

入門ガイド

iPlanet™ Application Server

Version 6.5

816-5270-01
2002 年 2 月

Copyright © 2002, Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

本製品は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。Sun および Sun のライセンサーの書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

フォントテクノロジーを含む第三者のソフトウェアの著作権は Sun の提供者により保護されており、ライセンス許諾されています。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、iPlanet、および iPlanet のロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc.(以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC の商標はライセンスに基づいて使用され、米国およびその他の国における SPARC International, Inc. の商標もしくは登録商標です。SPARC の商標に関連する製品は Sun Microsystems, Inc. によって開発されたアーキテクチャに基づいています。

UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

この製品には Apache Software Foundation (<http://www.apache.org/>) により開発されたソフトウェアが含まれています。Copyright © 1999 The Apache Software Foundation. All rights reserved.

Federal Acquisitions: Commercial Software - Government Users Subject to Standard License Terms and Conditions.

目次

| | |
|---|-----------|
| はじめに | 7 |
| このマニュアルについて | 7 |
| 前提事項 | 8 |
| このマニュアルの構成 | 8 |
| マニュアルの表記規則 | 9 |
| | |
| 第 1 章 iPlanet Application Server の概要 | 11 |
| アプリケーションサーバとは | 11 |
| iPlanet Application Server の機能 | 13 |
| J2EE プラットフォーム | 14 |
| 業界標準コンポーネント | 14 |
| 高いスケーラビリティ | 15 |
| アプリケーションパーティショニング | 15 |
| 分散データ同期 (DSync) | 16 |
| 高いパフォーマンス | 17 |
| ダイナミックロードバランス | 18 |
| データベースコネクションプーリング | 18 |
| 結果キャッシュ | 19 |
| データストリーミング | 20 |
| マルチスレッド機能 | 20 |
| EJB セッションプーリング | 21 |
| Web サーバとの最適な通信 | 21 |
| 高い可用性 | 21 |
| アプリケーション開発期間の短縮 | 22 |
| 多層アプリケーションモデル | 22 |
| コアサービス | 24 |
| アプリケーション開発ツール | 24 |

| | |
|--|-----------|
| セッションおよびステートの管理 | 25 |
| セキュリティ | 25 |
| ユーザ認証 | 26 |
| データソースへの安全なアクセス | 26 |
| 製品コンポーネント | 26 |
| プログラミング API | 27 |
| システムサービスとアプリケーションサービス | 27 |
| サンプルアプリケーション | 27 |
| コアアプリケーションサーバコンポーネント | 28 |
| Web Connector プラグイン | 28 |
| iPlanet Application Server Administration Tool | 28 |
| iPlanet Application Server Deployment Tool | 29 |
| iPlanet Directory Server | 29 |
| 第 2 章 アプリケーションの展開 | 31 |
| PointBase データベースサーバの起動 | 31 |
| iPlanet Application Server の停止と起動 | 32 |
| Windows でサーバを停止して起動するには | 32 |
| Solaris でサーバを停止して起動するには | 35 |
| データソースの登録 | 36 |
| Solaris と Windows にデータベースを登録するには | 36 |
| アプリケーションのビルド | 37 |
| アプリケーションをビルドするには | 37 |
| アプリケーションの表示 | 38 |
| アプリケーションの展開の解除 | 39 |
| 第 3 章 管理ツールと配置ツール | 41 |
| iPlanet Application Server Administration | 41 |
| 基本管理タスク | 42 |
| サーバを起動するには | 42 |
| コマンドラインツール | 42 |
| コマンドラインツールの使用 | 42 |
| サーバを起動して停止するには | 43 |
| アプリケーションの展開 | 43 |
| GUI ベースのツールを使うには | 43 |
| コマンドラインの Deployment Tool にアクセスするには | 44 |
| アプリケーションメッセージの記録 | 44 |
| ログメッセージの書式設定 | 44 |
| ログ送信先の決定 | 45 |
| iPlanet Directory Server の管理 | 45 |
| サードパーティ管理ツールのサポート | 46 |

| | |
|--|-----------|
| 第 4 章 iPlanet Application Server の問題と対処方法 | 47 |
| インストール関連の問題 | 47 |
| コアサーバ関連の問題 | 48 |
| Administration Tool 関連の問題 | 48 |
| 展開に関する問題 | 49 |
| データベースサポート関連の問題 | 50 |
| | |
| 第 5 章 その他の資料 | 51 |
| | |
| 付録 A iPlanet Application Server のアーキテクチャ | 55 |
| サーバプロセス | 55 |
| プロセス対話の要約 | 55 |
| Executive Server | 57 |
| Administration Server | 58 |
| Java サーバおよび C++ サーバ | 58 |
| システムコンポーネント | 59 |
| プロトコルマネージャ | 59 |
| ロードバランスシステム | 61 |
| ロードモニタ | 61 |
| ロードバランサ | 61 |
| リクエスト管理システム | 62 |
| アプリケーションコンポーネント | 62 |
| アプリケーションサービス | 63 |
| KJS または KCS で管理するサービス | 63 |
| KJS だけが管理するサービス | 65 |
| システムサービス | 65 |
| トランザクション管理システム | 67 |
| ローカルトランザクションとグローバルトランザクション | 67 |
| アーキテクチャの詳細 | 67 |
| セキュリティ | 68 |
| 管理サービス | 69 |
| | |
| 索引 | 71 |

はじめに

この章では『iPlanet Application Server 入門ガイド』の内容について説明します。この章には次の節があります。

- このマニュアルについて
- 前提事項
- このマニュアルの構成
- マニュアルの表記規則

このマニュアルについて

この『入門ガイド』では iPlanet Application Server のさまざまな機能とサンプルアプリケーションを展開する方法について説明します。また、iPlanet Application Server のアーキテクチャの概要についても説明します。

このマニュアルは、iPlanet Application Server で利用できるさまざまなタスクやツールの理解を必要とするシステム管理者、ネットワーク管理者、評価者、アプリケーションサーバ管理者、Web 開発者、およびソフトウェア開発者を対象にしています。

前提事項

このマニュアルは、次のトピックに関する知識があることを前提とします。

- アプリケーションサーバ
- クライアント/サーバプログラミングモデル
- インターネットおよび WWW (World Wide Web)
- Windows NT/2000 または Solaris™ オペレーティングシステム
- Java プログラミングおよび J2EE

このマニュアルの構成

このマニュアルの構成は次のとおりです。

第 1 章「iPlanet Application Server の概要」では、iPlanet Application Server の機能とコンポーネントの概要について説明します。

第 2 章「アプリケーションの展開」では、iPlanet Application Server にサンプルアプリケーションを展開する手順について説明します。

第 3 章「管理ツールと配置ツール」では、アプリケーションの管理および展開に利用できるさまざまなツールについて説明します。

第 4 章「iPlanet Application Server の問題と対処方法」では、一般的なエラーとその対処方法について説明します。

第 5 章「その他の資料」には、iPlanet Application Server のほかのマニュアルやリソースに関する参照資料があります。

付録 A「iPlanet Application Server のアーキテクチャ」では、J2EE の概念と iPlanet Application Server のアーキテクチャについて説明します。

マニュアルの表記規則

ファイルとディレクトリのパスは、Windows の形式で表記されます (ディレクトリ名を円記号で区切って表記)。UNIX バージョンでは、ディレクトリパスについては Windows と同じですが、ディレクトリの区切りには円記号ではなくスラッシュが使われます。

このマニュアルでは、`http://server.domain/path/file.html` のような URL 形式を使います。ここで、

- `server` は、アプリケーションを実行しているサーバの名前です。
- `domain` は、インターネットのドメイン名です。
- `path` は、サーバ上のディレクトリ構造です。
- `file` は、個々のファイル名です。

次の表は、iPlanet マニュアルで採用しているフォントの規約を示します。

表 1 フォントの規約

| 書体 | 意味 | 例 |
|--------|---|--|
| モノスペース | ファイル名、ディレクトリ、サンプルコード、コードの一覧表示、および HTML タグ | <p>Hello.html ファイルを開きます。</p> <p><HEAD1> は、最上位の見出しを作成します。</p> |
| イタリック | 変数、コードのプレースホルダ、およびリテラルに使われる語句 | <p>ユーザ <i>ID</i> を入力します。</p> <p>名前のフィールドに「<i>Login</i>」と入力します。</p> |
| 太字 | テキストに初めて登場した用語 | <p>テンプレートは、ページのアウトラインです。</p> |

iPlanet Application Server の概要

この章には次のトピックがあります。

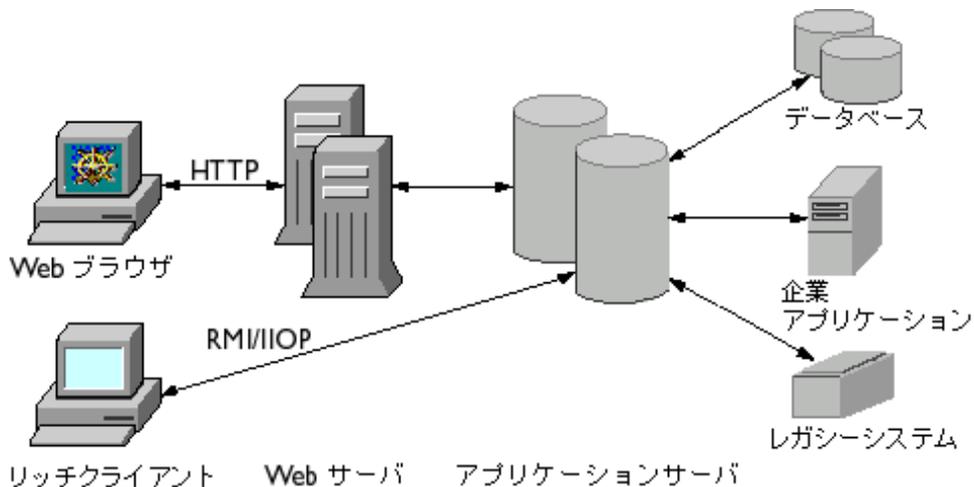
- アプリケーションサーバとは
- iPlanet Application Server の機能
- 製品コンポーネント

iPlanet Application Server はパフォーマンス、スケーラビリティ、可用性を基本に設計されています。このためユーザが大幅に増加したり、電子商取引の Web サイトやインターネットを利用している企業での処理が大幅に増えても、iPlanet Application Server なら対応することができます。

アプリケーションサーバとは

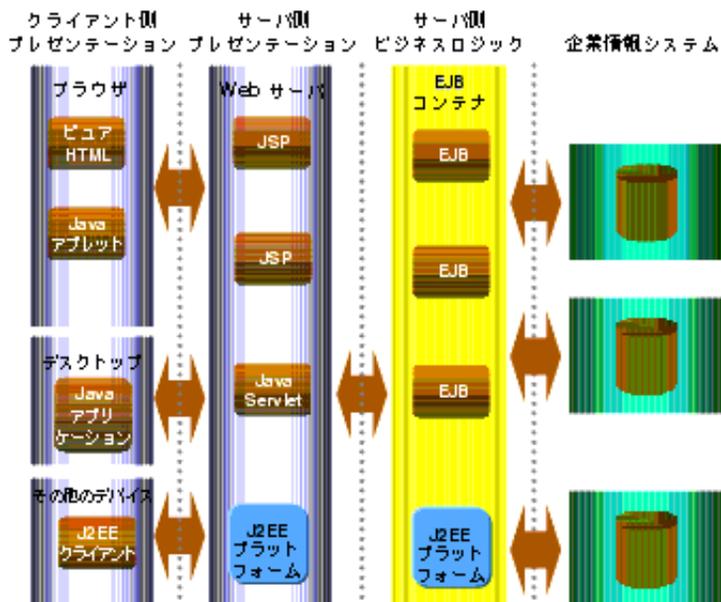
アプリケーションサーバは、クライアントからバックエンドリソースへの接続、アプリケーションロジックの実行、およびクライアントへの結果の通知のためのフレームワークを提供します。アプリケーションサーバは 3 層処理モデルの中間層に位置しています。

図 1-1 インターネット上のアプリケーションサーバ空間



もっとも一般的なアプリケーションサーバは Java™ アプリケーションサーバです。iPlanet Application Server は Java アプリケーションサーバで、Java 2 Enterprise Edition (J2EE™) 仕様書に完全に準拠しています。J2EE は完全で安全な基盤を提供し、セキュリティ、開発、展開、コードの再利用、移植性の標準セットが数多く記述されているので、企業は移植可能でベンダに依存しないアプリケーションを作成することができます。

図 1-2 J2EE プラットフォームベースのアプリケーションサーバコンポーネント



iPlanet Application Server の機能

iPlanet Application Server 6.5 には次のような機能があり、現在利用できるアプリケーションサーバの中でもっともスケーラブルかつ堅牢で、パフォーマンスの高いものの1つです。

- J2EE プラットフォーム
- 高いスケーラビリティ
- 高いパフォーマンス
- 高い可用性
- セッションおよびステートの管理
- アプリケーション開発期間の短縮
- セキュリティ
- 製品コンポーネント

J2EE プラットフォーム

iPlanet Application Server は Java 2 Enterprise Edition (J2EE) に完全に準拠したアプリケーションサーバプラットフォームを提供します。J2EE アプリケーションモデルを使うと、開発者は下位レベルの処理を J2EE コンポーネントに任せてビジネスロジックに集中することができます。これによりアプリケーションとサービスを簡単に拡張してすぐに展開することができ、競争の激しい変化にすばやく対応することができます。

J2EE プラットフォームを介してオープンな標準アーキテクチャを提供することにより、iPlanet Application Server は、スケーラブルで可用性が高く、安全で信頼できる多層サービスの開発にかかるコストと複雑さの問題を解決します。詳細については、55 ページの「iPlanet Application Server のアーキテクチャ」を参照してください。

業界標準コンポーネント

アプリケーションの開発は次の業界標準 Java に基づいて行います。iPlanet Application Server 6.5 がサポートする標準とコンポーネントは次のとおりです。

- JDK 1.3 仕様
- Java Servlet 2.2 仕様
- Enterprise JavaBeans 1.1 仕様
- JavaServer Pages 1.1 仕様
- JDBC 2.0 コア仕様
- JDBC 2.0 標準拡張仕様
- JTA 1.0 仕様
- JNDI 1.2 仕様
- RMI-IIOP 1.0
- JavaMail 1.0
- アプリケーションのアセンブリと展開 (XML)
- HTML
- XML
- LDAP
- SNMP AGENTS
- XA

高いスケーラビリティ

iPlanet Application Server は、プログラムを手直しすることなく、拡大するビジネスの需要に対応できるスケーラブルなアーキテクチャを持っています。開発者はユーザーベースが拡大してもアプリケーションロジックを変更する必要はありません。たとえば、サーバクラスタが3つある企業に、iPlanet Application Server を稼働させたまま50台のサーバをダイナミックに追加することができます。リソースを追加するとトランザクションとリクエストの数の増加が可能になり、パフォーマンスが向上します。

アプリケーションの拡張は、主にサーバクラスタにサーバを増設するか、またはマルチCPUシステムにCPUを追加することによって達成できます。次にアプリケーションロジックを新しいサーバに展開します。分散アーキテクチャの本質を考えると、拡張とはiPlanet Application Server への統合であり、ビジネスソフトウェアは物理的に離れた場所にある複数のサーバで実行されます。通常、それぞれのサーバにはアプリケーションのコピーがあり、クラスタのどのサーバでモジュールを実行するかは管理エンジンが実行時に決定します。アプリケーションタスクは、リクエストをもっとも効率的に処理できるサーバに割り当てることができます。

iPlanet Application Server のアーキテクチャはスケーラブルであり、次のどちらも効率的に行うことができます。

- アプリケーションパーティショニング
- 分散データ同期 (DSync)

アプリケーションパーティショニング

iPlanet Application Server アーキテクチャはアプリケーションパーティショニングをサポートしています。この機能を使ってアプリケーションロジックを複数のサーバに分散することにより、アプリケーションを拡張して負荷の増加に対応することができます。

アプリケーションロジックをグループ化して、それぞれのグループを関連のある操作でまとめることができます。アプリケーションコンポーネントを1つまたは複数のグループに組み入れることができます。アプリケーションはアプリケーションロジックを共有することもできます。システム管理者は、iPlanet Application Server Administrator Tool のダイナミックな「ポイントアンドクリック」パーティショニング機能を使って次のことができます。

- コンポーネントの負荷をリアルタイムで監視する
- 負荷監視パラメータをただちに変更する
- アプリケーションコンポーネントを管理する
- クラスタ内の任意のマシンまたは全マシンにコンポーネントを展開する
- コンポーネントやマシンの適性に合わせて展開する

- コンポーネントを無効にし、すぐに利用できる状態にしておく

オーダが登録されると、オーダ処理コンポーネントは必ずサーバ 1 から呼び出されませんが、チェックアウト関数は 3 つのサーバのどれからでも呼び出すことができます。アプリケーションのパーティションや分散の方法にかかわらず、アプリケーションは 1 つのまとまりとして機能します。

アプリケーションオブジェクトをグループ化して、それぞれのグループを関連のある操作でまとめることができます。たとえば、あるグループにオーダ処理に関する EJB がすべて含まれている場合、アプリケーションコンポーネントを 1 つまたは複数のグループに組み入れることができます。アプリケーションはアプリケーションロジックを共有することもできます。

システム管理者は、次の方法を使って、アプリケーションオブジェクトのグループをアプリケーションサーバにローカルまたはグローバルに展開することができます。

- アプリケーションの一部が別々の iPlanet Application Server に独自に存在しているのに、単一のアプリケーションとして動作していることがあります。この場合、アプリケーションオブジェクトはもっとも効率的に実行できるサーバに格納することができます。たとえば、データをよく利用するアプリケーションオブジェクトは、データソースにもっとも近いサーバで実行すると、離れた場所のデータにアクセスすることによる遅延を避けることができます。
- ロードバランスの取れたアプリケーションでは、同じアプリケーションオブジェクトのグループを複数のサーバに格納することができます。これにより、もっともよく利用するリソースを持つサーバでアプリケーションロジックを効率的に実行できます。
- アプリケーションが特定のアプリケーションロジックのオブジェクトをダイナミックに共有することがあります。たとえば、ネットワーク上のすべてのアプリケーションは同じアプリケーションロジックを共有して、ユーザのログインと認証、またはクレジットカードの認可を行います。

アプリケーションパーティショニングを使うと、システム管理者はアプリケーションパフォーマンスの拡張とチューニングを驚くほど柔軟に行うことができます。さらに、複数のサーバにアプリケーションコンポーネントを格納しておくことで、サーバのシャットダウン時にアプリケーションの高度な可用性が保証されます。

分散データ同期 (DSync)

分散データ同期により、プロセスでのクラスタ管理とデータ同期が提供され iPlanet Application Server のスケーラビリティを補助します。

クラスタは iPlanet Application Server サーバのセットで、セッション (クライアントの対話情報) とステート (データベースのオープンデータ、変数) のデータを 1 つのソースで共有します。クラスタは共有セッションとステートデータの統合を管理します。クラスタは、アプリケーションがセッションとステートデータを共有する必要があっても iPlanet Application Server を実行する必要がない場合に使います。

クラスタの使用中は、ほかのサーバから Sync Primary にアクセスしてセッションとステートデータを検索します。

クラスタは、管理ツールで簡単に管理し追加できます。DSync を使うと、クラスタの分散セッション情報 (ユーザ ID など)、ステートデータ (株価など)、およびステートフルセッション (ショッピングカートなど) を利用できます。サーバに障害が発生した場合も、データを失うことなく Sync Backup 上で情報を利用できます。

DSync とクラスタ管理は、iPlanet Application Server のスケーラビリティと可用性に関する機能です。DSync はサーバの可用性においても重要な役割を果たしています。21 ページの「高い可用性」を参照してください。DSync に関する詳細は、『iPlanet Application Server 管理者ガイド』で説明されています。

高いパフォーマンス

iPlanet Application Server は、パフォーマンスの高い、マルチスレッド、マルチプロセスのアプリケーションサーバです。iPlanet Application Server は多数の同時リクエスト、データベースコネクション、およびセッションを処理し、高負荷の環境でも高いパフォーマンスを提供します。

iPlanet Application Server には次のような機能があり、Web サーバ、ほかの iPlanet Application Server マシン、異種のバックエンドデータソースとの間で高いパフォーマンスを提供します。

- ダイナミックロードバランス
- データベースコネクションプーリング
- 結果キャッシュ
- データストリーミング
- マルチスレッド機能
- EJB セッションプーリング
- Web サーバとの最適な通信

アプリケーションサーバ以外にも、ネットワークトポロジ、ネットワークとサーバのハードウェア、データベースアーキテクチャ、およびアプリケーションプログラミングなどがアプリケーションのパフォーマンスに影響を与えます。これらはこのマニュアルの範囲外になるので、ここでは説明しません。

ダイナミックロードバランス

iPlanet Application Server は、高負荷環境で最適なパフォーマンスレベルを提供するダイナミックロードバランスをサポートします。ロードバランスは、そのときのコンポーネントのリクエストをもっともうまく処理できるシステムで、それぞれのアプリケーションコンポーネントを処理して最適なパフォーマンスにするようにします。

ロードバランスはロードモニタが取得したロードとパフォーマンスの情報を使って、送られてくるリクエストを処理するのに最適なリソースを持つサーバを決定します。iPlanet Application Server の各インスタンスには、ロードバランスのプロパティに基づいてルートを決する独自のロードバランスモジュールがあります。

ロードバランスはサーバで管理するか、プラグインで管理します。ロードバランスを有効にすると、iPlanet Application Server では、他の処理を行っているサーバが利用できるようになるまで待つ代わりに、リクエストを転送して利用可能なサーバで実行できます。あるサーバに負荷がかかりすぎると、別のサーバがリクエストの処理を引き継ぎます。ロードモニタでサーバリソースの利用状況を知ることができます。ウェイト指定のラウンドロビンのような単純なロードバランス方式を使って、収集されるサーバ統計の量を減らすことができます。

ロードバランスを使うには、実行時にアプリケーションロジックを処理するすべての iPlanet Application Server にアプリケーションロジックを分割する必要があります。システム管理者は、ロードバランスをアプリケーション全体に適用するかアプリケーションの一部に適用するかを決定する必要があります。

パフォーマンスを最適化してリソースを活用するには、システム管理者はそのロジックが実行できるように最適化されたサーバにアプリケーションロジックを展開します。リクエストされたロードバランスを実行するために管理者が使えるオプションにはさまざまなものがあります。「スティッキー」なロードバランスはその 1 つで、あるクライアントセッションで最初にリクエストを処理するサーバにすべてのリクエストを渡します。これはキャッシュするのに便利で、セッションやステート情報をサーバに分散できないオブジェクトにも便利です。

パーティション特性はクラスタのすべてのサーバにダイナミックに通知されます。iPlanet Application Server は定期的にロード統計を更新し、クラスタのほかの iPlanet Application Server に更新内容をブロードキャストします。ロードバランス要因に基づいて、リクエストはダイナミックにサーバに渡されます。ロードバランス要因は iPlanet Application Server Administration を使って設定されます。

データベースコネクションプーリング

パフォーマンスを向上させるため、iPlanet Application Server では、通常使う既存のコネクションを毎回確立し直すのではなく再利用するように、データベース接続をプールします。コネクションのプーリングによって、リクエストごとに新しいデータベースコネクションを作成するときに発生するオーバーヘッドを回避できます。

実行時にアプリケーションがデータベースへの新しいコネクションを作成する場合、実際には仮想コネクションを作成しています。iPlanet Application Server は「現実の」コネクションを作成して管理し、アプリケーションの仮想コネクションにリンクさせます。アプリケーションがコネクションを使用しない場合は、プールのコネクションには空きのマークが付けられます。同じパーミッションを持つ同じデータベースに新しいリクエストが出される場合（セッションやアプリケーションは違うこともある）、iPlanet Application Server は新しくコネクションを作成する代わりに既存の空きコネクションを使用することができます。これをコネクションの共有と呼びます。空きコネクションは、指定したタイムアウトが過ぎても使用されなければアプリケーションサーバによって解放されます。

通常、Web アプリケーションはグループごとにユーザに類似のアクセスパーミッションを与えます。これをロールと呼びます。セキュリティ上の理由から、それぞれのロールはユーザに必要なコネクションパーミッションだけを与えます。たとえば、管理ユーザはログインすると管理者用に設定されたデータベースコネクションを使用します。管理者グループのメンバーではないユーザは、このコネクションリソースがプールで利用できる状態にあっても使うことはできません。ただし、制限されたコネクションが現在使用されていないければ、iPlanet Application Server はこれを閉じて制限されないコネクションを新たに作成し、制限されたコネクションリソースの使用を最適化します。これをコネクションの流用と呼びます。

システム管理者は iPlanet Application Server Administration Tool を使ってサーバのデータベースコネクションキャッシュを設定します。これにはキャッシュの初期スロット番号、空きコネクションのタイムアウトなどがあります。

システム管理者は iPlanet Application Server Administration Tool を使ってサーバのパフォーマンスを監視し、キャッシュで利用できるコネクションの数を調整します。これにより、システムリソースに対するキャッシュされたコネクションの割合を最適にすることができます。

結果キャッシュ

iPlanet Application Server では、アプリケーションロジックの実行結果をキャッシュして、アプリケーションパフォーマンスを向上できます。リクエストの結果を保存して、次に同じリクエストを受信したらすぐに結果を渡します。開発者は、この機能をアプリケーションで実行するように設定できます。

キャッシュが有効になっていると、iPlanet Application Server はアプリケーションロジックの入力パラメータと結果をキャッシュに保存します。次に iPlanet Application Server が同じリクエストを実行したとき、サーバはまずキャッシュをチェックして、入力されたパラメータがキャッシュの入力パラメータと一致するかどうかを確認します。一致する場合は、リクエストを再実行する代わりに、サーバはキャッシュの結果

を取得します。結果キャッシュは、処理時間が長くなる大量のデータリクエストや頻繁にアクセスされるアプリケーションロジックには特に有効です。Java Server Pages のキャッシュも利用可能です。キャッシュは iPlanet Application Server の実行 (メイン) エンジンではなく iPlanet Application Server の Java エンジンで行われます。

データストリーミング

iPlanet Application Server にはデータストリーミング機能があります。ストリーミングにより、命令が完全に処理されるまで待たなくても、すぐにリクエストの結果を表示することができますので、パフォーマンスが向上します。アプリケーション開発者は、データのストリーミング方法を明示的に制御したり、自動ストリーミングを設定したりできます。

ストリーミングは、処理に時間がかかるクエリを含む大量のデータセットの場合、特に役に立ちます。たとえば、ユーザが 10,000 項目分の価格表をリクエストしたとします。アプリケーションは、データベースから 10,000 の項目すべてを検索するまで待機するのではなく、クエリを処理してたとえば一度に 40 項目が利用できるようになったら (または 1 ページ分を表示できるようになったら) ユーザに項目を表示します。

マルチスレッド機能

iPlanet Application Server はホストオペレーティングシステムのマルチスレッド機能をサポートします。アプリケーションでは、複数のスレッドでリクエストを処理することによってパフォーマンスを最適化しており、CPU リソースを最大限に活用できます。

アプリケーション開発者はアプリケーションのマルチスレッド機能を自動的に利用します。さらに、開発者はクエリ、挿入、更新、削除などのデータベース操作を非同期で実行できます。非同期操作により、大量のクエリなど時間がかかる操作をバックグラウンドで実行している間にアプリケーションでほかの作業をすることができます。

システム管理者は iPlanet Application Server Administration Tool を使って、次のようなマルチスレッド設定を行うことができます。

- すべてのリクエストを扱う最小スレッド数と最大スレッド数
- 非同期データベースリクエストを扱う最小スレッド数と最大スレッド数

通常、管理者はサーバのパフォーマンスを監視して利用できるスレッド数を調整し、システムリソースのスレッド割合を最適にします。

EJB セッションプーリング

ステートレスセッションビーンの実インスタンスを再利用および再プールして、各呼び出しに対するオーバーヘッドを少なくすることができます。重要な役割を持つビジネスアプリケーションとして、EJB はステートレスセッションビーンを要求し、クライアントと iPlanet Application Server 間の会話型ステートを追跡します。短期間に多数のクライアントがアクセスする場合、システムは失われた各セッションを再利用することができます。

Web サーバとの最適な通信

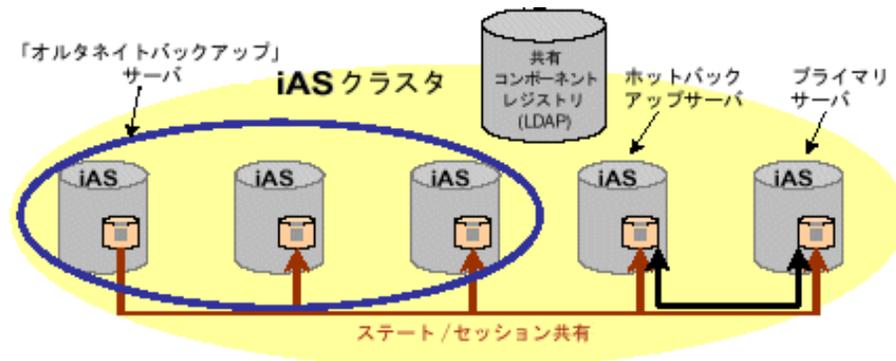
iPlanet Application Server は Web サーバとの統合を強化してアプリケーションのパフォーマンスを最適化します。この統合は、Web Connector プラグインおよび対応するリスナを通して行われます。iPlanet Application Server では、NSAPI、ISAPI、APACHEAPI と、iPlanet、Microsoft、および CGI 互換 Web サーバに最適化された CGI をサポートしています。

高い可用性

多くの企業アプリケーションは 24 時間休みなく利用できる必要があります。iPlanet Application Server は「ダイナミックフェールオーバー」(フェールリカバリとも呼ばれる)および「ダイナミックロードバランス」を通じて可用性と信頼性の高いソリューションを提供します。

iPlanet Application Server を使うと、複数のサーバにアプリケーションの全部または一部を分散させることができます。その結果、あるサーバに障害が発生してもほかのサーバで引き続きリクエストを処理することができます。

図 1-3 ダイナミックフェールオーバー



iPlanet Application Server にはアプリケーション自動再起動機能があり、ダウンタイムを最小限にしています。さらに、iPlanet Application Server は、分散ユーザセッション情報と分散アプリケーションステート情報の保持とレプリケートを行います。クラッシュしたサーバのクラスタ内で、複数の iPlanet Application Server が動作しているかぎり、情報は保持されます。

開発者はアプリケーションにリカバリ機能とスケーラビリティ機能を組み込むことを考える必要がありません。このような機能は、実行時環境でホストするだけでアプリケーションに受け継がれます。詳細については、『iPlanet Application Server 管理者ガイド』を参照してください。

アプリケーション開発期間の短縮

iPlanet Application Server を使うと、以下に説明する機能によって企業アプリケーションを短期間で開発することができます。

- 多層アプリケーションモデル
- コアサービス
- アプリケーション開発ツール

多層アプリケーションモデル

アプリケーションモデルとは、ソフトウェアアプリケーションを機能コンポーネントに分解する考え方です。iPlanet Application Server のアプリケーションモデルにより、コードの再利用化とアプリケーションの高速展開が促進されます。

iPlanet Application Server アプリケーションモデルは、アプリケーションをプレゼンテーション、ビジネスロジック、およびデータアクセスの3つの層に分解します。プレゼンテーションはさらに、ページレイアウトとプレゼンテーションロジックに分けることができます。データアクセスは、データベースとほかのデータソースの両方を参照します。

アプリケーションモデルはコンポーネント指向型であり、Java テクノロジーに基づいています。

「アプリケーションコンポーネントとその機能」の表に、iPlanet Application Server 環境のアプリケーション機能を構成する主なコンポーネントを示します。

表 1-1 アプリケーションコンポーネントとその機能

| 機能 | アプリケーションコンポーネント |
|----------------|------------------|
| アプリケーションモデルの機能 | Java アプリケーション |
| プレゼンテーションロジック | Servlet |
| ページレイアウト | JavaServer Pages |

表 1-1 アプリケーションコンポーネントとその機能 (続き)

| 機能 | アプリケーションコンポーネント |
|-----------------|---|
| ビジネスロジック | Enterprise JavaBeans |
| データベースアクセス | JDBC を使用する Enterprise JavaBeans、リレーショナルマッピングのオブジェクト |
| ほかのデータソースへのアクセス | エンタープライズコネクタ |

アプリケーションコンポーネントは次のどちらかのカテゴリに分類されます。

- 業界標準コンポーネント
- その他のコンポーネント

業界標準コンポーネント

Java アプリケーションを開発する場合は、できるだけ標準コンポーネントを使用することをお勧めします。標準コンポーネントには、Servlet、JavaServer Pages (JSP)、Enterprise JavaBeans (EJB) があります。

- Servlet はページロジックとページ移動を定義する Java クラスです。Servlet はビジネスコンポーネントの作成や呼び出しもサポートします。
- JSP は HTML、JSP タグ、Java を組み合わせて記述する Web ブラウザページです。
- EJB はアプリケーションのビジネスルールとビジネスエンティティをカプセル化します。

さらに、アプリケーションコンポーネントは JDBC 呼び出しを行います。JDBC はデータベース接続の標準 API です。

以上の標準コンポーネントの使い方に関する詳細については、『iPlanet Application Server 開発者ガイド』を参照してください。

その他のコンポーネント

その他のアプリケーションコンポーネントには、AppLogic、HTML テンプレート、クエリファイル、およびエクステンションがあります。C++ で記述されているアプリケーションや、iPlanet Application Server 2.x 環境で実行されるアプリケーションでは、以上のコンポーネントを使用する必要があります。また、使用するアプリケーションが正しい企業データソースにアクセスする必要がある場合は、あらかじめ作成したエクステンションを使ったり、独自に作成する必要があります。

- AppLogic は C++ や Java で記述したプログラム命令のセットで、アプリケーション内で明確に定義されたモジュール化タスクを実行します。

- HTML テンプレートは HTML とダイナミックデータを一体化したテキストファイルで、書式化した出力を生成します。テンプレートには GX タグという特別なマークアップタグが使用されます。
- クエリファイルは SQL コマンドを持つテキストファイルで、データベースのクエリや更新を行います。
- エクステンションにより、ビジネスロジックのコンポーネントをカスタムリソースまたは社外秘のリソースに接続することができます。エクステンションは C++ または Java で記述された持続モジュールです。iPlanet Application Server にダイナミックに読み込まれ、エクステンションが存在するかぎり複数の AppLogic や EJB からアクセスされます。
 - iPlanet Application Server Integration Solution には、すぐに使用できるエクステンションが用意されています。iPlanet Application Server Integration Solution は MQSeries、Tuxedo、CICS、IMS、および R/3 アプリケーションで使用することができます。これらのアプリケーションは、iPlanet Application Server に展開されるアプリケーションと簡単に統合できます。
 - 開発者は iPlanet Extension Builder を使って独自のカスタムエクステンションを作成できます。iPlanet Extension Builder は、iPlanet Application Server に統合される別パッケージの GUI ベースのツールです。

AppLogic、HTML テンプレート、およびクエリファイルの詳細については、『iPlanet Application Server 移行ガイド』を参照してください。拡張機能の詳細については、iPlanet Extension Builder のマニュアルをご覧ください。

コアサービス

iPlanet Application Server は、高いパフォーマンスとスケーラビリティを持つトランザクションアプリケーションを構築、展開、および管理するためのアプリケーションサービスとシステムサービスを備えています。これらのサービスには、組み込みのステートおよびセッション管理、リクエストおよびトランザクション管理、結果キャッシュ、コネクションキャッシュなどがあります。

アプリケーション開発ツール

iPlanet Application Server は、J2EE 環境の JSP、Servlet、EJB を完全にサポートしています。iPlanet Application Server は先進的な統合開発環境への統合を強力にサポートしています。

アプリケーション開発者はさまざまなツールから選択してアプリケーションを構築することができます。ツールには簡単なテキストエディタから、ビジュアル Java エディタやビジュアル HTML エディタ、統合開発環境 (IDE) までさまざまなものがあります。

Forte for Java 製品ファミリーは、入門レベルの企業アプリケーションをどのようなプラットフォームでも Java 言語で作成するためのもっとも優れた開発ツールです。これは完全なオープンソリューションであり、Netbeans™ オープンソースツールプラットフォームに基づいています。

iPlanet では、インターネット版ツールキット Forte for Java のプラグインも用意しています。開発者はこのツールキットにより提供される開発と展開の統合環境で Forte for Java の開発機能を十分に利用し、iPlanet Application Server 実行環境においてローカルでもリモートでも透過的に展開、テスト、およびデバッグができるアプリケーションを構築します。

また、プラグインを使って Servlet、JSP、および iPlanet Application Server 記述子のカスタマイザを WAR にパッケージすることもできます。開発者は、統合 Enterprise Java Beans (EJB) モジュールウィザードを使って、EJB テクノロジに基づくビジネスロジックを持つ、洗練された Web アプリケーションを簡単に開発し展開することができます。

セッションおよびステートの管理

iPlanet Application Server は Web アプリケーションが必要とするステートおよびセッションの管理機能をサポートします。

ステートおよびセッションの情報は分散環境にある各サーバに保存されます。たとえば、あるアプリケーションではログイン画面を表示し、ユーザに ID とパスワードの入力を求め、得られた情報をセッションオブジェクトに保存します。その後アプリケーションはこの情報を使って複数のデータベースにログインするので、ユーザは同じ内容を再入力する必要がありません。

同様に、オンラインショッピングのセッションオブジェクトでは、購入する製品のリスト (数量、価格など) と持続変数 (現在の注文総計) を格納します。

ステートおよびセッションの管理は、複数の複雑な操作があるアプリケーションでは特に重要です。アプリケーションロジックが異なるサーバに分割されている環境では、システム管理者は iPlanet Application Server Administration Tool を使って、1つのサーバをすべてのステート情報のセントラルリポジトリとして動作するように指定することもできます。

セッションおよびステートの管理の詳細については、『iPlanet Application Server 開発者ガイド』を参照してください。

セキュリティ

iPlanet Directory Server は、アクセス制御リスト、SSL (Secure Sockets Layer)、およびパスワードを使って iPlanet Application Server アプリケーションにデータセキュリティを提供します。

また、iPlanet Application Server では Web サーバ通信がセキュリティで保護されており、SSL、HTTP、および HTTP チャレンジ応答認証をサポートします。ブラウザとデータソース間のセキュリティギャップを埋めるために、iPlanet Application Server では、トランザクション操作を安全に行うユーザ認証、cookie、およびデータベースアクセス制御をサポートしています。イベントの記録および追跡を行うことによって、認可されていないアクセスを検出および排除します。

ユーザ認証

iPlanet Application Server Administrator と iPlanet Directory Server はユーザ認証機能を提供し、認可されたユーザだけがアプリケーション、データベース、およびディレクトリにアクセスできるようにします。

サーバ管理者は Administrator のセキュリティツールを使って、指定したリソースにアクセスできるユーザ、グループ、およびロールを作成します。

データソースへの安全なアクセス

iPlanet Application Server はリレーショナルデータベース管理システム用の既存のロールアクセスのフレームワーク内で動作します。ユーザやアプリケーションは、データベースにログインしたあとでデータにアクセスすることができます。

開発者はアプリケーションを作成して、ユーザが一度ログイン情報を入力するとアプリケーションがセッションオブジェクトにその情報を保存できるようにします。以後はアプリケーションが必要に応じてバックグラウンドで最初のログイン情報を使って別のデータベースにログインするので、ユーザは新たに入力する必要がありません。

iPlanet Application Server は Web ブラウザとリレーショナルデータベースシステム間の安全なゲートキーパーとして機能し、バックエンドのデータを保護します。

製品コンポーネント

この節では、iPlanet Application Server のソフトウェアと製品コンポーネントについて説明します。この節には次のトピックがあります。

- プログラミング API
- システムサービスとアプリケーションサービス
- サンプルアプリケーション
- コアアプリケーションサーバコンポーネント
- iPlanet Directory Server

プログラミング API

iPlanet Application Server は Java 2 Platform Enterprise Edition が定める標準に準拠しています。iPlanet Application Server は業界標準の Java API をサポートします。特に、iPlanet Application Server は次の API とテクノロジーをサポートしています。

- JDK 1.2 仕様
- Java Servlet 2.2 API 仕様
- Enterprise JavaBeans 1.1 仕様
- JavaServer Pages 1.0 仕様
- JDBC 2.0 コア API 仕様
- JDBC 2.0 標準拡張仕様

『iPlanet Application Server 開発者ガイド』で説明されているように、iPlanet Application Server 6.5 は JDBC 2.0 を完全にサポートしています。

iPlanet Application Server 6.5 でサポートされる業界標準の Java API の詳細については、該当する API の仕様を参照してください。すべての仕様書は installdir/ias/docs/index.htm からアクセス可能です。installdir は、iPlanet Application Server がインストールされている場所です。

システムサービスとアプリケーションサービス

システムサービスとアプリケーションサービスには、さまざまなアプリケーションレベルの機能とシステムレベルの機能が用意されています。これらのサービスにより、複雑なビジネスロジックとトランザクションベースのアプリケーションを開発、展開、および管理することができます。

サンプルアプリケーション

iPlanet Application Server には Web ベースのサンプルアプリケーションが含まれているので、iPlanet Application Server 環境で開発および展開するための技術をより短期間で習得することができます。

サンプルの1つにブックストアアプリケーションがあり、オンラインでの書籍の閲覧、検索、および注文をシミュレートしています。このJavaアプリケーションではServlet、JavaServer Pages、Enterprise JavaBeans、およびJDBCを使ったデータアクセスなどの業界標準コンポーネントが使われており、iPlanet Application Server アプリケーションモデルの例を示しています。オンラインブックストアアプリケーションのインストール方法と使い方については、『iPlanet Application Server 開発者ガイド』を参照してください。

また、バンキングアプリケーションのサンプルもあり、オンライン口座のユーザセッションをシミュレートしています。このサンプルでは、既存のアプリケーションを業界標準のJavaアプリケーションモデルに移行して準拠させる方法を示しています。バンクアプリケーションのインストール方法については、『iPlanet Application Server 開発者ガイド』を参照してください。アプリケーションコードの詳細については、『iPlanet Application Server 移行ガイド』を参照してください。

コアアプリケーションサーバコンポーネント

iPlanet Application Server には次のコアコンポーネントがインストールされています。

- Web Connector プラグイン
- iPlanet Application Server Administration Tool
- iPlanet Application Server Deployment Tool

Web Connector プラグイン

Web Connector プラグインを使うと、iPlanet Application Server と Web サーバ間で通信できるようになります。iPlanet Application Server をインストールすると、Web サーバが Web Connector プラグイン用に自動的に設定されます。つまり、Web サーバ上の必要なディレクトリと設定がすべて更新されます。

iPlanet Application Server と Web Connector プラグインの接続に問題がある場合は、『iPlanet Application Server 管理者ガイド』を参照してください。

iPlanet Application Server Administration Tool

iPlanet Application Server Administration Tool はグラフィカルユーザインタフェースを持つスタンドアロン Java アプリケーションで、1つまたは複数の iPlanet Application Server を管理することができます。

iPlanet Application Server Deployment Tool

iPlanet Application Server Deployment Tool を使うと、J2EE アプリケーションをパッケージ化して展開できます。Administration Tool と同様に、Deployment Tool もグラフィカルユーザインターフェイスを持つスタンドアロン Java アプリケーションです。

iPlanet Directory Server

iPlanet Application Server などのディレクトリに対応したアプリケーションでは、iPlanet Directory Server をネットワークアクセスできる共通の場所として使用し、ユーザおよびグループ ID、サーバ ID、アクセス制御情報などの共有データを保存します。Directory Server のサービスとして、Distinguished Name Service (DNS) がもっともよく知られています。

iPlanet Directory Server にはグローバルディレクトリサービスがあり、さまざまなアプリケーションに情報を提供します。グローバルディレクトリサービスは、ディレクトリ情報の単一集中管理リポジトリで、アプリケーションとディレクトリ間のネットワークベースの通信を介して、すべてのアプリケーションからアクセスできます。iPlanet Directory Server では、LDAP (Lightweight Directory Access Protocol) を使って、アプリケーションからグローバルディレクトリサービスにアクセスします。LDAP プロトコルを使えば、ハードウェアおよびネットワークインフラストラクチャにわずかに投資するだけで、iPlanet Directory Server を数百万のエントリに対応させることができます。

注 iPlanet Directory Server は、Windows では `slapd` サービスとして、Solaris では `ns-slapd` として実行されます。

iPlanet Directory Server 4.13 を iPlanet Application Server とともにインストールすると、設定情報および認証情報が iPlanet Directory Server に格納されます。iPlanet Application Server をインストールすると、この情報のブランチを持つ Directory Server データ情報ツリー (DIT) が設定されます。詳細については、次のサイトに掲載されている『iPlanet Directory Server インストールガイド』を参照してください。

<http://docs.iplanet.com>

設定されるディレクトリは、アプリケーションサーバ設定情報の保存に使われる Directory Server の一部です。このディレクトリには、`o=NetscapeRoot` データツリーが含まれます。このデータツリーには、iPlanet Application Server によって、組織の識別に設定した接尾辞を持つ設定情報が格納されます。複数のサーバインストールでは、設定ディレクトリ上に設定を保存できます。

iPlanet Application Server とともに Directory Server コンポーネントをインストールする場合は、インストール先に別の Directory Server がインストールされている場合でも、このインストールを設定ディレクトリとして指定する必要があります。

Directory Server のさまざまな機能の概要については、『iPlanet Directory Server インストールガイド』を参照してください。

アプリケーションの展開

この章では、iPlanet Application Server 6.5 に付属している Hello DB のサンプルアプリケーションを展開するために必要な手順について説明します。

この章で扱うトピックは次のとおりです。

- PointBase データベースサーバの起動
- iPlanet Application Server の停止と起動
- データソースの登録
- アプリケーションのビルド
- アプリケーションの表示
- アプリケーションの展開の解除

アプリケーションのビルドとパッケージングの詳細については、『iPlanet Application Server 開発者ガイド』を参照してください。

PointBase データベースサーバの起動

PointBase Network 3.5 は iPlanet Application Server に付属しているデータベースサーバで、アプリケーションをテストするためのものです。iPlanet Application Server をインストールしたあと、データベースサーバを停止したりシステムを再起動したことがない場合は、PointBase Database Server を起動する必要はありません。このガイドの例では、PointBase Database Server を実行していることを前提にしています。

PointBase Database Server が実行されていない場合は、次の手順で起動してから先へ進んでください。

- Solaris では、`installdir/pointbase/network/bin` に移動し、次のコマンドを入力します。

```
pointbaseServer Start
```

- Windows では、「スタート」>「プログラム」>「PointBase Network 3.5」>「PointBase Server」を選択してください。

iPlanet Application Server の停止と起動

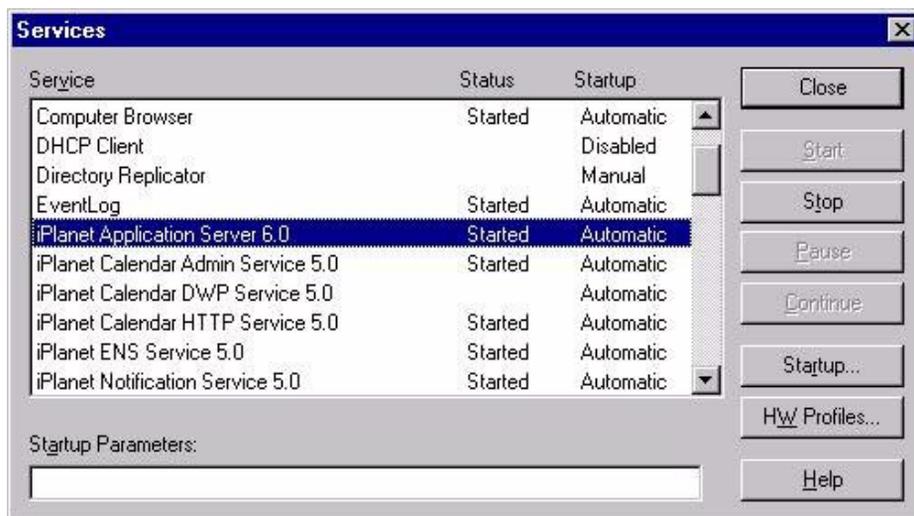
サーバはインストール中に自動的に起動するので、まずサーバを停止してから起動する必要があります。ここでは、Windows NT の場合と Solaris の場合について説明します。

Windows でサーバを停止して起動するには

iPlanet Application Server は Windows のサービスとして自動的にインストールされます。サーバを停止して起動し、サービスプロパティを設定するには、次の手順を実行します。

1. 「スタート」メニューから次のパスを選択して「コントロールパネル」を開きます。
「設定」>「コントロールパネル」
2. 「サービス (Services)」をダブルクリックします。図 2-1 の画面が表示されます。

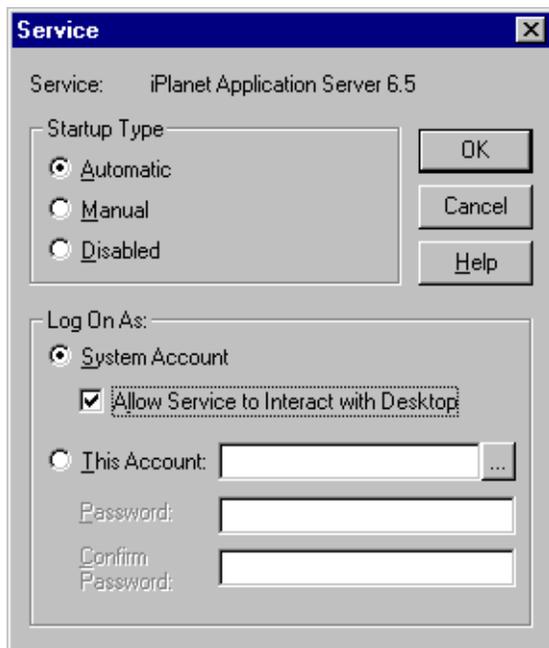
図 2-1 「サービス (Services)」画面



3. iPlanet Application Server 6.5 を強調表示して、「スタートアップ (Startup)」をクリックします。

図 2-2 の画面が表示されます。

図 2-2 「サービス (Services)」 のパラメータ画面



4. 「デスクトップとの対話をサービスに許可 (Allow Service to Interact with Desktop)」をオンにして、「OK」をクリックします。「サービス (Services)」画面 (図 2-1) に戻ります。

(この手順はサーバの起動や停止には関係がありませんが、あとでアプリケーションの実行中にログを表示する場合に役に立ちます。)

ヒント システム起動時に自動ではなく手動でサーバを起動する場合は、「スタートアップの種類 (Startup Type)」で「手動 (Manual)」を選択することもできます (図 2-2)。

5. 「停止」 ボタンをクリックします。
6. サーバプロセスがすべて終了するまで 30 秒待ちます。

7. 「起動 (Start)」 ボタンをクリックします。

デスクトップとの対話をサービスに許可しているので、デスクトップにはさらに 5 種類のポップアップウィンドウが表示されます。それぞれ別のコンポーネントのログが表示されます。

ヒント サーバプロセスのログは膨大な量になることがあります。バッファサイズとログウィンドウのサイズを大きくした方が便利な場合もあります。設定を確認するには、ログウィンドウの左上にあるアイコンをクリックしてウィンドウメニューを表示します。「プロパティ (Properties)」の「レイアウト (Layout)」タブを選択します。表示される画面でバッファとウィンドウのサイズを変更できます。

Solaris でサーバを停止して起動するには

Solaris で iPlanet Application Server を停止して再起動するには、次の手順を実行します。

1. 必ずルートユーザとしてログインしてください。
2. 現在のディレクトリを次のように変更します。 *ias install directory/ias/bin/*
3. サーバを停止するには、次のコマンドを実行します。

```
./iascontrol stop
```

4. サーバプロセスがすべて終了するまで 30 秒待ちます。
5. サーバを再起動するには、次のコマンドを実行します。

```
./iascontrol start
```

データソースの登録

データベース情報を jdbc/helloDb/HelloDB という名前のデータソースとして iPlanet Application Server に登録します。

登録すると、アプリケーションに必要なデータベースへの接続パラメータが定義されます。

Solaris と Windows にデータベースを登録するには

1. Solaris では、PATH 環境変数に *installdir/ias/bin* ディレクトリを追加します。*installdir* は iPlanet Application Server をインストールしたディレクトリです。
Windows では、「スタート」メニューで「プログラム」->「コマンドプロンプト」を選択して DOS コマンドプロンプトを開きます。
2. 次のコマンドを使って、現在のディレクトリを *helloDb* ソースディレクトリに変更します。

Solaris の場合

```
cd installdir/ias/ias-samples/database/helloDb/src/schema
```

Win NT の場合

```
cd installdir:¥ias¥ias-samples¥database¥helloDb¥src¥schema
```

3. 次のコマンドを使って、データソースを登録します。

```
iasdeploy regdatasource helloDb-pointbase-type4.xml
```

ファイル *helloDb-pointbase-type4.xml* には、データソースの配置記述子が含まれています。ファイルを表示すると、JDBC 接続の確立に必要な情報がすべて含まれていることがわかります。

アプリケーションのビルド

これでアプリケーションをビルドする準備ができました。iPlanet Application Server の `installdir/ias/ias-samples` にはいくつかのサンプルが用意されています。iPlanet Application Server 6.5 には Java ベースのビルドツール Ant が付属しています。これは、従来の make file に似た XML ベースの設定ファイルを使用します。それぞれのサンプルには、コンポーネントにビルドを指示する `build.xml` が含まれています。

アプリケーションをビルドするには

アプリケーションをビルドするには、次の手順を実行します。

1. Solaris を使う場合、次のディレクトリは PATH 環境変数の一部であることが必要です。

```
installdir/ias/bin
```

Windows NT を使う場合、この手順はインストール中に自動的に完了します。

2. 次のコマンドを使って、現在のディレクトリを `hellodb` ソースディレクトリに変更します。

Solaris の場合

```
cd installdir/ias/ias-samples/database/hellodb/src/
```

Windows の場合

```
cd installdir\ias\ias-samples\database\hellodb\src
```

3. 次のコマンドを使ってサンプルコンポーネントをビルドします。

```
build all
```

このコマンドは、サンプルの Java ソースファイルをすべてコンパイルし、WAR および JAR コンポーネントをビルドして EAR ファイルにまとめます。

| | |
|------------|---|
| ヒント | WAR、JAR、および EAR ファイルは標準の J2EE パッケージングコンポーネントです。 |
|------------|---|

4. 次のコマンドを使って、アプリケーションサーバにアプリケーションを展開します。

```
build install
```

このコマンドは、製品に同梱されている `iasdeploy` ユーティリティを使って `helloworldb` アプリケーションを `iPlanet Application Server` に登録します。このユーティリティは、`EAR` ファイルに格納されているアプリケーションコンポーネントと配置記述子を使って、アプリケーションサーバに `helloworldb` をインストールする方法を決定します。

アプリケーションの表示

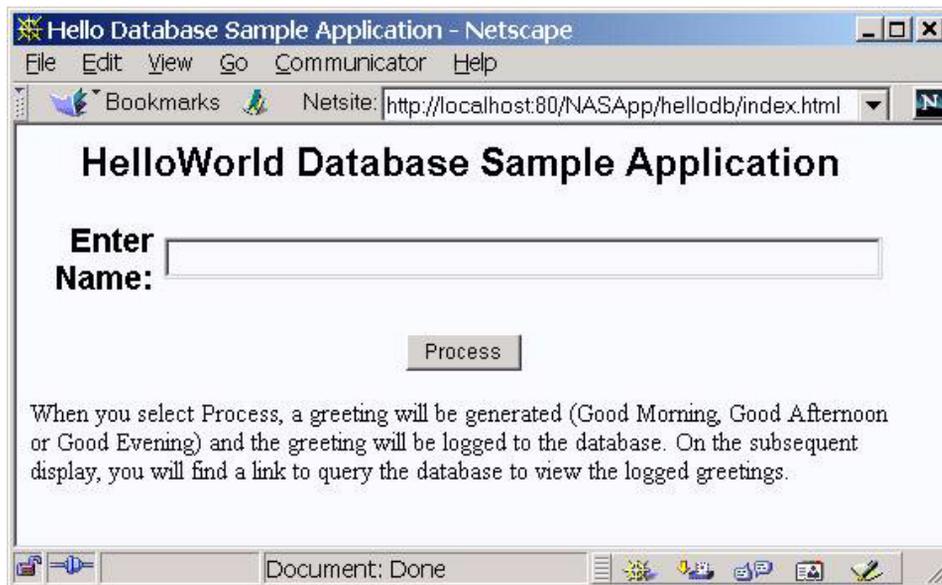
これで `helloworldb` アプリケーションのインストールが終了し、実行できるようになりました。

次の URL を入力します。[`PORT_NUMBER`] には Web サーバで使うネットワークポートの番号を入力します。

```
http://127.0.0.1:[PORT_NUMBER]/NASApp/helloadb/index.html
```

図 2-3 の画面が表示されます。

図 2-3 HelloDB サンプルアプリケーション



アプリケーションの展開の解除

`iasdeploy` を使ってアプリケーションの展開を完全に解除します。

次のようにします。 `iasdeploy removeapp application_ear_file`

次のように、J2EE のアプリケーション名を引数として使うこともできます。

```
iasdeploy removeapp j2ee_app_name
```

このほかにも、次のようなオプションがあります。

1. Web アプリケーションを削除する場合は、`removeweb webapp_name`
2. EJB を削除する場合は、`removeejb ejbjar_name`

使用可能なオプションをすべて表示する場合は、`iasdeploy -help` コマンドを使います。

管理ツールと配置ツール

この章では、アプリケーションを展開して iPlanet Application Server を管理するツールについて説明します。

iPlanet Application Server には統合管理機能があり、アプリケーションを簡単に管理できます。この機能の内容は次のとおりです。

- iPlanet Application Server Administration
- アプリケーションメッセージの記録
- iPlanet Directory Server の管理
- サードパーティ管理ツールのサポート

iPlanet Application Server Administration

iPlanet Application Server (iAS) Administration Tool はグラフィカルユーザインタフェースを持つ Java アプリケーションです。iPlanet Application Server Administrator を使うと次のことができます。

- 複数のサーバと分散アプリケーションのリモート管理
- ダイナミックな展開とアプリケーションの拡張
- パフォーマンスチューニングとサーバ環境の最適化

管理とチューニングには、データベースのコネクションスレッド数の調整、ロードバランスパラメータの調整、Web サーバの設定、ロールベースのセキュリティ管理などがあります。

iPlanet Application Server Administration Tool の詳細については、『iPlanet Application Server 管理者ガイド』を参照してください。

基本管理タスク

この節では、iASAT ツールへのアクセス方法と、iASAT ツールを使用しての iPlanet Application Server の起動および停止方法について説明します。

- Windows の場合は、「スタート」>「プログラム」>「iPlanet Application Server 6.5」>「iAS Administration Tool」を選択します。
- Solaris の場合は、`iASInstallDir/ias/bin` に移動し、「ksvradmin」と入力します。

サーバを起動するには

- 登録済みサーバのリストから、起動するサーバを選択します。
- 「サーバを起動」をクリックします。

同様に、サーバを停止するには、サーバを指定して「サーバの停止」をクリックします。

コマンドラインツール

iPlanet Application Server には、さまざまなコマンドラインツールおよび実行可能ファイルがあり、これらはコマンドラインプロンプト (Windows) や、シェルプロンプト (Solaris) から実行可能です。

コマンドラインツールを使うと、基本設定からアプリケーションの展開まで、さまざまなタスクを直接実行できます。

iPlanet Application Server の管理で使うコマンドラインツールのリストについては、『iPlanet Application Server 管理者ガイド』を参照してください。

コマンドラインツールの完全な説明を表示するには、プロンプトにコマンドとそのあとにスペースを入力し、「-help」と入力します。たとえば、`iascontrol` コマンドで実行できるすべてのオプションの完全なリストを表示するには、コマンドラインプロンプトに「`iascontrol-help`」と入力します。

コマンドラインツールの使用

iPlanet Application Server のコマンドラインツールの使用方法は、Windows プラットフォームと Solaris プラットフォームでは異なります。大半のコマンドラインツールは GUI ベースの iPlanet Application Server Administration Tool および iPlanet Application Server Deployment Tool と統合されています。

Solaris では、iPlanet Application Server Administration Tool および iPlanet Application Server Deployment Tool も使用できますが、コマンドラインツールの使用法は広範囲にわたります。Solaris では iPlanet Application Server の複数のインスタンスを持つことができるので、そのインスタンスだけの属性を変更する場合に、iPlanet Application Server の特定のインスタンスのインストールディレクトリからコマンドラインツールを実行することが必要になる場合があります。

Windows では、コマンドラインツールは実行可能 (.exe) ファイルの形式になっています。

Solaris システムと Windows システムのどちらも、コマンドラインツールはパス <iASInstallDir>/ias/bin にあります。

サーバを起動して停止するには

- Windows の場合は、「スタート」>「ファイル名を指定して実行」を選択し、「iascontrol」と入力します。
- Solaris の場合は、*iASInstallDir/ias/bin* に移動し、「./iascontrol」と入力します。

サーバを起動するには、次のコマンドを入力します。iascontrol start

サーバを停止するには、次のコマンドを入力します。iascontrol stop

iascontrol で使用できるさまざまなオプションの詳細については、iascontrol -help と入力して表示されるオンラインヘルプと『iPlanet Application Server 管理者ガイド』を参照してください。

アプリケーションの展開

次のツールを使って iPlanet Application Server にアプリケーションを展開します。

GUI ベースのツールを使うには

- Windows の場合は、「スタート」>「プログラム」>「iPlanet Application Server 6.5」>「iAS Deployment Tool」を選択します。
- Solaris の場合は、*iASInstallDir/ias/bin* に移動し、「./deploytool」と入力します。

コマンドラインの Deployment Tool にアクセスするには

- Windows の場合は、「スタート」>「ファイル名を指定して実行」を選択し、「iasdeploy」と入力します。
- Solaris の場合は、*IASInstallDir/ias/bin* に移動し、「./iasdeploy」と入力します。

アプリケーションメッセージの記録

メッセージログは、アプリケーションエラーを追跡してデバッグするのに有効です。log() メソッドを使って、サーバ管理者が iPlanet Application Server サービスに設定したものと同一ログの送信先にメッセージを送信できます。

管理および展開に関するログは、KAS として知られている Administration Server に保存されます。そのほかに、メッセージおよびイベントログが保存されるサーバとして、KJS (Java サーバ)、KCS (C++ サーバ)、および KXS (Executive Server) があります。KAS はこれらの3つのサーバを起動し、それらのサーバの稼動状況を監視します。

たとえば、アプリケーションがコードのセグメント内で問題に遭遇した場合は、関連するエラーメッセージを記録できます。エラーメッセージよりアプリケーションのステータスについての情報メッセージの方が役立ちます。

ログメッセージの書式設定

すべてのログメッセージには次の4つのコンポーネントがあります。

- メッセージが作成された日時
- 情報、警告、エラーなどのメッセージタイプ
- サービスまたはアプリケーションコンポーネント ID
- メッセージテキスト

ログメッセージをテキストベースの送信先ログに送信すると、次のように書式設定されます。

[メッセージの日時] メッセージタイプ: サービス ID: メッセージテキスト

たとえば、ASCII テキストファイルに送信された次のメッセージは、メッセージの書式を示しています。

```
[01/18/00 11:11:12:0] info (1):GMS-017:server shutdown (host  
0xc0a801ae, port 10818, group 'iAS') - updated host database
```

```
[01/18/00 11:11:18:2] warning (1):GMS-019:duplicate server (host  
0xc0a8017f, port 10818) recognized, please contact iPlanet  
Communications for additional licenses
```

ログ送信先の決定

ロギングサービスを設定して、次のいずれかまたはすべての送信先に、サーバおよびアプリケーションメッセージを記録できます。

- プロセスコンソール
- アプリケーションログ
- ASCII テキストファイル
- データベーステーブル

ロギングが有効であり、さらにコンソールが開いていてコンソールロギングも有効であれば、ロギングサービスは自動的に Windows および Solaris プラットフォームのプロセスコンソールにメッセージを送信します。Windows では、ロギングサービスはアプリケーションログにもメッセージを送信します。プロセスコンソールに記録しても、メッセージは記録されません。メッセージが画面からスクロールして消えてしまうと、メッセージを取得できません。

特定の送信先へのログを有効にする方法の詳細については、『iPlanet Application Server 管理者ガイド』を参照してください。

iPlanet Directory Server の管理

iPlanet Directory Server は iPlanet Application Server に同梱されており、Lightweight Directory Access Protocol (LDAP) を iPlanet に実装したものです。iPlanet Application Server は設定データを保存するだけでなく、ユーザ情報とグループ情報のセントラルリポジトリとしても iPlanet Directory Server を活用します。1 つの iPlanet Directory Server で、iPlanet Application Server のインスタンスを最大 5 クラスタまでサポートします。これにより、すべての iPlanet Application Server インストールの管理データを 1 か所に集中することができます。

iPlanet Application Server Administrator は LDAP クライアントとして動作するので、ユーザーやグループの情報にアクセスできます。このように LDAP と統合されているので、iPlanet Application Server では企業内のユーザ、グループ、およびロールを統一して管理することができます。

サードパーティ管理ツールのサポート

iPlanet Application Server には、HP Openview などの SNMP エージェントを経由して監視や管理を行う機能があります。SNMP は、ネットワークの稼動状況に関するデータを交換するために使われるプロトコルです。

iPlanet Application Server はサーバの MIB (Management Information Base) と呼ばれるツリー状の階層構造内に、ネットワーク管理に関する変数を格納しています。この MIB を通じて、iPlanet Application Server は主要な管理情報を SNMP を実行するサードパーティのツールに公開します。その結果、iPlanet Application Server を企業のサーバ管理ツールと統合することができ、ほかのソリューションをリモート管理することができます。

詳細については、『iPlanet Application Server 管理者ガイド』を参照してください。

iPlanet Application Server の問題と対処方法

インストール関連の問題

この節では、iPlanet Application Server Enterprise Edition 6.5 での既知のインストール関連の問題とその対処方法について説明します。

問題 (549653): iPlanet Application Server のインストール可能ファイルが、空白文字や特殊文字が含まれているディレクトリに置かれていると、Windows でのインストールおよびアップグレードが失敗する

対処方法

iPlanet Application Server のインストール可能ファイルは、空白文字や特殊文字が含まれていないディレクトリに置いてください。

問題 (546941): IDS 5.0 をマシンからアンインストールすると、iPlanet Application Server は Windows 2000 にディレクトリサーバをインストールできない

この問題は、デフォルトのディレクトリサーバとともに iPlanet Application Server をインストールする Windows 2000 マシンから、以前、iPlanet Directory Server 5.0 をアンインストールしたことがある場合に発生する

対処方法

iPlanet Application Server をインストールする前に `winnt¥System32` ディレクトリから `ns-dshttpd40.dll` を削除します。

問題 : スタティック IP アドレスのないマシンに iPlanet Application Server をインストールしたい

対処方法

iPlanet Application Server をインストールする Windows 2000 または Windows NT のマシンにループバックアダプタをインストールし、これにスタティック IP アドレスを割り当てます。

問題 : NFS のマウントされたファイルシステムにインストールしようとするとクラッシュする

iPlanet Application Server は、NFS のマウントされたファイルシステムにはインストールできません。

コアサーバ関連の問題

この節では、iPlanet Application Server Enterprise Edition 6.5 での既知のコアサーバ関連の問題とその対処方法について説明します。

問題 (546093): Windows プラットフォームでの iPlanet Application Server の自動再起動が、低速マシンでは機能しない場合がある

これは、iPlanet Directory Service の起動に関連する問題のためである

対処方法

低速マシンで自動起動を機能させるには、環境変数 `IAS_KASWAITTIME` を追加し、5 ~ 15 (秒) に設定します。手動起動の場合はこの変数を設定する必要はありません。

Administration Tool 関連の問題

この節では、iPlanet Application Server Enterprise Edition 6.5 での既知の Administration Tool 関連の問題とその対処方法について説明します。

問題 (544526): Administration Tool からデータソースを削除できない

対処方法

Administration Tool からは外部のデータソースや iPlanet Type 2 データソースを削除できません。kregedit を使ってレジストリからデータソースエントリを削除します。

詳細については、『iPlanet Application Server 管理者ガイド』の第 8 章「データベース接続の管理」を参照してください。

問題 (540597): `KIVAes.sh` 停止コマンドを使うと、StartupClass 内の Shutdown メソッドが起動しない

対処方法

`iascontrol stop` コマンドを使います。`KIVAes.sh` は廃止されているので、使わないでください。

詳細については、『iPlanet Application Server 管理者ガイド』の第 1 章「基本管理タスクの実行」を参照してください。

注 `iascontrol` コマンドを使うには、Administration Tool を使ってサーバを登録する必要があります。

展開に関する問題

この節では、iPlanet Application Server Enterprise Edition 6.5 での既知の展開に関する問題とその対処方法について説明します。

問題: EAR ファイルを展開しようとする、`iasdeploy` に「レジストリにはサーバが登録されていません」というエラーが表示される

対処方法

Administration Tool を使って初めに iPlanet Application Server を登録していないと、このメッセージが表示されます。

このエラーは、iPlanet Application Server に複数のインスタンスがある場合やサーバがリモートマシンに展開されている場合にも表示されます。そのような場合は、アプリケーションを展開する iPlanet Application Server のインスタンス、およびそのホストとポート番号を明確にする必要があります。

詳細については、『iPlanet Application Server 管理者ガイド』を参照してください。

問題: 「ビーンのホームにもリモートにも Document.class がありません」というエラーが表示される

このエラーメッセージは通常、Servlet、JSP、XML ファイル、および Beans を 1 つの EAR ファイルにまとめた場合に発生する

対処方法

1. 新しい EJB モジュールを作成し、EJB とその EJB に依存するクラスをモジュールに追加します。
2. Web モジュールを作成し、Servlet、JSP、XML ファイル、およびそれらに依存するクラスをモジュールに追加します。
3. 両方のモジュールを J2EE アプリケーションに追加し、未解決の EJB リンクを修正します。

問題 (548426): アプリケーションやモジュールを再展開しても、以前展開したアプリケーションやモジュールが削除されない。現在展開されているアプリケーションやモジュールが上書きされるだけである

アプリケーションを再展開しても、上書きされないレジストリ設定がある

対処方法

クリーンな再展開を行うには、まず `remove` を実行し、次に `deploy` を実行します。

データベースサポート関連の問題

この節では、iPlanet Application Server Enterprise Edition 6.5 での既知のデータベースサポート関連の問題とその対処方法について説明します。

問題 (481221): サードパーティドライバには JNDI からの RowSet 検索が実装されていない

対処方法

RowSet を使うには、RowSet オブジェクトを直接構築し URL を設定して、基礎となる接続を確立します。

問題 (427149): Informix および Sybase ネイティブドライバでは、コンパイル済みステートメント内のテキストおよび BLOB データタイプがサポートされない

対処方法

サードパーティ JDBC ドライバを使います。

その他の資料

iPlanet Application Server のマニュアルは、
<http://docs.iplanet.com/docs/manuals/ias.html> から PDF および HTML 形式で入手
 できます。開発者向けの付加的な資料は <http://developer.iplanet.com/> から入手でき
 ます。

次の参照資料一覧は、iPlanet Application Server のマニュアルおよび『リリースノート』に記述されているタスクと概念を示しています。特定のタスクを実行する場合や特定の概念について調べる場合は、該当するマニュアルを参照してください。

表 5-1 参照資料一覧

| 情報の内容 | 参照するマニュアル | 添付されている製品 |
|---|-----------|--|
| ソフトウェアおよびマニュアルに関する最新情報 | リリースノート | 最新のバージョンは http://docs.iplanet.com/docs/manuals/ias.html からオンラインで入手可能 |
| iPlanet Application Server およびそのコンポーネント (Web コネクタプラグイン、iPlanet Application Server Administrator) のインストールと、サンプルアプリケーションの設定 | インストールガイド | iPlanet Application Server 6.5 |

表 5-1 参照資料一覧 (続き)

| 情報の内容 | 参照するマニュアル | 添付されている製品 |
|---|-----------|--------------------------------|
| <p>次のタスクによる、オープン Java 標準モデル (Servlet、EJB、JSP、および JDBC) に準拠した iPlanet Application Server 6.5 アプリケーションの作成</p> <ul style="list-style-type: none"> • アプリケーションのプレゼンテーション層および実行層の作成 • EJB (Enterprise JavaBeans) コンポーネントへのビジネスロジックの個別部分およびエンティティの配置 • JDBC を使ったデータベースとの通信 • 反復テスト、デバッグなどアプリケーションの調整機能を使用した、正確かつ高速に動作するアプリケーションの生成 | 開発者ガイド | iPlanet Application Server 6.5 |
| <p>次のタスクを行うための、iPlanet Application Server Administrator Tool による 1 台または複数台のアプリケーションサーバの管理</p> <ul style="list-style-type: none"> • サーバの稼動状況の監視およびログ記録 • iPlanet Application Server へのセキュリティの実装 • サーバリソースの高利用度の実現 • Web コネクタプラグインの設定 • データベース接続の管理 • トランザクションの管理 • 複数のサーバの設定 • 複数のサーバでのアプリケーションの管理 • サーバのロードバランス • 分散データ同期の管理 • 開発用の iPlanet Application Server のセットアップ | 管理者ガイド | iPlanet Application Server 6.5 |

表 5-1 参照資料一覧 (続き)

| 情報の内容 | 参照するマニュアル | 添付されている製品 |
|---|--|--------------------------------|
| iPlanet Application Server に付属するオンラインバンクアプリケーションの移行サンプルを含む、Netscape Application Server バージョン 2.1 から新しい iPlanet Application Server 6.5 プログラミングモデルへのアプリケーションの移行 | 移行ガイド | iPlanet Application Server 6.5 |
| Java アプリケーションを作成する場合の iPlanet Application Server クラスライブラリの共有クラスとインターフェイス、およびそれらのメソッドの使用 | Server Foundation Class Reference (Java) | iPlanet Application Server 6.5 |
| C++ アプリケーションを作成する場合の iPlanet Application Server クラスライブラリの共有クラスとインターフェイス、およびそれらのメソッドの使用 | Server Foundation Class Reference (C++) | 別注文 |

iPlanet Application Server のアーキテクチャ

この付録では、次のトピックについて説明します。

- サーバプロセス
- システムコンポーネント

サーバプロセス

iPlanet Application Server のアーキテクチャには 4 つの内部サーバがあり、エンジンまたはプロセスと呼ばれることもあります。内部サーバは iPlanet Application Server のすべての処理を行います。表 A-1 「iPlanet Application Server のサーバと説明」は内部サーバについてまとめたものです。

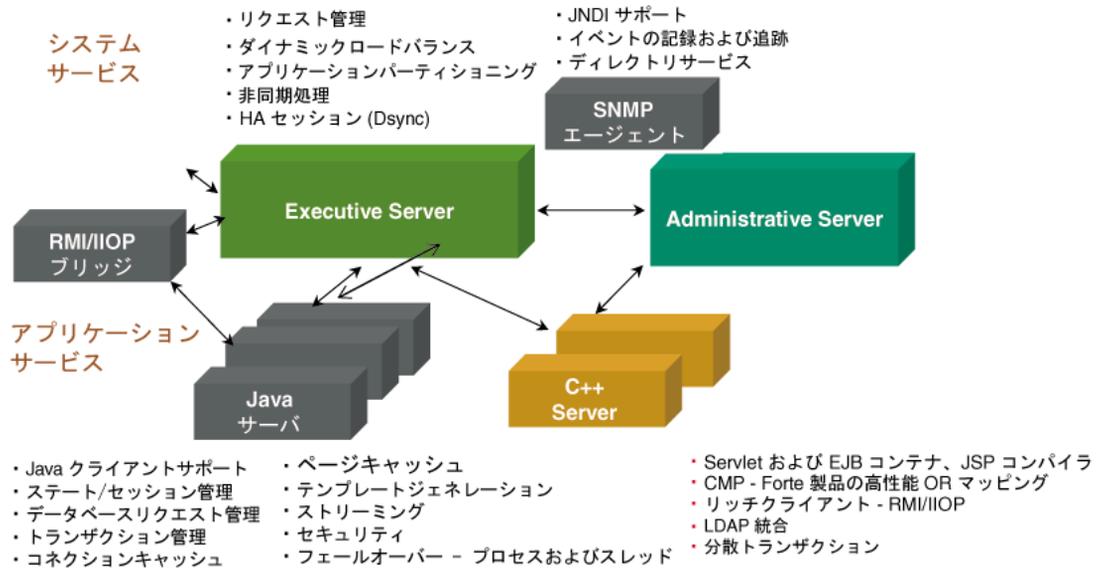
表 A-1 iPlanet Application Server のサーバと説明

| 内部サーバ | プロセス名 | 説明 |
|-----------------------|-------|---|
| Executive Server | KXS | ほとんどのシステムサービスを提供する (Administrative Server が管理する一部のサービスを含む) |
| Administration Server | KAS | iPlanet Application Server の管理とフェールリカバリのためのシステムサービスを提供する |
| Java サーバ | KJS | Java アプリケーションにサービスを提供する |

プロセス対話の要約

図 A-1 「iPlanet Application Server のプロセス」は iPlanet Application Server の 4 つのプロセスが、クライアントのリクエストを処理するために、どのように対話するかを表しています。

図 A-1 iPlanet Application Server のプロセス



Web サーバから iPlanet Application Server に送られたリクエストは、まず Executive Server (KXS) のプロセスで受信されます。KXS のプロセスは、Java サーバ (KJS) のプロセスまたは C++ サーバ (KCS) のプロセスにリクエストを送ります。KJS のプロセスは Java プログラミングロジックを実行し、KCS のプロセスは C++ プログラミングロジックを実行します。

KJS と KCS のプロセスにはそれぞれ多数の指定スレッドがあり、各スレッドでは最後まで同じプログラミングロジックが実行されます。Web サーバに返された結果はクライアントブラウザに渡されます。

アプリケーションのリクエストを処理する iPlanet Application Server 技術は、Web サーバの負荷を低減して反応速度を上げるための鍵となります。サーバ管理者は iPlanet Application Server 環境を最善のパフォーマンスにするために、次の設定ができます。

- iPlanet Application Server マシンを増設する
- KJS および KCS のプロセス数を指定する
- 各プロセスのスレッド数を保守する

高いパフォーマンスを提供すること以外にも、内部プロセスにより iPlanet Application Server を 24 時間稼働し続けるようにすることもできます。KJS または KCS のプロセスがダウンすると、Administrative Server process (KAS) がそれらを再起動します。iPlanet Application Server の補助監視機能により、KAS プロセスを常に実行しておくことができます。

iPlanet Application Server のすべてのプロセスがダウンした場合、クラスタ内のほかの iPlanet Application Server マシンがあとを受け継ぎます (複数サーバ環境を想定)。さらに、iPlanet Application Server は電子メールやファクスでシステム管理者に警告を発し、システムを別の場所にリダイレクトするように通知します。

次の節では、次の内部サーバについてさらに詳しく説明します。

- Executive Server
- Administration Server
- Java サーバおよび C++ サーバ

Executive Server

Executive Server は iPlanet Application Server のメインエンジンです。Executive Server は、iPlanet Application Server がシステムレベルのサービスを必要とするときにサービスの多くを管理します。

また、Executive Server プロセス (KXS) は、アプリケーションリクエストを該当するアプリケーションプロセス (Java サーバまたは C++ サーバプロセス) に分散します。

たとえば、iPlanet Application Server にアプリケーションリクエストが渡されると、次のように処理されます。

Executive Server はリクエストマネージャを起動します。これはシステムレベルのサービスです。

リクエストマネージャはリクエストにスレッドを割り当て、該当するアプリケーションプロセス (Java サーバまたは C++ サーバ) にリクエストを転送します。

リクエストマネージャは、終了時に Executive Server から解放され、ダイナミック利用モデルでのサーバのパフォーマンスが向上します。

Administration Server

Administration Server は1つまたは複数の iPlanet Application Server を管理します。GUI 管理ツールの iPlanet Application Server Administrator Tool (iPlanet Application ServerA) を使い、システムやアプリケーションの設定に対するすべての変更内容を該当するサーバに登録します。

また、Administration Server は、サーバプロセスが利用できなくなった場合にほかのサーバプロセスを起動するフェールリカバリサービスを管理します。このフェールリカバリサービスは、iPlanet Application Server に高度なフォールトトレランスを提供します。

Java サーバおよび C++ サーバ

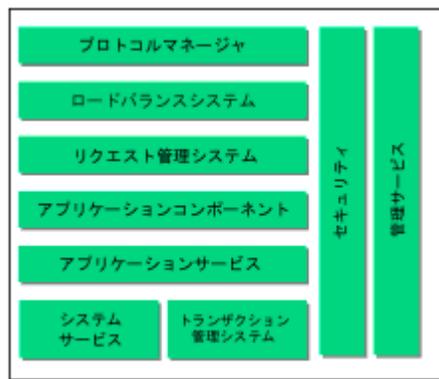
Java サーバおよび C++ サーバプロセスはアプリケーションサーバです。Java で記述したビジネスロジックコンポーネントは Java サーバで管理され、C++ で記述したコンポーネントは C++ サーバで管理されます。ビジネスロジックコンポーネントはアプリケーションの核となるもので、開発者が記述したコンパイル済みコード命令を保持しています。

また、Java サーバと C++ サーバプロセス (KJS および KCS) は、アプリケーションレベルのサービスを管理します。これらのサービスは、アプリケーションコンポーネントで必要になると適切なプロセスにダイナミックに読み込まれます。たとえば、EJB がデータベースへのアクセスを要求すると、Java サーバはデータアクセスエンジンにサービスを読み込んでそのサービスを使い、その後解放します。データアクセスエンジンが提供するデータベースコネクションは、Java サーバのアクティブメモリにキャッシュされます。別のリクエストがシステムに入力され、同じデータベースにアクセスした場合は、キャッシュされたコネクションが使用されます。このようにして iPlanet Application Server がデータアクセスエンジンを起動する回数を減らして、リクエスト処理のパフォーマンスを向上させます。

システムコンポーネント

この節では、iPlanet Application Server のアーキテクチャを構成する主な内部システムについて説明します。図 A-2 「iPlanet Application Server アーキテクチャ」にこのシステムを示します。

図 A-2 iPlanet Application Server アーキテクチャ



この節では、次のトピックについて説明します。

- プロトコルマネージャ
- ロードバランスシステム
- リクエスト管理システム
- アプリケーションコンポーネント
- アプリケーションサービス
- システムサービス
- トランザクション管理システム
- セキュリティ
- 管理サービス

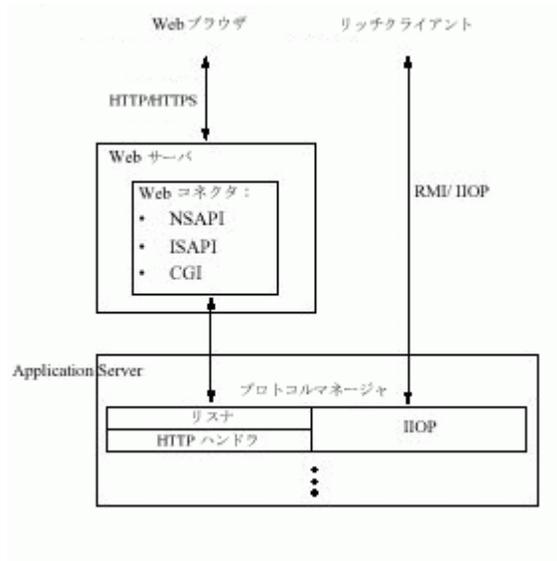
プロトコルマネージャ

iPlanet Application Server とクライアント間で通信できるようにプロトコルマネージャが管理するコネクタ、プロトコル、およびその機能は次のとおりです。

- Web サーバと iPlanet Application Server は、NSAPI、ISAPI、および最適化された CGI を介して通信します。ここで説明するリスナ、Web コネクタ、ストリーミングサービス、およびプロトコルマネージャにより、最適化と優れたパフォーマンスが達成されます。
- Web コネクタとリスナは Web サーバから iPlanet Application Server へのリクエストの送信を管理します。リスナは Web サーバからの要求を分散し、処理します。新しいリスナは HTTP ハンドラで追加します。
- HTTP ストリーミングサービスは、iPlanet Application Server から Web サーバおよび Web ブラウザへのデータストリームを扱います。
- iPlanet Application Server プロトコルマネージャは、ロードされたアクティブな全リスナにサービスを提供し管理します。プロトコルマネージャは HTTP、HTTPS (SSL を使った HTTP)、および IIOP をサポートします。

図 A-3 「プロトコルマネージャ」に、プロトコルマネージャが iPlanet Application Server とクライアント (Web ベースまたは非 Web ベース) 間の通信を可能にする方法を示します。

図 A-3 プロトコルマネージャ



Web ブラウザから送信されたリクエストは、HTTP または HTTPS プロトコルを介して Web サーバに渡されます。リクエストは適切な Web コネクタで処理されます。Web コネクタには、NSAPI Web コネクタ、ISAPI Web コネクタ、および最適化された CGI Web コネクタがあり、それぞれ iPlanet、Microsoft、および CGI 互換の Web サーバ用です。

送信されたリクエストは、iPlanet Application Server の対応するリスナに渡されます。iPlanet Application Server のリスナには NSAPI リスナ、ISAPI リスナ、および最適化された CGI リスナがあり、それぞれ iPlanet、Microsoft、および CGI 互換の Web コネクタと通信します。

ロードバランスシステム

複数の iPlanet Application Server をインストールしている環境では、受信されたリクエストはまずロードバランスシステムを通ります。ロードバランスシステムは、リクエストを処理するのに最適なサーバにリクエストを送信します。ロードバランスシステムはロードモニタとロードバランサで構成されています。

ロードモニタ

ロードモニタは、管理者が指定した内部設定に基づいてサーバロードのスナップショットを取得します。次にリソースの可用性とシステムのパフォーマンスを一覧表にまとめます。

ロードバランサ

ロードバランサはロードモニタが取得したロードとパフォーマンスの情報を使って、送られてくるリクエストを処理するのに最適なリソースを持つサーバを決定します。iPlanet Application Server の各インスタンスには、ロードバランスのパラメータに基づいてルートを決する独自のロードバランスモジュールがあります。管理者は、iPlanet Application Server Administrator を使ってパラメータを調整することができます。

適切な iPlanet Application Server マシンに振り分けられたリクエストは、リクエスト管理システムに渡されます。

リクエスト管理システム

送信されたリクエストはリクエスト管理システムで処理されます。iPlanet Application Server には複数のスレッドがあり、リクエスト管理システムはダイナミックスレッドプールのスレッドを割り当ててリクエストを処理します。リクエスト管理システムでは、大量のリクエストを同時に処理することができます。システム管理者は、最適なリクエスト処理ができるようにスレッドプールのパラメータを設定することができます。リクエスト管理システムには次のようなサブシステムがあります。

- スレッドマネージャ

スレッドマネージャはスレッドのダイナミックプールを提供します。このプールから、リクエストを処理するスレッドが割り当てられます。

- キューマネージャ

キューマネージャは、スレッドが利用できる状態になるまでリクエストをキューに入れておかなければならないときに使います。

キューマネージャは保留中のリクエストと記述的情報のリストを管理します。情報には、一意のリクエスト ID と、リクエストの現在の処理状況 (待機中、処理中、処理終了など) が含まれます。

- リクエストログ

システム管理者がリクエストログを有効に設定している場合、バックエンドデータベースまたはログファイルに Web サーバリクエストの情報ログが記録されます。

アプリケーションコンポーネント

受信した各リクエストは、1つまたは複数のアプリケーションコンポーネントを識別します。次にこれらのコンポーネントは GUID (グローバルに固有な識別子) で識別されます。GUID は iPlanet Application Server のグローバルディレクトリサービス (GDS)、iPlanet Directory Server、および LDAP サーバで検証されます。

GDS はアプリケーションコンポーネントの言語を判別し、リクエストの言語に適したハンドラを読み込みます。たとえば、Windows NT では .class ファイルは Java アプリケーションロジック用に読み込まれ、.dll ファイルは C++ アプリケーションロジック用に読み込まれます。UNIX では .class ファイルは Java アプリケーションロジック用に読み込まれ、共有ライブラリ (.so ファイル) が C++ アプリケーションロジック用に読み込まれます。

必要に応じて、iPlanet Directory Server は既知のロール情報でリクエストを検証して認証します。

次に適切なアプリケーションコンポーネントが実行され、リクエストが処理されます。たとえば、リクエストによって **Servlet** が起動され、次に 1 つまたは複数の **EJB** が呼び出されることがあります。

通常、リクエストの処理には、アプリケーションサービス、トランザクション管理システムその他のシステムサービスからの呼び出し機能が含まれます。たとえば、アプリケーションでステートとセッションの管理をしたり、ユーザのアカウントを更新するデータベースに接続することができます。

アプリケーションコンポーネントの詳細については、『**iPlanet Application Server 開発者ガイド (Java)**』を参照してください。

アプリケーションサービス

アプリケーションサービスでは、ユーザセッション、アプリケーションステート、cookie、電子メールによる通知、結果のキャッシュなどのアプリケーション機能を管理します。このようなサービスは API 呼び出しを使ってアプリケーションコンポーネントで起動します。アプリケーションサービスは KJS プロセス、KCS プロセス、またはその両方に読み込まれます。次の節では、Java および C++ アプリケーションで利用できるサービスについて説明します。

KJS または KCS で管理するサービス

表 A-2 「Java または C++ クライアントで利用できるサービス」に、Java または C++ で作成したアプリケーションで利用できるサービスを示します。

表 A-2 Java または C++ クライアントで利用できるサービス

| アプリケーションサービス | 説明 |
|---------------|---|
| ステートとセッションの管理 | ユーザログイン、ページ移動情報、「ショッピングカート」の選択など、ユーザのセッション情報を管理する。持続的なステート情報を管理する。分散型 iPlanet Application Server マシンはステートのワークスペースを利用して情報を共有する |
| cookie 管理 | cookie を識別できる Web ブラウザ用に HTTP cookie を生成する。cookie を識別しないブラウザでは、エミュレートされた cookie が URL や隠しフィールドに埋め込まれる |
| データアクセス管理 | データベースへのアクセスを提供し管理する |

表 A-2 Java または C++ クライアントで利用できるサービス (続き)

| アプリケーションサービス | 説明 |
|-----------------------|--|
| トランザクション管理 | データベーストランザクションを管理し、トランザクションへのコミットとロールバックのサポートを提供する。詳細については、「トランザクション管理システム」を参照してください。 |
| テンプレート管理 | データがユーザに返される前に、テンプレートを使ったリザルトセットデータのマージを管理する |
| データベースコネクション キャッシュ | データベースコネクションをキャッシュして、今後同じデータベースにアクセスした場合にすぐに提供できるようにする |
| 結果キャッシュ | リザルトセットのデータをキャッシュして、以降のリクエストをさらに効率よく処理できるようにする。リクエストが結果キャッシュに保存されていると、すでに計算された結果がすぐに返される。それ以外の場合は、アプリケーションロジックが実行され、結果が処理される。システム管理者は、キャッシュスロットの数、タイムアウト、キャッシュクリーニング間隔など、結果キャッシュの設定を指定できる |
| アプリケーションイベント | 時間基準やほかのイベント基準にもとづいて、アプリケーションが電子メールを送受信して AppLogic を起動したり、(Java アプリケーションでは) Servlet を起動できるようにする。この機能は管理に便利である |
| HTML ストリーミング | データをストリーミングして HTML クライアントにデータを効率よく返せるようにする |
| コネクタ | 企業アプリケーションが、iPlanet Application Server に展開したアプリケーションを統合できるようにする。コネクタは iPlanet Application Server にダイナミックに読み込まれる持続モジュールで、エクステンションが存在するかぎり複数の AppLogic または EJB からアクセスされる。コネクタはアプリケーションサービスとして動作するが、アプリケーションコンポーネントと見なすこともできる |

KJS だけが管理するサービス

表 A-3 「Java アプリケーションだけで利用できるサービス」に、Java で作成したアプリケーションだけで利用できるサービスを示します。

表 A-3 Java アプリケーションだけで利用できるサービス

| アプリケーションサービス | 説明 |
|--------------|--|
| JSP コンパイラ | JSP タグを解釈する。JSP タグは HTML タグに似ており、Web ブラウザに送信されるページのレイアウトを決定する。このコンパイラは JavaServer Pages 仕様のバージョン 0.92 をサポートする |
| Servlet コンテナ | リクエストや応答が設定されるネットワークサービスを提供し、MIME ベースのリクエストをデコードし、MIME ベースの応答を作成して、ライフサイクル期間中の Servlet を保持し管理する。HTTP および HTTPS をサポートする |
| EJB コンテナ | EJB にホームを提供し、ホーム内のビーンを管理する。管理には、ビーンの登録、ビーン用のリモートインターフェイスの提供、インスタンスの作成と削除、セキュリティチェック、セキュリティ活動状況の管理、および分散トランザクションの調整が含まれる。EJB コンテナでビーン内の持続データをすべて管理し、グローバルトランザクションを完全に管理することもできる |
| 分散トランザクション | 異なるタイプや異なる場所の複数のデータベースを扱うトランザクションをサポートする。分散トランザクションは EJB で起動され、iPlanet Application Server に組み込まれたトランザクション処理マネージャが使用される |
| LDAP サポート | セントラルリポジトリにユーザ、グループ、およびアクセス制御リストの情報を渡して、管理やセキュリティ管理を簡単に行えるようにする。iPlanet Application Server に付属する LDAP クライアントには、iPlanet Application Server Administration Tool と iPlanet Directory Server が含まれている |

システムサービス

システムサービスを使うと、アプリケーションリクエストの処理効率が向上します。このサービスはアプリケーションで直接使われることはありませんが、アプリケーションロジックの範囲外でアプリケーションを補助的にサポートします。システムレベルのサービスには API のアクセスはありません。このサービスについての説明を表 A-4 「システムサービス」に示します。次のサービスについては、この章で説明しているものもあります。

表 A-4 システムサービス

| システムサービス | 説明 |
|--------------------|---|
| プロトコル管理 | iPlanet Application Server が使うさまざまなプロトコルをサポートして、クライアントとの通信を管理する |
| リクエスト管理 | サーバに送られたリクエストを適切なプロセス (Java サーバまたは C++ サーバ) に転送し、リクエストスレッドを割り当てて、リクエストを管理する |
| グローバルディレクトリサービス | アプリケーションコンポーネントのプログラミング言語を決定し、適切なサーバプロセス (KJS または KCS) に読み込めるようにする |
| JNDI | Java Naming and Directory Interface (JNDI) は Java プラットフォームの標準拡張。JNDI API は、企業の複数のネーミングサービスやディレクトリサービスに対する統一インターフェイスを Java アプリケーションに提供する |
| イベントログ | アプリケーションロジックの実行ログを維持する。アプリケーション開発者は、アプリケーションロジックのログを有効にしてデバッグとチューニングに役立てることができる。また、システム管理者はイベントログを自動化し、ユーザーリクエストを処理する際に DLM (dynamically loadable module) とアプリケーションロジックオブジェクトで生成されるメッセージを記録することができる。イベントログはあらゆるプロセスで実行できる |
| ロードバランス | 複数のサーバ間でアプリケーションリクエストのロードバランスを保つ方法を指定する |
| アプリケーションリクエストキャッシュ | アプリケーションリクエストをキャッシュし、今後同じユーザからの同じアプリケーションコンポーネントへのリクエストがあった場合に対応できるようにする |
| 非同期処理 | 別々に届く複数のリクエストを同時に処理する |
| フェールリカバリ | Executive Server、Java サーバ、または C++ サーバのプロセスが利用できなくなった場合に、これらのサーバを再起動する |
| 分散データ同期 | データの同期を取ってフェールリカバリをサポートする。 iPlanet Application Server で実行中の KJS プロセスや KCS プロセス間だけでデータの同期を取るのではなく、クラスタにインストールされたすべての iPlanet Application Server 間でデータの同期を取る |
| SNMP サポート | SNMP エージェントを介した iPlanet Application Server へのアクセスを提供し、サードパーティの管理ツールによるリモート管理を可能にする |

表 A-4 システムサービス (続き)

| システムサービス | 説明 |
|----------|---|
| カーネルサービス | ほかのすべてのサービスやサブシステムに低レベルのサービスを提供する。カーネルサービスの例としては、言語バインドエンジンやロックマネージャがある |

トランザクション管理システム

トランザクション管理システムはバックエンドデータベースへのアクセスをサポートします。

- C++ アプリケーションは、Foundation Class Library と呼ばれる iPlanet Application Server API で提供されるクラスやインターフェイスを使ってデータベースにアクセスします。
- Java アプリケーションは標準の JDBC API を使ってデータベースにアクセスします。

ローカルトランザクションとグローバルトランザクション

Java アプリケーションでは、iPlanet Application Server はローカルとグローバルの両方のトランザクションをサポートします。グローバルトランザクションは分散トランザクションとも呼ばれます。

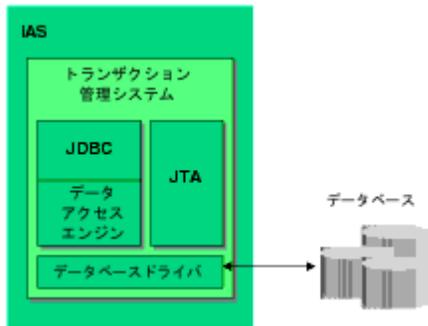
グローバルトランザクションは複数のデータベースにまたがり、データベースの種類が異なることもあります。グローバルトランザクションは、iPlanet Application Server に組み込まれた 2 階層コミットの Encina トランザクションマネージャを使って実行します。ローカルトランザクションでは、1 つのデータベースにアクセスします。複雑ではないのでアプリケーションのパフォーマンスが良くなります。

グローバルトランザクションは、EJB を介して宣言した場合だけ開始することができます。これとは対照的に、ローカルトランザクションは Servlet、JSP、EJB のどれかでプログラムした場合だけ開始することができます。

アーキテクチャの詳細

図 A-4 「トランザクション管理システム」に、トランザクション管理システムのアーキテクチャの詳細を示します。

図 A-4 トランザクション管理システム



JDBC と iPlanet Application Server API はどちらもデータアクセスエンジンに依存してデータベースドライバと対話します。iPlanet Application Server は、Oracle、DB2、Informix、Sybase、および (Windows NT の場合だけ) SQLServer の各データベースドライバをネイティブでサポートしています。

トランザクション管理システムには Java Transaction API (JTA) も含まれます。JTA は 1 つのデータベースへのコネクションに使用します。JTA はトランザクションマネージャともう一方のトランザクション要素 (iPlanet Application Server とトランザクションアプリケーションを含む) の間のローカル Java インターフェイスを指定します。JTA は業界標準の X/Open XA プロトコルの Java マッピングを提供します。このプロトコルは Encina トランザクションマネージャが使用します。

データベースアクセスの詳細については、『iPlanet Application Server 開発者ガイド』および『iPlanet Application Server 管理者ガイド』を参照してください。

セキュリティ

iPlanet Application Server では、LDAP やその他のセキュリティプロトコルの使用によって、すべてのサブシステムでセキュリティがサポートされています。詳細については、『iPlanet Application Server 開発者ガイド』および『iPlanet Application Server 管理者ガイド』を参照してください。

管理サービス

管理サービスは、Administration Server プロセス (KAS) で実行されます。KAS により、サーバとアプリケーションをリモートで管理することができます。また、KAS では、アプリケーションのパーティション分割、イベントログ、リクエストの監視、主なサーバ設定値の動的設定などのサービスもサポートしています。

管理サービスにアクセスするクライアントには、iPlanet Application Server Administration Tool、iPlanet Directory Server、およびサードパーティの SNMP エージェントがあります。

索引

A

Administration Server (KAS), 57, 58
API, 27
AppLogics, 23

C

C++ サーバ (KCS), 56, 58
cookie 管理, 63

D

Directory Server
設定ディレクトリ, 30

E

EJB コンテナ, 65
Enterprise JavaBeans, 23
Executive Server (KXS), 56, 57

J

JavaServer Pages, 23
Java サーバ (KJS), 56, 58
JSP コンパイラ, 65

L

LDAP, 65

N

Netscape Application Server Administrator, 41
Netscape Extension Builder, 24

S

Servlet, 23
Servlet コンテナ, 65

U

URL

形式、マニュアルでの～, 9

W

Web コネクタ, 60, 61

あ

アプリケーションイベント, 64

アプリケーションパーティショニング, 15

アプリケーションモデル, 22

い

イベントログ, 66

え

エクステンション, 24

き

キャッシュする

結果, 19, 64

データベースコネクション, 64

く

クエリファイル, 24

グローバルディレクトリサービス (GDS), 66

さ

サーバプロセス, 55

サービス

アプリケーションサービス, 63

システムサービス, 65

サンプルアプリケーション, 27

す

ステート管理, 25, 63

ストリーミング, 20

せ

セキュリティ, 25

セッション管理, 25, 63

設定ディレクトリ, 30

て

データストリーミング, 20

テンプレート, 24

と

トランザクション管理, 67

ふ

フェールリカバリサービス, 58

フォーマット

URL、マニュアルでの～, 9

プロトコルマネージャ, 59

分散トランザクション, 65

ま

マルチスレッドサポート, 20

り

リクエスト管理, 62

リスナ, 60, 61

ろ

ロードバランス, 18, 61, 66

