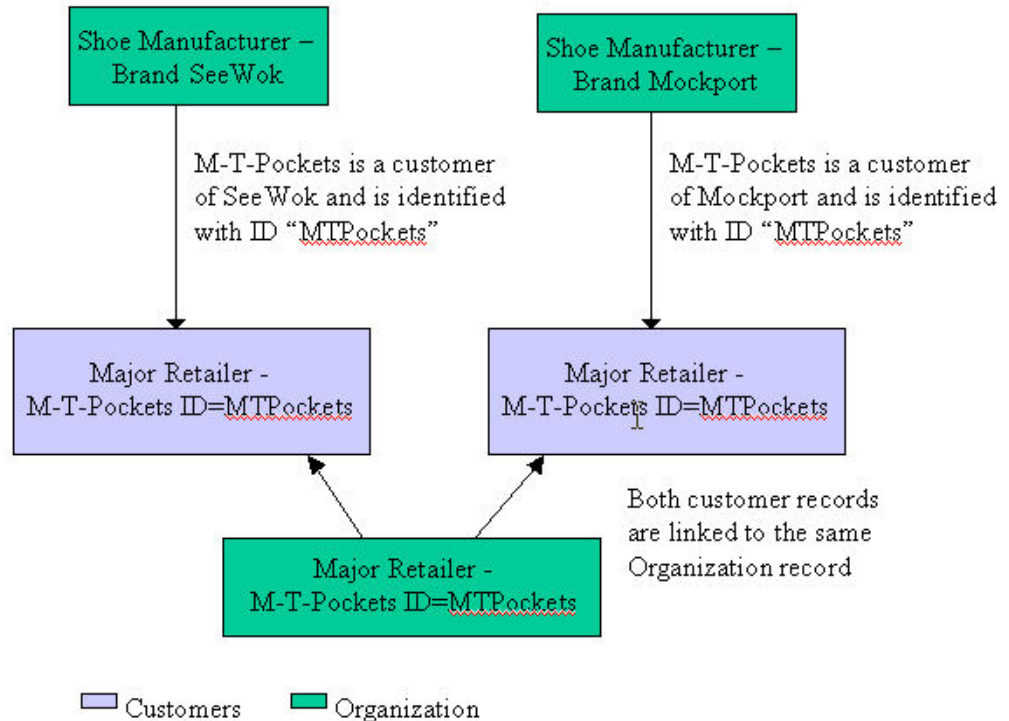


図8. Sterling Selling and Fulfillment Foundation 顧客モデル化オプション 2

このオプションの場合、組織「M-T-Pockets」について、主要小売店に対して定義されるすべての顧客コンプライアンス (customer compliance) ルール (ルーティング、ラベル付け、支払条件など) をこのモデルで「一度」だけ定義する必要があります。システムは、顧客が 2 つの異なる顧客としてモデル化されている場合でも、正しい構成を自動的にピックアップします。

## 顧客管理

ほとんどの企業間取引環境では、カート作成、オーダー発行、見積もり要請などを行うバイヤーが多数います。同様に、これらの顧客のニーズと、顧客が発行するオーダーを管理する顧客サービス担当者 (CSR) (customer service representatives (CSR)) も多数います。しかし、一般に、すべての顧客サービス担当者 (customer service representatives) が組織内のすべての顧客を管理するわけではありません。各顧客サービス担当者 (customer service representative) は通常、選ばれた一握りの顧客のニーズの管理を担当します。顧客は、分類または特定のオーダー属性でグループ分けすることができます。

Sterling Selling and Fulfillment Foundation では、各顧客サービス担当者 (customer service representative)、つまりユーザーを、特定の顧客を管理するために手動で割り当てるか、あるいは、複数の顧客を管理するために割り当てられたチームのメンバーとして自動的に割り当てることができます。管理者は、それぞれのチーム、またはそれぞれの子孫のチームが管理するすべての顧客に対してアクセス権を持ちます。

顧客サービス担当者 (customer service representatives) は、セラー組織のために顧客アカウント (customer account) を管理するユーザー連絡先です。しかし、バイヤー・ユーザー (buyer users) は、顧客組織のためにアカウントを管理するユーザー連

絡先です。バイヤー顧客の割り当てにより、1 人のバイヤー顧客を 1 つ以上の顧客組織に割り当てることができます。バイヤー顧客の割り当てを使用して、顧客は、それぞれ固有のオーダーを発行することができます。場合によっては、他の顧客に代わってオーダーを発行することができます。

顧客の割り当て (customer assignment) は、1 つ以上の顧客マスター組織が含まれている顧客階層でサポートされます。

この機能を可能にするには、顧客階層または顧客の割り当て (customer assignment) に対する変更が発生したときにいつでも、手動により、またはサービスを通じて、updateCustomerAssignmentLookup API を呼び出す必要があります。

**顧客の割り当て (customer assignment):** 顧客サービス担当者 (customer service representatives) (ユーザー) は、次のようにして顧客に割り当てられます。

- 手動で
- オーダー属性に基づいて手動で
- チームとして

バイヤー顧客の割り当てでは、バイヤー・ユーザー (buyer users) は 1 人の顧客または一連の顧客に割り当てられます。

#### **手動での顧客の割り当て (customer assignment)**

販売員が顧客に初めて販売したら、その顧客を管理するために、該当の販売員を割り当てることができます。このシナリオでは、2 人の販売員が同じチームに属している場合でも、1 人がもう 1 人の販売員に代わって顧客を管理できないように、各ユーザーについて手動での割り当てが必要です。手動でのユーザー割り当てが使用可能な場合は、常にユーザーを特定の顧客に直接割り当てる必要があります。そのユーザーが顧客オーダー (customer order) および情報にアクセスできるのは、その時のみです。ユーザーが管理者である場合は、自分のチームに割り当てられているすべての顧客と、自分のサブチームに割り当てられているすべての顧客のオーダーおよび情報にアクセスすることができます。

顧客を管理できるチームまたはユーザーの最大数と、特定のユーザーに割り当てることができる顧客の最大数を定義することができます。37 ページの図 9 は、手動での顧客の割り当て (customer assignment) についての顧客の所有権シナリオを示しています。

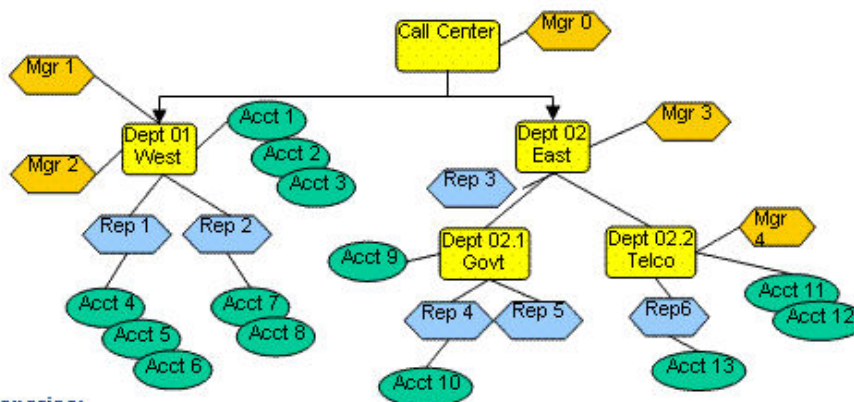
**KEY:**

- Team
- Supervisor
- Sales Rep/CSR
- Customer

**RULES:**

Max Cust per User = 3

Max Users per Cust = 2

**Customer Ownership Scenarios:**

- Mgr 0 has visibility into all orders and information for customers (1-13)
- Mgr 1 and Mgr 2 have visibility to all orders and information for customers assigned to the team or the reps of the team (Accts 1-8)
- Rep 1 can not be assigned any more accounts because he has max of 3
- Rep 3 has no accounts assigned to him so he does not see any orders or customer information
- Rep 4 & Rep 5 could both be assigned Acct 9 because they have less than 3 accounts and Acct 9 can be assigned to a maximum of 2 Reps

図 9. 手動での顧客の割り当て (customer assignment)

**オーダー属性に基づく手動での顧客の割り当て (customer assignment)**

Sterling Selling and Fulfillment Foundation は、オーダーに対してチーム・コード属性を提供します。チームに属しているユーザーは、それらのオーダーを見ることができます。オーダーの作成および変更中にチーム・コードを取り込むために、外部ロジックを実装する必要があります。

**チームへの顧客の割り当て (customer assignment)**

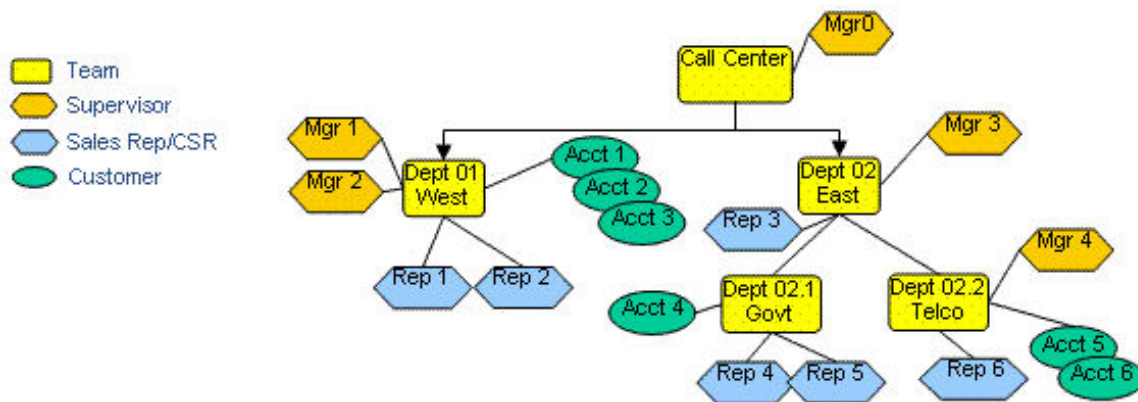
企業間取引環境では、多くの場合、各顧客は、顧客サービス担当者 (customer service representatives) (ユーザー) の特定のチームによって管理されます。Sterling Selling and Fulfillment Foundation では、チームへの顧客割り当ての選択肢が 3 つ用意されています。

- チームのユーザーはすべて、それぞれのチームに割り当てられたすべての顧客にアクセスすることができます。  
これにより、すべてのチーム・メンバーは、オーダーを表示し、そのチームに割り当てられた顧客と関連付けられている警告を受け取ることができます。これを構成するには、手動での顧客へのユーザーの割り当ては必要でないことを指定してください。
- チームのすべてのユーザーを顧客に直接割り当てる必要があります。  
これにより、チーム・メンバーは、顧客に直接割り当てられている場合に限り、顧客オーダー (customer order) および警告にアクセスすることができます。これを構成するには、手動での顧客へのユーザーの割り当てが必要です。
- すべての顧客にアクセスできるチームもあれば、メンバーを顧客に直接割り当てる必要があるチームもあります。

すべての顧客にアクセスできるチームは、顧客のオーダーおよび警告をすべて表示することができます。顧客への直接割り当てを必要とするチームは、顧客のオーダーおよび関連した警告を表示するためにアクセス権が必要です。この場合、手動での顧客へのユーザーの割り当てが必要です。

チーム管理者は、自分のチームまたはサブチームに割り当てられている顧客のすべてのオーダーおよび情報にアクセスすることができます。ユーザーは、顧客にアクセスできる場合は、この顧客のすべての子孫にアクセスすることができます。

図 10 は、チームへの顧客の割り当て (customer assignment) に関する顧客所有権シナリオを示しています。



#### Customer Ownership Scenarios:

- Mgr 0 has visibility of orders and information for all customers (1-6)
- Mgr 1 or Mgr 2 as well as Rep 1 and Rep 2 have visibility of orders and information for all customers assigned to their team (Accts 1-3)
- Rep 3 has no customers assigned to his team so he does not see any orders or customer information
- Rep 4 & Rep 5 have visibility into Acct 4

図 10. チーム割り当てに関する顧客所有権シナリオ

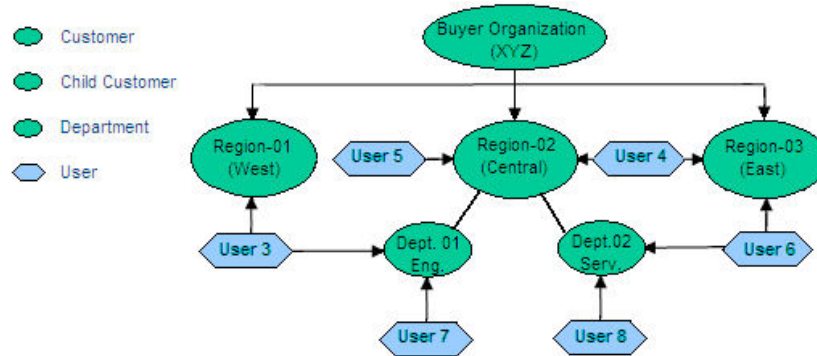
#### バイヤー顧客の割り当て

企業間取引環境では、複数の組織を顧客としてモデル化することができます。顧客連絡先は、バイヤー・ユーザー (buyer users) ともいい、これらの顧客のアカウントを管理するために割り当てることができます。バイヤー顧客の割り当てにより、顧客は自身のアカウントを管理することができ、場合によっては、他の顧客のアカウントを管理することができます。ただし、自身のアカウントも、他の顧客のアカウントも管理しない顧客もいます。

Sterling Selling and Fulfillment Foundation は、manageCustomerAssignment API から渡される組織およびユーザー情報を使用して、バイヤー顧客の割り当てを決定します。バイヤーへの顧客の割り当てを構成後、バイヤー・ユーザー (buyer users) のデータ・アクセス・ポリシーには、アカウント管理のために、以下の選択肢があります。

- バイヤー・ユーザー (buyer users) は、それぞれの割り当てられた顧客のみのデータにアクセスすることができます。
- バイヤー・ユーザー (buyer users) は、自分に割り当てられた顧客および子の顧客のデータにアクセスできます。

図 11 は、一部の顧客が他の顧客のアカウントを管理するバイヤー顧客の割り当てを示しています。



#### Account Ownership Scenarios:

- All users modeled here belong to XYZ Buyer Organization.
- Users 3, 4, 5, and 6 are all assigned regional offices for managing their respective supplies.
- Users 7 and 8 are assigned to specific departments for managing supplies.
- Users 3 and 6 can also be assigned specific departments to manage orders for that department.
- Users 4 and 5 can manage supplies for both departments based on hierarchy.

図 11. バイヤー顧客の割り当て

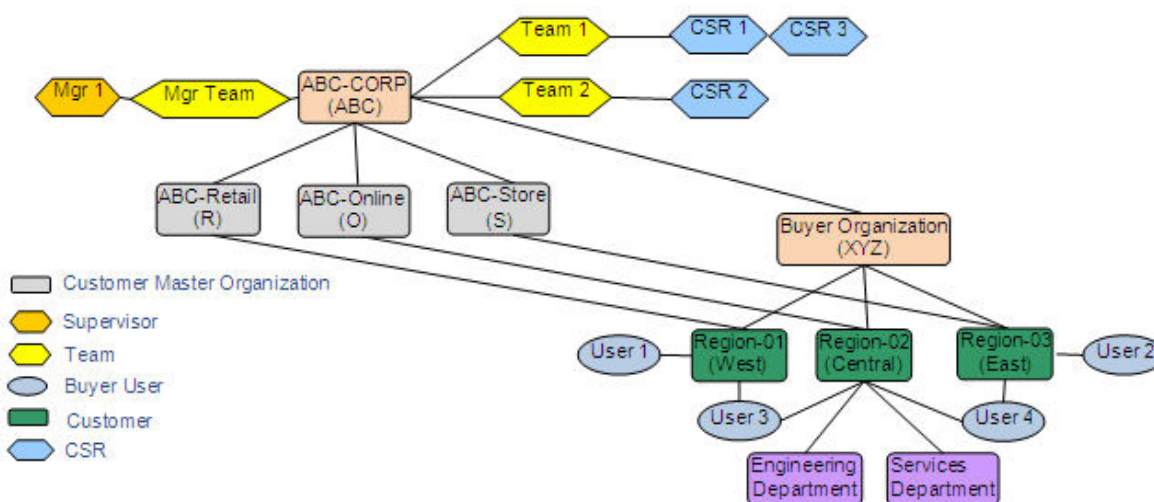
**複数の顧客マスター組織:** Sterling Selling and Fulfillment Foundation では、複数の顧客マスター組織間の顧客の割り当て (customer assignment) がサポートされ、ユーザーを 1 つ以上の顧客マスター組織の顧客に割り当てることができます。このビジネス・シナリオでは、顧客サービス担当者 (customer service representative)、またはバイヤー・ユーザー (buyer user) を、同一の顧客階層に含まれない複数の顧客に割り当てることができます。

顧客サービス担当者 (customer service representative) を複数の顧客マスター組織の顧客に割り当てるときは、顧客サービス担当者 (customer service representative) にチームを定義してから、顧客マスター組織をそのチームのエンタープライズ・リストに追加します。次に、manageCustomerAssignment API を使用して、顧客サービス担当者 (customer service representative)、およびチームを顧客に割り当てます。顧客サービス担当者 (customer service representative) は、割り当てられたすべての顧客のオーダーを管理できます。

バイヤー・ユーザー (buyer user) を複数の顧客マスター組織の顧客に割り当てるときは、顧客が同一のバイヤー組織に属していることを確認します。次に、manageCustomerAssignment API を使用して、バイヤー・ユーザー (buyer user) を顧客に割り当てます。バイヤー・ユーザー (buyer user) は、割り当てられたすべての顧客のオーダーを管理できます。

ユーザーおよびチームの構成について詳しくは、*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイドを参照してください。

図 12 は、複数の顧客マスター組織の顧客の割り当て (customer assignment) を示しています。



#### Account Ownership Scenarios:

- ABC-Retail, ABC-Online, and ABC-Store are customer master organizations, each having one Region as a customer.
- CSR 1's team, Team1, is assigned to the Region-01 customer and the Region-03 customer. CSR 2's team, Team 2, is assigned to the Region-02 customer. CSR 3's team, Team 1, is assigned to the Region-01 customer and the Region-03 customer.
- CSR 1 and CSR 3 can manage orders for the Region-01 customer and the Region-03 customer.
- CSR 2 can manage orders for the Region-02 customer.
- Mgr. 1 can manage all customers: Region-01, Region-02, and Region-03.
- Users of Buyer Organization XYZ (User 1, User 2, User 3, and User 4) can manage orders for their respective Regional customers.
- User 3 and User 4 can manage orders for both departments, based on hierarchy.

図 12. 複数の顧客マスター組織

図 12 の例は、複数の顧客マスター組織の顧客に割り当てられたユーザーのアカウント所有権のシナリオを示しています。

CSR 1 と CSR 3 を複数の顧客マスター組織の顧客に割り当てるため、CSR 1 と CSR 3 はチーム 1 に割り当てられ、ABC-Retail および ABC-Store の顧客マスター組織がチーム 1 のエンタープライズ・リストに追加されています。この例の場合、CSR 1 と CSR 3 は、ABC-Retail の顧客 Region-01 と ABC-Store の顧客 Region-03 のオーダーを管理できます。同様に、CSR 2 は、チーム 2 に割り当てられ、このチームのエンタープライズ・リストには ABC-Online の顧客マスター組織が追加されています。CSR 2 は、ABC-Online の顧客 Region-02 を管理できます。管理者 Mgr. 1 は、CSR 1、CSR 2、および CSR 3 に割り当てられたすべての顧客のオーダーを管理できます。

また、この例では User 3 が複数の顧客マスター組織の顧客に割り当てられています。User 3 は、ABC-Retail 顧客マスター組織の Region-01 の顧客と ABC-Online 顧客マスター組織の Region-02 の顧客に割り当てられています。Region-01 の顧客と Region-02 の顧客はいずれも同じバイヤー組織、Buyer Organization (XYZ) に属しています。



## 組織モデル化のためのガイドライン

Sterling Selling and Fulfillment Foundation 用に組織の階層モデルをセットアップする準備段階で、各組織に対して定義したい役割および責任を識別する必要があります。

どの組織がどの役割および責任を持つべきかを決定するには、次のように行います。

1. 組織階層に対して各組織を識別します。すなわち、ビジネス・グループ (business group)、会社、法人組織、セールス組織、および購入組織です。

ビジネス・グループ (business group) が複数の会社をもつ理由はいくつか考えられます。会社の買収が一例です。ビジネス・グループ (business group) が 2 つの関係のないビジネスを持っている場合も、他の一例です。

会社は、通貨、課税構造、またはその他の法的必要条件などの法律問題に関して互いに異なる地理的地域で活動する場合は、複数の法人組織を持つ必要があることがあります。

セールス組織を定義するための基準を決定します (例えば、販売チャネル、製品グループ、ジオグラフィー、または顧客)。セールス組織には、ソーシング・ルールおよび価格設定ルールがあります。このことは、セールス組織を決定する上で役に立ちます。例えば、異なる販売チャネルまたはジオグラフィーに対して異なる価格設定ルールまたはソーシング・ルールがある場合、セールス組織は、販売チャネルとジオグラフィーに基づくことができます。

2. 組織階層内の組織についてエンタープライズを識別します。エンタープライズの識別についての考慮事項は、43 ページの『モデル例 1: 電機会社』および 47 ページの『モデル例 2: サード・パーティーの物流管理会社』で説明しています。
3. カタログ組織 (catalog organization) (単数または複数) を識別します。組織階層内のどの組織でもカタログ組織 (catalog organization) にすることができます。一般に、主要会社がカタログ組織 (catalog organization) です。会社の 2 つの組織が 2 つの異なる物理的アイテムに対して同じアイテム ID を持っており、会社がその 2 つのアイテムに対して共通のアイテム ID を持っていない場合は、カタログ組織 (catalog organization) は、組織階層の下位レベルで定義する必要があります。2 つの法人組織が互いのアイテムを抱き合わせ販売しない場合は、カタログ組織 (catalog organization) を下位レベルで定義することができます。しかし、カタログ組織 (catalog organization) を階層構造の在庫組織よりも下位にすることはできません。
4. 在庫組織 (単数または複数) を識別します。組織階層内のどの組織でも在庫組織にすることができます。一般に、在庫組織は法人組織レベルに保持されます。在庫組織を階層内のこのレベルにしておくことには、長所と短所の両方があります。例えば、在庫組織が法人レベルにある場合、その法人組織のセールス組織はすべて、法人組織のロケーション内の在庫に対して同等のアクセス権を持ちます (セールス組織のアクセス権を在庫組織内の指定の物理的ロケーションにのみ制限する機能を使用)。これは、1 つのビジネスでは長所ですが、別のビジネスでは短所になり得ます。

5. 物理的ロケーションの所有者（ノード）を識別します。この定義は、ほとんどの場合、物理的ロケーションの実際の所有権をマップする必要があります。一般に、法人組織は物的資産の所有者であり、ノードの所有者組織としてモデル化されます。

---

## 組織モデル化のためのインストール・レベル・ルール

インストール・レベル・ルールが使用されると、参加者についてデフォルトの各種組織役割が自動的に採用されます。

組織モデル化は理解するのに複雑である可能性があるため、Sterling Selling and Fulfillment Foundationでは、インストール・レベルでセットアップできる強力なデフォルト設定ルールを提供します。

カタログ組織 (catalog organization) をモデル化する場合、品目マスターが以下に該当するかどうかを指定できます。

- ハブ・レベルに保持される

完全なインストールのための品目マスターの定義が 1 つのみである場合は、これを指定します。これをデフォルト・モデルとして選択した場合、参加者が作成された時点でそれぞれのカタログ組織 (catalog organization) としてハブ組織がすべての参加者に自動的に割り当てられます。拡張セットアップを使用した場合、このデフォルトは必要に応じて変更することができます。

- エンタープライズ・レベルに保持される

すべての参加者を、それぞれ固有のカタログ組織 (catalog organization) となるエンタープライズとして定義したい場合は、これを指定します。その他のすべての参加者については、カタログ組織 (catalog organization) が、それぞれの「主要エンタープライズ」として指定されます。

- あらゆる参加者によって定義される

在庫組織またはキャパシティー組織 (capacity organizations) をモデル化する場合は、以下に該当するかどうかを指定できます。

- 完全なインストールには、単一の集約在庫またはキャパシティー・ビューが必要です。これをデフォルト・モデルとして選択した場合、参加者が作成された時点でそれぞれの在庫組織またはキャパシティー組織 (capacity organization) としてハブ組織がすべての参加者に自動的に割り当てられます。拡張セットアップを使用した場合は、このデフォルトは必要に応じて変更することができます。
- 在庫組織またはキャパシティー組織 (Capacity organization) は、エンタープライズ・レベルに保持する必要があります。これをデフォルト・モデルとして選択した場合、エンタープライズとして定義されているすべての参加者が、それぞれの固有の在庫組織またはキャパシティー組織 (Capacity organization) となります。その他のすべての参加者については、在庫組織またはキャパシティー組織 (capacity organization) が、それぞれの「主要エンタープライズ」として指定されます。

これらの組織のデフォルト値は、拡張組織モデル化セットアップを使用して、必要に応じて他の任意の組織に変更することができます。ただし、提供される拡張セッ

トアップを使用する前に、組織のモデル化の意味を理解することがたいへん重要です。各アプローチには長所も短所もあり、本書を十分に理解することが、正しい決定を行う上で役立ちます。

## モデル例 1: 電機会社

Alphabet Electronics Corporation には、ABC と XYZ の 2 つの会社があります。

ABC は北アメリカに 1 つ、アジア太平洋に 1 つの法人組織があります。ABC には 2 つのセールス組織があります。1 つは、電子ガジェットを販売し (販売 - 電機と呼ばれる)、もう 1 つは音楽およびムービー を販売しています (販売 - エンターテインメントと呼ばれる)。

ABC 北アメリカの販売 - 電機には、カリフォルニアに 1 つ、マサチューセッツに 1 つの配送センター (distribution center) があります。

ABC 北アメリカの販売 - エンターテインメントには、カリフォルニアに 2 つの配送センター (distribution center) があり、1 つは音楽を、もう 1 つムービーを扱います。また、マサチューセッツにも 1 つ配送センター (distribution center) があり、ここでは音楽とムービーの両方を扱います。

ABC 北アメリカには、北アメリカの配送センター (distribution center) を担当する購入組織が 1 つあります。

ABC アジア太平洋地域には、電機とエンターテインメントの両方を受け持つセールス組織があります。この配送センター (distribution center) は中国にあります。アジア太平洋の配送センター (distribution center) を受け持つ購入組織が 1 つあります。

XYZ 社の組織構造は ABC と同じです。

### 電機会社の組織階層

Alphabet Electronics Corporation の組織階層は、44 ページの図 13 に示されています。わかりやすくするため、この図では ABC 社エンティティのみの組織階層を示しています。XYZ 社の組織階層は ABC と同じです。

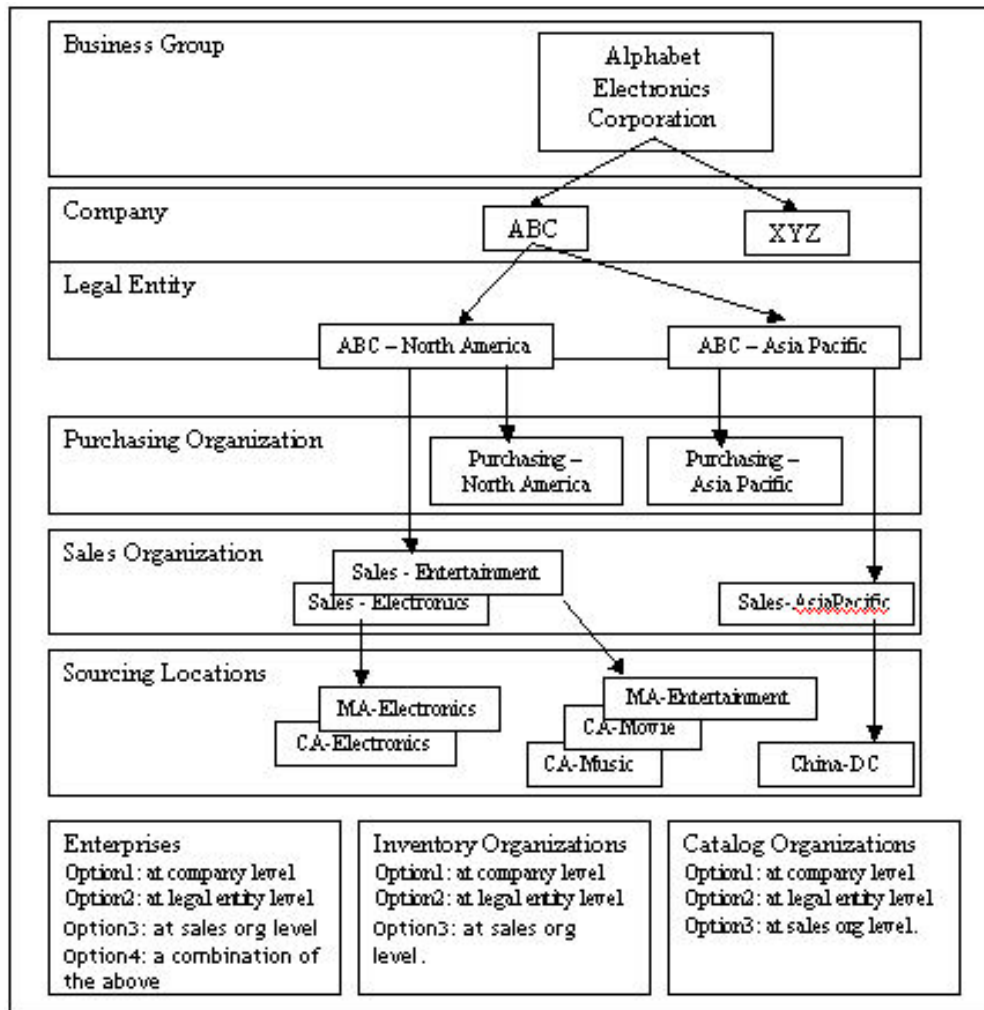


図 13. Alphabet Electronics Corporation の組織階層

## 電機会社にエンタープライズを選択する

エンタープライズを定義する必要がある階層レベルは、以下に基づいて決定します。

- ビジネス・ルール (business rules) およびプロセス定義を定義するのは誰か?

「エンタープライズ」は、ドキュメント・フローおよびドキュメント・フローに関連したビジネス・ルール (business rules) を所有し制御しています。セールス組織で、独自のプロセス定義およびビジネス・ルール (business rules) を定義できる必要がある場合、エンタープライズはセールス組織と同じレベルになければなりません。

- 顧客サービス組織は、組織構造のどこに該当するか?

オーダーは、オーダー上でエンタープライズ、バイヤー、またはセラーとして定義されている組織によってのみ、アクセスできます。顧客サービスを、セールス組織横断で中央から集中管理する場合は、顧客サービス組織と同じレベルに「エンタープライズ」を作成できます。

通常、エンタープライズは、組織構造内の可能な限り高いレベルでモデル化する必要があります。そうすることで、異なるグループ間で共通の業務の慣例を共有することができます。エンタープライズの責任をより高いレベルで指定したとしても、効率的にパイプライン決定ルールを使用することで、各セールス組織ごとに異なるプロセス定義を定義できることに注意してください。

Sterling Selling and Fulfillment Foundation は、単一の階層内で複数の組織の定義をサポートしており、それらを複数の「エンタープライズ」として指定できます。これによって、Sterling Selling and Fulfillment Foundation では、実際の組織構造をモデル化しながら、組織階層内の最も低いレベルですべてのプロセスを管理することも引き続き可能です。

## 電機会社に在庫組織を選択する

在庫組織を定義する必要のある階層レベルは、以下に基づいて決定します。

- 在庫の所有権は、物理ロケーションによって確立できるか？

在庫組織は、2 つの在庫組織間で明確に在庫を分別します。物理ロケーションによって在庫の法的所有権を確立できる場合、所有権の確立にこの機能は必要ない場合があります。そのような恵まれた状況にある場合、在庫組織を組織階層内でより高いレベルに定義し、複数の組織にまたがる在庫をより適切に拡大して表示できます。

- 組織には、販売オーダーを他の組織からソーシングできる機能が必要か？

現在、Sterling Selling and Fulfillment Foundation では、あるセールス組織の販売オーダーのオーダー約束およびスケジューリングを、自らの在庫組織以外の他の在庫組織から行うことができません（在庫が外部で管理されている場合を除く）。2 つの異なる法人に属する組織がお互いにソーシングできる必要がある場合、共通の在庫組織を共有しなければなりません。

このような状況は、ベンダーが在庫情報を組織に公開する場合にも発生する可能性があります。共通の在庫組織は、すべてのベンダーおよびベンダーの在庫へのアクセスが必要な組織について指定されます。これによって、組織はベンダーの在庫に対しても可視性を持つことができます。

なお、1 つの組織で保有できる在庫組織は 1 つのみですが、Sterling Selling and Fulfillment Foundation におけるすべての供給更新トランザクションによって、トランザクションの実行対象となる「在庫組織」を指定できます。この強力な機能によって、ベンダーは在庫情報を複数の在庫組織に同時に公開できます。在庫が同じ製品について 2 つの異なる在庫組織に公開された場合、それらの在庫組織が割り当ての重複を避けるために論理的に在庫を分割していることを、ベンダーは確認する必要があります。

- セールス組織または法人を在庫組織としてモデル化する必要があるか？

在庫組織は、以下の場合、問題なく法人レベルでモデル化できます。

- セールス組織が、完全に異なる製品の販売に携わっている
- 同じ製品が複数のセールス組織によって販売されているが、各セールス組織は物理ロケーションおよびこの在庫の完全な所有権を保有している。セールス組

組織が他のセールス組織の物理ロケーションにアクセスできないように、引き続き分配グループ (distribution group) を使用して、アクセス権を制御する必要があります。ことに注意してください。

同じ物理ロケーションが、同じ製品の販売に携わる複数のセールス組織によって共有されており、セールス組織ごとに在庫を明確に分離する必要がある場合、在庫組織を各セールス組織としてモデル化する必要があります。将来的には、Sterling Selling and Fulfillment Foundation では、このような状況进行处理できるように、分割機能がより強化される可能性があります。

組織全体の在庫の可視性が大きく向上するため、在庫組織を法人レベルで保持することを強くお勧めします。

- 法人または会社を在庫組織としてモデル化する必要があるか？

在庫組織を会社レベルで保持することによって、会社内のすべての組織に対して、会社の在庫が可視化されます。しかし、そのためには、以下の条件を評価する必要があります。

- 各物理ロケーションの在庫を法人に結びつけることができるか？結びつけることができる場合、在庫所有権による問題や障害は発生しません。結びつけることができない場合、Sterling Selling and Fulfillment Foundation は在庫所有権と無関係でいられるか？Sterling Selling and Fulfillment Foundation は純粋にフルフィルメント・ソリューションとして使用され、在庫所有権データは他のシステムで管理することができるため、この質問は場合によっては必要ありません。
- 法人間で共通カタログ組織 (catalog organization) を設定して、同じアイテム ID が 2 つの物理的な製品を表さないようにすることができるか？2 つの法人が同じ物理アイテムに異なる名前を付けても、また場合によっては、2 つの異なる物理製品に同じ名前をつけても、共通カタログ組織 (catalog organization) は、「グローバル・アイテム ID」を提供できる機能を備えます。

上記の質問に対して肯定的に回答できる場合、その会社は在庫組織としてモデル化できます。

通常、在庫組織は、組織全体の可視性を最大化するために、可能な限り高いレベルでモデル化する必要があります。

## 電機会社にカタログ組織 (catalog organization) を選択する

カタログ組織 (catalog organization) を定義する必要のある階層レベルは、以下に基づいて決定します。

- カタログ組織 (catalog organization) のレベルは、在庫組織と同じか、それより上である必要がある。
- 他のセットアップと同じく、カタログ組織 (catalog organization) をより高いレベルでセットアップすると、セットアップがより少なくなり、共通定義の共有が容易になる。
- 決定する際、ビジネス・グループ (business group) またはその下のレベル内で固有のグローバル ID を保有するために、アイテム ID を合理化できるかどうか判断する。アイテム ID はカタログ組織 (catalog organization) で固有であるため、この定義をより高いレベルに上げるためには、これは必須要件になります。

## モデル例 2: サード・パーティーの物流管理会社

3PL は、サード・パーティーの物流管理会社です。3PL のクライアントは ABC と XYZ です。

3PL は、自社の倉庫で ABC と XYZ の在庫を保管し、ABC と XYZ のオーダーに対応します。ABC と XYZ のビジネスは、非常に似ており、物理的に 2 つの異なるアイテムに同じアイテム ID が設定される場合があります。

3PL の収益は、クライアントの販売オーダーと購入オーダーの処理に対するサービス手数料から発生します。3PL は、サービス手数料をクライアントから集めるため、オーダー処理の詳細を自社の財務組織に送信します。

3PL の倉庫の在庫の所有者はクライアントです。3PL には、カリフォルニアに 1 つ、マサチューセッツに 1 つの倉庫があります。3PL は、両方のクライアント (ABC と XYZ) が所有する製品を同じ倉庫で保管しています。

ABC と XYZ の組織構造は、前出の例の説明と同様ですが、以下の違いがあります。

- 事業は北アメリカのみで行います。
- 配送センター (distribution center) (ソーシング場所) は所有していません。在庫は 3PL のロケーションで保管されます。
- ABC の 2 つのセールス組織である電機とエンターテインメントは、相互の在庫に対する可視性を必要としません。しかし、XYZ は異なります。

クライアントのそれぞれが 3PL の倉庫内の在庫を補充するには、購入オーダーを生成します。クライアントは購入オーダーをベンダーに送信します。また、同じ購入オーダーのコピーを 3PL にも送信し、3PL で製品を入庫できるようにします。

オーダーは、3PL によって管理されますが、クライアント (ABC と XYZ) には自社のオーダーに対する可視性が必要です。また、クライアントは一定のオーダー変更権限を必要としていますが、すべてについてはありません。倉庫オペレーターもオーダーに対する可視性を必要としており、返品を受け入れることができる必要もあります。

### サード・パーティーの物流管理会社の組織階層

48 ページの図 14 には、3PL Company の組織階層の例が示されています。

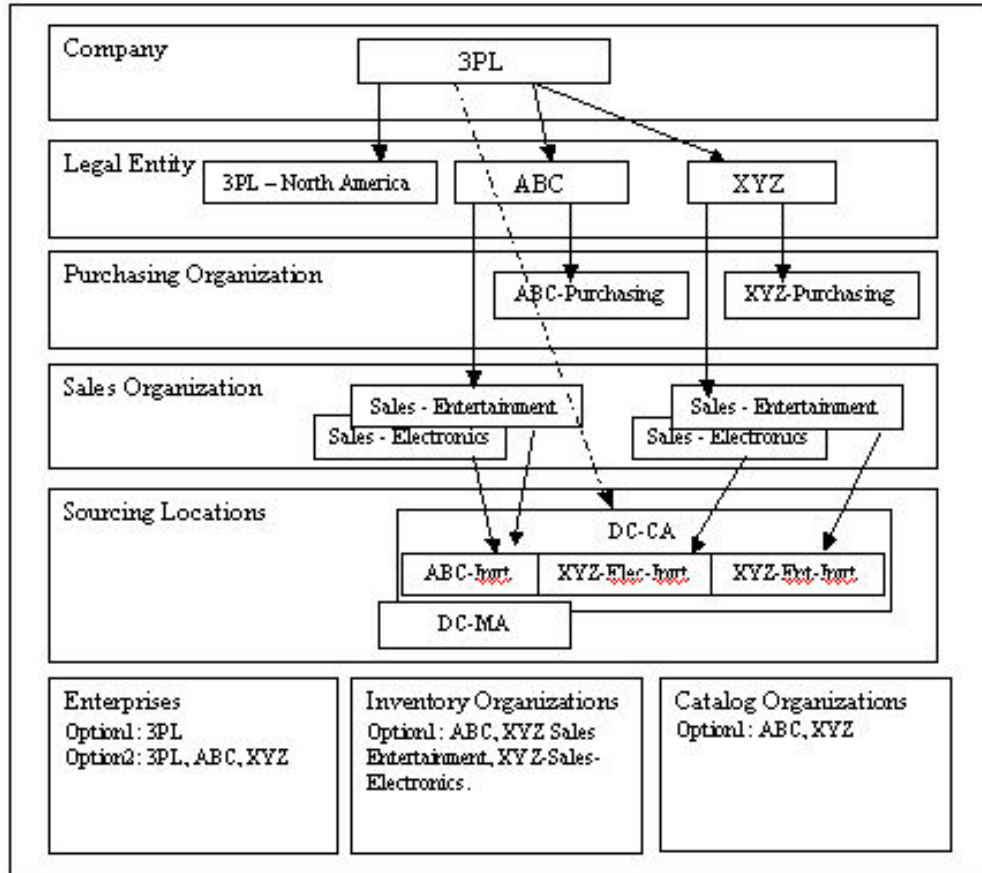


図 14. 3PL Company の組織階層

### サード・パーティーの物流会社にエンタープライズを選択する

前の例で説明したように、これはビジネス・ルール (business rule) 定義、オーダー・フルフィルメント・プロセス定義およびユーザー権限を、どのレベルで制御したいかによって決まります。

3PL は、クライアントのオーダーについて、ビジネス・ルール (business rules) およびオーダー・フルフィルメント・プロセスを定義し、管理します。したがって、3PL のみをエンタープライズとして定義するのが、適切なオプションになると考えられます。しかし、ABC および XYZ には、自らのオーダーに対する可視性および特定のオーダー変更許可が必要です。また、ABC および XYZ が自らのユーザーを作成し管理したい場合もあります。これは、ABC および XYZ も特定のエンタープライズ・レベルの機能が必要であることを意味します。したがって、3PL、ABC、および XYZ のすべてをエンタープライズとして定義することが、より適切なオプションになります。ただし、このオプションの場合、3PL は ABC および XYZ の権限を制限し、ABC および XYZ がオーダー・フルフィルメント・プロセスを定義および管理できないようにする必要があります。

ABC または XYZ で、あるセールス組織の特定ユーザー (例えば、顧客サービス担当者 (customer service representatives)) に、他のセールス組織のオーダーに対する可



視性を持たせない場合、それぞれのセールス組織に、権限を制限したエンタープライズの役割を割り当てることもできます。

### **サード・パーティーの物流会社に在庫組織を選択する**

前の例で述べたように、このモデルにおける在庫組織は、ABC、XYZ Sales-Electronics および XYZ Sales-Entertainment になります。これは、XYZ が、自らのセールス組織に対して他のセールス組織の在庫を表示させないにするためです。

同じ物理ロケーションを複数のクライアント間で共有するサード・パーティー物流などの状況では、各クライアントを「在庫組織」として指定する必要があります。3PL が同じクライアントの複数の組織を取り扱っている場合、クライアントまたはクライアント組織が要求する分割レベルに基づいて、在庫組織を選択できます。

### **サード・パーティーの物流会社にカタログ組織 (catalog organization) を選択する**

カタログ組織 (catalog organization) は、在庫組織と同じか、それより上のレベルで定義できます。XYZ の 2 つのセールス組織は、2 つの異なるアイテムに対して同じアイテム ID を保有しないため、カタログ組織 (catalog organization) は ABC と XYZ にすることができます。



---

## 第 4 章 プロセス・モデリングの概念

---

### プロセス・モデリングの概念

Sterling Selling and Fulfillment Foundation のプロセス・モデリングを使用し、オーダー、在庫の変更、返品、支払承認、またはその他の多くのシステム・イベントのビジネス・ワークフローをセットアップできます。

標準的なビジネス・プロセス・モデル (business process model) は以下で構成されます。

- 基本伝票種別および伝票種別 (document type)
- プロセス・タイプ・パイプライン
- リポジトリ
- トランザクション
- 条件
- アクション
- サービス
- プロセス・モデリング・タスク

---

### 基本伝票種別と伝票種別 (document type)

Sterling Selling and Fulfillment Foundation は、基本伝票種別および伝票種別 (document type) を使用して、構成済みのワークフロー・プロセス経由で情報を運びます。基本伝票種別は、Sterling Selling and Fulfillment Foundation が取り扱うビジネス・ドキュメントを定義し、すべての派生伝票種別に対して共通ストレージ構造を定義します。

Sterling Selling and Fulfillment Foundation では、以下の基本伝票種別が定義されています。

- オーダー
- 集合・混載
- 一般
- 棚卸
- 容器
- 出庫ピッキング
- 作業オーダー
- 案件

伝票種別 (document type) は、基本伝票種別から派生する固有のビジネス・ドキュメントです。例えば、販売オーダーや購入オーダーなどの伝票種別 (document type) は、オーダー基本伝票種別から派生することができます。

Sterling Selling and Fulfillment Foundation では、以下の基本伝票種別が定義されています。

- 計画済みオーダー
- 販売オーダー
- 購入オーダー
- 返品
- テンプレート・オーダー
- 転送オーダー
- マスター・オーダー
- 集合・混載
- 一般
- 棚卸
- 容器
- 出庫ピッキング
- 作業オーダー
- 見積

支払集金ルールや変更ルールなどのビジネス・ルール (business rules) は、各伝票種別 (document type) に対してセットアップする必要があります。伝票種別 (document type) のビジネス・ルール (business rules) のセットアップについては、

「Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド」を参照してください。

---

## プロセス・タイプ・パイプライン

Sterling Selling and Fulfillment Foundation では、オーダーなどのビジネス・ドキュメントは一連の定義済みプロセスを通過します。これらのプロセスは、基本プロセス・タイプと呼ばれます。あらゆるタイプの基本伝票には、基本プロセス・タイプの定義済みセットがあります。例えば、オーダー基本伝票種別には、以下の基本プロセス・タイプが定義されています。

- フルフィルメント
- 交渉
- 配達

集合・混載基本伝票種別には、集合・混載基本プロセス・タイプが定義されています。

汎用基本伝票種別には、汎用基本プロセス・タイプが定義されています。

プロセス・モデリングでプロセス・タイプ・パイプラインを作成し、これらのプロセスのフローを構成します。プロセス・タイプ・パイプラインとは、販売オーダーおよび購入オーダーなどの伝票種別 (document type) を、関連プロセスによってガイドする一連のトランザクションとステータスのことです。パイプラインは、フルフィルメント時にドキュメントが経るさまざまなステータスで構成されます。構成するパイプラインに関連するトランザクション (イベント、アクション、および条件から成る) を設定することもできます。

次の表は、各基本プロセス・タイプ、各基本プロセス・タイプから派生するプロセス・タイプ、およびプロセス・タイプに関連付けられているパイプラインを詳しく示したものです。

表2. オーダー・フルフィルメント基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
オーダー・フルフィルメント	販売オーダー・フルフィルメント
計画済みオーダー実行	計画済みオーダー実行
テンプレート・オーダー	なし
返品物流	返品物流、消費者返品
購入オーダー実行	購入オーダー実行、ドロップ出荷購入オーダー実行
転送オーダー実行	転送オーダー実行
見積フルフィルメント	見積フルフィルメント

表3. オーダー交渉基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
マスター・オーダー・フルフィルメント	マスター・オーダー・フルフィルメント
オーダー交渉	オーダー交渉
計画済みオーダー交渉	計画済みオーダー交渉
購入オーダー交渉	購入オーダー交渉

表4. 受入基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
返品受入	返品受入
購入オーダー受入	購入オーダー受入
転送オーダー受入	転送オーダー受入
販売オーダー受入	販売オーダー受入

表5. オーダー配達基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
出荷	出荷
入荷	入荷

表6. 集合・混載配達基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
集合・混載実行	集合・混載実行

表7. 汎用基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
全般	なし

表 8. 「梱包プロセス」プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
パック・プロセス	パック・プロセス

表 9. 棚卸実行プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
棚卸実行	棚卸実行

表 10. 出庫ピッキング・プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
出庫ピッキング	標準ピック・プロセス

表 11. VAS プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
VAS	VAS 作業オーダー

表 12. 案件基本プロセス・タイプ

派生プロセス・タイプ	プロセス・タイプ・パイプライン
案件フルフィルメント	案件フルフィルメント

## パイプラインの決定

パイプラインの決定は、ワークフローで使用されるパイプラインの選択に影響する条件のセットアップに使用されます。例えば、組織はときには危険物が含まれる販売オーダーを扱うことがあります。この組織には、2つのパイプラインがあります。1つは危険物が含まれないオーダー明細が記載されたオーダー用で通常のオーダープロセスに沿って処理され、もう1つは危険物を含むオーダー明細が記載されたオーダー用でオーダープロセスの処理を続行する前に検査 (inspection) を受ける必要があります。組織は、パイプラインの決定を使用して、オーダー明細に危険物が含まれるかどうかを判別してオーダーを適切なパイプラインに送る条件をセットアップします。

---

## リポジトリー

リポジトリーとは、特定のビジネス・プロセスを定義する、エンティティの論理的な集合体です。

リポジトリーには、以下のエンティティが含まれています。

- パイプライン
- トランザクション
- ステータス
- 条件
- アクション
- サービス

Sterling Selling and Fulfillment Foundation は、各プロセス・タイプにリポジトリを提供します。基本プロセス・タイプから新しいプロセス・タイプを作成する場合、対応する基本リポジトリ・エンティティがコピーされ、新しいプロセス・タイプに付加されます。例えば、販売オーダー・フルフィルメント・プロセス・タイプをフルフィルメント基本プロセス・タイプから作成した場合、フルフィルメントに含まれる基本リポジトリ・エンティティがコピーされて、販売オーダー・フルフィルメントに付加されます。

---

## トランザクション

各基本プロセス・タイプには、独自に定義された基本トランザクションのセットがあります。トランザクションとは、Sterling Selling and Fulfillment Foundation 内でアクティビティを実行するために必要な作業論理単位です。基本トランザクションとは、トランザクションの動作に関する情報 (プロセス・タイプにトランザクションのコピーをいくつ保持するか、構成可能な基本ピックアップ/ドロップ・ステータスをプロセス・タイプに含めるかどうかなど) を含む、事前定義されたトランザクションです。基本トランザクションは、新規のトランザクションを作成する場合に使用できます。これらのトランザクションは、基本トランザクションに定義された制限内で変更することができます。

Sterling Selling and Fulfillment Foundation では、API はトランザクションを完了するために使用されます。API を呼び出すと、API が完了したコンテキストに基づいてトランザクション ID が決定されます。トランザクション ID は、完了するトランザクションを識別します。状況に応じて、トランザクション ID を入力パラメーターとして渡すことができますし、または、呼び出し側 API に対して事前定義することもできます。API について詳しくは、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

トランザクションは、以下のタイプのいずれかとして分類できます。

- 外部トリガー
- ユーザー・トリガー
- 時間トリガー

### 外部でトリガーされたトランザクション (externally-triggered transaction)

外部でトリガーされたトランザクション (*externally-triggered transaction*) は、Sterling Selling and Fulfillment Foundation サービス定義フレームワークを使用して実行されます。このフレームワークは、トランザクションを完了するために Sterling Selling and Fulfillment Foundation 内で対応する API を呼び出します。

### ユーザー・トリガー・トランザクション

ユーザー・トリガー・トランザクションは、Sterling Selling and Fulfillment Foundation のユーザー・インターフェース、構成済みの警告キュー、または E メール交換で実行されるユーザー・アクションに基づいて実行されます。

## 時間トリガー・トランザクション

時間トリガー・トランザクションは、スケジュール済みの間隔で実行されます。Sterling Selling and Fulfillment Foundation では、時間トリガー・トランザクションはエージェントとも呼ばれます。

## トランザクション依存関係

オーダー・フルフィルメント・サイクルでは、特定のシーケンスで実行される必要があるオーダーに特定の製品およびサービスを含めることができます。ある条件を満たすまでオーダー明細を処理できないように、トランザクションの依存関係を設定できます。依存関係では、オーダー明細に関して定義できるアイテム分類、アイテム ID、サービス・タイプなどの多くの条件に基づいてルールを定義できます。

トランザクション依存関係は、特定のエンタープライズ、伝票種別 (document type)、およびプロセス・タイプ・レベルにおいて構成できます。例えば、スケジュール・トランザクションに関してエンタープライズによって指定される依存関係は、使用されるパイプラインに関係なく、そのエンタープライズの販売オーダーすべてに適用されます。

処理前、各オーダー明細は、トランザクションの依存関係に関してチェックされます。複数の依存関係がトランザクションに対して定義されている場合、明細を処理するには、すべての依存関係を満たす必要があります。

依存関係は、バンドル・コンポーネントに対しても定義でき、任意のオーダー日に基づくようにできます。バンドル・コンポーネントの場合、オーダー明細が、バンドル・オーダーを満たすと同時に、相互依存関係を持っていることがあります。バンドル親明細は、各子明細が指定されたトランザクションを完了するまで請求できません。したがって、一定の条件が満たされるまでオーダー明細を処理できないことをトランザクションが理解できるようにする必要があります。バンドル・コンポーネントについて詳しくは、125 ページの『バンドル』を参照してください。

オーダー日関連の依存関係は、特定のオーダー日に基づいてトランザクションに対して構成できます。Sterling Selling and Fulfillment Foundation に付属のテンプレートを使用してこれらの依存関係を指定できます。この依存関係の構成について詳しくは、「Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド」を参照してください。

## トランザクション依存関係の使用法のシナリオ

オーダーのトランザクション依存関係は、以下のシナリオで使用できます。

MODEM、FILTER、または CABLE のアイテム分類のいずれのオーダー明細の場合も、LINE ACTIVATION の分類のアイテムを含むすべてのオーダー明細が出荷オーダー・トランザクションを完了してから 24 時間経過するまで、スケジュール・トランザクションの処理は許可されません。

この依存関係が指定されると、スケジュール・トランザクションは、まずオーダーに MODEM 分類のアイテムがあるかどうかをチェックします。このアイテムが識別されると、オーダーの他の明細がチェックされ、それらの明細のいずれかに LINE ACTIVATION クラス・アイテムがあるかどうかを確認します。LINE ACTIVATION



のアイテム分類を含む各明細に対して、その明細が「SHIP\_ORDER」トランザクションに到達してから 24 時間以上経過したかどうかが判定されます。

## 他のトランザクションを一時停止するトランザクション依存関係

トランザクション依存関係がオーダー・レベル・トランザクション (決済処理など) を一時停止した場合、依存関係のためにオーダーを処理できないことを示すフラグが API の出力 XML に返されます。このフラグは、すべての API の出力 XML には返されません。詳しくは、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。ほとんどのトランザクションはステータス・ベースであるため、依存関係を満たさない明細は無視されます。ただし、(依存関係によって全部または一部を) 処理できないオーダーをエージェントが検出した場合、エージェントは、処理できない明細が何であれ、これを暗黙的に無視し、タスク・キューを更新して、後でそれらの明細を処理します。

## 依存関係のオーバーライド

トランザクション依存関係をオーバーライドするために、トランザクション依存関係が無視されるよう、トランザクションを呼び出す API にフラグを渡すことができます。このフラグは、一部の API でのみサポートされています。API について詳しくは、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

## 循環依存関係

循環依存関係は、あるトランザクションが、メイン・トランザクションがそのステータスを変更するか、または完了するのを待機している別のトランザクションに依存している場合に発生します。循環依存関係が原因でトランザクションがオーダーを処理しないような方法で依存関係をセットアップできます。

例えば、ITEM2 がスケジュール・トランザクションを完了するまで ITEM1 をスケジューリングできないというルールがあるとします。ITEM1 がスケジューリングされるまで ITEM2 のスケジューリングを妨げる別のルールがあるとします。その結果、循環依存関係が発生します。この場合、トランザクション依存関係ロジックによって循環依存関係が検出され、この循環ループの原因である依存関係のリストとともに例外がスローされます。

## サポートされるトランザクション

依存関係をサポートするシステム・トランザクションは、基本プロセス・タイプのフラグによって識別されます。

このフラグは、システムまたは組み込みトランザクションのために編集できません。依存関係を、拡張トランザクションまたはカスタム・トランザクションについてのみ使用可能または使用不可にできます。システム・リスナーおよびカスタム・リスナーは、トランザクションの依存関係をサポートしないことに注意してください。ただし、リスナーの各インスタンスに対してトランザクション完了を構成できます。トランザクション完了の構成について詳しくは、「*Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド*」を参照してください。

サポートされるトランザクションは以下のとおりです。

- 連鎖オーダーの作成 - このトランザクションは、依存関係を満たさないオーダー明細を無視します。
- オーダーを閉じる - このエージェントは、オーダー明細のいずれかが依存関係を満たさない場合には、オーダーを無視します。
- 支払集金 - この API は、いずれかの明細が依存関係を満たしていない場合に、依存関係がトランザクションの処理を妨げていることを示すフラグを返します。ただし、このエージェントは、現在のオーダーを無視し、次の実行時にこれを処理します。
- 支払実行 - この API は、いずれかの明細が依存関係を満たしていない場合に、依存関係がトランザクションの処理を妨げていることを示すフラグを返します。ただし、このエージェントは、現在のオーダーを無視し、次の実行時にこれを処理します。
- リリース - このトランザクションは、条件を満たさない明細を無視し、依存関係が原因で処理されなかった明細のスケジュール失敗理由を投入します。
- スケジュール - このトランザクションは、条件を満たさない明細を無視し、依存関係が原因で処理されなかった明細のスケジュール失敗理由を投入します。

抽象化トランザクションには、サポートされる依存関係のタイプを示す別のフラグがあります。

抽象化トランザクションがサポートする依存関係の 2 つのタイプは、以下のとおりです。

- なし - トランザクションの個別インスタンスのサポート依存関係は、常に N です。
- インスタンス固有 - サポート依存関係は、トランザクションの各インスタンスに対して構成可能です。

デフォルトでは、すべての抽象化トランザクションのこの依存関係フラグは、なしに設定されています。ただし、以下の抽象化トランザクションは、固有のインスタンスをサポートし、依存関係を満たさない明細を無視します。

- オーダー・ステータスの変更
- オーダー・インボイスの作成
- 派生したオーダー (derived order) の作成
- オーダーの送信

トランザクション依存関係の構成、関連ルール、および制約については、「*Sterling Selling and Fulfillment Foundation 分散オーダー管理 構成ガイド*」を参照してください。

## タスク・キューの更新

タスク・キュー・ベースのエージェントによるトランザクションの処理が依存関係によって妨げられた場合、オーダーの処理が準備完了になる日付が計算されます。

トランザクションの完了時に、このトランザクションの完了を待っている明細があるかどうかを確認するために、依存関係が検証されます。次に、従属トランザクションのタスク・キュー日付が再計算されます。さらに、オーダー日、出荷日、または配達日に変更された場合、依存関係は同様に再評価されます。

次に利用可能になる日の計算の使用例のシナリオは、「次に利用可能になる日の計算」で説明します。

トランザクションのエージェントは、処理前に更新され、トランザクション依存関係をオーダー明細に適用します。オーダー明細が依存関係のために処理されない場合、エージェントは、タスク・キュー日付を更新します。

## 次に利用可能になる日の計算

次に利用可能になる日が計算され、オーダーの処理が準備完了になるのはいつかが特定されます。以下の例は、次に利用可能になる日の計算方法を提示します。

例えば、オーダーに 2 つのオーダー明細があるとします。2 番目のオーダー明細は、最初のオーダー明細が SHIP\_ORDER トランザクションを完了してから 24 時間後までスケジューリングできません。

最初のオーダー明細の SHIP\_ORDER トランザクションが完了するまで、そのオーダー明細をスケジューリングできる最も早い日は、以下のように設定されます。

sysdate + 24h。

最初のオーダー明細の SHIP\_ORDER トランザクションが完了すると、2 番目のオーダー明細をスケジューリングするタスク・キュー日付は、以下のように設定されます。

sysdate + 24h。

オーダー全体の次のタスク・キュー日付は、残りの依存関係それぞれの最も早い日付を計算することによって決定されます。各明細の最大値の下限がその日付で、これは以下のように設定されます。以下に例を示します。

- 明細 1 - 依存 1 - 最も早い日付 = sysdate + 5h
- 明細 1 - 依存 2 - 最も早い日付 = sysdate + 48h
- 明細 2 - 依存 1 - 最も早い日付 = sysdate + 12h

明細 1 の最も早い日付は、(sysdate + 5h) と (sysdate + 48h) の最大値または sysdate + 48h です。ただし、オーダー全体に関しては、明細間の最小値は、以下のように計算されます。

明細 2 の sysdate + 12h。

最後に、トランザクションがタスク・キュー日付を押し出した場合、この日付の最小値およびトランザクションのタスク・キュー日付が使用されます。例えば、一方の明細が依存関係のためにスケジュールできない場合、他方の明細がバックオーダーになります。このバックオーダーの再ログ間隔が残りの依存関係用に計算された時間より短い場合、代わりに、バックオーダーの再ログ間隔が使用されます。

## トランザクション完了

パイプラインで構成されるトランザクションは、正常実行後にオーダー明細のステータスを変更します。トランザクションは、ステータス変更に基づいて、オーダー明細に対して完了または未完了のいずれかです。すべてのトランザクションが完了を理解する必要はありません。これは、構成に基づいています。完了は、そのトラ

ンザクションが完了に対応できるよう構成できる場合にのみ記録されます。トランザクションは、完了に対応できるようになると、オーダー明細でステータス変更が発生するたびに、評価されて、完了または未完了とマークされます。場合によっては、ステータスのレベルを下げることで、完了したトランザクションを未完了とマークできます。

トランザクションを完了または未完了とマークする方法は 2 つあります。1 つの方法では、単一のステータスが提供され、オーダー明細が終わると必ずこのステータス・トランザクションを完了とマークします。トランザクションを完了とマークするもう 1 つの方法では、ステータスのリストを提供します。オーダー明細がこれらのステータスのいずれかである場合、トランザクションは完了とマークされます。ステータスのリストが構成されないようにするには、オーダーのライフサイクルの進行に合わせて、ステータスを昇順で構成することをお勧めします。

例えば、トランザクション X がトランザクション Y より前に完了する必要がある場合、トランザクション X は、トランザクション Y になるステータスより下位のステータスになる必要があります。

オーダー・フルフィルメント・パイプラインでは、トランザクションは完了できるように事前構成されています。システム・トランザクションの完了条件は変更できません。ただし、カスタム・トランザクションまたは派生トランザクションの完了を構成することはできます。また、派生リスナーの完了も構成できますが、リスナーの各インスタンスに対して実行する必要があります。

トランザクションが完了して、オーダーが従属トランザクションを処理できる状態になっていると判定されると、イベントがトリガーされます。また、このイベントは、オーダーが処理できる状態にあることをカスタム・トランザクションが認識できるようにします。

1 つのトランザクション完了によって、複数のトランザクションを準備が整った状態にすることができます。このイベントは、オーダー明細の完了を待っていたオーダー明細の各トランザクションに対して発生します。

トランザクション完了構成および発生するイベントについて詳しくは、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

## イベント

**イベント** は、ビジネス・プロセスにおいて具体的に発生するもので、多くの場合、ステータスが変更されたり、警告が生成されたりします。オーダーのリリースおよびオーダーのキャンセルはどちらも、イベントの例です。トランザクションでイベントが発生すると、アクションがトリガーされます。

## ステータス

**ステータス**は、ドキュメントがパイプラインを移動するときの変更前または変更後の実際の状態です。トランザクションには、「ドロップ・ステータス (drop status)」と「ピックアップ・ステータス」の 2 つのタイプのステータスを含めることができます。ドキュメントは、トランザクションおよびそのイベントが完了すると、**ドロップ・ステータス (drop status)** に移動します。**ピックアップ・ステータス**は、ドロ

ップ・ステータス (drop status) からドキュメントを取り出し、次のトランザクションへと移動させます。ステータスの例としては、「作成済み」や「スケジュール済み」があります。

---

## 条件

**条件**は、決定点に照らして伝票種別属性を突き合わせて、指定された属性と値の組み合わせに基づいて、ドキュメントを適切なパスにルーティングします。条件を作成可能な伝票種別属性は、Sterling Selling and Fulfillment Foundation で事前に定義されます。これらの属性は任意の組み合わせで使用できます。または特定の状況向けに適切なアプリケーション・ロジックを実行する条件を作成できます。

例えば、販売オーダー・フルフィルメント・プロセス・タイプ・パイプラインの特定の点で、オーダーに危険物が含まれるかどうかを判断するための条件をセットアップします。オーダーがパイプラインでこの条件に達すると、この条件が明確に「Yes」値または「No」値を満たすまで、オーダーはそれ以上移動できません。この例では、オーダーに危険物が含まれていない場合、値は「No」であり、オーダーは通常のパイプラインを経て続行します。オーダーに危険物が含まれている場合、値は「Yes」であり、オーダーは、危険物オーダーを扱うように構成されているオーダー・パイプラインの代替ブランチに送られます。

## 拡張 XML 条件

拡張 XML 条件は、入力 XML ドキュメントで動作し、XPath 式を使用して条件を評価するための宣言メカニズムを提供します。これらの拡張 XML 条件は、Greex ルールとも呼ばれます。拡張 XML 条件の構文は、XML ベースです。

拡張 XML 条件は、Greex 構文に基づく XML ベースの条件またはルールの定義に役立ちます。拡張 XML 条件の最も重要な機能の例は、以下のとおりです。

- XML 対応
- 名前空間のサポート
- 決定表のサポート
- 組み込みライブラリーのセット。
- XML 要素、ストリング、またはブール値を返すことが可能。

拡張 XML 条件構造は、複数の IF および ELSE ブロックを使用してネスト化することも、AND または OR 演算子を使用してグループ化することもできます。それぞれの式は、1 つ以上の関数呼び出しから構成されます。これらの関数は、ネスト化できます。つまり、関数のパラメーターを他の関数呼び出しにすることができません。

これらの関数のセットは、Greex ライブラリーの一部として提供されます。これらの関数は、XPath 式を介して、入力データで呼び出すことができます。関数パラメーターに定数を使用することもできます。Greex ライブラリーは、jar として使用できます。このライブラリーは、既存のプラットフォームにおいても、その他のコンポーネントにおいても、依存関係を含みません。したがって、サード・パーティー・ライブラリーとして使用されます。

表 13 では、Greex ライブラリーの一部として Greex フレームワークによって提供される関数を説明します。

表 13. Greex フレームワークの関数

関数	説明
intSum	この関数は、Integer 型の属性の XPath を引数として使用します。この関数は、指定された属性の値の合計を計算し、返します。
IsTrue	この関数は、属性の XPath を引数として使用します。この関数は、属性の値が true と評価された場合に、true を返します。
dateGreater	この関数は、2 つの Date オブジェクトを引数として使用します。この関数は、1 番目の日付が 2 番目の日付よりも大きい場合に、true を返します。
IntGreater	この関数は、2 つの Integer オブジェクトを引数として使用します。この関数は、1 番目の値が 2 番目の値よりも大きい場合に、true を返します。
stringBegins	この関数は、2 つの String オブジェクトを引数として使用します。この関数は、1 番目のストリングが 2 番目のストリングで始まっている場合に、true を返します。
dateMin	この関数は、Date 型の属性の XPath を引数として使用します。この関数は、属性の値を比較し、最も古い日付を返します。
doubleSum	この関数は、Double 型の属性の XPath を引数として使用します。この関数は、指定された属性の値の合計を計算し、返します。
isVoid	この関数は、属性の XPath を引数として使用します。この関数は、指定された属性の値が NULL または空のストリングの場合に、true を返します。
doubleGreater	この関数は、2 つの Double オブジェクトを引数として使用します。この関数は、1 番目の値が 2 番目の値よりも大きい場合に、true を返します。
equals	この関数は、2 つの Object を引数として使用します。この関数は、1 番目の Object が 2 番目の Object と等しい場合に、true を返します。
棚卸	この関数は、属性の XPath を引数として使用します。この関数は、指定された属性の値の回数を計算し、返します。
dateAdd	この関数は、2 つの Date オブジェクトを引数として使用します。この関数は、指定された属性の値の合計を計算し、返します。
equalsIgnoreCase	この関数は、属性の XPath を引数として使用します。この関数は、1 番目の値が 2 番目の値と等しい場合に、true を返します。比較するときに、大/小文字は無視されます。

Greex フレームワークは、以下のタイプの拡張 XML 条件または Greex ルールをサポートします。

- 通常拡張 XML 条件—このタイプの拡張 XML 条件では、複数の IF および ELSE ブロックを使用して、ネスト化された条件を指定できます。通常拡張 XML 条件は、XML 要素、ストリング、またはブール値を返すことができます。

これらのタイプの拡張 XML 条件は、拡張 XML 条件に対して複数の条件を定義する必要があるが、それぞれの条件が異なる属性に関連付けられている場合に便利です。

例えば、以下のような条件で拡張 XML 条件を作成する必要があるとします。

```
If Ordertype="WEB" and OrderQty="100", then Discount="20".
```

```
Else
```

```
If OriginalTotalAmout="1000" and OrderLine>"5", then
```

```
TaxExemptFlag="Y".
```

```
Else HoldFlag="True".
```

この場合、複数の条件があり、それぞれの条件が異なる属性に関連付けられています。1 つ目の条件は、Ordertype、OrderQty および Discount に関連付けられており、2 つ目の条件は、OriginalTotalAmount、OrderLine、および TaxExemptFlag に関連付けられています。

- 決定表ベースの拡張 XML 条件—このタイプの拡張 XML 条件では、単一の拡張 XML 条件でネスト化された条件を指定できます。1 つの定数値の代わりに定数値の配列を持つ拡張 XML 条件を 1 つのみ保有します。また、決定表ベースの拡張 XML 条件では、1 つの条件しか必要ありません。そのため、1 つの IF ブロックがあるだけで、ELSE ブロックはありません。この決定表ベースの拡張 XML 条件は、ストリングのみを返すことができます。

これらのタイプの拡張 XML 条件は、拡張 XML 条件に対して複数のネスト化された条件を定義するが、それぞれの条件が同じ属性に関連付けられている場合に便利です。このような場合、1 つの条件のみを書き込み、切り替え文のように動作するパラメーターのテーブルを用意できます。

例えば、以下のように、同じ属性タイプで異なるケースを含む条件に対して、拡張 XML 条件を作成する必要がある場合です。

```
If Ordertype="WEB" and OrderLineQty="200", then Discount="5".
```

```
Else
```

```
If Ordertype="STORE" and OrderLineQty="500", then Discount="7"
```

```
Else
```

```
If Ordertype="CALL" and OrderLineQty="250", then Discount="3".
```

```
Else
```

```
default="0"
```

表 14 では、決定表の形式で上記のシナリオを説明します。

表 14. 決定表

オーダー・タイプ	オーダー明細の数量	割引
WEB	200	5
STORE	500	7
CALLCENTER	250	3
デフォルト	0	

この場合、複数の条件があり、それぞれの条件が同じ属性に関連付けられています。すべての条件が `Ordertype`、`OrderLineQty`、`Discount` の各属性に関連付けられています。定義された条件を満たすものがない場合、Greex エンジンではデフォルト値を返します。

拡張 XML 条件または Greex ルールは、Sterling Greex Editor という名前の IDE ツールを使用してのみ、作成できます。拡張 XML 条件は、拡張 XML ファイル (Greex ファイルとも呼ばれる) に定義されます。Sterling Greex Editor を使用した拡張 XML 条件の作成について詳しくは、「*Sterling Selling and Fulfillment Foundation* 条件ビルダーの拡張」を参照してください。

既存の拡張 XML 条件または Greex ルールの構造を、Applications Manager を使用して変更することはできません。新しい値を拡張 XML 条件の変更可変パラメーターに割り当てることができるだけです。拡張 XML 条件の変更について詳しくは、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

Greex ファイルをローカライズすることによって、拡張 XML 条件または Greex ルールをローカライズできます。Greex ファイルは、BundleReslover インターフェースを実装することで、ローカライズできます。Greex ファイルのローカライズについて詳しくは、「*Sterling Selling and Fulfillment Foundation* ローカライズ・ガイド」を参照してください。

また、GreexLogger インターフェースを実装することで、拡張 XML 条件または Greex ルールに関する情報を異なるレベルでログに記録することも可能です。拡張 XML 条件または Greex ルールの情報のログへの記録について詳しくは、「*Sterling Selling and Fulfillment Foundation* 条件ビルダーの拡張」を参照してください。

---

## アクション

アクションは、イベントにより起動されるプロセスまたはプログラムです。これらのプロセスやプログラムは、警告通知を送信し、問題を自動的に解決します。

例えば、オーダーがリリースされるときに (イベント)、顧客に E メール・メッセージを送信するアクションを設定できます。

---

## サービス

サービス は、Sterling Selling and Fulfillment Foundation と外部システム間のビジネス・プロセス・フローを定義します。



---

## プロセス・モデリング・タスク

プロセス・モデリングを完了するために必要なタスクには次が含まれます。

- プロセス・タイプ・リポジトリのロード
- パイプラインの作成および変更
- トランザクションの作成、変更、および削除
- トランザクションへのイベントの追加
- トランザクションのイベントの変更および削除
- トランザクションへのピックアップ・ステータスの追加
- トランザクションからのピックアップ・ステータスの削除
- トランザクションへのドロップ・ステータス (drop status) の追加
- トランザクションからのドロップ・ステータス (drop status) の削除
- イベント処理の設定
- ステータスの作成、変更、および削除
- ステータス・モニター・ルールの定義の設定
- 条件の作成、変更、および削除
- 条件の影響を受けるすべてのエンティティの表示
- アクションの作成、変更、および削除



---

## 第 5 章 カタログ管理概念

この章の情報は、カタログ管理概念のオンライン・ファイルに移動しました。



## 第 6 章 統合在庫管理

統合在庫管理 (Global Inventory Visibility) には、顧客が希望したときに製品が出荷に対応できるようにするために、在庫レベルの獲得およびモニターが必要です。

### 在庫 ID

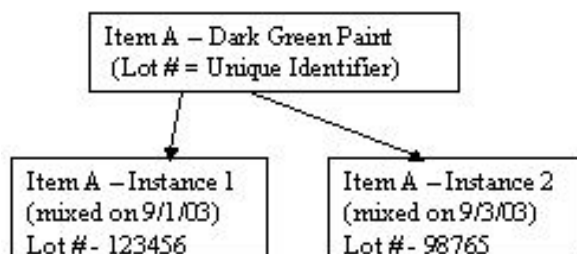
Sterling Selling and Fulfillment Foundation では、製品を差別化するために、在庫 ID 番号を使用します。これは、特性が少し異なる「製品インスタンス」について物理的にもシステムのにも区別する必要がある製品の場合も同じです。そのような ID 番号の一般的な例として、ロット番号、改訂番号、および製造バッチ番号または製造日付があります。各種の業界に、それぞれ固有の ID 番号があります。例えば、鋼鉄を扱ういくつかの会社では、工場認証番号で 1 巻の鋼鉄を識別します。

これらの ID 番号は、会社が販売する各製品に必ずしも関連しません。製品によっては、ロット番号が製品のすべての特性を一意的に定義するのに対し、別の製品は改訂番号で差別化します。Sterling Selling and Fulfillment Foundation では、用語「在庫タグ番号」を使用して、これらの固有の製品 ID 状態を合理的に説明します。ほとんどの場合、在庫タグ番号は、製品に基づくロット番号、バッチ番号、または改訂番号などの現実の製造またはカスタマイズ済み番号のいずれかを表すことが予想されます。製品拡張が小さいものであれば、在庫タグ番号は、2 つの在庫 ID 番号が共に一意的に製品を識別する場合に ID 番号の組み合わせを意味することもできます。

多くの場合、在庫タグ番号は Sterling Global Inventory Visibility によって使用され、物理的な製品には実際の関連在庫 ID 番号が付けられます。

### 在庫タグ番号として使用されるロット番号

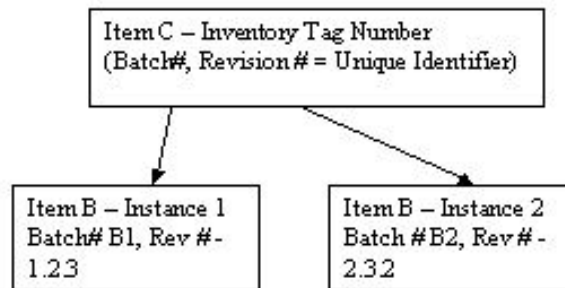
次の図は、同時に混合された特定のブレンドの識別にロット番号を使用する塗料製品を示しています。この例では在庫タグ番号がロット番号と同じになっています。



在庫タグ番号が製品 ID、または製品 ID の組み合わせと異なる場合、Sterling Selling and Fulfillment Foundation では、そのアイテムについて両方の ID が保管されます。

## バッチと改訂のタグ番号の組み合わせ

以下の図は、製品を一意的に特定するためのバッチ番号と改訂番号の組み合わせを使用して、製品を示しています。



この場合、Sterling Selling and Fulfillment Foundation は、在庫アイテム (inventory item) に関して、改訂番号およびバッチ番号の両方を保管します。在庫タグ番号の生成、および必要に応じてそれらの番号の照会に使用される、getTagNo() ユーザー EXIT が提供されています。このユーザーEXIT は、独自のロジックを備えており、2 つの ID をマージまたはその組み合わせを別のテーブルに保管し、外部タグ番号を返します。この外部タグ番号は、Sterling Selling and Fulfillment Foundation で固有の在庫 ID として使用できます。

Sterling Selling and Fulfillment Foundation は、タグ情報を保管し、すべての出荷について、タグ番号と出荷期限を追跡することができます。ただし、ノードで Sterling Selling and Fulfillment Foundation ソフトウェアを使用しておらず、タグ情報を追跡できない場合、出荷した製品の詳細を Sterling Selling and Fulfillment Foundation に通知することはできません。そのようなノードについては、タグ追跡がサポートされません。

Sterling Selling and Fulfillment Foundation では、タグ番号と出荷期限は、完全に無関係です。ロット番号をアイテムに割り当てても、出荷期限がアイテムに割り当てられることはありません。

---

## 供給と需要

Sterling Selling and Fulfillment Foundation では、アイテムの供給は、ノードで受入済みのアイテムの全数量です。供給には、手持ちの供給も含まれます。供給は、ノードによって受入済みの購入オーダー (PO) および事前出荷通知 (ASN) (advance shipment notice (ASN)) から構成されます。手持ちの供給の他に、供給には、ベンダーから到着していない、以前行われた購入オーダーである将来の在庫 (future inventory) も含まれます。

Sterling Selling and Fulfillment Foundation では、アイテムの需要は、多量のアイテムの消費を要望する表明です。需要には、アイテムに対して行われたオーダーおよび予約が含まれます。

需要は、十分な供給が存在する場合のみ満たすことができます。使用可能な数量は、アイテムのすべての需要が満たされた後に残された供給の量です。したがって、使用可能な在庫は、供給と需要の差です。

## 予約

Sterling Selling and Fulfillment Foundation では、需要は、「需要」と呼ばれるさまざまなタイプの特定のエンティティで構成されます。需要のタイプの 1 つが予約です。在庫が顧客のために確保されていることを顧客に保証するには、どうすればよいでしょうか。その方法の 1 つとして、システムで予約を作成する方法があります。顧客は、その予約を確認またはキャンセルする必要があります。予約とは、後日アイテムを購入する意向の顧客のためにセラーが確保しているアイテムの数量です。これによって、顧客の特定の需要を満たすために、利用可能な在庫から特定の数量の在庫が差し引かれます。予約はオーダーにアップグレードされることもあれば、キャンセルされることもあります。

例えば、除雪用ショベルを製造しているとします。顧客である ABC Hardware ストアが 11 月 5 日用に 150 個の除雪用ショベルを「予約」したとします。この場合、150 個の除雪用ショベルのオーダーを「予約済み」ステータスで作成します。ABC Hardware ストアは予約をキャンセルすることも、除雪用ショベルを出荷してもらうこともできます。

予約有効期限の日時を渡すことで、予約の有効期限が切れるようにスケジュールすることができます。有効期限日を個々のオーダーに指定することも、デフォルトの有効期間時刻を構成して、その時刻を過ぎたら既存の予約をキャンセルすることもできます。この機能を使用して、なんらかの理由でキャンセルされていない予約をクリーンアップできます。

**注:** セグメント化されている在庫を予約することはできません。セグメント在庫について詳しくは、『セグメンテーション』を参照してください。

お願い

## セグメンテーション

Sterling Selling and Fulfillment Foundation では、供給はさまざまなタイプの特定のエンティティで構成されます。「供給」の 1 つのタイプが在庫セグメンテーションです。在庫が顧客のために確保されていることを顧客に保証するには、どうすればよいでしょうか。在庫セグメンテーションとは、在庫を複数のセグメントに割り当てることです。在庫セグメントは、特権を持つ顧客グループからの需要に提供するために確保しておいた一定量の在庫です。この在庫は、コミットメントおよび契約を履行するために必要なものです。在庫がセグメントに割り当てられている場合、これはその在庫が、一致するセグメント、セグメント・タイプ、またはその両方を持つ需要以外の需要には消費されないことを示します。

例えば、シャンプーの製造業者またはサプライヤーだとします。大手薬局チェーンおよび百貨店チェーンと、各社の指定に従ってシャンプー・ブランドの 1 つをそれぞれ供給する契約を結んでいます。これらの会社がオーダーを発行すると、その需要は、事前に手配済みのセグメントまたはセグメント・タイプとともに記録されます。この需要は、シャンプー製造業者またはサプライヤーのノードでセグメントまたはセグメント・タイプが一致する供給から履行されます。

**注:** セグメント化されている在庫を予約することはできません。予約在庫について詳しくは、『予約』を参照してください。

お願い

## 在庫状況モニター

eコマース・ビジネスでは、多くの場合、Sterling Selling and Fulfillment Foundation を絶えず呼び出すことなく在庫ピクチャーのスナップショットを顧客に提供できるように、リアルタイムの在庫状況標識を必要とします。スナップショットをそのように提供できることは、オーダーが発行され、在庫が常に表示され、変更されている Web サイトでたいへん有用です。在庫状況標識の例は、「在庫あり」、「僅か (Low)」、「制限付き」、「バックオーダー/プレオーダー (Backorder/Pre-order)」、および「在庫切れ」です。各標識レベルの基準は、Applications Manager で定義できます。

Sterling Selling and Fulfillment Foundation は、リアルタイム在庫状況モニターを通じて、この機能を提供します。次の 3 とおりの方法で実行できます。

### アクティビティ・ベース・モード

このモードでは、Sterling Selling and Fulfillment Foundation では、在庫の変更がリアルタイムで追跡されます。指定アイテムの在庫レベルが Applications Manager のモニター・ルールに定義されたしきい値を超えたり、あるいは下回ったりすると、Sterling Selling and Fulfillment Foundation は、更新済みの在庫レベルを外部システムに公開します。

### 即時同期モード

このモードで稼働している場合、Sterling Selling and Fulfillment Foundation は、モニターが記録した最新の在庫状況情報を外部システムへ送信します。アイテムの在庫レベルが「在庫あり」から「僅か」になり、その後、「在庫あり」に戻った場合は、Sterling Selling and Fulfillment Foundation は、同期時に、そのアイテムについて「在庫あり」レベルを公開するのみです。

また、在庫状況情報には手持ち在庫状況および将来の在庫状況が含まれるため、このモードを使用して、計画とプロモーション・システムに在庫状況メッセージを送信することができます。

### 完全同期モード

一般に、エンタープライズは、これをスケジュール済みジョブとして通常、夜間に実行します。在庫状況情報は、すべてのアイテムについて、在庫状況の変更が済んでいるかどうかに関係なく、送信されます。

在庫情報が Sterling Selling and Fulfillment Foundation API またはサービスを介して Sterling Selling and Fulfillment Foundation にロードされていない場合、完全同期モードは、在庫状況モニターが初めて実行されるときに使用されると予想されます。

注: アクティビティ・ベース・モードで実行している場合でも、在庫状況モニターは、完全にはリアルタイムで作動しません。在庫の変更をデータベース・テーブルに公開する必要があり、これはモニターによって処理する必要があります。したがって、リアルタイムに近いものとして考えてください。



---

## 最適化

予想される受注に従って利用可能な在庫数量を微調整することで、在庫が最適化されます。在庫を最適化するには、「リード・タイム + 処理時間」という概念が役立ちます。

ある製品の在庫が倉庫に 20 ユニットあり、出荷可能な状態になっている場合を考えてみましょう。今から 12 日後にその製品を 20 ユニット出荷してほしいというオーダーを受けたとします。

その後、今から 6 日後に 10 ユニットを出荷するというオーダーを受け入れることができるかどうかを、別の顧客からたずねられました。このオーダーを受け入れた方がよいでしょうか。つまり、今から 6 日後に 10 ユニットの確保できますか。

この疑問に答えるには、以下のスケジュール・パラメーターが必要です。

- リード・タイム - 新しい入庫オーダーを発行してから、供給業者または製造業者から製品を入手するまでに要する時間。
- 入庫処理時間 - ノードへの入庫後、供給の処理に要する時間。
- 出庫処理時間 - ノードで製品を集め、流通加工サービスを実施して、製品を物理的に出荷するのに要する時間。

製品のこの 3 つのパラメーターの合計をリード・タイム + 処理時間といいます。リード・タイム + 処理時間内に、入庫オーダーを供給業者に送信し、供給をノードに入庫して、製品を処理し、出荷すると考えることができます。つまり、調達日がリード・タイム + 処理時間を超える需要は、現在の在庫状況に関係なく調達可能です。

例に戻ると、リード・タイム + 処理時間内に、今から 6 日後の 10 ユニットのオーダーと今から 12 日後の 20 ユニットのオーダーの両方に対応できる場合、10 ユニットのオーダーを受け入れることができます。

---

## 回転の速い在庫

場合によっては、現在、十分な在庫状況があっても、特定の日付を過ぎて在庫状況を保証しないことが最良である場合があります。該当アイテムが回転の速いアイテムであり、オーダーが出荷されるよりもはるか前に手持ち在庫がすべてなくなることが予想できる場合は、その将来のオーダー用に在庫を保留したくないことがあります。

この好例を、人気玩具の最近の売り出しに見出すことができます。小売店は、売り出し日から 2 週間の在庫状況を保証することを拒否しました。小売店は、オーダーが出荷される予定よりもはるか前に、すべての使用可能な予想在庫がなくなると、正しく見なしていたのです。

## 需要と供給の有効期間

アイテムの供給はアイテムが消費可能である期間に限って適切とされます。一部のアイテムは時間の経過とともに品質が低下します。この例としては、医薬、食料品、飲料および生花などが挙げられます。アイテムを消費在庫として保管できる期間は、保存期間と呼ばれます。有効期限 (expiration date) は、アイテムの保存期間の終了日を示します。

現在の供給を需要のフルフィルメントに使用できる時間枠は、有効期間と呼ばれます。供給品の劣化が原因で顧客への出荷ができなくなる境界の日付は、出荷期限と呼ばれます。出荷期限は常に供給の有効期限 (expiration date) 以前になります。

Sterling Selling and Fulfillment Foundation では、約束プロセスおよびスケジュールプロセス時に、時間の影響を受けるアイテムの出荷時における必要な残余有効期間を考慮して、オーダー作成時にユーザーが最小出荷期限を指定できます。最小出荷期限について詳しくは、155 ページの『最小出荷期限』を参照してください。

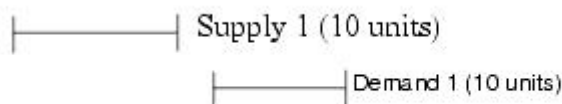
需要にも有効期間があります。例えば、顧客が本日、オーダー日付から 2 週間以内の配達を条件として製品をオーダーしたとします。この需要の有効期間は 2 週間です。

在庫管理で最も重要な側面の 1 つは、需要と供給の有効期間を管理し、理解することです。

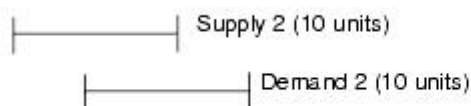
需要は、以下に該当する場合、かつその場合にのみ充足できます。

- 供給の数量が需要の充足に十分である。
- 1 つ以上の供給の有効期間が需要の有効期間とオーバーラップする。

以下の例を検討します。



供給 1 の有効期間は、需要 1 の有効期間とオーバーラップしません。したがって、この需要は充足されません。需要 1 は、充足可能な利用可能供給がなく、新たにオーダーされた供給がリード・タイム + 処理時間内に到着しないため、充足することはできません。



供給 2 の有効期間は、需要の有効期間とオーバーラップします。したがって、需要 2 は充足されます。

## 在庫集約

ハブ組織は、在庫アイテム (inventory item) をどのように識別して集約するかを決定するルールを指定します。ハブ (HUB) は、ハブ・レベルまたはエンタープライズ・レベルのどちらかでの集約を選択できます。在庫がハブ・レベルで集約される場合、アイテム ID および計測単位は、すべての組織全体で固有のものであると見なされ、在庫の分割はありません。エンタープライズ・レベル集約を選択すると、在庫が各エンタープライズによって分離されるものと見なされます。ハブ・レベル集約またはエンタープライズ・レベル集約の選択について詳しくは、「*Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド*」を参照してください。

在庫を集約するレベルを決定する場合、以下のシステム機能と、それらがどのような影響を受けるかを考慮する必要があります。

- ドロップ出荷 - 組織の販売オーダーをドロップ出荷するためにどのような組織を使用できるかを考慮する必要があります。いずれの組織も、それぞれの販売オーダーを、同じ在庫集約の一部である他の任意組織からドロップ出荷することができます。

注: まず最初に、正しい組織コードを `adjustInventory` API に渡して、所有者が別の在庫集約に属している出荷ノードについてのすべての在庫調整が確実に作成されるようにした場合、その出荷ノードへオーダーをドロップ出荷することができます。

注: 例えば、出荷ノード 1 を組織コード `DEFAULT` が所有している (在庫集約者 = `DEFAULT`) 場合、デフォルトで、(組織コードを呼び出さずに) この出荷ノードの調整在庫を呼び出すと、出荷ノード所有者の在庫集約者 (この場合 `DEFAULT`) に対して調整が行われます。

注: エンタープライズ 1 (在庫集約者 = エンタープライズ) が在庫を出荷ノード 1 から調達しようとする場合、`adjustInventory` API を呼び出す際に `OrganizationCode=Enterprise1` を渡すことによって、エンタープライズ 1 についての出荷ノード 1 におけるすべての在庫調整を行う必要があります。

- 在庫の表示可能性 - 同じ在庫集約の一部である組織はすべて、他のすべての組織の在庫に対してアクセス権を持っています。ただし、それぞれの固有の分配セットアップのみに限られます。
- 在庫モニター - 在庫および在庫状況モニターは、在庫集約者に対してのみ実行できます。

## ハブ・レベル集約

ハブ・レベル集約により、オーダーのドロップ出荷で最高の柔軟性が実現されます。いずれの組織も、他の任意の組織の出荷ノードを使用して、それぞれの販売オーダーをドロップ出荷することができます。組織の分配ルール・セットアップに基づき、ハブ (hub) 内のすべての組織全体でアイテムの在庫状況を検索できます。

ハブ・レベル集約を選択することによっても、組織の在庫がハブ (HUB) 内の他のすべての組織に公開され、在庫モニター、在庫状況モニター (availability monitor)、および FEFO ルールをハブ・レベルでのみセットアップすることができます。

## エンタープライズ・レベル集約

エンタープライズ・レベル集約により、オーダーの制限付きドロップ出荷が提供されます。いずれの組織も、販売オーダーがすべて同じ在庫集約を持っている限り、他の任意組織のノードを使用して、それぞれの販売オーダーをドロップ出荷することができます。組織の分配ルール・セットアップに基づき、同じ主要エンタープライズを持つすべての組織全体でアイテムの在庫状況を検索できます。

在庫は、エンタープライズごとに分離されているため、あるエンタープライズが他のエンタープライズの在庫を見ることはできません。各エンタープライズについて、在庫をモニターすることもできます。

また、1つの組織が複数のエンタープライズに参加することは可能ですが、その在庫集約者として1つのエンタープライズを選ぶ必要があり、在庫を維持しておけるのは、選んだそのエンタープライズのみです。

---

## ノード在庫との同期化

倉庫で在庫の変更が発生しても、Sterling Selling and Fulfillment Foundation が即時に認識しない可能性があります。例えば、商品が逸失または損傷した場合などです。このため、Sterling Selling and Fulfillment Foundation は、内部在庫ピクチャーとノードの在庫ピクチャーを定期的に一致させる必要があります。このプロセスは、以下の2つのステップに分かれます。

1. ノードからの在庫ピクチャーのロード

ノードの在庫ピクチャーが、Sterling Selling and Fulfillment Foundation にダウンロードされます。その情報は、一時表に格納されます。

2. Sterling Selling and Fulfillment Foundation 在庫ピクチャーと実際の在庫ピクチャーの同期化。

一時表と基本在庫表のデータが比較され、必要に応じて同期化されます。

このプロセスを完了したら、エージェントを構成して実行し、一時表をクリーンアップする必要があります。ノード在庫との同期の構成について詳しくは、「*Sterling Selling and Fulfillment Foundation 統合在庫管理 構成ガイド*」を参照してください。

---

## ノード需要との同期化

Sterling Selling and Fulfillment Foundation には、外部システムからの需要を同期化する機能が用意されています。Sterling Selling and Fulfillment Foundation は、一連の出荷ノード、需要タイプ、需要ピクチャーを入力として取ります。その後、入力として受け取った需要ピクチャーに基づいて、出荷ノードおよび需要タイプの需要情報をリセットします。

## 在庫コスト計算

Sterling Selling and Fulfillment Foundation は、Sterling Selling and Fulfillment Foundation を介して会計システムまたは調達システムからコストが渡されると見なして、特定の製品に対して在庫コストを管理する機能を提供します。Sterling Selling and Fulfillment Foundation は、返品アイテムのコストを決定するための手段も提供します。在庫コストを管理することにより、Sterling Selling and Fulfillment Foundation は、コスト情報を会計システムに渡し、「コスト中心の」レポート作成のためのリポジトリとして機能することができます。

Sterling Selling and Fulfillment Foundation を、ノード単位ベースで在庫コストを管理するようセットアップすることができます。各ノードについて、Sterling Selling and Fulfillment Foundation は、購入オーダー、アイテム、およびノード・レベルでのすべての在庫受入についてレコードを管理します。Sterling Selling and Fulfillment Foundation は、ノードからの出荷についてのレコードも管理します。

Sterling Selling and Fulfillment Foundation は、次のような 2 つの在庫コスト計算方法を提供します。

- 在庫コスト計算方法 1: 平均コスト計算
- 在庫コスト計算方法 2: 先入れ先出し (FIFO)

在庫値は、各アイテムの平均コストに基づくロケーションにあるすべての手持ち在庫の合計貨幣価値です。これは、手持ち分類への在庫の出し入れがあったときはいつでも更新されます。

Sterling Selling and Fulfillment Foundation は、以下の数式を使用して、新しい在庫値を計算します。

Old Inventory Value + Change in Inventory Value Due to Transaction

ここで、「Change in Inventory Value Due to Transaction」は次のようになります。

1. 正または負 (トランザクションが手持ち在庫からの受入を表すか、または送を表すかによる)
2. トランザクション伝票の transaction quantity \* unit value から計算される

Sterling Selling and Fulfillment Foundation では、製品明細または部門などによる掛け勘定の決定を変更するために、会計アプリケーションへの組み込みを提供します。アイテム定義中に指定される Posting Classification 属性は、アイテムを該当の会計元帳グループにグループ分けするのに使用されます。Sterling Selling and Fulfillment Foundation から一連の会計計上にトランザクションを変換する場合は、トランザクション内のアイテムの分類値はインターフェースで繰り上げられます。

### 在庫コスト計算方法 1: 平均コスト計算

平均コスト計算方法の場合、在庫調整は、レコードを突き合わせテーブルに計上する前にコスト計算係が手動で見直せるように累算されます。これにより、コスト計算係は、総計アイテム・レベルでコストを調整し、調整を相殺することができます。

Sterling Selling and Fulfillment Foundation 平均コスト計算機能には、次のものが含まれます。

- Sterling Selling and Fulfillment Foundation 内での在庫コストの定義
- 購入オーダー実行および販売実行中の関連コストの取り込み
- 正しく、かつ最新のコストが Sterling Selling and Fulfillment Foundation からターゲット会計 (A/P、A/R、および G/L) アプリケーションへ渡されることの確認
- Sterling Selling and Fulfillment Foundation でいつでも在庫の現在の貨幣価値を表示する機能

### 平均コスト計算方法: ユニット・コストの決定

ユニット・コストは、実際のコストに対する計測のベースとして使用される通常コストまたは指定されたコストを表します。製造されたアイテムのユニット・コストには、労働コスト、材料コスト、およびオーバーヘッド・コスト、ベンダー取得コスト、運賃、税金および購入アイテムに関するその他のカテゴリーが含まれます。物理キットのユニット・コストは、キットのすべてのコンポーネントのユニット・コストを合計し、アイテム・マスターでアイテムに定義されている WIP コスト要因を加算することで決定されます。WIP コスト要因がアイテム・マスターで使用不可の場合、カタログ組織 (catalog organization) の主要エンタープライズに定義された WIP コスト要因が使用されます。

ユニット・コストは、その計算に使用された基本コストに応じて、別に保管されません。

Sterling Selling and Fulfillment Foundation では、在庫価値の決定のために、以下の方法に基づいてユニット・コストを定義する機能があります。

- 置換
- 平均

**置換コスト:** 置換コストは、指定したサプライヤーから在庫を入手する際の単位あたりコストで、購入オーダーのデフォルトのコストとして使用されます。Sterling Selling and Fulfillment Foundation では、置換コストをサプライヤーの公開済み価格リストとして表します。

置換コストは、Sterling Selling and Fulfillment Foundation 内に、次の条件に応じて単位あたりのコストが変化する構造で保管されます。

- サプライヤー
- 日付

これをベンダー価格リストと呼びます。

Sterling Selling and Fulfillment Foundation には、次の方法で置換コストを使用する機能が備わっています。

- 購入オーダー明細のデフォルトのユニット・コストとして使用。これは手動でオーバーライドできます。
- ユニット・コスト計算用の基本コストとして使用。

**基本コストとして使用される置換コスト:** ユニット・コストはアイテム・レベルで保管されます。現行リリースでは、置換コストをサプライヤー・ロケーション・レベルで保持する方法は提供されていないため、置換コストがサプライヤーの出荷口

ケーションに応じて変わることはありません。置換コストが基本として使用されるため、ユニット・コストが各ロケーションに応じて変わることはなく、アイテム・レベルで保持できます。

Sterling Selling and Fulfillment Foundation には、API を通じてアイテムのユニット・コストを再計算するメカニズムが用意されています。この API は、置換コストに変更が加えられた場合 (またはその他の必要時) および Sterling Selling and Fulfillment Foundation のコスト決定ロジックが該当する要因に基づいてユニット・コストを再決定した場合、いつでも呼び出すことができます。Sterling Selling and Fulfillment Foundation は、置換コストが変わったときにユニット・コストを自動的に再計算することではなく、現状では、外部トリガーによってこの API を呼び出して、ユニット・コストを再計算する必要があります。

基本置換コストのソースを解決する必要がある場合、アイテム定義の際に、「主要サプライヤー」属性でこのアイテムの 1 次ソースを識別する必要があります。Sterling Selling and Fulfillment Foundation は、複数のロケーションから調達されているアイテムをサポートしています。この指定は、Sterling Selling and Fulfillment Foundation がサプライヤーからの調達時にコスト要因のリストを決定できるデフォルト・ソースを示します。

置換コストを基本コストとして使用すると、ユニット・コストは次のように再計算されます。

ユニット・コスト + 荷役コスト要因 + 標準コスト要因 (荷役コスト要因の追加後に適用されます) = 計算されたユニット・コスト

各要素の内容は次のとおりです。

- ユニット・コストとは、主要サプライヤーの価格リストからのユニット・コストです。
- 標準コスト要因とは、主要サプライヤーの標準コスト要因です。
- 荷役コスト要因とは、主要サプライヤーの荷役コスト要因です。主要サプライヤーの標準または荷役コスト要因のいずれかが定義されていない場合、主要サプライヤーの主要エンタープライズで定義されているコスト要因が使用されます。

コスト要因について詳しくは、83 ページの『平均コスト計算方法: コスト要因』を参照してください。

**平均コスト:** 平均コストは、特定のロケーションにおけるアイテムのコストに、運賃などの入庫コストを加えたものです。平均コストは、以下のときに再評価されます。

- 購入オーダーに対して、製品を受け入れたとき
- 「製造」操作を経て完成品を受け入れたとき。

**基本コストとして使用される平均コスト:** ユニット・コストは、アイテムおよびロケーション (Sterling Selling and Fulfillment Foundation 出荷ノード) のレベルで記録されます。ユニット・コストは、常にアイテム・レベルおよびロケーション・レベルでの金額として表されます。

ユニット・コストの再計算は、平均コストに変更があった場合、常に自動的にシステムによって行われます。この場合、外部トリガーは必要ありません。この場合と、置換コストが基本コストとして使用される場合との違いに注意してください。

**平均コストの計算 - 購入オーダー受入:** 平均コストは、購入オーダーの受入時に、以下の方法で計算されます。

$(\text{手持ち在庫価値合計} + \text{受け入れた荷役価値合計}) / (\text{新規の手持ち数量合計})$

ここでは、以下とします。

- 在庫価値は再計算済み
- 荷役価値合計 = 予定単価 \* 受入数量 + 関連コスト要因に基づく追加コスト。
- 購入オーダー上の単価はバイヤーのコストになる。

**注:** Sterling Selling and Fulfillment Foundation 供給コラボレーション・アプリケーションによって、ユニット・コスト以外の料金を指定できても、購入オーダー上に指定された単価のみが、受入価値合計の計算に使用されます。追加コストは、受入またはサプライヤーに適用可能なコスト・ロード要因に基づいて計算されます。

以下の例は、購入オーダー受入時の平均コストの計算を示しています。この例で参照されているコスト要因グループは、84 ページの『国際的なベンダーから輸入した製品のコスト要因グループ』に定義されています。この例では、コスト要因グループは、ベンダーに関連付けられています。アイテムの各ユニットの重さは 1 ポンドです。

#### 平均コストの計算 - 購入オーダー受入

受入前の現在の手持ち数量 = 500。

受入前の在庫価値 = \$2000。

1000 ユニットのアイテムの購入オーダー明細には、以下の価格が定義されています。

予定ユニット・コスト = \$3.00。

購入オーダー明細コスト合計 = \$3000。

関連コスト要因グループに基づいた追加コストは次のように計算されます。

運賃	(2.5%)	$0.025 * 3000 = \$75.00$
税金	(0.05 / ポンド)	$1000 * 1 * .05 = \$50.00$
保険料	(0.75%)	$0.0075 * 3000 = \$22.50$
仲介手数料	(1%)	$0.01 * 3000 = \$30.00$

追加コスト合計: \$177.50

この購入オーダー明細で 1000 ユニットの受入が行われた場合、平均コストは以下のように計算されます。



$$(\$2000 + \$3000 + \$177.50) / (500 + 1000) = \$3.451667$$

購入オーダー明細アイテムの受入が総勘定元帳の記帳用に公開されると、Sterling Selling and Fulfillment Foundation は、以下のように、在庫価値の内訳を提供します。

拡張コスト: \$3000

運賃: \$75.00

税金: \$50.00

保険料: \$22.50

仲介手数料: \$30.00

合計: \$3177.50

提供された内訳を使用して、適切な総勘定元帳の科目に記帳することができます。

**平均コストの計算 - 作業オーダーの完了:** 平均コストは、作業オーダー完了の受入時に、以下の方法で計算されます。

$(\text{手持ち在庫価値} + \text{完成品の合計価値}) / (\text{新規の手持ち数量})$

ここでは、以下とします。

- 在庫価値は再計算済み
- 完成品の合計価値は次と等しい。
  - すべてのコンポーネントの平均コストの合計 + 関連コスト要因グループに基づいて計算された追加コスト。
- コスト要因グループは、エンタープライズ・レベルまたは作業オーダーの親アイテム・レベルで定義できる。ビジネス・ユニット (エンタープライズ) 1 つに対して、定義できるコスト要因グループは 1 つのみです。

以下の例は、作業オーダー完了時の平均コストの計算を示しています。この例で参照されているコスト要因グループは、85 ページの『キット化された完成製品のコスト』に定義されています。この例では、コスト要因グループは、エンタープライズに関連付けられています。

作業オーダー完了前の現在の手持ち数量 = 500。

完了前の在庫価値 = \$5000。

1000 ユニットの完成品の作業オーダーを受入済み。

完成品は、以下のコンポーネントから構成。

- 12345 - 平均コスト = \$2.10
- 32456 - 平均コスト = \$6.20

1000 ユニットのこの完成品の作業オーダーが完了したとき、完成品の合計価値は以下のように計算されます。

コンポーネント・コスト		$(1000 * \$2.10) + (1000 * 6.20) = \$8300$
労働コスト	(20%)	$0.2 * 8300 = \$1660$
供給	(\$3.00/ユニット)	\$3000

在庫価値の増加合計 = \$12,960.00

新しい平均コストは以下のように計算されます。

$$(\$5000 + \$12960) / (500 + 1000) = \$11.973333$$

**負の在庫残高の取り扱い:** 負の手持ち残高のあるアイテムまたはノードに対して受け入れを行う場合は、在庫値および平均コストの計算は、以下のように変更されます。

1. 平均コストは、現行トランザクションの負荷コストに設定されます。負荷コストは、PO に対する単位価格に、PO に対するセラーの荷役コスト要因を加えたものです。
2. 在庫値は、on-hand inventory \* new average cost の値に設定されます。
3. Sterling Selling and Fulfillment Foundation は、標準在庫値の変更に付随して起こる 2 番目のイベントを生成します。この 2 番目のイベントは、再計算された在庫値と、次のように計算された損金処理金額との差分を公開します。

$$\text{Old Inventory Value} + \text{Change in Inventory Value} - \text{New Final Inventory Value}$$

以下に例を示します。

製品には、現在、\$3.50 の平均コストがかかっています。システム内の手持ち数量は -20 で、在庫値は -\$70 です。新規受け入れの負荷コストを \$4.00 として、このアイテムの 15 単位について受け入れが作成されると、Sterling Selling and Fulfillment Foundation は、以下のように在庫を更新します。

$$\text{Average cost} = \$4.00 - \text{loaded cost from the receipt}$$

$$\text{On hand quantity} = -5$$

$$\text{Inventory value} = -\$20.00$$

標準在庫イベントは、アイテムの在庫値に対して \$60.00 の増加を公開します。この増加には、\$4.00 \* 15 (受け入れられた単位数) の負荷コストが含まれます。

追加の在庫帳消しイベントは、(古い在庫値 + 受入による在庫値の変更 - 新しい最終在庫値) または  $(-\$70 + \$60 - (-\$20) = \$10.00)$  として計算される \$10.00 として帳消し金額を発行します。この場合、調整勘定への借方と在庫勘定への貸方を使って、会計アプリケーションへの入力を作成されます。結果が負の金額になった場合は、入力は、調整勘定への貸方と在庫勘定への借方になります。Sterling Selling and Fulfillment Foundation では、これが会計アプリケーションに対して確実にそのように表されます。

## 平均コスト計算方法: コスト要因

コスト要因は、新しいユニット・コストを生む基本コストからの追加機能またはコンポーネントで、値の変更因子を表します。コスト要因の例には、保険、運賃、材料取扱、梱包などがあります。これらのアクティビティーは、組織が派生コストを出すために追跡する必要がある、基本ポイントに対して相対的な付加価値を表します。例えば、ベンダーからの取得などです。

Sterling Selling and Fulfillment Foundation では、名目価値 (置換コストなど) と 1 つ以上のコスト要因の間の関係として、コスト要因およびユニット・コストの定義が使用可能です。

Sterling Selling and Fulfillment Foundation は、以下のポイントでコスト要因を使用し、派生コストを算出します。

- 受入処理時の在庫の積み付けコストの計算。
- 作業オーダー完了時のキット化された完成品の積み付けコストの計算
- 置換コストまたは平均コストからのユニット・コストの計算

**コスト要因の定義:** 組織は、多数のコスト要因を定義できます。各コスト要因には、次の属性があります。

名前	コスト要因の固有名
計算方法 - % または値のどちらか?	この要因がパーセンテージまたは金額のどちらで表されるか?
適用方法	計算方法が「パーセンテージ」として定義されている場合、適用方法に意味はありません。  計算方法が「金額」として定義されている場合、適用方法は以下のいずれかの値になります。 <ul style="list-style-type: none"><li>• 重量 - 指定された値は、製品のポンド当たりのドルの値を表します。基本重量は、任意の重量の測定単位 (UOM) で定義できます。</li><li>• 容積 - 指定された値は、製品の立法フィート当たりのドルの値を表します。基本容積は、任意の容積の測定単位 (UOM) で定義できます。</li><li>• 数量 - 指定された値は、製品のユニット当たりの金銭的価値を表します。</li></ul>

**コスト要因グループ:** ベンダーまたはトランザクション・タイプ (作業オーダーの完了または製品の受入など) に基づいて、異なるコスト要因の適用が必要となる場合があるため、Sterling Selling and Fulfillment Foundation では、コスト要因グループを作成することができます。コスト要因グループは、特定のシナリオの基本コストに適用されるコスト要因のセットを表します。グループ内の各コスト要因は、共通の定義への参照です。グループは、特定のシナリオにおいて、要因に特定の値を指定します。

**コスト要因グループの割り当て:** 柔軟性を最大化し、保守コストを最小化するために、コスト要因の割り当ておよび取得に階層型のアプローチが提供されています。

レベル 0: エンタープライズ・レベル

レベル 1: サプライヤー・レベル

レベル 2: アイテム・レベル - WIP 処理における物理キットの親アイテムの場合のみ

各レベルで、組織は異なるトランザクションに一連のコスト要因を割り当てることができ、各コスト要因に値を割り当てることができます。

**コスト要因の取得:** コスト要因の取得時に、アプリケーションは割り当て階層のレベル 2 で開始し、割り当て階層のレベル 0 まで戻りながら、特定のトランザクションに最も固有なコスト要因のリストを取得します。コスト要因は、単一のレベルのみから取得されます。

**コスト要因グループの例:** 『国際的なベンダーから輸入した製品のコスト要因グループ』から 85 ページの『置換コストからのユニット・コストの計算』では、特定の状況において、コスト要因グループをどのようにセットアップするかを説明しています。

#### 国際的なベンダーから輸入した製品のコスト要因グループ

このグループでは、以下のコスト・ロード要因が指定されます。

コスト要因名	計算方法	適用方法	パーセンテージ	値	コメント
運賃	パーセンテージ	-	2.5%	-	運賃は、基本コストに占めるパーセンテージとして計算されます
税金	金額	重量	-	0.050000	税金は製品の 1 ポンド当たり 0.05 で計算されます
仲介手数料	パーセンテージ	-	1%	-	仲介手数料は、製品コストの 1% になります
保険料	パーセンテージ	-	0.75%	-	保険料は、基本コストに占めるパーセンテージとして計算されます

#### 国内ベンダーからソーシングした製品のコスト要因グループ

このグループでは、以下のコスト・ロード要因が指定されます。

コスト要因名	計算方法	適用方法	パーセンテージ	値	コメント
運賃	パーセンテージ	-	2%	-	運賃は、基本コストに占めるパーセンテージとして計算されます
保険料	パーセンテージ	-	0.0050	-	保険料は、基本コストに占めるパーセンテージとして計算されます

## 運賃を単価に含める国内ベンダーからソーシングした製品のコスト要因グループ

このグループでは、以下のコスト・ロード要因が指定されます。

コスト要因名	計算方法	適用方法	パーセンテージ	値	コメント
保険料	パーセンテージ	-	0.0050	-	保険料は、基本コストに占めるパーセンテージとして計算されます

## キット化された完成製品のコスト

このグループでは、以下のコスト要因が指定されます。

コスト要因名	計算方法	適用方法	パーセンテージ	値	コメント
労働	パーセンテージ	-	20%		基本コストの 20% が労働コストとして追加されます
供給	金額	数量 (Quantity)	-	3.00	\$3.00 が、すべての完成したキット化されたユニットに供給コストとして追加されます

## 置換コストからのユニット・コストの計算

このグループでは、以下のコスト・ロード要因が指定されます。

コスト・ロード要因名	計算方法	適用方法	パーセンテージ	値	コメント
取り扱い	パーセンテージ	-	5%		基本コストの 5% が派生コストとして追加されます

## 平均コスト計算方法: 在庫製品の在庫コスト計算

以下のイベントは、在庫製品の取り扱い時に、コスト要素または総勘定元帳の記帳に影響を与えます。

1. 購入オーダーに関連したイベント (86 ページの『標準購入オーダーの処理』を参照)
  - a. 購入オーダー作成
  - b. 購入オーダー更新
  - c. 購入オーダー受入
  - d. 会計システムで実行されるインボイス・マッチング機能
2. 販売オーダーに関連したイベント (93 ページの『標準販売オーダーの処理』を参照)
  - a. 販売オーダー作成
  - b. 出荷の確認
  - c. インボイスの発行
3. 在庫の調整 (98 ページの『返品処理と在庫調整』を参照)
4. 商品の返品 (98 ページの『返品処理と在庫調整』を参照)

5. 作業オーダーの完了 (102 ページの『仕掛品の処理』を参照)

**標準購入オーダーの処理:** Sterling Selling and Fulfillment Foundation では、標準購入オーダーは次のように処理されます。

### PO の作成と更新

1. Sterling Selling and Fulfillment Foundation では、PO 明細に、Sterling Selling and Fulfillment Foundation で管理されるベンダー価格リストからの価格が記載されます。

ベンダー価格が Sterling Selling and Fulfillment Foundation で管理されていない場合は、外部のベンダー価格設定アプリケーションから価格が取り出されることがあります。

2. Sterling Selling and Fulfillment Foundation では、ベンダーの単価を PO 明細にロックする機能が提供されます。

ベンダーの価格リストが単価計算に使用される場合、Sterling Selling and Fulfillment Foundation は、購入オーダー明細の数量が変更されるたびに単価を再計算します。これにより、適切な数量の内訳が単価計算に反映されるようになります。購入オーダー上で PO 単価がオーバーライドされた場合、Sterling Selling and Fulfillment Foundation はこの計算を省略し、明細の拡張コストのみを再計算します。

3. 購入オーダーの作成と更新が正常に行われると、Sterling Selling and Fulfillment Foundation が標準イベントを発生します。これを使って、購入オーダーの予想コストに関連する情報で会計アプリケーションを更新できます。

### PO の受入

4. Sterling Selling and Fulfillment Foundation は、製品の受入時に、PO に対して在庫値を増やします。この計算は在庫値を決定するために行います。
5. Sterling Selling and Fulfillment Foundation は、製品を PO に対して受け入れるときに、ノードの製品の平均コストを再計算します。この計算は平均コストを決定するために行います。
6. Sterling Selling and Fulfillment Foundation は各受入明細のデータを公開します。このデータを使用して、会計アプリケーションにおいて総勘定元帳レベルの転記が行われます。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。
7. Sterling Selling and Fulfillment Foundation では、複数の受入を各 PO 明細に対して記録できるため、PO 明細が複数の総勘定元帳インターフェースのレコードに含まれる (受入明細ごとに 1 つ) 可能性があります。そのため、受入が PO に対して記録されると、購入オーダー明細ごとに 1 つのイベントが発行されます。同じ明細が複数の受入として記録されると、複数のイベントが発生します。
8. 負の既存手持ち残高に対して受入が作成されると、Sterling Selling and Fulfillment Foundation では、82 ページの『負の在庫残高の取り扱い』で説明されている特別な処理方法が使用されます。

9. Sterling Selling and Fulfillment Foundation は、標準購入オーダーと連鎖購入オーダーを異なる方法で処理します。

例えば、サプライヤーによるインボイスの送信を契機にフルフィルメントの通知が行われるような場合には、連鎖 PO の A/P インターフェースは抑制しなければならないことがあります。

#### インボイスの作成

10. 会計アプリケーションでインボイスの照合機能は次のように実行されます。
- インボイスの照合機能により検出された不一致は、在庫値勘定のほかに、差異勘定に転記されます。
  - 買掛金勘定アプリケーションが PO 予想コストと支払可能インボイスの実際のコストの差異を生成した場合、この差異を Sterling Selling and Fulfillment Foundation に再び渡して平均コストに反映させる必要があります。Sterling Selling and Fulfillment Foundation では、差異金額 (に加え PO 参照および請求済み数量) を受け入れて、この額により在庫値の調整を試行する API (updateInventoryCost) が提供されます。手持ち数量の合計が、(後続する出荷または出庫のため) 請求済みの数量を下回っている場合は、差異合計が比例計算され、残りの手持ち在庫に適用されます。適用されない金額は会計アプリケーションに再び渡され、適切な差異勘定に保存されます。

#### 標準購入オーダー・シナリオ:

注: 以下のシナリオでは説明されていませんが、コスト要因の取り出しはコスト計算の基本要素です。

#### 標準購入オーダー - シナリオ 1 - 89 ページの図 15 を参照

購入オーダーは、アイテム ABC に対して作成されます。購入オーダー作成メッセージは、この特定のトランザクションに対して、ABC の予定サプライヤー (S1) を特定します。数量 50 個が必要です。エントリーで、S1 から ABC を購入する置換コストが取り出され、明細価格として購入オーダーにスタンプされます。数量 50 個の ABC が購入オーダー明細 (または ASN) に対して受け入れられます。受入それぞれに対して、受入ロケーションの在庫の値が再計算され、受入数量および購入オーダー明細の予定コストに基づく新しい平均コストが決まります。最後に、受入に関する情報 (受入数量、アイテム、予定価格) が会計アプリケーションに送信されます。

- Sterling Selling and Fulfillment Foundation は、置換コスト取り出しを実行します。
  - 使用される条件: オーダー日、サプライヤー、アイテム、オーダー数量
  - 外部プログラムによって、標準の Sterling Selling and Fulfillment Foundation 取り出しをバイパスできます。
  - 外部プログラムが実装されていない場合、Sterling Selling and Fulfillment Foundation は、指定された価格リストから価格を取り出します。
  - Sterling Selling and Fulfillment Foundation は、予定コストを含む購入オーダー明細を更新します (予定コストは、外部プログラムの起動または Sterling Selling and Fulfillment Foundation からの取り出しによって取得されます)。
  - 結果として得られる購入オーダー価格 = \$2.00 (1 個あたり)

2. Sterling Selling and Fulfillment Foundation の受入明細トランザクションそれぞれが手順 3 から 5 を実行します。
3. Sterling Selling and Fulfillment Foundation は、在庫値を再計算します。
  - a. 現在の在庫値 = \$250
  - b. 新しい受入の値 =  $50 \times 2.00 = \$100$
  - c. 新しい在庫値 = \$350
4. Sterling Selling and Fulfillment Foundation は、平均コストを再計算します。
  - a. 手持ち在庫 (事前受入) = 120
  - b. 手持ち在庫 (事後受入) = 170
  - c. 新しい在庫値 = \$350
  - d. 新しい平均コスト =  $350/170 = \$2.058824$
5. Sterling Selling and Fulfillment Foundation は在庫値の変更 (会計アプリケーションへの統合に使用) を発行します。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。



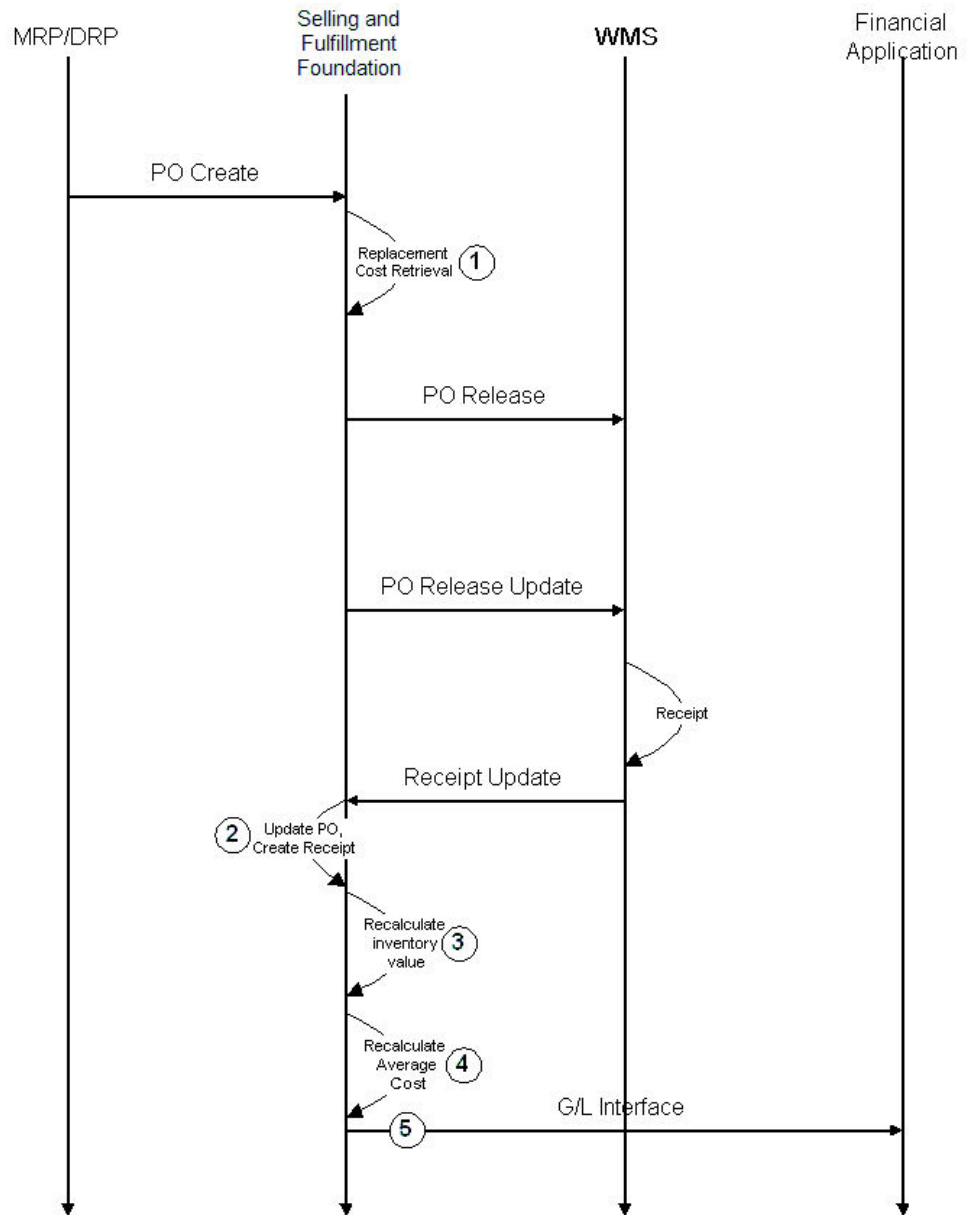


図 15. 購入オーダー・シナリオ 1

標準購入オーダー - シナリオ 2 - 91 ページの図 16 を参照

購入オーダーは、アイテム ABC に対して作成されます。購入オーダー作成メッセージは、この特定のトランザクションに対して、ABC の予定サプライヤー (S1) を特定します。数量 50 個が必要です。エントリーで、S1 から ABC を購入する置換コストが取り出され、明細価格として購入オーダーにスタンプされます。S1 から購入オーダー肯定応答を受信すると、単位あたり価格が増加したことが注記されます。この新しい価格は、元の購入オーダーに購入オーダー変更として入力されます。数量 50 個の ABC が購入オーダー明細/出荷に対して受け入れられます。受入それぞれに対して、受入ロケーションの在庫の値が再計算され、受入数量および購

入オーダー明細の予定コストに基づく新しい平均コストが決まります。最後に、受入に関する情報 (受入数量、アイテム、予定価格) が会計アプリケーションに送信されます。

1. Sterling Selling and Fulfillment Foundation は、置換コスト取り出しを実行します。
  - a. 使用される条件: オーダー日、サプライヤー、アイテム、オーダー数量
  - b. 外部プログラムによって、標準の Sterling Selling and Fulfillment Foundation 取り出しをバイパスできます。
  - c. 外部プログラムが実装されていない場合、Sterling Selling and Fulfillment Foundation は、指定された価格リストから価格を取り出します。
  - d. Sterling Selling and Fulfillment Foundation は、予定コストを含む購入オーダー明細を更新します (予定コストは、外部プログラムの起動または Sterling Selling and Fulfillment Foundation からの取り出しによって取得されます)。
  - e. 結果として得られる購入オーダー価格 = \$2.00 (1 個あたり)
2. Sterling Selling and Fulfillment Foundation は、サプライヤーからの購入オーダー価格を更新します。
  - a. インバウンド・メッセージからの価格は、購入オーダー価格を変更する必要があります。
  - b. 新しい価格 = \$2.10
3. Sterling Selling and Fulfillment Foundation の受入明細トランザクションそれぞれが手順 2 から 6 を実行します。
4. Sterling Selling and Fulfillment Foundation は、在庫値を再計算します。
  - a. 現在の在庫値 = \$250
  - b. 新しい受入の値 =  $50 \times 2.10 = \$105$
  - c. 新しい在庫値 = \$355
5. Sterling Selling and Fulfillment Foundation は、平均コストを再計算します。
  - a. 手持ち在庫 (事前受入) = 120
  - b. 手持ち在庫 (事後受入) = 170
  - c. 新しい在庫値 = \$355
  - d. 新しい平均コスト =  $355 / 170 = \$2.088235$
6. Sterling Selling and Fulfillment Foundation は在庫値の変更 (会計アプリケーションへの統合に使用) を発行します。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

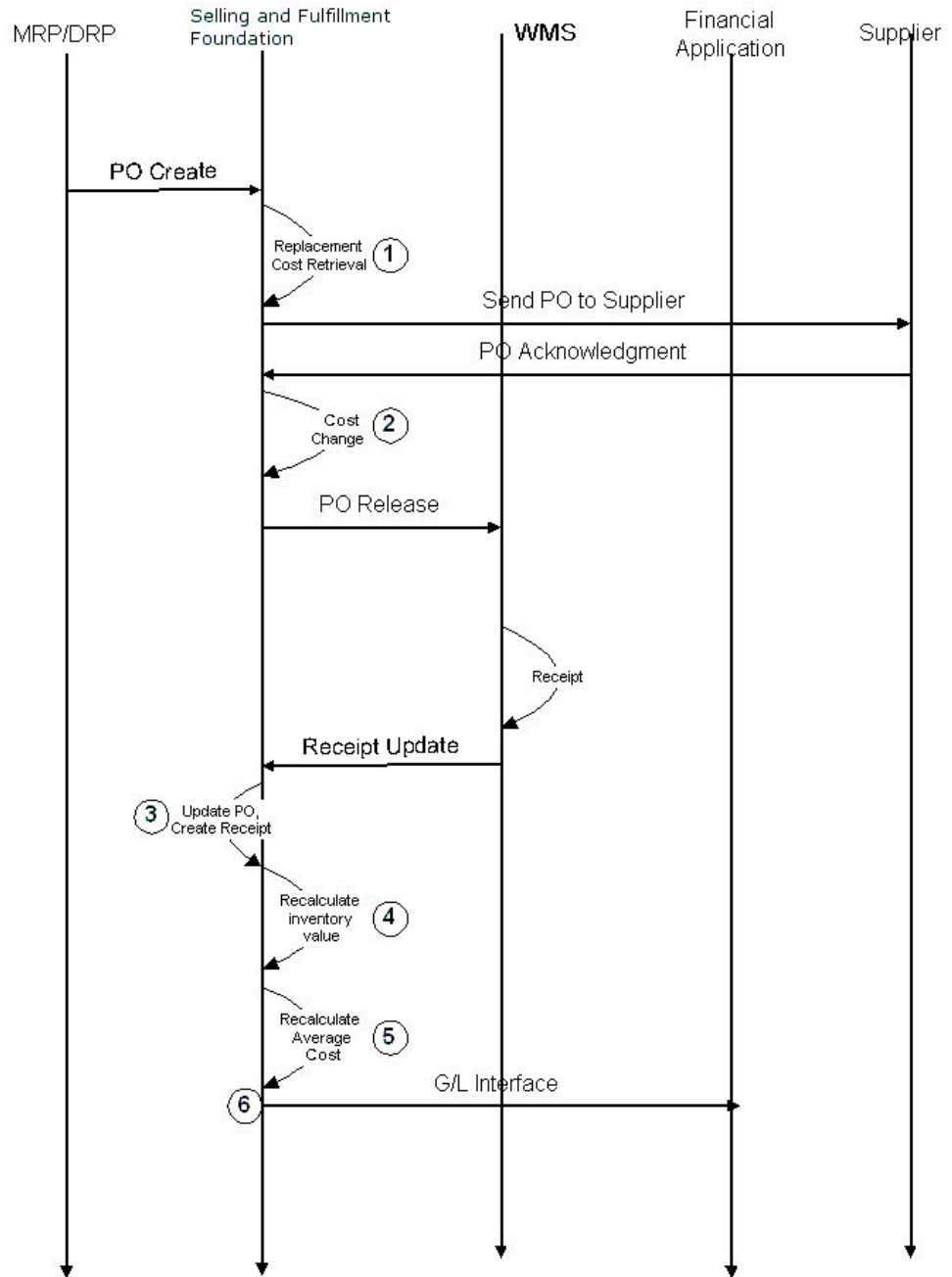


図 16. 購入オーダー・シナリオ 2

**標準購入オーダー - シナリオ 3 - 89 ページの図 15 を参照**

購入オーダーは、アイテム ABC に対して作成されます。購入オーダー作成メッセージは、この特定のトランザクションに対して、ABC の予定サプライヤー (S1) を特定します。数量 50 個が必要です。エントリーで、S1 から ABC を購入する置換コストが取り出され、明細価格として購入オーダーにスタンプされます。数量 40 個の ABC が購入オーダー/ASN に対して受け入れられます。受入それぞれに対して、受入ロケーションの在庫の値が再計算され、受入数量および購入オーダー明細の予定コストに基づく新しい平均コストが決まります。最後に、受入に関する情報 (受入数量、アイテム、予定価格) が会計アプリケーションに送信されます。

1. Sterling Selling and Fulfillment Foundation は、置換コスト取り出しを実行します。
  - a. 使用される条件: オーダー日、サプライヤー、アイテム、オーダー数量
  - b. 外部プログラムによって、標準の Sterling Selling and Fulfillment Foundation 取り出しをバイパスできます。
  - c. 外部プログラムが実装されていない場合、Sterling Selling and Fulfillment Foundation は、指定された価格リストから価格を取り出します。
  - d. Sterling Selling and Fulfillment Foundation は、予定コストを含む購入オーダー明細を更新します (予定コストは、外部プログラムの起動または Sterling Selling and Fulfillment Foundation からの取り出しによって取得されます)。
  - e. 結果として得られる購入オーダー価格 = \$2.00 (1 個あたり)
2. Sterling Selling and Fulfillment Foundation の受入明細トランザクションそれぞれが手順 3 から 5 を実行します。
3. Sterling Selling and Fulfillment Foundation は、在庫値を再計算します。
  - a. 現在の在庫値 = \$250
  - b. 新しい受入の値 =  $40 \times 2.00 = \$80$
  - c. 新しい在庫値 = \$330
4. Sterling Selling and Fulfillment Foundation は、平均コストを再計算します。
  - a. 手持ち在庫 (事前受入) = 120
  - b. 手持ち在庫 (事後受入) = 160
  - c. 新しい在庫値 = \$330
  - d. 新しい平均コスト =  $330 / 160 = \$2.0625$
5. Sterling Selling and Fulfillment Foundation は在庫値の変更 (会計アプリケーションへの統合に使用) を発行します。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

注: 非常に大きい数量を受け入れる追加の受入トランザクションについては、上記のプロセスに従ってください。倉庫管理システムで実行される既存の受入の調整は、会計アプリケーションで手動で入力する必要があります。

#### 標準購入オーダー - シナリオ 4

アイテム JJJ の現在の平均コストは \$3.50 です。システムの現在の手持ち数量は -20 で在庫値は -\$70 です。受入は、明細価格 \$4.00 の購入オーダーに対する 15 単位の JJJ に対して行われます。

1. Sterling Selling and Fulfillment Foundation は、平均コストを購入オーダー明細価格に設定します (平均コスト \$3.50 から \$4.00)。
2. 新しい手持ち数量 =  $-5 (-20+15)$
3. Sterling Selling and Fulfillment Foundation は、新しい在庫値を再計算します。
  - a. 手持ち数量が負の値で始まったため、Sterling Selling and Fulfillment Foundation は、変更された計算を使用します。
  - b. 新しい在庫値 = 手持ちあり \* 新しい平均コスト =  $-20$

4. Sterling Selling and Fulfillment Foundation は、在庫値の帳消しを決定します。
  - a. 標準在庫イベントは、アイテムの在庫値に対する \$60.00 (15 \* \$4) の増加を発行します。
  - b. 追加の在庫帳消しイベントは、(古い在庫値 + 受入による在庫値の変更 - 新しい最終在庫値) または  $(-\$70 + \$60 - (-\$20) = \$10.00)$  として計算される \$10.00 として帳消し金額を発行します。

#### 標準購入オーダー - シナリオ 5

アイテム JJJ の現在の平均コストは \$3.50 です。システムの現在の手持ち数量は -20 で在庫値は -\$70 です。受入は、明細価格 \$4.00 の購入オーダーに対する 25 単位の JJJ に対して行われます。

1. Sterling Selling and Fulfillment Foundation は、平均コストを購入オーダー明細価格に設定します (平均コスト \$3.50 から \$4.00)。
2. 新しい手持ち数量 = 5 (-20+25)
3. Sterling Selling and Fulfillment Foundation は、新しい在庫値を再計算します。
  - a. 手持ち数量が負の値で始まったため、Sterling Selling and Fulfillment Foundation は、変更された計算を使用します。
  - b. 新しい在庫値 = 手持ちあり \* 新しい平均コスト = \$20
4. Sterling Selling and Fulfillment Foundation は、在庫値の帳消しを決定します。
  - a. 標準在庫イベントは、アイテムの在庫値に対する \$100.00 (25 \* \$4) の増加を発行します。
  - b. 追加の在庫帳消しイベントは、(古い在庫値 + 受入による在庫の変更 - 新しい最終在庫値) または  $(-\$70 + \$100 - (\$20) = \$10.00)$  として計算される \$10.00 として帳消し金額を発行します。

**標準販売オーダーの処理:** Sterling Selling and Fulfillment Foundation では、標準販売オーダーは次のように処理されます。

#### オーダーの作成

販売オーダー明細の作成時とドラフト販売オーダーの確認時に、 Sterling Selling and Fulfillment Foundation は次のロジックを使って各明細の単価を取り出します。

1. オーダー明細上で単価がオーバーライドされた場合、Sterling Selling and Fulfillment Foundation はオーバーライド・コストを単価として使用します。
2. 置換コストが単価計算の基準として使用される場合、以下のようになります。
  - 製品マスター・テーブルにおいて、アイテム・レベルで単価が手動入力された場合は、オーダー明細には手動入力された単価が使用されます。
  - 手動入力が行われていない場合は、オーダー明細にはアイテム・レベルで保存されている計算済みの単価が使われます。単価が保存されていない場合は、販売オーダー明細に \$0.00 というコストが反映され、ORDER\_CREATE.ON\_ZERO\_UNIT\_COST イベントがトリガーされます。
  - 論理キットの単価は、コンポーネント・レベルで保存および管理されます。
3. 平均コストが単価計算の基準として使用される場合、このコスト基準はこのリリースではサポートされません。

- オーダー明細上で単価がオーバーライドされた場合、オーダー明細には指定のコストが単価として使用されます。
- 単価の計算では、オーダー明細に出荷ノードが指定されている必要があります。出荷ノードが指定されていない場合は、単価は保存されず、\$0.00 と表示されます。
- オーダー明細に出荷ノードが指定される都度、Sterling Selling and Fulfillment Foundation は、アイテム・ノード・レベルの単価を保存します。オーダー明細の出荷ノードが変更されると、オーダー明細の単価は、変更後のノードから抽出されます。

### 出荷の確認

1. 製品が出荷されると、Sterling Selling and Fulfillment Foundation は、フルフィルメント・ロケーションの在庫値を再計算します。77 ページの『在庫コスト計算』の在庫値の計算方法を参照してください。
2. Sterling Selling and Fulfillment Foundation で出荷が確認されると、Sterling Selling and Fulfillment Foundation は情報を公開し、会計アプリケーションで使用できるようになります。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。
3. 出荷確認イベントを使用して、総勘定元帳の売上原価、在庫、および各種の差異勘定のエントリーが更新されます。このイベントは、オーダー明細ごとに個別に発行され、通常、インボイス発行イベントにより売上勘定および売掛金勘定の転記処理が行われます。

### インボイスの作成

Sterling Selling and Fulfillment Foundation のインボイス発行インターフェースにより、総勘定元帳の売上勘定と売掛金勘定の各エントリーに転記されます。

### 標準販売オーダーに関するシナリオ

注: 以下のシナリオでは説明されていませんが、コスト要因の取り出しはコスト計算の基本要素です。

#### 標準販売オーダー - シナリオ 1 - 96 ページの図 17 参照

顧客が製品 ABC を 2 ピースの数量でオーダー発注しました。オーダーの価格は、顧客の価格設定ルールに従って決定されます。オーダー作成時に出荷ノードが決定されるため、ABC の予想単価もオーダー明細に記録されます。フルフィルメント・ロケーションにより ABC の出荷が報告され、この時点で、在庫値の変更イベントが発行されて各コスト・バケットの詳細が示されます。インボイス作成トランザクションの実行時に、出荷された数量と販売価格を使って、適切な会計アプリケーションのインターフェースのエントリーが生成されます。

1. オーダーの作成
  - a. セラーによって定義された価格リストに従い、オーダーの価格が設定されます。
  - b. 単価は、ABC の主要サプライヤーに基づき、オーダー明細に記録されます。

## 2. 出荷の確認

出荷またはオーダー明細に対して、製品 ABC の平均コストとフルフィルメント・ロケーションが記録されます。

## 3. Sterling Selling and Fulfillment Foundation は在庫値を再計算します。

- a. 現在の在庫値 = \$250
- b. 現在の平均コスト = \$2.05725
- c. 出荷値 =  $2 \times 2.05725 = \$4.1145$
- d. 新規の在庫値 = \$245.8855
- e. Sterling Selling and Fulfillment Foundation は、在庫の変更情報を公開します (会計アプリケーションへの統合に使用されます)。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

## 4. インボイスの作成

- a. 販売オーダーのインボイスは、コスト関連のデータを発行しません。
- b. 出荷確認時 (手順 2) に記録されるすべてのコスト関連のデータが渡されます。
- c. インボイスでは、記載される出荷を出荷番号により追跡し、必要に応じてコスト・データを取得できます。インボイスには、トランザクションに関する価格関連情報がすべて記載され、対応する売掛金のエントリーを作成するのに使用されます。

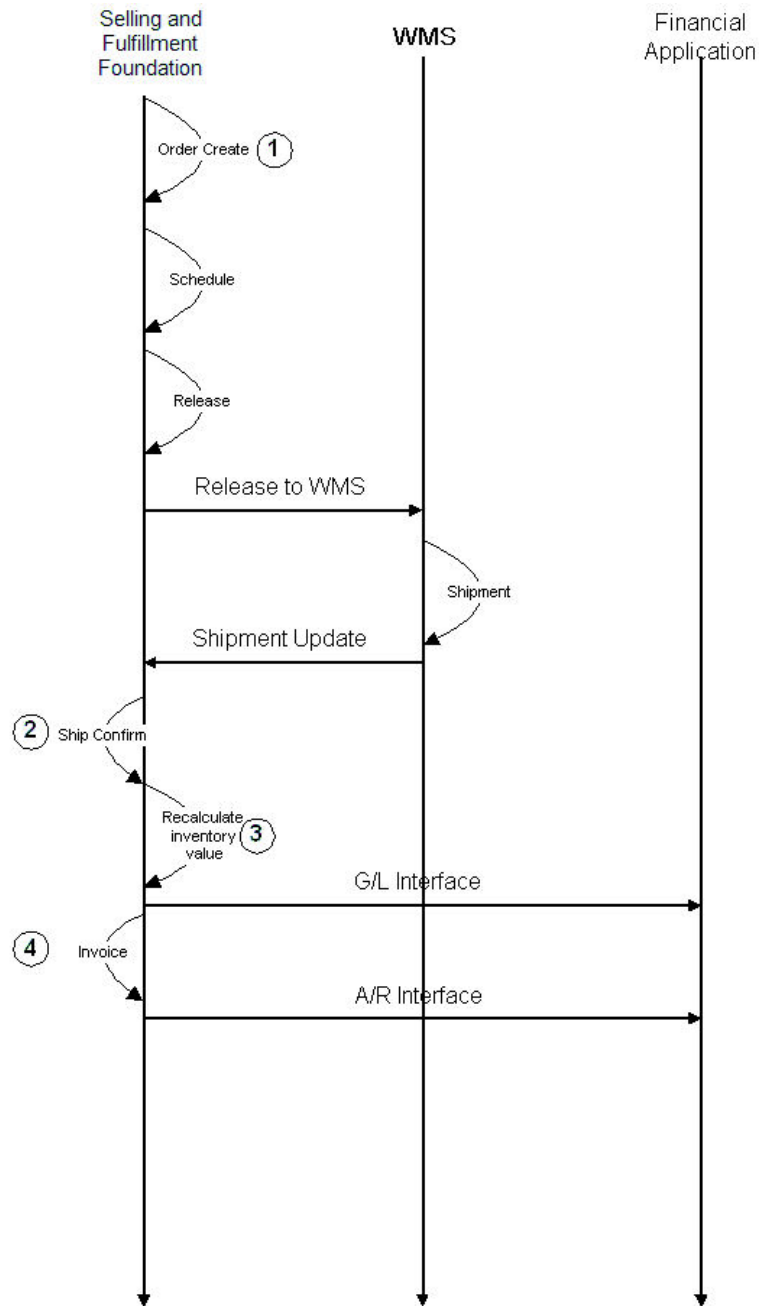


図 17. 販売オーダーのシナリオ 1

**標準販売オーダー - シナリオ 2 - 98 ページの図 18 参照**

顧客が製品 ABC を 10 ピースの数量でオーダー発注しました。オーダーの価格は、顧客の価格設定ルールに従って決定されます。ABC の予想単価は、オーダー明細にも記録されます。オーダーがスケジュールされ、フルフィルメントが Node1 (数量 = 7) と Node2 (数量 = 3) から行われると決定されます。オーダーがリリースされます。フルフィルメント・ロケーションにより ABC の出荷が報告され、この時点で、在庫値の変更イベントが発行されて各コスト・バケットの詳細が示されます。インボイス作成トランザクションを (出荷に基づくインボイス作成を使用し



て) 実行する際に、出荷された数量と販売価格を使って、適切な会計アプリケーションのインターフェースのエントリーが生成されます。

1. オーダーの作成

- a. オーダーの価格は、セラーによって定義された価格リストに従って決定されます。
- b. 単価は、ABC の主要サプライヤーに基づき、オーダー明細に記録されます。

2. スケジュール

オーダー明細は 2 つのスケジュール・エントリーに分割され、Node1 と Node2 という 2 つのフルフィルメント・ソースが示されます。

3. リリース

Sterling Selling and Fulfillment Foundation は各ノードを対象にオーダー・リリースを作成し、オーダー明細に対して実行します。

4. 出荷の確認 - Node1

Sterling Selling and Fulfillment Foundation は、出荷またはオーダー明細に対して、製品 ABC/フルフィルメント・ロケーションの平均コストを記録します。

5. Sterling Selling and Fulfillment Foundation は Node1 の在庫値を再計算します。

6. Sterling Selling and Fulfillment Foundation は、在庫の変更情報を公開します (会計アプリケーションへの統合に使用されます)。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

7. Sterling Selling and Fulfillment Foundation は、Node2 を対象に 4 から 6 の手順を繰り返します。

8. インボイスの作成

- a. 販売オーダーのインボイスは、コスト関連のデータを発行しません。
- b. 出荷確認時 (手順 4) に記録されるすべてのコスト関連のデータが渡されます。
- c. インボイスでは、記載される出荷を出荷番号により追跡し、必要に応じてコスト・データを取得できます。インボイスには、トランザクションに関する価格関連情報がすべて記載され、対応する売掛金のエントリーを作成するのに使用されます。

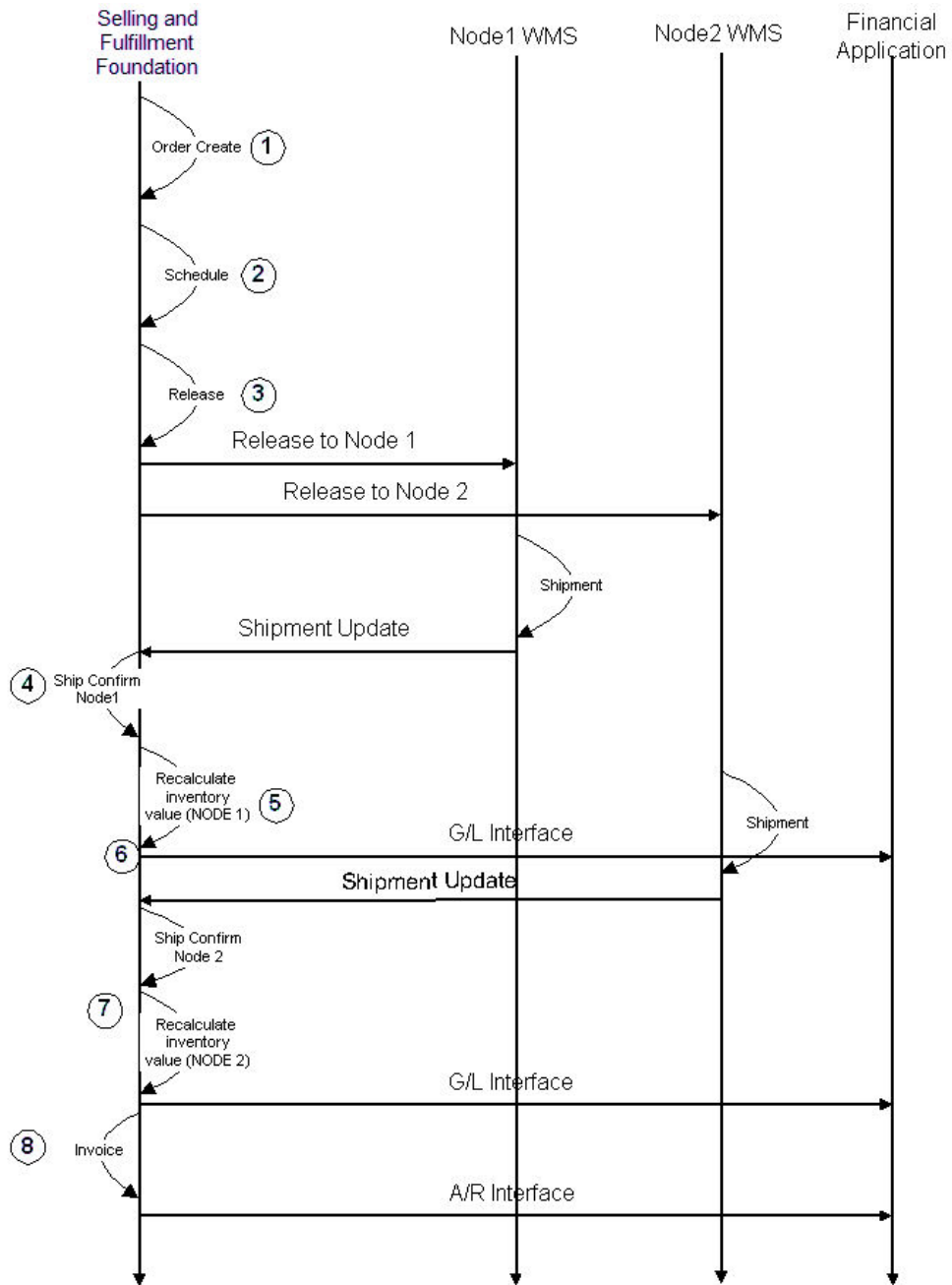


図 18. 販売オーダーのシナリオ 2

**返品処理と在庫調整:** Sterling Selling and Fulfillment Foundation は返品オーダーを処理し、次のように在庫調整を行います。

**一般**

1. 返品処理は、Sterling Selling and Fulfillment Foundation 返品物流モジュールの使用を前提としていません。在庫調整は、「返品」タイプとしてのみフラグを設定できます。
2. 返品処理と在庫調整により、平均コストの再計算が強制的に行われることはありません。

注: updateInventoryCost API を使用して在庫値の再計算を行うことができます。updateInventoryCost API について詳しくは、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

3. 返品と在庫調整の両方で、入庫または出庫された数量を基に、在庫値の再計算が強制的に行われます。
  - 返品受入は手持ちの供給タイプに影響します。供給タイプは、組織ごとに構成可能です。これには、数量が手持ち在庫計算に含まれるかどうかを（このレベルで）制御する機能が含まれます。
  - 倉庫管理システムに関連するすべての更新は、手持ちの供給を対象とするものと想定されます。
4. 返品（受入）処理または在庫調整（正または負の数値）を基に在庫値を調整する場合、Sterling Selling and Fulfillment Foundation により変更が公開され、総勘定元帳の会計アプリケーションで使用できるようになります。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド*」を参照してください。

#### 正の在庫調整

5. 製品の平均コストが不明な場合は、\$0.00 として在庫に入力されるため、手動のコスト調整が必要となります。

#### 負の在庫調整

6. 手持ち数量が負の数値となった場合は、在庫値にも負の数値が反映されます。このトランザクションにより、平均コストが変更されることはありません。
7. 在庫値がゼロから調整され、製品の平均コストが不明な場合は、在庫値は \$0.00 のままになります。

#### 返品

8. 製品の平均コストが不明な場合は、\$0.00 として在庫に入力されるため、手動のコスト調整が必要となります。

#### 返品と在庫調整のシナリオ

返品シナリオ 1 と 2 では、Sterling Selling and Fulfillment Foundation における返品処理と在庫調整の方法が説明されています。

#### 返品 - シナリオ 1 - 100 ページの図 19 参照

Sterling Selling and Fulfillment Foundation では、既存の販売オーダーから返品が作成されます。顧客が製品を指定の返品ノードに発送します。このシナリオでは、製品はその出荷元と同じ施設に返送されます。返品が受け入れられ、在庫（手持ち在庫）に移動します。返品ノードにおけるこの製品の現在の平均コストを使って、在庫値が再計算されます。在庫値の変更が総勘定元帳の会計アプリケーションに公開されます。

#### 返品 - シナリオ 2 - 100 ページの図 19 参照

Sterling Selling and Fulfillment Foundation では、既存の販売オーダーから返品が作成されます。顧客が製品を指定の返品ノードに発送します。このシナリオでは、製

品はその出荷元と異なる施設に返送されます。返品施設で返品が受け入れられ、在庫（手持ち在庫）に移動します。返品ノードにおけるこの製品の現在の平均コストを使って、在庫値が再計算されます。在庫値の変更が会計アプリケーションに公開されます。

返品シナリオ 1 と 2 の両方で、一般的な情報フローは次のようになります。

1. Sterling Selling and Fulfillment Foundation は在庫値を再計算します。
  - a. 現在の平均コスト = \$4.75
  - b. 現在の在庫値 = \$2500
  - c. 新規の在庫値 =  $\$2500 + (10 \times \$4.75) = \$2547.50$
2. Sterling Selling and Fulfillment Foundation は、在庫値の変更に関連する情報を公開します（会計アプリケーションへの取り込みに使用されます）。在庫トランザクションおよび発行される具体的なデータについては、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

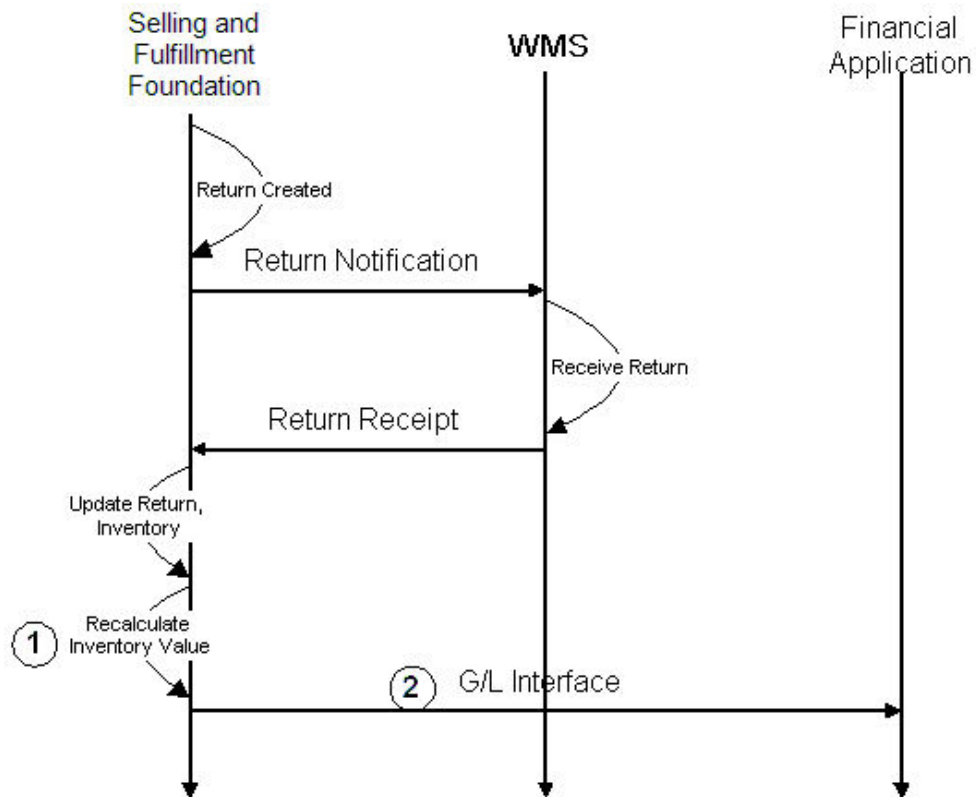


図 19. 返品シナリオ 1 および 2

#### 在庫調整 - シナリオ 1 - 102 ページの図 20 参照

循環棚卸 (cycle count) の後、Node1 の製品 ABC の在庫を 4 ユニット減らす必要があります。これは倉庫管理システムで報告され、Sterling Selling and Fulfillment Foundation にインターフェースされます。Sterling Warehouse Management System 内での調整の完了時には、Sterling Selling and Fulfillment Foundation により在庫値

が再計算されて Node1 の ABC の平均コストで 4 ユニットの ABC の減算が反映されます。在庫値の変更が会計アプリケーションに公開されます。

1. Sterling Selling and Fulfillment Foundation は在庫値を再計算します。
  - a. 現在の平均コスト = \$11.20
  - b. 現在の在庫値 = \$12,500
  - c. 新規の在庫値 =  $\$12,500 - (4 \times \$11.20) = \$12,455.20$
2. Sterling Selling and Fulfillment Foundation は、在庫値の変更に関連する情報を公開します (会計アプリケーションへの取り込みに使用されます)。在庫トランザクションおよび発行される具体的なデータについては、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

#### 在庫調整 - シナリオ 2 - 102 ページの図 20 参照

循環棚卸 (cycle count) の後、Node1 の製品 DEF の在庫を 13 ユニット増やす必要があります。これは倉庫管理システムで報告され、Sterling Selling and Fulfillment Foundation にインターフェースされます。Sterling Warehouse Management System 内での調整の完了時には、Sterling Selling and Fulfillment Foundation により在庫値が再計算されて Node1 の DEF の平均コストで 13 ユニットの DEF の加算が反映されます。在庫値の変更が会計アプリケーションに公開されます。

1. Sterling Selling and Fulfillment Foundation は在庫値を再計算します。
  - a. 現在の平均コスト = \$20.00
  - b. 現在の在庫値 = \$17,800
  - c. 新規の在庫値 =  $\$17,800 + (13 \times \$20.00) = \$18,060$
2. Sterling Selling and Fulfillment Foundation は、在庫値の変更に関連する情報を公開します (会計アプリケーションへの取り込みに使用されます)。在庫トランザクションおよび発行される具体的なデータについては、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

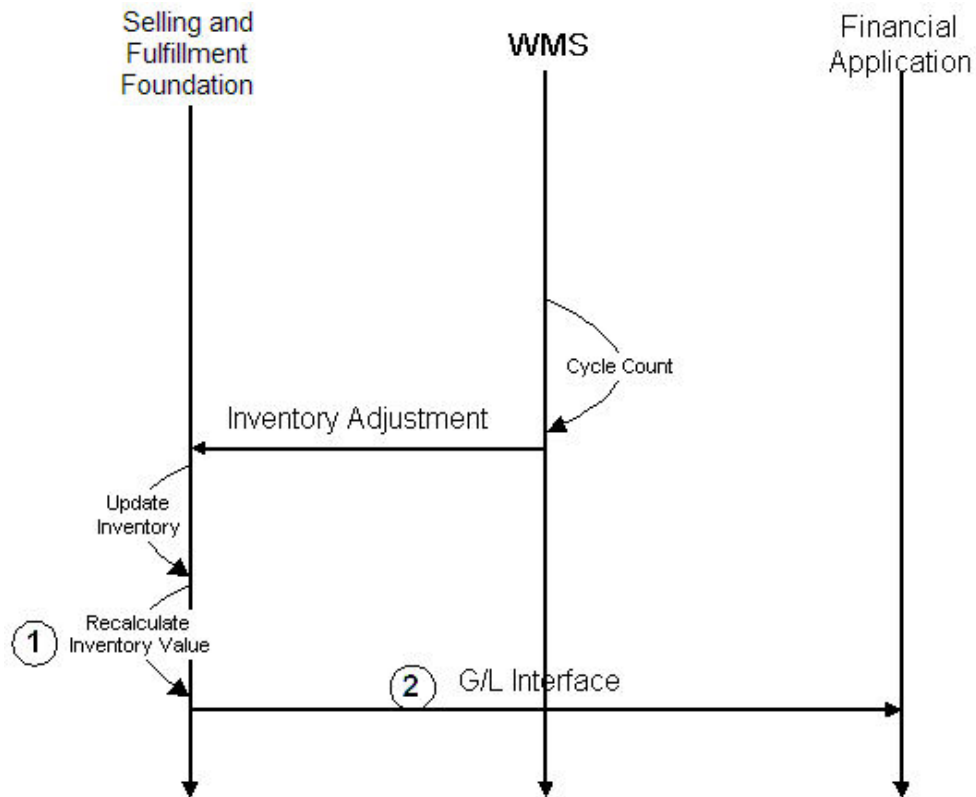


図 20. 在庫調整 - シナリオ 1 および 2

**仕掛品の処理:** 仕掛品の処理では、倉庫内のアSEMBル・キットの処理について説明します。この処理には、親部品表に応じてコンポーネント製品（個別販売可能）を消費し、多量の親アイテムを生産することが含まれます。親アイテムの生産およびその「受入」の在庫への報告によって、このタスクは完了します。

Sterling Selling and Fulfillment Foundation は、以下のように仕掛品を処理します。

1. 倉庫管理システム内のキット処理の場合、親アイテムのすべての「生産」アクティビティは、正の在庫調整として報告されます。コンポーネント・アイテムのすべての「消費」アクティビティは、負の在庫調整として報告されます。
2. コンポーネント・アイテムの場合、消費の報告に対して、Sterling Selling and Fulfillment Foundation は、生産ロケーションで在庫値を調整します。
3. コンポーネント・アイテムの場合、消費の報告に対して、Sterling Selling and Fulfillment Foundation は、会計アプリケーションに対して在庫値の負の変更を発行します。在庫トランザクションおよび発行される具体的なデータについては詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。
4. 親アイテムの場合、生産の報告に対して、Sterling Selling and Fulfillment Foundation は、77 ページの『在庫コスト計算』の説明に従って、在庫値の増加を計算します。Sterling Selling and Fulfillment Foundation は、生産ロケーションのアイテムの平均コストも再計算します。

5. 0 未満の既存の手持ち残高に対して受入が行われた場合、Sterling Selling and Fulfillment Foundation は、82 ページの『負の在庫残高の取り扱い』の説明に従って、受入を処理します。
6. 親アイテムの場合、生産の報告に対して、Sterling Selling and Fulfillment Foundation は、会計アプリケーションに対して在庫値の変更を発行します。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド」を参照してください。

### 仕掛品のシナリオ

『仕掛品のシナリオ』は、Sterling Selling and Fulfillment Foundation がどのように仕掛品オーダーを処理するかを示しています。

### 仕掛品のシナリオ

倉庫管理システムは、20 ユニットのキット・アイテム A12 の生産を Node1 で報告します。また、20 ユニットの A12 になるコンポーネント在庫 (20 ユニットの C1、40 ユニットの C2 および 60 ユニットの C3) の消費も報告します。

A12 (1 ユニット) の標準構成は、1 ユニットの C1、2 ユニットの C2、および 3 ユニットの C3 です。以下のコスト要因グループがこのエンタープライズの作業オーダーの生産と関連付けられているとします。

コスト要因名	計算方法	適用方法	パーセンテージ	値	コメント
労働	パーセンテージ	-	20%		基本コストの 20% が労働コストとして追加されます
供給	金額	数量 (Quantity)	-	3.00	\$3.00 が完成したキット・ユニットそれぞれに供給コストとして追加されます

1. Sterling Selling and Fulfillment Foundation 内のコンポーネント在庫の減少に関して、各コンポーネントの在庫値は、以下のように再計算されます。
  - a. C1 については、 $20 * C1$  の平均コスト  $(\$1.50) = \$30$
  - b. C2 については、 $40 * C2$  の平均コスト  $(\$0.50) = \$20$
  - c. C3 については、 $60 * C3$  の平均コスト  $(\$1.75) = \$105$
2. Sterling Selling and Fulfillment Foundation は、会計アプリケーションに対して別のイベントで各在庫値の変更を発行します。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド」を参照してください。
3. 20 ユニットの A12 の生産の値は、以下ようになります。
  - a. コンポーネント・コスト =  $\$30 + \$20 + \$105 = \$155$
  - b. 労働コスト (20%) =  $\$31$

- c. 供給 =  $\$3/\text{ユニット} = \$60$
  - d. A12 の総コスト =  $155+31+60 = \$246$
  - e. 既存の在庫値 =  $\$850$ 、数量 = 100
  - f. 新しい在庫値 =  $\$1096$ 、数量 = 120
  - g. A12 の新しい平均コスト =  $1096/120 = \$9.133333$
4. Sterling Selling and Fulfillment Foundation は、在庫値の変更に関連する情報を公開します (会計アプリケーションへの取り込みに使用されます)。在庫トランザクションおよび発行される具体的なデータについて詳しくは、時間トリガー・トランザクション付録についての「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

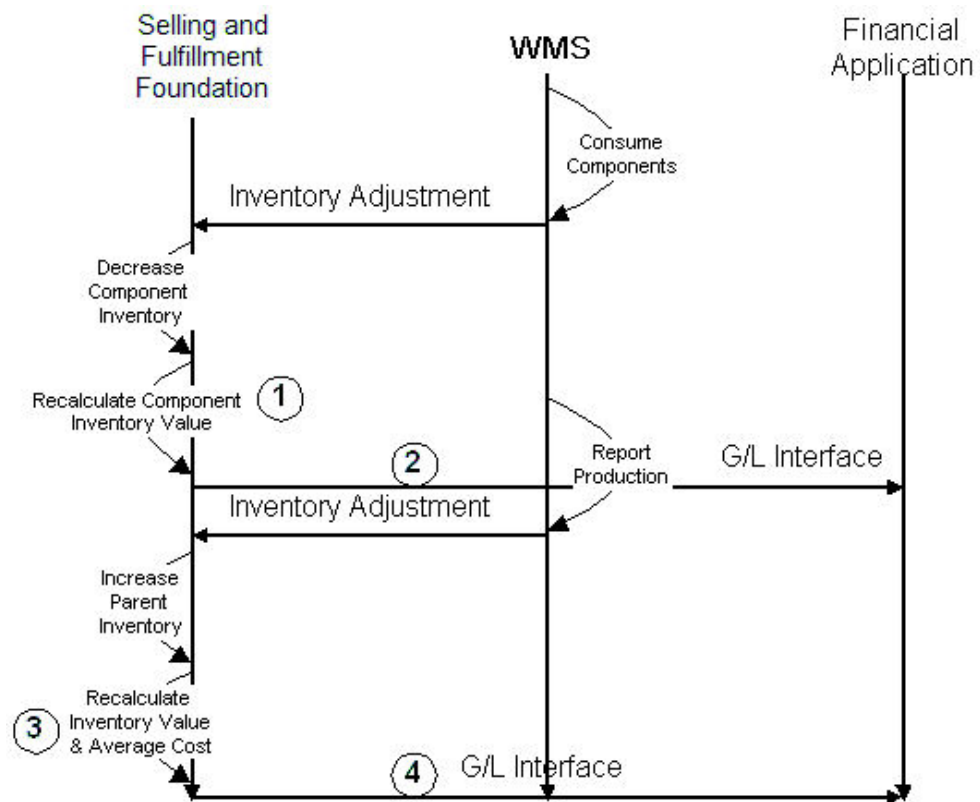


図 21. 仕掛品のシナリオ

### 平均コスト計算方法: ドロップ出荷製品の在庫コスト計算

Sterling Selling and Fulfillment Foundation は、直送オーダーを以下のように処理します。

#### オーダー作成

1. Sterling Selling and Fulfillment Foundation は、85 ページの『平均コスト計算方法: 在庫製品の在庫コスト計算』の説明のとおり、ユニット・コストを決定します。
2. Sterling Selling and Fulfillment Foundation は、親販売オーダー明細上のユニット・コストを連鎖購入オーダーの単価に転送します。



販売オーダー明細でユニット・コストがオーバーライドされない場合、Sterling Selling and Fulfillment Foundation が、ユニット・コストを連鎖購入オーダーの単価に転送する前に、標準コスト要因がユニット・コストから差し引かれます。

3. Sterling Selling and Fulfillment Foundation は、会計アプリケーションに対して、新しく作成された連鎖購入オーダーを公開します。これはバンダーのインボイス照合にも使用できます。
4. 単価は、この連鎖購入オーダーについてロックされます。

#### オーダー変更

5. Sterling Selling and Fulfillment Foundation は、親販売オーダー明細上のユニット・コストに対するあらゆる更新を対応する連鎖購入オーダー明細に転送します。
6. Sterling Selling and Fulfillment Foundation は、連鎖購入オーダー単価上の単価に対する更新を、親販売オーダー明細上のユニット・コストに伝搬して戻すことはしません。

#### 出荷の確認

7. Sterling Selling and Fulfillment Foundation が、出荷確認通知を受け取ったとき、「成功時」イベントが公開されます。このイベントを使用して、会計アプリケーション内の受入をシミュレートすることができます。また、同じイベントを使用して、買掛および売上原価に勘定記帳することもできます。これは、直接出荷された出荷にのみ行うことができます。

#### インボイス作成

8. Sterling Selling and Fulfillment Foundation は、販売オーダーにインボイスを作成し、このインボイスは、通常通り売掛および売上に記帳されます。すべてのトランザクションには、出荷の参照番号があり、これを使用してトランザクションが結びつけられています。

### 直送オーダーのシナリオ

『直送オーダーのシナリオ』では、Sterling Selling and Fulfillment Foundation がどのように直送オーダーを処理するかを説明しています。

#### 直送オーダーのシナリオ

顧客は、数量 2 ピースでアイテム DEF に対するオーダーを発行します。オーダーの価格は、顧客の価格設定ルールに従って決定されます。製品は、サプライヤー 1 からソーシングされ、顧客に直接出荷されます。予定コストは、DEF の主要サプライヤーに基づいて、DEF の置換コストから取得されます。オーダーがスケジュールされリリースされます。これによって、サプライヤー 1 に「連鎖」購入オーダーが生成され、2 ピースの DEF が顧客に出荷されます。サプライヤー 1 は、DEF を顧客に出荷し、インボイスをエンタープライズに送付します。インボイスを記録すると、「連鎖」購入オーダーに対する出荷の確認がトリガーされ、次に「連鎖」購入オーダーによって、元の販売オーダーに対する出荷の確認が伝搬されます。

出荷確認イベントが連鎖オーダーに対して発行されると、会計アプリケーション内で受入がシミュレートされます。また、この時点で、買掛および売上原価の総勘定元帳の勘定記帳が作成されます。これらの勘定記帳は、連鎖購入オーダー明細上の

情報に基づいて作成されます。インボイスは、元の販売オーダーに対して作成され、会計アプリケーションに公開されます。これは、売掛および売上に記帳されません。

#### 1. オーダーの作成

- a. オーダーの価格は、セラーによって定義された価格リストに従って決定されます。
- b. ユニット・コストが、オーダー明細上にスタンプされます。

#### 2. 連鎖購入オーダーの作成

ユニット・コストがオーバーライドされている場合 (特に販売オーダー明細に設定されている場合)、販売オーダー明細のユニット・コストが、連鎖購入オーダー上の単価になります。この時点で、単価はこの連鎖購入オーダーについてロックされます。ユニット・コストがオーバーライドされていない場合、オーダーの価格は、セラーによって定義された価格リストに従って決定されます。論理キット・アイテムの場合、販売オーダーのユニット・コストは購入オーダーに伝搬されません。この時点から購入オーダーの出荷確認まで、購入オーダーの単価は以下のいずれかで管理できます。

- 購入オーダーの場合。販売オーダーのユニット・コストに戻って伝搬されることはしません。
- ユニット・コストの販売オーダー管理の場合。購入オーダーの単価に伝搬されます。

#### 3. 出荷の確認

- a. 会計アプリケーション (サプライヤーからの買掛インボイスの受入によってトリガーされる) から出荷確認を受け取ると、Sterling Selling and Fulfillment Foundation は、連鎖購入オーダーに対するトランザクションを記録します。
- b. 連鎖購入オーダー・パイプラインには、インボイス・トランザクションは含まれていません。このシナリオでは必要ありません。
- c. 連鎖購入オーダーに対する出荷確認は、内部の Sterling Selling and Fulfillment Foundation の仕組みを介して、元の販売オーダーに対する出荷確認イベントをトリガーします。

#### 4. 受入インターフェース

エンタープライズでの在庫の受入はありませんが、サプライヤー・インボイスとの突き合わせに受入が使用されます。これには、会計アプリケーションの統合に必要な購入オーダー明細の詳細が含まれています。

#### 5. インボイス作成

販売オーダー明細の場合、Sterling Selling and Fulfillment Foundation は、売掛勘定記帳の作成で使用されるインボイス明細を作成します。

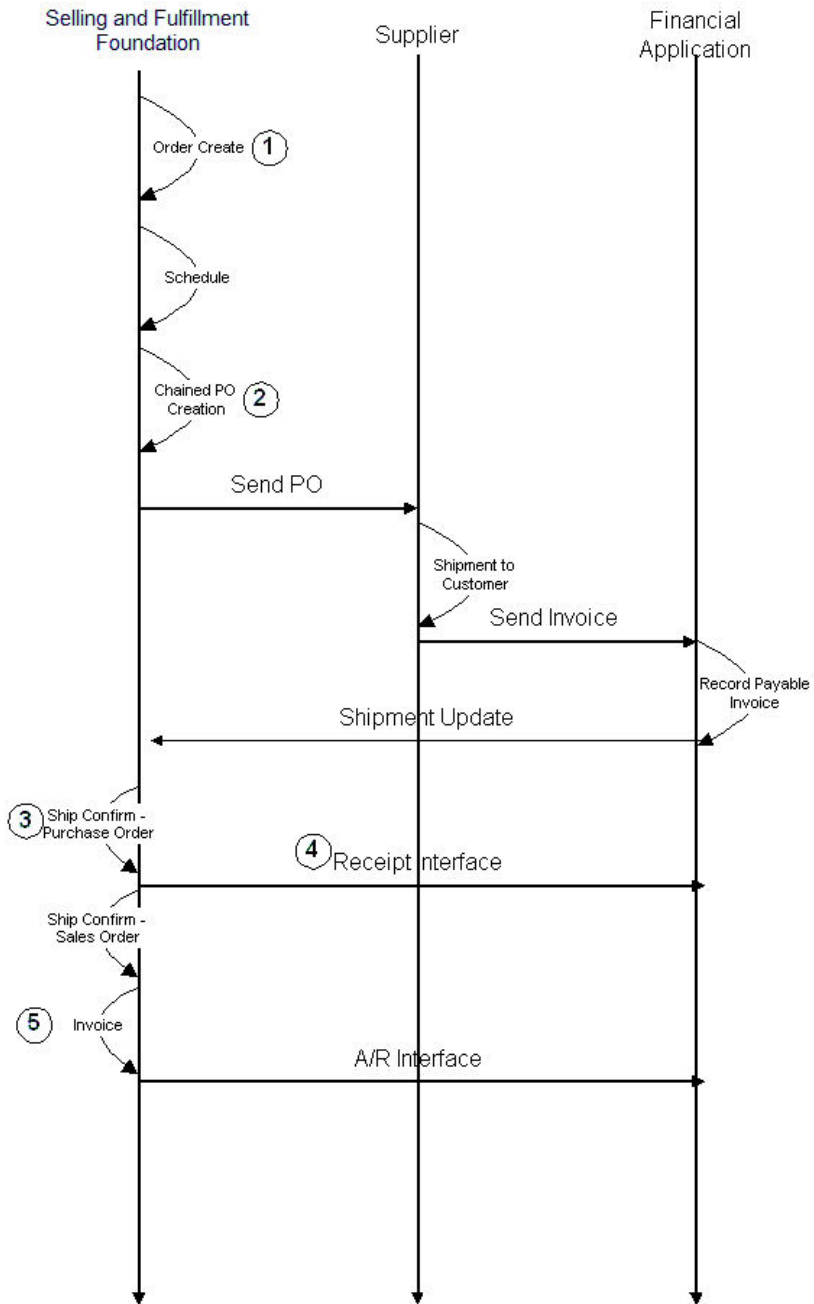


図 22. ドロップ出荷シナリオ

## 在庫コスト計算方法 2: 先入れ先出し (FIFO)

FIFO コスト計算方法の場合、Sterling Selling and Fulfillment Foundationエージェントは、先入れ先出し (FIFO) 順で受入と出荷を突き合わせ、突き合わせたコスト情報を記録します。この突き合わせ済みの情報は、レポート作成と会計システムへの計上のために使用されます。エージェントも、在庫コスト・データを会計システムおよび在庫システムに計上します。

返品処理に合わせて、在庫に対する減少は、FIFO 順にロールバックされます。つまり、返品アイテムのコストを決定するために、最後に出荷されたアイテムが返品されるものと見なされるということです。

Sterling Selling and Fulfillment Foundation は、在庫、出荷、および返品の実入を実行するために API を提供します。一般的な掛け勘定番号の決定と、必要に応じてコスト決定のロジックを提供するために、調整、出荷、および返品について外部プログラムが呼び出されます。在庫調整は、レコードを突き合わせテーブルに計上する前にコスト計算係が手動で見直せるように累算されます。これにより、コスト計算係は、総計アイテム・レベルでコストを調整し、調整を相殺することができます。

---

## ホット SKU

ホット SKU とは、特定の期間中に大量の要請のある人気アイテムのことです。例えば、非常に人気の高いアーティストの新しい CD のリリースにより、発売と同時にそのアイテムに対する大量の要請が生成される場合があります。

高いトランザクション負荷のもとで、ホット SKU が存在すると、同じアイテムに対して同時に複数の検索要求が行われる状態が生じる可能性があります。その状況が発生すると、1 つのトランザクションがロックを掛け、残りのトランザクションはブロックされます。

Sterling Selling and Fulfillment Foundation は、特定のアイテムをホット SKU として自動的に識別することができます。通常の状態では、ホット SKU が検出されなければ、システムは在庫変更中に 1 つのアイテムに対してロックを掛け、供給テーブルと需要テーブルを更新してから、ロックを解除します。アイテムがホットであると見なされた場合、システムはそれをロックしません。その代わりに、2 つの追加テーブル (1 つは需要用で、もう 1 つは供給用) に変更が挿入されます。

Sterling Selling and Fulfillment Foundation は、常に基本テーブルと追加テーブルの両方を見ています。追加テーブルが無制限に成長しないようにするために、Sterling Selling and Fulfillment Foundation は、基本テーブル、追加の供給テーブルおよび需要テーブルを調整するエージェントを提供します。

ホット SKU 機能について詳しくは、「*Sterling Selling and Fulfillment Foundation* パフォーマンス管理ガイド」を参照してください。

---

## 梱卸

在庫レベルの精度は、サプライ・チェーンにとって重要です。在庫レベルは、お客様の満足度の向上、需要計画、およびコストの削減の鍵となります。ラベル付け、一貫したマーク付け、およびバーコード付けが含まれている梱包制約を導入することにより、倉庫内のロケーションへ在庫を受け入れ、格納するときにヒューマン・エラー要因を低減することができます。また、ロケーションを必要とするシステムおよびスキャンされるパッケージ内で移動要請があるたびに、ヒューマン・エラーがさらに低減されます。

ただし、顧客または消費者がパッケージ計測単位と異なる数量でオーダーすることがよくあります。例えば、靴は、通常、ケース梱包で梱包されます。しかし、消費者が 1 足注文する場合があります。その場合は、パッケージを開ける必要があるため、人が製品を取り扱うこととなります。

梱卸には 2 つのタイプ、すなわち、物理梱卸と循環梱卸 (cycle count) があります。

物理棚卸は、通常、ゾーンなどの大きな領域、または倉庫に対して実行されるプロセスです。物理棚卸では、ユーザーは、通常は倉庫を停止して、在庫棚卸を実行します。この棚卸の後、この倉庫に関係するアクティビティー、例えば、格納、補充、取り出しなど、物理棚卸を開始する前にユーザーが停止したことを再開できません。

循環棚卸 (cycle count) は、通常、ロケーション範囲などの小さい領域に対して定期的に行われるプロセスです。循環棚卸 (cycle count) では、ユーザーは、格納、補充、取り出しなどのプロセスを同時に実行しながら、在庫棚卸を実行します。

棚卸システムは、計画済みの方法または暫定的な方法で棚卸を実行できるようにします。採用される一般的なタイプの棚卸は、年末棚卸です。これは、徹底的ですが、同時に、時間がかかるものでもあり、年間を通して精度が保障するわけではありません。もう 1 つの方法は、四半期ごとに製品回転または価格に基づいてアイテムをカウントするだけです。ただし、最良の方法は、システム内で棚卸が確実に定期的に行われるようにすることです。棚卸タスクは、循環棚卸 (cycle count) プログラム用に生成されるか、在庫制御部門により暫定的に要請されるか、あるいはロケーション・レベルで発生するシステム・イベントを通じて開始されます。

Sterling Warehouse Management System では、棚卸要請 (count request) を暫定ベースでコンソールから開始することができます。棚卸要請 (count request) は、1 つのノードまたはエンタープライズの一連のノードで採用される循環棚卸 (cycle count) プログラムに対して自動的に生成できます。格納、取り出し、またはピックと同様に実行されるアクティビティーの最中に記録される例外、あるいは最小レベルを下回るか、またはゼロまで減少したロケーション数量などのシステム・イベントを使用して、棚卸要請 (count request) を開始することもできます。

さまざまな棚卸要請 (count request) の例として、以下のものについての棚卸要請 (count request) があります。

- ロケーション
- アイテム、製品クラス、および単位の組み合わせ
- 受入番号を使用する購入オーダーまたは出荷に基づく、すべての受入済み製品
- ゾーン全体
- ケースまたはパレット
- 製品明細またはその他の任意のアイテム分類
- 地域レベル
- ノード・レベルまたはノード間
- 1 つのゾーン内のロケーション・レベル、通路、またはベイ・レベル
- 一連のロケーション

Sterling Warehouse Management System では、棚卸要請 (count request) に優先順位を付けることができ、開始時刻または終了時刻 (あるいはその両方) について日時ベースの条件を付けることができます。例えば、ユーザーは、低速移動アイテムがあるゾーンの棚卸を要請し、低い優先順位を割り当てることで棚卸を翌日に開始するよう要請することができます。開始時刻と終了時刻により計画および追跡を行うことができます。

Sterling Warehouse Management System は、最小レベルか、数量ゼロのロケーションについてのロケーション・レベル・アクティビティーに基づいてイベントを発生させる在庫モニターを提供します。

Sterling Warehouse Management System は、また、ピック後にロケーションが空であることを検出すると、必ずユーザーに確認を求めます。ユーザーがロケーションが空でないことを指定した場合は、棚卸要請 (count request) を作成するイベントがトリガーされます。これにより、次の循環棚卸 (cycle count) が行われる前でも、早期に差異が検出されます。確認は、取り出し、補充、およびピックを行っているときに行われます。ユーザーは少しも不便を感じることはありません。確認後、それぞれの割り当てられた作業を実行することができます。

棚卸要請 (count request) の作業は、ロケーション、アイテム、アイテムのカテゴリ、または容器レベルのものか、あるいは要請がなされたレベルに基づいてそれらを組み合わせるものになります。

---

## 在庫委託

在庫委託とは、1 つの在庫組織の在庫状況を別の在庫組織の在庫状況の一部と見なすことができる概念です。在庫委託プログラムに参加している在庫組織は、消費可能在庫組織 (consumable inventory organization) および消費在庫組織 (consuming inventory organization) と呼ばれます。

消費在庫組織 (consuming inventory organization) は、一緒に参加する消費可能在庫組織 (consumable inventory organization) のリストを選ぶことができます。消費可能在庫組織 (consumable inventory organization) の在庫状況は、消費在庫組織 (consuming inventory organization) の在庫状況の一部と見なすことができます。消費可能在庫組織 (consumable inventory organization) が、別の在庫組織の消費在庫組織 (consuming inventory organization) であることは許されません。つまり、1 つのレベルの関係のみが許されます。

在庫委託により、消費在庫組織 (consuming inventory organization) は、出荷の直前まで消費可能在庫組織 (consumable inventory organization) に在庫を管理することができます。出荷時に、在庫の所有権は消費可能在庫組織 (consumable inventory organization) から消費在庫組織 (consuming inventory organization) へ移されます。これにより、消費在庫組織 (consuming inventory organization) はその持ち越し費用を大幅に減らす一方で、正確な在庫ピクチャーを管理することができます。消費在庫組織 (consuming inventory organization) は在庫が出荷されるまで理論上はそれを所有しませんが、使用可能な在庫として、まだカウントすることができるためです。

エンタープライズおよびベンダーのモデルでは、エンタープライズの在庫組織を消費在庫組織 (consuming inventory organization) として構成できます。ベンダーの在庫組織は、エンタープライズと共に参加しますが、消費可能在庫組織 (consumable inventory organization) として構成することができます。

注: 在庫の所有権を移動する場合は、システムでは、消費可能在庫組織 (consumable inventory organization) と消費在庫組織 (consuming inventory organization) は同じ製品クラスを持つものと見なされます。

約束機能を使用して消費可能在庫組織 (consumable inventory organization) の在庫状況を入手する一方で、システムは、可能であれば、消費在庫組織 (consuming inventory organization) の過剰な需要を、参加している他の消費可能在庫組織 (consumable inventory organization) に最初に割り当てることにより、在庫状況を最大化しようとしています。

在庫ルールは、消費在庫組織 (consuming inventory organization) に基づいて読み取られます。消費可能在庫組織 (consumable inventory organization) が複数の消費在庫組織 (consuming inventory organization) と共に参加している構成では、システムは、参加している消費在庫組織 (consuming inventory organization) すべてに同じ一連の在庫ルールがあるかどうかを検証します。これらの在庫ルールは、「アイテム・ベースの割り当てを使用する」、「在庫ノード制御の将来の供給を使用する (Use Future Supply of Inventory Node Control)」、「将来の供給タイプのセーフティー要因率 (Future Supply Type Safety Factor Percentage)」、「手持ち供給タイプのセーフティー要因率 (Onhand Supply Type Safety Factor Percentage)」、および「デフォルト分配ルール ID (Default Distribution Rule ID)」など、ATP ルール内のルールです。

消費在庫組織 (consuming inventory organization) の在庫ルールは、消費在庫組織 (consuming inventory organization) および消費可能在庫組織 (consumable inventory organization) の在庫状況の入手時に使用されます。照会する組織のソーシングとスケジューリングソーシング関連ルールは、約束機能のために使用されます。

---

## バリエーションのあるアイテムの在庫状況の確認

Sterling Selling and Fulfillment Foundation では、バリエーションのあるアイテムの在庫状況を、指定された属性のセットに基づいて確認できます。この機能は、以下の機能を提供します。

- 属性および子アイテム (child item) による在庫のフィルタリング
- バリエーションおよび子アイテム (child item) のあるアイテムに関する供給と需要の管理

注: バリエーションのあるアイテムの在庫状況を確認できる機能を導入した結果、いくつかの API の動作が変更されています。これらの API の動作の変更について詳しくは、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

### 属性および子アイテム (child item) による在庫のフィルタリング

Sterling Selling and Fulfillment Foundation では、子アイテム (child item) または属性のセット (スタイル、サイズ、色など) を指定して、バリエーションのあるアイテムの在庫状況を確認することができます。例えば、オープントゥのサイズ 7 の靴の在庫状況を確認できます。

バリエーションのあるアイテムの在庫状況を確認する際に、子も属性も指定されていない場合は、Sterling Selling and Fulfillment Foundation は、バリエーションのあるアイテムおよびそのすべての子アイテム (child item) の情報を提供します。子アイテム (child item) または属性のセットを指定した場合、Sterling Selling and

Fulfillment Foundation は、指定された条件に合致するアイテムについてのみ、在庫状況を確認します。

## 属性および子アイテム (child item) によるフィルタリング

以下の表は、バリエーションのあるアイテム (T シャツ) の子アイテム (child item) に関する在庫情報を提供しています。

属性および子アイテム (child item) によるフィルタリング

アイテム ID	定義済み属性	手持ち数量
Tshirt_red_small	T シャツ、赤、S サイズ	10
Tshirt_red_medium	T シャツ、赤、M サイズ	11
Tshirt_red_large	T シャツ、赤、L サイズ	20
Tshirt_blue_small	T シャツ、青、S サイズ	10
Tshirt_blue_medium	T シャツ、青、M サイズ	11
Tshirt_blue_large	T シャツ、青、L サイズ	2

この例では、属性を何も指定しないで T シャツの在庫状況を確認した場合、システムは手持ち数量として 64 を返します。T シャツの在庫状況を確認し、「赤」属性を指定した場合、システムは手持ち数量として 41 を返します。Tshirt\_red\_large 子アイテム (child item) の在庫状況を確認した場合、システムは手持ち数量として 20 を返します。

## バリエーションおよび子アイテム (child item) のあるアイテムに関する供給と需要の管理

バリエーションのあるアイテムの在庫状況全体は、すべての子アイテム (child item) の在庫状況の合計と等しくなります。需要は、バリエーションのあるアイテムに対しては出されません。

個々の子アイテム (child item) に関する供給と需要は、アイテム別に管理されます。したがって、ある子アイテム (child item) の供給を、別の子アイテム (child item) の需要を満たすために使用することはできません。例えば、オープントウの靴の在庫を、バックストラップの靴の在庫不足に充当することはできません。

## 子アイテム (child item) に関する供給と需要の計算

以下の表は、バリエーションのあるアイテム (T シャツ) の子アイテム (child item) に関する在庫情報を提供しています。

子アイテム (child item) に関する供給と需要の計算

アイテム ID	手持ち数量	需要数量	購入可数量 (Quantity)	不足
Tshirt_red	10	0	10	0
Tshirt_blue	20	11	9	0
Tshirt_green	40	40	0	0
Tshirt_white	80	90	0	10



この例では、バリエーションのあるアイテム (T シャツ) の在庫状況は、すべての子アイテム (child item) の在庫数量を加算して (10 + 9 + 0 + 0) 計算されます。つまり、T シャツの在庫は 19 になります。

またこの例では、各子アイテム (child item) の供給と需要が分けて計算されることも示しています。したがって、在庫数量は Tshirt\_red が 10、Tshirt\_blue が 9、Tshirt\_green が 0、Tshirt\_white が 0 で返されます。また、Tshirt\_red または Tshirt\_blue の在庫数量を、Tshirt\_white の在庫不足に充当することはできません。Tshirt\_white は、10 の在庫不足を返します。

---

## 単一オーダー・アイテムに対する複数のフルフィルメント・アイテム

Sterling Selling and Fulfillment Foundation には、オーダーの 1 件のアイテムについて、カタログ内の複数のフルフィルメント・アイテムで在庫状況を確認する機能があります。このシナリオの「フルフィルメント・アイテム」とは、オーダーに記載されたアイテムである、オーダー・アイテムの需要の充足に使用され、1 つ以上のベンダーによって供給されるアイテムのことです。

Sterling Selling and Fulfillment Foundation では、グローバル・トレード・アイテム・ナンバー (GTIN) を使用して、フルフィルメント・アイテムとオーダー・アイテムが識別されます。オーダー・アイテムと対応するフルフィルメント・アイテムには、同じ GTIN が割り当てられます。また、オーダー・アイテムとフルフィルメント・アイテムは同じ単位、および同じ組織コードを使用する必要があります。ただし、すべてのアイテムに固有のアイテム ID が割り当てられます。

あるアイテムに対するオーダーが発生すると、同一の GTIN を共有するすべてのフルフィルメント・アイテムを対象として在庫状況が確認されます。オーダーのスケジュールはフルフィルメント・アイテムではなく、オーダー・アイテムに対して行われます。ただし、Sterling Selling and Fulfillment Foundation が、フルフィルメントを目的として外部ソースに通知を行う場合、システムは常に、オーダー・アイテムではなく、フルフィルメント・アイテムを基準とします。

外部ソースの通知は、オーダー・リリース、直送オーダー、および調達オーダーの際に行われます。パラメーター「連鎖オーダーとしてオーダーが作成された場合にフルフィルメント・アイテムを使用する」を使用すると、調達オーダーおよび直送オーダーで在庫を追跡できます。調達オーダー、および直送オーダーの在庫追跡について詳しくは、*Sterling Selling and Fulfillment Foundation: Supply Collaboration Configuration Guide*を参照してください。

需要は、オーダー・アイテムではなくフルフィルメント・アイテムに対して発生します。特定のアイテム ID のフルフィルメント・アイテムが利用可能でない場合、需要は一致する GTIN を持つ別のフルフィルメント・アイテムに対して設定されます。同じ GTIN を持つフルフィルメント・アイテムが利用可能でない場合、そのアイテムはオーダー上のアイテム ID に対してバックオーダーされます。

単一オーダー・アイテムに対する複数のフルフィルメント・アイテムの機能を有効にするには、以下の条件に合致していることを確認します。

- オーダー・アイテムとフルフィルメント・アイテムが同じカタログに属している。

- すべての組織が同じ在庫組織に属している。
- オーダー・アイテムとフルフィルメント・アイテムに同じ GTIN が割り当てられている。また、オーダー・アイテムとフルフィルメント・アイテムは、同じ単位、組織コード、および在庫パラメーターを共有していなければなりません。
- *Sterling Global Inventory Visibility* アプリケーションの「共有しているアイテム全体の在庫状況を読み取る」パラメーターが「GTIN」に設定されている。このオプションについて詳しくは、*the Sterling Selling and Fulfillment Foundation 統合在庫管理 構成ガイド* を参照してください。

単一オーダー・アイテムに対する複数のフルフィルメント・アイテムの機能を使用すると、カタログ組織 (catalog organization) 間で同じ GTIN を在庫委託に使用することはできません。在庫委託について詳しくは、「在庫委託 (Inventory Consignment)」を参照してください。

**注:** 単一オーダー・アイテムに対する複数のフルフィルメント・アイテムの機能を導入した結果、一部の API の振る舞いに変更されています。これらの API の動作の変更について詳しくは、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

### 単一オーダー・アイテムに対する複数のフルフィルメント・アイテム

下表は、オーダー・アイテム「用紙」の需要の充足に使用可能なフルフィルメント・アイテムの情報です。

ベンダー	フルフィルメント・アイテム	GTIN	手持ち数量
ベンダー A	用紙 A	1234	10
ベンダー B	用紙 B	1234	11
ベンダー C	用紙 C	1234	20

この例では、顧客が用紙を発注すると、*Sterling Selling and Fulfillment Foundation* では、GTIN 1234 のすべてのフルフィルメント・アイテムの在庫状況が確認されます。この例の場合、*Sterling Selling and Fulfillment Foundation* では、用紙の手持ち数量がアイテム数 41 としてレポートされます。用紙のオーダーが発生した場合、GTIN 1234 のいずれのアイテムもオーダーの充足に使用できます。オーダーを充足するにはこのようにして用紙 A、用紙 B、用紙 C が使用されます。

オーダーはアイテム「用紙」に対してスケジュールされますが、リリース・オーダーは、用紙 A などの対応するフルフィルメント・アイテムのいずれかに対して作成されます。

---

## 第 7 章 オーダー管理

---

### オーダー管理

Order Management では、プロセス・モデリングで開発されたプロセス・タイプ・パイプラインを使用して、作成、スケジュール、リリース、および出荷などのさまざまな段階にオーダー伝票を送り、そのライフサイクル全体を追跡します。

オーダー・コンソールによって、オーダー情報へのアクセスが提供されます。マネージャー、および顧客サービス担当者 (customer service representatives) は、リアルタイムでオーダー情報を表示して、警告の処理や問題の修正を行い、オーダーの定刻実行を保證できます。

---

### オーダーの構成要素

Sterling Selling and Fulfillment Foundation のオーダーは、オーダー・ヘッダー・レベル、オーダー明細レベル、およびオーダー・リリース・レベルに分けることができます。

オーダーはオーダー明細で構成されます。オーダー明細は、オーダー対象の個別のアイテムに関連する情報が記載されたオーダーの明細です。

例えば、バイヤーが 3 台のコンピューター・モニター、3 台のキーボード、および 3 台のプリンターが含まれる 1 件のオーダーを作成するとします。オーダー全体を見るときは、オーダー・ヘッダー・レベルを表示します。モニター、キーボード、およびプリンターは、それぞれに数量 3 が指定されたオーダー明細です。

一部の属性は、オーダー・ヘッダーとオーダー明細の両レベルに関係します。以下に例を示します。

- オーダーが送られる番地などの出荷先情報
- 贈答品のラッピングなどのカスタマイズ指示
- 出荷および取り扱いコスト
- 価格プロモーション

一部の属性は、オーダー・ヘッダー・レベルのみに関係します。以下に例を示します。

- インボイスの送付先の番地などの請求先情報
- バイヤーがクレジット・カード (credit card) と小切手のいずれで支払うかなどの支払い情報
- オーダー ID、オーダーの作成、出荷、および財務情報などの追加属性
- オーダーが交換オーダー (exchange order) の場合の交換タイプ (exchange type)。これには、標準交換、拡張交換、前納交換を使用できます。

一部の属性はオーダー明細レベルのみに関係します。以下に例を示します。

- アイテム ID

- 単一ユニットやダースなどの単位
- オーダーされた明細アイテムの数量
- 作成済みや出荷済みなどのオーダー明細ステータス
- 明細スケジュール
- ロット番号や改訂番号などの在庫属性
- シリアル番号
- 明細アイテムの単価などの価格設定情報
- 住所のマーク
- オーダー明細が贈答品である場合の贈答品属性

---

## 分類

製品アイテム分類、および分類階層を定義することができます。これらは、Sterling Selling and Fulfillment Foundation 内で、サービスのソーシングや関連付け、出荷プリファレンスの決定、オーダー対象のアイテムのリストの取得などのアクションに使用できます。分類を定義することにより、Sterling Selling and Fulfillment Foundation の特定の用途を持つアイテム属性を識別できます。

また、分類のために階層的なアイテムのグループ化を定義することもできます。これらのグループ化を使用して、分類目的 (classification purpose) によって影響を受けるアイテムをさらに詳細化できます。

---

## オーダー・パイプライン

ビジネス・プロセスをモデル化する際に、オーダー、交渉、計画オーダー、返品、購入オーダーのプロセス・タイプのパイプラインを作成できます。パイプラインは、フルフィルメント時にドキュメントが経るさまざまなステータスで構成されます。構成するパイプラインに関連するトランザクション (イベント、アクション、および条件から成る) を設定することもできます。

オーダー伝票は、次のプロセス・タイプのパイプラインを通過します。

- オーダー・フルフィルメント
- 交渉
- 購入オーダー
- マスター・オーダー
- 返品
- 見積フルフィルメント

## オーダー・フルフィルメント・パイプライン

オーダー・フローは作成から開始され、一連のトランザクションとステータスを経て完了に至ります。このトランザクションとオーダー・ステータスのチェーンは、オーダー・フルフィルメント・パイプラインと呼ばれます。

Sterling Selling and Fulfillment Foundation は、オーダー・フルフィルメント・パイプラインによって、そのパイプラインにおけるオーダーのロケーションに応じた通知の送信や警告のロギングなどのアクションを実行する手段を得ることができま

す。また、作成から完了までオーダーを追跡し、必要に応じて手操作による介入を実行する手段も提供されます。次の図は、オーダー・フルフィルメント・パイプラインの構造を示しています。

各企業のビジネスの起点となるオーダー・フルフィルメント・パイプラインは、システム管理者によるビジネス環境のセットアップ方法に応じて固有のものとなります。ただし、一般にオーダー・パイプラインはすべて、オーダーを作成するトランザクションによって開始され、オーダーの出荷が完了したこと、場合によっては返品されたことを示すトランザクションで終了します。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。オーダー・ステータスは、オーダーの現在状況を記述し、トランザクションごとに移動します。

オーダー・フルフィルメント・パイプラインでは、システム内での構成に応じて以下のステータスを使用することができます。

- 受諾済み - 交渉された条件が両当事者によって受諾され、オーダーのリリース準備が完了しています。
- 連鎖オーダー作成待機中 - オーダーが連鎖オーダー作成済みステータスに移動することになります。133 ページの『連鎖オーダー』
- 出荷集約待機中 - オーダーは、さらなる出荷および配達の実行前にノードで集約される複数の出荷で構成されます。
- バックオーダー済み - オーダーは作成されましたが、在庫が不足してオーダーのスケジュールを実行できません。在庫が利用可能になるまで、オーダーはバックオーダー状態で保持されます。
- ノードからバックオーダー済み - オーダーは作成され、ノードにリリースされましたが、ノードで在庫が不足しているため、オーダーを充足できません。この時点で、構成によっては、ノードで物理棚卸が実行されるまで、オーダー約束に利用できる在庫がないとされます。

構成によっては、Sterling Selling and Fulfillment Foundation ノード内の内部在庫を将来の在庫供給に対して約束できます。

- 交渉中 - オーダーは交渉パイプラインで交渉中です。
- キャンセル - オーダーはキャンセルされています。
- 連鎖オーダー作成済み - 連鎖オーダーが作成され、該当するノードに送信されています。
- 作成済み - オーダーは作成されています。
- 配達済み - 作業オーダーに記載されたサービスおよび製品明細が配達されています。
- ドラフト・オーダー作成済み - オーダー作成コンソール、「オーダー・エントリー」画面でドラフトオーダーが作成されています。確認済みになるまでは、このオーダーのあらゆる事項を変更できます。
- ドラフト・オーダー予約済み - ドラフト・オーダー (draft order) が作成され、このオーダーに在庫が予約されています。このオーダーが確認されておらず、オーダーが削除またはキャンセルされたときは、このオーダーに予約された在庫を他のオーダーに使用できます。

- 出荷に含まれている - このオーダーは出荷に含まれています。
- オーダー・インボイス作成済み - オーダーにインボイスが作成されています。
- 調達購入オーダー作成済み - オーダー作成コンソール、「オーダー・エントリ」画面で調達購入オーダーが作成されています。
- 調達購入オーダーが出荷済み - 調達購入オーダーが出荷されています。
- 調達転送オーダー作成済み - オーダー作成コンソール、「オーダー・エントリ」画面で調達転送オーダーが作成されています。
- 調達転送オーダー出荷済み - 調達転送オーダーが出荷されています。
- 交渉準備完了 - エンタープライズと出荷ノードで交渉が実行される時は、オーダーが独立した交渉パイプラインに送信されます。
- 受入済み - バイヤーがオーダーを受領しています。
- コンポーネントとして受入済み - オーダー内のアイテムがキット・コンポーネントで構成される場合は、このステータスによってバイヤーがすべてのコンポーネントを受領したことが示されます。
- リリース済み - フルフィルメントに向けたオーダーのスケジュールが可能な、十分な在庫が存在しています。オーダーは、アプリケーション・コンソール、Sterling Selling and Fulfillment Foundation 倉庫管理システム、またはサード・パーティーの倉庫管理システムにリリースされます。
- 予約済み - オーダーは作成されていますが、まだ出荷のスケジュールの準備が完了していません。
- 返品作成済み - オーダーに含まれる 1 つ以上のアイテムをバイヤーが返品しています。
- 返品受入済み - 返品アイテムが返品ノードで受け入れられています。
- スケジュール済み - 適切な 1 つ以上のノードにオーダーを充足できるだけの在庫が存在し、リリースのスケジュールが可能な状態です。
- ノードに送信 - オーダーのリリースとして、オーダーがノードに送信されています。
- 出荷済み - オーダーは出荷されています。
- 出荷遅延 - オーダー出荷のすべてまたは一部が遅延しています。
- 不足 - オーダーに記載された数量が要求数量より少なくなっています。オーダーはクローズされます。
- 受入取り消し済み - バイヤーがオーダーを受け入れていません。
- スケジュール解除 - オーダーのスケジュール済みステータスが削除され、スケジュール・ノードでこのオーダーに予約されていたすべての在庫がキャンセルされています。

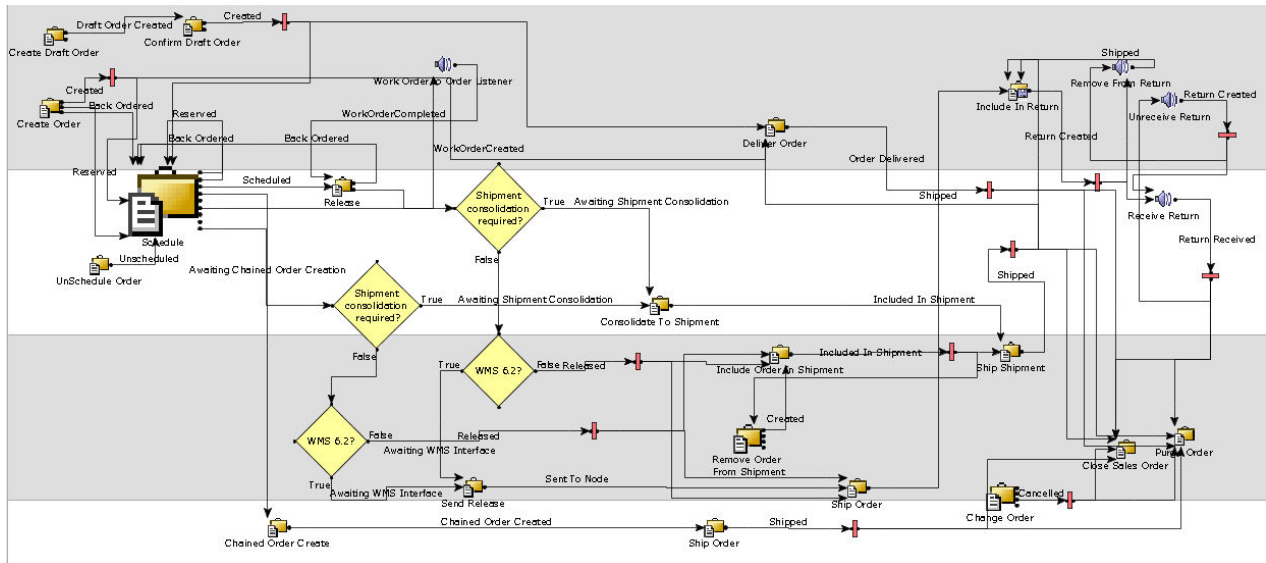


図 23. オーダー・フルフィルメント・パイプライン

デフォルトのパイプラインとステータスは、ユーザーのビジネス慣習に従って構成できます。オーダー・フルフィルメント・パイプライン、およびステータスの構成について詳しくは、*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイドを参照してください。

次の図は、デフォルトの販売オーダー・サービス・フルフィルメント・パイプラインを Application Manager に表示されるとおりにグラフィックで示しています。

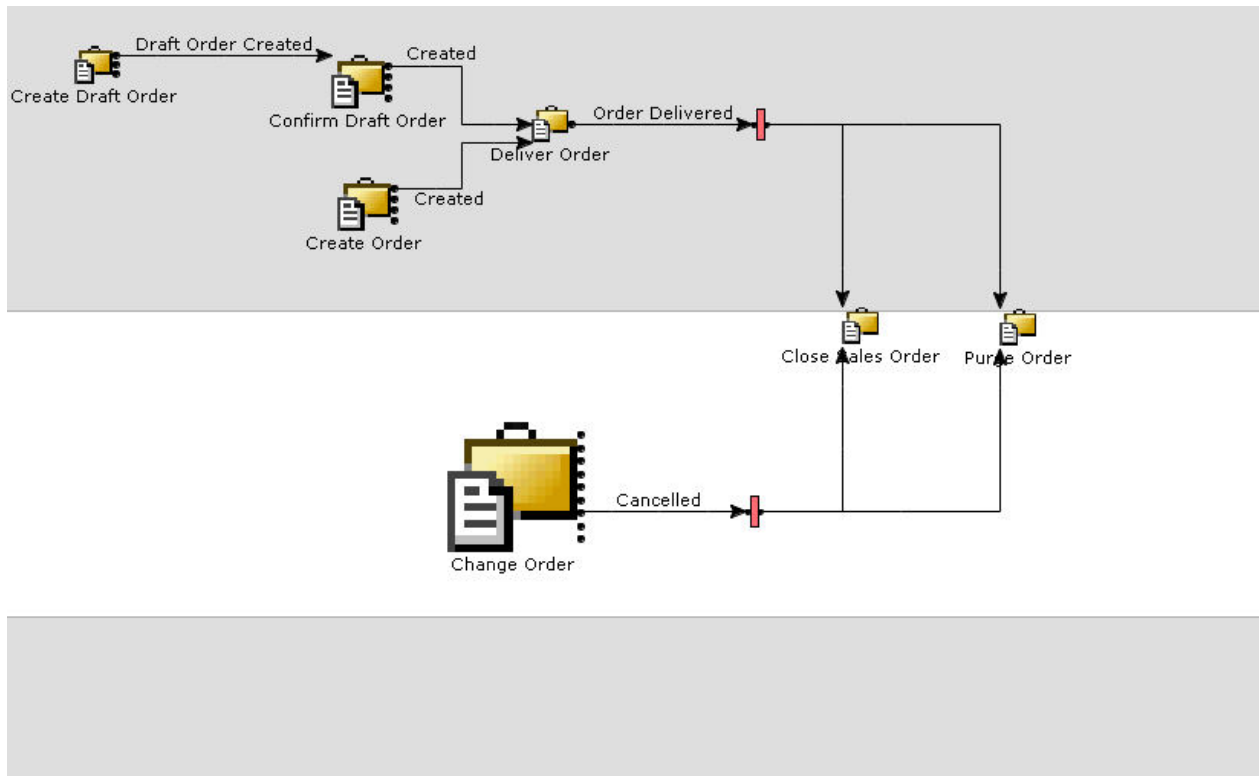


図 24. 販売オーダー・サービス・フルフィルメント・パイプライン

## 交渉パイプライン

交渉とは、2つの組織がプロセス・タイプ・ドキュメントの条件を確定するためのプロセスです。交渉は、オーダー作成後に構成できます。

交渉プロセスをオーダー・フルフィルメント・パイプラインに取り込む必要がある場合は、必ずフルフィルメント伝票（オーダー、計画オーダーなど）から交渉ドキュメントを作成します。交渉はすべて交渉ドキュメントを通じて実施され、結果は元のドキュメントに適用されます。

交渉パイプラインを使用すると、参加者は現在の取引の詳細を交渉できます。このパイプラインは、既存パイプライン内でリリース前のいずれかの状況で行われるように構成できます。交渉プロセスからの結果は、交渉価格の一部を更新する可能性がある形式の受諾、あるいは、一方の参加者または双方による取引条件の拒否のいずれかになります。

### 交渉パイプラインにおける交渉の応答

交渉プロセスには2つの組織が関係します。これは、イニシエーター組織とネゴシエーター組織です。交渉は相互に応答を送信することによって実行されます。応答は、1つのオーダー明細を対象とすることも、すべてのオーダー明細を対象とすることもできます。一方の組織の応答がもう一方の組織に受諾されると、交渉は完了ステータスに進みます。交渉の完了後、交渉された条件が元のドキュメントに適用されます。



応答は、応答番号によって識別されます。すべての応答に他方の組織からの最終応答の応答番号を「参照応答番号」として指定する必要があります。表 15 は、交渉の応答を示しています。

表 15. 交渉の応答

応答番号	応答名	説明
1100	申し込み	この応答は、イニシエーターからの申し込みです。イニシエーター組織のみがこの応答を送信できます。
1200	修正申し込み	この応答は、ネゴシエーターからの修正申し込みです。ネゴシエーター組織のみがこの応答を送信できます。
1300	拒否	この応答はネゴシエーターからの拒否です。ネゴシエーター組織のみがこの応答を送信できます。
1400	削除	イニシエーターが交渉からの明細の削除を希望しています。イニシエーターのみがこの応答を送信できます。この応答は、明細レベルのみで使用できます。明細が削除されると、その明細は交渉が完了したとされ、これ以降その明細の交渉はできなくなります。
1500	承認 (Accept)	送信側組織が他方の組織の条件を受諾します。イニシエーターとネゴシエーターの双方がこの応答を送信できます。ヘッダー、または明細を受諾されると、そのヘッダーまたは明細は交渉が完了したとされ、これ以降はそのヘッダーまたは明細の交渉はできなくなります。

## 交渉パイプラインにおける交渉アクション

応答アクションが受諾または削除の場合は、関連アクションが設定されています。オーダー、および計画オーダーのアクションは、キャンセルです。アクション情報は、交渉された条件が元のドキュメントに適用される場合に使用されます。2 つの組織が元のドキュメントの数量より少ない数量で合意する場合があります。その場合は、残余数量について指定された交渉アクションが実行されます。

交渉には、ヘッダーと複数の明細を設定できます。交渉ステータスはヘッダー・レベルで管理され、交渉に関する最終応答から派生します。

表 16. 交渉ステータス

ステータス・コード	ステータス名	説明
1000	申し込み済み	交渉での最終応答はイニシエーター組織からの応答です。ネゴシエーター組織が元の申し込みに対して応答する前ならば、イニシエーターは別の申し込みを送ることができます。

表 16. 交渉ステータス (続き)

ステータス・コード	ステータス名	説明
1100	修正申し込み済み	交渉の最終応答は、ネゴシエーター組織からの応答です。イニシエーター組織が元の修正申し込みに対して応答する前ならば、ネゴシエーター組織は別の修正申し込みを送ることができます。
1200	完了	2 組織間でヘッダー条件およびすべての明細の明細条件の交渉が完了しています。
1300	公開済み	交渉された条件が公開され、元のドキュメントに適用されています。

## 交渉パイプラインにおける交渉プロセス

図 25 は、一般的な交渉プロセスに関連する取引とステータスを示しています。交渉ステータスは、交渉対象のドキュメント全体のヘッダー・レベルに設定されます。

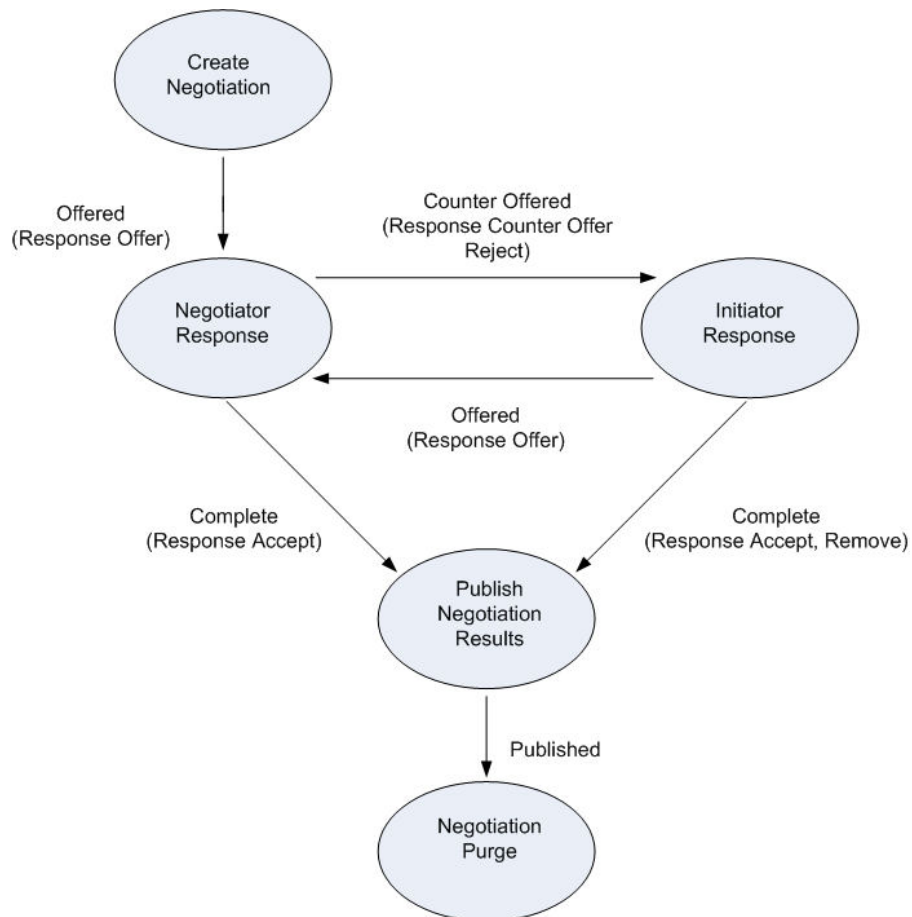


図 25. 交渉プロセス

交渉は、オーダー作成後に構成できます。

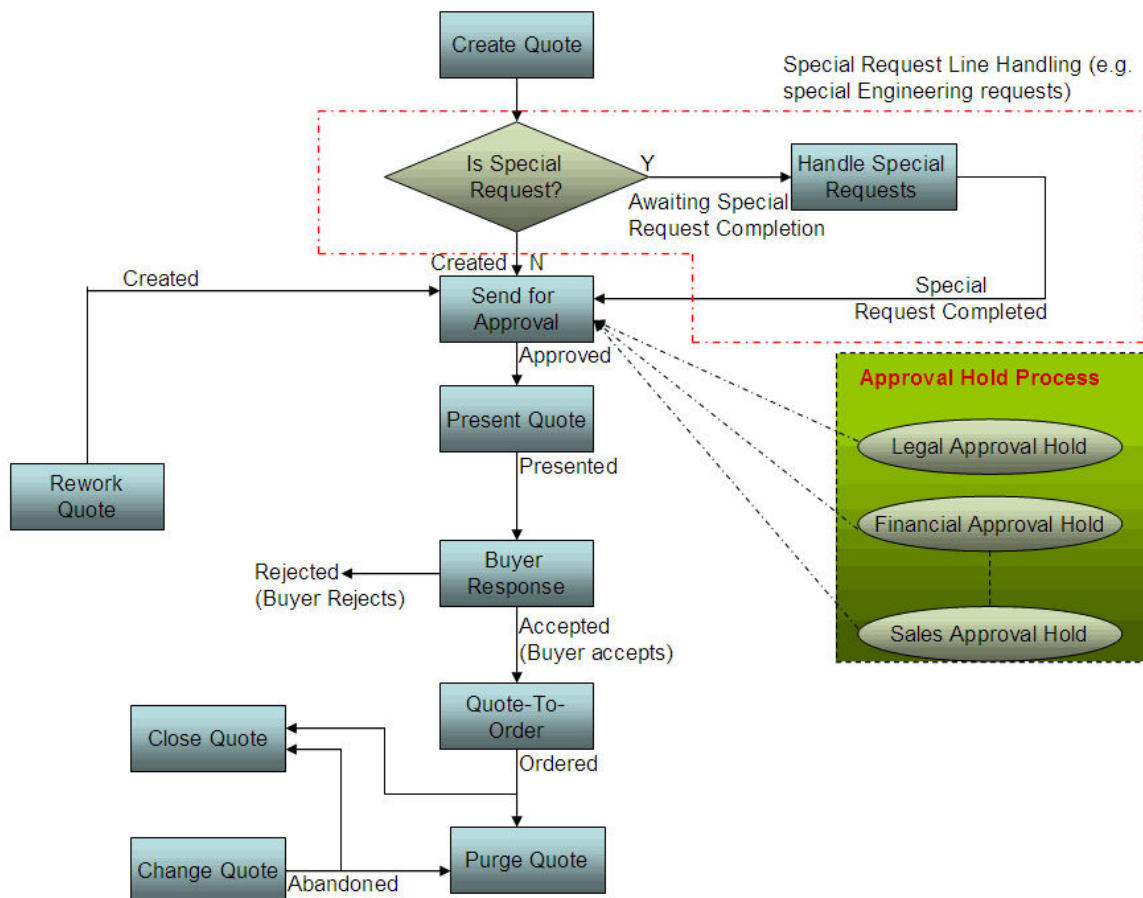
## 見積フルフィルメント・パイプライン

作成から完了に至るまで、見積は一連のトランザクションとステータスを経由します。この見積のトランザクションとステータスのチェーンを「見積フルフィルメント・パイプライン」と呼びます。

見積フルフィルメント・パイプラインでは、次のステータスを使用できます。

- 作成済み - 見積は作成済みです。
- 承認済み - 見積は必要な承認者すべて（該当する場合）によって承認済みです。
- 提示済み - 見積は顧客に提示済みです。
- 拒否済み - 見積は顧客により拒否済みです。
- 受諾済み - 見積は顧客により受諾済みです。
- オーダー済み - 見積からオーダーが作成済みです。
- 中止 - 次のいずれかの理由で見積が中止されました。
  - 見積が顧客により受諾された際に、その案件の関連する見積すべてが中止された。
  - 顧客はすでに見積について交渉を求めている。
  - 見積の有効期限が切れている。

以下は、見積フルフィルメント・パイプラインのグラフィカル表示です。



注: 特別要求は、パイプライン拡張機能によって処理できます。これを実装するには、明細アイテムが特別要求かどうかを判断するための条件を見積フルフィルメント・パイプラインに設定します。この条件フラグが Y を返し、明細アイテムが特別要求であることを示している場合、createOrder API 呼び出しにより、その明細アイテムのステータスは自動的に「特別要求の完了待ち (Awaiting Special Request Completion)」に移され、特別要求を特別要求トランザクションで取得できるようになります。エンジニアリング・チームは、特別要求を解決したら、見積明細を手動で「特別要求完了済み (Special Request Completed)」ステータスに移し、明細アイテムを送信して見積フルフィルメント・パイプラインで承認できるようにします。

## マスター・オーダー・パイプライン

マスター・オーダーは、一定の期間にわたり複数の製品の反復的な販売が予測されることを示す伝票です。一般に、マスター・オーダーは一連の出荷製品を購入する顧客に対して作成されます。

図 26 は、一般的なマスター・オーダー・プロセス・タイプのパイプラインを示しています。

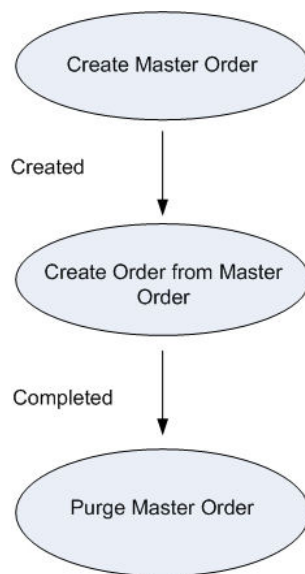


図 26. マスター・オーダー・パイプライン

---

## 交換オーダー (exchange order)

交換オーダー (exchange order) は、ORDER\_PURPOSE フラグが「EXCHANGE」に設定された販売オーダーです。これは、返品オーダーに基づいてのみ可能であり、販売オーダーと同様に動作します。

交換として顧客に送付される製品の値が、顧客が返品した製品の値と同じ場合、交換は同等です。例えば、顧客が、青いシャツを返品し、交換希望は、色が赤というだけで、スタイルとサイズは同じシャツです。

交換として顧客に送付される製品の値が、顧客が返品した製品の値より大きいか、または小さい場合、交換は同等ではありません。例えば、顧客が、青いシャツを返品し、返品に緑色のシャツのほかに、赤いシャツも希望しています。その場合、新しい決済処理を行う必要があります。

交換には、次のように 3 つのタイプがあります。

- 標準交換
- 拡張交換
- 拡張、前納交換

### 標準交換

標準交換が発生するのは、エンタープライズが、交換を送る前に、顧客からの返品を待つ場合です。交換アイテムは、返品アイテムを受け取っている場合および必要に応じて差引残高が支払済みになっている場合にのみ、送られます。

例えば、顧客が、赤いシャツを送り返し、交換で青いシャツを要求して、追加で緑色のシャツの購入も希望しています。エンタープライズは、赤いシャツを受け取ったら、緑色のシャツの代金を顧客に対して請求し、両方のシャツを一度に送ります。

### 拡張交換

拡張交換が発生するのは、エンタープライズが、返品アイテムの受け取りを待たずに、交換が作成されたら即時に交換を出荷する場合です。例えば、携帯電話会社が、CSR が要請を受け取ると即時に新しい携帯電話を顧客に送ることができます。会社は、返品を受け取るものと仮定しており、交換を発行するのを待つことはありません。

### 拡張前納交換

拡張前納交換が発生するのは、エンタープライズが顧客に対して交換アイテムの料金を請求し、それを出荷して、その後、返品アイテムを受け取った時点で顧客に返金する場合です。例えば、顧客が赤いシャツを青いシャツと交換することを希望している場合、エンタープライズは、顧客に対して青いシャツの料金を請求し、それを出荷して、赤いシャツを受け取った時点で、その赤いシャツの値を顧客に返金します。

交換オーダー (exchange order) について提供サービスが関係している場合、Sterling Selling and Fulfillment Foundation は、交換アイテムの配達および搬入と共に古いアイテムの引き上げとピックアップが含まれる単一の作業オーダーを作成できるようにするために、交換オーダー (exchange order) を返品オーダーと同期化します。

---

## バンドル

バンドルは、製品およびサービスの両方を含むパッケージです。バンドルは、バンドルの親と 1 つ以上の子明細から構成されています。子明細は、製品、サービスまたは他のバンドルのいずれかになります。バンドルの親および子明細バンドルは、マーケティング目的でのみ作成され、オーダーの一部として履行することはできません。多くの家具会社から提供されている寝室のパッケージなどが、バンドル・オ

オーダーの 1 つの例です。寝室パッケージには、化粧台、ナイトテーブル、寝具セットのバンドル (マットレスやベッドのフレームなど)、および寝室パッケージの設置サービスなど、多数の異なる製品およびサービスが含まれています。図 27 では、一般的なバンドルを説明しています。



図 27. バンドルの例 - 寝室パッケージ

## バンドルのモデル化

Sterling Selling and Fulfillment Foundation では、バンドルをモデル化するため、オーダー構造で階層オーダー明細がサポートされます。表 17 は、Sterling Selling and Fulfillment Foundation におけるバンドルのモデル化を示しています。

表 17. バンドルのモデル化

オーダー明細	バンドル親明細
寝室パッケージ (親明細)	空欄
ドレッサー	寝室パッケージ
ナイトテーブル	寝室パッケージ
ベッド据え付け	寝室パッケージ
ベッド・セット・パッケージ (親明細)	寝室パッケージ
マットレス	寝具パッケージ
ベッド・フレーム	寝具パッケージ
ヘッドボード	寝具パッケージ
フットボード	寝具パッケージ

126 ページの表 17 に示されるように、各コンポーネントにその親明細が示され、親明細のそれぞれには、バンドルの親明細であるかどうかを示されます。オーダー明細にバンドルの親明細が存在する場合、その明細はバンドルの構成部分とみなされ、バンドル・コンポーネントと呼ばれます。

## 追加のバンドル機能

Sterling Selling and Fulfillment Foundation は、バンドルに関連して以下の追加機能を提供しています。

- オーダー作成 OK
- バンドル・コンポーネントのオーダー数量計算
- 在庫更新
- サービス関連付け
- オーダー明細の属性の同期
- オーダー変更
- オーダーのスケジューリング
- 価格
- 連鎖オーダー
- 派生したオーダー (derived order)
- 返品

## オーダーの作成

n 層バンドルを含むオーダーを作成できます。オーダーの作成プロセスでは、バンドル親情報が渡される場合に限り、カタログからバンドル定義を読み取り、製品を予約するためのソーシング制約を受け入れます。外部システムを使用してカタログを管理する場合は、Sterling Selling and Fulfillment Foundation はバンドル親情報をその外部システムに渡すことによってバンドル定義を読み取ります。

Sterling Distributed Order Management は、サード・パーティーと通信したり、他の関連作業単位を実行したりするために、バンドルが従属バンドルを作成できるようにします。これにより、必ず、従属バンドルは、該当する場合は親バンドルにステータス更新のレポートを返し、親バンドルは、リンクされた値の変更を各従属バンドルに連絡できるようになります。126 ページの表 17 に示す例は、n 層バンドルのものです。

126 ページの図 27 を使用して、以下の例では、126 ページの表 17 で説明しているバンドルがカタログに存在するものと見なして、顧客の要件に基づいて渡される情報について詳細が提供されます。

### 顧客が標準寝室パッケージを要請する

この例では、顧客は標準寝室パッケージを要請します。寝室パッケージ・バンドルに対する変更はないため、オーダーの作成時に、寝室パッケージ情報のみが渡されます。寝室パッケージ・バンドルに対して、以下のオーダー明細が作成されます。

- 寝室パッケージ (親バンドル)
- ドレッサー

- ナイトテーブル
- ベッド据え付け
- ベッド・セット・パッケージ (親バンドル)
  - マットレス
  - ベッド・フレーム
  - ヘッドボード
  - フットボード

### 顧客が寝室パッケージの変更済みバージョンを要請する

この例では、顧客は標準寝室パッケージを希望していますが、ナイトスタンドは不要です。要請された寝室パッケージの構成要素に変更があったため、寝室パッケージ・バンドルとその直接構成要素が渡されます。変更済み寝室パッケージ・バンドルに対して、以下のオーダー明細が作成されます。

- 寝室パッケージ (親バンドル)
- ドレッサー
- ベッド据え付け
- ベッド・セット・パッケージ (親バンドル)
  - マットレス
  - ベッド・フレーム
  - ヘッドボード
  - フットボード

### 顧客が寝室パッケージの変更済みバージョンとベッド・セット・パッケージを要請する

この例では、顧客は標準寝室パッケージを希望していますが、ナイトスタンドまたはマットレスは不要です。寝室パッケージ・バンドルおよびベッド・セット・バンドルの両方に変更があったため、寝室パッケージ・バンドルとその直接構成要素およびベッド・セットとその構成要素の両方が渡されます。寝室パッケージ・バンドルに対して、以下のオーダー明細が作成されます。

- 寝室パッケージ (親バンドル)
- ドレッサー
- ベッド据え付け
- ベッド・セット・パッケージ (親バンドル)
  - ベッド・フレーム
  - ヘッドボード
  - フットボード

## バンドル・コンポーネントのオーダー数量計算

バンドル・コンポーネントのオーダー数量は、親明細に依存しない場合も、親に依存する場合もあります。これは、オーダー明細のキット数量、またはコンポーネント・オーダー明細の比率数量によって決まります。オーダー明細でキット数量が指定されている場合、コンポーネント明細の数量は、親明細のオーダー数量にコンポ



ーメント明細のキット数量を掛けた数量です。キット数量が指定されている場合は、親明細の数量が変更されると、コンポーネントの数量に影響し、また、逆にコンポーネントの数量の変更も親明細の数量に影響します。コンポーネントでキット数量が維持される場合は、コンポーネントから親へのステータス変更、および親からコンポーネントへのステータス変更も伝搬される可能性があります。

コンポーネントは、固定オーダー数量で作成することもできます。この場合、親の数量の変更が必ずコンポーネントに適用されるとは限りません。コンポーネントの数量変更についても同じです。変更の処理方法の相違について詳しくは、130ページの『オーダー変更』を参照してください。

**注:** 親からコンポーネント、およびコンポーネントから親への数量関連の変更の伝搬を容易にするには、キット数量を使用してバンドル・コンポーネントを作成します。

## 在庫更新

バンドル親明細は、非在庫アイテムです。バンドル親明細については、在庫更新は発生しません。

## サービス関連付け

バンドルをカタログから読み取るときに、サービス・アイテムとの関連付けを自動的に設定できます。

バンドル内の各製品コンポーネントは、バンドル内に存在するアイテムに適用可能なサービスに関連付けられます。

126ページの図27を例に取ると、化粧台、ナイトスタンド、および寝具パッケージには、関連付けられたサービスとしてベッドの設置があり、オーダーには次の関連付けが作成されます。

表 18. バンドル内のサービス関連付けの例

商品 (Product)	関連付けられているサービス
ドレッサー	ベッド据え付け
ナイトスタンド	ベッド据え付け
寝具パッケージ	ベッド据え付け

バンドルに配送サービスをコンポーネントとして設定することはできません。これは、バンドルのコンポーネントが複数配達されるのを防ぐためです。バンドルまたはバンドル・コンポーネントを配達する必要がある場合は、配送サービス明細をオーダーに渡すことができます。

配送サービスをバンドル親に関連付けることができます。バンドル親に関連付けられている配送サービスがオーダーに渡されると、すべてのコンポーネントは1つの配送サービスを使用して配達されます。配送サービス明細をバンドル・コンポーネント・レベルで渡すことはできません。特定の配送サービスを使用してバンドル・コンポーネントを配達する必要がある場合は、そのサービスをコンポーネントに関連付けることができます。

## オーダー明細のスケジュール属性の同期

バンドルの親オーダー明細を充足できないときは、その親の出荷ノード、予定出荷日、および予定配達日の属性は計算されません。バンドル・コンポーネントでこれらの属性が変更されると、バンドル親のオーダー明細のスケジュール属性が決定されます。

例えば、126 ページの表 17 の説明どおりにバンドルにオーダーが作成されたとします。スケジュール・トランザクションが実行されると、バンドル親の出荷ノード、予定出荷日、および予定配達日が決定されます。

## オーダー変更

コンポーネント・レベルで実行された変更が親に影響することもあるれば、親レベルで実行された変更がコンポーネントレベルに影響することもあります。ここでは、親明細と子明細の両方に影響するすべての変更について説明します。

### 数量の追加

数量がバンドル親に追加されると、キット数量が指定されたバンドル内の全コンポーネントの数量も追加されます。これに使用される式では、親の数量の変更がコンポーネントのキット数量に掛けられます。126 ページの表 17 を例として使用し、顧客が寝室パッケージ一式をオーダーし、さらにもう一式、寝室パッケージの購入を希望しているとします。オーダー明細での寝室パッケージの数量の変更によって、コンポーネントが比例していると想定してすべてのコンポーネントの数量が増加します。

数量がコンポーネント・レベルで追加された場合は、バンドル親の数量が変更されます。親明細の数量は、すべてのコンポーネントの最大数量です。この結果として親明細の数量が増加される場合、この変更は許可されません。コンポーネント・レベルの数量変更で親の数量を増加させることはできません。コンポーネントの数量の増加は、いったんキャンセルされた数量を、顧客が再度増加することを希望した場合にのみ許可されます。

### 明細の分割

コンポーネントの数量を分割すると、テンプレートに基づいてキット数量が新規明細に伝搬されます。新規明細が固定数量明細の場合は、親明細で実行された変更による影響を受けません。バンドル全体の分割はサポートされません。比率が維持される場合は、キット数量が両明細で同じ場合、数量変更においては分割明細と元の明細が 1 つの明細であるとみなされます。

### 明細の追加

コンポーネントは、既存のバンドルの親明細に追加できます。コンポーネントは、製品、サービス、または別のバンドル明細になります。

### 明細の削除

コンポーネント明細は、バンドルから削除できます。コンポーネントが削除されると、親バンドルの数量が変更されます。バンドル親が削除されると、すべての子明細も削除されます。

## バンドルのキャンセル

セット・キット数量が設定されているコンポーネントはすべて、親明細で数量が減少すると、親明細数量の変更にコンポーネントのキット比率を掛けた数量だけ、コンポーネントも減少します。親がすべてキャンセルされると、固定数量コンポーネントもキャンセルされます。

バンドルのコンポーネントの「キット数量」が設定されていない場合、親明細で数量減少が発生しても、コンポーネントの数量は減少しません。

コンポーネント数量も、別個に減らすことができます。コンポーネントをキャンセルすると、親の数量が調整されます。

## オーダーのスケジューリング

バンドルでは親のそれぞれがバンドルのスケジュール方法を指示します。バンドルで使用可能なスケジュール値は以下のとおりです。

- 一括出荷
- 別々に出荷
- 一括配達

これらの値は、カタログから取得されます。カタログが使用できない場合、Sterling Selling and Fulfillment Foundation により、バンドルを一括出荷するように決定されます。これらの値は、製品明細コンポーネントのみに適用され (サービス明細コンポーネントは無視されます)、バンドルのすべてのレベルのコンポーネントに適用されます。

## 価格

価格設定に関しては、バンドル親はその他のオーダー明細と同様に処理されます。明細合計はその他のオーダーと同じ方法で計算されます。オーダー合計には、バンドル親およびすべてのコンポーネントが含まれます。バンドルには、その他の明細と同様に、単価、料金、および税が含まれます。

オーダー明細レベルの属性には、明細の価格を抑制し、顧客が支払義務を負うオーダー合計または全額から除外できるものがあります。明細の価格は表示して、それがオーダー合計から除外されるようにするには、そのコンポーネントにこの属性を設定します。バンドルのコンポーネントの価格がパッケージ自体の価格に含まれていても、キャンセルや返品に備えて、各明細の価格も保存したい場合には、その明細にこの属性を設定できます。バンドルの特別価格を反映するには、コンポーネントの価格がオーダーの合計価格に含まれないように、この属性をコンポーネントに設定できます。

表 19 を例として使用すると、寝室パッケージ・バンドルおよびそのコンポーネントの価格を次のように設定できます。

表 19. 寝室パッケージの価格設定

寝室パッケージ	明細合計額	情報提供のための価格
寝室パッケージ (バンドル)	\$500	
ドレッサー	\$200	はい

表 19. 寝室パッケージの価格設定 (続き)

寝室パッケージ	明細合計額	情報提供のための価格
ナイトテーブル	\$100	はい
ベッド・セット・パッケージ (バンドル)	\$300	はい

通常は、オーダー合計はすべての明細の合計に一致し、この場合は \$1100 になります。バンドル親を基に、オーダーの価格を別の方法で設定し、コンポーネント・レベルの価格がインボイスに含まれないようにする場合は、オーダー作成時にコンポーネント・レベルで属性を設定できます。これらの属性の詳細については、「*Sterling Selling and Fulfillment Foundation Javadocs*」を参照してください。

## バンドルおよび連鎖オーダー

スケジュール済みのトランザクションがコンポーネントを検索して、直送オーダーを作成するとき、バンドル親は可視性を目的として、連鎖オーダーに取り込まれます。バンドル親は、連鎖オーダー内で親オーダーにリンクされません。これは、親オーダーでバンドルの親が変更されても、連鎖オーダーに表示されないことを意味します。スケジュール済みの属性および数量の変更のみが表示されます。

連鎖オーダーの概要については、133 ページの『連鎖オーダー』を参照してください。

## 派生したオーダー (derived order)

派生したオーダー (derived order) が作成された時点で、オンライン明細がバンドル親である場合は、比率になっている適切なコンポーネントもすべて、派生したオーダー (derived order) に入れられます。コンポーネントに対するバンドル親の関係は、新しいオーダーで管理されます。親バンドルに含まれているサービス明細はどれも、派生したオーダー (derived order) には含まれません。

一般に、派生したオーダー (derived order) について詳しくは、136 ページの『派生したオーダー (derived order)』を参照してください。

## 返品

返品は、コンポーネントに対しても、バンドル親に対しても作成できます。バンドル親に対して返品を作成した場合、比率内の返品可能コンポーネントがすべて返品オーダーに含まれます。コンポーネントに対して返品を作成した場合、すべてのコンポーネントが返品オーダーに含まれていても、バンドル親は返品オーダーに含まれません。例えば、126 ページの図 27 で、ベッドのフレームのみが返品される場合、バンドル親は返品オーダーに含まれません。寝室パッケージを返品する場合は、すべての子明細 (化粧台、ナイトスタンド、寝具セット・パッケージ) が返品オーダーに含まれます。

返品全般について詳しくは、305 ページの『第 14 章 返品物流』を参照してください。

---

## 多層オーダー

Sterling Selling and Fulfillment Foundation Order Management には、サード・パーティーとの通信、またはその他の関連作業単位を実行するため、オーダーに従属オーダーを作成する機能があります。これにより、必要に応じて従属オーダーのステータス更新が親オーダーにレポートされ、親オーダーではリンクされた値の変更を各従属オーダーに通知できることが保証されます。

Sterling Selling and Fulfillment Foundation には、次のタイプの多層オーダーがあります。

- 連鎖オーダー
- 派生したオーダー (derived order)

多層オーダーのそれぞれは独自のインターフェースを使用し、独自のパイプラインに従います。連鎖オーダーにパイプラインを構成し、そこからさらに別の多層オーダーを作成できます。理論的には、指定されたシナリオ内で適用可能であれば、N 個の階層のオーダーを連鎖または派生できます。

多層オーダーの作成は、連鎖オーダー作成トランザクションに結合され、親オーダーでリリースが発生すると、指定された組織ペアの単一の連鎖オーダーに変換されます。多層オーダーの伝票タイプは、親オーダーの伝票タイプの構成時に指定できます。

### 連鎖オーダー

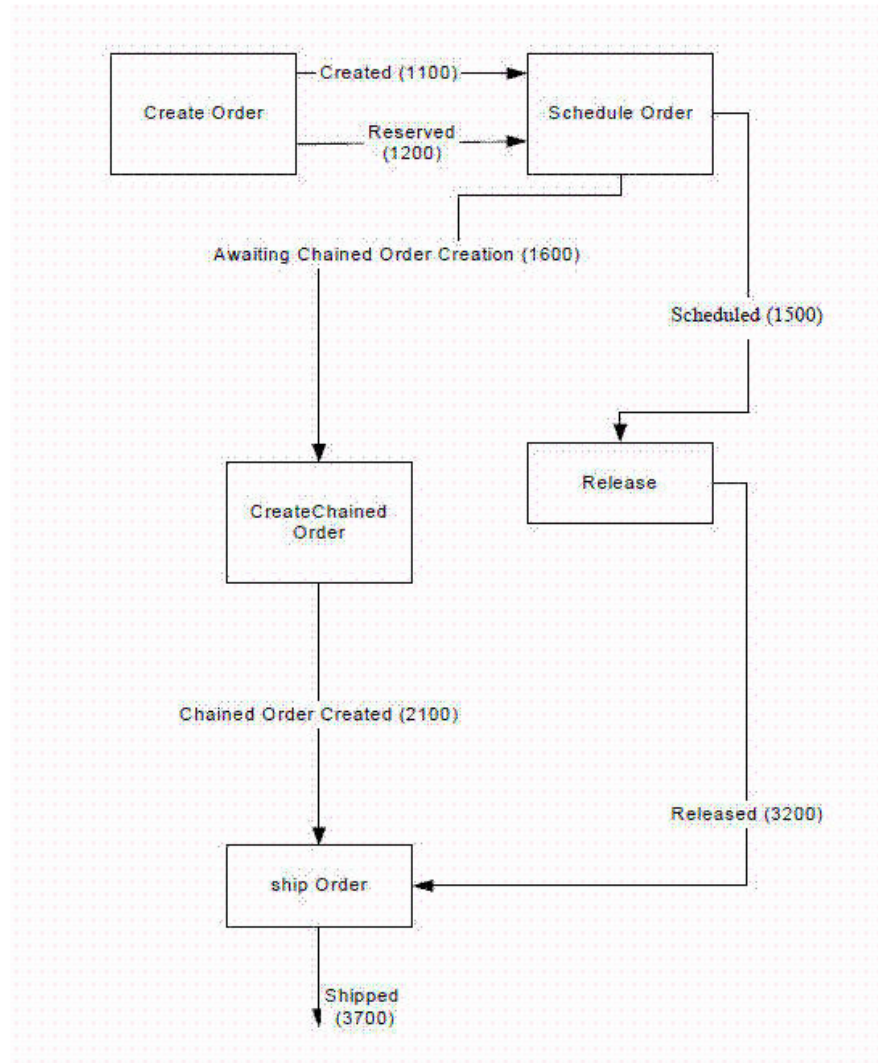
連鎖オーダー は、オーダー・フルフィルメント実行の一部を親オーダーからサード・パーティーに伝達する必要性が生じた結果、作成される従属オーダーです。従属オーダーがフルフィルメント・プロセスを完了しない限りその親オーダーが履行済みと見なされない場合、階層型オーダーは「連鎖されている」として定義されます。例えば、エンタープライズでオーダーが出され、一方でエンタープライズは、顧客に対してオーダーを直接履行させるために、ドロップ出荷サプライヤーで購入オーダーを作成します。親オーダーに「連鎖されている」オーダーは、ステータス更新を親オーダーに報告しなければなりません。連鎖オーダーの作成後に行われた親オーダーに対するあらゆる変更は、各連鎖オーダーに伝搬されます。

連鎖オーダーの場合、親オーダーのセラーは、バイヤーとしても構成される必要があります。

基本的に、手動またはスケジューリング・プロセスによって、特定の明細をドロップ出荷サプライヤーからソーシングすると決定された場合、その明細は連鎖購入オーダーに含められ、そのオーダーは独立して構成されたパイプラインを介して履行することができます。このプロセスによって作成される 2 番目のオーダーを連鎖オーダーと呼びます。

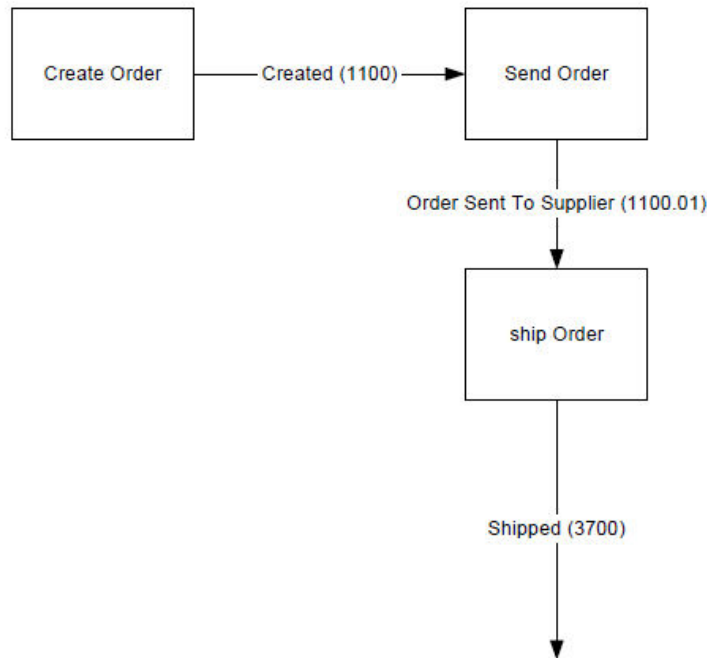
## 連鎖オーダーを作成するオーダー・ワークフロー

下図は、元のオーダーの大まかなワークフローを示しています。



## 連鎖オーダーのパイプライン

連鎖オーダーのパイプラインは、以下の図で説明する方法で構築できます。



## サービス用の連鎖オーダーの作成

連鎖オーダーは、IncludeInChainedOrder トランザクションを使用して、サービス明細についても作成することができます。連鎖オーダーのサービス明細は、提供サービスまたは配送サービスのいずれかになります。スタンドアロンの明細も可能であり、製品に関連付けることもできます。サービス明細が連鎖されている場合、関連付けられた明細も、明細間の関連付けも連鎖オーダーに持ち越されません。サービス明細を部分的に連鎖することはできません。

連鎖サービス明細は、キャンセルできます。作業オーダーに含めることや、完了することもできます。連鎖サービス明細が完了すると、親サービス明細にも完了を示すマークが付けられます。完了したサービス明細は、製品明細に自動的に完了を示すマークを付けることはありません。連鎖サービス明細が作業オーダーの一部として完了した場合、配達オーダー・トランザクションによって、連鎖明細と親明細の両方に完了を示すマークが付けられます。サービス明細は、作業オーダーまたは連鎖オーダーのいずれかによって、完了することができます。サービス明細が作業オーダーに含まれる場合、連鎖オーダーは作成できません。また、その逆の場合も同じです。

DELIVER\_ORDER トランザクションは、作業オーダーを保有している場合、連鎖オーダーのパイプラインの親とパイプラインの両方に含める必要があります。含まれていない場合、連鎖オーダーの作業オーダーを確認するときに、「数量が不足しています」を示すエラーを受け取ります。

## 派生したオーダー (derived order)

派生したオーダー (*derived order*) とは、親オーダーから作成された従属オーダーです。ただし、いったん作成されると、維持するのは、親に対する 1 つの参照のみです。そのライフサイクルは、親から独立しており、親オーダーへの状況伝搬は提供されず、親オーダーに対する変更があっても更新されません。フルフィルメント・プロセスを終了しなくても親オーダーを実行済みと見なすことができる場合は、階層型オーダーは「派生 (Derived)」と定義されます。例えば、購入オーダーを計画済みオーダーから「派生」することができます。

一般に、親オーダーのライフサイクルが、派生したオーダー (*derived order*) が作成された (例えば、計画済みオーダーからフルフィルメント・オーダーを作成した) 時間だけすでに超えてしまっているか、あるいは何らかの並列処理のために完全に新規のオーダーを生成する (例えば、返品処分が行われる前に返品用の交換オーダー (*exchange order*) を作成) 必要があります。他の各種オーダー伝票への親オーダーの進行を追跡する目的でのみ、派生したオーダー (*derived order*) と親オーダーとの間でリンクが維持されます。

派生したオーダー (*derived order*) の作成には、2 つの抽象化トランザクション、つまり、「派生オーダーに含める (*Include in Derived Order*)」と「派生したオーダーの作成 (*Create Derived Order*)」が作成されることが必要です。該当の重要なトランザクションを作成するために、これらのトランザクションをトランザクション構成中にコピーすることができます。どちらのトランザクションも、複数のピックアップ・ステータスと単一のドロップ・ステータス (*drop status*) を持つことが可能です。

1 つのオーダーが、同じ数量から複数の派生したオーダー (*derived order*) を持つことはできますが、親オーダーのフルフィルメントを完了するために連鎖オーダーが作成されるため、親オーダーでの 1 単位は、1 つの連鎖オーダーにのみ属します。

---

## カスタム・オーダー

カスタム・オーダーは、カスタマイズ済み製品の供給をサポートします。Sterling Selling and Fulfillment Foundation は、以下の 2 つのタイプのカスタム・オーダーをサポートしています。

- オーダー用 のオーダー
- 顧客用 のオーダー

オーダー用 オーダーには、このオーダー用に一意的に作成されたアイテムが含まれています。該当するアイテムは、特にこのオーダー向けに未完成の素材から作成されます。該当アイテムについての既存の在庫はなく、このアイテムとまったく同じ別のアイテムがオーダーされる見込みはありません。

顧客用 オーダーは、バイヤーの要件に基づいて作成されます。このオーダーでは、バイヤーがこの特定のアイテムについて、複数のオーダーを発行する予定であると見なします。顧客用オーダーは、要件に合わせて構成されたアイテムを反復購入するバイヤー組織が使用できるように設計されています。



## 受注生産オーダー

受注生産 (MTO) オーダーは、そのオーダー用に作成または構成されるアイテムのオーダーです。例えば、コンピューター・ベンダーは、カスタム・コンピューター構成を販売することがあります。この場合バイヤーは、ハード・ディスク・サイズ、プロセッサ・タイプ、DVD プレイヤー、およびその他のハードウェアを選択して指定できます。コンピューター・ベンダーは、既存のコンポーネントからコンピューターを組み立て、バイヤーの要求した構成で出荷します。

受注生産オーダーの特定のアイテムの数量は、1 より大きくなる場合があります。例えば、シャツの製造業者がバイヤーの企業ロゴが刺繍された青いシャツ 2000 枚のオーダーを受けることがあります。これも、受注生産のオーダーになります。

オーダーが受注生産である場合、カタログ上には購入されるアイテムのコンポーネントの定義がありません。これは、作成されるアイテムがそのオーダー専用であることを反映しています。「動的物理キット (dynamic physical kit)」(DPK) のコードが設定される場合があります。

オーダーが作成されると、「動的物理キット (dynamic physical kit)」(DPK) のキット・コードが割り当てられ、そのアイテムを構成するコンポーネントがオーダー明細で指定されます。セグメントがオーダー明細キーに、セグメント・タイプは「MTO」に設定され、このオーダーが受注生産であることが示されます。

オーダーのスケジュール時は、個別明細のソーシング・ルールが使用され、スケジュールリングでは単一出荷ノードへの配置が試行されます。単一のノードを指定できない場合、アイテムはバックオーダーされます。オーダー・スケジュールについて詳しくは、151 ページの『第 9 章 オーダー約束とスケジュールリング』を参照してください。



図 28. 受注生産オーダー

受注生産オーダーの作成によって、作業オーダー伝票も生成されます。作業オーダー伝票は、カスタム・オーダーに対応するための一連のアクションの管理に使用されます。作業オーダーについて詳しくは、work orders, see 227 ページの『第 10 章 付加価値サービス』を参照してください。

## 特注生産オーダー

バイヤーのコンプライアンスをサポートするため、組織は付加価値サービス (VAS) をオファーできます。このサービスではバイヤーにアイテムのカスタマイズが提供されます。

例えば、携帯電話の製造メーカー、ABC が、2 社の携帯電話サービス・プロバイダーに携帯電話を提供しているとします。1 社目の携帯電話サービス・プロバイダー、EFG は、電話のそれぞれに EFG のロゴを貼り付けることを希望しています。2 社目の携帯電話サービス・プロバイダー、XYZ は、10 代向けの市場をターゲットとし、人気のある 10 代の歌手のロゴを電話に貼り付けることを希望しています。さらに、KLM は、電話の箱に特別なマーケティング資料を入れることを希望しています。これらの資料には、ゲーム、インスタント・メッセージ、音楽のレビュー・サービスなど、10 代の市場の興味をそそる特別なサービスが記載されています。

携帯電話製造メーカー、ABC は、これらの 2 社のバイヤー用に別個の在庫セグメントをセットアップすることによってこれらの要件に対応できます。類似のオブジェクトのオーダーが発生するという前提のない受注生産モデルとは異なり、この例では、携帯電話サービス・プロバイダーがカスタマイズされた携帯電話について複数のオーダーを発注できます。ABC は、携帯電話の在庫を構築できます。

これらの付加価値サービスをサポートするには、組織の構成時に Requires\_VAS\_Compliance フラグを設定することによって、特注生産オーダーがサポートされるようにバイヤー組織を構成します。これによって、バイヤーが独自の仕様で生産されるアイテムをオーダーできることが指定されます。

カタログでカスタマイズ可能なアイテムが指定されると、そのアイテムに「VAS」(付加価値サービス) の分類が割り当てられ、バイヤーの要件に応じてアイテムがカスタマイズされることが示されます。アイテムの定義では、コンプライアンス・サービスが指定されます。コンプライアンス・サービスでは、バイヤーの要件に合わせてアイテムを作成するために実行されるアクティビティーが定義されます。VAS として分類されたアイテムにキットを設定することはできません。

オーダー作成時に、セグメントはバイヤー組織、セグメント・タイプは「MTC」(特注生産) が設定され、このオーダーが特注生産オーダーであることが示されます。

特注生産オーダーの送信時に、そのオーダーに対して十分な在庫が存在する場合は、作業オーダーが作成されます。

作業オーダーの確認によって在庫変換が発生する場合があります。Sterling Selling and Fulfillment Foundation は、割り当ての考慮事項の構成を使用することによって、このような在庫変換を可視化します。

Sterling Selling and Fulfillment Foundation の作業オーダー割り当ては以下のように使用されます。

- 作業オーダーの作成時には、元の在庫 (消費された在庫) に対して需要が出され、新規在庫 (作成された在庫) に対する供給が増やされます。ただし、増加した供給は、手持ちの供給ではありません。これは将来入手可能になる指示的な供給です。また、増加した需要は、約束された需要ではありません。

- 作業オーダーが割り当てられると、出された需要は、約束された需要を示すように変更されます。供給も、着荷の可能性が増えたことを示すように変更されることがあります。これらの需要と供給は利用可能在庫の評価に使用できます。
- 作業オーダーを確認すると、元の在庫の供給は削除され、新規在庫の供給が作成されます。

作業オーダー実行について詳しくは、227 ページの『第 10 章 付加価値サービス』を参照してください。

---

## オーダー保留処理

Sterling Selling and Fulfillment Foundation では、1 件のオーダーに対して 1 つまたは複数の保留タイプを適用できます。保留タイプを使用すると、オーダーではその保留タイプに関連付けられたトランザクションを処理できなくなります。

保留には、次の 3 つの異なるステータスがあります。

- 作成済み。保留がオーダーに適用された直後で、まだ調査は行われていません。
- 拒否済み。管理者が保留を調査し、オーダーを処理するべきではないと決定されています。
- 解決済み。管理者が保留を調査し、オーダーの処理を行うべきだと決定されています。

オーダー保留タイプは、次の方法で適用できます。

- 手動: 不正チェックを行うため、オーダーに保留を適用するべきであると管理者が判断し、アプリケーション・コンソールから保留を適用します。特殊な保留タイプを構成して、特定グループまたは一連のグループのユーザーのみがオーダーにその保留を適用できるように構成することができます。
- ドラフト・オーダー (draft order) の作成時に自動適用: ドラフト・オーダー (draft order) が作成されるたびに、デフォルトで特定の保留タイプが適用されます。
- ドラフト・オーダー (draft order) の確認時またはオーダー作成時に自動適用: ドラフト・オーダー (draft order) の確認のたび、またはドラフト・ステータスを介さずにオーダー作成を実行するたびに、デフォルトで特定の保留タイプが適用されます。
- 別の保留タイプの解決時に自動適用: 特定の保留タイプが解決された場合に、別の保留タイプが自動的に起動されます。これは、他の保留タイプの解決時に適用される保留タイプに指定されます。
- 特定の変更タイプの発生時に自動適用: 保留タイプは、オーダー、オーダー明細、リリース、またはリリース明細のレベルで特定の変更タイプが発生した場合に自動的に適用されるように構成できます。例えば、新規支払タイプが追加された際に常に不正チェック保留を適用します。
- 条件付き: オーダーまたはオーダー明細レベルで特定の条件に合致した場合に保留を適用できます。
- オーダーに承認が必要な場合に自動適用: オーダーが承認を必要とする条件に合致した場合に、構成されている保留タイプが起動されます。

注: ドラフト・オーダー (draft order) の作成時は、オーダーに承認が必要な場合でも、オーダーに保留は適用されません。ただし、ドラフト・オーダー (draft

order)を確認し、そのオーダーが承認の要求に必要なとされる条件に合致する場合は、そのオーダーに保留が適用されます。

保留の適用方法に関係なく、オーダーに保留を適用するかどうかを決定する条件を指定できます。例えば、特定の支払ステータスのオーダーのみに保留を適用する必要がありますとします。保留タイプに対してこれを行うには、条件ビルダーを使用できます。条件ビルダーの使用について詳しくは、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

デフォルトでは、すべてのトランザクションで保留中のオーダーまたはオーダー明細の処理が許可されます。アプリケーション・マネージャーを使用して、特定の保留タイプが適用されたオーダーの処理を抑止するトランザクションを指定できます。保留時にオーダーの処理が行われないように構成できるトランザクションは、保留タイプ対応トランザクションと呼ばれ、特定のエンタープライズの伝票レベルで定義されます。抽象化トランザクションから派生していないカスタム・トランザクションは、全保留タイプ対応です。

保留中のオーダー、またはオーダー明細を処理し、保留を削除するため、作成済みステータスの保留用に1つ、拒否済みステータスの保留用に1つの、2つのトランザクションを構成する必要があります。保留は手動で解決でき、また、これを実行する権限を持つ一連のユーザー・グループを指定できます。アプリケーション・マネージャーでは、特定のエンタープライズの伝票タイプ・レベルでも同様に実行できます。さらに、構成ルールに基づいて、オーダーの変更時にオーダー承認保留を自動解決することもできます。

## オーダー明細の保留タイプ

オーダー上の単一の明細に保留を適用することもできます。機能的にオーダー明細の保留は、オーダー・レベルの保留と同様です。例えば、複数の明細が含まれるオーダーを作成され、明細ごとに不正チェックを実行する必要がありますとします。明細ごとに不正チェック保留を有効にすることによって、チェックに合格した明細がフルフィルメント・プロセスに進むことができます。

オーダー明細レベルの保留は、オーダー・レベルの保留と同じ方法で適用できますが、以下の追加があります。

- 明細を含むドラフト・オーダー (draft order) の作成時、またはドラフト・オーダー (draft order) への明細の追加時に自動適用。
- 標準オーダーへの明細追加時に自動適用。

## 保留効果レベル

オーダー・レベルの保留と同様、指定された保留の適用がオーダー明細にもたらす効果を定義できます。ただし、オーダー明細レベルの保留には、オーダー・レベルか明細レベルのいずれかでトランザクションの処理が実行されないようにする機能もあります。例えば、いずれかの明細の出荷ノードが変更された場合に、自動的にSHIP\_NODE\_HOLD保留を適用することができます。この保留は、オーダーの残りの明細の処理を可能にするため特定の明細のみを対象とするか、その変更によってオーダー全体を対象として、スケジュール・トランザクションが実行されないように構成できます。

以下のトランザクションは、オーダー・レベルでのみ実行されないようにすることができます。これらのトランザクションは、オーダーの全明細をまとめて処理します。

- 支払実行
- 支払集金
- オーダーを閉じる
- リリースの送信
- ドラフト・オーダー (draft order) の確認

## タスク・キューでの保留フラグの設定

タスク・キューの保留フラグは、次のシナリオに基づいてオーダー明細の保留に設定できます。

- タスク・キュー・ベースのトランザクションの対象となるオーダーの全明細に、そのトランザクションが実行されないようにする保留が設定されている場合、タスク・キューの保留フラグは「Y」に設定されます。
- オーダー・レベルのタスク・キュー・トランザクションが実行されないようにする保留が適用された明細がオーダーに含まれる場合、タスク・キューの保留フラグは「Y」に設定されます。

---

## マルチクライアント・オーダー

1 バージョンの Sterling Selling and Fulfillment Foundation 上に複数バージョンの PCA がインストールされている場合、いずれかの PCA バージョンで作成されたオーダーに他のバージョンからアクセスできます。

このため顧客は、特定セットのストアに新バージョンの PCA をロールアウトし、その他のストアではアップグレードの準備ができるまで旧バージョンのままにしておくことができます。

このシナリオでは、顧客サービス担当者 (CSR) (Customer Service Representative (CSR)) が IBM Sterling Call Center および IBM Sterling Store リリース 8.2 を使用してオーダーを作成し、別の CSR がリリース 9.1 を使用してそのオーダーのピックアップをスケジュールすることができます。バージョン間で一部の機能にアクセス制限がある場合は、複数のバージョン間で機能に制限があるか使用できないことを示すメッセージがユーザーに提供されます。

マルチクライアント・オーダー機能は、特定の変更ルールに従って自動的に起動されるオーダー・タグをセットアップすることによって有効化されます。オーダー作成時、または変更時に、これらのタグが自動的に設定されます。

オーダーにおけるタグ作成は、次の基準に基づいて行われます。

- 構成されたオーダー・ヘッダー条件を満たしている
- 構成されたオーダー明細条件を満たしている
- 特定の変更タイプが発生している
- 手動での設定が可能である

修飾タグの定義について詳しくは、*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイドを参照してください。オーダー・タグの構成については、*Sterling Selling and Fulfillment Foundation* 分散オーダー管理 構成ガイドを参照してください。

---

## オーダー取り込み中の在庫予約

ビジネス要件により、製品在庫が顧客に対して保証され、オーダーが受け入れられた時点で保留になることが義務付けられる場合が少なくありません。これは、顧客のために在庫を予約することによって実現されます。

予約とは、後で購入する意志を示す顧客のために在庫を保留するプロセスです。これにより、オーダー取り込み中に、顧客のために希望アイテムの在庫状況が確保されます。既存の在庫予約に対してオーダーを予約するか、あるいは目的のオーダーについて新たに予約を試みることができます。

既存の在庫予約を使用する場合、オーダーがシステムに入力される前に、指定のアイテムについてシステムで予約を作成する必要があります。この予約は、オーダー作成の時刻で指定されます。次に、オーダーにより保留中の在庫が消費されます。

指定のアイテムについて予約が存在しない場合、システムは在庫の確認を試み、それが使用可能であれば、目的のオーダー用に予約します。

類似の明細で、配達明細および提供されたサービス明細についてキャパシティー割り当てを実行できます (製品アイテムの在庫予約と類似)。詳細については、『オーダー取り込み中のキャパシティー割り当て』のセクションを参照してください。

**注:** *Sterling Selling and Fulfillment Foundation* では、時間依存在庫のためのオーダー予約は現在サポートしていません。

*Sterling Selling and Fulfillment Foundation* では、上述の予約プロセスに加えて、以下のことを実行することができます。

- ドラフト・オーダー (draft order) についての予約
- 調達ノードについての予約
- 予約のみについてのオーダー明細の作成

### ドラフト・オーダー (draft order) の予約

ドラフト・オーダー (draft order) とは、そのオーダー用にデータが作成済みであるが、システムで確認されていないオーダーです。これらは、オーダー・プロポーザルか、または顧客の側での購入意志を示すオーダーです。オーダーが確認されるまで、ドラフト・オーダー (draft order) に対してオーダー実行プロセスを実行することはできません。

*Sterling Selling and Fulfillment Foundation* は、予約を通じてドラフト・オーダー (draft order) について在庫を保留にする機能を備えています。システムで在庫が使用可能な場合、「ドラフト・オーダー予約済み」ステータスでドラフト・オーダー (draft order) を作成することができます。このステータスで作成されると、システム内で「予約」需要が作成されます。

## 調達ノードについての予約

オーダーは、指定の調達ノードを使用して作成できます。オーダー明細に対して調達ノードが定義されている場合、明細は、該当ノードから調達する（購入オーダーまたは転送オーダーを使用）必要があります。このシナリオでは、オーダー明細に対して定義されている出荷ノードは、調達ノードから在庫を受け入れて、オーダーを満たします。

オーダー明細が調達ノードと関連付けられており、予約が必要な場合は、在庫は調達ノードで予約する必要があります。このシナリオでは、出荷ノードと同様、調達ノードについて「予約」在庫需要が作成されます。出荷ノードでは供給も作成され、出荷ノードにおける調達ノードからの入庫供給を表します。オーダーがスケジュールされ、調達オーダーが作成されると、調達ノードについての超過分需要と出荷ノードにおける供給が削除されます。

## 予約のみについてのオーダー明細作成

在庫が目的のオーダーを満たすことができる場合にのみ、オーダー明細が考慮されることがあります。Sterling Selling and Fulfillment Foundation は、オーダー作成時にオーダー明細について予約が作成できる場合にのみオーダー明細が作成されるようにすることで、この機能をサポートします。オーダー作成後は、在庫チェックを必要とするオーダー明細に対する変更は、変更後の属性に対して在庫が使用可能である場合にのみ、許されます。オーダー明細に対する一部予約または複数予約が可能です。これにより、ノード、アイテム、または明細に関する日付に関係なく、明細に関連付けられている予約を保留することができます。予約は、オーダー明細の異なる複数の属性に対して許可されます。明細上のノードがノード 1 の場合、予約は、ノード 2 に対して受け入れられます。既存の明細予約についての属性は、数量に関する場合を除き、変更できません。可能なのは、数量を減らすか、あるいは予約を削除することのみです。（オンザフライで予約する場合は、数量を増やすこともできます）。

数量の追加、ノード変更などの明細レベル変更は、関連付けられている予約に影響を与えません。明細が完全にキャンセルされた場合にのみ、その予約が削除されます。いったん明細がリリースされると、追加の明細予約は許されません。ただし、リリース済みステータスより前であれば、明細がスケジュール済みの場合でも、予約を追加できます。

在庫の予約は、明細レベルでの予約以外、各予約により行われます。各予約は、明細そのものに異なる属性があるためです。しかし、明細属性と一致する予約済み数量と同じく、予約属性が明細属性と一致する場合は、明細上のステータスはさらに更新されます。このような場合、オーダー明細予約は明細そのものにロールアップされます。例えば、ノード 1 からの 10 単位のアイテム 1 の数量に対してオーダーが作成され、2 つの明細予約が渡されます。1 つはノード 1 からの 7 単位のアイテム 1 に対するもので、もう 1 つはノード 2 からの 3 単位のアイテム 1 に対するものです（おそらく調達はオプションです）。この場合、数量 7 の予約はオーダー明細属性と一致するため、予約はオーダー明細予約テーブルから削除され、オーダー明細の数量 7 は「予約済み」ステータスに移ります。残りの 3 数量はオープン・オーダー・ステータスです。これにより、何らかの再割り振りが発生します。オーダーによってブロックされる需要の最終ピクチャーは、ノード 1 で予約される 7（ステータスによって行われる）、ノード 2 で予約される 3（オーダー明細予約に

よって行われる)、およびノード 1 での 3 オープン・オーダー需要 (ステータスによって行われる) です。オーダー明細と一致しない予約は、予約されますが、オーダー明細にはロールアップされません。スケジュールは、後でこれらの予約を調べて、それぞれの有効期間を判断します。ただし、スケジュールが実行されるまで、需要はそのオーダー明細に対してブロックされます。

## 複数明細予約

特定のオーダー明細に対して、一部予約または複数予約を行うことができます。明細のノード、アイテム、または日付にかかわらず、その明細に関連付けられたすべての予約を保留する機能が提供されます。予約は、オーダー明細の異なる複数の属性に対して許可されます。例えば、明細のノードがノード 1 である場合に、ノード 2 に対する予約を実行できます。scheduleOrder API が実行される場合は常に、在庫が利用可能であると想定することによって、ノードにかかわらず予約の使用が試行されます。明細予約はすべて、スケジュールリングの後に削除されます。

オーダー明細予約属性が、出荷ノード、アイテム情報、日付などの明細属性と一致する場合は、常に明細ステータスによってオーダー明細予約が消費されます。このシナリオでは、明細に予約済みステータスが設定され、オーダー明細予約は作成されません。

## 予約パラメーター

さまざまなシナリオでエンタープライズが予約を処理できるよう、複数のパラメーターを定義できます。

### 要予約 (Reservation Required)

「要予約 (Reservation Required)」は、オーダーに対して在庫を予約できる場合に、予約作業が必要なことを示します。

### 予約が必須

「予約が必須」は、オーダー作成時またはオーダー変更時に、オーダー明細に対して予約できる在庫がなければならないことを指定します。オーダー明細に対して在庫が使用可能でない場合、システムにオーダー明細は作成されません。

### 明細変更時に予約の検証を抑制 (Suppress Validation of Reservation on Line Change)

「明細変更時に予約の検証を抑制 (Suppress Validation of Reservation on Line Change)」を使用すると、在庫チェックなしに予約オーダーに変更を加えることができます。ただし、このパラメーターが使用されるのは、予約が必須でない場合のみです。

### 正確な日付の予約を使用 (Use Reservation With Exact Date)

Sterling Selling and Fulfillment Foundation で、特定の日付の予約を作成できます。在庫はこの日からシステム内で予約されます。このパラメーターを使用可能にした場合、予約日がオーダーの出荷指定日と同じである、既存の予約のみを使用できます。



ただし、このパラメーターが使用可能でない場合、予約日がオーダー希望日以前の予約もすべてオーダーの予約に使用できます。

---

## オーダー取得時のキャパシティの割り当て

Sterling Selling and Fulfillment Foundation には、オーダー取得時に、提供サービスおよび配送サービスの明細にキャパシティを割り当てる（またはリソース利用可能性を約束する）機能があります。これは、オーダー明細作成時に、提供サービスおよび配送サービスを実行するキャパシティをチェックすることによって、実行されます。使用可能なキャパシティがある場合、明細の作成時にキャパシティを割り当てる（または約束する）ことができます。これで、このキャパシティは、他のオーダー明細では使用不可とみなされます。サービス明細の変更の場合も、使用可能なキャパシティがあるか、チェックできます。以下の主要機能は、提供サービスおよび配送サービスの明細のキャパシティ割り当てを処理するために、Sterling Selling and Fulfillment Foundation によって提供されています。

- オーダー明細作成時にキャパシティが使用可能かどうかチェックし、使用可能な場合は、キャパシティを割り当てる（ブロックする）ことが可能。
- 明細に有効な予定を提供することによって、オーダー明細作成後に、キャパシティを割り当てることが可能。
- 明細の出荷または作業オーダーの完了前ならいつでも、提供サービス明細の約束された予定を変更することが可能。

**注:** 提供サービス明細または配達明細のステータスには、その明細にキャパシティが割り当て済みかどうかは示されません。予定ステータスには、サービス明細にキャパシティが割り当て済みかどうか示されます。

## キャパシティの割り当てのパラメーター

Sterling Selling and Fulfillment Foundation では、キャパシティの割り当ての取り扱いのために、以下のパラメーターを提供しています。

キャパシティの割り当てを試行する前に、有効な予定がオーダー明細上に存在する必要があります。予定が使用不可の場合、キャパシティを割り当てる試行は、明細の作成後に行われます。

キャパシティが使用不可の場合、作成時には、予定は取得されますが、キャパシティは割り当てられません。

以下のようなオーダー明細の変更を行うと、キャパシティの使用可能性が再度考慮されるようになります。

- オーダー数量の追加またはオプションの追加
- 出荷先住所
- 明細フルフィルメント用に選択された出荷ノード。
- サービスを実行する必要がある予定日時。
- 既存の配送サービスまたは提供サービスに別の製品明細を追加することによる、スキル変更。

## キャパシティー・チェックのオーバーライド

各サービス明細について、明細作成時、または以降に何らかの変更があった場合に、キャパシティーをチェックするかどうかを示すパラメーターを指定できます。このパラメーターでキャパシティー・チェックがオーバーライドされるように指定されている場合、そのサービス明細でキャパシティーが使用可能であることが前提となり、チェックは行われません。キャパシティーは、明細で指定された予定日に割り当てられます。ただし、必要な配送サービスまたは提供サービスを提供できるように、有効なリソース・プールがセットアップされている必要があります。

## キャパシティー予約

キャパシティー予約によって、販売オーダーの作成前に、キャパシティーをブロックできます。オーダー作成時に、キャパシティー予約 ID を渡すことができ、ブロックされたキャパシティーは作成されたオーダーによって消費されます。Sterling Selling and Fulfillment Foundation は、キャパシティー・チェックを使用して、使用可能なキャパシティーに対して確実に予約が行われるようにします。

キャパシティー予約は、指定されたスロットで、キャパシティーが完全に使用可能な場合にのみ作成されます。例えば、リソース・プールで 9-12 および 1-5 の 2 つのスロットが定義されており、各スロットで使用可能なキャパシティーが 4 時間とします。

予約 ID	スロット	数量 (Quantity)	結果
Res1	9-12	3	予約が作成されます。
Res2	1-4	1	スロットが定義されていないため、予約は作成されません。
Res3	12-6	1	スロットが定義されていないため、予約は作成されません。
Res4	1-5	7	使用可能なキャパシティーが十分でないため、予約は作成されません。

キャパシティー予約は、リソース・プールおよびリソースの両方でサポートされています。リソースについては、キャパシティーがリソースに対してブロックされている場合、実際にブロックされるキャパシティーは、リソースのカレンダーに基づきます。

スロット・ベースの予約およびスロット・ベースでない予約を作成できます。リソース・レベルでキャパシティーを管理していないリソース・プールに対してスロット・ベースの予約が行われた場合、キャパシティー・チェックを実行して、使用可能なキャパシティーがあるかチェックできます。

キャパシティー予約の作成時に、有効期限 (expiration date) を渡すことができます。これによって、予約の有効期限 (expiration date) が切れると、予約をバージョンアップすることができるようになります。これで予約済みのキャパシティーは解放され、新しいオーダーに割り当てまたは予約することができます。予約キャパシティーおよび有効期限 (expiration date) は、変更可能です (キャパシティーが使用可能であることが前提)。予約は手動で削除することもできます。

---

## アイテム検証

アイテム検証により、エンタープライズは、オーダーの製品アイテムがエンタープライズのカタログに属しているかどうかを検証することができます。

デフォルトでは、アイテム検証はオフになっています。

### 拡張検証

拡張検証により、エンタープライズはオーダーに対して以下の追加アイテム検証を実行できるようになります。

- 有効日付範囲
- 個別販売不可
- 最小数量および最大数量
- アイテム・ステータス
- 顧客資格 (customer entitlement)

システムが検証を拡張する前に、アイテム検証を使用可能にしておく必要があります。

デフォルトでは、拡張検証は使用可能になっています。

`validateItemForOrder` API を呼び出すことによっても、オーダーを検証できます。

### 有効日付範囲

オーダー内のアイテムの有効日付範囲は、そのオーダーが作成、変更、または確認される時点で検証されます。システム日付が有効な日付範囲外の場合、システムは、アイテムの検証中にエラーをスローします。しかし、そのアイテムについて取り替えが有効になっている場合は、検証は無視されます。

### 個別販売不可

検証により、オーダー内の該当アイテムがバンドルまたは物理キットの一部であることが裏付けられます。検証は、オーダーが作成、変更、または確認されるときに実行されます。

### 最小数量および最大数量

オーダー内のアイテムの最小数量および最大数量は、オーダー確認時に検証されます。1 つのオーダー内の 1 つのアイテムの数量がそのアイテムの最小数量と最大数量の範囲外にある場合は、システムはエラーをスローします。

### アイテム・ステータス

検証により、該当アイテムのステータスはオーダーの作成および変更時にオーダー可能であることが裏付けられます。アイテムのステータスが「公開済み」でない場合は、システムはエラーをスローします。アイテム・ステータスについて詳しくは、「*Sterling Selling and Fulfillment Foundation: Catalog Management Configuration Guide*」を参照してください。

## 顧客資格 (customer entitlement)

オーダーが作成、変更、または確認される時点で、オーダー内のアイテムについての顧客資格 (customer entitlement) が検証されます。顧客が、資格のないアイテムをオーダーしようとする、システムはアイテム検証中にエラーをスローします。顧客資格 (customer entitlement) について詳しくは、「*Sterling Selling and Fulfillment Foundation* カタログ管理概念ガイド」を参照してください。

注: アイテムが非公開であるか、期限切れであるか、あるいは個別販売できない場合は、現行日付が関連する有効な日付範囲内にある製品関連が存在する限り、そのアイテムはまだ製品カタログに掲載されます。しかし、そのアイテムは、オーダーには使用できません。

以下の基準の 1 つを満足する場合は、アイテムは無効になります。

- 現在日付がアイテムの有効日付範囲外である。
- 製品が個別販売不可である (IsSoldSeparately="N")。
- アイテムが公開されていない (ステータスが 3000 に等しくない)。

---

## 確認済みオーダーの変更保留

Sterling Selling and Fulfillment Foundation には、オーダーが変更が許可されるステータスである場合に、ユーザーが確認済みオーダーを変更できる機能があります。変更を保持するには、ユーザーが変更内容を実際に保存する必要があります。ただし、Sterling Selling and Fulfillment Foundation には変更をロールバックする機能もあります。確認済みオーダーに加えられた変更は、ユーザーが変更を保存またはロールバックするまで保留状態で保持されるか、事前構成された時間フレームの経過後に変更が自動的に有効期限切れになります。

例えば、ユーザーは確認済みオーダーにアイテムを追加して、そのオーダーをモデル化できます。ユーザーが変更内容を保存すれば、その変更が永続的にオーダーに適用されます。しかし、オーダーのモデル化後に、アイテムの価格が高すぎると、ユーザーが判断したとします。確認済みオーダーには追加されたアイテムが表示されますが、ユーザーが保留中の変更を保存しないでブラウザーを閉じたとします。

Sterling Distributed Order Management Applications Manager で定義された指定時間フレームの経過後に、保留中の変更は自動的に期限切れになり、オーダーは元の状態にロールバックされます。

確認済みオーダーに保留中の変更が含まれる場合、自動的に保留がオーダーに設定され、その後のオーダー処理ができなくなります。保留中の変更が保存またはロールバックされると、保留が自動的に削除され、オーダー処理を続行できます。保留中の変更が期限切れになった場合、ProcessOrderHoldType エージェントが起動され、保留が自動的に削除されるまで、オーダーが保留されたままになります。

確認済みオーダーの変更保留機能は、ユーザーによるオーダー変更、オーダーのモデル化、およびオーダーの確認が可能なドラフト・オーダー (draft order) に似ています。しかし、ドラフト・オーダー (draft order) には保留が設定されず、通常の場合ドラフト・オーダー (draft order) はユーザーがブラウザーを閉じた後も期限切れにならない点が異なります。ユーザーが再度ログインすると、一般にドラフト・オーダー (draft order) は変更および確認が可能です。

---

## 第 8 章 案件管理

---

### 案件管理とは

案件とは、1 つのセールス案件と関連付けられた 1 つ以上の見積のコンテナ (container) です。例えば、外回りの営業担当員に、特定の顧客にラップトップ・コンピューターとデスクトップ・コンピューターを販売する案件があります。この外回りの営業担当員は、顧客に提示する 1 つ以上の見積を生成します。以下に例を示します。

- 見積 A: 総コスト \$4,000.00 で 5 台のラップトップ・コンピューターと 5 台のデスクトップ・コンピューターを購入。
- 見積 B: 総コスト \$3,500.00 で 4 台のラップトップ・コンピューターと 6 台のデスクトップ・コンピューターを購入。
- 見積 C: 総コスト \$3,000.00 で 2 台のラップトップ・コンピューターと 8 台のデスクトップ・コンピューターを購入。

顧客は、これらの見積のどの条件についても、外回りの営業担当員と交渉できます。ただし、顧客が受け入れることができる見積は 1 つのみであり、その結果、この販売案件と関連付けられた他の見積はすべて破棄されます。顧客がある見積を受け入れると、その見積からオーダーが作成され、すべての関連見積は破棄され、案件が成立します。顧客が見積のいずれに関しても交渉しないか、または見積の有効期限が切れた場合、見積は破棄され、案件は失注になります。

案件の作成および管理について詳しくは、「*Sterling Selling and Fulfillment Foundation* 分散オーダー管理 (Quote セットアップ)」を参照してください。



---

## 第 9 章 オーダー約束とスケジューリング

---

### オーダー約束の概要

サプライ・チェーン業界において、オーダーによって要請される製品またはサービスは、在庫状況のチェック、約束、出荷または配達に合わせたスケジューリング、その後のリリース（出荷済みまたは配達済み）が必要です。

Sterling Selling and Fulfillment Foundationは、そのオーダー約束機能の一部として、製品とサービスの両方についてノードの選択および出荷日の制御に使用できる構成可能なルールを提供します。これらのオーダー約束フィーチャーの能力を理解する上で役立つように、本章では、約束機能と関連するSterling Selling and Fulfillment Foundationの概念を説明し、お客様のビジネス目的を実現するために必要なセットアップに関する情報を提供します。

オーダー約束機能は、サービス要請（配送サービスおよび提供サービス）と同様に、出荷中の製品についても提供されます。これらの特定の約束フィーチャーについて詳しくは、153 ページの『出荷する製品の約束』または 201 ページの『サービス要請の約束』を参照してください。

このすべてが Sterling Selling and Fulfillment Foundation でどのように実現されるかを理解するためには、まず最初に、オーダー約束と関連付けられている以下の共通用語に精通している必要があります。

#### ノード

ノードとは、製品の出荷、返品、または配達の宛先または元となる物理的ロケーションです。

ノードがオーダー明細に指定されると、そのノードからのオーダー明細を満たすことを示します。ノードが確定事前定義済みノードの場合、オーダー約束機能は、オーダー明細が確実にこのノードからのみ満たされるようにします。

ただし、特定のビジネス要件では、オーダーが作成されたら即時に在庫をそのオーダー用に予約することが必要な場合があります。この在庫予約は、在庫が予約済みのオーダー明細に出荷ノードを提供することによって達成されます。予約用に指定されたノードが、目的のオーダーを満たすのに最適のノードでない場合があります。このシナリオでは、該当ノードを非確定ノードとしてマーク付けできます。オーダーを約束する際に、Sterling Selling and Fulfillment Foundation スケジューリング機能は、ソーシング・セットアップに基づいて、在庫が使用可能な代替ノードを見つけようと試みます。

デフォルトでは、Sterling Selling and Fulfillment Foundation は、別段の記載がない限り、オーダー明細に提供された各ノードを確定ノードとして扱います。

## 分配グループ (distribution group)

分配グループ (distribution group) (以前のリリースでは分配ルール (distribution rule) と呼んでいました) は、製品またはサービスを分配するために定義された一連のノードまたは組織です。

## 在庫状況の照会

ほとんどのシステムでは、在庫状況照会機能とサービスの可用性機能は別々に提供されます。通常、製品の在庫状況を知るためには、在庫モジュール内で照会する必要があります。別個のキャパシティー・モジュールは、サービスに使用可能であれば、キャパシティーが対応可能かどうかについて照会対象にすることができます。より複雑な照会機能が必要で、出荷先ロケーションや、同梱出荷が必要であるといった制約のある複数製品の在庫状況を考慮しなければならなかった場合は、システム上で見積もりを作成してからでないと、在庫状況を照会できません。

しかし、Sterling Selling and Fulfillment Foundationでは、別のアプローチを採用しており、システム上で存続ドキュメントを作成しなくても単一の照会機能で複数の製品「および」サービスが使用可能であるかどうかを照会できる API を提供します。この API は、可能なソーシング・オプションを提案する際に、複数の明細間のすべての相互依存関係を考慮します。システムでオーダーを作成してある場合は、同じ照会機能も使用できます。

## 納期回答ルール (Available to Promise Rules)

納期回答 (ATP) ルール (Available to Promise (ATP) rules) は、アイテムの在庫状況を追跡したり、在庫が指定最小レベルを下回った場合に特定のアクションを実行したりするモニター・システムをセットアップするのに役立ちます。アイテムの在庫状況は、現在日付、ATP 時間フレーム内の後続日数、および ATP 時間フレーム外の後続日数に基づいて追跡できます。これにより、現在および将来の需要を満たすために、より正確に供給をオーダーできます。

## スケジューリング

スケジューリングは、以下を行うプロセスです。

- 製品フルフィルメントのための出荷ノードまたはサプライヤー、あるいはサービス・フルフィルメントのためのサービス・プロバイダーを決定する。この論理プロセス・ステップを「ソーシング」と呼びます。
- 製品またはサービスが出荷または配達される日付を決定する。
- 出荷ロケーションにある在庫を出荷日用に予約する。ただし、在庫が製品明細用にすでに予約済みの場合は、スケジューリングでは在庫の再チェックを試みません。

スケジューリング機能ではオーダー伝票が必要であることに注意してください。

## スケジューリング・ルール

スケジューリング・ルールは、以下のような一般的なスケジュール・パラメーターを制御します。

- オーダーをスケジュールする必要がある場合



- 製品の在庫状況チェックに先立つ日数 (将来の在庫 (future inventory) に対して約束する場合)
- スケジューリングに使用する必要のある最適化のタイプ

例えば、Sterling Selling and Fulfillment Foundation が、最短日付で配達可能な出荷ノードを選択するように、日付に基づいて最適化することができます。また、出荷数に基づいて最適化することもできます。つまり、Sterling Selling and Fulfillment Foundation は、最後にはそれより後の日付で配達される可能性がある場合でも、必要な出荷の総数を最小化します。

- 出荷完了オーダーまたは出荷完了明細のパラメーター

## ソーシング・ルール

ソーシング・ルールは、製品、アイテム分類、出荷先地域、およびその他のパラメーターに基づいて製品をソーシングするために考慮すべきノード、外部組織、またはノードのグループを制御します。

## 通知

通知とは、オーダー要請を満たす際に出荷ノード、ベンダー、またはサービス・プロバイダーに通知するプロセスです。Sterling Selling and Fulfillment Foundation は、この通知プロセスについていくつかの異なるドキュメントを作成します。

## 配達ノード

配達ノードとは、最終マイル・サービスを使用して製品が配達されるロケーションです。

---

## 出荷する製品の約束

約束機能の一部として、Sterling Selling and Fulfillment Foundation には次の機能があります。

- 製品の在庫状況の照会。製品の在庫状況の照会機能は、約束機能に関連付けられたすべてのビジネス・ルール (business rules) を考慮した上で 1 つ以上のソーシング・オプションおよび可能な出荷日を提示します。ユーザー・インターフェースを使用し、特定のオーダーについて製品の在庫状況を照会することができます。既存のオーダーを必要としない約束 API (`findInventory ()`) を使用して、アイテムのセットの製品の在庫状況を照会することができます。
- オーダーまたはオーダー明細のスケジューリング。Sterling Selling and Fulfillment Foundation には、適時に出荷できるようにオーダーをスケジュールする API およびトランザクションが用意されています。スケジューリング機能は、ノードおよび予定出荷日を設定し、これと同じになるように出荷をスケジュールします。さらにスケジューリング機能は、その予定出荷日のために出荷ノードで在庫を確保します。
- ノードまたはサプライヤーに適時に通知するための通知および連鎖ドキュメントの作成。

## 製品の在庫状況の検索

このセクションでは、製品の在庫状況の検索と関連する概念について説明します。

## 納期回答ルール (Available to Promise Rules)

納期回答 (ATP) ルール (Available to Promise (ATP) rules) により、現在および将来の需要のためのアイテムの在庫状況を判別できます。この判別によって在庫を最も有効に利用でき、より差し迫った需要を満たすためにアイテムを使用できる場合に、それを将来のオーダーのために取り置いておくことはなくなります。アイテムの在庫状況は、現在と将来の供給、リード・タイム、および ATP 構成に基づいています。リード・タイムは、サプライヤー (配送センターまたはドロップ出荷サプライヤー) が出荷用のアイテムを購入するためにかかる時間です。処理時間は、サプライヤーがアイテムを受け入れ、出荷 (入庫処理) および倉庫からの出荷 (出庫処理) の準備ができるまでにかかる時間を範囲とします。

ATP ルールにより、アイテムのオーダーを効率的に管理できます。現在および将来のオーダーにアイテムが使用できる時間についてのパラメーターを設定できます。期限切れ順先出し (FEFO) 在庫管理システムを用いることで、傷みやすいアイテムを、その有効期限 (expiration date) が切れる十分前に販売および出荷することができます。先に期限切れとなる在庫を必ず先に消費することができます。

## 在庫状況の安全要因

在庫状況の安全要因は、さまざまな目的で在庫状況から除外される固定数量またはパーセントを定義します。在庫状況の安全要因を適用するかどうかは、スケジューリング・ルール、供給タイプ、およびノード・タイプで制御されます。

## 約束手持ち在庫

在庫状況がオンライン消費者と店内顧客で共有されているシナリオで店内在庫状況を扱う場合、アイテムが置き間違えられているか、アイテムがショッピング・カートに入っている、あるいはアイテムが予約されているために、いずれかの指定時刻での製品の実際の在庫状況が正確でない可能性があります。これは、オンライン消費者にとって過剰約束手につながる可能性があります。

在庫状況の安全要因を使用して、店は、オンライン消費者に表示される在庫状況から除外される在庫の数量またはパーセントを、アイテムまたは在庫供給タイプごとに定義することができます。

## 約束将来在庫

将来在庫とは、サプライヤーに到着する予定であるが、まだ手元にない供給です。購入オーダーにより生成される将来在庫は、エンタープライズが新規オーダーおよび既存オーダーに対して約束するのに使用できます。ただし、場合によっては、すべての将来在庫を完全に信頼できると見なせないことがあります。例えば、エンタープライズがそのサプライヤーの 1 つに購入オーダーを発行したばかりの場合、そのオーダーがバックオーダー済みになる可能性があります。したがって、エンタープライズは、購入オーダーによって作成される将来供給の 40% のみをオーダー約束用の対象にする可能性があります。この供給が出荷済みで、輸送中であれば、エンタープライズは、供給をタイムリーに受け取れることをより確信することができます。したがって、エンタープライズは、輸送中供給の最大 95% を安全に約束することができます。

Sterling Selling and Fulfillment Foundation では、この柔軟性が提供されます。エンタープライズは、需要を約束するのに使用できる各供給タイプのパーセントを定義することができます。

## 計画外の在庫

計画外の在庫とは、アイテムのリード・タイムに基づいて使用可能と仮定されている在庫です。例えば、10 日のリード・タイムがあるアイテム 1 の 10 数量についてオーダー照会があり、アイテム 1 の 7 数量のみが手持ち在庫にあります。計画外の在庫を考慮すると、オーダーは、以下の在庫に対してスケジュールすることができます。

手持ち在庫 = 7 数量については現在日付で使用可能

計画外の在庫 = 3 数量については現在日付 + 10 日 (リード・タイム) で使用可能

## 最小出荷期限

時間の影響を受けるアイテムを扱う場合、顧客によって好まれる製品出荷後の有効期間は異なります。例えば、正規小売店は残余有効期間が 240 日のアイテムを要求しますが、ディスカウント小売店は経費節減のため同じアイテムでも 120 日の有効期間のものを要求したりします。

最小出荷期限は、オーダー作成時にオーダー明細で指定できます。オーダー作成プロセスで、時間の影響を受けるアイテムの出荷後の必要な残余有効期間が考慮されます。最小出荷期限は、製品の要求残余寿命に追加される必要な出荷期限です。

時間の影響を受けるアイテムの約束、およびスケジュール・プロセスの際に、Sterling Selling and Fulfillment Foundation では、オーダー明細の最小出荷期限が在庫供給の出荷期限 (有効期限 (expiration date)) 以前にある場合にのみ利用可能数量が約束されることが保証されます。

例えば、次の手持ち在庫が利用可能であるとします。

供給	商品 (Product)	有効期限 (expiration date)	数量 (Quantity)
供給 1	アイテム 1	7/15/2005	5
供給 2	アイテム 1	7/30/2005	10

次のオーダーが発生したとします。

商品 (Product)	数量 (Quantity)	需要出荷日	最小出荷期限
アイテム 1	5	7/1/2005	7/21/2005

この例は、このオーダーを充足するには、出荷時点において約束したアイテム 1 の数量に、20 日の残余有効期限が必要であることを意味します。

供給 1 は、出荷後 14 日で有効期限が切れるため、この在庫は約束に算入されません。一方、供給 2 は、出荷後 29 日で有効期限が切れるため、指定された基準に合致します。

## ソーシング

ソーシングは、製品の出荷元のノードまたはサプライヤーを決定するプロセスです。

配送センター (distribution center) が 1 つしかないという最も単純なシナリオでは、出荷ノードは既にわかっています。このような場合、出荷ノードを照会機能で事前に指定するか、オーダー明細で指定できます。このノードが固定の事前定義ノードの場合、オーダー明細は、その出荷ノードからのみ調達されます。この場合、ソーシング構成を行う必要はありません。

ただし、ノードが固定の事前定義ノードではない場合には、残りのソーシング設定が使用されます。明細で指定されたノードが固定ノードでない場合や、もっと複雑な状況では、製品の調達元のノードおよびサプライヤーが複数存在することがあります。

正しい出荷ロケーションの選択の要件は、以下の事柄に基づきます。

- 出荷される製品
- 出荷先ロケーション - 通常、最も近い出荷ロケーションから出荷しますが、地政学的要件を満たしていることを確認する必要があります。例えば、テキサス州のエルパソに出荷する場合、メキシコの配送センター (distribution center)の方がエルパソに近くても、米国内にある配送センター (distribution center) から製品を出荷する必要があります。
- さまざまなロケーションでの製品の在庫状況。指定のアイテムの在庫が倉庫にある場合とない場合があります。しかし、製造業者は、ノードにある原材料を使用して、需要を満たすことができることを常に想定している必要があります。これを実現するために、いずれのノードでも、特定のアイテムのスケジュールおよびリリースに関する在庫チェックを実行しないように Sterling Selling and Fulfillment Foundation を構成できます。この機能は、アイテム・レベルで使用可能にできます。
- 要求を完了するために必要な総出荷数。これは、何らかの方法で、出荷全体に関連付けられた輸送コストおよび取り扱いコストを反映します。3 アイテムのオーダーが行われ、あるノードには 3 つすべてがあり、別のノードには 2 つしかない場合、2 回出荷するのではなく、3 アイテムすべてがあるノードから一度だけ出荷する必要があります。
- ノードの優先順位。まず独自のロケーションから出荷し、次に独自のノードに製品がない場合に製造元サプライヤーを使用する必要があります。
- 一括出荷依存や選択したノードから補充数を満たすなどの顧客指定の制約。
- 作業オーダーのサービスを実行する機能。これには、キット化サービスとキット化解除サービス、およびバイヤー準拠のサービスを提供する機能が含まれます。
- 贈答品用の包装サービスを行う機能。出荷ノードは、アイテムのフルフィルメント・オプションとして贈答品用の包装を提供する場合と提供しない場合があります。この機能は、ノード属性レベルで使用可能にできます。
- カスタマイズ可能なオーダー・ソーシング分類。例えば、エンタープライズは、顧客属性をソーシング・パラメーターとして使用する必要がある場合があります。オーダー・ソーシング分類は、この柔軟性を可能にします。

以下の各セクションでは、これらの要件を満たすのに必要な構成について説明します。ソーシングおよび日付を決定する際に、Sterling Selling and Fulfillment Foundation は、以下の構成設定を使用します。

- オーダーのエンタープライズ。このルールの例外は、個別のケースで注記します。
- 処理中の照会 API で渡される組織コードの主要エンタープライズ。

**ソーシング構成:** 製品のソーシングを決定するとき、以下のパラメーターをどのように構成するかについて検討します。これは、暫定的な製品ソーシング構成の定義に役立ちます。

1. 「ソーシング・ルールは定義されますか (Are sourcing rules defined?)」

ソーシング・ルールは、エンタープライズによって定義されます。このパラメーターを使用して、エンタープライズがソーシング・ルールを定義するかどうかを指定できます。エンタープライズが所有している配送センター (distribution center) からのみ出荷し、エンタープライズの配送センター (distribution center) が数カ所の場合、複雑なソーシング設定を行う必要はありません。このパラメーターを「いいえ (No)」に設定すると、Sterling Selling and Fulfillment Foundation は、エンタープライズが所有するノードのいずれかから製品を調達します。最適化ロジックを使用して、距離などのパラメーターに基づいて、適切なノードを選択します。

**注:** この機能は、前のリリース (5.0 以前) から変更されました。以前のリリースでは、Sterling Selling and Fulfillment Foundation は、オーダーのセラー組織が定義した設定を検索していました。この機能には下位互換性はありません。ソーシング・ルールを再構成して、必要な機能を設定する必要があります。

2. 「在庫情報はシステムで使用可能ですか (Is inventory information available to the system?)」

このパラメーターは、在庫を所有できるすべての組織に対する設定です。このパラメーターは、組織が在庫情報を Sterling Selling and Fulfillment Foundation で使用できるようにするかどうかを指定します。情報は、Sterling Global Inventory Visibility アプリケーションまたはリアルタイム・インターフェースを使用して使用可能にできます。

在庫情報をシステムで使用可能にできる場合、このパラメーターを「はい」に設定します。こうすると、通常、ソーシングの決定が向上します。この情報を使用できない場合、これを「いいえ (No)」に設定する必要があります。

また、このパラメーターの他に、Sterling Selling and Fulfillment Foundation は、以下の場合のみ在庫状況を制約の 1 つとして使用することに注意してください。

- オーダーのセラー組織が在庫情報を発行する場合 (このパラメーターが「はい」に設定されている場合)
- オーダーの伝票種別 (document type) が「スケジューリングのための在庫の読み込み (read inventory for scheduling)」に設定されている場合
- オーダーの伝票種別 (document type) が「セラー組織の在庫の更新 (update inventory for seller organization)」に設定されている場合

上記のどれにも当てはまらない場合、Sterling Selling and Fulfillment Foundation は、在庫チェックを実行する必要がないと見なし、ソーシング・プロセスは、在庫を決定要因の 1 つとして見なしません。ただし、距離やノード優先順位などの他の要因は、出荷ノードの選択に関して考慮されます。

3. 「在庫が外部に保持されていますか (Is inventory kept externally?)」

このパラメーターは、上記のパラメーター「在庫情報はシステムで使用可能ですか (Is inventory information available to the system?)」が「はい」に設定されている場合のみ関係があります。このパラメーターは、組織の在庫情報を Sterling Global Inventory Visibility アプリケーション内で管理するか、またはリアルタイム・インターフェースを介して使用できるようにするかを指定します。

単一の分配グループ (distribution group) が、以下のものを指定できます。

- Sterling Global Inventory Visibility アプリケーション内で保持される在庫のノード。
- 在庫を外部に保持する組織

在庫をチェックするとき、Sterling Selling and Fulfillment Foundation は、このパラメーターを使用して、在庫ストアを判別し、外部で保持されている場合、コールアウトして製品の在庫状況を取得します。

4. 「デフォルトの配布ノード・グループですか (Default distribution node group?)」

このパラメーターは、ノードまたはサプライヤーのデフォルト・セットを定義し、Sterling Selling and Fulfillment Foundation が一致するソーシング・ルールを検出しない場合にソーシングを考慮します。ソーシング・ルールの設定については、159 ページの『ソーシング・ルール』を参照してください。このパラメーターは、エンタープライズがソーシング・ルールを設定する場合のみ関係があります。

**分配グループ (distribution group):** 分配グループ (distribution group) は、定義済みの一連のノードまたは外部組織です。その後、定義済み分配グループをソーシング・ルールと関連付けることができます。Sterling Selling and Fulfillment Foundation は、このグループの一部であるすべてのノードまたは組織を考慮し、ノード選択のために各種要因について最適化します。各ノードまたは組織について、優先番号も指定できます。この優先番号は、スケジューリングに使用される最適化モードに基づくノードの選択に使用されます。最適化に基づくスケジューリングについて詳しくは、167 ページの『スケジューリング・ルール』を参照してください。

例えば、159 ページの図 29では、「East Coast」や「Central」などの複数の分配ノード・グループについて説明しています。単一ノードが、同時に複数のノード・グループの一部になれることに注意してください。

また、出荷ロケーションが隣接しているため、分配グループ (distribution group) を作成することが常に必要であるとは限らないことにも注意してください。固有のノードと外部サプライヤーを区別するためにグループを作成し、さらにソーシング・ルールの順序付けフィーチャーを使用して、最初に固有のノードからの調達を試みることができます。在庫が不十分である場合は、Sterling Selling and Fulfillment Foundation は、サプライヤー・ノードを検索します。

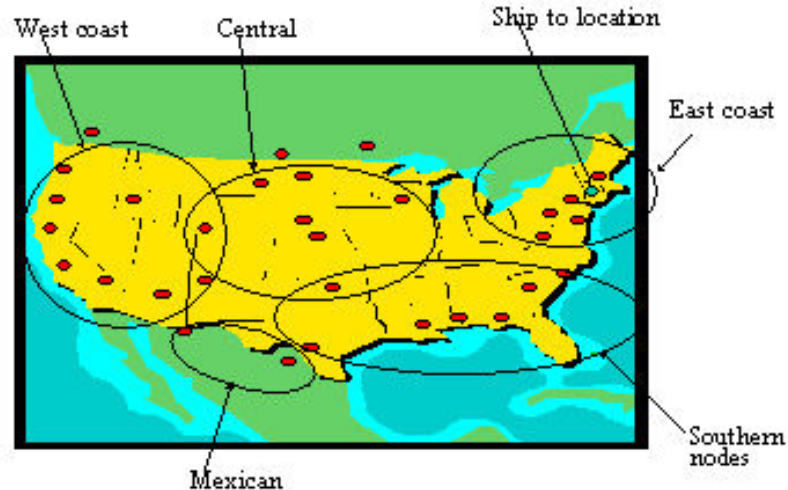


図 29. 分配グループ・ノード

**ソーシング・ルール:** 「ソーシング・ルールは定義されていますか (Sourcing rules defined)」パラメーターを「いいえ (No)」に設定している場合、ソーシング・ルールの設定は関係なく、完全にスキップできます。Sterling Selling and Fulfillment Foundation は、エンタープライズが所有しているすべてのノードを自動的に製品のソーシングの潜在的な候補と見なし、残りの構成に基づいて最適なノードを選択します。

このセクションでは、ソーシング・ルールの設定およびこの構成を Sterling Selling and Fulfillment Foundation がどのように使用するかについて説明します。

ソーシング・ルールは、以下のキー・パラメーターを 1 つ以上指定することによって作成できます。

- アイテム分類またはアイテム ID
- 出荷先ロケーションまたは出荷先ノードの地理的地域
- フルフィルメント・タイプ
- セラー組織
- ソーシング条件

ソーシング・ルールで上記の任意のパラメーター (フルフィルメント・タイプを除く) をボイドのままにするという柔軟性があり、これは、ソーシング・ルールがそのパラメーターのすべての値に適用可能であることを意味します。

各ソーシング・ルールに対して、製品をソーシングするために使用するノードまたは分配グループ (distribution group) のシーケンスを指定できます。

### オーダー・ソーシング分類

Sterling Selling and Fulfillment Foundation は、ソーシング優先順位が高いカスタム・パラメーターを構成するという柔軟性を備えています。例えば、エンタープライズが製品を多数の小売店に提供していて、小売店の 1 つが外国にある倉庫から製品を入荷しないとします。これは、国境を超えることに伴う面倒 (関税、商品チェックによる遅れ) によるものである可能性があります。顧客名またはオーダー・ソ

ーシング分類などの顧客属性を指定して、ソーシング・ロジックにおいてこの制約を考慮することができます。

### アイテム分類またはアイテム ID

Sterling Selling and Fulfillment Foundation は、複数のアイテム分類に基づいてソーシング・ルールを構成する柔軟性を備えており、各アイテム・レベルに必要な多くの設定を回避します。同時に、特別の状況に対してアイテム・レベルでこの設定を指定する柔軟性も備えています。

Sterling Selling and Fulfillment Foundation が実行できる 1 つの方法は、製品明細などの事前定義された「分類」のみをこの設定に使用することです。さまざまな状況で、独自の分類を作成し、ソーシング・ルールの設定に使用することが必要な場合があります。例えば、鋼鉄を扱うプロセス業界では、分類は「鋼鉄の等級」で、小売店では「製品明細」になります。

この柔軟性を実現するために、Sterling Selling and Fulfillment Foundation では、ソーシング・ルールの設定に使用する独自のアイテム属性の指定が可能です。カタログ組織 (catalog organization) (エンタープライズが独自のカタログを管理している場合あり) は、「製品ソーシング」の目的に使用する分類を選択できます。例えば、組織は、製品明細の分類に基づいたソーシングの設定を選択できます。同じインストール済み環境の別のエンタープライズは、ソーシングに他のカスタム分類を選択できます。

### 地理的地域

Sterling Selling and Fulfillment Foundation は、製品の出荷先の地域に基づいてソーシング・ルールを作成する柔軟性を備えています。組織は、出荷地域の決定に使用する地域スキーマを選択できます。それぞれの出荷先住所は、郵便番号に基づいて地域スキーマで出荷地域階層に変換されます。この出荷地域が、正しいソーシング・ルールの検索に使用されます。以下の 163 ページの『例 1: ソーシング・ルール』および 164 ページの『例 2: ソーシング・ルールの地域階層』は、ソーシング・ルールの決定における地域の使用方法を示しています。この機能の詳しい説明については、201 ページの『地域および地域スキーマ』を参照してください。

### 出荷設定の調達

製品の在庫状況を評価するとき、システムは、選択した出荷ソーシング・ルールで各ソーシング・ルールの詳細を検討します。製品が最低シーケンスのノードまたは分配ルール (distribution rule) にない場合、調達ソーシング・ルールが検討され、指定されたシーケンスの順序で評価されます。すべての調達詳細が評価されたが、在庫が見つからない場合、出荷ソーシング・ルールの次の詳細が検討されます。

調達ソーシング・ルールは、論理キット・アイテムではサポートされていません。

ある状況では、特定のロケーションに製品がない場合でも、そのロケーションから製品を出荷する必要がある場合があります。このような状況では、Sterling Selling and Fulfillment Foundation は、選択した出荷ロケーションに製品がない場合、製品の転送オーダーまたは購入オーダーを自動的に作成できます。そのためには、「出荷するために調達する (Procure for shipment)」フラグを「はい」に設定する必要があります。



これを行う必要がある状況の例は、最終出荷先ロケーションに近い 1 つのノードから全オーダーを出荷したいが、希望の出荷ノードにすべてのアイテムが保管されていないか、またはこの出荷ノードが単なるクロス・ドック・ロケーションである場合です。このパラメーターを「はい」に設定します。Sterling Selling and Fulfillment Foundation は、オーダーを希望の出荷ノード・ロケーションに調達し、他のロケーションからこのロケーションへの転送オーダーまたは調達オーダーを作成し、この特定のオーダーを完了します。これらの「連鎖調達オーダー」は、転送オーダーとして働くか、または顧客出荷オーダーに結合された購入オーダーになります。図 30 は、出荷調達の代表的な設定を示しています。

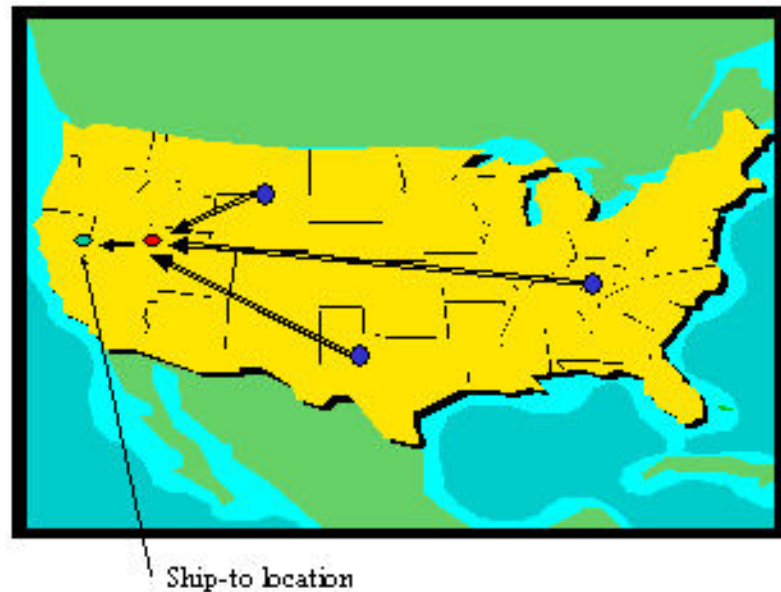


図 30. 調達分配設定

これは、「転送中のマージ」機能の実装に非常に効果的に使用できます。マージ・ロケーションは、オーダーの出荷ノードとして機能できます。Sterling Selling and Fulfillment Foundation は、この設定に基づいて、マージ・ロケーションへの転送を作成できます。マージ・ロケーション自体の選択は、ソーシング・ルールの地理的次元から決定できます。

これを「はい」に設定する他に、該当のノードの「出荷するための調達可能 (Procure for shipment allowed)」フラグも「はい」に設定する必要があります。

転送オーダーまたは調達オーダーの作成を効果的に行うために、以下も設定する必要があります。

- 出荷するための調達のソーシング・ルール

これらは、調達するノードによって設定されるソーシング・ルールです。これらのルールは、ここで説明しているその他のソーシング・ルールに似ています。出荷ソーシング・ルールと同様に、対応するシーケンスを使用して複数のソーシング・ルールを定義できます。各ソーシング詳細には、単一ノードまたは分配ルール (distribution rule) (販売ノードのノード在庫状況に基づいて最適化されている) のいずれかが含まれており、これによって、調達に関して複数のノードを検討で

きます。Sterling Selling and Fulfillment Foundation は、アイテムの調達元のノードの在庫を確認し、在庫がある場合、調達オーダーを作成します。

- 「連鎖オーダーが必要」フラグを「はい」に設定する必要があります
- 転送スケジュール

必須ではありませんが、このスケジュールを作成して、任意の 2 つのノードの間の転送スケジュールを「曜日」ベースで指定できます。通過時間も指定できます。この設定は、予定日付の計算に使用され、174 ページの『予定日付の計算』で説明されています。

**注:** 在庫状況 API を呼び出すとき、Sterling Selling and Fulfillment Foundation は、この情報すべてを考慮し、調達オーダーが作成されることを考慮して日付を提示します。ただし、Sterling Selling and Fulfillment Foundation は、スケジュールリング機能を使用してオーダーをスケジュールリングするまで、実際には調達オーダーを作成しません。

### フルフィルメント・タイプ

エンタープライズは、フルフィルメント・タイプの有効な値のリストを定義し、これをオーダー明細で指定できます。Sterling Selling and Fulfillment Foundation は、フルフィルメント・タイプに関連付けられたソーシング・ルールを使用して、正しいソーシング・ルールを判別します。オーダー明細でフルフィルメント・タイプにブランクの値を指定できますが、ソーシング・ルールでフルフィルメント・タイプをブランクにはできません。オーダー明細でこれをブランクにすると、ソーシング・ルールを選択するのに、オーダーのデフォルトのフルフィルメント・タイプのエンタープライズが使用されます。Sterling Selling and Fulfillment Foundation は、ソーシング・ルールのパラメーターの値とオーダー明細のフルフィルメント・タイプの間で完全一致突き合わせを実行します。他のソーシング・ルール・パラメーターと異なり、このパラメーターのブランク値は、すべてのフルフィルメント・タイプに使用可能であることを意味しません。

この制御を使用して、Sterling Selling and Fulfillment Foundation が理解できないまたはこのリリースで制御を提供していないパラメーター (顧客、オーダー明細の数量、オーダー・タイプなど) に基づいて、さまざまなソーシング・ロケーションを使用する必要があるカスタム要件に適応できます。要件をさまざまなフルフィルメント・タイプに変換し、さまざまなソーシング・ルールを取得できます。

### セラー組織

Sterling Selling and Fulfillment Foundation は、トランザクションに関するセラー組織に基づいて、別の方法で製品を調達する柔軟性を備えています。ストアをさまざまなセラー組織としてモデル化し、製品の販売元のストアに基づいてさまざまなソーシングを実行できます。

**ソーシング・ルールの決定:** パラメーターの一部をボイドのままにすることによって、競合しているソーシング・ルールのセットを作成できることが容易に分かります。Sterling Selling and Fulfillment Foundation は、キー・パラメーターの優先順位を使用して、このような競合を解決します。Sterling Selling and Fulfillment Foundation が使用する優先順位は、以下のとおりです。

1. セラー組織

2. オーダー・ソーシング分類
3. アイテム ID
4. 基本アイテム分類
5. 2 次アイテム分類
6. 3 次アイテム分類
7. 出荷先ロケーションの地理的地域

フルフィルメント・タイプは、オーダー明細パラメーターまたは照会パラメーターと正確に突き合わされます。

一方のルールに関して「アイテム ID」のみを指定し、他方のルールに関して「基本アイテム分類」を指定し、両方のルールが調達される製品の値と一致した場合、Sterling Selling and Fulfillment Foundation は、基本アイテム分類で指定されたルールではなく、アイテム ID で指定されたルールを選択します。また、パラメーターに値が指定されているルールの方が、値がブランクのままのルールよりも優先されます。「例 1: ソーシング・ルール」では、この概念をさらに詳しく説明します。

### 例 1: ソーシング・ルール

この例では、基本的なソーシング・シナリオについて説明します。ソーシング・ルール・パラメーターは、表 20 で説明されているように定義されます。

表 20. 優先のソーシング・ルール・パラメーターの例

ルール	アイテム ID	基本アイテム分類	地理的地域
1	ITEM1	-	-
2	-	TV	-
3	ITEM1		カリフォルニア
4	-	-	カリフォルニア
5	ITEM1		サンフランシスコ、カリフォルニア

オーダー明細のアイテム ID、アイテム分類、および出荷ロケーションの各値、つまり ITEM1、TV、およびニューヨークを使用してオーダー明細をソーシングしている場合、Sterling Selling and Fulfillment Foundation は、ルール #1 を使用します。ルール #2 も条件を満たしますが、ルール #1 の方が他のルールよりも具体的です。アイテム ID パラメーターには、基本アイテム分類パラメーターよりも高い優先順位が与えられます。

オーダー明細のアイテム ID、基本アイテム分類、および出荷ロケーションの各値、つまり、それぞれ、ITEM1、TV、およびロサンゼルス - カリフォルニアを使用してオーダー明細をソーシングしている場合、Sterling Selling and Fulfillment Foundation は、ルール #3 を使用します。ルール #1、2、および 4 もオーダー明細の条件を満たしますが、ルール #3 以外の各ルールは一部のパラメーターがブランク (ボイド) になっていますが、ルール #3 はそうならないため、ルール #3 に他のルールよりも高い優先順位が与えられます。アイテム ID パラメーターには、地理的地域パラメーターよりも高い優先順位が与えられます。

オーダー明細のアイテム ID、基本アイテム分類、および出荷ロケーションの各値の ITEM1、TV、およびサンフランシスコ-カリフォルニアのそれぞれを使用してオーダー明細をソーシングしている場合、Sterling Selling and Fulfillment Foundation は、ルール #5 を使用します。ルール #1、2、3、および 4 もオーダー明細の条件を満たしますが、ルール #5 以外の各ルールは一部のパラメーターがブランク (ボイド) になっていますが、ルール #5 はそうになっていないため、ルール #5 に他のルールよりも高い優先順位が与えられます。また、ルール #5 で指定された地域の方がルール #3 の地域よりも具体的なため、ルール #5 にルール #3 よりも高い優先順位が与えられます。

「例 2: ソーシング・ルールの地域階層」では、定義済みの地域を使用してソーシング・ルールを決定するための階層についてさらに詳しく説明します。

## 例 2: ソーシング・ルールの地域階層

ソーシング・ルールを選択するときに、最も低いレベルの地域と一致するルールには、高いレベルの地域が指定されているいずれのルールよりも高い優先順位が与えられます。

地域の階層は、以下のように定義されます。

```
- USA
-- New York State
--- New York
---- Manhattan
----- Down Manhattan
----- Midtown Manhattan
```

ソーシング・ルールが以下のように定義されていると仮定します。

- ニューヨークの任意の住所への出荷に「Node1」を使用します。
- ダウンタウン・マンハッタンの任意の住所への出荷に「Node2」を使用します。

出荷住所に対する Sterling Selling and Fulfillment Foundation のノード選択では、ダウンタウン・マンハッタンの住所へ出荷する場合には Node2 を選択し、ダウンタウン・マンハッタン以外のマンハッタンの任意の住所に対しては Node1 を選択します。

## ソーシング・テンプレートのシーケンスの使用

各ソーシング・ルールに対して、オーダー明細をソーシングするために使用するソーシング・テンプレートのシーケンスを指定できます。

Sterling Selling and Fulfillment Foundation は、最高のシーケンス (最も小さい数字) のソーシング・テンプレートから製品を調達しようとします。複数の選択が Sterling Selling and Fulfillment Foundation で可能な場合、ノード選択は、オーダーに関連付けられたスケジューリング・ルールの設定に基づいて最適化されます。最適化設定については、167 ページの『スケジューリング・ルール』で詳細に説明します。

このシーケンスで指定されたソーシング・テンプレートに製品がない場合、Sterling Selling and Fulfillment Foundation は、次のシーケンスのソーシング・テンプレートから調達しようとします。

## 順序付けを使用する場合

特定の状況においては、ソーシングのために Sterling Selling and Fulfillment Foundation が使用する最適化ロジックをより詳細に制御する必要があります。

例えば、テキサス州エルパソのロケーションに出荷する必要があるとします。北アメリカには複数の倉庫があります。最初に米国の倉庫を検索し、米国の倉庫に在庫がない場合のみメキシコの倉庫を検索します。北アメリカのすべてのロケーションを同じシーケンスに指定した場合、Sterling Selling and Fulfillment Foundation は、出荷元と出荷先のロケーションの間の距離に基づいて最適化し（これは、他のパラメーターに関して最適化するためにも制御できます）、その結果、メキシコの倉庫から出荷される場合があります。考慮する分配グループ (distribution group) のシーケンスの制御を可能にすると、このような地政学的状況进行处理できるようになります。図 31 は、この概念を説明しています。

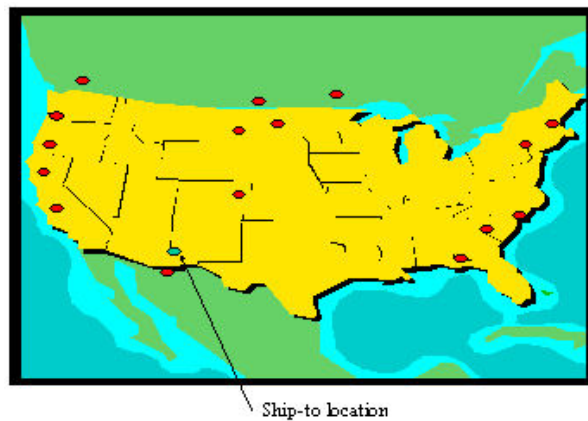


図 31. ソーシング・テンプレートの順序付け

特定のシーケンスを指定する他の理由は、システム・パフォーマンス上の理由です。

ニューヨークの住所に出荷する必要があると仮定します。国内には 30 から 40 の配送センター (distribution center) があり、製品がある最も近い配送センター (distribution center) から出荷する必要があります。東海岸ノード、西海岸ノード、中央アメリカ・ノード、および南部アメリカなどの地理的地域に基づいて、出荷ノードを 4 から 5 個の配送ノード・グループにグループ化できます。

この特定の状況では、Sterling Selling and Fulfillment Foundation がまず東海岸ノードから調達しようとし、次に、これに続く地域から調達しようとするように、シーケンスを設定します。これは、166 ページの図 32 で説明されています。同じノードを複数のグループの一部として含むことができることがわかります。

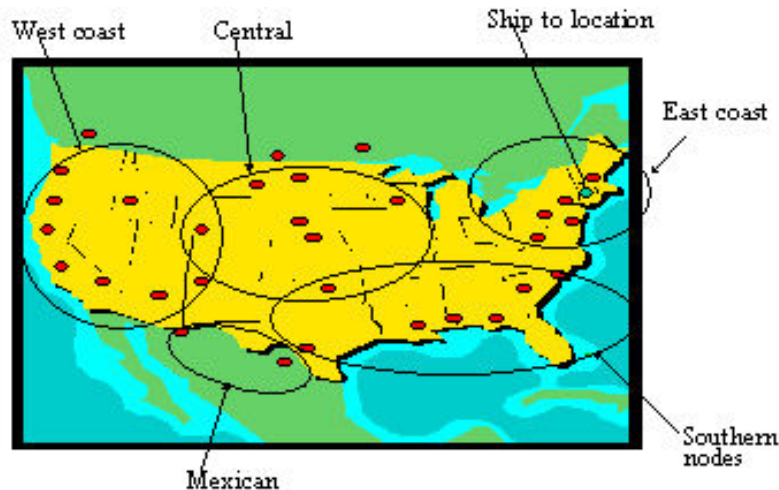


図 32. ソーシング・テンプレートの順序付け - 例 2

上記の設定での明白な疑問は、なぜ単一のシーケンスを指定して、米国のロケーションすべてを含む配送ノード・グループを関連付けないのかということです。Sterling Selling and Fulfillment Foundation は、既に最も近いロケーションを選択するように最適化しているため、必要な設定が少なくなり、同じ結果を得ることができます。

これを複数のシーケンスとして設定する主な理由は、Sterling Selling and Fulfillment Foundation が余分な在庫検索を実行することを防止し、成功の見込みが高い機能からの応答時間を向上するためです。Sterling Selling and Fulfillment Foundation は、指定されたノードすべての中で選択を最適化するため、指定されたノードすべての在庫を読み取ってから最終的な選択を行う必要があります。米国のすべてのロケーションを指定した場合、Sterling Selling and Fulfillment Foundation は、各ロケーションの在庫を読み取ってから最終的な選択を行います。検索するノードの小さいサブセットを作成できる場合、おそらくシステム・リソースの節約になり、応答時間が向上します。設定を少なくすることとパフォーマンスを良くすることの間の選択は、主に検索するロケーション数に基づきます。1 つのシーケンスに 5 から 10 のロケーションを配置することが適切です。1 つのシーケンスにこれ以上のロケーションを配置すると、パフォーマンスがいくらか低下する可能性があります。

#### 複数をシーケンス使用時のノードの優先順位

Sterling Selling and Fulfillment Foundation は、ソーシング・ルールのシーケンス設定それぞれを検索し、製品の在庫状況を検出するまで製品を調達します。指定の時刻における個別ノードの優先順位は、シーケンスのすべてのグループのノードの最高優先順位として計算されます。ノードがシーケンスの 1 つとして明示的に指定されている場合、優先順位は 0 (最高) として処理されます。

各ソーシング・ルールに対して、ソーシング・ルール構成に基づいて将来の在庫 (future inventory) を考慮するスケジューリングを拡張するシーケンスも指定できます。優先の代替がアイテムに対して構成されている場合、手持ちの在庫の前にそのノードに出荷する必要があります。例えば、以下について考えてみます。

シーケンス番号	シーケンス
シーケンス 1	地域の配送センターから調達
	5 日以内に入荷する将来の在庫 (future inventory) を考慮
	代替可能
シーケンス 2	地域の配送センターから調達
	30 日以内に入荷する将来の在庫 (future inventory) を考慮
	代替可能

手持ち在庫が 5 個で、優先の代替が 2 日後に 5 個、20 日後に 10 個入荷します。

顧客 1 が 5 アイテムをオーダーすると、優先の代替が 5 日以内に入荷するため、優先の代替が顧客に出荷されます。

顧客 2 がさらに 5 アイテムをオーダーすると、優先在庫が 0 になったため、手持ち在庫が出荷されます。

ここで、顧客 3 が 5 アイテムをオーダーすると、優先の代替および手持ち在庫が 0 になったため、30 日以内にアイテムを取得します。

したがって、これらの構成は、追加のノードの選択を優先順位付けし、将来の在庫 (future inventory) を確認します。

**スケジューリング・ルール:** このセクションでは、製品のソーシングにのみ関連するパラメーターについて説明します。日付の決定のために使用されるスケジューリング・ルールの一部として定義されているパラメーターは他にもありますが、これらについては、174 ページの『予定日付の計算』で説明します。

スケジューリング・ルールはエンタープライズ・レベルでセットアップします。Sterling Selling and Fulfillment Foundation は、オーダー・トランザクションのエンタープライズによって定義されているルールを使用します。オンザフライの照会を使用する場合、Sterling Selling and Fulfillment Foundation は、要求を行った組織コードの主要エンタープライズによって定義されているルールを使用します。

### 地理を使用するかどうか?

このパラメーターを使用して、ノードと出荷先ロケーション間の距離に基づいて、ノードの優先順位付けをオンまたはオフにすることができます。「地理を使用」の設定が「はい」の場合に、「優先順位」に基づいてノード選択を最適化すると、Sterling Selling and Fulfillment Foundation はノード優先順位を次のように計算します。

Weight factor for distance \* distance in miles as calculated based on longitude and latitude + Weight factor for priority \* priority setup in the distribution node group. If using multiple sequences, see "Priority of Nodes When Using Multiple Sequences".

この結合された優先順位を使用して、最も小さい優先順位番号のノードを選択します。スケジューリング・ルールの一部として、重み要因もセットアップされます。

ノード優先順位のセットアップを最優先したい場合、距離にかかわらず、最優先する「優先順位」の重み要因を 100,000 にセットアップします。2 つのノードの優先順位が同じ場合は、距離が決定基準になります。距離の重要性を増やしたい場合は、優先順位の重み要因を 0 か、別の低い数字にセットアップします。使用する重み要因と優先順位番号は、距離 (マイル単位) の計算は内部で実行されるという事実に基づいて決定できます。

## 最適化タイプ

複数のノードと日付がソーシングに使用可能な場合、このパラメーターを使用することで、以下に基づいてソーシングの選択を行うことができます。

- ノードの優先順位
- 出荷先ロケーションからノードまでの距離
- 配達可能な日付
- 結果として生じる出荷の数
- コスト・ベースのスケジューリング

必要な最適化タイプは、スケジューリング・ルールの一部として設定できます。以下の最適化タイプを設定できます。

- 優先順位 - これを設定した場合、ノードの選択は、分配グループ (distribution group) での優先順位のセットアップおよびノードから出荷先ロケーションの距離に基づきます。優先順位は、ノード優先順位の重み \* ノード優先順位 + 距離の重み \* ノードから出荷先ロケーションまでの距離を使用して計算されます。距離の計算は 2 つのロケーションの経度と緯度に基づいており、単なる「直線」距離です。ほとんどのシナリオでは、距離の計算には実際の道路距離から約 20% の許容誤差があります。
- 出荷先ロケーションからノードまでの距離 - スケジューリング・ソリューションを最適化する際、スケジュールはノード優先順位コストを考慮に入れることができます。ノードの優先順位は、優先順位コスト要因を使用して、スケジュール・コストに変換されます。ノード優先順位と優先順位コスト要因の積によって、その特定のノードの使用における追加コストが計算されます。次に例を示します。

分配グループ (distribution group) が明細 DG1 に渡されます: N1 - 優先順位 1、N2 - 優先順位 10。

N1 からの最終レグ出荷コストは \$10 で、N2 からは \$2 です (他のコストは使用していないものと想定)。

最適なソリューションは N2 です。

ノード優先度コスト要因を使用可能にすると、次のようになります。

優先度コスト要因 - \$1

Option1 - N1 -  $1 * \$1 + \$10 = \$11$

Option2 - N2 -  $10 * \$1 + \$2 = \$12$

N1 の方がコストが低くノードの優先度も低いため、N1 が最適なソリューションです。



- 日付 - Sterling Selling and Fulfillment Foundation は、製品を最短で配達できるノードを選択します。配達日は、出荷ロケーションと出荷先ロケーション間で計算された通過時間に基づいて計算されます。通過時間計算について詳しくは、183ページの『通過時間の決定』を参照してください。
- 出荷数 - 出荷日とノードの選択は、最終的に行う合計出荷数が少なくなるような方法で実行されます。これは、将来の在庫 (future inventory) に対して出荷を集約することによっても実現されています。結果として、顧客に通知した内容よりも出荷が遅延することがあります。この問題を解決するため、出荷を遅延できる最大日数 (遅延期間) を構成できます。

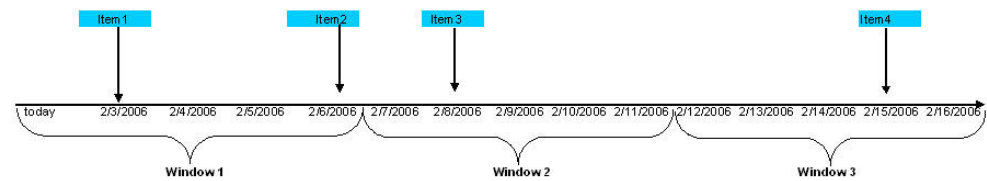


図 33. 日付範囲の最適化ロジック

図 33 は、各期間が最適化日付範囲 (5 日に構成されています) をどのように示しているかを説明しています。アイテムはさまざまな日付で使用可能です。Item1 は 06/2/3 に使用可能になり、06/2/15 まで引き続き使用可能です。同様に、Item2 は 06/2/6 に使用可能になり、Item3 は 06/2/8 から 06/2/15 まで使用可能です。Item 4 が使用可能なのは 06/2/15 だけです。顧客がすべてのアイテムのオーダーを発行し、出荷を待つことに同意した場合、これらのアイテムは 1 つの出荷に集約され、06/2/15 に出荷されます。これにより、複数回出荷することによるコストが削減されます。この場合、日付範囲の最適化ロジックのルールが適用されます。一方、顧客がアイテムが使用可能になったときにアイテムを出荷するよう希望している場合、出荷最適化が考慮され、顧客には、単一の期間内に収まる最短日である 06/2/6 に Item1 と Item2 が出荷されます。

遅延期間は、高チケット・アイテムに対応するために、アイテム・レベルで指定できます。この期間は 0 日に設定することもできます。つまり、製品は遅延しないことになります。さらに、将来の在庫 (future inventory) に関する情報 (日付) は正確ではないことが多いため、出荷を遅らせて将来の在庫 (future inventory) と集約しないように選択することもできます。これにより、手持ちおよび調達済みの在庫は確実にただちに出荷されるようになります。

- コスト・ベースのスケジューリング - コスト・ベースの最適化では、フルフィルメント・オプションを決定する際にスケジューリング・プロセスで荷役コストを考慮できます。荷役コストは次のコストで構成されます。
  - アイテムの在庫コストでは、Sterling Selling and Fulfillment Foundation で維持されている平均コストを使用して、特定のノードのアイテムのユニット・コストを計算します。
  - 取り扱いコスト (入庫および出庫取り扱いコストの両方) は、ユニットあたり、明細あたり、および出荷あたりで構成できます。入庫コストは、スケジューリング中に調達を選択した場合に考慮され、調達先ノードに適用されます。出庫取り扱いコストは出荷について考慮され、出荷ノードに適用されます。

- 転送コストは、スケジューリングの際に調達オプションを選択した場合に考慮されます。転送コストは、ユニットごと、ユニット重量ごと、およびユニットの距離ごと (per unit distance) に構成できます。または、個々の転送スケジュールに対して、固定転送コストを指定することができます。
- 最終レグ・コストは、顧客への最終的な出荷のコストを計算するために使用されます。これは通常、指定された配送業者または配送業者サービスについて計算されます。最終レグ・コストを計算するためのユーザーEXIT が提供されています。

スケジューリングの際に、荷役コスト・オプションが評価されて、最小荷役コストが決定され、これがコスト・ベース・オプションとして使用されます。荷役コストは、ユニットあたりの荷役コストを反映するように割り当てられます。優先順位はスケジューリング・ルールで構成されます。上記にリストしたさまざまなコストに加え、コスト要因を使用して、優先順位をコストに変換することができます。例えば、分配グループ (distribution group) 内により低い優先順位を持つ (優先順位番号が大きい) ノードがある場合、そのノードは、優先順位が高いノードと比較して、よりコストが高くなります。

最適なコスト・ベースの最適化タイプを決定するには、より多くのソリューションを評価する必要があります。より多くのソリューションを評価するために、約束およびスケジューリング API の MaximumRecords を増加させることができます。日付の最適化を使用している場合、最も早いオプションが最初に考慮されるため、コスト・ベースのスケジューリングは使用されません。

### コスト・ベースのスケジューリング: 例

次の例では、特定のオプションについて荷役コストを計算します。顧客が DVD プレイヤー 1 つと DVD を 3 つオーダーしたと想定しましょう。DVD プレイヤーは DC1 で使用可能で、DVD は DC2 で使用可能です。この場合、スケジュールには次の 2 つのオプションがあります。

- オプション 1 は、個々のアイテムを DC1 および DC2 から 2 つの別個の出荷として顧客に出荷することです。
- オプション 2 は、両方のアイテムを DC3 に転送した後、DC3 から 1 つの出荷として顧客に出荷することです。

オプション 2 の方が費用が安いので、スケジュールでは、コスト・ベースのスケジュール・オプションとしてオプション 2 が選択されます。これらのオプションについて、次のセクションで説明します。

#### オプション 1

すべてのコストが構成済みと仮定すると、オプション 1 の荷役コストは次のように計算されます。

- 出荷 1 では、DVD プレイヤーが顧客に直接出荷されます。次の式を使用して、出荷 1 の荷役コストを計算します。

(合計アイテム・コスト + 出庫取り扱いコスト + 最終レグ・コスト) / 合計ユニット

この式を使用した場合、次の計算に示すように、出荷 1 の荷役コストは \$120.50 になります。

$(1 \text{ ユニット} * \$100 \text{ (平均コスト)} + \$3 \text{ (出荷あたり)} + \$17.50 \text{ (顧客への陸路出荷)}) / 1 \text{ ユニット} = \$120.50$

- 出荷 2 では、DVD 3 枚が DC3 から直接顧客に出荷されます。次の式を使用して、出荷 2 の荷役コストを計算します。

$(\text{アイテム・コスト} + \text{出庫取り扱いコスト} + \text{最終レグ・コスト}) / \text{合計ユニット}$

この式を使用した場合、次の計算に示すように、出荷 2 の荷役コストは \$14.33 になります。

$(3 \text{ ユニット} * \$10 \text{ (平均コスト)} + \$3 \text{ (出荷あたり)} + \$10 \text{ (顧客への陸路出荷)}) / 3 \text{ ユニット} = \$14.33$

オプション 1 を履行する荷役コストを決定するために、出荷 1 と出荷 2 の計算を加えると、\$124.83 になります。

## オプション 2

すべてのコストが構成済みと仮定すると、オプション 2 の荷役コストは次のように計算されます。

- 出荷 1 では、DVD プレイヤーが DC1 から DC3 に出荷されます。次の式を使用して、出荷 1 の荷役コストを計算します。

$(\text{合計アイテム・コスト} + \text{出庫取り扱いコスト}) / \text{合計ユニット}$

この式を使用した場合、次の計算に示すように、出荷 2 の荷役コストは \$106 になります。

$(1 \text{ ユニット} * \$100 \text{ (平均コスト)} + \$3 \text{ (出荷あたり)} + \$2 \text{ (DC1 から DC3 間の固定転送コスト)} + \$1 \text{ (DC3 での出荷あたり)}) / 1 \text{ ユニット} = \$106$

- 出荷 2 では、DVD が DC2 から DC3 に出荷されます。次の式を使用して、出荷 2 の荷役コストを計算します。

$(\text{合計アイテム・コスト} + \text{出庫取り扱いコスト} + \text{転送コスト} + \text{入庫取り扱いコスト}) / \text{合計ユニット}$

この式を使用した場合、次の計算に示すように、出荷 2 の荷役コストは \$11.67 になります。

$(3 \text{ ユニット} * \$10 \text{ (平均コスト)} + \$3 \text{ (出荷あたり)} + \$1 \text{ (DC2 から DC3 間の固定転送コスト)} + \$1 \text{ (DC3 での出荷あたり)}) / 3 \text{ ユニット} = \$11.67$

- 出荷 3 では、DVD プレイヤーと DVD が DC3 から顧客に出荷されます。次の式を使用して、出荷 3 の荷役コストを計算します。

$(\text{在庫コストなし (アイテム/調達済み)} + \text{出庫取り扱いコスト} + \text{最終レグ・コスト}) / 4 \text{ ユニット}$

この式を使用した場合、次の計算に示すように、出荷 3 の荷役コストは \$3.33 になります。

$$(\$3 \text{ (出荷あたり)} + \$7 \text{ (DC3 から顧客への陸路出荷)}) / 3 \text{ ユニット} = \$3.33$$

オプション 2 を履行する荷役コストを決定するために、出荷 1、出荷 2、および出荷 3 の計算を加えると、\$121 になります。

### 完全なオーダーを出荷 (Ship Complete Order)

このパラメーターは、約束照会要求内のすべての製品明細が完全にスケジュールされているかまったくスケジュールされていないかのどちらかであることを保証します。ただし、それぞれの明細の調達元は異なる出荷ロケーションである可能性があります。

### 単一出荷ノードからオーダーを出荷 (Ship Order from Single Ship Node)

このパラメーターは、約束照会要求内のすべての製品明細が完全にスケジュールされているかまったくスケジュールされていないかのどちらかであることを保証します。また、完全な要求が単一の日付に単一のノードから調達されることも保証します。これは、「完全なオーダーを出荷 (ship complete order)」のスーパー・セットで、このパラメーターを設定する場合、「出荷完了」が前提になります。

### 完全な明細を出荷 (Ship Complete Line)

このパラメーターは、個別明細ベースのすべての製品明細が完全に調達されているかまったく調達されていないかのどちらかであることを保証します。ただし、それぞれの明細の調達元は異なる出荷ロケーションである可能性があります。このルールと「完全なオーダーを出荷 (ship complete order)」ルールの違いは、このルールでは要求のすべての明細が完全に調達されることは強制されない点です。特定の明細を調達する一方で、同じ要求の別の明細をバックオーダーできます。

### 単一出荷ノードから明細を出荷 (Ship Line from Single Ship Node)

このパラメーターは、照会要求内の製品明細が完全にスケジュールされているかまったくスケジュールされていないかのどちらかであることを保証します。また、個々の各明細が単一の日付に単一のノードから調達されることも保証します。これは、「完全な明細を出荷 (ship complete line)」のスーパー・セットで、このパラメーターを設定する場合、「完全な明細を出荷 (ship complete line)」が前提になります。ただし、このルールでは、すべての明細が同じノードから出荷されることは強制されません。特定の明細をノード 1 から完全に出荷する一方で、別の明細をノード 2 から完全に出荷できます。

**ソーシングのその他の制約:** 製品ソーシングには、次の追加の制約も設定できません。

#### 一連の製品明細の一括出荷

単一のオーダー内のオーダー明細のサブセットを、同一ノードから同時に一括出荷することが要求される状況があります。Sterling Selling and Fulfillment Foundation は、「オーダー明細の一括出荷グループ」を作成する機能を備えています。

補充数は明細ごとに指定できます。少なくとも、一括出荷グループに含まれるすべての明細の補充数が使用可能である必要があります。このグループの構成部分である明細はすべて、同じノードから同時に出荷されます。

これに必要な構成はありません。オーダー・トランザクションのそれぞれで、「一括出荷依存関係 (Ship together dependency)」パラメーターの設定を使用することで、一括出荷グループを関連付けることができます。一括出荷する必要がある一連の明細を選択して、依存関係グループを作成します。Sterling Selling and Fulfillment Foundation の約束機能によって、これらの明細に一括出荷がスケジュールされることが保証されます。

**注:** この制約は、オーダーがシステム内に存在する場合のみ、明細のサブセット・レベルで指定できます。オーダーなしで問い合わせを行う場合は、問い合わせレベルでこの制約を指定できます。これにより、Sterling Selling and Fulfillment Foundation で要求内のすべての明細が一括出荷されることが保証されます。

### 一連の明細の一括配達

この制約と「一括出荷」制約の主な違いは、この制約では依存グループの構成部分である個別の明細を異なるロケーションから出荷できるが、Sterling Selling and Fulfillment Foundation によって、依存グループを構成するすべての明細の配達がスケジュールされるか、すべてがされないかのどちらかであることが保証される点です。これらの明細を特定のロケーションで集約するために「マージ・ノード」を指定して、これらの明細が一括配達されるように保証することができます。オーダー・リリースが実行されると、「マージ・ノード」がリリースに適用されます。また、出荷集約プロセスによって、「マージ・ノード」がその出荷に適用されることも保証されます。出荷倉庫は、実際の出荷先の場所ではなく、この「マージ・ノード」への出荷が実行されることを確認する必要があります。

これに必要な構成はありません。オーダー・トランザクションのそれぞれで、「一括配達依存関係 (Deliver together dependency)」パラメーターを使用して一括配達グループを指定できます。一括配達する必要がある一連の明細を選択して、依存関係グループを作成します。Sterling Selling and Fulfillment Foundation の約束機能によって、これらの明細に一括配達がスケジュールされることが保証されます。

**注:** この制約は、オーダーがシステム内に存在する場合のみ、明細のサブセット・レベルで指定できます。オーダーなしで問い合わせを行う場合は、問い合わせレベルでこの制約を指定できます。これにより、Sterling Selling and Fulfillment Foundation で要求内のすべての明細が一括出荷されることが保証されます。

特定のオーダー明細に対して実行されるすべての出荷で、指定された最小数量がその出荷に含まれることを保証するには、任意の要求明細で補充数を指定できます。Sterling Selling and Fulfillment Foundation によって、ノードからの出荷がこの数量を最小数量として実行されることが保証されます。オーダー明細の残余数量がこの補充数に満たない場合はスケジュールされません。必要に応じて残余数量を取り消すか、補充数を減らすことができます。

### 明細フルフィルメントに単一ノードを使用

このルールでは、オーダー明細を 1 つのノードのみから充足することが要求されます。ただし、スケジュールリング・ルールの「単一ノードから明細を出荷 (Ship Line

From Single Node)』とは異なり、このルールでは単一ノードからの明細の完全出荷は要求されません。オーダー明細の一部を出荷でき、残余数量は後の時点で出荷できます。

このルールが有効化されていると、次のような追加機能が使用可能になります。

- 一部バックオーダーの明細分割 (事前定義の出荷ノードがある明細、およびない明細)
- 最適出荷ノードへの明細のバックオーダー

これらのパラメーターについて詳しくは、see 200 ページの『バックオーダーの取り扱い』を参照してください。

## 予定日付の計算

指定された照会に対する予定出荷日および予定配達日の計算には、以下のようにさまざまな要因が関連しています。

- ノードによって要求される最小通知時間
- 通知スケジュール
- サービス・レベル (Level of Service)
- 運送会社ピックアップ・スケジュール
- 運送会社保留時間
- ノードの出荷カレンダー
- ノードの受入カレンダー
- 受入処理時間
- 出荷に必要な輸送時間
- 出荷元と出荷先間の事前定義済み転送スケジュール
- 製品およびサービスの明細の間に定義されている順序制約
- 製品の検索ウィンドウ - 要請日より後にオーダー明細をスケジュールできる期間

## 出荷日の計算

このセクションでは、「可能な限り早く (ASAP)」オーダーと将来のオーダーの出荷日の計算について説明します。

ASAP オーダーとは、出荷指定日または配達指定日が指定されていないオーダーのことです。これらのオーダーはいつでも可能なときに出荷できます。また、出荷指定日または配達指定日が過去になっているすべてのオーダーも ASAP オーダーとみなされ、同じ方法で取り扱われます。

将来のオーダーは、少なくとも出荷指定日または配達指定日のいずれかが定義されていて、その指定日が将来になっているオーダーとして定義されます。

正確な出荷日および配達日になるように、システムは最初にオーダー明細の出荷日と配達日の範囲を計算します。システムは、続いて、製品の調達元のさまざまなロケーションに基づいて、正確な日付になるように範囲を絞り込みます。

**日付範囲の初期計算:** ASAP オーダーおよび将来のオーダーの初期日付範囲は、以下の基準に基づいて計算されます。

- ASAP オーダー
  - 出荷日の下限は、現在時刻として設定されます。
  - 出荷日の上限は、キャンセル日に設定されます (指定する場合)。指定しない場合は、現在時刻に、「許容出荷遅延」ウィンドウのスケジューリング・ルールのパラメーター・セットアップを加えた日付として設定されます。
  - 配達日の下限は、現在時刻として設定されます。
  - 配達日の上限は、キャンセル日に設定されます (指定する場合)。配達指定日を指定せずに出荷指定日およびキャンセル日が指定されていた場合は、唯一の例外です。この場合のキャンセル日は、出荷日の上限として扱われ、配達日としては扱われません。キャンセル日が指定されなかった場合、これは、通過時間に配慮して、出荷日の上限に 60 日を加えた日付として計算されます。実際の日付が計算されると、この通過は実際の通過時間までに絞り込まれます。
- 将来のオーダー
  - 出荷日の下限は、出荷指定日です (指定する場合)。指定されない場合は、現在時刻に設定されます。
  - 出荷日の上限は、キャンセル日に設定されます (指定する場合)。キャンセル日は指定されていないが、出荷指定日が指定されていた場合は、出荷指定日に、「許容出荷遅延」ウィンドウのスケジューリング・ルールのパラメーター・セットアップを加えた日付に設定されます。出荷指定日が指定されなかった場合、これが配達日の上限として設定されます。
  - 配達日の下限は、配達指定日として設定されます。指定されない場合、出荷日の下限として設定されます。
  - 配達日の上限は、キャンセル日に設定されます (指定する場合)。配達指定日を指定せずに出荷指定日およびキャンセル日が指定されていた場合は、唯一の例外です。この場合のキャンセル日は、出荷日の上限として扱われ、配達日としては扱われません。キャンセル日が指定されなかった場合、これは、配達指定日に「許容出荷遅延」日数をプラスしたものと計算されます。配達指定日が指定されなかった場合は、これは、通過時間を考慮して、出荷日の上限に 60 日を加算した日付として設定されます。実際の日付が計算されると、この通過は実際の通過時間までに絞り込まれます。

「出荷遅延」ウィンドウは経過日数で指定されることにも注意してください。これは通常、少なくとも、いずれかのノードまたはアイテムが必要とする最大通知時間よりも長いものとして設定されると予想されます。より微細な制御がオーダーの最後の出荷可能日まで必要な場合は、そのより微細な制御にキャンセル日を使用してください。Sterling Selling and Fulfillment Foundation では、「許容出荷遅延日数」のデフォルトとして 30 日を使用します。

#### 初期日付範囲計算の組み合わせ

176 ページの表 21 は、配達指定日、出荷日、およびキャンセル日のさまざまな組み合わせと、Sterling Selling and Fulfillment Foundation により初期出荷および配達日付範囲がどのように計算されるかについて詳しく説明します。

表 21. 初期日付範囲計算の組み合わせ

配達指定日 (Requested Delivery Date)	要求された 出荷日	キャンセル日	初期出荷開 始 (Initial Ship Start)	初期出荷終 了 (Initial Ship End)	初期配達開 始日 (Initial Delivery Start Date)	初期配達終 了日 (Initial Delivery End Date)
指定なし	指定なし	指定なし	現在時刻	現在時刻から許容出荷遅延日数後の日の終了時刻	現在時刻	初期出荷終了日 + 60 日
指定なし	指定なし	指定済み	現在時刻	キャンセル日	現在時刻	キャンセル日
指定なし	将来の日付	指定なし	将来の日付が指定済み	将来の日から許容出荷遅延日数後の日の終了時刻	将来の日付が指定済み	初期出荷終了 + 60 日
指定なし	将来の日付	指定済み	将来の日付が指定済み	キャンセル日	将来の日付が指定済み	初期出荷終了 + 60 日
指定なし	過去の日付	指定なし	現在時刻	現在時刻から許容出荷遅延日数後の日の終了時刻	現在時刻	現在時刻 + 60 日から許容出荷遅延日数後の日の終了時刻
指定なし	過去の日付	指定済み	現在時刻	キャンセル日	現在時刻	キャンセル日 + 60 日
将来の日付	指定なし	指定なし	現在時刻	将来の日付から許容出荷遅延日数後の日の終了時刻	将来の日付	将来の日付から許容出荷遅延日数後の日の終了時刻
将来の日付	指定なし	指定済み	現在時刻	キャンセル日	将来の日付	キャンセル日
将来の配達日	将来の出荷日	指定なし	将来の出荷日	将来の配達日から許容出荷遅延日数後の日の終了時刻	将来の配達日	将来の配達日から許容出荷遅延日数後の日の終了時刻
将来の配達日	将来の出荷日	指定済み	将来の出荷日	キャンセル日	将来の配達日	キャンセル日
将来の配達日	過去の出荷日	指定なし	現在時刻	将来の配達日から許容出荷遅延日数後の日の終了時刻	将来の配達日	将来の配達日から許容出荷遅延日数後の日の終了時刻
将来の配達日	過去の出荷日	指定済み	現在時刻	キャンセル日	将来の配達日	キャンセル日



表 21. 初期日付範囲計算の組み合わせ (続き)

配達指定日 (Requested Delivery Date)	要求された 出荷日	キャンセ ル日	初期出荷開 始 (Initial Ship Start)	初期出荷終 了 (Initial Ship End)	初期配達開 始日 (Initial Delivery Start Date)	初期配達終 了日 (Initial Delivery End Date)
過去の配達 日	指定なし、 または過去 の出荷日	指定なし	現在時刻	現在時刻から 許容出荷 遅延日数後 の日の終了 時刻	現在時刻	現在時刻から 許容出荷 遅延日数後 の日の終了 時刻
過去の配達 日	指定なし、 または過去 の出荷日	指定済み	現在時刻	キャンセル 日	現在時刻	キャンセル 日

### 初期日付範囲の計算結果

以下の例は、ソーシングとスケジューリングを決定するために使用される初期日付範囲を示しています。

以下のパラメーターを指示どおりに設定した場合、初期日付範囲の計算結果は、表 22 に示されているものになります。

- 現在時刻 = 9/8/2003 3PM
- 許容出荷遅延 = 30 日 (このパラメーターは一般に、少なくとも、任意のノードの通知時刻に最大連続非稼働日数を加えたものに設定する必要があることに注意してください)

表 22. 日付範囲の計算結果例

配達指定日 (Requested Delivery Date)	要求され た出荷日	キャンセ ル日	初期出荷開 始 (Initial ship start)	初期出荷終 了 (Initial ship end)	初期配達開 始日 (Initial delivery start date)	初期配達終 了日 (Initial delivery end date)
指定なし	指定なし	指定なし	9/8 3PM	10/8 00AM	9/8 3PM.	12/8 00AM
指定なし	指定なし	9/30	9/8 3PM	9/30 00AM	9/8 3PM.	9/30 00AM
指定なし	9/15 2PM	指定なし	9/15 2PM	10/15 00AM	9/15 2PM	12/15 00AM
指定なし	9/15 2PM	9/30	9/15 2PM	9/30 00AM	9/15 2PM	11/30 00AM
指定なし	過去の日 付	指定なし	9/8 3PM	9/9 00 AM	9/8 3PM	11/9 00AM
指定なし	過去の日 付	9/30	9/8 3PM	9/30 00AM	9/8 3PM	11/30 00AM
9/15 2PM	指定なし	指定なし	9/8 3PM	10/15 00AM	9/15 2PM	10/15 00AM
9/15 2PM	指定なし	9/30	9/8 3PM	9/30 00AM	9/15 2PM	9/30 00AM
9/15 2PM	9/12 2PM	指定なし	9/12 2PM	10/12 00AM	9/15 2PM	10/15 00AM
9/15 2PM	9/12 2PM	9/30	9/12 2PM	9/30 00AM	9/15 2PM	9/30 00AM
9/15 2PM	9/6 2PM	指定なし	9/8 3PM	10/15 00AM	9/15 2PM	10/15 00AM
9/15 2PM	9/6 2PM	9/30	9/8 3PM	9/30 00AM	9/15 2PM	9/30 00AM

表 22. 日付範囲の計算結果例 (続き)

配達指定日 (Requested Delivery Date)	要求され た出荷日	キャンセ ル日	初期出荷開 始 (Initial ship start)	初期出荷終 了 (Initial ship end)	初期配達開 始日 (Initial delivery start date)	初期配達終了 日 (Initial delivery end date)
9/6 2PM	指定なし	指定なし	9/8 3PM	10/8 00AM	9/8 3PM	10/8 00AM
9/6 2PM	指定なし	9/30	9/8 3PM	9/30 00AM	9/8 3PM	9/30 00AM

**出荷カレンダー:** ノードは、出荷カレンダーを設定したり、あるいはその主要エンタープライズのカレンダーを継承したりすることができます。出荷が確実にノードの作業時間内でのみスケジュールされるように、このカレンダーが出荷のスケジュールに使用されます。ノードで出荷カレンダーが設定されていない場合、Sterling Selling and Fulfillment Foundation は、「週 7 日、24 時間」操業と見なし、他のパラメーターに基づいて出荷をスケジュールします。

**最小通知時間:** 最小通知時間は、オーダーがノードでスケジュールされた後、その出荷に要する最小営業時間数を表します。一部のノード (または外部サプライヤー) は、オーダーを出荷するための出荷通知を受領後、2 から 3 日を必要とする場合があります。また、通知の受領後 2 時間以内でオーダーの出荷が可能なノードもあります。このパラメーターによって、オーダーの約束時点で予定が適正に設定されることが保証されます。このパラメーターは、出荷ノードとアイテム・レベルの両方で指定できます。現在時間または次の使用可能なシフトを基準にして計算された 2 つの値の最大値を使用して、通知時間が判定されます。出荷カレンダーのパラメーターと結合することによって、実行可能な正確なオーダー約束を行うことができます。

例えば、最小通知時間が 2 時間に設定され、出荷ノードの営業時間が 8 時から午後 5 時の場合、その日のうちに出荷するためにはオーダーを午後 3 時までスケジュールする必要があります。午後 3 時を過ぎた場合は出荷が翌日にスケジュールされる必要があります。

オーダーのソーシング先であるサプライヤーから在庫情報を入手できない場合は、このパラメーターを使用してリード・タイムを表すことができます。少なくとも、現在時刻からの最小処理時間と同等であることを保証するように予定出荷日が計算され、約束の遵守が可能であることが保証されます。

**通知スケジュール:** ノードでは、オーダーの出荷の通知を受け取る特定の日時を構成できます。例えば、月曜日から木曜日の午前 9 時のみ通知を受け取るように、ノードを構成できます。

これは、ノードが通知の受け取り後 2 時間で出荷できる能力があったとしても、金曜日には通知を受け取れないため、通知時間に影響します。この通知スケジュールでは、金曜日の通知は強制的に翌週の月曜日の午前 9 時の受け取りになります。

**サービス・レベル (Level of Service):** ビジネスについてさまざまなサービス・レベルをセットアップできます。まず、ハブ・レベルでサービス・レベルを定義し、その後、ノード・レベルでサービス・レベルの通知時間とスケジュールを構成します。Sterling Selling and Fulfillment Foundation のデフォルトでは、標準的なサービ

ス・レベル (Level of Service) が提供されます。例えば、ビジネスのサービス・レベル (Levels of Service) に標準とプレミアムのサービスをセットアップするには、まず、ハブ・レベルでプレミアム・サービス・レベル (Level of Service) を定義し、その後、標準オーダーの最小通知時間 10 時間とプレミアム・オーダーの最小通知時間 5 時間を構成します。

サービス・レベル (Level of Service) は、オーダー・ヘッダーとオーダー明細のレベルで指定できます。サービス・レベル (Levels of Service) が構成されていないノードでサービス・レベル (Level of Service) が要求された場合、アイテムの通知時間とノードの通知時間およびスケジュールが使用されます。ただし、ノードでサービス・レベル (Level of Service) が構成されているときは、ノードのサービス・レベル (Level of Service) の通知時間およびスケジュールのみが使用されます。

Sterling Selling and Fulfillment Foundation は、オーダーのサービス・レベル (Level of Service) に基づいて、オーダーのヘッダー/明細を検索して通知スケジュールを判別し、特定された通知スケジュールを使用して予定出荷日を計算します。同一リリースにサービス・レベル (Levels of Service) が異なる複数のオーダー明細が存在することはなく、バンドル・アイテムには常に同一のサービス・レベル (Level of Service) が設定されます。

サービス・レベル (Level of Service) は、配達および提供サービスでも指定できます。例えば、顧客が配達/取り付けでサービス・レベル (Level of Service) を要求する場合があります。この例では、プレミアム・サービスの最小通知時間が標準サービスの最小通知時間より短いため、予約がより早くなります。Sterling Selling and Fulfillment Foundation では、作業オーダーの全明細 (製品および配達) に同じサービス・レベル (Level of Service) が割り当てられることが保証されます。

**運送会社ピックアップ・スケジュール:** ノードの運送会社ピックアップ・スケジュールによって、運送会社がノードから出荷をピックアップする日時が決定されます。運送会社は、さまざまな時期 (休日や夏季休暇など) のスケジュールを保有でき、スケジュールには、日時に関するオーバーライド (休日など) を含めることができます。Sterling Selling and Fulfillment Foundation は、出荷日を計算する際に、ノードの異なる運送会社のピックアップ・スケジュールを参照します。

例えば、陸路サービスが月曜日、水曜日、および金曜日の午後 3:00 にピックアップする場合、月曜日の午後 12:00 に出されたオーダーは、陸路サービスを使用すると月曜日の午後 3:00 に出荷されます。月曜日の午後 3:30 に出された 2 番目のオーダーは、水曜日の午後 3:00 まで出荷されません。2 番目のオーダーの場合、別の運送会社が水曜日の午後 3:00 より早く配達することができれば、その運送会社が出荷をピックアップすることがあります。

ノードの最小通知時間も、運送会社が出荷をピックアップする時期に影響を与えます。例えば、上記の例のノードでオーダーの出荷前に 4 時間の通知時間が必要な場合、月曜日の午後 12:00 にオーダーが出されると、そのオーダーは陸路サービスを使用して水曜日の午後 3:00 に出荷されます。月曜日の午後 3:30 に出された別のオーダーも、水曜日の午後 3:00 に出荷されます。

**運送会社保留時間:** 運送会社保留時間は、有効な配達日に配達できない配達をサービスの運送会社が保留する日数を指定します。このパラメーターによって、適切な出荷日が運送会社に設定され、指定された日数を超えて配達を保留することができなくなります。

例えば、Sterling Selling and Fulfillment Foundation は、保留日数が 1 日で、土曜日と日曜日に配達を行わない運送会社に対しては、土曜日を出荷日に設定しません。しかし、この例では、日曜日を出荷日に設定することはできます。運送会社は、日曜日に配達を保留し、月曜日に配達を行うためです。

**出荷時刻としてシフト終了時刻を使用:** このパラメーターは、出荷日の時刻コンポーネントを計算するために使用されます。製品が特定の日に出荷されることを予定されている場合、このパラメーターを「Y」に設定することにより、時刻をその日の「次のシフトの終了 (end of next shift)」時刻として選択できます。このパラメーターが「N」に設定された場合、時刻コンポーネントは、「次の作業」時刻として設定されます。

以下の例では、正しいスケジューリング日を計算するためにこれらの設定をどのように使用するかを説明します。

#### シフト終了時刻の出荷時刻、例 1

以下のシナリオを考えると、オーダーは、以下の表に示されているとおりに出荷するようスケジューリングされます。

- シアトルの配送センター (distribution center) は、週 6 日間作業します (日曜日が休日)。
- 土曜日の作業時間は、午前 8 時から正午までで、その他のすべての曜日の作業時間は、午前 8 時から午後 4 時と午後 4 時から午後 8 時の 2 シフトに指定されます。
- 最小通知時間 = 2 時間です。
- 製品の在庫状況は「手持ちあり」です。
- 「出荷オーダーの要求」は「できる限り早く」です。

#### 出荷時刻のスケジュール (例 1)

現在の時刻	「シフト終了時刻を使用」の設定	オーダーが出荷されるスケジュール
金曜日の午後 1 時	あり	同日の午後 4 時
金曜日の午後 1 時	なし	同日の午後 3 時
金曜日の午後 3 時	あり	同日の午後 8 時
金曜日の午後 3 時	なし	同日の午後 5 時
金曜日の午後 6:01	あり	土曜日の正午
金曜日の午後 6:01	なし	土曜日の午前 8:01
土曜日の午後 11:59	あり	月曜日の午後 4 時
土曜日の午後 11:59	なし	月曜日の午前 9:59
月曜日の午前 7 時	あり	月曜日の午後 4 時
月曜日の午前 7 時	なし	月曜日の午前 10 時

## シフト終了時刻の出荷時刻、例 2

以下のシナリオを考えると、オーダーは、以下の表に示されているとおりに出荷するようスケジューリングされます。

- ワシントンの配送センター (distribution center) は、週 5 日間作業します (土曜日と日曜日が非作業日)。
- すべての作業時間は、午前 8 時から午後 5 時までです。
- 最小通知時間 = 3 日間 (72 時間) です。
- 製品の在庫状況は「手持ちあり」です。
- 「出荷オーダーの要求」は「できる限り早く」です。

### 出荷時刻のスケジュール (例 2)

現在の時刻	「シフト終了時刻を使用」の設定	オーダーが出荷されるスケジュール
金曜日の午後 1 時	あり	水曜日の午後 5 時
金曜日の午後 1 時	なし	水曜日の午後 1 時
月曜日の午後 1 時	あり	木曜日の午後 5 時
月曜日の午後 1 時	なし	木曜日の午後 1 時

**2 つのノード間の事前定義転送スケジュールの使用:** 受入ノードおよび出荷ノードに事前定義転送スケジュールがある場合、転送スケジュールは、正しい予定出荷日を決断するために使用されます。転送スケジュールで、出荷ノードから受入ノードに出荷する曜日を指定できます。Sterling Selling and Fulfillment Foundation では、転送スケジュールで指定された曜日にのみ受入ノードで出荷を行うことができると想定しています。正しい出荷日を見つけるために、最小通知時間のほかに、この制約が追加されます。時間計算は変更されません。

出荷ノードから受入ノードへの出荷に要するデフォルトの通過時間も指定できます。このデフォルトの時間は、特定の曜日に指定された通過時間によってオーバーライドできます。

『事前定義転送スケジュールを使用した転送出荷』は、事前定義転送スケジュールを使用した場合の出荷予定日時を示しています。

### 事前定義転送スケジュールを使用した転送出荷

以下のシナリオを考えると、オーダーは、表 8-6、「事前定義転送スケジュールを使用した出荷時刻」に示されているとおりに出荷するようスケジューリングされます。

- 出荷は、シアトルの配送センター (distribution center) からオレゴン州のポートランドにある配送センター (distribution center) へ行われます。
- シアトルの配送センター (distribution center) は、週 6 日間作業します (日曜日が休日)。
- 土曜日の作業時間は、午前 8 時から正午までで、その他のすべての曜日の作業時間は、午前 8 時から午後 4 時と午後 4 時から午後 8 時の 2 シフトに指定されます。

- 最小通知時間 = 2 時間です。
- 転送スケジュールは、シアトルの配送センター (distribution center) からポートランドの配送センター (distribution center) に毎週月曜日と木曜日に行われるように 2 つの配送センター (distribution center) の間で定義されています。
- 製品の在庫状況は「手持ちあり」です。
- 「出荷オーダーの要求」は「できる限り早く」です。

事前定義転送スケジュールを使用した出荷時刻

現在の時刻	「シフト終了時刻を使用」の設定	オーダーが出荷されるスケジュール
金曜日の午後 1 時	あり	次の月曜日の午後 4 時
金曜日の午後 1 時	なし	次の月曜日の午前 8 時
月曜日の午後 5 時	あり	同日の午後 8 時
月曜日の午後 5 時	なし	同日の午後 7 時
月曜日の午後 7 時	あり	水曜日の午後 8 時
月曜日の午後 7 時	なし	水曜日の午前 8 時

**希望日の同期:** 希望日は、出荷指定日、配達日、およびキャンセル日と同期されます。日付は、オーダー・ヘッダーまたはオーダー明細で同期できます。ヘッダー・レベルのオーダー日 (yfs\_order\_dates) は、オーダー・ヘッダー・レコードの日付だけでなく、オーダー全体のスケジュール・レコードの影響も受けます。明細レベルのオーダー日 (yfs\_order\_dates) は、オーダー明細レコードの日付だけでなく、その明細のスケジュール・レコードの影響も受けます。

オーダー日テーブルがオーダー明細で変更された場合は、オーダー明細の日付は次のように同期されます。

- OrderLine.RequestShipDate=MinShipDate.Requested
- OrderLine.RequestedDeliveryDate=MinDeliveryDate.Requested
- OrderLine.CancelDate=MaxDeliveryDate.Requested

## 配達日の計算

配達日は、出荷日が分かると計算されます。配達日を計算する場合は、以下の要因が考慮されます。

- 出荷元ロケーションと出荷先ロケーションとの間の転送時間。
- 2 つのロケーション間の、事前定義済み転送スケジュール
- 曜日配達は、選択されたサービスに応じて行われます。例えば、Ground サービスでは、月曜日から金曜日までのみ配達を行います。土曜日配達では、土曜日にも配達を行うことができます。Express 配達は、1 週間の 7 日のすべての曜日に行うことができます。
- 曜日配達は、サービスの異なる配達スケジュールと転送スケジュールに基づいて行われます。例えば、Ground サービスは、その定期的スケジュール中は月曜日から金曜日まで配達を行い、また、休日スケジュール中は月曜日から日曜日まで配達を行います。

**初期配達日付範囲 (Initial Delivery Date Range):** Sterling Selling and Fulfillment Foundation では、顧客希望日に基づいて製品を配達できる場合に初期配達日付範囲を計算します。

初期配達日付範囲の下限は、ASAP オーダーの現在時刻および将来のオーダーの配達希望時刻として設定されます。

この範囲の上限は、関連付けられたスケジューリング・ルールから、下限 + 「最大許容出荷遅延日数」として設定されます。

**通過時間の決定:** 通過時間は、製品が出荷ロケーションから出荷された時点で、それをその出荷先ロケーションに配達するために必要な時間として定義されます。

2 つのロケーション (出荷ロケーションおよび受入ロケーション) が転送ノードとして定義されている場合、Sterling Selling and Fulfillment Foundation は、その 2 つのロケーション間での転送スケジュール・セットアップを調べます。そのようなセットアップが存在する場合は、通過時間は転送スケジュールからピックアップされます。

出荷ロケーションと受入ロケーションの間に転送スケジュール・セットアップが存在しない場合、通過時間は、外部プログラムか、または Sterling Selling and Fulfillment Foundation のデフォルト・ロジックを使用して計算されます。

- `getDeliveryLeadTime` 外部プログラムを使用して通過時間を決定することができます。この外部プログラムは、Sterling Selling and Fulfillment Foundation が通過時間を計算する必要があるときに、必ず呼び出されます。この外部プログラムを実装しない場合は、通過時間を計算するために、Sterling Selling and Fulfillment Foundation が以下の 2 つのモードのいずれかで動作するようにセットアップすることができます。
- Sterling Selling and Fulfillment Foundation のデフォルト・ロジックを使用した場合、通過時間は次のように計算されます。

A fixed unit of time + (the distance between ship-to and shipping location/average distance per day). Distance is calculated using the longitude and latitude definitions of the two zip codes involved.

**注:** ロケーション (出荷先ロケーションと出荷ロケーション) の郵便番号が提供されない場合は、ロケーションの市区町村、都道府県、および国別情報を使用して、距離を計算します。市区町村に複数の郵便番号がある状態では、データベースから最初に返された郵便番号を使用して距離を見積もります。

- 2 つのパラメーター「時間の固定単位 (fixed unit of time)」および「平均距離/日 (average distance/day)」は、別のエンタープライズ・レベル構成に基づいて異なる構成からピックアップされます。すべてのエンタープライズが、「通過時間計算モード (Transit time calculation mode)」パラメーターを以下のいずれかとして指定できます。
  - 基本モード - 通過時間の計算では、通過時間に使用される運送会社およびサービスは考慮されません。2 つのパラメーターは、エンタープライズ・レベルで構成済みとして使用されます。
  - 拡張モード - 通過時間の計算は、運送会社と運送会社サービスの組み合わせレベル・セットアップに基づいて行われます。2 つのパラメーターは、組み合わせレベルで構成済みのもので使用されます。運送会社と運送会社サービス

の組み合わせのデータ・レコードがデータベースに存在しない場合は、Sterling Selling and Fulfillment Foundation は、自動的に基本モードに切り替わります。

- サービス・モードに基づく - 通過時間の計算パラメーターは、サービス・レコードからピックアップされます。このモードでは、運送会社は役割を果たしません。サービスについてセットアップが存在しない場合は、Sterling Selling and Fulfillment Foundation は、デフォルトとして自動的に基本モードになります。

**曜日ベースの配達をサービス・レベルで考慮する:** 配達を行うことができる特定の曜日を、各サービス・レベルで指定することができます。例えば、速配サービスを、週 7 日のすべての日に配達できるようにセットアップし、通常サービスを、月曜日から金曜日のみに配達できるようにセットアップすることができます。

また、運送会社が出荷を転送する特定の曜日を、各サービス・レベルで指定することもできます。運送会社の転送スケジュールは、出荷の輸送時間に影響を与える可能性があります。例えば、金曜日の出荷に対して計算された輸送時間が 48 時間で、運送会社が土曜日と日曜日には転送を行わない場合、出荷の配達は火曜までに完了しません。

出荷日に対する配達日を計算する場合、出荷日に輸送時間を加算した日付以降が配達日として計算されます。配達日は、選択されたサービスについて、週内の有効な日でなければなりません。

例えば、2 つのロケーション間の輸送時間が 48 時間で計算され、配達に「陸路」サービスが選択され、月曜日から金曜日に配達が行われるように構成されているとします。このサービスは、月曜日から日曜日までの間に出荷を転送します。この例では、配達日は、木曜日または金曜日に出荷された場合、配達日は月曜日になります。

ただし、月曜日から金曜日までの間にのみ、出荷の転送が行われるようにサービスが構成されている場合、木曜日に出荷された場合の配達日は月曜日になり、金曜日に出荷された場合の配達日は火曜日になります。

## 製品調達の予定出荷日の計算

前に説明した日付計算に加えて、製品調達で、調達ノードの出荷カレンダーと受入カレンダー、およびノードの受入処理時間を考慮したオーダーの予定出荷日 (ESD) です。

例えば (185 ページの図 34 の説明を参照)、アイテムが Nashua 倉庫から Boston 倉庫に条件付きで調達されるとします。



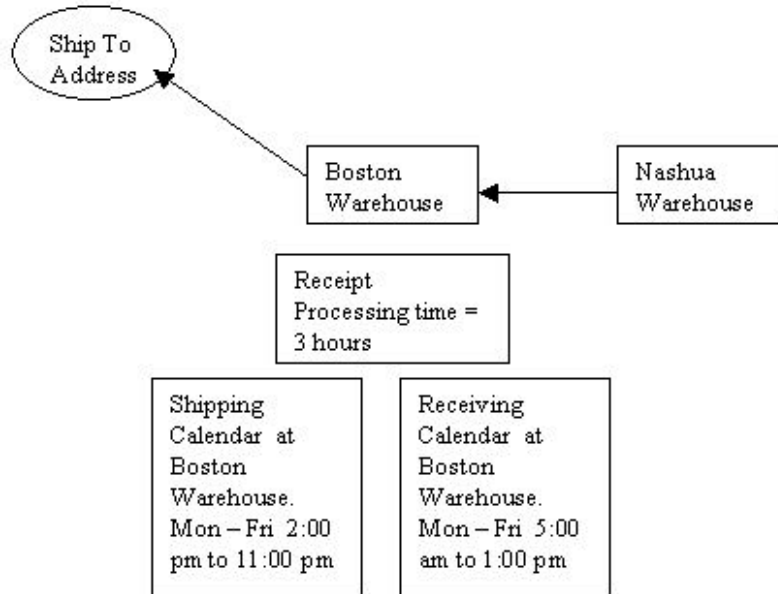


図 34. 予定出荷日の例

- Boston 倉庫の予定配達日 (EDD) は 2003/10/29 1:00 PM です。
- Boston 倉庫への予定配達日 (EDD) は、前に説明した日付計算に基づいて計算されます。
- オーダーの予定出荷日 (ESD) は以下のように計算されます。
  - Boston 倉庫での予定受入日は 2003/10/30 5:00 AM です。(受入カレンダーに基づいています。Boston 倉庫は、月曜日から金曜日までの 5:00 AM から 1:00 PM に、受入を行います)。
  - Boston 倉庫での予定受入 (完了) 日は 2003/10/30 8:00 AM です。(2003/10/30 5:00 AM に 3 時間の受入処理時間を加算)。受入処理時間は、Boston 倉庫の受入カレンダーの作業時間に基づいて追加されます。
  - Boston 倉庫からの予定出荷日は 2003/10/30 2:00 PM です (出荷カレンダーに基づいて、「シフト終了時刻」が選択されていないことを前提として計算されています。Boston 倉庫は、月曜日から金曜日までの 2:00 PM と 1:00 PM の間に、出荷を行います)。
  - オーダーの予定出荷日 (ESD) は 2003/10/30 2:00 PM と計算されます。

予定受入日および予定入庫日は、データベースに格納されません。これらは、この例のためにのみ使用されています。

### フォワーディングの予定出荷日の計算

予定出荷日は、フォワーディング・ノードおよび出荷ノードに基づいて計算されます。上記の例 (184 ページの『製品調達の予定出荷日の計算』) で説明したように、日付計算は、予定受入日の計算にのみ変更がありますが、他は製品調達の場合と同じです。この場合、Boston 倉庫がフォワーディング・ノードとみなされ、フォワーディングの受入処理時間は、0.5 時間です。したがって、フォワーディングの Boston 倉庫での予定受入 (完了) 日は、2003/10/30、5:30 AM (2003/10/30 5:00 AM にフォワーディングの受入処理時間の 0.5 時間を加算) になります。

## 日付同期

日付は、オーダー明細またはオーダー・ヘッダーの日付に変更があるとき、およびマスター・オーダー日付とオーダー明細およびスケジュールの日付との間で日付を同期するルールが構成されるときに、同期されます。出荷指定日と出荷予定日に基づいて、オーダー明細およびオーダー・ヘッダーの両方について、以下の 4 つのオーダー日付が作成されます。

- MinShipDate
- MaxShipDate
- MinDeliveryDate
- MaxDeliveryDate

どのオーダー日付にも、予定日付、希望日、実際の日付、およびコミット日付があります。これらのオーダー日付は、オーダー・ヘッダーか、またはオーダー明細レベルのどちらかで指定されます。オーダー日付の希望日は、出荷指定日、配達指定日、およびキャンセル指定日と同期されます。

明細レベルのオーダー予定日は、出荷予定日、およびオーダー明細スケジュールの配達予定日と同期されます。オーダー・ヘッダーのオーダー予定日は、すべてのオーダー明細の出荷予定日および配達予定日と同期されます。

ヘッダー・レベルのオーダー日付は、オーダー・ヘッダー・レコード上の日付の影響を受け、全オーダーのスケジュール・レコードによる影響も受けます。明細レベルのオーダー日付は、オーダー明細レコード上の日付の影響を受け、その明細のスケジュール・レコードによる影響も受けます。

### 希望日

オーダーの希望日は、出荷指定日、配達日、およびキャンセル日と同期されます。希望日が同期されるのは、createOrder および changeOrder API が呼び出されたときです。

オーダー希望日テーブルがオーダー明細で変更された場合は、オーダー明細の日付は次のように同期されます。

- OrderLine.RequestShipDate=MinShipDate.Requested
- OrderLine.RequestedDeliveryDate=MinDeliveryDate.Requested
- OrderLine.CancelDate=MaxDeliveryDate.Requested

出荷指定日、配達指定日、またはキャンセル指定日が明細レベルで変更された場合は、オーダー希望日は次のように同期されます。

MinShipDate.RequestedDate = RequestedShipDate

MaxShipDate.RequestedDate = RequestedCancelDate

MinDeliveryDate.RequestedDate= RequestedDeliveryDate

MaxDeliveryDate.RequestedDate = RequestedCancelDate.

オーダー・ヘッダーの同期も同じロジックに従います。

## 予定日付

オーダーの予定日付は、オーダー・スケジュールと同期されます。予定日付は、オーダー明細が作成または変更された時点で同期されます。オーダー・スケジュールに変更があると、出荷日および配達日は、次のように同期されます。

MinShipDate= スケジュールにあるすべての出荷日のうちの最小のもの (Min.)

MaxShipDate= スケジュールにあるすべての出荷日のうちの最大のもの (Max.)

MinDeliveryDate= スケジュールにあるすべての配達日のうちの最小のもの (Min.)

MaxDeliveryDate= スケジュールにあるすべての配達日のうちの最大のもの (Max.)

オーダーの予定日付に変更があった場合、オーダー日付はオーダー・スケジュールと同期されません。

## ソーシング・モデルの影響

これまでのところ、ソーシングに関する考察は、製品在庫状況を決定するために、ソーシング・ロケーションおよび予定日付のシステム決定に関連付けられています。ただし、製品のソーシングに使用されるソーシング・モデルに応じて、以下の場合があります。

- 在庫を別の方法で検索しなければならない場合がある
- 出荷通知を別の方法で処理しなければならない場合がある
- 在庫供給および需要の更新を別の方法で行わなければならない場合がある

このセクションでは、実装可能な各種ソーシング・モデルについて、また、Sterling Selling and Fulfillment Foundation がそれぞれのモデルをどのように処理するかについて説明します。

### ソーシング・モデル A

お客様の組織（または「3PL ロケーション」としてマーク付けられた同じ法人またはノード）が所有するロケーションの 1 つで在庫から出荷する場合、Sterling Selling and Fulfillment Foundationは、以下のように在庫検索、出荷通知、および在庫更新を処理します。

- 在庫検索
  - セラーの在庫組織の在庫状況が検索されます
- 出荷指示通知
  - 出荷指示は、オーダー・リリースを介して行われます
- 在庫更新
  - オーダーのステータス移動を介して制御されます

### ソーシング・モデル B

パートナー（出荷ロケーションは別の法人に属しており、「3PL ロケーション」としてのマーク付けなし）から出荷をドロップする場合、Sterling Selling and Fulfillment Foundationは、以下のように在庫検索、出荷通知、および在庫更新を処理します。

パートナー組織（ベンダー）は購入オーダーが必要です。

- 在庫検索
  - パートナーからの在庫情報が利用可能な場合は、以下のようになります。
    - パートナーの在庫組織がセラーと同じ場合、セラーの在庫組織の在庫状況が検索されます。
    - パートナー組織が在庫を外部で保持している場合、在庫情報を取り込むのに外部プログラムが呼び出されます。
    - パートナー組織が別の在庫組織の一部として Sterling Selling and Fulfillment Foundation 内に在庫を保持している場合、その在庫にはアクセスできないことに注意してください。
  - パートナーからの在庫情報が利用「不能」な場合は、以下のようになります。
    - 在庫チェックは行われません。出荷日の計算については、178 ページの『最小通知時間』を参照してください。
- 出荷指示通知
  - 新しい「ドロップ出荷」連鎖オーダーが作成されます。この連鎖オーダーは、「作成済み」ステータスで作成されます。通知は、パイプラインの該当するステップで作成されたオーダーに基づいてベンダーへ送信できます。
- 在庫更新
  - すべてのバイヤー更新は、親オーダーのみを介して制御されます。
  - セラー供給および需要の更新は、親オーダーから行われます。
- パートナー組織では、別個の購入オーダーは不要です

このタイプの組織に対してサポートされる唯一の在庫モデルは、「Sterling Selling and Fulfillment Foundation の一部として管理されている在庫」です。

- 在庫検索
  - 在庫は、セラーの在庫組織内で検索されます。
- 出荷指示通知
  - 出荷指示は、オーダー・リリースを介して行われます。
- 在庫更新
  - オーダーのステータス移動を介して制御されます

## ソーシング・モデル C

パートナーから在庫を調達し、お客様自身のロケーションから出荷する場合、Sterling Selling and Fulfillment Foundation は、以下のように在庫検索、出荷通知、および在庫更新を処理します。

- 在庫検索
  - パートナーからの在庫情報が利用可能な場合は、以下のようになります。
    - パートナーの在庫組織がセラーと同じ場合、在庫は、セラーの在庫組織内で検索されます。
    - パートナー組織が在庫を外部で保持している場合、在庫情報を取り込むのに外部プログラムが呼び出されます。

- パートナー組織が別の在庫組織の一部として Sterling Selling and Fulfillment Foundation 内に在庫を保持している場合、その在庫にはアクセスできないことに注意してください。
- パートナーからの在庫情報が利用「不能」な場合は、以下のようになります。
  - 在庫チェックは行われません。

出荷日の計算については、199 ページの『調達購入オーダー』を参照してください。Sterling Selling and Fulfillment Foundation には、最終出荷日を計算するために調達ロケーションと出荷ロケーションとの間の通過時間が含まれます。Sterling Selling and Fulfillment Foundation では、最終出荷ノードに対して、受入カレンダーと受入処理時間も考慮されます。

- 出荷指示通知
  - オーダーのスケジュール時点で、連鎖オーダーを出荷する調達が即時に作成されます。この連鎖オーダーは、「予約済み」ステータスで作成されます。
  - 通知は、パイプラインの該当するステップで作成されたオーダーに基づいてベンダーへ送信できます。「事前通知日数」設定を介して通知時刻を制御できるリリース・プロセスを、別途構成することができます。
- 在庫更新
  - 在庫更新は、バイヤーとサプライヤーの両方について、両方のオーダーに対して行われます。パートナー組織が在庫情報を一切保持しない場合、調達オーダーに合わせたセラー在庫更新は実行されません。

## ソーシング・モデル D

お客様が所有する別のロケーションから在庫を調達し、お客様所有の別のロケーションから出荷する場合、Sterling Selling and Fulfillment Foundationは、以下のように在庫検索、出荷通知、および在庫更新を処理します。

- 在庫検索
  - 在庫は、セラーの在庫組織内で検索されます。

出荷日の計算については、199 ページの『調達転送オーダー』を参照してください。Sterling Selling and Fulfillment Foundation には、最終出荷日を計算するために調達ロケーションと出荷ロケーションとの間の通過時間が含まれます。

- 出荷指示通知
  - オーダーのスケジュール時点で、連鎖オーダーを出荷する調達が即時に作成されます。この連鎖オーダーは、「予約済み」ステータスで作成されます。この連鎖オーダーに使用される伝票種別 (document type) は、別個の伝票種別構成を介して制御され、特に指定しない限り、「転送オーダー」に設定されます。
  - 通知は、パイプラインの該当するステップで作成されたオーダーに基づいてノードへ送信できます。「事前通知日数」設定を介して通知時刻を制御できるリリース・プロセスを、別途構成することができます。
- 在庫更新
  - 在庫更新は、バイヤーとサプライヤーの両方について、両方のオーダーに対して行われます。

## オーダーまたはオーダー明細の出荷のスケジュールリング

Sterling Selling and Fulfillment Foundation には、さまざまなレベルで次のパラメーターが用意されており、特定の時間まで出荷のオーダーまたはオーダー明細をスケジュールするために、またはスケジュール「しない」ために必要なコントロールを確実に持つことができます。

- ステータスのコントロール
- 最も早い予定日

### ステータスのコントロール

これはオーダー明細レベルのコントロールで、オーダー明細のステータスはスケジュールリング・トランザクションで使用可能なステータスとして構成する必要があります。

### 最も早い予定日

このコントロールを使用して、将来のオーダーが事前に過剰にスケジュールされないようにし、それによって、現在または「可能な限り早く」のオーダーで使用可能な在庫がブロックされるのを防ぎます。

オーダー明細をスケジュールできる場合の最も早い日は、次のように計算されます。

- 日付はオーダー明細の作成時に指定できます。指定した場合、その日付が最も早い予定日として使用されます。
- 日付を指定しなかった場合は、Sterling Selling and Fulfillment Foundation が以下のように適格な予定日を計算します。
  - 事前にスケジュールする最大日数を決定するための、以下の値のうち最小のもの
    - スケジュールリング・ルール・レベルのパラメーター「リード・タイムのスケジュール」。
    - ノード・レベルのパラメーター「スケジュール可能な最大前日数」。このパラメーターが使用されるのは、ノードがオーダー明細で事前指定されていた場合のみです。ノードが事前指定されていない場合は、ソーシング・ステップによって最も早い予定日が再決定され、その時点でオーダー明細をスケジュールしないように決定できます。
  - 続いて、Sterling Selling and Fulfillment Foundation はこの日付をオーダー明細の「出荷日」として計算します (以前に決定した事前スケジュール可能な最大前日数)。オーダーで指定された出荷指定日は「出荷日」として使用されます。特定の出荷指定日が指定されていない場合は、Sterling Selling and Fulfillment Foundation が、指定配達日および最も予測される通過時間計算に基づいて、概算の出荷日を計算します。出荷元の実際のノードが不明なため、この概算の出荷日の計算は不正確な場合があることに注意してください。「スケジュール可能な最大前日数」パラメーターは、スケジュールリング操作がオーダーに対して十分余裕を持って実行されるように設定してください。

## オーダー明細の出荷のスケジューリング

次のようなシナリオの場合、最も早い予定日は、表 23 に示されている方法で計算されます。

- 2003 年 6 月 29 日に製品 ITEM1 が含まれるオーダー明細を作成します。
- この明細でノードを NODE1 として事前指定しています。

パラメーターは次のように設定されます。

- スケジューリング・ルール - リード・タイムのスケジュール = 30 日
- NODE1 - スケジュール可能な最大前日数 = 10 日

表 23. 最も早い予定日の計算

出荷指定日...	最も早い予定日の計算方法...
2003 年 7 月 15 日	2003 年 7 月 5 日。オーダーはそのときまでスケジュールされません。
2003 年 6 月 30 日	2003 年 6 月 29 日。オーダーはこのコントロールのパーспекティブからスケジューリングの準備ができています。
指定されていないが、指定配達日は 2003 年 7 月 15 日として指定されており、初期通過時間は 3 日と計算されている	2003 年 7 月 2 日。これは、「出荷日」が 2003 年 7 月 12 日として計算されるため、このため 7 月 2 日という計算になります。
指定されておらず、指定配達日も指定されていない	2003 年 6 月 29 日。

これらのどの例でも、オーダー明細のスケジューリングは、日付の計算後に限って考慮されます。作成と同時にオーダー明細をスケジュールしたい場合は、最大スケジューリング日数を 999 などの高い値に設定する必要があります。

## ソーシング決定後のスケジューリングのスキップ

オーダー作成時に出荷ロケーションがわからない場合、最も早い予定日の計算は、ノード・レベル・パラメーターを考慮しません。オーダー明細に対してソーシングが実行されると、ノードは内部ノードまたは外部サプライヤーになります。これが外部サプライヤーの場合、すぐにオーダーをスケジューリングする必要があります。また、あまり早い時期に独自のノードにオーダーをスケジューリングせずに、未決定の状態のままにしておく場合もあります。スケジューリング・プロセスは、オーダー明細が最初の計算に基づくスケジューリングに対して考慮されても、明細の調達元のノードでの設定により、その決定が早すぎると示唆された場合にはオーダー明細をスケジューリングしません。

場合によっては、依存関係グループ内の特定のオーダー明細のスケジューリングがスキップされ、残りのオーダー明細がスケジューリングされるときに、この理由から依存関係グループの一部がスケジューリングされない場合があることに注意してください。この状態を受け入れられない場合は、ノード・レベルで「最大スケジュール日 (Max schedule days)」パラメーターを 999 などの大きい値に設定します。

この概念を説明するために、製品 ITEM1 を含むオーダー明細を 2003 年 6 月 29 日に作成しているとします。このオーダー明細に対してノードを事前指定していないものとします。パラメーターは次のように設定されます。

- スケジューリング・ルール - リード・タイムのスケジュール = 30 日
- NODE1 - 最大スケジュール日 = 10 日
- NODE2 - 最大スケジュール日 = 40 日

オーダー明細の出荷指定日が 2003 年 7 月 31 日に指定されている場合、最も早いスケジュール日は 2003 年 7 月 1 日と計算され、オーダーはその時までスケジュールリングを考慮されません。ここでは、2003 年 7 月 1 日に、このオーダーはスケジュールリングを考慮されます。結果は、以下のいずれかになります。

- Sterling Selling and Fulfillment Foundation がソーシング・ノードを NODE1 に決めた場合。NODE1 の「最大スケジュール日 (Max schedule days)」が 10 日に設定されているため、Sterling Selling and Fulfillment Foundation は、このオーダー明細をスケジュールリングしません。
- Sterling Selling and Fulfillment Foundation がソーシング・ノードを NODE2 に決めた場合。オーダー明細はスケジュールリングされます。

## オーダーおよびオーダー明細がスケジュールリングされない理由

Sterling Selling and Fulfillment Foundation のスケジュールリング機能は、以前に説明した制御に基づいて、スケジュールリングの準備ができていないすべてのオーダー明細を取得します。スケジュールリング機能は、出荷ノード (まだ決まっていない場合) および予定日付を決定し、決定したノードおよび日付に対してオーダーをスケジュールリングします。スケジュールリング機能が何らかの理由で製品を調達できない場合、オーダー明細をバックオーダーします (または、失敗の理由に基づいて現在のステータスのままにします)。Sterling Selling and Fulfillment Foundation は、オーダー明細の「スケジュール失敗理由」を更新します。この理由は、スケジュールリング機能が完了しなかった理由を説明します。スケジュールリング失敗の考えられる理由は、以下のとおりです。

- DATE\_BEYOND\_MAX\_SCHEDULE - (指定または計算された) 出荷日が未来であるため、スケジュールリング・ルール (リード・タイムのスケジュール) またはノード・パラメーター (最大スケジュール・リード日数) では、このような早い時期のスケジュールリングはできません。これらのパラメーターが正しく設定されていない場合、修正し、スケジュールリングを再実行してください。
- FILL\_QTY\_NOT\_REMAINING - オーダー明細に指定の補充数があるが残りの数量が指定の補充数より少ないです。オーダーは補充数が調整されるまでスケジュールリングできません。残数量のオーダー明細をキャンセルするか、または補充数の指定が正しくない場合、これを調整します。
- FILL\_QTY\_NOT\_AVAILABLE - このオーダーについてスケジュールする残りの数量は、補充数量を超えています。まだスケジュールできません。利用不可能な数量がスケジュールリングの準備ができていないステータスになっています。オーダー明細で手動のステータス変更が必要な場合、その変更を実行します。オーダー明細の準備ができると、Sterling Selling and Fulfillment Foundation は、自動的に明細をスケジュールリングします。
- NO\_MORE\_LEFT\_TO\_SCH - オーダー明細には、スケジュールする数量がこれ以上ありません。



- UNAVAIL\_FOR\_SHIP\_COMPLETE - オーダー明細またはオーダー・スケジュールリング・ルールが明細出荷完了またはオーダー出荷完了に対して設定されていて、すべてのオーダー数量がまだスケジュールリングできません。
- SOME\_QTY\_NOT\_AVAIL - 製品明細用 - オーダー明細にこれに関連付けられた配送サービスがあるが、オーダー明細の一部の数量をまだスケジュールリングできません。Sterling Selling and Fulfillment Foundation は、完了した数量がスケジュールリング・トランザクションに利用可能になるまで、オーダー明細をスケジュールリングできません。
- QTY\_NOT\_AVL\_FOR\_SCH\_YET - オーダー明細にスケジュール外の数量がありますが、スケジュールできるユニットはありません。
- APPT\_BEFORE\_CONSTRAINT\_DATE - 配送サービス明細または提供サービス明細にある予定日が、それより前のオーダー明細でスケジュールされた配達日より前の日付になっています。
- APPT\_IN\_THE\_PAST - サービス明細の予定日が、過去の日付です。再度予定する必要があります。
- CANCEL\_DATE\_IN\_THE\_PAST - オーダー明細で指定されているキャンセル日が過去の日付です。オーダー明細をキャンセルするか、キャンセル日を適切な将来の日付に調整してください。
- NEEDS\_AN\_APPT\_TO\_SCH - スケジュールするには、この明細にサービスの予定が必要です。
- NO\_DEL\_SRVC\_ATTACHED - 明細の配達方法が「DEL」(配達) に設定されていますが、このオーダー明細に関連付けられた配送サービス明細がありません。配送サービス明細を関連付けてください。
- PARENT\_NOT\_COMPLETE - オーダー明細がシーケンス明細グループの一部で、このオーダー明細をスケジュールする前に親明細を配達する必要があります。先行の明細が完了した後に、この明細をスケジュールできます。
- PARENT\_NOT\_SCHEDULED - オーダー明細がシーケンス明細グループの一部であって、このオーダー明細をスケジュールする前に親明細をスケジュールする必要があります。先行の明細がスケジュールされていない理由を確認し、その明細に対して修正アクションを実行してください。先行の明細をスケジュールできたら、この明細をスケジュールできます。
- CONFIRM\_ASSIGNMENT\_REJECTED - スケジュール機能がオーダー明細を出荷できるノードを検出しましたが、確認割り当て外部プログラムがこの明細のスケジュールリングを拒否しました。
- PARENT\_LINE\_NOT\_READY\_TO\_SCH - オーダー明細がシーケンス明細グループの一部で、このオーダー明細をスケジュールする前に親明細をスケジュールする必要があります。親明細はまだスケジュールリングできません。したがって、この明細をスケジュールリングできません。先行の明細をスケジュールできない理由を確認し、その明細に対して修正アクションを実行してください。先行の明細のスケジュールリングの準備ができれば、この明細をスケジュールリングできます。
- DEPENDENCY\_CONSTRAINT - オーダー明細が他の明細 (依存グループまたはシーケンス・グループ) の依存関係にあり、依存関係の制約によってこの明細をスケジュールリングできません。

- DELIVERY\_SERVICE\_CONSTRAINT - この明細には配送サービスを利用できません。
- NOT\_ENOUGH\_PRODUCT\_CHOICES - この明細の製品の在庫がありません。在庫状況は、製品を出荷できる日付範囲によって決まります。「遅延期間」を増やすことによってこの制約を取り除けるかどうか、もう一度オーダーの日付を確認してください。
- CAPACITY\_NOT\_AVAILABLE - サービス明細のキャパシティが足りないために、要請された予定スロットが作成できません。
- DATE\_CONSTRAINT - このオーダー明細の許可済みの出荷終了日または配達終了日より前に先行の明細をスケジューリングできません。
- NO\_RESOURCE\_POOL\_FOUND - オーダー明細の地域またはアイテムを提供するリソース・プールがありません。オーダー明細をキャンセルするかまたはその地域を提供する類似のサービスを検索する必要があります。
- NO\_ADDRESS\_SPECIFIED - オーダー明細に出荷先住所が指定されていません。
- NO\_SOURCING\_RULE\_DEFINED - オーダー明細に出荷ノードや分配ルール (distribution rule) が事前指定されておらず、ソーシング・ルールも定義されていません。
- NO\_PRODUCT\_AVAIL - このエラーが表示された配送サービス明細に配達能力はありますが、製品を予定日または予定時刻に入手できません。
- CONFLICTING\_CONSTRAINTS/OTHER\_CONSTRAINTS - これは、「一括出荷」などの矛盾する制約があるが、各明細に異なる出荷ノードが指定されているために、Sterling Selling and Fulfillment Foundation が一連のオーダー明細をスケジューリングできないという稀な状況で発生します。

## オーダーまたはオーダー明細のスケジューリング

Sterling Selling and Fulfillment Foundation は、オーダーをスケジュールするための次のメカニズムを備えています。

- スケジューリングにスケジュール・エージェントを使用する。パイプラインの構成に基づいて、Sterling Selling and Fulfillment Foundation は、適切な時間にスケジュール・エージェントを起動でき、スケジューリング機能が実行されます。
- スケジュール API を使用して、特定のオーダーのスケジューリングを要求する。この API を使用して、特定のオーダーのスケジューリングを要求することができます。Sterling Selling and Fulfillment Foundation はスケジュール・エージェントと基本的に同じ処理を実行します。ただし、スケジューリングのためにピックアップするオーダーのシーケンスを制御するのはユーザーである点が異なります。
- オーダー明細を手動でスケジュールする。この場合、オーダー明細をスケジュールするノードと日付はユーザーが決定します。この API を使用する前に、在庫状況照会 API を使用してください。

## 予約済みのオーダーまたはオーダー明細のスケジューリング

スケジューリング・プロセス中に、本製品は、ノード、代替品、および日付を検索します。ただし、在庫を確認するだけでなく、関連付けられている予約が事前予約済み在庫として使用されます。これらの予約が使用されるのは、予約が予約パラメーターに一致した場合のみです。以下に例を示します。

オーダー明細: Item1、数量 5、Node1

オーダー明細に関連付けられている予約は次のとおりです。

1. Item2、数量 3、Node1
2. Item2、数量 1、Node2

Item2 は Item1 の優先代替で、Node1 は Node2 から調達できます。スケジューリングの実行中、使用可能な数量は、Node1 または Node2 の Item2 の数量 1 だけです。残りの数量は、Node1 および Node2 で事前予約済みとみなされます。優先代替品が構成されていない場合は、Node1 および Node2 で合計 5 台のユニットが使用可能で、予約パラメーターが一致しないため、関連付けられている予約は使用されません。

オーダーに対してスケジュールが実行されると、明細およびヘッダーに関連付けられている予約はすべて削除されます。これは、予約が使用されたかどうかとは関係なく行われます。リリースも同様に機能し、スケジュールを行ってすべての予約を削除します。

### 予約モードのスケジューリング

通常、スケジュールを実行してオーダーをリリース対象としてスケジュールできなかった場合、常にその数量はバックオーダーされます。在庫が使用可能であるのに一部のオーダー制約を満たすことができない場合、スケジューリング・ルールの構成を行うことで在庫を予約できます。例えば、100 ユニットのオーダーを発行し、補充数が 100 だとします。それぞれ 50 ユニットの購入オーダー 2 つが 5 日間隔で入った場合、スケジュールはまず、最初の購入オーダーに対して 50 ユニットの予約します。続いて、5 日以内に、完全な在庫数量が使用可能になったら、明細全体がスケジュールされます。したがって、部分的に使用可能な在庫を予約することで、スケジュールによって、2 つ目の購入オーダーが到着したら、確実に明細がスケジュールされるようになります。その他の場合、より少ない数量の別のオーダーは最初の購入オーダーを消費できますが、最初のオーダーはバックオーダーされます。

### 配達中のオーダー明細のスケジューリング

配達方法が「配達」に設定されているオーダー明細は、その明細用にサービス作業オーダーが作成済みでない限り、スケジュールできません。

デフォルトでは、サービス作業オーダーが作成済みであれば、予定の確保後、明細をスケジュールできます。予定が確保されていないのにスケジューリングを試行すると、明細は「予約済み」ステータスにドロップされます。

ただし、サービス作業オーダーの予約が確保される前にオーダーをスケジュールできるように Sterling Selling and Fulfillment Foundation を構成することができます。詳しくは、「*Sterling Selling and Fulfillment Foundation* 分散オーダー管理 構成ガイド」を参照してください。サービス作業オーダーについて詳しくは、230 ページの『サービス作業オーダーのタイプ』を参照してください。

**注:** オーダーのスケジュールとリリースを同時に行うよう選択した場合、Applications Manager で許可されているように構成していても、作業オーダーの予約を確保しないと配達オーダー明細はスケジュールされません。

**バンドルのスケジューリング:** バンドルは、コンポーネントと親を持つ複数のオーダー明細として表されます。個々の明細は、オーダー上で、バンドルの親および別のバンドルの子の両方になることができます。各コンポーネントはその親を示します。バンドル親は、バンドルをどのようにスケジュールする必要があるかを示します。制約に設定可能な値は、「一括出荷」、「一括配達」、および「個別出荷」です。この制約はカタログから取得されます。キット比率はスケジューリング時に強制的に適用されます。

このセクションで説明するロジックは、スケジューリング・アクティビティー実行時に考慮されます。

### 出荷制約

出荷制約はカタログから取得されます。カタログが定義されていない場合、バンドルは一括出荷されます。これらの制約は、バンドルのすべてのレベルのコンポーネントに適用されます。

親が「別々に出荷」を示している場合、制約は適用されず、コンポーネントは独立した明細として扱われます。

親が「一括出荷」を示している場合、コンポーネントは同じノードから一括出荷されます。ただし、異なるノードからコンポーネントを調達して、出荷ノードでマージして、1つの出荷として出荷することができます。

親が「一括配達」を示している場合、予定配達日によって出荷スケジュールが決まり、比率内で同じ配達日を達成できるように、コンポーネントの出荷日は延期されます。

### 比率外

バンドルの1つ以上のコンポーネントがキャンセルされた場合、コンポーネントは比率外になります。このシナリオでは、比率外の明細のスケジューリング時に制約「一括出荷」は適用されません。例えば、以下について考えてみます。

スキー・パッケージ - P1

スキー - C1 (数量 2)

ポール - C2 (数量 2)

ヘルメット - C3 (数量 1)

在庫状況:

N1 では 4 つの C1、4 つの C2 が使用可能で、N2 では 2 つの C1、2 つの C2 が使用可能です。

2 つのスキー・パッケージに対してオーダーが作成され、ヘルメット 1 つが手動でキャンセルされました。スケジューリング中に、C1 および C2 は N1 に対して完全にスケジュールされますが、C3 はバックオーダーされます。1 つのヘルメットが N2 で使用可能であれば N2 に対してスケジュールされますが、C1 と C2 は N1 に対してスケジュールされています。同様に、バンドルが 1 つのノードに対してス

スケジュールされていて、コンポーネントの 1 つが手動でバックオーダーに入れられた場合も、「一括出荷」制約は適用されません。したがって、上の例では、N2 で 2 つ目のヘルメットがバックオーダーになった場合、N1 の方が近いため C1 および C2 を N1 にスケジュールします。その結果、スキーとポールはすべて N1 から出荷され、N2 からはヘルメット 1 つが出荷されます。

### 予約

コンポーネントを比率内でスケジュールできず、その比率が必要な場合、いくつかのコンポーネントが予約されます。予約するため、スケジュールリングは 2 回反復して実行されます。1 回目は、すべてのコンポーネントをスケジュールしようとし、スケジュールされていないコンポーネントがもう一度スケジュールリングされます。2 回目の反復では、最初の反復の一部としてすでにスケジュール済みの明細がスケジュールされ、最適化、および「一括出荷」制約と「一括配達」制約の目的でのみ使用されます。例:

スケジュールリングにより、出荷の数が最適化されます。

L1 P1 2 (一括出荷)

L2 C1 キット比率 3 (合計数量 6 (2\*3))

L3 C2 キット比率 2 (合計数量 4 (2\*2))

L4 Item1 (数量 2)

在庫: Item1 2 N1、C1 - 5 N1、5 N2

最初の反復時に、L4 を N1 に対してスケジュールし、L2 および L3 はバックオーダーになります。2 回目の反復で、L2 (数量 5) は N2 ではなく N1 に対してスケジュールされます。これは、N2 と比較した場合に出荷数がより少ないためです。

## 製品出荷の通知

オーダーの出荷準備が完了すると、次の伝票のいずれかがこの情報の連絡に使用されます。

- オーダー・リリース
- ドロップ出荷連鎖オーダー
- 調達転送オーダー
- 調達購入オーダー

### オーダー・リリース

自社所有ノードのいずれか、または別個の連鎖伝票を必要としないノードからの出荷の場合、オーダーのリリースは、出荷指示として送信されます。

スケジュール済みの出荷日を基準にして、Sterling Selling and Fulfillment Foundation によって通知日が計算されます。Sterling Selling and Fulfillment Foundation は、通知日にオーダー・リリースを作成します。このオーダー・リリースが出荷ノードに通知され、出荷プロセスが実行されます。

### 通知日の計算:

通知日は、「事前通知日数」パラメーターおよび「最大作業時間」パラメーターに基づいて計算されます。

事前通知日数は、出荷ノードが、予定出荷日より前に通知するために必要な日数です。このパラメーターは、出荷ノードのカレンダーを考慮しません。例えば、出荷ノードが2日の事前通知日数を必要とし、それらの日数の中の1日が休日に該当した場合、このノードの出荷前の事前通知には実際には3日必要です。

最大作業時間は、出荷ノードが、予定出荷日より前に通知するために必要な作業時間数です。これは、出荷ノードのカレンダーを考慮します。このパラメーターは、通知日の計算時に、出荷ノードが作業時間でない場合を考慮して対応しています。

「事前通知日数」パラメーターおよび「最大作業時間」パラメーターは、以下のよう

- 出荷ノード・レベル - 「事前通知日数」および「最大作業時間」。オーダーを出荷ノードまたはサプライヤーに出荷するための伝達は、予定出荷日前に、最大でこの日数およびこの作業時間に行われます。
- アイテム・レベル - 「事前通知日数」および「最大作業時間」。オーダーを出荷ノードまたはサプライヤーに出荷するための伝達は、予定出荷日前に、最大でこの日数で行われます。
- ATP ルール - 「事前通知日数」。これは、後方互換性のためだけに使用され、アイテム設定と比較されて、2つのうち大きい方の時間が選択されます。2つの中での最大時間が、アイテムの事前通知時間として使用されます。

Sterling Selling and Fulfillment Foundation は、すべての事前通知日数の中の最大日数を、事前通知日数として使用します。同様に、すべての作業時間の中の最大時間を、最大作業時間として使用します。特定の出荷ノードで、より長い通知時間が必要な場合、出荷ノード・レベルでオーバーライドできます。特定のアイテムで特殊な取り扱い、および長い通知時間が必要な場合には、アイテム・レベルでオーバーライドできます。Sterling Selling and Fulfillment Foundation は、最大値を選択するため、適時な伝達が確保されます。

通知日は、すべてサービスの要求レベルに基づいて、予定出荷日 - 最大通知時間 (出荷ノードのカレンダーを含む) - 事前通知日数で計算されます。

通知日は、通知スケジュールによっても影響を受けます。通知スケジュールがオーダーのサービスのレベルで存在する場合、計算された日付より前の通知時間を使用して計算されます。例えば、通知スケジュールで、通知が毎日午後 5:00 に行われるように指定されており、通知日が元々木曜日の午後 3:00 で計算されていた場合、通知スケジュールを考慮すると、新しい通知日は、前日 (水曜日) の午後 5:00 となり、その結果木曜日の午後 3:00 までにノードは通知を受けることができます。

オーダー・リリースは、出荷ノードに伝達できる通知日に作成されます。Sterling Selling and Fulfillment Foundation は、オーダー・リリースを作成する前に、通知日に在庫の状況を再チェックすることに注意してください。これによって、スケジュールリング処理とリリース処理に大きな間隔が空いたとしても、在庫がまだ入手可能な場合にのみ、通知が送信されます。

これらの計算は、直送オーダーおよび調達オーダーの作成の両方に適用されます。

## ドロップ出荷連鎖オーダー

調達機能が、作成される別個の購入ドキュメントを送信しなければならない外部組織のノードを選択した場合、Sterling Selling and Fulfillment Foundation は、「ドロップ出荷連鎖オーダー」を作成します。

このオーダーをベンダーに連絡すべき日付に関して制御がさらに必要な場合は、制御するパイプライン内でプロセスを作成することができます。また、連鎖購入ドキュメントについて明示的なリリース・プロセスを含めることもできます。

## 調達転送オーダー

転送オーダーは、次の場合に Sterling Selling and Fulfillment Foundation で作成されます。

- 顧客への最終出荷点が自社のいずれかのノードの場合 (または 3PL ノードあるいは同じ法人組織所有のノード)
- 出荷ノードに十分な在庫がなく、オーダーを満たすために自社所有の別のノードから補充する必要がある場合

Sterling Selling and Fulfillment Foundation は、次の場合に元のオーダーのスケジュール時点で調達転送オーダーを作成します。

- 出荷ノードと調達元のノードがオーダー明細で渡されている場合、または
- ソーシング・ルールが使用されていて、分配ルール (distribution rules) の代わりに出荷ノードを使用するように構成されている場合

転送オーダーは「予約済み」ステータスで作成されます。転送オーダーをリリースするリリース・エージェントを構成する必要があります。オーダー・リリースは、転送オーダーで伝達用に作成されるため、通知日は 197 ページの『オーダー・リリース』に説明されているように制御できます。

## 調達購入オーダー

購入オーダーは、次の場合に Sterling Selling and Fulfillment Foundation で作成されます。

- 顧客への最終出荷点が自社のいずれかのノードの場合 (または 3PL ノードあるいは同じ法人組織所有のノード)
- 出荷ノードに十分な在庫がなく、外部組織のノードから補充する必要がある場合

Sterling Selling and Fulfillment Foundation は、次の場合に元のオーダーのスケジュール時点で購入オーダーを作成します。

- 出荷ノードと調達元のノードがオーダー明細で渡されている場合、または
- ソーシング・ルールが使用されていて、分配ルール (distribution rules) の代わりに出荷ノードを使用するように構成されている場合

購入オーダーは「スケジュール済み」ステータスで作成されます。リリース・エージェントを構成して購入オーダーをリリースするか、購入オーダーと直接通信できます。

購入オーダーを通じて伝達する場合には、パイプライン構成を使用し、必要に応じて制御できます。

オーダー・リリースが購入オーダーで伝達用に作成される場合は、197 ページの『オーダー・リリース』に説明されているように、通知日をコントロールできます。

## バックオーダーの取り扱い

在庫がない場合、オーダー明細はバックオーダーされます。オーダー明細がバックオーダーされると、通常は使用可能な在庫を検索するためのスケジューリングを待機する必要があります。

ビジネス要件によっては、バックオーダーされた明細を別の方法で処理する必要があります。

以下に例を示します。

- 明細は、フルフィルメントに単一のノードを使用する必要があります。一部のみバックオーダーされた場合、バックオーダーされた数量を異なる明細に分割し、実行によってスケジュールされた数量を前に移動しなければならない場合があります。
- バックオーダー上の明細は、ソーシングのセットアップに基づいて、ノードにバックオーダーされる必要があります (出荷ロケーションに最も近いか、または最優先のノード)。このシナリオでは、バックオーダーされた明細について、最適な出荷ノードが特定され、使用される必要があります。

Sterling Selling and Fulfillment Foundation は、エンタープライズと伝票種別 (document type) の組み合わせで、追加ルールを使用してこの機能を提供します。

### 固定事前定義ノードを持つ明細のバックオーダー時の分割

このルールは、「明細フルフィルメントに単一ノードを使用」ルールが有効になっている場合のみ、適用されます。このルールを使用すると、部分的にスケジュール解除またはバックオーダーされる固定事前定義ノードを持つあらゆるオーダー明細が、システムによって分割されます。スケジュール解除またはバックオーダーされた数量は、新しい明細に分割されます。

### 固定事前定義ノードを持たない明細のバックオーダー時の分割

このルールは、「明細フルフィルメントに単一ノードを使用」ルールが有効になっている場合のみ、適用されます。このルールを使用すると、部分的にスケジュール解除またはバックオーダーされる固定事前定義ノードを持たないあらゆるオーダー明細が、システムによって分割されます。スケジュール解除またはバックオーダーされた部分は、新しい明細に分割されます。

### 最優先の出荷ノードへの明細のバックオーダー

このルールは、「明細フルフィルメントに単一ノードを使用」が有効になっており、出荷ノードが固定ノードでない場合のみ、考慮されます。このルールは、ソーシング・セットアップに基づいて、この明細を履行する上で最適なノードに、明細をバックオーダーします。どの出荷ノードでも在庫が使用できない場合、システムは在庫チェックを実行せずに、最適な出荷ノードを見つけようとします。明細が完全にバックオーダーされると、オーダー明細上に新規ノードがスタンプされません。



このルールは、スケジュール・プロセスおよびリリース・プロセスでのみ使用されます。

---

## サービス要請の約束

このセクションでは、配送サービスまたは提供サービスなどのサービス・アイテムを顧客に約束する処理の、さまざまなシステム・コンポーネントについて説明します。

### 地域および地域スキーマ

地域および地域スキーマは、Sterling Selling and Fulfillment Foundation で地理を定義するためのビルディング・ブロックです。

地域スキーマは、特定の地理エリアを定義する地域の完全なセットです。

地域スキーマは、階層状の地域のグループで構成されます。各地域はそれ自体、他の地域のセット、またはその地域を構成する郵便番号のセットです。

例えば、米国のような国の地理定義は次のようになります。

- 他の 50 の地域、つまり 50 の州で構成される地域である米国。
- 各州はさまざまな数の地域 (州内の郡) で構成されます。
- 各郡は複数の市または街で構成される可能性があります。
- 各市および街は、それを構成する郵便番号のセットによって定義できます。

Sterling Selling and Fulfillment Foundation では、階層的な方法で地域を定義するため、地域の集合体を最小限のデータ入力で簡単に定義できます。

Sterling Selling and Fulfillment Foundation は広範な機能を備えているため、地理を定義および分割する方法が 1 つでは非常に制限が大きくなることがあります。例えば、出荷用に定義した顧客販売地域の内容は、インストール・サービスの提供に必要な定義の内容とは異なる場合があります。このような理由から、Sterling Selling and Fulfillment Foundation では、地域スキーマを必要な数だけ定義できるようになっています。その後、組織は、適切な地域スキーマを次のそれぞれの目的で関連付けることができます。

- 出荷済み製品地域スキーマ - 出荷のため、非常に広範な地域を定義して、国を大幅に少ない数の地域に分割できます。これらの地域は、製品を宛先に出荷する元の出荷ノードに関連付けることができます。
- 配達地域スキーマ - 配送サービスを提供するためには、市または街、さらに場合によっては市の地区レベルという、さらに細かい地域定義が必要です。
- 提供サービス地域スキーマ - 配送サービスと同様に、この目的のためにも、さらに細かい地域定義が必要です。ただし、地理の分割方法は、配達地域スキーマ定義用の分割方法とは同じにならない場合があります。
- 分析地域スキーマ - データウェアハウス分析のためです。
- 販売地域スキーマ - 異なる地域に対して、異なる製品価格設定や資格を提供するためです。

現在のところ、地域スキーマ定義を行えるのは、ハブ (Hub) (デフォルト) 組織だけです。エンタープライズは、ハブ (hub) が作成したスキーマのリストから適切なス

スキーマを選択できます。Sterling Selling and Fulfillment Foundation では、オプションのデフォルト・セットアップとして米国 (US) 用の基本的なスキーマ定義が提供されています。これを開始点として使用して、独自のスキーマ定義を行うことができます。

地域スキーマを定義できるのはハブ・レベルに限られますが、関連付けはリソース・プール・レベルで行えます。地域スキーマをリソース・プール・レベルで関連付けた場合、それはプロバイダー組織レベルの地域スキーマよりも優先されます。

スキーマによっては大量の地域が定義される可能性があるため、地域レベルという概念が導入されており、管理の容易性が実現されています。地域レベルは、地域を異なるカテゴリーに分類し、後から容易に検索できるようにするものです。地域レベルの例としては、「国」、「都道府県」、「市区町村」などが挙げられます。また、地域レベルはデータ入力エラーの防止にも役立ちます。一度セットアップするだけで、どの「レベル」をどの「レベル」の子にできるかが定義されます。これは、誤って「国」を「都道府県」に追加するなどのデータ入力エラーの防止に役立ちます。

## **Sterling Selling and Fulfillment Foundation が指定の郵便番号の枝地域を識別する方法**

1 つの地域で、一連の郵便番号の範囲を表すことができます。Sterling Selling and Fulfillment Foundation は、必要な郵便番号が一連の一部として含まれている地域を選択します。複数の地域に同じ郵便番号が含まれる可能性があるため、以下の方法で競合解決が行われます。

- 最大有効数字が範囲内にある地域定義に最高位の優先度が与えられます。
  - 例えば、地域 R1 が 901-902 という範囲を指定しているのに対し、地域 R2 は郵便番号範囲を 90101-90103 として指定してあります。90101 という郵便番号を検索した場合、正しい地域として、90102 - R2 または 90103 -R2 が選択されます。901-902 間の他の郵便番号を検索した場合は、正しい地域として R1 が選択されます。
- 最小範囲を持つ地域に、より高い優先度が与えられます。
  - 例えば、地域 R3 が 90101-90220 という範囲を指定しているのに対し、地域 R2 は郵便番号範囲を 90101-90103 として指定された郵便番号範囲を持っています。90101 という郵便番号を検索した場合は、正しい地域として、90102 - R2 または 90103 -R2 が選択されます。90101-90220 間の他の郵便番号を検索した場合は、正しい地域として R3 が選択されます。
- 必要な郵便番号に最も近い開始郵便番号に、より高い優先度が与えられます。
  - 例えば、地域 R4 が 90099-90102 という範囲を指定しているのに対し、地域 R2 は郵便番号範囲を 90101-90103 として指定しています。90101 または 90102 という郵便番号を検索した場合は、正しい地域として R2 が選択されます。90099-90100 間の他の郵便番号を検索した場合は、正しい地域として R4 が選択されます。

## **地域照合プリファレンス**

Sterling Selling and Fulfillment Foundation の地域照合を使用すると、アプリケーションの地域が存在する国に応じて、さまざまな住所フィールドに基づいて出荷先住所をその地域に一致させることができます。

国の照合基準として定義できる住所フィールドは次のとおりです。

- 国
- 都道府県
- 市区町村
- 郵便番号
- 住所 6

地域を定義する場合、地域レベルを指定します。例えば、地域「Boston」は「市区町村」として指定し、「MA」は「都道府県」として指定します。

地域照合プリファレンスで、特定の国の地域と住所を突き合わせる際に使用する住所フィールドを指定できます。例えば、米国の地域照合プリファレンスとして「市区町村」を指定すると、米国の国別コードのスタンプが付いた住所は、その住所の「市区町村」フィールドの値で照合されます。

## サービス・スロット

サービス（配達または提供）を提供する際には、顧客との予定が必要です。予定は、サービスを提供する時間帯のタイム・スロットに作成します。

Sterling Selling and Fulfillment Foundation 内のキャパシティー組織（capacity organization）は、それぞれに複数のサービス・スロットが含まれた複数のスロット・グループを定義できます。

### サービス・スロット

サービス・スロットは、開始時刻と終了時刻によって特定されます。定義したスロットの 1 つに対して、サービスの約束を作成します。

### スロット・グループ

スロット・グループはスロット・グループ ID によって識別される、指定されたサービス・スロットのセットです。1 つのスロット・グループをリソース・プールに関連付けることができます。

複数のスロットやスロット・グループを定義できるため、さまざまなリソース・プールについて異なる精度の予定を確保できます。例えば、配送サービスを提供するリソース・プールでは、4 時間間隔でしか約束できませんが、特定の提供サービスで使用されているリソース・プールでは、2 時間間隔で約束できます。精度は、サービスの提供に使用しているサービス・プロバイダーによっても異なることがあります。

この概念を説明するため、次の 2 つのタイプの配送サービスを提供することを想定してみます。

- 軒先渡し配達
- 設置工事付き配達

軒先渡し配達には、4 時間のタイム・スロット内でしか配達を約束できないサード・パーティーの配送サービス・プロバイダーを使用します。一方、設置工事付き配達には、専用の車両を使用し、2 時間のタイム・スロット内の配達を約束できません。

この例の場合、それぞれ表 24 および表 25 にリストされたサービス・スロットが含まれる 2 つのスロット・グループを定義します。

表 24. 軒先渡し配達のスロット・グループ

開始時刻	終了時刻	スロット名
8:00 AM	12:00 PM	午前
1:00 PM	5:00 PM	午後
5:00 PM	9:00 PM	深夜

表 25. 設置工事付きスロット・グループ

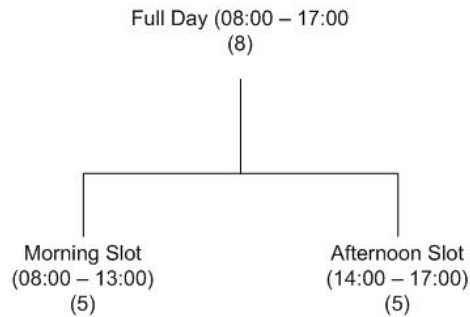
開始時刻	終了時刻	スロット名
8:00 Am	10:00 PM	午前早く
10:00 AM	12:00 PM	午前遅く
1:00 PM	3:00 PM	午後早く
3:00 PM	5:00 PM	午後遅く

軒先渡し配達スロット・グループ (表 24) は、サード・パーティー配達を提供するリソース・プールに関連付け、設置工事付きスロット・グループ (表 25) は、社内配達を提供するリソース・プールに関連付けます。

## 階層サービス・スロット

スロット・グループ内のスロットは、さまざまなレベルの階層で定義できます。階層スロット・グループは、親スロットと続く子スロットで構成できます。階層スロット・グループ内では通常、親タイム・スロットが、その子スロットすべてで提供できる最大キャパシティを提供します。これは、実際のリソースに対して過剰な約束が発生するのを防ぐためです。205 ページの図 35 に示すように、FullDay スロットを親スロットとみなすと、FullDay スロット内にある朝スロットおよび午後スロットは子スロットです。子スロットそのものは、多数の連続する子スロットに分岐させることができます。例えば、午前スロットは AM1 と AM2 に分割し、午後スロットは PM1 と PM2 に分割することができます。

この階層スロット・モデルのおかげで精度を高めることができるとともに、より短い時間間隔内に予定を確保することができます。これは、拡張顧客サービスを提供する際にも役立ち、過剰な約束を防止します。親スロットの利用可能性によって、子スロットの利用可能性が制限されます。



(8) – Indicates the total capacity of the parent slot  
 (5) – Indicates the individual capacity of the child slots

図 35. 階層スロット

## 顧客スロット・プリファレンス

Sterling Selling and Fulfillment Foundation で顧客の予定のスロット・プリファレンスを定義して、そのそれぞれにサービス・タイム・スロットを関連付けることが可能です。関連付けられたタイム・スロットは、優先または必須のどちらにもできます。Application Console で顧客に対して配達またはサービスの予約を確保する顧客サービス担当者 (customer service representative) は、該当する顧客にとってどのスロットが優先されているかを確認できます。必須タイム・スロットが使用されている場合、それらは、Sterling Selling and Fulfillment Foundation がその顧客に対して予定を計画する唯一のタイム・スロットです。顧客スロット・プリファレンスの定義について詳しくは、「*Sterling Selling and Fulfillment Foundation 分散オーダー管理構成ガイド*」を参照してください。

## サービス・アイテム

サービス・アイテムは、Sterling Selling and Fulfillment Foundation カタログ管理モジュール内で定義できます。Sterling Selling and Fulfillment Foundation では、サービス・アイテムと物理的な製品は明確に区別されます。サービスと物理的な製品には核心部分にいくつかの違いがありますが、類似点も数多くあります。このような理由から、物理的な製品とサービス・アイテムは、Sterling Selling and Fulfillment Foundation カタログ管理モジュールの一部として定義されます。

サービス・アイテムは、大きく分けてさらに次の 2 つのカテゴリに分割されません。

- 配送サービスは、製品の配達に関連付けられるサービスです。代表的な例は、軒先渡し配達と設置工事付き配達です。
- 提供サービスは、製品の配達前後に行われるサービスです。代表的な例は、「測定サービス」または「設置サービス」です。

物理的な製品とその製品で使用可能なサービスの間に関連付けを定義することもできます。この関連付けは、個々のアイテム・レベルまたはアイテム分類レベルで行うことができます。

これらのサービスを提供するためのスキルは、次のレベルで維持されます。

- サービス・アイテム

- サービスと製品の関連付け
- サービスとアイテム分類の関連付け

## サービス・リソース

サービス・リソース は、Sterling Selling and Fulfillment Foundation において、チームとして作業する 1 人以上の人を定義するために使用され、提供サービスや配送サービスを実行する単一のエンティティとして表されます。各リソースにはカレンダーが関連付けられます。このために、出荷ノードまたはその主要エンタープライズによって定義されたカレンダーの 1 つを選択するか、あるいはノードの出荷カレンダーを使用します。この出荷カレンダーはノードで定義することも、ノードの主要エンタープライズから継承することもできます。各サービス・リソースは単一のリソース・プールに関連付けられます。

スキルと地域はリソース・プール・レベルで定義され、リソースのカレンダーに加え、リソース・プール・カレンダーも使用されます。

この機能により、予定の予約時にリソースを割り当てることができます。例えば、複数回の訪問が必要な仕事では、同じリソースを複数日予約する必要があります。また、タスクの実行に必要なスキルを系統的にモデル化するのが難しいカスタム・タスクの場合も、リソース・スケジューリングを行う必要があります。この場合、スケジューラーがリソースの選択時に仕事に固有の要件を考慮します。

さらに、サービス・リソースを所有するノードのレベルでユーザーが定義されている場合は、Applications Manager でチーム・メンバーをサービス・リソースに関連付けることもできます。その後、チーム・メンバーをサービス作業オーダーの予定に関連付けることができます。チーム・メンバーとサービス・リソースの関連付けについて詳しくは、「Sterling Selling and Fulfillment Foundation 統合在庫管理 構成ガイド」を参照してください。チーム・メンバーとサービス作業オーダーの予定の関連付けについて詳しくは、「Sterling Selling and Fulfillment Foundation: Distributed Order Management User Guide」を参照してください。

リソース・レベルでキャパシティーを維持しているサービス・リソースまたはリソース・プールのキャパシティー、消費、および利用可能性は、リソース・プールのキャパシティーの単位にかかわらず、常に時間単位で計算および保管されます。

### サービス・リソースのキャパシティー計算

リソース・プールがサービス・リソース・レベルでキャパシティーを管理している場合、リソース・プールの総キャパシティーは、リソースのすべてのキャパシティーの合計になります。

例えば、リソース・プール (RP1) が以下のように構成されているとします。

スロット	午前 8:00 - 午後 12:00、午後 2:00 - 午後 6:00
キャパシティー	作業時間
Mike のチーム	午前 9:00 - 午後 12:00、午後 2:00 - 午後 6:00 (7 時間)

スロット	午前 8:00 - 午後 12:00、午後 2:00 - 午後 6:00
キャパシティー	作業時間
John のチーム	午前 10:00 - 午後 12:00、午後 2:00 - 午後 6:00 (6 時間)
Bob のチーム	午前 10:00 - 午後 12:00、午後 2:00 - 午後 6:00 (6 時間)

RP1 の総キャパシティーは 19 時間です。つまり、Mike のチーム、John のチームおよび Bob のチームのキャパシティーの合計になります。

### サービス・リソースの利用可能性の計算

割り振り可能なリソース・プールの最大利用可能性は、そのリソース・プールで管理されているサービス・リソースの最大利用可能性です。

例えば、上記の構成を使用した場合、8 AM から 12 PM までスロットにおける Mike のチームの消費は、以下のとおりです。

- 2 時間のスロット・ベースの予定
- 1 時間のスロット・ベースでない予定

これで、8 AM から 12 PM までのスロットで Mike のチームのすべてのキャパシティーを消費します。

	8 AM から 12 AM までのスロットの利用可能性	2 PM から 6 PM までのスロットの利用可能性
Mike のチーム	0 時間	4 時間
John のチーム	2 時間	4 時間
Bob のチーム	2 時間	4 時間
RP1	4 時間	12 時間

8 AM から 12 PM までのスロットの RP1 の利用可能性は 4 時間ですが、割り振り可能な最大利用可能性は、サービス・リソースの最大利用可能性であるため、2 時間になります。

### サービス・スロット間のリソースのキャパシティーのスパン

サービス・スロット間のリソースのキャパシティーをスパンするとき、サービス・スロットは、リソースが到着し、サービスの実行を開始するようにスケジューリングされた時間の範囲を指定します。

サービス・リソース・レベルでキャパシティーを管理し、階層的でないサービス・スロット・グループを持つリソース・プールでは、リソース・キャパシティーは、スロット間でスパンすることもできます。このシナリオでは、サービス・リソース状況は、隣接シフトを考慮することによって決まります。

キャパシティー組織 (capacity organization) は、適用可能なリソースのキャパシティー・ルールを設定することによってリソースのキャパシティー・スロットのスパンを可能にします。キャパシティー組織 (capacity organization) がリソースのキャパシ

ティー・スロットのスパンを可能にしたら、そのキャパシティー組織 (capacity organization) は、非作業シフトがスパンされる最大分数を指定する必要があります。例えば、午前 8 時から正午までと午後 1 時から午後 5 時まで作業するリソース・プールがある場合、このルールは、正午から午後 1 時の間の非作業時間をスパンできるかどうかと、その時間の長さを決定します。このルールが 60 分未満に設定された場合、スロットはスパンされず、午前 8 時から午前 9 時までのスロットは予定に 4 時間を使用できます。このルールが 60 分以上に設定された場合、スロットはスパンされ、午前 8 時から午前 9 時までのスロットは予定に 8 時間を使用できます。リソース・プールが隣接スロットの先頭で作業していない場合、隣接スロットのキャパシティーは使用されません。同様に、現在のスロットの終端が作業していない場合、隣接スロットのキャパシティーは使用されません。スロットが隣接スロットの先頭または現在のスロットの終端で取り込まれている場合、同じことが発生します。

以下の表は、スロットのスパンが消費の前に 60 分使用可能な場合の最終リソース状況を示しています。この例では、ある人 (ジョー) のリソース・プールの 1 日が午前 9 時に始まり、午後 8 時に終わります (昼休みの非作業時間 1 時間を含む)。

スロット	消費	在庫状況
9am-11am	0 時間	10 時間
11am-1pm	0 時間	8 時間
昼休み	0 時間	0 時間
2pm-4pm	0 時間	6 時間
4pm-6pm	0 時間	4 時間
6pm-8pm	0 時間	2 時間

予定が作成され、必須キャパシティーが消費されると、消費が、利用可能なスロットより大きくなる可能性があります。例えば、ジョーが午後 2 時ごろ到着するという 4 時間の予定が作成された場合、ジョーの 2pm-4pm スロットの在庫状況は、4pm-6pm スロットの在庫状況とともに消費されます。以下の表は、スロットのスパンが 60 分使用可能で 4 時間の予定が作成された場合の最終リソース状況を示しています。

スロット	消費	在庫状況
9am-11am	0 時間	4 時間
11am-1pm	0 時間	2 時間
昼休み	0 時間	0 時間
2pm-4pm	4 時間	0 時間
4pm-6pm	0 時間	0 時間
6pm-8pm	0 時間	2 時間

## リソース・プール (Resource Pools)

Sterling Selling and Fulfillment Foundation のリソース・プール とは、サービスを実行する個々のリソースのグループを表します。これは、Sterling Selling and Fulfillment Foundation 在庫管理モジュール内でキャパシティーを定義および管理する主要メカニズムです。これは、実際にサービスを実行する個々のリソースすべて



を表すためのものではなく、単にサービス・リソースの集合体を表すための方法に過ぎないことに注意してください。リソース・プールを定義することで、キャパシティーをリソース・プール・レベルまたはサービス・リソース・レベルで定義できるようになります。キャパシティーがリソース・プール・レベルで維持されている場合でも、サービス・リソースを情報目的で定義できます。サービス・リソースの定義について詳しくは、206 ページの『サービス・リソース』を参照してください。

すべてのリソース・プールは、そのリソース・プールを所有している 1 つのプロバイダー組織に属します。1 つのリソース・プールは、配送サービスまたは提供サービスのいずれかを提供できます。

また、リソース・プールは 1 つのノードに関連付けられます。配送サービス・リソース・プールの場合、これは、配達元のノードを表します。提供サービス・リソース・プールの場合は、単にリソース・プール・キャパシティーの管理を担当するノードを表します。

以下を定義できます。

- リソース・プールがサービスを提供する地域、およびそのリソース・プールがこれらの地域にサービスを提供する曜日と時刻
- 曜日ベースの標準キャパシティー
- 追加キャパシティー
- 追加キャパシティー
- 例外日ベースのキャパシティー・オーバーライド (capacity overrides)
- リソース・プールに関連付けられたカレンダー

Sterling Selling and Fulfillment Foundation の約束機能は、オーダー明細のサービスおよび地理要件に一致するリソース・プールに対してキャパシティーが使用可能かどうかを確認します。

## サービス・スキル

各サービス要請では、アクティビティーを実行するために特定のスキルが必要な場合があります。例えば、洗濯機の設置には配管および電気のスキルが必要です。Sterling Selling and Fulfillment Foundation は、サービス・アイテムに対して選択したリソース・プールがサービス明細に必要なすべてのスキルを確実に提供できるようにします。

## 追加の固定キャパシティー

配送サービスにおいては、特定のサービス・タイプの場合に、特定の地域にサービスを提供することが、他の地域よりも難しいとみなされる場合があります。例えば、複雑なサービス・タイプの場合に、郊外エリアへの配達には都市部への配達と比較して 30 分余分にかかるように、Sterling Selling and Fulfillment Foundation を構成したい場合があります。追加キャパシティーは、任意の地域とサービス・タイプの組み合わせに対して、関連付けることができます。追加キャパシティーの構成について詳しくは、「*Sterling Selling and Fulfillment Foundation: Catalog Management Configuration Guide*」を参照してください。

## 多次元キャパシティー

多次元キャパシティーは、計画中にキャパシティーを制限することで、予定のスケジュールリングを支援します。主要キャパシティーに加え、重量や容積など、追加のキャパシティー制限を提供できます。

### 重量

これは、特定の発送で運送できる製品の重量を制限します。

### 容積

これは、特定の発送で運送できる製品の容積を制限します。

キャパシティーは、各タイム・スロットごとに定義および制限されます。この結果、消費されるキャパシティーは、要求されたタイム・スロットに応じて変わることになります。スロットのキャパシティーをブロックするには、すべての次元の要求キャパシティーを、リソース・プールのすべての次元の使用可能キャパシティーよりも小さくする必要があります。例えば、午前および午後のスロットを子スロットとして持つ終日スロットについて考えてみましょう。キャパシティーは終日スロットに対して定義されており、10 時間、100 kg、250 立法フィートなど任意の単位で制限できます。また、各子スロットのキャパシティーは 5 時間、50 kg、125 立法フィートなどにすることができます。詳しくは、図 36 を参照してください。

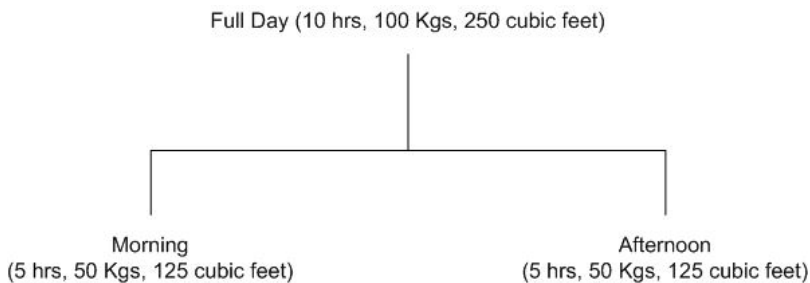


図 36. 多次元キャパシティー

## 追加キャパシティー

必要と思われる場合、エンタープライズは、追加キャパシティーを使用できることがあります。例えば、配達トラックを 10 台所有する会社で、重要な顧客が特定の日までの製品配達を要求している場合、サード・パーティーのトラックを 2 台使用できます。Applications Manager を使用して顧客を作成する際に、そのユーザーにはデフォルトで追加キャパシティーを考慮するように指定できます。追加キャパシティーは、各標準キャパシティー期間内で、特定の曜日に定義できます。作業オーダーの予定を取得する際に、追加キャパシティーを考慮するよう選択することができます。また、キャパシティー・コンソールで使用可能なキャパシティーを表示する際に、追加キャパシティーを考慮するよう選択することもできます。

## 外部保持キャパシティー

キャパシティー情報がリソース・プール・レベルで使用可能な場合、リソース・プールを指定できます。この情報を約束機能で使用可能にするには、Sterling Selling and Fulfillment Foundation の在庫管理モジュールでこれらを定義するか、またはキャパシティー組織 (capacity organization) に「外部保持キャパシティー」を所有しているというフラグを立てて、定義済みのユーザーEXIT によってこの情報をリアルタイム・ベースでシステムに提供します。

上記いずれのメカニズムでもキャパシティー情報を使用可能にできない場合、そのリソース・プールにフラグを立てて、キャパシティー情報が使用不可であることを示すことができます。Sterling Selling and Fulfillment Foundation は、これを「無限の」キャパシティーと同様に扱いますが、サービス提供曜日および地域の考慮事項は引き続き考慮します。

リソース・プールを定義する際は、リソース・プールを 1 つのキャパシティー組織 (capacity organization) に関連付ける必要もあります。このリソース・プールを使用できるのは、リソース・プールと同じキャパシティー組織 (capacity organization) を持つセラー組織だけです。キャパシティー組織 (capacity organizations) について詳しくは、『キャパシティー組織 (Capacity Organization)』および 25 ページの『第 3 章 参加者の管理』を参照してください。

各リソース・プールはスロット・グループに関連付けられます。そのリソース・プールに対して確保された予定はすべて、このグループ内で定義されたスロット用になります。キャパシティー定義もすべて、このスロット・グループで定義されたスロット用になります。

## 配送サービスの約束

製品明細は、配送サービス要請に関連付けることができます。通常、配送サービスは製品が次に該当する場合に提供されます。

- 重量物、特大サイズ、または壊れ物であり、一般の配送業者では輸送できない製品
- 特別な器材や人員を使用する特殊な扱いが必要となる製品

配送サービスを必要とする製品例としては、背面投射型テレビや洗濯機などが挙げられます。

このセクションでは、ラスト・マイル配送サービスを使用して配達する製品の、約束機能に関連する製品機能について説明します。製品のセットを配達するには、配送サービス要請を配達対象の製品明細に関連付ける必要があります。Sterling Selling and Fulfillment Foundation では、配送サービスをスケジュールする際に、すべての製品明細を同時にまとめて配達する必要があると想定します。製品明細の一部が (在庫、ステータス状況、またはその他の理由から) ない場合には、まとめての配達 はスケジュールされません。

配送サービスの約束プロセスにおける一般的なステップは次のとおりです。

- 配達に利用可能なスロットの照会およびその利用状況に応じた顧客予定の記録
- 記録された予定に基づいた配達のスケジュールリング
- 配達ノードへの配達指示の通知

## 利用可能なスロットの照会

このセクションでは、配送サービスに利用可能なスロットの検索に関連する機能について説明します。

**基本構成:** Sterling Selling and Fulfillment Foundation で必要な配送サービスの約束には、いくつかの事前準備構成が関連しています。このセクションで説明する基本構成の多くは、エンタープライズ組織によって定義されます。スロットの利用可能性の照会に必要な基本構成をセットアップするために、以下のパラメーターを設定する必要があります。

### 配達ソーシング・ルールは定義済みか?

ソーシング・ルールは、配送ノードの選択を管理します。このパラメーターを「Yes」という値で設定した場合、配達ロケーションは、ソーシング・ルールのセットアップに基づいて決定されます。この値を「No」に設定した場合、「出荷地域」にサービスを提供する配達ロケーションを、Sterling Selling and Fulfillment Foundation によって選択できます。配達地域が相互に交わらないように配達ロケーションを編成済みの場合、この構成を「No」に設定できます。そうすれば、Sterling Selling and Fulfillment Foundation は、各ノードの配達地域に基づいて正しい配達ロケーションを選択します。

複数の配達ロケーションが同じ地域にサービスを提供している可能性があり、正しい配達ロケーションを決定するためのソーシング・ルールをセットアップする場合、このパラメーターを「Yes」に設定する必要があります。

### キャパシティー組織 (Capacity Organization) は?

Sterling Selling and Fulfillment Foundation は、キャパシティー組織 (Capacity organization) のパラメーターを、キャパシティー定義を区別されたサイロに分離するための仕組みとして使用します。Sterling Selling and Fulfillment Foundation で定義されるすべてのリソース・プールは、ただ 1 つのキャパシティー組織 (capacity organization) のみに属します。組織は、同じキャパシティー組織 (capacity organization) 内に定義されているリソース・プールのみを使用できます。この定義によって、Sterling Selling and Fulfillment Foundation は、リソース・プールを共有してはいけない組織を完全に分離することができます。

正しいサイロを参照するように、キャパシティー組織 (capacity organization) をセットアップする必要があります。組織は、キャパシティー組織 (capacity organization) を 1 つのみ保有することができ、この組織によって実行されたサービスは、同じキャパシティー組織 (capacity organization) のリソース・プールを介してスケジュールされる必要があります。

### キャパシティーは外部で保管されるか?

このパラメーターは、リソース・プールのキャパシティーが Sterling Selling and Fulfillment Foundation 内で管理されるかどうかを制御します。このパラメーターは、キャパシティー組織 (capacity organization) レベルで保持されます。これが「Y」に設定された場合、Sterling Selling and Fulfillment Foundation は、スロット利用可能性のチェック時に、すべてのリソース・プールのキャパシティーをリアルタイムで取り出す必要があることを前提とします。

「Y」に設定する必要がある状況は、以下のとおりです。

- サード・パーティーのフリート管理ツールを使用しており、Sterling Selling and Fulfillment Foundation 内の各リソース・プールにスロット・キャパシティーを定義しない場合。スロット利用可能性のチェック時に、Sterling Selling and Fulfillment Foundation は、リアルタイムのユーザーEXIT 呼び出しを行い、リソース・プールのスロット利用可能性を検索します。ご使用のフリート管理ソフトウェアから読み取った利用可能性を返して渡すことができます。キャパシティーを外部に保持できても、依然としてリソース・プールを Sterling Selling and Fulfillment Foundation 内に定義する必要があることに注意してください。
- スロット利用可能性をリアルタイムで提示できる、サード・パーティー・サービス・プロバイダーを使用している場合。サード・パーティー・プロバイダーを表すリソース・プールを定義し、スロット利用可能性のチェック時に、このプロバイダーに対してリアルタイムの呼び出しを行い、最新の利用可能性を把握することができます。

## ソーシング・ルール

ソーシング・ルールは、以下のキー・パラメーターを 1 つ以上指定することによって作成できます。

- 出荷先ロケーションまたは出荷先ノードの地理的地域
- フルフィルメント・タイプ
- セラー組織
- ソーシング条件

ソーシング・ルールで上記の任意のパラメーター (フルフィルメント・タイプを除く) をボイドのままにするという柔軟性があり、これは、ソーシング・ルールがそのパラメーターのすべての値に適用可能であることを意味します。

各ソーシング・ルールに対して、製品をソーシングするために使用するノードまたは分配グループ (distribution group) のシーケンスを指定できます。

製品ソーシングで説明したとおり、Sterling Selling and Fulfillment Foundation は、十分な在庫が配達ロケーションにない場合、転送または購入を作成しようとしません。調達オーダーの設定は、出荷される製品のオーダーと同じです。調達構成について詳しくは、153 ページの『出荷する製品の約束』を参照してください。

## 配達ロケーションの検索

配達ロケーションは、以下のことに基づいて決定されます。

- ソーシング・ルールがセットアップされる場合は、ソーシング・ルール
- ソーシング・ルールがセットアップされない場合は、サービス地域を担当するリソース・プール

リソース・プール・セットアップに基づいてノードを決定する場合は、Sterling Selling and Fulfillment Foundation は、該当配達地域をサービス対象とし、配送サービスを実行するのに必要なスキル (必要な場合) を持つすべてのリソース・プールについて検索します。使用可能なリソース・プールと関連付けられたすべてのノードが、製品を調達するために考慮されます。

Sterling Selling and Fulfillment Foundation は、確実に次のようにします。

- 選択されたノードに、配達要請に応えられるキャパシティーがあること。
- 選択されたノードが、この配達要請の一部として配達されているすべての製品について使用可能であること。Sterling Selling and Fulfillment Foundation では、配達ロケーションに十分な在庫がない場合は、他のロケーションからの自動転送オーダーまたは自動調達オーダーを生成するように構成することができます。図 37 は、配達の適応範囲を示します。

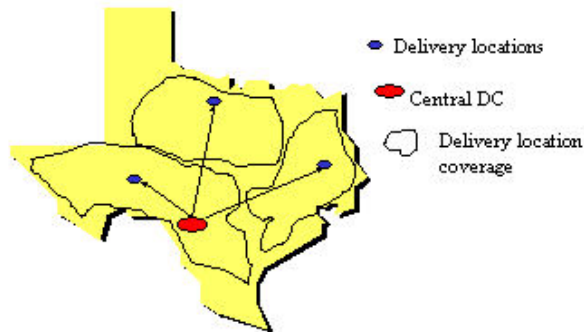


図 37. 配達ロケーション

## 利用可能なスロットの検索

Sterling Selling and Fulfillment Foundation は、可能な配達ロケーションと関連付けられており、かつ以下の追加の制約事項を満足するすべてのリソース・プールを考慮します。

- 配達地域は、考慮されるように、リソース・プールによるサービス対象である必要があります。
- 配達を実行するのに必要なスキルがある場合、リソース・プールはそれらのスキルをすべて提供します。

Sterling Selling and Fulfillment Foundation は、次のことに基づいて、利用可能なスロットを提示します。

- リソース・プールが使用可能かどうか
  - リソース・プールが使用可能かどうかをチェックするために、Sterling Selling and Fulfillment Foundation は、次のことを確実なものにします。
    - リソース・プールに、考慮されているスロット内に十分なキャパシティーが残っていること。これは、リソース・プールでキャパシティー情報が使用可能である場合にのみ、考慮されます。キャパシティー情報が使用可能でない場合は、Sterling Selling and Fulfillment Foundation では、これは「無限のキャパシティー」と見なされるため、このスロットを予定の記録用に提示することができます。
    - リソース・プールが、該当地域の「曜日」セットアップに基づいて、その日付での配達地域として機能すること。このチェックは、キャパシティーが使用可能であるかどうかに関係なく、行われます。
    - この日付が、リソース・プールのカレンダーで非稼働日としてマークされていないこと。リソース・プールのカレンダーをリソース・プールについてオ

オーバーライドすることができます。それ以外の場合は、リソース・プールのノードのデフォルト・カレンダーが使用されます。

- 製品の在庫状況

- 配達されるすべての製品の製品の在庫状況 - Sterling Selling and Fulfillment Foundation では、すべてのアイテムが最初に提示されたスロットで確実に使用可能になります。
- 順序制約 - スロットは、オーダー明細の順序制約に基づいて提示されます。

- 最小サービス通知時刻

最小サービス通知時刻をノード・レベルで指定できます。現在時刻およびこの通知時刻 (営業時間) 内に開始しているサービス・スロットがあっても、使用不可と見なされ、提示されません。

- サービス検索期間

このパラメーターは、スケジューリング・ルールの一部としてセットアップされます。このパラメーターは、サービスが使用可能かどうかを調べる日付の上限を設定します。これは通常、30 から 60 日の範囲に設定します。Sterling Selling and Fulfillment Foundation は、このパラメーターによって定義された境界内でスロットが使用可能であるかどうかを調べます。

スロットが使用可能であるかどうかを照会した場合、Sterling Selling and Fulfillment Foundationは、一連の使用可能なスロットを提示します。1 つのスロットを複数のリソース・プールによるサービス対象とすることができるため、その一連のスロットは、その地域およびサービス・アイテムをサービス対象とすることができるすべてのリソース・プール間で使用可能であることを表します。実際に使用するリソース・プールは、オーダー明細がスケジュールされるまでロックされません。

## スケジューリングのキャパシティー数量の計算

Sterling Selling and Fulfillment Foundation は、スロット利用可能性のチェック対象のキャパシティー数量を決定します。計算は、「配送サービス・アイテムに使用される固定キャパシティー・ユニット + 関連製品の数量に基づくキャパシティー・ユニット」で行われます。

使用されるキャパシティー・ユニットは、製品およびサービス関連付けをカタログ管理システムに作成するときに定義された、製品関連付けに基づいています。ユーザーは、「使用されるサービス・キャパシティー/製品数量」を定義できます。

Sterling Selling and Fulfillment Foundation は、キャパシティーの可変部分を (オーダーされる製品数量) \* (サービス・キャパシティー数量/製品数量) で計算します。これらすべてに加えて、オプションに基づくキャパシティーが総数量に追加されます。

チェック対象のキャパシティー合計 = 固定数量 + オーダーされる製品数量 \* (サービス数量/製品数量) + オーダーされたオプション数量

注: 配達が既にスケジュール済みで、配達に追加される追加明細に対する照会が行われている場合、固定キャパシティーおよびオプション数量は、キャパシティー合計の数字に含まれません。

作業オーダー作成時の推定キャパシティー計算について詳しくは、235 ページの『作業オーダーのキャパシティー計算』を参照してください。

## 配送サービスのスケジューリング

配送サービスをスケジュールできるようにするには、その配達に関連付けられているすべての製品明細がスケジューリングで使用可能になっている必要があります。製品のコントロールについて詳しくは、153 ページの『出荷する製品の約束』を参照してください。

サービス・アイテムは、作業オーダーまたは予定が記録されていなければ、スケジュールされたときに「予約済み」ステータスに移ります。スケジューリング・プロセス中に、Sterling Selling and Fulfillment Foundation は、配送サービスに使用するリソース・プールを割り当てます。配達ノードの選択は、213 ページの『配達ロケーションの検索』で説明されているように実行されます。

複数のリソース・プールが使用可能な場合、Sterling Selling and Fulfillment Foundation は、出荷先地域用の「2 次」リソース・プールのマークが付いていないリソース・プールを使用して、配送サービスをスケジュールしようとします。「2 次」リソース・プールがスケジューリングに使用されるのは、他に使用可能なリソース・プールがない場合です。

スケジューリング・プロセスの一環として、選択されたリソース・プールに対するキャパシティー消費が記録されます。

## 出荷済み製品とのその他の主な相違点

Sterling Selling and Fulfillment Foundation では、配達製品の輸送時間計算は実行されません。予定が記録される場合、出荷日と配達日は予定開始時間に設定されません。

最終出荷先ロケーションと配達ロケーション間の転送スケジュールは、スロットの可用性の判定や提案に使用されません。スロットの可用性は、リソース・プールの制約に基づいて判定されます。

## 提供サービスの約束

オーダーされる製品明細に、1 つ以上の提供サービスを関連付けることができます。複数のサービス明細は 1 つの作業オーダーに関連付けられます。販売オーダーの各サービス明細は、1 つの作業オーダー・サービス明細として表されます。ただし、予定をとる際には、すべてのサービス・スキルを提供するリソース・プールのみが考慮されます。作業オーダーも、サービス・スロットに関係なく、サービス・リソースに対して作成されます。

提供サービスの約束プロセスにおける一般的なステップは次のとおりです。

- サービスに利用可能なスロットの照会およびその利用状況に応じた顧客予定の記録
- 記録された予定に基づいたサービスのスケジューリング
- 配達ノードへの配達指示の通知



**注:** 提供サービスのオーダー明細は、連鎖オーダーの作成に使用できません。同様に、提供サービス・オーダー明細に関連付けられたノードが別の法人組織に属している場合には、そのオーダーは連鎖オーダーを作成せずにスケジュールされます。

## 利用可能なスロットの照会

このセクションでは、提供サービスに利用可能なスロットの検索に関連する機能について説明します。

**基本構成:** Sterling Selling and Fulfillment Foundation で必要な提供サービスの約束には、いくつかの事前準備構成が関連しています。このセクションで説明する基本構成の多くは、エンタープライズ組織によって定義されます。スロットの利用可能性の照会に必要な基本構成をセットアップするために、以下のパラメーターを設定する必要があります。

### サービス・ソーシング・ルールは定義済みか?

ソーシング・ルールは、サービス・ノードの選択を制御します。ソーシング・ルールをセットアップして、ソーシング・ルールを定義するかどうかを示すことができます。このパラメーターの値を「Yes」で設定した場合、サービス・ロケーションは、ソーシング・ルールのセットアップに基づいて決定されます。この値を「No」に設定した場合、ロケーションによってサービスが提供される「サービス地域」に基づいて、Sterling Selling and Fulfillment Foundation がサービス・ロケーションを選択します。

キャパシティー組織 (capacity organization) 内のリソース・プールを使用でき、通常複数のサービス・プロバイダー間で重複するサービス地域がない場合、このパラメーターの値を「N」に設定すると、サービス・プロバイダーの選択に必要な詳細な構成を省略することができます。

プロバイダーの選択をより詳細に制御する場合は、このパラメーターを「Yes」に設定します。Sterling Selling and Fulfillment Foundation は、ユーザーが定義したソーシング・ルールを使用して、正しいプロバイダーを選択します。

### キャパシティー組織 (Capacity Organization) は?

配送サービスと提供サービスに同じキャパシティー組織 (capacity organization) が使用されます。このパラメーターについて詳しくは、配送サービスのセクションの『キャパシティー組織 (Capacity Organization) は?』を参照してください。

### キャパシティーは外部で保管されるか?

このパラメーターは、配送サービスで定義された場合と同じ動作をします。このパラメーターについて詳しくは、配送サービスのセクションの『キャパシティーは外部で保管されるか?』を参照してください。

## 分配グループ (distribution group)

分配グループ (distribution group) は、一連のノードまたはプロバイダー組織を作成するための機能を提供します。その後、このグループをソーシング・ルールと関連付けることができます。Sterling Selling and Fulfillment Foundation は、このグループの一部であるすべてのノードまたは組織を考慮し、最終選択を行うために各種要

因について最適化します。ノードまたはプロバイダーの選択に使用される各ノードまたは組織について、優先番号も指定できます。

複数のノードまたはプロバイダーがあり、Sterling Selling and Fulfillment Foundation が最適化ロジックでそのビルドに基づいてノードを選択する場合は、そのようなグループを作成して、Sterling Selling and Fulfillment Foundation が最適化パラメーターに基づいてノードの最終選択を行います。事前定義済みの固定された方法でノード選択を順序付けたい場合は、Sterling Selling and Fulfillment Foundation が固定順序でノードを検索できるように、ソーシング・ルール・セットアップの「順序付け」機能を使用します。

## ソーシング・ルール

ソーシング・ルールは、出荷する製品のソーシング・ルールと同様に定義されます。製品クラス・パラメーターおよびアイテム分類パラメーターがソーシング・ルールの設定に使用できないという違いを除いて、残りのソーシング・ルール構成は、出荷する製品の 159 ページの『ソーシング・ルール』で説明している構成と同じです。

## サービス・ロケーションの検索

サービス・ロケーションは、次のものに基づいて決定されます。

- ソーシング・ルールがセットアップされる場合は、ソーシング・ルール
- ソーシング・ルールがセットアップされない場合は、サービス地域を担当するリソース・プール

リソース・プール・セットアップに基づいてノードを決定する場合は、Sterling Selling and Fulfillment Foundation は、該当地域をサービス対象とするすべてのリソース・プールについて検索します。使用可能なリソース・プールと関連付けられたすべてのノードが、サービスを調達するために考慮されます。

Sterling Selling and Fulfillment Foundation は、選択されたノードに、確実にサービス要請に応えられるキャパシティーがあるようにします。

## 利用可能なスロットの検索

Sterling Selling and Fulfillment Foundation は、可能なサービス・ロケーションと関連付けられており、かつ以下の追加の制約事項を満足するすべてのリソース・プールを考慮します。

- サービス地域は、考慮するためにリソース・プールによるサービス対象にする必要があります。
- 配達を実行するのに必要なスキルがある場合は、リソース・プールはそれらのスキルをすべて提供します。

Sterling Selling and Fulfillment Foundation は、次のことに基づいて、利用可能なスロットを提示します。

- リソース・プールが使用可能かどうか
  - リソース・プールが使用可能かどうかを調べるために、Sterling Selling and Fulfillment Foundation は、次のことを確実なものにします。
    - リソース・プールに、考慮されているスロット内に十分なキャパシティーが残っていること。これは、リソース・プールでキャパシティー情報が使用可

能である場合にのみ、考慮されます。キャパシティー情報が使用可能でない場合は、Sterling Selling and Fulfillment Foundation では、これは「無限のキャパシティー」と見なされるため、このスロットを予定の記録用に提示することができます。

- リソース・プールが、該当地域の「曜日」セットアップに基づいて、その日付での配達地域として機能すること。このチェックは、キャパシティーが使用可能であるかどうかに関係なく、行われます。
  - この日付が、リソース・プールのカレンダーで非稼働日としてマークされていないこと。リソース・プールのカレンダーをリソース・プールについてオーバーライドすることができます。それ以外の場合は、リソース・プールのノードのデフォルト・カレンダーが使用されます。
- 最小サービス通知時刻

最小サービス通知時刻をノード・レベルで指定できます。現在時刻およびこの通知時刻 (営業時間) 内に開始しているサービス・スロットがあっても、使用不可と見なされ、提示されません。

- サービス検索期間

このパラメーターは、スケジューリング・ルールの一部としてセットアップされます。このパラメーターは、サービスが使用可能かどうかを調べる日付の上限を設定します。これは通常、30 から 60 日の範囲に設定します。Sterling Selling and Fulfillment Foundation は、このパラメーターによって定義された境界内でスロットが使用可能であるかどうかを調べます。

スロットが使用可能であるかどうかを照会した場合は、Sterling Selling and Fulfillment Foundationは、一連の使用可能なスロットを提示します。1 つのスロットを複数のリソース・プールによるサービス対象とすることができるため、その一連のスロットは、その地域およびサービス・アイテムをサービス対象とすることができるすべてのリソース・プール間で使用可能であることを表します。実際に使用するリソース・プールは、オーダー明細がスケジュールされるまでロックされません。

## 提供サービスのスケジューリング

提供サービス明細をいつスケジュールするかを判断する場合の管理は、配送サービスの場合と同様です。詳しくは、216 ページの『配送サービスのスケジューリング』を参照してください。

---

## オーダー明細の複雑な順序付け

さまざまな製品の配達および関連サービスを順序付けできるように、コントロールが提供されています。製品が配達される前に特定の配達前サービスを完了し、さらに確実に製品が配達された後にのみ配達後サービスを実行するようにスケジュールできます。

製品とサービス間の関連付けをセットアップする際、オフセットを指定できます。-ve オフセットは、サービスが実行されなければならない、製品配達前の 最小限のオフセット時間数を示します。+ve オフセットは、サービスが実行されなければならない、製品配達後の最小限のオフセット時間数を示します。指定された時間

は、経過時間であり、営業時間ではありません。ここでは、サービス明細が製品明細よりも前に順序付けされた場合、そのサービス明細は先行の明細と呼ばれ、その逆の場合は後続の明細と呼ばれます。

オフセット時間と組み合わせると、前に説明した明細の依存関係によって、複雑な順序付けの状況が作成されます。ここでは、明細は特定の時間より前または後にスケジュールされません。Sterling Selling and Fulfillment Foundation は、以下の 2 つの制約日を計算します。

- 日付前は完了できません - この日付より前は完了を目的としてオーダー明細をスケジュールできない日付です。完了とは、製品明細の配達、およびサービス明細の場合にはサービス完了の記録を指します。この日付は、以下のいずれかの理由で明細に課することができます。
  - 先行の明細が既にスケジュール済み。制約日は、先行の明細の最新のスケジュール済みの配達日に、製品とサービス間の関連付けで指定されたオフセット時間を加算した日付を表します。
  - 先行の明細に配達指定日が指定済み。この日付は、先行の明細の配達指定日にオフセット時間を加算した日付を表します。
  - 先行の明細に既に予定が記録済み。この日付は、最新の予定時間にオフセット時間を加算した日付を表します。
- 日付後は完了できません - この日付より後は完了を目的としてオーダー明細をスケジュールできない日付です。Sterling Selling and Fulfillment Foundation が、ユーザー・インターフェースでこの日付を計算し表示しても、この日付は先行の明細に対する実際の制約として使用されません。状況によっては、先行の明細と後続の明細の間でデッドロックが作成され、スケジュールリング操作が実行できなくなる可能性があることが、主な理由です。このような状況を回避するために、この制約はソフトな制約として扱われ、明細には課されません。Sterling Selling and Fulfillment Foundation は、「日付後は完了できません」の制約を満たさない明細を強調表示します。この日付は、以下のいずれかの理由で明細に課することができます。
  - 後続の明細にキャンセル指定日が指定済み。この日付は、後続の明細のキャンセル指定日からオフセット時間を差し引いた日付を表します。
  - 後続の明細に予定日が記録済み。この日付は、後続の明細の最初の予定日からオフセット時間を差し引いた日付を表します。

Sterling Selling and Fulfillment Foundation は、上記の制約を満たさない予定または配達スケジュールを持つオーダー明細を強調表示します。

Sterling Selling and Fulfillment Foundation は、「スケジュールできません」というインディケータも計算します。このインディケータは、このインディケータが「No」に設定されているオーダー明細がスケジュールできないことを示します。この理由は、以下のいずれかです。

- 先行の明細がまだスケジュール未済。すべての先行の明細は、後続の明細がスケジュールされる前にスケジュールされていなければなりません。
- 製品をサービスに関連付ける際、「完了のスケジュールリングを保留 (Hold scheduling for completion)」に true のマークが付けられ、先行の明細はまだ完了していません。

明細に「スケジュールできません」というマークが付けられている場合でも、Sterling Selling and Fulfillment Foundation では、ユーザーは最も使用できる可能性が高い予定を照会することができます。また、希望する予定を記録することもできます。ただし、明細をスケジュールできないようにしている制約が除去されない限り、その明細をスケジュールすることはできません。

上記のパラメーターを使用して、複雑な順序付けを必要とする明細に対して、効率的に予定を記録し、適切な時点でそれらをスケジュールすることができます。

## アイテム・ベース割り当て

アイテム・ベース割り当て (IBA) はプロセスの 1 つで、システム内でのユーザー構成とサプライおよび需要の変更に基づいて、既存オーダーのアイテムについてのコミットされていない需要およびコミット済み需要を、より適切なサプライに再割り当てするプロセスです。

Sterling Selling and Fulfillment Foundation は、次のような 2 つのタイプのアイテム・ベース割り当てオプションを提供します。

- FIFO 優先順位付け。最初に発行されるオーダーに、後で発行されるものよりも高い優先順位を付与します。
- オーダー属性に基づく、ユーザー構成の優先順位付け。指定のオーダーに、他のオーダーよりも高い優先順位を付与します。

ユーザー構成の IBA 優先順位が設定されていない場合、あるいはそれが設定されて、完了している場合は、Sterling Selling and Fulfillment Foundation は、デフォルトとしてシステム構成の IBM 優先順位を採用します。

この後続セクションでは、両方のタイプの IBA について説明します。

### FIFO IBA

デフォルト FIFO IBA では、影響を受けるオーダーの需要は、FIFO 順序で再割り当てされます。つまり、最初に発行されたオーダーが、オーダーの発行に先立って約束されます。例えば、表 26 に示されているオーダーは、対応する供給および需要と一緒にシステム内に存在しています。購入オーダーが受け入れられてからオーダーを出荷するまでに 1 日要すると仮定します。

表 26. IBA の例

購入オーダー	製品	ノード	数量	ETA	販売オーダー	製品	ノード	数量	需要出荷日	オーダー日付
PO1	アイテム 1	ノード 1	5	11/15/08	SO1	アイテム 1	ノード 1	5	11/16/08	11/03/08
PO2	アイテム 1	ノード 1	5	11/20/08	SO2	アイテム 1	ノード 1	5	11/21/08	11/05/08
PO3	アイテム 1	ノード 1	5	11/21/08	SO3	アイテム 1	ノード 1	5	11/22/08	11/10/08

販売オーダー「SO1」が取り消され、FIFO アイテム・ベース割り当てが考慮される場合は、供給が表 27 に示されている早い方の日付で対応可能になるため、「SO1」より後で発行されたオーダー「SO2」および「SO3」は早い方の日付に再割り当てされます。

表 27. FIFO IBA の結果

購入オーダー	製品	ノード	数量	ETA	販売オーダー	製品	ノード	数量	需要出荷日	オーダー日付
PO1	アイテム 1	ノード 1	5	11/15/08	SO1	アイテム 1	ノード 1	0	該当なし	11/03/08
PO2	アイテム 1	ノード 1	5	11/20/08	SO2	アイテム 1	ノード 1	5	11/16/08	11/05/08
PO3	アイテム 1	ノード 1	5	11/21/08	SO3	アイテム 1	ノード 1	5	11/21/08	11/10/08

アイテム・ベースの割り当て機能は、以下に説明するとおり、順番に実行する必要がある 2 つのプロセスで構成されています。

1. アイテム・ベース割り当てを実行するアイテムの識別。

アイテムを識別するプロセスは以下のとおりです。

a. アイテムについてアイテム・ベース割り当てが必要な場合の検出。

Sterling Selling and Fulfillment Foundation は、手持ちの在庫状況がない場合にノードのアイテムにアイテム・ベースの割り当てプロセスが必要であることを検出します。これは、Sterling Selling and Fulfillment Foundation が、reserveAvailableInventory、modifyWorkOrder、scheduleOrder、および releaseOrder などの API を約束することにより更新を行うためにアイテムの在庫状況を検索しているときに行われます。アイテム・ベースの割り当ての使用ルールが有効になっており、どのノードにも手持ち在庫状況がない場合は、IBA\_REQUIRED フラグを「Y」に設定した状態 (アイテムとノードで IsItemBasedAllocationAllowed = "Y" が指定されている場合のみ) で、該当のアイテムとノードの組み合わせの YFS\_IBA\_TRIGGER テーブルにアイテム・ベースの割り当てトリガーが挿入されます (まだ存在していない場合)。これは、指示されたノードのアイテムに IBA プロセスが必要であること、また、システムがアイテムとノードの組み合わせのトリガーを IBA エージェントが処理できるようにしていることを表します。

b. アイテムとノードを処理するためのアイテム・ベース割り当てエージェントのトリガー。

ノードのアイテムにアイテム・ベース割り当てプロセスが必要である (IBA\_REQUIRED="Y" が設定された YFS\_IBA\_TRIGGER テーブルにレコードが存在する) ことを Sterling Selling and Fulfillment Foundation が検出すると、Sterling Selling and Fulfillment Foundation は、以下のオカレンスに基づいて、アイテム・ベース割り当てエージェントがそのアイテムとノードの組み合わせの処理をトリガーできるようにします。

c. 供給変更が発生した場合

供給に変更があり、供給タイプが TriggerItemBasedAllocation="Y" である場合、Sterling Selling and Fulfillment Foundation は、IBA\_RUN\_REQUIRED フィールドを「Y」に更新することによって、在庫アイテム (inventory item) とノードを IBA エージェントが処理するようにトリガーします。

d. 特定の需要変更が発生した場合

需要に変更があると、YFS\_IBA\_TRIGGER テーブルの IBA\_RUN\_REQUIRED フィールドを「Y」に更新することにより、在庫アイテム (inventory item) とノードは IBA エージェントによって処理されるようにトリガーされます。

以下の変更が発生すると、YFS\_IBA\_TRIGGER テーブルの IBA\_RUN\_REQUIRED フィールドを更新して「Y」に設定することにより、在庫アイテム (inventory item) とノードは IBA エージェントによって処理されるようにトリガーされます。

オーダーの取り消し

オーダーのバックオーダーまたはノードからのバックオーダー済み

予約の取り消し

2. アイテム・ベースの割り当てプロセスの実行

アイテム・ベース割り当てプロセスは、「一般」プロセス・タイプの下にあるアイテム・ベース割り当てエージェントによって実行されます。トリガーされたアイテムとノードの組み合わせごとに、エージェントは、そのアイテムとノードの組み合わせを含んでいる該当するオーダー明細またはオーダー明細予約をすべて検索し、それらのコミットされていない需要とコミット済み需要を、使用可能な供給に基づき、さらにユーザー構成の IBA 選択ルールまたは FIFO (先入れ先出し) IBA 選択ルールに基づいて、より適切なフォーマットに移動しようとしません。

その後、システムは、一致した供給の最初の出荷日で新規の正のオーダー明細予約を作成し、既存の需要の出荷日には負のオーダー明細予約を作成します。オーダーは、処理されると、保留状態にされ、オーダーのオーダー明細予約で変更が検出された場合にスケジュールが変更されます。再スケジュール・プロセスは、アイテム・ベースの割り当てプロセスで作成されたオーダー明細予約を使用することによって、オーダーの実際の再スケジュールを実行します。

注: オーダーをスケジュールリングのために保留状態にできるように、保留タイプをオーダー明細予約変更の変更タイプ用にセットアップする必要があります。

## ユーザー構成の IBA

ユーザー構成の IBA では、オーダーおよびオーダー明細にシーケンスと条件を指定するようにルールを定義できます。条件を作成することができ、この条件と一致するオーダーが優先されます。例えば、224 ページの表 28 では、Buyer1、Buyer 2、および Buyer3 が在庫に対して競合しています。

表 28. バイヤー組織コードを含む IBA の例

購入オーダー	製品	ノード	数量	ETA	販売オーダー	製品	ノード	数量	需要出荷日	オーダー日付	バイヤー組織
PO1	Item1	Node1	5	11/15/08	SO1	Item1	Node1	5	11/16/08	11/03/08	Buyer1
PO2	Item1	Node1	5	11/20/08	SO2	Item1	Node1	5	11/21/08	11/05/08	Buyer2
PO3	Item1	Node1	5	11/21/08	SO3	Item1	Node1	5	11/22/08	11/10/08	Buyer3
PO4	Item1	Node1	5	11/25/08	SO4	Item1	Node1	5	11/26/08	11/15/08	Buyer4

表 29 は、Buyer3 が優先されるように構成されている場合、Buyer3 の需要出荷日が 11/16/08 に変更され、Buyer3 が Buyer2 および Buyer4 より優先されることを示しています。ユーザー構成の優先順位が完了したため、FIFO 優先順位が有効になり、Buyer2 と Buyer4 が在庫に対して競合します。Buyer2 の方がオーダー日が早いいため、Buyer2 が早い需要出荷日を取得します。

表 29. ユーザー構成の IBA の結果

購入オーダー	製品	ノード	数量	ETA	販売オーダー	製品	ノード	数量	需要出荷日	オーダー日付	バイヤー組織
PO1	Item1	Node1	5	11/15/08	SO1	Item1	Node1	0	該当なし	11/03/08	Buyer1
PO2	Item1	Node1	5	11/20/08	SO2	Item1	Node1	5	11/21/08	11/05/08	Buyer2
PO3	Item1	Node1	5	11/21/08	SO3	Item1	Node1	5	11/16/08	11/10/08	Buyer3
PO4	Item1	Node1	5	11/25/08	SO4	Item1	Node1	5	11/22/08	11/15/08	Buyer4

## スケジュールの変更

オーダーのスケジュールの変更は、スケジュールリングにとって重要な属性が変更された場合に可能です。これは、オーダー保留および以下のユーザー構成を使用した場合にのみ可能です。

- ユーザーが保留タイプを作成する必要がある
- この保留によって、ReleaseOrder および CreateChainedOrder トランザクションでオーダーをピックできなくなる
- この保留が、スケジュールリング (オーダー明細予約変更、出荷先住所の変更、ノードの変更など) に影響する変更タイプによってトリガーされる
- この保留が SCHEDULE トランザクションによって解決可能である

**注:** 在庫の変更のためにオーダーのスケジュールを変更する必要がある場合は、スケジュール変更機能ではなく IBA を使用してください。IBA について詳しくは、221 ページの『アイテム・ベース割り当て』を参照してください。

スケジュールでオーダーがピックアップされるときに、スケジュールが解決するよう構成されているアクティブな保留がオーダーにあるかどうか確認されます。ある場合、スケジュールがスケジュール変更モードであることが確認されます。スケジュールが終了すると、オーダーを処理できなかった場合を除き、解決するよう構成されている保留はすべてクローズされます。



明細に変更があり、明細がスケジュール済みの場合、オーダー明細のスケジュールは変更されないままです。明細は保留になり、スケジュールが変更されます。スケジュールが終了すると、オーダーから保留タイプが削除されます。



---

## 第 10 章 付加価値サービス

---

### 付加価値サービスの概要

付加価値サービス (VAS) は、顧客の需要を満たすために実行されます。このサービスは、製品が顧客に配達される前に製品に対して実行されるアクティビティー、または顧客のサイトで実行される提供サービスです。

アクティビティーは、製造設備および中継センターまたは配送センター (distribution center) を含む、すべてのタイプの設備で実行できます。ベンダー・サイトで実行される VAS の例を、次に示します。

- ベンダーは、製造業者の仕様に基づいて、ライセンス・プレート番号 (LPN) ラベルをカートン (carton) に適用します。
- 製造業者は、小売店の要件に基づいて、価格チケットまたは SKU ラベルを適用します。
- 倉庫は、顧客要件を使用して、パレットを作成します。
- 倉庫は、特定の小売店への出荷前に、アイテムのクラスにセキュリティ・タグを適用します。
- 倉庫は、出荷前に、コンポーネントを組み立ててキットを作成します。

提供サービスは通常、顧客が追加手数料を支払って購入します。提供サービスとして提示されるアクティビティーは、次のものです。

- 顧客のホーム・シアター・システムの設置。
- サービス契約の一部として炉に対するメンテナンスの提供。
- 新規コンピューターへのソフトウェアのインストール、およびそのコンピューターをホーム・ネットワーク上で稼働させるための構成。

---

### 付加価値サービスの使用

付加価値サービス (VAS) は、顧客の需要を満たすために実行されます。このサービスは、製品が顧客に配達される前に製品に対して実行されるアクティビティー、または顧客のサイトで実行される提供サービスです。

アクティビティーは、製造設備および中継センターまたは配送センター (distribution center) を含む、すべてのタイプの設備で実行できます。ベンダー・サイトで実行される VAS の例を、次に示します。

- ベンダーによる、製造業者の仕様に基づく、ライセンス・プレート番号 (LPN) ラベルのカートン (carton) への貼り付け。
- 製造業者による、小売店の要求に基づく、値札ラベルまたは SKU ラベルの貼り付け。
- 倉庫による、顧客の要求に従った、パレットの作成。
- 倉庫による、特定の小売店に出荷する前の、アイテムのクラスに関するセキュリティ・タグの貼り付け。

- 倉庫による、出荷前のコンポーネントの組み立ておよびキットのビルド。

提供サービスは通常、顧客が追加手数料を支払って購入します。提供サービスとして提示されるアクティビティーは、次のものです。

- 顧客のホーム・シアター・システムの設置。
- サービス契約の一部として炉に対するメンテナンスの提供。

新規コンピューターへのソフトウェアのインストール、およびそのコンピューターをホーム・ネットワーク上で稼働させるための構成。

付加価値サービスが製造、ベンダー、または出荷の各設備内で使用される方法の例を以下に示します。

- **在庫に対する VAS** — 設備は、製品の予想需要に基づいて、在庫に対する VAS 操作を定期的に行います。例えば、倉庫は、ペン、鉛筆および消しゴムを別々に出荷します。また、この倉庫は、2 本のペン、2 本の鉛筆および 1 つの消しゴムから構成される学生パッケージも出荷します。
- **顧客に対する VAS** — このタイプの VAS 操作は、特別チケットやセキュリティのタグ付けなどの特定の顧客の要求を満たすために実行されます。Sterling Selling and Fulfillment Foundation では、バイヤーのためにノードで在庫をカスタマイズして、その顧客への割り当てを確保できます。このシナリオは、**顧客用**、または**バイヤー準拠**と呼ばれます。通常顧客は、これらの特殊アイテムを繰り返し購入します。このタイプの VAS は、需要予測に基づいて倉庫で実行され、**オーダー・サイクル・タイム**を短縮します。
- **オーダー用 VAS** — これは、以下の条件が満たされたときに適用可能です。
  - シャツへのモノグラムなどのアイテムをオーダーするときのカスタマイズを顧客が選択。Sterling Selling and Fulfillment Foundation は、作業オーダーを作成し、倉庫でこれらのサービスを実行し、在庫がそのオーダーに対してのみ割り当てられるようにします。これは、**オーダー用**とも呼ばれます。
  - 顧客が動的物理キット (dynamic physical kit) をオーダーし、そのアイテムに必須のコンポーネントを選択。Sterling Selling and Fulfillment Foundation は、作業オーダーを作成してキットを組み立て、組み立てられたユニットの在庫がそのオーダーに対してのみ割り当てられるようにします。これは、**受注生産**、または**オーダー用**とも呼ばれます。
  - パッケージ化が必要な入荷。受入済みの部品は、指定の単位にパッケージ化する必要があります。
- **キットの分解またはキット化解除** — これは、不要になった在庫のキットに対して実行されます。例えば、季節の販売促進の在庫をキット化解除し、そのコンポーネントを別のキットに使用、または個別販売できます。
- **単位 (UOM) 変換** — 倉庫は、作業オーダーを作成し、在庫をある単位から別の単位に変換できます。例えば、倉庫は、ねじの在庫を個別または 50 個のパックで追跡できます。作業オーダーを作成して、1 つの在庫単位を 50 個のパックに変換します。
- **贈答品用の包装サービス** — 出荷ノードは、贈答品用の包装の対象アイテムに対して贈答品用の包装サービスを提供できます。このフラグは、**ソーシング/スケジューリング・ノード属性**の参加者の設定中にノードに対して使用可能にされません。

## アイテム作成の延期

倉庫、および他の施設における付加価値サービス提供のアプローチの 1 つとして、オーダーを受けるまでカスタマイズ・プロセスを遅延させることがあります。これは「延期」と呼ばれます。倉庫には、顧客の製品カスタマイズの需要を待ち受ける汎用的な製品が保管されます。オーダーの受領後に、汎用的な在庫に追加のサービスが実行され、汎用的な在庫が顧客が要求する固有の在庫に変換されます。これにより、使用不可の在庫が発生するリスクを減らす一方で、オーダー・サイクル・タイムが増加します。次の図は、延期プロセスの例を示しています。

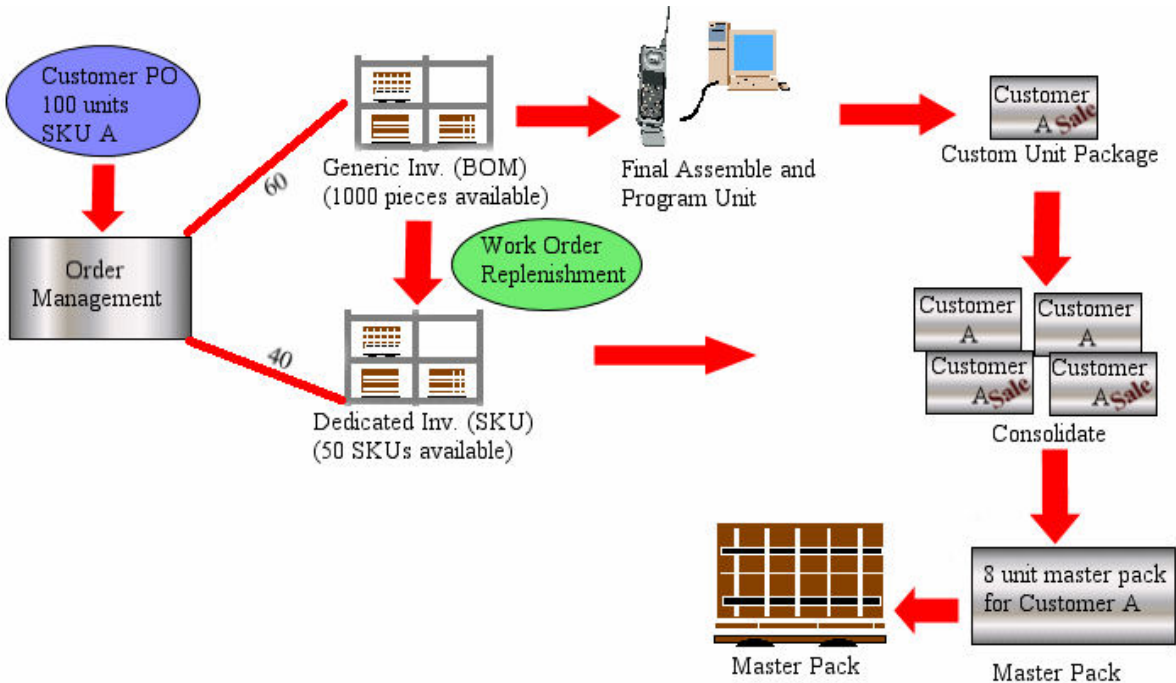


図 38. 延期プロセス

Sterling Selling and Fulfillment Foundation では、付加価値サービスが作業オーダーの作成によって実行されます。作業オーダーは、需要に応じて自動的に開始することも、ユーザーが手動で開始することもできます。Sterling Selling and Fulfillment Foundation では、VAS 部門による在庫取得、最終在庫へ向けた完了の記録、および製品の適切な部門への格納によって、作業オーダーの実行が管理されます。

## 作業オーダーの作成

Sterling Selling and Fulfillment Foundation の作業オーダーは、サービスの実行に必要なアクティビティーを取り込みます。作業オーダーは、以下のいずれかの方法によって Sterling Selling and Fulfillment Foundation で作成されます。

- 手動作成 - ユーザーは、再パッケージ化、単位変換、ラベル貼り付け、またはキット解除のための作業オーダー・コンソールを使用して要求を開始します。
- 在庫レベルに基づく自動作成 - 要求は、SKU に達する最小レベルまたは最大レベルに基づいて、Sterling Selling and Fulfillment Foundation の在庫モニターによって発行されます。

- オーダーに基づく自動作成 - 要求は、オーダーのアイテムの販売オーダーによって発行されます。これは、通常、**オーダー用**または**顧客用**のアイテムに対して行われます。

作業オーダーが作成される時、これは、以下のタイプのサービスの 1 つ以上から構成されます。

- キット化サービス - キット・アイテムのコンポーネントのアセンブルです。
- キット解除サービス - キット・アイテムの逆アセンブルです。これは、別のオーダーを完成させるため、個別コンポーネントを取得するために実行できます。
- コンプライアンス・サービス - **顧客用**オーダーに関して特定のバイヤーに対してアイテムを供給する際に実行される、付加価値サービスです。
- 在庫の変更 - 在庫をある単位から別の単位に変換します。
- 提供サービス - 顧客のサイトで実行されるサービスです。
- 配送サービス - 顧客のサイトへの製品配達のために行われるサービスです。

1 つ以上のサービスを 1 つの作業オーダーに組み込み、オーダーを実行する順序を示すシーケンス番号を割り当てることができます。各サービスには、1 つ以上のサービス・アクティビティを組み込むことができます。サービス・アクティビティは、コンポーネントのアセンブル、ロゴの貼り付け、コンポーネントの包装などのアクティビティのカテゴリを指定します。提供サービスは、他のサービスと結合されません。

---

## 作業オーダー保留タイプ

作業オーダーをライフサイクルのある時点で保留する必要がある場合があります。例えば、ある複雑なサービスでは、多数の互換性のある製品、サービス、およびサービス・ツールを必要とする場合があります。管理者は、予定を取得する前に、このようなサービスの作業オーダーを確認する必要がある場合があります。

139 ページの『オーダー保留処理』で説明しているオーダー保留タイプ機能も作業オーダーで使用できます。唯一の相違点は、作業オーダーをドラフト・ステータスにできないことです。したがって、作業オーダー作成時、別の保留タイプの解決時、または変更タイプを使用した場合のみ、保留を自動的に作成できます。

---

## サービス作業オーダーのタイプ

サービス作業オーダーは、作業オーダー・サービス明細、配達明細、およびその作業オーダーに対して確保する必要があった予定で構成されます。サービス作業オーダーとその他の付加価値サービスの重要な違いは、サービス作業オーダーに関連付けられているサービスは顧客のロケーションで顧客に対して提供されるのが一般的なのに対し、付加価値サービスは倉庫または他のアセンブリー・ロケーションで実施されるのが一般的である点です。これらはこのように異なりますが、多くの企業では、提供サービスを、セラーが実施できる作業オーダーまたはサード・パーティーが実行できる作業オーダーとして取り扱っています。

サービス作業オーダーに対してサービスの管理者を指定できます。通常、管理者は、特定のノードのセットから発生した作業オーダーの管理に責任を持ちます。サ

サービス作業オーダーにストア管理者が指定されている場合は、そのユーザーが管理しているリソース・プールに対してしか予定を確保できません。

管理者は、優先順位順に次の 3 つのレベルで設定できます。

- 特定のノードとセラー組織の組み合わせに対して
- 特定のリソース・プールに対して
- ノードに対して

例えば、特定のサービス作業オーダーが特定のセラー組織に発行された場合、そのノードに対して定義されているデフォルトの管理者よりも、そのセラーとノードの組み合わせに対して定義されている管理者が選択されます。

サービス作業オーダーは次の 2 つの基本的なタイプに分割されます。

- 提供サービス作業オーダー
- 配送サービス作業オーダー

## 単一作業オーダーでの複数のサービス明細

単一の作業オーダーに複数のサービス明細を設定できます。販売オーダーのサービス明細のそれぞれが、作業オーダーのサービス明細として示されます。

例えば、顧客が TV、レシーバー、スピーカーをオーダーし、その各製品の取り付けが個別のサービス明細としてモデル化される場合を考えてみましょう。顧客が 1 回の訪問ですべての取り付けを要求した場合は、この 3 製品の取り付けを 1 件の作業オーダーに記載できます。3 つのすべての取り付けサービスに対して、1 件の予定が作成されます。この例では、予定の作成時にこれらの 3 つのサービスすべての提供が可能なリソース・プールのみが考慮されます。したがって、顧客が後日、予定の変更を必要とした場合に必要になる変更は 1 つのみです。

複数のサービス明細が含まれるサービス作業オーダーは、サービス・アイテムに関連付けられたサービス・タイプに基づいて、最上位のサービス・レベルが設定されます。さらに、作業オーダーと同じアイテム・グループ・コードを持つサービス・アイテムは、サービス・レベルの決定で考慮されます。例えば、提供されたサービス作業オーダーにサービス・レベルが「低」の提供サービス・アイテムとサービス・レベルが「高」の配送サービス・アイテムが記載されている場合、作業オーダーのサービス・レベルは「低」になります。このサービス・レベルは、サービス明細の追加または削除が発生するたびに再計算されます。

## サービス・リソース付きの作業オーダー

作業オーダーが複雑で、さまざまなスキルを持つ人を必要とする場合、リソース・プール・ブロッキングでは不十分であり、サービス・リソースに対して予定を行うことが適切です。リソースは、提供サービスまたは配送サービスのいずれかを提供できるチームです。サービス・リソースの定義について詳しくは、「*Sterling Selling and Fulfillment Foundation 統合在庫管理 構成ガイド*」を参照してください。

リソース・プールがリソースを定義する場合には必ず、カレンダー日付またはタイム・スロットのいずれかに基づいて予定をリソースに対して行うことができます。例えば、リソースの作業オーダーの予定は、午前 8:30 から午後 5:30 の間です。

## 複数日の作業オーダー

単一の作業オーダーに複数の予定を作成する機能は、数日に広がるジョブがある場合に役立ちます。複数の予定を作成するには、作業オーダーの作成時に、「複数の予定」チェック・ボックスにチェックを付ける必要があります。このフラグが有効化されると、追加の予定を加えたり、既存の予定を変更することができます。

注: 有効化された「複数の予定」チェック・ボックスを無効にすることはできません。

一般に複数の予定を使用する場合は、ジョブに経験豊富な専門家がスケジュールされるため、作業オーダー能力の過剰割り振りが可能です。例えば、カスタム・ジョブの完了に 35 時間を要する場合に、特定のリソースに対して、各日を 8 時間とする合計 5 日の構成を割り振ることができます。

## 提供サービス作業オーダー

作業オーダーのそれぞれの提供サービス明細は、オーダーの 1 明細に対応します。顧客は、製品をオーダーする場合と同様に提供サービスをオーダーします。サービスは、Sterling Selling and Fulfillment Foundation においてサービス・アイテムとしてモデル化されます。サービスはそれだけでオーダーされることも、購入された製品と一緒にオーダーされることもあります。提供サービス明細を作業オーダーに含めても、該当する提供サービス明細に関連付けられる、すべての製品明細が作業オーダーで完了したことを意味するわけではありません。

### 配達予定の製品がある提供サービス作業オーダー

顧客が製品を購入し、その設置と配達を一緒にオーダーした場合には、設置者が製品を配達します。そのような状況に対応するため、「作業オーダーの詳細」画面の「配達予定の製品 (Products To Be Delivered)」オプションを使用できます。その結果、提供サービスと配送サービスの両方を提供する 1 つの明細が作成されます。

作業オーダーの明細は、複数の販売オーダーから構成できます。例えば、製品交換の場合では、古い製品の撤去と取り外しを、新しい製品の配達と設置と同じ作業オーダーに含めることができます。ただし、予定に対応するために使用可能なリソース・プールを探す場合は、サービス・アイテムまたはサービス・スキルの配送サービス明細は無視されます。

作業オーダーが提供サービスと配送サービスの両方の明細を含む場合、その作業オーダーは提供サービスと見なされます。サービス・アイテムの単位は、提供サービスのキャパシティの単位と同じです。ServiceItemGroupCode は「PS」です。

### 作業オーダーと交換オーダー (exchange order) の同期

作業オーダーの明細は、複数の販売オーダーから構成できます。例えば、顧客が新しく設置した皿洗い機を返品し、別の色のものと交換する場合、古い皿洗い機の撤去とピックアップおよび皿洗い機自体を含む返品オーダーが作成されます。また、配達および設置のサービス付きでの新しい皿洗い機を含む交換オーダー (exchange order) も作成されます。ユーザーは、以下の項目を含む 1 つの作業オーダーを生成できます。

- 新しい皿洗い機
- 皿洗い機の配達 (配送サービス)



- 皿洗い機の設置 (提供サービス)
- 古い皿洗い機
- 皿洗い機のピックアップ (提供サービス)
- 皿洗い機の撤去 (提供サービス)

これにより、ユーザーは、両方のタスクに対して 1 つのみの予定を作成することが可能になります。

## 配送サービス作業オーダー

作業オーダーには、製品が関連付けられている配送アイテムを含めることができます。配達される製品が大きくて、設置者がそれらを配達できない場合、あるいは顧客が設置なしで配達のみをオーダーしている場合は、これはきわめて重要です。作業オーダーに配送サービス明細を含めても、その配達明細と関連付けられているすべての製品明細がその作業オーダーについて配達されることを意味するわけではありません。

作業オーダーには、独立した配送サービス明細も含めることができ、後で必要であれば製品明細と関連付けることができます。

作業オーダーに含まれている製品明細のみが、作業オーダー実行の一部として配達されます。

## 作業オーダー・タイプの動的決定

作業オーダー・タイプ (提供サービス (PS) または配送サービス (DS)) は、作業オーダーに存在する明細のタイプに基づいて、自動的に決定されます。この決定は、オーダー作成時、または作業オーダーでの変更中に発生する可能性があります。作業オーダー・タイプは、以下のロジックに基づきます。

- 少なくとも 1 つの PS 明細が存在する場合は、作業オーダーは PS 作業オーダーであると見なされます。
- 存在するすべての明細が DS 明細である場合は、作業オーダーは DS 作業オーダーであると見なされます。

作業オーダー・タイプは、明細が作業オーダーに追加されるか、または作業オーダーから除去されるときに、動的に切り替えることができます。例えば、PS 明細と DS 明細の両方が含まれている PS 作業オーダーで、すべての PS 明細を除去すると、その作業オーダーは DS 作業オーダーになります。

作業オーダー・タイプを更新すると、アイテム・グループ・コードが変更されるため、以下の情報も自動的に更新されます。

- サービス・アイテム・グループ・コード
- 必須キャパシティー
- キャパシティーの単位
- 複雑さレベル

作業オーダー・タイプの変更時に、作業オーダー予定のリソース・プールも更新されます。

リソース・プールが予定上で手動で渡された場合は、作業オーダー・タイプの変更後に、手動で予定を取り戻す必要があります。

---

## サービス作業オーダーの作成

サービス作業オーダーは、製品に関連付けられているサービス明細が存在する場合には、販売オーダーから作成されます。作業オーダーが作成されてもオーダーのステータスに変化はありません。作業オーダーを作成できるのは、まだスケジュールされていないオーダー明細に対してです。各作業オーダーは単一のオーダー明細に関連しています。ただし、完全に出荷およびキャンセルされたオーダー明細の作業オーダーを含めることはできません。この考慮事項に注意しておけば、提供サービス、配送サービス、および製品明細を好きな数だけ作業オーダーに追加できます。

さらに、特定のアイテムを、作業オーダーに追加可能なサービス・ツールとして構成することもできます。一般的には、これらは、作業オーダーの必要なタスクを実行するためにサービス・リソースが使用するアイテムです (例: はしご、工具一式)。

作業オーダーの作成時には、次の検証が実行されます。

- 提供サービス明細すべての単位が同じである。
- 作業オーダーが、提供サービス明細のない配送サービス・オーダーの場合、サービス明細すべての単位が同じである。
- 「配達する製品 (Products To Be Delivered)」オプションは、提供サービスまたは配送サービスの作業オーダーに対してしか設定できない。
- 作業オーダーに含まれているすべての明細で出荷 (「出荷先」) の住所が同じである。

## 提供サービスまたは配送サービスの作業オーダーの決定

作業オーダーのタイプは、次の基準を使用して定義されます。

- 配送サービス明細および製品明細のみが作業オーダーに含まれている場合には、その作業オーダーは配送サービス作業オーダーと呼ばれます。
- 次の組み合わせで、少なくとも 1 つの提供サービス明細が作業オーダーにある場合には、その作業オーダーは提供サービス作業オーダーです。
  - 提供サービス明細。
  - 提供サービス明細および製品明細。
  - 提供サービス明細、製品明細、および配送サービス明細。

作業オーダーに配送サービス明細のみが含まれる場合には、提供サービス明細を追加することはできません。作業オーダーが提供サービス作業オーダーの場合には、配達明細を追加できますが、作業オーダー・タイプは変化しません。

## 作業オーダー・ノードの決定

各作業オーダーは、作業オーダー実行を所有している特定のノードに対して作成される必要があります。Sterling Selling and Fulfillment Foundation は、以下の条件に基づいて、作業オーダー・ノードを決定します。

- 作業オーダー作成中にノードが指定された場合、そのノードが使用されます。

- 任意のサービス明細でノードが指定されている場合、そのノードが作業オーダー・ノードとして使用されます。
- キャパシティーがオーダー明細によってブロックされている場合、キャパシティーのブロックに使用されたリソース・プールのノードが使用されます。
- 作業オーダーまたはサービス明細に対して事前定義されたノードがない場合、(提供サービス作業オーダーの) 提供サービスまたは (配送サービス作業オーダーの) 配送サービスのいずれかのソーシング・ルールが使用されます。この場合、1 次ノードが作業オーダー・ノードとして使用されます。

注: 上記の状態のすべてにおいて、サービス明細に複数のノードが指定されている場合、作業オーダー・ノードを作成できません。したがって、作業オーダー作成中に作業オーダー・ノードを指定しない場合、すべてのサービス明細において 1 つのノードのみを引用できることを確認する必要があります。

ノードは、作業オーダーでは変更できません。ノードを変更する必要がある場合、作業オーダーをキャンセルし、再作成する必要があります。

## 作業オーダーのプロバイダーの決定

Sterling Selling and Fulfillment Foundation は、以下の条件に基づいて、作業オーダーのプロバイダー組織を決定します。

- 作業オーダーが提供サービス作業オーダーの場合、作業オーダーの出荷ノードの所有者がプロバイダー組織になります。プロバイダー組織は、常に作業オーダー作成時に決定されます。したがって、プロバイダー組織は、作業オーダー作成後には変更できません。
- 作業オーダーが配送サービス作業オーダーの場合、プロバイダー組織は、作業オーダー作成時に指定できます。ただし、作業オーダーの出荷ノード用に定義されたリソース・プールがプロバイダー組織にあることを確認するための検証は行われません。

この条件を満たした後でも、プロバイダー組織を決定できない場合、作業オーダーのプロバイダー組織は空白のままになります。modifyWorkOrder API を使用してプロバイダー組織を渡すか、または最初の予定を行うことによって、プロバイダー組織を後で設定できます。後者の場合、リソース・プールのプロバイダー組織が作業オーダーとみなされます。

## 作業オーダーのキャパシティー計算

作業オーダーに追加された各サービス明細 (提供サービス明細と配送サービス明細が同じ作業オーダーに混在する場合を除く) によって、キャパシティーが計算されます。アイテムのキャパシティーは、製品関連付けまたはオーダーされた数量のいずれかを使用して計算できます。

例えば、提供サービス明細に 2 つの関連付けられた製品明細があり、各明細に 3 つの数量があるとします。この場合、作業オーダーに必要なキャパシティーの合計は、 $2 \times 3 \times CapacityMultiplier$  になります。CapacityMultiplier は、提供サービスと配送サービスの作業オーダー間で変化する乗算係数です。

## 配送サービス作業オーダーのキャパシティー計算

$ComputedCapacity = Max$  (各配送サービス明細の固定キャパシティー) +

$Sum$  (すべての配送サービス明細の可変キャパシティー)

固定キャパシティーおよび可変キャパシティーの決定には、製品カタログが使用されます。キャパシティーがオーダーされた数量に基づいて評価される場合、計算されるキャパシティーは、固定キャパシティーについて計算されます。

また、配送サービス明細の必要数量は、作業オーダーに含まれる製品明細に基づいて計算されます。ただし、サービス明細が追加または削除されると、キャパシティーのオーバーライド・オプションが作業オーダーで有効になっていない場合、必要数量は再計算されます。配達する必要がある製品が販売オーダー上の配送サービス明細に関連付けられていない場合、可変キャパシティーに対して、関連付けが製品カタログから計算されます。

配送サービス作業オーダー上に複数の配送サービス明細が存在する場合、可変キャパシティーは、製品カタログに関連付けられたアイテムについて計算されます。配送サービス明細が複数ある場合、配送サービス作業オーダーのキャパシティーは、ランダムに選択されます。製品が、カタログ内の配送サービス・アイテムとまったく関連付けられていない場合、その製品のキャパシティーは 0 として計算されます。

また、配送サービス作業オーダーの必要キャパシティーの計算時に、追加要因を考慮することも可能です。追加の固定キャパシティーおよび追加キャパシティーについて詳しくは、208 ページの『リソース・プール (Resource Pools)』を参照してください。キャパシティーのインパクトについて詳しくは、「*Sterling Selling and Fulfillment Foundation* 分散オーダー管理 構成ガイド」を参照してください。

## 提供サービス作業オーダーのキャパシティー計算

$ComputedCapacity = Sum$  (すべての提供サービス明細上の固定キャパシティー) +

$Sum$  (すべての提供サービス明細の可変キャパシティー)

オーダーされた数量に基づいて計算される場合、キャパシティーは固定とみなされます。

$RequestedCapacity = ComputedCapacity + AdditionalRequestedCapacity$

$AdditionalRequestedCapacity$  を使用して、必要な追加キャパシティーを手動で指定できます。この値には、正または負の数字を使用できます。例えば、インストール作業者が製品を配達し、現地の管理者が、ハブ (hub) からの製品のピックアップに追加の 30 分を要することを認識しているとします。この場合、追加の 30 分が  $AdditionalRequestedCapacity$  とみなされます。

ただし、この追加キャパシティーは、要求数量を手動で指定することによって、作業オーダー作成時にオーバーライドできます。このような状況の事例は、必要なキャパシティーをサービス・アイテム関連付けに基づいて算出できない、複雑なジョブの場合に発生します。

要求数量がオーバーライドされると、オーダー明細上の数量が変更されても、要求されたキャパシティーは変更されません。ただし、最新のサービス明細がキャンセルされると、作業オーダー全体がキャンセルされます。

QuantityRequested は、作業オーダーの完了に必要なキャパシティーの数量を指します。予定の計画時に、ブロックされたキャパシティーは、AllocatedQuantity として提供されます。多くの場合、予定の作成処理が完了した後は、割り当て数量および要求数量は変わりません。

提供サービス作業オーダーの必要キャパシティーの計算時に、追加の要因を考慮することも可能です。キャパシティーのインパクトについて詳しくは、「*Sterling Selling and Fulfillment Foundation 分散オーダー管理 構成ガイド*」を参照してください。

## サービス作業オーダーの製品予約

製品予約の試行は、スケジュール・トランザクションによって行われます。このトランザクションは、販売オーダーに指定された出荷指定日に基づいて製品を予約するために呼び出されます。製品のリリースをスケジュールできない場合には、スケジュール・トランザクションは、その製品明細を予約済みステータスにします。

製品明細が予約されると、作業オーダー明細の出荷ノードが未確定の出荷ノードとして指定されます。作業オーダーの製品明細の出荷ノードを決定する際、Sterling Selling and Fulfillment Foundation は次を考慮します。

- 製品明細に定義済みの出荷ノードがある場合には、そのノードが製品予約に使用されます。
- 製品明細にノードが指定されていない場合で、それが配送サービス作業オーダーの場合には、配送サービス明細の出荷ノードが使用されます。
- 製品明細にノードが指定されていない場合で、それが提供サービス作業オーダーの場合には、次が考慮されます。
  - サービス設置者のノードが製品の配達ノードとして使用されます。このオプションが選択されると、作業オーダー明細の出荷ノードは、製品予約の試行に使用されます。このオプションは、在庫が物理的に設置者のハブ (hub) にある場合に使用できます。
  - 製品の配達ノードを決定する標準ルールが使用されます。サービス設置者が製品を配達するか、製品が別に配達されるかが重要でない場合には、このオプションを使用できます。

---

## 作業オーダーの確認

作業オーダーを確認するには、確認対象の予定、予定の結果、および実行の詳細を指定する必要があります。ただし、作業オーダー上の配送サービス明細は、完了としてマークは付けられません。それらは、「配達済み」のステータスに移動します。すべての提供サービス明細が完了し、すべての製品が配達されたときに、作業オーダーは完了とみなされます。1 つの作業オーダーの確認で、複数の予定を完了することができます。作業オーダーの確認のために、confirmWorkOrder API が呼び出された場合、すべての未完了の予定はキャンセルされ、割り当てられたキャパシティーは除去されます。

Sterling Selling and Fulfillment Foundation では、予定の詳細を指定せずに、作業オーダーを確認することもできます。作業オーダー完了の一部として、すべてのサービス明細 (提供サービスおよび配送サービス) が完了し、すべての製品が配達された場合、作業オーダー上の未完了の予定は、以下のように解決されます。

- 予定が 1 つある作業オーダーの場合、その予定は完了としてマークが付けられません。
- 複数の予定がある作業オーダーの場合、過去のすべての予定は完了としてマークが付けられ、将来の予定はキャンセルされます。

---

## 作業オーダーのキャンセル

作業オーダーのキャンセルは、サービス明細のステータスに依存します。サービス明細のいずれかが完了済みまたは製品明細のいずれかが配達済みの場合、作業オーダーは完了済みとしてマークが付けられます。オープンである予定のみがキャンセルされ、関連するキャパシティー予約が削除されます。

製品出荷後に配達のカンセルを要求することも可能です。オーダーしたアイテムが不要であると顧客が判断したが、そのアイテムが既に出荷済みの場合、配達停止要求を作成できます。この要求が成功すると、返品オーダーが自動的に作成されます。

---

## 予定ステータス

予定ステータスは、作業オーダー予定のステータスを表します。予定ステータスは、ブロックされたキャパシティーと関連付けられています。予定開始時刻は、現在キャパシティーをブロックしている、すべての予定開始時刻の中で最小の時刻を表します。予定終了時刻は、現在キャパシティーをブロックしている、すべての予定終了時刻の中で最大の時刻を表します。予定がキャパシティー・チェックなしで作成されている場合、常にそれらをオーバーライドできます。

予定に使用可能なステータスは以下のとおりです。

- オープン
- 完了済み
- 失敗
- キャンセル済み (Cancelled)

### オープン

作業オーダーに対して予定が取得されると、予定ステータスは「オープン」に設定されます。キャパシティーがオーバーライドされた場合、またはリソース・プールに確認が必要な場合でも、これは該当します。このステータスは、予定が取得済みであることを示します。

### 完了済み

予定が正常に実行されたと記録されたとき、予定ステータスは、「完了済み」に変更されます。予定に一度完了済みとマークが付けられると、予定の詳細を変更することはできません。

## 失敗

顧客が利用不可の場合、予定は「失敗」に変更されます。予定に一度「失敗」とマークが付けられると、その予定を他のステータスに変更することはできません。顧客が後日利用可能になった場合は、予定を再作成する必要があります。

## キャンセル済み (Cancelled)

作業オーダーのキャンセル、または作業オーダーの完了の結果、予定がキャンセルされた場合、予定ステータスは「キャンセル済み (Cancelled)」に設定されます。作業オーダーが確認中で、すべてのサービスが製品の配達とともに完了している場合、失敗、または完了済みでないすべての予定は、「キャンセル済み (Cancelled)」とマークされます。

## 予定を入れるための手順

作業オーダーが作成されると、作業オーダーに直接予定を作成できます。予定は、リソース・プールまたはサービス・リソースに対して作成されます。サービス明細のすべてで必要とされるスキルまたはサービス・アイテムのすべてを提供するリソース・プールのみが、作業オーダーに含まれます。複数のリソース・プールから選択できるようにするには、リソース・プールの利用状況を確認する間に、製品の在庫状況が無視するように選択できます。予定は、スロットまたはカレンダーのいずれかに基づくようにできます。

### スロットに基づく予定

予定がスロットに基づく場合には、予定の開始時刻および終了時刻は実際の時間スロットを示します。リソース・プールが時間に基づく場合には、約束した予定の開始日と終了日の間の差は必須キャパシティーを示します。スロットに基づく予定は、リソース・プールまたはサービス・リソースに対して作成できます。

### カレンダーに基づく予定

時間以外に基づくリソース・プールに対してカレンダーに基づく予定を作成する場合には、必須キャパシティーを指定します。サービス・リソースがリソース・プール、キャパシティー、サービス地域、およびスキルと共に指定された場合には検証が行われます。指定されたサービス・リソースまたはリソース・プールで必須キャパシティーがブロックされます。

### 作業オーダーの作成と予定計画の例

次の例は、シンプルな作業オーダーおよび予定計画の作成を説明したものです。

顧客が TV と一緒に、合計 2 時間かかる設置サービスをオーダーします。その 2 時間のうち、0.5 時間が固定キャパシティーで 1.5 時間が可変キャパシティーです。

さらにその顧客は DVD プレイヤーと一緒に、1 時間の固定キャパシティーと 0.5 時間の可変キャパシティーを要する設置サービスもオーダーします。販売員がその顧客のオーダーを受け、両方の設置サービス明細 (TV 設置と DVD 設置) を選択して作業オーダーを作成し、出荷ノードを自身のストアに指定します。TV 設置と DVD 設置の現在の作業オーダー・フィールドには WorkOrderKey と表示されます。

出荷ノードの所有者としてプロバイダー組織が指定され、作業オーダーが作成されます。要請済みキャパシティーは 3 時間 (必要数量 2 時間と 1 時間の合計) です。販売員は予定を確認し、顧客に利用可能なスロットを提示します。11 月 12 日の 9:00 AM から 12:00 PM のスロットが選択されます。予定ステータスは「未完了」と表示され、その予定が作業オーダー・ヘッダーと両方のオーダー明細 (TV 設置と DVD 設置) に表示されます。そのため、指定されたりソース・プール (この場合は小売店) の 11 月 30 日の 9:00 AM から 12:00 PM の 3 時間のキャパシティーがブロックされます。

### 予定日前の事前電話

かなり先の将来日付で予定が作成された場合は、サービス日付が近づいた時点で、顧客へのお知らせとして事前電話を行うことが推奨されます。Sterling Selling and Fulfillment Foundation では、必須ではない事前電話の管理およびモニターのための高度なプロセスがサポートされます。

Sterling Selling and Fulfillment Foundation では、事前電話が失敗した場合の予定確認もサポートされます。

サービス・プロバイダーが顧客への事前電話を行うと、次のいずれかの結果が発生します。

- **事前電話の正常完了** — 事前電話が顧客に通じ、顧客が当初の予定を問題なく確認した場合に発生します。
- **事前電話の失敗** — 事前電話の実施時に、顧客に電話が通じなかった場合に発生します。ただし、後の時点で顧客に対して予定を再確認することはできます。Sterling Selling and Fulfillment Foundation では、事前電話の失敗時における予定の確認もサポートされます。
- **予定の変更** — 事前電話の実行時に、顧客が予定の変更を要求する場合があります。新規の予定がどの程度将来の日付であるかに基づいて、その後、新規の予定日が近づいた時点で、別途の事前電話を実行できます。

---

## 提供サービスおよび配送サービスのインボイスの作成

提供サービスおよび独立した配送サービスのインボイス作成は、販売オーダー明細のオーダー済み数量全体に基づいて行われます。

関連製品のインボイスが作成されるときに、関連製品明細のある配送サービスのインボイスが作成されます。オーダー済み数量が、配送サービス明細の作成時に指定された数量と同じである場合は、完全数量のインボイスが、製品明細と同時に作成されます。

例えば、以下の 2 つの事例について考えます。

### 事例 1: 入力済み数量戦略

関連製品明細のある配送サービス明細のインボイスを作成するために入力済み数量戦略が使用される場合は、製品明細のインボイスが作成されるときに配送サービス明細数量全体のインボイスが作成されます。



製品数量が 10 であり、配送サービス明細に入力された数量が 5 であるとして、製品数量のインボイスが一部または全体のどちらについて作成される場合でも、配送サービス明細のインボイスは、製品数量のインボイスが作成されるときに数量全体について作成されます。

## 事例 2: 関連付けられた数量戦略

関連製品明細のある配送サービス明細のインボイスを作成するために関連付けられた数量戦略が使用される場合は、インボイスが作成される製品数量に基づいて、配送サービス明細数量についての請求済み数量が計算されます。

製品数量が 10 であり、配送サービス明細数量が 5 であるとして、配送サービス明細のインボイスは、最初の 2 つの製品単位のインボイスが作成されるときに 1 つについて作成されます。この場合、製品数量および配送サービス明細数量は、それぞれ、2:1 の比率に維持されます。

---

## 作業オーダー・パイプライン

ビジネス・プロセスをモデル化する場合、作業オーダーにパイプラインを作成できます。パイプラインは、ライフサイクル時にドキュメントが経るさまざまなステータスで構成されます。構成するパイプラインに関連するトランザクションを設定することもできます。

作業オーダー・ドキュメントは、付加価値サービス・パイプラインを使用します。作業オーダーは、オーダー・パイプラインのスケジューリング・アクティビティーの一部にすることもできます。

## 作業オーダーの付加価値サービス・パイプライン

作業オーダーは、その作成から始まり、一連のトランザクションとステータスが続き、完了します。この一連のトランザクションと作業オーダー・ステータスは、付加価値サービス (VAS) パイプラインと呼ばれます。VAS パイプラインを構成して、特別なビジネス要件を満たすことができます。ただし、各 VAS は、通常、作業オーダーを作成するトランザクションで始まり、作業オーダーが完了したときのトランザクションで終了します。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。オーダー・ステータスは、作業オーダーの状態を示します。

以下のステータスは、システム内での構成方法に応じて、VAS パイプラインで使用できます。

- 作成済み - 作業オーダーは作成済みです
- 確認済み - 作業オーダーのサービスは完了しました
- キャンセル - 作業オーダーのサービス要求はキャンセルされました

242 ページの図 39 は、Sterling Selling and Fulfillment Foundation のデフォルトの VAS パイプラインを示しています。

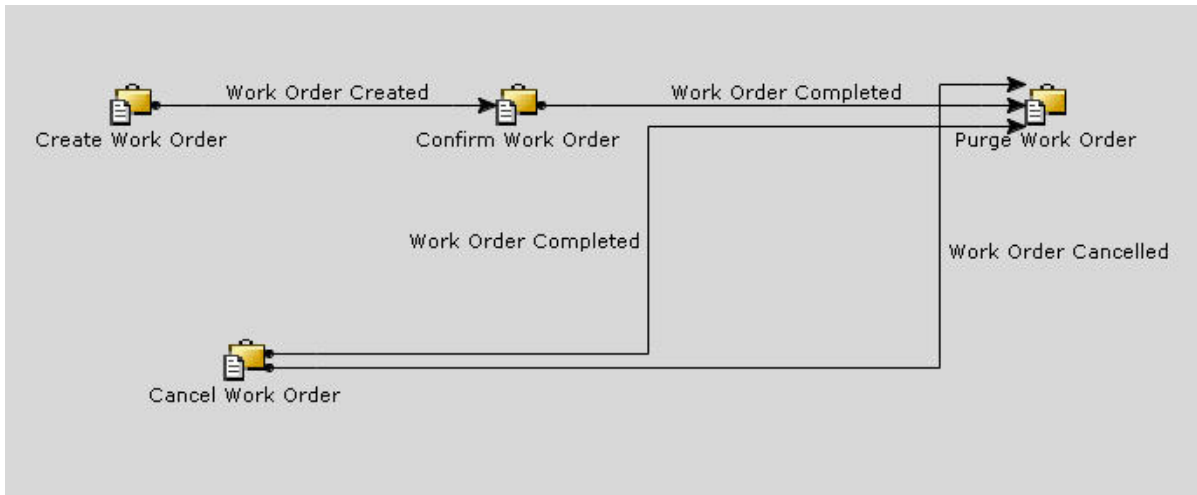


図 39. VAS パイプライン

## コンプライアンス・サービス

コンプライアンス・サービスは、バイヤーの詳細な仕様を満たす、バイヤーのためのアイテムのカスタマイズを可能にします。例えば、ベンダーは、百貨店のチェーン・ブランド *Big Store* でカスタマイズできる、冷蔵庫を供給するとします。冷蔵庫が出荷される時、その製品が百貨店のブランドであることを識別するための小さなエンブレムがあり、保証やメール・イン・リポートなど、すべてのサポート関係物は、元のベンダーではなく、百貨店に固有のものになります。

バイヤー・コンプライアンスをサポートするために、バイヤー組織構成で、`Requires_VAS_Compliance` フラグを有効にする必要があります。これは、バイヤーの仕様に合わせて作成されたアイテムのオーダーを、バイヤーが保有できることを示しています。

バイヤーに関する情報を構成する際、コンプライアンス・サービスが実行されていないアイテム分類を識別できます。このコンプライアンス・サービスは、バイヤーの要件を満たすために、そのアイテム分類でアイテムを準備するために実行されなければならないアクティビティを定義します。コンプライアンス・サービスの実行が必要なアイテムに、キットは使用できません。

例えば「音楽 CD」という分類があるとします。このバイヤーが購入するすべての音楽 CD について、新しい音楽サービスを宣伝するステッカーが CD ケースに貼られます。このシナリオをサポートするようにバイヤーを構成するときに、コンプライアンス・サービスが作成され、ステッカーの貼付方法が説明されます。次に、このコンプライアンス・サービスは、バイヤーのコンプライアンス・サービス構成内のアイテム分類「音楽 CD」に関連付けられます。

## コンプライアンス・サービスの構成

コンプライアンス・サービスの構成時に、コンプライアンス・サービスの実行時に作業オーダーを生成するかどうか指定できます。実行するタスクが、在庫スケジューリングまたは全体処理に影響を与えないため、特定の作業オーダーが必要でない場合があります。

例えば、バイヤーがすべての CD プレイヤーにメールイン・リポート・クーポンを追加したいとします。この場合、このプロセスを記述したコンプライアンス・サービスは、特定の作業オーダーが生成されないように構成できます。在庫スケジューリング全体に変更がないためです。

別のシナリオでは、シャツにフットボール・チームのロゴを付けるとします。これは、在庫状況に影響を与えるため、作業オーダーを生成するように構成しなければなりません。両方のシナリオ共に、処理の追跡が可能であり、コンプライアンス・サービス処理の実行回数最適化が使用可能です。このサービスに作業オーダーが生成される際、販売オーダーには反映されません。また、作業オーダーがキャンセルされた場合、販売オーダーにその変更が戻されて反映されることはありません。オーダーがリリースされ、在庫チェックが必要な場合、作業オーダーがキャンセルされ、使用可能な在庫がないため、販売オーダーは自動的にバックオーダー (back ordered) されます。このケースで在庫チェックが必要ない場合、バックオーダーするには、販売オーダーを手動で変更する必要があります。

## コンプライアンス処理の最適化のための実行数量の使用

コンプライアンス・サービスが構成された場合、実行数量を構成して、特別なアイテムの処理を制御できます。実行数量は、アイテムのオーダーに対応するための在庫が不足している場合に、作成する必要のあるアイテム数の計算に使用されます。

コンプライアンス・サービスを必要とするアイテムのオーダーが行われた場合、システムは、これらの特別なオーダーに対応するためにオーダーで指定された出荷ノードに取り置かれた (または区分された) 使用可能な在庫に対してスケジューリングしようとしています。在庫が出荷ノードにない場合、システムは、構成された調達ノードにある任意の区分された在庫に対してスケジューリングしようとしています。これでもオーダーに対応する十分な在庫がない場合、そのアイテムのコンプライアンス・サービスに基づいて、作業オーダーが作成されます。この作業オーダーが使用されて、このアイテムを供給する特定の要求を管理します。

例えば、大規模ストアは、毎週 150 台を超える冷蔵庫を購入します。この冷蔵庫に対するコンプライアンス・サービスは、作業オーダーが作成されるときに 100 台の冷蔵庫を作成するように構成されます。

別のバイヤー (地元のストア) は、毎週 10 から 15 台の冷蔵庫を購入します。この冷蔵庫にコンプライアンス・サービスを構成する場合、新しい作業オーダーが作成されるときに冷蔵庫を 10 台だけ作成する必要があります。表 30 は、毎週の購入およびコンプライアンス・サービスの構成のまとめです。

表 30. コンプライアンス・サービスと在庫

バイヤー	毎週購入する 冷蔵庫	コンプライアンス・サービス: 実行数量
大規模ストア	150	100

表 30. コンプライアンス・サービスと在庫 (続き)

バイヤー	毎週購入する 冷蔵庫	コンプライアンス・サービス: 実行数量
地元のストア	10-15	10

カスタマイズされた冷蔵庫の現在の在庫 (current inventory) については、表 31 で説明します。

表 31. カスタムの冷蔵庫の現在の在庫 (current inventory)

バイヤー	手持ちあり
大規模ストア	21
地元のストア	7

大規模ストアがさらに 30 台の冷蔵庫をオーダーすると、これを供給する十分な在庫がありません。コンプライアンス・サービスの手順を指定する作業オーダーが作成されます。コンプライアンス・サービスの実行数量は 100 です。したがって、合計 100 台の冷蔵庫が大きいストアの要求に対してカスタマイズされます。

大規模ストアが特売を行っていて、150 台の冷蔵庫をオーダーする場合、200 台の冷蔵庫の作業オーダーが作成されます (100 台ずつ 2 回実行)。これを行うのは、手持ちの冷蔵庫が 21 台のみで、実行数量 100 を追加しても冷蔵庫は 121 台にしかならないためです。したがって、実行数量を繰り返し使用し、オーダーの要求を満たす十分な量を作成します。

オーダー済み数量が実行数量と等しくない場合、スケジューリング中に在庫チェックが 1 回ではなく、2 回行われます。

地元のストアがさらに 7 台の冷蔵庫をオーダーした場合、作業オーダーは作成されず、追加の冷蔵庫は作成されません。これにより、地元のストアの冷蔵庫の在庫が完全になります。地元のストアが 7 台以上の冷蔵庫をオーダーした場合、追加で 10 台の冷蔵庫が作成されます。

---

## 第 11 章 供給コラボレーション

---

### 供給コラボレーションとは

供給コラボレーションは、複数のビジネス・ユニット、部門、サプライヤー、外部製造元、および輸送業者間での直接材料の供給計画、購入オーダー、補充、在庫、およびインバウンド・フルフィルメントの調整や共同実行を行います。

供給コラボレーションには、製品の購入オーダー処理の管理が含まれます。

---

### 購入オーダー

Sterling Selling and Fulfillment Foundation では、購入オーダーは、購入オーダー・ヘッダー・レベル、購入オーダー明細レベル、および購入オーダー・リリース・レベルという 3 つのレベルに分類できます。購入オーダー・ヘッダー・レベルには購入オーダーの購入オーダー明細がすべて含まれます。購入オーダー明細レベルは、個々の各購入オーダー明細ごとに分類されます。購入オーダー・リリース・レベルには、出荷ノードにリリース済みの明細がすべて含まれます。

1 つの購入オーダーは、複数の購入オーダー明細で構成されます。購入オーダー明細は、固有のアイテム、単位、および製品クラスで識別されます。

---

### 購入オーダー実行パイプライン

購入オーダーは、その作成から始まり、一連のトランザクションとステータスが続き、完了します。この一連のトランザクションと購入オーダーのステータスは、購入オーダー実行パイプラインと呼ばれます。2 つのデフォルトのパイプラインがあります。独自のノードを持つ組織用の標準購入オーダー実行パイプラインと購入オーダー処理にドロップ出荷ノードを使用する組織用のドロップ出荷購入オーダー実行パイプラインです。

購入オーダー実行パイプラインは、Sterling Selling and Fulfillment Foundation にパイプラインの購入オーダーのロケーションに応じてアクション (通知の送信、警告のロギングなど) を実行する手段を提供します。また、購入オーダーを作成から完了まで追跡し、必要に応じて手操作による介入を実行する手段も提供します。

ビジネスが稼働する購入オーダー実行パイプラインは、システム管理者によるビジネス環境の設定方法に固有のものです。ただし、各購入オーダー実行パイプラインは、一般に、購入オーダーを作成するトランザクションで始まり、購入オーダーが出荷されたことを示すトランザクションで終了します。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。購入オーダー・ステータスは、購入オーダーの状態を説明し、トランザクションからトランザクションに移動します。

## 購入オーダーのステータス

購入オーダー実行パイプラインでは、システム内での構成内容に応じて、次のステータスが使用される場合があります。

- 受諾済み - 交渉条件が両者によって受諾済みで、購入オーダーはリリース準備が整っています。
- 連鎖オーダー作成待機中 - 連鎖オーダーを作成して、該当するノードに送信する必要があります。
- バックオーダー済み - 購入オーダーは作成済みですが、十分な在庫がないためスケジュールできません。購入オーダーは、在庫が入手可能になるまで、バックオーダー済みのままです。
- ノードからバックオーダー済み - 購入オーダーは作成されていてノードにリリース済みですが、ノードにオーダーを満たす十分な在庫がありません。
- 交渉中 - 購入オーダーは交渉パイプラインで交渉中です。
- キャンセル - 購入オーダーはキャンセル済みです。
- 連鎖オーダー作成済み - 連鎖オーダーが購入オーダーから作成済みです。
- 作成済み - 購入オーダーは作成済みです。
- ドラフト・オーダー作成済み - 購入オーダーの作成コンソールでドラフト購入オーダーが作成済みです。確認済みになるまでは、このオーダーのあらゆる事項を変更できます。
- 出荷に含まれている - 購入オーダーは配達対象として出荷に追加されています。
- 受入終了 - 購入オーダーで必要な返品取り扱いが完了し、返品は終了しています。
- 受入済み - 購入オーダーはバイヤーによって受入済みです。
- コンポーネントとして受入済み - 購入オーダーのアイテムがキットのコンポーネントで構成されている場合、このステータスは、バイヤーがすべてのコンポーネントを受入済みであることを示します。
- リリース済み - フルフィルメント用に購入オーダーにスケジュールする十分な在庫があります。購入オーダーは、アプリケーション・コンソール、Sterling Selling and Fulfillment Foundation 倉庫管理システム、または他のサード・パーティー製倉庫管理システムにリリースされます。
- 予約済み - 購入オーダーは作成済みですが、スケジュールの準備ができていません。
- スケジュール済み - 該当する 1 つ以上のノードに購入オーダーを履行できる在庫があり、リリース用にスケジュールできます。
- ノードに送信 - 購入オーダーはオーダー・リリースの形式で送信されます。
- 出荷済み - 購入オーダーは出荷済みです。
- 出荷遅延 - 購入オーダーの出荷のすべてまたは一部が遅延しています。
- 不足 - 要求されたものよりも少ない数量が購入オーダーに含まれています。オーダーはクローズされます。
- 受入取り消し済み - 購入オーダーはバイヤーによって受け入れられていません。
- スケジュール解除 - 購入オーダーが「スケジュール済み」のステータスから削除され、スケジュールされたノードでこの購入オーダー用に予約されていた在庫が

すべてキャンセルされます。

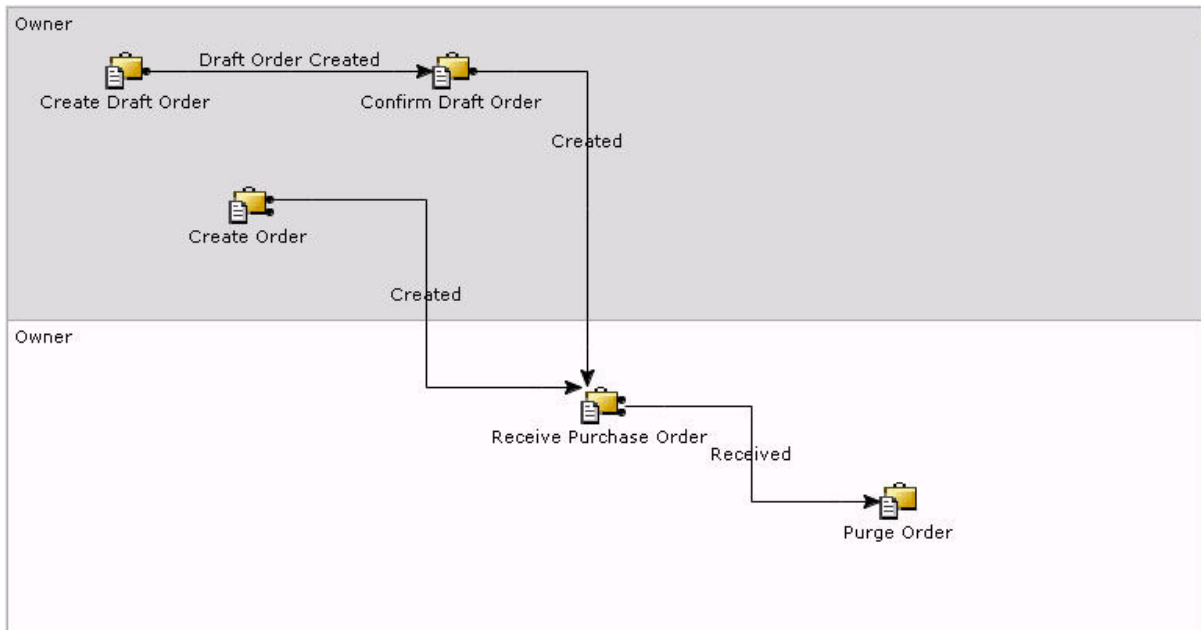


図 40. 購入オーダー実行パイプライン

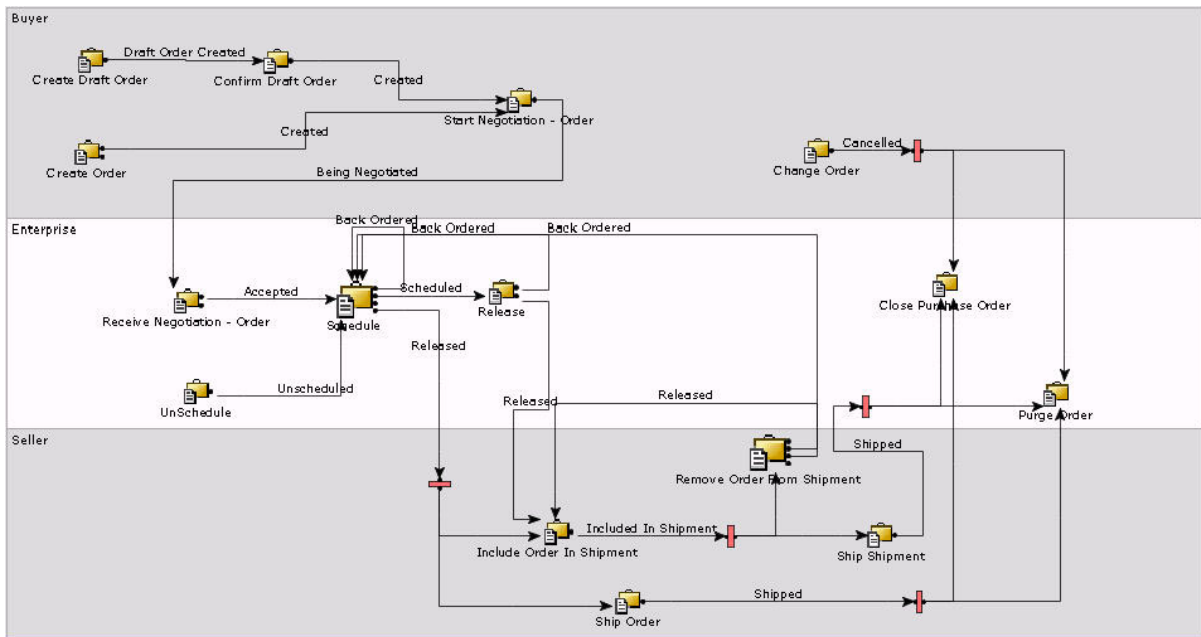


図 41. ドロップ出荷購入オーダー実行パイプライン

デフォルトのパイプラインとステータスは、ユーザーのビジネス慣習に従って構成できます。購入オーダー実行パイプラインおよびステータスの構成については、「*Sterling Selling and Fulfillment Foundation アプリケーション・プラットフォーム構成ガイド*」を参照してください。





---

## 第 12 章 支払システム

---

### 支払システム

Sterling Selling and Fulfillment Foundation は、オーダー管理の処理の際に重要な支払関連プロセスを実行するために使用でき、外部決済システムとの統合を実現します。

決済処理には、支払いの承認、決済、および請求が関係します。この章では、Sterling Selling and Fulfillment Foundation のオーダーのライフサイクルのさまざまな時点で発生する承認および決済に関連する、決済処理の機能、モジュール、コンポーネント、およびイベントについて説明します。

---

### 決済処理の概要

Sterling Selling and Fulfillment Foundation には、次の決済処理機能があります。

- 複数の支払方法
- 同期決済処理
- 非同期決済処理
- 売掛勘定による外部集金
- 顧客ロジックを処理するための外部プログラム、および API
- 料金の集約
- 記録済みの前払い
- 返される要請済みの金額
- 遅延再承認 (delayed reauthorization)
- 請求トランザクション要求 ID
- 承認取り消し
- 事前決済のサポート
- 請求
- 返品決済処理および返金フルフィルメント
- 交換決済処理
- ドラフト・オーダー (draft order) の決済処理
- 手操作による介入
- 複数の制御レベル
- 支払ルール

### 複数の支払方法

Sterling Selling and Fulfillment Foundation では、1 件のオーダーに関連付けることができる支払方法の数に制限がありません。さらに、単一の支払タイプ内 (クレジット・カード (credit card) など) についても、Sterling Selling and Fulfillment Foundation によって許可される個別のクレジット・カード (credit card) の数に制限

はありません。例えば、特定の 1 件のオーダーの支払いに割引クーポン、ギフト・チェック、および 2 枚のクレジット・カード (credit card) を使用することができます。

## 同期決済処理

Sterling Selling and Fulfillment Foundation トランザクションおよび外部プログラムは、外部決済処理システムと同期するように接続され、承認または請求の結果を待機するように構成できます。要求の成功または失敗に応じて、オーダーの支払ステータスが即時に更新されます。

## 非同期支払処理

Sterling Selling and Fulfillment Foundation トランザクションおよびユーザーEXIT を構成して、外部の支払処理システムと非同期で通信することができます。次に承認および請求の要請をバッチ・モードでディスパッチし、要請の結果に基づいて更新を一度に実行できます。Sterling Selling and Fulfillment Foundation におけるオーダーの支払ステータスは、要請が未処理か処理済みかを示しています。

## アカウントを介した外部集金受け入れ可能

Sterling Selling and Fulfillment Foundation では、アカウント受け入れ可能情報を扱う外部システムと統合することができます。これが使用可能である場合、すべてのインボイスの詳細は、外部アカウント受け入れ可能システムに対して公開されません。決済プロセスは、Sterling Selling and Fulfillment Foundation では扱われません。Sterling Selling and Fulfillment Foundation の外部で発生している集金はいずれも、レポートは返されません。

## カスタム・ロジックを処理するために提供される外部プログラムおよび API

外部プログラムは、支払タイプに応じて選択的に呼び出されます。外部プログラムは、必要性に合わせてカスタマイズできます。カスタマイズされた決済処理が可能になるように、API も提供されています。

## 料金の集約

料金の集約によって、外部の支払処理システムに対する同期呼び出しの回数を削減できます。セラーに対して定義されている各支払タイプに基づいて、料金の集約の時間フレーム・ウィンドウをセットアップします。後で実動モードで、特定のオーダーに対する同じ要求が指定された時間フレーム内で発生した場合、それらは一緒にグループ化されてから、外部呼び出しが行われます。

## 記録済みの前払い

オーダーが作成されると、支払い承認と集金を前払いで実行し、Sterling Selling and Fulfillment Foundation に送信することができます。その場合、Sterling Selling and Fulfillment Foundation で事前承認および事前集金されたこれらの金額を記録できます。これらが一部のみの支払いである場合は、オーダーの合計金額からその金額が差し引かれ、オーダーの残りの金額については、一般的な決済処理の動作が続行します。

## 返される要請済みの金額

getOrderDetails API は、ドラフト・オーダー (draft order) を含むオーダーで承認される要請済みの金額を返します。要請済みの金額は、オーダーの合計から以下を差し引いたものと等しくなります。

- アクティブな支払方法の MaxChargeLimits
- 現在保留の支払方法で作成された料金
- 支払方法に結合されていないクレジットの合計

UnlimitedCharges = Y というマークが付いた支払方法は、制限が 0 としてカウントされます。オーダーに追加料金または承認がある場合、要請済みの金額は負になることがあります。例えば、オーダーの合計が \$100 で、次の支払いの場合、要請済みの金額は \$60 ( $\$100 - 0 - \$30 - \$10$ ) になります。

- 最大料金限度が \$30 のギフト・カード
- 最大料金限度が \$20 (\$10 請求済み) の保留クレジット・カード (credit card)
- 上限なし料金のクレジット・カード (credit card)

## 遅延再承認 (delayed reauthorization)

遅延再承認 (delayed reauthorization) 機能は、バックオーダーまたは事前オーダーされた在庫に対してオーダーをスケジューリングする場合に発生する可能性のある支払い再承認の数を制御します。出荷日が近づくのを待つ間に、支払い承認が期限切れになり、顧客のクレジット明細がロックされ、再承認を複数回行われる可能性があります。この場合、支払いゲートウェイとインターフェースを取る場合に費用もかかります。2 つの遅延再承認 (delayed reauthorization) 構成オプションは、再承認の回数を、オーダー・サイクルの構成可能ポイントに制限します。

遅延承認は、ドラフト・オーダー (draft order) についてはサポートされません。

## 請求トランザクション要求 ID

請求トランザクション要求 ID 機能は、オーダーに対する支払承認の発生時期および承認金額を制御するための仕組みを提供します。この機能によって、承認がオーダー・レベルで自動的に処理されるのではなく、オーダーに関する支払承認をよりきめ細かく制御することができます。承認は ID と結びつけられており、ID は、オーダーにおける 1 つのエンティティまたはエンティティ・グループを表します。例えば、支払承認をオーダー・レベルだけでなく、いずれかのエンティティに要求 ID を作成することで、オーダー・リリース・レベルまたはオーダー明細レベルで処理できます。

請求トランザクション要求 ID 機能は、「要決済」の支払いオプションをサポートしません。

## 承認取り消し

クレジット・カード (credit card) 会社の中には、使用されていないクレジット・カード (credit card) 承認を明示的に取り消さない業者に追加手数料を請求する会社もあります。「承認取り消し」機能を使用することで、業者は、使用されていない承

認を有効期限が切れる前に取り消すための承認戦略を実装できます。この機能は、ユーザーEXITを使用することによって、異なる承認金額と決済金額を処理するための方法も提供します。

## 事前決済のサポート

事前決済を使用すると、オーダー処理のどの段階でも、オーダーの一部または全部を請求できます。これは、アイテムが顧客に既に販売され、取り置きが行われている、事前オーダーまたはストア・ピックアップの状況で一般的に使用されます。このため、実際の製品出荷の前に集金プロセスを開始する必要があります。

## 名前付き外部プログラム

invokeUE と呼ばれる新規 API を実装することによって、外部プログラムを直接呼び出すことができます。これは、外部プログラムとして実装されている不正チェックなどの機能呼び出す必要がある場合に役立ちます。invokeUE API は、名前付き外部プログラムの呼び出しを必要とするすべてのアプリケーションへのエントリー・ポイントです。invokeUE API は、入力 XML での指定に従って内部的に外部プログラムを呼び出し、外部プログラムの出力を呼び出し元に戻します。invokeUE API について詳しくは、*Sterling Selling and Fulfillment Foundation Javadocs*を参照してください。

## ドラフト・オーダー (draft order) の決済処理

ハブ・レベルで「ドラフトについて決済処理を有効化」フラグを設定することにより、決済処理をドラフト・オーダー (draft order) に有効にすることができます。これにより、ドラフト・オーダー (draft order) を確認する前にドラフト・オーダー (draft order) に対する支払いの処理が可能になります。さらに、ドラフト・オーダー (draft order) の作成時に請求トランザクション・テーブルおよび支払ステータスにデータが設定されます。このフラグは、デフォルトでオンになっています。ドラフト・オーダー決済処理には、以下のフィーチャーが含まれています。

- processOrderPayment API は、新規支払方法を受け入れ、持続します
- 失敗した承認に関するエラー・メッセージを表示します
- 失敗した支払方法を変更します
- ドラフト・オーダー (draft order) に対する料金を無視します
- 料金シーケンスに基づいて決済処理を実行します
- 請求トランザクションを集約します

### processOrderPayment API による新規支払方法の受入と持続

processOrderPayment API は、オーダーに対する新規支払方法を受け入れ維持するために拡張されます。これにより、オーダーについて無請求の支払方法および無許可の支払方法を修正したり、削除したりすることができます。

### 失敗した承認に関するエラー・メッセージを表示

エラー・メッセージは、processOrderPayments API 出力 XML に返され、YFS\_PMNT\_TRANS\_ERROR テーブルに格納されます。決済処理が完了した後にエラー・メッセージを取り出すためには、getOrderDetails API を呼び出します。

## 失敗した支払方法を変更

ドラフト・オーダー (draft order) に対する支払方法が承認に失敗した場合は、その支払方法を削除したり、修正したりすることができます。この場合、オーダーの支払ステータスは「失敗」で、請求トランザクション・ステータスはエラーでなければなりません。支払方法を変更するには、支払いキーと支払タイプを除き、支払方法の属性のいずれかを変更します。

## ドラフト・オーダー (draft order) に対する料金の無視

「ドラフトに対する料金を無視」フラグが有効であれば、このフラグを使用して、ドラフト・オーダー (draft order) に対する要請金額を計算するときに請求要請を無視することができます。例えば、合計 \$100 のドラフト・オーダー (draft order) に \$20 のクレジット・カード (credit card) 承認と、\$20 の小切手 (請求要請) がある場合、getOrderDetails API は \$80 の要請金額を返します。「ドラフトに対する料金を無視」フラグは、デフォルトではオフであり、ハブ・レベルで構成されます。

「ドラフトに対する料金を無視」フラグを構成する前に、「ドラフト・オーダー処理を有効化」フラグをオンに設定する必要があります。

## 料金シーケンスに基づいて決済処理を実行

processOrderPayment API で、各支払方法の料金シーケンスにより、ドラフト・オーダー (draft order) に対して処理される認証および料金の順序が決定されます。デフォルトでは、支払方法は同等の料金シーケンスを持っているため、ランダムな順序で処理されます。

## 請求トランザクションを集約

オーダーが変更されるたびに、請求トランザクション・レコードが作成されます。このため、1 つのオーダーについて多くの請求トランザクション・レコードが存在することがあります。confirmDraftOrder API は、1 つのオーダーについて、すべての請求トランザクション・レコードを単一のレコードに集約します。

## インボイス作成

インボイス作成プロセスにより、通常、オーダー・ライフサイクルで決済プロセスが始まります。インボイス作成は、時間トリガー・トランザクションまたは API を介して行うことができます。

出荷のインボイス作成 は、オーダーが出荷された後で発生します。これにより、集金プロセスが始まります。

オーダー・インボイスの作成 は、出荷の直後ではなく、オーダーのライフサイクルの任意の段階で決済プロセスを開始できるようにします。これは、出荷がない状態や、インボイスが複数の出荷にまたがる必要がある場合に使用できます。

提供サービスおよび配送サービスは、オーダーされた製品明細数量に基づいて請求されます。詳しくは、240 ページの『提供サービスおよび配送サービスのインボイスの作成』を参照してください。

返品インボイス は、返品プロセス中に顧客に対して返金を実行できるようにします。返金は、返品商品の受入の前または後に発生するよう構成できます。返品が贈

答品受取人によって作成された場合は、Sterling Selling and Fulfillment Foundation では、販売オーダーのバイヤーに対してではなく、贈答品受取人に対して返金を実行します。

インボイスの作成中にオーダーの税金および変更内容に対してなんらかの変更がある場合は、オーダーは、オーダー全体が請求された時点でのみ更新されます。また、インボイスの作成中に明細に関する税金および変更内容に対して変更がある場合は、これらのオーダー明細は、該当のオーダー明細全体が請求された時点でのみ更新されます。

インボイスは、作成直後（一般に、外部集金用）または Sterling Selling and Fulfillment Foundation アプリケーションの範囲内での決済直後に公開できます。

## 仮インボイスの作成

多くのビジネス・シナリオにおいて、各オーダーに複数の出荷がある場合は、料金と税は、出荷ごとに分割されます。実装済みのビジネス・プロセスに応じて、これらの金額は、いくつかの方法でさまざまな出荷間で分割されます。オーダーの後続の出荷について適切に請求および課税するには、その対象として持続するように、以前の料金と税がデータベース内に存在している必要があります。

仮インボイス (Pro Forma invoice)、すなわち、出荷作成時に生成されるインボイスは、追加の出荷について料金と税が計算されるときに以前の料金と税が考慮されるように、料金と税をデータベース内に存続させるドラフト・インボイスとして機能します。

例えば、明細が 2 つあるオーダーが \$70 で作成されます。エンタープライズは、出荷に対して以下のようなインクリメンタル料金を使用します。

- \$0 から \$49.99 までのオーダーには \$4.99 の出荷料が追加されます。
- \$50 から \$99.99 までのオーダーには、\$6.99 の出荷料が追加されます。

オーダーの明細 1 は \$40 で、オーダーの明細 2 は \$30 です。仮インボイスの作成を使用しない場合は、インボイス作成時に以下のような料金が発生します。

表 32. 例: 仮インボイスの作成を使用しないオーダー

明細	金額	料金
明細 1	\$40	\$4.99
明細 2	\$30	\$6.99

仮インボイスの作成を使用した場合は、明細 2 に対する出荷料金を計算するとき、明細 1 についての既存の料金が考慮されます。

表 33. 例: 仮インボイスの作成を使用したオーダー

明細	金額	料金
明細 1	\$40	\$4.99
明細 2	\$30	\$6.99 - \$4.99 = \$2.00

## 事前集金済み支払いのあるオーダーの一部キャンセル

オーダーがまだ出荷されていないが、そのオーダーの支払いがすでに事前集金済みである場合、オーダーの全部または一部のキャンセルに一定の金額の返金が必要であることがあります。その場合、Sterling Selling and Fulfillment Foundation は、返金される金額（該当する場合）で外部支払いシステムによりピックアップされるように構成できる通知を送信します。

例えば、\$140 のオーダーに \$100 相当の事前集金済み支払いがあり、\$50 がキャンセルされた場合、\$10 という金額を事前集金済み支払タイプに返金する必要があります。その場合、\$10 が返金されるように Sterling Selling and Fulfillment Foundation によって通知が発生します。

さらに、支払ルール・レベルにおいて Applications Manager で返品時に繰延クレジット項目が必要ルールを有効にすることができます。そのルールが有効になっている場合で、しかもキャンセルされた金額が返金不要の場合は、Sterling Selling and Fulfillment Foundation は、その金額で外部支払いシステムによりピックアップされるように構成できる通知を送信します。

上記の例で、返品時に繰延クレジット項目が必要ルールが有効になっている場合、キャンセルされたが返金されなかった \$40 について、Sterling Selling and Fulfillment Foundation により通知がもう 1 つ発生します。

## 価格変更のインボイスの転記

Sterling Selling and Fulfillment Foundation は、オーダーがすでに請求されている場合に、オーダーについてのあらゆる料金または単価の変更に対応して、調整インボイスの作成をトリガーします。これは、指定されたエンタープライズの伝票種別レベルで請求済み数量への変更の適用ルールを有効にすることによって可能になります。この調整インボイスは、支払エージェントによってピックアップされ、適宜処理されます。

例えば、いくつかのストアがそれぞれの顧客に最低価格保証を提案します。顧客が 1 足の靴を \$50 で購入し、同じ靴が別のストアで \$45 であることを知った場合、\$5 の返金を要求するはずですが、請求済み数量への変更の適用ルールが有効になっており、販売オーダーのインボイスの転記作成で靴の単価が \$45 に変更された場合、\$5 の調整インボイスが作成され、その金額の返金が顧客に対して実行されます。

## 税変更のインボイスの転記

Sterling Selling and Fulfillment Foundation は、オーダーがすでに請求されている場合に、オーダーについての税率のあらゆる変更に対応して、調整インボイスの作成をトリガーします。これは、指定されたエンタープライズの伝票種別レベルで請求済み数量への変更の適用ルールをオンにすることによって可能になります。この調整インボイスは、支払エージェントによってピックアップされ、適宜処理されます。

例えば、非課税ステータスを持つ顧客が購入したが、該当オーダーが出荷され、請求された後までストアに自分のステータスを通知しなかった場合、調整インボイスが作成され、その顧客に返金が行われます。

税率を渡すことによりオーダーまたはオーダー明細で税金が適用されると、そのオーダーに対する税の金額、オーダー明細、および税金記録はインボイスの作成後に更新されます。オーダー、オーダー明細、およびオーダー API の出力の税金記録に正しい税金額を反映させるには、対応する税率の正しい税金額を入力に入れて渡すか、または税金の再計算外部プログラムを実装する必要があります。

## 返品支払処理および返金フルフィルメント

返品プロセス中に、消費者から集金した料金を消費者に返金する必要があります。返金金額は元の支払方法を使用して入金されます。

### 返金シーケンス

販売オーダーで複数の支払方法が使用されている場合、料金は、Applications Manager で定義されている返金シーケンスに従って返金されます。2 つ以上の支払方法の優先順位が同じ場合、返金は、それらの方法に照らして販売オーダーで使用された支払シーケンスの逆の順序で実行されます。このデフォルトのシーケンスは、confirmRefundPayments ユーザーEXIT を実装することによってオーバーライドできます。

### 前払い

部分的に前払いと通常の支払いが含まれる販売オーダーから返金が派生している場合、Sterling Selling and Fulfillment Foundation は最初に自己の支払方法に照らして返金を行います。その時点までに全額が返金されていない場合に、前払い方法が使用されます。

### デフォルトの返金支払タイプ

デフォルトの返金支払タイプは、返金発行対象の有効な支払方法がオーダーになれば、デフォルトの返品支払タイプの新しい支払方法に照らして返金を発行するように定義できます。デフォルトの返金支払タイプは、confirmRefundPayments ユーザーEXIT を実装することによってオーバーライドできます。

### 同じアカウントへの返金

「同じアカウントに返金 (Refund to Same Account)」フラグは、支払タイプ・レベルで定義し、支払いを請求先と同じアカウントに返金するか、それとも新しいアカウントを作成するかを示します。例えば、顧客がストアード・バリュー・カードを使用してオーダーを発行した場合にこのフラグが使用可能になっていると、返金は同じストアード・バリュー・カードに対して行われます。

### 新しい支払タイプに対する返金

「同じアカウントに返金 (Refund to Same Account)」フラグが使用不可の場合、Sterling Selling and Fulfillment Foundation は、Applications Manager で定義された新しい支払タイプに対して返金を行おうとします。新しい支払タイプとして同じ支払タイプを選択すると、Sterling Selling and Fulfillment Foundation は同じ支払タイプに返金しますが、アカウントは新規のものになります。例えば、「同じアカウントに返金 (Refund to Same Account)」フラグの設定が「N」で新しい支払タイプの設定がストアード・バリュー・カードの場合に、ストアード・バリュー・カードに返金を行うと、返金は新しいストアード・バリュー・カードに対して発行されます。



YFS\_CHARGE\_TRANSACTION テーブルに新しい請求要請を作成したくない場合は、新しい支払タイプへの返金を作成されます。この場合、不完全な料金トランザクションが発生するため、支払タイプの料金詳細を完了するには、ユーザーによる介入が必要になります。詳細が入力されると、要求収集エージェントが返金の必要な料金をピックアップして、「支払済み」ステータスに移します。

## 代替返金支払タイプ (制約あり)

特定の支払タイプに対する返金に制約を課す必要がある場合があります。例えば、エンタープライズでストアード・バリュー・カードの生産コストに \$4 かかっており、返金金額が \$5 より小さい場合、エンタープライズは小切手を発行できます。このために、代替返金支払タイプが含まれるように支払タイプを構成できます。これにより、返金金額が制約を満たした場合に返金が発行されます。

## 返金フルフィルメント・オーダー

返金フルフィルメント・オーダーは、「以下を使用した返金フルフィルメント・オーダーの作成」フラグが使用可能な場合に支払集金トランザクションによって作成されます。このフラグが使用可能な場合、新しい支払方法ではなく返金フルフィルメント・オーダーが作成されます。例えば、顧客がギフト・カードを使用してオーダーを発行し、そのオーダーを返品したとします。この場合、通常は、顧客に新しいギフト・カードが出荷されます。ギフト・カードを出荷するために、新しいフルフィルメント・オーダーが自動的に作成されます。

決済が必要なくとも、承認に代わる請求支払タイプに返金が行われます。

## 交換決済処理

交換が作成された時点で、返品オーダーの合計オーダー金額が交換オーダー (exchange order) の PENDING\_TRANSFER\_IN フィールドに追加されます。この値は、常に、交換オーダー (exchange order) と返品オーダーについて同期され続けます。たとえば、交換が合計金額が \$40 の返品オーダーから最初に派生し、しかも、後になって明細が返品に追加されて、その金額が \$60 まで増やされた場合、交換の PENDING\_TRANSFER\_IN も増やされます。

拡張交換オーダーおよび標準交換オーダーの場合、交換オーダー (exchange order) の合計金額が PENDING\_TRANSFER\_IN の金額を超えたら、追加の支払い情報を入力する必要があります。

交換オーダー (exchange order) に対して TRANSFER\_IN 交換レコードが作成され、金額の転送元となったオーダーに対して TRANSFER\_OUT 交換レコードが作成されます。

前納交換の場合、交換の金額全体に対して、支払い情報を前もって収集する必要があります。

## 手操作による介入

ユーザーは必要に応じて、さまざまな UI コントロール、API、トランザクション、および外部プログラムを使用して、標準の動作をオーバーライドできます。これらの手動作業へのアクセスは、権限の構成を使用して制御できます。

手操作による介入を使用して顧客サービス担当者 (customer service representatives) は、返金のためのクレジット伝票や料金のデビット伝票を作成できます。手動による与信および請求要請も実行できます。オーダーに支払保留ステータスを設定し、その後、解除して処理を続行できます。必要に応じて新規の支払方法を追加できます。既存の支払方法の保留および再アクティブ化を実行できます。支払方法が保留されると、その支払いに対して作成されている他のすべての未処理の与信および請求要請がクローズされ、その支払方法に対して新規の与信および請求要請を作成できなくなります。

## 複数の制御レベル

支払い処理は、伝票タイプ・レベルで有効と無効を切り替えることができます。例えば、計画済みオーダー伝票タイプで支払い処理が無効化されている場合、その伝票タイプに属するすべてのオーダーで支払い関連アクティビティは行われません。また、支払い処理は、セラー・レベルで制御することもできます。特定のセラーの支払い処理を無効化して、そのオーダーで支払い処理が行われないようにすることができます。伝票タイプの権限設定は、セラー・レベルの制御設定をオーバーライドします。

## 支払ルール

セラー組織には、1 つ以上の支払ルールを構成できます。これらのルールのいずれかを、セラー組織のデフォルト支払ルールとして設定する必要があります。オーダー作成時にオーダーに適用可能な支払ルールを指定するか、セラーのデフォルトの支払ルールをオーダーに使用します。支払ルールによって、決済処理が売掛勘定と Sterling Selling and Fulfillment Foundation のどちらを使用して実行されるか、Sterling Selling and Fulfillment Foundation の承認または決済、あるいはその両方で実行される決済処理のタイプなどが指定されます。

---

## 支払タイプのグループ

支払タイプのグループとは、同様に動作する支払タイプの追跡手段のことです。使用可能な支払タイプのグループは次のとおりです。

- クレジット・カード (Credit Card)
- 顧客アカウント (Customer Account)
- ストアード・バリュー・カード
- その他 (Other)

---

## 支払タイプ

支払タイプとは、セラー組織で使用される支払関連情報を追跡する手段のことです。提供される標準的な支払タイプは次のとおりです。

- CREDIT\_CARD
- CUSTOMER\_ACCOUNT
- CHECK
- OTHER

セラー組織は、任意の数のカスタム支払タイプを定義できます。オーダーで使用される支払方法はすべて、そのセラーの有効支払タイプとして構成済みの支払タイプであることが要求されます。

セラーの構成で、支払タイプのそれぞれに料金シーケンスと返金シーケンスを指定できます。0 の料金シーケンス番号 (または返金シーケンス番号) の優先順位が最も高く、次が 1 で、以下同様に続きます。複数の支払タイプに同じシーケンス番号を設定できます。シーケンスが指定されていない場合は、すべての支払タイプの優先順位が同等であるとされます。料金シーケンスと返金シーケンスは相互に独立しています。

## 支払方法

支払方法は、支払タイプに固有の支払い情報を格納し、オーダーとともに送られます。また、事前承認または事前請求情報に関する決済詳細が含まれる場合もあります。Sterling Selling and Fulfillment Foundation は、各オーダーで任意の数の支払方法の受け渡しをサポートします。

オーダーの支払方法内で使用する場合に、各支払タイプに必須のフィールドは次のとおりです。

表 34. 支払タイプの必須フィールド

支払タイプのグループ	必須フィールド
クレジット・カード (Credit Card)	CreditCardNo
顧客アカウント (Customer Account)	CustomerAccountNo
ストアード・バリュー・カード	SVCNo
その他 (Other)	PaymentReference1

オーダーの 1 つ以上の支払方法に料金シーケンスを割り当てることもできます。支払方法に対して、またはセラーの構成レベルで料金シーケンスが指定されていない場合は、そのオーダーのすべての支払方法で請求に関する優先順位が同等になります。支払方法の料金シーケンスは、セラー構成レベルで定義されたシーケンスに従属します。

支払方法に割り当てられる料金を上限なしにすることができます。上限なし料金 = 「N」の場合、最大料金限度を設定できます。支払方法で上限なし料金 = 「N」の場合、支払方法に対して行われる承認と料金のすべてに、最大料金限度によって設定される上限が適用されます。上限なし料金 = 「Y」が設定された支払方法には、オーダーの承認/料金の残余金額が割り当てられ、料金シーケンスでその下位にある支払方法はすべて無視されます。

オーダー内の支払方法は、変更、保留、削除が可能です。また、既存オーダーへの新規支払方法の追加も可能です。詳しくは、*Sterling Selling and Fulfillment Foundation Javadocs*で、`changeOrder()` API を参照してください。

## 機密情報のトークン化

デフォルトでは、クレジット・カード (credit card) 番号およびストアード・バリュー・カード番号は、暗号化される代わりにトークン化されます。トークン化は、クレジット・カード (credit card) 番号やストアード・バリュー・カード番号などの機密情報を、セキュリティー・ストアード機密情報にトークンを関連付けるポールド・システムに格納するプロセスです。

トークン化のプロセスは、IBM Sterling Sensitive Data Capture Server アプリケーションによって使用可能にされます。Sterling Sensitive Data Capture Server は、クレジット・カード (credit card) 番号およびストアード・バリュー・カード番号を解釈、検証、およびトークン化し、IBM Sterling Selling and Fulfillment Suite を使用可能にして、Payment Application Data Security Standard (PA-DSS) 準拠を達成します。

Sterling Sensitive Data Capture Server アプリケーションおよびクレジット・カード (credit card) 番号およびストアード・バリュー・カード番号のトークン化の追加情報については、「*Sterling Selling and Fulfillment Foundation: Sterling Sensitive Data Capture Server リリース 1.1 構成ガイド*」を参照してください。

PCI-DSS 準拠を達成するための Sterling Sensitive Data Capture Server アプリケーションの構成方法について詳しくは、「*PA-DSS 実装ガイド*」を参照してください。

## 請求シーケンス

請求シーケンスによって、Sterling Selling and Fulfillment Foundation が承認要請または請求要請を作成する順序が決まります。ロジックは、次のようになります。

オーダーの各支払方法には、実際の料金シーケンスが割り当てられており、これに基づいて承認要請および請求要請が作成されます。要求は、セラー構成に定義されている支払タイプの構成済み料金シーケンスの昇順で、次に特定のオーダーに定義されている支払方法のトランザクション料金シーケンスの昇順で、その支払方法に照らして作成されます。

例えば、このシナリオでは、セラーの料金シーケンスは以下のように指定されています。

- 0 (CUSTOMER\_ACCOUNT)
- 1 (CREDIT\_CARD)
- 2 (OTHER)

オーダーの支払方法は、以下の料金シーケンスを使用するように指定されています。

- 1 (CreditCard1)
- 2 (CustomerAccount1)
- 3 (Other1)
- 4 (CreditCard2)

この結果、これらの支払方法での請求は以下のシーケンスで行われます。

1. CustomerAccount1

2. CreditCard1
3. CreditCard2
4. Other1

また、請求に関連した以下の動作に注意してください。

- 作成された要請は、支払方法で指定された「最大料金限度」を超えることはありません。
- 特定の支払方法が有効でない場合 (情報が不完全など)、システムはその支払方法をスキップして、オーダー内の優先順位がより低い支払方法に照らして請求を作成します。
- 中断されている支払方法に照らして、承認要請および請求要請が作成されることはありません。
- オーダー上の支払方法が、要求されている資金を充足するには不十分な場合、オーダーは `AWAIT_PAY_INFO` ステータスにロールバックされます。プロセスのこの時点で、`PAYMENT_COLLECTION.INCOMPLETE_PAYMENT_INFORMATION` イベントが発生する場合があります。

## 返金シーケンス

返金シーケンスは、Sterling Selling and Fulfillment Foundation が負の料金または返金要求を作成する順序を決定します。これが従うロジックについて以下に説明します。

1. 「実際の返金シーケンス」の順序で決まっている支払方法に照らして、返金要求が作成されます。要求は支払方法に照らして作成されます。この際、まず、セラーの構成で定義されている支払タイプの構成済み返金シーケンスの昇順で作成され、次に特定のオーダーで定義されている支払方法のトランザクション料金シーケンスの降順で作成されます。
2. 返金要求は、`VALID_FOR_RETURN` というマークが付いている支払タイプに対してのみ作成されます。支払方法が保留の場合、返金対象とみなされません。返金金額を決定する際のステップは次のとおりです。
  - a. オーダーの支払方法の最初の解析では、支払方法に対する返金は、支払方法に対する合計料金を超えません。
  - b. 最初の解析が終了した後も、関連する資金が請求されていても、有効な支払方法が見つからず、発行する必要がある返金が依然として存在している可能性があります。この追加の返金金額には、次のロジックが適用されます。
    - 「`PAYMENT->EXCEED_CHARGE_AMOUNT_FOR_REFUND`」ルールの設定が `Y` の場合 (伝票種別 (document type) またはルール・セット・レベル定義)、オーダーで優先順位が最初の (`Actual_Refund_Sequence` が最低の) 支払方法が残りのクレジットすべてを発行するよう選択されます。
  - c. 返品に有効な支払方法がオーダーにない場合、上のオプションも失敗することがあります。この場合は、「返品のデフォルト」というマークが付いた支払方法がオーダー用に作成され、この支払タイプに対して返金が発行されます。この支払タイプを完了するために詳細情報が必要な場合は、`INCOMPLETE_PAYMENT_INFORMATION` イベントが発生します。

上のどちらも失敗した場合 (Default\_For\_Return 支払タイプがない場合)、INCOMPLETE\_PAYMENT\_INFORMATION イベントが発生します。返金フルフィルメントについて詳しくは、256 ページの『返品支払処理および返金フルフィルメント』を参照してください。

---

## 支払ステータス

オーダーのそれぞれに、AUTHORIZED、INVOICED、PAID などの支払ステータスが設定されます。これらのステータスは、オーダーに関連付けられます。支払ステータスは、次の値のいずれかになります。

- AWAIT\_PAY\_INFO - オーダー作成時、または承認や請求の際に、支払情報が使用可能ではありません。また、指定された支払方法が不十分でオーダー金額または返金金額に対応できない場合も、オーダーがこのステータスになる場合があります。
- AWAIT\_AUTH - オーダー金額の一部について、承認が保留されています。
- REQUESTED\_AUTH - 承認要請が送信されていますが、支払システムからの応答を受信していません。このステータスは、非同期の環境のみで発生します。
- REQUESTED\_CHARGE - 請求要請が送信されていますが、支払システムからの応答を受信していません。このステータスは、非同期の環境のみで発生します。
- AUTHORIZED - オーダー金額が承認された金額以下です。遅延再承認 (delayed reauthorization) が構成されている場合、AUTHORIZED のステータスは、オーダーがパイプラインを使用して移動可能であるが、完全な承認が示されていないことを示します。詳しくは、267 ページの『遅延再承認 (delayed reauthorization)』を参照してください。
- INVOICED - オーダーの全体が請求され、未完了のオーダー金額はありません。
- PAID - オーダーの支払いが集金されています。
- FAILED\_AUTH - 承認要請に失敗しています。
- FAILED\_CHARGE - 請求要請に失敗しています。
- NOT\_APPLICABLE - オーダーの伝票種別 (document type) で決済処理が要求されていないか、セラー組織が決済処理を要求していません。

---

## 一般支払いプロセス

Sterling Selling and Fulfillment Foundation での支払いプロセスは、2 つのサブプロセス、つまり承認と決済で構成されます。

### 承認処理

支払承認は、支払方法で支払われた金額を確認する処理です。クレジット・カード (credit card) の場合、承認には、支払システムへの接続、およびクレジット・カード (credit card) に請求された必要な資金額のブロックが特に含まれます。支払タイプによっては、この承認ステップが必要な場合も、必要でない場合もあります。これは、セラーの支払ルールに Sterling Selling and Fulfillment Foundation で構成できます。オーダーに支払処理が必要な場合、Sterling Selling and Fulfillment Foundation は、支払処理が承認されるまで、スケジューリングその他の処理に対するオーダーをピックアップしないことに注意してください。

支払集金の時間トリガー・トランザクションは、オーダーを分析して、承認要請を作成します。支払実行の時間トリガー・トランザクションは、作成された承認要請をモニターし、承認を実行するためのユーザーEXIT を提供します。ユーザーEXIT は、次のいずれかの方法で承認要請を処理できます。

- 同期処理を実行して、売掛データベースにインターフェースして即時に承認を行い、承認された金額を返します。
- 支払システムとのインターフェースが操作不能の場合、後で再試行するように要求を出します。
- 非同期処理を要求します (つまり、コンソールがこのオーダーのために支払システムに接続することはありません)。

外部支払システムからの応答に応じて、Sterling Selling and Fulfillment Foundation 内で適切に応答を処理するために、異なるイベントを発生させることができます。使用可能なイベントについて詳しくは、「Sterling Selling and Fulfillment Foundation Javadocs」の支払処理 API を参照してください。

承認を受け取った場合、またはオーダーがオーダー金額全体についてフロントエンドで事前承認済みの場合、支払集金トランザクションは、支払ステータスを AUTHORIZED に変更します。

### 有効期限 (expiration date)

Sterling Warehouse Management System では、有効期限 (expiration date) 別に時間依存アイテムとして指定されたアイテムの在庫を追跡し、期限切れのアイテムはどのオーダーにも割り振られないようにします。有効期限属性の例について、以下の表で説明します。

表 35. 有効期限属性

ロケーション	SKU	有効期限 (expiration date)	数量 (Quantity)
13C01	SKU1	2004 年 12 月 20 日	100
	SKU1	2005 年 12 月 25 日	30

業界の例には、以下のものがあります。

- 缶詰食品
- 加工食品 (牛乳、肉)
- 薬剤

### 請求トランザクション要求 ID

請求トランザクション要求 ID の構成オプションには、「支払ルール」画面からアクセス可能です。ここで顧客は、オーダーに対してよりきめ細かな承認の制御を行うことができます。支払承認をオーダー・レベルでのみ処理する代わりに、この構成オプションでは、事前に定義されたオーダーのグループ化に基づいた支払承認が可能です。オーダー内の各グループ化には、それぞれ請求トランザクション要求 ID が関連付けられています。

注: 請求トランザクション要求 ID 機能は、「要決済」の支払いオプションをサポートしません。

YFS\_CHARGE\_TRAN\_REQUEST および YFS\_CHARGE\_TRAN\_RQ\_MAP の 2 つのハングオフ・テーブルが、オーダーに関する承認のグループ化要求を管理します。YFS\_CHARGE\_TRAN\_REQUEST には、承認および ID に対する要求が含まれ、YFS\_CHARGE\_TRAN\_RQ\_MAP は、その要求を請求トランザクションにマップします。

**承認要請のシーケンス:** 各承認要請には MaxRequestAmount 属性が含まれており、これによって、該当する要求に必要な合計承認数を定義します。要求が完全に承認されると、次の要求の承認が開始します。要求のシーケンスは指定可能です。指定しない場合、要求は要求 ID 順に実行されます。

**支払ステータス:** 承認要請のそれぞれに、その要請に関連付けられている支払ステータスが設定されます。支払ステータスには、AWAIT\_AUTH、REQUEST\_AUTH、FAILED\_AUTH、または AUTHORIZED の値を設定できます。デフォルトの支払ステータスは、AWAIT\_AUTH です。要請 ID の支払ステータスが変更されると、PAYMENT\_COLLECTION.REQUEST\_PAYMENT\_STATUS イベントが起動します。

要請の支払ステータスは、オーダーの支払ステータスと同じ方法で計算されます。

## 承認の取り消し

クレジット・カード (credit card) 会社は、通常請求するトランザクション手数料の他に、使用されていないクレジット・カード (credit card) 承認を明示的に取り消さない業者に追加手数料を請求する場合があります。「承認取り消し」機能を使用することで、業者は、使用されていない承認の有効期限が切れる前に取り消し要求を生成する承認戦略を実装できます。

クレジット・カード (credit card) 会社は、承認金額と決済金額が一致しない場合にも手数料を徴収することがあります。Sterling Selling and Fulfillment Foundation では、YFSCollectionCreditCardUE ユーザーEXIT の実装によって、有効期限が切れる承認を取り消す方法、および異なる承認金額と決済金額を処理する方法が提供されています。承認取り消し要求は、PAYMENT\_EXECUTION エージェントによって処理されます。

承認取り消し機能は、顧客アカウント (customer account) を除き、承認が必要なあらゆる支払方法をサポートしています。

**承認取り消しウィンドウ:** クレジット・カード (credit card) の承認処理時に、業者は、支払システムに接続して、決済金額でクレジット・カード (credit card) をブロックして、オーダーに対する承認を取得しようとします。通常、この承認の期限が切れると、再承認の取得が自動的に試みられます。ただし、承認の取り消しが有効になっている場合は、再承認の取得だけでなく、同じ承認 ID で、負の金額のみで承認要請が作成されます。例えば、\$100 の承認の期限が切れると、\$-100 に対する別の承認が開始されます。

期限が到来する時期、および取り消し期間が終了する時期は、クレジット・カード (credit card) 会社および銀行によって異なりますが、ユーザーのビジネスの測定基準に基づいて、承認取り消しウィンドウを定義することができます。これらの測定基



準には、支払システムに対するスループットの量、オーダー変更が許可される頻度、将来の在庫 (future inventory) に対するオーダーの割合、支払集金エージェントの実行頻度などがあります。

図 42 では、通常の承認ライフサイクル、およびそのライフサイクルでの取り消しウィンドウの位置について説明しています。

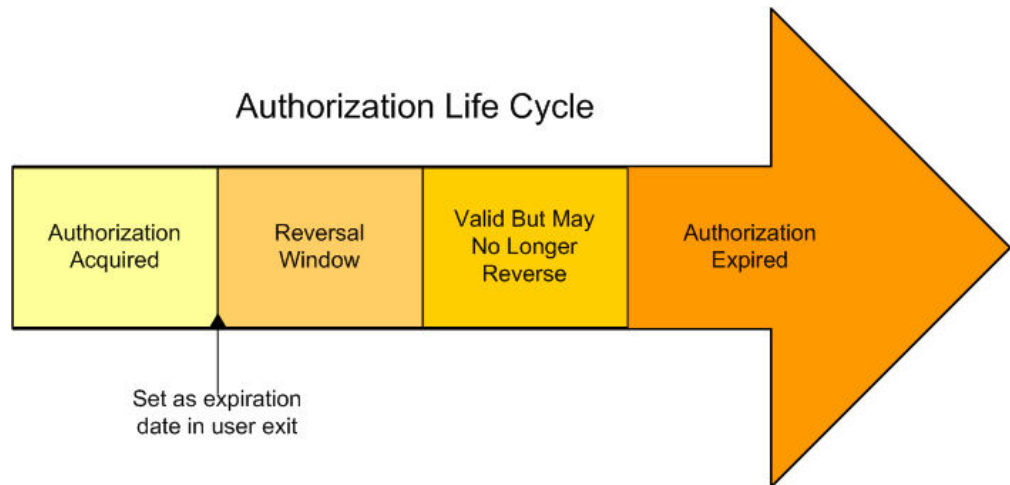


図 42. 承認の取り消しのウィンドウ

図 42 では、承認の取得の開始から取り消しウィンドウの終了まで、どの時点でも承認の取り消しが可能です。承認の取り消しで適切な時点は、可能な限り取り消し期間の終了に近い時点になります。追加料金を抑えることができるためです。この取り消し期間の早すぎる時点は適切ではありません。追加コストが発生するためです。

承認を取り消す場合、YFSCollectionCreditCardUE の有効期限 (expiration date) を、実際の承認の有効期限 (expiration date) ではなく、図 42 で示す時点に設定します。その結果、システムはこの時点で承認の取り消しを開始するため、承認の取り消しの処理、および最初の試行が失敗した場合の再試行を行うための十分な時間があります。

承認を取り消さない場合、有効期限 (expiration date) を実際の有効期限 (expiration date) に設定します。

実装やクレジット・カード (credit card) によって、取り消しウィンドウが図 42 で示したウィンドウと異なる場合があります。追加料金が発生することなく実際の有効期限 (expiration date) まで承認の取り消しが可能な場合があるため、「有効だが取り消し不可 (Valid But May No Longer Reverse)」期間が存在しない場合があります。

**構成オプション:** このセクションでは、2 つの承認取り消し構成オプションについて説明します。これらは、Distributed Order Management Applications Manager の「支払タイプの詳細」ウィンドウで使用可能です (「Sterling Selling and Fulfillment Foundation 分散オーダー管理 構成ガイド」を参照)。

- 取り消さない

- 有効期限満了時に取り消す

### 取り消さない

承認取り消し方法を実装しない場合、このオプションを選択できます。これは、通常の承認ライフサイクルが発生することを意味します。承認が取得され、一定期間後に、有効期限が切れます。有効期限が到来すると、新しい承認が取得されます。取り消しは行われません。

### 有効期限満了時に取り消す

このオプションが有効になっている場合、承認取り消しが同じ承認 ID の承認要請の形式で自動的に発生します。必ず負の金額になります。例えば、元の承認が \$60 に対するものであった場合、承認取り消し要請は、-\$60 に対するものになります。

承認の自動取り消しを有効にしており、該当する支払方法が承認取り消しをサポートしていない場合、Sterling Selling and Fulfillment Foundation は、承認に取り消し用のマークを付けますが、支払システムに伝達せず、コストも発生しません。

**承認取り消しの決済シナリオ:** 承認の取り消しについて、次に 3 つの決済シナリオを示します。

- 承認金額と決済料金が一致する場合
- 承認が期限切れの場合
- 承認金額と決済金額が異なる場合

### 承認金額と決済料金が一致する場合

ほとんどのシナリオでは、決済要求は、まだ期限切れでない承認に対して正確な金額分、作成されます。この状態では、取り消しや再承認の必要はなく、通常どおり目的の承認に対して決済を実行できます。

### 承認が期限切れの場合

承認要請が期限切れではあるものの再承認の前であれば、請求要請を作成できます。このシナリオでは、承認を取り消す必要はありません。承認期限が取り消しが許可されている最終日に設定されているときは、多くの場合、承認はまだ決済で有効です。

YFSCollectionCreditCardUE など、executeCollection のユーザーEXIT を使用して、料金をどのように処理するかを決定できます。実際の承認日は、どの程度前に支払タイプを取り消す必要があるかを示す支払タイプ情報に基づいて判断できます。

承認が実際に期限切れの場合、YFSCollectionCreditCardUE は請求に失敗したことを返すことができます。requestCollection は新しい請求要請を作成し、executeCollection で結果を返します。このユーザーEXIT の出力の内部には、recordAdditionalTransactions にフィールドがあります。これを使用して、YFSCollectionCreditCardUE ユーザーEXIT の内部で発生している追加の支払システム通信を記録できます。このユーザーEXIT は通常、1 つのレコードのみを更新します。executeCollection API は、この文書を使用して内部的に recordExternalCharges API を呼び出します。再承認または決済に失敗した場合、

YFSCollectionCreditCardUE は請求に失敗したことを返し、requestCollection は新しい請求要請を作成します。

### 承認金額と決済料金が異なる場合

インボイスを決済する際、承認要請が実際の請求要請と一致しないことがあります。このような不一致は、部分的な出荷、価格変更、またはその他の調整シナリオがあった場合に発生する可能性があります。例えば、\$100 の承認に対して \$60 の料金が発生したとします。この場合、次の戦略を使用します。

1. 元の \$100 の承認を取り消す。
2. \$60 に対する新しい承認を獲得する。
3. \$60 の新しい承認で決済する。

この戦略は、YFSCollectionCreditCardUE 内部に実装でき、recordAdditionalTransactions 出力フィールドを使用して必要な情報をレポートできます。さらに、料金レコードの承認 ID を更新することもできますが、recordAdditionalTransactions を使用して取り消し情報および再承認情報を入力する必要があります。

#### 承認の手動調整:

オーダー変更およびインボイスの承認を取り消すことによって、不一致の決済を最小化する選択ができます。オーダー変更の合計頻度によっては、これによって取引の手数料を節約できます。請求時の税および料金の調整は決済の直前に行われるため、これも取り消しおよび再承認の候補にできます。

手動調整を実行するには、recordExternalCharges API を使用して未完了承認取り消し要請を作成します。この時点で再承認を作成することはできません。取り消しが作成されると、支払エージェントによって、訂正金額の再承認が自動作成されます。

### 遅延再承認 (delayed reauthorization)

オーダーが在庫を待っている間に、承認が期限切れになり、顧客のクレジット明細がロックされ、セラーに対する費用の原因となる再承認が複数回行われる可能性があります。これらのトランザクションのコストを減らすために、構成可能な支払ルールにより、以下の遅延再承認 (delayed reauthorization) の選択肢が提示されます。

- **遅延再承認 (delayed reauthorization)** - 初期承認の作成後の遅延再承認 (delayed reauthorization)。初期承認が期限切れになると、再承認は、リリース日より構成可能な時間数前になるまで発生しません。
- **承認なしのスケジュール** - 初期承認なしで済ませ、リリース日より構成可能な時間数前になるまで、承認の実行を待ちます。

2 つの構成オプションには、「支払ルール」画面でアクセスすることができます。

**リリース日:** リリース日は、オーダーをリリースするよう通知が倉庫に送信される時期で、最終承認のタイミングの決定要因です。これは、オーダーの出荷が行われる出荷日と同じではありません。最終承認時間をリリース日より前にスケジュールリングすることで、最終承認が行われるまで明細がリリースまたは出荷されないよう保証されます。

**バンドル:** バンドル親は、「最初のバンドル・コンポーネントで請求 (Invoice on first bundle component)」ルールが選択されているときのインボイス作成と同じように、承認のために価格設定されます。インボイス作成について詳しくは、「*Sterling Selling and Fulfillment Foundation 分散オーダー管理 構成ガイド*」を参照してください。

**PS および DS 作業オーダー:** サービス明細では、リリース日ではなく「約束した予定開始日」を使用して、承認日を決定します。作業オーダーのアイテムは引き続きリリース日を使用します。

**調達、転送、および連鎖オーダー:** 調達、転送、および連鎖オーダーは、承認なしで開始できます。

**オーダー変更:** オーダーの支払関連エンティティーに関連する変更が発生すると、オーダーの支払ステータスが `AWAIT_AUTH` に変更されます。このような変更には、支払方法の変更、新規明細の追加やいずれかの明細の数量の増加による明細数量の増加、および価格、料金、税の変更などが含まれます。これらの変更が発生すると、支払集金トランザクションによってオーダーが再評価され、適切なアクション・コースが決定されます。

遅延再承認 (delayed reauthorization) が構成されている場合は、以下に該当するトランザクションや API もオーダー変更に含まれます。

- オーダーの価格の変更 (出荷調整と出荷に関する最終インボイスを除く)
- 予定出荷日の変更
- PS/DS 明細の予測予定日の変更

**支払ステータス:** スケジューリングの前に承認が必要な場合は、アクティブな承認に含まれていないオーダーの全体が再承認されます。支払ルールによってスケジューリング前の承認が要求され、価格変更が正 (増加) である場合、支払ステータスは `AWAIT_AUTH` になります。

スケジューリングの前の承認が要求されない場合、オーダーのリリース準備が完了するまで支払ステータスは `AUTHORIZED` で保持されます。

**注:** 遅延再承認 (delayed reauthorization) が構成されている場合、在庫を待機しているオーダーの支払ステータスが `AUTHORIZED` であるときは、完全な承認を示していない可能性があります。例えば、アイテム 1 とアイテム 2 の \$100 のオーダーがあらかじめ承認されているときに、そのうちの \$60 のアイテム 1 がただちに出荷され、アイテム 2 が 3 週間遅延するとします。この場合、承認の有効期限が切れた後に最初の \$100 の `AUTHORIZED` ステータスが、必ず残りの \$40 分の承認を構成するとは限りません。ただし、最終承認の発生時は、この `AUTHORIZED` ステータスにより、リリースまでのオーダー・プロセスに沿ってこのアイテムまたはオーダーの処理を続けます。

## 決済プロセス

支払いの決済には、オーダーに対して記録された金額の資金を集金する作業が含まれます。例えば、クレジット・カード (credit card) を使用した場合、決済プロセスには特に、支払システムへの接続、およびそのクレジット・カード (credit card) に対して必要な金額の資金を集金する処理が含まれます。支払タイプによっては、こ

の決済ステップは必要な場合も、必要ない場合もあります。これは、Sterling Selling and Fulfillment Foundation でセラーの支払ルールで構成可能です。オーダーで支払処理が必要な場合、そのオーダーは、決済が完了してオーダーが「支払済み」ステータスにならない限りパージされません。使用可能なイベントについては、「Sterling Selling and Fulfillment Foundation Javadocs」で支払処理 API を参照してください。

支払集金は、オーダーを分析して決済要求を作成します。決済要求は、Sterling Selling and Fulfillment Foundation に対する決済完了の要求です。オーダーに適用される支払ルールで承認と決済の両方が要求される場合は、期限切れでない既存の承認に対してのみ決済要求が作成されます。

外部集金システムを使用して決済を取り扱う場合、支払集金を売掛金を使用して記録しないよう選択できます。決済が Sterling Selling and Fulfillment Foundation データベースに記録されない場合、オーダーの決済要求は作成されません。決済は外部システム経由で、インボイスとともに公開された情報を使用して実行されることを前提としています。

支払実行は、決済要求をモニターします。この時間トリガー・トランザクションが、決済を実行するためのユーザーEXIT を提供します。このユーザーEXIT は、次のいずれかの方法で決済要求を処理できます。

- 決済をただちに実行して (売掛金データベースと接続して)、決済金額を返す。
- 支払システムとのインターフェースが操作不能の場合、後で再試行するように要求を出す。
- 非同期処理を要求する (つまり、コンソールがこのオーダーのために支払システムに接続することはない)。支払インターフェース (カスタム拡張機能) は、Sterling Selling and Fulfillment Foundation API 経由で支払に対応する詳細を更新する必要があります。これは、一般的に、支払システムとのインターフェースがバッチ・モードで実行されている場合に該当します。

完全なオーダー金額の支払いが受け取られると、Sterling Selling and Fulfillment Foundation の時間トリガー・トランザクションにより、オーダーの支払ステータスが「支払済み」に変更されます。

---

## 顧客アカウント (customer account) 支払いプロセス

顧客アカウント (customer account) がシステム内でセットアップされた場合は、それらの顧客アカウント (customer account) に属するオーダーの支払い処理は、一般支払いプロセスの支払い処理と異なります。

顧客アカウント (customer account) は、支払いルールに関する「内部で管理される顧客アカウント (customer account)」フラグを設定して、その顧客アカウント (customer account) についてアカウント限度を設定することにより、システム内でセットアップすることができます。この機能が効率よく動作するように、顧客アカウント (customer account) の資金不足により適用する保留タイプ・ルールを構成する必要があります。さらに、推奨されない保留タイプの機能はオンにしないでください。保留タイプについて詳しくは、139 ページの『オーダー保留処理』を参照してください。

## 承認処理

顧客アカウント (customer account) を管理する際、承認のために外部システムに接続されることも、AUTHORIZATION 要求に対して YFSCollectionCustomerAccountUE ユーザーEXIT が呼び出されることもありません。システムは、顧客アカウント (customer account) で使用可能な資金額を確認します。要請金額が顧客アカウント (customer account) で使用可能な金額以下の場合、AUTHORIZATION 要求はクローズされ、顧客アカウント (customer account) に対する未承認の金額が増加されます。要請金額が使用不可の場合、Applications Manager で構成された保留、「顧客アカウント (customer account) の資金不足により適用する保留タイプ」がオーダーに適用されます。要請金額が、負の承認要請で顧客アカウント (customer account) に追加されます。

### 資金不足の場合の決済処理

アカウントの資金が不足している場合は、オーダーが保留になります。保留は、システムで構成されているルールによって決定されます。保留中のオーダーが承認されると、保留が解除されます。

保留中のオーダーを再承認の対象とするには、次のようにします。

- 承認の手動オーバーライド

例外的なシナリオでは、十分な資金を使用できない場合でもオーダーが処理されます。顧客アカウント (customer account) の予約額が増加し、オーダーの合計額が処理済みの金額と見なされます。

- 顧客アカウント (customer account) 上限の変更

顧客がエンタープライズに対しアカウントの上限の増加を要求する場合があります。顧客アカウント (customer account) の上限が増加されたときは、それまで処理できなかったオーダーを再処理する必要があります。

このような保留中のオーダーの決済処理を起動するため、triggerRequestCollection API が用意されています。この API は、顧客アカウント (customer account) 上限の変更時に呼び出す必要があります。

- 未完了オーダーの金額の削減

顧客アカウント (customer account) の承認オーダーがキャンセルされた場合、または内部か外部でオーダーが決済された場合、その顧客アカウント (customer account) の未完了オーダーの金額が削減されます。

このような保留中のオーダーの決済処理を起動するため、triggerRequestCollection API が用意されています。この API は、オーダー変更後の正常完了時に呼び出す必要があります。

- 顧客アカウント (customer account) のオーダーの決済

顧客アカウント (customer account) のオーダーが決済されると、未完了オーダーの実行が開始されます。

このような保留中のオーダーの決済処理を起動するため、triggerRequestCollection API が用意されています。この API は、顧客勘定残高の変更時に呼び出す必要があります。

## 有効期限 (expiration date)

顧客アカウント (customer account) について獲得された承認には有効期限 (expiration date) はありません。

## 決済プロセス

支払決済プロセスには、オーダーに対して記録された金額の資金を集金する作業が含まれます。内部的に維持されている顧客アカウント (customer account) に関連するオーダーの決済プロセスは、一般的な支払処理の決済プロセスとほぼ同じです。一般的な支払プロセスについて詳しくは、268 ページの『決済プロセス』を参照してください。内部的に維持されている顧客アカウント (customer account) の決済プロセスにおける唯一の違いは、これらのアカウントの未完了オーダーの金額は、決済された金額分、減額される点です。

---

## 決済処理のトランザクション

次の標準の時間トリガー・トランザクションによって、Sterling Selling and Fulfillment Foundation は、外部システムと会計情報を交換できます。

- オーダー・インボイスの作成 - 1 件のオーダーに対して 1 つ以上のインボイスを作成します。これはトランザクションから派生し、インボイスが出荷とリンクされていないシナリオ (返品など) での使用が推奨されます。
- 出荷インボイスの作成 - 出荷に対して 1 つ以上のインボイスを作成します。これはトランザクションから派生し、出荷に対するインボイスの作成での使用が推奨されます。
- 支払実行 - 承認、および請求が保留されているすべての要請を処理します。承認を必要とするすべての未完了の要請と請求を読み取り、適切な外部プログラムを起動して要請を実行します。
- 支払集金 - オーダーを分析して、承認または請求要請の作成が必要な金額を判定します。
- インボイスの送付 - 外部の売掛勘定システムに送信可能なインボイス・データを発行します。インボイスは、その作成時または支払の集金後に発行できます。

これらの時間トリガー・トランザクションについて詳しくは、see the *Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイドを参照してください。

---

## カスタム・トランザクションの使用

ビジネスで、外部システムとの追加の会計関連の統合が必要な場合、カスタム時間トリガー・トランザクションを作成することができます。カスタム時間トリガー・トランザクションを作成する際に、Sterling Selling and Fulfillment Foundation 支払関連の外部プログラムおよび API を呼び出します。

## 支払関連 API

次の API は外部会計システムとの統合を実現します。

- `executeCollection()` - 要請が作成されたオーダーの承認および請求の個別要請を処理します。要請を個別に処理する必要がある場合は、必要に応じてこの API を呼び出して、個別の要請を実行できます。この API は、適切な外部プログラムを呼び出して、決済処理を実行します。
- `recordCollection()` - Sterling Selling and Fulfillment Foundation によって作成された個別の要請に対して処理された承認金額と請求金額を記録します。これは、バッチ・モードで外部決済処理システムとのインターフェースを行い、一連の要請を処理してから、Sterling Selling and Fulfillment Foundation で結果を更新する場合に便利です。
- `processOrderPayments()` - 1 回の呼び出しで、`requestCollection()` API と `executeCollection()` API を起動して、オーダー全体の金額をオンラインで承認して、不必要な費用を回避できるようにします。`processOrderPayments()` API は、オーダー上の支払方法のリストを取得します。
- `executePaymentTransactions()` - オーダーの作成または確認前に、承認および請求の要請を処理します。要請が行われたオーダーの複数の支払要請の承認および請求が行われます。この API では、該当する外部プログラムを呼び出して決済処理を実行する前に、支払条件が検証されます。検証時に API でエラーが発生した場合は、外部プログラムを呼び出さずに処理が終了します。外部プログラムの処理中にエラーが発生した場合は、その外部プログラムを終了して、それまでの正常完了したすべての支払トランザクションと最初にエラーになったトランザクションを返します。
- `scheduleOrder()` - オーダーを再承認する時点を計算します。
- `getOrderDetails()` - トランザクション・テーブルまたは履歴テーブルから、特定のオーダー番号に関する詳細情報を返します。情報には、オーダー明細、価格情報、総合計、請求トランザクション詳細、オーダー状況、およびその他のオーダー関連情報が含まれます。
- `recordExternalCharges()` - Sterling Selling and Fulfillment Foundation の外部で実行された承認および請求のすべてが記録されます。この API は、外部システムで受領した支払いの記録にも使用されます。この例としては、相当期間の後に支払いを受け取るバイヤー・アカウントが挙げられます。料金が非同期で処理されることが `executeCollection` API によって既に Sterling Selling and Fulfillment Foundation に通知されている場合、`recordExternalCharges` が実際の料金を記録します。
- `requestCollection()` API - オーダーを分析して、承認または請求要請を作成する必要がある金額を判定します。
- `recordInvoiceCreation()` - オーダーに対するクレジット伝票、デビット伝票、または情報インボイスを記録します。この API を使用してクレジット伝票やデビット伝票を記録すると、Sterling Selling and Fulfillment Foundation のトリガーとして機能し、請求対象金額の決済プロセスが起動されます。Sterling Selling and Fulfillment Foundation では、この API によって作成された情報インボイスの決済処理は行われません。これらのインボイスは、外部システムによって実際の処理が行われ、可視性のためにオーダーに記録される運送会社の請求などのキャプチャーに使用できます。
- `createAccessToken` - ランダムで固有のアクセス・トークンを生成します。これは、Sterling Sensitive Data Capture Server アプリケーションに対する呼び出しの認証に使用されます。



- `manageChargeTransactionRequest` および `getChargeTransactionRequestList` - これらの API は、オーダーに関連付けられた請求トランザクション要請 ID の承認プロセスの管理に使用されます。`getChargeTransactionRequestList` API は、承認が必要な請求トランザクション要請を取得します。`manageChargeTransactionRequest` API は、オーダーの承認有効期限をシステム日付にリセットして、これによって `requestCollection` API で承認および請求要請の作成が必要な金額を判定します。API 呼び出しのそれぞれは、1 件のオーダーにおける複数の請求トランザクション要請を同時に処理できます。

これらの API およびそれによって発生するイベントについて詳しくは、*Sterling Selling and Fulfillment Foundation Javadocs*を参照してください。

## 支払い関連の外部プログラム

支払実行の時間トリガー・トランザクション、および `executeCollection()` API によって呼び出される外部プログラムは次のとおりです。

- `YFSCollectionCreditCardUE` - クレジット・カード (credit card) に対する承認および請求を処理します。
- `YFSCollectionCustomerAccountUE` - バイヤー・アカウントに対する承認および請求を処理します。
- `YFSCollectionOthersUE` - 贈答用小切手、割引クーポンなど、その他の支払方法に対する承認および請求を処理します。
- `YFSgetFundsAvailableUE` - オーダーの支払方法における使用可能資金を判定する外部ロジックへの接続を有効にします。
- `YFSCollectionStoredValueCardUE` - ストアード・バリュー・カードの承認または請求のためのカスタム・ロジックに接続する機能を提供します。

---

## データベースの詳細

*Sterling Selling and Fulfillment Foundation* データベースでは、主に 6 つのテーブル (`YFS_CHARGE_TRANSACTION`、`YFS_CHARGE_TRAN_DIST`、`YFS_ORDER_HEADER`、`YFS_CHARGE_TRAN_REQUEST`、`YFS_CHARGE_TRAN_RQ_MAP`、および `YFS_PMNT_TRANS_ERROR`) に支払データを保持します。

システムでは、ドライバー・テーブルとして `YFS_CHARGE_TRANSACTION` テーブルを使用して、承認および料金を処理するための時間トリガー・トランザクションとインターフェースを取ります。また、ドライバー・テーブルは、いつでもオーダーに対するすべてのクレジットおよびデビットのジャーナルとしても機能します。

### YFS\_CHARGE\_TRANSACTION テーブル

`YFS_CHARGE_TRANSACTION` テーブルのキー・フィールドおよびそれぞれの値について、以下に詳述します。

表 36. YFS\_Charge\_Transaction\_Table フィールド

フィールド	説明
STATUS	<p>このフィールドでの有効値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• OPEN - YFS_CHARGE_TRANSACTION テーブルに新しい要求が作成されます。これらのレコードは、支払トランザクションによってピックアップされます。</li> <li>• CLOSED - 支払集金が Sterling Selling and Fulfillment Foundation に対して外部で実行され、recordCollection() API または支払いを介したレポート済みの集金の詳細が、支払実行時間トリガートランザクションを使用して承認または収集されました。</li> <li>• CHECKED - YFS_CHARGE_TRANSACTION テーブル内のレコードは妥当性検査済みです。これらのレコードに対して、これ以上の処理は発生しません。</li> <li>• ERROR - 関連した支払方法の承認または請求が失敗しました。</li> </ul>
CHARGE_TYPE	<p>このフィールドでの有効値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• ADDITION - オーダー明細のオーダー数量が増えました。</li> <li>• ADJUSTMENTS - オーダーのユーザー指定 TotalAmount と計算 TotalAmount が一致していません。</li> <li>• AUTHORIZATION - 承認が要求されます。この要求は、requestCollection API を呼び出すか、または支払集金の時間トリガー・トランザクションを介して作成できます。</li> <li>• 取り消し - 1 つのオーダーまたは 1 つのオーダーの一部が取り消されました。</li> <li>• CHANGE_PRICE - オーダーの価格が再設定されました。オーダーの価格は、単位価格、料金、および税金を変更することによって再設定できます。</li> <li>• CHARGE - 料金が要求されます。この要求は、requestCollection API を呼び出すか、または支払集金の時間トリガー・トランザクションを介して作成できます。このような要求は、オープン・オーダーまたは出荷に対してなされます。</li> <li>• CREATE_ORDER - システム内でオーダーが作成されました。</li> <li>• ORDER_INVOICE - CREATE_ORDER_INVOICE トランザクションを使用して、オーダーの一部または全体が請求されました。</li> <li>• REQUEST_SETTLEMENT - オーダー明細のオーダー数量 (一部または全体) が請求前に決済されています。これは、事前決済要請で発生します。</li> <li>• RETURN - 返品オーダーの一部または全体が請求されます。</li> </ul>

表 36. YFS\_Charge\_Transaction\_Table フィールド (続き)

フィールド	説明
	<ul style="list-style-type: none"> <li>• SHIPMENT - 1 つのオーダーまたは 1 つのオーダーの一部が、出荷インボイス作成の時間トリガー・トランザクションを使用して出荷された後、請求されました。</li> <li>• SHIPMENT_ADJUSTMENT - この請求要請は、オーダー全体が出荷された後でなんらかの調整が必要な場合に作成されます。</li> <li>• SPLIT_LINE - オーダー明細が分割され、新規明細に決済処理が必要です。</li> <li>• TRANSFER_IN - このレコードは、金額が別のオーダーから転入されている場合に表示されます。このリンクをクリックすると、金額の転送元であるオーダーに移動します。</li> <li>• TRANSFER_OUT - このレコードは、金額が別のオーダーへ転送されている場合に表示されます。このリンクをクリックすると、金額の転送先であるオーダーに移動します。</li> </ul>
CREDIT_AMOUNT	オーダーの関連付けられた支払方法に対して顧客へのクレジットとして扱われる金額。顧客から集金された金額。この金額は、実際の資金集金が行われた後でのみ、増加します。
DEBIT_AMOUNT	このオーダーに対して顧客へのデビットとして扱われる金額。
BOOK_AMOUNT	未完了オーダーの金額。どの時点でも、この金額は、まだ出荷されていない合計オーダー金額を反映します。この金額は、オーダーの作成、オーダー数量の追加、明細の追加、価格、料金、および税の増加によって増加され、取り消し、価格、料金、および税の削減、およびインボイス作成によって削減されます。
OPEN_AUTHORIZED_AMOUNT	顧客の支払方法に対して承認された金額。承認は、実際の資金集金に使用されます。
REQUEST_AMOUNT	承認要請または料金要請が作成されている金額。
SETTLED_AMOUNT	事前決済要請が作成されている金額。
USER_EXIT_STATUS	このフィールドは、支払実行トランザクションが処理のためにレコードをピックアップすると「INVOKED」に更新されます。外部支払いシステムからの処理の待機中は、このフィールドは「ONLINE」に更新され、エージェントは該当レコードをピックアップできなくなります。トランザクションが正常に戻ると、元どおり <空欄> に設定されます。このフィールドにデータが設定されたままである場合は、外部システム呼び出しで問題が発生しており、手操作による介入が必要であることを表します。

以下の表では、オーダー・ライフサイクルの様々な時点で YFS\_CHARGE\_TRANSACTION テーブルにログ記録される各種金額について説明します。

表 37. オーダー・ライフサイクル内での支払料金の金額

時点	料金タイプ	クレジット金額	デビット金額	簿価	承認済み金額	要請済みの金額
オーダーの作成	ORDER_CREATE			+ オーダーに関する合計金額		

表 37. オーダー・ライフサイクル内での支払料金の金額 (続き)

時点	料金タイプ	クレジット金額	デビット金額	簿価	承認済み金額	要請済みの金額
フロントエンドに対して実行され、CreateOrder XML と共に渡される承認	AUTHORIZATION				+ 渡し済みとしての承認済み金額	
フロントエンドに対して集金され、CreateOrder XML と共に渡される資金	CHARGE	+ 渡し済みとしての料金の金額				
取り消し	CANCEL			- 以下のものとしての取り消し金額		
支払集金がコンソール・データベースに記録される場合の出荷	SHIPMENT		+ 出荷金額	- 出荷金額		
支払集金がコンソール・データベースに記録されない場合の出荷	SHIPMENT	+ 出荷金額	+ 出荷金額	- 出荷金額		
承認要請の作成	AUTHORIZATION					承認金額
請求要請の作成	CHARGE					料金の金額
実際の承認					承認された金額	
実際の資金集金	CHARGE	集金した金額				
承認の取り消し	AUTHORIZATION			- 金額承認済み		
API/コンソールを介した価格変更	CHANGE_PRICE			価格変更金額		
デビット伝票	ADJUSTMENT		伝票上の金額			
クレジット伝票	ADJUSTMENT		伝票上の負の金額			
返品	RETURN		顧客に提供される合計クレジットの負数			

表 37. オーダー・ライフサイクル内での支払料金の金額 (続き)

時点	料金タイプ	クレジット金額	デビット金額	簿価	承認済み金額	要請済みの金額
明細のオーダー数量が増加します	ADDITION			追加金額		
オーダーが請求されます	ORDER_INVOICE		請求済み金額	請求額の負数		
オーダー明細が分割されます	SPLIT_LINE			合計の変更		
最終出荷の作成	SHIPMENT_ADJUSTMENT			オーダーの簿価と請求額合計との差		
返品オーダーの作成	TRANSFER_IN	オーダーへ転送済みの金額				
返品オーダーの作成	TRANSFER_OUT	オーダーへ転送済みの金額の負数				
繰延クレジット項目	DEFERRED_CREDIT	据え置き金額				

## YFS\_CHARGE\_TRAN\_DIST テーブル

このテーブルには、返金中の事前集金済み資金のレコードが含まれます。このテーブルには、事前集金済み請求トランザクション (転入、請求)、返金先の請求トランザクション (転出、負の請求)、および返金される金額のキーも含まれます。

表 38. YFS\_CHARGE\_TRAN\_DIST テーブルのフィールド

フィールド	説明
DISTRIBUTED_FROM_CT_KEY	この請求トランザクション分配と関連付けられている請求トランザクション。これは資金を分配した請求トランザクションです。
DISTRIBUTED_TO_CT_KEY	この請求トランザクション分配と関連付けられている請求トランザクション。これは、資金の分配先となった請求トランザクションです。
CHARGE_AMOUNT	請求トランザクション・レコード DISTRIBUTED_TO_CT_KEY から請求トランザクション・レコード DISTRIBUTED_TO_CT_KEY に分配された金額。

## YFS\_ORDER\_HEADER テーブル

YFS\_ORDER\_HEADER テーブル内の決済処理に関連したキー・フィールドは PAYMENT\_STATUS です。262 ページの『支払ステータス』を参照してください。

## YFS\_CHARGE\_TRAN\_REQUEST テーブル

このテーブルには、決済承認および ID の要求が含まれています。ORDER\_HEADER\_KEY と ID の組み合わせにより、該当オーダーについての一連の固有グループが作成されます。承認は、各承認が 1 つの ID のみに対応するように分割されます。

表 39. YFS\_CHARGE\_TRAN\_REQUEST テーブルのフィールド

フィールド	説明
CHARGE_TRAN_REQUEST_KEY	基本キー
ORDER_HEADER_KEY	要求の対象であるオーダー。
CHARGE_TRAN_REQUEST_ID	オーダーに関する金額の要求の固有 ID。
PAYMENT_STATUS	YFS_ORDER_HEADER テーブル内の支払ステータスに類似。
REQUEST_SEQUENCE	グループが承認されているオーダー。REQUEST_SEQUENCE がヌルでないレコードが最初に処理されます。null シーケンスは、CHARGE_TRAN_REQUEST_ID で昇順にオーダーされます。
MAX_REQUEST_AMOUNT	承認されるために別のグループへの分配が始まる前に、この固有 ID が表す最大金額。

## YFS\_CHARGE\_TRAN\_RQ\_MAP テーブル

このテーブルは、請求トランザクション要求を、それら要求を満たす請求トランザクションにマップします。

表 40. YFS\_CHARGE\_TRAN\_RQ\_MAP テーブル・フィールド

フィールド	説明
CHARGE_TRAN_REQUEST_MAP_KEY	基本キー
CHARGE_TRANSACTION_KEY	YFS_CHARGE_TRANSACTION テーブルへの参照。
CHARGE_TRAN_REQUEST_KEY	YFS_CHARGE_TRAN_REQUEST テーブルへの参照。
REQUEST_AMOUNT	マップされた請求トランザクションに関して要求された金額。
PROCESSED_AMOUNT	マップされた請求トランザクションに関して処理された金額。

## YFS\_PMNT\_TRANS\_ERROR テーブル

このテーブルには、決済処理中に外部プログラム出力から取り込まれた一連のエラー・メッセージが含まれます。請求トランザクションのレコードが削除される場合、該当の請求トランザクションに関連付けられたトランザクション・エラーもすべて削除する必要があります。これは、changeOrder() API、deleteOrder() API、および PurgeOrderAgent にあてはまります。

表 41. YFS\_PMNT\_TRANS\_ERROR テーブル・フィールド

フィールド	説明
PMNT_TRANS_ERROR_KEY	この支払トランザクション・エラー・レコードと関連付けられている固有 ID。
CHARGE_TRANSACTION_KEY	この支払トランザクション・エラー・レコードと関連付けられている請求トランザクション。請求トランザクションごとに、複数のメッセージが用意されていることがあります。

表 41. YFS\_PMNT\_TRANS\_ERROR テーブル・フィールド (続き)

フィールド	説明
MESSAGE_TYPE	この支払トランザクション・エラー・レコード用のエラー・メッセージのタイプ。
MESSAGE	この支払トランザクション・エラー・レコード用のエラー・メッセージ。





---

## 第 13 章 物流管理

---

### 物流管理とは

物流管理には、顧客が望むタイミングおよび方法でオーダーが出荷されるようにオーダー明細レベルの出荷処理を計画し、モニターすることが含まれます。Sterling Selling and Fulfillment Foundation は、バイヤーの出荷要求、エンタープライズの設定の検討、および出荷処理をカスタマイズする機能とともに入庫コンプライアンスをサポートする多数の機能を提供します。

---

### 出荷の計画に関する考慮事項

1 つ以上のオーダー明細の出荷を計画する際は、次のような目標が存在します。

- コスト効率の高い方法でアイテムを配達する。
- 顧客の出荷に関する要件に合わせる。例えば、同じ組織内の複数の部門からのオーダーは、確実にそれぞれの部門に個別に出荷されるようにします。
- アイテムを適時に配達する。

実務上、効率的な出荷計画の作成は、多数の実際的な考慮事項に応じて異なります。多くの場合、配達の速さと配達のコストの間にはトレードオフがあります。アイテムを出荷するエンタープライズの多くは、配送される出荷および集合・混載の数を最小化しようとしませんが、バイヤーはバイヤーの社内プロセスに適した形式で出荷を受け取ることが重要であると考えます。

Sterling Selling and Fulfillment Foundation を使用すると、これらの出荷計画に関する考慮事項の評価方法を設定できます。Sterling Selling and Fulfillment Foundation を使用して、バイヤーとの対話の条件およびプロセスを設定することにより、配送計画の効率を向上できます。

---

### 条件

出荷計画で使用する重要な条件の一部を以下に示します。

#### 運送会社

運送会社は、配送サービスの提供者です。オーダーの一部であるアイテムの配達に複数の運送会社を含めることができます。

#### 配送計画

配送計画とは、1 つまたは複数の出荷地から 1 つ以上の宛先に 1 つ以上のオーダーを配達するために必要な一連の移動全体を指します。配送計画は、出荷、集合・混載、出荷地、停止場所、および出荷先で構成されます。

## 出荷先

出荷先とは、集合・混載の移動ルートの内、集合・混載内の残りすべての出荷がドロップオフされる最後のノードまたは住所です。

## 集合・混載

集合・混載では、2 地点間の 1 つ以上の完全な出荷（部分的出荷は対象外）が配送されます。出荷地と出荷先はそれぞれ 1 つですが、中継地の停止場所は複数指定できます。出荷地またはすべての中継地の停止場所で、集合・混載に出荷を追加できます。ドロップオフは、集合・混載の出荷先または中継地の停止場所で可能です。

## 作成元

出荷地は、集合・混載の発信元ノードです。

## 出荷

出荷とは、単一の出荷者から単一の荷受人に、1 つ以上のオーダーおよびオーダー明細を配達することです。出荷は、複数の集合・混載および複数の配送業者により配達されることがあります。

## 停止場所

停止場所は、出荷がピックアップまたはドロップオフされる任意のロケーションです。集合・混載には、その移動ルートを指定する停止順序が設定されます。

## 出荷計画戦略

計画にあたり、ユーザーは、アイテムを安価で迅速に配達し、顧客の要件を満たすことを考えます。これらの配達計画目標を達成するには、次の戦略を使用します。

### 複数アイテムの単一の出荷への集約

多くの場合、オーダーの複数のアイテムを 1 つの出荷で出荷した方が効率的です。すべてのアイテムを同じロケーションに出荷する場合は、すべてのアイテムを単一の出荷で一括して出荷した方が安いことがほとんどです。

### 複数の出荷の 1 つの集合・混載への集約

複数の出荷を、配送業者によって輸送される単一の集合・混載にまとめることができます。出荷を 1 つの集合・混載に集約する際は、次のように多くの要因を考慮する必要があります。

- 可能なルート
- 使用可能な配送業者
- 配送業者の貨物輸送キャパシティー
- 中間ノードまたは顧客の特別な要件

例えば、低コストのバルク配送業者が使用可能でも、その配送業者で使用している容器が顧客の受入ドックには大きすぎる可能性があります。

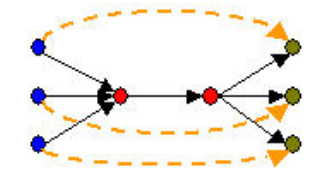
## ルーティングの最適化

事情によっては、異なるルーティング戦略を使用することで、配達時間や配達コストを最適化できます。差出人からの直接ルートが最適なこともあれば、複数の出荷をピックアップして、それらを単一の宛先に配達するのが最も合理的なこともあります。

次の図は、最も一般的な配達ルートのシナリオをいくつか示しています。この図から、使用可能な幅広いルーティング戦略を理解できます。

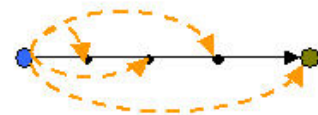


**Model 1 – Single Shipment, Single Load.**  
For simple shipments.



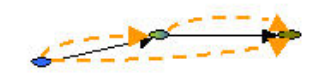
**Model 2 – Many Shipments, Consolidate & Break Into loads**

Allows multiple origins to consolidate (for example, at port) and be moved by second transport as a single load. Then broken back into initial structures to complete separate voyages.



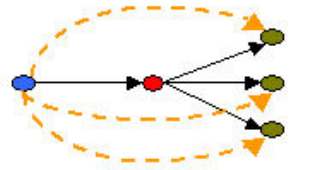
**Model 3 – Many Shipments, Single load**

Records a single load that makes stops to drop-off x amount of shipments.



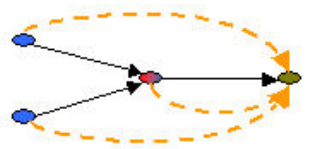
**Model 4 – Many Shipments, Intermediate Drop-Off & Pick-Up**

Allows efficient use of available transport resources to create loads in transit.



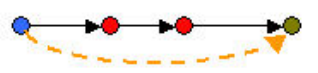
**Model 5 – Single Origin, Multiple Shipments, Break**

Shipments are shipped from an origin as a single load and subsequently broken at 'nearest' point to different destinations. Separate loads make final delivery.



**Model 6 – Single Shipment, Merge at one Origin**

Example of a merge in transit where components are sourced from disparate sites and merged at the source of the high value component.



**Model 7 – Single Shipment, Interline loads**

Although a single origin/destination pair, the load, for cost/optimum performance reasons, moves with more than one carrier.

図 43. 配達ルートのシナリオ

## 出荷計画機能

Sterling Selling and Fulfillment Foundation は、柔軟な出荷計画を提供するさまざまな機能を備えています。出荷の集約、集合・混載の作成、およびルーティングをどのように実行するかをコントロールする条件を設定できます。これらの条件はエンタープライズを対象に設定される他、各バイヤー組織を対象に設定することもできます。その後、競合する条件があれば、優先する組織の条件を設定できます。

主要な配達計画機能には次のようなものがあります。

- 経済的な出荷パラメーターの使用
- ルーティングの決定

さらに、手動ルーティングを使用することで、出荷のルーティング方法を明示的に設定することもできます。288 ページの『配送計画』を参照してください。

### 経済的な出荷パラメーターの使用

2 つ以上の出荷をまとめて出荷することによって、輸送コストの大幅な節約がもたらされる可能性が高まります。これは、まとめられたオーダーによるコスト節約と配達日を遵守する可能性との間のトレードオフももたらします。

経済的な出荷パラメーター設定は、以下の項目間の前記のトレードオフを評価するために使用されます。

- 多くのアイテムを同時に出荷できるようになるまで出荷を保留すること
- タイムリーに出荷すること

出荷は、以下の 2 つの方法で保留できます。

1. 複数のオーダーが指定の時刻にあり、オーダーを単一出荷にまとめてコストを削減する。
  - a. 同じ出荷日の複数のオーダー
  - b. 異なる出荷日の複数のオーダー
2. 集約できる別のオーダーを期待して既存のオーダーを保留する。

#### シナリオ 1-A: 同じ出荷日で指定の時刻に対応可能な複数のオーダーがある

これは、配達日に支障をきたすことなく、すべてのオーダーを 1 つの出荷に集約できるという理想的なシナリオです。

これは、すべてのオーダーが同じ出荷日を共有するため、可能です。

#### シナリオ 1-B: 異なる出荷日で指定の時刻に対応可能な複数のオーダーがある

このシナリオでは、各オーダーの配達日を満たすことに関してトレードオフが発生します。ESP 設定が有効になり、許容限界内に遅延を保持します。

このセットのオーダーの 2 つ以上のオーダーが「許容出荷遅延」パラメーターで指定された最大日数または適用可能出荷期間 (オーダーを出荷する必要がある日付範囲) を超えている場合、適用可能なオーダーのみがまとめられます。

例えば、オーダー O1、O2、および O3 の出荷期限が 1 日おき (D1、D2、および D3) になっていて、許容出荷遅延パラメーターが 1 日に設定されている場合、ESP は、オーダー O1 と O2、または O2 と O3 のいずれかをまとめ、3 つすべてはまとめません。これは、オーダー O1 と O3 の出荷期限の 2 日の差が許容出荷遅延を超えているためです。

## シナリオ 2: まとめることができる別のオーダーがあることを期待して既存のオーダーを保留する。

同じ出荷先住所への別のオーダーがあることを期待して、出荷できる 1 つのオーダー/出荷が保留されます。

ESP 設定は、次の処理のために解放する必要がある前に、オーダーを保留できる期間を決定します。ESP 設定により、通常オーダーは、重量または体積のしきい値を満たすまで、または「許容出荷遅延」パラメーターで指定された最大日数あるいは出荷の配送期間に達するまで保留されます。

出荷の配送期間は、出荷を配送する必要がある期間を表し、出荷の「出荷指定日」と「出荷期限日」によって決まります。

「出荷期限日」は、以下のように計算されます。

### 手順 1 顧客の出荷指定日の確認

オーダーに顧客の出荷指定日がある場合、これを使用します。

オーダーに顧客の出荷指定日がない場合、以下のいずれかの方法を使用します。

- オーダーに指定配達日がある場合、顧客の出荷指定日は、指定配達日からグローバル通過時間を引き、結果として得られた日付のシフト終了時刻を使用して「シフト終了時刻」パラメーターに基づいて計算されます。
- 指定配達日がオーダーに対して指定されていない場合、現在の時刻が顧客の出荷指定日とみなされます。

### 手順 2 出荷期限日の計算

顧客の出荷指定日を確認した後、以下の式を使用して出荷期限日を計算します。

$$\text{MustShipBeforeDate} = \text{Min} (\text{Max} (\text{ReqShipDate}, \text{CustReqShipDate} + \text{ESPDelayDays}), \text{ReqCancelDate})$$

ここで、

- MustShipBeforeDate は、出荷を配送する必要がある日です。
- ReqShipDate は、顧客が要求する最も早い出荷日です。
- CustReqShipDate は、手順 1 で確認した日です。
- ESPDelayDays は、出荷を構成するオーダーの最も早い出荷指定日から出荷を遅延できる日数です。ESP の遅延日数は、オーダーの運賃条件 (freight terms) の構成および ESP パラメーターの管理者に基づいて、パイヤーの入庫コンプライアンスまたはエンタープライズの出庫制約あるいはハブ (HUB) の出庫制約から取得されます。
- ReqCancelDate は、顧客が指定した、出荷を受け取る最後の日付です。

## 異なる出荷指定日の出荷をまとめるために使用されるロジック

新しいオーダー・リリースをまとめた出荷の出荷予定日と出荷期限日の間に出荷指定日がある場合のみ、異なる出荷指定日の 2 つの出荷を 1 つの出荷にまとめられます。

出荷集約を行うためには、出荷期限日を適切な方法で設定する必要があります。例えば、キャンセル指定日が出荷期限日より早い場合、出荷期限日をキャンセル指定日に設定する必要があります。

注: シナリオ 1-A または 1-B は、複数のオーダーを 1 つの出荷にまとめるとシナリオ 2 に切り替わり、この出荷は、この出荷にまとめることができる別のオーダーを期待して保留されます。

例えば、オーダー O1 と O2 または O2 と O3 が出荷 S1 にまとめられると、この出荷は、S1 にまとめることができる別のオーダーがあることを期待して保留できます。

ESP の使用はオプションですが、バイヤーとエンタープライズの両方が ESP を設定できます。これが設定されると、運賃条件 (Freight Terms) の設定が使用されて、最初に使用する ESP を決定します。

## ルーティングの決定

ルーティングにより、出荷を配達する方法が決定されます。ルーティング・プロセスの結果、出荷が既存の集合・混載に割り当てられるか、またはその出荷用に新規集合・混載が作成されます。

Sterling Selling and Fulfillment Foundation は、外部ルーティングを使用するか、あるいは手動ルーティングにより、Sterling Selling and Fulfillment Foundation 内で定義されたルーティング・ガイドに基づいてルーティングを実行します。外部ルーティングは、動的ルーティングとも呼ばれ、外部リソースを使用して、集合・混載への出荷の割り当てを決定します。

### ルーティング・ガイド

ルーティング・ガイド は、出荷のルーティング方法、および使用する配送業者とサービスと決定する条件のリストです。ルーティング・ガイドは、有効である期間と、適用時期の条件があります。これらの条件は、運賃条件 (Freight Terms) と部門に基づいています。ルーティング・ガイドは、Sterling Selling and Fulfillment Foundation 内で維持されています。

各ルーティング・ガイドには、ルーティング・ガイドライン のリストが含まれており、それぞれには配送業者、サービス、および出荷モードの選択についての詳細な条件が記述されています。次のような条件があります。

- 出荷の出荷地および宛先
- 任意の配送業者のサービス要請
- 重量や容積など、出荷自体の特性

ルーティング・ガイド情報は、VICS (Voluntary Inter Industry Commerce Standards) ルーティングにより使用されるデータに基づいています。

例えば、同じバイヤー組織内の異なる部門は、さまざまな配送サービスと別個の契約を結んでいることがあります。各部門で、配送業者の選択基準が含まれたルーティング・ガイドを提供できます。

Jolly Toy Company 社を例に挙げてみましょう。この会社には Stuffed Toys 部門と Board Games 部門の 2 つの部門があります。Stuffed Toys 部門はカリフォルニアにあり、その 1 か所ですべての配達を行っています。Board Games 部門の業務形態はまったく異なり、全米にある 7 つの異なるロケーションで出荷を受け入れることができます。

Jolly Toy Company 社は、これらの各部門にルーティング・ガイドを用意しています。

Stuffed Toys 部門用のルーティング・ガイドは非常にシンプルで、ほとんどの場合、この部門は優先配送業者 1 社を使用しています。出荷が非常に重い場合、Stuffed Toys 部門は異なる配送業者を使用します。

表 42. 例: Stuffed Toy 部門のルーティング・ガイド

条件	アクション
出荷の重量が 300 ポンドより軽い	配送業者サービス <i>Happy Toy Carriers</i> を使用してカリフォルニアのサンディエゴの施設へ出荷する
出荷の重量が 300 ポンドより重い	配送業者サービス <i>Big and Bulky Carrier</i> を使用してサンディエゴの施設へ出荷する

Board Games 部門のルーティング・ガイドは複雑です。この部門は、出荷地と宛先、出荷の重量、および配送業者の要求タイプ (翌日、陸路など) に基づいて一連の条件を設けています。出荷の時間とコストを最適化するために、幅広い柔軟な選択肢を使用しています。

表 43. 例: Board Game 部門のルーティング・ガイド

条件	アクション
出荷の重量が 5 ポンドより軽く、サービスは「翌日」	FedEx を使用する
出荷の重量が 10 ポンドより軽く、サービスは「翌々日」	UPS を使用する
出荷の重量が 10 ポンドより重く、CT、NY、または RI から NY または NJ への出荷	配送業者サービス <i>NY Special Freight</i> を使用して NJ の流通業者へ出荷する
出荷の重量が 10 ポンドより重く、CT、NY、または RI からカリフォルニア、コロラド、またはワシントンへの出荷	配送業者サービス <i>Cross Country Carrier</i> または <i>Big and Bulky Carrier</i> を使用して出荷先住所に出荷する
など.....	

Jolly Toy Company 社へ出荷する際に、エンタープライズ組織はこれらのルーティング・ガイドを参照します。出荷を担当する部門に応じて、適切な条件を評価し、適切なアクションを使用します。

ルーティング・ガイドの使用は任意です。バイヤーとエンタープライズのどちらもルーティング・ガイドを設定できます。バイヤーとエンタープライズの両方がルーティング・ガイドを設定している場合は、「運賃条件 (Freight Terms)」の設定を使用して、どちらのルーティング・ガイドを最初に使用するかが判断されます。

配送業者サービスを決定するためのルーティング・ガイドは複雑です。要求する配送業者タイプ、コスト最適化、配送業者に割り当てられている最も小さな優先順位番号など、指定されている基準に基づいて、満たすべき一連の条件があります。

表 44. 例: 配送業者サービスの決定

条件	アクション
出荷属性	危険なアイテムを出荷する場合、配送業者サービス UPSN Ground を使用してカリフォルニアのタスティンの施設に出荷する。
サービス料金	UPSN Ground の場合、サービス・コストは \$200。
優先順位	UPSN Ground の場合、割り当てられている優先順位は 2。

## 動的ルーティング

動的ルーティングは、外部リソースを使用して、出荷を集合・混載に割り当てる方法を決定します。動的ルーティングが使用される場合、ルーティング・ガイドは使用されません。

動的ルーティングの実行方法は、いくつかあります。1 つの方法として、輸送管理システム (TMS) を Sterling Selling and Fulfillment Foundation と統合する方法があります。統合については詳しくは、「*Sterling Selling and Fulfillment Foundation 統合ガイド*」を参照してください。輸送管理システムは、集合・混載のルーティングの詳細な計画を提供します。

動的ルーティングの 2 つ目の方法として、外部プログラムを実装する方法が挙げられます。この方法では、ルーティング計画について外部ソースを調べます。例えば、バイヤーが、固有のルーティング・システムを管理しており、そのシステムでは、提示された集合・混載出荷の日にはどの集合・混載をどの設備へ転送するかを考慮します。バイヤーは、そのときに、この集合・混載のルーティングを決定できます。外部プログラムを使用して、バイヤーに相談し、Sterling Selling and Fulfillment Foundation ルーティング処理を更新することができます。

動的ルーティング処理については詳しくは、292 ページの『出荷のルーティング』を参照してください。

## 配送計画

配送計画とは、1 つまたは複数の出荷地から 1 つ以上の宛先に 1 つ以上のオーダーを配達するために必要な一連の移動全体を指します。配送計画は、出荷、集合・混載、出荷地、停止場所、および出荷先で構成されます。配送計画の作成、つまり、手動での出荷のルーティングは、出荷管理コンソールを使用して行われます。コンソール・ユーザーは、出荷のルーティング方法を決定します。



## 入庫コンプライアンスと出庫制約

バイヤーおよびエンタープライズは両方とも、オーダーのアイテムを出荷する方法の要件および条件を持つことができます。

### バイヤーおよび入庫コンプライアンス

入庫コンプライアンス という用語は、集約および配達に関するバイヤーの条件をエンタープライズが遵守することを指します。

バイヤーは、バイヤーへの出荷時に満たさなければならない 必須条件を保有できます。例えば、バイヤーは異なるオーダー番号のオーダーを、分けて出荷することを要求できます。同様に、バイヤーは異なる部門番号のオーダーを、分けて出荷することを要求することもできます。

Sterling Selling and Fulfillment Foundation では、入庫コンプライアンスの条件は、参加者モデリングで設定されます。

入庫コンプライアンスに、以下の条件を設定できます。

表 45. 入庫コンプライアンスのカテゴリ

カテゴリ	説明
集約	アイテムを同じ出荷に入れることができるかどうかを記述します。条件には、異なる購入オーダー番号、異なるオーダー番号、異なる部門、異なる「マーク対象」、および異なるオーダー・タイプなどがあります。
ESP	経済的な出荷パラメーターを使用するかどうか、およびそれらのパラメーターの条件です。
運送会社プリファレンス	優先される運送会社のリストです。これらの運送会社プリファレンスは、可能な場合は優先する必要がありますが、条件に一致する使用可能な優先される運送会社がない場合は、別の運送会社を使用して出荷を行うことができます。
ルーティング・ガイド	配達のルーティングの条件を記述したガイドです。
梱包	アイテムを同じケースまたはパレットに混在させる場合の条件が記述されています。例えば、異なる測定単位 (UOM) のアイテムを同じパレットに混在させることができない、などです。  また、出荷容器マーキングを適用する時期に関する条件も示します。
流通加工サービス (Value Added Services)	付加価値サービスは、バイヤーの要件に合致する在庫を提供するために使用される、追加処理または取り扱いを定義します。詳しくは、227 ページの『第 10 章 付加価値サービス』を参照してください。

### エンタープライズと出庫制約

出庫制約 という用語は、集約と出荷のためのエンタープライズの固有の条件を表します。例えば、出荷コストを削減するために、エンタープライズは、一定の重量を満たすまで、できるだけ多くのアイテムを 1 つの出荷に入れることで、アイテムを

集約しようとしています。エンタープライズは、284 ページの『経済的な出荷パラメーターの使用』で説明する経済的な出荷パラメーターを設定して、この戦略を実現することができます。

Sterling Selling and Fulfillment Foundation では、分散オーダー管理構成コンソールの物流管理のセクションにアウトバウンド・コンプライアンスの条件を設定することができます。

これらのカテゴリについて出庫制約を設定できます。

表 46. 出庫制約カテゴリ

カテゴリ	説明
集約	アイテムを同じ出荷に入れることができるかどうかを記述します。条件には、異なる購入オーダー番号、異なるオーダー番号、異なる部門、異なる「マーク対象」、および異なるオーダー・タイプなどがあります。
ルーティング・ガイド	配達ルーティングの条件を記述したガイドです。
ESP	経済的な出荷パラメーターを使用するかどうか、およびそれらのパラメーターの条件です。

## 競合条件の解決

入庫コンプライアンス (バイヤーの条件) および出庫制約 (エンタープライズの条件) の両方を設定できます。両方を設定した場合、出荷集約またはルーティングを決定する際に競合条件が発生する可能性があります。

例えば、バイヤーとエンタープライズの両方が経済的な出荷パラメーターを設定しているとします。この場合、バイヤーは、アイテムを集約して出荷するためにエンタープライズよりも長い待ち時間を希望します。

表 47. バイヤーとエンタープライズの経済的な出荷パラメーター

ESP 属性	設定	
	バイヤー	セラー
重量のしきい値	300 キロ	150 キロ
容積のしきい値	40 立方メートル	20 立方メートル
保留日数	7	4

この競合を解決するには、プリファレンスを適用するオーダーを判断するための簡単な設定を使用します。分散オーダー管理で運賃条件 (Freight Terms) を設定する際に、次のようなオーダーを設定できます。

- 最初にバイヤー、次にエンタープライズ
- 最初にエンタープライズ、次にバイヤー

例えば、「最初にバイヤー、次にエンタープライズ」を選択した場合、バイヤーの設定が使用されます。以下のいずれかが発生するまで、出荷は集約されます。

- 重量のしきい値 300 キロを満たした
- 容積のしきい値 40 立方メートルを満たした
- 7 日が経過した

## 出荷

出荷作成のプロセスは、オーダー明細が出荷に向けてオーダー・リリースを作成してリリースされたときに開始され、出荷が配達されたときに完了します。このプロセスには、以下が含まれます。

- 出荷の作成
- 出荷のルーティング
- 出荷パイプライン
- 出荷コンソール

### 出荷の作成

オーダー明細が出荷用にリリースされると、オーダーの作成がリリースされます。以下の図は、使用されるロジックを示します。このロジックの詳細は、出荷パイプラインをカスタマイズすることによって変更できます。

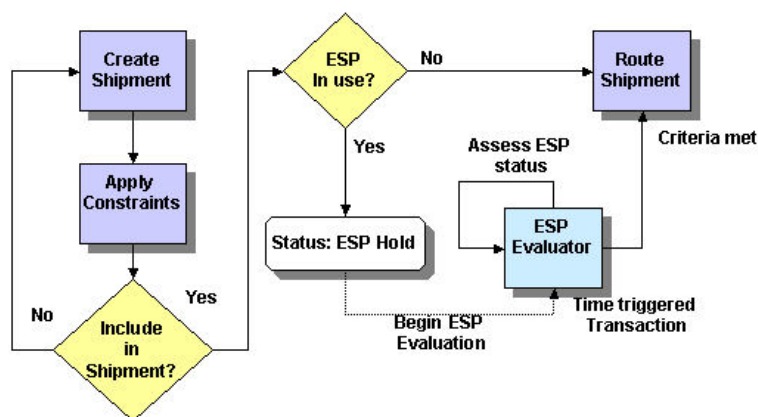


図 44. 出荷ロジックの作成

図 44 は、オーダー・リリース・アイテムを出荷に割り当てるためのプロセスを示します。

### 制約およびオーダー・リリースの作成

オーダー・リリースは、オーダー・リリース内のアイテムを確実に一緒に出荷できるようにするために、一連のルールを適用して作成されます。これらのルールには次のようなものがあります。

- 「混在させない」制約 (必須) - これらの制約は、Sterling Selling and Fulfillment Foundation システムの一部です。一緒に出荷できないアイテムの部分的なリストは、異なる出荷ノード、出荷先住所、運賃条件 (Freight Terms)、またはバイヤー組織などを持つアイテムを指します。完全なリストは、販売オーダー・ドキュメント・タイプに定義されます。
- 入庫コンプライアンス - 異なる顧客 PO 番号、オーダー番号、部門、またはマーケット対象を持つアイテムを一緒に出荷できないなど、オプションのバイヤー・ベースのルールです。

- 出庫コンプライアンス - 異なる顧客 PO 番号、オーダー番号、部門、またはマーク対象を持つアイテムを一緒に出荷できないなど、オプションのエンタープライズ・ベースのルールです。
- ルーティング・ベースの制約 - ルーティング・ガイドを使用している場合、そのガイドに基づいた制約です。

これらの制約を評価することによって、システムは一緒に出荷可能なオーダー・リリースを作成します。この評価は、出荷の作成処理で検証されます。

## 出荷の集約

Sterling Selling and Fulfillment Foundation では、IBM またはサード・パーティーの ERP、あるいはその両方からリリースされたオーダーを、出荷に集約することができます。出荷の集約は、出荷先、バイヤー、セラー、マーク対象、経済的な出荷パラメーターなど、さまざまなパラメーターに基づいて実行されます。

例えば、バイヤーは、オーダー明細が異なる部門に関連付けられている場合など、何らかの条件に基づいて、いくつかの出荷に分割しなければならないオーダーを頻繁に出す場合があります。ESP 処理が使用されている場合、同じ部門宛のアイテムは集約できます。ここでは、ESP 条件が適用され満たされた後に、ルーティング処理が開始されます。

## 出荷のルーティング

ルーティングとは、出荷に最適な配送業者とモードを割り当てることを指します。モードは、小包、小口貨物未満 (LTL)、および小口貨物 (TL) を指します。これは、出荷のルーティングが必要な場合に使用可能な選択肢で、配送業者を決定する際の、ノード、ビジネス・エンティティー (エンタープライズ)、または顧客に固有の条件の考慮などの要件が含まれます。出荷の配送業者とモードの選択は、基準に従って適切なガイドを使用すると同時に、出荷の宛先と重量を考慮してからにします。例えば、小包配送業者は、重量制限を設けていたり、特定の ZIP コード (郵便番号) には供給しなかったりすることが多くあります。

Sterling Warehouse Management System は、ノード、エンタープライズ、または顧客レベルでルーティング・ガイドを保持する機能をサポートしています。また、ルーティング・ガイドは、ガイドの有効日もキャプチャーします。業界では、ルーティング・ガイドはバイヤーによって 2、3 年ごとに変更されるのが一般的です。

外部のルーティング・ガイドまたは輸送管理システムとの統合機能も使用可能です。

Sterling Warehouse Management System の出荷には、事前割り当て済みの配送業者とサービスを設定することができ、その場合、ルーティング・ガイドは使用されません。出荷は、要求されるサービスを指定することができます。例えば、Best Parcel を指定すると、ルーティング・ガイドでも、選択した配送業者と出荷モードを最適化するために使用されます。

ルーティングされる出荷の要件は、出荷パイプラインで定義します。次の図は、使用されるロジックを示しています。

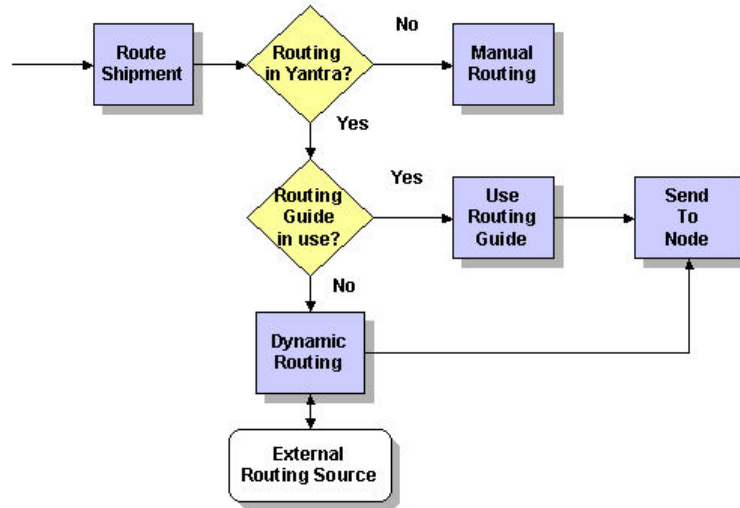


図 45. ルーティング処理の概要

図 45 は、ルーティングの概要を示しています。ルーティングを実行する方法には、次の 3 つがあります。

- ルーティング・ガイドの使用 - ルーティングは、バイヤーまたはエンタープライズのルーティング・ガイドに基づきます。
- 動的 - ルーティングは、輸送管理システムや、バイヤーとのカスタマイズされた対話など、外部リソースに基づきます。
- 手動ルーティング - 出荷ルーティングはコンソール・オペレーターによって設定されます。

## ルーティング・ガイドの使用

Sterling Warehouse Management System のルーティング・ガイドでは、要求サービス、重量、郵便番号などの条件に基づいて、最適の運送会社および出荷のモード (小包、LTL または TL) を指定できます。複数のルーティング・ガイドが使用される状況では、まず最初にバイヤーのルーティング・ガイドを検討して運送会社を決定するようにシステムを構成できます。バイヤーのルーティング・ガイドで運送会社を決定できない場合、システムは、次にエンタープライズのルーティング・ガイドを検討します。システムは、これに失敗すると、次にハブ (HUB) のルーティング・ガイドを検討します。

まず最初にセラーのルーティング・ガイドを検討して運送会社を決定するようにもシステムを構成できます。これに失敗したら、システムは、ハブ (Hub) のルーティング・ガイドを検討します。ただし、この場合、バイヤーのルーティング・ガイドは検討されません。

小包モードの出荷が選択された場合、出荷はルーティング済みにマークを付けられます。この出荷には、これ以上の処理は必要ありません。新しいオーダーを出荷にまとめることができます。この場合、この出荷に対して再ルーティングが行われません。

LTL 出荷の場合、出荷は、条件に基づいて既存の集合・混載に追加されるか、または新しい集合・混載が作成されます。集合・混載は、同じ配達先住所に出荷する出荷のグループです。すべての LTL 出荷は、出荷体積が TL に指定された体積を超えると、自動的に TL 出荷に変換されます。出荷が既存の LTL 集合・混載に追加されるたびに、新しい集合・混載に対して再ルーティングが実行されます。この出荷は、ルーティング済みにマークを付けられます。TL としてルーティングされた出荷には、さらなる集約の試行は行われません。この出荷は、ルーティング済みにマークを付けられます。

## ゾーン・スキップ

通常、運送会社は、いくつかの地理的地域をゾーンにグループ化し、これらのゾーン間の出荷の移動に対する出荷料金を指定します。運送会社は、ゾーン内の現地出荷の配達元である各ゾーンにバルク統合および仕分けを行うセンターを持っています。

ゾーン間の出荷を行うとき、倉庫は、宛先ゾーンの運送会社に LTL または TL として出荷する方が利益が大きくなる場合があります。これらの出荷は、後にゾーン内で小包で出荷されます。このため、倉庫は、配達元と配達先のゾーンの間のいくつかのゾーンをスキップします。

Sterling Warehouse Management System のゾーン・スキップ機能では、倉庫は、個別の出荷を LTL または TL の集合・混載にまとめ、これらを地域に出荷し、その地域の運送会社のバルク仕分けノード (break bulk node) に送ることによって、輸送コストを削減できます。このまとめられた集合・混載は、バルク仕分けロードと呼ばれます。バルク仕分けノード (break bulk node) に到達したら、バルク仕分けロードは分解され、個々の出荷が運送会社および運送会社サービスに基づいて宛先に出荷されます。

### ゾーン・スキップでのルーティング

ゾーン・スキップ中は、出荷は次のようにルーティングされます。

- 出荷ノードからバルク仕分けノード (break bulk node) へ
- バルク仕分けノード (break bulk node) から宛先ノードへ

出荷の SCAC および配送業者サービスは、ルーティング・ガイドに従ってセットアップされます。出荷ノードからバルク仕分けノード (break bulk node) への間にバルク仕分け集合・混載が存在するかどうかシステムによって判断されます。バルク仕分け集合・混載が存在する場合、出荷はそこに追加されます。そのような集合・混載が存在しない場合は、次の条件を満たすその他の出荷があるかどうかシステムによって確認されます。

- ルーティング・ソースが Sterling Warehouse Management System である。
- Sterling Warehouse Management System にルーティング・ガイドが維持されている。
- 出荷の配送業者タイプが小包である。
- 出荷が積荷目録記載されていない。
- 運賃が出荷者払いである。

上記の基準に一致する出荷が見つかった場合、システムは、すべての出荷の集約リストを準備し、この集約出荷が、そのバルク仕分けノード (break bulk node) に指定されている重量または容積の最小基準を満たすかどうかを判断します。システムによって集合・混載が作成され、これらの出荷が集合・混載に追加されます。この集合・混載の出荷地には出荷ノードのマークが付けられ、宛先にはバルク仕分けノード (break bulk node) のマークが付けられます。この集合・混載はルーティングされ、バルク仕分け集合・混載のマークが付けられます。

**注:** バルク仕分け集合・混載に属する出荷に加えられた変更は、集合・混載に伝搬されます。バルク仕分け集合・混載が削除された場合、そこに属している出荷は自動的に再ルーティングされます。

次の条件を満たす場合、バルク仕分け集合・混載は削除されます。

- 集合・混載に属する出荷のいずれも積荷目録記載されていない。
- 重量および容積の最小基準を満たしていない。

### 出荷の出荷ゾーンの積荷目録作成

ルーティングの一環として、Sterling Warehouse Management System では、次のゾーン・スキッピングの条件を満たす小包の出荷が考慮されます。

- 出荷にバルク仕分けノード (break bulk node) のスタンプがある。
- 出荷がどの積荷目録にも含まれていない。
- Sterling Warehouse Management System で出荷のルーティング・ガイドが管理されている。
- Sterling Warehouse Management System によってルーティング・ソースが割り当てられている。

ゾーン・スキッピングのしきい値の重量および体積に適合する場合は、それらの出荷をバルク仕分けノード (break bulk node) に運送するための LTL または TL 集合・混載がシステムで作成されます。バルク仕分けの集合・混載に属する出荷は、自動的にバルク仕分けノード (break bulk node) から積荷目録に記載されます。

**注:** バルク仕分けノード (break bulk node) は、非 WMS 出荷ノードとして構成する必要があります。また、バルク仕分けノード (break bulk node) の積荷目録に記載されたすべての出荷が (バルク仕分けの集合・混載構成部分として) バルク仕分けノード (break bulk node) に同時に到着することを保証するため、出荷ノードのそれぞれに別個のバルク仕分けノード (break bulk node) を構成する必要があります。積荷目録の完了は、その他の出荷ノードの集合・混載の到着には依存しません。

システムでは、次の方法で 1 つのバルク仕分けの集合・混載に属するすべての出荷が自動的に積荷目録に記載されます。

- 出荷レベルの統合 — 出荷の梱包プロセス完了時に特定の出荷のすべての容器が積荷目録に記載されます。
- パッケージ・レベルの統合 — 出荷の梱包プロセスの完了時に容器のそれぞれが積荷目録に記載されます。

ユーザーが積荷目録に記載された容器を開梱すると、その出荷は自動的に積荷目録から削除されます。FedEx との出荷レベル統合が設定された出荷については、その出荷の容器のすべてが積荷目録に記載されません。

バルク仕分けノード (break bulk node) の未完了の積荷目録は、集合・混載が確認されると完了します。バルク仕分けの集合・混載に属する出荷は、集合・混載が出荷地ノードから出荷されたときに確認されます。

## 小包出荷用の 2 番目の区間の運送会社の決定

Sterling Warehouse Management System は、バルク仕分けノードから最終宛先までの出荷 (バルク仕分けの集合・混載の一部) のルーティングを行う機能を提供します。これを実現するためには、ユーザーは、バルク仕分けノードから最終宛先までのルーティング・ガイドラインを構成する必要があります。

小包の出荷に対する 2 番目の区間のルーティングが失敗した場合は、以下のようになります。

- 国内出荷 — 出荷に、最初の SCAC および運送会社サービス・コードが刻印されます。
- 海外向け出荷 — 出荷に、「2 番目の区間の運送会社を決定できませんでした」ルーティング・エラー・コードが刻印されます。それにも関わらず、出荷は「出荷ルーティング済み」ステータスに変わります。このエラーは、手動で解決する必要があります。

## 出荷パイプライン

出荷の実際の処理は、一連のトランザクションとステータスを経て完了に至ります。このトランザクションと出荷ステータスの連鎖は、出荷パイプラインと呼ばれます。

出荷パイプラインは、Sterling Selling and Fulfillment Foundation に、パイプライン内の出荷のロケーションに応じた通知の送信や警告のロギングなどのアクションを実行する手段を提供します。また、作成から配達まで、出荷を追跡する手段も提供されます。

各企業のビジネスの起点となる出荷パイプラインは、システム管理者によるビジネス環境のセットアップ方法に応じて固有のものとなります。ただし、一般に出荷パイプラインはすべて、出荷を作成するトランザクションによって開始され、オーダーの配達完了したことを示すトランザクションで終了します。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。出荷ステータスは、出荷の現在状況を記述し、トランザクションごとに移動します。

## 出荷パイプラインのステータス

Sterling Selling and Fulfillment Foundation のデフォルトの出荷パイプラインにおける出荷のフロー・ステータスは、次のとおりです。

- 出荷作成済み - 出荷は、オーダー・リリースまたは連鎖オーダーから作成されました。
- ピック中の出荷 - 出荷の準備のため、明細アイテムが物理的にピックアップされています。
- 出荷完了 - 出荷は出荷先住所に出荷されました。
- 出荷配達済み - 出荷は出荷ノード住所へ配達されました。



- 受入に組み込み済み - 出荷は受入に組み込まれました。
- 受入終了済み - 出荷は受入が終了し、完了したと見なされます。
- 出荷請求済み - 出荷に対してインボイスが作成されました。
- ESP チェックが必要 - ESP エバリュエーターを実行して、ESP 条件に一致しているかどうかを判定する必要があります。
- ESP 保留時 - ESP 条件に合致するまで、出荷が保留されます。
- ESP 保留解除 - ESP 条件に合致し、ESP 保留が解除されます。
- 出荷ルーティング済み - 出荷が集合・混載に割り当てられています。
- ノードに送信 - 梱包に向けて出荷が送信されています。
- 出荷キャンセル済み - 出荷がキャンセルされています。

次の図は、デフォルトの出荷パイプラインを Application Manager に表示されるとおりにグラフィックで示しています。

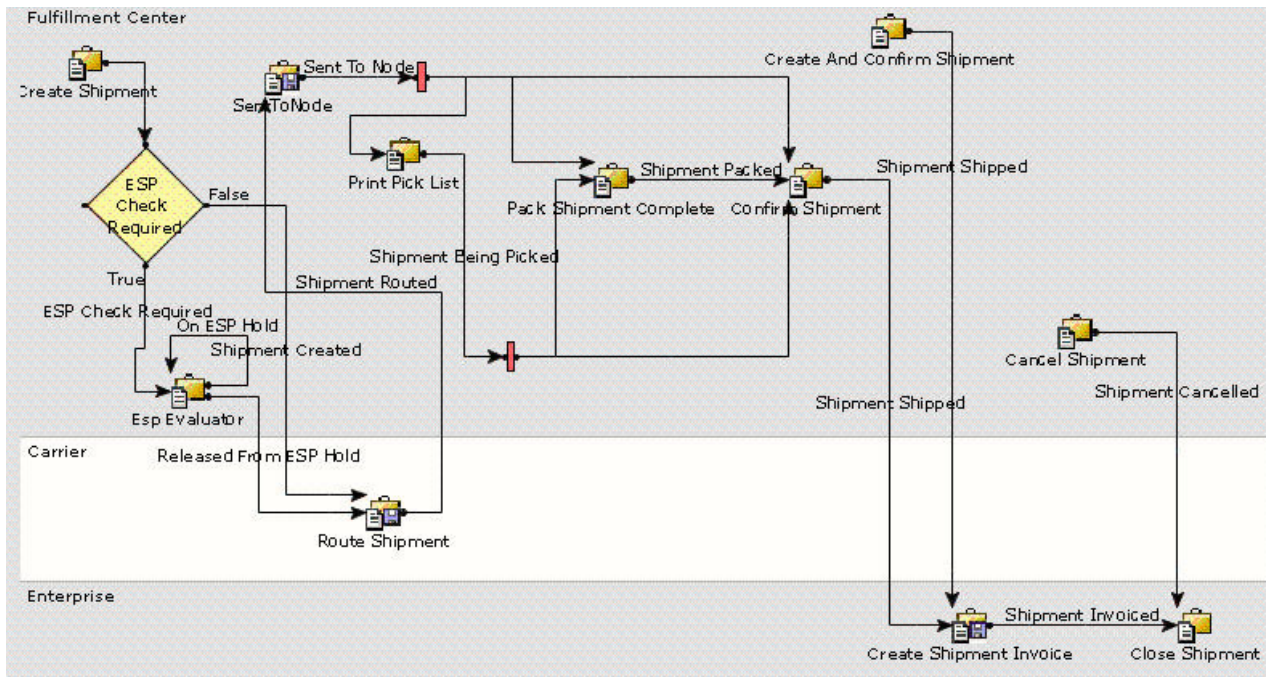


図 46. 出荷パイプライン

## 出荷コンソール

出荷コンソールは、販売オーダーから実行される出荷および出荷容器に関する情報を提供します。このコンソールは、適正なオーダーの定刻出荷を確実にするために必要な情報を、セラー、ノード・ユーザー、および運送会社に提供します。

オーダー・リリースが梱包され、出荷ノードによる出荷の準備が物理的に完了したときに、オーダーが出荷になります。出荷には、出荷先住所、寸法、出荷料金、運送会社情報、および出荷される明細の情報の詳細が記載されます。

エンティティとしての出荷は、集合・混載に添付することによって、配送計画で使用できます。配送計画と集合・混載のセットアップについて詳しくは、*Sterling Selling and Fulfillment Foundation: Logistics Management User Guide*を参照してください。

## バンドル出荷の作成

バンドルのオーダー情報が使用可能な場合、バンドルについての出荷の作成は、バンドル親が出荷に追加されるのか、バンドルのコンポーネントが出荷に追加されるのかに応じて発生します。

バンドル親が出荷に追加される場合は、バンドルのコンポーネントが出荷に自動的に追加されることはありません。

バンドル・コンポーネントが出荷に追加されるか、またはコンポーネントについて出荷が作成される場合は、その親明細およびすべての n レベル明細が以前に追加されていないければ、それらも出荷に追加されます。

バンドルのオーダー情報が使用可能でない場合は、バンドルのコンポーネントはカタログから取られます。

## バンドルの出荷の確認

バンドル内の最後のコンポーネントを出荷すると、親明細のステータスは、パイプラインのセットアップに基づいて、「出荷済み」またはその拡張ステータスに変更されます。

バンドル・コンポーネントの比率が管理されていない場合、バンドル全体が出荷されたときにのみ、オーダー明細のステータスは更新されます。

バンドル親の場合、出荷数量が出荷レベルでのコンポーネント数量の最大数量でスタンプされていることを、確認します。ただし、オーダー明細上には、出荷された親の正確な数量が表示されます。

## バンドル出荷のインボイス作成

エンタープライズ・レベル・ルール「バンドル親明細のインボイスを作成する時期」は、バンドルのインボイスをどのように作成するかを決定します。このルールには、2 つの値、First と Last があります。ルールが First に設定された場合は、そのコンポーネントの 1 つが出荷されるときにいつでも、バンドルのインボイスが作成されます。ルールが Last に設定された場合は、そのコンポーネントのすべてが出荷されたときにのみ、バンドルのインボイスが作成されます。

『顧客が 2 つのベッド・セット・バンドルをオーダー』では、このエンタープライズ・レベル・ルールに基づいてバンドルのインボイスがどのように作成されるかについて説明します。

### 顧客が 2 つのベッド・セット・バンドルをオーダー

この例では、顧客は、2 つのベッド・セットをオーダー済みです。ベッド・セットは、以下のもので構成されるバンドルです。

- ベッド・セット (親バンドル)

- マットレス
- ベッド・フレーム

表 48 は、エンタープライズ・レベル・ルール「バンドル親明細のインボイスを作成する時期」が値 First または Last に設定されている場合に、バンドルのインボイスがどのように作成されるかを示します。

表 48. バンドル出荷のインボイス作成

出荷済み日付	出荷済みの内容	親バンドルに対して請求済みの金額	
		First	Last
08/01	1 ベッド・セット	1	1
08/05	2 マットレス	2	0
08/10	2 ベッド・フレーム	0	2
08/20	2 ベッド・フレーム	2	2

## 集合・混載

集合・混載では、2 地点間の 1 つ以上の完全な出荷 (部分的出荷は対象外) が配送されます。集合・混載は、出荷のルーティングの結果として作成されます。詳しくは、292 ページの『出荷のルーティング』を参照してください。

### 集合・混載の作成

集合・混載の作成は、ルーティング・プロセスの一部として行われます。ルーティング・プロセス中に、出荷を既存の集合・混載に追加したり、あるいは新規の集合・混載を作成したりすることができます。出荷の行き先であるロケーションへ現在出荷中の集合・混載がないか、あるいは現在の集合・混載は出荷を収容することができないために、新規集合・混載が作成されることがあります。

詳しくは、292 ページの『出荷のルーティング』を参照してください。

### 集合・混載実行パイプライン

集合・混載ドキュメントのフローは作成から開始され、一連のトランザクションとステータスを経て完了に至ります。このトランザクションと出荷ステータスのチェーンは、集合・混載実行パイプラインと呼ばれます。

Sterling Selling and Fulfillment Foundation は、集合・混載パイプラインによって、そのパイプラインにおける集合・混載のロケーションに応じた通知の送信や警告のロギングなどのアクションを実行する手段を得ることができます。また、作成から配達まで、集合・混載を追跡する手段も提供されます。

各企業のビジネスの起点となる集合・混載実行パイプラインは、システム管理者によるビジネス環境のセットアップ方法に応じて固有のものになります。ただし、一般に集合・混載実行パイプラインはすべて、集合・混載を作成するトランザクションによって開始され、集合・混載が割り当てられた宛先に到着したことを示すトランザクションによって終了します。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。集合・混載ステータスは、集合・混載の現在状況を記述し、トランザクションごとに移動します。

Sterling Selling and Fulfillment Foundation のデフォルトの集合・混載実行パイプラインにおける集合・混載のフロー・ステータスは次のとおりです。

- 集合・混載作成済み
- トレーラー積み付け完了
- 輸送中の集合・混載
- 完了した集合・混載

次の図は、デフォルトの集合・混載実行パイプラインを Applications Manager に表示されるとおりにグラフィックで示しています。

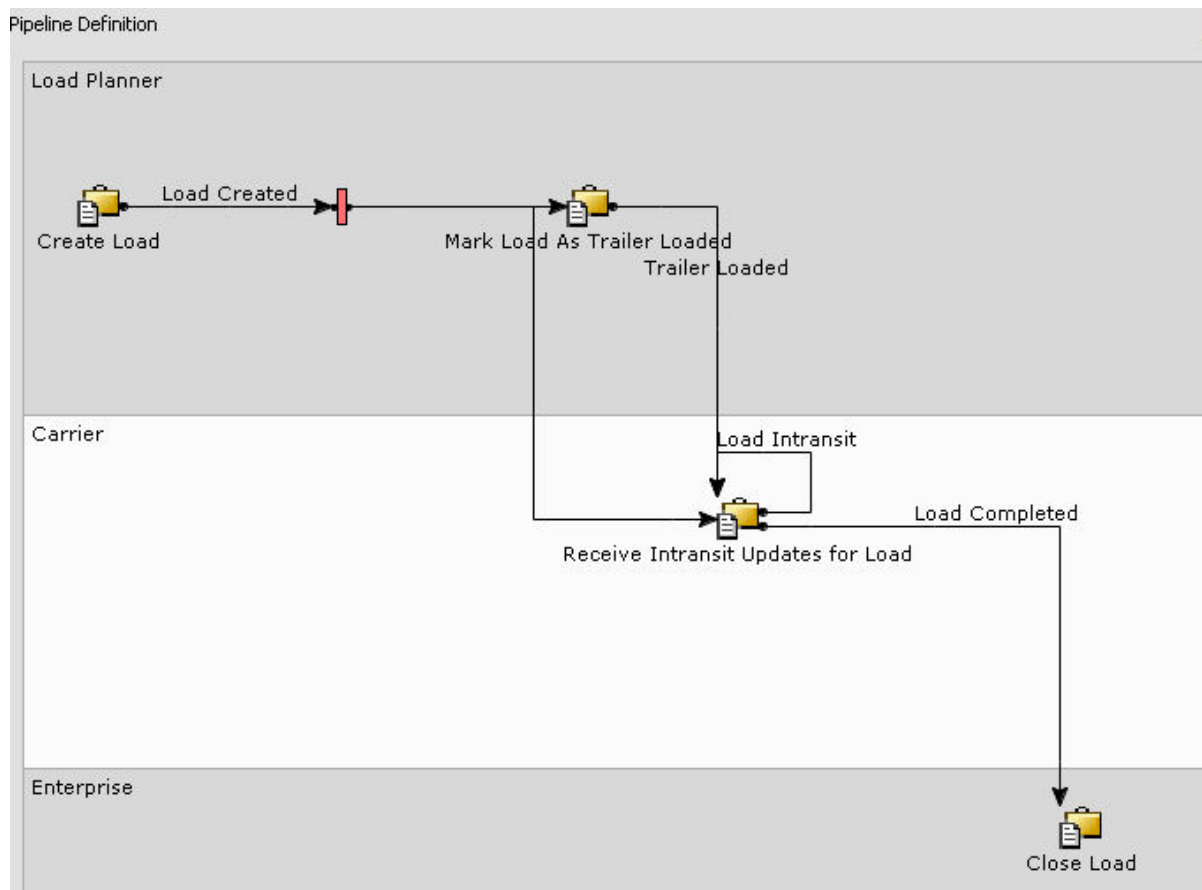


図 47. 集合・混載実行パイプライン

## 入荷

オーダーは、購入オーダー・リリースが梱包された時点で入荷になり、物理的に出荷ノードによる出荷準備が整います。

## 入荷パイプライン

入荷は、その作成から始まり、一連のトランザクションおよびステータスを経て、完了します。このトランザクションと入荷ステータスの連鎖を、入荷パイプラインと呼びます。

入荷パイプラインは、そのパイプライン内での入荷のロケーションに応じて、通知の送付または警告のロギングなどのアクションを実行するための手段を **Sterling Selling and Fulfillment Foundation** に提供します。また、作成から配達まで入荷を追跡するための手段も提供します。

ビジネス運営の基礎となる入荷パイプラインは、システム管理者がビジネス環境をセットアップした方法に特有のものです。しかし、各入荷パイプラインは一般に、入荷を作成するトランザクションで始まり、オーダーが配達されたことを示すトランザクションで終わります。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。入荷ステータスは、入荷がどのような状態であるかを説明し、トランザクション間で入荷を移動させます。

入荷が **Sterling Selling and Fulfillment Foundation** のデフォルト入荷パイプライン内で通過する可能性があるステータスは、以下のとおりです。

- 出荷作成済み - 出荷は、オーダー・リリースまたは連鎖オーダーから作成されました。
- ピック中の出荷 - 出荷の準備のため、明細アイテムが物理的にピックアップされています。
- 出荷完了 - 出荷は出荷先住所に出荷されました。
- 出荷配達済み - 出荷は出荷ノード住所へ配達されました。
- 受入に組み込み済み - 出荷は受入に組み込まれました。
- 出荷請求済み - 出荷に対してインボイスが作成されました。

## 入荷コンソール

入荷コンソールには、購入オーダーから作成される入荷および入荷容器に関する情報が表示されます。このコンソールでは、正しいオーダーの定刻出荷を確実にするために必要な情報が、セラー、ノード・ユーザー、および運送会社に提供されます。

オーダーは、購入オーダー・リリースが梱包された時点で入荷になり、物理的に出荷ノードによる出荷準備が整います。入荷は、出荷先住所、寸法、入荷料金、運送会社情報、および出荷される明細に関する情報の詳細を示します。

エンティティーとしての入荷は、集合・混載に追加することにより、配送計画で使用できます。配送計画および集合・混載の設定について詳しくは、*Sterling Selling and Fulfillment Foundation: Logistics Management Configuration Guide* を参照してください。

## 保留処理

Sterling Selling and Fulfillment Foundation では、出荷または集合・混載を複数の理由で保留することができます。特定のタイプの保留では、その保留タイプと関連付けられたトランザクションで出荷または集合・混載を処理できません。特定のタイプの保留では、特定の変更も出荷または集合・混載に適用されません。エンタープライズが、住所が変更になるたびに住所確認を実行したり、配達日が変更になるたびに日付確認を実行しようとする場合は、それらの確認が完了するまで、出荷または集合・混載が処理されないようにする必要があります。

保留には、次の 3 つの異なるステータスがあります。

- 作成済み。保留が出荷または集合・混載に適用されたばかりで、まだチェックを受けていない場合です。
- 拒否済み。保留は、出荷または集合・混載が処理されないように決定した管理者によって精査されています。
- 解決済み。保留は、出荷または集合・混載が処理されるように決定した管理者によって精査されています。

保留タイプは、以下のように適用できます。

- 手動: 管理者は、出荷または集合・混載を不正チェックのために保留中にする必要があると考える場合があり、アプリケーション・コンソールを通じて保留を適用します。1 つの特定のグループまたは一連のグループのユーザーのみが保留を出荷または集合・混載に適用できるように、特殊な保留タイプを構成することが可能です。
- 出荷または集合・混載の作成時に自動的に: デフォルトでは、出荷または集合・混載が作成されるたびに、特定の保留タイプにセットされます。
- 別の保留タイプの解決時に自動適用: 特定の保留タイプが解決された場合に、別の保留タイプが自動的に起動されます。これは、他の保留タイプの解決時に適用される保留タイプに指定されます。
- 特定の変更タイプの発生時に自動適用: 例えば、集合・混載にテロリスト組織のための爆弾またはドラッグが含まれている可能性があるという内密の情報がある場合のテロリスト脅威保留など、出荷または集合・混載レベルで特定の変更タイプが発生したときに保留タイプが自動的にセットされるように構成することができます。

保留がどのように適用されるかに関係なく、出荷または集合・混載を保留にするかどうかを決定する条件を指定することができます。例えば、特定の配達住所を持つ出荷のみを保留にしたい場合があります。保留タイプに対してこれを行うには、条件ビルダーを使用できます。条件ビルダーの使用については、「*Sterling Selling and Fulfillment Foundation* アプリケーション・プラットフォーム構成ガイド」を参照してください。

デフォルトでは、すべてのトランザクションが、保留中の出荷または集合・混載を処理することができます。Applications Manager を使用すると、特定の保留タイプになっている出荷または集合・混載を処理しないようにするトランザクションを指定することができます。保留中の出荷または集合・混載を処理しないように構成できるトランザクションは、保留タイプ対応トランザクションと呼ばれ、指定のエンタ

ープライズに対してドキュメント・タイプ・レベルで定義されます。抽象化トランザクションから派生していないカスタム・トランザクションは、全保留タイプ対応です。

1 つの保留タイプにある出荷または集合・混載を処理し、その保留からそれらを削除するように、2 つのトランザクションを構成する必要があります。1 つは作成済みステータスでの保留用で、もう 1 つは拒否済みステータスでの保留用です。さらに、保留は手動で解決することが可能であり、権限を持つ一連のユーザー・グループがそれを行うように指定することができます。それも、Applications Manager で、指定のエンタープライズのドキュメント・タイプ・レベルで行われます。





---

## 第 14 章 返品物流

返品物流とは、サプライ・チェーンを使用して、消費者または小売店から製造業者への製品の返品を取り扱うプロセスのことです。返品物流には、パッケージや容器のリサイクル、製品の修理や整備、および旧商品の処分も含まれます。

返品物流は多くの会社にとって、最も注目されていない分野の 1 つです。しかし、サプライ・チェーンでアセットと収入を回復できるため、この分野は改めて注目を集めつつあります。現在、何がどのような理由で返品されるかは、十分に理解されていません。返品物流における戦略は、取り扱う SKU タイプによって異なります。例えば、消耗品のオンライン小売店と電気製品の小売店では異なります。

Sterling Warehouse Management System Reverse Logistics は、小売店、製造業者、サード・パーティー返品物流プロバイダーをはじめとするあらゆるタイプの会社における返品物流の「エンドツーエンド」プロセスを管理するための包括的な機能を備えています。返品物流プロセスは、返品承認の作成、製品の処分、返品された在庫の再在庫、ベンダーへの返品、廃棄など、あらゆる側面を網羅します。

Sterling Warehouse Management System では、返品を処分とともに個々の SKU に分解します。返品部門での処分が可視化されるため、それぞれのエリアへの在庫の集約とルーティングが可能になります。Sterling Warehouse Management System でのルーティングは、複数のステップで構成されるプロセスで、在庫の再在庫前における整備や再パッケージの要件が組み込まれています。追加で記録されるフィールド在庫属性には、理由コード、ロット番号、改訂番号、シリアル番号などがあります。

このような情報の記録によって提供されるきめ細かな可視性のおかげで、受入時から、アイテムが適切なエリアに格納される前まで、複数ステップの処分プロセスを通じて、在庫の追跡が可能です。細かなレベルで実行されたアクティビティを追跡するために監査証跡も提供されており、返品番号、アイテム、理由コード、処分、在庫属性、およびユーザーが含まれています。

多くの会社は、返品および修理プロセスに関連する在庫の金額を認識していません。価値の高い製品や部品が、修理のために発送されたり、在庫に戻されたりする前に、何週間もの間、技術者のトラックや検査センターに放置されている場合があります。多くの場合、このような部品には修理の必要はなく、すぐに別の顧客に再販売できます。Sterling Warehouse Management System の Reverse Logistics を使用することで、会社は返品および修理プロセス全体でアイテムを効果的に追跡して、アイテムを在庫に戻す手順を自動化できます。ステータスはサービス組織および修理組織からリアルタイムで更新されるため、会社では、返品物流サイクルを供給ソースとして真に活用することができます。

---

### 返品物流とは

返品物流には、製品の返品処理の管理が含まれます。返品オーダーを表示して、返品アイテムの検査、返品アイテムの処理方法の決定、パイヤーへの返品クレジットの必要額の提供を実行できます

---

## 返品オーダーの構成要素

Sterling Selling and Fulfillment Foundation の返品オーダーは、返品オーダー・ヘッダー・レベル、返品オーダー明細レベル、および返品オーダー・リリース・レベルの 3 つのレベルに分けることができます。返品オーダー・ヘッダー・レベルには、返品された返品オーダー明細のすべてが記載され、返品オーダー明細レベルは、返品された個別の明細のそれぞれに分けられ、返品オーダー・リリース・レベルには、返品ノードにリリースされたすべての明細が記載されます。

---

## 返品物流パイプライン

作成から完了に至るまで、返品オーダーは一連のトランザクションとステータスを経由します。このトランザクション・チェーンと返品オーダー・ステータスを返品物流パイプラインと呼びます。パイプラインには、標準の返品物流パイプライン、および顧客から小売サイトへの返品を取り扱う際に使用される消費者返品パイプラインという 2 つのデフォルトのパイプラインがあります。

返品物流パイプラインは、パイプラインでの返品オーダーのロケーションに応じて、通知の送信、警告のロギングなどのアクションを実行するための方法を Sterling Selling and Fulfillment Foundation に提供します。さらに、返品オーダーの作成から完了までを追跡する方法、および必要に応じて手操作による介入を実行するための方法も提供します。

ビジネスが実行される元の返品物流パイプラインは、システム管理者がビジネス環境をどのようにセットアップしているかによって固有です。ただし、すべての返品物流パイプラインは、返品オーダーを作成するトランザクションから始まり、返品オーダーがクローズされたことを示すトランザクションで終了するのが一般的です。

トランザクションは、追跡される必要がある出来事で、実行する特定のアクションを呼び出すことができます。返品オーダー・ステータスは返品オーダーがどのような状態にあるかを説明し、トランザクションごとに移り変わります。

## 返品物流パイプラインのステータス

返品物流パイプラインでは、システム内での構成内容に応じて、次のステータスが使用される場合があります。

- 承認済み - 返品は、ビジネス慣習に従って該当者によって承認済みです。
- 交換オーダー作成の待機中 - オーダーは返品済みで、交換オーダー (exchange order) を作成する必要があります。
- キャンセル - 返品はキャンセル済みです。
- 作成済み - 返品は作成済みです。
- ドラフト作成済み - 返品の実成コンソールでドラフト返品が通知なし返品 (blind return) 用に作成済みです。この返品は、確認済みになるまでは、すべての側面を変更できます。
- 交換オーダー作成済み - 交換オーダー (exchange order) が作成済みです。
- 保留 - なんらかの理由により購入オーダーが保留されています。保留からリリースされるまで購入オーダーに変更を加えることはできません。

- 出荷に含まれている - 返品は出荷に含まれています。
- 検査済み - 返品は返品ノードにより検査済みです。
- コンポーネントとして検査済み (Inspected As Components) - 1 つ以上の個々のコンポーネントが検査済みです。
- 未承認 - 返品は、ビジネス慣習に従って該当者によって承認されていません。
- 未リリース - 返品は返品ノードへリリースされていません。
- 受入済み - 返品は返品ノードによって受入済みです。
- コンポーネントとして受入済み - 返品は 1 つ以上の個々のコンポーネントとして受入済みです。
- 受入終了 - 必要な返品取り扱いが完了し、返品はクローズされています。
- リリース済み - 返品オーダーの伝票は該当する返品ノードへリリース済みです。
- リリースから削除済み - 返品に含まれる 1 つ以上のアイテムがリリースから削除されました。
- 請求済み返品 - 返品のインボイスが作成済みです。
- 出荷済み - 返品は出荷済みです。
- 不足 - 元々オーダーされたものよりも少ない数量が返品に含まれています。返品はクローズされます。
- 受入取り消し済み - 返品は返品ノードによって受け入れられていません。

## デフォルト・パイプライン

以下の図は、Applications Manager で表示されるデフォルト・パイプラインを示します。

- 返品物流パイプライン
- 消費者返品パイプライン
- 返品物流サービス・パイプライン
- バンドル親の返品物流パイプライン

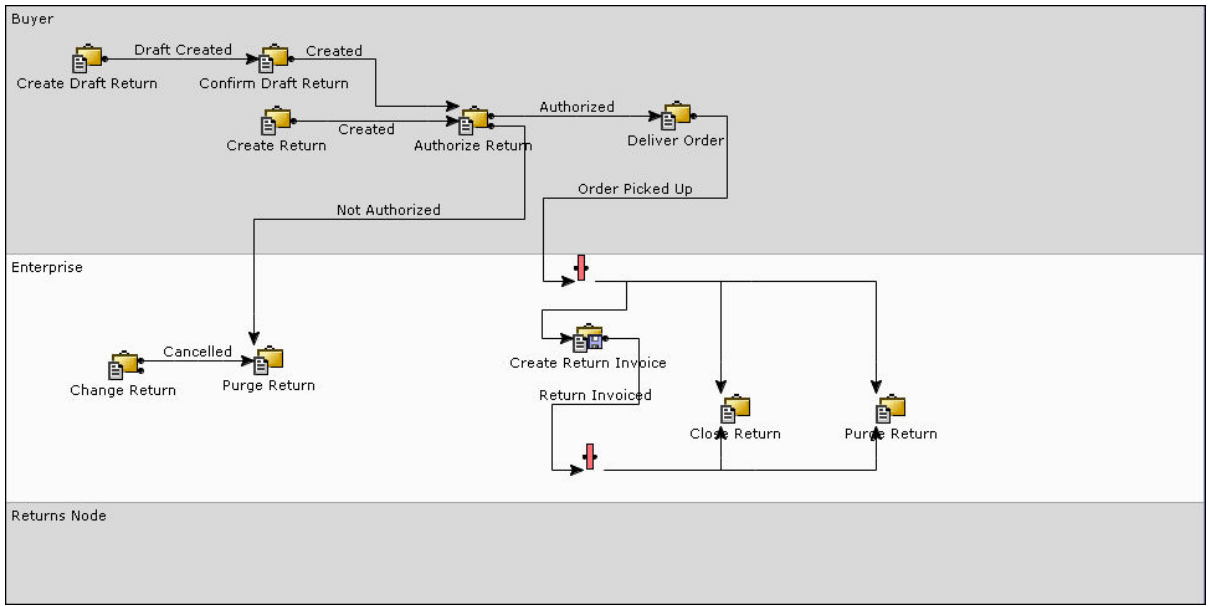


図 48. 返品物流パイプライン

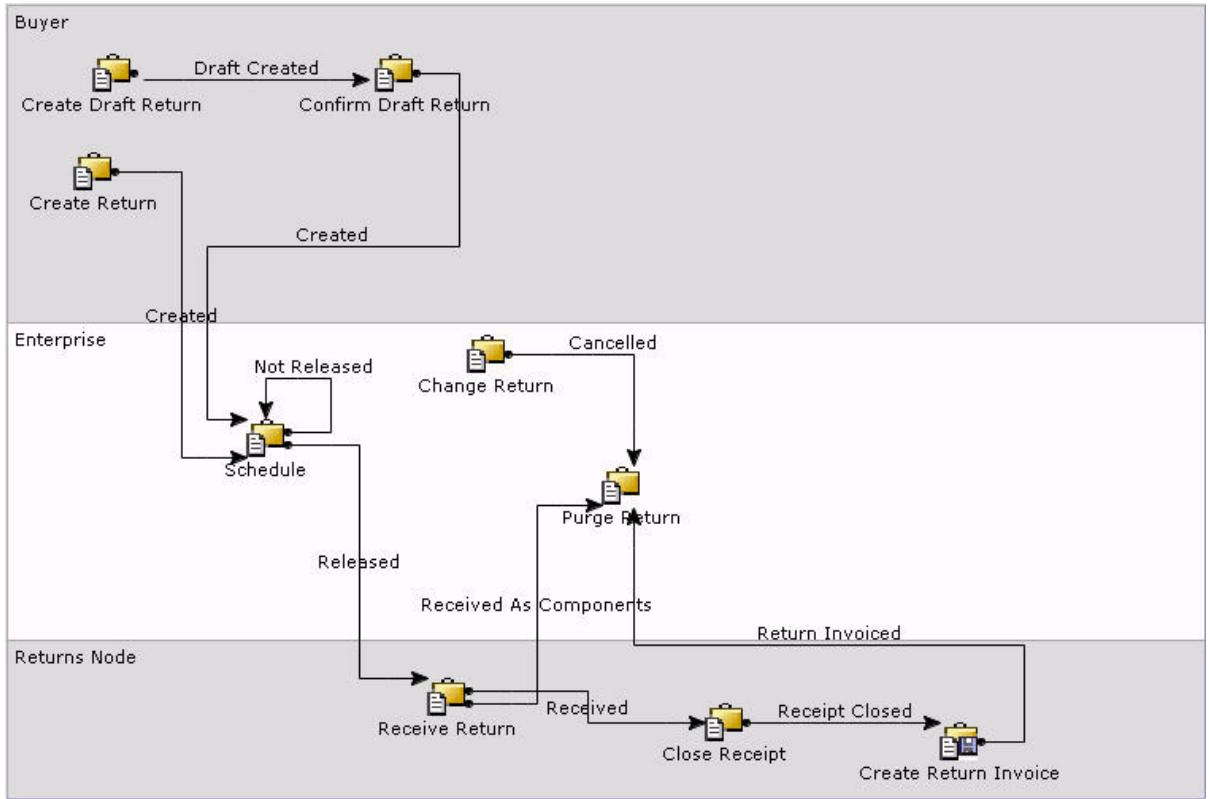


図 49. 消費者返品パイプライン

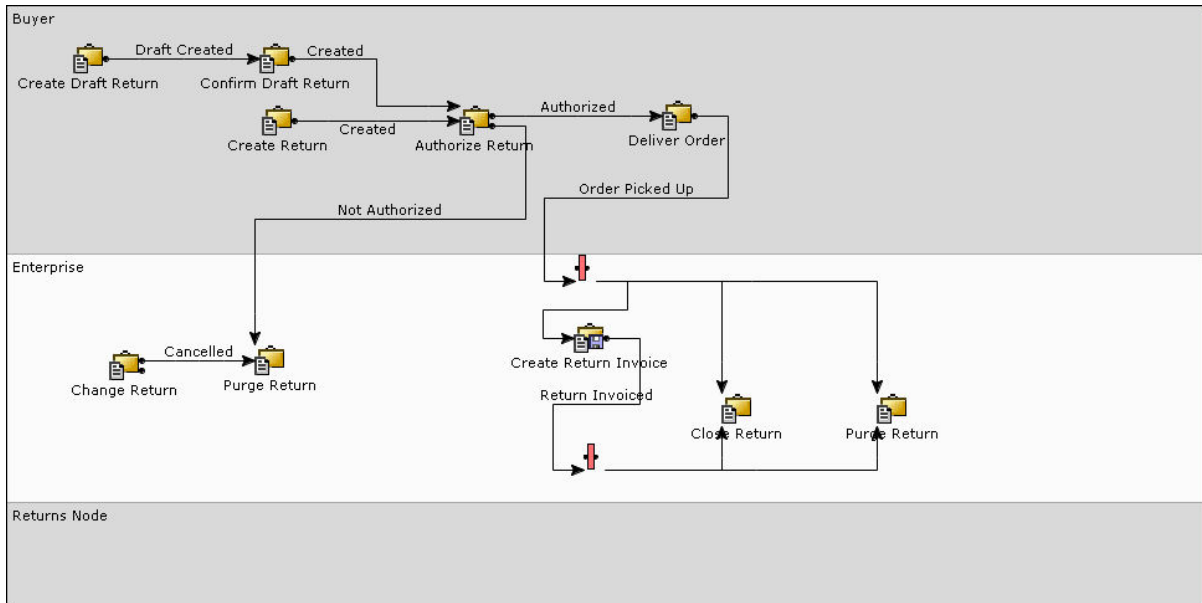


図 50. 返品物流サービス・パイプライン

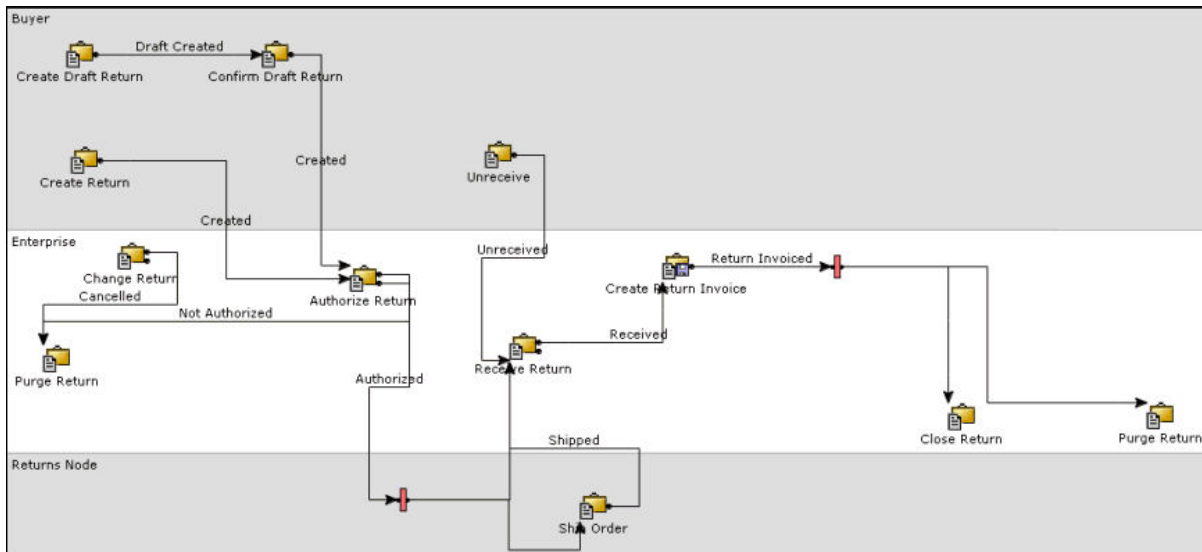


図 51. バンドル親の返品物流パイプライン



---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、**IBM** 所定のプログラム契約の契約条項、**IBM** プログラムのご使用条件、またはそれと同等の条項に基づいて、**IBM** より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

**IBM** 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。**IBM** は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。**IBM** 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

**IBM** の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている **IBM** の価格は **IBM** が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:



本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェース (application programming interface) に準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© IBM 2012。このコードの一部は、IBM Corp. のサンプル・プログラムの派生物です。© Copyright IBM Corp. 2012。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com)<sup>®</sup> は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、および PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は、英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

Intel、Intel (ロゴ)、Intel Inside、Intel Inside (ロゴ)、Intel Centrino、Intel Centrino (ロゴ)、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Cell Broadband Engine, Cell/B.E は、米国およびその他の国における Sony Computer Entertainment, Inc. の商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open, LTO, LTO ロゴ、Ultrium および Ultrium ロゴは、米国およびその他の国における HP、IBM Corp. および Quantum の商標です。

Connect Control Center<sup>®</sup>、Connect:Direct<sup>®</sup>、Connect:Enterprise<sup>™</sup>、Gentran<sup>®</sup>、Gentran<sup>®</sup>:Basic<sup>®</sup>、Gentran:Control<sup>®</sup>、Gentran:Director<sup>®</sup>、Gentran:Plus<sup>®</sup>、Gentran:Realtime<sup>®</sup>、Gentran:Server<sup>®</sup>、Gentran:Viewpoint<sup>®</sup>、Sterling Commerce<sup>™</sup>、Sterling Information Broker<sup>®</sup>、および Sterling Integrator<sup>®</sup> は、Sterling Commerce<sup>™</sup>、Inc.、IBM Company の商標です。

# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アイテム 282  
アクション 64, 116, 296, 299, 306  
アクセス・ポリシー 14  
宛先 299  
アラート 116, 296, 299, 306  
案件管理 149  
依存関係のオーバーライド 57  
イベント 60, 64, 116  
印刷拡張性 22  
インバウンド処理 154  
インボイス作成 240  
運送会社 281, 297  
    定義 281  
運送会社プリファレンス 289  
延期  
    定義 229  
エンタープライズのオンボード化 29  
オーダー  
    オーダー用 136, 137  
    管理 154  
    交換オーダー (exchange order) 124  
    顧客用 136  
オーダー 保留処理 139  
オーダー管理 115  
オーダー作成コンソール 117, 118  
オーダー保留処理 139  
オーダー明細レベル 115  
オーダー・ヘッダー・レベル 115  
オーダー・リリース・レベル 115  
オーダー・インボイス作成済みステータス 118  
    「オーダー・エントリー」画面 117, 118  
オーダー・コンソール 115  
オーダー・ソーシング分類 156, 163  
オーダー・フルフィルメント・パイプライン 116  
オーダー・プロセス・タイプ 116  
オーダー・ヘッダー・レベル 115  
オーダー・ヘッダー・レベル 115  
オーダー・リリース  
    作成 291  
    制約 291  
オーダー・リリース・レベル 115

オンボード化 29

## [カ行]

回転の速い在庫 73  
外部でトリガーされたトランザクション (externally-triggered transaction) 55  
外部プログラム  
    動的ルーティング 288  
拡張、販売テーブルおよびフルフィルメント・ファウンデーション・テーブルの 12  
カスタム・オーダー 136  
カスタム・トランザクション 271  
    支払関連 API 271  
    支払い関連の外部プログラム 273  
カレンダー  
    出荷 178  
    「完全なオーダーを出荷 (Ship Complete Order)」フラグ 172  
    「完全な明細を出荷」フラグ 172  
管理者 230  
完了した集合・混載ステータス 300  
期限切れ順先出し (First Expiration First Out)。FEFO を参照。154  
キット化解除サービス 230  
キット化サービス 230  
基本プロセス・タイプ 52  
キャパシティ 206, 209, 212  
    外部保持キャパシティー 211  
キャパシティーの割り当て 145  
    パラメーター 145  
キャパシティー予約 146  
キャパシティー・チェックのオーバーライド 146  
キャンセル・ステータス 117, 246, 306  
供給 70  
    現在の供給 154  
    将来の供給 154  
    手持ちの供給 70  
警告管理 69  
警告の集約 20  
    時間別 21  
    集約テンプレート 21  
    日別 21  
経済的な出荷パラメーター 284, 289, 290  
決済システム  
    外部プログラムおよび API 250  
    同期決済処理 250  
決済処理のトランザクション 271  
    「検査済み」ステータス 307

交換オーダー (exchange order) 124, 125  
    「交換オーダー作成済み」ステータス 306  
    「交換オーダー作成の待機中」ステータス 306  
交換決済処理 252, 257  
交渉 118, 120  
    アクション 121  
    イニシエーター組織 120  
    応答 120  
    組織 120  
    プロセス 122  
交渉準備完了ステータス 118, 246  
交渉パイプライン 116, 118, 120  
購入オーダー 245  
    ステータス 246  
購入オーダー・パイプライン 116  
購入組織 26  
顧客スロット・プリファレンス 205  
コンプライアンス用に実行 243  
コンプライアンス・サービス 230, 242  
    実行 243  
    設定 243  
    「コンポーネントとして検査済み (Inspected As Components)」ステータス 307  
コンポーネントとして受入済みステータス 118, 246, 307  
梱包  
    入庫コンプライアンス 289

## [サ行]

サービス作業オーダー  
    確認 237  
    キャパシティーの計算 235  
    キャンセル 238  
    決定 234  
    サービス作業オーダーのタイプ 230  
    サービス・リソース 231  
    作成 234  
    製品 予約 237  
    提供 サービス作業オーダー 232  
    ノード  
        決定 234  
    配送サービス作業オーダー 233  
    パイプライン 241  
    複数日 232  
    複数のサービス明細 231  
    プロバイダー組織 235  
サービスの約束 211

- サービスの管理者 230
- サービスの約束 201
- サービス要請
  - 約束 201
- サービス・アイテム 205
- サービス・スキル 209
- サービス・スロット 203
  - 定義 203
- サービス・リソース 206, 231
  - キャパシティの計算 206
  - 利用可能性の計算 207
- 在庫
  - コンプライアンス・サービス 243
  - 在庫の追跡、有効期限 (expiration date) 別の 263
  - 実行 243
- 在庫管理 69, 245
- 在庫管理システム 70
- 在庫集約 75
- 在庫状況 207
  - スロット 212
- 在庫状況の照会
  - 定義 152
- 在庫状況モニター 72
  - アクティビティ・ベース 72
  - 完全同期 72
  - 即時同期 72
- 在庫変更サービス 230
- 在庫予約 142
  - 調達ノード 143
  - ドラフト・オーダー (draft order) 142
- 最小通知時間 178
- 最適化タイプ 168
- 作業オーダー 138
  - 作成 229
- 作業オーダー・パイプライン 241
- 作成済みステータス 117, 246, 306
- サプライヤー 154
- サポートされるトランザクション
  - トランザクションの依存関係 57
- 参加者 19
- 参加者モデリング 19, 25
- 時間トリガー・トランザクション 56
- 事前出荷通知 70
- 実行 243
- 支払システム
  - インボイス作成 253
  - 外部集金 250
  - 記録済みの前払い 250
  - 交換決済処理 252, 257
  - 事前決済のサポート 252
  - 支払ルール 258
  - データベースの詳細 273
  - 非同期支払処理 250
  - 複数の支払方法 249
  - 複数の制御レベル 258
- 支払システム (続き)
  - 返金フルフィルメント 256
  - 料金の集約 250
- 支払処理
  - 承認処理 262
  - 顧客 アカун ト (customer account) 270
- 支払ステータス 262
- 支払タイプ 258
  - 支払 方法 259
  - 請求シーケンス 260
  - 返金シーケンス 261
- 支払タイプのグループ 258
- 支払プロセス 262
  - 決済プロセス 268
  - 顧客アカウント (customer account) 271
  - 顧客アカウント (customer account) 269
- 集合・混載 282, 288, 299
  - 作成 299
  - 定義 282, 299
  - パイプライン 299
  - 保留処理 302
  - ルーティング 286
- 集合・混載作成済みステータス 300
- 集合・混載実行パイプライン 299
- 集約 21, 289, 290
  - 時間別 21
  - 出荷 282
  - 日別 21
- 受諾済みステータス 117, 246
- 出荷 282, 288, 291, 297
  - 作成 291
  - 集約 282, 292
  - ステータス 296
  - 定義 282
  - パイプライン 296
  - 保留処理 302
  - 容器 297
  - ルーティング 286, 292
  - ルーティング・ガイド 293
- 出荷 パイプライン 296
- 出荷完了ステータス 296, 301
- 出荷キャンセル済みステータス 297
- 出荷計画 284
  - エンタープライズ 289
  - 競合条件の解決 290
  - 経済的な出荷パラメーター 284
  - 決定、ルーティングの 286
  - 在庫制約 289
  - 動的ルーティング 288
  - 配送計画 288
  - パイヤー 289
  - ルーティング・ガイド 286
- 出荷コンソール 297
- 出荷先 282
  - 定義 282
- 出荷作成済みステータス 296, 301
- 出荷集約 292
- 出荷集約待機中ステータス 117
- 出荷済みステータス 118, 246, 307
- 出荷済み製品
  - 約束 153
- 出荷請求済みステータス 297, 301
- 出荷地 282
  - 定義 282
- 出荷遅延ステータス 118, 246
- 出荷日
  - 計算 174
- 出荷に含まれているステータス 118, 246, 307
- 出荷配達済みステータス 296, 301
- 出荷パイプライン 296, 297
- 出荷ルーティング済みステータス 297
- 出庫処理 154
- 出庫処理時間 73
- 出庫制約 289
- 受入終了済みステータス 246, 297, 307
- 受入済みステータス 118, 246, 307
- 受入取り消し済みステータス 118, 246, 307
- 受入に組み込み済みステータス 297, 301
- 需要 70
  - 現在の需要 154
  - 将来の需要 154
- 需要タイプ
  - 予約 71
- 循環依存関係 57
- 順序付け 219
- 条件 61, 116
- 承認処理 262
  - オーダー変更 268
  - 顧客アカウント (customer account) 270
  - 承認取り消し 264
  - 有効期限 (expiration date) 271
  - 「承認済み」ステータス 306
- 承認取り消し 264
- 将来のオーダー 175
- スケジューリング 191, 194, 195
  - キャパシティの計算 215
  - 出荷 190
  - ステータスのコントロール 190
  - 定義 152
  - 提供サービス 219
  - 配送サービス 216
  - 最も早い予定日 190
- スケジューリング・ルール 167
  - 地理 167
  - 定義 152
- スケジュール解除ステータス 118, 246

スケジュール済みステータス 118, 246  
ステータス 60, 116, 296, 299, 301, 306  
ドロップ・ステータス (drop status) 60  
ピックアップ・ステータス 60  
見積フルフィルメント・パイプライン 123  
スロットのスペン 207  
スロット・グループ  
定義 203  
「請求済み返品」ステータス 307  
製造日 70  
セキュリティ管理 12  
セグメンテーション 71  
セラー 162, 297  
ソーシング  
アイテム分類 160  
構成 157  
出荷の調達 160  
制約 172  
ソーシング分類 159  
地域 160  
定義 156  
ノードのシーケンス 164  
複数のシーケンス 166  
分配グループ (distribution group) 164  
ソーシング・モデル 187  
ソーシング・ルール 159, 163, 212, 213, 218  
ソーシング・ルールの決定 162  
地域  
階層 164  
定義 153  
倉庫 154  
贈答品用の包装 156, 228

## [夕行]

タグ 70  
タグ番号 70  
タスク・キュー日付の計算 59  
梱卸  
梱卸の実行 109  
定義 108  
梱卸要請 (count request) 110  
「単一出荷ノードからオーダーを出荷 (Ship Order from Single Ship Node)」フラグ 172  
「単一出荷ノードから明細を出荷 (Ship Line from Single Ship Node)」フラグ 172  
チーム 13  
地域 202  
定義 201  
地域スキーマ  
定義 201

地域スキーマ (続き)  
目的 201  
調達 160, 199  
予定出荷日 184, 185  
調達購入オーダー作成済みステータス 118  
調達購入オーダー出荷済みステータス 118  
調達転送オーダー作成済みステータス 118  
調達転送オーダー出荷済みステータス 118  
追加キャパシティー 210  
追加の固定キャパシティー 209  
通知  
オーダー・リリース 197  
出荷製品 197  
定義 153  
通知日  
計算 197  
データ・アクセス・ポリシー 14  
テーブル  
拡張 12  
定義 125  
提供サービス 230  
停止場所 282  
定義 282  
転送オーダー 199  
動的物理キット (dynamic physical kit)。DPK 参照。 137  
動的ルーティング 288  
「ドラフト作成済み」ステータス 306  
ドラフト・オーダー (draft order) 117, 246  
ドラフト・オーダー予約済みステータス 117  
ドラフト・ステータス 117, 246  
トランザクション 55, 116, 296, 299, 301, 306  
外部トリガー 55  
時間トリガー 56  
ユーザー・トリガー 55  
トランザクション依存関係 56  
エージェントの処理 59  
オーバーライド 57  
サポートされるトランザクション 57  
カスタム、拡張 57  
abstract 58  
システム・トランザクション 57  
循環依存関係 57  
回避 57  
使用法のシナリオ 56  
設定 56  
エンタープライズ・レベル 56  
伝票種別レベル 56  
タスク・キューの更新 58

トランザクション依存関係 (続き)  
他のトランザクションを一時停止 57  
次に利用可能になる日の計算 59  
バンドル・コンポーネント 56  
リスナー 57  
ルールの定義 56  
トレーラー積み付け完了ステータス 300  
ドロップ出荷サブライヤー 154

## [ナ行]

入荷 300  
入荷コンソール 301  
入荷パイプライン 301  
入庫オーダー 70, 199  
パイプライン 245  
入庫コンプライアンス 289  
入庫処理時間 73  
ノード 234  
出荷ノード 297  
定義 151  
ノードからバックオーダー済みステータス 117, 246  
ノードに送信ステータス 118, 246, 297  
ノード・ユーザー 297

## [ハ行]

配送計画 281, 288, 298  
定義 281  
配送サービス 230  
約束 211  
配達済みステータス 117  
配達日  
計算 182  
配達ノード  
定義 153  
配達ルート・モデル 283  
パイプライン 61, 115, 116, 241, 245, 296, 299, 306  
オーダー・フルフィルメント 116  
交渉 116  
交渉パイプライン 118  
購入オーダー・パイプライン 116  
作業オーダー・パイプライン 241  
集合・混載実行 299  
集合・混載実行パイプライン 299  
出荷 296  
出荷パイプライン 296  
入荷パイプライン 301  
パイプラインの決定 54  
プロセス・タイプ 52  
返品パイプライン 116  
返品物流 306

パイプライン (続き)

マスター・オーダー・パイプライン 116  
見積パイプライン 116  
見積フルフィルメント・パイプライン 123  
パイプラインの決定 54  
パイヤー 115  
エンタープライズ 27  
入庫コンプライアンス 289  
派生したオーダー (derived order) 136  
バックオーダー 117, 246  
バックオーダー済みステータス 117, 246  
バックオーダーの取り扱い 200  
バンドル 125  
出荷  
インボイス作成 298  
確認 298  
作成 298  
スケジューリング 196  
販売オーダー 297  
ピック中の出荷ステータス 296, 301  
日付範囲 174  
付加価値サービス。VAS を参照 227  
複雑な順序付け 219  
複数のトランザクション依存関係 56  
不足ステータス 118, 246, 307  
物流 管理  
計画に関する考慮事項 281  
物流管理 281  
計画戦略 282  
条件 281  
ルートの最適化 283  
フルフィルメント・タイプ 162  
プロセス・タイプ・パイプライン 52, 115  
プロバイダー組織 235  
分配グループ (distribution group) 217  
定義 152, 158  
返金フルフィルメント 256  
返品オーダー 306  
返品作成済みステータス 118  
返品受入済みステータス 118  
返品パイプライン 116  
返品物流  
ステータス 306  
定義 305  
パイプライン 306  
保存期間 74  
ホット SKU 108  
「保留」ステータス 306

## [マ行]

マスター・オーダー 124  
マスター・オーダー・パイプライン 116, 124  
マルチテナント・エンタープライズ・アーキテクチャー 30  
マルチ・スキーマ・デプロイメント 30  
「未承認」ステータス 307  
見積  
ステータス 123  
パイプライン 123  
見積パイプライン 116  
見積フルフィルメント・パイプライン 123  
「未リリース」ステータス 307  
モバイル・アプリケーションの拡張性 22

## [ヤ行]

約束 151  
出荷済み 製品 153  
提供サービス  
サービスの約束 216  
ユーザー 12  
ユーザー・グループ 13  
ユーザー・トリガー・トランザクション 55  
有効期間 74  
有効期限 (expiration date) 74  
郵便番号の範囲 202  
輸送管理システム 288  
輸送中の集合・混載ステータス 300  
容器 297  
予定  
カレンダーに基づく 239  
事前電話 240  
ステータス 238  
スロットに基づく 239  
手順 239  
予定出荷日  
計算 184, 185  
予定日付  
計算 174  
予約 70, 71  
サービス作業オーダー 237  
予約済みステータス 118, 246  
予約パラメーター 144

## [ラ行]

リード・タイム 73, 154  
リアルタイムの在庫状況モニター 72  
リソース 13  
リソース・アクセス権 13

リソース・プール  
定義 208  
リポジトリ 54  
利用可能なスロット 214  
料金の集約 250  
リリース 297  
「リリースから削除済み」ステータス 307  
リリース交渉中ステータス 117, 246  
リリース済みステータス 118, 246, 307  
ルーティング 299  
外部 286  
決定 286  
出荷 292  
動的 286, 288  
ルーティング・ガイド 286  
ルーティング・ガイド 286, 293  
在庫制約 290  
入庫コンプライアンス 289  
明細 286  
VICS 286  
ルーティング・ガイドライン 286  
例外キュー 20  
連鎖オーダー 133, 199  
連鎖オーダー作成済みステータス 117, 246  
連鎖オーダー作成待機中ステータス 117, 246  
ロット番号 70

## [ワ行]

ワークフロー管理 115

## A

ASAP オーダー 175  
ATP (納期回答) ルール  
定義 152  
ATP (納期回答) ルール (ATP (Available To Promise) rules) 154

## C

CSR (顧客サービス担当者) (CSR (Customer Service Representatives)) 115

## D

DPK (動的物理キット) DPK (Dynamic Physical Kit) 137

## E

- ESP チェックが必要ステータス 297
- ESP 保留解除ステータス 297
- ESP 保留時ステータス 297

## F

- FEFO (期限切れ順先出し) 154

## V

- VAS (付加価値サービス) (VAS (Value Added Services)) 138, 227
  - オーダー用 228
  - キット化解除 228
  - キットの分解 228
  - 顧客の要求 228
  - 在庫 228
  - 贈答品用の包装 228
  - 単位変換 228
  - 提供 サービス 227, 228
- VICS 286









Printed in Japan