



Making Software Work Together™

IONA iPortal Application Serverä ホワイトペーパー

IONA Technologies
1999 年 12 月

IONA iPortal Suite、IONA iPortal Application Server、IONA iPortal Integration Server、IONA iPortal Server、IONA Orbix 2000 は、IONA Technologies PLC の商標です。

Orbix は、IONA Technologies PLC の登録商標です。

IONA Technologies PLCは本書に記載されている内容が正確であるものと確信していますが、特定用途への適合性や市場性の黙示保証など、本製品に対するこの種の保証は一切行いません。また、本書に含まれる誤りや、本製品の設置、作動、または運用に伴う偶発的または間接的な損害に対しては一切責任を負いません。

著作権表記

コピー、記録、その他のいかなる形式または手段を問わず、IONA Technologies PLCの文書による事前の承諾なしに、本書を一部たりとも複製したり、検索システムに保存したり、転送したりすることはできません。弊社は、本書に記載されている内容の使用に関して、第三者の知的財産権に対する責任は保証しません。また、本書に誤りや脱漏があっても、その責任は負いません。本書の内容やここで取り上げた機能は、予告なしに変更することがあります。

Copyright © 1999 IONA Technologies PLC. All rights reserved.

本書で取り上げた製品やサービスはすべて、それらの販売会社が指定している商標、サービス・マーク、または製品名で記載しています。

M2367

要約

IONA iPortal Application Server™ は、コンポーネントベースのエンタープライズ・アプリケーションを開発し、デプロイし、運用するためのアプリケーション・サーバです。iPortal Application Server はグラフィカルな環境を備えており、その環境の中で Enterprise JavaBeans (EJB) 1.1 コンポーネントを使用してアプリケーションをアセンブリし、それらのアプリケーションをデプロイし、稼働中にそれらの動作を中央から制御することができます。iPortal Application Server は、IONA が特許権を所有しているアダプティブ・ランタイム・テクノロジー (**Adaptive Runtime Technology™** :ART) を使用して構築されており、IONA の次世代プラットフォーム・ミドルウェアである Orbix 2000 の基盤となるものです。これにより、アプリケーション・サーバに、実証済みのスケーラビリティ、パフォーマンス、信頼性を提供するとともに、CORBA と COM コンポーネントの完全な統合と高性能マイクロカーネル・アーキテクチャを提供し、高度にフレキシブルで適合性の優れた開発・運用環境を実現します。IONA iPortal Application Server は、エンタープライズ・ポータルおよびその他の大規模な分散アプリケーションを開発、運用、および管理するためのマルチプラットフォーム・スイート製品である IONA iPortal Suite のメンバーです。

目次

はじめに	1
Enterprise JavaBeans を使用する利点	3
CORBA を使用する利点	4
Enterprise JavaBeans と CORBA の結合	4
iPortal Application Server のさらなる利点	5
iPortal Application Server の概要	6
iPortal Application Server アーキテクチャの概要	7
iPortal Application Server によるアプリケーションのアセンブリ	9
アプリケーションのコンポーネントの選択	9
コンポーネントの連結	10
アプリケーション特性の指定	11
セキュリティ要件の指定	11
トランザクション要件の指定	13
デフォルト・コンフィギュレーション値の指定	14
iPortal Application Server によるアプリケーションの配備	16
アプリケーションのインストール	16
アプリケーションのコンフィギュレーション	17
用語集	19
関連資料	22
連絡先の詳細	23

はじめに

私たちは、特定の業界で E-コマース (電子商取引) のパイオニアとなって一夜のうちに名を馳せた企業の名前をすべて列挙することができます。もちろん、こういった "amazon.com" (通販専門のインターネット書店) のような企業に対して、いわゆる "ブリックス・アンド・モルタル" 企業が存在します。これらの企業はインターネット上で効果的にアピールすることができなかったために、ビジネス上で大きなダメージを受けています。IONA では、このことを "ブリックス・アンド・モルタルの完敗" と呼んでいます。それぞれの大企業がこの現象をみて、会社が生き残るかどうかは E-ビジネス戦略をいかに俊敏かつ効果的に実現できるかで決まるということを学んだのです。

これらの大企業の E-ビジネス戦略は、エンタープライズ・ポータルという概念を中心にますますエスカレートしてきています。インターネットの利用という観点からすると、ポータルはエキサイティングな進化の過程を示しており、大企業の開発組織にとって、次に目指すべき大きな開発目標となっています。

エンタープライズ・ポータルは、次の機能や特性を備えていなければなりません。

- パーソナル性、フレキシビリティ、スケーラビリティに優れている
- ビジネス・ロジックを手早く作成し、運用することができる
- 統一された標準ベースのアーキテクチャを提供する
- パッケージ化された既存のレガシー・アプリケーションと緊密に統合する

ポータル・システムは、隔絶された状態で構築されるわけではありません。実際、大きな組織はすべて、広範囲のさまざまなテクノロジー、オペレーティング・システム、ハードウェア・プラットフォーム、およびプログラミング言語に対応しなければなりません。これらはそれぞれが特定の重要なビジネス・タスクに適したものであり、これらのすべてを連携、統合しないとビジネスはうまく機能しません。

新しいエンタープライズ・アプリケーションを産み出すためのプラットフォームは、既存のレガシー・アプリケーションからサードパーティによるパッケージ・ソフトウェアに至るまで、可能な限り広範なアプリケーションに適応できるものでなければならず、アプリケーション開発モデルは、企業が新しい市場機会やテクノロジー機会に俊敏に対応できるものでなければなりません。

Sun Microsystems の Java 2 Platform Enterprise Edition (J2EE) は、エンタープライズ・アプリケーションを開発し、運用するためのコンポーネント・ベースの標準モデルです。このモデルは多階層アーキテクチャを記述するもので、軽量クライアントが中央のビジネス・サービスにアクセスし、この中央のビジネス・

サービスがエンタープライズの個々の部分に存在する情報システムにアクセスします。このアーキテクチャを、図 1 に示します。

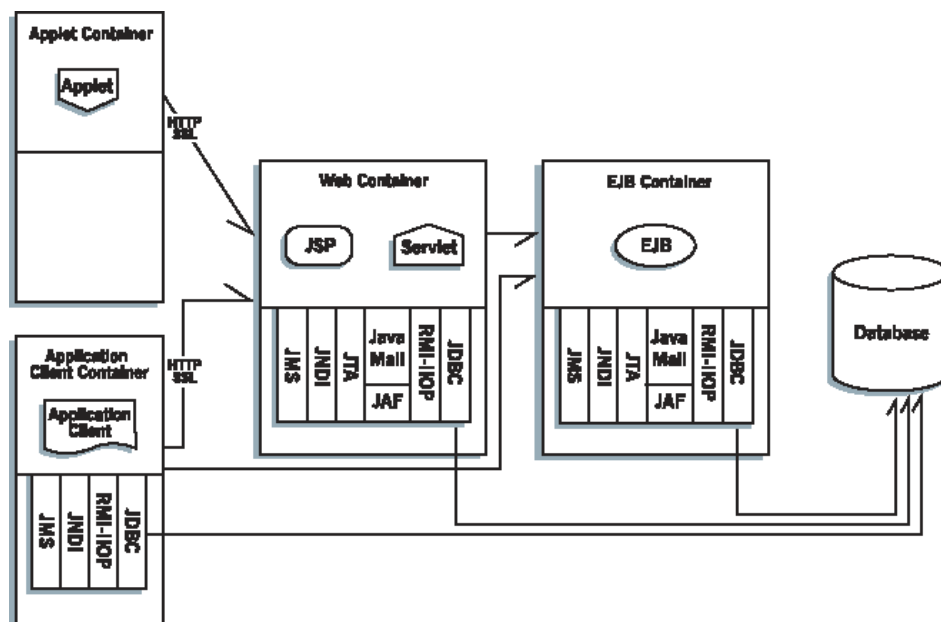


図1. J2EE アーキテクチャの概要

J2EE アプリケーションは、エンタープライズ・システムの間階層で動作し、バックエンド情報システムと統合して、エンタープライズ・サービスをエンドユーザ・クライアントにもたらしめます。J2EE は、ビジネス・ロジックを中間階層に配置し、クライアントには配置しません。このことが J2EE アプリケーションをスケーラブルで、アクセスしやすく、制御しやすいものにしています。なぜなら、クライアントの数によってアプリケーションの管理機能が影響されないからです。

J2EE では、中間階層のアプリケーションは EJB (Enterprise JavaBeans) コンポーネントで構築されます。EJB コンポーネントはコンテナの中で実行され、コンテナが、トランザクションやネットワーク・セキュリティの管理などのコンポーネントの分散プログラミング要件を請け負います。

iPortal Application Server は、J2EE プラットフォームで使用するための EJB 1.1 サーバです。これは、EJB アプリケーションを構築し、それらのアプリケーションをデプロイし、稼働時にそれらの動作を中央から監視し変更するために必要なツールを備えています。アプリケーションは EJB 1.1 コンテナの中で実行されますが、このコンテナは IONA の CORBA 製品セット上で構築されます。

iPortal Application Server では、IONA の Orbix 製品セットによって EJB コンテナの基礎となるインフラストラクチャを提供しています。セキュリティやトラン

ザクシオンなどの EJB コンテナのサービスは、対応する CORBA の各サービスの上に構築されます。iPortal Application Server は、EJB と CORBA を結合して、両方のテクノロジーのあらゆる利点を引き出します。

IONA iPortal Application Server はエンタープライズ・ポータルの開発、運用、管理の問題に対処した業界初のマルチプラットフォーム・スイート製品である IONA iPortal Suite™ のメンバーで、iPortal Application Server™、iPortal Integration Server™、iPortal Server™、および Orbix 2000 という 4 つの製品を緊密に統合する新しいアーキテクチャを基盤としています。IONA iPortal Suite によって、E-ビジネスのプロジェクト・チームの各メンバーが一体化され、既存のエンタープライズ・アプリケーションと統合された新たなエンタープライズ・ポータルを簡単に運用、管理できるようになります。

Enterprise JavaBeans を使用する利点

EJB は、新しいエンタープライズ・サービスを産み出すための、コンポーネントベースの標準モデルです。EJB の 開発モデルは、コンポーネントを容易に開発し、再利用し、あらゆる標準仕様の EJB システムに移植できるように設計されています。

EJB により、ユーザはビジネス・ロジックの構築に専念し、分散プログラミングの複雑さから解放されるので、開発作業が容易になります。EJB は、コンテナの中で実行されます。セキュリティ、永続性、トランザクション処理サービスなどのエンタープライズ・コンポーネントが必要とする分散コードの開発はコンテナ・プロバイダが請け負います。コンポーネントは、これらのサービスを透過的に使用します。したがって、ユーザは、低レベルの分散プログラミングを行う必要は一切ありません。

EJB があらゆるコンテナ上で実行できることを保証するために、コンテナがコンポーネントにサービスを提供するためのインタフェースが完全に標準化されています。その結果、各コンテナは異なる方法で分散サービスを実装できるにもかかわらず、各コンポーネントはすべての標準 EJB コンテナに移植可能です。

EJB はあらゆるコンテナで実行できることに加えて、多くの異なるアプリケーションで再利用可能です。特定のアプリケーション用にコンポーネントの動作をカスタマイズするには、そのコンポーネントのソース・コードではなくコンフィギュレーションを修正します。

iPortal Application Server を利用することにより、Java 統合開発環境 (IDE) を使用して開発した EJB や コンポーネント・ベンダーが提供する EJB をアプリケーションにアセンブリすることができます。iPortal Application Server は、わかりやすいポイント・アンド・クリック式のインタフェースを備えており、これを使用してコンポーネントを結合し、カスタマイズすることができます。

iPortal Application Server のインタフェースによって、このグラフィカルなアプローチがアプリケーションの運用と管理にまで拡張されます。中央のコンソール

から、アプリケーションを運用環境にインストールし、アプリケーションの運用を制御し、動的にコンフィギュレーションを設定することができます。

CORBA を使用する利点

CORBA は、分散オブジェクトを作成するためのオープンな標準ソリューションです。CORBA を使用することにより、オブジェクトの作成に使用されるプログラミング言語や、オブジェクトが実行されているオペレーティング・システムやプラットフォームに関係なく、オブジェクトをネットワーク上で直接やりとりできるようになります。

CORBA ソリューションはあらゆる共通環境で使用可能であり、Visual Basic、C、C++、Java、Ada、Smalltalk、および COBOL などの言語で作成された、組み込みシステム、PC、UNIX ホスト、およびメインフレーム上で動作するアプリケーションを統合します。これらの環境で動作する CORBA オブジェクトは、シームレスに相互運用することができます。IONA の CORBA と COM 間の動的ブリッジである OrbixCOMet を利用して、COM オブジェクトと相互運用させることも可能です。

CORBA は広範な利用が可能であり、分散ビジネス・オブジェクトに必要な機能をすべてサポートする拡張インフラストラクチャを提供します。このインフラストラクチャには、トランザクション、セキュリティ、メッセージングなどの重要な分散サービスが含まれます。

また、CORBA は成熟度の高いインフラストラクチャを提供することができ、既存の複数のエンタープライズを統合する際に非常に重要な役割を果たします。これらの要因が組み合わされて、CORBA は今日最も重要なシステム統合テクノロジーとなっています。

Enterprise JavaBeans と CORBA の結合

CORBA と EJB は相互に補完し合うテクノロジーです。EJB コンポーネントは、サーバ・サイドのコンポーネントの開発と運用のための優れたモデルを提供します。CORBA によって、EJB を、きわめて広範な言語およびプラットフォームを使用して構築されたアプリケーションと統合することができます。

iPortal Application Server を使用することにより、EJB コンポーネントは、Windows ホスト、UNIX ホスト、またはメインフレーム上で実行されている CORBA オブジェクトと直接やりとりできます。また、OrbixCOMet を使用して COM コンポーネントとも完全に相互運用が可能です。

EJB と CORBA の結合の例として、次のようなものがあります。

- EJB は、OrbixCOMet を使用して、リモートの Microsoft Excel スプレッドシートを動作させることができます。

- Orbix Wonderwall は、EJB コレクションのための安全なインターネット・ファイアウォールを提供できます。
- Orbix for OS/390 を使用することにより、EJB は、メインフレーム上で稼働している COBOL アプリケーションとシームレスに統合できます。
- OrbixOTM を使用することにより、C++ アプリケーションを、EJB と CORBA を組み合わせた 2 相コミットを使用する完全なトランザクション伝播を含むトランザクション・サーバ・グループに収容することができます。
- OrbixNotification は、EJB システムに対し、完全なメッセージング機能を提供します。これにはコンテンツ・ベースのフィルタリングも含まれます。

EJB と CORBA を連携させて使用する利点は、相互運用性だけではありません。iPortal Application Server では、EJB の機能は、実証済みのスケーラビリティ、信頼性、およびパフォーマンスを備えた IONA 製品の上に構築されています。アプリケーションは、十分に確立され広範に使用されているサービスを活用できる運用環境にデプロイされます。CORBA インフラストラクチャは、分散エンタープライズの堅固な土台となります。

iPortal Application Server のさらなる利点

iPortal Application Server は、EJB と CORBA の結合以上のことを行います。iPortal Application Server 環境には多くのユニークな機能が含まれており、企業規模でアプリケーションを開発し、運用できます。たとえば、次のようなことが可能です。

- EJB アプリケーションの動作とコンフィギュレーションは、XML ドキュメントに記述されます。iPortal Application Server では、明確なビジュアル・パラダイムによって運用とコンフィギュレーションを管理することができます。iPortal Application Server により、すべての XML ファイルの内容が自動的に管理されます。
- iPortal Application Server を使用すれば、中央のコンソールからその運用環境内のアプリケーションをコンフィギュレーションできます。各アプリケーションは、複数の事前定義済みコンフィギュレーションを保持することができます。
- アプリケーションの可用性に影響を与えずに、任意の時点でアプリケーションに新しいコンフィギュレーションを適用することができます。
- EJB コンテナの中で実行されているコンポーネントの場所は、コンポーネントを使用しているクライアントにとって透過的です。したがって、中央のコンソールから、クライアントに影響を与えずに、稼働させたままでコンポーネントを移動することが可能です。

iPortal Application Server の概要

iPortal Application Server を使用して、単一のユーザ・インタフェースから、エンタープライズ・アプリケーションのライフサイクルの全段階を制御します。このインタフェースを図 2 に示します。

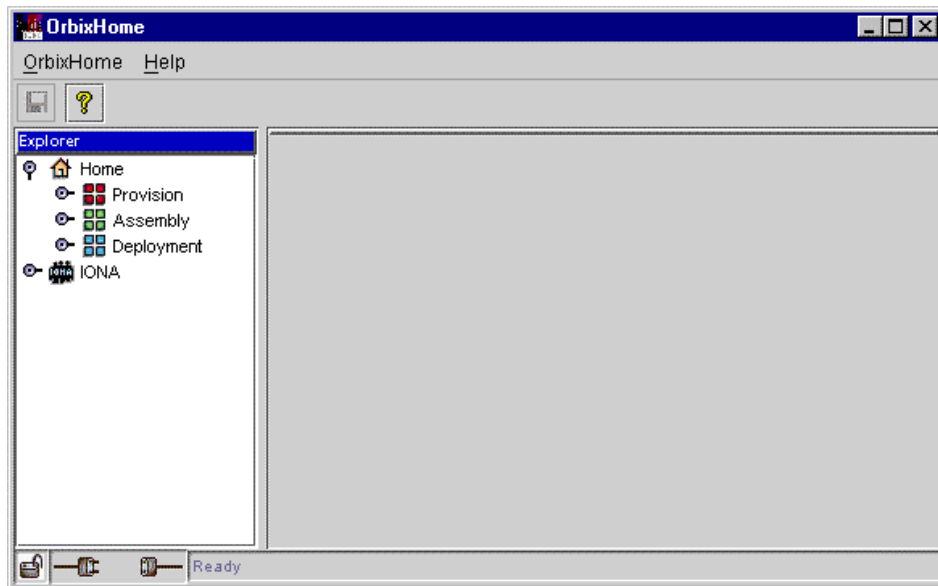


図2. iPortal Application Server ユーザ・インタフェース

iPortal Application Server インタフェースには、主要な 4 つの領域があります。各領域は、アプリケーションのライフサイクルの主要段階に対応しています。

- コンポーネントの準備。iPortal Application Server でアプリケーションを作成する前に、既存のコンポーネントをインポートします。たとえば、JAVA IDE を使用してコンポーネントを開発することも、コンポーネント・ベンダーから購入することもできます。
- アプリケーションのアセンブリ。この領域で、完成しているアプリケーションにコンポーネントを結合することができます。
- アプリケーションのディプロイメント。アプリケーションをアセンブリした後、そのアプリケーションを運用環境にディプロイします。このディプロイメントインタフェースを使って、システム内の使用可能なコンテナを表示し、コンテナにアプリケーションをインストールし、アプリケーションのコンフィギュレーションを設定することができます。

- アプリケーションの管理。管理インターフェースを使用して、アプリケーションを起動および停止させ、動作を監視し、アプリケーションのコンフィギュレーションを動的に変更し、稼働させたままコンポーネントをアップグレードすることができます (アプリケーション管理機能は、iPortal Application Server の現在のベータ・リリースでは使用できません)。

iPortal Application Server アーキテクチャの概要

図 3 に、iPortal Application Server のインターフェースを使ってアクセスすることができる分散アーキテクチャを示します。

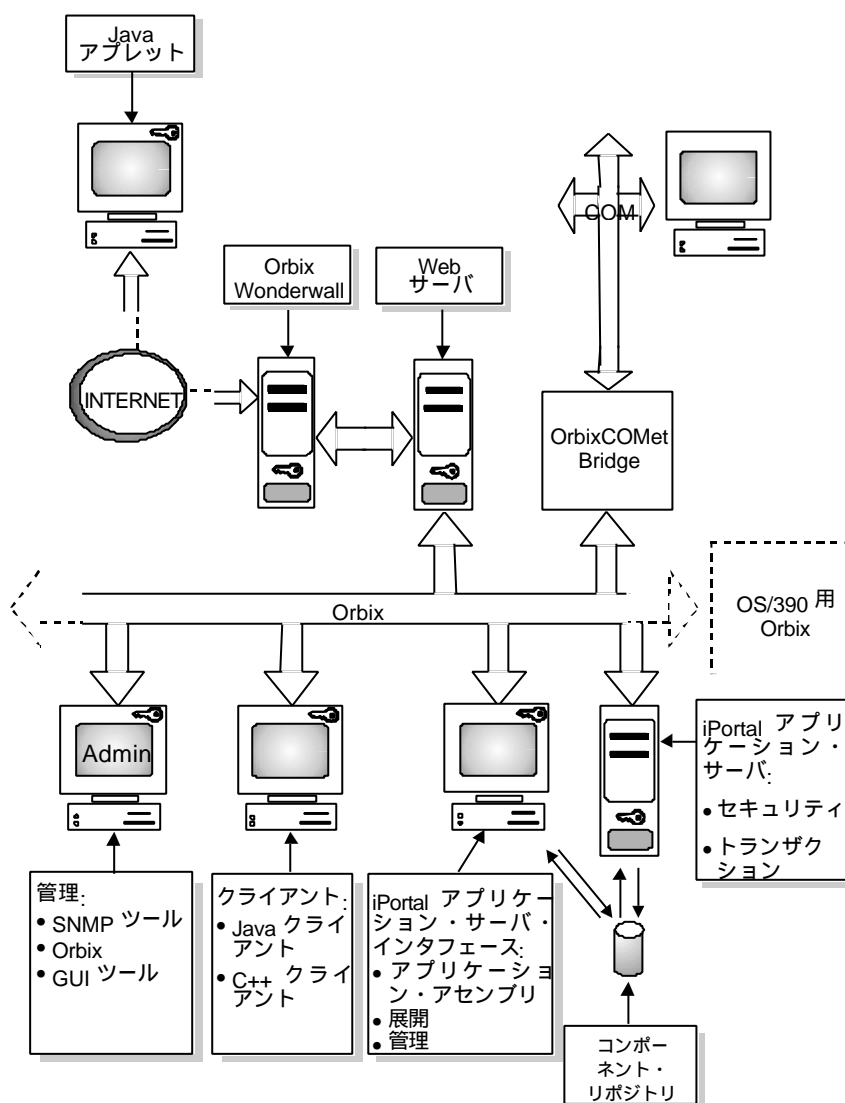


図3. iPortal Application Server アーキテクチャ

iPortal Application Server によって、使用可能なコンポーネントが中央のコンポーネント・リポジトリに格納されます。iPortal Application Server を使用してアプリケーションをアセンブリする際には、このリポジトリから必要なコンポーネントを選択します。これで、iPortal Application Server によって、アセンブリされたアプリケーションが格納されます。

iPortal Application Server コンテナは、アプリケーションのホストである各サーバ上で実行されます。このコンテナには、OrbixSSL や OrbixOTS などの IONA 製品を使って実装された、EJB 1.1 規格に規定されているすべてのサービスが含まれます。

特に、iPortal Application Server コンテナは、次の機能を提供します。

- セッションおよびエンティティ Bean の完全サポート
- RMI-IIOP のサポート
- コンテナおよび Bean で管理される永続性と、Java Database Connectivity (JDBC) による主要なリレーショナル・データベース・システムのサポート
- Java Transaction API (JTA) によるトランザクション・サポート。これには、2 相コミット、さらに、コンポーネント、クライアント、およびコンテナで管理されるトランザクション境界が含まれます。
- Java Naming and Directory Interface (JNDI) のサポート
- セキュリティ管理
- ネットワークの障害回復
- 稼動システム内の EJB アプリケーションに対する動的なコンフィギュレーション変更
- 複数のアプリケーション・コンフィギュレーションのグラフィカルな選択
- EJB コンテナ間でのアプリケーションの動的な移動

サーバ側のコンポーネントにアクセスするクライアントは、ローカル・ネットワーク内に存在する場合もあるし、Web サーバを通じてインターネットに配布されることもあります。たとえば、Java アプレット、Java Server Pages (JSP)、サーブレット、または、コンポーネントと通信する CORBA クライアント・アプリケーションを通じて、情報をユーザに配布することができます。

iPortal Application Server によるアプリケーションのアセンブリ

Symantec 社の Visual Café のような統合開発環境 (IDE) における EJB のサポートによって、個々のコンポーネントの開発のタスクが簡素化されます。場合によっては、個々のコンポーネントが完全なアプリケーションの役割を果たすこともあります。しかし、コンポーネントは、より複雑なアプリケーションの一部であることがほとんどです。アプリケーションには、複数の EJB と、JSP や CORBA オブジェクトのような他のタイプのサーバ側アプリケーション・ロジックが含まれる場合があります。

たとえば、オンライン・ショッピング・アプリケーションを構築するには、ショッピング・カート Bean とカタログ Bean を組み合わせてビジネス・ロジックを実装し、ユーザが品目をブラウズし、選択し、購入することができるようにするための JSP を使用することになるでしょう。これらの EJB は、その会社のカタログ・データベースや注文処理データベースとやりとりすることになります。

EJB 1.1 仕様には、アプリケーションのアセンブリに関する明確なプロセスが定義されています。このプロセスに正しく従えば、一連のコンポーネントが相互に機能し合っって単一の再利用可能なエンティティが形成されます。

EJB アプリケーションをアセンブリするには、アプリケーション・コンポーネントとそのアプリケーションを記述した XML ドキュメントを収容する ejb-jar ファイルを作成します。たとえば、XML ドキュメントには、アプリケーションに関連するセキュリティ上の役割 (role)、コンポーネント間の関係などが記述されます。このドキュメントは、多くの場合、長くて複雑なものになります。XML ドキュメントの作成は、時間がかかる難しい作業になる場合があります。

iPortal Application Server にはグラフィカルな環境が提供されており、ユーザはこの環境の中で EJB 1.1 規格に完全に適合するアプリケーションをアセンブリすることができます。iPortal Application Server によって自動的に ejb-jar ファイルが生成され、すべての関連する XML ドキュメントが作成されます。ユーザは、簡単なグラフィカル・システムを使用してアプリケーションの特性を指定します。

アプリケーションのコンポーネントの選択

アプリケーションをアセンブリするには、まず、アプリケーションに必要な EJB コンポーネントをインポートします。これらのコンポーネントは、ユーザが JAVA IDE で作成した EJB である場合や、ユーザがカスタマイズした既存の EJB である場合、あるいはサードパーティから入手した EJB の場合もあるでしょう。各 EJB は、1 つの ejb-jar ファイルにパッケージされます。EJB を iPortal Application Server にインポートするには、ローカル・ディスクまたは

ローカル・ネットワーク上の、あるいはインターネットを介して、該当する jar ファイルの場所を指定します。

EJB をインポートすると、iPortal Application Server によってそれがコンポーネント・リポジトリに追加されます。コンポーネント・リポジトリとは、すべてのコンポーネントの中央保管場所です。コンポーネントをリポジトリにインポートしたら、図 4 に示すように、アプリケーションに必要なコンポーネントを選択し、コンポーネント・リポジトリからアセンブリ作業領域にドラッグします。

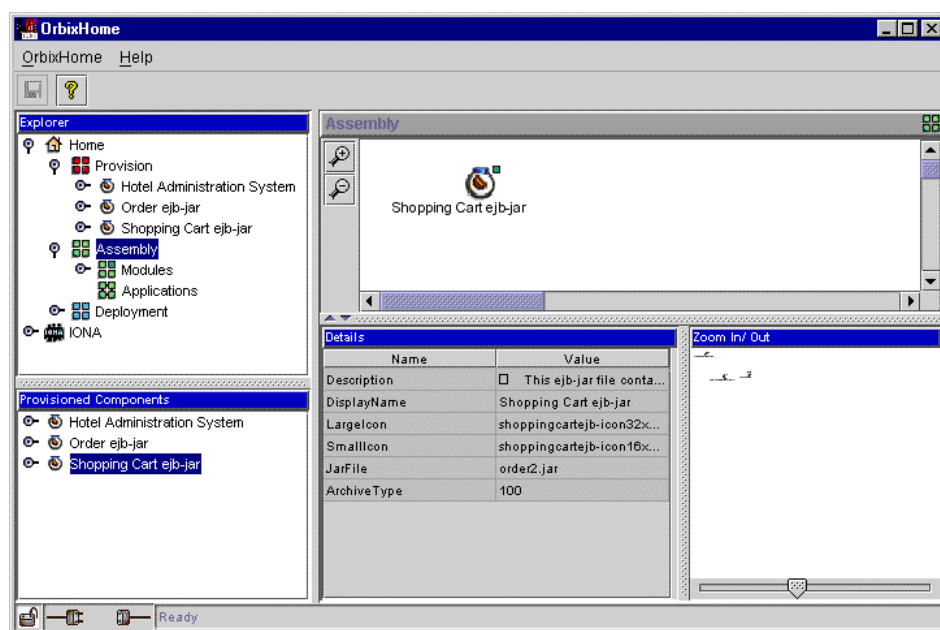


図4. アプリケーションのコンポーネントの選択

コンポーネントの結合

このインタフェースにおいては、アセンブリ作業領域には選択されているコンポーネントが表示されます。また、リソースなどの、それらのアプリケーション・コンポーネントが必要とするその他のアプリケーション特性も表示されます。たとえば、コンポーネントの中には他のコンポーネントへの明示的な参照を宣言するものもあります。アセンブリ作業領域の中でこのことを表現するには、図 5 に示すように、それらのコンポーネントを接続する線を描きます。

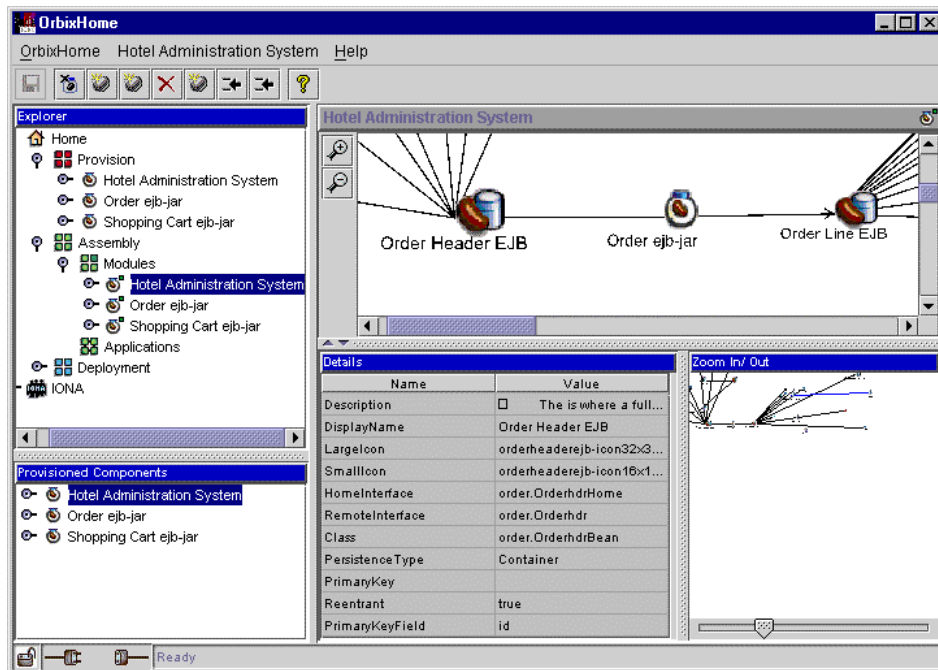


図5. コンポーネント間の関係の定義

アプリケーション特性の指定

アプリケーションを完成させるには、iPortal Application Server を使用して、アプリケーションのセキュリティ要件、トランザクション要件、およびコンフィギュレーション設定項目のデフォルト値を指定します。

セキュリティ要件の指定

EJB は、1 つ以上のセキュリティ上の役割 (*role*) を担っています。セキュリティ上の役割とは、その EJB にアクセスできるユーザのタイプの論理名およびそのユーザ・タイプに関連付けられている一連の許可 (*permission*) です。

EJB アプリケーションをアセンブリするときには、アプリケーション・レベルのセキュリティ上の役割を定義します。次に、Bean レベルのセキュリティ上の役割がアプリケーションのセキュリティ上の役割にどのように対応するのかを指定します。たとえば、2 つの Bean が類似したセキュリティ上の役割を持つ場合であっても、たとえば *shopper* と *customer* といった、異なる名前を使用します。これらの役割は、1 つのアプリケーション・レベルの役割に対応します。

図 6 は、"receptionist" というセキュリティ上の役割を iPortal Application Server で簡単に表現したものです。この役割は一連のメソッド許可 (*Method*

Permission) に関連付けられていて、ここにはこのアプリケーションのコンポーネントの中でこのタイプのユーザがアクセスできるメソッドが記述されています。

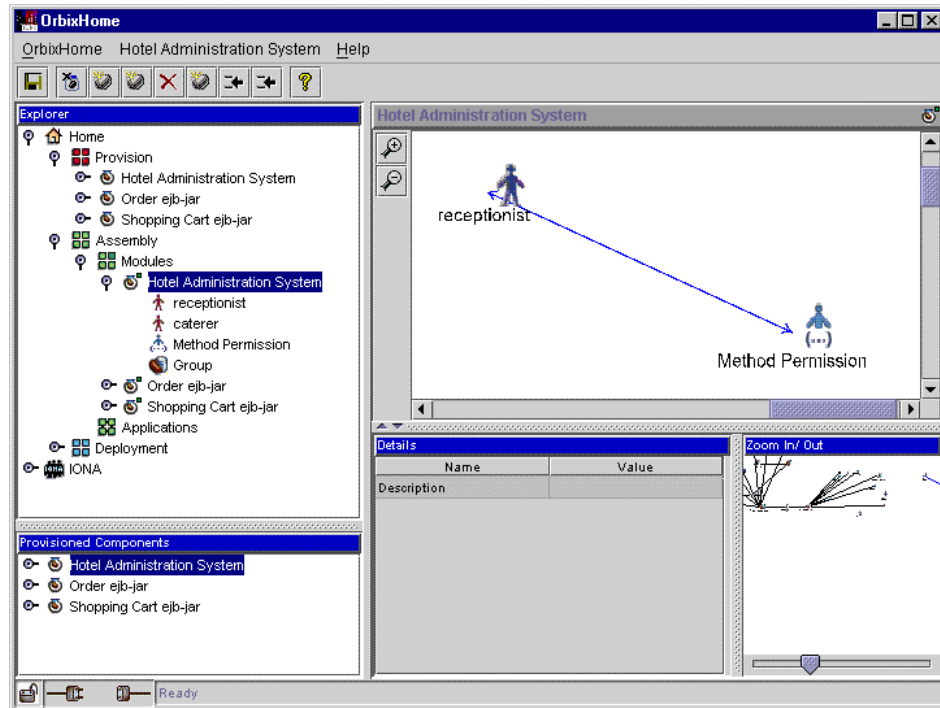


図6. アプリケーション・レベルのセキュリティの役割の作成

アセンブリ作業領域には、各コンポーネントに関連付けられている Bean レベルの役割も表示されます。たとえば、図7は、"orders" という Bean レベルの役割を示しています。Bean レベルの役割をアプリケーション・レベルの役割に関連付けるには、Bean レベルの役割アイコンをドラッグして、アプリケーション・レベルの役割アイコンの上にドロップします。

iPortal Application Server を使用してアプリケーションをディプロイする際には、ディプロイメント環境の中で、ユーザ・アカウントなどの実際のセキュリティ・エンティティにセキュリティ上の役割を割り当てます。

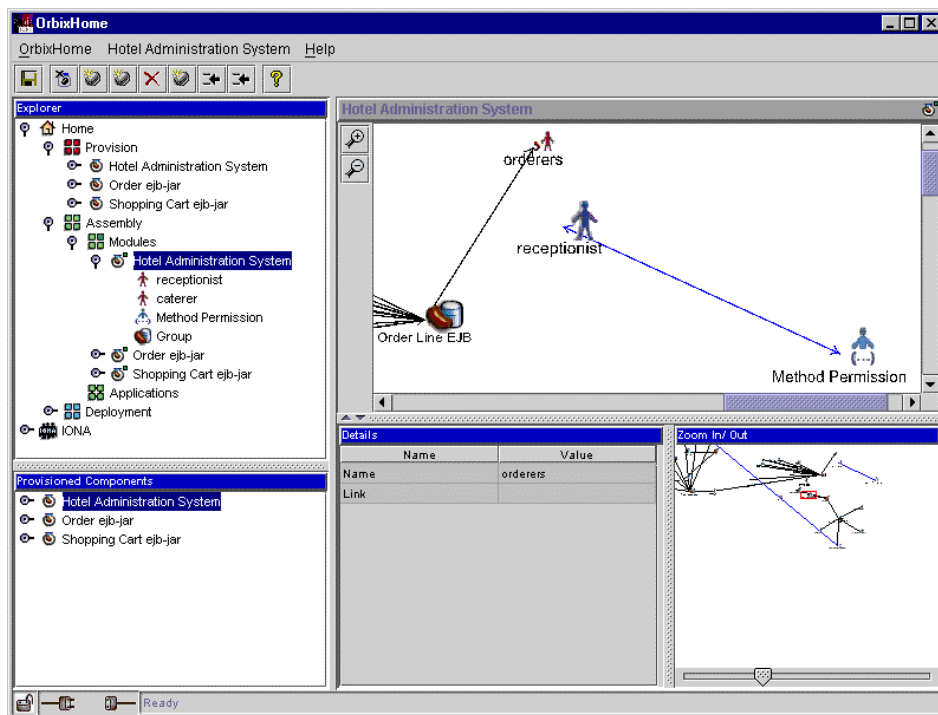


図7. アプリケーションのセキュリティ上の役割の割り当て

トランザクション要件の指定

EJB プロバイダがコンポーネントを開発する際には、プロバイダは、コンポーネントがトランザクションを必要とするかどうか、また、どのタイプのトランザクション制御を使用するかを指定します。2つのタイプのトランザクション制御が使用可能で、*Bean 管理トランザクション*と*コンテナ管理トランザクション*があります。*Bean 管理トランザクション*の場合には、EJB はメソッド呼び出しを使用して、いつトランザクションを開始し終了するかを制御します。コンテナ管理トランザクションの場合には、コンテナがトランザクションを制御します。

アプリケーションをアセンブリする際には、*Bean 管理トランザクション*を使用する EJB の場合には、特別の処置は必要ありません。コンテナ管理トランザクションの場合には、コンテナがその EJB に対する特定のメソッドについてトランザクションを管理する方法をユーザが指定する必要があります。

iPortal Application Server のアセンブリ作業領域には、アプリケーションのコンポーネント間のトランザクションのフローが表示されます。iPortal Application Server のインタフェースを使用して、特定のコンポーネントの特定のメソッドが必ずトランザクションを使用しなければならないこと、トランザクションを使用できること、トランザクションを決して使用できないこと、などを指定することができます。たとえば、図 8 では、iPortal Application Server において、特定の

コンポーネントのすべてのメソッドにトランザクションが必要であることを指定する方法を示しています。

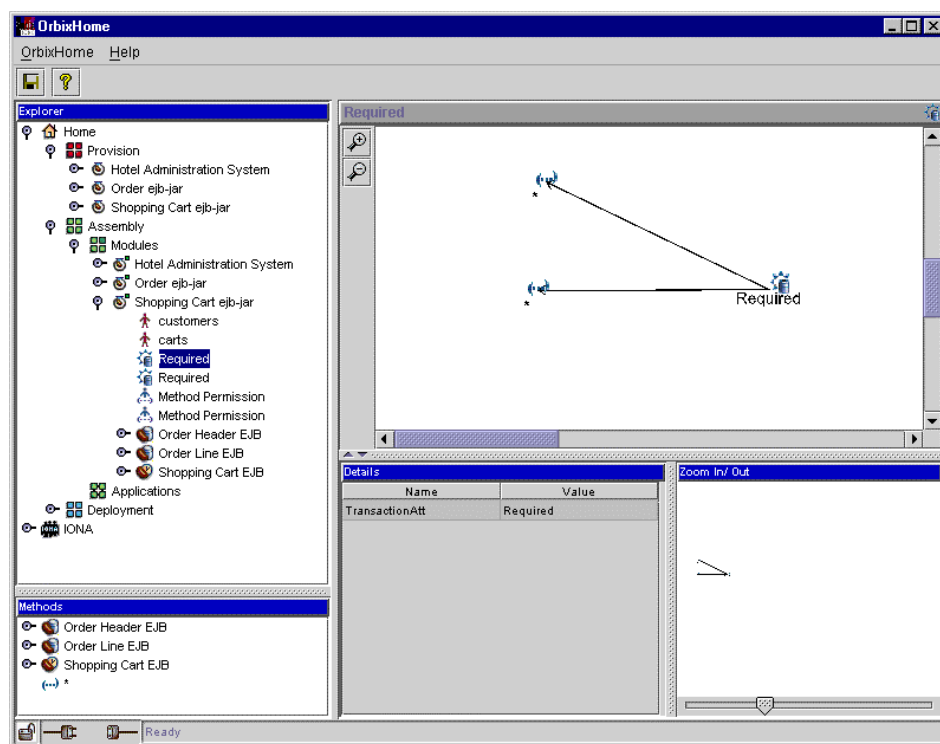


図8. アプリケーションのトランザクション要件の指定

デフォルト・コンフィギュレーション値の指定

アプリケーションの各コンポーネントは、アプリケーションのユーザがコンフィギュレーションの設定値を指定するように要求することができます。ユーザは、アプリケーションをアセンブリする際に、デフォルトのコンフィギュレーション設定値を指定することができます。

図9は、アセンブリ作業領域の中のアプリケーション・コンポーネントと、コンフィギュレーション値 `maxTotal` に対して関連付けられているデフォルト値を示しています。この値を編集するには、この値を選択し、[Details] パネルの中で設定を変更します。

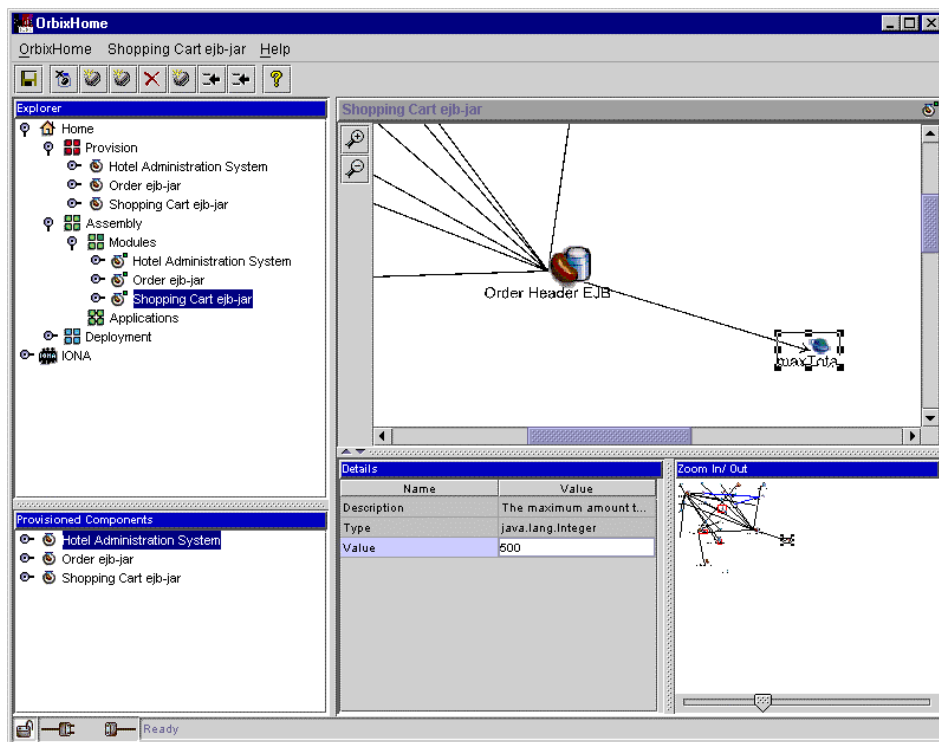


図9. デフォルトのコンフィギュレーション値の指定

iPortal Application Server によるアプリケーションのディプロイメント

アプリケーションのアセンブリが完了したら、アプリケーションを運用環境にディプロイする準備ができたことになります。iPortal Application Server によって、中央コンソールからネットワークのホスト上にアプリケーションをインストールし、コンフィギュレーションを行い、そこで稼働させることができます。

アプリケーションのインストール

図 10 に示すように、iPortal Application Server のディプロイメント作業領域には、システム内の iPortal Application Server コンテナおよびユーザがアセンブリしたすべてのアプリケーションが表示されます。ネットワーク内のどこからでも、iPortal Application Server コンテナを実行することができます。コンテナにアプリケーションをインストールするには、ディスプレイからアプリケーションをドラッグし、目的のコンテナ上にドロップします。iPortal Application Server が、インストールに関するあらゆる専門処理を実行します。

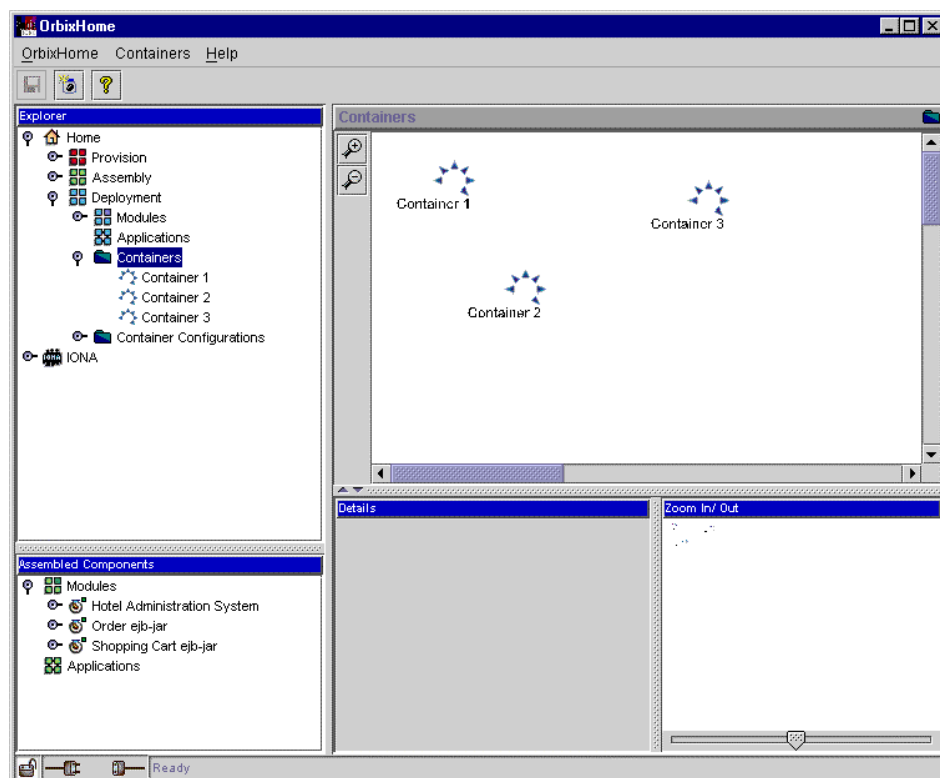


図10. コンテナへのアプリケーションのインストール

アプリケーションのコンフィギュレーション

アプリケーションをどのようにコンフィギュレーションするかは、アプリケーションが実行される環境によって異なります。たとえば、データベースを使用するアプリケーションで考えてみましょう。テスト中には、特に指定されたテスト・データベースを使用するようにコンフィギュレーションすることが考えられます。このアプリケーションが完全にデプロイされた時点では、テスト・データベースではなく本番データベースを使用することになります。

iPortal Application Server では、コンテナごとに複数のコンフィギュレーションを作成することができます。図 11 では、1 つのコンテナに対して、Base Config、High Watermark、および Low Watermark という名前の 3 つのコンフィギュレーション・セットが示されています。各コンテナは、コンフィギュレーションをいくつでも保持することができ、アプリケーションを停止して再起動させなくても、稼働させたままアクティブ・コンフィギュレーションを変更することができます。

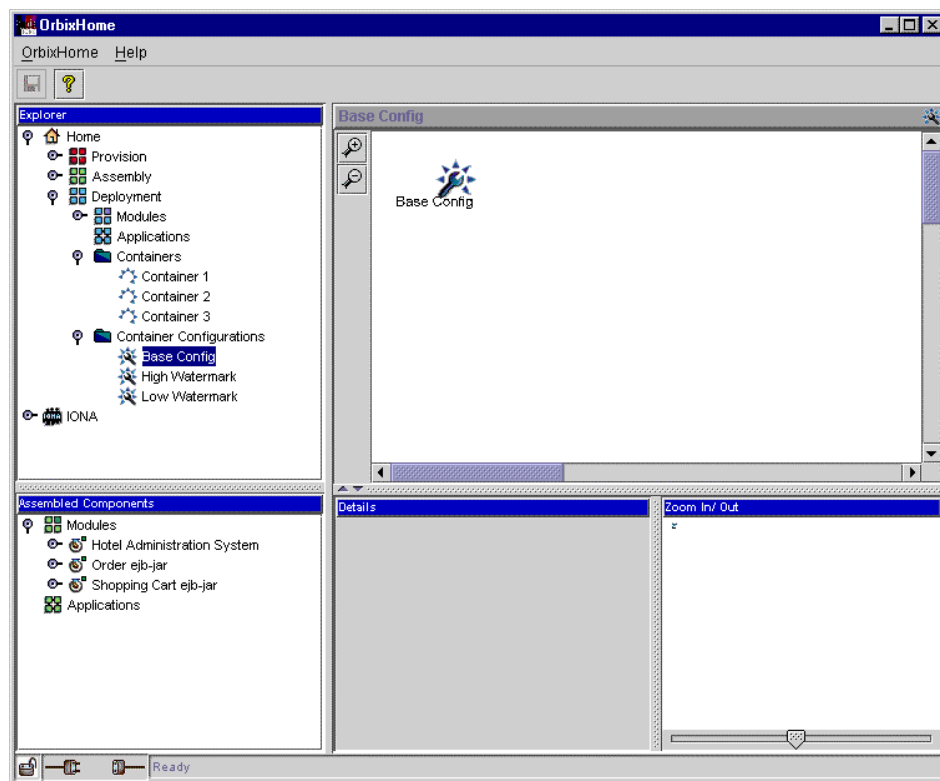


図11. コンテナ・コンフィギュレーションの作成

コンフィギュレーションを作成するには、iPortal Application Server のグラフィカルな環境を使用して、コンポーネントにとって必要なデータ・ストアを運

用環境の実際のデータベースにマップします。たとえば、EJB が、`java:comp/jdbc/Orders` という名前のデータベースをアドレス指定する場合があります。コンフィギュレーションを使って、この名前をリレーショナル・システムの中の `SALES` という名前のデータベースにマップすることができます。

コンフィギュレーションには、アプリケーションのセキュリティ要件を、運用環境で使用可能なセキュリティ・メカニズムに関連付ける情報を設定することができます。たとえば、システムでの組織的役割が LDAP 準拠のディレクトリに格納されている場合、これらの役割を、アプリケーションが規定しているセキュリティ上の役割に対応させる方法を指定することができます。デプロイメント作業領域には、セキュリティ上の役割と、役割を目的の環境に関連付けるためにユーザが入力する基本フィールドが表示されます。

複数のコンフィギュレーションを使用することには、多くの利点があります。たとえば、それによってシステム管理者は、処理負荷が予想より高くなる期間の計画を立てることが可能になります。毎週の売上報告によって、月曜の朝にユーザからの需要が高いことを示す結果が得られた場合には、アプリケーションを実行するコンテナを、より多くのメモリ常駐コンポーネントの作成を許すプロファイルに切り換えて、増加分の需要を処理するようにします。そして、月曜日以外の曜日については、通常レベルにスケールダウンさせます。同様に、昼間と夜間のサービスに対して異なるコンフィギュレーションを使用することもできます。

iPortal Application Server は明解なグラフィカル・インタフェースを備えており、このインタフェースを使用してコンフィギュレーションを作成することができます。アプリケーションの稼働時に、中央コンソールからそれらのコンフィギュレーションをアプリケーションに適用させることができます。

用語集

ActiveX COM をベースにした広範な技術を記述するために Microsoft が使用している用語。ActiveX コントロールは、ActiveX 環境で作成されるコンポーネントで、インターネットを介してダウンロードできます。

Adaptive Runtime Technology コア・アプリケーション機能およびサービスの動的な運用とコンフィギュレーションのための IONA の革新的なランタイム・アーキテクチャ。

COM (Component Object Model) 仕様に準拠した任意のアプリケーションがアクセスすることのできるオブジェクトの開発およびサポートのための Microsoft のフレームワーク。

CORBA (Common Object Request Broker Architecture) OMG が定義した、分散オブジェクト間の通信に関する標準アーキテクチャ。ORB は、分散コンポーネントの開発および運用のための、OMG のより広範なフレームワークのコア要素です。

EJB (Enterprise JavaBeans) Java で作成されたサーバ側コンポーネント用に Sun が開発したコンポーネント仕様。

HTTP (Hypertext Transfer Protocol) TCP/IP 上の単純なプロトコル層で、Web ブラウザが Web ページや他のファイルをダウンロードするために使用します。

IDL (Interface Definition Language) CORBA 標準言語で、プログラマはこれによって CORBA オブジェクトへのインタフェースを定義することができます。クライアントはこれらのインタフェースを使用して、ネットワーク上のオブジェクトにアクセスします。

IIOP (Internet Inter-ORB Protocol) 分散アプリケーション間の通信に関する CORBA 標準プロトコル。IIOP は、TCP/IP より上位のプロトコル層として定義されています。

IOR (Interoperable Object Reference) CORBA オブジェクトの位置、固有の ID、およびサポートするサービスを指定します。

JDBC (Java Database Connectivity) Java アプリケーションからデータベースにアクセスするための標準インタフェース。

OMG (Object Management Group) 分散型オブジェクト指向プログラミングのための標準フレームワークを定義することを目的とするコンソーシアム。OMG は、CORBA 仕様を策定します。

ORB (Object Request Broker) ORB は、クライアントと分散オブジェクトの間の仲介者の役割を果たすミドルウェア・コンポーネントです。ORB は、ネットワークを介してクライアントとオブジェクトの間でメッセージの受け渡しを行います。ORB により、ハードウェア、オペレーティング・システムやプログラミング言語の違いなどの分散システムに起因する煩雑さからアプリケーション・プログラマは解放されます。

OTS (Object Transaction Service) OMG が定義した、標準 CORBA サービスの 1 つで、CORBA システムにおける分散トランザクションをプログラミングする方法を記述しています。

SNMP (Simple Network Management Protocol) TCP/IP 上の簡易なプロトコル層で、本来は大規模ネットワークのノードをモニタリングするために開発されたものです。

SSL (Secure Sockets Layer) Netscape Communications が開発したオープンなセキュリティ標準。TCP/IP ネットワークを介した通信を保護するために使用されます。

TCP/IP (Transmission Control Protocol/Internet Protocol) TCP/IP は、ホストをインターネット、イントラネット、およびエクストラネットに接続するために使用されるプロトコルの基本スイートです。

XA インタフェース X/Open グループが定義したインタフェースで、Distributed Transaction Processing 標準の一部です。データベース・トランザクション (またはその他のリソース・トランザクション) をトランザクション・モニタによって制御できるようにします。

インターネット グローバルで分散的なコンピュータ・ネットワーク。イントラネットとは異なり、インターネットへのアクセスに制限はありません。

イントラネット TCP/IP プロトコルに基づいたネットワークで、組織内の情報を共有するために使用されます。イントラネットは、通常、組織内のメンバーまたは社員でなければアクセスできません。

オブジェクト オブジェクト指向プログラミングにおける単一のソフトウェア・エンティティで、データとそのデータを操作する手続きで構成されます。CORBA では、オブジェクトはネットワークのどこにでも存在することができます。CORBA オブジェクトの機能には、IDL で定義されたインタフェースを通じてアクセスします。

オペレーション IDL 定義の関数またはメソッド。オペレーションは IDL インタフェースで定義され、CORBA オブジェクト上で呼び出すことができます。

クライアント サーバからのサービスを要求するプログラム。

サーバ クライアントにサービスを提供するプログラム。CORBA サーバは、CORBA オブジェクトのコンテナの役割を果たし、クライアントは IDL インタフェースを使用してこれらのオブジェクトにアクセスできます。

デーモン 常時実行されているプログラムで、定期的なサービス要求を処理します。

ファイアウォール。他のネットワークのユーザによる無許可アクセスから内部ネットワークのリソースを保護するシステム。

ミドルウェア 複数の別個のソフトウェアを接続するソフトウェア。

要塞ホスト 外部ネットワークと内部ネットワークの間のファイアウォール・セキュリティを提供する専用ホスト。

レガシー・アプリケーション 現在のテクノロジーより古い言語、プラットフォーム、あるいは技法を使用している既存のアプリケーション。

関連資料

1. Object Management Group (OMG). *The Common Object Request Broker: Architecture and Specification, Revision 2.1*. August 1997¹.
2. Object Management Group (OMG). *CORBA services: Common Object Services Specification*. March 1995.
3. Sun Microsystems. *Enterprise JavaBeans 1.1 Specification*. June 1999.
4. Sun Microsystems. *Java 2 Platform Enterprise Edition Specification, v1.2*.
5. Slama, Dirk, and others, *Enterprise CORBA*. Prentice Hall, March 1999.
邦訳 『Enterprise CORBA - CORBA による実用システムの設計ガイドライン - 』 ピアソン・エデュケーション.
6. Geraghty, Ronan, and others. *COM/CORBA Interoperability*. Prentice Hall, January 1999.

¹ OMG ドキュメントは、次のサイトから入手できます。
<http://www.omg.org>

連絡先の詳細

日本アイオナテクノロジーズ株式会社

〒107-0052

東京都港区赤坂 3-21-16

赤坂 3 丁目ビル 7F

電話: 03-3560-5611

ファックス: 03-3560-5612

Email: info-japan@iona.com

IONA Technologies PLC

The IONA Building

Shelbourne Road

Dublin 4

Ireland

電話: +353 1 637 2000

ファックス: +353 1 637 2888

IONA Technologies Inc.

200 West St

Waltham, MA 02451

USA

電話: +1 781-902-8000

ファックス: +1 781-902-8001

IONA の FTP サイト ftp.iona.com

World Wide Web: <http://www.iona.com/>