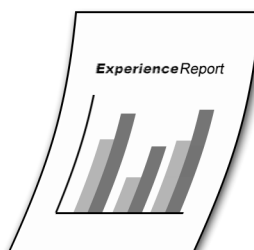


iSeries



iDrink と Cola 接続 SMB シナリオ概要

# Experience Report





iSeries



## **iDrink と Cola 接続 SMB シナリオ概要**

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： iSeries  
iDrink and Cola Connections SMB scenario overview  
Experience Report

発 行： 日本アイ・ビー・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 2005.8

この文書では、平成明朝体™W3、平成明朝体™W7、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™W7、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 2005. All rights reserved.

© Copyright IBM Japan 2005

---

# 目次

## 第 1 章 iDrink と Cola 接続 SMB シナ

リオ概要 . . . . .	1
iDrink の概要 . . . . .	1
Cola Connections の概要 . . . . .	2
環境の概要 . . . . .	3
SMB ディスカバリー . . . . .	5

## 第 2 章 WebSphere Application Server

- Express 環境 . . . . .	7
WebSphere <sup>(R)</sup> Application Server の概要 . . . . .	8
アプリケーションのフロー . . . . .	8
開発環境 . . . . .	10
アプリケーションの詳細 . . . . .	10
アプリケーション設計のポイント . . . . .	11
UserServlet . . . . .	16

CustomerServlet . . . . .	17
EmployeeServlet . . . . .	20
SupplierServlet . . . . .	30
エンタープライズ・アプリケーションのインストー ル . . . . .	33
WebSphere ディスカバリー . . . . .	35

## 第 3 章 Lotus Domino 環境 . . . . . 37

OS/400 LDAP を使用する Lotus Domino の構成 . . . . .	37
アプリケーションの詳細 . . . . .	38
アプリケーション設計のポイント . . . . .	38
アプリケーションのセットアップ . . . . .	39
Lotus Domino のフォームおよびビュー . . . . .	40
Lotus Domino エージェント . . . . .	40
Lotus Domino ディスカバリー . . . . .	41



---

## 第 1 章 iDrink と Cola 接続 SMB シナリオ概要

この報告書は、IBM<sup>(R)</sup> eServer<sup>(TM)</sup> i5 カスタマー・ソリューション・テスト・チームの、WebSphere<sup>(R)</sup> Application Server - Express および Lotus<sup>(R)</sup> Domino<sup>(R)</sup> を使用した、中小および中堅規模ビジネス (SMB) シナリオをインプリメントする経験について文書化しています。このシナリオでは WebSphere Application Server - Express for iSeries<sup>(TM)</sup> を使用して動的 Web ページ、データベース・アクセス、およびディレクトリー・サーバー・アクセスを介した初期の Web 環境を確立します。Lotus Domino アプリケーションは、iDrink 従業員に内部用の動的 Web ページを提供します。

iDrink SMB シナリオでは、飲料流通会社をシミュレートしています。この架空の会社には、200 人未満の従業員がいて、中小規模のビジネスと見なされています。飲料流通業者は、販売する品目の仕入れ先として飲料製造業者と提携しています。iDrink は、さまざまなストアで販売するために大量の飲料を購入する顧客と提携します。従業員は、ビジネスの成功させるために、サプライヤーと顧客の両方と共に仕事をします。iDrink 会社は、Java<sup>(TM)</sup> サーブレットおよび JavaServer Pages (JSP) を介して、顧客、サプライヤー、および製品に関する情報を処理します。ほとんどの顧客、サプライヤー、および製品情報は、DB2 Universal Database<sup>(TM)</sup> に格納され、従業員の情報とパスワード、顧客のパスワード、およびサプライヤーのパスワードは、Lightweight Directory Access Protocol (LDAP) ディレクトリーに保持されています。

さらに、iDrink 従業員は、分類別広告の Web サイト (Cola Connections) へのアクセス権を持っており、従業員はそれを使用して販売品目を広告することができます。このアプリケーションは、Lotus Domino にインプリメントされます。

---

### iDrink の概要

iDrink 社では、従業員、顧客、およびサプライヤーが、それぞれの役割に適した情報にアクセスできるようにしています。

#### シナリオ設計の決定

以下は、最初に行われたシナリオ設計の決定事項です。

- アプリケーション開発には、顧客、従業員、およびサプライヤーの Web 環境をインターフェースにすばやく確立するための手段が用意された WebSphere<sup>(R)</sup> Application Server - Express を使用する。
- WebSphere アプリケーションの認証のために eServer<sup>(TM)</sup> i5 LDAP ディレクトリーを使用して、セキュリティを確保する。

#### iDrink アプリケーション

iDrink アプリケーションは、JavaServer Pages、JavaBeans<sup>(TM)</sup>、および Java<sup>(TM)</sup> サーブレットを使用した WebSphere Application Server - Express 環境で構成され、顧客、従業員、およびサプライヤーは以下の機能を実行することができます。

- 顧客
  - 登録またはログイン
  - 製品情報の表示
  - 製品の注文
  - 顧客情報の更新

- 注文履歴の表示
- 従業員
  - サプライヤー情報の更新
  - サプライヤーからの商品の注文
  - 商品価格の変更
  - 顧客注文の追跡
  - 製品インベントリーの追跡
  - 機密広告の作成、更新、および表示
- サプライヤー
  - 製品情報の更新

図 1 は、iDrink アプリケーションのフローを示します。

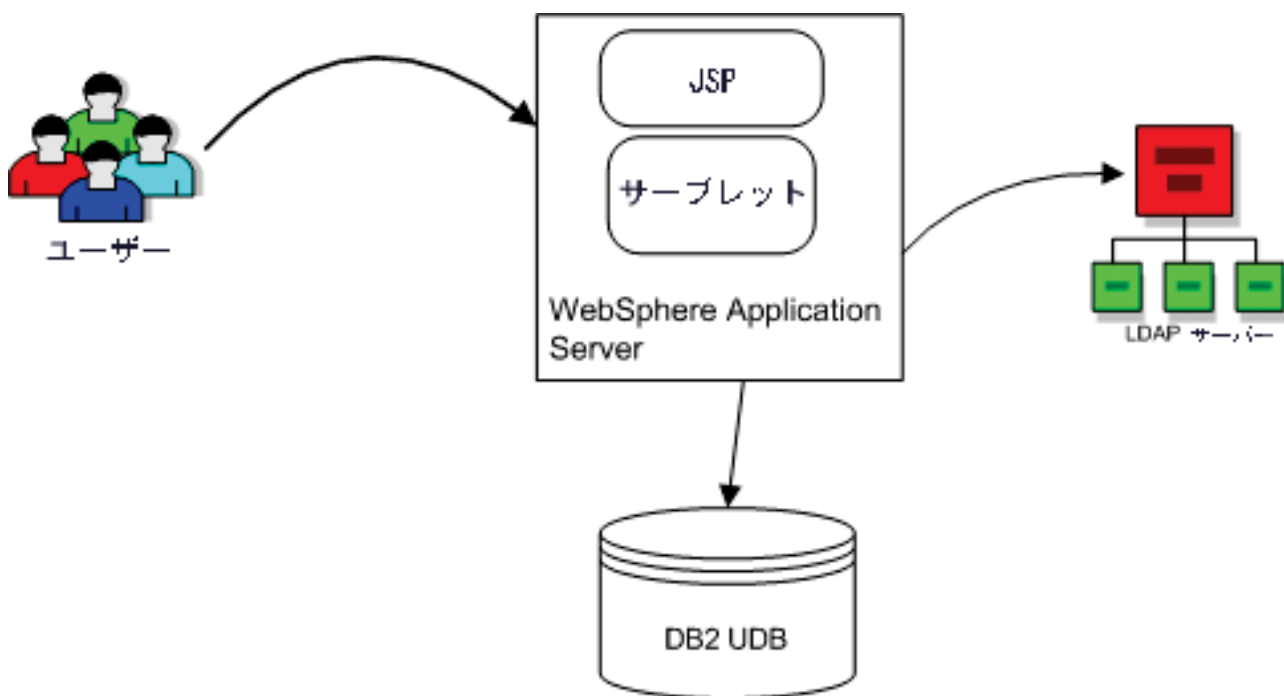


図 1: iDrink アプリケーションのフロー

## Cola Connections の概要

Cola 接続アプリケーションは、従業員に、他の Drink の従業員に機密広告をリストするロケーションを提供します。

### アプリケーション設計の決定

以下は、最初に行われた設計の決定の説明です。

- アプリケーション開発には、すばやく Web 環境を確立するための手段が用意された Lotus<sup>(R)</sup> Domino<sup>(R)</sup> を使用する。



- Lotus Domino アプリケーションの認証のために eServer<sup>(TM)</sup> i5 LDAP ディレクトリーを使用して、セキュリティを確保する。

### Cola 接続アプリケーション

Cola 接続アプリケーションは、Lotus Domino データベース、エージェント、ビュー、およびフォームで構成されており、iDrink 従業員に、機密広告を作成、更新、または表示する機能を提供します。図 2 は、Cola 接続アプリケーションのフローを示したものです。

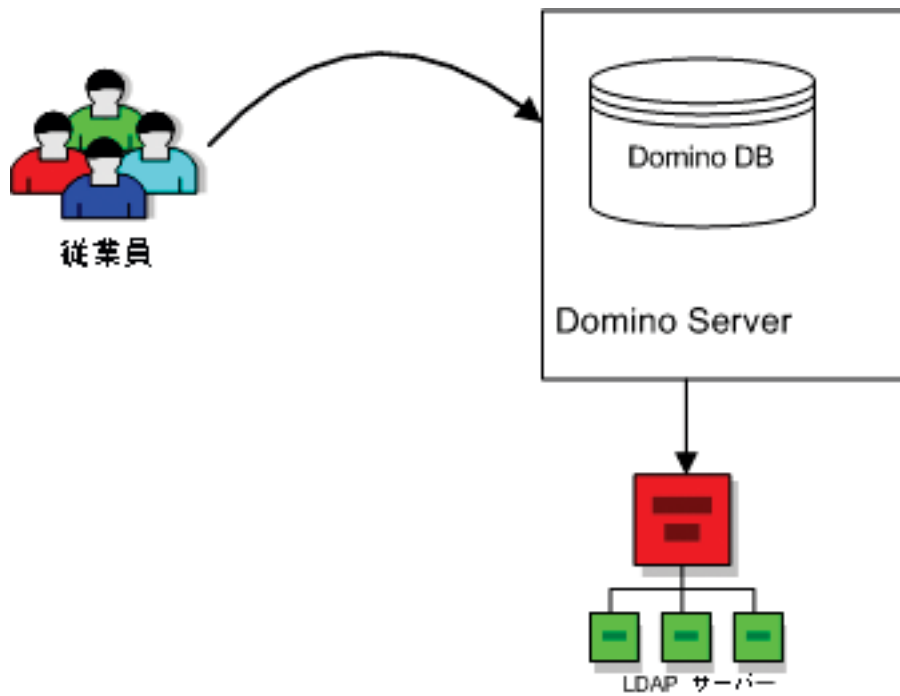


図 2: Cola 接続アプリケーションのフロー

### Domino Directory Assistance を使用したセキュリティ

Domino Directory は、アプリケーションおよびデータ・アクセスの安全性を確保するためのサポートを提供します。セキュリティ・メカニズムは、ユーザー ID を判別および検証し、指定されたユーザーのみに保護リソースへのアクセスを許可します。Cola Connections では、従業員のみが機密広告にアクセスできます。Domino は、各データベースのアクセス制御リスト (ACL) にエントリーを格納するアクセス制御モデルを使用します。セキュリティは、既存の i5/OS<sup>(TM)</sup> LDAP ディレクトリーを使用するように Domino Directory Assistance を構成することによって実装されます。セキュリティのセットアップの詳細については、37 ページの『第 3 章 Lotus Domino 環境』セクションを参照してください。

---

## 環境の概要

iDrink シナリオの SMB 環境のセットアップについては、以下のランタイム環境、システム・ハードウェア、および使用された主な製品のトピックで説明しています。

### ランタイム環境

iDrink 会社は、中小規模のビジネスと考えられ、目的の要求に見合った設計の手ごろなエントリー・レベルの eServer<sup>(TM)</sup> i5 に投資しました。この小型サーバーは、セキュリティーおよび可用性、ビジネス・アプリケーションおよび e-business ソリューションを実行するための能力を提供しながら、iDrink が多くのアプリケーションを同時に実行できるようにします。

## システム・ハードウェア

このシナリオの実施では、1 つの eServer i5 システムが使用されます。この eServer i5 システムは、1Way プロセッサ、1 GB のメモリー、i5/OS<sup>(TM)</sup> V5R3 を実行する 175 GB の主ストレージを含みます。

## 主な製品

このシナリオで使用される主なソフトウェア製品は以下のとおりです。

- **WebSphere<sup>(R)</sup> Application Server - Express for iSeries<sup>(TM)</sup>** は、しっかりと統合された開発ツールであり、動的 Web サイトを作成する会社に対して、e-business への手ごろな価格のエントリー・ポイントとなるアプリケーション・サーバーです。WebSphere Application Server - Express は、JavaServer Pages (JSP)、Java<sup>(TM)</sup> サンプルおよび Web サービスの仕様をサポートします。WebSphere Application Server - Express では、Web サービスを作成および使用する能力を提供しつつ、データベースの情報にアクセスしたり、簡単な更新を実行することにより、静的および動的 Web サイトの構築が可能になります。低価格で、使いやすく、すぐに使用可能なソリューションであるため、中間市場の会社の必要性に見合っています。

IBM<sup>(R)</sup> WebSphere Application Server - Express for iSeries の詳細については、Web の <http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/indexexp51.html> を参照してください。

- **Lotus<sup>(R)</sup> Domino<sup>(R)</sup>** は、コラボレーションと e-business の基礎を提供し、企業内のメッセージングから Web ベースのトランザクションおよび両者間のすべてのソリューションを実現していきます。このエンタープライズ・クラスのメッセージングおよびコラボレーション・システムは、個人、チーム、および拡張コミュニティーの経験と知識を開放することによって、生産性を最大化するように構築されています。

Lotus Domino for iSeries の詳細については、Web の <http://www.ibm.com/servers/eserver/iseries/domino/> を参照してください。

- **Lightweight Directory Access Protocol (LDAP)** は、TCP/IP を使用するディレクトリー・サービス・プロトコルです。LDAP ディレクトリー・サービスは、クライアント/サーバー・モデルに準じています。1 つ以上の LDAP サーバーに、ディレクトリー・データが含まれています。LDAP クライアントは、LDAP サーバーに接続し、要求を行います。サーバーは、返信または別の LDAP サーバーへのポインター (参照) によって応答します。LDAP は、データベースというよりもディレクトリー・サービスであるため、LDAP ディレクトリー内の情報は、通常、記述可能であり、属性ベースの情報です。一般的に、LDAP では、ディレクトリー内の情報の変更頻度よりも読み取り頻度の方が高くなります。更新は、通常、全か無かの単純な変更です。LDAP ディレクトリーの一般的な使用方法には、以下のものがあります。

- オンライン電話帳
- E メール・ディレクトリー
- 許可
- 認証

LDAP についての詳細は、Web の <http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm> を参照してください。

- **Apache で稼働する IBM HTTP Server for iSeries** は、素早く簡単に Web を開設し、ビジネスで Web を活用できるようにするためのツールを提供する Web サーバーです。

Apache で稼働する HTTP Server for iSeries についての詳細は、Web の <http://www.ibm.com/servers/eserver/series/software/http/> を参照してください。

- **DB2 Universal Database<sup>(TM)</sup> (UDB) for iSeries** は、IBM の eServer i5 のオンデマンド機能を活用する高機能 64 ビットのリレーショナル・データベース管理システムです。IBM の最先端の製品群である DB2<sup>(R)</sup> 製品の 1 つ DB2 UDB for iSeries は、固有のオートノミック・コンピューティング (自己管理) 機能によって、さまざまなアプリケーションおよび開発環境を低コストでサポートします。

DB2 UDB についての詳細は、Web の <http://www.ibm.com/servers/eserver/series/db2> を参照してください。

---

## SMB ディスカバリー

以下は、iDrink SMB ソリューションの作成中に明らかになった主なディスカバリーのリストです。

- サブレットおよび JSP 間のデータ通信のセッション・データに情報を格納する場合は、格納するオブジェクトをできるだけ小さくしてください。セッションに格納するオブジェクトは、Java<sup>(TM)</sup> ヒープの一部になり、メモリーが消費されます。メモリーの少ない小型のコンピューターでは、オブジェクトを格納する余地がありません。システムのメモリー割り振りを超える場合は、メモリーの内容をディスクにページングするため、システム・パフォーマンスが全体的に低下します。
- サイトで予想される要求数を処理するため、WebSphere<sup>(R)</sup> データ・ソースの接続プールのサイズを増やしてください。また、経過タイムアウト値を設定して、接続が使用されない場合には接続を廃棄するようにして、メモリー使用率を低く保ってください。WebSphere 管理コンソールでこれらの値を設定するには、次のようにします。
  - 左のフレームにある「リソース (Resources)」の横のプラス記号を展開して、「JDBC プロバイダー (JDBC Providers)」リンクをクリックする。メインフレームで、「サーバー有効範囲 (Server Scope)」ラジオ・ボタンを選択し、「適用 (Apply)」ボタンを押します。
  - メインフレームで、アプリケーションが使用する JDBC プロバイダー・リンクをクリックする。追加プロパティの下に JDBC プロバイダーのページの下部に進んで、「データ・ソース (Data Sources)」リンクを押します。
  - メインフレームで、アプリケーションが使用するデータ・ソース・リンクをクリックする。データ・ソース・ページの下部に進んで、「接続のプール (Connection Pool)」リンクをクリックします。
  - 接続のプールのページで、最小、最大接続数と、経過タイムアウトを設定する。
  - 「適用 (Apply)」ボタンをクリックしてから、画面の左上隅の「保存」リンクをクリックする。次のページで、「保存」ボタンをクリックします。
  - WebSphere Application Server を再始動する。
- WebSphere Application Server インスタンス内の Web コンテナで要求を処理するためのスレッド数を、ユーザーの最大数に合わせて増やしてください。また、最大スレッド・サイズを超えたスレッドの割り振りを許可するようにしてください。WebSphere 管理コンソールでこれらの値を設定するには、次のようにします。
  - 左のフレームにある「サーバー (Servers)」の横のプラス記号を展開して、「アプリケーション・サーバー (Application Servers)」リンクをクリックする。メインフレームで、アプリケーションが実行されているアプリケーション・サーバーのリンクをクリックします。

- アプリケーション・サーバーのページで、「**Web コンテナ (Web Container)**」リンクをクリックする。
- Web コンテナのページで、「**スレッド・プール (Thread Pool)**」リンクをクリックする。
- このページで、最小および最大サイズを設定して、「**拡張可能 (Is Growable)**」ラジオ・ボックスをクリックする。
- 「**適用 (Apply)**」ボタンをクリックしてから、画面の左上隅の「**保存**」リンクをクリックする。次のページで、「**保存**」ボタンをクリックします。
- WebSphere Application Server を再始動する。
- HTTP サーバーで要求を処理するためのスレッド数をユーザーの最大数に合わせて増やしてください。これらの値を設定するには、IBM<sup>(R)</sup> タスク・ページで次のようにします。
  - 「**管理 (Manage)**」タブをクリックし、「**HTTP サーバー (HTTP Servers)**」タブをクリックする。
  - ドロップダウン・ボックスから WebSphere Application Server に関連付けられた HTTP サーバーを選択する。
  - 左のフレームにある「**システム・リソース (System Resources)**」リンクを選択する。
  - メインフレームで、「**拡張**」タブを選択する。
  - 要求を処理するためのスレッド数を設定する。
  - 「**適用 (Apply)**」ボタンを押してから、「**OK**」ボタンを押す。
  - HTTP サーバーを再始動する。
- システム規模の TCP 送信バッファおよび受信バッファのサイズを、パーティションで実行するアプリケーションの性質に応じて増やしてください。これらの設定を変更するには、5250 セッションで次のようにします。
  - **CHGTCPA**
  - TCP 受信バッファおよび送信バッファのサイズを変更する。
  - **Enter** を押す。

---

## 第 2 章 WebSphere Application Server - Express 環境

IBM<sup>(R)</sup> WebSphere Application Server - Express for iSeries<sup>(TM)</sup> バージョン 5.1 は、開発ツールおよび Web ベース・アプリケーションの統合パッケージを提供するアプリケーション・サーバーが両方組み合わさったものです。 WebSphere Application Server - Express には、以下のものが含まれます。

- アプリケーション・サーバーには以下のものが含まれています。
  - サブレット 2.3 および JSP 1.2 サポート
  - 組み込み Web サーバー
  - Web コンテナ
  - Web サービス・サポート
  - XML および XSL サポート
  - JDBC 2.0 サポート
  - 接続のプール
  - Simple WebSphere Authentication Method (SWAM)
  - サブレット 2.3 および JSP 1.2 サポート
- IBM WebSphere Development Studio Client for iSeries バージョン 5.1 開発ツールには、以下のものが含まれています。
  - サブレット 2.3 仕様
  - JSP 1.2 仕様
  - HTML
  - JavaScript<sup>(TM)</sup> (クライアント・サイドおよびサーバー・サイド)
  - 動的 HTML (DHTML)
  - XML および XHTML
  - Web サービスの使用および作成
  - CVS を使用したチーム開発
  - JDBC 2.0
  - リモート・サーバーの構成および操作のサポート
  - 顧客タグ・ライブラリー
  - Struts

iDrink アプリケーションは、Enterprise Archive (EAR) ファイルに含まれています。 EAR ファイルは、J2EE アプリケーションを含む圧縮 JAR ファイルです。 J2EE アプリケーションは、サブレット、JSP、JavaBeans<sup>(TM)</sup>、および XML 構成ファイルなどのエレメントを含んでいます。 EAR ファイルは、WebSphere Development Studio Client、WebSphere Studio Application Developer、または WebSphere Studio Application Site Developer を使用して作成することができます。

---

## WebSphere<sup>(R)</sup> Application Server の概要

WebSphere Application Server - Express 環境は、eServer<sup>(TM)</sup> i5 カスタマー・ソリューション・テストの iDrink シナリオの動作中にセットアップされました。目標は、Web 環境を速やかに確立し、iDrink 社が、必要なビジネス機能を実行するためのインターフェースを顧客やサプライヤーに提供できるようにすることでした。iDrink 社はさらに、WebSphere Application Server - Express 環境を使用して、従業員が作業を実行するためのインターフェースを提供します。iDrink インターフェースは、JavaServer Pages (JSP)、JavaBeans<sup>(TM)</sup>、およびサーブレットを使用して構築されました。

---

## アプリケーションのフロー

iDrink アプリケーション内部では、多くの JSP が使用されています。これらの JSP は、ユーザー、顧客、およびサプライヤーのセクションでリストおよび説明されています。

次の図は、iDrink アプリケーション内部のハイレベル・フローを示しています。

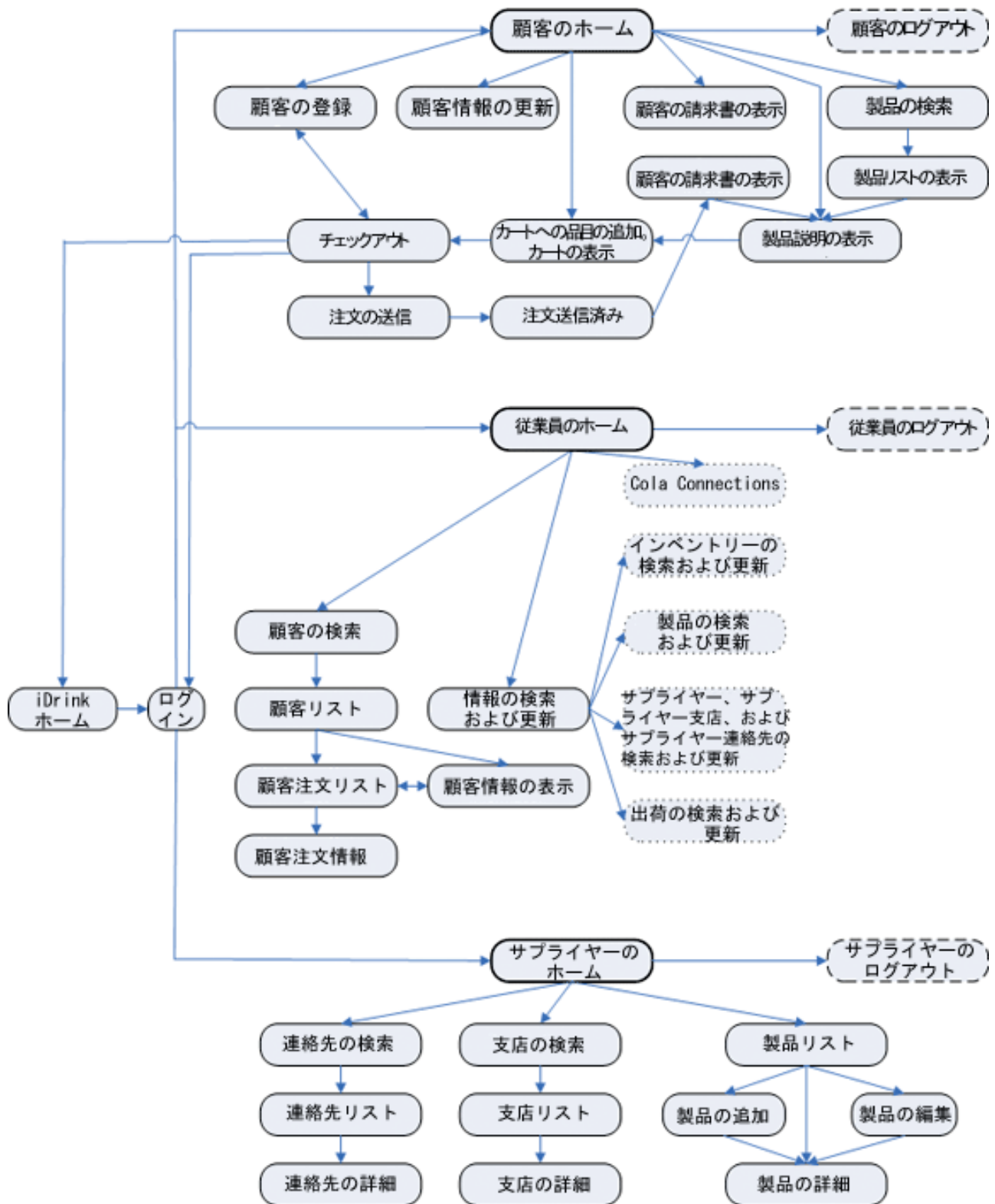


図 3: iDrink アプリケーションのフロー



---

## 開発環境

iDrink チームは、IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Studio Application Developer (WSAD) バージョン 5.0 を選択して、iDrink アプリケーションを形成しているサーブレット、JavaServer Pages (JSP) および JavaBeans<sup>(TM)</sup> を開発しました。WSAD は、開発者が実動システムにデプロイメントする Java<sup>(TM)</sup> 2 Platform, Enterprise Edition (J2EE) アプリケーションを簡単に作成、コンパイル、およびパッケージすることを可能にする Eclipse ベースの開発環境を提供します。一度コード化およびコンパイルされたアプリケーションは、WSAD のテスト環境で実行したり、IBM WebSphere Application Server インスタンスにデプロイして WebSphere 管理コンソールを介して構成したりできます。Web アプリケーションの開発に使用可能なもう 1 つのツールとして、WebSphere Development Studio Client (WDSC) があります。

---

## アプリケーションの詳細

iDrink アプリケーションは、Java<sup>(TM)</sup> サーブレット、JavaBeans<sup>(TM)</sup> および JavaServer Pages (JSP) から構成されるエンタープライズ・アプリケーションです。iDrink アプリケーションには、4 つのサーブレットがあります。これらのサーブレットは、UserServlet、CustomerServlet、EmployeeServlet および SupplierServlet です。UserServlet は、すべてのユーザーのすべてのログイン、ログアウト、およびパスワードの変更要求の処理を行います。CustomerServlet は、顧客および iDrink にログインしていないすべてのユーザーからのすべての要求の処理を行います。EmployeeServlet は、ログインした iDrink 従業員からのすべての要求を処理し、SupplierServlet はログインした iDrink サプライヤーからのすべての要求を処理します。

各 iDrink サーブレットは、ユーザーから要求を受信すると、その要求の内容を解析し、取るべき処置を判別します。例えば、サーブレットが Lightweight Directory Access Protocol (LDAP) のエントリーやデータベース・ファイル内のレコードにアクセスしたり、それらを作成したりする必要がある場合があります。データが取得されると、該当する JavaBean が作成され、そのデータは JavaBean に格納されます。次に、JavaBean は要求オブジェクトまたはセッション・オブジェクトに追加され、該当する JSP が表示されます。図 4 は、これらのコンポーネントがどのように協働しているかを示したものです。



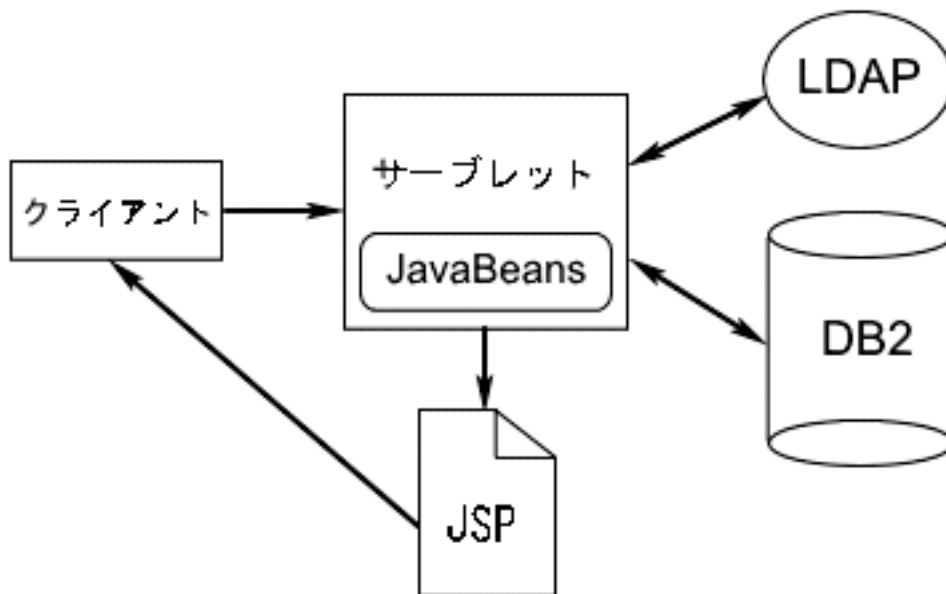


図 4: iDrink アプリケーションのコンポーネント

このセクションには、UserServlet、CustomerServlet、EmployeeServlet および SupplierServlet のメソッド、およびこれらのサーブレットによって呼び出される JSP の詳細情報が含まれます。また、このセクションは、iDrink などのエンタープライズ・アプリケーションを IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Application Server - Express サーバーにインストールする手順と共に、サーブレットのアプリケーション設計のポイントの説明も含まれています。

## アプリケーション設計のポイント

アプリケーションを設計する前に、LDAP サーバーおよびデータベース・スキーマがどのように設計されているかを確認します。このセクションには、これらの考慮事項に関する情報と、iDrink シナリオ用の JavaBeans<sup>(TM)</sup> および Java<sup>(TM)</sup> サーブレットの設計に関する情報が掲載されています。

### Lightweight Directory Access Protocol (LDAP) 設計

LDAP ディレクトリーは、iDrink および Cola 接続アプリケーション内で認証メカニズムとして使用されます。図 5 は、LDAP ディレクトリーの構成方法を示しています。

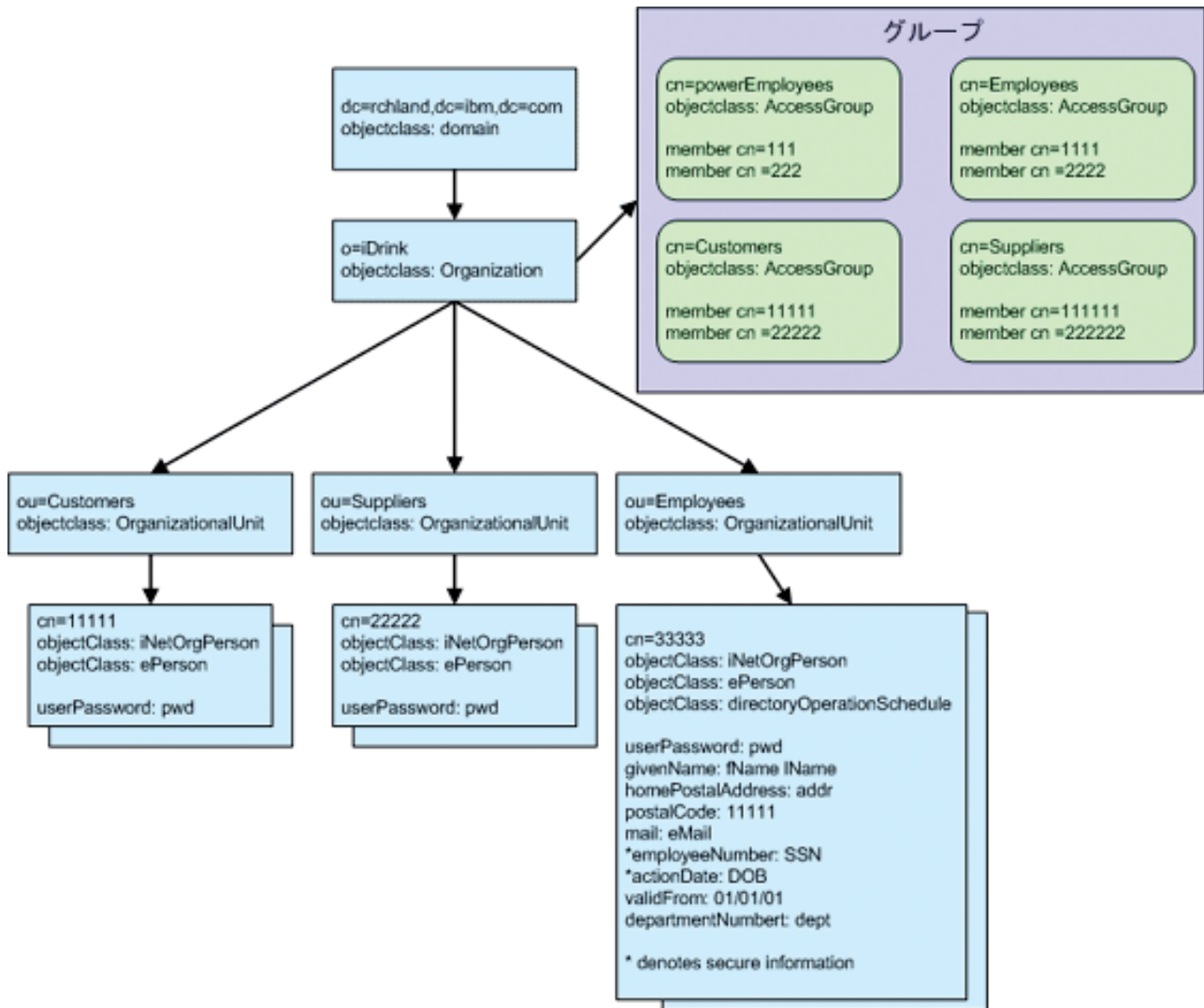


図 5: LDAP ディレクトリー

LDAP ディレクトリーの構成時に、オブジェクト・クラスおよび ACL に関する以下の選択項目が使用されます。

- 顧客およびサプライヤーは、オブジェクト・クラス `inetOrgPerson` および `ePerson` を使用する。すべての顧客は、`cn=customers` と呼ばれるグループのメンバーであり、すべてのサプライヤーは `cn=suppliers` と呼ばれるグループのメンバーです。
- 従業員は、オブジェクト・クラス `inetOrgPerson` および `ePerson` を使用する。すべての従業員は、`cn=employees` と呼ばれるグループのメンバーです。一部の従業員は、`cn=powerEmployees` と呼ばれるグループのメンバーです。powerEmployees グループは、通常の従業員には許可されていない、LDAP エントリーに対する特定のアクションを行うことが許可されています。
- アクセス制御リスト (ACL) フィルターが、`actionDate` (DateOfBirth) および `employeeNumber` (社会保障番号) に含まれているデータを保護するために、従業員レコードに追加された。使用されている ACL フィルターは、従業員がレコードを読み取ることを許可しますが、`employeeNumber` および `actionDate` の値を表示できるのは `powerEmployees` のみです。

– `aclentry: group:cn=employees,o=iDrink,dc=domainName,dc=domainSuffix:normal:grant:rs`

- aclentry: group:cn=employees,o=iDrink,dc=*domainName*,dc=*domainSuffix*:  
at.employeeNumber:deny:rsc:at.actionDate:deny:rsc
- aclentry: group:cn=powerEmployees,o=iDrink,dc=*domainName*,dc=*domainSuffix*:  
at.employeeNumber:grant:rsc:at.actionDate:grant:rsc
- LDAP ディレクトリーにデータを追加する場合、すべての日付フィールドのフォーマットは、  
YYYYMMDDHHMMSS[.l,fraction][(+|-HHMM)IZ] になります。

## データベース設計

以下の図は、iDrink のデータベース設計を示しています。

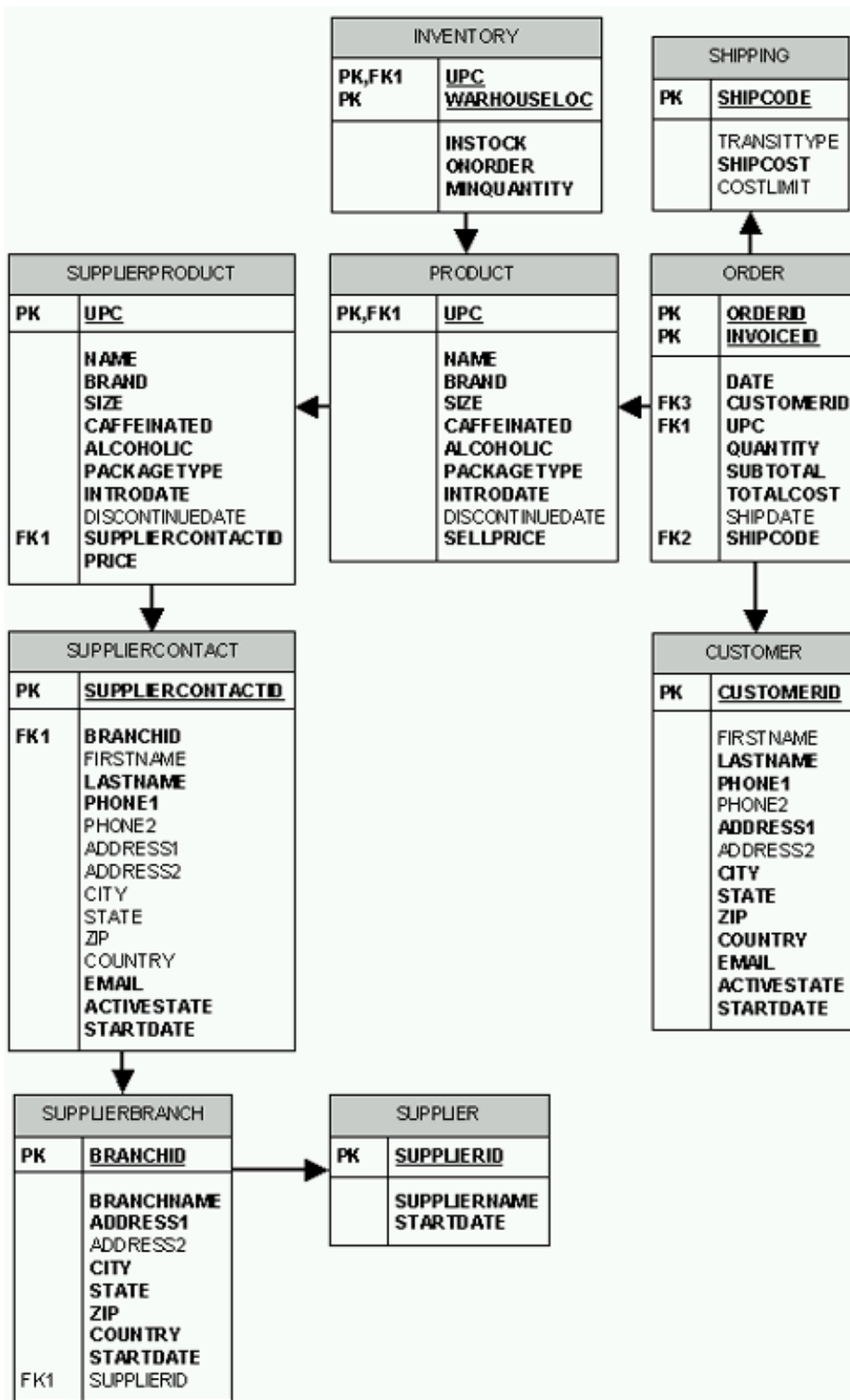


図 6: iDrink データベース設計

以下は、各テーブルの簡潔な要約です。

- **SUPPLIER:** iDrink のサプライヤーに関する情報を含みます。
- **SUPPLIERBRANCH:** iDrink のサプライヤーの支店に関する情報を含みます。
- **SUPPLIERCONTACT:** iDrink のサプライヤーの連絡先に関する情報を含みます。
- **SUPPLIERPRODUCT:** iDrink のサプライヤーによって提供される製品に関する情報を含みます。このデータベースには、iDrink が販売しない製品も含まれています。
- **PRODUCT:** iDrink によって提供される製品に関する情報を含みます。PRODUCT テーブルの値は、SUPPLIERPRODUCT テーブルの値のサブセットです。
- **INVENTORY:** iDrink が現在所有している製品の数量、およびこれらの製品がある場所に関する情報が含まれています。
- **CUSTOMER:** iDrink から製品を購入する顧客を含みます。
- **ORDER:** iDrink 顧客のオーダー情報を含みます。
- **SHIPPING:** iDrink がオーダーを出荷するために使用する出荷タイプを含みます。

## JavaBean 設計

JavaBeans は、サーブレットによって取得されるデータを格納するために作成されました。iDrink アプリケーションでは、以下の JavaBeans が作成されました。

- **CustomerBean** - CustomerBean は、CUSTOMER データベースからの情報を格納します。
- **CustomerInvoiceBean** - CustomerInvoiceBean は、ORDER データベースからの情報を格納します。
- **InventoryBean** - InventoryBean は、INVENTORY データベースからの情報を格納します。
- **InvoiceItemBean** - InvoiceItemBean は、ORDER データベース内の特別オーダーの単一品目に関する情報を格納します。
- **ProductBean** - ProductBean は、PRODUCT データベースからの情報を格納します。
- **ShippingBean** - ShippingBean は、SHIPPING データベースからの情報を格納します。
- **SupplierBean** - SupplierBean は、SUPPLIER データベースからの情報を格納します。
- **SupplierBranchBean** - SupplierBranchBean は、SUPPLIERBRANCH データベースからの情報を格納します。
- **SupplierContactBean** - SupplierContactBean は、SUPPLIERCONTACT データベースからの情報を格納します。
- **SupplierProductBean** - SupplierProductBean は、SUPPLIERPRODUCT データベースからの情報を格納します。
- **UserBean** - UserBean は、LDAP にあるユーザー情報を格納します。

## サーブレット設計

4 つのサーブレットが、iDrink アプリケーションで使用されました。以下で、各サーブレットについて簡単に説明しています。

- **CustomerServlet** - CustomerServlet は、顧客アプリケーション・フローで使用されるメソッドを含みます。
- **EmployeeServlet** - EmployeeServlet は、従業員アプリケーション・フローで使用されるメソッドを含みます。一部のメソッドは、サプライヤー・アプリケーション・フローでも使用されます。
- **SupplierServlet** - SupplierServlet は、サプライヤー・アプリケーション・フローで使用されるメソッドを含みます。

- **UserServlet** - UserServlet は、確実に正しい権限を持つユーザーのみが該当するページにログオンおよびアクセスできるようにする (顧客が従業員のページを表示できなくなるようにするなど) ための、アプリケーション・フロー全体を通じて使用されるメソッドを含みます。

---

## UserServlet

iDrink UserServlet は、iDrink ユーザーが iDrink にログインしたり、パスワードを変更したり、iDrink からログアウトするのに必要な機能を提供します。サーブレットは、これらのタスクを遂行するために、LDAP ディレクトリーを使用します。

### 設計のポイント

iDrink アプリケーションには、ユーザーがログインしているかどうかを判別するためのメカニズムが必要でした。ユーザーがログインすると、アプリケーションもまたそのユーザーのユーザー・タイプを認識する必要がありました。これらのタスクを遂行するために、iDrink アプリケーションは UserBean クラスに依存しています。UserBean オブジェクトは、ユーザーのユーザー ID、それぞれの iDrink ユーザー・タイプと関連する 3 つの定数値の 1 つを割り当てられている整数値を含んでいます。ユーザーが iDrink にログインすると、UserServlet は、新規 UserBean オブジェクトを作成し、ユーザー ID およびユーザー・タイプの値を設定し、そのオブジェクトをセッションに追加します。ユーザーがログアウトするまで、そのオブジェクトはセッション内にあります。このように、ユーザーが一度ログインすると、そのユーザーがアクセスするサーブレットや JSP は、そのユーザーのユーザー ID やユーザー・タイプを容易に判別できます。

### メソッド

以下は、UserServlet にあるメソッドのリストです。doPost メソッドは、顧客からすべての着信要求を受け取り、このメソッド・リストから該当するメソッドを呼び出します。

- **assertUserType** - assertUserType メソッドは、要求から UserBean を取得します。UserBean がヌルであるか、正当なユーザー・タイプではない場合、メソッドが UserLogin JSP を表示します。
- **changePassword** - changePassword メソッドは、ユーザー ID、旧パスワード、および新規パスワードを受け入れます。指定したユーザー ID の LDAP エントリーがある場合は、そのエントリーの userPassword 属性は新規パスワードと共に更新されます。これが成功すると、メソッドは真のブール値を返します。この処理中に例外が throw されると、メソッドは偽のブール値を返します。
- **handlePasswdRequest** - handlePasswdRequest メソッドは、新規パスワードが確認パスワードと一致しているかどうかを検証し、changePassword メソッドを呼び出して、既存のパスワードを変更します。次に、メソッドが UserPasswdResults JSP を表示します。
- **loginUser** - loginUser メソッドは、要求からユーザー ID およびパスワードを取得し、verifyUser メソッドを呼び出して、指定したパスワードがそのユーザー ID のパスワードとして正しいかどうかを検証します。パスワードが正しい場合、UserBean が作成され、セッション・オブジェクトに追加されます。ログオンするユーザーのタイプに応じて、該当する JSP が表示されます。顧客は iDrinkHome JSP、従業員は EmployeeHome JSP、およびサプライヤーは SupplierHome JSP を表示します。
- **logoutUser** - logoutUser メソッドは、セッション・オブジェクトから UserBean オブジェクトを除去し、iDrinkHome JSP を表示します。
- **verifyUser** - verifyUser メソッドは、ユーザー ID およびパスワードを受け入れ、そのパスワードが、指定したユーザー ID のパスワードとして正しいかどうかを検証します。パスワードが正しい場合は、メソッドは iDrink ユーザーのタイプを表す整数を返します。パスワードが正しくない場合は、メソッドはゼロを返します。

## JavaServer Pages



以下は、UserServlet によって呼び出される JSP のリストです。

- **CustomerProceedToCheckout** - CustomerProceedToCheckout JSP は、顧客に CustomerCheckout JSP を参照させるリンクを表示します。この JSP は、チェックアウトの処理中に顧客がログインするとき、中間ステップとして使用されます。
- **EmployeeHome** - EmployeeHome JSP は、従業員がログインした後に表示されます。
- **iDrinkHome** - iDrinkHome JSP は、顧客がログインした後に表示されます。また、ユーザーがログアウトした後にも表示されます。
- **SupplierHome** - SupplierHome JSP は、サプライヤーがログインした後に表示されます。
- **UserLogin** - UserLogin JSP は、ログイン処理の最初に表示されます。ユーザー ID およびパスワード入力用のテキスト・フィールドを表示します。
- **UserPasswdResults** - UserPasswdResults ページは、ユーザーがパスワードを変更した後に表示されません。ユーザーに、パスワード変更要求の成否を通知します。

---

## CustomerServlet

iDrink CustomerServlet は、iDrink の顧客が iDrink と業務を行うのに必要な機能を提供します。また、iDrink での登録、顧客情報の更新、商品の検索、商品説明の表示、ショッピング・カートへ品目の追加、発注、発注履歴の表示などが行える機能を顧客に提供します。CustomerServlet およびそれによって呼び出される JavaServer Pages (JSP) は、UserServlet と共に作動して、ユーザーが iDrink のログインおよびログアウト、パスワードの変更を行うようにできる機能性を提供します。デフォルトでは、CustomerServlet は、iDrink 従業員または iDrink のサプライヤーとしてログインするまで、iDrink ユーザーのすべての要求を処理します。iDrink ユーザーが iDrink の顧客としてログインする場合、CustomerServlet はそのユーザーからの要求を処理し続けます。

### 設計のポイント

iDrink によって顧客がショッピングをしている間に、CustomerServlet は、iDrinkCart オブジェクトを使用して、顧客がショッピング・カートに追加する品目を格納します。iDrinkCart オブジェクトは java.util.Hashtable を含んでおり、iDrinkCart に追加された iDrinkCartItems をこれに格納します。各 iDrinkCartItem には、カート内の品目情報 (UPC 番号、数量、および価格など) が含まれています。ハッシュ・テーブルの各 iDrinkCartItem のキー値は、その品目の UPC 番号であり、これによりハッシュ・テーブルから品目を容易に取得できます。

新規の顧客が iDrink で登録された場合、新規で固有のカスタマー ID が生成され、新しい顧客のアカウントに割り当てられる必要があります。各顧客の顧客表には、1 つのレコードのみがあるため、ID 列はカスタマー ID 値を生成するために使用されます。ID 列は、表内の各行に固有の主キーを容易に自動生成する方法を提供します。これにより、アプリケーションが自身の固有値を生成する必要があり、それらの値が通常、すでにテーブル内にある値に基づいている場合に発生する可能性がある並行性およびパフォーマンス上の問題がなくなります。カスタマー ID は、顧客表内の ID 列であるため、CustomerServlet は、次のカスタマー ID が何であるかを判別する必要はありません。単に、ID 列の機能性がこの作業を自動的に処理します。

顧客が iDrink で注文した場合、新規で固有の請求書 ID が生成され、その注文に割り当てられる必要があります。CustomerServlet が、顧客から注文情報を受け取るとき、注文の個々の品目の注文表に 1 つのレコードを挿入します。これらのレコードのそれぞれは、同じ請求書 ID を含む必要があります。つまり、同じ請求書 ID を持つ注文表に複数のレコードが入ることになります。このため、ID 列が、全体の注文に対して 1 つの固有値ではなく、注文の各品目に固有値を生成するため、固有請求書 ID を生成するために ID 列を使用できません。固有の請求書 ID を生成する際、CustomerServlet は DB2<sup>(R)</sup> シーケンス・オブジ

ェクトに依存します。DB2 シーケンス・オブジェクトは、シーケンス値を生成するオブジェクトです。値がシーケンス・オブジェクトから要求されると、シーケンス・オブジェクトが作成されたときに定義されたシーケンス内の次の値が戻されます。請求書 ID のシーケンス・オブジェクトで定義されたシーケンスは、単に、昇順で 1 ずつ増加する整数のシーケンスです。CustomerServlet が顧客の注文を実行するとき、シーケンス・オブジェクトにアクセスして、次の請求書 ID を取得します。この値は、実行された注文で実行される、すべての SQL INSERT ステートメントで使用されます。

iDrink の Web サイトを使用するまで、顧客はログインを強制されません。つまり、ロジックを CustomerServlet、UserServlet、およびさまざまな JSP に追加して、顧客がカートに品目を追加し、チェックアウト処理を開始するまでログインや登録しない場合に処理を行うようにすることが必要でした。顧客がチェックアウト・ボタンを押す前にログインを行っていない場合は、追加のパラメーターが HTTP 要求で CustomerServlet (顧客登録の場合) または UserServlet (ログインの場合) に送信されます。これらのサーブレットのいずれかが登録またはログイン要求でこのパラメーターを検出した場合、サーブレットは要求を処理し、チェックアウト処理を続行するためのカスタマー ID およびリンクを表示する中間チェックアウト・ページにユーザーをリダイレクトします。顧客が iDrink Web サイトの他の場所からログインおよび登録処理を開始した場合、登録またはログイン要求の正常終了後に、顧客はウェルカム画面にリダイレクトされます。この場合、中間ページは必要ではなく、表示もされません。

## アプリケーションのフロー

iDrink お顧客インターフェースは、顧客に事前定義したパスを強制しません。このように、顧客は、1 つのタスクから別のタスクにいつでも容易にジャンプすることができます。この柔軟性によって、iDrink でショッピング中に、顧客インターフェースを介して顧客が取る可能性のあるパスをすべて例示することはかなり困難です。図 7 は、iDrink 顧客が最もよく使用するパスを示しています。



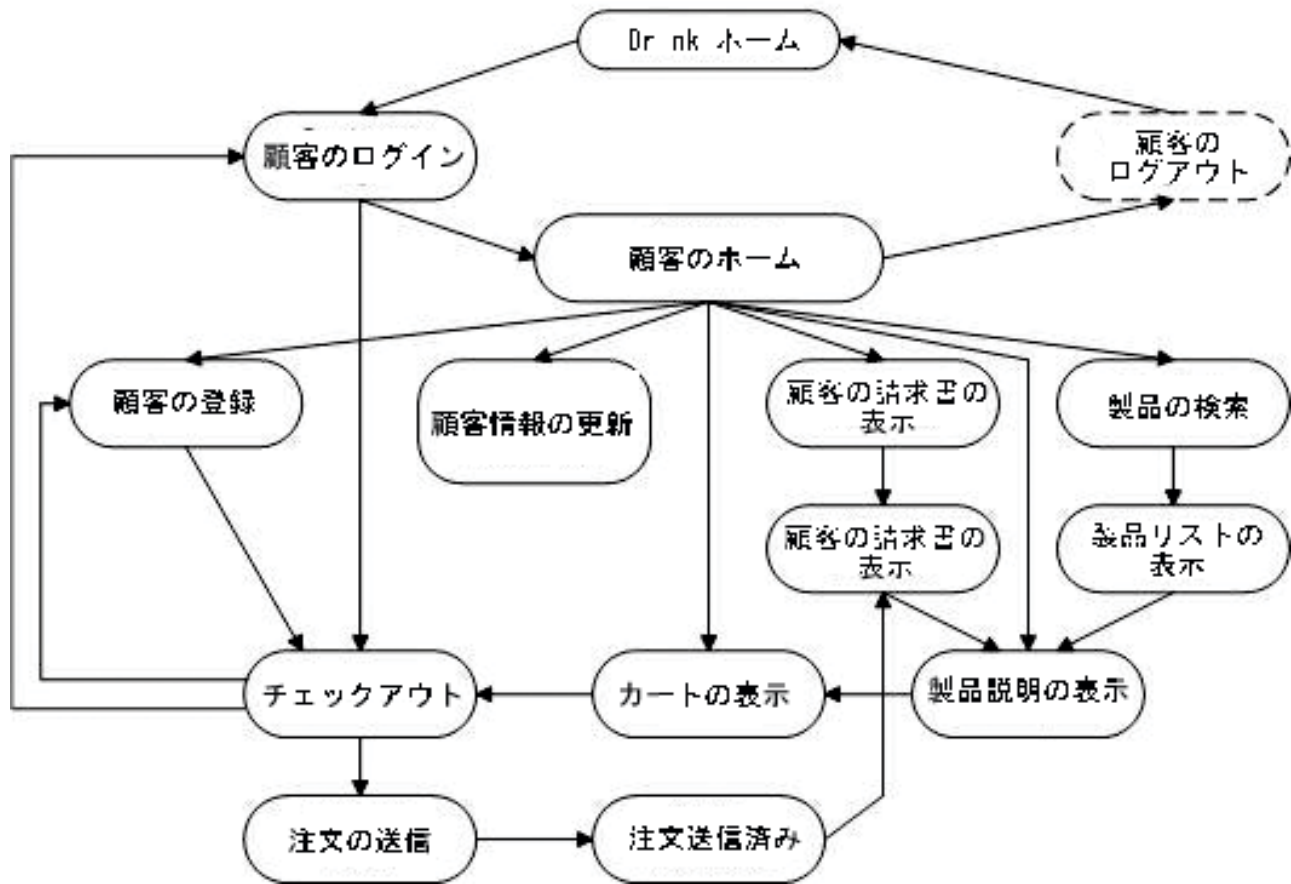


図 7: iDrink 顧客アプリケーションのフロー

## メソッド

以下は、CustomerServlet にあるメソッドのリストです。doPost メソッドは、顧客からのすべての着信要求を受け取り、このメソッド・リストから該当するメソッドを呼び出します。

- **addItemToCart** - addItemToCart メソッドは、指定した商品を指定数量だけ顧客の iDrinkCart に追加します。更新されたカートの内容は CustomerViewCart JSP に表示されます。
- **checkout** - checkout メソッドは、iDrinkCart 内の品目の値に基づいて、顧客の注文の有効な配送コードを取得します。これらの配送コードは、CustomerCheckout の JSP に顧客のセレクション用に表示されます。
- **displayMatchingProducts** - displayMatchingProducts メソッドは、顧客が指定した検索基準に一致する商品の基本情報を取得します。この情報は、CustomerMatchingProducts JSP に表示されます。
- **getNextInvoiceID** - getNextInvoiceID メソッドは、DB2 シーケンス・オブジェクトによって提供される次の請求書 ID を戻します。
- **registerNewCustomer** - registerNewCustomer メソッドは、新規のレコードを iDrink 顧客テーブルに挿入します。このレコードは、顧客の情報を含みます。また、顧客の新規 LDAP エントリーも作成します。このエントリーは、顧客のパスワードを含みます。
- **submitOrder** - submitOrder メソッドは、getNextInvoiceID メソッドから新規請求書 ID を取得します。次に、顧客の iDrinkCart のそれぞれの品目の iDrink 注文表に 1 つのレコードを挿入します。

- **updateCustomer** - updateCustomer メソッドは、現在の顧客の顧客情報を顧客が指定した値で更新します。
- **updateItemsInCart** - updateItemsInCart メソッドは、顧客の iDrinkCart にある各品目の数量を、顧客が指定した値に基づいて更新します。更新された内容は、CustomerViewCart JSP に表示されます。
- **viewInvoices** - viewInvoices メソッドは、現在の顧客が実行した各注文の基本情報を取得します。この情報は、CustomerInvoices JSP に表示されます。
- **viewInvoiceInformation** - viewInvoiceInformation メソッドは、指定の顧客注文の詳細情報を取得します。この情報は、CustomerInvoiceInfo JSP に表示されます。
- **viewProduct** - viewProduct メソッドは、指定の UPC を持つ製品の詳細情報を取得します。この情報は、CustomerProductInfo JSP に表示されます。
- **viewUpdateCustomerScreen** - viewUpdateCustomerScreen メソッドは、現在の顧客の顧客情報を取得します。この情報は、CustomerUpdate JSP に表示されます。

## JavaServer Pages

以下は、CustomerServlet によって呼び出される JSP のリストです。

- **CustomerCheckout** - CustomerCheckout JSP は、顧客が注文で選択する使用可能な出荷オプションのリストを表示します。
- **CustomerDisplayCart** - CustomerDisplayCart JSP は、顧客の iDrinkCart の内容を表示します。
- **CustomerInvoiceInfo** - CustomerInvoiceInfo の JSP は、指定の顧客注文の詳細情報を表示します。
- **CustomerInvoices** - CustomerInvoices JSP は、顧客が実行したすべての注文のリストを表示します。
- **CustomerMatchingProducts** - CustomerMatchingProducts JSP は、顧客が指定した検索基準と一致するすべての商品を表示します。
- **CustomerOrderSubmitted** - CustomerOrderSubmitted JSP は、注文を受けたことをユーザーに通知し、請求書番号を表示します。
- **CustomerProceedToCheckout** - CustomerProceedToCheckout JSP は、顧客に CustomerCheckout JSP を参照させるリンクを表示します。この JSP は、チェックアウトの処理中に顧客が登録またはログインするとき、中間ステップとして使用されます。
- **CustomerProductInfo** - CustomerProductInfo JSP は、指定の iDrink 品目の詳細情報を表示します。
- **CustomerRegistration** - CustomerRegistration JSP は、新規顧客が顧客情報を入力および実行依頼できるようにする入力フィールドを提供します。
- **CustomerSearchProducts** - CustomerSearchProducts JSP は、顧客が一連の製品検索基準を入力および実行依頼できるようにする入力フィールドを提供します。
- **CustomerUpdate** - CustomerUpdate JSP は、テキスト・フィールドに現在の顧客情報を表示して、ユーザーが更新された顧客情報を変更および実行依頼できるようにします。

---

## EmployeeServlet

iDrink EmployeeServlet は、iDrink 従業員が、顧客、サプライヤー、オーダー、製品、およびインベントリーを含む、会社のすべてのエンティティを処理するために必要な機能性を提供します。次の機能が従業員に提供されます。

- 顧客オーダーの表示
- iDrink が提供する製品の追加、表示、更新、および削除
- インベントリーからの製品の追加、表示、更新、および削除

- サプライヤー、サプライヤー支店、およびサプライヤー連絡先の作成、表示、更新、および削除
- 出荷タイプの作成、表示、更新、および削除
- Cola Connections サイトへのアクセス

EmployeeServlet およびそれによって呼び出される JSP は、UserServlet と連動して、EmployeeServlet に関連するページに、従業員だけがログオンおよびアクセスできるようにする機能性を提供します。

### 設計のポイント

- 顧客アプリケーション・フローの設計で ID 列を使用するのに並行して、従業員アプリケーション・フローでも ID 列が使用されます。ID 列は、以下の主キー値を簡単に作成するために、これらのテーブルで使用されます。
  - SHIPPING
  - SUPPLIER
  - SUPPLIERBRANCH
  - SUPPLIERCONTACT
- 従業員およびサプライヤーのアプリケーション・フローの比較において、次の処理が類似していることがわかりました。
  - サプライヤー情報の検索および表示
  - サプライヤー支店情報の検索および表示
  - サプライヤー連絡先情報の検索および表示

これらの処理が同一であることから、EmployeeServlet および SupplierServlet の両方にメソッドを書き込む代わりに、EmployeeServlet にこれらの処理のメソッドを 1 セットだけ書き込むことが決まりました。

また、情報を検索および表示するため、JSP ページが 1 セットだけ作成されました。JSP が 1 セットだけ使用されたので、サプライヤーと従業員の両方がこれらのページへのアクセスを許可されるように指定することが重要になりました。このことは、次のコードで例示するように、サプライヤー・チェック条件を検索および表示 JSP に追加することによって行われました。

```
UserServlet.assertUserType
(request, response, UserBean.USER_TYPE_SUPPLIER | UserBean.USER_TYPE_EMPLOYEE, "UserLogin.jsp");
```

従業員アプリケーション・フローは、サプライヤー、サプライヤー支店、およびサプライヤー連絡先情報をこれらのページから更新および削除するためのリンクを必要としていたので、これらのリンクを表示する前にユーザー・タイプを確認することが重要になりました。ユーザーのタイプがサプライヤーである場合、更新および削除へのリンクが表示されませんでした。この JSP コードの例は、次のようになります。

```
<%
  if(userbean.getUserType() == UserBean.USER_TYPE_EMPLOYEE)
  {
  %>

  <A HREF="EmployeeServlet?action=viewSupplierContactToUpdate&contactID=<%=
    supplierContactBean.getContactID()%>">Update Supplier Contact</A>
  <br /><BR>
  <A HREF="EmployeeServlet?action=viewSupplierContactToDelete&contactID=<%=
    supplierContactBean.getContactID()%>">Delete Supplier Contact</A>
  <br /><BR>

  <%
  }
```

従業員がサプライヤー、サプライヤー連絡先、またはインベントリーを削除すると、そのエンティティはデータベースから削除されます。しかし、製品および出荷コードの削除は、異なる意味を持ちます。情報は、履歴上の理由からデータベースに残す必要があるため、その品目はデータベースに残され、特定のフィールドが、その品目が削除されていることを示すように設定されます。

- 製品を削除するには、DISCONTINUEDATE を設定します (ヌルの DISCONTINUEDATE を持つ製品は、製品が削除されていないことを意味します)。
- 出荷コードを削除するには、COSTLIMIT を 0 に設定します (0 より大きい COSTLIMIT を持つ出荷コードは、オーダーで使用できるアクティブ出荷コードです)。

また、品目によっては、次のような条件を満たしていなければ削除できないものもあります。

- 関連するサプライヤー支店を持っているサプライヤーは、削除してはならない。
- 関連するサプライヤー連絡先を持っているサプライヤー支店は、削除してはならない。
- 関連するサプライヤー製品を持っているサプライヤー連絡先や、アクティブ状態のサプライヤー連絡先は、削除してはならない。
- 新規製品を顧客オファリングに追加する場合、iDrink 従業員は、すでに PRODUCT テーブルではなく、SUPPLIERPRODUCT テーブルから製品を選択する必要があります。SUPPLIERPRODUCT テーブルが、iDrink サプライヤーによって提供されるすべての製品を含んでいることを再確認してください。PRODUCT テーブルは、iDrink が顧客に提供する製品を含んでいます。PRODUCT テーブルの製品は、SUPPLIERPRODUCT テーブルの製品のサブセットです。PRODUCT テーブルにはない、SUPPLIERPRODUCT テーブル内の製品を表示するために、次の例に示すように、副選択が SUPPLIERPRODUCT テーブルで実行されました。

```
SELECT *  
FROM IDRINK.SUPPLIERPRODUCT  
WHERE UPC NOT IN (SELECT UPC FROM  
IDRINK.PRODUCT)
```

- 既存のインベントリーを更新する際に、インプリメント可能な、次のようないくつかのオプションがありました。
  - 新規金額で既存のインベントリーを上書きする (Overwriting the existing inventory with a new amount)
  - 既存のインベントリーから金額を追加または減算する (Adding or subtracting an amount from the existing inventory)

充分検討した上で、最初のオプション「新規金額で既存のインベントリーを上書きする (overwriting the existing inventory with a new amount)」をインプリメントすることに決めました。この結果、インプリメントが最も簡単であるだけでなく、従業員が最も簡単に使用できるオプションが提供されました。

- 出荷作業のとき、各出荷コードのオーダーに関して、従業員が最小および最大コストの制限を表示できることは有益です。しかし、SHIPPING テーブルでは、出荷コードの最大コスト制限のみが格納されません。最小コスト制限は、出荷コードの出荷タイプで、次の最小コスト制限の最大コスト制限に暗黙に含まれています (iDrink によって現在使用されている出荷タイプには、GROUND、NEXT DAY AIR、および 2ND DAY AIR などがあります)。JSP が簡単に最小コスト制限情報にアクセスできるように、costLimitMin と呼ばれる変数が ShippingBean に作成され、この情報を保持します。

## アプリケーションのフロー

iDrink 従業員のインターフェースでは、従業員に事前定義したパスが強制されません。したがって、従業員は、あるタスクから別のタスクにいつでも簡単にジャンプできます。この柔軟性の結果、従業員が従業員インターフェースを介して使用する可能性のあるパスをすべて明らかにすることは、非常に困難であると予想されます。図 8 は、iDrink 従業員が使用できる基本的なアプリケーション・フローを示しています。

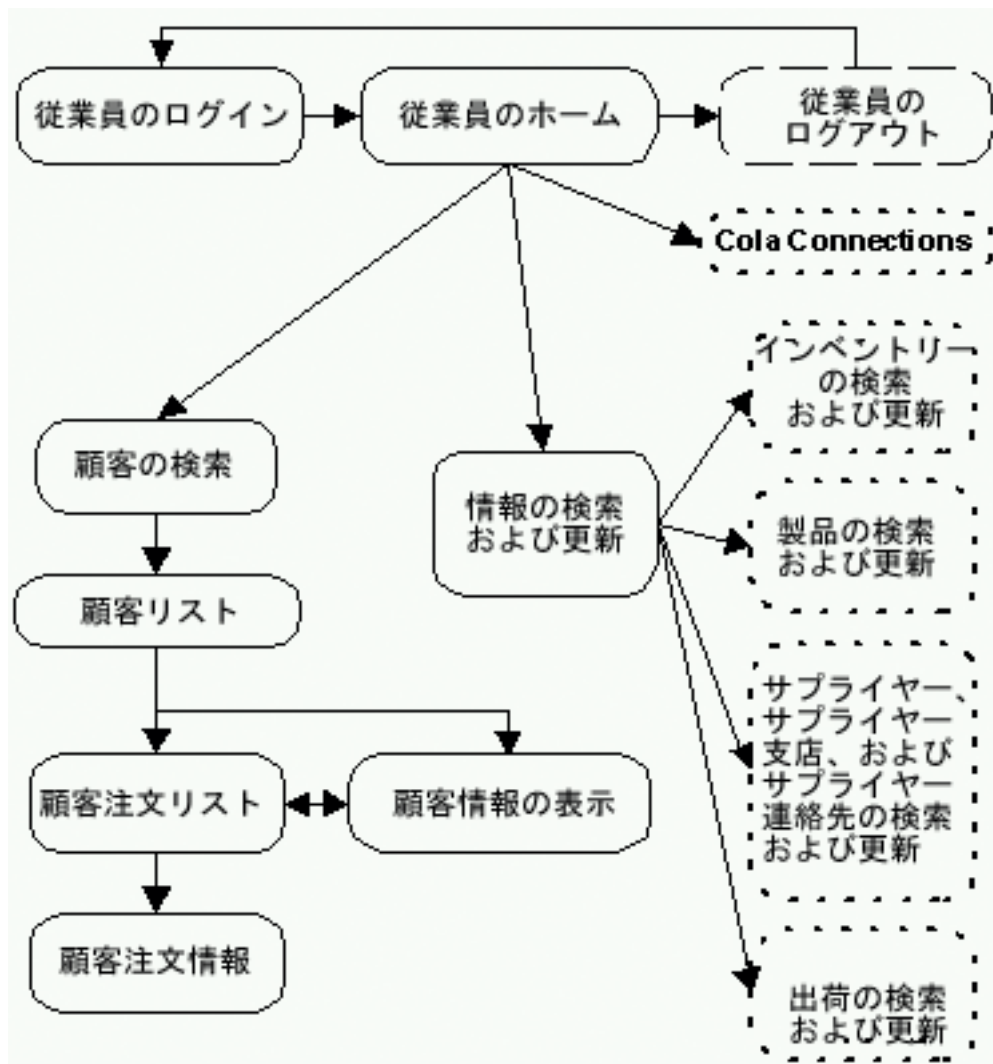


図 8: iDrink 従業員アプリケーションのフロー

これらの品目のアプリケーション・フローは、以下の点で類似しています。

- インベントリーの検索および更新
- 製品の検索および更新
- サプライヤー、サプライヤー支店、およびサプライヤー連絡先の検索および更新
- 出荷の検索および更新

例えば、図 9 は、製品の検索および更新のアプリケーション・フローを例示します。



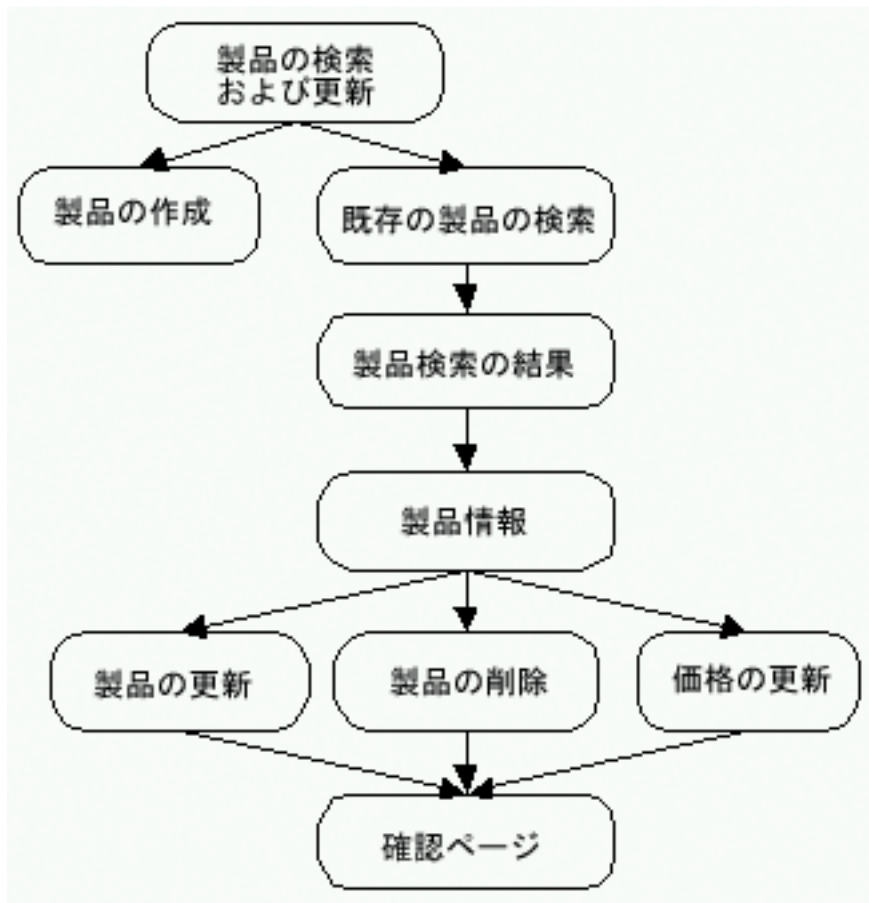


図 9: 製品の検索および更新のアプリケーション・フロー

若干の例外はありますが、同様のアプリケーション・フローが、インベントリ、出荷、サプライヤー、サプライヤー支店、およびサプライヤー連絡先で使用されます。

Cola Connections のアプリケーション・フローについては、Lotus<sup>(R)</sup> Domino<sup>(R)</sup> 環境セクションで説明されています。

### メソッド

以下のテーブルは、EmployeeServlet にあるメソッドを示しています。大部分の iDrink エンティティが、同様のアクションを実行するメソッドを持っているため、それらのメソッドはアクション別にカテゴリー化されています。

doPost メソッドは、顧客からすべての着信要求を受け取り、このメソッド・リストから該当するメソッドを呼び出します。

エンティティ	メソッド名	使用されるテーブル	メソッドの完了後に呼び出されるもの
インベントリ	createInventory	INVENTORY	Confirm.jsp
製品	createProduct	PRODUCT、SUPPLIER PRODUCT	Confirm.jsp
出荷	createShipping	SHIPPING	Confirm.jsp

サプライヤー	createSupplier	SUPPLIER	Confirm.jsp
サプライヤー支店	createSupplierBranch	SUPPLIER、 SUPPLIERBRANCH	Confirm.jsp
サプライヤー連絡先	createSupplierContact	SUPPLIER、 SUPPLIERBRANCH、 SUPPLIERCONTACT	Confirm.jsp

表 1: 新規オブジェクトを作成し、そのオブジェクトを該当するテーブルに挿入するメソッド

エンティティ	メソッド名	使用されるテーブル	メソッドの完了後に呼び出されるもの
インベントリ	deleteInventory	INVENTORY	Confirm.jsp
製品	deleteProduct	PRODUCT	Confirm.jsp
出荷	deleteShipping	SHIPPING	Confirm.jsp
サプライヤー	deleteSupplier	SUPPLIER	Confirm.jsp
サプライヤー支店	deleteSupplierBranch	SUPPLIER、 SUPPLIERBRANCH	Confirm.jsp
サプライヤー連絡先	deleteSupplierContact	SUPPLIER、 SUPPLIERBRANCH、 SUPPLIERCONTACT	Confirm.jsp

表 2: 従業員によるオブジェクトの除去を可能にするメソッド。**deleteProduct** および **deleteShipping** を除くすべてのメソッドは、該当する条件が満たされると、データベースからオブジェクトを削除します。

エンティティ	メソッド名	使用されるテーブル	メソッドの完了後に呼び出されるもの
顧客	searchCustomerOrders	ORDER	EmployeeCustomerOrdersList.jsp
顧客	searchCustomers	CUSTOMER	EmployeeCustomerList.jsp
インベントリ	searchInventory	INVENTORY、 PRODUCT	InventorySearchResults.jsp
製品	searchProduct	PRODUCT	ProductSearchResults.jsp
出荷	searchShipping	SHIPPING	ShippingSearchResults.jsp
サプライヤー	searchSupplier	SUPPLIER	SupplierSearchResults.jsp
サプライヤー支店	searchSupplierBranch	SUPPLIER、 SUPPLIERBRANCH	SupplierBranchSearchResults.jsp
サプライヤー連絡先	searchSupplierContact	SUPPLIER、 SUPPLIERBRANCH、 SUPPLIERCONTACT	SupplierContactSearchResults.jsp
サプライヤー製品	searchSupplierProduct	PRODUCT、 SUPPLIERCONTACT、 SUPPLIERPRODUCT	ProductSearchResults.jsp

表 3: 従業員によるオブジェクトの検索を可能にするメソッド

エンティティ	メソッド名	使用されるテーブル	メソッドの完了後に呼び出されるもの
--------	-------	-----------	-------------------

インベントリ	updateInventory	INVENTORY	Confirm.jsp
製品	updatePrice	PRODUCT	Confirm.jsp
製品	updateProduct	PRODUCT	Confirm.jsp
出荷	updateShipping	SHIPPING	Confirm.jsp
サプライヤー	updateSupplier	SUPPLIER	Confirm.jsp
サプライヤー支店	updateSupplierBranch	SUPPLIER、 SUPPLIERBRANCH	Confirm.jsp
サプライヤー連絡先	updateSupplierContact	SUPPLIER、 SUPPLIERBRANCH、 SUPPLIERCONTACT	Confirm.jsp

表 4: 従業員によるオブジェクトの更新を可能にするメソッド

エンティティ	メソッド名	使用されるテーブル	メソッドの完了後に呼び出されるもの
顧客	viewCustomer	CUSTOMER	EmployeeCustomerInfo.jsp
顧客	viewCustomerOrder	ORDER、 PRODUCT、 SHIPPING	EmployeeOrderInfo.jsp
インベントリ	viewInventoryItem	INVENTORY	InventoryInfo.jsp
製品	viewPrice	PRODUCT	PriceUpdate.jsp
製品	viewProductItem	PRODUCT	ProductInfo.jsp
出荷	viewShippingItem	SHIPPING	ShippingInfo.jsp
サプライヤー	viewSupplierItem	SUPPLIER	SupplierInfo.jsp
サプライヤー支店	viewSupplierBranchItem	SUPPLIER、 SUPPLIERBRANCH	SupplierBranchInfo.jsp
サプライヤー連絡先	viewSupplierContactItem	SUPPLIER、 SUPPLIERBRANCH、 SUPPLIERCONTACT	SupplierContactInfo.jsp

表 5: 従業員によるオブジェクトの詳細の表示を可能にするメソッド

エンティティ	メソッド名	使用されるテーブル	メソッドの完了後に呼び出されるもの
インベントリ	inventoryInfo	INVENTORY、 PRODUCT	
インベントリ	viewInventoryToDelete (inventoryInfo を呼び出す)		InventoryDelete.jsp
インベントリ	viewInventoryToUpdate (inventoryInfo を呼び出す)		InventoryUpdate.jsp
製品	productInfo	PRODUCT	
製品	viewProductToDelete (productInfo を呼び出す)		ProductDelete.jsp
製品	viewProductToUpdate (productInfo を呼び出す)		ProductUpdate.jsp
出荷	fetchShippingInfo	SHIPPING	



出荷	viewShippingToDelete (fetchShippingInfo を呼び出す)		ShippingDelete.jsp
出荷	viewShippingToUpdate (fetchShippingInfo を呼び出す)		ShippingUpdate.jsp
サプライヤー	fetchSupplierInfo	SUPPLIER	
サプライヤー	viewSupplierToDelete (fetchSupplierInfo を呼び出す)		SupplierDelete.jsp
サプライヤー	viewSupplierToUpdate (fetchSupplierInfo を呼び出す)		SupplierUpdate.jsp
サプライヤー支店	fetchSupplierBranchInfo	SUPPLIER、 SUPPLIERBRANCH	
サプライヤー支店	viewSupplierBranchToDelete (fetchSupplierBranchInfo を呼び出す)		SupplierBranchDelete.jsp
サプライヤー支店	viewSupplierBranchToUpdate (fetchSupplierBranchInfo を呼び出す)		SupplierBranchUpdate.jsp
サプライヤー連絡先	fetchSupplierContactInfo	SUPPLIER、 SUPPLIERBRANCH、 SUPPLIERCONTACT	
サプライヤー連絡先	viewSupplierContactToDelete (fetchSupplierContactInfo を呼び出す)		SupplierContactDelete.jsp
サプライヤー連絡先	viewSupplierContactToUpdate (fetchSupplierContactInfo を呼び出す)		SupplierContactUpdate.jsp

表 6: 現行情報と一緒にセッションに関連付けられた Bean を準備するメソッドか、現行情報と一緒にセッションに関連付けられた Bean を準備するメソッドを呼び出すメソッド。情報は、該当するページに表示されます。

それ以外のメソッドについては、詳しく説明します。

- **getInventoryCreateInfo** - getInventoryCreateInfo メソッドは、PRODUCT テーブルから有効な UPC のリストを取得します。この情報は、InventoryCreate JSP に表示されます。
- **getInventorySearchInfo** - getInventorySearchInfo メソッドは、PRODUCT テーブルからすべての有効な UPC を、INVENTORY テーブルからすべての有効なウェアハウスのリストを取得します。この情報は、InventorySearch JSP に表示されます。
- **getProductCreateInfo** - getProductCreateInfo メソッドは、SUPPLIERPRODUCT テーブルにはあるが PRODUCT テーブルにはない製品情報を取得します。この情報は、ProductCreate JSP に表示されます。
- **getProductSearchInfo** - getProductSearchInfo メソッドは、検索にデフォルト値を提供するために、有効なパッケージ・タイプのリストを取得します。この情報は、ProductSearch JSP に表示されます。
- **fetchCostLimitMin** - fetchCostLimitMin メソッドは、配送コードが適用可能である最低限のコスト制限を計算し、この情報を出荷 Bean に設定します。このメソッドは、メソッド fetchShippingInfo によって呼び出されます。

- **fetchSupplierIDs** - fetchSupplierIDs メソッドは、検索にデフォルト値を提供するため、有効なサプライヤー ID および名前の一覧を取得します。この情報は、SupplierBranchSearch JSP に表示されます。
- **preCreateSupplierBranch** - preCreateSupplierBranch メソッドは、fetchSupplierIDs メソッドを呼び出して、有効なサプライヤーの一覧を取得します。この情報は、SupplierBranchCreate JSP に表示されます。
- **preSearchSupplierBranch** - preSearchSupplierBranch メソッドは、fetchSupplierIDs メソッドを呼び出して、有効なサプライヤーの一覧を取得します。この情報は、SupplierBranchSearch JSP に表示されます。
- **fetchSupplierBranchIDs** - fetchSupplierBranchIDs メソッドは、検索のデフォルト値を提供するため、有効なサプライヤー支店 ID および名前の一覧を取得します。この情報は、SupplierContactSearch JSP に表示されます。
- **preCreateSupplierContact** - preCreateSupplierContact メソッドは、fetchSupplierBranchIDs メソッドを呼び出し、有効なサプライヤー支店のリストを取得します。この情報は、SupplierContactCreate JSP に表示されます。
- **preSearchSupplierContact** - preSearchSupplierContact メソッドは、メソッド fetchSupplierIDs および fetchSupplierBranchIDs を呼び出し、有効なサプライヤーおよびサプライヤー支店のリストを取得します。この情報は、SupplierContactSearch JSP に表示されます。

## JavaServer Pages

以下のテーブルは、EmployeeServlet が使用する JSP を示しています。大部分の iDrink エンティティが、同様のアクションを実行する JSP と関連付けられているため、それらの JSP は、表示または要求される情報別にカテゴリー化されています。

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
インベントリ	InventoryCreate.jsp		InventoryHome.jsp
製品	ProductCreate.jsp		ProductHome.jsp
出荷	ShippingCreate.jsp		ShippingHome.jsp
サプライヤー	SupplierCreate.jsp		SupplierHome.jsp
サプライヤー支店	SupplierBranchCreate.jsp		SupplierHome.jsp
サプライヤー連絡先	SupplierContactCreate.jsp		SupplierHome.jsp

表 7: 新規オブジェクトを作成するための情報を求めるプロンプトを出す JSP

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
インベントリ	InventoryDelete.jsp	viewInventoryToDelete	
製品	ProductDelete.jsp	viewProductToDelete	
出荷	ShippingDelete.jsp	viewShippingToDelete	
サプライヤー	SupplierDelete.jsp	viewSupplierToDelete	
サプライヤー支店	SupplierBranchDelete.jsp	viewSupplierBranchToDelete	
サプライヤー連絡先	SupplierContactDelete.jsp	viewSupplierContactToDelete	

表 8: オブジェクトを削除する JSP

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
顧客	EmployeeCustomerInfo.jsp	viewCustomer	
顧客	EmployeeCustomerOrderInfo.jsp	viewCustomerOrder	
インベントリ	InventoryInfo.jsp	viewInventoryItem	
製品	ProductInfo.jsp	viewProductItem	
出荷	ShippingInfo.jsp	viewShippingItem	
サプライヤー	SupplierInfo.jsp	viewSupplierItem	
サプライヤー支店	SupplierBranchInfo.jsp	viewSupplierBranchItem	
サプライヤー連絡先	SupplierContactInfo.jsp	viewSupplierContactItem	

表 9: オブジェクトに関する詳細情報を表示する JSP

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
インベントリ	InventoryHome.jsp		EmployeeHome.jsp
製品	ProductHome.jsp		EmployeeHome.jsp
出荷	ShippingHome.jsp		EmployeeHome.jsp
サプライヤー、サプライヤー支店、サプライヤー連絡先	SupplierHome.jsp		EmployeeHome.jsp

表 10: さまざまなエンティティのホーム・ページである JSP

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
顧客	EmployeeCustomerOrdersSearch.jsp		EmployeeHome.jsp
インベントリ	InventorySearch.jsp		InventoryHome.jsp
製品	ProductSearch.jsp		ProductHome.jsp
出荷	ShippingSearch.jsp		ShippingHome.jsp
サプライヤー	SupplierSearch.jsp		SupplierHome.jsp
サプライヤー支店	SupplierBranchSearch.jsp		SupplierHome.jsp
サプライヤー連絡先	SupplierContactSearch.jsp		SupplierHome.jsp

表 11: 従業員によるオブジェクトの検索を可能にする JSP

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
顧客	EmployeeCustomerList.jsp	searchCustomers	
顧客	EmployeeCustomerOrdersList.jsp	searchCustomerOrders	
インベントリ	InventorySearchResults.jsp	searchInventory	
製品	ProductSearchResults.jsp	searchProduct	
出荷	ShippingSearchResults.jsp	searchShipping	

サプライヤー	SupplierSearchResults.jsp	searchSupplier	
サプライヤー支店	SupplierBranchSearchResults.jsp	searchSupplierBranch	
サプライヤー連絡先	SupplierContactSearchResults.jsp	searchSupplierContact	

表 12: 指定した検索基準と一致するオブジェクトを表示する JSP

エンティティ	JSP 名	JSP を呼び出すメソッド	JSP へのリンクを含むページ
インベントリ	InventoryUpdate.jsp	viewInventoryToUpdate	
製品	ProductUpdate.jsp	viewProductToUpdate	
出荷	ShippingUpdate.jsp	viewShippingToUpdate	
サプライヤー	SupplierUpdate.jsp	viewSupplierToUpdate	
サプライヤー支店	SupplierBranchUpdate.jsp	viewSupplierBranchToUpdate	
サプライヤー連絡先	SupplierContact.jsp	viewSupplierContactToUpdate	

表 13: 既存オブジェクトを更新するための情報を求めるプロンプトを出す JSP

## SupplierServlet

iDrink SupplierServlet は、iDrink サプライヤーが、iDrink に提供された製品を管理するために必要とする機能を提供します。次の機能がサプライヤーに提供されます。

- iDrink に提供する製品の詳細情報の表示
- iDrink が製品リストに追加するために選択できる新規製品の追加
- 製品情報の編集 (iDrink がサプライヤーからまだ製品を購入していない場合)
- サプライヤーの支店および連絡先の検索
- サプライヤーの支店および連絡先の詳細情報の表示

SupplierServlet およびそれによって呼び出される JSP は、UserServlet と連動して、SupplierServlet に関連するページに、サプライヤーだけがログオンおよびアクセスできるようにする機能性を提供します。

### 設計のポイント

- EmployeeServlet セクションでも示しているように、EmployeeServlet および SupplierServlet で必要とされるメソッドおよび JSP の多くは類似したものです。コードの重複を避けるために、これらのメソッドおよび JSP は一度だけコード化されました。共通するメソッドは、EmployeeServlet でコード化されました。詳細については、EmployeeServlet セクションを参照してください。
- サプライヤーは、iDrink で現在生産段階にある製品は編集できません。PRODUCT テーブルにある製品は、iDrink での生産段階に入っています。JSP ページは、SupplierProductBean の isEditable という名前のフィールドを検査することで、これを反映します。このフィールドには、製品が iDrink の生産段階に入っている場合は真が、それ以外の場合は偽が設定されています。SupplierProductListing.jsp からの以下のコードは、このフィールドの検査方法を示しています (以下のコードで、spb は SupplierProductBean です)。

```

<td>
<%
if(spb.isEditable()) {
%>
<b><a href="SupplierServlet?action=productedit&upc=<%=spb.getUPC()%>">edit</a></b>
<%
} else {
%>
<span class="noteditable">in distribution</span>
<%
}
%>
</td>

```

- ログインしたサプライヤーが提供する製品のみを表示し、それ以外のサプライヤーが提供する製品を表示させないため、以下のように SQL 文での検索にサプライヤー ID が使用されました。

```

String queryString = "select
idrink.product.upc as pupc,
idrink.supplierproduct.suppliercontactid,
idrink.supplierproduct.upc,
idrink.supplierproduct.brand,
idrink.supplierproduct.name,
idrink.supplierproduct.size,
idrink.supplierproduct.packagetype
from idrink.supplierproduct,
idrink.suppliercontact,
idrink.supplierbranch
left outer join idrink.product
on idrink.product.upc=idrink.supplierproduct.upc
where idrink.supplierproduct.suppliercontactid=idrink.suppliercontact.suppliercontactid
and idrink.suppliercontact.branchid=idrink.supplierbranch.branchid
and idrink.supplierbranch.supplierid=?";

```

## アプリケーションのフロー

iDrink サプライヤーのインターフェースでは、サプライヤーに事前定義したパスが強制されません。したがって、サプライヤーは、あるタスクから別のタスクにいつでも簡単にジャンプできます。この柔軟性の結果、サプライヤーがサプライヤー・インターフェースを介して使用する可能性のあるパスをすべて明らかにすることは、非常に困難であると予想されます。図 10 は、iDrink サプライヤーが使用できる基本的なアプリケーション・フローを示しています。

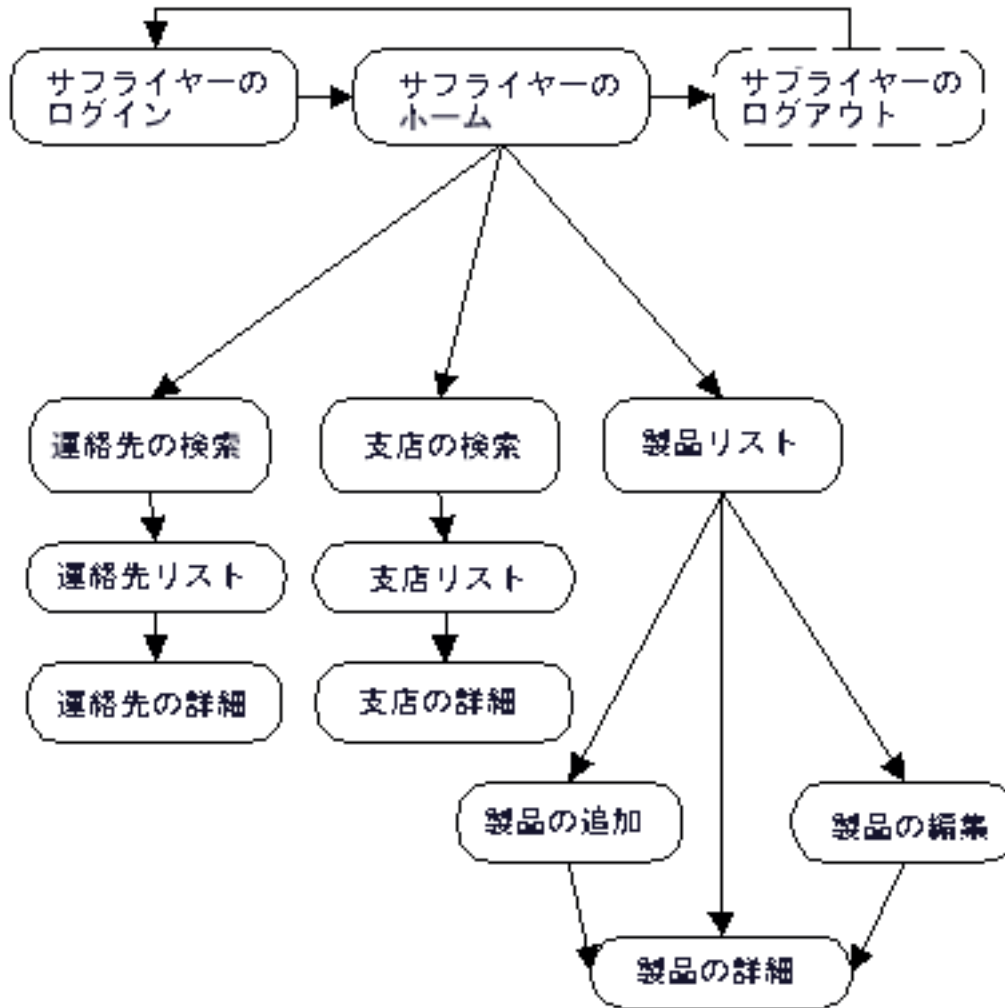


図 10: iDrink サプライヤー・アプリケーションのフロー

## メソッド

以下は、SupplierServlet にあるメソッドのリストです。doPost メソッドは、顧客からすべての着信要求を受け取り、このメソッド・リストから該当するメソッドを呼び出します。

- **addSupplierProduct** - addSupplierProduct メソッドは、フォーム/要求データに基づいて、SUPPLIERPRODUCT テーブルに新規製品を挿入します。
- **displaySupplierProductDetails** - displaySupplierProductDetails メソッドは、fetchSupplierProductDetails メソッドを呼び出して、現在の情報を使ってセッションが関連付けられている SupplierProductBean を準備します。この情報は、SupplierProductInfo.jsp に表示されます。
- **editSupplierProduct** - editSupplierProduct メソッドは、サプライヤーが提供する値でサプライヤー製品情報を更新します。
- **fetchSupplierProductDetails** - fetchSupplierProductDetails メソッドは、(情報が表示される時点と情報の更新が必要とされる時点の間で変更が行われた場合に備えて) 現在のサプライヤー製品を表示するのに必要な情報を使って、セッションが関連付けられた SupplierProductBean を準備します。
- **loadSupplierContacts** - loadSupplierContacts メソッドは、現在のサプライヤーの有効なサプライヤー連絡先リストを取得します。

- **prepareForSupplierProductAdd** - prepareForSupplierProductAdd メソッドは、loadSupplierContacts メソッドを呼び出して、現在のサプライヤー連絡先のリストを取得します。この情報は、SupplierProductAdd.jsp に表示されます。
- **prepareForSupplierProductEdit** - prepareForSupplierProductEdit メソッドは、fetchSupplierProductDetails メソッドを呼び出して、現在の情報を使ってセッションが関連付けられている SupplierProductBean を準備します。この情報は、SupplierProductEdit.jsp に表示されます。
- **searchSupplierProducts** - searchSupplierProducts メソッドは、ログインしたサプライヤーが提供する製品を取得します。この情報は、SupplierProductListing.jsp に表示されます。

## JavaServer Pages

以下は、サプライヤーが使用する JSP のリストです (\* マークが付けられた JSP は、従業員アプリケーションのフローでも使用されます)。

- **\*SupplierBranchInfo** - SupplierBranchInfo の JSP は、特定のサプライヤー支店の詳細情報を表示します。
- **\*SupplierBranchSearch** - SupplierBranchSearch の JSP は、サプライヤーが一連の支店検索基準を入力および実行依頼するために使用する入力フィールドを提供します。
- **\*SupplierBranchSearchResults** - SupplierBranchSearchResults の JSP は、指定された検索基準に一致するサプライヤー支店をすべて表示します。
- **\*SupplierContactInfo** - SupplierContactInfo の JSP は、特定のサプライヤー連絡先の詳細情報を表示します。
- **\*SupplierContactSearch** - SupplierContactSearch の JSP は、サプライヤーが一連の連絡先検索基準を入力および実行依頼するために使用する入力フィールドを提供します。
- **\*SupplierContactSearchResults** - SupplierContactSearchResults の JSP は、指定された検索基準に一致するサプライヤー連絡先のすべてを表示します。
- **SupplierHome** - SupplierHome の JSP は、サプライヤー・アプリケーション・フローのホーム・ページです。
- **SupplierProductAdd** - SupplierProductAdd の JSP は、サプライヤーが新規サプライヤー製品を作成するために使用する入力フィールドを提供します。
- **SupplierProductEdit** - SupplierProductEdit の JSP は、サプライヤーが既存のサプライヤー製品を編集するために使用する入力フィールドを提供します。
- **SupplierProductInfo** - SupplierProductInfo の JSP は、特定のサプライヤー製品の詳細情報を表示します。
- **SupplierProductListing** - SupplierProductListing の JSP は、サプライヤーが提供するすべての製品を表示します。

---

## エンタープライズ・アプリケーションのインストール

IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Application Server インスタンスでのエンタープライズ・アプリケーションのインストールは、WebSphere Studio Application Developer (WSAD)、IBM WebSphere 管理コンソールおよび IBM Web Administration for iSeries<sup>(TM)</sup> Console のウィザードによって処理されます。アプリケーションをデプロイするには、アプリケーションをまず Java<sup>(TM)</sup> 2 Platform, Enterprise Edition (J2EE) の Enterprise ARchive (EAR) ファイルにエクスポートする必要があります。EAR ファイルは、すべてのクラス・ファイル、および実動システムにアプリケーションをインストールするのに必要な追加情報を含んでおり、



WSAD のウィザードによって作成できます。EAR ファイルが作成されると、開発者は WebSphere 管理コンソールまたは IBM Web Administration for iSeries Console を使用して、実動システムに EAR ファイルをデプロイします。

## EAR ファイルへのアプリケーションのエクスポート

エンタープライズ・アプリケーションがエラーなしで正常にコンパイルされると、開発者は WebSphere Application Server インスタンスの **installableApps** ディレクトリー (¥QIBM¥UserData¥WebASE51¥ASE¥instanceName¥installableApps) に配置される EAR ファイルにアプリケーションをエクスポートします。この処理は、「IBM WebSphere Application Server - Express V5.0.2 Developer Handbook, SG24-6555」レッドブックの『Section 13.8.1 - Exporting an enterprise application to a file location』で説明しています。

**IBM WebSphere Application Server インスタンスでの EAR ファイルのデプロイ** EAR ファイルが WSAD ウィザードによって正常に作成されると、EAR ファイルは実動システムにデプロイできます。開発者は、このタスクを完了するために、WebSphere 管理コンソールまたは IBM Web Administration for iSeries Console を使用します。WebSphere 管理コンソールを介した EAR ファイルのデプロイのプロセスは、「WebSphere Application Server - Express V5.0.2 Administrator Handbook, SG24-6976」の『Section 6.4 - Installing applications』で説明しています。IBM Web Administration for iSeries Console で EAR ファイルをデプロイするには、EAR ファイルはすでにターゲット・システムにある必要があります。EAR ファイルをデプロイするには、次のステップを実行します。

- Web ブラウザーで、ターゲット・システムの IBM Web Administration for iSeries Console に移動する。URL は、*http://mysystem:2001/HTTPAdmin* です。ここで、「mysystem」はターゲット・システムの名前です。情報のプロンプトが出されたら、ターゲット・システムのユーザー ID とパスワードを入力します。
- ページの一番上にある「**管理 (Manage)**」タブを選択し、その下の「**アプリケーション・サーバー (Application Servers)**」タブを選択する。
- WebSphere Application Server のインスタンスが、「**インスタンス/サーバー (Instance/Server)**」メニューで選択されていることを確認する。サーバー・インスタンスが停止している場合は、メニューの左側にある緑のスタート・ボタンをクリックし、サーバー・インスタンスを始動します。サーバー・インスタンスが稼働中になったら、次のステップに進み、アプリケーションのインストールを開始します。
- 左のフレームで、「**新規アプリケーションのインストール (Install New Application)**」リンクをクリックする。
- 「**アプリケーションのロケーションを指定 (Specify Application Location)**」画面で、「参照」をクリックする。ファイル・ブラウザー・ウィンドウで、EAR ファイルを選択し、「OK」ボタンをクリックします。「**アプリケーションのロケーションを指定 (Specify Application Location)**」画面で、「次へ」ボタンをクリックします。
- 「**インストールを実行するオプションを指定 (Provide Options to Perform Install)**」画面で、「**アプリケーション名 (Application Name)**」テキスト・フィールドにアプリケーション名を入力し、JSP をプリコンパイルする場合は、「**JSP のプリコンパイル (Pre-compile JSPs)**」チェック・ボックスにチェック・マークを付ける。「次へ」ボタンをクリックします。
- 「**Web モジュールの仮想ホストをマップ (Map Virtual Hosts for Web Modules)**」画面で、「**仮想ホスト (Virtual host)**」メニューにある **default\_host** を選択する。「次へ」ボタンをクリックします。
- 「**要約**」画面で、この情報が正しいかを確認し、「完了」ボタンをクリックする。アプリケーションがインストールされるまで数分かかる場合があります。



---

## WebSphere ディスカバリー

- WebSphere Application Server - Express for iSeries<sup>(TM)</sup> バージョン 5.1 以前のバージョンでは、JSP は、デフォルトのパッケージからクラスをインポートできました。しかし、WebSphere Application Server - Express for iSeries バージョン 5.1 の段階で、JSP は、デフォルトのパッケージからクラスをインポートできなくなりました。当初は、iDrink サブレットおよび JavaBeans<sup>(TM)</sup> は、デフォルトのパッケージに配置されていました。これらを非デフォルトのパッケージに移行するには、WebSphere Studio Application Developer で以下のステップを実行します。
  - 新規パッケージを作成し、Java ソース・ファイルをその新規パッケージに追加します (WebSphere Studio Application Developer は、移行時に、該当する「パッケージ」文をすべてのソース・ファイルに追加します)。
  - コードが Bean をインスタンス化する場合、常にパッケージ名には Bean 名が含まれます。以下に例を示します。

```
Beans.instantiate(getClass().getClassLoader(), "packageName.beanName");
```
  - JSP が Bean をインポートする場合、常にパッケージ名には Bean 名が含まれます。以下に例を示します。

```
<%@ page import="packageName.beanName" %>
```
  - パッケージ名を含むサブレットをポイントするように、web.xml ファイルのサブレット・リストを更新します。WebSphere Studio Application Developer が、移行時に、該当するパッケージ文を各ソース・ファイルに追加しても、新規パッケージのサブレットをポイントする web.xml ファイルは更新されません。GUI エディターを使用する場合、リストからサブレットを除去してから、追加し直します。「サブレット・クラス (Servlet class)」フィールドには packageName.servletName が含まれます。

### 参照

- 「WebSphere Application Server - Express V5.0 for iSeries」IBM<sup>(R)</sup> Redpaper (REDP-3624-00)
- 「WebSphere J2EE Application Development for the IBM eServer<sup>(TM)</sup> iSeries Server」IBM Redbook (SG24-6559-00)
- 「WebSphere Development Studio Client for iSeries V5.0」IBM Redbook (SG24-6961-00)
- 「WebSphere Studio Application Developer Programming Guide」IBM Redbook (SG24-6585-00)



---

## 第 3 章 Lotus Domino 環境

iDrink 社は、従業員が機密広告を参照および保存できる Web サイトを提供することを望んでいました。この Web サイト Cola Connections は、eServer<sup>(TM)</sup> i5 上の Lotus Domino バージョン 6.5 を使用して作成されました。

Lotus Domino では、iDrink 社は、機密広告の動的 Web サイトを素早く作成することができました。加えて、Lotus Domino により、iDrink 社はワークフロー処理を利用できるようになり、また他の数多くのプラットフォームとの相互運用性がもたらされました。

Cola 接続アプリケーションは、さまざまなフォームやビューを使用して、広告情報を作成および保守します。広告情報は、従業員による販売のためにリストされた機密広告で構成されています。従業員データを安全に管理するために、i5/OS<sup>(TM)</sup> Lightweight Directory Access Protocol (LDAP) ディレクトリーの参照には、Lotus Domino Directory Assistance データベースが使用されています。

iDrink 社は、Lotus Domino フレームセットを使用して Web インターフェースを提供しています。Cola Connections Web サイトを介して、iDrink 従業員は以下のタスクを実行できます。

- 機密広告の表示
- 機密広告の作成
- 機密広告を売却済みとしてマーク付けするオプションによる既存の機密広告の編集

---

### OS/400 LDAP を使用する Lotus Domino の構成

セキュリティーが Cola Connections Web サイトの要件であるため、Lotus Domino は、ユーザー認証に i5/OS<sup>(TM)</sup> LDAP サービスを使用するように構成されました。デフォルトでは、Domino は自分自身の LDAP 機能を使用します。Domino が、自分自身の LDAP 機能の代わりに i5/OS LDAP サービスを使用するには、ディレクトリー・アシスタンス・データベースを作成する必要があります。

ディレクトリー・アシスタンス・データベースをセットアップする方法については、レッドブック「*Integrating Lotus Domino 6 and WebSphere<sup>(R)</sup> Express V5 on the IBM<sup>(R)</sup> eServer<sup>(TM)</sup> iSeries<sup>(TM)</sup> Server (SG24-6998-00)*」のセクション『4.4.1 Creating a Directory Assistance database』を参照してください。

これらのヒントは、レッドブックに記載されている指示に従うとき、Cola Connections のディレクトリー・アシスタンス・データベースのセットアップに有効です。

- ステップ 8 では、「**基本 (Basics)**」タブのフィールドに入力する際、Domino サーバーが使用するドメイン以外の任意の名前としてドメイン名を指定できます。例えば、iDrinks の Domino サーバーは、iDrink ドメイン内に構成されていました。そのため、「**基本 (Basics)**」タブで iDrink はドメイン名として指定できませんでした。
- ステップ 10 では、「**LDAP**」タブのフィールドに iDrink の LDAP セットアップに基づいて入力する際、フィールド「使用する検索フィルターのタイプ (Type of search filter to use)」には「カスタム」を指定し、「カスタマイズされたフィルター (Customized filters)」セクションでは、以下を「許可フィルター (Authorization Filter)」フィールドに追加します。

```
(|(&(objectclass=groupofuniquenames)(UniqueMember=*))(&(objectclass=groupofnames)(objectclass=AccessGroup))(member=*))
```

この検索フィルターによって、Domino で LDAP グループを検索できるようになります。次に、従業員グループだけがアクセスできるように、Domino データベースでアクセス制御リスト (ACL) が設定されます。すべての従業員はこのグループの一部であるため、Cola 接続アプリケーションにアクセスすることができます。

セクション『4.4.2 *Configuring Domino to use OS/400 LDAP*』から章の終わりまでの指示に従って、Domino サーバーの構成を続行してください。

Lotus Domino Database で ACL を構成するには、セクション『5.2.3 *Updating Domino ACLs for adding users registered in OS/400 LDAP*』の指示に従ってください。例えば、Cola Connections データベースでは、iDrink LDAP スキーマから `cn=employees/o=idrink/dc=domainName/dc=domainSuffix` という個人グループ ACL を使用します。Domino がグループを LDAP スキーマで検索できるように、上記のようにスラッシュを入力する必要があります。

---

## アプリケーションの詳細

Cola Connections Web サイトから、iDrink 従業員は、いくつかのタスクを実行できます。iDrink 従業員は、「新規広告 (New Ad)」ボタンを押して、広告を作成することができます。この新規文書は一度作成されると、更新、売却済みとしてマーク付け、または表示が可能です。

---

## アプリケーション設計のポイント

Lotus<sup>(R)</sup> Domino<sup>(R)</sup> アプリケーションの設計時に、iDrink 社は、Web インターフェースとして、フレームセットとナビゲーターのどちらを使用するかを選択する必要がありました。以下は、フレームセットとナビゲーターの差異をリストしたものです。

- フレームセットは、フォーム、フォルダー、ページ、文書、ビュー、ナビゲーター、またはフレームセットを含むことができます。また、フレームは Web ページを含み、特定の URL と関連付けることができます。フレーム間のリンクおよび関係は、フレームセットと共に作成することができます。
- ナビゲーターは、ほとんどのユーザーが情報を見つけやすくするために、フォルダー、ビューおよび設計エレメントのグラフィック表示を提供します。

iDrink 社では、Web サイト全体を通じて、さまざまなフォームやビューを表示可能な一貫性のある構造を提供したいと考えており、フレームには、フォーム、フォルダー、ページ、文書、およびビューを含めることができることから、フレームセットが最適な選択であるとされました。

また、iDrink 社には、データ保護の最適な方法を探して実装するタスクも必要でした。Cola Connections セキュリティ要件には、次のものがあります。

- 信頼性、デプロイメントの速度、および管理の容易性を実現する堅固なアーキテクチャー
- 既存のアプリケーションと統合しやすいソリューション
- 他のソフトウェア・パッケージと統合可能なソリューション

これらの要件によって、iDrink 社が i5/OS<sup>(TM)</sup> LDAP データベースを参照する Domino Server 上に、ディレクトリー・アシスタンス・データベースを実装することができます。

iDrink 社には、Java<sup>(TM)</sup> 開発のスキルを持つ従業員が数名在籍していたため、同社ではアプリケーションのバックグラウンド・エージェントを Java で作成しました。このことは、既存のスキルとリソースを使用するという観点で、価値があると判明しました。

図 11 は、Cola 接続アプリケーションのフローを示します。

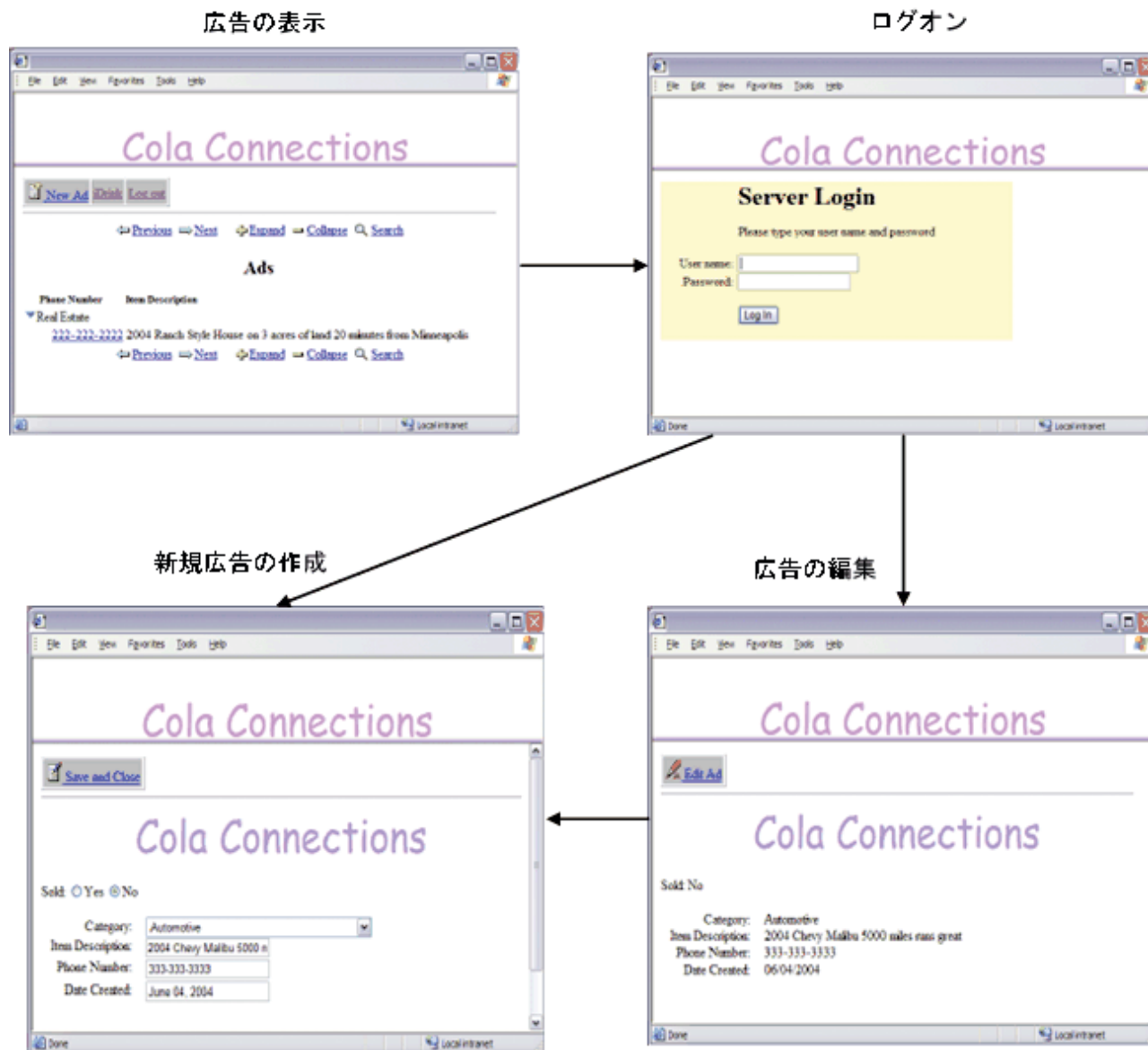


図 11: Cola 接続アプリケーションのフロー

## アプリケーションのセットアップ

Lotus<sup>(R)</sup> Domino<sup>(R)</sup> Designer<sup>(R)</sup> が、Lotus Domino 実装作業のための開発ツールとして使用されました。Lotus Domino Designer は、Lotus Domino アプリケーションを素早く作成および変更するために使用できる、Lotus<sup>(R)</sup> クライアント・アプリケーションです。このアプリケーションは、フォーム、ビュー、およびエージェントなど、すべてのデータベース・エレメントにアプリケーション構築ブロックを提供します。フォームは、データベース内で新規文書を作成し、現行の文書を表示するために iDrink で使用されました。ビューを使用すると、柔軟かつ直感的な方法で文書を作成することができます。ユーザーは、素早く文書を作成、ソート、表示、および編集することができます。

## Lotus Domino のフォームおよびビュー

iDrink 社の従業員は、Lotus Domino アプリケーションの広告情報フォームと、広告およびすべての広告ビューを作成しました。

広告情報ビューには、iDrink 従業員によって作成されたすべての使用可能な広告のリストが含まれています。広告情報フォームは、新規広告を作成したり、既存の広告を表示するのに使用されます。

広告およびすべての広告ビューは、さまざまな広告を表示し、自動車、コンピューター/ホーム・オフィス、および不動産などのカテゴリーごとにソートされます。

図 13 は、広告情報フォームのフィールドを示しています。

名前	フィールド名	型	説明
カテゴリー	Category	ダイアログ・リスト	広告のカテゴリー。可能なカテゴリーは次のとおり。自動車、他の車、ボート/RV/スノー・モービルおよび他の車両、衣服、コンピューター/ホーム・オフィス、世帯、子供、その他、バイク、音楽、写真/ステレオ/TV、スポーツ用品、不動産、取引、求人、懸賞
品目の説明	ItemDescription	テキスト	広告の品目の説明。字数制限は 100 文字
電話番号	PhoneNumber	テキスト	品目を販売する従業員の電話番号
作成日	DateCreate	日時	広告の作成日を示すフィールドを自動的に生成
売却済み	Sold	ラジオ・ボタン	品目が売却されたかを示す。指定可能な値は、「はい」または「いいえ」です。

表 13: 広告情報フォームのフィールド

## Lotus Domino エージェント

Lotus Domino エージェントは、作業を自動化するために Lotus Domino データベースに追加される設計要素です。エージェントは、ユーザー・アクションによって開始したり、スケジュール・ベースで実行することができます。エージェントは、一般に、文書を更新および作成したり、データベースにアクセスしたりする際に使用されます。Lotus Domino エージェントは、Java<sup>(TM)</sup>、LotusScript、または Formula Language で作成することができます。

エージェントの作成には、Lotus Domino Designer<sup>(R)</sup> が必要です。エージェントが作成される時点で必要になる決定事項には、エージェントをいつ実行すべきか、エージェントを作成する言語は何か、およびエージェントがどの文書の下で実行されるか、などがあります。エージェントを作成し、コンパイルすると、特定の時刻に実行するよう自動的にスケジュールされます。Lotus Domino Designer の組み込みデバッグ機能は、エージェントの作成時に役立ちます。

Lotus Domino の Cola 接続アプリケーションには、Delete Ads という 1 つのエージェントが含まれています。この Java エージェントは、スケジュール・ベースで実行されます。購入状況が「はい」の場合の広告文書や 1 カ月以上古い広告文書は削除されます。

Delete Ads エージェントのコード断片は、以下のようになります。

```
private void deleteAd(AgentContext agentContext, Session session) {  
  
    Vector adDateVector;  
    DateTime adDateTime = null;  
    Date adDate = null;  
    Date currentDate = null;  
    int timeBetween = 0;  
    float timePassed = 0;  
    try {  
        Database db = agentContext.getCurrentDatabase();  
        DateTime currentDateTime = session.createDateTime("Today"); // get the current date  
        currentDateTime.setNow(); // get all documents where the Sold Status is Yes from All Ads  
        DocumentCollection adInformationDC = db.search("SELECT ((Form = ¥\"Ad Information¥\"))");  
        Document adDoc = adInformationDC.getFirstDocument(); // get first ad document  
        while (adDoc != null) // while there are ad documents  
            String adStatus = adDoc.getItemValueString("Sold"); // obtain the value of the Sold variable  
            adDateVector = adDoc.getItemValue("DateCreate"); // obtain value of create date for ad  
            adDateTime = (DateTime)adDateVector.elementAt(0); // obtain value of create date for ad  
            timeBetween = currentDateTime.timeDifference( adDateTime); // returns difference in seconds  
            timePassed = timeBetween/2629744; //divide by seconds in a month  
            Document adDoc1 = adInformationDC.getNextDocument(); // get the next ad document  
            if ((adStatus.equals("Yes")) || (timePassed > 1)) // if item is Sold, remove the document  
                adDoc.remove(true);  
            adDoc = adDoc1; // get the next ad to process  
        } // end while adDoc is not null }  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
} // end delete ads
```

---

## Lotus Domino ディスカバリー

以下は、Cola Connections シナリオの作成中に明らかになった主なディスカバリーのリストです。

- 最初のセットアップの後、iDrink 従業員は、新規広告の作成時に、ログインするプロンプトが出されませんでした。時々、「保管してクローズ (Save and Close)」ボタンが押されるまで、ログイン・プロンプトが表示されませんでした。「新規広告 (New Ad)」ボタンを押しているときに従業員にログインを求めようとするため、新規広告操作に次の Formula Language コマンドが設定されました。

```
@URLOpen(@WebDbName + "/Ad+Information?OpenForm&login")
```

- iDrink 社は、従業員が Cola 接続アプリケーションからログアウトする方法を実装しようとしていました。結果的に、ログアウト機能が組み込まれました。しかし、ログアウト・ボタンを押しても、従業員は Cola Connections ホーム・ページにリダイレクトされませんでした。ホーム・ページにリダイレクトできるようにするために、次の Formula Language コマンドがログアウト操作に設定されました。

```
@SetTargetFrame(" top");  
@URLOpen("http://nenotes:2500/ColaConn.nsf?logout&RedirectTo=http://NotesServerName:2500/ColaConn.nsf/Cola%20Connections?OpenFrameSet")
```

### 参照

- IBM<sup>(R)</sup> Lotus Domino for iSeries<sup>(TM)</sup> - OS/400<sup>(R)</sup> Web サイト  
<http://www.ibm.com/servers/eserver/series/domino>
- IBM Lotus Domino for iSeries (PartnerWorld for Developers)  
<http://www.ibm.com/servers/eserver/series/developer>



- Lotus Web サイト  
<http://www.lotus.com>







Printed in Japan