

IBM WebSphere Commerce



管理ガイド

バージョン 5.5

IBM WebSphere Commerce



管理ガイド

バージョン 5.5

ご注意!

本書および本書で紹介する製品をご使用になる前に、193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 WebSphere Commerce バージョン 5.5 (プロダクト番号 5724-A18)、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。製品のレベルに合った適切な版を使用していることをご確認ください。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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： IBM WebSphere Commerce
Administration Guide
Version 5.5

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2003.8

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2003. All rights reserved.

© Copyright IBM Japan 2003

目次

+

本書について	vii
情報源	vii
WebSphere Commerce の資料	vii
WebSphere Commerce オンライン・ヘルプ	vii
Web 上の詳細情報	viii
変更の要約	viii
始める前に	ix
必要な知識	x
本書の規則	x
パスの変数	x
第 1 章 WebSphere Commerce の管理の概要	1
管理ツール	2
管理コンソール	2
構成マネージャー	3
WebSphere Commerce Payments	3
組織管理コンソール	10
WebSphere Commerce アクセラレーター	10
WebSphere Application Server	11
データベース	11
第 2 章 基本的な管理タスク	13
1 回限りの基本的な管理タスク	13
インストールと構成	13
定期的な管理タスク	13
WebSphere Commerce の開始と停止	13
WebSphere Commerce 構成マネージャーの起動	14
インスタンスの作成、構成、および削除	17
管理コンソールを開く	19
組織管理コンソールを開く	20
WebSphere Commerce コンポーネントを使用可能、または使用不可にする	21
WebSphere Commerce Payments の開始と停止	21
インストール・ガイドの管理用タスク	23
第 3 章 構成	25
構成サービス	25
アーキテクチャー上の構成	25
メッセージング・サービス	25
サービスのスケジューリング	26
ロギング・サービス	26
WebSphere Commerce Payments の問題判別	30
ルール・サービス	30
LDAP — Lightweight Directory Access Protocol	31
ステージング・サーバー	38
接続サービス	64
構成タスク	68
メッセージング・サービスの構成	68
スケジューリング・サービスの構成	68
ロギングの構成	68

ログ・ファイルの表示	72
LDAP の構成	74
ルール・サービスの構成	74
ステージング・サーバーを構成する	74
接続の構成	81
WebSphere Commerce Payments の構成	81
第 4 章 ホストされるストアの管理	83
販売店の管理	83
ディストリビューターの管理	84
サイト管理	84
第 5 章 動的キャッシング	85
WebSphere Commerce サブレットまたは JSP 結果のキャッシング	85
WebSphere Application Server 動的キャッシュ	85
動的キャッシュ	86
動的キャッシュ・サービスおよびサブレット・キャッシングの使用可能化	86
動的キャッシュ・モニターの使用可能化	86
WebSphere 動的キャッシュ・モニターの依存 ID のツリー・ビュー	87
キャッシング可能なオブジェクトの構成	87
キャッシングのデフォルト	89
全ページおよびフラグメントのキャッシング	89
全ページ・キャッシング	90
フラグメント・キャッシュ	91
キャッシュ・サブレット・フィルター操作	91
ストア・ページのキャッシング・ストラテジー	93
WebSphere Commerce におけるキャッシュ無効化メカニズム	93
cachespec.xml で無効化ポリシーを定義する	94
キャッシュ・モニターを使用してキャッシュ・エントリを除去する	97
動的キャッシュ無効化 API を呼び出す	97
コマンド・ベースの無効化のためのコマンドを書く	98
WebSphere Commerce でコマンド無効化を使用した例	98
キャッシュ無効化の例	99
キャッシュ無効化 API (DynaCacheInvalidation)	102
CACHEIVL テーブル	102
外部キャッシュ内の無効化	102
Web サーバー・プラグイン・キャッシュ	102
単純ファイル・サブレット	103
問題判別	104
第 6 章 システムの保守	107
Database Cleanup ユーティリティ	107
Database Cleanup ユーティリティのオブジェクト	108
Database Cleanup のコマンド	114
データベースのクリーンアップ	118
WebSphere Commerce ユーティリティ用に PATH 環境変数を設定する	119
オブジェクトを削除する	120
アカウント・オブジェクトを削除する	120
アドレス・オブジェクトを削除する	120
予定可能在庫オブジェクトを削除する	121
付加項目オブジェクトを削除する	122
オークション・オブジェクトを削除する	122

オークション・ログ・オブジェクトを削除する	123
自動入札ログ・オブジェクトを削除する	123
基本アイテム・オブジェクトを削除する	124
入札ログ・オブジェクトを削除する	125
計算コード・オブジェクトを削除する	125
キャンペーン・オブジェクトを削除する	126
キャンペーン統計オブジェクトを削除する	126
カタログ・エントリー・オブジェクトを削除する	127
カタログ・グループ・オブジェクトを削除する	127
契約オブジェクトを削除する	128
クーポン・オブジェクトを削除する	129
予定在庫レコード・オブジェクトを削除する	129
予定在庫レコード・オブジェクトに関する詳細情報を削除する	130
フォーラム・メッセージ・オブジェクトを削除する	130
配送センター・オブジェクトを削除する	131
在庫コード調整オブジェクトを削除する	132
在庫調整オブジェクトを削除する	132
指定アイテム情報オブジェクトを削除する	133
メンバー・メッセージ関係オブジェクトを削除する	133
メッセージ・オブジェクトを削除する	134
オーダー・オブジェクトを削除する	134
組織オブジェクトを削除する	135
商品アドバイザー統計オブジェクトを削除する	136
商品比較統計オブジェクトを削除する	136
商品探査統計オブジェクトを削除する	137
ポリシー・オブジェクトを削除する	138
商品セット・オブジェクトを削除する	138
見積依頼オブジェクトを削除する	139
返品オブジェクトを削除する	139
返品理由オブジェクトを削除する	140
セールス・アシスタント統計オブジェクトを削除する	140
ステージ・オブジェクトを削除する	141
ストア・オブジェクトを削除する	142
ユーザー・オブジェクトを削除する	142
ユーザー通信ログ・オブジェクトを削除する	143
取引先オブジェクトを削除する	144
Database Cleanup ユーティリティに新規構成を追加する	144
オブジェクトの削除の例	145
第 7 章 パフォーマンス	149
WebSphere Commerce PMI モジュールを使用したパフォーマンスのモニター	150
PMI モジュールの設定	151
Tivoli Performance Viewer の開始	152
WebSphere Commerce PMI モジュールのレポート値	153
他のパフォーマンス・ツール	154
スケジューラーの保守	154
LDAP の複製	155
ステージング・サーバーでサイトをテストする	155
ステージング・サーバーを構成する	155
カスタム・テーブルにトリガーを作成する	156
リモート・データベースを構成する	156
ステージング・データベースにデータをコピーする	156

Stage Check コマンドを実行する	157
実動データベースにデータを伝搬する	158
ファイルを実動サーバーにコピーする	158
ステージ・オブジェクトを削除する	159
WebSphere Commerce Payments のパフォーマンス・チューニング・パラメータ ー	159
iSeries 上での Payments インスタンス・データベースの管理.	162
第 8 章 トラブルシューティング	163
Adapter for CrossWorlds®	163
WebSphere Commerce Payments.	164
ハイパフォーマンス環境用のチューニング	164
付録 A. LDAP シナリオ	167
LDAP シナリオ: メンバー・リポジトリとしての LDAP サーバー.	167
付録 B. LDAP ファイル	169
ldapentry.dtd	169
ldapentry.xml	169
付録 C. WebSphere Commerce Payments チュートリアル	175
ステップ 1: WebSphere Commerce Payments ユーザー・インターフェースへの アクセス	175
ステップ 2: WebSphere Commerce Payments マーチャントの作成とカセットの 認可.	176
ステップ 3: WebSphere Commerce Payments ユーザーの定義	177
ステップ 4: ユーザー役割の割り当て	177
マーチャント管理者としてログオン	179
ステップ 5: アカountの作成.	179
アカウント設定の指定	180
ステップ 6: ブランドの作成	181
ステップ 7: Sample Checkout を使用したオーダーの作成	181
ステップ 8: オーダーの承認	183
「オーダー」ページでのオーダーの承認	183
販売機能を使用したオーダーの承認	184
ステップ 9: 決済のデポジット.	185
ステップ 10: バッチの清算	185
ステップ 11: クレジットの発行	187
ステップ 12: バッチ合計の表示	188
付録 D. WebSphere Commerce Payments の各国語 (NLS) サポート情報	191
NLS のヒント	191
サポートされる言語とロケール	192
特記事項	193

本書について

本書は管理者を対象としており、管理コンソールや他の管理ツールを使用して WebSphere® Commerce コンポーネントを管理する方法について取り上げます。


情報源


WebSphere Commerce には、e-commerce ソリューション一式について説明するオンライン情報とハードコピー情報が用意されています。さらに、WebSphere Commerce にバンドルされているソフトウェア・プロダクトには、ソフトウェアの特定のフィーチャーや機能について説明する詳細情報が準備されています。このセクションでは、さまざまな種類の情報がどこで提供されているかについて簡単に説明します。

WebSphere Commerce の資料

- 「WebSphere Commerce 基本 バージョン 5.5」
- 「WebSphere Commerce プログラミング・ガイドとチュートリアル バージョン 5.5」
- 「WebSphere Commerce インストール・ガイド バージョン 5.5」
- 「WebSphere Commerce クイック・スタート バージョン 5.5」
- 「WebSphere Commerce 追加ソフトウェア・ガイド バージョン 5.5」
- 「WebSphere Commerce マイグレーション・ガイド バージョン 5.5」
- 「WebSphere Commerce ストア開発ガイド バージョン 5.5」
- 「WebSphere Commerce サンプル・ストア・ガイド バージョン 5.5」
- 「WebSphere Commerce 管理ガイド バージョン 5.5」
- 「WebSphere Commerce セキュリティー・ガイド バージョン 5.5」
- 「WebSphere Commerce Payments プログラミング・ガイドとリファレンス バージョン 5.5」
- 「WebSphere Commerce Payments OfflineCard Cassette 補足 バージョン 5.5」
- 「WebSphere Commerce Payments CustomOffline Cassette 補足 バージョン 5.5」

これらの資料の更新情報は、次の Web アドレスで確認してください。

 Business Edition
(http://www.ibm.com/software/webservers/commerce/wc_be/)

 Professional Edition
(http://www.ibm.com/software/webservers/commerce/wc_pe/)

WebSphere Commerce オンライン・ヘルプ

WebSphere Commerce オンライン・ヘルプは、Web ブラウザーで表示可能なオンライン情報で構成されています。

オンライン・ヘルプにアクセスするには、Web ブラウザ (Internet Explorer バージョン 5.5 以上) に次のアドレスを入力してください。

https://host_name:8000/wchelp

host_name は、WebSphere Commerce マシンの名前です。

Web 上の詳細情報

WebSphere Commerce サポート

サポート情報 (ニュースグループ、FAQ、技術情報、トラブルシューティングの情報、ダウンロードなど) を検索するには、次の Web アドレスにアクセスしてください。

WebSphere Commerce Support Web ページ
(<http://www.ibm.com/software/commerce/support/>)

WebSphere Commerce Technical Library Web ページ
(<http://www.ibm.com/software/commerce/library/>)

ソフトウェア・パートナー

WebSphere Commerce を強化する製品とサービスが、多くのソフトウェア・パートナーから提供されています。パートナーの詳細は、Commerce Zone
(<http://www.ibm.com/software/wsdd/zones/commerce/>) にアクセスし、Software Developers のリンクをクリックして確認してください。

Redbooks™

さらに高度な技術情報を検索するには、Redbooks Web サイト
(<http://www.ibm.com/redbooks>) にアクセスして、WebSphere Commerce を検索してください。

+ 変更の要約

+ 本書ならびに本書の更新版はすべて、WebSphere Commerce Technical Library Web
+ ページ (<http://www.ibm.com/software/commerce/library/>) から入手可能です。ご
+ 使用の WebSphere Commerce の版については、以下の概説ページを参照してください。
+

+ • Business Edition
+ (http://www.ibm.com/software/webservers/commerce/wc_be/)

+ • Professional Edition
+ (http://www.ibm.com/software/webservers/commerce/wc_pe/)

+ サポートに関する詳細情報については、WebSphere Commerce Support サイト
+ (<http://www.ibm.com/software/commerce/support/>) を参照してください。

+ 最後の段階で製品に加えられた変更については、更新された製品の README ファ
+ イルを参照してください。これは上記の Web サイトでもご覧になれます。
+ WebSphere Commerce とサポートされている製品をインストールする方法について
+ は、「WebSphere Commerce インストール・ガイド」を参照してください。

+ 本書の最後の版以降に加えられた更新箇所を、余白の改訂文字で示します。本書で
+ は、改訂文字に関する以下の規則を使用しています。

+ • 「+」(プラス) 文字は、本書の現行バージョンで加えられた更新情報を示してい
+ ます。

+ 以下の表に、本書に加えられた主な変更内容を示します。

変更内容	関係する章またはページ
iSeries™ プラットフォーム用の dbclean の例の訂正。	107 ページの『第 6 章 システムの保守』
iSeries プラットフォーム用の dbclean、stagingcopy、stagingcheck、および stagingprop の例の訂正。	149 ページの『第 7 章 パフォーマンス』と 38 ページの『ステージング・サーバー』
『独立型ツールのロギング』のセクションは使用されなくなったため、除去されました。	68 ページの『ロギングの構成』
動的キャッシングについての情報の明確化と、その他の更新。	87 ページの『キャッシュ可能なオブジェクトの構成』

始める前に

IBM® WebSphere Commerce 管理ガイドでは、サイト管理者またはシステム管理者が WebSphere Commerce サイトで実行する必要がある作業について説明します。本書では、両方の役割をサイト管理者の役割と呼びます。

重要

本書では、随所にプラットフォーム固有の情報が 있습니다。ご使用のプラットフォームに固有の指示に従うようにしてください。

本書は、以下の章に分かれています。

WebSphere Commerce の管理の概要

基本的な管理タスク

構成

ホストされるストアの管理

動的キャッシング

システムの保守

パフォーマンス

トラブルシューティング

付録 A: LDAP シナリオ

付録 B: LDAP ファイル

付録 C: WebSphere Commerce Payments チュートリアル

付録 D: WebSphere Commerce Payments の各国語 (NLS) サポート情報

必要な知識

本書の対象読者は、WebSphere Commerce サイトをセットアップして保守する方法について理解する必要があるサイト管理者またはシステム管理者です。管理者は、以下の分野について理解している必要があります。

- e-commerce ソリューションのアーキテクチャー
- ネットワーク・インターフェース・コンポーネント (ファイアウォール、DFS™、DNS、ルーター、およびネットワーク・ハブ) のインストールおよび構成
- データベース・テクノロジー
- XML
- ストア・ビジネスの手順

本書の規則

本書では、以下の規則を使用しています。

太文字は、フィールドの名前、ボタン、またはメニュー選択項目などのグラフィカル・ユーザー・インターフェース (GUI) を表します。

モノスペース (Monospace) は、記載どおりに入力するテキストの例と、ディレクトリー・パスを表します。

イタリック は、強調を表したり、実際の値で置き換える変数を表します。



は、作業を進めるのに役立つ追加情報を表します。

400 は、WebSphere Commerce for the IBM Eserver iSeries 400® (旧称 AS/400®) に固有の情報を表します。

AIX は、WebSphere Commerce for AIX® に固有の情報を表します。

Linux は、WebSphere Commerce for Linux® に固有の情報を表します。

Solaris は、WebSphere Commerce for Solaris オペレーティング環境ソフトウェアに固有の情報を表します。

Windows は、WebSphere Commerce for Windows®に固有の情報を表します。

Professional は、WebSphere Commerce Professional Edition に固有の情報を表します。

Business は、WebSphere Commerce Business Edition に固有の情報を表します。

パスの変数

本書では、ディレクトリー・パスを次の変数で表します。

WC_installdir

これは、WebSphere Commerce のインストール・ディレクトリーです。各オペレーティング・システム上の WebSphere Commerce のデフォルト・インストール・ディレクトリーは、次のとおりです。

▶ AIX	/usr/WebSphere/CommerceServer55
▶ 400	/QIBM/ProdData/CommerceServer55
▶ Linux	/opt/WebSphere/CommerceServer55
▶ Solaris	/opt/WebSphere/CommerceServer55
▶ Windows	C:¥Program Files¥WebSphere¥CommerceServer55

▶ 400 *WC_userdir*

これは、WebSphere Commerce で使用するすべてのデータのディレクトリーです。このディレクトリーは変更可能であり、ユーザーが構成しなければならない場合もあります。このようなデータの例として、WebSphere Commerce インスタンス情報があります。これは、OS/400® に固有のディレクトリーです。

WC_userdir 変数は、/QIBM/UserData/CommerceServer55 ディレクトリーを表します。

WAS_installdir

これは、WebSphere Application Server のインストール・ディレクトリーです。各オペレーティング・システム上の WebSphere Application Server のデフォルト・インストール・ディレクトリーは、次のとおりです。

▶ AIX	/usr/WebSphere/AppServer
▶ 400	/QIBM/ProdData/WebAS5
▶ Linux	/opt/WebSphere/AppServer
▶ Solaris	/opt/WebSphere/AppServer
▶ Windows	C:¥Program Files¥WebSphere¥AppServer

WAS_userdir

これは、WebSphere Application Server で使用するすべてのデータのディレクトリーです。このディレクトリーは変更可能であり、ユーザーが構成しなければならない場合もあります。

▶ 400	/QIBM/UserData/WebAS5/Base/WAS_instance_name
-------	--

第 1 章 WebSphere Commerce の管理の概要

本書では、サイト管理者が日常的に実行するタスクに焦点を当てています。本書で説明されているサイト管理者の役割は、多くの会社に見られるシステム管理者の役割と同義であり、これ以降の部分ではサイト管理者と呼びます。

サイト管理者は、WebSphere Commerce に付属する定義済みのデフォルトの役割の 1 つです。通常、サイト管理者は、WebSphere Commerce および関連したソフトウェアおよびハードウェアのインストール、構成、および保守を行います。サイト管理者はシステム警告、アラート、およびエラーに応答し、システム問題を診断および解決します。この役割は通常、アクセス制御と権限制御 (メンバーを作成して、適切な役割へ割り当てる)、Web サイトの管理、パフォーマンスのモニター、およびロード・バランシング・タスクの管理を行います。サイト管理者は、さまざまな開発段階 (テスト、ステージング、および実動など) で必要となる、いくつかのサーバー構成の設定と保守も担当する場合があります。また、この役割は、重要なシステム・バックアップを扱ったり、パフォーマンス上の問題を解決したりします。

WebSphere Commerce の他の役割について詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。サイト管理者は以下の分野でタスクを実行します。

アクセス管理

- ユーザー
- 組織
- 役割
- アクセス・グループ
- ポリシー
- リソース・グループ
- アクション・グループ

セキュリティー

- アカウント・ポリシー
- パスワード・ポリシー
- アカウント・ロックアウト・ポリシー
- セキュリティー・チェッカー

パフォーマンス

システム・パフォーマンスのモニター

構成

- トランスポート
- メッセージ・タイプ
- メッセージ・ログ
- トレース
- コンポーネント構成
- スケジューラー
- レジストリー

- この商品について

WebSphere Commerce Payments

- ユーザー
- マーチャント設定
- WebSphere Commerce Payments 設定
- カセット
- トレース
- アクセス管理
- サイトのメッセージング・システムを定義する
- ルール・サービスを実動サーバーに伝搬する
- WebSphere Commerce Payments を構成する
- キャッシュを構成および保守する
- スケジューラーを構成する
- ログインを構成する
- データベースを保守する
- WebSphere Application Server を管理する
- Web サーバーを管理する
- システム・パフォーマンスのモニター
- サイトのセキュリティーを確認する
- トラブルシューティング

管理ツール

サイト管理者のタスク実行をより容易にするいくつかのツールがあります。このセクションでは主なツールを説明します。以降の章では、他のツールとそれを使用して実行できるタスクが扱われます。

管理コンソール

サイト管理者は、管理コンソールを使用することによってサイト構成およびストア構成に関連したタスクを実行することができます。管理コンソールの使用についての詳細は、19 ページの『管理コンソールを開く』を参照してください。サイト管理者が管理コンソールを使用して実行するタスクには、以下のものが含まれます。

- ユーザー、組織、役割、およびメンバー・グループを管理する
- アクセス制御を管理する
- サイトのトランスポートとメッセージ・タイプを定義する
- WebSphere Commerce Payments 設定を指定する
- WebSphere Commerce コンポーネントを使用可能、または使用不可にする
- サイトのために実行されるジョブをスケジュールする
- レジストリー・コンポーネントを更新する
- ストア・アーカイブを発行する

ストア発行ユーティリティー

管理コンソールに含まれているストア発行ユーティリティーを使用すると、WebSphere Commerce に付属のサンプルに基づいて、ストア・アーカイブを即座に作成できます。このユーティリティーを使用して、ストアで特定のフィーチャー (コラボレーションなど) を使用可能または使用不可にすることにより、発行済みの

ストアを構成することもできます。ストアの発行について詳しくは、「WebSphere Commerce オンライン・ヘルプ」の『ストア・アーカイブの発行』を参照してください。

構成マネージャー

構成マネージャーは、WebSphere Commerce のデプロイおよび実行に必要なインフラストラクチャーを確立および変更します。インフラストラクチャーの確立には、ご使用のデータベース、Web サーバー、および WebSphere Commerce Payments のロケーションについての情報、および WebSphere Commerce と共に使用するこれらのアプリケーションの構成に必要な他の情報を入手することが含まれます。また、ご使用のマシンのセットアップ情報および WebSphere Commerce アプリケーションの初期化に必要な他の情報を判別することも含まれます。構成マネージャーの使用についての詳細は、14 ページの『WebSphere Commerce 構成マネージャーの起動』、または「WebSphere Commerce インストール・ガイド」を参照してください。

サイト管理者は、構成マネージャーを使用することによって、構文に慎重な扱いを要する構成ファイルを扱わずに、管理タスクおよび構成タスクを実行できるようになります。サイト管理者が実行できる機能には以下のものがあります。

- WebSphere Commerce インスタンスを作成または削除する
- WebSphere Commerce インスタンスの構成設定を変更する
- Web サーバーを構成する
- データベースをステージング・サーバーとして使用するよう構成する
- WebSphere Commerce Payments のパラメーターを入力する

WebSphere Commerce Payments

WebSphere Commerce Payments は、WebSphere Commerce のコンポーネントで、このようなセキュリティー問題を解決して、インターネット・マーチャントに安全な電子決済処理を提供します。業界標準のオープンなテクノロジーに基づいて、WebSphere Commerce Payments は決済カセット を処理し、以下のものを含む複数の決済プロトコルをサポートします。

- VisaNet。世界的な通信データおよび決済処理のプロバイダーであり、決済を許可および清算することができます。
- BankServACH。Automated Clearing House Network (ACH) と交信する決済ゲートウェイで、オンライン電子小切手決済をサポートします。
- Paymentech。VISA およびマスターカードへの直接リンクを持つオンライン・クレジット・カード決済のマーチャント・プロセッサー。
- WebSphere Commerce Payments 用に作成された第三者の決済カセット。

簡単に言えば、WebSphere Commerce Payments は WebSphere Commerce と統合して、決済処理を管理するための現金レジのような機能を提供します。WebSphere Commerce Payments はインターネット・マーチャントのストアフロントの後ろに置かれ、決済の受理および銀行や他の金融機関とのこれらの決済処理を行うので、顧客が WebSphere Commerce Payments と直接対話することはありません。

注: IBM WebSphere Commerce Payments (これ以降、WebSphere Commerce Payments といいます) は、以前は Payment Manager と呼ばれていました。バージョン 3.1.3 以降は決済アプリケーションの名前が WebSphere Commerce Payments に変更され、本書での製品の呼称が変更されました。

複数の支払メソッドへの対応

WebSphere Commerce Payments は複数決済フレームワーク・アーキテクチャーを実装しており、複数の支払メソッドを受け入れなければならないマーチャントの便宜を図って柔軟で拡張可能な方法を提供します。複数決済フレームワークは、決済管理であるフレームワークを、特定の支払メソッドであるソフトウェア・カセットから分離して、それぞれが別個に展開するようにします。

WebSphere Commerce Payments は、プラグイン・アーキテクチャーを提供します。それによって、各支払メソッドのソフトウェア・カセットは決済フレームワークに接続します。フレームワークは、さまざまな支払メソッドの決済を行ったり、受け取ったりするのに必要な一般的なインフラ機能を提供します。

決済カセットは、WebSphere Commerce Payments フレームワークのデータ・フローおよび制御規則に準拠するソフトウェア・アプリケーションです。各決済カセットには、特定の支払メソッドおよびプロトコルが実装されています。

カセットは IBM または第三者の決済システム提供者によって作成されます。IBM は、カセットの開発をサポートしており、独自の決済カセットを作成する開発者に対して詳細な指示を与えます。カセット開発について詳しくは、「*WebSphere Commerce Payments Cassette Kit Programming Guide*」および <http://www.software.ibm.com/commerce/payments/download.html> の資料を参照してください。

WebSphere Commerce Payments ユーザー・インターフェース

Payments の機能には、WebSphere Commerce UI または独立した UI を使ってアクセスできます。ホストされる WebSphere Commerce Payments を使用する Payments 管理者および個々のマーチャントは、この両方を使用して以下のことを行えます。

- WebSphere Commerce Payments を構成する
- 決済とは関係ない以下の定期的な決済処理を実行する
 - オーダーの承認
 - 決済のデポジット
 - バッチの清算
 - クレジットの発行
 - 日ごとのバッチ合計の表示

WebSphere Commerce Payments ユーザー・インターフェースはブラウザー・ベースのもので、Web ブラウザーの SSL 機能を使用してリモートで安全にアクセスできます。

WebSphere Commerce Payments の役割

WebSphere Commerce Payments は、ユーザーの役割に基づいて各ユーザーに対して異なったビュー (たとえば、マーチャント管理者に対する Payments 管理者の観点) が表示されるようにして、役割を強制します。マーチャント組織内でも、WebSphere

Commerce Payments は様々な役割という概念を可能にして、マーチャントが自分のユーザーを監視できるようにします。たとえば、クラークはオーダーの承認などの操作だけに制限して、マーチャントまたは Payments 管理者が金融機関との関係を変更できるようにすることができます。

Payments フレームワークには 4 つの役割があり、それぞれが WebSphere Commerce の役割と対応して関連付けられています。ユーザーを作成して役割を割り当てる管理者にとって、このことを知っておくと役立ちます。WebSphere Commerce 組織管理コンソール内でユーザーを作成する場合は、最初に以下の WebSphere Commerce の役割をユーザーに割り当てる必要があります。次に、ユーザーが Payments UI に表示されるので、対応する Payments の役割をユーザーに割り当てることができます。次の表は、Payments のユーザー役割と対応する WebSphere Commerce の役割のマッピングを示しています。

表 1. 役割のマッピング

Payments の役割	WebSphere Commerce の役割
Payments 管理者	サイト管理者
マーチャント管理者	サイト管理者
スーパーバイザー	オペレーションまたはセールス・マネージャー
クラーク	顧客サービス・スーパーバイザー

Payments 管理者とマーチャント管理者は両方とも、WebSphere Commerce Payments を管理できます。スーパーバイザーおよびクラークは財務上の役割です。スーパーバイザーおよびクラークは、WebSphere Commerce Payments を管理しませんが、支払い処理機能を管理します。次の表に、Payments の各役割の職責を示します。

表 2. 役割の職責

役割	職責
Payments 管理者	<ul style="list-style-type: none"> • マーチャント管理者、スーパーバイザー、およびクラークを定義する • マーチャントおよびそれらのカセットを構成する • Payments ホスト名および状況を識別する • インストール済みのカセットを構成する • イベント・リスナーを追加、削除、および更新する • 決済を清算する • オーダーを承認または販売する • クレジットを発行し、クレジットを取り消す • オーダーをデポジットする • オーダーおよびバッチを検索する • 日ごとのバッチ合計の表示
マーチャント管理者	<ul style="list-style-type: none"> • マーチャント管理者、スーパーバイザー、およびクラークを定義する • マーチャントおよびそれらのカセットを構成する • イベント・リスナーを追加、削除、および更新する

表 2. 役割の職責 (続き)

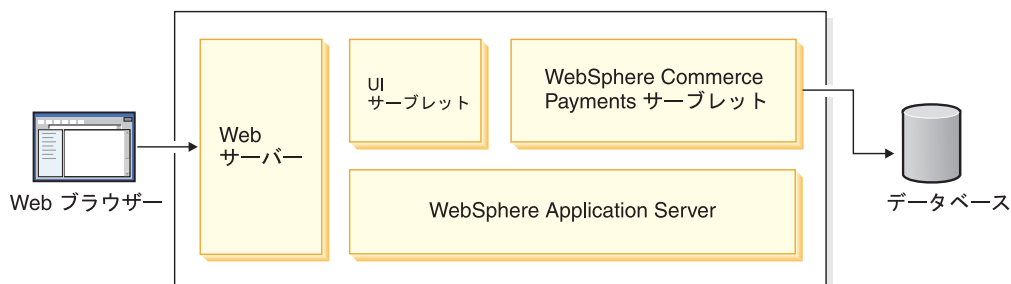
役割	職責
スーパーバイザー	<ul style="list-style-type: none"> • 決済を清算する • オーダーを承認または販売する • クレジットを発行し、クレジットを取り消す • オーダーをデポジットする • オーダーおよびバッチを検索する • 日ごとのバッチ合計の表示
クラーク	<ul style="list-style-type: none"> • 決済を清算する • オーダーを承認または販売する • オーダーをデポジットする • オーダーおよびバッチを検索する • 日ごとのバッチ合計の表示

WebSphere Commerce Payments のコンポーネント

WebSphere Commerce Payments を理解するには、以下の WebSphere Commerce Payments コンポーネントについて考察して、そのコンポーネントを定義することが役立ちます。

- WebSphere Application Server によって構成された Web サーバー
- WebSphere Commerce Payments サブレット
- ユーザー・インターフェース・サブレット
- データベース

以下の図は、WebSphere Commerce Payments の内部コンポーネントを示しています。



WebSphere Commerce Payments サブレットは、WebSphere Commerce Payments の主要なコンポーネントです。このサブレットは、WebSphere Application Server (すべてのプラットフォームで共通するサブレット環境を提供する製品) と動作するように設計されています。

このセクションの残りの部分は、WebSphere Commerce Payments サブレット、およびそれに付随する以下のサポート製品についての定義を行います。

- Web サーバー
- WebSphere Application Server
- データベース

Web サーバー: なぜ Web サーバーを使用する必要があるのでしょうか。この章で前述されているとおり、WebSphere Commerce Payments は、着信 HTTP 要求を処理します。HTTP 要求はまず、WebSphere Application Server によって構成された Web サーバーによってフィールド付けされます。これらの要求はそれから、他の WebSphere Commerce Payments コンポーネントに中継されます。

IBM WebSphere Application Server: なぜ WebSphere Application Server で Web サーバーを構成するべきなのでしょう。それには、以下の 2 つの理由があります。

1. WebSphere Application Server は、複数のプラットフォーム上の様々な Web サーバーと動作する製品です。
2. WebSphere Application Server は、WebSphere Commerce Payments サブプレットを実行できる環境を提供しています。

WebSphere Commerce Payments サブプレット: Payment サブプレットとも呼ばれる WebSphere Commerce Payments サブプレットは、WebSphere Commerce Payments の中心的な部分です。これはユーザー・インターフェースまたは他のマーチャント・アプリケーションからの要求を受け取り、セキュリティ検査の後に、Payment カセットを呼び出して、金融ネットワークとの通信などの必要な決済処理を扱い、最後に呼び出し元のアプリケーションに応答を送信します。これは WebSphere Commerce Payments データベースに接続し、そこに保管されているセキュア・データにアクセスできる唯一のコンポーネントです。

WebSphere Commerce Payments および e-commerce エンティティ

このセクションでは、WebSphere Commerce Payments がマーチャントのショッピング・ソフトウェアをサポートする方法を取り上げます。3 つのシナリオで、WebSphere Commerce Payments がマーチャント・ソフトウェアおよび汎用決済プロトコルとともに、以下の e-commerce エンティティと対話する方法を説明します。

- マーチャント
- バイヤー
- 金融機関

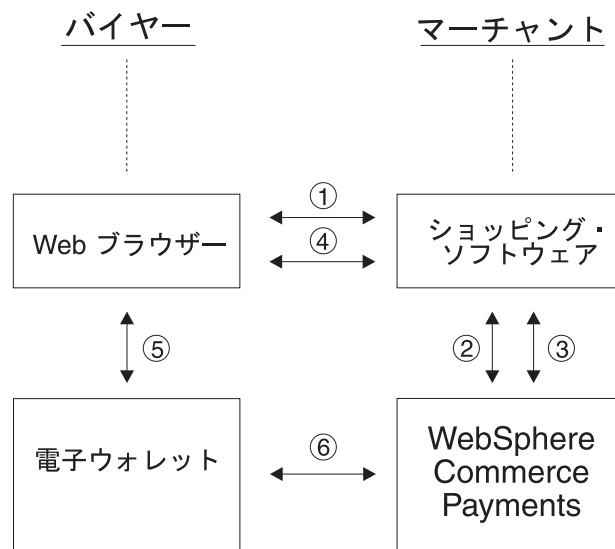
以下のシナリオでは、WebSphere Commerce Payments を使用した e-commerce について説明します。重要な点として、これらのシナリオは一般的なものなので、特定の決済プロトコルでの対話を示すものではないことに注意してください。むしろ、これらのシナリオは、多くの e-commerce ビジネス・シナリオに共通する基本的なトランザクションの流れを経験するためのものです。

シナリオ 1: 現金および電子ウォレット: この最初のシナリオは仮想の現金プロトコルを使用し、バイヤーおよびマーチャントという 2 つのエンティティが関係しています。資金はバイヤーからマーチャントへ送られます。このトランザクションでは、バイヤーはプロトコル固有の電子ウォレット・アプリケーションを使用します。トランザクションのステップが、下記のシナリオ図で説明されています。

1. バイヤーとマーチャント・ショッピング・ソフトウェアとの対話は、バイヤーがマーチャントに対して購入要求を出すことによって締結します。

2. 購入要求の応答として、マーチャント・ソフトウェアは決済開始コマンド（つまり、決済を受け取る）を WebSphere Commerce Payments に送ります。
3. WebSphere Commerce Payments は、バイヤーのウォレット・プログラム用のプロトコル固有情報を含む決済開始メッセージを返します。
4. マーチャント・ソフトウェアはこの決済開始メッセージをバイヤーのブラウザに転送します。
5. バイヤーのブラウザで受け取られた決済開始メッセージは、バイヤーのウォレット・プログラムを開始します。
6. WebSphere Commerce Payments とウォレット・ソフトウェアの対話によってバイヤーはトランザクションを完了します。

シナリオ 1: 現金および電子ウォレットを使用した WebSphere Commerce Payments のやり取り



このシナリオと後続の 2 つの例との違いは、この例では e-commerce トランザクションには以下に示す 2 つのエンティティーしか関係していないところにあります。

- バイヤー (電子ウォレットによって表される)
 - WebSphere Commerce Payments を使用するマーチャント
- 金融機関は関係していません。

シナリオ 2: クレジット・カードおよび電子ウォレット: 2 番目のシナリオは、仮想のクレジット・カード・プロトコルを使用した購入を示しており、以下の 3 つのエンティティーが関係しています。

- マーチャント
- バイヤー (今回も電子ウォレットを使用する)
- 金融機関

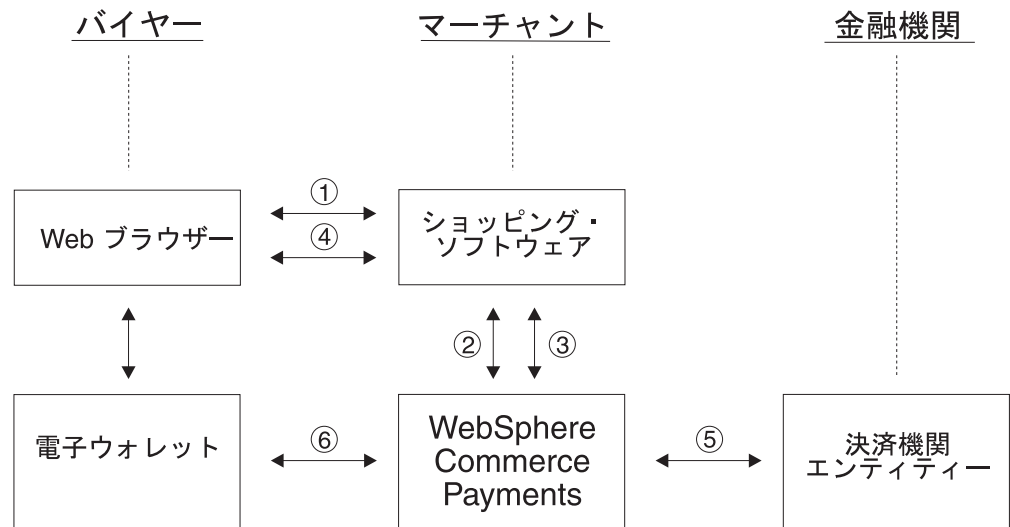
このシナリオの流れは、以下の点を除いて、シナリオ 1 と同じです。

- バイヤーのウォレットが購入要求を完了する際に、WebSphere Commerce Payments は金融機関の決済機関エンティティーに承認要求を出して、カード所有者が十分な資金を持っていることを確認します。

- WebSphere Commerce Payments が承認状況を示す確認応答をバイヤーのウォレットに送信することによって、トランザクションは終了します。

下記のシナリオ図では、この違いをステップ 5 と 6 で示しています。

シナリオ 2: クレジット・カードおよび電子ウォレットを使用した WebSphere Commerce Payments のやり取り

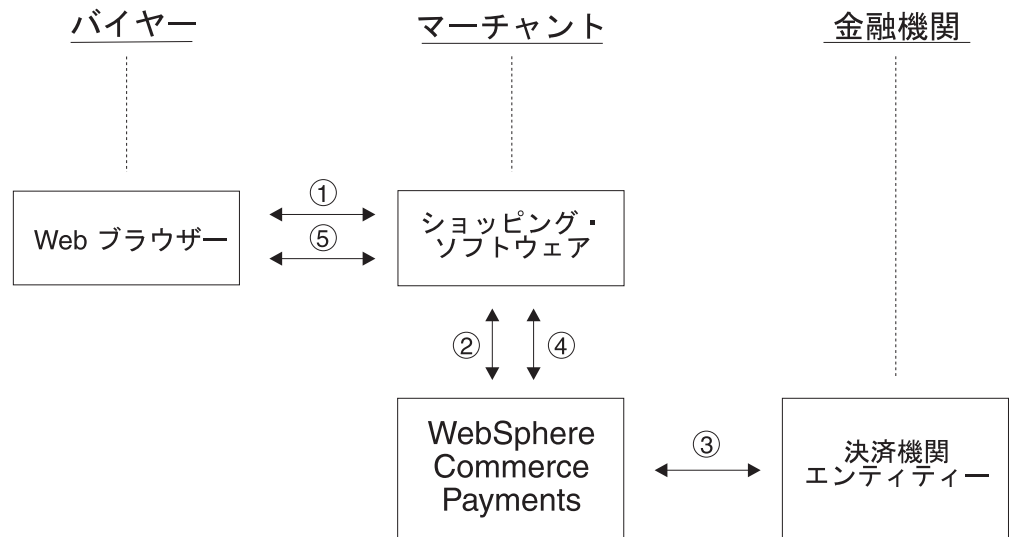


顧客のウォレット・プログラムが開始して取引が完了すると、金融機関の決済機関エンティティはカード所有者が十分な資金を持っていることを確認し、WebSphere Commerce Payments は資金が送金されたことをバイヤーに確認します。

シナリオ 3: クレジット・カードおよび SSL: この最後のシナリオでは、バイヤーがウォレットまたは消費者固有のソフトウェアを持っていない場合のクレジット・カード・プロトコルを示します。消費者の情報はすべて安全な方法 (たとえば、SSL ソケット) でマーチャント・ソフトウェアに渡されます。トランザクションのステップが、下記のシナリオ図で説明されています。

1. バイヤーとマーチャント・ショッピング・ソフトウェアとの対話は、バイヤーがマーチャントに対して購入要求を出すことによって締結します。
2. 購入要求の応答として、マーチャント・ソフトウェアは決済開始コマンドを WebSphere Commerce Payments に送ります。
3. WebSphere Commerce Payments は、決済機関エンティティの金融機関を通して、十分なカード所有者資金があることを確認することによって、トランザクションを承認しようとしています。
4. WebSphere Commerce Payments は結果 (資金のデポジットが行われたことなど) をマーチャント・ソフトウェアに戻します。
5. 最後に、購入要求が成功したかまたは失敗したかがバイヤーに通知されます。

シナリオ 3: クレジット・カードおよび SSL を使用した WebSphere Commerce Payments のやり取り



上記の図は、実際に起こりうる e-commerce のシナリオを示しており、それらのシナリオでの WebSphere Commerce Payments の役割を表しています。これら以外にも多くのビジネスの流れが存在しますが、WebSphere Commerce Payments はそのような対話においても同様の決済機能を想定します。

組織管理コンソール

サイト管理者は、組織管理コンソールを使用することによって、サイトまたはストアにアクセスする組織を制御できます。実行権限のあるタスクは、種々のメニューから「組織管理コンソール」ホーム・ページに表示されます。これらのタスクはユーザーの役割および権限レベルを基にしています。詳しくは、「WebSphere Commerce オンライン・ヘルプ」で組織管理コンソールに関する情報を参照してください。

WebSphere Commerce アクセラレーター

WebSphere Commerce アクセラレーターを使用すれば、ストアのルック・アンド・フィールの管理から、オーダーを作成および保守してストア・アクティビティーをトラッキングすることに至るまで、種々のストア操作を完全に実行することにより、オンライン・ストアを保守できるようになります。自分の役割において実行権限のあるタスクは、「WebSphere Commerce アクセラレーター」ホーム・ページ・メニューに表示されます。これらのタスクは、サイト管理者が管理コンソールを使用して定義する、ユーザーの役割および権限レベルを基にしています。詳しくは、「WebSphere Commerce オンライン・ヘルプ」で WebSphere Commerce アクセラレーターに関する情報を参照してください。

注: サイト管理者が組織管理コンソールまたは WebSphere Commerce アクセラレーターを使用して実行するタスクについては、オンライン・ヘルプで説明されています。

WebSphere Application Server

WebSphere Application Server は、インターネットおよびイントラネット Web アプリケーションの作成、デプロイ、および管理のための、Java 関連のアプリケーション環境です。WebSphere Application Server は、Java™ Server Page ファイルのランタイム・サポートも備えています。

WebSphere Application Server の管理コンソールには、管理者が次のことを行うためのツールやフィーチャーが備わっています。

- WebSphere Commerce サーバーを停止または開始する
- IBM WebSphere Commerce Payments を停止または開始する
- リソース (サーブレットおよびエンタープライズ Bean など) をインストールおよび構成する
- リソースにセキュリティーを割り当てる
- すべてのアプリケーションが使用可能であることを確認する
- ユーザー・アクセスを付与したり取り消したりする (新規従業員のために新規アカウントを作成して、特定のユーザー・グループに追加するなど)
- Commerce Server のパフォーマンスをモニターする
- パフォーマンス向上のためにアプリケーション・コンポーネントを複製する
- WebSphere アプリケーションの実行に関するトレースとデバッグを提供する

WebSphere Application Server の情報は、WebSphere Application Server Web サイト (<http://www.ibm.com/software/webservers/appserv>) から入手できます。

WebSphere Application Server のシステム管理についての詳細情報は、以下の資料を参照してください。

- *Systems Administration*
- *Tuning and Troubleshooting*

データベース

データベース管理システムとして DB2® または Oracle を使用できます。詳しくは、各システムの「WebSphere Commerce インストール・ガイド」を参照してください。

第 2 章 基本的な管理タスク

この章では、定期的に行う必要がある基本的な管理タスクや、1 回だけ行う必要がある基本的な管理タスクについて説明します。ビジネス・プロセスによっては、これらのタスクを実行しなくてよい場合もあります。

1 回限りの基本的な管理タスク

以下のタスクは、通常は 1 回だけ実行します。

インストールと構成

サイト管理者として実行するタスクには、WebSphere Commerce サイトをインストールして構成するというタスクがあります。一般に、これは最初に行うなければならないステップです。通常、インストールとは、使用しているオペレーション・システムに応じた WebSphere Commerce ソフトウェアをインストールすることを意味します。インストール・ステップは、製品に付属のインストール・ガイドで説明されています。インストールと構成の詳細については、「*WebSphere Commerce インストール・ガイド*」を参照してください。構成についての情報がさらに必要な場合は、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

定期的な管理タスク

以下のタスクは、定期的に行う必要があります。

- WebSphere Commerce の開始と停止
- 構成マネージャーを開く
- WebSphere Commerce インスタンスの作成、構成、および削除
- 管理コンソールを開く
- 組織管理コンソールを開く
- WebSphere Commerce コンポーネントを使用可能、または使用不可にする
- WebSphere Commerce Payments の開始と停止

WebSphere Commerce の開始と停止


WebSphere Commerce の各インスタンスには、独自の WebSphere Commerce Server があります。WebSphere Commerce Server は、WebSphere Commerce システム内の情報の流れを制御するための Java ベースのコマース・サーバーです。WebSphere Commerce Server の各コンポーネントについての詳細は、「WebSphere Commerce オンライン・ヘルプ」の『WebSphere Commerce Server』を参照してください。

WebSphere Commerce インスタンスは、`WC_instance_name` というアプリケーション・サーバーとして開始および停止されます。ここで、`instance_name` は、開始または停止する WebSphere Commerce インスタンスの名前です。たとえば、`demo` という WebSphere Commerce インスタンスを開始するには、`WC_demo` というアプリケーション・サーバーを開始する必要があります。

WebSphere Commerce インスタンスを開始または停止するには、以下のようになります。

1. 開始または停止する WebSphere Commerce インスタンスを決定します。
2. ご使用のプラットフォーム用のアプリケーション・サーバーを停止および開始するための説明に従います。これらの説明については、「WebSphere Commerce オンライン・ヘルプ」の『Starting and stopping an application server on.<your platform>』を参照してください。

WebSphere Commerce Payments がまだ開始されていない場合は、WebSphere Commerce の開始後に開始できます。WebSphere Commerce Payments を開始する方法については、21 ページの『WebSphere Commerce Payments の開始と停止』を参照してください。

注:  WebSphere Commerce Studio 内で WebSphere Commerce を開始する方法については、「WebSphere Commerce オンライン・ヘルプ」の『Starting and stopping WebSphere Commerce within WebSphere Commerce Studio』を参照してください。

WebSphere Commerce 構成マネージャーの起動

構成マネージャーでは、WebSphere Commerce のさまざまなコンポーネントを使用可能にしたり構成したりすることができます。可能な場合は常に、WebSphere Commerce の構成ファイルを変更するという方法ではなく、構成マネージャーを使用してコンポーネントを使用可能にしてください。WebSphere Commerce インスタンスを作成または構成するには、構成マネージャーを使用してください。

WebSphere Commerce 構成マネージャーの AIX、Linux、および Solaris ソフトウェア上での起動

WebSphere Commerce 構成マネージャーを AIX、Linux、および Solaris ソフトウェア上で起動するには、以下のようになります。

1. 次のようにして、サーバーを始動します。
 - a. 端末ウィンドウをオープンします。
 - b. WebSphere Commerce のインストール手順の一部としてインストール後スクリプトが実行されたときに作成された WebSphere Application Server ユーザーでログインします。デフォルトの WebSphere Application Server ユーザー名は *wasuser* です。
 - c. 以下のコマンドを発行します。

```
export DISPLAY=host_name:0.0
xhost +host_name
cd /usr/WebSphere/CommerceServer55/bin
./config_server.sh
```



```
export DISPLAY=host_name:0.0
xhost +host_name
cd /opt/WebSphere/CommerceServer55/bin
./config_server.sh
```

host_name は、構成マネージャーへのアクセスに使用するマシンの完全修飾ホスト名です。

注:

- 1) `config_server.sh` コマンドを入力した端末ウィンドウを閉じないでください。閉じると、構成マネージャーが停止します。
 - 2) 構成マネージャー・サーバーをバックグラウンド・プロセスとして実行しないでください。セキュリティ上の危険につながります。
2. 以下のようにして、クライアントを開始します。
- a. 別の端末ウィンドウをオープンします。
 - b. WebSphere Commerce のインストール手順の一部としてインストール後スクリプトが実行されたときに作成された WebSphere Application Server ユーザーでログインします。デフォルトの WebSphere Application Server ユーザー名は *wasuser* です。
 - c. 以下のコマンドを発行します。

▶ AIX ▶ Solaris

```
export DISPLAY=host_name:0.0
xhost +host_name
cd /usr/WebSphere/CommerceServer55/bin
./config_client.sh &
```

▶ Linux

```
export DISPLAY=host_name:0.0
xhost +host_name
cd /opt/WebSphere/CommerceServer55/bin
./config_client.sh &
```

host_name は、構成マネージャーへのアクセスに使用するマシンの完全修飾ホスト名です。

iSeries 上での WebSphere Commerce 構成マネージャーの起動

WebSphere Commerce 構成マネージャーを iSeries 上で起動するには、以下のようになります。

1. 次のようにして、サーバーを始動します。
 - a. プロファイルが *SECOFR ユーザー・クラスをもっていることを確認し、さらに、英語に固有の設定を使用するか、またはインスタンス用のデフォルト言語として選んだ言語に合った設定を使用して、そのプロファイルがセットアップされていることを確認したうえで、iSeries マシンにログオンします。
 - b. 次のコマンドを入力して QShell セッションを開始します。

```
STRQSH
```

さらに、QShell セッションで次のようにします。

- 1) 次のコマンドを実行して、WebSphere Commerce サーバーの bin ディレクトリーに切り替えます。

```
cd /qibm/proddata/commerceserver55/bin
```
- 2) 次のコマンドを実行して、構成マネージャー・サーバー・プログラムを開始します。

```
config_server.sh -port server_port_number
```

ここで *server_port_number* は、構成マネージャーが listen する iSeries サーバー上のポート番号です。このパラメーターはオプションで、デフォルトは 1099 です。この値の範囲は 1024 から 65535 で、現在は使用されていません。

注: 1 次言語がインスタンスの作成で使用している言語とは異なるシステムを使用している場合は、*QSYSlanguage_feature_number* ライブラリーをユーザー・プロファイルのライブラリー・リストに追加する必要があります。追加しないと、ユーザー・プロファイルは *QSYS* の下でライブラリーを探そうとします。言語機能ライブラリーを追加するには、*EDTLIBL* コマンドを使用します。

- c. 今回初めて構成マネージャーをシステムで実行する場合は、以下のメッセージが表示されます。

```
Attaching Java program to /QIBM/ProdData/WebCommerce55/lib/wcsconfig.JAR.  
Attaching Java program /QIBM/ProdData/WebCommerce55/lib/wcsruntime.JAR.  
Attaching Java program to /QIBM/ProdData/WebCommerce55/wc.ear/lib  
/wcslogging.JAR.  
Attaching Java program to /QIBM/ProdData/WebCommerce55/lib/xml4j.JAR.  
Attaching Java program to /QIBM/ProdData/WebCommerce55/lib/sslite.ZIP.
```

- d. 以下のメッセージを受け取ったら、次のステップに進みます。

```
Registry created.  
CMServer bound in registry.
```

- 2. 以下のようにして、クライアントを開始します。

- a. クライアント・マシンのコマンド・プロンプトを使用して、*WCS400* ディレクトリーに変更します。
- b. 以下のコマンドを入力して、クライアントを構成します。

```
config_client.bat iSeries_Host_name Server_port_number
```

ここで、

iSeries_Host_name

サーバーの完全修飾ホスト名です。

Server_port_number

構成マネージャーが listen する iSeries サーバー上のポート番号です。

Windows での WebSphere Commerce 構成マネージャーの起動

Windows で WebSphere Commerce 構成マネージャーを起動するには、次のようにします。

- 1. IBM WC 構成マネージャー・サーバー・プロセスが稼働中であることを確認します。それには、「スタート」→「設定」→「コントロール パネル」→「管理ツール」→「サービス」を選択して、IBM WC 構成マネージャーのサービスの状況が「開始済み」になっているかどうかを調べます。

重要

IBM WC 構成マネージャー・サーバーのサービスを実行中のままにすると、セキュリティ上の問題を生じる可能性があります。構成マネージャーを使用しないときは、WC 構成マネージャー・サーバー・サービスを停止してください。

セキュリティ上の問題が起きないようにするには、IBM WC 構成マネージャーを自動ではなく手動で始動するように設定してください。

2. 「スタート」→「IBM WebSphere Commerce」→「構成」を選択します。

インスタンスの作成、構成、および削除

WebSphere Commerce インスタンスを作成、構成、または削除するには、構成マネージャーを使用してください。このツールは、無効な値の入力を防止してインスタンスを構成できるように設計されています。

大部分の WebSphere Commerce 構成パラメーターの変更には、構成マネージャーを使用してください。ただし、一部のパラメーターは、WebSphere Commerce 構成ファイル (*instance_name.xml*) によって変更する必要があります。コンポーネントを構成する際に、構成マネージャーと WebSphere Commerce 構成ファイル (*instance_name.xml*) のどちらを使用すべきかを判別するには、オンライン・ヘルプの個々のコンポーネントの資料を参照してください。

インスタンスの作成と構成

インスタンスを作成するには、以下のようにします。

1. 構成マネージャーを開きます。構成マネージャーの開き方については、14 ページの『WebSphere Commerce 構成マネージャーの起動』を参照してください。
2. マシン名が付いたノードを拡張して、「インスタンス・リスト」を選択し、「アクション」メニューから「インスタンスの作成」を選択します。
3. インスタンス作成ウィザードが、インスタンスを作成するプロセスをガイドします。パネルでの操作が分からない場合は、いつでも「ヘルプ」を選択してください。

インスタンスを構成するには、以下のようにします。

1. 構成マネージャーを開きます。
2. マシン名が付いたノードを拡張して、「インスタンス・リスト」を拡張表示し、変更するインスタンスを選択します。
3. 変更するコンポーネントを選択します。
4. そのコンポーネントがすでに存在する場合は、コンポーネントを選択すると適切なパネルが開きます。次いで、必要に応じてコンポーネントを変更します。
5. コンポーネントが存在しない場合は、コンポーネントを右マウス・ボタン・クリックして、コンポーネントの作成を選択します。ウィザードがコンポーネントを作成するプロセスをガイドします。

インスタンスの削除

WebSphere Commerce インスタンスを削除するには、以下のようになります。

1. WebSphere Commerce インスタンスをバックアップします (推奨)。
2. `WC_installdir¥instances¥instance_name¥xml`、または `400` `WC_userdir/instances/instance_name/xml` ディレクトリーに切り替えて、`instance_name.xml` ファイルをバックアップ・ディレクトリーにコピーします。
3. WebSphere Application Server を開始します。
4. 以下のようにして、WebSphere Commerce Server を除去します。

- `400` QSHHELL セッションのコマンド行から、以下を入力します。

```
WC_installdir/bin/rmCommerceServer.sh instance_name
```

`AIX` `Solaris` `Linux` コマンド・プロンプトから、以下を入力します。

```
WC_userdir/bin
./rmCommerceServer.sh instance_name
```

- `Windows` コマンド・プロンプトから、以下を入力します。

```
WC_installdir/bin
./rmCommerceServer instance_name
```

5. WebSphere Commerce 構成マネージャーで、インスタンスを右マウス・ボタン・クリックして「削除」を選択します。
6. 以下のようにして、WebSphere Commerce Business Edition データベースをドロップします。

- `DB2` DB2 データベースを使用している場合は、以下のようになります。

- a. DB2 コマンド・ウィンドウを開いて、以下を入力します。

```
db2 drop db db_name
db2 uncatalog node node_name
```

ここで `db_name` は、構成マネージャーに入力した WebSphere Commerce データベースの名前で、`node_name` は、これらのデータベース用にカタログしたノードの名前です。デフォルトのノード名は、マシンのホスト名の短縮形です (つまり、完全修飾されていません)。

- `Oracle` Oracle データベースを使用している場合は、以下のようになります。

- a. SQL*Plus セッションを開始して、以下を入力します。

```
sqlplus system/manager@wc_SID
```

ここで、`wc_SID` は、WebSphere Commerce データベース用の Oracle システム ID (SID) です。

- b. SQL*Plus セッションで以下のコマンドを発行します。

```
DROP TABLESPACE wc_tablespace_name
INCLUDING CONTENTS;
DROP USER wc_oracle_user_ID
CASCADE;
EXIT;
```


注: 実際の表スペース・ファイルも除去する必要があります。表スペース・ファイルには、`wc_SID.ora` という名前が付けられます。ここで、`wc_SID` は、WebSphere Commerce データベース用の Oracle システム ID (SID) です。

▶ 400 SQL ステートメント `drop schema instance_name` を実行して、インスタンス・データベース・ライブラリーを削除します。スキーマを削除しようとして、接続されている間は接続側を削除できないというメッセージを受け取る場合は、`ENDJRPF FILE(*ALL) JRN(instance_name/QSQJRN)` コマンドを実行してみてください。

7. WebSphere Commerce インスタンスを完全に削除するには、以下のディレクトリを削除します。

• ▶ AIX ▶ Solaris ▶ Linux ▶ Windows

`WC_installdir/instances/instance_name`

• ▶ 400 `WC_userdir/instances/instance_name`

8. ▶ 400 DLTUSRPF コマンドを使用して、インスタンス・ユーザー・プロファイルを削除します。

管理コンソールを開く

管理タスクを実行するには、多くの場合、管理コンソールを使用する必要があります。管理コンソールでは、管理操作と構成タスクを行ってサイトまたはストアを制御できます。管理コンソールの詳細については、2 ページの『管理コンソール』を参照してください。

1. 管理コンソールは、以下のいずれかの方法で開始します。

• ブラウザーで以下の URL にアクセスします。

`https://host_name:8002/adminconsole`

ここで `host_name` は、WebSphere Commerce Server の完全修飾 TCP/IP 名です。

- ▶ Studio WebSphere Commerce Studio をインストールしてあり、このマシンから管理コンソールを開く場合は、以下のようになります。
 - a. WebSphere Commerce Studio 内で WebSphere Commerce を開始してあることを確認します。WebSphere Commerce Studio 内で WebSphere Commerce を開始する方法については、「WebSphere Commerce オンライン・ヘルプ」の『Starting and stopping WebSphere Commerce within WebSphere Commerce Studio』を参照してください。
 - b. ブラウザーで以下の Web アドレスにアクセスします。

`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

ここで `host_name` は、WebSphere Commerce Studio マシンの完全修飾 HTTP ホスト名です。

2. 以下のようにして、管理コンソールにログオンします。

- a. 「ログオン」ページで管理コンソールのログオン・ユーザー名とパスワードを入力して、「ログオン」をクリックします。複数のストアまたは言語を扱

う権限がある場合は、「管理コンソールのサイト/ストアの選択」ページが表示されます。単一のストアおよび言語しか扱う権限がない場合は、「管理コンソール」ホーム・ページが表示されます。実行する権限があるタスクは、「管理コンソール」ホーム・ページに表示されます。

3. 「管理コンソールのサイト/ストアの選択」ページで「**サイト (site)**」ラジオ・ボタンまたは「**ストア (store)**」ラジオ・ボタンを選択して、「**OK**」をクリックします。
 - 「**サイト (site)**」を選択すると、「管理コンソール」ホーム・ページが表示されます。
 - 「**ストア (store)**」を選択すると、ストアの名前と言語を「**名前**」ドロップダウン・ボックスと「**言語**」ドロップダウン・ボックスから選択できます。「管理コンソール」ホーム・ページが表示されます。

組織管理コンソールを開く

以下のいずれかの方法で、組織管理コンソールにログオンします。

- 組織管理コンソールは、以下のいずれかの方法で開始します。
 - ブラウザーで以下の URL にアクセスします。
`https://host_name:8004/orgadminconsole`
ここで *host_name* は、 WebSphere Commerce Server の完全修飾 TCP/IP 名です。
- WebSphere Commerce Studio をインストールしてあり、このマシンから組織管理コンソールを開く場合は、以下のようになります。
 1. WebSphere Commerce Studio 内で WebSphere Commerce を開始してあることを確認します。詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。
 2. ブラウザーで以下の Web アドレスにアクセスします。
`https://host_name/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=buyerconsole.BuyAdminConsoleLogon`
ここで *host_name* は、 WebSphere Commerce Studio マシンの完全修飾 HTTP ホスト名です。
- 組織管理コンソールにログオンするには、以下のようになります。
 1. 「ログオン」ページで組織管理コンソールのログオン・ユーザー名とパスワードを入力して、「ログオン」をクリックします。複数のストアまたは言語を扱う権限がある場合は、「組織管理コンソール」ホーム・ページが表示されます。実行する権限があるタスクは、「組織管理コンソール」ホーム・ページに表示されます。
- 組織管理コンソールにログオンしてパスワードを変更するには、以下のようになります。
 1. 「ログオン」ページで組織管理コンソールのログオン・ユーザー名とパスワードを入力します。
 2. 「パスワード変更」チェック・ボックスを選択して、「ログオン」をクリックします。「パスワード変更」ページが表示されます。

3. 「旧パスワード (Old password)」フィールドに、組織管理コンソールの現在のログオン・パスワードを入力します。このフィールドには、最大 128 文字の英数字を入力できます。
4. 「新規パスワード」フィールドに、新規ログオン・パスワードを入力します。このフィールドには、最大 128 文字の英数字を入力できます。
5. 「新規パスワードの確認 (New password confirmation)」フィールドに、新規ログオン・パスワードを再入力します。
6. 「変更」をクリックして、新規パスワードを保管します。「組織管理コンソール」ホーム・ページが表示されます。実行する権限があるタスクは、「組織管理コンソール」ホーム・ページに表示されます。

WebSphere Commerce コンポーネントを使用可能、または使用不可にする

管理者として、WebSphere Commerce の特定のコンポーネントを特定の時期にのみ使用可能にしたいという場合があります。顧客によってニーズが異なるため、WebSphere Commerce では、管理コンソールの「構成」メニューを使用して、WebSphere Commerce コンポーネントを必要に応じて使用可能にしたり使用不可にしたりすることができます。

WebSphere Commerce コンポーネントを使用可能または使用不可にするには、以下のようにします。

1. 管理コンソールを開いて、サイト管理者としてログオンします。
2. 「構成」メニューから、「コンポーネント構成」をクリックします。サイトで使用可能なすべてのコンポーネントがリスト表示されたページが表示されます。
3. コンポーネントを使用可能にするには、以下のようにします。
 - a. 「使用可能なコンポーネント (Available components)」リストから使用可能にするコンポーネントを選択して、「追加」をクリックします。
 - b. 使用可能にするすべてのコンポーネントが「選択済みのコンポーネント (Selected components)」リストに入っていることを確認して、「OK」をクリックします。

注: コンポーネントを使用可能にする前に、そのコンポーネントについての資料を参照して、コンポーネントが正しく構成されていることを確認します。

4. コンポーネントを使用不可にするには、以下のようにします。
 - a. 「選択済みのコンポーネント (Selected components)」リストから使用不可にするコンポーネントを選択して、「除去」をクリックします。
 - b. 使用不可にするすべてのコンポーネントが「使用可能なコンポーネント (Available components)」リストに入っていることを確認して、「OK」をクリックします。

WebSphere Commerce Payments の開始と停止

WebSphere Commerce Payments は、オンライン・マーチャント用のプロトコル独立型支払トランザクション・サーバーです。これは、サイトにレジのような機能を提供し、プロトコル固有のカセットを使用して複数の支払メソッドをサポートします。これらのカセットは、Payments フレームワークに接続可能なソフトウェア・コ

ンポーメントで、汎用支払コマンドと管理コマンドを支払プロトコル固有の要求に解釈します。次いでこの要求は、決済機関の Payment Gateway などの適切な宛先に転送されます。最終結果は、従来のストアでレジ係がチェックアウト・カウンターに支払カードを通して読ませるときと似ています。Payments は、インターネットを介した支払いのすべてのバックグラウンド詳細情報をマーチャント（またはストア・グループなどのマーチャントの一团）に代わって処理し、取引の管理を単純化するグラフィカル・インターフェースを提供します。Payments は、以下の機能を実行します。

- 始動時にカセットに依存して、決済機関のすべての証明書が揃っていることを確認します。足りない証明書があれば、決済機関または Payment Gateway から自動的に要求されます。
- 専用ポートを listen し、ウォレットとマーチャント・サーバーからのすべての着信要求を処理して、永続的に稼働します。
- TCP/IP プロトコルまたはカセットに依存した他のプロトコルを使用して、インターネットを介して決済機関と通信します。
- 取引の状況、承認要求、およびデポジット要求についての情報を保守するデータベース・テーブルを使用します。これらのテーブルのレコードは、作業用、トラッキング用、および記録用に保持されます。データベースには、構成テーブルも含まれます。このテーブルに保存されるのは、マーチャントの情報、支払カードのブランドの情報、およびマーチャントに対してカード与信と支払資金化サービスを行う決済機関についての情報です。

マーチャント・サーバーは、Payments API を使用することで Payments と対話します。これらの API は十分に汎用性のある設計となっており、Secure Electronic Transactions (SET™) および他の支払テクノロジーをサポートします。


Payments のサービスの開始と停止は、管理者が行う必要があります。支払エンジンを開始および停止するには、Payments のインストール先のディレクトリーに .payment ファイルを置いておく必要があります。 .payment ファイルは、Payments のインストール中に作成されます。

WebSphere Commerce Payments の開始

WebSphere Commerce Payments を開始するには、以下のようにします。

1. WebSphere Application Server 管理コンソールを開始して、WebSphere Payments アプリケーション・サーバーが開始していることを確認します。
2. 以下のようにしてログインします。

    データベース・インスタンス ID でログインします。

 管理者のグループ特権を持つユーザーとしてログインします。

3. WebSphere Commerce Payments のインストール先のディレクトリーに移動します。デフォルトのインストール場所は、以下のとおりです。

 /usr/lpp/WebSphere/CommerceServer55/payments

 /QIBM/ProdData/CommercePayments/V55

  /opt/WebSphere/CommerceServer55/payments

 drive:¥Program Files¥IBM¥WebSphere Commerce Payments

4. 以下のコマンドを入力します。

   ./IBMPayServer

 IBMPayServer

 このコマンドの実行後にパスワードを指定する必要があります。

注: WebSphere Commerce Payments の構成時に「パスワードを要求 (Password Required)」を選択した場合は、 WebSphere Commerce Payments を完全に初期化するには、 IBMPayServer コマンドを呼び出す必要があります。

IBMPayServer コマンドを使用して、データベース内に保管される機密データを暗号化解除するために使用する WebSphere Commerce Payments パスワードを指定します。このコマンドについての詳細は、「WebSphere Commerce オンライン・ヘルプ」の『Issuing the IBMPayServer command』を参照してください。 WebSphere Commerce Payments の構成について詳しくは、 81 ページの『WebSphere Commerce Payments の構成』を参照してください。 WebSphere Commerce Payments の構成についてのさらに詳細な情報は、 175 ページの『付録 C. WebSphere Commerce Payments チュートリアル』を参照してください。

WebSphere Commerce Payments の停止

WebSphere Commerce Payments を停止するには、以下のようにします。

- WebSphere Commerce Payments を停止します。
- WebSphere Application Server の下の Payment Manager アプリケーション・サーバーを停止します。

WebSphere Commerce Payments を停止するには、 22 ページの『WebSphere Commerce Payments の開始』のステップ 1 ~ 3 を実行し、ステップ 4 で以下のコマンドを使用します。

   ./StopIBMPayServer

 StopIBMPayServer

注: 構成マネージャーまたは WebSphere Commerce Studio を使って Commerce Payments を停止する方法については、「WebSphere Commerce オンライン・ヘルプ」の『WebSphere Commerce Studio での WebSphere Commerce Payments の開始と停止』および『構成マネージャーからの WebSphere Commerce Payments の開始と停止』を参照してください。

インストール・ガイドの管理用タスク

以下の管理用タスクについては、「WebSphere Commerce インストール・ガイド」に記載されています。

- WebSphere Commerce インスタンスまたは Commerce Payments インスタンスの変更
- WebSphere Commerce インスタンスの削除
- その他のコマンド行構成タスク
- リモート DB2 データベースのカタログ
- WebSphere Commerce の暗号化されたパスワードの生成
- WebSphere Commerce Payments の暗号化されたパスワードの生成

第 3 章 構成

このセクションでは、WebSphere Commerce サイトの機能を最大限に引き出すために、サイト管理者が使用する構成ツールおよびタスクについて説明します。

構成サービス

アーキテクチャー上の構成

サイト管理者は、WebSphere Commerce が機能する構成を実装するために、設計者と協力する必要があります。たとえば、WebSphere Commerce および必要な一群のアプリケーション (Web サーバー、WAS、データベース、および Commerce Payments) は、すべて物理的に同じマシンまたはノード上にあっても構いません。これを単一ノード・アーキテクチャーと言います。あるいは、WebSphere Commerce のさまざまな層 (ソフトウェア・プロダクト) は、異なる物理ノード (マシン) 上にあっても構いません。これをマルチノード・アーキテクチャーと言います。ソリューションの複雑さによっては、さまざまなマシンまたは複製の間で負荷を分散する、クラスター化環境とすることもできます。サイト管理者は、さまざまな開発段階 (テスト、ステージング、および実動など) で必要となる、いくつかのサーバー構成の設定も担当する場合があります。WebSphere Commerce で使用可能な他の構成の詳細については、「*WebSphere Commerce インストール・ガイド*」を参照してください。

メッセージング・サービス

管理コンソールのメッセージング・サービス・インターフェースを使用することで、サイト管理者は、サイトおよびストア用のメッセージの送付をセットアップしたり管理したりすることができます。WebSphere Commerce メッセージング・システムを使用すれば、WebSphere Commerce で生成されるメッセージの定義と送信のすべての局面を管理できます。メッセージング・システムは、E メールやプレーン・ファイルなどのトランスポートを使用して、メッセージを送信します。Eメールの場合、サポートされるアウトバウンド・プロトコルは SMTP です。メッセージのエンコード方式は、指定した言語によって異なります。プレーン・ファイルのメッセージでは、UTF-8 エンコード標準が使用されます。WebSphere MQ および IBM CrossWorlds[®] を使用してメッセージをバックエンド・システムに送信するよう、メッセージング・システムを構成することもできます。また、アウトバウンド・メッセージを送信し、一定の時間応答メッセージを待つよう、メッセージング・サービスを構成することもできます。アウトバウンド・メッセージング・システムを構成するには、管理コンソールの「構成」メニューを使用します。

WebSphere Commerce は、Java 2 Enterprise Edition Connector Architecture (J2EE/CA) を実装するプラグイン・モデルを使用して、システムとさまざまなトランスポートとの間の共通インターフェースを提供します。外部トランスポートとの一般インターフェースを使用すると、トランスポートの実装の詳細を、メッセージング・システムの操作と分けておくことができます。このアーキテクチャーにより、J2EE/CA インターフェース準拠の追加トランスポートのプラグインが可能となります。このようにして、ビジネス・ニーズと環境に適合するよう、ソリューションを簡単にカスタマイズできます。

サービスのスケジューリング

スケジューラーは Commerc Server のコンポーネントであり、主にタイミング・スキームに基づいたジョブをスケジュールしたり起動したりするために使用されます。スケジュールされた各ジョブは、別々のスレッドとして実行されます。複数ジョブの同時実行をスケジュールすることは可能ですが、ワークロードを制御し、長時間のアイドル時間を避けるため、ジョブを分散させることをお勧めします。ジョブとは、指定した時刻または間隔で実行されるようスケジュールされた WebSphere Commerce のコマンドのことです。ジョブのタイミングを指定するには、AddJob コマンドのコマンド開始パラメーターと間隔パラメーターを使用します。AddJob コマンドについて詳しくは、「WebSphere Commerce オンライン・ヘルプ」の『AddJob コマンド』を参照してください。

スケジューラーは、以下の場合に実行しておく必要があります。

- IBM WebSphere Commerce Payments の実行中。
- オークションの実行中。
- ルール・サービスの実動環境への伝搬中。
- サイト管理者がシステムを変更し、クラスターの各インスタンスを再始動せずに変更をすべてのクローンに伝えたい場合。
- 販売可能在庫数量の確認による納期回答 (ATP) 機能を使用する場合。
- サイトで処理済みメッセージを送信する場合。

これらのいずれの状況も当てはまらない場合は、サイト管理者は、スケジューラーを使用不可にしても構いません。

ロギング・サービス


WebSphere Commerce サーバーでのロギング・メッセージの目的は、WebSphere Commerce アプリケーションで予期しないエラーや異常条件が発生する場合に、管理者に通知することです。メッセージのログやトレースは、サイト管理者が問題の源を判別するのを助ける重要な診断ツールです。トレースは、問題判別の手段です。トレースは、開発者が開発段階でコードをデバッグしたり、技術サポート・チームが顧客の問題を解決したりするときに役立ちます。WebSphere Commerce からのロギングは、WebSphere Application Server のロギング機能を利用するので、ログ・アナライザーを使用できます。ログ・アナライザーは、ログ・ファイルの表示と分析を容易にするグラフィカル・ユーティリティーです。ログ・アナライザーについて詳しくは、「WebSphere Application Server InfoCenter」を参照してください。

WebSphere Commerce は、ロギングのための機能を提供します。既存の顧客については引き続き、ECTrace および ECTrace をサポートします。新規インプリメンテーションについては、WebSphere Application Server によって提供され、WebSphere Commerce アプリケーションで使用できる JRAS を使用することをお勧めします。

WebSphere Commerce ログ API は、以下のとおりです。

- ECTrace は、データ・フローをトレースします。デバッグのためにトレース・エントリーがファイルに取り込まれます。

- `ECMessage` は診断メッセージを記録します。メッセージはロケールに依存し、ファイル・システムのログ・ファイルに保管されます。エラー通知が使用可能になっている場合、技術サポートはアラート通知を受け取ります。問題判別のために診断ログが使用されます。デフォルトでは、ログ・ファイル名は `activity.log` です。
- `JRas` は、メッセージ・ロギングと診断トレースのプリミティブを提供するスタンドアロン・ロギング・ツールキットです。

デフォルトの出力ファイルの場所は、`WAS_installdir/logs/WC_<instance name>/`です。  `WAS_userdir/logs/WC_instance_name/` です。デフォルトの出力ファイルは、以下のとおりです。

- `native_stderr.log` は、`STDERR` ストリームに書き込まれるテキストを含むプロセス・ログです。
- `native_stdout.log` は、`STDOUT` ストリームに書き込まれるテキストを含むプロセス・ログです。
- `startServer.log` は、サーバーを始動するときのログです。
- `stopServer.log` は、サーバーを停止するときのログです。
- `SystemErr.log` は、サーバー稼働中のシステム・エラーをすべて記録します。
- `SystemOut.log` は、サーバー稼働中のシステム出力ファイルを記録します。
- `activity.log` は、継続するアクティビティを記録します。このログは、`WAS_installdir/logs` ディレクトリにあります。

WebSphere Application Server ログ API について詳しくは、「WebSphere Application Server InfoCenter」にあるロギングとトレースに関するセクションを参照してください。

ECMessage

問題判別のために診断ログが使用されます。 `ECMessageLog` は診断メッセージを記録し、`ECMessages` はローカライズされます。 `ECMessages` は以下のカテゴリで構成されています。

- システム・メッセージ
- ユーザー・メッセージ

システム・メッセージとユーザー・メッセージ: システム・メッセージはログに現れます。これは問題をデバッグするためのものです。システム・メッセージは、サイト管理者に診断情報を提供します。これらのメッセージはシステムの誤動作を追ったり、他の重大なイベントを示す場合があります。

システム・メッセージには、製品固有のメッセージ ID が割り当てられます。

`CMNnnnns`

ここで、

`nnnn` は、影響を受ける対象を識別するのに使用されるかぎとなる番号です。

`s:` は、メッセージの重大度です。

ユーザー・メッセージは頻繁にブラウザーに表示されます。これはサイトにアクセスする顧客のためのものです。ユーザー・メッセージは、問題に関する詳細を提供

します。たとえば、指定したパラメーターが無効かどうかを示しますが、これは、要求を送信するときに修正すべき値を顧客に示します。サイト管理者は、メッセージ ID を使用して、そのメッセージに関する詳細を検索することができます。顧客はメッセージを報告することによって、トラブルシューティングの担当者をサポートできます。

ロギング・レベル: WebSphere Commerce ロギング・システムには、5 つのロギング・レベル、つまり重大度があります。システム・メッセージの重大度は、エラー、状況、警告、通知、デバッグの 5 つです。

- エラー・メッセージは、デフォルトで常時記録されます。エラー・メッセージは、システムの誤動作につながりかねないエラー状態を明らかにします。エラー・メッセージは、E メールでの WebSphere MQ メッセージとして、または別の通知形式によって、メッセージングに登録されたサイト管理者に送信することができます。
- 状況メッセージは、WebSphere Commerce が到達する特定の状態を示します。たとえば、各コンポーネントは、ロード時、またはメッセージが特定の状態に移ったときに、メッセージを保持します。状況メッセージは、技術サポートがコンポーネントおよびアプリケーションの状態を理解するのを助けます。
- 警告メッセージは、潜在的な問題を明らかにします。
- 通知メッセージは、WebSphere Commerce システム内で発生するイベントを追います。通知メッセージは、システム状態の変更をトリガーするイベントと関係があります。たとえば、オーダーが送信されたときに、通知メッセージが保持されます。
- デバッグ・メッセージは、トレース・コンポーネントとオーバーラップします。ただし、デバッグ・メッセージによって、サイト管理者は、技術サポートに頼ることなく、問題そのものを調査することができます。

WebSphere Commerce は WebSphere Application Server ロギング機能を使用しているので、WebSphere Application Server には 3 種類のロギング・レベルしかないため、WebSphere Commerce ロギング・レベルは次のように WebSphere Application Server にマップされます。

表 3.

WebSphere Commerce のロギング・レベル	WebSphere Application Server のロギング・レベル
ERROR/ERR	TYPE_ERROR/TYPE_ERR
INFORMATION/INFO	TYPE_INFORMATION/TYPE_INFO
DEBUG	TYPE_INFORMATION/TYPE_INFO
WARNING/WARN	TYPE_WARNING/TYPE_WARN
STATUS	TYPE_INFORMATION/TYPE_INFO

ロギング・レベルまたは重大度タイプの使用可能化の制御に、WebSphere Application GUI が使用されます。サイト管理者は WebSphere Application Server 管理コンソールを使って、記録するロギング重大度を指定できます。ロギング・レベルの使用可能化について詳しくは、「WebSphere Application Server InfoCenter」を参照してください。

ECTrace

トレースは、問題判別のために使用されます。トレースは、開発者が開発段階でコードをデバッグしたり、技術サポート・チームが顧客の問題を解決したりするとき役に立ちます。

トレース・データは将来の参照のためにトレース・ファイルに保持されます。データ構造は、クラス名、メソッド名、およびテキスト・メッセージなどのコンテキスト情報によって構成されます。複数のデータ構造が、ソフトウェア・アプリケーション内のデータ・フローを記述します。データ構造のシーケンスを分析することにより、開発者は実行されたパスを知ることができます。これは、誤動作の原因の判別に役に立ちます。

JRas

JRas は、メッセージ・ロギングと診断トレースのプリミティブを提供する複数の java パッケージで構成されています。これらのプリミティブは、どの特定の製品またはプラットフォームとも結び付いていません。JRas は基本的に、複数のコンポーネントから成ります。

- **ロガー:** ロガーは、ユーザー・コードが対話する基本オブジェクトです。ロガーには、メッセージ・ロガーとトレース・ロガーの 2 種類があります。メッセージ・ロガーはメッセージ・レコードのみを作成し、トレース・ロガーはトレース・レコードを作成します。ロガーには、さらに処理を行うためのイベントの転送先となる 1 つ以上のハンドラーが含まれます。
- **ハンドラー:** ハンドラーはロガーからイベントを受け取り、出力装置またはイベント・コンシューマーを介した抽象を設けます。一例として、ファイル・ハンドラーは、イベントをファイルに書き出すことができます。
- **フォーマッター:** ハンドラーはフォーマッターとともに構成されます。これは、特定のタイプのイベントをフォーマットすることができます。
- **イベント・タイプ - メッセージとトレース**には固有の事前定義イベント・タイプがあります。
- **イベント・クラス - スタンドアロン JRas ロギング・ツールキット**は、メッセージとトレースの両方のイベント・クラスを定義します。

注: JRas ロギング・ツールキットについて詳しくは、「WebSphere Application Server InfoCenter」にある「WebSphere Application Server JRas」の資料を参照してください。

WebSphere JRas 拡張機能: WebSphere Application Server ランタイムに統合するため、または J2EE 環境で使用するために、WebSphere は拡張クラスのセットを提供します。WebSphere Commerce からのロギングは WebSphere Application Server ロギング機能を利用しますが、これらの拡張クラスは、異なる WebSphere 製品から生成される、より相関性の高いメッセージとトレースを提供します。この拡張クラスのコレクションは、WebSphere JRas 拡張機能と呼ばれます。WebSphere JRas 拡張機能は、適切なロガー・インプリメンテーション・クラスを提供します。これらのメッセージおよびトレースのロガー・クラスのインスタンスは、WebSphere Manager クラスから直接かつ排他的に入手できます。このクラスは `com.ibm.websphere.ras` パッケージにあります。Payments や JCA メッセージング・フレームワークなどの他のコンポーネントも、WebSphere JRas 拡張機能を利用します。WebSphere Commerce および Payments のトレース・コンポーネントに

関連した WebSphere JRes 拡張機能のリストについては、69 ページの『コンポーネントのトレースの使用可能化』を参照してください。

注: WebSphere Commerce は、WebSphere JRes 拡張機能 API を呼び出す ECTMessage および ECTTrace のラッパーを提供しますが、サイト管理者は JRes API を直接呼び出すよう勧められています。

WebSphere Commerce Payments の問題判別

WebSphere Commerce Payments を使用し始めると、実際のアプリケーションの使用時に発生する問題の診断が必要になる場合があります。WebSphere ログ・アナライザーは、エラーとトレースのログを分析して一連のアクションを決定するときに使用できるインターフェースを提供します。Payments エラーおよびトレースのログはログ・アナライザーから入手できます。ただし、以下のメッセージは入手できません。

- サード・パーティーのカセット
- 独立型 Payments の UI
- SampleCheckout
- Cashier

ルール・サービス

WebSphere Commerce は、ルール・サービスを使用して Advisor ルール・サーバーと対話します。ルール・サービスは、2 つのアプリケーション間の通信を促進するインターフェースとして機能します。また、ルール・サービスは、サイトのルール・ベースの部分で、WebSphere Commerce 全体を停止しないで更新するための便利な方法も提供します。ルール・サービスは、キャンペーンとの 1 対 1 の関係の中に存在します。WebSphere Commerce アクセラレーターを使って発行された各キャンペーンにはそれぞれ、対応するルール・サービスがなければなりません。この設計は柔軟性を付与します。というのは、各マーケティング・サービス、そしてキャンペーンは、それぞれ互いのサービスに依存せずに開始、停止、および最新表示ができるからです。WebSphere Commerce アクセラレーターを使って顧客プロフィール、キャンペーン、またはルール・サービスを更新するときは常に、実動サーバー上の適切な場所にファイルを転送した後、ルール・サービスを最新表示する必要があります。それぞれのアプリケーションの複製、または Java 仮想マシンには、ルール・サーバーのローカル・インスタンスがあります。アプリケーションの複製の境界を越えて通信する必要はほとんどありませんが、これが必要なケースがただ 1 つあります。それは、ルール・サービスの状況を更新する場合です。各アプリケーション複製から状況レポートを受け取るには、システムはスケジューラーを使用して要求をブロードキャストし、ルール・サービスの状況を更新します。ユーザーが「状況の表示」ボタンを押すと、結果が収集され、表示されます。

Advisor ルール・サーバー

WebSphere Commerce は Advisor ルール・サーバーを使ってルールを処理し、広告やおすす​​め商品提示商法の技法から成る個別設定されたマーケティング・コンテンツを提供します。このサーバーは WebSphere Commerce Server に統合されています。ルール・サーバーは、現行のショッピング環境に基づいて情報を渡す WebSphere Commerce Server によって呼び出されます。ルール・サーバーは、マーチャントまたはマーケティング担当者によって作成されるルールのコレクションに

対してこの情報を処理し、特定の環境に適した出力をコンパイルします。出力は WebSphere Commerce アクセラレーターを使って定義した基準に基づきます。ルール・サーバーは、ライセンス情報ブックレットに入っている特定の制限によって制御されます。これらの制限を越えるには、HNC Inc. からの別のライセンスが必要です。

LDAP — Lightweight Directory Access Protocol

LDAP は、ディレクトリー・サービスにアクセスするためのクライアント/サーバー・プロトコルです。もともと X.500 のフロントエンドとして使用されていましたが、独立型ディレクトリー・サーバーや他の種類のディレクトリー・サーバーでも使用できます。LDAP は、さまざまなアプリケーション間の情報の共有をサポートするための中央情報リポジトリーとして使用されます。

LDAP 情報モデルは、何らかのオブジェクト (たとえば個人) についての情報が入っているエントリーに基づいています。エントリーは、属性 (タイプと 1 つ以上の値を持つ) で構成されます。各属性には構文があり、それによって属性に指定できる値の種類や、それらの値がディレクトリー操作中にどのように動作するかが決まります。属性の構文の例には、IA5 (ASCII) ストリング、JPEG 写真、u-law 形式でエンコードされた音、URL、PGP 鍵などがあります。一般に、エントリーは識別名 (DN) によって一意的に識別されます。

WebSphere Commerce では、登録済みユーザーに固有 ID が与えられます。これは、識別名と同じ形式のストリングです。WebSphere Commerce データベースをメンバー・リポジトリーとして使用する場合、DN の形式は「uid=logonIDvalue」となり、その後ろにユーザーが属する親組織エンティティの DN が続きます。したがって、ユーザーのログオン ID が「john」で、IBM 組織の中の SWG 組織単位に属している場合、固有 ID は「uid=john, ou=SWG, o=IBM, o=Root Organization」になります。ディレクトリー・サーバーをメンバー・リポジトリーとして使用する場合、登録済みユーザーの固有 ID はディレクトリー・サーバー内にあるそのユーザーの DN になります。この場合、「o=Root Organization」の部分は、ユーザーの DN には含まれません。WebSphere Commerce がサポートするのは、ルート組織の下にはないディレクトリー・サーバー内の既存のユーザー・エントリーだからです。

WebSphere Commerce はディレクトリー・サーバーに組織エントリーを保管することもサポートし、すべての組織エントリーは DN と同じ形式の固有 ID を持ちます。WebSphere Commerce データベースをメンバー・リポジトリーとして使用する場合は、固有 ID の形式は「ou=organizationalEntityName」となり、その後ろに親組織エンティティの DN が続きます。したがって、「Marketing」という組織単位が、IBM 組織の中の SWG 組織単位の中にある場合、固有 ID は「ou=Marketing,ou=SWG,o=IBM,o=Root Organization」となります。ディレクトリー・サーバーをメンバー・リポジトリーとして使用する場合、組織エンティティの固有 ID はディレクトリー・サーバー内にあるその組織エンティティの DN になります。この場合、「o=Root Organization」の部分は、組織エンティティの DN には含まれません。WebSphere Commerce がサポートするのは、ルート組織の下にはないディレクトリー・サーバー内の既存の組織エントリーだからです。また、DN で「o」と「ou」のどちらが使用されるかは、組織エンティティが組織であるか組織単位であるかによって決まります。

ディレクトリー・サーバーをメンバー・リポジトリーとして使用する場合は、ユーザーと組織エンティティをディレクトリー・サーバーに保管できます。さらに、RDN 属性としてどの LDAP 属性を使用するのか、どの WebSphere Commerce 属性を使用して RDN 属性に値を提供するのかを構成できます。標準的な DN は、以下のとおりです。

```
uid=jsmith, l=Toronto, st=Ontario, c=CA
```

ここで、

uid これは、ユーザーの固有 ID です。この例の uid 属性は、RDN 属性として知られています。これは、DN が *l=Toronto, st=Ontario, c=CA* である親エントリーの下での LDAP エントリーを一意的に識別します。WebSphere Commerce では、認証のユーザー確認のための質問がログオン ID とパスワードの場合は、WebSphere Commerce は、RDN = logonID を持つノードを求めて特定の検索スペースの下での LDAP サーバーを検索します。X.509 証明書がユーザー確認のための質問形式の場合、WebSphere Commerce では、証明書のサブジェクト名が RDN 属性値として使用されます。

l ユーザーの地域または市区町村。

st ユーザーの都道府県。

c ユーザーの国または地域。

LDAP の追加情報については、以下の IBM(R) Redbooks を参照してください。

- 「*LDAP Implementation Cookbook*」
- 「*Understanding LDAP*」

Redbooks は、<http://www.redbooks.ibm.com> で入手可能です。

WebSphere Commerce Server と LDAP サーバーとの間で SSL を使用する場合は、LDAP サーバーの説明書を参照して、セットアップを正しく行い (たとえば、鍵リング・ファイルを作成する)、`instancename.xml` ファイルの中の `JNDIEnvPropNameX` 属性と `JNDIEnvPropValueX` 属性を使用して、WebSphere Commerce Server に必要な環境プロパティを指定します。

LDAP による認証

ユーザー・プロファイル情報は、WebSphere Commerce データベースまたはディレクトリー・サーバーに保管できます。認証オプションは、以下のとおりです。

- LDAP

ユーザーは、自分の DN または RDN の値とパスワードでログオンできます。RDN を使用する場合は、`ldapentry.xml` ファイルで構成された検索ベースを使用してユーザーが検索されます。ユーザーは、LDAP サーバーに対して認証されます。LDAP サーバー上のユーザー情報は、ランタイム操作で使用できるよう WebSphere Commerce データベースに複製されます。

- データベース

ユーザーは、ユーザーが提供するログオン ID とパスワードを使用して WebSphere Commerce データベースに対して認証されます。

- その他

サード・パーティーのインターフェースを使用して、WebSphere Commerce データベース内のストア・プロファイル・データを認証します。

認証モードを指定するには、構成マネージャーの「メンバー・サブシステム」ページを使用します。X.509 証明書は、LDAP 認証またはデータベース認証のどちらでも使用できます。X.509 証明書を使用する場合は、Web サーバーがユーザーの認証を行います。この場合、認証モード設定によって、プロファイル・データの保管先 (LDAP か WebSphere Commerce データベース) が決まります。

LDAP レジストリー

WebSphere Commerce では、LDAP または WebSphere Commerce データベースをメンバー・リポジトリとして使用できます。ユーザー・エンティティーと組織エンティティーは、LDAP サーバーに保管できます。現在、メンバー・グループは、WebSphere Commerce データベースにしか保管できません。LDAP をレジストリーとして使用する場合、これは *instancename.xml* ファイル内の `AuthenticationMode=LDAP` と `ProfileDataStorage=LDAP` という設定に対応します。

データは LDAP サーバーと WebSphere Commerce データベースとの間で複製されます。複製するデータを構成するには、*ldapentry.xml* ファイルを使用します。

以下の表は、WebSphere Commerce で提供される LDAP サービスのレベルを示しています。

表 4. ユーザーの登録

	LDAP をレジストリーとして使用	LDAP のサポートなし
WebSphere Commerce を介した新規ユーザーの登録または登録済みユーザー情報の更新	ユーザー情報は、WebSphere Commerce データベースと LDAP サーバーの両方で作成または更新されます。	ユーザー情報は、WebSphere Commerce データベースで作成または更新されます。
LDAP ユーザーが、別の LDAP アプリケーションから情報を登録または更新する	ユーザーが WebSphere Commerce Server にログオンする場合、またはユーザー情報が WebSphere Commerce Server 内のロジックで必要な場合にのみ、ユーザー情報が WebSphere Commerce データベースに複製されます。	適応されない
LDAP サーバーの可用性	LDAP サーバーは常に使用可能でなければなりません。	適応されない
LDAP 接続の失敗	ユーザーにエラー・ページが表示されます。 エラー・メッセージは、システム・ログ・ファイルに記録されます。	適応されない

表 5. 組織エンティティの登録

	LDAP をレジストリーとして使用	LDAP のサポートなし
WebSphere Commerce を介した新規組織エンティティの登録または組織エンティティ情報の更新	組織エンティティ情報は、WebSphere Commerce データベースと LDAP サーバーの両方で作成または更新されます。	組織エンティティ情報は、WebSphere Commerce データベースで作成または更新されます。
組織エンティティ情報は、別の LDAP アプリケーションで作成または更新されます。	組織エンティティ情報が WebSphere Commerce Server 内のロジックで必要な場合にのみ、組織エンティティが WebSphere Commerce データベースで作成または更新されます。	適応されない

登録が承認保留となっているユーザーも、ディレクトリー・サーバーに保管されません。コンテキスト依存の属性は、デフォルトではディレクトリー・サーバーに保管されません。

ユーザーの登録住所だけが LDAP に持続されます。LDAP のユーザーの住所録にある友人、親族、または仕事の同僚の住所は持続されません。

注: LDAP サーバーをメンバー・リポジトリーとして選択してある場合は、ローダー・パッケージを使用して、WebSphere Commerce データベース内にユーザー・エ

ンティティーと組織エンティティーの情報を一括ロードしないでください。
WebSphere Commerce 5.5 には、データベースにデータを一括ロードするビジネス・モデルが導入されています。ビジネス・モデルについて詳しくは、「*WebSphere Commerce* ストア開発ガイド」を参照してください。

LDAP 構成パラメーター

使用する LDAP システムのインストール後に、構成マネージャーの「メンバー・サブシステム」ページの該当するフィールド、または LDAP サポートを手動で構成する場合は、*instancename.xml* ファイルの該当するフィールドに入力する必要があります。ここでは、*instancename.xml* ファイルの MemberSubSystem セクションのパラメーターについて説明します。*instancename.xml* ファイルは、WebSphere Commerce の各インスタンスに含まれていることに注意してください。競合を避けるために、LDAP 構成情報はすべてのインスタンスで同じにしてください。

注: 以下にコードの例を示します。ここで使用する値は架空のものであり、この例以外では意味をなしません。実際のファイルでは、パラメーターに使用する有効な値を指定する必要があります。

```
<MemberSubSystem name="Member SubSystem"
    ProfileDataStorage="LDAP"
    AuthenticationMode="LDAP">
<Directory LdapPort="636"
    LdapType="SECUREWAY"
    LdapAuthenticationMode="SIMPLE"
    EntryFileName="d:/WebSphere/CommerceServer55/
xml/ldap/ldapentry.xml"
    LdapAdminPW="EaDPFd9VAf0="
    LdapVersion="3"
    LdapHost="basswood.torolab.ibm.com"
    SingleSignOn="0"
    LdapAdminDN="cn=root"
    display="false"
    MigrateUsersFromWCdb="ON"
    JNDIEnvPropName1="java.naming.referral"
    JNDIEnvPropValue1="ignore"
    JNDIEnvPropName2="java.naming.security.
protocol"
    JNDIEnvPropValue2="ssl"
    ...
    JNDIEnvPropNameN="java.naming.language"
    JNDIEnvPropValueN="en-US"
```

```
LdapTimeOut="0" />
```

```
</MemberSubSystem>
```

パラメーターの説明

ProfileDataStorage

この属性の値は、ユーザー・エンティティと組織エンティティのプロファイル・データの保管先を決定します。値に LDAP を指定しても、一部のプロファイル・データは WebSphere Commerce データベースに複製されません。

AuthenticationMode

この属性の値は、ユーザーが認証されるシステムを決定します。有効な値は次のとおりです。

- LDAP (デフォルト)
- DB
- Other (サード・パーティーの機構用)

LdapAdminDN

LDAP 上で操作を実行するために使用される管理者の DN。この DN のセットアップと許可は、LDAP 上で行う必要があります。

LdapAuthenticationMode

この属性の値は、使用するディレクトリー・サーバーによって決まります。Simple と None は両方とも、IBM SecureWay[®] および Netscape iPlanet で有効です。

- Simple (デフォルト)
- None

LdapTimeOut

LDAP 検索がタイムアウトになる時間 (秒単位)。タイムアウト値は、LDAP サーバー上でも指定できます。両方で値を指定する場合は、値の小さい方が優先されます。LdapTimeOut の値が 0 の場合、検索タイムアウトは LDAP サーバー構成によってのみ決定されます。デフォルト値は 0 です。

LdapVersion

LDAP プロトコルのバージョン番号。有効な値は 3 だけです。

EntryFileName

WebSphere Commerce 属性がどの LDAP 属性にマップされるかについての情報が入っている XML ファイル。
デフォルト値は、以下のとおりです。

 Solaris Linux Windows

WC_installdir/
xml/ldap/ldapentry.xml

 AIX 400

WC_installdir/xml/ldap/ldapentry.xml

LdapPort

LDAP サーバーで使用されるポート。デフォルト値は 389 です。

SingleSignOn

「0」の値は、単一サインオンがオフになっていることを意味します。「1」の値は、オンになっていることを意味します。

LdapAdminPW

デフォルトなし - ブランク

管理者の暗号化されたパスワード。XML が手動で変更される場合、wcs_encrypt.bat または wcs_encrypt.sh を使って生成されます。

LdapHost

LDAP サーバーのホスト名。デフォルト値は、現在のホスト名です。

MigrateUsersFromWCdb

ストア発行プロセスによって作成されるブートストラップ・ユーザーおよびすべての組織を LDAP にプッシュするために、このオプションはオンにしておく必要があります。そうでない場合は、OFF にしてください。デフォルト値は OFF です。

LdapType

使用する LDAP ディレクトリー・サーバー。たとえば、以下のようなものです。

    IBM SecureWay V5.1 (default)



- OS400 - DIRECTORY SERVICES
- IBM SecureWay V3.2.2

JNDIEnvPropNameX および JNDIEnvPropValueX

これらの属性は、WebSphere Commerce Server で設定する JNDI 環境プロパティーを指定する場合に使用します。これらの属性は、instancename.xml ファイル内で手動で指定する必要があります。

WebSphere Commerce 5.1 から、以下の点に変更されていることに注意してください。

- MappingFileName が EntryFileName に変更されました
- ldapmap.xml が ldapentry.xml に変更されました
- LdapPersonRDN、LdapPersonSearchRoot、LdapPersonDefaultBase、LdapPersonOCS、LdapOrgOCS、および LdapOrgUnitOCS が ldapentry.xml ファイルの一部となり、その名前が次のように変更されています。
 - LdapPersonRDN は rdnName に変更されました
 - LdapPersonSearchRoot は searchBase に変更されました
 - LdapPersonDefaultBase は defaultBase に変更されました
 - LdapPersonOCS は objClass に変更されました
 - LdapOrgOCS は objClass に変更されました
 - LdapOrgUnitOCS は objClass に変更されました
- LDAP プロトコル バージョン 3 だけがサポートされるようになりました

ldapentry.xml ファイルと ldapentry.dtd ファイルの詳細については、169 ページの『付録 B. LDAP ファイル』を参照してください。

LDAP 設計上の特徴

WebSphere Commerce での LDAP インプリメンテーションの設計上のいくつかの特徴を以下に示します。

- 複数の WebSphere Commerce 属性を、区切り文字を指定した単一の LDAP 属性に複製できます。
- 複数の LDAP 属性から、単一の WebSphere Commerce 属性に複製できます。これを行うには、ldapentry.xml ファイルの中で複数の <map> セクションを使用します。ただし、上書きされる場合があることに注意してください。
- 単一の WebSphere Commerce 属性を、複数の LDAP 属性に複製できます。
- 単一の LDAP 属性を、複数の WebSphere Commerce 属性に複製できます。
- ユーザーのデフォルトの住所だけが LDAP に持続されます。デフォルトの登録住所は、selfaddress=1 および logonid=nickname という設定でユーザーの住所録に定義されます。ここで、logonID は USEREG テーブルの logonid で、nickname は ADDRESS テーブルの NICKNAME です。ユーザーの住所録にあるその他の住所は、LDAP 上に保管されません。
- WebSphere Commerce から登録されたユーザーだけが LDAP に持続され、ゲスト・ユーザーは持続されません。
- WebSphere Commerce では、LDAP 上のユーザー・エントリーを検索する場合に複数の検索ベースを指定できます。ただし、検索基準を満たすユーザー・エントリーが複数検出される場合は、エラーが発生します。

以下の表は、WebSphere Commerce での LDAP サポートのいくつかの制限を示しています。

制限	説明
LDAP での多値属性	WebSphere Commerce では、多値属性を保管したり検索することはできません。LDAP での多値属性は許容されますが、最初の値しか検索されません。更新時に、LDAP 上に新しい値がすでに存在している場合は、更新は実行されません。存在しない場合は、多値 LDAP 属性の最初の値は更新されます。
LDAP 参照元および参照	WebSphere Commerce では、LDAP 参照元および参照はサポートされません。
複数コンポーネント RDN	RDN 属性として指定できるのは、1 つの LDAP 属性だけです。複数の LDAP 属性を組み合わせることはできません。

ステージング・サーバー

ほとんどのオンライン・ストアは、一日 24 時間、一年 365 日運営されるため、保守を実行したり、システムの変更をテストしたりすることは困難です。WebSphere Commerce ステージング・サーバーを使用することにより、サイト管理者は、実動データベースをステージング・データベースにコピーし、顧客に影響を与えることなく更新をテストすることが可能になります。これは、商品カタログの更新をテストする場合に便利ですが、新しいショッピング処理コマンドをテストする場合にも重要です。

ステージング・サーバーは、以下のコンポーネントで構成されます。

WebSphere Commerce インスタンス

データのテストと変更を行います。

データベース・スキーマ・スクリプト

ステージング・データベースのステージング・テーブルとトリガーを作成します。ステージング・データベースには、実動データベースと同じスキーマおよびテーブルだけでなく、ステージング・データベースで行われる変更を記録するための一連のトリガーが入っています。ステージング・データベース・スキーマ・スクリプトは、データベースにトリガーを追加します。

変更は、データベース・トリガーを使用して STAGLOG テーブル (ステージング・テーブル) に記録されます。ステージング・データベースのデータベース・テーブル・レコードを変更するたびに、変更が STAGLOG テーブルに記録されます。

ステージ・コピー・ユーティリティ

管理者が、実動データベースのデータをステージング・データベースにコピーできるようにします。データは、サイト関連のテーブル、マーチャント関連のテーブル、または個々のテーブルにコピーできます。

ステージ伝搬ユーティリティ

管理者が、ステージング・データベースに行った変更を実動データベースに伝搬できるようにします。STAGLOG テーブル内の情報は、実動データベースで挿入、更新、または削除する必要がある、ステージング・データベースのレコードを識別します。次いで、識別されたレコードが実動データベースで更新されます。処理済みのレコードは、STAGLOG テーブル内の STGPROCESSED 列の 1 という印によって示されます。

ステージ検査ユーティリティ

管理者が、ステージング・サーバーと実動サーバーの 2 つのテーブルの間で、固有索引キーが競合していないかどうかを検査できるようにします。

ステージング・サーバーの制限

ステージング・サーバーを使用する前に、以下の制限事項を知っておく必要があります。

- ステージング・サーバーは、バイヤー組織の自己管理フィーチャーと併用できない。
- (MEMBER、MBRREL、MBRROLE、および MBRATTRVAL を除く) ステージング・テーブルの member_id 列はすべて、ユーザーではなく、組織グループまたはメンバー・グループでなければならない。
- member_id は、すべてのサイト・テーブルで、-2001 または 0 でなければならない。サイト・データに関連する行の member_id は、サイト・データとマーチャント・データの両方を含むすべてのテーブルで、0 または -2001 でなければならない。
- 実動システム上で RFQ フィーチャーを使用している場合は、ステージ・コピー・コマンドを使用できない。ステージ・コピー・コマンドが使用できない場合は、プッシュ・モードを使用しなければなりません。実動サイトを立ち上げる前に、ステージング・サーバーを作成し、ステージング・データベースをセットアップ

ップしてください。ステージング・サーバー上でデータをデプロイし、テストした後、ステージ伝搬コマンドを使用して実動サーバーにプッシュします。

- ステージング・サーバー上では RFQ を作成または更新できない。

キーの分割: 販売店のマーケットプレイスでは、IBM とサイト管理者がステージング・サーバー上のデータを更新し、販売店が実動サーバー上のデータを更新します。これは、基本キーの衝突を引き起こす可能性があります。

WebSphere Commerce は、キー・マネージャーを使用してテーブルの基本キーを生成します。キー範囲は KEYS テーブルで定義されます。実動サーバーとステージング・サーバーが同じキー範囲を使用する場合、キー・マネージャーが実動サーバーとステージング・サーバー両方の同じテーブルに同じ基本キー値を割り当てると、基本キーの衝突が生じます。そうすると、ステージング・インスタンスと実動インスタンスが作成された直後に、キー範囲を分割しなければなりません。こうすることで、ステージング・サーバーと実動サーバーが使用するキー範囲を別にします。

現在、WebSphere Commerce はステージング・サーバーと実動サーバーのキー範囲を分割するために、以下の SQL を使用します。この SQL ステートメントは、現行のキー範囲をそれぞれが同等の 3 つの部分に分けます。

第 1 部

```
update keys set upperbound=(upperbound-lowerbound)/3 + lowerbound
where tablename in (select tablename from stgmertab) or tablename
in (select tablename from stgsitetab)
```

第 2 部

```
db2 update keys set upperbound = (upperbound-lowerbound)/3*2 + lowerbound,
lowerbound = (upperbound-lowerbound)/3 +lowerbound+1, counter =
counter+(upperbound-lowerbound)/3 +1 where tablename in
(select tablename from stgmertab)
or tablename in (select tablename from stgsitetab)
```

第 3 部

```
db2 update keys set lowerbound = (upperbound-lowerbound)/3*2 + lowerbound +1
, counter = counter + (upperbound-lowerbound)/3*2 +1 where tablename in
(select tablename from stgmertab) or tablename in (select tablename from stgsitetab)
```

最初の SQL ステートメント (第 1 部) は、実動サーバー上で実行しなければなりません。実行後、実動サーバーは、全キー範囲のうちの 3 分の 1 を占めるようになります。2 番目の SQL ステートメント (第 2 部) は、ステージング・サーバー上で実行します。実行後、ステージング・サーバーは、全キー範囲のうちの次の 3 分の 1 を占めるようになります。3 番目の SQL ステートメント (第 3 部) は、ステージング・サーバーの予備として確保しておきます。

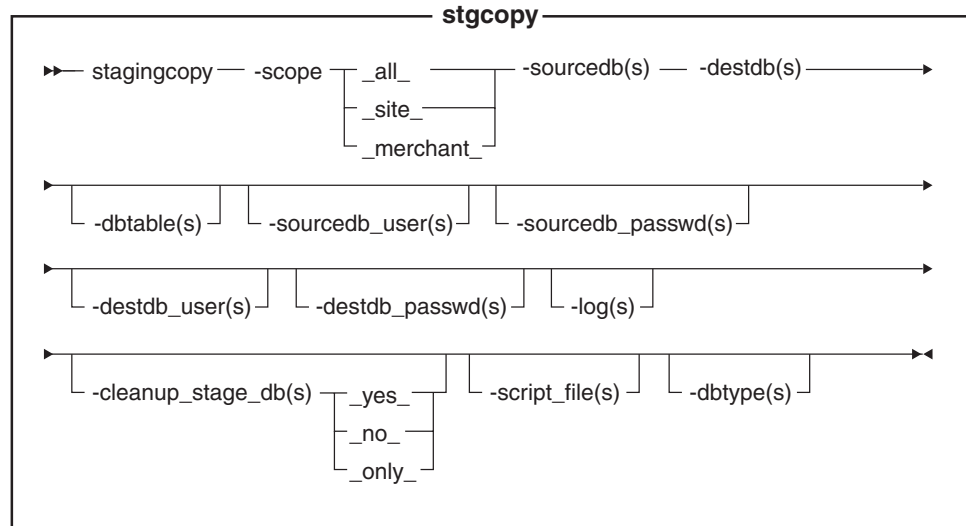
注: 上記の SQL のキーの分割はすでに、インスタンスの構成プロセスに組み込まれていますので、手動で実行する必要はありません。

ステージング・サーバー・コマンド

ステージ・コピー・ユーティリティー・コマンド (Windows 2000、AIX、Linux、iSeries、および Solaris): ステージ・コピー・ユーティリティーは、実動データベースからステージング・データベースにデータをコピーします。RFQ が実動シ

システム上に存在する場合は、このコマンドを使用できません。詳しくは、39 ページの『ステージング・サーバーの制限』を参照してください。

ステージ・コピー・ユーティリティを実行するには、ステージング・サーバーと実動サーバー両方のデータベースに接続可能なマシン上のコマンド行から以下のコマンドを入力してください。すべてのコマンドを 1 行に入力してください。ここでは、表示の都合で複数の行に分けているに過ぎません。



注:

1. **Oracle** 現在このユーティリティを同じユーザー ID を使用して実行している場合でも、オプション・パラメーター、ログオン・ユーザー ID、およびパスワードをコマンドに組み込まなければなりません。
2. **400** iSeries は現在 stagingCopy.sh を使用します。iSeries でシェル・スクリプトを実行するには、以下のようにします。
 - a. 65535 以外の CCSID を持つユーザー・プロファイルとしてログオンします。
 - b. OS/400 行で STRQSH を入力することにより、QSHELL コマンド・ウィンドウをオープンします。
 - c. 以下のようにしてコマンドを実行してください。

/QIBM/ProdData/CommerceServer55/bin/stagingCopy.sh (parameters . . .)

注: ユーザー・パラメーターは必ず指定しなければなりません。

パラメーター値

scope ステージング・サーバーにコピーする際の有効範囲のレベル。以下のいずれかを指定します。

- **_all_**
サイトに関連するレコード、および全マーチャントに関連するレコードの両方をコピーするには、_all_ を入力します。

- `_site_`
サイトに関連するレコードのみをコピーするには、`_site_` を入力します。
- `_merchant_`
全マーチャントに関連するレコードのみをコピーするには、`_merchant_` を入力します。

sourcedb

実動サーバー上のデータベースの名前。

▶ **400** リレーショナル・データベース・ディレクトリーに表示される実動サーバー上のデータベースの名前。

▶ **Oracle** `host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

destdb ステージング・サーバー上のデータベースの名前。

▶ **400** リレーショナル・データベース・ディレクトリーに表示される実動サーバー上のデータベースの名前。

▶ **Oracle** `host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

dbtable

(オプション) コピーする任意の特定のテーブルの名前。レコードが `scope` パラメーターで指定した有効範囲内にある場合は、そのテーブル内のすべてのレコードがコピーされます。有効範囲内にはない場合は、どのレコードもコピーされません。

sourcedb_user

(オプション) ソース・データベースのスキーマを作成したデータベース管理者のログオン ID。これを指定しない場合、現在ユーティリティーを呼び出しているユーザーの ID が使用されます。

▶ **400** (必須) コマース・インスタンスと関連したユーザー・プロファイル。これは、ソース・データベース・スキーマと同じになります。

sourcedb_passwd

(オプション) `sourcedb_user` パラメーターで指定するログオン ID のパスワード。

destdb_user

(オプション) 宛先データベースのスキーマを作成したデータベース管理者のログオン ID。

▶ **400** (必須) コマース・インスタンスと関連したユーザー・プロファイル。これは、宛先データベース・スキーマと同じになります。

注: これを指定しない場合、ユーティリティーを呼び出しているユーザーの ID が使用されます。

destdb_passwd

(オプション) `destdb_user` パラメーターで指定するログオン ID のパスワード。これを指定しない場合、パスワードを入力するように求められます。

+ **log** (オプション) ステージ・コピー・ユーティリティーがそのアクティビティー

+ とエラーを記録するファイルのパスと名前。このパラメーターを指定しない
+ 場合、 stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log という名前のログ・フ
+ ァイルが以下のディレクトリーに作成されます。

▶ AIX ▶ Solaris ▶ Linux WC_installdir/logs

▶ Windows WC_installdir¥logs

▶ 400 WC_userdir/instances

cleanup_stage_db

(オプション) ステージ・コピー・ユーティリティーを使用する前にステージング・テーブルをクリーニングするには、このパラメーターを使用します。
-cleanup_stage_db パラメーターを使用してサイト・データをクリーニングすると、削除カスケードによってマーチャント・データも削除されてしまう可能性があります。サイト・データ、マーチャント・データの順でクリーニングおよびコピーを行う必要があります。Yes がデフォルトです。no を指定すると、ステージング・テーブルから何も削除されません。コピー・データとの競合が起きたり、基本キーまたは固有索引でキーが重複したりすると、コピーは失敗する可能性があります。ステージ・コピーを使用して、実動データベースからデータ・コピーをクリーンアップせず、ステージ・データベースのみをクリーンアップするには、-cleanup_stage_db に only を指定ください。

script file

(オプション) エクスポートとインポートを使用して指定された範囲内で実動データベースからステージング・データベースにコピーするとき、ステージ・コピー・ユーティリティー・コマンドによって生成される SQL スクリプト・ファイルの名前。このスクリプト・ファイルにより、削除ステートメントも生成されます。これは、デフォルト値を使用するか、-cleanup_stage_db に yes を指定した場合にステージング・データベースをクリーニングします。

注: このスクリプト・ファイルは、iSeries ではサポートされません。スクリプトを実行する前に、エクスポートされるテーブルを保管するための十分なディスク・スペースがあるか確認してください。スクリプト・ファイルは、ステージ・コピー・ユーティリティーを呼び出すステージ・コピー・ユーティリティーのディレクトリーにあります。

▶ DB2 スクリプト・ファイルを実行するには、db2 -vtd# -f script_file_name を使用します。

dbtype (オプション) データベースのタイプ (DB2 または Oracle)。デフォルトは DB2 です。

ステージング・サーバーのコピーについては、例を参照してください。

ステージング・データベースへのデータのコピーの例: 以下の例は、実動データベースからステージング・データベースにテーブルをコピーする方法を示しています。RFQ が実動システム上に存在する場合はステージ・コピー・ユーティリティーを使用できない、ということ覚えておいてください。詳しくは、39 ページの『ステージング・サーバーの制限』を参照してください。

すべてのコマンドを 1 行に入力してください。ここでは、表示の都合でコマンドを複数の行に分けているに過ぎません。

例 1

ステージング・データベースをクリーニングした後、`scope` を `all` に設定して実動データベースからステージング・データベースにコピーします。

1. `PATH` 環境変数を設定します。▶ 400 適応されない
2. データベースを構成します。▶ 400 適応されない
3. ログ・ファイルを書き込むディレクトリに移動します。▶ 400 適応されない。
iSeries の場合、ログ・ファイルはデフォルトの以下のディレクトリに入ります。

```
/QIBM/UserData/CommerceServer55/instances/stagingcopy_{sourcedb_user}
_{destdb_user}_{timestamp}.log)
```

4. 次のように入力します。

- ▶ Windows ▶ DB2
`stagingcopy -scope _all_ -sourcedb production_database_name -destdb staging_database_name`

- ▶ Windows ▶ Oracle
`stagingcopy -scope _all_ -sourcedb production_database_name -destdb staging_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2
`. stagingcopy.sh -scope _all_ -sourcedb production_database_name -destdb staging_database_name -sourcedb_user user -destdb_user user`

- ▶ AIX ▶ Solaris ▶ Oracle
`. stagingcopy.sh -scope _all_ -sourcedb production_database_name -destdb staging_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

注: Oracle データベースの名前には、`host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

5. `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。









ログ・ファイルの名前とパスを指定するには、以下のログ・ファイル・パラメータを使用します。

- ▶ Windows ▶ DB2
`stagingcopy -scope _all_ -sourcedb production_database_name -destdb staging_database_name -log log_file_name`

- ▶ Windows ▶ Oracle
`stagingcopy -scope _all_ -sourcedb production_database_name -destdb staging_database_name -log log_file_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

+
+
+

+
+
+
+

-     
 . stagingcopy.sh -scope _all_ -sourcedb *production_database_name* -destdb *staging_database_name* -log *log_file_name* -sourcedb_user *user* -destdb_user *user*
-    . stagingcopy.sh -scope _all_ -sourcedb *production_database_name* -destdb *staging_database_name* -log *log_file_name* dbtype *oracle* -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*





注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

DB2 を使用していて、データベース管理者としてログオンしていない場合は、*-sourcedb_user*、*-sourcedb_passwd*、*-destdb_user*、および *-destdb_passwd* のそれぞれのオプションに値を指定する必要があります。









例 2

ステージング・データベースからマーチャント・テーブルをクリーニングした後、実動データベースからステージング・データベースにマーチャントに関するテーブルをコピーします。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name`
-   `stagingcopy -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`





+
+
+

-      `stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -sourcedb_user user -destdb_user user`
-    `stagingcopy.sh -scope _merchant_ -sourcedb production_database_name -destdb staging_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`









注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

5. `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ステージング・データベースのみをクリーニングするには、*-cleanup_stage_db* パラメーターを次のように指定します。





-   stagingcopy -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only
-   stagingcopy -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

+
+
+
+









-      stagingcopy.sh -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only -sourcedb_user *user* -destdb_user *user*
-    stagingcopy.sh -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

コピー・データのみをコピーするには、*-cleanup_stage_db no* パラメーターを次のように指定します。

-   stagingcopy -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no
-   stagingcopy -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

+
+
+
+

-      stagingcopy.sh -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no -sourcedb_user *user* -destdb_user *user*
-    stagingcopy.sh -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*



注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。



重要: scope を merchant に設定してコピーする際は必ず、最初にサイトの有効範囲のデータをコピーしておいてください。そうしないと、コピーは失敗してしまいます。

例 3






ステージング・データベースからサイト・テーブルをクリーニングした後、実動データベースからステージング・データベースにサイト・テーブルをコピーします。




1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

-   stagingcopy -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name*

-   stagingcopy -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

+
+
+



-      stagingcopy.sh -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -sourcedb_user *user* -destdb_user *user*



-    stagingcopy.sh -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。






5. stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

サイト・テーブルを削除すると、削除カスケードの制限によりマーチャント・テーブルに影響を与える場合があります。ですから、マーチャント・データ、サイト・データの順でクリーニングした後、次のようにデータをコピーしてください。

-   stagingcopy -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only

-  
stagingcopy -scope _merchant_
-sourcedb *production_database_name*
-destdb *staging_database_name*
-cleanup_stage_db only -dbtype oracle -sourcedb_user
user -sourcedb_passwd
password -destdb_user
user -destdb_passwd
password

+
+
+
+

-      stagingcopy.sh -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only -sourcedb_user *user* -destdb_user *user*

- ▶ AIX ▶ Solaris ▶ Oracle stagingcopy.sh -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

注: Oracle データベースの名前には、 *host:port:sid* の形式を使用してください。たとえば、 *myhost:1521:mydb* のようにします。

- ▶ Windows ▶ DB2 stagingcopy -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only

- ▶ Windows ▶ Oracle stagingcopy -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

- + ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2 stagingcopy.sh -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only -sourcedb_user *user* -destdb_user *user*

- ▶ AIX ▶ Solaris ▶ Oracle stagingcopy.sh -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db only dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

注: Oracle データベースの名前には、 *host:port:sid* の形式を使用してください。たとえば、 *myhost:1521:mydb* のようにします。

- ▶ Windows ▶ DB2 stagingcopy -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no











- ▶ Windows ▶ Oracle stagingcopy -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

- + ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2 stagingcopy.sh -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no -sourcedb_user *user* -destdb_user *user*

- ▶ AIX ▶ Solaris ▶ Oracle stagingcopy.sh -scope _site_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*


注: Oracle データベースの名前には、 *host:port:sid* の形式を使用してください。たとえば、 *myhost:1521:mydb* のようにします。

- ▶ Windows ▶ DB2 stagingcopy -scope _merchant_ -sourcedb *production_database_name* -destdb *staging_database_name* -cleanup_stage_db no












-   `stagingcopy -scope _merchant_ -sourcedb
production_database_name -destdb staging_database_name -cleanup_stage_db
no -dbtype oracle -sourcedb_user user -sourcedb_passwd password
-destdb_user user -destdb_passwd password`
- +      `stagingcopy.sh -scope
merchant -sourcedb production_database_name -destdb
staging_database_name -cleanup_stage_db no -sourcedb_user user
-destdb_user user`
- +    `stagingcopy.sh -scope _merchant_ -sourcedb
production_database_name -destdb staging_database_name -cleanup_stage_db
no dbtype oracle -sourcedb_user user -sourcedb_passwd password
-destdb_user user -destdb_passwd password`

注: Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

例 4

`scope` を `all` に設定してクリーニングをし、実動データベースからステージング・データベースにコピーするために、以下のスクリプトを生成します。  この例は、OS/400 for iSeries には適用されません。というのは、`-script` オプションは iSeries でサポートされないからです。

1. `PATH` 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

-   `stagingcopy -scope _all_ -sourcedb
production_database_name -destdb staging_database_name -script_file
stage_copy.sql`
-   `stagingcopy -scope _all_ -sourcedb
production_database_name -destdb staging_database_name -script_file
stage_copy.sql -dbtype oracle -sourcedb_user user -sourcedb_passwd
password -destdb_user user -destdb_passwd password`
- +     `stagingcopy.sh -scope _all_
-sourcedb production_database_name -destdb staging_database_name
-script_file stage_copy.sql -sourcedb_user user -destdb_user user`
-    `stagingcopy.sh -scope _all_ -sourcedb
production_database_name -destdb staging_database_name -script_file
stage_copy.sql dbtype oracle -sourcedb_user user -sourcedb_passwd
password -destdb_user user -destdb_passwd password`

注: Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

5. `stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ステージ・コピー・ユーティリティーは、データベースをクリーニングしてコピーする `stage_copy.sql` スクリプトを生成します。 **DB2** DB2 を使用している場合は、以下のスクリプトを実行します。

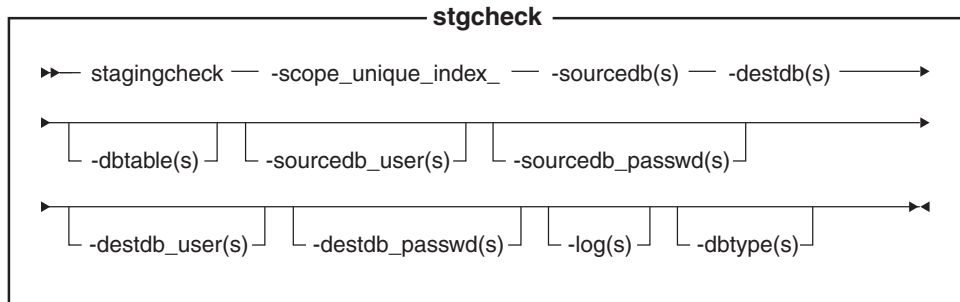
1. データベース管理者 (DBA) としてログオンします。
2. DB2 コマンド・ウィンドウを開きます。
3. `db2 -vtd# -f stage_copy.sql` と入力します。

Oracle Oracle を使用している場合は、以下のスクリプトを実行します。

1. SQLPlus ウィンドウを開きます。
2. `dba` として接続します。
3. `@stage_copy.sql` と入力します。

ステージ検査コマンド (Windows 2000, AIX, iSeries, Linux, および Solaris):

ステージ検査コマンドは、ステージング・データベースと実動データベースとの間に固有索引キーの競合があるかどうかを判別します。ステージ検査ユーティリティーを実行するには、ステージング・サーバーか実動サーバー上のコマンド行から以下のコマンドを入力してください。すべてのコマンドを 1 行に入力してください。ここでは、表示の都合で複数の行に分けているに過ぎません。



Oracle 注: 現在このユーティリティーを同じユーザー ID を使用して実行している場合でも、オプション・パラメーター、ログオン・ユーザー ID、およびパスワードをコマンドに組み込まなければなりません。

400 iSeries は現在 `stagingCheck.sh` を使用します。iSeries でシェル・スクリプトを実行するには、以下のようになります。

1. 65535 以外の CCSID を持つユーザー・プロファイルとしてログオンします。
2. OS/400 行で STRQSH を入力することにより、QSHELL コマンド・ウィンドウをオープンします。
3. 以下のようにしてコマンドを実行してください。

```
/QIBM/ProdData/CommerceServer55/bin/stagingCheck.sh (parameters . . .)
```

注: ユーザー・パラメーターは必ず指定しなければなりません。

パラメーター値

scope ステージング・サーバーにコピーする際の有効範囲のレベル。
_unique_index_ を指定します。

sourcedb

ステージング・サーバー上のデータベースの名前。

▶ Oracle `host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

destdb

実動サーバー上のデータベースの名前。

▶ Oracle `host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

dbtable

(オプション) 固有キーの競合が検査される特定のテーブルの名前。

sourcedb_user

+ (オプション) ステージング・データベースのスキーマを作成したデータベース
+ 管理者のログオン ID。これを指定しない場合、現在ユーティリティを
+ 呼び出しているユーザーの ID が使用されます。

+ ▶ 400 (必須) コマース・インスタンスと関連したユーザー・プロファイル。
+ これは、ソース・データベース・スキーマと同じになります。

sourcedb_passwd

(オプション) sourcedb パラメーターで指定するログオン ID のパスワード。

destdb_user

+ (オプション) 実動データベースのスキーマを作成したデータベース管理者の
+ ログオン ID。これを指定しない場合、ユーティリティを呼び出している
+ ユーザーの ID が使用されます。

+ ▶ 400 (必須) コマース・インスタンスと関連したユーザー・プロファイル。
+ これは、宛先データベース・スキーマと同じになります。

destdb_passwd

(オプション) destdb_user パラメーターで指定するログオン ID のパスワード。これを指定しない場合、パスワードを入力するように求められます。

log

+ (オプション) ステージ・コピー・ユーティリティがそのアクティビティ
+ とエラーを記録するファイルのパスと名前。このパラメーターを指定しない
+ 場合、`stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log` という名前のログ・フ
+ ァイルが以下のディレクトリーに作成されます。

+ ▶ AIX ▶ Solaris ▶ Linux `WC_installdir/logs`

+ ▶ Windows `WC_installdir\logs`

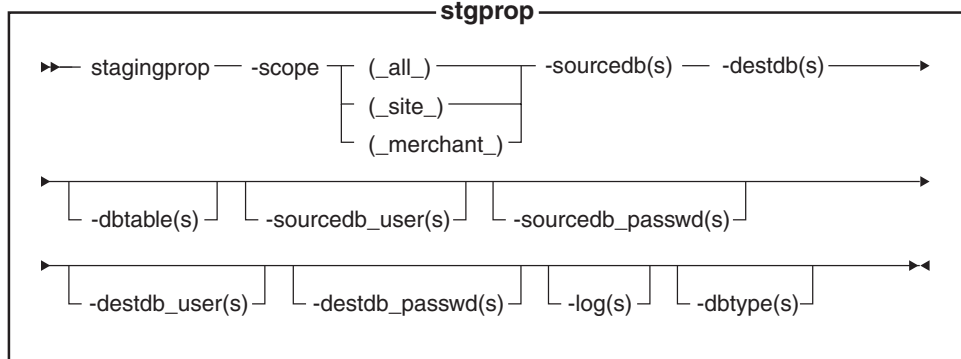
+ ▶ 400 `WC_userdir/instances`

dbtype (オプション) データベースのタイプ (DB2 または Oracle)。デフォルトは DB2 です。

固有索引キーの競合の検査について詳しくは、例を参照してください。

ステージ伝搬ユーティリティ・コマンド (Windows 2000, AIX, iSeries, Linux, および Solaris): ステージ伝搬ユーティリティは、ステージング・データベースから実動データベースにデータをコピーします。ステージング・サーバー上に RFQ オブジェクトを作成したり、それを更新したりすることはできません。

ステージ伝搬ユーティリティを実行するには、ステージング・サーバーと実動サーバー両方のデータベースに接続可能なマシン上のコマンド行から以下のコマンドを入力してください。すべてのコマンドを 1 行に入力してください。ここでは、表示の都合で複数の行に分けているに過ぎません。



▶ Oracle

注: 現在このユーティリティを同じユーザー ID を使用して実行している場合でも、オプション・パラメーター、ログオン・ユーザー ID、およびパスワードをコマンドに組み込まなければなりません。

▶ 400 iSeries は現在 stagingProp.sh を使用します。iSeries でシェル・スクリプトを実行するには、以下のようにします。

1. 65535 以外の CCSID を持つユーザー・プロファイルとしてログオンします。
2. OS/400 行で STRQSH を入力することにより、QSHELL コマンド・ウィンドウをオープンします。
3. 以下のようにしてコマンドを実行してください。

/QIBM/ProdData/CommerceServer55/bin/stagingProp.sh (parameters . . .)

注: ユーザー・パラメーターは必ず指定しなければなりません。

パラメーター値

scope 実動サーバーへの伝搬の有効範囲レベル。以下のいずれかを指定します。

- **_all_**
サイトに関連するレコード、および全マーチャントに関連するレコードの両方を伝搬するには、**_all_** を入力します。
- **_site_**
サイトに関連するレコードのみを伝搬するには、**_site_** を入力します。
- **_merchant_**
全マーチャントに関連するレコードのみを伝搬するには、**_merchant_** を入力します。

sourcedb

ステージング・サーバー上のデータベースの名前。 ▶ Oracle

host:port:sid の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

destdb 実動サーバー上のデータベースの名前。

▶ **Oracle** `host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

dbtable

(オプション) 伝搬する任意の特定のテーブルの名前。レコードが `scope` パラメーターで指定した有効範囲内にある場合は、そのテーブル内のすべての変更レコードが伝搬されます。有効範囲内にはない場合は、どのレコードも伝搬されません。

sourcedb_user

+ (オプション) ソース・データベースのスキーマを作成したデータベース管理者のログオン ID。これを指定しない場合、現在ユーティリティを呼び出しているユーザーの ID が使用されます。

+ ▶ **400** (必須) コマース・インスタンスと関連したユーザー・プロファイル。これは、ソース・データベース・スキーマと同じになります。

sourcedb_passwd

(オプション) `sourcedb_user` パラメーターで指定するログオン ID のパスワード。

destdb_user

+ (オプション) 宛先データベースのスキーマを作成したデータベース管理者のログオン ID。これを指定しない場合、ユーティリティを呼び出しているユーザーの ID が使用されます。リモート・データベースを使用する場合は、このパラメーターは必須です。

+ ▶ **400** (必須) コマース・インスタンスと関連したユーザー・プロファイル。これは、宛先データベース・スキーマと同じになります。

destdb_passwd

(オプション) `destdb_user` パラメーターで指定するログオン ID のパスワード。これを指定しない場合、パスワードを入力するように求められます。リモート・データベースを使用する場合は、このパラメーターは必須です。

log

+ (オプション) ステージ伝搬ユーティリティがそのアクティビティとエラーを記録するファイルのパスと名前。このパラメーターを指定しない場合、`stagingprop_yyyy.mm.dd_hh.mm.ss.zzz.log` という名前のログ・ファイルが以下のディレクトリーに作成されます。

+ ▶ **AIX** ▶ **Solaris** ▶ **Linux** `WC_installdir/logs`
+ ▶ **Windows** `WC_installdir¥logs`
+ ▶ **400** `WC_userdir/instances`

dbtype (オプション) データベースのタイプ (DB2 または Oracle)。デフォルトは DB2 です。

ステージング・サーバーの伝搬について詳しくは、以下の例を参照してください。













実動データベースへのデータの伝搬の例: 以下の例は、ステージング・データベースから実動データベースに変更レコードを伝搬する方法を示しています。

コマンドは 1 行で入力してください。ここでは、表示の都合でコマンドを複数の行に分けているに過ぎません。

例 1

ステージング・サーバー・データベースから実動データベースにすべての変更を伝搬します。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。










-   stagingprop -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name*
-   stagingprop -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name* -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*
-      stagingprop.sh -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name* -sourcedb_user *user* -destdb_user *user*
-    stagingprop.sh -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name* -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

5. log ファイルを調べて、コマンドが正常に実行されたかどうか検査します。ログ・パラメーターの説明を調べて、このデータベース・タイプおよびプラットフォームでのログ・ファイルのデフォルトの場所を判別してください。

ステージ伝搬ユーティリティーは、実動データベースにまずすべてのサイト・データを伝搬した後、すべてのマーチャント・データを伝搬します。エラーが発生する場合、トランザクション全体がロールバックします。

ログ・ファイルの名前とパスを指定するには、以下のログ・ファイル・パラメーターを使用します。

-   stagingprop -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name* -loglog_file_name
-   stagingprop -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name* -loglog_file_name -dbtype oracle -sourcedb_user *user* -sourcedb_passwd *password* -destdb_user *user* -destdb_passwd *password*
-      stagingprop.sh -scope _all_ -sourcedb *staging_database_name* -destdb *production_database_name* -loglog_file_name -sourcedb_user *user* -destdb_user *user*

- ▶ AIX ▶ Solaris ▶ Oracle

```
stagingprop.sh -scope _all_ -sourcedb staging_database_name-destdb
production_database_name -loglog_file_name-dbtype oracle -sourcedb_user
user -sourcedb_passwd password -destdb_user user -destdb_passwd password
```

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

DB2 を使用していて、データベース管理者としてログオンしていない場合は、*-sourcedb_user*、*-sourcedb_passwd*、*-destdb_user*、および *-destdb_passwd* のそれぞれのオプションに値を指定する必要があります。

例 2

ステージング・データベースから実動データベースに、変更されたすべてのサイト・データを伝搬します。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

- ▶ Windows ▶ DB2 `stagingprop -scope _site_ -sourcedb staging_database_name -destdb production_database_name`

- ▶ Windows ▶ Oracle `stagingprop -scope _site_ -sourcedb staging_database_name -destdb production_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

+
+
+

- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2 `stagingprop.sh -scope _site_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user`

- ▶ AIX ▶ Solaris ▶ Oracle `stagingprop.sh -scope _site_ -sourcedb staging_database_name-destdb production_database_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。













5. log ファイルを調べて、コマンドが正常に実行されたかどうか検査します。ログ・パラメーターの説明を調べて、このデータベース・タイプおよびプラットフォームでのログ・ファイルのデフォルトの場所を判別してください。

例 3

(サイト・データを伝搬した後) ステージング・データベースから実動データベースに、変更されたすべてのマーチャント・データを伝搬します。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

+
+
+

-   `stagingprop -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name`
-   `stagingprop -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name -dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`
-      `stagingprop.sh -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name -sourcedb_user user -destdb_user user`
-    `stagingprop.sh -scope _merchant_ -sourcedb staging_database_name -destdb production_database_name dbtype oracle -sourcedb_user user -sourcedb_passwd password -destdb_user user -destdb_passwd password`

注: Oracle データベースの名前には、`host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

5. log ファイルを調べて、コマンドが正常に実行されたかどうか検査します。ログ・パラメーターの説明を調べて、このデータベース・タイプおよびプラットフォームでのログ・ファイルのデフォルトの場所を判別してください。

カスタマイズされたデータベース・テーブルの要件

新規テーブルを作成してデータベース・スキーマをカスタマイズしている場合、ステージング・サーバーを使用するために以下の要件を満たさなければなりません。

- **基本キーまたは固有索引を定義しなければならない。**
ステージング・サーバーの機能はキーに基づいています。STAGLOG テーブルへの過度のデータ記録を避けるために、キー (基本キーまたは固有索引) のみを記録してください。ステージ・ユーティリティーは、圧縮を行ったり伝搬されたデータを見つけたりするのにキーを使用します。キーがないと、ステージ・ユーティリティーは機能しません。
- **テーブル間に参照保全 (RI) 制約サイクルがあってはならない。**
ステージング・サーバーは常に、子テーブルの前に親テーブルを伝搬します。RI 制約サイクルがあると、ステージング・サーバーは親テーブルと子テーブルを区別できません。
- **データベース・テーブルには、構成データしか含められない。**
企業消費者間取引のシナリオでは、カタログやカタログ項目などの構成データはサイト管理者の制御下にあります。テーブルに操作データが含まれる場合、顧客はサイト管理者がステージング・サーバーにテーブルをコピーした後、実動データベース内の同じテーブルを変更することができます。これを行うと、キーの競合や RI 制約違反が生じる可能性があります。
- **データベース・テーブルに操作テーブルへの参照を含めることはできない。**
伝搬されるテーブルに、外部キーから操作テーブルの基本キーへの参照を含めることはできません。そのような参照があると、顧客はステージ・コピーの後基本キーを削除する場合にデータを実動データベースにリストアすることができません。

- ▶ Windows
▶ AIX
▶ 400
▶ Solaris
▶ Linux
 実動データベースの 2 つのテーブルに挿入する際に、挿入トリガーがあってはならない。
 ステージング・サーバーが網羅する任意の 2 つのテーブル (たとえば R1 と R2) に関して、実動データベース内の R2 および R1 に行を挿入する際、それらのテーブルに行を挿入するためのトリガーがあってはなりません。挿入トリガーは両方のデータベースに更新を作成するため、キーの問題が生じます。
- MEMBER** テーブルに固有索引を含められない。
- カスタマイズされたデータベース・テーブル上で削除ルールを使用する際は、注意する必要がある。
 削除ルールを使用すると、Database Cleanup ユーティリティのパフォーマンスが低下します。また、ステージング・データベースをクリーニングする際に障害が発生する可能性もあります。ステージング・データベースをクリーニングする前に、force オプションとともに Database Cleanup ユーティリティ・コマンドを手動で実行して、テーブルをクリーニングする必要があります。そうしないと、ステージング・データベースをクリーニングできません。

カスタマイズされたテーブルに関してステージング・サーバーを準備するには、『カスタマイズされたテーブルのためのステージング・サーバーの構成』を参照してください。

ステージング・サーバーのトラブルシューティング (Windows 2000、AIX、Linux、および Solaris)

1. ステージング・サーバー・コマンド `stgprop` の実行中に、次のメッセージが出ることがあります。
 SQLSTATE 54001: ステートメントは長すぎるか複雑すぎます (SQLSTATE 54001: The statement is too long or too complex)

データベースを構成する際に提案されたとおりにデータベースの `stmthep` サイズを設定しているか確認してください。増加をサポートするのに十分なメモリーがあるか確認する必要があります。

2. ステージ・コピー・ユーティリティが処理を完了できない場合、データベース・ログが小さすぎる可能性があります。DB2 コマンド・ウィンドウから、次のコマンドを発行してください。

```
update database configuration for staging_server using logprimary 50
logfilsiz 1000
db2 terminate
db2stop
db2start
```

ここで、`staging_server` は、ステージング・サーバー・データベースの名前です。問題が解決されない場合は、より大きい値をパラメーター `logprimary` または `logfilsiz` に使用してみてください。

3. ▶ AIX `db2inst1` などの DB2 ユーザーのリソースが正しく設定されているか確認してください。 `ulimit -a` というコマンドを入力すると、リソースの制限が表示されます。
 データ・セグメントを 240MB に、スタックを 16MB に構成してください。追加情報は、「DB2 コマンド・リファレンス」を参照してください。

WebSphere Commerce のステージング・テーブル

WebSphere Commerce ステージング・テーブルを、実動サーバーからステージング・サーバーにコピーすることができます。テーブルは、サイト関連のデータを含むテーブル、マーチャント関連のデータを含むテーブル、サイト関連のデータとマーチャント関連のデータの両方を含むテーブル、という具合にグループ分けされます。

各テーブル・グループは、以下のページにリストされます。

- サイト・データの有効範囲
- マーチャント・データの有効範囲
- サイト・データおよびマーチャント・データの有効範囲

ステージング・テーブルのトリガー

WebSphere Commerce のステージング・テーブルに関して、以下のトリガーが定義されています。カスタム・テーブルに同じデータ有効範囲とキー特性が含まれる場合は、その設定をカスタム・テーブルに適用することができます。データの有効範囲別のテーブルを表示するには、以下のリンクを参照してください。

- サイト・データの有効範囲
- マーチャント・データの有効範囲
- サイト・データまたはマーチャント・データの有効範囲

サイト・データの有効範囲: 以下の図表は、サイト・データの有効範囲にあるデータベース・テーブルをリストしています。

テーブル名	固有索引				
	Integer 1	Integer 2	Char 1	Integer 3	Char 2
ACACGPDESC	○	○			
ACACTACTGP	○	○			
ACACTDESC	○	○			
ACACTGRP	○				
ACACTION	○				
ACATTR	○				
ACATTRDESC	○	○			
ACCCMDTYPE			○		
ACRELATION	○				
ACRELDDESC	○	○			
ACRESACT	○	○			
ACRESATREL	○	○			
ACRESCGRY	○				
ACRESGRP	○				
ACRESGPDES	○	○			
ACRESGPRES	○	○			
ACRESPRIM	○				
ACRESREL	○	○			
ATTACHUSG			○		

ATTRTYPE			○		
BUYERPOTYP	○				
CALUSAGE	○				
CATENTTYPE			○		
CATRELTYPE			○		
CHKARRANG	○	○			
CHKCMD	○				
COUNTCODE			○		○
COUNTRY	○		○		
DEVICEFMT	○				
FLCOMPOSE	○	○		○	
FLDOMNDESC	○	○			
FLOW	○				
FLOWDESC	○	○			
FLOWDOMAIN	○				
FLOWTYPE	○				
FLSTATEDCT	○				
FLSTATEGP	○				
FLSTATEREL	○	○			
FLSTDCTDSC	○	○			
FLSTGPDSC	○	○			
FLTRANSDSC	○	○			
FLTRANSITN	○				
FLTYPEDESC	○	○			
ICDATAREG	○				
INVRSRVTYP	○				
INVRSRVDSC	○	○			
ITEMTYPE			○		
LANGUAGE	○				
LANGUAGEDS	○	○			
MASSOC			○		
MASSOCTYPE			○		
MBRATTR	○				
MBRGRPTYPE	○				
OPERATOR	○				
OPERATRDSC	○	○			
ORDCHNLTY	○				
OUTPUTQ	○				
OUTPUTQDSC	○	○			
PARTROLE	○				
PARTROLED	○	○			
PLCYTYPDSC	○		○		

POLICYTYPE	○				
QTYCONVERT			○		○
QTYUNIT			○		
QTYUNITDSC	○		○		
ROLE	○				
SCHCMD	○				
SETCURR			○		
SETCURRDSC	○		○		
STATECODE			○		○
STATEPROV	○		○		
STORECGRY	○				
TAXTYPE	○				
TCSUBTYPDS	○		○		
TCSUBTYPE			○		
TCTYPE			○		
TFALGOPOL	○				
TFALGOTYPE	○				
TFALGPOLDS	○	○			
TFALGTYPDS	○	○			
TFDOMAIN	○				
TFDOMDSC	○	○			
TFSBDOMAIN	○				
TFSBDOMDSC	○	○			
TRDTYPE	○				
TRDTYPEDSC	○	○			
TXCDScheme	○				

マーチャント・データの有効範囲: 以下の図表は、マーチャント・データの有効範囲にあるデータベース・テーブルをリストしています。

テーブル名	固有索引				
	Integer 1	Integer 2	Char 1	Integer 3	Char 2
ACCCMDGRP	○				
ACCCUSTEXC	○	○			
ACCOUNT	○				
ACORGPOL	○	○			
ACPOLDESC	○	○			
ACPOLICY	○				
ATTACHMENT	○				
ATTRIBUTE	○	○			
ATTRVALUE	○	○			
BASEITEM	○				

BASEITEM	<input type="radio"/>	<input type="radio"/>			
BUYERPO	<input type="radio"/>				
CALCODE	<input type="radio"/>				
CALCODEDSC	<input type="radio"/>	<input type="radio"/>			
CALCODEMGP	<input type="radio"/>	<input type="radio"/>			
CALCODTXEX	<input type="radio"/>	<input type="radio"/>			
CALMETHOD	<input type="radio"/>				
CALRANGE	<input type="radio"/>				
CALRLOOKUP	<input type="radio"/>				
CALRULE	<input type="radio"/>				
CALRULEMGP	<input type="radio"/>	<input type="radio"/>			
CALSCALE	<input type="radio"/>				
CALSCALEDS	<input type="radio"/>	<input type="radio"/>			
CATALOG	<input type="radio"/>				
CATALOGDSC	<input type="radio"/>	<input type="radio"/>			
CATCNTR	<input type="radio"/>	<input type="radio"/>			
CATCONFINF	<input type="radio"/>				
CATENCALCD	<input type="radio"/>				
CATENTATTR	<input type="radio"/>				
CATENTDESC	<input type="radio"/>	<input type="radio"/>			
CATENTREL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		
CATENTRY	<input type="radio"/>				
CATENTSHIP	<input type="radio"/>				
CATGPCALCD	<input type="radio"/>				
CATGPENREL	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	
CATGROUP	<input type="radio"/>				
CATGRPATTR	<input type="radio"/>	<input type="radio"/>			
CATGRPDESC	<input type="radio"/>	<input type="radio"/>			
CATGRPPS	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	
CATGRPREL	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	
CATGRPTPC	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	
CATTOGRP	<input type="radio"/>	<input type="radio"/>			
CHARGETYPE	<input type="radio"/>				
CHRGTYPDSC	<input type="radio"/>	<input type="radio"/>			
CNTRDISPLY	<input type="radio"/>				
CNTRNAME	<input type="radio"/>		<input type="radio"/>		
CONTRACT	<input type="radio"/>				
CREDITLINE	<input type="radio"/>				
CRULESCALE	<input type="radio"/>	<input type="radio"/>			
CURCONVERT	<input type="radio"/>				
CURCVLIST	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>
CURFMTDESC	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		

CURFORMAT	○		○		
CURLIST	○		○		
DISPCGPREL	○				
DISPENTREL	○				
DISTARRANG	○				
FLOWADMIN	○				
INVADJCODE	○				
INVADJDESC	○	○			
ITEMSPC	○				
ITEMVERSN	○				
JURST	○				
JURSTGPREL	○	○			
JURSTGROUP	○				
LANGPAIR	○	○		○	
LISTPRICE	○		○		
MASSOCCECE	○				
MASSOCGPGP	○				
MBRATTRVAL	○				
MBRGRPUSG	○	○			
MBRREL	○	○			
MBRROLE	○	○		○	
MGPTRDPCSN	○	○			
OFFER	○				
OFFERDESC	○	○			
OFFERPRICE	○		○		
PARTICIPNT	○				
PATTRDESC	○	○			
PATTRIBUTE	○				
PATTRPROD	○	○			
PKGATTR	○				
PKGATTRVAL	○				
PKGITEMREL	○	○		○	
POLICY	○				
POLICYCMD	○		○		
POLICYDESC	○	○			
POLICYTC	○	○			
PRODSETDSC	○	○			
PRODUCTSET	○				
PRSETCEREL	○	○			
QTYFMTDESC	○	○	○		
QTYFORMAT	○		○		
RTNDNYDESC	○	○			

RTNDNYRSN	○				
RTNDSPCODE	○				
RTNDSPDESC	○	○			
RTNREASON	○				
RTNRSNDESC	○	○			
SHPARJURGP	○	○			
SHPARRANGE	○				
SHPJCRULE	○				
SHPMODEDESC	○	○			
STENCALUSG	○	○			
STORECAT	○	○			
STORECENT	○	○			
STORECGRP	○	○			
STORECNTR	○	○			
STOREDEF	○				
STOREITEM	○	○			
STORELANG	○	○			
STOREMBRGP	○	○			
STORITMFFC	○	○		○	
STORLANGDS	○	○		○	
TAXCGRY	○				
TAXCGRYDS	○	○			
TAXJCRULE	○	○		○	
TCDESC	○	○			
TDPSCNCNTR	○	○			
TERMCOND	○				
TRADEPOSCN	○				
TRADING	○				
TRDATTACH	○	○			
TRDDESC	○	○			
TXCDCLASS	○				
VENDOR	○				
VENDORDESC	○	○			
VERSIONSPC	○				
VIEWREG	○	○	○		

サイト・データおよびマーチャント・データの有効範囲: 以下の図表は、サイト・データおよびマーチャント・データの有効範囲にあるデータベース・テーブルをリストしています。

テーブル名	固有索引				
	Integer 1	Integer 2	Char 1	Integer 3	Char 2
CMDREG	○		○		
FFMCENTDS	○	○			
FFMCENTER	○				
MBRGRP	○				
MBRGRPCOND	○				
MEMBER	○				
ORGENTITY	○				
SHIPMODE	○				
STADDRESS	○				
STORE	○				
STOREENT	○				
STOREENTDS	○	○			
STOREGRP	○				
URLREG	○		○		
VIEWREG	○	○	○		

接続サービス

WebSphere Commerce は外部システムとの接続に、Program Adapter for HTTP、Adapter for WebSphere MQ、および Adapter for CrossWorlds を使用します。以降のセクションでは、各アダプターについて説明し、それらのアダプターを WebSphere Commerce で使用方法を示します。

Program Adapter for HTTP

Program Adapter for HTTP は、HTTP プロトコル上で XML 要求を渡すことにより、外部システムと WebSphere Commerce との通信を可能にします。Program Adapter for HTTP は調達システムなどの外部システムに、HTTP を介した WebSphere Commerce との通信という一般的な方法を提供します。WebSphere Commerce は、バイヤー/サプライヤーのトランザクションに関して、外部システムに対するサプライヤーとして機能します。Program Adapter for HTTP は、以下のアクションを実行することで着信 XML 要求を処理します。

- 要求を認識し、それが XML 要求かどうかを検証する。以下の 3 つの要求属性がサポートされる場合、XML 要求として識別できます。
 - content-type
 - method
 - character encoding
- サポートされる要求属性は、アダプター構成で指定されます。
- 要求の入カストリームを抽出する。
- メッセージ・マッパーを呼び出し、入カストリームの内容を渡す。
- メッセージ・マッパーによって戻される WebSphere Commerce コマンドを表す CommandProperty オブジェクトを受け取る。

- 応答を生成する際に使用する適切なデバイス形式を決定する。
- コマンドを実行する。
- XML 応答メッセージを送信する。この応答メッセージは JSP によって作成され、実行されたコマンドによって指定されたビュー名と受け取った要求のデバイス形式に基づきます。

それぞれの要求は、別個のセッションとして扱われます。メッセージの認証情報は、メッセージの制御域で指定されます。デフォルトでは、Program Adapter for HTTP は、メッセージ内のユーザー ID とパスワードのパラメーターを検査して、要求の確実性を判別します。Program Adapter for HTTP は、レガシー・メッセージをサポートしていません。レガシー・メッセージは、認証情報の仕様をサポートしていないからです。

Program Adapter for HTTP のライフ・サイクルは、WebSphere Commerce インスタンス全体に渡って存在します。これは、構成パラメーターが除去されていないかアダプターが使用可能になっていれば、インスタンスの始動時に初期化され、インスタンスが実行されている限り存続します。

注: Program Adapter for HTTP は、デフォルトでは使用不可になっています。

WebSphere Commerce がデバイスから受信する要求の処理の仕方に関する構造的な情報については、「*WebSphere Commerce プログラミング・ガイドとチュートリアル*」を参照してください。Program Adapter for HTTP の構成、およびその他の接続サービスについては、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

Listener for WebSphere MQ

Listener for WebSphere MQ は、WebSphere MQ 経由でインバウンド・メッセージを処理することによってバックエンド・システムとの統合を可能にする WebSphere Commerce コンポーネントの 1 つです。Listener for WebSphere MQ は、MQ メッセージの検索に使用する Adapter for Websphere MQ と、そのメッセージを実行するために呼び出されるプログラム・アダプターを組み合わせたものです。

Listener for WebSphere MQ は、WebSphere Commerce ビジネスの処理とバックエンド・システムまたは外部システムの処理を統合するのを助ける一連の事前定義メッセージを持ちます。それぞれの着信メッセージは WebSphere Commerce 内の処理をアクティブにし、データベース・テーブルを更新したり、他の操作を実行したりします。提供されているメッセージについては、WebSphere Commerce オンライン・ヘルプにあるバックエンドの統合およびフルフィルメントの統合に関するメッセージの情報を参照してください。Listener for WebSphere MQ は、既存の事前定義メッセージの他にも、メッセージの拡張機能や新規メッセージをサポートします。

ミドルウェアとしての WebSphere MQ: Adapter for WebSphere を使用すると、ミドルウェアとして WebSphere MQ を使用して、バックエンド・システムおよび外部システムを WebSphere Commerce に統合することができます。さらに、Adapter for WebSphereMQ によって、WebSphere Commerce が、バックエンド・システムおよび外部システムからメッセージを受け取ることができるようになります。サポートされるソフトウェアは、WebSphere MQ クライアント製品の拡張機能を持つ

WebSphere MQ バージョン 5.3 以降です。WebSphere MQ Client は WebSphere MQ 用のクライアントで、ダウンロードによって別に入手できます。

バインディング・モードまたはクライアント・モードを使用することによって、WebSphere MQ をミドルウェアとしてセットアップすることができます。WebSphere Commerce が WebSphere MQ サーバーと同じマシンにインストールされ、Java Native Interface (JNI) を使用する WebSphere MQ Java を介して WebSphere MQ サーバーに接続されると、バインディング・モードが使用されます。通信は、ネットワーク経由ではなく JNI がキュー・マネージャーの API を直接呼び出すことによって行われるため、バインディング・モードを使用するほうが、ネットワーク接続を使用するクライアント・モードを使用するよりも一般に高いパフォーマンスを示します。クライアント・モードが使用されるのは、WebSphere Commerce が 1 つのマシンにインストールされ、WebSphere MQ サーバーがバックエンド・システム上でインストールされる場合です。

WebSphere MQ 接続、キュー、およびチャネルを検証するには、テスト・プログラムを実行して、WebSphere MQ キューとの間でメッセージの送受信を行ってください。詳しくは、WebSphere MQ の資料および「*WebSphere Commerce 追加ソフトウェア・ガイド*」を参照してください。

Adapter for CrossWorlds

Adapter for CrossWorlds は、IBM WebSphere Commerce ビジネスと IBM CrossWorlds との統合を拡張する新しい機構を提供します。この新しいアダプターは CrossWorlds Server Access Interface というアプリケーション・プログラミング・インターフェース (API) を使用します。これを使用すると、外部プロセスによって IBM CrossWorlds InterChange Server (CW ICS) 内部のコラボレーションを実行することができます。このアダプターを使用すると、WebSphere Commerce は、IBM CrossWorlds を介してこれらのシステムに同期メッセージを送信することによって、外部システムと統合することができます。また、このアダプターを使用することによって WebSphere Commerce は CrossWorlds サーバーにメッセージを送信し、応答を待つことができます。応答が戻された後、呼び出しコマンドは他のビジネス・ロジックの処理を継続できます。

IBM CrossWorlds システムは、一組のソフトウェア統合製品です。その中には、一般的なビジネス統合要件のための、コラボレーションと呼ばれる事前に構築されたビジネス・ロジック・テンプレートや、各種開発ツールおよび管理ツールが含まれています。コラボレーションは、オーダー管理や製造部品表管理などの業界固有の一般的なビジネス・プロセス・ステップを定義し、自動化します。また、コラボレーションは、異なるエンタープライズ・ソフトウェア製品のビジネス・プロセスを調整および拡張したり、それら製品間での意味のあるデータ交換を容易にしたりするためにも使用されます。

注: Adapter for CrossWorlds について詳しくは、「*WebSphere Commerce 追加ソフトウェア・ガイド*」および「WebSphere Commerce オンライン・ヘルプ」を参照してください。

WebSphere Commerce と IBM CrossWorlds InterChange Server との統合:

WebSphere Commerce は、IBM CrossWorlds InterChange Server (ICS) と統合できるように設計されています。これにより、WebSphere Commerce は、ICS を介してバ

ックエンド・システムと統合することができます。この WebSphere Commerce と ICS の統合は、以下の 3 種類のメッセージ・フローをサポートします。

- アウトバウンド非同期メッセージ・フローを使用すると、 Websphere Commerce は処理のためにメッセージを ICS に送信できます。
- インバウンド・メッセージ・フローを使用すると、 Websphere Commerce は ICS からメッセージを受信できます。
- 同期メッセージ・フローを使用すると、 WebSphere Commerce は ICS に要求/応答メッセージを送信できます。

アウトバウンド・フローおよびインバウンド・フローは、 Websphere Business Integration (WBI) Adapter for Websphere Commerce を使用します。要求/応答フローは、 Websphere Commerce SAI Adapter for ICS を使用します。

WebSphere Commerce アウトバウンド・プロセス: WebSphere Commerce のトランザクションは、 WebSphere Commerce メッセージング・システムを介して他のアプリケーションに送信されるメッセージをトリガーできます。メッセージング・システムを介して生成されるメッセージを使用して、 Websphere Business Integration Server のような統合サーバーを介して他のアプリケーションと統合することができます。たとえば、 WebSphere Commerce から OrderCreate XML メッセージを生成して、バックエンド・システムにオーダーを作成することができます。

WebSphere Commerce にオーダーが作成されると、さらに処理を行うために配送などの他のシステムにオーダー関連の情報を送信するように構成することができます。 IBM Websphere Business Integration Server を使用する顧客は、処理を行うために Websphere Commerce からの XML メッセージを IBM CrossWorlds ICS に送信するように Websphere Commerce および IBM CrossWorlds ICS を構成することができます。この場合、オーダーは XML メッセージを介して WebSphere Commerce から ICS に送信されます。この XML メッセージは、 IBM CrossWorlds ICS によって処理し、 SAP のような他のシステムに送信することができます。

WebSphere Commerce にオーダーを発行すると、 XML 形式の OrderCreate メッセージが生成され、 WebSphere MQ 出力キューに発行されます。 WBI Adapter for WebSphere Commerce エージェントは定期的に新規メッセージをポーリングします。その後そのメッセージは、 WebSphere MQ 出力キューからコネクタ・コントローラーに渡されます。

コントローラーは WebSphere Commerce 固有のビジネス・オブジェクトを受け取り、マップを呼び出して汎用ビジネス・オブジェクト (GBO) を生成します。 GBO は対応するコラボレーションに渡されます。そしてそのコラボレーションはそれを処理し、バックエンド・システムに要求を送信します。

Websphere Commerce と IBM CrossWorlds ICS の統合について詳しくは、
<http://www-3.ibm.com/software/websphere/crossworlds/library/doc/v411/welcome.html> にある WBI Adapter for Websphere Commerce の資料を参照してください。

WebSphere Commerce インバウンド・プロセス: Websphere Commerce は、IBM CorssWorlds ICS と統合して、ICS から Websphere Commerce への要求をサポートできるように設計されています。Websphere Commerce メッセージング・システムは Websphere Commerce ビジネス・ロジックを呼び出して ICS から来る要求をサポートします。

このシナリオでは、IBM CrossWorlds ICS から来るビジネス・ロジックをトリガーする要求を呼び出すために、既存の WebSphere Commerce メッセージを使用します。

WBI Adapter for WebSphere Commerce はコラボレーションからビジネス・オブジェクトを受け取り、データ・ハンドラーを使ってそれらを XML 形式のメッセージに変換し、その後 WebSphere MQ キューにメッセージを送信します。Websphere Commerce と IBM CrossWorlds ICS の統合について詳しくは、<http://www-3.ibm.com/software/websphere/crossworlds/library/doc/v411/welcome.html> にある Adapter for CrossWorlds の資料を参照してください。

構成タスク

メッセージング・サービスの構成

システムの管理中に、サイト管理者は以下のタスクを実行できます。

- メッセージを送付するための伝達手段 (トランスポートと呼ばれる) を追加する
- トランスポートを構成する
- メッセージ・トランスポートの割り当てを表示する
- メッセージ・タイプを構成する (メッセージの説明、重大度、トランスポート、またはデバイス形式を追加、変更、または削除する)
- ストアにトランスポート・メソッドを追加する
- トランスポート・メソッドをアクティブまたは非アクティブにする

メッセージング・システムの構成について詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

スケジューリング・サービスの構成

スケジューラーにアクセスするには、管理コンソールを使用します。スケジューラーの使い方の詳細については、「WebSphere Commerce オンライン・ヘルプ」を参照してください。スケジューラーの保守について詳しくは、154 ページの『スケジューラーの保守』を参照してください。

ロギングの構成

ロギングとトレースの構成はすべて、WebSphere Application Server 管理コンソールから行えます。以下のセクションでは、独立型のツールのみを扱っています。

- ECTrace 始動オプションの変更
- ECMessage 始動オプションの変更

WebSphere Application Server のロギングとトレースについて詳しくは、「WebSphere Application Server InfoCenter」を参照してください。

コンポーネントのトレースの使用可能化

WebSphere Commerce システムでトレースを使用可能にするには、WebSphere Application Server 管理コンソールを使用する必要があります。

サーバー始動時にトレースを使用可能にする:

1. WebSphere Application Server 管理コンソールをオープンします。この使い方については、「WebSphere Application Server InfoCenter」を参照してください。
2. コンソール・ナビゲーション・ツリーから「トラブルシューティング (Troubleshooting)」>「ロギングとトレース (Logging and Tracing)」をクリックし、その後「サーバー (Server)」>「診断トレース (Diagnostic Trace)」をクリックします。
3. 「構成」をクリックします。
4. トレースを使用可能にするには「トレースを使用可能にする (Enable Trace)」チェック・ボックスを選択し、トレースを使用不可にするにはチェック・ボックスをクリアします。
5. 適切な TraceString を入力して、トレースの仕様を希望する状態に設定します。
6. トレース出力をファイルに送信するか、メモリー内の循環バッファーに送信するか選択します。
7. 生成されたトレースに対して希望する形式を選択します。
8. 変更された構成を保管します。
9. サーバーを開始します。

注: これらのステップのいずれかについて詳しくは、「WebSphere Application Server InfoCenter」を参照してください。

実行中のサーバーに対してトレースを使用可能にする: 以下の手順を使用することにより、実行中のサーバーで現在トレースされているコンポーネントを判別するトレース・サービス状態を変更することができます。

1. WebSphere Application Server 管理コンソールをオープンします。
2. コンソール・ナビゲーション・ツリーから「トラブルシューティング (Troubleshooting)」>「ロギングとトレース (Logging and Tracing)」をクリックし、その後「サーバー (Server)」>「診断トレース (Diagnostic Trace)」をクリックします。
3. 「ランタイム (Runtime)」タブを選択します。
4. トレースの仕様を希望する状態に変更することにより、既存のトレース状態を変更します。
5. (オプション) 既存のトレースからの変更を希望する場合は、トレース出力を構成します。
6. 「適用」をクリックします。

注: これらのステップのいずれかについて詳しくは、「WebSphere Application Server InfoCenter」を参照してください。

WebSphere Commerce のトレース・コンポーネント: 以下は、WebSphere Commerce V5.5 のトレース・コンポーネントです。

表 6.

トレース・コンポーネント	WebSphere JRAS 拡張機能のトレース・ロガー
SERVER	com.ibm.websphere.commerce.WC_SERVER
CATALOG	com.ibm.websphere.commerce.WC_CATALOG
DATASOURCE	com.ibm.websphere.commerce.WC_DATASOURCE
ORDER	com.ibm.websphere.commerce.WC_ORDER
USER	com.ibm.websphere.commerce.WC_USER
COMMAND	com.ibm.websphere.commerce.WC_COMMAND
CF	com.ibm.websphere.commerce.WC_CF
NEGOTIATION	com.ibm.websphere.commerce.WC_NEGOTIATION
RAS	com.ibm.websphere.commerce.WC_RAS
DB	com.ibm.websphere.commerce.WC_DB
METAPHOR	com.ibm.websphere.commerce.WC_METAPHOR
SCHEDULER	com.ibm.websphere.commerce.WC_SCHEDULER
DEVTOOLS	com.ibm.websphere.commerce.WC_DEVTOOLS
TOOLSFRAMEWORK	com.ibm.websphere.commerce.WC_TOOLSFRAMEWORK
RULESYSTEM	com.ibm.websphere.commerce.WC_RULESYSTEM
MERCHANDISING	com.ibm.websphere.commerce.WC_MERCHANDISING
MARKETING	com.ibm.websphere.commerce.WC_MARKETING
REPORTING	com.ibm.websphere.commerce.WC_REPORTING
TRANSPORT_ADAPTER	com.ibm.websphere.commerce.WC_TRANSPORT_ADAPTER
PERFMONITOR	com.ibm.websphere.commerce.WC_PERFMONITOR
MESSAGING	com.ibm.websphere.commerce.WC_MESSAGING
STOREOPERATIONS	com.ibm.websphere.commerce.WC_STOREOPERATIONS
CACHE	com.ibm.websphere.commerce.WC_CACHE
EVENT	com.ibm.websphere.commerce.WC_EVENT
EJB	com.ibm.websphere.commerce.WC_EJB
CURRENCY	com.ibm.websphere.commerce.WC_CURRENCY
CATALOGTOOL	com.ibm.websphere.commerce.WC_CATALOGTOOL
CONTRACT	com.ibm.websphere.commerce.WC_CONTRACT
UBF	com.ibm.websphere.commerce.WC_UBF
BI	com.ibm.websphere.commerce.WC_BI
INVENTORY	com.ibm.websphere.commerce.WC_INVENTORY
UTF	com.ibm.websphere.commerce.WC_UTF
RFQ	com.ibm.websphere.commerce.WC_RFQ
EXCHANGE	com.ibm.websphere.commerce.WC_EXCHANGE
ACCESSCONTROL	com.ibm.websphere.commerce.WC_ACCESSCONTROL

表 6. (続き)

AC_UNITTEST	com.ibm.websphere.commerce.WC_AC_UNITTEST
APPROVAL	com.ibm.websphere.commerce.WC_APPROVAL
COLLABORATION	com.ibm.websphere.commerce.WC_COLLABORATION
THREAD	com.ibm.websphere.commerce.WC_THREAD
SENSITIVE_INFO	com.ibm.websphere.commerce.WC_SENSITIVE_INFO

JCA コネクタ・トレース・コンポーネント: 以下は、WebSphere Commerce V5.5 の JCA コネクタ・トレース・コンポーネントです。

表 7.

トレース・コンポーネント	WebSphere JRAS 拡張機能のトレース・ロガー
JCA E メール・コネクタ	com.ibm.websphere.commerce.jcaemail
JCA ファイル・コネクタ	com.ibm.websphere.commerce.jcafile
JCA JMS コネクタ	com.ibm.websphere.commerce.jcajms
JCA サンプル・コネクタ	com.ibm.websphere.commerce.jcasample

Commerce Payments トレース・コンポーネント: 以下は、WebSphere Commerce V5.5 の Commerce Payments トレース・コンポーネントです。

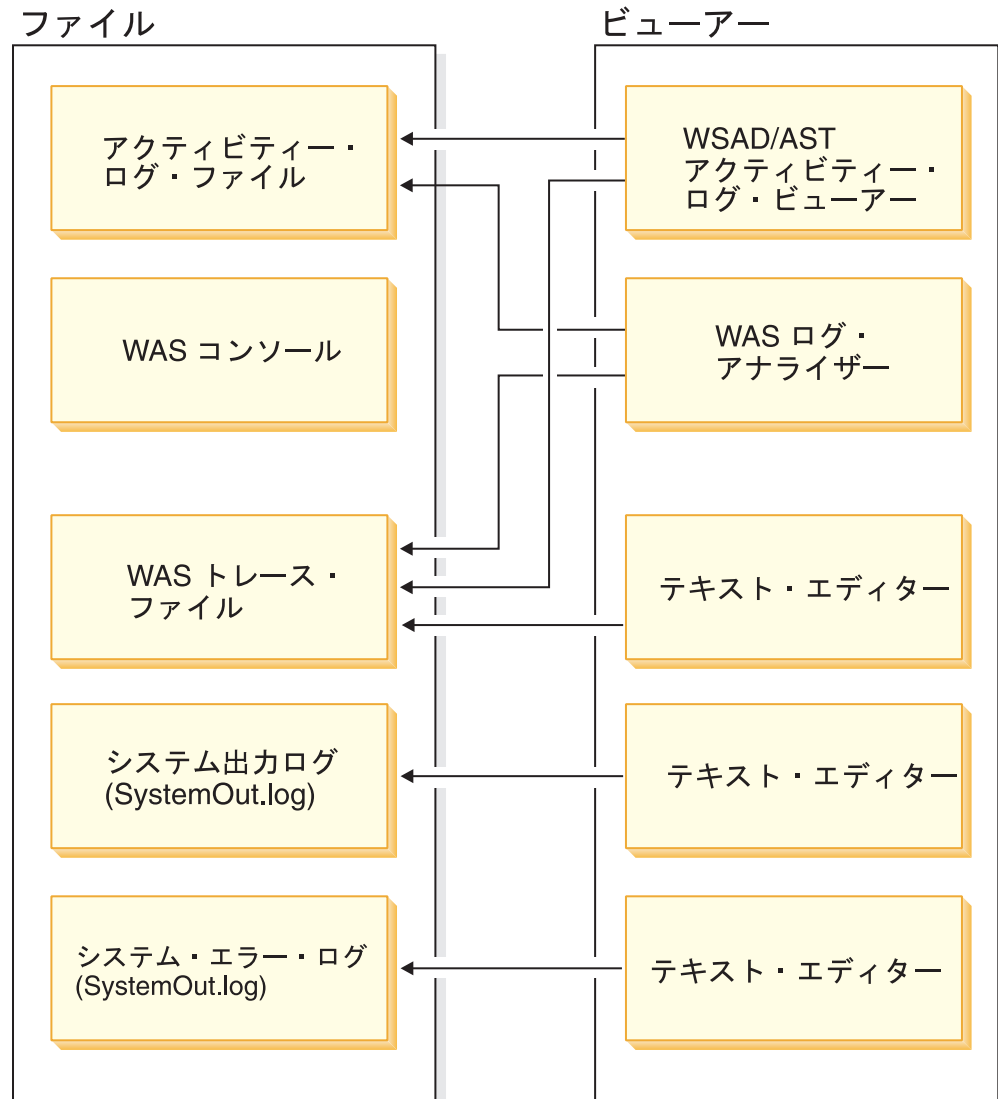
表 8.

トレース・コンポーネント	WebSphere JRAS 拡張機能のトレース・ロガー
* (任意のトレース・コンポーネント)	com.ibm.websphere.commerce.payments.*
MPF (Payments フレームワーク)	com.ibm.websphere.commerce.payments.MPF
MPFUI (Payments UI)	com.ibm.websphere.commerce.payments.MPFUI
<Cassette_name> (これはすべての IBM またはサード・パーティーのカセットのトレースを使用可能にします。) WebSphere Commerce には、VisaNet、Paymentech、BankServACH、OfflineCard、および CustomOffline の各種カセットが付属しています。	com.ibm.websphere.commerce.payments.<Cassette_name>

注: WebSphere Commerce Payments について詳しくは、「WebSphere Commerce Payments Library」を参照してください。

ログ・ファイルの表示

以下の図は、ログ・ファイルの表示の違いを示しています。



ログ・アナライザーの使用

WebSphere Application Server の一部であるログ・アナライザーは、1 つ以上のサービスまたはアクティビティー・ログを取り、すべてのデータをマージして、エントリーを表示します。ツールは問題の診断を助けるために、その症状データベースに基づいてログ・エントリー内のイベント条件またはエラー条件を分析し、解釈します。ログ・アナライザーには、その機構を使用可能することにより、IBM Web サイトから最新の症状データベースをダウンロードする特殊機構があります。最新の症状データベースの更新をダウンロードするには、ログ・アナライザー・インターフェースから、「ファイル」->「データベースの更新 (Update Database)」->




「WebSphere Application Server 症状データベース (WebSphere Application Server Symptom Database)」オプション (WebSphere Application Server の場合) または

「WebSphere Application Server Network Deployment 症状データベース (WebSphere Application Server Network Deployment Symptom Database)」オプション (WebSphere Application Server Network Deployment の場合) を使用します。

サービスおよびアクティビティ・ログ: WebSphere Application Server サーバーは、さまざまな WebSphere Application Server コンポーネントのアクティビティからサービスおよびアクティビティ・ログ・ファイルを作成します。サービスまたはアクティビティ・ログ・ファイル activity.log は、`installation_directory/logs/activity` ログにあるバイナリー・ファイルです。ログ・アナライザーを使用してサービスまたはアクティビティ・ログ・ファイルを表示することができます。また、それらのファイルを 1 つのログ・ファイルにマージすることもできます。サービスまたはアクティビティ・ログはテキスト・エディターでは表示できません。ログ・アナライザー・ツールはこのファイルを表示するために設計されています。activity.log について詳しくは、「*WebSphere Application Server System Administration Guide*」を参照してください。

ログ・アナライザーを使用したサービスおよびアクティビティ・ログの表示: ログ・アナライザーを使用してサービスまたはアクティビティ・ログを表示するには、以下のようにします。

1. `WASinstall_dir/bin` ディレクトリーに移動します。
2. `waslogbr` スクリプト・ファイルを実行します。

-  `v waslogbr.bat`
-   `v waslogbr`

これにより、ログ・アナライザーのインターフェースが開始します。

3. 「ファイル」 > 「オープン」を選択します。
4. サービスまたはアクティブ・ログ・ファイルを含むディレクトリーにナビゲートします。
5. サービスまたはアクティビティ・ログ・ファイルを選択し、「オープン」をクリックします。
6. レコードを分析するには、左側のツリーのレコードを右マウス・ボタンでクリックし、右クリック・メニューから「UnitOfWorkView」を選択して、「分析 (Analyze)」を選択します。

これで隣りに緑色のチェック・マークが付いているレコードはすべて、症状データベースのレコードと一致しています。チェック・マークのレコードを選択すると、右下のペインに問題の説明が表示されます。ログ・アナライザー・ユーザー・インターフェースについて詳しくは、「WebSphere Application Server InfoCenter」を参照してください。

テキスト・ファイルの使用

ログ・ファイルを表示するには、任意のテキスト・エディターを使用できます。ログ・ファイルは、WebSphere Commerce サーバーがインストールされているログ・ディレクトリーにあります。通常これは `../appserver/logs/` ディレクトリーになりますが、変更することもできます。以下に示すのは、ログ・ファイルを見つけるために使用できるパスの例です。

`WAS_installdir/logs/WC_instanceName/trace.log`

注: **AIX** vi 以外のテキスト・エディターを使用して、生成された WebSphere Commerce ログ・ファイルを表示します。vi で表示するには行が長すぎるため、正しく表示することができません。

LDAP の構成

WebSphere Commerce のインストール後は、LDAP サーバーをメンバー・リポジトリとして使用することができます。または、最初はデータベースをメンバー・リポジトリとして使用し、後で LDAP サーバーに切り替えることもできます。LDAP の構成について詳しくは、「*WebSphere Commerce 追加ソフトウェア・ガイド*」の『LDAP 用の WebSphere Commerce の構成』を参照してください。

ルール・サービスの構成

ルール・サービスを構成するには、管理コンソールにログオンして、「**ルール・サービス**」メニューから「**管理**」をクリックします。サイト管理者は、管理コンソールを使用して、以下のルール・サービス・タスクを実行できます。

- ルール・サービスの追加
- ルール・サービスの編集
- ルール・サービスの状況の検査
- ルール・サービスの最新表示
- ルール・サービスの削除
- ルール・サービスの開始
- ルール・サービスの停止

ルール・サービスを構成するためのタスクについて詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

ステージング・サーバーを構成する

ステージング・サーバー・ユーティリティーの使用

ステージング・ユーティリティーは、ステージング・サーバー上に存在し、標準の DB2/JDBC 通信を使用して実動データベースとステージング・データベースの両方に接続します。管理者は、DB2 クライアント/サーバー環境が正しくセットアップされていることを確認する必要があります。DB2 クライアントは、TCP/IP ポートを介して DB2 サーバーと通信します。番号は、DB2 インスタンス変数 SVCENAME で設定します。このポート番号を知っておく必要があり、このポートがプロキシによってブロックされないようにする必要もあります。DB2 がデフォルトで使用するポート番号は 50000 です。JDBC と DB2 クライアント/サーバー環境のセットアップの詳細については、「*DB2 管理ガイド*」を参照してください。

環境をセットアップし、ステージング・サーバーのデータを実動サーバーにコピーする準備ができたなら、ステージング・ユーティリティーを使用してデータをプッシュします。コピー可能なテーブルは、ステージング・テーブルにリストされます。ステージング・サーバーをカスタマイズ・テーブル用に構成すれば、カスタム・データを保管するために作成した新規テーブルもステージング・サーバーにコピーできます。

STAGLOG テーブルは、内部ログの機能を果たします。ステージング・サーバー上のテーブルのレコードを変更するたびに、トリガーは変更を STAGLOG テーブルに記録します。変更されたレコードごとに、トリガーは、変更のタイプ (挿入、削除、または更新)、レコードが存在するテーブルの名前、レコードの基本キーまたは固有索引を記録します。ステージング・サーバー上でデータベース・レコードの変更とテストを終えたら、ステージ伝搬ユーティリティを使用して変更を実動サーバーに伝搬します。

ステージング・サーバーで処理の対象となるデータベース・テーブルは、ステージング・セッション中に実動データベースで更新されることはありません。ステージング・セッションは、ステージ・コピー・ユーティリティを使用して実動データベースをステージング・データベースにコピーするときに開始し、ステージ・コピー・ユーティリティを使用して別のステージング・セッションを開始するときに終了します。実動データベースのデータをステージング・データベースにコピーし終えたら、ステージング・データベースと実動データベースは、ステージング・サーバーで処理の対象となったテーブルに関して同期化されます。テーブルが同期化されると、実動データベース上でテーブルを変更することができなくなります。ステージング・データベースを更新し、次いでステージ伝搬ユーティリティを使用して、変更を実動データベースに伝搬することしか行えません。両方のデータベースを更新すると、キー競合または参照健全性違反が原因で、伝搬は失敗するでしょう。ステージング・セッション中に実動データベースを更新する必要がある場合は、ステージ・コピー・ユーティリティを使用して両方のデータベースを同期化し、新しいステージング・セッションを開始してください。

ステージング・セッション中に実動データベースで更新が行われないようにするには、サイト管理者だけがテーブルを管理できるようにする必要があります。ステージ・コピー・ユーティリティを使用してデータをコピーした後で、個々の顧客またはマーチャントが、実動データベースのステージング・テーブルを更新する場合があります。たとえば、実動データベースへのコピー後に、マーチャントが実動データベースの OFFER テーブルを変更することを禁止することはできません。このような場合、ステージング・サーバーは使用できなくなります。ただし、RFQ オブジェクトは例外です。実動データベース上で RFQ オブジェクトを作成すると、実動データベースの trading テーブルに行が挿入されます。ステージング・データベース内で契約を作成する場合は、ステージング・データベースの trading テーブルにも行が挿入されます。この場合は、ステージング・データベースと実動データベースの両方の同じテーブルが更新されます。

RFQ オブジェクトを使用する場合は、ステージング・サーバーに適用される制限がいくつかあります。ステージ伝搬ユーティリティを使用して変更を実動データベースに伝搬する前に、ステージ検査ユーティリティを使用し、潜在的な固有索引キーの競合を検出して訂正する必要があります。

標準的な企業消費者間取引サイトでは、テーブルは構成データと操作データという 2 つのグループに分けることができます。構成テーブルには、ストア、カタログ、カタログ・エントリー、言語、税、および割引などのデータが入ります。これらのテーブルは管理者が管理し、個々の顧客はテーブルを変更できません。操作テーブルには、顧客情報、住所、オーダー、および SET 関連データが入ります。顧客は、操作テーブルを変更できます。ステージング・サーバーでは、構成テーブルだけが

処理の対象となります。ステージング・サーバーで処理の対象となるテーブルのリストについては、「WebSphere Commerce ステージング・テーブル」を参照してください。

ステージング・サーバーで処理の対象となるテーブルに、操作テーブルに対する外部キー参照を入れないようにすることも重要です。もし入っている場合は、実動データベースから基本キーが削除されていることが原因で、伝搬は失敗するでしょう。ステージング・サーバーを使用する前に、操作データを所有しているのが、カタログ管理者などの個々のユーザーではなく組織であることを確認する必要があります。

ステージング・サーバーを使用する前に、以下の点に注意してください。

- ステージ・レコードによって参照されるすべての新しいイメージ・ファイル、HTML ファイル、または JSP ファイルは、ステージング・サーバーから実動サーバーに手動でコピーする必要があります。
- ステージング・サーバーは、データベース・スキーマの変更、イメージ・ファイル、HTML ファイルまたは JSP ファイルをコピーして伝搬することはできません。たとえば、新しい索引またはテーブルをステージング・データベースに作成する場合は、その索引とテーブルを実動データベースに手動で作成する必要があります。
- ステージ伝搬ユーティリティーは、ローダー・パッケージ (ロード・モード) または DB2 ロード・ユーティリティーによってロードされるレコードを伝搬できません。ステージング・トリガーは、これらの両方でバイパスされるからです。どちらかのユーティリティーを使用した場合は、ステージ・コピーを使用してデータベース・テーブルを再同期化し、新しいステージング・セッションを開始してください。ステージング・セッション中は、ステージング・データベースまたは実動データベース上で、ローダー・パッケージ (ロード・モード) または DB2 ロードを絶対に使用しないでください。
- ステージ・コピー・ユーティリティーを使用した後は、ステージング・サーバーを停止してから再始動する必要があります。
- ステージング・サーバーでは、DB2 Text Extender はサポートされません。
- ステージング・サーバー上では、STAGLOG テーブルをクリーンアップするという目的以外には、Database Cleanup ユーティリティーを実行しないでください。
- ステージング・サーバーの制限がいくらかあります。ステージング・サーバーを使用する前に、ステージング・サーバーの制限を理解しておいてください。詳しくは、39 ページの『ステージング・サーバーの制限』を参照してください。

ステージ・コピー・ユーティリティー: ステージ・コピー・ユーティリティー (iSeries では stagingCopy.sh) は、実動データベースからステージング・データベースにデータをコピーします。コピーできるのは、サイト関連のテーブル、マーチャント関連のテーブル、または個々のテーブルにあるデータです。コマンド構文で `cleanup_stage_db` パラメーターを使用することにより、ステージ・コピー・ユーティリティーの前にステージング・サーバーをクリーニングすることもできます。yes を指定すると、ステージ・コピー・ユーティリティーはデータをコピーする前に、すべてのステージング・テーブルをクリーニングします。これは、他のテーブルが削除カスケードによる影響を受ける場合があります。no を指定すると、ステージ・コピー・ユーティリティーはステージング・テーブルから何も削除しませ

ん。コピー・データとの競合が起きたり、基本キーまたは固有索引でキーが重複したりすると、コピーは失敗する可能性があります。ステージング・データベースをクリーニングするためだけにステージ・コピー・ユーティリティを使用する場合は、`only` を指定してください。

ステージ・コピー・ユーティリティとステージ伝搬ユーティリティは、サイト関連とマーチャント関連の 2 つの有効範囲レベルにデータベースのデータを分けます。サイトの有効範囲には、システム内のすべてのマーチャントに共通するデータが含まれます。たとえば、システムで使用される言語、および国または地域のコードなどです。マーチャントの有効範囲には、個々のマーチャントに関連したデータが含まれます。たとえば、ストア情報は個々のマーチャントに合わせてカスタマイズされますし、ストア・テーブルの行を各マーチャントごとの固有の行とすることもできます。サイト情報とマーチャント情報の両方が含まれているデータベース・テーブルもあります。ステージ・コピー・ユーティリティの `scope` パラメーターに `_all_` を指定すると、サイト・データ、全マーチャント・データの順にコピーされます。 `scope` に `_site_` を指定すると、サイト・データだけがコピーされ、 `_merchant_` を指定すると、マーチャント・データだけがコピーされます。個々のマーチャントのデータをコピーすることはできず、すべてのマーチャントのデータをコピーすることしかできません。 `scope` に `_all_` を設定しない場合は、マーチャント・データの前にサイト・データをコピーしてください。すべてのマーチャントがサイト・データを使用するからです。これを行わないと、外部キーと基本キーの間でミスマッチが生じ、コピーは失敗します。 `cleanup_stage_db` を使用してサイト・データをクリーニングすると、削除カスケードによってマーチャント・データも削除されてしまう可能性があります。ですから、まずマーチャント・データをクリーニングしてからサイト・データをクリーニングするようにしてください。それから `scope` に `_all_` を設定していない場合は、サイト・データ、マーチャント・データの順でコピーしてください。

ステージ・コピー・ユーティリティには、 `script_file` パラメーターというオプションもあります。スクリプト・ファイルの名前を指定してステージ・コピー・ユーティリティを実行すると、指定された有効範囲に基づいて実動データベースからステージング・データベースへコピーするためにエクスポートとインポートを使用する SQL スクリプト・ファイルが生成されます。それと同時に、削除ステートメントも生成されます。これは、デフォルト値を指定するか、 `cleanup_stage_db` に `yes` を指定した場合にステージング・データベースをクリーニングします。スクリプト・ファイルは、ステージ・コピー・ユーティリティを開始するディレクトリにあります。これは、エクスポートとインポートを使用することによりデータベース・コピー・プロセスを速めます。生成されたスクリプト・ファイルを変更することで、ステージ・コピー・ユーティリティの動作を変えることもできます。たとえば、スクリプト・ファイルを変更してインポート・ユーティリティの代わりに DB2 ロード・ユーティリティを使用するようにするなら、コピー・プロセスはさらに早まります。生成されたスクリプトは、ユーティリティを実行するディレクトリにテーブルをすべてエクスポートするので、十分なディスク・スペースを確保しておく必要があります。

トランザクションの有効範囲を理解しておくのは重要なことです。ステージ・コピー・ユーティリティは、ステージング・データベースをクリーニングするとき、各テーブルをクリーニングした後トランザクションをコミットし、データをコピーするときは、各テーブルをコピーして KEYS テーブルを同期した後トランザクシ

ンをコミットします。生成されたスクリプトのトランザクション有効範囲は、DB2 インポート・ユーティリティーのために若干異なります。DB2 インポート・ユーティリティーはインポートを終了した後自動的にトランザクションをコミットします。トランザクションのコミットは、KEYS テーブルを同期する前に行われます。ですから、KEYS テーブルの同期化は、別個のトランザクションで行われることとなります。

クリーニングまたはコピーするテーブルは、`dbtable` パラメーターを使用して指定できます。クリーニングまたはコピーするテーブルを指定する際に、テーブルを分けることはできません。テーブルの中には、参照制約によって互いに関連しているものがあります。指定したテーブルをクリーニングすると、削除カスケードによって子テーブルもクリーニングされます。指定したテーブルをコピーする場合は、まず親テーブルをコピーしてください。そうしないと、クリーニングやコピーは失敗してしまいます。

ステージ・コピー・ユーティリティーは構成および拡張可能です。カスタマイズされたテーブルを処理するために、それらのテーブルはいくつかの条件を満たしていなければなりません。それらの条件を、ステージング構成テーブルにセットアップする必要があります。ステージ・コピー・ユーティリティーを使用する前に、規定のステップに従ってデータベースを構成しなければなりません。ステージ・コピー・ユーティリティーは、コマンドが成功すると STAGLOG テーブルからすべてのレコードを削除します。

注: RFQ がシステム上に存在する場合、ステージ・コピー・コマンドは使用できません。詳しくは、39 ページの『ステージング・サーバーの制限』を参照してください。

ステージ検査ユーティリティー: 構成データと操作データが同じテーブルを共用すると、ステージング・データベースと実動データベースの間で固有索引キーの競合が生じる場合があります。変更を実動データベースに伝搬する前に、ステージ検査コマンド (iSeries の場合は `stagingCheck.sh`) を使用して、潜在的な固有索引の競合を判別し、伝搬の前に競合を訂正してください。

実動データベース上で RFQ を使用しており、ステージング・データベース上で契約を作成している場合、両方のデータベース上で同じテーブルを更新していることとなります。たとえば、サイト管理者が、ステージング・データベース上に契約を作成し、それをステージング・データベース上の TRADING テーブル (および他のテーブル) 内の 1 行に挿入するとします。同時に、ユーザーが、実動データベース上に RFQ を作成し、それを実動データベース上の TRADING テーブル (および他のテーブル) 内の 1 行に挿入します。TRADING テーブル内の 2 つの行の固有索引が同じ値になります。ステージング・テーブルから実動データベースに契約を伝搬すると、固有索引キーの競合が生じ、伝搬は失敗します。伝搬の前に、ステージ検査ユーティリティーを使用して、競合している固有索引キーを見つけてそれを修正してください。その後、変更を伝搬できます。

ステージ検査ユーティリティーを使用する際、`_unique_index_` に `-scope` パラメーターを指定すると、ステージング・データベース内の差分変更における潜在的なキーの競合がないか検査できます。挿入操作および更新操作を行う際は常に、STGUINDTAB テーブルで指定されたすべてのテーブルに対して、潜在的なキーの競合がないか検査が行われます。実動データベースとステージング・データベース

の間の競合を検査するために、各テーブルのすべての固有索引が網羅されます。潜在的なキーの競合が検出された場合は、テーブル名、固有索引、および競合するキーの値が報告されます。

ステージ検査コマンドは、データベースの変更は行いません。解決しなければならない潜在的なキーの競合を報告するだけです。このコマンドを使用する際は、ステージング・データベースに `-sourcedb` パラメーターを指定してください。ソース・データベースに実動データベースを指定すると、ステージ検査コマンドは正しく機能しません。

ステージ検査ユーティリティーは構成および拡張可能です。他にもテーブル (独自にカスタマイズしたテーブルも可) を `STGUINDTAB` テーブルに追加して、ステージ検査コマンドを実行して潜在的なキーの競合を検査することができます。

注: 絶対に構成データと操作データが同じテーブルを共有しないようにしてください。

ステージ伝搬ユーティリティー: ステージング・サーバー上のデータベース・レコードを変更し、テストした後、ステージ検査ユーティリティーを使用して潜在的な固有索引キーの競合を検査し、訂正します。これで、変更を実動データベースに伝搬する準備が整いました。

ステージ伝搬ユーティリティー (iSeries の場合は `stagingProp.sh`) は、ステージング・データベースから実動データベースに変更を移動します。ステージ伝搬ユーティリティーは `STAGLOG` テーブルを使用してステージング・データベース内の変更レコードを識別し、その後それらのレコードを実動データベース内で更新します。処理済みのレコードは、`STAGLOG` テーブル内の `STGPROCESSED` 列の `1` という印によって示されます。

`scope` パラメーターを指定すると、伝搬するデータのタイプを選択できます。`_site_` を設定すると、変更されたサイト・データがすべてステージング・テーブルから実動データベースに伝搬されます。`_merchant_` を設定すると、全マーチャントの変更データが伝搬されます。個々のマーチャント・データを伝搬することはできません。`_all_` を設定すると、サイト・データとマーチャント・データの両方が伝搬されます。

`dbtable` パラメーターを使用すると、特定のテーブルを伝搬できます。テーブルは必ず、親テーブルを伝搬した後で指定してください。

ステージ伝搬ユーティリティーのトランザクションの有効範囲は、ステージ・コピー・ユーティリティーと異なります。ステージ伝搬ユーティリティーの場合は、コマンドを実行する度に `1` トランザクションとして数えられます。たとえば、ステージ伝搬ユーティリティーの `scope` に `_site_` を指定すると、変更されたすべてのサイト・データに対して新規トランザクションを開始し、伝搬が成功した後トランザクションをコミットします。伝搬が失敗した場合、伝搬がロールバックされ、実動データベースは前と同じ状態に戻ります。

ステージ伝搬ユーティリティーは構成および拡張可能です。独自にカスタマイズしたテーブルを伝搬する前に、テーブルはある一定の条件を満たさなければなりません。詳しくは、『カスタマイズされたテーブルのためのステージング・サーバーの

構成』を参照してください。ステージ伝搬ユーティリティを使用する前に、規定のステップに従ってデータベースを構成しなければなりません。

データベースの構成

ステージ・コピー・ユーティリティ、ステージ伝搬ユーティリティ、または Database Cleanup ユーティリティを使用する前に、以下の手順に従ってください。

注:

1. 以下のステップは提案にすぎないので、ステージ・コピー・ユーティリティの使用で問題が発生する場合にのみ使用してください。

2. **▶ 400** によって示されない場合、以下は iSeries に適用されません。

1. PATH 環境変数を設定します。

2. **▶ Windows** **▶ AIX** **▶ Solaris** **▶ Linux** **▶ DB2** DB2 データベースを使用している場合、以下のコマンドを発行して、ステージング・データベースおよび実動データベースを構成します。

```
db2 update db config for db_name using LOGPRIMARY 80
db2 update db config for db_name using LOGBUFSZ 512
db2 update db config for db_name using DBHEAP 2048
db2 update db config for db_name using APPLHEAPSZ 2048
db2 update db config for db_name using PCKCACHESZ 8200
```

ここで、*db_name* は、ご使用のデータベースの名前です。

注:

a. デフォルトの STMTHEAP サイズは 60000 です。

b. デフォルトの LOCKLIST は 2400 です。

c. デフォルトの STAT_HEAP_SZ は 2048 です。

d. デフォルトの APP_CTL_HEAP_SZ は 4096 です。

3. **▶ Windows** **▶ AIX** **▶ Solaris** **▶ Linux** パフォーマンスの向上のため、バッファ・プールのサイズを大きくします。バッファ・プールの最適サイズは、ご使用の DB2 データベースのサイズと使用可能メモリーに基づいて決定してください。デフォルトのバッファ・プール・サイズを変更するには、次のコマンドを実行してください。

```
db2 connect to db_name
db2 alter bufferpool IBMDEFAULTBP size n
db2 terminate
```

ここで、*n* は、バッファ・プールの最適サイズです。

4. **▶ 400** secofr 権限、および 65535 以外の cssid を持つユーザー・プロファイルでログオンします。

カスタマイズされたテーブルのためのステージング・サーバーの構成

独自にカスタマイズしたデータベース・テーブルとともにステージング・サーバーを使用するには、以下の構成を行ってください。

1. 独自にカスタマイズしたテーブルの有効範囲 (サイト・データ、マーチャント・データ、またはその両方) を確認する。

2. テーブルの有効範囲および索引タイプに基づく対応するトリガー例を使用して、データベース・テーブル用のトリガーを作成する。
3. カスタマイズされたテーブルを STGSITETAB、STGMERTAB、および STGMRSTTAB テーブルに挿入する。
 - サイト・テーブルの場合は、STGSITETAB にのみ挿入する。
 - マーチャント・テーブルの場合は、STGMERTAB にのみ挿入する。
 - サイト・データとマーチャント・データの両方を含むテーブルの場合は、STGSITETAB、STGMERTAB、および STGMRSTTAB に挿入する。

注: すべてのテーブルを正しく挿入し、親テーブルの TABNBR 列の数を子テーブルの TABNBR 列の数よりも少なくしなければなりません。独自にカスタマイズしたテーブルが WebSphere Commerce テーブルの親テーブルとなる場合は、その TABNBR 列の数を子テーブルの TABNBR 列の数よりも少なくする必要もあります。

接続の構成

各種アダプターの構成について詳しくは、「*WebSphere Commerce 追加ソフトウェア・ガイド*」および「*WebSphere Commerce オンライン・ヘルプ*」を参照してください。

WebSphere Commerce Payments の構成

Payments を構成する前に、以下のようにします。

- WebSphere Commerce Payments が WebSphere Commerce インストールの一部となっていることを確認します。
- WebSphere Commerce Payments インスタンスを作成します。
- 構成マネージャーを使用して、WebSphere Commerce Payments インスタンスにカセットを追加します。
- WebSphere Commerce Payments インスタンス、および WebSphere Commerce インスタンスを開始します。
- マーチャント、およびそのマーチャントのマーチャント管理者を作成します。

カセットを構成して Payments を構成するには、マーチャント管理者として WebSphere Commerce Payments にログオンする必要があります。 WebSphere Commerce Payments は CustomOffline Cassette および OfflineCard Cassette とともにインストールされます。これらのカセットがサポートする最低のフレームワークは、WebSphere Commerce Payments バージョン 5.5 です。詳しくは、「*WebSphere Commerce インストール・ガイド*」を参照してください。

以下は、WebSphere Commerce と Payments のデフォルト・インストール・ディレクトリです。

表9. デフォルト・インストール・ディレクトリー

デフォルト・インストール・ディレクトリー	説明
<i>WC_installdir</i>	Windows: <i>drive:¥WebSphere¥CommerceServer55</i> AIX: /usr/WebSphere/CommerceServer55 Solaris、Linux: /opt/WebSphere/CommerceServer55 400: /QIBM/ProdData/CommerceServer55
<i>Payments_installdir</i>	Windows: <i>drive:¥WebSphere¥CommerceServer55¥payments</i> AIX: /usr/lpp/WebSphere/CommerceServer55/payments Solaris、Linux: /opt/WebSphere/CommerceServer55/payments 400: /QIBM/ProdData/CommercePayments/55

WebSphere Commerce Payments の構成について詳しくは、175 ページの『付録 C. WebSphere Commerce Payments チュートリアル』を参照してください。

第 4 章 ホストされるストアの管理

サイト管理者は、ホストされるストアでいくつかのタスクを実行する必要があります。この章では、以下のカテゴリでサイト管理者が実行するタスクについて概説します。

- 販売店の管理
- ディストリビューターの管理
- サイト管理

販売店の管理

ホストされるストアにおいて、サイト管理者には、販売店組織をセットアップし、販売店との関係を維持する責任があります。この調整には、以下のタスクが含まれます。

- 新規販売店組織の登録の E メール承認を送信する。

いったん販売店がその組織を登録したら、サイト管理者は、その組織に関する要求が承認されたことを通知する E メールを販売店に送信する必要があります。組織の登録の承認は、実際には組織管理コンソールで実行されます。組織管理コンソールと其中で実行されるタスクについて詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

いったん販売店組織が承認されると、デマンド・チェーン・ビジネス・モデルの販売店は、WebSphere Commerce アクセラレーターを使ってそのストアを変更できるようになります。ストアはそれぞれクローズ状態で作成されますが、販売店は WebSphere Commerce アクセラレーターを使ってストアに対する更新や変更を行うことができ、最後にはそれを開くことができます。ホスティング・ビジネス・モデルにおいて、ストアはそれぞれ中断状態で作成されるため、サイト管理者やチャネル・マネージャーがそれを再開する必要があります。この場合、ストアが再開されるまで、マーチャントはそのストアに対して変更を加えることができません。ビジネス・モデルについて詳しくは、「WebSphere Commerce 基本」、および「WebSphere Commerce ストア開発ガイド」を参照してください。WebSphere Commerce アクセラレーターについて詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

- マーチャントによってホストされるストアの承認を送信する (ホスティング・ビジネス・モデルを使用するストアの場合のみ)。

いったんホストされるストアをセットアップした後、サイト管理者はホストされるストアの要求承認を添付した E メールをマーチャントに送信します。この E メールには、新規のホストされるストアへのリンクと、その管理方法に関する指示も含まれている必要があります。

注: 販売店組織の管理について詳しくは、「WebSphere Commerce オンライン・ヘルプ」の『販売店組織の管理』を参照してください。

ディストリビューターの管理

ホストされるストアにおいて、サイト管理者には、ディストリビューターをセットアップし、そのディストリビューターと販売店との関係を作成する責任があります。この管理には、以下のタスクが含まれます。

- 新規ディストリビューターを作成する。

サイト管理者は、ディストリビューター・サービス契約を作成することによって、新規ディストリビューターを作成することができます。契約に追加すべき情報は、IBM と直接連絡を取ることによって得られます。ディストリビューター・サービス契約について詳しくは、「WebSphere Commerce オンライン・ヘルプ」を参照してください。

注: このタスクはチャンネル・マネージャーによって実行することもできます。

- ディストリビューター設定を変更する。

サイト管理者は、時々、ディストリビューター設定の一部、たとえばディストリビューターとの接続に関するタイムアウト設定などの変更要求を受け取ることがあります。ディストリビューター設定は、SQL ステートメントを介してデータベースにアクセスすることによって変更できます。詳しくは、「WebSphere Commerce ストア開発ガイド」を参照してください。

- Adapter for CrossWorlds の構成およびインストールを行う。

Adapter for CrossWorlds は、WebSphere Commerce と、ホストされるストアの外部システムおよびバックエンド・システムの間での対話を確認するために使用されるコネクタです。サイト管理者には、販売店サイトとディストリビューター・サイトの間での接続を確認するためにアダプターを構成し、インストールする責任があります。WebSphere MQ を使った WebSphere Commerce と SAP R/3 バックエンド・システムの間での対話も作成され、サイト管理者はこれをセットアップする必要があります。IBM CrossWorlds および WebSphere MQ について詳しくは、「WebSphere Commerce 追加ソフトウェア・ガイド」を参照してください。

サイト管理

ホストされるストアに固有のサイト管理タスクには、以下のタスクが含まれます。

- ステージング・サーバーを操作する。詳しくは、38 ページの『ステージング・サーバー』を参照してください。
- DBClean を操作する。詳しくは、107 ページの『Database Cleanup ユーティリティ』を参照してください。
- WebSphere Commerce アナライザーを操作する。詳しくは、WebSphere Commerce のインストール CD を参照してください。
- WebSphere Commerce サーバーを始動および停止する。詳しくは、13 ページの『WebSphere Commerce の開始と停止』を参照してください。

第 5 章 動的キャッシング

このセクションでは、WebSphere Commerce サブレットまたは JavaServer Pages (JSP) 結果の動的キャッシングについて説明します。

WebSphere Commerce サブレットまたは JSP 結果のキャッシング

顧客がリンクをクリックして商品またはカテゴリ・ページを表示する場合、ほとんどの時間は、HTTP 要求の構文解析、データベースへのアクセス、および動的にページを作成することに費やされます。サイト・トラフィックが過密な場合や、データベースにある商品やカテゴリ・エントリーが多い場合は、サブレットまたは JavaServer Pages (JSP ファイル) のロードにかかる時間がさらに増えるかもしれません。

サーバーで処理される HTTP 要求の大半は、カタログ情報に関するものです。WebSphere Commerce コマンド — `CategoryDisplay`、`ProductDisplay`、`TopCategoriesDisplay`、および `StoreCatalogDisplay` — は、データベースから情報を検索し、その結果を JSP ファイルとして表示します。前回表示したときからカタログ情報が変わっていないなら、次回顧客が要求を出す際、そのサブレットまたは JSP ファイルを再実行する必要はありません。キャッシュに保管されている同等の静的サブレットまたは JSP ファイルを使用したほうが早いかもしれません。Web モジュールの `WEB-INF` ディレクトリーにある `cachespec.xml` ファイル内のキャッシュ・エントリー・エレメントを定義することによって、サブレットまたは JSP のキャッシングを構成することができます。

アクセスされるページに対応するキャッシュ・エントリーがメモリーにない場合、ページが動的に生成されます。そしてページはメモリーに保管され、基になっているデータが変更されるまで、再生成する必要がなくなります。

注: WebSphere Commerce 動的キャッシングと URL 再書き込みは同時に使用できません。URL 再書き込みがオンになっている場合、WebSphere Commerce 動的キャッシングを使用不可にする必要があります。サブレットまたは JSP ファイルをキャッシュすべきではありません。URL 再書き込みについて詳しくは、「*WebSphere Commerce セキュリティー・ガイド*」のセッション管理に関する章を参照してください。

WebSphere Application Server 動的キャッシュ

WebSphere Commerce は、サブレットまたは JSP ファイル、および WebSphere Application Server `CacheableCommand` インターフェースから拡張するコマンドをキャッシュするための WebSphere Application Server 動的キャッシュ・サービスを使用します。WebSphere Commerce インスタンスの作成中、動的キャッシュ・サービス、サブレット・キャッシング、およびディスク・オフロードは、デフォルトでは使用可能になっています。

WebSphere Application Server 動的キャッシュについて詳しくは、「WebSphere Application Server InfoCenter」
(<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae>)

/infocenter-ja/infocenter/index.html) で、『Improving performance through the dynamic cache service』 のトピックを参照してください。

動的キャッシュ

サーブレットまたは JSP ファイル結果のキャッシングによって、アプリケーションのパフォーマンスが向上します。WebSphere Application Server は、サーブレット、Web サービス、および WebSphere コマンドを含む、いくつかのキャッシング・アクティビティを動的キャッシュと呼ばれる 1 つのサービスに統合します。これらのキャッシング・アクティビティは、共に作業してアプリケーションのパフォーマンスを向上し、アプリケーション・サーバーの動的キャッシュ・サービスで設定される多くの構成パラメーターを共用します。

メモリー内のキャッシュからの要求に対処することによって、動的キャッシュを使用して、サーブレットおよび JSP ファイルのパフォーマンスを向上させることができます。キャッシュ・エントリーにはサーブレット出力、サーブレット実行結果、およびメタデータが含まれます。

詳しくは、「WebSphere Application Server InfoCenter」(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) で、『Improving performance through the dynamic cache service』 のトピックを参照してください。

動的キャッシュ・サービスおよびサーブレット・キャッシングの使用可能化

キャッシングを使用可能にするには、動的キャッシュ・サービスを使用可能にしてサーブレット・キャッシングを構成しなければなりません。これらのステップの実行について詳しくは、「WebSphere Application Server InfoCenter」(<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter-ja/infocenter.html>) で、『Enabling globally the dynamic cache service』 および 『Configuring servlet caching』 を参照してください。

動的キャッシュ・モニターの使用可能化

WebSphere Application Server は、キャッシュ統計、Edge Side およびディスク統計、キャッシュ・エントリー、依存 ID、およびキャッシュ・ポリシー情報を表示する動的キャッシュ・モニター・アプリケーションを提供します。WebSphere Application Server キャッシュのコンテンツおよび動作を検査するには、WebSphere Application Server 動的キャッシング・モニターをインストールしなければなりません。動的キャッシュ・モニターを使用するには、動的キャッシングを使用するそれぞれのアプリケーション・サーバーに、`WAS_installdir` ディレクトリーの下に `installableApps` サブディレクトリーにある `CacheMonitor.ear` ファイルをインストールする必要があります。

WebSphere Commerce の場合、セキュリティ上の理由で、動的キャッシュのモニターとして、仮想ホスト `VH_instance_name` の代わりに、`VH_instance_name_admin` (例: `VH_demo_admin`) を使用することをお勧めします。仮想ホスト名は、WebSphere Application Server 管理コンソールから変更することができます。

Web ブラウザーを使用して、以下の Web アドレスから Web アプリケーションにアクセスできます。

`http://host_name:port/cachemonitor`

しかし、さらに安全なアクセスを望まれる場合には、管理ホスト・マシンにアクセスすることをお勧めします。

`https://admin_host_name:port/cachemonitor`

たとえば、仮想ホスト `VH_instance_name_admin` がキャッシュ・モニターのインストールに使用される場合は、以下のようにキャッシュ・モニターにアクセスすることができます。

`https://admin_host_name:8002/cachemonitor`

注:

1. それぞれの WebSphere Commerce インスタンスごとに 1 つの動的キャッシュ・モニターを起動する必要があります。
2. キャッシュ・モニターを使用して、WebSphere Application Server EJB セキュリティーを使用可能にしたい場合は、「WebSphere Commerce セキュリティー・ガイド」のセクション『動的キャッシュ・モニターのセキュリティ構成』で説明されているように追加のセットアップをいくつか実行する必要があります。

動的キャッシュ・モニターのインストールについて詳しくは、「WebSphere Application Server InfoCenter」

(<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter-ja/infocenter/index.html>) で、トピック『Displaying cache information』を参照してください。

WebSphere 動的キャッシュ・モニターの依存 ID のツリー・ビュー

WebSphere Commerce は「ツリー・ビュー (Tree View)」ツール、メモリー内の依存 ID のツリー・ビューを提供するキャッシュ・モニターの拡張機能を提供します。

「ツリー・ビュー (Tree View)」ツールは依存 ID のツリー・ビューを提供します。ツリー・ノードを拡張表示または縮小表示して、階層ビューを取得し、そのノードの下のキャッシュ・エントリー・グループを無効にすることができます。オリジナルのビューには処理を行いませんが、代わりに新規リンクが「依存 ID ツリー・ビュー (Dependency IDs Tree View)」として表示されます。

+ 「ツリー・ビュー (Tree View)」ツールのインストールおよび使用について詳しく
+ は、以下のディレクトリーにある README ファイルで『Tree View for
+ Dependency IDs for WebSphere Dynamic Cache Monitor』を参照してください。

[▶ AIX](#) [▶ 400](#) [▶ Linux](#) [▶ Solaris](#)

`WC_installdir/features/dynacache/CacheMonitor/`

[▶ Windows](#) `WC_installdir¥features¥dynacache¥CacheMonitor¥`

キャッシュ可能なオブジェクトの構成

キャッシュ可能なオブジェクトが、Web アプリケーション・アーカイブ (WAR) WEB-INF または enterprise bean WEB-INF ディレクトリーにある `cachespec.xml` ファイル内に定義されます。グローバル `cachespec.xml` ファイルをアプリケーション・サーバーのプロパティー・ディレクトリーに置くことができますが、お勧めす

る方法は、キャッシュ構成ファイルとデプロイメント・モジュールを一緒に置くことです。 `cachespec.xml` ファイルのルート・エレメントは `<cache>` であり、`<cache-entry>` エレメントが含まれます。

サーブレットまたは JSP 結果キャッシングのキャッシュ・エントリーを指定するには、以下のセクションを `cachespec.xml` ファイルに追加します。

```
+ <cache-entry>
+   <class>servlet</class>
+   <name>name</name>
+   ...
+ </cache-entry>
```

`name` は、サーブレットまたは JSP の相対 Web パスまたはサーブレット・マッピングです。

コマンド・キャッシングのキャッシュ・エントリーを指定するには、以下のセクションを `cachespec.xml` ファイルに追加します。

```
+ <cache-entry>
+   <class>command</class>
+   <name>name</name>
+   ...
+ </cache-entry>
```

`name` はコマンド・クラスへの完全パスです。例:

`com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl`。

`<cache-entry>...</cache-entry>` エレメント内で、キャッシュ ID、依存 ID、および無効化規則を作成することができます。オブジェクトをキャッシュするには、WebSphere Application Server は、そのオブジェクトの別の呼び出しに対して固有の ID を生成する必要があります。 `<cache-id>` エレメントはこのタスクを実行します。各キャッシュ・エントリーは、規則が空ではないキャッシュ ID を戻すか、または実行する規則がなくなるかのいずれかまで、順に実行する複数のキャッシュ ID 規則を持つことができます。有効なキャッシュ ID を生成するキャッシュ ID 生成規則がない場合、オブジェクトはキャッシュされません。

+ **注:** このリリースでは、コマンド・キャッシングのキャッシュ・エントリーを指定
+ する場合に、 `<invalidate>...</invalidate>` を使ったコマンド無効化しかサ
+ ポートしません。(コマンド・キャッシングに `<cache-id>...</cache-id>` また
+ は `<dependency-id>...</dependency-id>` エレメントを使用することはできま
+ せん。)

```
+ <cache-entry>
+   <class>command</class>
+   <name>name</name>
+   <invalidate>
+   ...
+   </invalidate>
+ </cache-entry>
```

動的キャッシュはこの `cachespec.xml` ファイル内の変更に応答します。新規バージョンの `cachespec.xml` が検出されると、古いポリシーは置換されます。古いポリシー・ファイルによってキャッシュされたオブジェクトは、自動的にキャッシュから無効になるわけではありません。それらは新規ポリシーとともに再利用されるか、置換アルゴリズムによってキャッシュから除去されます。 `cachespec.xml` ファイルについて詳しくは、「WebSphere Application Server InfoCenter」

(<http://www-3.ibm.com/software/webervers/appserv/doc/v40/ae/infocenter-ja/infocenter/index.html>) で、トピック 『Cachespec.xml file』 を参照してください。さらに、ストアを作成する場合の追加考慮事項については、「*WebSphere Commerce* ストア開発ガイド」を参照してください。

キャッシングのデフォルト

インストール中、*WebSphere Commerce* は以下のデフォルトを使用してキャッシング・システムをセットアップします。

- 動的キャッシングは使用可能。
- ディスク・オフロードは使用可能。
- サブレット・キャッシングは使用可能。
- それぞれのビジネス・モデルのデフォルト・キャッシュ・ポリシー (サンプルとして提供される) は、以下の表示コマンドに対して JSP ファイルをキャッシュします。
 - CategoryDisplay
 - ProductDisplay
 - StoreCatalogDisplay
 - TopCategoriesDisplay

キャッシュ・エントリーを以下から手動でマージする必要があります。

```
WC55_installdir/samples/dynacache/business_model/cachespec.xml
```

マージ先:

```
WAS_installdir/installedApps/cell_name/WC_instance_name.ear/Stores.war/WEB-INF/cachespec.xml
```



```
WC55_installdir%samples%dynacache%business_model%cachespec.xml
```

マージ先:

```
WAS_installdir%installedApps%cell_name%WC_instance_name.ear%Stores.war%WEB-INF%cachespec.xml
```

ストア内でのキャッシュ・ポリシーの使用可能化については、「*WebSphere Commerce* サンプル・ストア・ガイド」または「*WebSphere Commerce* ストア開発ガイド」を参照してください。

全ページおよびフラグメントのキャッシング

WebSphere Application Server 動的キャッシュが JSP ファイルをキャッシュする方法は、JSP が書き込まれる方法に基づいています。特定の *WebSphere Commerce* コマンドのページ出力が URL パラメーターおよび要求属性に基づいて常に同じ結果を出す場合、このページ出力をプロパティ・エレメントを使用してキャッシュ・エントリーでキャッシュすることができます。consume-subfragments (CSF) とともに *WebSphere Commerce* コントローラー・サブレット (com.ibm.commerce.server.RequestServlet) をサブレット名とします。キャッシュ・エントリーをこの方法で定義すると、ページ出力は 全ページ (full page) キャ

ッシングとして知られる方法でキャッシュされます。コントローラー・サーブレットで `consume-subfragments` を使用することの大きな利点はパフォーマンスですが、このメカニズムを使用して Web ページをキャッシュする場合、ページ出力は個別設定情報を持つことはできません。

ページ出力にはユーザー依存のセクションがあり、ページ出力はフラグメント (*fragment*) キャッシングとして知られる方法でキャッシュされます。つまり、JSP ページは別々のキャッシュ・エンタリーとしてキャッシュされ、要求されると再アセンブルされます。フラグメント (JSP) キャッシングの場合、WebSphere Commerce がコマンドを実行してどの JSP を実行したらよいか判別してから、動的キャッシング・メカニズムは JSP をキャッシュから取り出せるかどうかを判別します。この方法の利点は、異なるキャッシュ・エンタリーを再組み立てして、ユーザー情報に基づいてページを作成できるため、柔軟であるということです。

全ページ・キャッシング

プロパティ・エレメント `consume-subfragments` (CSF) を使用すると、親エンタリー (CSF でマークされている) は、すべてのフラグメントからのコンテンツをそのキャッシュ・エンタリー内に組み込むため、結果として、組み込みまたは転送はないがエンタリーの全体のツリーからのコンテンツを持つ 1 つの大きなキャッシュ・エンタリーになります。

サーブレットがキャッシュされると、そのサーブレットのコンテンツのみが保管されます。キャッシュには、組み込まれるかまたは転送される、他のフラグメントのプレースホルダーが含まれます。Consume-subfragments (CSF) は、子サーブレットが含まれる場合、キャッシュがコンテンツの保管を停止しないように伝えます。親エンタリー (CSF としてマークされている) は、すべてのフラグメントからのコンテンツをそのキャッシュ・エンタリー内に組み込むため、結果として、組み込みまたは転送はないがエンタリーの全体のツリーからのコンテンツを持つ 1 つの大きなキャッシュ・エンタリーになります。これによって、かなりの量のアプリケーション・サーバー処理を節約できますが、これは通常、組み込まれるフラグメントのツリー全体の判別に必要なすべての情報が外部 HTTP 要求に含まれる場合にのみ役立ちます。

たとえば、以下のように `<cache-entry>` が定義される場合。

```
<cache-entry>
  <class>servlet</class>
  <name>com.ibm.commerce.server.RequestServlet.class</name>
  <property name="consume-subfragments">true</property>
  <property name="save-attributes">false</property>
  <property name="store-cookies">false</property>

  <!-- StoreCatalogDisplay?storeId=<storeId> -->
  <cache-id>
    <component id="" type="pathinfo">
      <required>true</required>
      <value>/StoreCatalogDisplay</value>
    </component>
    <component id="storeId" type="parameter">
      <required>true</required>
    </component>
  </cache-id>
</cache-entry>
```

+

+

save-attributes プロパティが false に設定されると、その要求属性はキャッシュ・エントリーとともに保管されません。store-cookies プロパティが false に設定されると、その要求 cookie はキャッシュ・エントリーとともに保管されません。

上の例では、キャッシュ・サブレット・エントリーには、StoreCatalogDisplay コマンドによって転送される JSP ファイル StoreCatalogDisplay.jsp の使用済みの組み込みが含まれます。

フラグメント・キャッシュ

動的に組み込まれるそれぞれの JSP ファイルは、それ自身の <cache-entry> を cachespec.xml ファイル内に定義して、動的キャッシュが要求を受け取った場合に取り出せるようにする必要があります。そうでない場合、動的に組み込まれるそれぞれの JSP ファイルが要求ごとに再実行されます。たとえば、StoreCatalogDisplay.jsp が動的に header.jsp および footer.jsp を組み込み、StoreCatalogDisplay.jsp に対して <cache-entry> のセットアップのみを行う場合を考慮します。次に、StoreCatalogDisplay ページを要求すると、header.jsp および footer.jsp ファイルはキャッシュされない場合に実行します。以下は、StoreCatalogDisplay.jsp のための <cache-entry> を定義する方法の例です。

```
<cache-entry>
  <class>servlet</class>
  <name>/ToolTech/ShoppingArea/CatalogSection/CategorySubsection/StoreCatalogDisplay.jsp</name>
  <property name="save-attributes">>false</property>

  <cache-id>
    <component id="storeId" type="parameter">
      <required>>true</required>
    </component>
    <component id="catalogId" type="parameter">
      <required>>false</required>
    </component>
  </cache-id>
</cache-entry>
```

キャッシュ・サブレット・フィルター操作

WebSphere Commerce バージョン 5.5 より前では、WebSphere Commerce はそれ自身のキャッシング・メカニズムを提供していました。以前のメカニズムを使用すると、Web ページは以下の 2 つの方法に基づいてキャッシュすることができます。

- セッション独立 (SI) キッシング - ページは URL パラメーターに基づいてキャッシュされる
- セッション依存 (SD) キッシング - ページは URL パラメーター、ユーザーの言語、希望する通貨、親組織、契約 ID、およびメンバー・グループに基づいてキャッシュされる

SI キッシングのキャッシュ ID は URL パラメーターに基づいて生成され、また SD キッシングの場合、キャッシュ ID は URL パラメーターとセッション情報を使用して作成されます。

以前のセッション依存キャッシングと同じ機能性を WebSphere Application Server 動的キャッシング・メカニズムを使用して提供するために、WebSphere Commerce 5.5 はキャッシュ・フィルターとして知られるサブレット・フィルターを導入しています。このキャッシュ・フィルターは、動的キャッシュがキャッシュ ID を構成

するのに使用するセッション情報から、要求属性をセットアップするのに使用されます。セッション情報は WebSphere Commerce サーバー・ランタイムによって設定されるため、キャッシュ・フィルターは Web サイトに対する 2 番目の要求まで要求属性のすべてを設定することはできません。

以下の表には、キャッシュ・フィルターによって設定される要求属性がリストされています。

表 10. キャッシュ・フィルターによって設定される要求属性

要求属性	説明
DC_curr	ユーザーの希望する通貨
DC_lang	ユーザーの希望する言語
DC_porg	ユーザーの親組織
DC_cont	ユーザーの現在の契約
DC_mg	ユーザーの明示的メンバー・グループ
DC_storeId	ストア ID
DC_userId	ユーザー ID

ユーザーは複数の契約を実行する資格があり、複数のメンバー・グループに属することができるため、要求属性 DC_cont および DC_mg には複数の値が含まれる場合があります。こうしたユーザーの場合、値はソートされ、区切り文字としてセミコロン (;) で連結されます。さらに、複数の契約およびメンバー・グループ要求属性が定義されます。(たとえば、DC_cont0、DC_cont1、...DC_contN。ここで、N はユーザーが権限を持つ契約の数です。) たとえば、ユーザーが契約 10004 および 10005 の実行資格を持っている場合、以下の要求属性がセットアップされます。DC_cont は 10004;10005、DC_cont0 は 10004、DC_cont1 は 10005 です。

要求属性 DC_cont の設定の目的は、限定された数のコンポーネントを持つキャッシュ ID の構造を許可することであり、個別の要求属性 DC_cont0、DC_cont1、...、DC_contN の設定の目的は、さらに詳細なキャッシュ無効化のために依存 ID の構造を許可することです。

メンバー・グループ情報はセッション・データの一部ではないため、キャッシュ・フィルターは、この情報をユーザー ID に基づいてデータベースから検索する必要があります。データベース照会が繰り返されることによる性能低下を避けるために、キャッシュ・フィルターは、WebSphere Command Caching を使用して、このタスクを完了します。WebSphere Command Framework から直接拡張する新規コマンド - com.ibm.commerce.dynacache.command.MemberGroupsCacheCmdImpl - を使用して、ユーザー ID に基づいてユーザーが属するメンバー・グループをキャッシュします。WebSphere Commerce は以下のキャッシュ・エントリーを cachespec.xml に追加して、このコマンドをキャッシュする動的キャッシング・サービスを通知します。

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl</name>
  <cache-id>
    <component type="method" id="getUserId">
      <required>true</required>
    </component>
  </cache-id>
```



```

<dependency-id>DC_userId</dependency-id>

<dependency-id>DC_userId
  <component type="method" id="getUserId">
    <required>true</required>
  </component>
</dependency-id>
</cache-entry>

```

WebSphere Application Server 動的キャッシングは、キャッシュ ID としての GET メソッド getUserId() に基づいて、クラス MemberGroupsCacheCmdImpl のコマンド・オブジェクトをキャッシュします。このため、次回に同じユーザー ID のメンバー・グループに対する要求がある際に、動的キャッシングはすでに計算されたメンバー・グループを持つキャッシュされたコマンド・オブジェクトを戻します。このメンバー・グループ情報は、フィルター呼び出しをキャッシュして、このユーザーのメンバー・グループを要求属性 DC_mg に設定する、コマンド・メソッド getMemberGroups() を使用すると即座にアクセス可能になります。

注: WebSphere Commerce は、WebSphere Command Caching を上記のキャッシュ・フィルター MemberGroupsCacheCmdImpl などとともに内部で使用します。しかし、WebSphere Commerce は、WebSphere Commerce コマンドの WebSphere Command Caching を公式にはサポートしません。

ストア・ページのキャッシング・ストラテジー

ストア・ページを設計する際のキャッシング考慮事項について詳しくは、「WebSphere Commerce ストア開発ガイド」の『ストア・ページのキャッシング』を参照してください。これには、以下の問題に関するガイダンスが含まれます。

- どのページをキャッシュするか?
- ページ全体またはページ・フラグメントをキャッシュすべきか?
- キャッシング・ストラテジーの開発

さらに、「WebSphere Commerce ストア開発ガイド」で、キャッシング時のストア関係の含意を説明しているセクションを参照してください。

WebSphere Commerce におけるキャッシュ無効化メカニズム

WebSphere Application Server 動的キャッシュ・サービスによって生成されたキャッシュ・エントリには、サーブレット出力、サーブレット実行の結果 (他のサーブレットまたは JSP ファイルなど) に関する情報、およびタイムアウトやエントリ優先順位情報といったエントリに関するメタデータが含まれます。時々 WebSphere Commerce は、キャッシュ内にとどまっている間に、この情報が最新で整合性があり、正確であることを検査する必要があります。このため、WebSphere Commerce は、もはや有効ではないキャッシュ情報を識別して除去するための効果的なメカニズムを必要とします。WebSphere Application Server 動的キャッシュ・サービスは、キャッシュ無効化 と呼ばれるプロセスを使用して、キャッシュ情報を維持するためのサポートを提供します。キャッシュ無効化によって、WebSphere Commerce はサービスが提供する異なる無効化メカニズムを活用して、異なる状況の下で無効化を実行できるようになります。これらのメカニズムはルール・ベース、タイム・ベース、グループ・ベースから、プログラマチック・アプローチに変わります。

このセクションは、WebSphere Commerce でのキャッシュ無効化を実行する方法の説明のみを提供しており、一般的なキャッシングの各局面を網羅するためのものではありません。ここでは、WebSphere Application Server 動的キャッシュによって提供される無効化メカニズムについて説明します。

- cachespec.xml で無効化ポリシーを定義する
- WebSphere Application Server によって提供されるキャッシュ・モニターを使用してキャッシュ・エントリーを除去する
- WebSphere Commerce アプリケーション内で、WebSphere Application Server 動的キャッシュ無効化 API をプログラマチックに呼び出す

cachespec.xml で無効化ポリシーを定義する

WebSphere Application Server 動的キャッシングは、cachespec.xml ファイルを使用することによってキャッシュ無効化の構成をサポートします。Web アプリケーション・アーカイブ (WAR) WEB-INF または enterprise bean WEB-INF ディレクトリーにある、キャッシュ可能なオブジェクトの構成に使用する、同じ cachespec.xml で無効化ポリシーを動的に定義できます。cachespec.xml 構文について詳しくは、「WebSphere Application Server InfoCenter」(<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter-ja/infocenter/index.html>) で、トピック『cachespec.xml file』を参照してください。

対応するキャッシュ・エントリーを無効にする必要のあるイベントを定義する <invalidation> タグ、またはキャッシュ・エントリーを保持する期間を構成する <timeout> タグのあるキャッシュ・エントリー内で、無効化規則を指定できます。これらは依存 ID と全く同じ方法で定義され、無効化規則によって生成されるその ID を使用して、同じ依存 ID を持つキャッシュ・エントリーを無効にします。キャッシュ・エントリーごとに複数の無効化規則が存在することができます。すべての無効化規則は別々に実行します。

以下のセクションでは、cachespec.xml にあるこれらの無効化規則の構文について説明し、WebSphere Commerce で以下の規則を適用する方法についての例を示します。

- サブレット・ベースの無効化規則
- コマンド・ベースの無効化規則
- タイムアウトおよび優先順位の無効化規則

サブレット・ベースの無効化規則

WebSphere Application Server 動的キャッシングによって提供されるこの無効化の方法は、サブレット要求の実行時に他のキャッシュ・エントリーを無効にします。

cachespec.xml 内の構文:

```
<cache-entry>
  <class>servlet</class>
  <name>servlet_name</name>
  <invalidation>invalidation_id
    <component id="" type="pathinfo">
      ...
    </component>
  </invalidation>
</cache-entry>
```

注:

<class>

"servlet" の値は、この無効化がサーブレットによってトリガーされることを示しています。

<invalidation>

無効化をトリガーするイベント、および指定された ID を持つどのキャッシュ・エントリーを無効にするかを識別するのに使用するエレメント。

pathinfo

WebSphere Commerce では、すべてのサーブレット要求は、次のいずれかの単一コントローラー・サーブレットを通過します。

com.ibm.commerce.server.RequestServlet.class または

com.ibm.commerce.tools.common.ToolsRequestServlet.class。

固有の要求 URI を識別するには、"pathinfo" コンポーネント・タイプを使用して、pathinfo 名を要求からフィルター操作することができます。

cachspec.xml で定義されたポリシーの例: InterestItemDelete URL を発行してカタログ・エントリーをリストから削除する際に、参照番号によって識別される、興味のあるアイテム・リストのキャッシュ・ページを無効にするには、以下のようになります。

```
<cache-entry>
  <class>servlet</class>
  <name>com.ibm.commerce.server.RequestServlet.class</name>

  <invalidation>listId
    <component id="" type="pathinfo" ignore-value="true">
      <required>true</required>
      <value>/InterestItemDelete</value>
    </component>
    <component id="listId" type="parameter">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>
```

コマンド・ベースの無効化規則

この方法では、他のキャッシュ・エントリーをコマンド実行の副次作用として無効にします。これは、WebSphere Commerce でコマンド・レベルで無効化をトリガーするためのイベント・ドリブン・メカニズムを提供します。ユーザーはコマンドに書き込まれたビジネス・ロジックを調べたり、コマンドの正常な実行に影響を与える可能性のある特定のキャッシュ・オブジェクトを識別して、それに応じて無効化規則を定義することができます。このようなコマンドの 1 つの基準は、WebSphere Commerce プログラミング・モデルをインプリメントして、無効化 ID が入力パラメーターに基づいて生成される場合には、入力パラメーターを処理するコマンド内のメソッドおよびフィールドを外部化する必要があるということです。

このサポートについて詳しくは、WebSphere Application Server の資料を参照してください。98 ページの『コマンド・ベースの無効化のためのコマンドを書く』も参照してください。

cachspec.xml 内の構文:

```

<cache-entry>
  <class>command</class>
  <name>fully_qualified_class_name_of_the_command</name>
  <invalidation>invalidation_rules</invalidation>
</cache-entry>

```

注:

<class>

command の値は、この無効化がコマンドによってトリガーされることを示しています。

<name>

コマンドの完全修飾パスが含まれます。

cachespec.xml で定義されたポリシーの例: CatalogEntryUpdate コマンドを呼び出して、カタログ・エントリーを変更する際に、参照番号でキャッシュされた商品表示ページを無効にするには、以下のようにします。

```

<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.catalogmanagement.commands.CatalogEntryUpdateCmdImpl</name>

  <invalidation>productId
    <component id="getCatEntryId" type="method">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>

```

タイムアウトおよび優先順位ベースの無効化規則

ポリシー内の <timeout> タグを使用したタイムアウト・メカニズムは、事前設定期間の後にキャッシュ・エントリーを除去する簡単な方法を提供します。このメカニズムは、キャッシュ・エントリーを除去する時刻が指定された特定の無効化規則またはトリガー・イベントのセットアップを実行できない場合に、役に立ちます。

WebSphere Application Server 動的キャッシング・サービスは、キャッシュがいっぱいになると、変更された最低使用頻度 (LRU) アルゴリズムに基づいてキャッシュ・エントリーを自動的に除去します。その際、<priority> タグを使用して加重する優先順位を使用して、どのエントリーをキャッシュから除去するかを判別します。他のキャッシュ・エントリーに対して特定のキャッシュ・エントリーの優先順位の値が高くなればなるほど、このキャッシュ・エントリーが他のキャッシュ・エントリーに比べてキャッシュ内に長くとどまる傾向があります。

キャッシュがいっぱいになった場合、メモリーからエントリーを除去せずに、ディスク・オフロードを構成して、後で使用するためにエントリーをファイル・システムにコピーすることができます。(ロケーションは構成可能です。) また、キャッシュ・エントリーは、シャットダウンされたサーバー上のディスクにオフロードされ、サーバーが再始動された場合に、再利用することができます。加えて、以下の動的キャッシュ・サービス設定も無効化プロセスに影響します。

キャッシュ・サイズ

キャッシュが保持する最大エントリー数を決定します。

デフォルト優先順位

デフォルトで、エントリーがフル・キャッシュ内にとどまる時間を決定します。

ディスク・オフロード

ディスク・オフロードを使用可能にするかどうか、またディスク・オフロードのロケーションを指定します。

cacheSpec.xml 内の構文:

```
<cache-entry>
  .....
  <cache-id>
    <component >.....</component>
    <priority>priority_value</priority>
    <timeout>time_in_cache</timeout>
  </cache-id>
</cache-entry>
```

cacheSpec.xml で定義されたポリシーの例: ショッピング・カートのキャッシュ・コンテンツを、作成または使用してから 3600 秒後に除去するには、以下のようになります。

```
<cache-id>
  <component id="" type="pathinfo">
    <required>true</required>
    <value>/OrderItemDisplay</value>
  </component>
  <component id="orderId" type="parameter">
    <required>true</required>
  </component>
  <timeout>3600</timeout>
</cache-id>
```

キャッシュ・モニターを使用してキャッシュ・エントリーを除去する

WebSphere Application Server キャッシュ・モニターは、キャッシュ・エントリーを手動で無効にするオプションを提供します。キャッシュ・コンテンツ全体を空にする「**キャッシュのクリア (Clear Cache)**」ボタンがあります。それぞれのキャッシュ・エントリーを個別に無効にしたり、キャッシュ・エントリーのグループをテンプレートとして無効にすることができます (例: ProductDisplay.jsp に関連したすべてのエントリー)。依存 ID で識別されるキャッシュ・エントリーのグループもモニター上のブロックとして無効にすることができます。

動的キャッシュ無効化 API を呼び出す

WebSphere Application Server 動的キャッシングは以下の API を提供して、プログラマチックな無効化をサポートします。

- com.ibm.websphere.cache.invalidateById
- com.ibm.websphere.cache.invalidateByTemplate

これらの API の使用について詳しくは、WebSphere Application Server の資料を参照してください。

WebSphere Commerce はこれらの API を利用して、ホームメイドの無効化ラッパー、DynaCacheInvalidation コマンドを提供して、WebSphere Commerce 内のテーブル・ドリブン無効化 API をサポートします。これは、CACHEIVL テーブル内のレコードを処理することによって、キャッシュ・オブジェクトを無効にします。詳しくは、102 ページの『**キャッシュ無効化 API (DynaCacheInvalidation)**』を参照してください。

コマンド・ベースの無効化のためのコマンドを書く

コマンド呼び出しを動的キャッシュによってインターセプトするには、コマンドを WebSphere Command Framework に書き込み、そのインプリメンテーション・クラスを CacheableCommandImpl (com.ibm.websphere.command パッケージ内にある) から継承する必要があります。コマンド・ベースの無効化のためにコマンド書き込みを単純化するには、WebSphere Commerce は抽象クラス ControllerCommandImpl および TaskCommandImpl を更新して、CacheableCommandImpl から継承し、これらの抽象クラスから継承するコマンドが CacheableCommandImpl から継承してコマンド・ベースの無効化に適するようにします。

こうしたコマンドを書き込む場合、どれが無効化 ID であるかを知ること、またコマンドへの呼び出しをインターセプトする無効化規則を理解することも重要です。無効化 ID は、コマンド内に存在するメソッドおよびフィールドに基づいて、入力パラメーターとして生成されるため、無効化 ID の構成に必要なすべてのメソッドがコマンド・インターフェースで提供され、インプリメントされる必要があります。

WebSphere Commerce でコマンド無効化を使用した例

以下の例は、WebSphere Commerce はコマンド無効化を使用する方法を示しています。特定のメンバー・グループに属している特定のメンバーを削除するコマンド DeleteMemberGroupMemberCmdImpl が正常に実行されると、動的キャッシュは無効化規則で定義されたキャッシュ・エントリを無効にします。この例では、これは "DC_userId:userId" として定義され、userId は getMemberId メソッドから戻されている値です。(WebSphere Commerce でのメンバー・グループ情報のキャッシングについて詳しくは、91 ページの『キャッシュ・サブレット・フィルター操作』を参照してください。) 例: DC_userId:-1000, DC_userId:-1001 など。このコマンドには、削除中のユーザー ID を検索する GET メソッド getMemberId() があり、このメソッドを使用して、ユーザー ID に基づいて依存 ID を持つどのキャッシュ・エントリを削除するかを計算します。同じ論理は、GET メソッド getMemberId() を持つコマンド AddMemberGroupMemberCmdImpl にも適用されます。

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.membergroup.commands.AddMemberGroupMemberCmdImpl</name>
  <name>com.ibm.commerce.membergroup.commands.DeleteMemberGroupMemberCmdImpl</name>

  <invalidation>DC_userId
    <component type="method" id="getMemberId">
      <required>true</required>
    </component>
  </invalidation>
</cache-entry>
```

注: 上記の無効化規則はすべて、サンプル・キャッシュ・ポリシー内で WebSphere Commerce とともに提供されます。WebSphere Commerce インストール・ディレクトリーの /samples/dynacache/invalidation サブディレクトリーの下に、無効化規則のサンプルをさらに見い出すことができます。cachespec.xml ファイルに無効化規則を取り込む方法について詳しくは、README ファイルの『Sample invalidation cache policies for Dynacache』という項を参照してください。

キャッシュ無効化の例

以下の例では、cachespec.xml ファイルにキャッシング・ポリシーをセットアップして、WebSphere Commerce で消費者向けビジネス・モデル用に ProductDisplay JSP ページをキャッシュする方法や、同じ XML ファイル内の無効化規則を定義することによってキャッシュ・エントリを無効にする方法を示します。この例では、JSP ファイル用のキャッシュ ID 生成規則とともに複数の依存 ID を定義します。それぞれの依存 ID は、キャッシュ・エントリがそれぞれ異なる状況で更新されたときに、キャッシュ・エントリを無効するために使用されます。この例では、ProductDisplay JSP を無効にするのに必要なポリシーのサブセットのみを表示します。完全な例および詳細情報については、*WC_installdir/samples/dynacache/invalidation* ディレクトリーにある README ファイルを参照してください。

```
<cache>
<cache-entry>
  <class>servlet</class>
  <name>/FashionFlow/ShoppingArea/CatalogSection/CatalogEntrySubsection/ProductDisplay.jsp</name>
  <property name="save-attributes">>false</property>

  <!-- Cache ProductDisplay.jsp -->
  <cache-id>
    <component id="storeId" type="parameter">
      <required>>true</required>
    </component>
    <component id="catalogId" type="parameter">
      <required>>true</required>
    </component>
    <component id="productId" type="parameter">
      <required>>true</required>
    </component>
    <component id="DC_lang" type="attribute">
      <required>>true</required>
    </component>
    <component id="DC_curr" type="attribute">
      <required>>true</required>
    </component>
    <component id="DC_porg" type="attribute">
      <required>>true</required>
    </component>
    <component id="DC_cont" type="attribute">
      <required>>true</required>
    </component>
    <component id="DC_mg" type="attribute">
      <required>>true</required>
    </component>
  </cache-id>

  <!-- Used for invalidating the product display cache entry -->
  <!-- that belongs to a specific store -->
  <dependency-id>storeId
    <component id="storeId" type="parameter">
      <required>>true</required>
    </component>
  </dependency-id>

  <!-- Used for invalidating the cache entry of a specific product -->
  <dependency-id>productId
    <component id="productId" type="parameter">
      <required>>true</required>
    </component>
  </dependency-id>
</cache-entry>
</cache>
```



```

    </dependency-id>

    <!-- Used for invalidating the product display cache entry -->
    <!-- that belongs to a specific catalog in the store -->
    <dependency-id>storeId:catalogId
<component id="storeId" type="parameter">
<required>true</required>
    </component>
    <component id="catalogId" type="parameter">
    <required>true</required>
    </component>
</dependency-id>

    <!-- Used for invalidating the product display cache entry -->
    <!-- that is under a specific contract -->
    <dependency-id>contractId
    <component id="DC_cont0" type="attribute">
    <required>true</required>
    </component>
</dependency-id>

</cache-entry>

<cache-entry>
<class>command</class>
<sharing-policy>not-shared</sharing-policy>
    <name>com.ibm.commerce.catalogmanagement.commands.AddCatalogDescCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.UpdateCatalogDescCmdImpl</name>

    <!-- ***** -->
<!-- Invalidate all the product page cache entries that -->
<!-- might be affected when the catalog description is changed -->
<!-- ***** -->
<invalidation>storeId:catalogId
    <component id="getStoreId" type="method">
    <required>true</required>
    </component>
    <component id="getCatalogId" type="method">
    <required>true</required>
    </component>
</invalidation>

</cache-entry>

<cache-entry>
<class>command</class>
<sharing-policy>not-shared</sharing-policy>
    <name>com.ibm.commerce.catalogmanagement.commands.ListpriceAddCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.ListpriceDeleteCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.ListpriceUpdateCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.OfferAddCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.OfferDeleteCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.OfferUpdateCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.ProductAttributeUpdateCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.AttributeValueUpdateCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.AddListpriceCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.DeleteListpriceCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.UpdateListpriceCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.AddOfferCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.DeleteOfferCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.UpdateOfferCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.UpdateAttributeCmdImpl</name>
    <name>com.ibm.commerce.catalogmanagement.commands.UpdateAttributeValueCmdImpl</name>

    <!-- ***** -->
<!-- Invalidate the specific product page cache entry when the -->
<!-- product is updated -->

```



```

        <!-- ***** -->
        <invalidation>productId
            <component id="getCatentryId" type="method">
                <required>true</required>
            </component>
        </invalidation>

    </cache-entry>

<cache-entry>
    <class>command</class>
    <sharing-policy>not-shared</sharing-policy>
    <name>com.ibm.commerce.contract.commands.ContractSuspendCmdImpl</name>
    <name>com.ibm.commerce.contract.commands.ContractTCDeployCmdImpl</name>

    <!-- ***** -->
    <!-- Invalidate all the product page cache entries under a -->
    <!-- specific contract -->
    <!-- ***** -->
    <invalidation>contractId
        <component id="getContractId" type="method">
            <required>true</required>
        </component>
    </invalidation>
</cache-entry>

<cache-entry>
    <class>command</class>
    <name>com.ibm.commerce.tools.devtools.store.commands.StoreProfileUpdateCmdImpl</name>
    <name>com.ibm.commerce.tools.devtools.flexflow.ui.commands.impl.FlexflowUpdateCmdImpl</name>
    <name>com.ibm.commerce.store.commands.StoreOpenCmdImpl</name>
    <name>com.ibm.commerce.store.commands.StoreCloseCmdImpl</name>

    <!-- ***** -->
    <!-- Invalidate all the product page cache entries that -->
    <!-- belong to the store when the store is updated -->
    <!-- ***** -->
    <invalidation>storeId
        <component id="getStoreId" type="method">
            <required>true</required>
        </component>
    </invalidation>

</cache-entry>

<cache-entry>
    <class>command</class>
    <sharing-policy>not-shared</sharing-policy>
    <name>com.ibm.commerce.catalogimport.commands.CatalogImportJobAddCmd</name>

    <!-- ***** -->
    <!-- Invalidate all the product page cache entries that -->
    <!-- belong to the store when the store catalog is updated -->
    <!-- ***** -->
    <invalidation>storeId
        <component id="getStoreId" type="method">
            <required>true</required>
        </component>
    </invalidation>
</cache-entry>

</cache>

```

キャッシュ無効化 API (DynaCacheInvalidation)

WebSphere Commerce スケジューラーは、DynaCacheInvalidation コマンドを定期的に呼び出して、CACHEIVL テーブル内のレコードを処理してから、動的キャッシュ無効化 API を呼び出してキャッシュ・エントリーを無効にします。startTime はチェックポイント・ファイルから取得されます。チェックポイント・ファイルは instance_name/cache ディレクトリーに書き込まれます。このディレクトリーは、DynaCacheInvalidation が初めて実行されるときに作成されます。最終レコードのタイム・スタンプは、コマンドの最終段階の間にチェックポイント・ファイルに書き込まれます。

CACHEIVL テーブル

CACHEIVL テーブルを処理する際の DynaCacheInvalidation コマンドの規則は以下のとおりです。

- テンプレート列が設定されると、DynaCacheInvalidation コマンドは動的キャッシュ無効化 API (invalidateByTemplate) を呼び出して、テンプレート ID としてその名前を使用します (例:
/webapp/wcs/stores/ToolTech/ShoppingArea/CatalogSection/
CategorySubsection/StoreCatalogDisplay.jsp)。
- DATA_ID 列が設定され、テンプレート名が設定されないと、DynaCacheInvalidation コマンドは、動的キャッシュ無効化 API (invalidateById) を呼び出して、DATA_ID を ID として使用します (例:
StoreCatalogDisplay:storeId:10151)。
- 動的キャッシュ無効化 API を呼び出すと、それはキャッシュ・エントリーを無効にします。

外部キャッシュ内の無効化

結果キャッシュを、IBM HTTP Server や Edge Server といった外部キャッシュにブッシュすることができます。外部キャッシュで無効化が生じるのは以下の場合です。

- キャッシュが混雑している
- タイムアウトが発生している
- 無効化メッセージが WebSphere Application Server から送信された

Web サーバー・プラグイン・キャッシュ

WebSphere Application Server 5.0 では、Web サーバー・プラグインには Edge Side Include (ESI) プロセッサーが組み込まれています。この ESI プロセッサーには、ページ全体やフラグメントをキャッシュする機能があり、より高いキャッシュ・ヒット率を提供します。ESI プロセッサーによってインプリメントされるキャッシュは、ディスク・キャッシュではなくメモリー内のキャッシュにあります。このため、キャッシュ・エントリーは、Web サーバーの再始動時に保管されません。

Web サーバー・プラグインによって要求を受け取ると、ESI プロセッサーが使用不可でない限り、それは ESI プロセッサーに送信されます。(デフォルトでは使用可

能になっています。) キャッシュ・ミスが発生すると、Surrogate-Capabilities ヘッダーが要求に追加され、その要求は WebSphere Application Server に転送されます。動的サーブレット・キャッシュがアプリケーション・サーバー内で使用可能であり、応答が Edge 対応可能な場合、アプリケーション・サーバーは WebSphere プラグインに対して、Surrogate-Control ヘッダーを戻します。

ESI プロセッサーは、WebSphere Web サーバー・プラグイン構成ファイル plugin-cfg.xml を通して構成可能です。以下に例を示します。

```
<Property Name="esiEnable" Value="true"/>
<Property Name="esiMaxCacheSize" Value="1024"/>
<Property Name="esiInvalidationMonitor" Value="true" />
```

ここで、

- esiEnable を使用して、値を FALSE にすることによって ESI プロセッサーを使用不可にします。デフォルトでは、ESI は使用可能です。
- esiMaxCacheSize は、1 キロバイト単位の最大キャッシュ・サイズです。デフォルトの最大キャッシュ・サイズは 1 MB です。キャッシュがいっぱいになると、キャッシュから除去される最初のエントリーは、有効期限に最も近いエントリーです。(デフォルトの ESI キャッシュ・エントリーの有効期限は 24 時間です。)
- esiInvalidationMonitor は、ESI プロセッサーがアプリケーション・サーバーから無効化を受け取るかどうかを指定します。

キャッシュ・エントリーが ESI キャッシュから除去される方法には以下の 3 つがあります。

- エントリーの有効期限タイムアウトを呼び出す。
- エントリーがパージされることにより、新規エントリー用の余地を作る。
- アプリケーション・サーバーが、エントリーのグループに対して明示的な無効化を送信する。

このメカニズムを使用可能にするには、esiInvalidationMonitor を true に設定し、installableApps ディレクトリーにある DynaCacheEsi.ear アプリケーションをアプリケーション・サーバー上にインストールする必要があります。

注: DynaCacheEsi.ear は、VH_instance_name を仮想ホストとして使用すべきです。

ESI プロセッサーについて詳しくは、「WebSphere Application Server InfoCenter」(<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter-ja/infocenter/index.html>) で、トピック『Configuring Edge Side Include caching』を参照してください。

単純ファイル・サーブレット

デフォルトでは、WebSphere Commerce はイメージや HTML といった静的データをキャッシュしません。これは、システム・プロパティ `com.ibm.servlet.file.esi.timeOut` を値 0 (ゼロ) に設定していることによるものです。イメージおよび HTML のキャッシングを使用可能にしたい場合、以下のいずれかを実行できます。

- キャッシュ・エントリーを SimpleFileServlet 用に cachespec.xml ファイルにセットアップします。キャッシュ・エントリーを cachespec.xml にセットアップすると、WebSphere Application Server からキャッシュされるものを制御することによって、必要なときにキャッシュ・エントリーを無効にすることができるようになるため、アプローチが改善されます。

ほとんどすべてのイメージをキャッシュ・エントリーとしてアプリケーション・サーバーに保管していると、最大キャッシュ・エントリー数に達した場合に、パフォーマンスが犠牲になる可能性があることに注意してください。

SimpleFileServlet キャッシュ・エントリーには、他のキャッシュ・エントリーと比べて低い優先順位の値を設定することをお勧めします。

- システム・プロパティ `com.ibm.servlet.file.esi.timeOut` の設定値をゼロ以外のタイムアウト値に変更します。

一部のイメージ・ファイルで無効化が必要な理由は、ビジネス・モデルの中にロゴやバナーを簡単に変更できるものがあるためです。

以下に例を示します。

```
<cache-entry>
  <class>Servlet</class>
  <name>com.ibm.ws.webcontainer.servlet.SimpleFileServlet.class</name>
  <property name="EdgeCacheable">true</property>
  <cache-id>
    <component id="" type="pathinfo">
      <required>true</required>
    </component>
  </cache-id>
  <dependency-id>
    <component id="" type="pathinfo">
      <required>true</required>
    </component>
  </dependency-id>
  <timeout>1800</timeout>
</cache-entry>
```

以下は、イメージの無効化のためのキャッシュ・エントリーの例です。

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.tools.devtools.store.commands.StoreLogoUpdateCmdImpl</name>
  <invalidation>FashionFlow/images/logo.gif</invalidation>
</cache-entry>
```

```
<cache-entry>
  <class>command</class>
  <name>com.ibm.commerce.tools.devtools.store.commands.StoreBannerUpdateCmdImpl</name>
  <invalidation>FashionFlow/images/banner.gif</invalidation>
</cache-entry>
```

問題判別

動的キャッシングを使用して、ページを正常にキャッシュする際に問題が起きた場合には、以下の情報を参考にしてください。

- **問題:** キャッシュ・モニターにアクセスできません。

解決策: Web サーバー・プラグインを再生成して、Web サーバーとアプリケーション・サーバーを再始動し、新規の仮想ホストおよびポート番号のマッピングがロードされるようにします。

- **問題:** キャッシュするページがありません。

解決策: cachespec.xml ファイルが WebSphere Application Server にロードされているかどうか調べてください。 cachespec.xml がロードされていれば、サイド・バーの「**キャッシュ・ポリシー (Cache Policies)**」リンクをクリックすることによって、ポリシーはキャッシュ・モニターから表示可能になるはずです。さらに、WebSphere Application Server SystemOut.log ファイルが以下のような行を表示します。

```
[6/6/03 21:21:12:635 EDT] 65c60609 ConfigManager I DYNA0047I: Successfully loaded
cache configuration file D:\WebSphere\AppServer5\installedApps\buzz\WC_demo.ear\
Stores.war\WEB-INF\cachespec.xml.
```

上記の行は表示する目的で分割されていることに注意してください。

- **問題:** ページの一部しかキャッシュされません。

解決策: おそらく cachespec.xml ファイルに問題があります。トレースをオンにしてから、調査中のページにアクセスすることができます。このトレースによって、問題が示されるはずです。

トレースの使用可能化について詳しくは、69 ページの『コンポーネントのトレースの使用可能化』を参照してください。

動的キャッシュ・トレース (WebSphere Commerce サイド) をオンにするには、**Trace Specification** に "com.ibm.websphere.commerce.WC_CACHE=all=enabled" が含まれていなければなりません。

動的キャッシュ・トレース (WebSphere Application Server サイド) をオンにするには、**Trace Specification** に "com.ibm.ws.cache.*=all=enabled" が含まれていなければなりません。

- **問題:** DynaCacheInvalidation コマンドが無効になりません。

解決策: このコマンドが URL から機能しているかどうか調べてください。このコマンドを URL から実行する必要がある場合には、cacheinvalidation.jsp ファイルは docpath の中になければなりません。これがパス内にない場合でも、コマンドは実行しますが、完了したことをレポートする表示ページを受け取りません。URL が正しく呼び出されると、コマンドが WebSphere Commerce スケジューラーを通して実行するようスケジュールされているかどうかを調べることができます。デフォルトでは、実行するようスケジュールされていません。

- **問題:** 依存 ID による無効化が正常に動作しません。

解決策: 対象とするサーブレットまたは JSP ファイルの依存 ID が、同じキャッシュ・エントリー・ブロック内にあるかどうかを調べてください。そうである場合、異なるキャッシュ・エントリー・ブロック内で同じ依存 ID を複数定義して、同一の定義を持つようにする必要があります。つまり、依存 ID が以下のようにいくつかのキャッシュ・エントリーに対して定義されている場合、他のキャッシュ・エントリー・ブロックにある依存 ID の定義も全く同じでなければなりません。

```
<dependency-id>storeId
  <component id="storeId" type="parameter">
    <required>true</required>
  </component>
</dependency-id>
```






第 6 章 システムの保守





この章では、Database Cleanup ユーティリティについて、また Database Cleanup ユーティリティを使用してシステム上のデータベースを保守する方法について説明します。

Database Cleanup ユーティリティ

Database Cleanup ユーティリティを使用することにより、データベースから複数のオブジェクトを同時に削除することができます。データベース内の多くの情報を変更したり、未使用のテーブルや行がある場合に、これを行うことができます。データベースから削除できるオブジェクトにはいくつかの種類があります。これらのオブジェクトについて詳しくは、118 ページの『データベースのクリーンアップ』を参照してください。

Database Cleanup ユーティリティがオブジェクトを削除する場合、データベースの参照保全を保持するために、オブジェクトのテーブル内のレコードは削除されません。Database Cleanup ユーティリティ・コマンドは、トップダウンまたはボトムアップのいずれかの方法でデータベースをクリーンアップします。トップダウンは削除カスケードで子テーブルからすべての行を削除します。削除ルールが参照保全で指定されている場合、削除カスケードは失敗します。ボトムアップ法を使用することをご検討ください。

    ボトムアップ法を使用するには、コマンド構文の FORCE パラメーターに対して `yes` を指定します。これにより子テーブルをまず削除し、親テーブルがその後続くこととなります。

    ボトムアップ法をトリガーする別の方法は、コマンド構文で LOGLEVEL パラメーターを 2 と指定することです。0 を指定すると何も記録されません。1 を指定すると先頭テーブルからの削除ステートメントのみ記録されます。LOGLEVEL 2 を指定すると、削除されるそれぞれの子テーブルからの削除ステートメントが記録され、これが先頭テーブルに至るまで続きます。2 を選択するとボトムアップ法がトリガーされますが、参照保全に削除ルールが存在する場合は正常な削除を保証できません。削除ルールのあるレコードを削除するには、FORCE パラメーターに `yes` と指定してください。

テーブルに多くの子テーブルが含まれている場合、ボトムアップ法ではより長い応答時間が予想されます。例えば、MEMBER テーブルには 500 以上の子テーブルが含まれています。パフォーマンス上の理由から、トップダウン法を使用することをお勧めします。

- + **注:** Loglevel 0 または Loglevel 1 が失敗する場合にだけ、Loglevel 2 を使用するようしてください。FORCE パラメーターを `no` に設定すると失敗する場合にだけ、FORCE パラメーターを `yes` に設定するようしてください。

Database Cleanup ユーティリティは構成、拡張、適合が可能です。事前設定のクリーンアップ構成は別として、新規オブジェクトをデータベース・テーブルに追加

し、どのテーブルや行をクリーンアップするかを定義できます。『Database Cleanup ユーティリティに新規構成を追加する』を参照してください。

新規テーブルを作成してデータベース・スキーマを拡張した場合、新規テーブルをクリーンアップするために Database Cleanup ユーティリティを使用できます。データベース・スキーマを変更した場合（新規列をテーブルに追加する、外部キーと基本キーの関係を変更する、新規子テーブルを参照保全パスに追加するなど）は、Database Cleanup ユーティリティは自動的に変更に対応します。列名を変更する場合は、CLEANCONF テーブルの構成データを更新してください。

Database Cleanup ユーティリティは、データベース・スキーマの参照保全定義の削除規則に基づき、子テーブルのレコードを削除します。削除規則を `on delete cascade`、`on delete set null`、または `on delete restrict` に設定できます。新規テーブルを追加する場合は、参照保全および削除規則が適切に定義されているかを確認してください。適切に定義されていないならば、Database Cleanup ユーティリティは新規テーブルを処理できません。

注: ステージング・サーバー上では、`staglog` オブジェクトをクリーンアップするという目的でのみ、Database Cleanup ユーティリティを実行してください。ステージング・データベースは実動データベースとは異なります。ステージング・データベースは構成データのみ持っており、運用データは持っていません。構成データを削除すると、運用データに対し削除カスケードが実行されます。ステージ伝搬ユーティリティが実動データベースに削除を伝搬する場合、運用データ（保存しておきたいデータ）に対しカスケード削除が実行されることがあります。構成データをクリーンアップするには、実動データベースで Database Cleanup ユーティリティを実行してください。

Database Cleanup ユーティリティのオブジェクト

Database Cleanup ユーティリティは CLEANCONF テーブルを参照して、特定のオブジェクトやオブジェクト・タイプが指定された場合にどのテーブルおよび行を削除するかを決定します。以下の表は、CLEANCONF テーブルの事前に構成された削除シナリオを示しています。CLEANCONF テーブルに同様の行を追加することにより、削除オブジェクトを構成できます。

オブジェクト	タイプ	ステートメント
account	obsolete	<code>delete from account where markfordelete = 1 and trdtype_id = 0 and trading_id not in (select account_id from trading) and trading_id not in (select distinct account_id from ordpaymthd)</code>
address	obsolete	<code>delete from address where status = 'T' and (days(CURRENT_TIMESTAMP) - days(lastcreate)) >= ? and (address_id not in (select distinct address_id from orderitems where address_id is not null)) and (address_id not in (select distinct address_id from orders where address_id is not null)) and (address_id not in (select distinct allocaddress_id from orderitems where allocaddress_id is not null))</code>

オブジェクト	タイプ	ステートメント
atp_inventory	obsolete	delete from receipt where qtyonhand = 0 and qtyinkits = 0 and receipt_id not in (select distinct receipt_id from ordpickhst where receipt_id is not null) and receipt_id not in (select distinct receipt_id from ordshiphst where receipt_id is not null)
attachment	obsolete	delete from attachment where days(current timestamp) - days(timeupdated) >=? and attachment_id not in (select attachment_id from trdattach)
auction	retracted	delete from auction where austatus = 'R' and (days(CURRENT TIMESTAMP) - days(updatetime)) >= ?
auction	settlement_closed	delete from auction where austatus = 'SC' and (days(CURRENT TIMESTAMP) - days(updatetime)) >= ?
auctionlog	obsolete	delete from auctionlog where (days(CURRENT TIMESTAMP) - days(actiontime)) >= ?
autobidlog	obsolete	delete from autobidlog where (days(CURRENT TIMESTAMP) - days(actiontime)) >= ?
baseitem	obsolete	delete from baseitem where markfordelete = 1 and baseitem_id not in (select baseitem_id from catentry) and baseitem_id not in (select distinct baseitem_id from itemspc where markfordelete = 0 and itemspc_id in (select distinct itemspc_id from orderitems) or itemspc_id in (select distinct itemspc_id from oicomplst) or itemspc_id in (select distinct itemspc_id from versionspc where versionspc_id in (select distinct versionspc_id from receipt)) or itemspc_id in (select distinct itemspc_id from radetail) or itemspc_id in (select distinct itemspc_id from bkordalloc) or itemspc_id in (select distinct itemspc_id from invreserve) or itemspc_id in (select distinct itemspc_id from rmaitem) or itemspc_id in (select distinct itemspc_id from rmaitemcmp) or itemspc_id in (select distinct itemspc_id from catentry))
bidlog	obsolete	delete from bidlog where (days(CURRENT TIMESTAMP) - days(actiontime)) >= ?
calculation_code	obsolete	delete from calcode where published = 2 and calcode_id not in (select distinct calcode_id from ordadjust where calcode_id is not null) and calcode_id not in (select distinct calcode_id from stencalusg where calcode_id is not null) and calcode_id not in (select distinct calcode_id from ordcalcd where calcode_id is not null) and calcode_id not in (select distinct calcode_id from ordicalcd where calcode_id is not null)
catalog_group	obsolete	delete from catgroup where markfordelete = 1

オブジェクト	タイプ	ステートメント
catentry	without_orderitems	delete from catentry where markfordelete = 1 and(days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and catentry_id not in (select distinct catentry_id from auction) and catentry_id not in (select distinct catentry_id from orderitems where catentry_id is not null) and catentry_id not in (select distinct catentry_id from oicomplis where catentry_id is not null) and catentry_id not in (select distinct catentry_id from rmaitem where catentry_id is not null) and catentry_id not in (select distinct catentry_id from offer where offer_id in (select distinct offer_id from orderitems))
catentry	without_orderitems-iitems	delete from catentry where markfordelete = 1 and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and catentry_id not in (select distinct catentry_id from auction) and catentry_id not in (select distinct catentry_id from orderitems) and catentry_id not in (select distinct catentry_id from oicomplis where catentry_id is not null)and catentry_id not in (select distinct catentry_id from iitem) and catentry_id not in (select distinct catentry_id from rmaitem where catentry_id is not null) and catentry_id not in (select distinct catentry_id from offer where offer_id in (select distinct offer_id from orderitems))
contract	obsolete	<ol style="list-style-type: none"> 1. delete from trading where markfordelete = 1 and trdtype_id = 1 and trading_id not in (select distinct trading_id from orderitems) and trading_id not in (select distinct trading_id from rma) and trading_id not in (select distinct trading_id from ordpaymthd) and trading_id not in (select distinct account_id from ordpaymthd) 2. delete from productset where markfordelete = 1 and productset_id not in (select distinct productset_id from tradeposcn where tradeposcn_id in (select distinct tradeposcn_id from offer where offer_id in (select distinct offer_id from orderitems))) 3. delete from tradeposcn where markfordelete = 1 and tradeposcn_id not in (select distinct tradeposcn_id from offer where offer_id in (select distinct offer_id from orderitems))
coupon_promotion	expired	delete from cppm where days(current timestamp) - days(enddate) >=?
cpgnlog	obsolete	delete from cpgnlog
cpgnstats	obsolete	delete from cpgnstats
expected_inventory_records	obsolete	delete from ra where markfordelete = 1 and ra_id not in (select distinct ra_id from receipt, radetail where receipt.radetail_id = radetail.radetail_id)

オブジェクト	タイプ	ステートメント
expected_inventory_ record_details	obsolete	delete from radetail where markfordelete = 1 and radetail_id not in (select distinct radetail_id from receipt)
forummsg	obsolete	delete from forummsg where msgstatus = 'D' or (days(CURRENT TIMESTAMP) - days(posttime)) >= ?
fulfillment_center	obsolete	delete from ffmcenter where markfordelete = 1 and ffmcenter_id not in (select distinct ffmcenter_id from radetail) and ffmcenter_id not in (select distinct ffmcenter_id from inventory) and ffmcenter_id not in (select distinct ffmcenter_id from rma) and ffmcenter_id not in (select distinct ffmcenter_id from orderitems) and ffmcenter_id not in (select distinct allocffmc_id from orderitems) and ffmcenter_id not in (select distinct ffmcenter_id from store) and ffmcenter_id not in (select distinct rtnffmctr_id from store) and ffmcenter_id not in (select distinct ffmcenter_id from receipt) and ffmcenter_id not in (select distinct ffmcenter_id from auction) and ffmcenter_id not in (select distinct ffmcenter_id from auctionlog)
inventory_adjustments	obsolete	delete from invadjust where days(CURRENT TIMESTAMP) - days(adjustmentdate) >= ?
inventory_adjustment_ codes	obsolete	delete from invadjcode where markfordelete = 1 and invadjcode_id not in (select distinct invadjcode_id from invadjust)
itemspecification	obsolete	delete from itemspc where markfordelete = 1 and itemspc_id not in (select distinct itemspc_id from orderitems) and itemspc_id not in (select distinct itemspc_id from oicomplis) and itemspc_id not in (select distinct itemspc_id from versionspc where versionspc_id in (select distinct versionspc_id from receipt)) and itemspc_id not in (select distinct itemspc_id from radetail) and itemspc_id not in (select distinct itemspc_id from bkordalloc) and itemspc_id not in (select distinct itemspc_id from invreserve) and itemspc_id not in (select distinct itemspc_id from rmaitem) and itemspc_id not in (select distinct itemspc_id from rmaitemcmp) and itemspc_id not in (select distinct itemspc_id from catentry)
message	obsolete	delete from message where message_id not in (select message_id from msgmemrel) or (days(CURRENT TIMESTAMP) - days(posttime)) >= ?
msgmemrel	obsolete	delete from msgmemrel where message_id in (select m.message_id from message ms, msgmemrel m where ms.message_id = m.message_id and (status = 'D' or ((status = 'O' or sendstat = 'S') and (days(CURRENT TIMESTAMP) - days(posttime)) >= ?)))
order	canceled	delete from orders where status = 'X' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?

オブジェクト	タイプ	ステートメント
order	completed	delete from orders where status = 'C' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?
order	deposited	delete from orders where status = 'D' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?
order	shipped	delete from orders where status = 'S' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?
order	stale_guest	delete from orders where (status = 'P' or status = 'I' or status = 'W' or status = 'N') and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and orders.member_id in (select distinct users_id from users where registertype = 'G') and (orders_id not in (select distinct orders_id from orderitems where inventorystatus != 'NALC' and inventorystatus is not null))
order	stale_non_guest	delete from orders where (status = 'P' or status = 'I' or status = 'W' or status = 'N') and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and orders.member_id in (select distinct users_id from users where registertype != 'G') and (orders_id not in (select distinct orders_id from orderitems where inventorystatus != 'NALC' and inventorystatus is not null))
organization	obsolete	delete from member where member_id in (select orgentity_id from orgentity where orgentity_id = ?)
pastats	obsolete	delete from pastats
pcstats	obsolete	delete from pcstats
pestats	obsolete	delete from pestats
policy	obsolete	delete from policy where days(current timestamp) - days(endtime) > ? and policy_id not in (select distinct policy_id from ordpaymthd) and policy_id not in (select distinct policy_id from rma)
rfq	obsolete	delete from trading where markfordelete = 1 and trdtype_id in (2, 3, 4) and trading_id not in (select distinct trading_id from orderitems) and trading_id not in (select distinct trading_id from rma) and trading_id not in (select distinct trading_id from ordpaymthd) and trading_id not in (select distinct account_id from ordpaymthd)
rma	abandoned	delete from rma where status in ('PRC', 'EDT') and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id not in (select rma_id from rmaitem where rmaitem.status in ('APP', 'MAN')) and rma_id not in (select rma_id from rtnreceipt)
rma	approved_or_partly_ approved	delete from rma where status = 'PND' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id in (select rma_id from rmaitem where rmaitem.status in ('APP', 'MAN')) and rma_id not in (select rma_id from rtnreceipt)
rma	canceled	delete from rma where status = 'CAN' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ?

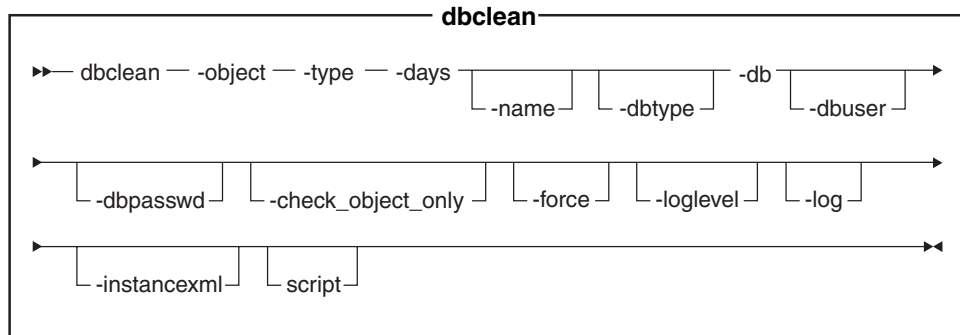
オブジェクト	タイプ	ステートメント
rma	completed	delete from rma where status = 'CLO' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id not in (select rma_id from rtnreceipt)
rma	not_approved	delete from rma where status = 'PND' and (days(CURRENT TIMESTAMP) - days(lastupdate)) >= ? and rma_id not in (select rma_id from rmaitem where rmaitem.status in ('APP', 'MAN')) and rma_id not in (select rma_id from rtnreceipt)
rtnreasons	obsolete	delete from rtnreason where markfordelete = 1 and rtnreason_id not in (select distinct rtnreason_id from rtnrcptdsp) and rtnreason_id not in (select distinct rtnreason_id from rmaitem)
sastats	obsolete	delete from sastats
staglog	obsolete	delete from staglog where stgprocessed = 1 and (days(CURRENT TIMESTAMP) - days(stgstmp)) >= ?
store	obsolete	delete from storeent where storeent_id = ? and type='S'
users	guest	delete from member where member_id in (select users_id from users where registertype='G' and (days(CURRENT TIMESTAMP) - days(lastsession)) >= ? And (users_id not in (select member_id from orders where status != 'Q')) and (users_id > 0) and (users_id not in (select member_id from address where address_id in (select address_id from orderitems where address_id is not null and status != 'Q') or address_id in (select allocaddress_id from orderitems where allocaddress_id is not null and status != 'Q') or address_id in (select address_id from orders where address_id is not null and status !='Q'))))
users	registered	delete from member where member_id in (select users_id from users where registertype= 'R' and (days(CURRENT TIMESTAMP) - days(lastsession)) >= ? and (users_id not in (select member_id from orders where status != 'Q')) and (users_id > 0) and (users_id not in (select member_id from address where address_id in (select address_id from orderitems where address_id is not null and status != 'Q') or address_id in (select allocaddress_id from orderitems where allocaddress_id is not null and status != 'Q') or address_id in (select address_id from orders where address_id is not null and status !='Q'))))
usrtraffic	obsolete	delete from usrtraffic where (days(CURRENT TIMESTAMP) - days(stmp)) >= ?
vendor	obsolete	delete from vendor where markfordelete = 1 and vendor_id not in (select distinct vendor_id from ra) and vendor_id not in (select distinct vendor_id from receipt where vendor_id is not null)
productset	obsolete	delete from productset where markfordelete = 1 and productset_id not in (select productset_id from tradeposcn where productset_id is not null)

オブジェクト	タイプ	ステートメント
productset	obsolete	delete from productset where productset_id in (select productset_id from tradeposcn where productset_id is not NULL and markfordelete = 1 and type = 'C')
tradeposcn	obsolete	delete from tradeposcn where markfordelete = 1 and type = 'S'

Database Cleanup のコマンド

Database Cleanup ユーティリティ・コマンド (AIX、iSeries、Linux、Solaris、および Windows 2000)

Database Cleanup ユーティリティはデータベースからオブジェクトを除去します。Database Cleanup ユーティリティを実行するには、コマンド行から以下のコマンドを入力してください。すべてのコマンドを 1 行に入力してください。



400 iSeries は dbclean.sh を使用します。iSeries でシェル・スクリプトを実行するには、以下のようにします。

- 65535 以外の CCSID を持つユーザー・プロファイルでログオンします。
- OS/400 行で STRQSH を入力することにより、QSHELL コマンド・ウィンドウをオープンします。
- 以下のようにしてコマンドを実行します:
/QIBM/ProdData/CommerceServer55/bin/dbclean.sh (parameters . . .)

+ 注: パラメーター -dbuser は必ず指定しなければなりません。

Oracle 注:

- 現在このユーティリティを同じユーザー ID を使用して実行している場合でも、オプション・パラメーター、ログオン・ユーザー ID、およびパスワードをコマンドに組み込まなければなりません。
- Linux** **400** Oracle に関する参照事項は iSeries または Linux プラットフォームには適用されません。

パラメーター値:

object 古くなったレコードの削除元となるオブジェクトの名前。以下のオブジェクト名のいずれかを入力してください。

- アカウント・オブジェクトを削除する場合は `account`。
 - アドレス・オブジェクトを削除する場合は `address`。
 - 予定可能在庫オブジェクトを削除する場合は `atp_inventory`。
 - 付加項目オブジェクトを削除する場合は `attachment`。
 - オークション・オブジェクトを削除する場合は `auction`。
 - オークション・ログ・オブジェクトを削除する場合は `auctionlog`。
 - 自動入札オブジェクトを削除する場合は `autobidlog`。
 - 商品情報オブジェクトを削除する場合は `baseitem`。
 - 入札ログ・オブジェクトを削除する場合は `bidlog`。
 - 計算コード・オブジェクトを削除する場合は `calculation_code`。
 - カタログ・エントリー・オブジェクトを削除する場合は `catentry`。
 - カタログ・グループ・オブジェクトを削除する場合は `catalog_group`。
 - 契約オブジェクトを削除する場合は `contract`。
 - クーポン・オブジェクトを削除する場合は `coupon_promotion`。
 - キャンペーン・オブジェクトを削除する場合は `cpnlog`。
 - キャンペーン統計オブジェクトを削除する場合は `cpnstats`。
 - 予定在庫オブジェクトを削除する場合は `expected_inventory_records`。
 - 予定在庫の詳細情報オブジェクトを削除する場合は `expected_inventory_records_details`。
 - サイト管理者と顧客との間のメッセージ・オブジェクトを削除する場合は `forummsg`。
 - 配送センター・オブジェクトを削除する場合は `fulfillment_center`。
 - 在庫オブジェクトを削除する場合は `inventory_adjustments`。
 - 在庫コード・オブジェクトを削除する場合は `inventory_adjustment_codes`。
 - 指定されたアイテム・オブジェクトを削除する場合は `itemspecification`。
 - `dbclean` で使用するためにカスタマイザー・パラメーターの位置を指定する場合は `jdbccustomizer`。 `dbclean` を構成してツールボックス JDBC ドライバーを `iSeries` で使用するには、`dbtype` として `DB2/iSeries` を指定する必要があります。また、カスタマイザー・ファイルには以下が含まれていなければなりません。

```
jdbcDriver=com.ibm.as400.access.AS400JDBCdriver
jdbcUrlPrefix=jdbc:as400:// iSeries での DB2 のデフォルトは次の通りです: jdbcDriver=com.ibm.db2.jdbc.app.DB2Driver
jdbcUrlPrefix=jdbc:db2://。他のプラットフォームでの DB2 のデフォルトは次の通りです: jdbcDriver=COM.ibm.db2.jdbc.app.DB2Driver
jdbcUrlPrefix=jdbc:db2。
```
- 注:** 他の組み合わせでも動作するかもしれませんが、テストは実施されていません。
- オークション関連のメッセージ・オブジェクトを削除する場合は `message`。

- メッセージ・メンバー関係オブジェクトを削除する場合は `msgmemrel`。
- オーダー・オブジェクトを削除する場合は `orders`。
- 組織オブジェクトを削除する場合は `organization`。
- 商品アドバイザー統計オブジェクトを削除する場合は `pastats`。
- 商品比較統計オブジェクトを削除する場合は `pcstats`。
- 商品探査統計オブジェクトを削除する場合は `pestats`。
- ポリシー・オブジェクトを削除する場合は `policy`。
- 商品セット・オブジェクトを削除する場合は `product_sets`。
- 見積依頼オブジェクトを削除する場合は `rfq`。
- 返品オブジェクトを削除する場合は `rma`。
- 返品理由オブジェクトを削除する場合は `rtnreasons`。
- セールス・アシスタント統計オブジェクトを削除する場合は `sastats`。
- ステージ・オブジェクトを削除する場合は `staglog`。
- ストア・オブジェクトを削除する場合は `store`。
- ユーザー・オブジェクトを削除する場合は `user`。
- ユーザー通信ログ・オブジェクトを削除する場合は `usrtraffic`。
- 取引先オブジェクトを削除する場合は `vendor`。

type 削除したいオブジェクトのタイプ。データベースのクリーンアップの下にある各コマンドを参照してください。

days 削除されるレコードが存在する最小日数。

name (オプション) 削除されるオブジェクトの ID。 `member` が `organization` パラメーターの値として指定され、 `organization` が `type` の値として指定されている場合に、このパラメーターが必要となります。

dbtype

+ (オプション) データベースのタイプは DB2、DB2/iSeries、または Oracle
 + です。デフォルトは DB2 です。 `400` デフォルトは DB2/iSeries で
 + ず。

db データベースの名前。 `400` リレーショナル・データベース・ディレクトリーで検索されるとおりのデータベースの名前。

注: ツールボックス JDBC ドライバーを使用するために `jdbccustomizer` パラメーターを構成した場合、データベース名の代わりにデータベースが常駐するマシンのホスト名を指定しなければなりません (例えば、`-db hostname.ibm.com`)。

`Oracle`

`host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

dbuser

(オプション) スキーマを作成した管理者またはデータベースのサイト管理者のログオン ID。このパラメーターを指定しない場合、ユーティリティを

呼び出しているユーザーの ID が使用されます。▶ 400 (必須)
WebSphere Commerce インスタンスと関連したユーザー・プロファイル。これはスキーマ名でもあります。

dbpasswd

(オプション) DB2 データベース用に `dbuser` パラメーターで指定されたログオン ID のパスワード。指定されない場合、システムはパスワードを入力するようにプロンプトを出します。

check_object_only

(オプション) Database Cleanup ユーティリティーは `yes` を指定した場合、データベースのクリーンアップの影響を受けた可能性がある子テーブルおよび削除ルールをすべてリストします。パラメーターを `no` (デフォルト) のままにしておくと、ユーティリティーはチェックを実行しません。

force (オプション) `force` オプションは `yes` または `no` に設定されます。 `yes` を設定した場合、ユーティリティーは子テーブルを削除し、続いて親テーブルを削除します。

loglevel

(オプション) データベースのクリーンアップ中に実行されるロギングのレベル。値が指定されていない場合、デフォルトのロギング・レベルは 0 です。

- ログ・アクティビティーが記録されないように指定するには、0 を入力してください。これはデフォルトです。
- 指定されたテーブルに対してのみ `DELETE` ステートメントが記録されるように指定するには、1 を入力してください。
- 指定されたテーブルおよび子テーブルに対して `DELETE` ステートメントが記録されるように指定するには、2 を入力してください。Loglevel 2 は Database Cleanup ユーティリティーがボトムアップ法を使用するよう強制します。

+ **log** (オプション) ユーティリティーがそのアクティビティーを記録するログ・ファイルのパスと名前。このコマンドの発行者は指定されたパスに対する書き込み権限を持っている必要があります。また、パスはすでに存在していなければなりません。このパラメーターを指定しない場合、`dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` という名前のログ・ファイルが以下のディレクトリーに作成されます。

+ ▶ AIX ▶ Solaris ▶ Linux `WC_installdir/logs`

+ ▶ 2000 `WC_installdir¥logs`


+ ▶ 400 このパラメーターを指定しない場合、`dbclean_dbuser_yyyy.mm.dd_hh.mm.ss.zzz.log` という名前のログ・ファイルが `WC_userdir/instances` ディレクトリーに作成されます。


instancexml

ストアを削除する場合に使用されます。 `instancexml` は、`WC_installdir/instances/instancexml/xml/instancexml.xml` という形式の絶対ファイル名の値を持ちます。

注: ユーザー、組織、ストア、付加項目などファイル資産を持つオブジェクトにはこのパラメーターが必要です。

+ **script** (オプション) 保管される出力スクリプトのパスと名前を指定します。デフォルトでは、以下のディレクトリーに保管されます。

+  `WC_installdir\logs\assetclean_userName_timestamp_.cmd`

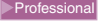

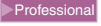

+  `/WC_userdir/instances/assetclean_userName_timestamp_.sh`





+ その他のプラットフォームの場合は、

+ `WC_installdir/logs/assetclean_userName_timestamp_.sh` です。

データベースのクリーンアップ


Database Cleanup ユーティリティを使用して WebSphere Commerce データベースから未使用のまたは古くなったオブジェクトを削除するには、以下のいずれかを選択してください。















- アカウント・オブジェクト
- アドレス・オブジェクト
- 予定可能 (ATP) 在庫オブジェクト
- 付加項目オブジェクト
-  オークション・オブジェクト
-  オークション・ログ・オブジェクト
-  自動入札ログ・オブジェクト
- 基本アイテム・オブジェクト
-  入札ログ・オブジェクト
- 計算コード・オブジェクト
- キャンペーン・オブジェクト
- キャンペーン統計オブジェクト
- カタログ・エントリー・オブジェクト
- カタログ・グループ・オブジェクト
- 契約オブジェクト
- クーポン・オブジェクト
- 予定在庫レコード・オブジェクト
- 予定在庫レコード詳細情報オブジェクト
- フォーラム・メッセージ・オブジェクト
- 配送センター・オブジェクト
- 在庫調整コード・オブジェクト
- 在庫調整オブジェクト
- アイテム指定オブジェクト
- メンバー・メッセージ・オブジェクト
- メッセージ・オブジェクト
- オーダー・オブジェクト

- 組織オブジェクト
-  Professional 商品アドバイザー統計オブジェクト
-  Professional 商品比較統計オブジェクト
-  Professional 商品探査統計オブジェクト
- ポリシー・オブジェクト
- 商品セット・オブジェクト
- 見積依頼 (RFQ) オブジェクト
- 返品オブジェクト
- 返品理由オブジェクト
-  Professional セールス・アシスタント統計オブジェクト
- ステージ・オブジェクト
- ストア・オブジェクト
- ユーザー・オブジェクト
- ユーザー通信ログ・オブジェクト
- 取引先オブジェクト

注: Database Cleanup ユーティリティーを使用する前に、データベースを構成して PATH 環境変数を設定する必要があります。

WebSphere Commerce ユーティリティー用に PATH 環境変数を設定する

注:  400 このセクションは iSeries プラットフォームには適用されません。Database Cleanup ユーティリティーまたはステージング・サーバーを使用する前に、PATH 環境変数で以下のように指定されているか確認してください。

-  2000 
WC_installdir/bin および WC_installdir%sqllib%bin
-  2000 
WC_installdir%bin および WC_installdir%ora9i%bin
-  2000 
WC_installdir%bin および WC_installdir%sqllib%bin
-  2000 
WC_installdir%bin および WC_installdir%ora9i%bin
-  AIX 
WC_installdir/bin および WC_installdir/lpp/db2_08_01
-  AIX 
WC_installdir/bin および /usr/ora9i/bin
-  Solaris 
WC_installdir/bin および WC_installdir/IBMdb2/V8.1

- ▶ Linux ▶ DB2
`WC_installdir/bin` および `opt/IBM/db2/V8.1`
- ▶ Solaris ▶ Oracle
`WC_installdir/bin` および `WC_installdir/ora9i/bin`

オブジェクトを削除する

このセクションでは、Database Cleanup ユーティリティーを使用した、WebSphere Commerce データベースからの未使用または古くなった特定のオブジェクトの削除について説明します。

アカウント・オブジェクトを削除する

アカウント・オブジェクトを削除するには、次のようにします。

- 次のように入力します。

- ▶ Windows ▶ DB2
`dbclean -object account -type obsolete -db dbname -days daysold -loglevel loglevel`

- ▶ Windows ▶ Oracle
`dbclean -object account -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

+
+
+

- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2
`. dbclean.sh -object account -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`

- ▶ AIX ▶ Solaris ▶ Oracle
`. dbclean.sh -object account -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。たとえば、`myhost:1521:mydb` のようにします。

- `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

アカウント・オブジェクトに関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。











アドレス・オブジェクトを削除する

アドレス・オブジェクトを削除するには、次のようにします。

- 次のように入力します。

- ▶ Windows ▶ DB2
`dbclean -object address -type obsolete -db dbname -days daysold -loglevel loglevel`

+
+
+

-  
`dbclean -object address -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object address -type obsolete -db dbname -dbuser user
-days daysold -loglevel loglevel`
-   
`. dbclean.sh -object address -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。













アドレス・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

予定可能在庫オブジェクトを削除する

予定可能 (ATP) 在庫オブジェクトを削除するには、以下のようにします。

1. 次のように入力します。

+
+
+

-  
`dbclean -object atp_inventory -type typename -db dbname -loglevel
loglevel`
-  
`dbclean -object atp_inventory -type typename -db dbname -logLevel
loglevel -dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object atp_inventory -type typename -db dbname -dbuser
user -loglevel loglevel`
-   
`. dbclean.sh -object atp_inventory -type typename -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ATP 在庫オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

付加項目オブジェクトを削除する

付加項目オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object attachment -type obsolete -db dbname -days daysold  
-loglevel loglevel
```

- Windows Oracle

```
dbclean -object attachment -type obsolete -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object attachment -type obsolete -db dbname -dbuser user  
-days daysold -loglevel loglevel
```

- AIX Solaris Oracle

```
. dbclean.sh -object attachment -type obsolete -db dbname -days  
daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd  
password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

注: 付加項目の *-type* は *markfordelete* とすることもできます。この場
合、*-instancexml* が必要となります。以下に例を示します。

```
dbclean.sh -object attachment -type markfordelete -instancexml  
/WC_userdir/instances/instance/xml/instance.xml  
-db dbname -dbuser instance -loglevel
```

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実
行されたかどうか検査します。

付加項目オブジェクトの削除に関するその他の例は、145 ページの『オブジェクト
の削除の例』を参照してください。

オークション・オブジェクトを削除する

オークション・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object auction -type typename -db dbname -days daysold  
-loglevel loglevel
```

- Windows Oracle

```
dbclean -object auction -type typename -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object auction -type typename -db dbname -dbuser user  
-days daysold -loglevel loglevel
```

- ▶ AIX ▶ Solaris ▶ Oracle

```
. dbclean.sh -object auction -type typename -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。注: *-type* パラメーターに対して、完了したオークション・レコードを示すには *settlement_closed*、撤回したオークションを示すには *retracted* を指定します。

2. *dbclean.log_yyyy.mm.dd_hh.mm.ss.zzz* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

オークション・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

オークション・ログ・オブジェクトを削除する

オークション・ログ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- ▶ Windows ▶ DB2

```
dbclean -object auctionlog -type obsolete -db dbname -days daysold
-loglevel loglevel
```

- ▶ Windows ▶ Oracle

```
dbclean -object auctionlog -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2

```
. dbclean.sh -object auctionlog -type obsolete -db dbname -dbuser user
days daysold -loglevel loglevel
```

- ▶ AIX ▶ Solaris ▶ Linux

```
. dbclean.sh -object auctionlog -type obsolete -db dbname -dbuser user
-days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd
password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。













2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

オークション・ログ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

自動入札ログ・オブジェクトを削除する

自動入札ログ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  
`dbclean -object autobidlog -type obsolete -db dbname -days daysold
 -loglevel loglevel`
-  
`dbclean -object autobidlog -type obsolete -db dbname -days daysold
 -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
`. dbclean.sh -object autobidlog -type obsolete -db dbname -dbuser user
 -days daysold -loglevel loglevel`
- +   
`. dbclean.sh -object autobidlog -type obsolete -db dbname -days
daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd
password`

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
 たとえば、`myhost:1521:mydb` のようにします。













2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

自動入札ログ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

基本アイテム・オブジェクトを削除する

同じ名前や説明を持つ一群の商品全般に関する情報を含む基本アイテム・オブジェクトを削除するには、以下のようにします。

1. 次のように入力します。

-  
`dbclean -object baseitem -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object baseitem -type obsolete -db dbname -loglevel loglevel
 -dbtype oracle -dbuser user -dbpasswd password`
- +     
`. dbclean.sh -object baseitem -type obsolete -db dbname -dbuser user
 -loglevel loglevel`
- +   
`. dbclean.sh -object baseitem -type obsolete -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
 たとえば、`myhost:1521:mydb` のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

基本アイテム・オブジェクトの削除に関するその他の例は、145ページの『オブジェクトの削除の例』を参照してください。

入札ログ・オブジェクトを削除する

入札ログ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object bidlog -type obsolete -db dbname -days daysold  
-loglevel loglevel
```

- Windows Oracle

```
dbclean -object bidlog -type obsolete -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object bidlog -type obsolete -db dbname -dbuser user  
-days daysold -loglevel loglevel
```

- AIX Solaris Oracle

```
. dbclean.sh -object bidlog -type obsolete -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

入札ログ・オブジェクトの削除に関するその他の例は、145ページの『オブジェクトの削除の例』を参照してください。

計算コード・オブジェクトを削除する

計算コード・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object calculation_code -type obsolete -db dbname -loglevel  
loglevel
```

- Windows Oracle

```
dbclean -object calculation_code -type obsolete -db dbname -loglevel  
loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object calculation_code -type obsolete -db dbname  
-dbuser user -loglevel loglevel
```

- AIX Solaris Oracle

```
. dbclean.sh -object calculation_code -type obsolete -db dbname  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。



計算コード・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。





キャンペーン・オブジェクトを削除する




キャンペーン・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  
`dbclean -object cpnlog -type all -db dbname -loglevel loglevel`

-  
`dbclean -object cpnlog -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`

-     
+
+
+
`. dbclean.sh -object cpnlog -type all -db dbname -dbuser user
-loglevel loglevel`

-   
`. dbclean.sh -object cpnlog -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。



2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。



キャンペーン・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。






キャンペーン統計オブジェクトを削除する

キャンペーン統計オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  
`dbclean -table cpnstats -type all -db dbname -loglevel loglevel`

-  
`dbclean -table cpnstats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`

-     
+
+
+
`. dbclean.sh -object cpnstats -type all -db dbname -dbuser user
-loglevel loglevel`

- AIX
 - Solaris
 - Oracle

```
. dbclean.sh -object cpgnstats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

CPGNSTATS テーブルからのキャンペーン統計レコードの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

カタログ・エン트리・オブジェクトを削除する

カタログ・エン트리・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows
 - DB2

```
dbclean -object catentry -type typename -db dbname -days daysold
-loglevel loglevel
```

- Windows
 - Oracle

```
dbclean -object catentry -type typename -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

- AIX
 - 400
 - Solaris
 - Linux
 - DB2

```
. dbclean.sh -object catentry -type typename -db dbname -dbuser user
-days daysold -loglevel loglevel
```

- AIX
 - Solaris
 - Oracle

```
. dbclean.sh -object catentry -type typename -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

注: `-type` パラメーターに対して、オーダー・アイテムと関連していないカタログ・エントリを示すには `without_orderitems` を、オーダー・アイテムまたは買い物候補リスト・アイテムと関連していないカタログ・エントリを示すには `without_orderitems_iitems` を指定できます。













2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

カタログ・エン트리・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

カタログ・グループ・オブジェクトを削除する

カタログ・グループ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  
`dbclean -object catalog_group -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object catalog_group -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
`. dbclean.sh -object catalog_group -type obsolete -db dbname -dbuser user -loglevel loglevel`
- +   
`. dbclean.sh -object catalog_group -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。













2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

カタログ・グループ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

契約オブジェクトを削除する

契約オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  
`dbclean -object contract -type obsolete -db dbname -loglevel loglevel`
-  
`dbclean -object contract -type obsolete -db database -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- +     
`. dbclean.sh -object contract -type obsolete -db dbname -dbuser user -loglevel loglevel`
- +   
`. dbclean.sh -object contract -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

注:      `-dbor DATABASE` パラメーターの場合は、`productset`、`tradeposn`、または `trading` の 3 つのデータベース・テーブルの中から 1 つを指定する必要があります。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

契約オブジェクトの削除に関するその他の例は、145ページの『オブジェクトの削除の例』を参照してください。

クーポン・オブジェクトを削除する

クーポン・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object coupon_promotion -type expired -db dbname -days  
daysold -loglevel loglevel
```

- Windows Oracle

```
dbclean -object coupon_promotion -type expired -db dbname -days  
daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd  
password
```

+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object coupon_promotion -type expired -db dbname -dbuser  
user -days daysold -loglevel loglevel
```

- AIX Solaris Oracle

```
. dbclean.sh -object coupon_promotion -type expired -db dbname -days  
daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd  
password
```

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

クーポン・オブジェクトの削除に関するその他の例は、145ページの『オブジェクトの削除の例』を参照してください。

予定在庫レコード・オブジェクトを削除する

予定在庫レコード・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object expected_inventory_records -type obsolete -db dbname  
-days daysold -loglevel loglevel
```

- Windows Oracle

```
dbclean -object expected_inventory_records -type obsolete -db dbname  
-days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd  
password
```

+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object expected_inventory_records -type obsolete -db  
dbname -dbuser user -days daysold -loglevel loglevel
```

- ▶ AIX
▶ Solaris
▶ Oracle

```
. dbclean.sh -object expected_inventory_records -type obsolete -db
dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user
-dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

予定在庫レコード・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

予定在庫レコード・オブジェクトに関する詳細情報を削除する

予定在庫レコード・オブジェクトに関する詳細情報を削除するには、次のようにします。

1. 次のように入力します。

- ▶ Windows
▶ DB2

```
dbclean -object expected_inventory_record_details -type obsolete -db
dbname -loglevel loglevel
```

- ▶ Windows
▶ Oracle

```
dbclean -object expected_inventory_record_details -type obsolete -db
dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd
password
```

- ▶ AIX
▶ 400
▶ Solaris
▶ Linux
▶ DB2

```
. dbclean.sh -object expected_inventory_record_details -type obsolete
-db dbname -dbuser user -loglevel loglevel
```

- ▶ AIX
▶ Solaris
▶ Oracle

```
. dbclean.sh -object expected_inventory_record_details -type obsolete
-db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd
password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

予定在庫レコード・オブジェクトに関する詳細情報の削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

フォーラム・メッセージ・オブジェクトを削除する

フォーラム・メッセージ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2
`dbclean -object forummsg -type obsolete -db dbname -days daysold -loglevel loglevel`
- Windows Oracle
`dbclean -object forummsg -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- AIX 400 Solaris Linux DB2
`. dbclean.sh -object forummsg -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel`
- AIX Solaris Oracle
`. dbclean.sh -object forummsg -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

フォーラム・メッセージ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

配送センター・オブジェクトを削除する

配送センター・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2
`dbclean -object fulfillment_center -type obsolete -db dbname -loglevel loglevel`
- Windows Oracle
`dbclean -object fulfillment_center -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`
- AIX 400 Solaris Linux DB2
`. dbclean.sh -object fulfillment_center -type obsolete -db dbname -dbuser user -loglevel loglevel`
- AIX Solaris Oracle
`. dbclean.sh -object fulfillment_center -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

配送センター・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

在庫コード調整オブジェクトを削除する

在庫コード調整オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2
dbclean -object inventory_adjustment_codes -type obsolete -db *dbname* -loglevel *loglevel*

- Windows Oracle
dbclean -object inventory_adjustment_codes -type obsolete -db *dbname* -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

- AIX 400 Solaris Linux DB2
+
+
+
. dbclean.sh -object inventory_adjustment_codes -type obsolete -db *dbname* -dbuser *user* -loglevel *loglevel*

- AIX Solaris Oracle
. dbclean.sh -object inventory_adjustment_codes -type obsolete -db *dbname* -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

在庫コード調整オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

在庫調整オブジェクトを削除する

在庫調整オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2
dbclean -object inventory_adjustments -type obsolete -db *dbname* -days *daysold* -loglevel *loglevel*

- Windows Oracle
dbclean -object inventory_adjustments -type obsolete -db *dbname* -days *daysold* -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

- AIX 400 Solaris Linux DB2
+
+
+
. dbclean.sh -object inventory_adjustments -type obsolete -db *dbname* -dbuser *user* -days *daysold* -loglevel *loglevel*

- AIX Solaris Oracle
. dbclean.sh -object inventory_adjustments -type obsolete -db *dbname* -days *daysold* -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

在庫調整オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

指定アイテム情報オブジェクトを削除する

指定アイテム情報オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  

```
dbclean -object itemspecification -type obsolete -db dbname -loglevel loglevel
```

-  

```
dbclean -object itemspecification -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object itemspecification -type obsolete -db dbname -dbuser user -loglevel loglevel
```

-   

```
. dbclean.sh -object itemspecification -type obsolete -db dbname -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

指定アイテム情報オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

メンバー・メッセージ関係オブジェクトを削除する

メンバー・メッセージ関係オブジェクトを削除するには、次のようにします。






1. 次のように入力します。




-  

```
dbclean -object msgmemrel -type obsolete -db dbname -days daysold -loglevel loglevel
```

-  

```
dbclean -object msgmemrel -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

- +
 -     

```
. dbclean.sh -object msgmemrel -type obsolete -db dbname -dbuser user
-days daysold -loglevel loglevel
```
 -   

```
. dbclean.sh -object msgmemrel -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。



2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

メンバー・メッセージ関係オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。



メッセージ・オブジェクトを削除する

メッセージ・オブジェクトを削除するには、次のようにします。






1. 次のように入力します。

-  




```
dbclean -object message -type obsolete -db dbname -days daysold
-loglevel loglevel
```

-  

```
dbclean -object message -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

- +
 -     

```
. dbclean.sh -object message -type obsolete -db dbname -dbuser user
-days daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object message -type obsolete -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。



2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。



メッセージ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。






オーダー・オブジェクトを削除する




オーダー・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  

```
dbclean -object order -type typename -db dbname -days daysold
-loglevel loglevel
```
-  

```
dbclean -object order -type typename -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```
- +     

```
. dbclean.sh -object order -type typename -db dbname -dbuser user
-days daysold -loglevel loglevel
```
- +   

```
. dbclean.sh -object order -type typename -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

注: `-type` パラメーターに対して、次の 6 つの中からいずれかを指定できます。完了したオーダーを示すには `completed`、キャンセルされたオーダーを示すには `canceled`、配送済みのオーダーを示すには `shipped`、入金済みのオーダーを示すには `deposited`、ゲスト顧客からの期限切れのオーダーを示すには `stale_guest`、ゲストではない顧客からの期限切れのオーダーを示すには `stale_non_guest` を指定します。



2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。



オーダー・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。






組織オブジェクトを削除する

組織オブジェクトを削除するには、次のようにします。




1. 次のように入力します。

-  

```
dbclean -object organization -type specified -db dbname -loglevel
loglevel -name organizationid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
-  

```
dbclean -object organization -type specified -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -name
organizationid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
- +     

```
. dbclean.sh -object organization -type specified -db dbname -dbuser
user -loglevel loglevel -name organizationid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```


- 



```

dbclean.sh -object organization -type specified -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -name
organizationid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml

```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。注: *-type* パラメーターに対して、組織レコードを示す *organization* を指定できます。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

組織オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

商品アドバイザー統計オブジェクトを削除する

商品アドバイザー統計オブジェクトを削除するには、次のようにします。



1. 次のように入力します。

- 


```

dbclean -object pastats -type all -db dbname -loglevel loglevel

```






- 


```

dbclean -object pastats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password

```




+
+
+

- 





```

dbclean.sh -object pastats -type all -db dbname -dbuser user
-loglevel loglevel

```

- 



```

dbclean.sh -object pastats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password

```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

商品アドバイザー統計オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

商品比較統計オブジェクトを削除する

商品比較（商品アドバイザー）統計オブジェクトを削除するには、次のようにします。

1. 次のように入力します。











- 


```

dbclean -object pcstats -type all -db dbname -loglevel loglevel

```

+
+
+

-  
`dbclean -object pcstats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object pcstats -type all -db dbname -dbuser user
-loglevel loglevel`
-   
`. dbclean.sh -object pcstats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。













商品比較 (商品アドバイザー) 統計オブジェクトの削除に関するその他の例は、
145 ページの『オブジェクトの削除の例』を参照してください。

商品探査統計オブジェクトを削除する

商品探査 (商品アドバイザー) 統計オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

+
+
+

-  
`dbclean -object pestats -type all -db dbname -loglevel loglevel`
-  
`dbclean -object pestats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`
-     
`. dbclean.sh -object pestats -type all -db dbname -dbuser user
-loglevel loglevel`
-   
`. dbclean.sh -object pestats -type all -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

商品探査 (商品アドバイザー) 統計オブジェクトの削除に関するその他の例は、
145 ページの『オブジェクトの削除の例』を参照してください。

ポリシー・オブジェクトを削除する

ポリシー・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2
dbclean -object policy -type obsolete -db *dbname* -days *daysold* -loglevel *loglevel*

- Windows Oracle
dbclean -object policy -type obsolete -db *dbname* -days *daysold* -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

- AIX 400 Solaris Linux DB2
+ . dbclean.sh -object policy -type obsolete -db *dbname* -dbuser *user*
+ -days *daysold* -loglevel *loglevel*
+

- AIX Solaris Oracle
+ . dbclean.sh -object policy -type obsolete -db *dbname* -days *daysold*
+ -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ポリシー・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

商品セット・オブジェクトを削除する

商品セット・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2
dbclean -object product_sets -type obsolete -db *dbname* -loglevel *loglevel*

- Windows Oracle
dbclean -object product_sets -type obsolete -db *dbname* -loglevel *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

- AIX 400 Solaris Linux DB2
+ . dbclean.sh -object product_sets -type obsolete -db *dbname* -dbuser
+ *user* -loglevel *loglevel*
+

- AIX Solaris Oracle
+ . dbclean.sh -object product_sets -type obsolete -db *dbname* -loglevel
+ *loglevel* -dbtype oracle -dbuser *user* -dbpasswd *password*

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

商品セット・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

見積依頼オブジェクトを削除する

見積依頼 (RFQ) オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  

```
dbclean -table rfq -type obsolete -db dbname -loglevel loglevel
```



-  

```
dbclean -object rfq -type obsolete -db dbname -loglevel loglevel  
-dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object rfq -type obsolete -db dbname -dbuser user  
-loglevel loglevel
```

-   

```
. dbclean.sh -object rfq -type obsolete -db dbname -loglevel loglevel  
-dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

RFQ オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

返品オブジェクトを削除する

返品オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  

```
dbclean -object rma -type typename -db dbname -days daysold -loglevel  
loglevel
```

-  

```
dbclean -object rma -type typename -db dbname -days daysold -loglevel  
loglevel -dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

-     

```
. dbclean.sh -object rma -type typename -db dbname -dbuser user -days  
daysold -loglevel loglevel
```

- ▶ AIX
▶ Solaris
▶ Oracle

```
. dbclean.sh -object rma -type typename -db dbname -days daysold
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

注: `-type` パラメーターに対して、中止されたレコードを示すには `abandoned`、キャンセルされたレコードを示すには `canceled`、拒否されたレコードを示すには `not_approved`、承認されたまたは一部承認されたレコードを示すには `approved_or_partly_approved` 完了したレコードを示すには `completed` を指定できます。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

返品オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

返品理由オブジェクトを削除する

商品に対して顧客が不満を感じる理由、または単なる返品理由オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- ▶ Windows
▶ DB2

```
dbclean -object rtnreason -type obsolete -db dbname -loglevel loglevel
```

- ▶ Windows
▶ Oracle

```
dbclean -object rtnreason -type obsolete -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password
```

- ▶ AIX
▶ 400
▶ Solaris
▶ Linux
▶ DB2

```
. dbclean.sh -object rtnreason -type obsolete -db dbname -dbuser user
-loglevel loglevel
```

- ▶ AIX
▶ Solaris
▶ Oracle

```
. dbclean.sh -object rtnreason -type obsolete -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

返品理由オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

セールス・アシスタント統計オブジェクトを削除する

セールス・アシスタント（商品アドバイザー）統計オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

•  

```
dbclean -object sastats -type all -db dbname -loglevel loglevel
```




•  

```
dbclean -object sastats -type all -db dbname -loglevel loglevel  
-dbtype oracle -dbuser user -dbpasswd password
```

+
+
+

•     

```
. dbclean.sh -object sastats -type all -db dbname -dbuser user  
-loglevel loglevel
```

•   

```
. dbclean.sh -object sastats -type all -db dbname -loglevel loglevel  
-dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

セールス・アシスタント (商品アドバイザー) 統計オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

ステージ・オブジェクトを削除する

ステージ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。


•  

```
dbclean -object staglog -type obsolete -db dbname -days daysold  
-loglevel loglevel
```



•  

```
dbclean -object staglog -type obsolete -db dbname -dbuser user -days  
daysold -loglevel loglevel -dbtype oracle -dbuser <user> -dbpasswd  
<password>
```

+
+
+

•     

```
. dbclean.sh -object staglog -type obsolete -db dbname -dbuser user  
-days daysold -loglevel loglevel
```

•   

```
. dbclean.sh -object staglog -type obsolete -db dbname -days daysold  
-loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ステージ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

ストア・オブジェクトを削除する

ストア・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

- Windows DB2

```
dbclean -object store -type specified -db dbname -loglevel loglevel
-name storeid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

- Windows Oracle

```
dbclean -object store -type specified -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password -name storeid
-instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

+
+
+
+

- AIX 400 Solaris Linux DB2

```
. dbclean.sh -object store -type specified -db dbname -dbuser user
-loglevel loglevel -name storeid -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

- AIX Solaris Oracle

```
. dbclean.sh -object store -type specified -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -name storeid
-instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。
たとえば、*myhost:1521:mydb* のようにします。

2. *dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ストア・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

ユーザー・オブジェクトを削除する

ユーザー・オブジェクトを削除するには、次のようにします。






1. 次のように入力します。




- Windows DB2

```
dbclean -object user -type typename -db dbname -days daysold -loglevel
loglevel -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

- Windows Oracle

```
dbclean -object user -type typename -db dbname -days daysold -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password -instancexml
WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```


- +
 -     

```
. dbclean.sh -object user -type typename -db dbname -dbuser user -days daysold -loglevel loglevel -instancexml WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```
 -   

```
. dbclean.sh -object user -type typename -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password -instancexml WC_installdir/instances/INSTANCE_NAME/xml/INSTANCE_NAME.xml
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

注: `-type` パラメーターに対して、ゲスト顧客を示すには `guest`、登録済み顧客を示すには `registered` を指定できます。



2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。



ユーザー・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。






ユーザー通信ログ・オブジェクトを削除する




ユーザー通信ログ・オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

-  

```
dbclean -object usrtraffic -type obsolete -db dbname -days daysold -loglevel loglevel
```
-  

```
dbclean -object usrtraffic -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```
- +
 -     

```
. dbclean.sh -object usrtraffic -type obsolete -db dbname -dbuser user -days daysold -loglevel loglevel
```
 -   

```
. dbclean.sh -object usrtraffic -type obsolete -db dbname -days daysold -loglevel loglevel -dbtype oracle -dbuser user -dbpasswd password
```

Oracle データベースの名前には、*host:port:sid* の形式を使用してください。たとえば、*myhost:1521:mydb* のようにします。

2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。



ユーザー通信ログ・オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。






取引先オブジェクトを削除する




取引先オブジェクトを削除するには、次のようにします。

1. 次のように入力します。

•  
`dbclean -object vendor -type obsolete -db dbname -loglevel loglevel`

•  
`dbclean -object vendor -type obsolete -db dbname -loglevel loglevel
-dbtype oracle -dbuser user -dbpasswd password`

+
+
+
•     
`. dbclean.sh -object vendor -type obsolete -db dbname -dbuser user
-loglevel loglevel`

•   
`. dbclean.sh -object vendor -type obsolete -db dbname -loglevel
loglevel -dbtype oracle -dbuser user -dbpasswd password`

Oracle データベースの名前には、`host:port:sid` の形式を使用してください。
たとえば、`myhost:1521:mydb` のようにします。


2. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

取引先オブジェクトの削除に関するその他の例は、145 ページの『オブジェクトの削除の例』を参照してください。

Database Cleanup ユーティリティーに新規構成を追加する

Database Cleanup ユーティリティーに新規構成を追加するには、参照として次の構文を使用してください。例えば、オブジェクト `o1` は次の列を含むテーブル `R1` から構成されているとします: `col1`、`col2`、`lastupdate`、および `col3`。Database Cleanup ユーティリティーを使用して、`col1 > 10` かつ `lastupdate` が `n` 日前であるすべてのオブジェクトを削除するには、以下のようにします。



+
+
1.     DB2 コマンド・プロンプトを開きます。

2. 次のように入力します。

```
db2 insert into cleanconf (objectname, type, statement, namearg,  
sequence, daysarg) values ('o1', 'obsolete', 'delete from r1 where col1 >  
10 and (days(CURRENT TIMESTAMP) - days(lastupdate)) > ?', 'no', 1, 'yes')
```

1.  次の SQL ステートメントを実行します。

```
insert into cleanconf (objectname, type, statement, namearg, sequence,  
daysarg) values ('o1', 'obsolete', 'delete from r1 where col1 > 10 and  
(days(CURRENT TIMESTAMP) - days(lastupdate)) > ?', 'no', 1, 'yes')
```





1. SQLPlus コマンド・ウィンドウを開きます。
2. 次のように入力します。




```
insert into cleanconf (objectname, type, statement, namearg, sequence,
daysarg) values ('o1', 'obsolete', 'delete from r1 where col1 > 10 and
(sysdate - lastupdate) > ?', 'no', 1, 'yes')
```






? は、次のコマンド行から `-days` パラメーターに置き換えられます。'no' は name パラメーターがステートメントで使用されていないことを示しています。'yes' は `-days` パラメーターがステートメントで使用されていることを示しています。'obsolete' はオブジェクト o1 のクリーンアップ・タイプを表します。他のワードも使用できますが、Database Cleanup ユーティリティ・コマンドを呼び出す際に、`-type` 引き数で同じワードを使用する必要があります。




例

Database Cleanup ユーティリティ・コマンドを呼び出して、2 日間存在しているレコードを新規テーブルからクリーンアップするには、次のように入力します。

-  

```
dbclean -object o1 -db dbname -type obsolete -days 2 -loglevel 1
```
-  

```
dbclean -object o1 -db dbname -type obsolete -days 2 -loglevel 1 -dbtype
oracle -dbuser user -dbpasswd password
```
- +     

```
. dbclean.sh -object o1 -db dbname -dbuser user -type obsolete -days 2
-loglevel 1
```
- +   



```
. dbclean.sh -object o1 -db dbname -type obsolete -days 2 -loglevel 1
-dbtype oracle -dbuser user -dbpasswd password
```

注: Oracle `dbname` パラメーターに対しては、`host:port:sid` を使用してください。たとえば、`myhost:1521:mydb` のようにします。

オブジェクトの削除の例

Database Cleanup ユーティリティ・コマンドでオプションのパラメーターを使用して、データベース・テーブルからオブジェクトを削除する例を以下に示します。パラメーターについては、『Database Cleanup ユーティリティ・コマンド』を参照してください。

例 1: どのテーブルが削除ルール・パラメーターを指定しているか検査するには、以下のように入力します。

-  

```
dbclean -object objectname -type typename -db database -check_object_only
yes
```




- ▶ Windows ▶ Oracle
`dbclean -object objectname -type typename -db host:port:sid
 -check_object_only yes -dbtype oracle -dbuser user -dbpasswd <password>`
- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2
`. dbclean.sh -object objectname -type typename -db database -dbuser user
 -check_object_only yes`
- ▶ AIX ▶ Solaris ▶ Oracle
`. dbclean.sh -object objectname -type typename -db host:port:sid
 -check_object_only yes -dbtype oracle -dbuser user -dbpasswd password`

例 2: 削除ルール・パラメーターを指定したテーブルで `force` オプションを使用するには、以下のように入力します。

- ▶ Windows ▶ DB2
`dbclean -object objectname -type typename -db database -days daysold
 -loglevel loglevel -force yes`
- ▶ Windows ▶ Oracle
`dbclean -object objectname -type typename -db host:port:sid -days daysold
 -loglevel loglevel -force yes -dbtype oracle -dbuser user -dbpasswd
 <password>`
- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2
`. dbclean.sh -object objectname -type typename -db database -dbuser user
 -days daysold -loglevel loglevel -force yes`
- ▶ AIX ▶ Solaris ▶ Oracle
`. dbclean.sh -object objectname -type typename -db host:port:sid -days
daysold -loglevel loglevel -force yes -dbtype oracle -dbuser user
 -dbpasswd <password>`

例 3: デフォルトのログ・ファイル名は必ず `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` が変化した名前になります。ログ・ファイル名を指定するには、以下のように入力します。

- ▶ Windows ▶ DB2
`dbclean -object objectname -type typename -db database -days daysold
 -loglevel loglevel -log logfile`
- ▶ Windows ▶ Oracle
`dbclean -object objectname -type typename -db host:port:sid -days daysold
 -loglevel loglevel -log logfile -dbtype oracle -dbuser user -dbpasswd
 <password>`
- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2
`. dbclean.sh -object objectname -type typename -db database -dbuser user
 -days daysold -loglevel loglevel -log logfile`

-   
 . dbclean.sh -object *objectname* -type *typename* -db *host:port:sid* -days
 daysold -loglevel *loglevel* -log *logfile* -dbtype oracle -dbuser *user*
 -dbpasswd <password>

第 7 章 パフォーマンス

WebSphere Commerce は、相互に作用し合う、複数のプロダクトから成り立っています。各プロダクトにはそれぞれ自身のパフォーマンス特性があります。さまざまなコンポーネント同士の相互作用の中には、構成が間違っていたり、リソースが不十分だったりすると、パフォーマンスに影響が及ぶ箇所が多数あります。パフォーマンスの目標には、以下のタイプの要求をタイムリーに処理することが含まれます。

- 顧客の複数の要求を処理する
- WebSphere Commerce データベースにあるデータにアクセスする
- データを Web ページとしてフォーマットする
- ショッパーのブラウザーに応答を戻す

WebSphere Commerce を最適化するには、以下のコンポーネントを考慮してください。

- ハードウェア

マシンが「*WebSphere Commerce インストール・ガイド*」で説明されている最小マシン要件を満たしていることを確認します。

多くの同時ユーザーが存在する実稼働環境では、複数 CPU を使用するとパフォーマンスが向上します。一般に、より高速の CPU を使用すると、ほとんどの操作が速くなります。

- データベース

DB2 データベースをチューニングする方法については、レッドブック「*DB2/UDB WebSphere Performance Tuning Guide*」(<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246417.html?Open> で入手可能) を参照してください。

最大データベース接続プールのサイズが、すべての並行タスク (HTTP 接続、スケジューラー・スレッドなど) を処理するのに十分であることを確認します。

- WebSphere Commerce

サーバーが入出力制約を受けていることを確認します。WebSphere Commerce システム・パフォーマンスは、多くのファイル・アクセスまたはネットワーク・アクセスが発生すると影響を受ける場合があります。たとえば、すべてのロギングやトレースをオンにすると、システムは、ほとんどの時間をワークロードを処理するのではなく、データをディスクに書き込むことに費やします。

85 ページの『第 5 章 動的キャッシング』に説明されているとおりに、動的キャッシングを使用します。

サーバー・ベースのセッション管理を使用する場合は、「*WebSphere Application Server 5.0 Tuning Guide*」のガイドラインを参照してください。この資料は、

「WebSphere Application Server InfoCenter」

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) にあります。「**Monitor and Troubleshooting**」 → 「**Performance**」 → 「**Tuning Performance**」を拡張表示します。

- WebSphere Application Server

システムをチューニングする方法のガイドラインとして、「*WebSphere Application Server 5.0 Tuning Guide*」を使用します。この資料は、「WebSphere

Application Server InfoCenter」

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) にあります。「**Monitor and Troubleshooting**」 → 「**Performance**」 → 「**Tuning Performance**」を拡張表示します。

- その他の考慮事項:
 - WebSphere データ・ソース (最大および最小接続プール・サイズ、ステートメント・キャッシュ・サイズ)
 - Web サイトの設計
 - セキュリティー (構成、タイムアウト、認証、およびアクセス制御)
 - Web サーバーの問題 (プロセス処理、リソース使用、高速応答キャッシュ・アクセラレーター)
 - WebSphere エンジンの問題 (Java 仮想マシン (JVM)、トランスポート・キュー、JSP ファイルのキャッシュ、EJB コンテナ)
 - WebSphere Commerce セッション管理 (キャッシング、セッションをメモリーに保管、またはセッションをデータベースに保管)
 - WebSphere Application Server セッション管理 (メモリー内のセッション・カウソントの設定、オーバーフローの許可、タイムアウト間隔、および分散環境設定)
 - NFS (ネットワーク・ファイル・システム) のパフォーマンス・チューニング (ファイル・サーバーのチューニング)

WebSphere Commerce PMI モジュールを使用したパフォーマンスのモニター

WebSphere Application Server Performance Monitoring Infrastructure (PMI) を使用して、WebSphere Commerce システムのパフォーマンスをモニターできます。以前の WebSphere Commerce パフォーマンス・モニターの機能は、WebSphere Commerce PMI モジュールに移動しました。WebSphere Application Server で使用可能なさまざまなツールを使って、パフォーマンス・データをモニターおよび分析できます。サイト管理者はツールから収集された情報を使用して、パフォーマンス上の問題の検出およびパフォーマンス傾向の分析を行えます。これらのツールを使用して、ローカルまたはリモート・マシンから WebSphere Commerce Application Server のパフォーマンスを測定できます。

WebSphere Commerce Application Server は、URL、タスク、および JSP の統計を収集します。各データ・キーには、以下の情報を提供する、関連したカウンターの設定があります。

- 平均応答時間
- 最終応答時間
- 最小応答時間
- 最大応答時間
- ヒット
- 合計応答時間
- 標準偏差

PMI をセットアップするには、「WebSphere Application Server InfoCenter」(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) の『Monitoring performance』を参照してください。

PMI モジュールの設定

Performance Monitoring Infrastructure (PMI) は、クライアント/サーバー・アーキテクチャーを使用します。サーバーは、さまざまな WebSphere Application Server コンポーネントからパフォーマンス・データを収集します。クライアントは、1 つ以上のサーバーからパフォーマンス・データを検索して、データを処理します。

WebSphere Application Server サーバーは、メモリー内の PMI データを収集します。このデータは、サブレット応答時間やデータ接続プール使用量などのカウンターで構成されます。次いで、データ・ポイントが、Web クライアント、Java クライアント、または JMX クライアントを使用して検索されます。WebSphere Application Server には、Tivoli® Performance Viewer が準備されています。これは、パフォーマンス・データを表示およびモニターする Java クライアントです。

パフォーマンス・モニター・ツールを使用するには、WebSphere Commerce PMI モジュールを以下のようにしてセットアップする必要があります。

1. PMI インターフェースを使ってパフォーマンス・データをモニターするには、最初に WebSphere Application Server 管理コンソールを使ってパフォーマンス・モニター・サービスを有効にする必要があります。詳細なステップについては、「WebSphere Application Server InfoCenter」(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) の『Enabling PMI services in application server through the administrative console』セクションを参照してください。
2. 管理コンソールを使って NodeAgent のパフォーマンス・モニター・サービスを有効にすることによって、PMI インターフェースを使ってパフォーマンス・データをモニターすることもできます。詳細なステップについては、「WebSphere Application Server InfoCenter」(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) の『Enabling PMI services in NodeAgent through the administrative console』セクションを参照してください。
3. セットアップが完了したら、データを収集できます。有効にするデータ・カウンターを決定するモニター・レベルは、サーバーを再起動することなく動的に設定できます。これは、以下の 3 つのいずれかの方法で行います。
 - WebSphere Application Server 管理コンソールを使って、データ収集を有効にする。
 - Tivoli Performance Viewer (以前のリソース・アナライザー) を使って、パフォーマンス・モニター・サービスを有効にする。
 - コマンド行を使用して、パフォーマンス・モニター・サービスを有効にする。詳細なステップについては、「WebSphere Application Server InfoCenter」(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) の該当するタスクのセクションを参照してください。
4. パフォーマンス・データをモニターおよび分析します。

以下のツールを使って、データをモニターおよび分析できます。

- a. Tivoli Performance Viewer を使って、パフォーマンス・データをモニターする。このツールは、WebSphere Application Server に組み込まれています。
- b. その他の Tivoli モニター・ツールを使って、パフォーマンス・データをモニターする。
- c. ユーザーが開発したツールを使ってパフォーマンス・データをモニターする。パフォーマンス・データをモニターする独自のアプリケーションを作成してください。
- d. サード・パーティーのモニター・ツールを使用して、パフォーマンスをモニターする。

詳細なステップについては、「WebSphere Application Server InfoCenter」(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) の該当するタスクのセクションを参照してください。

Tivoli Performance Viewer の開始

Tivoli Performance Viewer は、WebSphere Application Server 5.0 に同梱された WebSphere Application Server 用のパフォーマンス・モニターです。ビューアーは、管理サーバーを周期的にポーリングすることによってパフォーマンス・データを検索します。データは継続的に収集され、ビューアー内から必要に応じて検索されます。Performance Viewer を起動して、WebSphere Application Server 管理コンソールから収集するデータのレベルを指定してください。グラフィカル・インターフェースを使用して、データを検索して表または図表に表示したり、データをログ・ファイルに保管したりすることができます。ビューアーを使用して、Commerce PMI だけでなく、WebSphere Commerce と関係のある他の WebSphere Application Server の一般的なパフォーマンス・データもモニターできます。

詳しくは、「WebSphere Application Server InfoCenter」の『Monitoring performance with Tivoli Performance Viewer (formerly Resource Analyzer)』を参照してください。

Tivoli Performance Viewer for WebSphere Commerce を開始する場合は、WebSphere Commerce に関連したカウンターを測定するために、WC_instance_name アプリケーション・サーバー用の SOAP または RMI ポート番号を指定する必要があります。そうしないと、Tivoli Performance Viewer は、デフォルトで server1 アプリケーション・サーバーに接続しようとします。WC_instance_name アプリケーション・サーバーの SOAP または RMI ポート番号を判別する方法の 1 つは、WC_instance_name の System.Out ログ・ファイルを調べるという方法です。サーバーの始動時に、SOAP または RMI ポート番号が表示されます。

Tivoli Performance Viewer を開始するためのステップは、以下のとおりです。

1. 管理コンソールを使って、PMI サービスを有効にします。
2. Tivoli Performance Viewer を開始します。これは、以下の 2 つの方法で行えます。
 - a. コマンド行からパフォーマンス・モニターを開始する。WAS_install_dir/bin ディレクトリーに移動して、tperfviewer スクリプトを実行します。

Windows 2000 環境では、以下のようにしてホストおよびポートを指定します。

```
tperfviewer.bat host_name port_number connector_type
```

AIX および他の UNIX® プラットフォームでは、以下のようにします。

```
tperfviewer.sh host_name port_number connector_type
```

たとえば、`tperfviewer.bat localhost 8879 SOAP` のように指定します。

`connector_type` は、SOAP または RMI です。8879 は、SOAP コネクターのデフォルト ND ポートです。9809 は、RMI コネクターのデフォルト ND ポートです。

- b. 「スタート」 → 「プログラム」 → 「IBM WebSphere」 → 「Application Server v.50」 → 「Tivoli Performance Viewer」をクリックします。

Tivoli Performance Viewer は、使用している WebSphere Application Server のパッケージを検出して、デフォルトの Remote Method Invocation (RMI) コネクターを使用して接続します。接続に失敗すると、ダイアログが表示され、新しい接続パラメーターが提供されます。Performance Viewer を開始するコマンド行を使用して、リモート・ホストまたは別のポート番号に接続できます。

3. データ収集設定を調整します。

「WebSphere Application Server InfoCenter」

(<http://www.ibm.com/software/webservers/appserv/infocenter.html>) の

『Setting performance monitoring levels』を参照してください。

WebSphere Commerce PMI モジュールのレポート値

Tivoli Performance Viewer を開始したら、「リソースの選択 (Resource Selection)」パネルで「Commerce カウンター・グループ」を拡張表示します。データが、ストア ID に属するタスク、URL、およびビューに基づいてグループ化されます。`StoreId=0` のタスク・コマンドしかないことに注意してください。「リソースの選択 (Resource Selection)」パネルから、特定の URL、ビュー、またはタスク・コマンドを選択すると、「カウンターの選択 (Counter Selection)」パネルに WebSphere Commerce カウンターが表示されます。

これらは、各タスク、URL、またはビュー用に組み込まれたカウンターです。時間値は経過時間を表しています。

カウンター名

定義

平均応答時間

タスクの平均応答時間。

最終応答時間

タスクの最終応答時間。

最小応答時間

タスクの最小応答時間。

最大応答時間

タスクの最大応答時間。

ヒット タスクが呼び出された合計回数。

合計応答時間

タスクの合計応答時間。

標準偏差

応答時間の標準偏差。標準偏差の計算に使用される公式は、データが標準分布に従うことを前提としています。

他のパフォーマンス・ツール

サイト管理者は、以下のツールを使用することもできます。

- IBM Tivoli Web Site Analyzer

Web サイトの使用量、正常性、保水性、およびサイトのコンテンツをキャプチャー、分析、保管、およびレポートすることによって、IBM Tivoli Web Site Analyzer は、訪問者サイトの対話とサイトの全体的なパフォーマンスを明らかにします。この見識を活用して、顧客の定着度と e-business の有効性を向上させるためにサイトを最適化できます。Web Site Analyzer は、ページ・コンテンツの人気と、特定の訪問者または顧客セグメントに対する目標オファーまたはキャンペーンの商品購入をトラッキングできます。Web ページまたは商品ページへのアクセスが少ないために、投資を減らすべき場所や、Web ナビゲーションにおける変更箇所を示すことができます。

Web サイト・アナライザーについて詳しくは、以下の Web アドレスを参照してください。

<http://www.ibm.com/software/sysmgmt/products/web-site-analyzer.html>

- Commerce Studio Page Detailer

Web ページを分析し、ID、サイズ、ソース、およびそのページの各アイテムを送信するのに要する時間を表示します。これらの詳細情報を使用して、パフォーマンス改善の余地があるエリアを識別し、エンド・ユーザーの体験を一層充実したものにできます。Page Detailer の実行は、Commerce Studio の他のコンポーネントを実行せずに行うことができます。

スケジューラーの保守

パフォーマンスが適切かどうかを確認するには、スケジューラーを適切に保守する必要があります。日常的な保守を行って、スケジューラーが正しく実行しているかを確認するのは、サイト管理者の責任です。

SCHCONFIG テーブルに含まれているジョブの各実行ごとに、1 つのエントリーが SCHSTATUS テーブルに含まれています。スケジューラーによって実行されるジョブは大量なので、SCHSTATUS テーブルに保持されるジョブ状況の記録は非常に多くなります。これは、WebSphere Commerce Server のパフォーマンスに悪影響を及ぼす場合があります。それで、「スケジューラー状況表示」ページを使用して、SCHSTATUS テーブルを定期的にクリーンアップすることをお勧めします。このページは、管理コンソールの「構成」メニューにあります。ジョブ状況の記録を削除すると、タイム・スタンプおよびジョブ参照番号に基づいてジョブをクリーンアップすることによって、SCHSTATUS テーブルのサイズを削減することができます。

あるいは、CleanJob ジョブの実行をスケジュールして、SCHSTATUS テーブルのサイズをトリムすることもできます。endTime パラメーターを使用して、前週または前月のジョブ状況の記録を除去するかどうか指定してください。CleanJob コマンドの構文とスケジュールングについては、ご使用のプラットフォーム用の「WebSphere Commerce オンライン・ヘルプ」を参照してください。

注: CleanJob を実行すると、ストア発行ページからストア発行状況が除去されます。

ジョブの履歴を保持する必要がある場合は、スケジューラーの autoClean プロパティをいつでも「ON」に設定できます。これで、ジョブの実行が終了すると、SCHSTATUS テーブルからジョブが自動的に除去されます。

スケジューラーについての詳細は、26 ページの『サービスのスケジュールング』、または「WebSphere Commerce オンライン・ヘルプ」を参照してください。

LDAP の複製

WebSphere Commerce は、LDAP を使用した認証、および認証とプロファイル・データの保管をサポートしています。一部のデータは、WebSphere Commerce データベースと LDAP サーバーの間で複製されます。複製のほとんどは、ldapentry.xml ファイルを使用して構成することができます。複製ロジックが特定の Member Subsystem コマンドからしか起動されなかった以前のバージョンの WebSphere Commerce とは異なり、最新のバージョンの WebSphere Commerce では、オブジェクト・レベルで複製を実行します。つまり、必要なときにいつでも複製が生じるということです。その結果、LDAP サーバーがダウンした場合には、エラーが生成されます。

ステージング・サーバーでサイトをテストする

ステージング・サーバーでサイトをテストするには、以下を行います。

1. ステージング・サーバーを構成します。
2. あらゆるカスタム・テーブルにトリガーを作成します。
3. リモート・データベースを構成します (該当する場合)。
4. データをステージング・データベースにコピーします。
5. サイトをテストします。
6. Stage Check コマンドを実行して、固有索引キーが競合していないことを確認します。
7. データを実動データベースに伝搬します。
8. ファイルを実動サーバーにコピーします。
9. ステージ・オブジェクトを STAGLOG テーブルから削除します。

ステージング・サーバーを構成する

WebSphere Commerce マシンをどれでもステージング・サーバーとしてセットアップすることができます。ステージング・サーバーは、インストール中またはインストール後に構成することができます。WebSphere Commerce インストール中にステージング・サーバーをセットアップすることについては、「WebSphere Commerce

インストール・ガイド」で説明されています。インストール後にステージング・サーバーをセットアップする場合は、以下のようにします。

1. 構成マネージャーを使用して、ステージング・サーバーとして使用するための別の WebSphere Commerce インスタンスを作成および構成します。
2. 「データベース」パネル上のチェック・ボックス「ステージング・サーバーの使用」を選択し、そのインスタンスをステージング・サーバーとして構成します。
3. 構成マネージャーの「キャッシュ」パネルで、キャッシングが使用不可になっていることを確認してください。

カスタム・テーブルにトリガーを作成する

トリガーは、データベース・レコードの変更を識別する STAGLOG テーブルにエントリーを作成します。テーブルに同じデータ有効範囲およびキー特性が含まれる場合、新規のテーブルに対する既存のトリガーの設定を変更することができます。

新規テーブルを何も作成しなかった場合、このステップを実行する必要はありません。新規テーブルを作成した場合、カスタマイズされたテーブル用にステージング・サーバーを構成する場合の説明を参照してください。

リモート・データベースを構成する

ステージング・サーバーを実動サーバー以外のマシンにセットアップした場合、リモート・データベースを構成する必要があります。ステージング・ユーティリティーをステージング・サーバーから実行する計画の場合は、ご使用の実動サーバーをステージング・サーバーのリモート・データベースとして構成する必要があります。ステージング・ユーティリティーを実動サーバーから実行する計画の場合は、ステージング・データベースを実動サーバーのリモート・データベースとして構成する必要があります。



 DB2 データベースの場合、「DB2 管理ガイド」を参照してください。

 Oracle データベースの場合、製品の資料を参照してください。

ステージング・データベースにデータをコピーする

データを実動データベースからステージング・データベースにコピーするには、以下のようにします。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

•  

```
stagingcopy -scope_all -sourcedb production_database_name  
-destdb staging_database_name
```

•  


```
stagingcopy -scope _all_ -sourcedb production_database_name
             -destdb staging_database_name -dbtype oracle -sourcedb_user
             user -sourcedb_passwd password -destdb_user
             user -destdb_passwd password
```

+

- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2

+

```
. stagingcopy.sh -scope _all_ -sourcedb production_database_name
                -destdb staging_database_name -sourcedb_user user -destdb_user user
```

+

- ▶ AIX ▶ Solaris ▶ Oracle

```
. stagingcopy.sh -scope _all_ -sourcedb production_database_name
                -destdb staging_database_name dbtype oracle -sourcedb_user
                user -sourcedb_passwd password
                -destdb_user user -destdb_passwd password
```

注: Oracle データベース名には *host:port:sid* を使用してください。
myhost:1521:mydb などです。

5. *stagingcopy_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。
6. ステージング・サーバーのインスタンスを停止して、再始動します。

Stage Check コマンドを実行する

ステージング・データベースと実動データベースの間で、固有索引キーが競合する可能性がないかどうかを検査するには、以下のようにします。

Stage Check コマンドを実行して、固有索引キーが競合していないことを確かめるには、以下のようにします。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

- ▶ Windows ▶ DB2

```
stagingcheck -scope _unique_index_ -sourcedb staging_database_name
              -destdb production_database_name
```

- ▶ Windows ▶ Oracle

```
stagingcheck -scope _unique_index_ -sourcedb staging_database_name
              -destdb production_database_name -dbtype oracle -sourcedb_user
              user -sourcedb_passwd password -destdb_user
              user -destdb_passwd password
```

+

- ▶ AIX ▶ 400 ▶ Solaris ▶ Linux ▶ DB2

+

```
. stagingcheck.sh -scope _unique_index_ -sourcedb staging_database_name
                 -destdb production_database_name -sourcedb_user user -destdb_user user
```

+

- ▶ AIX ▶ Solaris ▶ Oracle

```
. stagingcheck.sh -scope _unique_index_ -sourcedb staging_database_name
                 -destdb production_database_name dbtype oracle -sourcedb_user
                 user -sourcedb_passwd password -destdb_user
                 user -destdb_passwd password
```

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。 *myhost:1521:mydb* などです。

5. *stagingcheck_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

実動データベースにデータを伝搬する

データをステージング・データベースから実動データベースに伝搬するには、以下のようにします。

1. PATH 環境変数を設定します。
2. データベースを構成します。
3. ログ・ファイルを書き込むディレクトリーに移動します。
4. 次のように入力します。

-  

```
stagingprop -scope _all_ -sourcedb staging_database_name -destdb  
production_database_name
```

-  

```
stagingprop -scope _all_ -sourcedb staging_database_name  
-destdb production_database_name -dbtype oracle -sourcedb_user  
user -sourcedb_passwd password -destdb_user  
user -destdb_passwd password
```

-      

```
stagingprop.sh -scope _all_ -sourcedb staging_database_name  
-destdb production_database_name -sourcedb_user user -destdb_user user
```

-   

```
stagingprop.sh -scope _all_ -sourcedb staging_database_name  
-destdb production_database_name dbtype oracle -sourcedb_user  
user -sourcedb_passwd password -destdb_user  
user -destdb_passwd password
```

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。 *myhost:1521:mydb* などです。

5. *stagingprop_yyyy.mm.dd_hh.mm.ss.zzz.log* ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

データを実動データベースに伝搬することの詳細については、例を参照してください。

ファイルを実動サーバーにコピーする

ステージング・サーバー上のイメージまたは HTML ファイル (データベースのデータではない) を追加、変更、削除するには、実動サーバーにこれらのファイルを手動でコピーする必要があります。

実動サーバーにファイルをコピーするには、以下のようにします。

1. 実動サーバーの正しいディレクトリーにファイルをコピーします。

- a. すべての静的 HTML ファイル、関連したイメージ・ファイル、および他の組み込みファイルを含む ZIP ファイルを作成します。このファイルには、新規、更新済み、および未変更のファイルが含まれていなければなりません。
 - b. この ZIP ファイルを実動サーバーに転送します。
 - c. 各 ZIP ファイルを、実動サーバー・ディレクトリー構造の対応するディレクトリーに unzip します。HTML ファイルを移動した場合、WebSphere Commerce 構成が新規のディレクトリー内のファイルを指すように編集します。
2. 実動サーバー上の未使用ディレクトリーを削除します。

ステージ・オブジェクトを削除する

ステージ・オブジェクトを削除するには、次のようにします。

1. PATH 環境変数を設定します。
2. ログ・ファイルを書き込むディレクトリーに移動します。
3. 次のように入力します。

-  

```
dbclean -object staglog -type obsolete -db dbname
-days daysold -loglevel loglevel
```

-  

```
dbclean -object staglog -type obsolete -db dbname
-days daysold -loglevel loglevel -dbtype oracle
-dbuser user -dbpasswd password
```

-     

```
. dbclean.sh -object staglog -type obsolete -db dbname
-dbuser user -days daysold -loglevel loglevel
```

-   

```
. dbclean.sh -object staglog -type obsolete -db dbname
-days daysold -loglevel loglevel -dbtype oracle
-dbuser user -dbpasswd password
```

注: Oracle データベースの名前には、*host:port:sid* の形式を使用してください。 *myhost:1521:mydb* などです。

4. `dbclean_yyyy.mm.dd_hh.mm.ss.zzz.log` ファイルを調べて、コマンドが正常に実行されたかどうか検査します。

ステージ・オブジェクトを削除することについての追加の例は、オブジェクトの削除の例を参照してください。

WebSphere Commerce Payments のパフォーマンス・チューニング・パラメーター

WebSphere Commerce Payments には、WebSphere Commerce Payments 内部リソース割り当てを制御できるチューニング・パラメーターが含まれています。

重要: これらのパラメーターは、WebSphere Commerce Payments によく精通している管理者だけが変更すべきです。これらのパラメーター値のいずれかを高く

設定しすぎると、パフォーマンスの低下や、WebSphere Commerce Payments の起動時の明らかな障害の原因となる場合があります。

以下のようにすることを、強く 推奨します。

- 変更は小規模な増分のみとする。
- 一度に 1 つのパラメーターだけを変更する。次の変更を加える前に、この効果を調べる。
- それらを実動システムで使用する前に、非基幹システム上で変更を十分にテストする。

WebSphere Commerce Payments は、以下のスレッド・プールのセットを管理します。

プロトコル・スレッド・プール

このプールのスレッドは、カセット内のサーバー側の プロトコル固有のメッセージを処理するタスクに割り当てられます。このようなプールは、独自の ComPoints を定義するカセットごとに 1 つあります。

400 プロパティ `wpm.poolsizesize` は、プロトコル・スレッド・プール・サイズを変更します。

プール・サイズは、WebSphere Application Server 管理コンソールで以下のステップを実行することによって変更できます。

1. 「サーバー」を拡張表示します。
2. 「**Application Servers**」 > 「**wpm Commerce Payments Server**」 > 「**プロセス定義 (Process Definition)**」 > 「**Java 仮想マシン (Java Virtual Machine)**」 > 「**カスタム・プロパティ (Custom Properties)**」 > 「**wpm.poolsizesize**」をクリックします。
3. 希望する値を入力します。デフォルトは、8 です。
4. 「適用」をクリックします。

サーバー側のプロトコル・メッセージに一般的に関係する処理の性質（つまり、レベルが高くなるネットワーク通信）として、これらのスレッドは、それぞれの特定の要求について一般に長時間保持されます。したがって、プロトコル・スレッドに依存してプロトコル・メッセージを処理するカセットは、重要なプロトコル・メッセージに関係する一般的な数の同時トランザクションを処理するために、十分なプロトコル・スレッド・プールで構成する必要があります。

サービス・スレッド・プール

このプール内のスレッドは、フレームワークに加えカセットによって使用され、同時タスクまたは将来のタスクをバックグラウンドで実行します。このプール内のスレッドのデフォルト数値は 6 です。

400 プロパティ `wpm.spoolsizesize` は、サービス・スレッド・プール・サイズを変更します。

プール・サイズは、WebSphere Application Server 管理コンソールで、以下のようにして変更できます。

1. 「サーバー」を拡張表示します。
2. 「Application Servers」 > 「wpm Commerce Payments Server」 > 「プロセス定義 (Process Definition)」 > 「Java 仮想マシン (Java Virtual Machine)」 > 「カスタム・プロパティ (Custom Properties)」 > 「wpm.spoolsize」をクリックします。
3. 「値 (Value)」フィールドで、希望するプール・サイズを指定します。デフォルトは、6 です。
4. 「適用」をクリックします。

当初は、サービス・スレッドの最も一般的な使用法は、以降のいずれかの時点において実行をスケジュールされたタスクを処理することでした。このようなタスクの例は、カセットのバックエンド・プロセッサとの再試行トランザクションや、定期保守タスクなどです。

wpm.disableDuplicateOrderCheck

このパラメーターは、WebSphere Commerce Payments に、新規のオーダーの処理時に、重複オーダーの検査をバイパスし、データベースへのアクセスを除去するように指示します。結果として、トランザクション・スループットが増えます。このパラメーターは、マーチャントが重複するオーダー番号を生成しない場合にのみ使用することを推奨します。そうでない場合に使用すると、オーダーが失敗する原因となります。このパラメーターを使用可能にするには、WebSphere Application Server 管理コンソールから以下のステップを実行します。

1. 「サーバー」を拡張表示します。
2. 「Application Servers」 > 「wpm Commerce Payments Server」 > 「プロセス定義 (Process Definition)」 > 「Java 仮想マシン (Java Virtual Machine)」 > 「カスタム・プロパティ (Custom Properties)」をクリックします。
3. 「新規」をクリックします。
4. 「名前 (Name)」フィールドで、パラメーター名 wpm.spoolsize を入力します。
5. 「値 (Value)」に 1 を入力します。
6. 「適用」をクリックします。

ヒント: ご使用のデータベース製品が、一度にそのデータベースに対して確立できる同時接続の数を制限している場合があります。これは、技術的な限界である場合もあれば、購入したデータベース製品のご使用条件によって課されたものである場合もあります。上述のパラメーターを調整する場合は、これらの要因も考慮する必要があります。これらのパラメーターのどちらかを高く設定しすぎてそのような制限を超えてしまうと、Payment サーブレットの初期化は失敗します。

iSeries 上での Payments インスタンス・データベースの管理

新しいオーダーや決済が WebSphere Commerce Payments によって処理されるにつれて、Payments インスタンス・データベース・コレクションが大きくなっていきます。iSeries のストレージ・スペースがいっぱいになった場合、システム・オペレーターは CPI099C - Critical storage lower limit reached のような警告メッセージを受け取ることがあります。そのような場合、WebSphere Commerce Payments インスタンス・データベース内のスペースを解放することを考慮してください。

WebSphere Commerce Payments インスタンス・データベース・コレクションが作成される時、ジャーナリングは *YES に設定されます。つまり、インスタンス・ライブラリーにデータベース・ジャーナル・レシーバー (データベース・ファイル変更などのイベントがいつ発生したかを示すエントリー) が含まれることとなります。iSeries の場合、ジャーナル QSQJRN が WebSphere Commerce Payments インスタンス・ライブラリーに作成され、QSQJRNXXXX という名前のジャーナル・レシーバーがインスタンス・ライブラリーの中に作成されます。WebSphere Commerce Payments インスタンス・ライブラリーのサイズが大きくなり過ぎたとき、スペースを解放するために、インスタンス・ライブラリー内の古いジャーナル・レシーバー (名前が QSJRXXXX、タイプが *JRNRCV のもの) を削除することができます。最新のジャーナル・レシーバーは依然としてデータベースに付加されますから、削除するべきではありません。さらに、IBM サービスから入手可能な Commerce Payments Pruneorders ユーティリティーを使用して、データベース・テーブルを枝取りできます。

ストレージのクリーンアップについて、詳しくは「AS/400 システム解説書 (SC88-5454)」を参照してください。

第 8 章 トラブルシューティング

サイト管理者には、トラブルシューティングを行い、システムの問題の解決策を見定める責任があります。このセクションでは、Adapter for CrossWorlds® と WebSphere Commerce Payments のトラブルシューティングを取り上げます。以下のトラブルシューティングの詳細は、「WebSphere Commerce インストール・ガイド」を参照してください。

- ダウンロード可能なツール
- ログ・ファイル
- WebSphere Application Server の問題
- Web サーバーの問題
- WebSphere Commerce の問題
- データベースの問題

Adapter for CrossWorlds®

このセクションでは、アダプターの構成中に発生する可能性のある問題と、問題を解決するためのアクションを列挙します。

- **問題:** あるマシンに InterChange Server (ICS) とリポジトリがあり、別のマシンに WebSphere Commerce Server があります。Visigenic ORB エージェントが全サブネットで機能するために必要な構成情報は何か? 接続しようとする、ダイアログ・ボックスがポップアップし、サーバーが稼働していることを別途確認できるにもかかわらず、WebSphere Commerce Server が「Can't locate the InterChange Server '<server name>'. The server is probably not running」と表示されるという症状が見られます。

解決方法 1:

1. Visibroker インストールの adm ディレクトリー (d:\%inprise%\vbroker\adm) を指す VBROKER_ADM というユーザー・レベルの環境変数を作成します。
2. adm ディレクトリーに、この ICS に接続する必要のあるクライアント・マシンの IP アドレスまたはホスト名が入っている agentaddr というテキスト・ファイルを作成します。
3. osagent プロセスを再始動し、VBROKER_ADM が、このプロセスを開始するユーザーまたはシェルの環境にあることを確認します。これで、WebSphere Commerce Server を ICS に接続できるはずですが、各 osagent には、WebSphere Commerce Server のリモート IP が入っているそれぞれの agentaddr テキスト・ファイルが含まれている必要があります。

注: agentaddr には拡張子がなく、実行されている必要がある唯一の osagent は ICS サイドに置きます。

解決方法 2: 代替環境変数 OSAGENT_ADDR_FILE を使用してうまく対処することもできます。この環境変数は、agentaddr ファイルのパスとファイル名を指定します。

ORB がディスカバリーを行う際は数分かかるので、いずれかの方法を放棄する前に 10 分以上再接続を試行してください。

- **問題:** WebSphere Commerce Server が、複数の InterChange Server に接続できません。OSAGENT_ADDR 環境変数を WebSphere Commerce Server マシン上で設定しましたが、指定したマシンで稼働している InterChange Server にしか接続できません。2 つの InterChange Server が 2 つの異なるサブネット上で稼働していて、WebSphere Commerce Server が第 3 のマシン上にあります。osagent が各 InterChange Server マシン上で実行されていて、各マシンの VBROKER_ADM 環境変数は、各 InterChange Server/osagent マシンの IP アドレスが入っている agentaddr ファイルを含むディレクトリーを指すよう正しく構成されています。**解決方法:** 各 osagent マシンの VBROKER_ADM 環境変数で指定されたディレクトリーに localaddr テキスト・ファイルを作成し、他の osagent マシンの IP アドレスまたは DNS 別名を指定します。
- **問題:** NT マシンの WebSphere Commerce Server を Solaris 上で稼働している ICS に接続しようとしています。**解決方法:** WebSphere Commerce Server と ICS の両方の agentaddr ファイルに、接続先のマシンの IP アドレスだけでなく、マシン自体の IP アドレスも入っていることを確認します。したがって、NT マシン上の agentaddr ファイルには、マシン自体の IP アドレスと Solaris マシンの IP アドレスの両方が入っている必要があります。

注:

1. このセクションで扱われていない問題については、IBM サポート担当者に連絡してください。
2. Adapter for CrossWorlds[®] は、iSeries、Linux、IBM eServer[™] zSeries[®]、または S/390[®] Linux 上ではサポートされていません。

注: このセクションで扱われていない問題については、IBM サポート担当者に連絡してください。

WebSphere Commerce Payments

各ビジネス・モデルには固有の要件があり、それゆえに、WebSphere Commerce Payments の機能を使用する方法も少し異なっています。WebSphere Commerce Payments は、多くの場合、そのパフォーマンスおよび機能性を最大限に発揮するように、指定されたビジネス環境に合わせてチューニングすることができます。

ハイパフォーマンス環境用のチューニング

TCP/IP ソケットの待ち時間

WebSphere Commerce Payments への各要求は TCP ソケットに入れられ、TIME_WAIT 状態になります。要求は、ソケットで数分間保持されます。大量の要求にサービスするマシンでは、より多くのソケットが TIME_WAIT 状態か TIME_CLOSED 状態に入れられるため、結果としていくつかの要求が拒否されるようになります。この場合、戻りコードは Cannot connect to WebSphere Commerce Payments です。これは、すべての TCP 接続にあてはまり、そして必要な振る舞いです。

TCP ソケットはしばらくの間 TIME_WAIT 状態になり、続いて行われるソケットでのどんな通信も、新しく結び付けられたソケットでの新規の通信と間違えられないようにします。この状態になる時間は、理論上では 2 MSL (最大セグメント存続期間の 2 倍) とされています。実際のところでは、Windows オペレーティング・シス

テムおよび Solaris でのデフォルトの TIME_WAIT は 4 分、AIX では 2 分になっています。デフォルトは iSeries システムでは約 2 分です。TIME_WAIT 値をオペレーティング・システム上で変更することによって、大量の要求を処理するユーザーはこの問題を減らすことができます。以下の例は、Windows、AIX、および Solaris オペレーティング・システムで間隔を再構成する方法を示しています。

例外: 代替のスタックが使用されている場合は、他の手段が必要です。

Windows:

1. 次のレジストリーを探します。

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\tcpip\Parameters
  \TcpTimeWaitDelay
```

ご使用の Windows レジストリーに上記の項目がない場合は、以下のステップ 2 に従って、これを作成する必要があります。

2. このレジストリーを新しい DWORD 項目として編集します。
3. 30 から 300 の間で任意の値を設定します (この値は秒数を表します)。お勧めする値は 30 です。

AIX:

```
no -o tcp_timewait=1
```

値 (1) は、15 秒の増分を表します。たとえば、1 は 15 秒を、2 は 30 秒を表し、以下同様です。お勧めする値は 1 または 2 です。

Solaris:

```
ndd -set /dev/tcp tcp_close_wait_interval 30000
```

値 (30000) はミリ秒です。

付録 A. LDAP シナリオ

LDAP シナリオ: メンバー・リポジトリとしての LDAP サーバー

LDAP サーバーをメンバー・リポジトリとして使用するためのシナリオとして、WebSphere Commerce の新しいインスタンスを作成し、それを指定して LDAP サーバーをメンバー・リポジトリとして使用するという方法があります。このシナリオでは、ルート組織、デフォルト組織、および「wcsadmin」ユーザー・エントリーのブートストラップ・データが、WebSphere Commerce データベースに追加されます。

このシナリオを使用する場合は、以下を実行する必要があります。

1. ディレクトリー・サーバーで必要な接尾部を作成します。WebSphere Commerce で使用されるユーザーおよび組織のエンティティー・エントリーは、これらの接尾部の下に置かれます。
2. ディレクトリー・サーバー上で「ルート組織」、「デフォルト組織」、および「wcsadmin」用のエントリーを作成し、「デフォルト組織」と「wcsadmin」の両方を「ルート組織」の下に置きます。LDIF ファイルのサンプルについては、『LDAP サーバー内でのブートストラップ・エントリーの作成』関連リンクを参照してください。
3. LDAP サーバーで「wcsadmin」エントリー用のパスワードを作成します。
4. ディレクトリー・サーバーに各エントリーを作成したら、ルート組織 (MEMBER_ID -2001) とデフォルト組織 (MEMBER_ID -2000) の正確な識別名を使用して、ORAGENTITY テーブルの DN 列を更新する必要があります。
5. WebSphere Commerce 属性を LDAP 属性にマッピングする ldapentry.xml ファイルをセットアップします。「wcsadmin」ユーザーを検出できるように、ユーザー用の検索ベースが ldapentry.xml ファイルに指定されていることを確認します。
6. 正しいパスワードを指定して、「wcsadmin」でログオンします。

WebSphere Commerce や別のアプリケーションを使用して、さらに多くのユーザーがディレクトリー・サーバー上に作成されると、それらのユーザーは自分の RDN 値または DN 値を使用して、WebSphere Commerce にログオンできます。ユーザーが RDN を使用してログオンする場合は、ldapentry.xml ファイルに検索ベースが正しく指定されていることを確認してください。

注: WebSphere Commerce インスタンスを複数作成し、それらのインスタンスで LDAP サーバーを共用する場合は、各 WebSphere Commerce インスタンス内の LDAP サーバーの構成を同じにしてください。

付録 B. LDAP ファイル

この付録では、以下の LDAP ファイルについて詳しく説明します。

- ldapentry.dtd
- ldapentry.xml
-

ldapentry.dtd

WebSphere Commerce の以前のバージョンで使用されていた ldapmap.dtd ファイルに代わって、ldapentry.dtd が使用されることになりました。以下は、LDAP で使用される DTD です。この DTD には変更を加えないでください。

```
<!-- new mapping file -->
<!ELEMENT ldapentry (entry+)>
<!ELEMENT entry (ldapsetting,ldapmap)>
<!ELEMENT ldapsetting (ldaprtn,ldapocs,ldapbase)>
<!ELEMENT ldaprtn EMPTY>
<!ELEMENT ldapocs EMPTY>
<!ELEMENT ldapbase EMPTY>
<!ELEMENT ldapmap (map+)>
<!ELEMENT map (objectAttribute+,objectSeparator?,ldapAttribute)>
<!ELEMENT objectAttribute EMPTY>
<!ELEMENT objectSeparator EMPTY>
<!ELEMENT ldapAttribute EMPTY>

<!ATTLIST entry
entryName      (User|Organization|OrganizationalUnit) #REQUIRED>

<!ATTLIST ldapbase
defaultBase    CDATA      #REQUIRED
searchBase     CDATA      #REQUIRED>

<!ATTLIST ldaprtn
rdnName        CDATA      #REQUIRED  keyAttrName    CDATA      #REQUIRED
keyObjName     CDATA      #REQUIRED>

<!ATTLIST ldapocs
objClass       CDATA      #REQUIRED>

<!ATTLIST objectAttribute
attrName       CDATA      #REQUIRED>

<!ATTLIST objectSeparator
attrSeparator  CDATA      #IMPLIED>
<!ATTLIST ldapAttribute
name           CDATA      #REQUIRED
operation      (replace|add) #REQUIRED
flow           (ldapToWcs|wcsToLdap|bothDirections) #REQUIRED>
<!-- End of DTD -->
```

ldapentry.xml

WebSphere Commerce の以前のバージョンで使用されていた ldapmap.xml ファイルに代わって、ldapentry.xml ファイルが使用されることになりました。以前使用していた ldapmap.xml ファイルの内容は、ldapentry.xml ファイルに移してください。ldapentry.xml ファイルは、最も頻繁に使用されるフィールドのデフォルト・マッ

ピングを提供します。デフォルト・スキーマ・マッピングの LDAP 属性は、ユーザーのデフォルト・オブジェクト・クラス (person、organizationalPerson、inetOrgPerson および ePerson) と、組織エンティティのデフォルト・オブジェクト・クラス (organization および organizationalUnit) に基づいています。これらのオブジェクト・クラスは、WebSphere Commerce によってサポートされるすべての LDAP サーバーに対して定義されます。デフォルト・スキーマ・マッピングを拡張して、これらのオブジェクト・クラス以外のものを追加するには、まずスキーマ拡張機能を LDAP サーバーで実行してください。その後、新しい属性のマッピングを ldapentry.xml ファイルに入れることができます。WebSphere Commerce の内部で生成されるキーや、USERS テーブル内の Lastsession のような頻繁に変更されるデータなどの属性を、無理に LDAP で管理しないようお勧めします。

LDAP 属性について詳しくは、www.as400.ibm.com/ldap/schema を参照してください。

```
<?xml version="1.0"?>
<!DOCTYPE ldapentry SYSTEM "ldapentry.dtd">
<ldapentry>
  <entry entryName="User">
    <ldapsetting>

<ldaprdn rdnName="uid"
  keyAttrName="logonId" keyObjName="UserRegistry"/>
    <ldapocs
objClass="top;person;organizationalPerson;inetOrgPerson"/>
    <ldapbase
defaultBase="o=Default Organization,o=Root Organization"
searchBase="o=Root Organization"/>
    </ldapsetting>
    <ldapmap>
    <map>

    <
objectAttribute attrName="logonPassword"/>

<ldapAttribute name="userPassword"
  operation="replace" flow="wcsToLdap"/>
    </map>
    <map>

<objectAttribute attrName="lastName"/>

<objectAttribute attrName="firstName"/>

<objectSeparator attrSeparator="/" />

<ldapAttribute name="cn" operation="replace"
  flow="wcsToLdap"/>

    </map>

<map>

    <objectAttribute attrName="lastName"/>

    <ldapAttribute name="sn" operation="replace
```



```

" flow="bothDirections"/>
</map>
  <map>

<objectAttribute attrName="firstName"/>
<ldapAttribute name="givenName" operation="replace
" flow="bothDirections"/>
  </map>
  <map>

<objectAttribute attrName="phone1"/>

<ldapAttribute name="homePhone" operation="replace
" flow="bothDirections"/>
  </map>
  <map>

<objectAttribute attrName="zipCode"/>

<ldapAttribute name="postalCode" operation="replace
" flow="bothDirections"/>
  </map>
  <map>

<objectAttribute attrName="address1"/>
<objectAttribute attrName="address2"/>
<objectAttribute attrName="address3"/>
<objectSeparator attrSeparator="/"/>
<ldapAttribute name="postalAddress" operation="replace
" flow="bothDirections"/>
  </map>
</ldapmap>
</entry>

entry entryName="Organization">
  <ldapsetting>

  <ldaprdn rdnName="o" keyAttrName="orgEntityName"
keyObjName="Organization"/>
  <
ldapocs objClass="top;organization"/>
  <
ldapbase defaultBase="o=Root Organization" searchBase="
o=Root Organization"/>
  </ldapsetting>
  <ldapmap>
  <map>

  <objectAttribute attrName="businessCategory"/>

  <ldapAttribute name="businessCategory" operation="
replace" flow="bothDirections"/>
  </map>
  <map>

  <objectAttribute attrName="description"/>

  <ldapAttribute name="description" operation="
replace" flow="bothDirections"/>

```

```

        </map>
        <map>

<objectAttribute attrName="address1"/>

<objectAttribute attrName="address2"/>

<objectAttribute attrName="address3"/>

<objectSeparator attrSeparator="/"/>

<ldapAttribute name="postalAddress" operation="replace
" flow="bothDirections"/>
        </map>
        <map>

<objectAttribute attrName="phone1"/>

<ldapAttribute name="telephoneNumber" operation="replace
" flow="bothDirections"/>
        </map>
    </ldapmap>
</entry>
<entry entryName=
"OrganizationalUnit">
    <ldapsetting>

        <ldaprdn rdnName="ou" keyAttrName="orgEntityName"
keyObjName="Organization"/>

<ldapocs objClass="top;organizationalUnit"/>

<ldapbase defaultBase="o=Root Organization"
searchBase="o=Root Organization"/>
        </ldapsetting>

        <ldapmap>
        <map>

<objectAttribute attrName="businessCategory"/>

<ldapAttribute name="businessCategory" operation="replace
" flow="bothDirections"/>
        </map>
        <map>

<objectAttribute attrName="description"/>

<ldapAttribute name="description" operation="replac
e" flow="bothDirections"/>
        </map>
        <map>

<objectAttribute attrName="address1"/>

<objectAttribute attrName="address2"/>

<objectAttribute attrName="address3"/>

<objectSeparator attrSeparator="/"/>

<ldapAttribute name="postalAddress" operation="replace
" flow="bothDirections"/>
        </map>
        <map>

<objectAttribute attrName="phone1"/>

```

```

<ldapAttribute name="telephoneNumber" operation="replace
" flow="bothDirections"/>
  </map>
</ldapmap>
</entry>
</ldapentry>

```

ldapentry.xml ファイルの形式は、次のとおりです。

entry entry エレメントは、マッピング対象のメンバーのタイプを示します。有効な値は、User、Organization、または OrganizationalUnit です。

ldapsetting - ldaprdn - rdnName

RDN 属性となる LDAP 属性を指定します。

ldapsetting - ldaprdn - keyAttrName

RDN 属性にマップする WebSphere Commerce 属性を指定します。ユーザーの属性名は、UserRegistrationAdd コマンドの構文に示されるプロパティ名と対応します。属性名には、大文字小文字の区別があります。組織および組織単位の場合 (OrgEntityAdd コマンドが使用される) も同様です。

ldapsetting - ldaprdn - keyObjName

WebSphere Commerce 属性が keyAttrName に指定されている WebSphere Commerce アクセス bean を指定します。

ldapsetting - ldapocs - objClass

LDAP エントリーの作成に使用される LDAP オブジェクト・クラスを指定します。

ldapsetting - ldapbase - defaultBase

ログオンに RDN が使用される場合は、ログオン時に WebSphere Commerce がユーザーを検索するための、最初の検索ベース DN を指定します。組織エンティティーの場合、defaultBase は、必要な場合にいつでも WebSphere Commerce が組織エンティティーを検索するための、最初の検索ベース DN を指定します。

ldapsetting - ldapbase - searchBase

WebSphere Commerce がユーザーや組織エンティティーを検索するための、残りのベース DN を指定します。

map -objectAttribute - attrName

WebSphere Commerce 属性名。ユーザーの属性名は、UserRegistrationAdd コマンドの構文に示されるプロパティ名と対応します。属性名には、大文字小文字の区別があります。組織および組織単位の場合 (OrgEntityAdd コマンドが使用される) も同様です。

map -ldapAttribute - name

attrName に指定された WebSphere Commerce 属性にマップされる LDAP 属性の名前。

map -ldapAttribute - flow

属性値が LDAP から読み取られるのか、LDAP に書き込まれるのか、あるいは読み書きの両方が行われるのかを指定します。有効な値は、ldapToWcs、wcsToLdap、または bothDirections です。

map -ldapAttribute - operation

LDAP のための属性値の変更方法を指定します。有効な値は、replace または add です。値 replace を指定すると、現行のユーザー情報を、提供される新規情報に更新します (たとえば、既存の電話番号を新しい電話番号に置き換える)。値 add を指定すると、ユーザーに新規エントリーが組み込まれます (たとえば、現行のユーザー情報に新しい電話番号を追加したり、ユーザーの電話番号リストを作成したりする)。

map - objectSeparator - attrSeparator

1 つの LDAP 属性に対して複数の WebSphere Commerce 属性を保管/検索する際に使用される区切り文字。

付録 C. WebSphere Commerce Payments チュートリアル

このチュートリアルでは、OfflineCard Cassette を使った WebSphere Commerce Payments の初めてのセットアップをガイドします。この初期セットアップの一部として、さらに、一般的な管理機能および決済機能を実演するため、WebSphere Commerce Payments には、OfflineCard Cassette および Sample Checkout を使うチュートリアル・サポートが備えられています。管理機能、構成機能、および決済機能についての詳細は、「WebSphere Commerce Payments ユーザー・インターフェース」のオンライン・ヘルプを参照してください。

OfflineCard Cassette を使った運用 WebSphere Commerce Payments をセットアップするには、以下の 6 つの構成タスクを実行する必要があります。

1. WebSphere Commerce Payments ユーザー・インターフェースへのアクセス
2. WebSphere Commerce Payments マーチャントの作成と決済カセットの認可
3. WebSphere Commerce Payments ユーザーの定義
4. ユーザー役割の割り当て
5. アカウントの作成
6. ブランドの作成

構成タスクが完了したら、オーダーの発行や、マーチャントが通常は日常的に行っているような、次のような決済処理作業を開始することができます。

7. オーダーの作成
8. オーダーの承認
9. 決済のデポジット
10. バッチの清算
11. クレジットの発行
12. 日ごとのバッチ合計の表示

ステップ 1: WebSphere Commerce Payments ユーザー・インターフェースへのアクセス

まず、Payments 管理者として Payments にログインします。

WebSphere Commerce Payments ユーザー・インターフェースにログインするには、以下のようにします。

1. Web ブラウザーで、`http://host_name:port/webapp/PaymentManager` を指定します。ここで `host_name` は、Web Server for Payments を実行しているホスト名で、`port` は Payments が稼働しているポート番号です。
2. WebSphere Commerce サイト管理者 ID を入力します。
3. WebSphere Commerce サイト管理者パスワードを入力します。
4. 「ログイン」をクリックします。

重要: WebSphere Commerce Payments インスタンスが使用している HTTP サーバーが、デフォルト以外のポート番号に対して構成されている場合、このチュ

ートリアル全体で WebSphere Commerce Payments Web アドレス・リンクに、ホスト名に続けてそのポート番号も指定する必要があります。

ユーザー・インターフェースの右上にあるアイコンは、次のようなときに使います。

- 多方向矢印をクリックすると、ページを最新表示します。
- 左矢印をクリックすると、最後にアクセスしたページに戻ります。
- 疑問符をクリックすると、ページのコンテキスト依存オンライン・ヘルプにアクセスします。

ステップ 2: WebSphere Commerce Payments マーチャントの作成とカセットの認可

WebSphere Commerce Payments にログオンしていない場合は、Payments 管理者としてログオンします。これでグローバル表示が可能になり、グローバル権限を持つこととなります。 WebSphere Commerce Payments の構成の最初のステップは、マーチャントを作成し、そのマーチャントによる決済カセットの使用を認可することです。マーチャントを作成してカセットを認可するには、以下のようになります。

1. ナビゲーション・フレームで「**マーチャント設定**」をクリックします。
2. 「**マーチャント設定**」ページで、「**マーチャントの追加**」をクリックします。
3. 「**マーチャント設定**」ページで、以下の情報を入力します (チュートリアルでは、モノスペースのテキストをそれぞれのフィールドに入力する必要があります)。

表 11. マーチャント・フィールドの作成

フィールド名	説明
マーチャント名	Test Store と入力します。これが、マーチャントに割り当てる名前になります。その唯一の機能は、ユーザー・インターフェースに情報を表示することです。
マーチャント番号	123456789 と入力します。これは、すべてのトランザクション・データにおいて、マーチャントを固有に識別するために割り当てる番号です。
許可カセット	OfflineCard のボックスをチェックします。このボックスをチェックすると、マーチャントがこの決済カセットを使用することが認可されます。

4. 「**マーチャントの作成**」をクリックして、マーチャントの構成を保管します。

このカセットの使用を許可する対象のマーチャントをすでに作成している場合は、以下のようになります。

1. 「**マーチャント設定**」をクリックします。
2. 「**マーチャント名**」をクリックします。
3. **OfflineCard** のボックスを選択します。
4. 「**マーチャントの作成**」をクリックします。

これでマーチャントは、このカセットを使用する許可を与えられました。

ステップ 3: WebSphere Commerce Payments ユーザーの定義

このチュートリアルでは、以下のユーザーで作業します。

- インストール中に作成されたデフォルト・ユーザー (詳しくは、「*WebSphere Commerce* インストール・ガイド」を参照)。
- *Pat* (システム管理者が定義するユーザー)

WebSphere Commerce 組織管理コンソールを使用して、ユーザーの定義や管理などのタスクを実行することができます。WebSphere Commerce Payments へのユーザーの定義は、2 つの部分で構成されるプロセスです。たとえば、ユーザー *Pat* を定義するには、WebSphere Commerce 組織管理コンソールを使用して、*Pat* にサイト管理者の役割を割り当てる必要があります。次に、Payments UI ディレクトリー内で、または組織管理コンソールを使って、*Pat* のユーザー役割をマーチャント管理者に割り当てます。アクセスをユーザーに割り当てる前に、マーチャントを作成する必要があります。

Payments ユーザーを構成するには、以下のようになります。

1. Web ブラウザーで `https://host_name:port/adminconsole` を指定します。
2. 「アクセス管理」>「ユーザー」をクリックします。
3. 「新規」をクリックします。
4. 新規ユーザー・ウィザードを使用して、新規ユーザー *Pat* を作成します。
5. 「アクセス」ページで、*Pat* にサイト管理者の役割を割り当てます。
6. 「Payments」>「ユーザー」をクリックします。
7. ユーザー名 *Pat* を入力します。
8. 「検索」をクリックします。
9. *Pat* の名前をクリックします。そのユーザーの構成ページが表示されます。
10. 希望するマーチャント名を選択します。
11. 希望する役割 (ここでは、マーチャント管理者) のラジオ・ボタンを選択します。
12. 「更新」をクリックして、指定した内容を保管します。

WebSphere Commerce Payments を WebSphere Commerce 製品に対してリモート側のシステムにインストールする場合で、WebSphere Commerce スタイルシートを WebSphere Commerce Payments ユーザー・インターフェースで使用する場合は、`PMCustomUI.properties` ファイルを `Payments_install_dir\samples\wcs\PMCustomUI.properties` ディレクトリーから WebSphere Commerce Payments のメイン・インストール・ディレクトリーにコピーする必要があります。

ステップ 4: ユーザー役割の割り当て

以下のいずれかの WebSphere Commerce Payments 役割にユーザーを割り当てる必要があります。これらの役割は、対応する WebSphere Commerce の役割にマッピングされています。

表 12. 役割のマッピング

Payments の役割	WebSphere Commerce の役割
Payments 管理者	サイト管理者
マーチャント管理者	サイト管理者
スーパーバイザー	オペレーションまたはセールス・マネージャー
クラーク	顧客サービス・スーパーバイザー

以下のユーザーを作成します。

- ユーザー *Pat*
- マーチャント *Test Store*

その後、WebSphere Commerce Payments 構成で *Pat* の役割を割り当てることができます。

例外: 非 *WebSphere Commerce Payments* アクセスの役割を割り当てて、WebSphere Commerce Payments へのユーザー・アクセスを拒否することもできます。これは、たとえば、休暇中の従業員などのユーザー・アクセスを一時的に拒否する必要がある場合に役立ちます。WebSphere Commerce Payments 役割の許可についての詳細は、「*WebSphere Commerce Payments* プログラミング・ガイドとリファレンス」の『役割の許可に関する表』を参照してください。

Pat に *Test Store* のマーチャント管理者の役割を割り当てするには、Payments UI で以下のようにします。

1. ナビゲーション・フレームで「ユーザー」をクリックします。
2. 「ユーザー検索」ページで、ユーザー名 *Pat* を入力してから、「検索」をクリックします。
3. 「ユーザー」ページで、ユーザー名 **Pat** をクリックします。
4. 「マーチャント (Merchant)」スクロール・ボックスから、「**Test Store**」を選択します。
5. 「マーチャント管理者 (Merchant Administrator)」のラジオ・ボタンを選択します。
6. 「更新」をクリックして、ユーザー構成を保管します。

この時点で、WebSphere Commerce Payments ユーザー・インターフェースからログオフし、今度はマーチャント管理者 *Pat* としてもう一度ログオンします。

重要: ホストされている環境 (すなわち、Commerce Service Provider (CSP) によって、複数のマーチャントをリモートで管理する WebSphere Commerce Payments が確立されている環境) では、CSP が Payments 管理者であり、マーチャントとプロバイダーとの各契約ごとに役目を果たします。このシナリオでは、マーチャントは CSP (Payments 管理者) によって付与されたマーチャント管理者権限を使い、自分のマーチャント設定を構成することになります。

マーチャント管理者としてログオン

ログオフしてもう一度ログオンするには、以下のようにします。

1. ナビゲーション・フレームで、「**admin をログオフ (Logoff admin)**」をクリックして WebSphere Commerce Payments のメイン・ログオン・ページに戻ります。
2. **ユーザー ID** Pat を入力します。
3. Pat に定義されたパスワードを入力します。
4. 「**OK**」をクリックします。

これ以降、このチュートリアルでは、役割は、Test Store のマーチャント管理者権限を持つユーザー Pat になります。WebSphere Commerce Payments ユーザー・インターフェースのビューは、マーチャント管理機能に制限されます。以前は、Payments 管理者として、マーチャントと Payments 管理者の両方の機能をグローバルに表示することができました。

ステップ 5: アカウムの作成

ここまでで、1 つのマーチャント Test Store を定義し、1 つの決済カセットの、OfflineCard Cassette が使用できるようになりました。マーチャント管理者としての最初の作業では、OfflineCard Cassette のアカウントを作成します。

アカウントとは、マーチャントと、そのマーチャントのトランザクションを処理する金融機関との関係のことです。決済カセットごとに、複数のアカウントを設けることができます。このチュートリアルでは、OfflineCard Cassette 用にアカウントを 1 つ作成します。

アカウントを作成するには、以下のようにします。

1. ナビゲーション・フレームで「**マーチャント設定**」をクリックします。
2. 「マーチャント設定」ページで、「Test Store」の「OfflineCard Cassette」アイコンをクリックします。
3. 「OfflineCard Cassette」ページで、「**アカウント**」をクリックします。
4. 「アカウント」ページで、「**アカウントの追加**」をクリックします。
5. 以下のフィールドに入力します (チュートリアルでは、モノスペースのテキストを入力する必要があります)。

表 13. 「アカウントの追加」フィールド

フィールド名	説明
アカウント名	Test Account と入力します。これが、アカウントに割り当てる名前になります。その唯一の機能は、ユーザー・インターフェースに情報を表示することです。
アカウント番号	11111111 と入力します。これは、すべてのトランザクション・データにおいて、アカウントを固有に識別するために、ホスティング・サービス・プロバイダーまたはマーチャント管理者が割り当てる番号です。
金融機関名	Test Bank と入力します。これは、このアカウントを保持している金融機関の名前です。その唯一の機能は、ユーザー・インターフェースに情報を表示すること。

表 13. 「アカウントの追加」フィールド (続き)

フィールド名	説明
通貨	このアカウントの通貨を選択します。
バッチ・クローズ時刻	午前 0 時を何分すぎたらこのアカウントのバッチの自動クローズをカセットが行えばよいかを指示する分数。ゼロの値は午前 0 時を表します。指定できる最大値は 1439 です。ヌル値を指定すると、自動バッチ・クローズは使用不可になります。

6. 「アカウントの作成」をクリックし、OfflineCard Cassette のアカウントを作成します。

アカウント設定の指定

このチュートリアルを使用するために、カセットに対して作成したばかりのアカウントおよびブランドの特定の基本または拡張設定を指定することは必要ありません。ただし、アカウントには特定のオプションを指定できるということには留意する必要があります。たとえば、マーチャント管理者として、アカウントに決済承認と関連付けた有効期限を持たせるかどうかを定義することができます。

OfflineCard Cassette のアカウント設定を WebSphere Commerce Payments ユーザー・インターフェースから表示するには、以下のようにします。

1. ナビゲーション・フレームで「マーチャント設定」を選択します。
2. 「マーチャント設定」ウィンドウにある「Test Store」ウィンドウで、「OfflineCard Cassette」アイコンをクリックします。
3. 「OfflineCard Cassette」ウィンドウで、「アカウント」をクリックします。
4. 「アカウント」ウィンドウで、「Test Account」を選択します。
5. 「Test Account」ウィンドウで、「アカウント設定」を選択します。
6. 基本または拡張設定を、該当する選択項目をクリックすることによって表示または変更できます。たとえば、「拡張設定」を選択して、自動決済処理オプションの承認有効期限設定を表示します。
7. 該当するオプションを入力し、「更新」を選択してアカウント設定を更新します。(取り消すには、「アカウント設定」ウィンドウの戻り矢印を選択します。)

処理オプションの設定の詳細については、オンライン・ヘルプを参照することができます。ただし、ここではそのうちの 1 つの承認有効期限オプションを考察します。

決済承認が有効である期間を制御する、承認有効期限を設定することが必要な場合があります。承認有効期限を設定すると、承認有効期限が切れた後の資金のデポジットに対して金融機関が課す課金を避けるために役立つことがあります。たとえば、クレジット・カードのオーダーの場合、決済承認は、クレジット・カード発行者が設定した日数だけ続きます。承認の有効期限が切れた後のオーダー決済の資金をデポジットすると、金融機関はその資金のデポジットを拒否するか、またはデポジットを受け入れてさらに課金する場合があります。承認有効期限を WebSphere Commerce Payments で設定することによって、WebSphere Commerce Payments は承認有効期限が切れた資金を資金化しないので、こうした状況を避けることができます。

決済承認が切れてから資金をデポジットしようとする、決済が承認状態ではないことを示すメッセージを受け取ります。 WebSphere Commerce Payments ユーザー・インターフェースでは、金額に 0 を指定して決済を取り消すことができます。決済が取り消されると、承認操作を実行して、決済を承認状態に戻すことができます。

承認有効期限オプションを使用するには、使用するカセットが承認有効期限をサポートしている必要があります。 WebSphere Commerce Payments が提供している OfflineCard、VisaNet、Paymentech、および CustomOffline カセットは、承認有効期限をサポートしています。

重要: 承認有効期限として入力した値は、アカウントに関連付けられたすべてのブランドに適用されます。それぞれのブランドが異なる決済承認有効期限の規則を持っている場合、承認有効期限の値を最低共通水準に設定するか、またはブランドごとに別個のアカウントを作成します。ブランドごとに別個のアカウントを作成する場合、バッチ決済をアカウントごとに別個に清算する必要があります。

オーダーの承認、決済のデポジット、およびバッチの清算の詳細については、この章で後述します。

ステップ 6: ブランドの作成

OfflineCard Cassette を使用するには、まず、ストアと連動するクレジット・カード会社のブランド名を構成する必要があります。これによって、ショッピングは、チェックアウト時にリストから適切なクレジット・カードを選択できるようになります。 OfflineCard Cassette のブランドを作成するには、以下のようにします (このチュートリアルでは、ブランドとして ROBO を使用します)。

1. ナビゲーション・フレームで「マーチャント設定」をクリックします。
2. 「マーチャント設定」ページで、「Test Store」の「OfflineCard Cassette」アイコンをクリックします。
3. 「OfflineCard Cassette」ページで、「アカウント」をクリックします。
4. 「アカウント」ページで、アカウント名「オフライン・アカウント (Offline Account)」をクリックします。
5. 「オフライン・アカウント (Offline Account)」ページで、「ブランド」をクリックします。
6. 「ブランド」ページで、「ブランドの追加」をクリックします。
7. 「ブランド名」フィールドに、ROBO と入力します。
8. 「ブランドの作成」をクリックします。

ステップ 7: Sample Checkout を使用したオーダーの作成

マーチャント管理者として、グローバルなマーチャント権限を持っているので、以下のことを実行できます。

- マーチャント固有の管理機能
- すべての決済処理機能

実際のビジネスのシナリオでは、決済処理作業を、限定された決済処理権限を持つ他のマーチャント定義ユーザー（たとえば、スーパーバイザーやクラーク）に委任できます。このチュートリアルでは、マーチャント管理者がこの作業を実行します。決済処理を開始するときに必要なすべての WebSphere Commerce Payments およびマーチャント管理作業を完了しているため、次に示す作業を開始する準備が整っています。

- オーダーの承認
- 決済のデポジット
- バッチの清算
- クレジットの発行
- 日ごとのバッチ合計の表示

このチュートリアルでは、Sample Checkout ツールを使用して、支払い処理用に 3 件のオーダーを作成します。Sample Checkout ツールにはユーザー・インターフェースがあり、これを使用して、カセットのインプリメンテーションをテストするためのサンプル・オーダーを作成できます。Sample Checkout にアクセスするには、まず、以下のようにしてデフォルトのユーザー ID とパスワードを変更する必要があります。

WebSphere Commerce Payments Sample Checkout にアクセスしてオーダーを作成するには、以下のようにします。

1. 以下のディレクトリーから、

```
WAS_installdir/installedApps/host_name/
payments_instance_Commerce_Payments_App.ear/SampleCheckout.war
```

構成ファイル SampleCheckout.xml を開きます。

2. SampleCheckout エレメントで、以下の属性値を変更します。

```
pmHostName="fully_qualified_host_name "
pmPort="port "
default_userid="wc_userid "
password="wc_password "
```

3. ファイルを保管してください。ブラウザで、
`http://host_name:port/webapp/SampleCheckout` を指定します。ここで `host_name` は、Web Server for Payments を実行しているマシンのホスト名で、`port` は Payments が稼働しているポート番号です。
4. 「Sample Checkout」ページで、以下の情報を入力します（チュートリアルでは、モノスペースのテキストをそれぞれのフィールドに入力する必要があります）。

表 14. Sample Checkout のフィールド

名前	説明
マーチャント番号	OfflineCard Sample Checkout マーチャント番号 987654321 を入力します。
オーダー番号	オーダー番号を表す任意の数字を入力します。
金額	オーダーの合計金額を表す任意の金額を入力します。
通貨	US dollar を入力します。これはこのオーダーを行うときに使う通貨です。

表 14. Sample Checkout のフィールド (続き)

名前	説明
支払メソッド	支払いのタイプとして OfflineCard を選択します。
ブランド	使用する OfflineCard のブランドを選択します。
クレジット・カード番号	4111111111111111 と入力します。
有効期限	ご使用のクレジット・カードの有効期限 (月と年) を選択します。

▶ 400 iSeries システムの場合は、使用するクレジット・カードのブランドを指定する必要があります。

5. 「購入」をクリックします。

決済を処理しなければならないオーダーが 3 つあるので、あと 2 回このステップを繰り返します。

ステップ 8: オーダーの承認

Sample Checkout を使って 3 件のオーダーを作成し終わったら、それらのオーダーを承認できます。次のステップに従ってオーダーを承認します。

1. ブラウザーで、`http://host_name:port/webapp/PaymentManager` を指定し (`host_name` は、Web Server for Payments を実行しているマシンのホスト名で、`port` は Payments が稼働しているポート番号)、Pat でログオンします。
2. ナビゲーション・フレームで、「承認」をクリックします。
3. 「承認」ページで、発行したオーダーのいずれか 1 つのボックスを選択します。
4. 「選択して承認」をクリックします。「承認の結果」ページに、承認要求の状況が表示されます。
5. 承認が完了したら、「承認画面に戻ります」をクリックします。

まだ 2 つのオーダーが、承認を待機しています。「承認」ページで「すべてを承認」をクリックすると、それらのオーダーを全額、同時に承認できます。しかし、承認機能を分かりやすく説明するために、このチュートリアルでは各オーダーを別々に処理する方法について説明します。

「オーダー」ページでのオーダーの承認

このセクションでは、「オーダー」ページからオーダーを承認しますが、オーダー全体の合計の一部だけを承認します。オーダーに関連した商品の一部がオーダー処理の段階で発送できない場合 (商品が在庫切れの場合など) には、オーダーの一部だけを承認する方法が便利です。

1. 「承認」ページで、承認を待機している残りのオーダーのいずれかの、「オーダー番号」をクリックします。
2. 「オーダー番号」ページで、オーダーの詳細を見ることができます。「承認」をクリックして、このオーダーを承認します。

3. 「オーダー番号の承認」ページには、以下のフィールドが表示されます。

表 15. オーダー番号の承認のフィールド

フィールド名	説明
通貨	このオーダーを行うときに使う通貨のタイプ。これは、読み取り専用フィールドです。
オーダー金額	オーダーの合計であり、オーダーを行うときに使う通貨で示されます。これは、読み取り専用フィールドです。
承認済み金額	承認されたオーダーがないため、この読み取り専用フィールドはゼロを表示します。
デPOSIT済み金額	承認されたあるいはデPOSITされた金額がないため、この読み取り専用フィールドはゼロを表示します。
承認金額	これは、オーダーの合計金額です。
許可コード	これは、オフラインの手動許可要求プロセスから戻される許可コードです。決済状態は「承認済み」に変わります。
拒否の理由	これは、オフラインの手動許可プロセスから戻される拒否理由です。決済状態は「拒否」に変わります。
AVS 結果コード	Address Verification System 結果コード。

承認金額を 3.00 に変更します。オプションで、その額を承認する許可理由、または承認を拒否する拒否理由を指定します。

4. このオーダーを 3 ドルで承認するには、「承認」をクリックします。承認処理が完了すると、「オーダー」ページは最新表示され、承認状況が表示されます。

販売機能を使用したオーダーの承認

処理した最後のページの一部を承認したため、「承認」ページには、2 つのオーダー項目が残っています。このステップでは、販売機能を使用し、残りのオーダーを承認します。

販売機能を使用すると、オーダーを承認したり、「承認済み」状態をバイパスして、直接に「デPOSIT済み」状態に持ってゆくことができます。販売機能は、オーダーの支払いを自動的に承認およびデPOSITします。したがって、販売を自動デPOSIT付きの承認と見なすことができます。商品をバイヤーに迅速に発送し、資金を確実に獲得したい場合（たとえば、ダウンロード可能なソフトウェアまたは電子情報を販売する場合）は、この販売機能を使用してください。ただし、販売機能を使うと、承認の許可理由や拒否理由を設定できません。販売機能は、承認とデPOSITを 1 つのトランザクションに組み合わせるため、トランザクションごとに課金される場合にも便利です。

販売機能を使用してオーダーを承認するには、以下のようになります。

1. ナビゲーション・フレームで、「承認」をクリックします。
2. 「承認」ページで、「すべて販売」をクリックします。処理が完了すると、販売するオーダーごとに承認状況が表示されます。
3. 販売が完了したら、「承認画面に戻ります」をクリックします。

ステップ 9: 決済のデポジット

デポジット機能により、オーダーの決済をデポジットすることができます。1つのオーダー番号で、複数の決済を関連付けることができます。同じリストに同じオーダー番号が複数回示されることがあります。その場合、それぞれの回の決済情報は異なります。

決済をデポジットするには、以下のようにします。

1. ナビゲーション・フレームで「**デポジット**」をクリックします。
2. リストされているいずれかの決済のボックスを選択し、「**選択してデポジット**」をクリックします。処理が完了すると、デポジットする決済のデポジット状況が表示されます。
3. デポジットが完了したら、「**「デポジット」画面に戻ります**」をクリックします。

決済の一部をデポジットすることもできます。決済の一部をデポジットするには、以下のようにします。

1. 「デポジット」ページで、デポジットを待機しているいずれかの決済の、「**決済番号**」をクリックします。
2. 「決済」ページで、「**デポジット**」をクリックします。
3. 「オーダー - 決済」ページで、デポジット合計を **2.00** に変更し、「**デポジット**」をクリックします。

ステップ 10: バッチの清算

バッチとは、金融機関によって一単位として処理される、決済とクレジットの集まりのことです。バッチはマーチャントおよびアカウントと関連付けられています。ここでは、前の演習でデポジットした決済をバッチで扱います。金融機関による処理を開始するため、このバッチを**清算**しなければなりません。清算が完了すると、金融機関側が資金の振替を行います。

バッチを清算するには、以下のようにします。

1. ナビゲーション・フレームで、「**バッチの検索**」をクリックします。または、「**清算**」をクリックしてもかまいません。
2. 「バッチの検索」ページで、以下の情報を入力して検索を絞り込むことができます（このチュートリアルでは、これらのフィールドに入力する必要はありません）。

表 16. バッチ検索のフィールド

フィールド名	説明
マーチャント	検索対象のバッチを持つマーチャントの名前。 WebSphere Commerce Payments データベースにあるマーチャントが 500 より少ない場合、マーチャント名をドロップダウン・リストから選択します。 WebSphere Commerce Payments データベースにあるマーチャントが 500 を超えている場合、マーチャントの名前を入力します。
バッチ番号	マーチャント内のバッチを固有に識別する番号。決済をデポジットするときに割り当てられます。

表 16. バッチ検索のフィールド (続き)

フィールド名	説明
状態	バッチの状態 <ul style="list-style-type: none"> • オープン • クローズ
貸借状況	このバッチのバランス状況。 <ul style="list-style-type: none"> • 貸借一致 バッチのバランスが正常である (つまり、総合計が合っている)。 • 貸借不一致 このバッチのバランスが正常ではない (つまり、総合計が合っていない)。
決済タイプ	オーダーするときに使う決済タイプ、またはプロトコルを示します (たとえば、OfflineCard)。
バッチ・オープン日付	検索開始日付および検索終了日付フィールドを使い、指定した時間範囲内にオープンしたバッチを検索します。 <ul style="list-style-type: none"> • 検索開始日付 特定の日付以降を指定して、オープンしているすべてのバッチを検索します。 • 検索終了日付 特定の日付以前を指定して、オープンしているすべてのバッチを検索します。
バッチ・クローズ日付	検索開始日付および検索終了日付フィールドを使い、指定した時間範囲内にクローズしているバッチを検索します。 <ul style="list-style-type: none"> • 検索開始日付 特定の日付以降を指定して、クローズしているすべてのバッチを検索します。 • 検索終了日付 特定の日付以前を指定して、クローズしているすべてのバッチを検索します。
アカウント	このオーダーを処理する対象のアカウント。定義されているアカウントの数が 500 を超えている場合は、入力フィールドにアカウント番号を入力します。

3. 「検索」をクリックします。

ヒント: 検索開始日付および検索終了日付フィールドを使用して特定のバッチを検索から除外し、検索結果を絞り込むこともできます。たとえば、08/01/2003 より前にオープンしたすべてのバッチを検索し、08/15/2003 より後にオープンしたすべてのバッチを検索することにより、08/02/2003 から 08/14/2003 の間にオープンしたバッチを除外することができます。

4. バッチ番号をクリックして、そのバッチについての情報を表示します。
5. このバッチでのすべての決済とクレジットの詳細リストを表示するときは、「バッチ詳細」をクリックします。
6. 「清算」をクリックして、バッチを清算します。処理が完了すると、清算状況が「清算結果」ページに表示されます。

古い情報を枝取りするには、「清算結果」ページの「削除」をクリックして、清算されたバッチを削除できます。バッチを削除すると、そのバッチに関連する情報、すなわち、決済、クレジット、およびカセット固有のデータもすべて削除されます。決済データすべてを保持しておくのであれば (たとえば、監査のために)、バッチを削除しないでください。

ステップ 11: クレジットの発行

クレジットは、オーダーに対して発行され、オーダーの合計金額まで任意の金額で付与できます。

クレジットを発行するには、以下のようにします。

1. クレジットを発行するオーダーを検索するには、ナビゲーション・フレームで「**オーダーの検索**」をクリックします。
2. 「**オーダーの検索**」ページで、以下の情報を入力できます (このチュートリアルでは、これらのフィールドに入力する必要はありません)。

表 17. オーダー検索のフィールド

フィールド名	説明
マーチャント	検索対象のオーダーを持つマーチャントの名前。 WebSphere Commerce Payments データベースにあるマーチャントが 500 より少ない場合、マーチャント名をドロップダウン・リストから選択します。 WebSphere Commerce Payments データベースにあるマーチャントが 500 を超えている場合、マーチャントの名前を入力します。
オーダー番号	マーチャントによって割り当てられる番号で、オーダーを固有に識別します。
状態	オーダーの状態 <ul style="list-style-type: none">• 要求済み• オーダー済み• 払い戻し可能• キャンセル• クローズ
決済タイプ	オーダーするときを使う決済タイプ、またはプロトコルを示します (たとえば、OfflineCard)。
オーダー日付	次の検索開始日付および検索終了日付フィールドを使い、指定した時間範囲内にオープンしたオーダーを検索します。 <ul style="list-style-type: none">• 検索開始日付 特定の日付以降を指定して、オープンしているすべてのオーダーを検索します。• 検索終了日付 特定の日付以前を指定して、オープンしているすべてのオーダーを検索します。
オーダー金額	<ul style="list-style-type: none">• 通貨 このオーダーを行うときに使う通貨。ドロップダウン・リストから、通貨のタイプを選択します。• 検索金額の下限 すべてのオーダーを検索するときの値を指定しますが、指定する値以上のオーダー数が示されます。• 検索金額の上限 すべてのオーダーを検索するときの値を指定しますが、指定する値以下のオーダー数が示されます。
アカウント	このオーダーを処理する対象のアカウント。定義されているアカウントの数が 500 を超えている場合は、入力フィールドにアカウント番号を入力します。

3. 「**検索**」をクリックします。

- 「オーダー検索結果」ページで、「払い戻し可能」状態のオーダーのオーダー番号をクリックし、そのオーダーの詳細を表示します。
- 「オーダー」ページで「クレジット」をクリックし、このオーダーに対するクレジットを作成します。
- 「クレジットの作成」ページに、以下の情報が表示されます。

表 18. クレジットの作成のフィールド

フィールド名	説明
通貨	このオーダーを行うときに使う通貨のタイプ。これは、読み取り専用フィールドです。
オーダー金額	オーダーの合計であり、オーダーを行うときに使う通貨で示されます。これは、読み取り専用フィールドです。
承認済み金額	承認されたオーダー金額であり、オーダーを行うときに使う通貨で示されます。これは、読み取り専用フィールドです。
デPOSIT済み金額	デPOSITされたオーダー金額であり、オーダーを行うときに使う通貨で示されます。これは、読み取り専用フィールドです。
クレジット金額	このフィールドは、マーチャント管理者によって、ショッパーにクレジットされる合計金額が記入されます。
許可の理由	これは、オフラインの手動許可要求プロセスから戻される許可コードです。クレジット状態は「リファンド済み」に変わります。
拒否の理由	これは、オフラインの手動許可プロセスから戻される拒否理由です。クレジット状態は「拒否」に変わります。

クレジット金額を入力して、「クレジット」をクリックします。

クレジット処理が完了すると、「オーダー」ページが最新表示され、クレジット状況が表示されます。新しく作成されたクレジットが、「クレジット」の下に表示されます。

ステップ 12: バッチ合計の表示

このチュートリアル最後のステップは、日ごとのバッチ合計を表示することです。WebSphere Commerce Payments レポート機能を使うと、クローズされた状態で、バッチの日ごとの合計を表示できます。

日ごとのバッチ合計レポートを生成するには、以下のようにします。

- ナビゲーション・フレームで、「レポート」をクリックします。
- 「レポート」ページで、「日次バッチ合計」をクリックします。
- 「バッチ合計レポート」ページで、バッチ合計レポートを得る日付を入力します。現在の日付のレポートを生成する場合は、フィールドをブランクのままにしておいてください。
- マーチャント名を入力するか選択します。マーチャント名を入力しない場合、指定した日付に基づくすべてのバッチのリストが表示されます。バッチの数が 500 個を超える場合、最初の 500 個のバッチだけが表示されます。
- 「検索」をクリックします。

「日次バッチ合計」レポートは、「**検索**」ページで指定した日付に終了したすべてのバッチの合計を計算します。検索日付を指定しない場合、生成されるレポートには現在の日付のバッチ合計が入ります。これらの合計は通貨単位で計算され、それぞれの通貨が 1 行ずつ示されます。これらの合計は、OfflineCard Cassette によってなされるものだけでなく、すべての決済タイプに対してなされるすべての決済とクレジットが含まれることに注意してください。

以上で、Payments 管理者およびマーチャント管理者の 1 日を終了しました。個々のビジネス・モデルはさまざまですが、このチュートリアルでは、実際に機能する WebSphere Commerce Payments を確立するための基本的な道筋を強調し、WebSphere Commerce Payments で実現される基本的な決済処理を紹介しました。WebSphere Commerce Payments ユーザー・インターフェースの特定の分野についての詳細は、オンライン・ヘルプを参照してください。

付録 D. WebSphere Commerce Payments の各国語 (NLS) サポート情報

この章では、WebSphere Commerce Payments 用の各国語サポート (NLS) に関する補足情報を提供しています。WebSphere Commerce Payments の各国語版 (NLV) をインストールする場合は、この付録にある情報を検討する必要があります。

NLS のヒント

Adobe Acrobat Reader の制約事項

AIX プラットフォームでは、いくつかの言語で PDF ファイルが正しく表示されない場合があります。WebSphere Commerce Payments PDF ファイルの表示や印刷ができない場合は、Windows® または Windows 2000 オペレーティング・システムのブラウザから表示および印刷を行うことができます。これが AIX Adobe Acrobat Reader の制約事項です。

コード・ページ

スペイン語で文字が正しく表示されない場合は、コード・ページ ISO-8859-1 を使用していることを確認してください。

WebSphere Commerce Payments ユーザー名

WebSphere Commerce Payments は、次の表にある文字で構成されるユーザー名のみをサポートします。この表にある文字セット以外の文字を含むユーザー名は、使用できません。

この表では、WebSphere Commerce Payments ユーザー名を割り当てるためにサポートされている文字セットをリストしています。

表 19. WebSphere Commerce Payments ユーザー名およびその他で使用する Portable Character Set (PCS)

ASCII 16 進値	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
文字	スペース 文字 (空白)	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
ASCII 16 進値	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
文字	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
ASCII 16 進値	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
文字	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
ASCII 16 進値	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
文字	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	—
ASCII 16 進値	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F

表 19. WebSphere Commerce Payments ユーザー名およびその他で使用する Portable Character Set (PCS) (続き)

文字	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
ASCII 16 進値	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	
文字	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

サポートされる言語とロケール

以下の表は、WebSphere Commerce Payments がサポートする言語および対応するロケールをリストしています。

表 20. WebSphere Commerce Payments でサポートされる言語とロケール

言語	ロケール
ブラジル・ポルトガル語	pt_BR
英語	en_US
フランス語	fr_FR
ドイツ語	de_DE
イタリア語	it_IT
日本語	Ja_JP
韓国語	ko_KR
中国語 (簡体字)	zh_CN
スペイン語	es_ES
中国語 (繁体字)	Zh_TW

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あり、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。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

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。

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

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すが、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。

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。

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。

この文書には、他社のインターネット・サイトへの参照を含む、他社製品の情報が含まれている場合があります。IBM は、そのような情報の正確性、完全性、または使用については何ら責任を負いません。

この製品は、SET プロトコルに基づいています。

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。お客様は、IBM のアプリケーション・プログラミング・インターフェース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。

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。

(C) (お客様の会社名) (年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 (C) Copyright IBM Corp. 1996, 2003. All rights reserved.

商標

IBM ロゴ および以下は、IBM Corporation の商標です。

- AIX
- CrossWorlds
- DB2 Extenders™
- DB2 Universal Database™
- IBM iSeries
- Intelligent Miner™
- Lotus®
- OS/400
- pSeries™
- S/390
- VisualAge®
- WebSphere
- xSeries®
- zSeries
- Redbooks
- SecureWay
- DFS
- VisualAge
- DB2
- AS/400

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. 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。

Microsoft、Windows 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。

SET および SET ロゴは、SET Secure Electronic Transaction LLC の商標です。

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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



Printed in Japan