

**WebSphere® Application Server V4.0 for z/OS
and OS/390**



J2EE アプリケーションのアセンブル

**WebSphere® Application Server V4.0 for z/OS
and OS/390**



J2EE アプリケーションのアセンブル

お願い

本書および本書で紹介する製品をご使用になる前に、159ページの『付録B. 特記事項』に記載する一般情報をお読みください。

本書は、WebSphere Application Server V4.0 for z/OS and OS/390 (5655-F31) に適用されます。また、新版で特に指示のない限り、その後のリリース、およびモディフィケーション・レベルにも適用されます。

WebSphere Application Server V4.0 for z/OS and OS/390 資料の最新バージョンは、下記の Web サイトにあります。
<http://www.ibm.com/jp/software/websphere/appserv/>

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)

原典： SA22-7836-00
WebSphere® Application Server V4.0 for z/OS and OS/390
Assembling J2EE Applications

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 2001.5

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

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

© Copyright International Business Machines Corporation 2000, 2001. All rights reserved.

Translation: © Copyright IBM Japan 2001

目次

図	v	第6章 J2EE アプリケーションのアセンブル	39
表	vii	アプリケーション組み立てツールをインストールするためのステップ	40
本書について	ix	新規 J2EE アプリケーションをアセンブルするためのステップ	41
本書の対象者	ix		
関連情報の検索場所	ix		
<hr/>			
第1部 WebSphere for z/OS J2EE		第7章 J2EE サーバーのランタイム環境の作成	49
サーバーの紹介	1	OS/390 の手動作業を完了するためのステップ	50
		制御およびサーバー領域用の JCL プロシージャを作成するためのステップ	53
第1章 WebSphere for z/OS J2EE サーバーの概要	3	JVM のプロパティを設定するためのステップ	53
		Web アプリケーションに対する J2EE サーバーのサポートの使用可能化	54
第2章 アプリケーション・ツールの概要	9	サーバー構成の定義	55
		管理アプリケーションを始動するためのステップ	56
第3章 J2EE サーバーの定義と活動化の概要	13	会話を開始するためのステップ	56
		J2SERV サーバーを追加するためのステップ	57
<hr/>			
第2部 J2EE サーバー・アプリケーションの作成、アセンブル、および配置	19	J2SERV1 サーバー・インスタンスを追加するためのステップ	59
		J2EE リソースを追加するためのステップ	59
第4章 アプリケーション開発環境の設定	21	J2EE リソース・インスタンスを追加するためのステップ	60
ワークステーションをセットアップするためのステップ	21	J2EE アプリケーションをインストールするためのステップ	61
z/OS または OS/390 をセットアップするためのステップ	26	新規会話モデルの妥当性検査のためのステップ	64
		会話をコミットするためのステップ	64
第5章 J2EE サーバーにインストールされる新規アプリケーション・コンポーネントの作成	29	z/OS または OS/390 タスクの完了のマーク付けをするためのステップ	65
Enterprise bean の作成	30	サーバー構成を活動化するためのステップ	65
Enterprise bean の開発のためのチェックリスト	30		
Enterprise bean の開発	32		
JAR ファイルへの bean のパッケージ化	34		
Web アプリケーションの作成	34	第8章 J2EE アプリケーション・クライアントの作成と実行	67
Web コンポーネントの開発	35	z/OS または OS/390 クライアント用の WebSphere Application Server スタンダード版	67
WAR ファイルへの Web コンポーネントのパッケージ化	36	ネイティブ z/OS または OS/390 Java クライアント	71

非 z/OS および非 OS/390 プラットフォーム上の WebSphere Application Server アドバンスド版およびスタンダード版クライアント . . .	72
Enterprise bean に非 z/OS または非 OS/390 プラットフォームからアクセスするためのステップ	73
J2EE アプリケーションのセキュリティ上の考慮事項	74

第3部 ランタイム環境での J2EE アプリケーションに関する作業 77

第9章 WebSphere for z/OS サーバーへのアプリケーションのインストール	79
管理アプリケーションを介してエクスポート / インポート・プロセスを使用するためのステップ	79
スクリプトによるアプリケーションのインストール	81

第10章 J2EE アプリケーションの活動に関するデータの収集	83
SMF レコードによる J2EE アプリケーション情報の収集	83
Java アプリケーションのロギング・メッセージおよびトレース・データ	83
アプリケーション・メッセージを MVS マスター・コンソールに発行するときのバックグラウンド	86
ユーザーのアプリケーションに対してトレース要求を発行するときのバックグラウンド	89
Java アプリケーションをコード化してメッセージとトレース要求を発行するためのステップ	91
Java アプリケーションのメッセージとトレース要求をログに記録するための z/OS または OS/390 環境を作成するステップ	98
メッセージとトレース・データを表示するときのバックグラウンド	102

第4部 J2EE サーバーへのアプリケーションのマイグレーション 107

第11章 マイグレーションのバックグラウンド	109
----------------------------------	-----

第12章 WebSphere for z/OS の新規リリースへのアプリケーションのマイグレーション 113	113
z/OS または OS/390 スタンダード版の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ	114
OS/390 エンタープライズ版 V3.0.2 の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ	114

第13章 WebSphere for z/OS プラットフォームへのアプリケーションのマイグレーション	119
---	-----

第14章 WebSphere for z/OS J2EE サーバーにすでにインストールされているアプリケーションのアップグレード	123
--	-----

第5部 付録 125

付録A. 環境ファイルおよび JVM プロパティ・ファイル	127
環境ファイルおよび環境変数	127
WebSphere for z/OS による環境変数と環境ファイルの管理方法	127
ランタイム・サーバー開始プロシージャによる環境ファイルの指示方法	129
OS/390 クライアント向け環境変数	129
置換変数使用上の注意事項	130
環境変数構文	130
環境変数の使用	131
環境変数の説明	136
JVM プロパティおよびプロパティ・ファイル	155
JVM プロパティの管理方法	156
JVM プロパティの使用	156
プロパティの説明	157

付録B. 特記事項	159
本書で使用している例について	161
プログラミング・インターフェース情報	161
商標	161

用語集	163
---------------	-----

索引	165
--------------	-----



1.	J2EE アプリケーション用の新規 WebSphere for z/OS サーバー	3	5.	サポートされる J2EE アプリケーシ ョン・コンポーネント	11
2.	J2EE サーバーにインストールされるアプ リケーション・コンポーネントの潜在的な クライアント	6	6.	J2EE サーバーのモデル	14
3.	スタンダード版 for z/OS and OS/390 と J2EE サーバーの可能な構成	7	7.	J2EE サーバーへの J2EE アプリケーシ ョンのインストール	16
4.	J2EE サーバーでのコンポーネントの開 発、アセンブル、およびインストールを 行うためのツールと出力	10	8.	アクティブ J2EE アプリケーション・サ ーバー	17

表

1. J2EE テクノロジーに対する現在の WebSphere for z/OS のサポート	4
2. Java 2 Enterprise Edition アプリケーション・コンポーネントのソフトウェア要件	22
3. アプリケーション開発ソフトウェアのインストールまたはマイグレーション情報を得るための参照先	23
4. アセンブリーおよび展開ソフトウェアのインストールまたはマイグレーション情報を得るための参照先	24
5. Enterprise bean の開発のためのチェックリスト	31
6. トレース設定プロパティ・タイプとこれに対応する JRas トレース・タイプ	100
7. アプリケーションのマイグレーション・パス一覧	107
8. 役割および実行する可能性のあるマイグレーション・アクションのチェックリスト	109
9. WebSphere Application Server for OS/390 エンタープライズ版 V3.02 アプリケーションのマイグレーション作業の要約	115
10. 他の WebSphere ファミリー環境でのアプリケーションのマイグレーション・シナリオの要約	120
11. 環境変数を使用する場所	132
12. JVM プロパティが使用される場所	157

本書について

WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル, SA88-8654 では、WebSphere Application Server V4.0 for z/OS and OS/390 環境で稼働する Java 2 エンタープライズ版 (J2EE) アプリケーションの作成、アセンブル、およびインストールを行う方法について説明します。この J2EE アプリケーションは、場合により、Enterprise Java bean、Java サブレット、および JavaServer Pages (JSP) から構成されます。WebSphere for z/OS J2EE サーバーでは、このアプリケーションの高度な管理を行うことができ、z/OS または OS/390 上のデータベースおよびトランザクション・システムとこのアプリケーションを統合することができるアプリケーション環境が提供されます。

注: 正式なプロダクト名は「WebSphere Application Server V4.0 for z/OS and OS/390」ですが、本書では「WebSphere for z/OS」と呼んでいます。

本書の対象者

本書は、アプリケーション・コンポーネント提供者、アプリケーション・アセンブル担当者、および配置担当者の役割について Sun Microsystems Java 2 エンタープライズ版 Specification V1.2 に定義されている作業を遂行するプログラマーを主な対象として作成されています。これらの役割および関連する責任の詳細については、Sun Microsystems J2EE の仕様を参照してください。これは、下記から入手できます。

<http://java.sun.com/>

関連情報の検索場所

WebSphere for z/OS ライブラリーの中にある資料のリストを以下に示します。これらの資料は、以下の Web サイトで見つけることができます。

<http://www.ibm.com/jp/software/websphere/appserv/>

- *WebSphere Application Server V4.0 for z/OS and OS/390: プログラム・ディレクトリ*, GA22-7833 では、WebSphere for z/OS のエレメントとインストールについて説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: License Information*, LA22-7855 では、WebSphere for z/OS のライセンス情報について説明しています。

- *WebSphere Application Server V4.0 for z/OS and OS/390*: インストールおよびカスタマイズ, GA88-8652 では、WebSphere for z/OS の計画、インストール、およびカスタマイズの作業とガイドラインについて説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390*: メッセージおよび診断, GA88-8655 では、診断情報を提供し、WebSphere for z/OS に関連したメッセージとコードについて説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390*: 操作および管理, SA88-8653 では、システム操作および管理作業について説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 では、WebSphere for z/OS J2EE サーバーで J2EE アプリケーションの開発、アセンブル、およびインストールを行う方法について説明しています。また、この資料には、WebSphere Application Server for OS/390 の以前のリリースから、またはその他の WebSphere ファミリー・プラットフォームからアプリケーションをマイグレーションする方法についての情報も記載されています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: CORBA アプリケーションのアセンブル*, SA88-8658 では、WebSphere for z/OS (MOFW) サーバーで CORBA アプリケーションの開発、アセンブル、および配置を行う方法について説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390*: システム管理ユーザー・インターフェース, SA88-8656 では、システム管理ユーザー・インターフェースで提供されるシステム管理および運用作業について説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390*: システム管理スクリプト API, SA88-8657 では、WebSphere for z/OS システム管理スクリプト API 製品の機能について説明しています。

場合によっては、z/OS または OS/390 のその他のエレメントと製品に関する情報を参照することも必要です。その情報のすべてを、インターネット上の次のロケーションのリンクを介して利用できます。

<http://www.ibm.com/servers/eserver/zseries/zos/>
<http://www.ibm.com/servers/s390/os390/>

特に役立つ資料を以下に示します。

- *Getting Started with WebSphere Application Server*, SC09-4581 では、WebSphere for z/OS について概説し、環境を設定するための要件について説明しています。
- *WebSphere* ビジネス構築のソリューション, SD88-7362

第1部 WebSphere for z/OS J2EE サーバーの紹介

第1章 WebSphere for z/OS J2EE サーバーの概要

WebSphere Application Server for z/OS and OS/390 では、可用性、安全性、信頼性、および拡張性の高いランタイム環境が Java 2 エンタープライズ版 (J2EE) アプリケーション用に提供されます。この WebSphere for z/OS ランタイムには、J2EE アプリケーションと CORBA アプリケーションの両方に対するサーバーが、それぞれその J2EE サーバーとマネージメント・オブジェクト・フレームワーク (MOFW) サーバーを介して組み込まれます。WebSphere for z/OS 環境について、図1 で説明します。

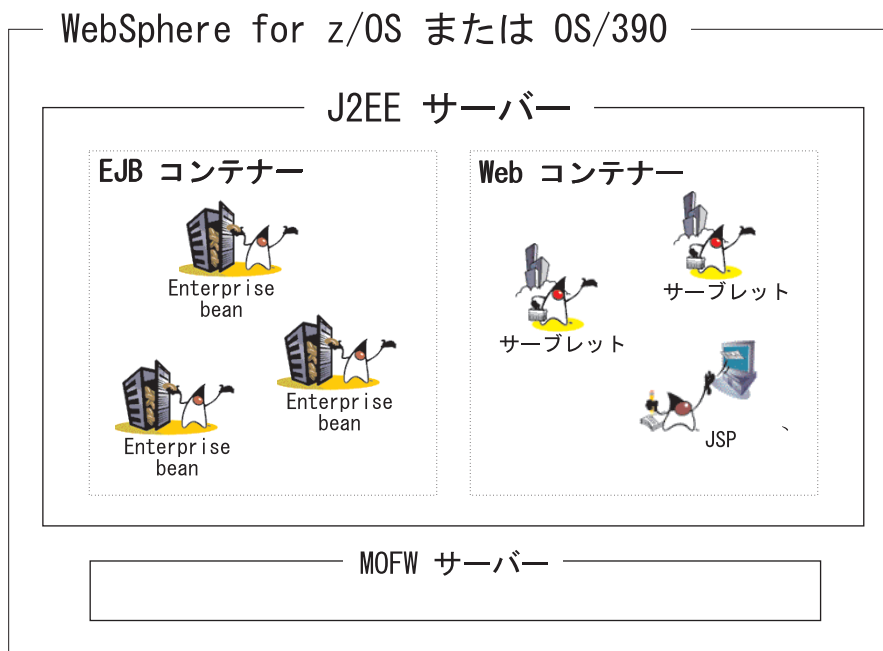


図1. J2EE アプリケーション用の新規 WebSphere for z/OS サーバー

WebSphere for z/OS は、その J2EE サーバーに重点を置いています。このサーバーは、Sun Microsystems が公表した J2EE 仕様およびパッケージ化規格に準拠する Enterprise bean と Web アプリケーションの両方をサポートします。WebSphere for z/OS J2EE サーバーで稼働するこれらの J2EE アプリケーション・コンポーネントは、以下の両方を使用できます。

- Java 2 スタンダード版 (J2SE) Software Development Kit (SDK) V1.3 によって提供されるアプリケーション・プログラミング・インターフェース (API) およびサービス
- Java Database Connectivity (JDBC)、Java Naming and Directory Interface (JNDI)、Java Transaction Service (JTS) および API などのエンタープライズ・サービス

現在の WebSphere for z/OS のサポートには、表1 にリストされた J2EE テクノロジーが含まれています。

表 1. J2EE テクノロジーに対する現在の WebSphere for z/OS のサポート

J2EE テクノロジー	WebSphere for z/OS J2EE サーバーでのサポート
Java 2 スタンダード版 (J2SE) Software Development Kit (SDK)	V1.3
Enterprise JavaBeans	V1.1 (ただし、V1.0 仕様もサポートします。)
Java サンプル	V2.2 仕様
JavaServer Pages (JSP)	V1.1 仕様
Java Transaction Service (JTS) および API (JTA)	分散トランザクションでサポートされるV1.0
Java Database Connectivity (JDBC)	JDBC V2.1 および JDBC スタンダード・エクステンション V2.0 (JDBC V1.x は、互換目的でサポートされています。)
Java Naming and Directory Interface (JNDI)	1.2
Java リモート・メソッド呼び出し (RMI) および RMI/IIOP	1.0
Java IDL	サポートされています。
Java メッセージ・サービス (JMS)	現在はサポートされていません。
JavaBeans Activation Framework (JAF)	現在はサポートされていません。
JavaMail	現在はサポートされていません。
コネクタ	Web コンポーネント専用: IBM Common Connector Framework (CCF) コネクタ: <ul style="list-style-type: none"> • IMS Connect 製品 (5655-E51) • CICS Transaction Gateway 製品 (5648-B43)

WebSphere for z/OS J2EE サーバーは、多様なクライアント・アプリケーションをサポートしています。そのクライアント・アプリケーションは、6ページの

図2 に示された Enterprise bean または Web アプリケーションにアクセスできません。以下のことに注目してください。

- WebSphere for z/OS J2EE サーバーのコンポーネントは、場合により、別の J2EE サーバー内、同じかまたは別の z/OS または OS/390 イメージ上のコンポーネントのクライアントです。
 - サブレットは、MOFW サーバー内の CORBA ベースの Java ビジネス・オブジェクト (BO) を駆動できます。
 - Enterprise bean と Java BO は相互協調処理が可能です。
- しかし、MOFW サーバー内の C++ ビジネス・オブジェクトは、J2EE サーバー 内のアプリケーション・コンポーネントにアクセスできません。
- WebSphere Application Server スタンダード版 で稼働する Web コンポーネント (サブレットと JavaServer Pages) として、次のものがあります。
 - WebSphere for z/OS J2EE サーバー内の Enterprise bean を駆動するクライアント
 - WebSphere for z/OS J2EE サーバーで稼働するようマイグレーションすることができる Web コンポーネント

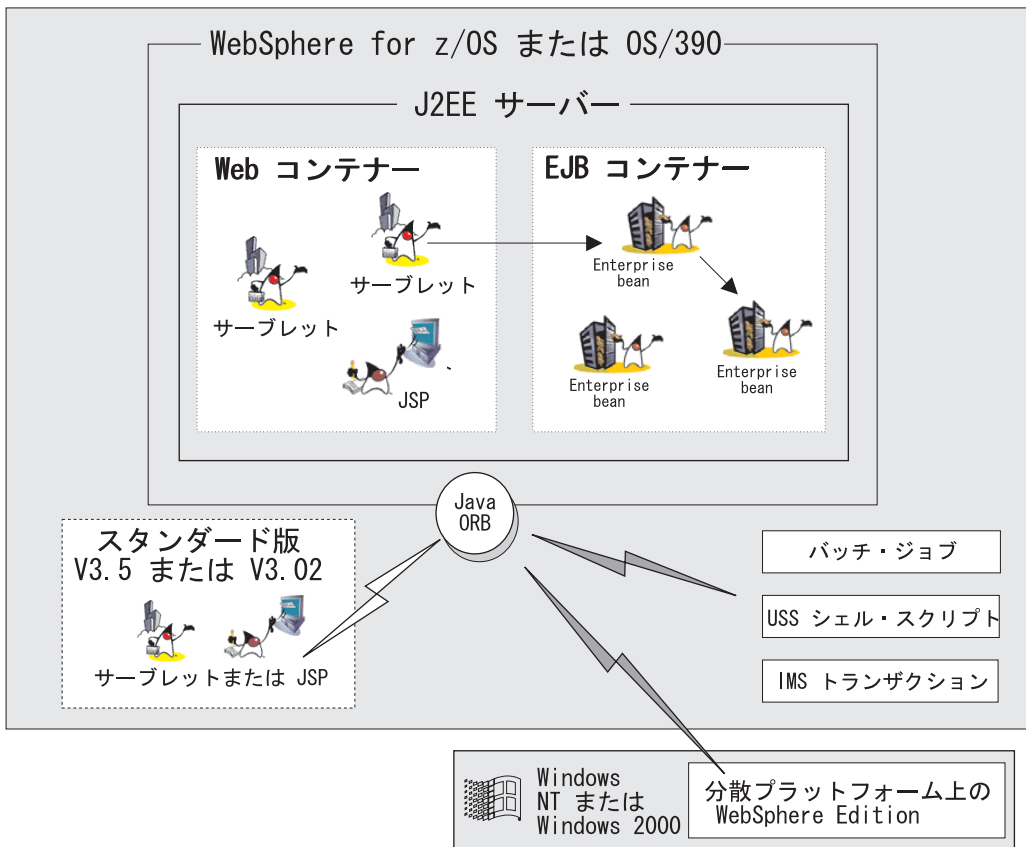


図2. J2EE サーバーにインストールされるアプリケーション・コンポーネントの潜在的なクライアント

7ページの図3 では、z/OS または OS/390 上で Web サービス提供を可能にすることができる 2 とおりの可能な構成を説明しています。HTTP サーバーから次のいずれかのアプリケーションへ要求を送信します。

1. WebSphere Application Server スタンダード版 バージョン 3.5 または 3.02 の Web アプリケーション。
2. WebSphere for z/OS J2EE サーバーの Web コンテナ内の Web アプリケーション。このオプションでは、WebSphere Application Server プラグインを使用して WebSphere for z/OS に要求を送信します。

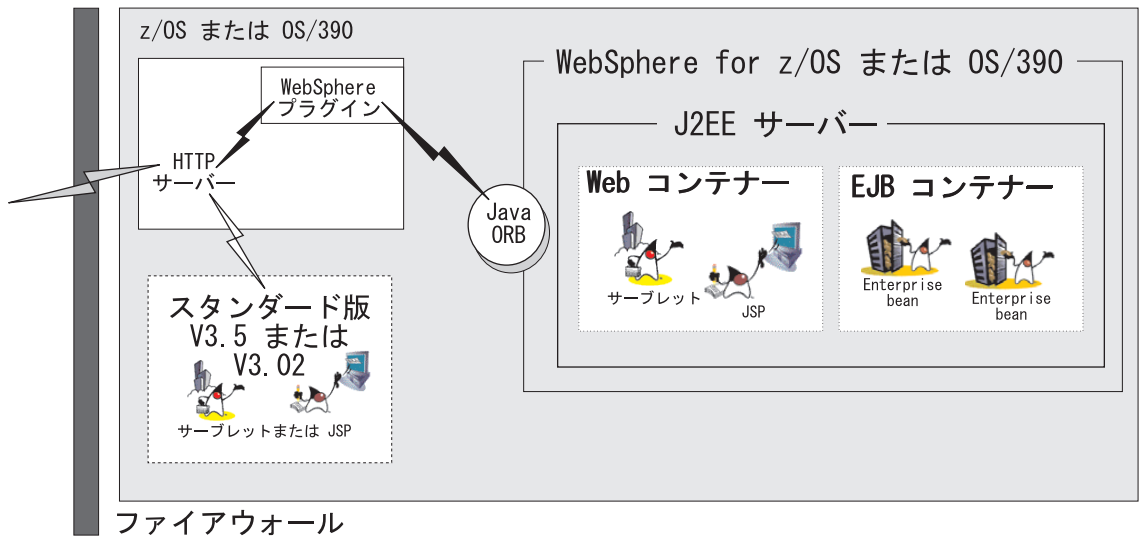


図3. スタンダード版 for z/OS and OS/390 と J2EE サーバーの可能な構成

Enterprise bean と Web アプリケーションは、J2EE パッケージ化規格に準拠しなければならないので、WebSphere for z/OS では、WebSphere for z/OS J2EE サーバーにインストールするための J2EE アプリケーションの準備を行うのに役立つ新規ツールを導入しています。以下のトピックでいくつかの処理を行う方法について、簡単に検討します。

- WebSphere for z/OS J2EE サーバーの J2EE アプリケーションの作成、アセンブル、およびインストール (または配置) を行う方法。アセンブリー段階で新規ツール、WebSphere for z/OS アプリケーション組み立てツールを使用する必要があります。9ページの『第2章 アプリケーション・ツールの概要』を参照してください。
- WebSphere for z/OS J2EE サーバーを定義してアクティブにする方法。このプロセスでは、WebSphere for z/OS 管理アプリケーションを使用する必要があります。管理アプリケーションは、システム管理エンドユーザー・インターフェース (SM EUI) とも呼ばれます。

WebSphere Application Server エンタープライズ版 for OS/390 製品をこれまでに使用したことがあれば、管理アプリケーションを使用するときいくつかの違いに気が付きます。13ページの『第3章 J2EE サーバーの定義と活動化の概要』を参照してください。

第2章 アプリケーション・ツールの概要

この章のプロセスの説明を理解するために、J2EE 用語を復習することが必要な場合があります。Java 2 エンタープライズ版 (J2EE) アプリケーションは J2EE モジュールから構成され、また、このモジュールは、以下の J2EE コンポーネントから構成されます。

- Enterprise bean
- Web コンポーネント。つまり、サーブレットまたは JavaServer Pages (JSP)
- アプリケーション・クライアント
- アプレット

J2EE モジュールには、場合により、同じタイプのコンポーネントが 1 つ以上含まれます。J2EE モジュールはアーカイブ、つまり、Java アーカイブ (JAR) ファイルまたは Web アプリケーション・アーカイブ (WAR) ファイルです。J2EE モジュールの集合である J2EE アプリケーションは、Enterprise アーカイブ (EAR) ファイルにパッケージされます。J2EE アプリケーションを WebSphere for z/OS にインストールできるのは、このアプリケーションが EAR ファイルにパッケージされるときだけです。

WebSphere for z/OS がその J2EE サーバーで現在サポートしている J2EE アプリケーション・コンポーネントは、Enterprise bean と Web コンポーネントです。J2EE サーバーは、その EJB コンテナを介して Enterprise bean をサポートし、その Web コンテナを介してサーブレットと JSP をサポートしません。

J2EE アプリケーションについてサポートされるコンポーネントを作成するには、コンポーネント (Enterprise bean、サーブレット、または JSP) のタイプごとの Sun Microsystems の仕様、および WebSphere for z/OS がサポートする仕様レベルに精通していることが必要です。その知識があれば、10ページの図4に示すように、開発および配置プロセスを開始できます。

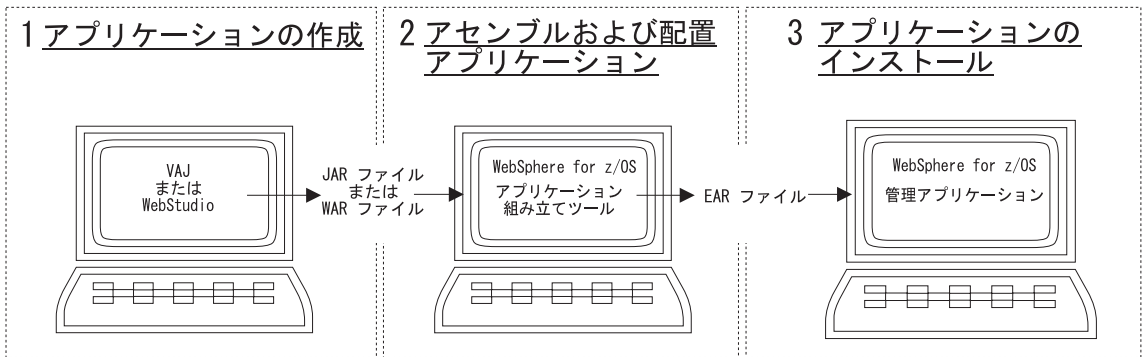


図4. J2EE サーバーでのコンポーネントの開発、アセンブル、およびインストールを行うためのツールと出力

1. ビジネスの目的に応じて、アプリケーション・コンポーネントと関連クラス、または各コンポーネントで必要とされるファイルを定義してインプリメントします。Enterprise bean を開発する場合は、IBM の VisualAge for Java のようなツールを使用し、サーブレットと JSP については、IBM の WebSphere Studio のようなツールを使用してください。

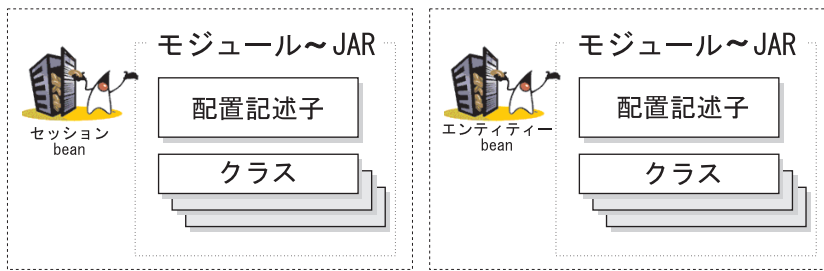
コンポーネント開発プロセスの一環として、使用するツールで配置記述子を作成します。この記述子には、J2EE サーバーによる各アプリケーション・コンポーネントのライフ・サイクルとリソースの管理のしかたを定義するために選択する属性と環境設定値が含まれます。その定義は、プラットフォームに依存しない指定であるため、ワークステーション開発環境でその定義をテストできます。

開発プロセスのこの段階を完了した時点で、11ページの図5 に示すように、1 つ以上の次の ART ファイルが存在しています。

- Java アーカイブ (JAR) ファイルにパッケージされた Enterprise bean、そのクラスと配置記述子
- Web アプリケーション・アーカイブ (WAR) ファイルにパッケージされたサーブレット、そのクラスと記述子、JSP、および HTML や GIF などの静的ファイルで構成された Web アプリケーション

各 JAR ファイルには、場合により 1 つ以上の Enterprise bean が含まれることに注目してください。同様に、各 WAR ファイルには、場合により複数のサーブレットまたは JSP が含まれます。これらの JAR ファイルと WAR ファイルは、次の段階、つまり、アセンブリーと配置のための入力データとなります。

Enterprise bean:



Web コンポーネント:

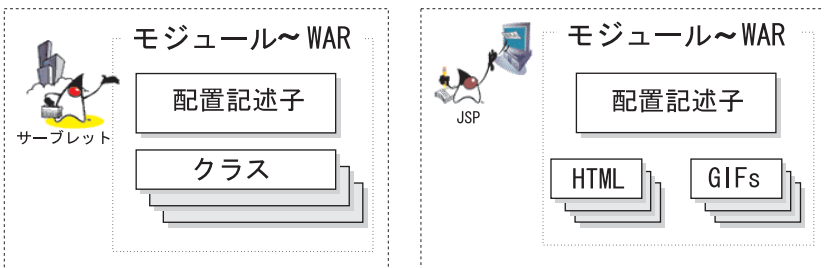


図5. サポートされる J2EE アプリケーション・コンポーネント

2. J2EE アプリケーション・コンポーネントをアセンブルして配置します。

アプリケーションのアセンブリーと配置の間に、そのモジュールを Enterprise アーカイブ (EAR) ファイルにアセンブルする必要があります。この EAR ファイルのタイプだけが、WebSphere for z/OS で J2EE サーバーへのインストールを受け入れられます。

WebSphere for z/OS にインストールするためにアプリケーションをアセンブルするには、WebSphere for z/OS アプリケーション組み立てツールを使用する必要があります。このツールで以下のことを行います。

- リモート・インターフェース、ホーム・インターフェース、タイ、スタブ、キー、ハンドル、ファインダー・ヘルパー、およびパーシスタンスに関連したコードを含めて、z/OS または OS/390 用のコードを生成します。
 - V1.1 仕様レベルに一致するように、V1.0 Enterprise bean の配置記述子を変換します。この機能によって、WebSphere for z/OS が V1.0 bean をサポートできるようになります。
3. J2EE アプリケーションをインストールします。

WebSphere for z/OS 上でアプリケーションをインストールするには、WebSphere for z/OS 管理アプリケーションを使用する必要があります。イ

インストール・プロセスを通じて、配置記述子は WebSphere for z/OS 環境用にカスタマイズされ、アプリケーション・コンポーネントは WebSphere for z/OS J2EE サーバーにロードされます。具体的には、以下のとおりです。

- アプリケーション ART ファイル (JAR ファイル、EAR ファイルなど) は、ワークステーションから z/OS または OS/390 に転送されます。
- WebSphere for z/OS が J2EE アプリケーション・コンポーネントにアクセスして管理できるように、アプリケーション・メタデータは保管されません。
- bean とサーブレットのリソースと参照は、Java Naming and Directory Interface (JNDI) を使用して解決されます。

注: WebSphere for z/OS ではクライアント・コンテナ・サポートが提供されないため、サーバー側の `java:comp` ルックアップだけが可能です。つまり、J2EE サーバーで実行中の Enterprise bean、サーブレット、および JSP だけが、これらのコンポーネントの作成に使用される開発ツールも `java:comp` 名に対する JNDI ルックアップをサポートする限り、`java:comp` ルックアップを使用できます。クライアントとアプレットは、アプリケーション・コンポーネントおよびホームに対して明示的な名前を使用する必要があります。

- J2EE サーバーのコンテナ・パラメーターは、インストールされるアプリケーション、モジュール、コンポーネント、およびメソッドに対する配置記述子によって構成されます。
- クライアントが要求したときに HTTP プロトコルを介して WAR ファイルと EJB JAR ファイルにアクセスできるようにする完全修飾 URI で、Web アプリケーションが提供されます。

第3章 J2EE サーバーの定義と活動化の概要

実行可能 J2EE アプリケーションを開発してアSEMBルしたあと、このアプリケーションを適切なランタイム環境にインストールする必要があります。z/OS または OS/390 プラットフォームの場合、ランタイム環境は、WebSphere for z/OS アプリケーション・サーバー (J2EE サーバー) です。ご使用のシステムの方針によっては、アプリケーションのテストに使用するために、J2EE サーバーがすでに使用可能になっている場合もあります。そうでない場合は、アプリケーション・サーバーを新たに作成する必要があります。そのためには、手動で完了する作業と WebSphere for z/OS 管理アプリケーションを介して完了する作業を合わせて行う必要があります。管理アプリケーションは、システム管理エンドユーザー・インターフェース (SM EU) とも呼ばれます。

その手動作業では、z/OS または OS/390 オペレーティング・システムに関する専門知識が多少要求されるので、それぞれの作業は、通常、システム・プログラマー、データベース管理者、またはセキュリティー管理者が行います。z/OS または OS/390 に精通していなくてもその作業を完了するための役に立つサンプルを提供してありますが、必要に応じて、ご使用のシステムの専門家に相談してください。

管理アプリケーションを介して実行する作業では、必ずしも z/OS または OS/390 に精通していることが必要ではありません。しかし、最初にモデルのランタイム環境をユーザーのアプリケーション用に定義し、そのモデルをアクティブにして作業用 z/OS または OS/390 アプリケーションに入れてプロセスを完了することを理解すると、作業が比較的楽に実行できる可能性があります。この章では、このサーバー・モデルを漸次に作成し、z/OS または OS/390 オペレーティング・システムの従来の長所を利用するランタイム環境にこのサーバー・モデルを変える方法を概説します。その作業を実行するためには、正しく構成されてアクティブになっている WebSphere for z/OS インストール・システムとの接続を管理アプリケーションが確立していることを確認する必要があります。

1. J2EE サーバー、J2EE サーバー・インスタンス、および J2EE リソースを定義します。

管理アプリケーションを使用して J2EE サーバーを定義するとき、14ページの図6 に示されたエレメントを含むモデルを作成します。

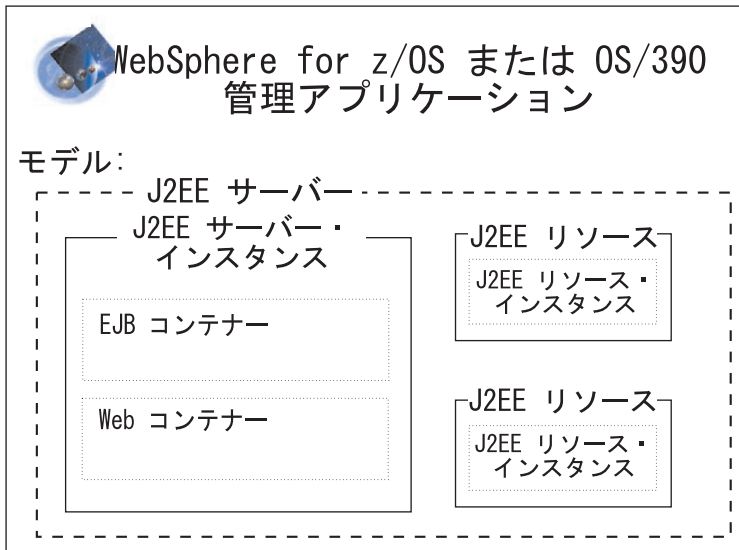


図 6. J2EE サーバーのモデル

このモデルには、テスト用または実動用の z/OS または OS/390 構成に J2EE サーバーがどのように適合するかを定義する、システムやシスプレックタスなどの追加エレメントが組み込まれます。しかし、ユーザーのアプリケーションのランタイム環境を定義するという目的上、以下のモデル・エレメントに専心することができます。

- アプリケーション環境に対応する汎用サーバー。サーバーとは、z/OS または OS/390 上で実行されるある種の作業を担当するエンティティーです。
 - ユーザーのアプリケーションが稼働するサーバー・インスタンス。サーバー・インスタンスとは、ユーザーのアプリケーション・コンポーネントが稼働する Java 仮想マシン (JVM) に特に対応するエンティティーです。サーバー・インスタンスは、Enterprise bean 用の EJB コンテナまたはサーブレットと JSP 用の Web コンテナを介したアプリケーション・コンポーネントの実行と管理を担当します。
 - J2EE リソースと J2EE リソース・インスタンス。これらは、汎用タイプのシステム・リソースと、そのタイプを管理する特定のサブシステムにそれぞれ対応します。たとえば、DB2 は z/OS または OS/390 システム・リソースのタイプであり、特定の DB2 サブシステムがそのシステム上のすべての DB2 データベースおよび表を管理する場合があります。
2. ユーザーの J2EE アプリケーションを含んでいる Enterprise アーカイブ (EAR) ファイルをインストールします。

ユーザーの J2EE アプリケーションも、サーバー・モデルの一部になります。インストール・プロセスの一部として、以下の処理が行われます。

- アプリケーション ART ファイル (JAR ファイル、EAR ファイルなど) は、ワークステーションから z/OS または OS/390 に転送されます。
- WebSphere for z/OS が J2EE アプリケーション・コンポーネントにアクセスして管理できるように、アプリケーション・メタデータは保管されます。
- bean とサーブレットのリソースと参照は、Java Naming and Directory Interface (JNDI) を使用して解決されます。

注: WebSphere for z/OS ではクライアント・コンテナ・サポートが提供されないため、サーバー側の `java:comp` ルックアップだけが可能です。つまり、J2EE サーバーで実行中の Enterprise bean、サーブレット、および JSP だけが、これらのコンポーネントの作成に使用される開発ツールも `java:comp` 名に対する JNDI ルックアップをサポートする限り、`java:comp` ルックアップを使用できます。クライアントとアプレットは、アプリケーション・コンポーネントおよびホームに対して明示的な名前を使用する必要があります。

- J2EE サーバーのコンテナ・パラメーターは、インストールされるアプリケーション、モジュール、コンポーネント、およびメソッドに対する配置記述子によって構成されます。
- クライアントが要求したときに HTTP プロトコルを介して WAR ファイルと EJB JAR ファイルにアクセスできるようにする完全修飾 URI で、Web アプリケーションが提供されます。

サーバーの定義およびアプリケーションのインストール処理の結果を、16ページの図7 で説明しています。これは、次のものが存在するアプリケーション・サーバー・インスタンスのモデルです。

- EJB コンテナ (Enterprise bean 用)、および多くの場合、Web コンテナ (サーブレットと JSP 用)
- J2EE リソースとの接続。これは、WebSphere for z/OS がアプリケーションのインストールの一環として作成します。

注: 現在、サーブレットまたは JSP を含む J2EE アプリケーションをインストールしようとする場合は、追加作業をいくつか完了して J2EE サーバー用の Web コンテナを構成する必要があります。その場合、管理アプリケーションを使用してアプリケーションをインストールできます。詳細は、本書の J2EE サーバーの作成方法に関する段階的説明の中に後述されています。



WebSphere for z/OS または OS/390 管理アプリケーション

モデル:

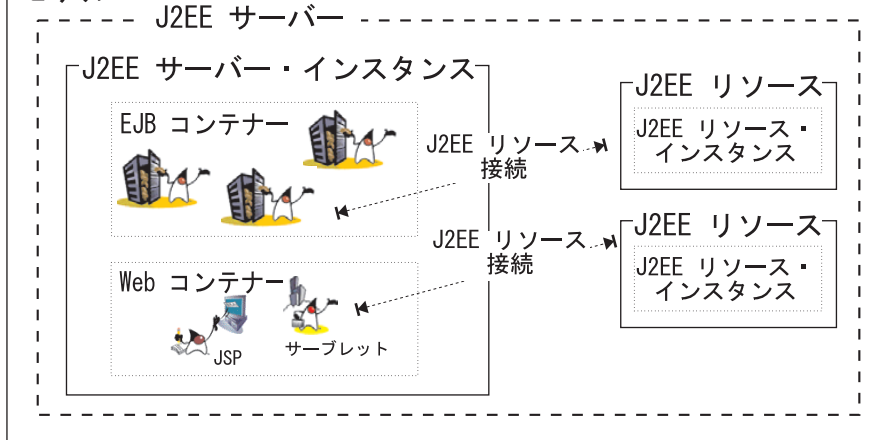


図7. J2EE サーバーへの J2EE アプリケーションのインストール

3. 会話モデルをアクティブな J2EE サーバー・ランタイム環境に変換します。

サンプル・アプリケーション用のランタイム環境のモデルが得られたなら、このプロセスの最終段階、つまり、z/OS または OS/390 上のアクティブなランタイム環境へのモデルの変換を開始します。

まず、管理アプリケーションを使用してサーバー・モデルをコミットします。これは、定義を永久に保管することに似ています。

その場合、管理アプリケーションを使用してサーバーをアクティブにします。17ページの図8 では、どのようにモデル・エレメントが z/OS または OS/390 システム内のアクティブ・エレメントに対応するかを説明しています。



ランタイム環境

モデル:

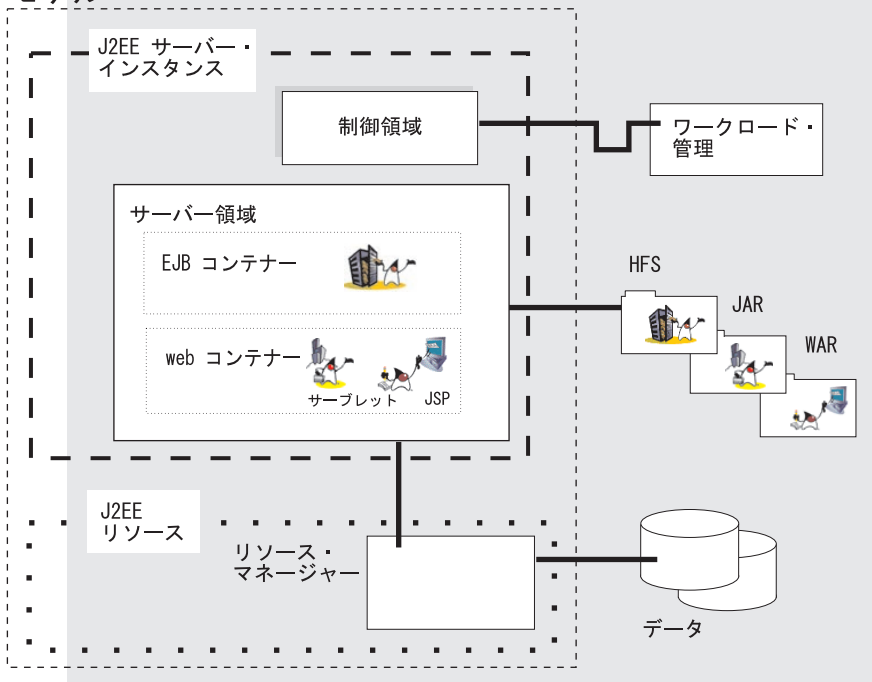


図8. アクティブ J2EE アプリケーション・サーバー

以下のエレメントに特に注目してください。

- サーバー・インスタンスは、次のものから構成されています。
 - z/OS または OS/390 ワークロード管理 (WLM) に対するクライアント要求を受信してキューに入れる制御領域
 - 1 つ以上のサーバー領域 (z/OS または OS/390 アドレス・スペース)。サーバー領域は、アプリケーションのコードを一緒に実行して管理するいくつかの機能から構成されています。WLM は、着信要求の量に応じて追加サーバー領域を始動します。
- サーバー領域コンテナは、このサーバーにインストールされたアプリケーション・コンポーネントのライフ・サイクルを管理します。各サーバー領域は、HFS にインストールされた実行可能アプリケーション・コードを検出できます。
- J2EE リソースは、あるタイプのリソースを管理する 1 つのサブシステムに相当します。各サーバー・インスタンスは、1 つ以上の J2EE リソース・タイプおよびサブシステムに接続される場合があります。

サーバー・モデルをアクティブにすると、WebSphere for z/OS は、クライアント要求を管理する WLM に制御領域を接続し、データを管理する DB2 などの実際のサブシステムにサーバー領域を接続します。クライアント要求が着信すると、直ちに、J2EE サーバーは、その要求が Enterprise bean か Web コンポーネントのいずれに対するものであるかを判別し、その要求を EJB コンテナか Web コンテナのいずれかに送信します。次に、コンテナは、参照されるアプリケーション・コンポーネントのコードを検出し、ライフ・サイクルが開始されます。

第2部 J2EE サーバー・アプリケーションの作成、アセンブル、および配置

第4章 アプリケーション開発環境の設定

J2EE アプリケーション・コンポーネントの開発、アSEMBル、およびインストールを行うために使用するツールはすべて、ワークステーション・ツールです。このワークステーション・ツールのインストールと使用に関する資料はすでに利用可能になっているので、以下の手順では、インストール・ステップの要約だけを示していますが、より詳細な説明がある情報源について言及しています。J2EE アプリケーション・コンポーネントの開発を開始するとき、正しいツールと情報源を用意してあることを確認するために、この手順をチェックリストとして利用してください。

ワークステーションをセットアップするためのステップ

Java 2 エンタープライズ版 (J2EE) アプリケーションの開発、アSEMBル、およびインストールを行うためにご使用のワークステーションをセットアップするには、以下のステップを実行してください。

1. アプリケーションの開発のためにどのアプリケーション開発ツールおよびその他のソフトウェアが必要になるかを判断します。必要なソフトウェアを判断するために、次の表を役立ててください。

EJB コンポーネントの開発に関する推奨: bean、サーブレット、および JSP の開発とテストを行うために VisualAge for Java の IBM EJB Development Environment フィーチャーを使用する場合、Java 2 Standard Edition (J2SE) ソフトウェア開発キット (SDK) または WebSphere Application Server ランタイムをインストールする必要はありません。ワークステーション上で bean 配置環境をシミュレートするために使用できるコード生成されたテスト・クライアントおよび WebSphere Application Server ランタイムが、VisualAge for Java のこのフィーチャーで提供されます。IBM EJB Development Environment を使用すると、開発者は、リモート・メソッドの呼び出し、およびホーム・インターフェース上のメソッドの呼び出しを含めて、エンティティーおよびセッション bean を十分にテストすることができます。また、この機能にはサーブレット・エンジンもあるので、サーブレットと JSP が HTTP およびアプリケーション・サーバーを経由しているかのように、Web ブラウザーにサーブレットと JSP を提供できます。

Web コンポーネントの開発に関する推奨: サーブレットまたは JSP を開発し、これらを自動的に Web アプリケーション・アーカイブ (WAR) ファイ

ルにパッケージするには、IBM WebSphere Studio を使用してください (その他のツールを使用するときには、WAR ファイルを手動で作成しなければならない場合があります)。次に、WebSphere Application Server for Multiplatforms を使用すると、これらの Web コンポーネントをアセンブルして J2EE サーバーにインストールする前に、これらの Web コンポーネントをテストすることができます。

表 2. Java 2 Enterprise Edition アプリケーション・コンポーネントのソフトウェア要件

J2EE	使用するソフトウェア
アプリケーション・コンポーネント	<p data-bbox="279 526 467 548">Enterprise bean</p> <p data-bbox="518 526 740 548">開発およびテスト用:</p> <ul data-bbox="518 565 1190 869" style="list-style-type: none"> <li data-bbox="518 565 1190 626">• 以下のフィーチャーがあるパッチ 2 を伴った VisualAge for Java 3.5 <ul data-bbox="545 638 1016 869" style="list-style-type: none"> <li data-bbox="545 638 807 661">– Data Access Beans 3.5 <li data-bbox="545 673 989 696">– IBM EJB Development Environment 3.5 <li data-bbox="545 708 975 730">– IBM Enterprise Extension Libraries 3.5 <li data-bbox="545 743 975 765">– IBM WebSphere Test Environment 3.5 <li data-bbox="545 777 995 800">– IBM Common Connector Framework 3.5 <li data-bbox="545 812 1016 835">– IBM Enterprise Access Builder Library 3.5 <li data-bbox="545 847 881 869">– IBM Java Record Library 3.5 <p data-bbox="545 904 1220 1069">ヒント: VisualAge for Java を使用する代わりに、IBM 以外のツール、たとえば、JBuilder や Visual Cafeなどをアプリケーション開発用に使用することもできます。これらの製品の資料を利用して、ハードウェアおよびソフトウェア要件を判別してください。</p> <ul data-bbox="518 1086 1220 1338" style="list-style-type: none"> <li data-bbox="518 1086 1204 1147">• IBM または Sun Microsystems Java 2 スタンダード版 (J2SE) ソフトウェア開発キット (SDK)V1.3 <li data-bbox="518 1164 1220 1225">• WebSphere Application Server アドバンスド版 V3.5。アプリケーション・コンポーネントのテスト用 <li data-bbox="518 1242 1220 1338">• (オプション) DB2 ユニバーサル・データベース・バージョン 7.1。パーシスタント・データ・ストアを使用する必要がある bean をテストする場合にのみ必要 <hr/> <p data-bbox="518 1355 1220 1416">アセンブリー用: WebSphere for z/OS アプリケーション組み立てツール</p> <hr/> <p data-bbox="518 1433 1220 1489">J2EE サーバーでのインストール用: WebSphere for z/OS 管理アプリケーション</p>

表2. Java 2 Enterprise Edition アプリケーション・コンポーネントのソフトウェア要件 (続き)

J2EE アプリケーション・ コンポーネント	使用するソフトウェア
サーブレットおよび JavaServer Pages (JSP)	<p>開発およびテスト用:</p> <ul style="list-style-type: none"> WebSphere Studio 3.5.2 <p>ヒント: WebSphere Studio を始動すると、VisualAge for Java と WebSphere Application Server アドバンスド版の両方がご使用のワークステーションにインストールされているかどうかの検査がこのツールで行われます。</p> <ul style="list-style-type: none"> IBM または Sun Microsystems Java 2 スタンダード版 (J2SE) ソフトウェア開発キット (SDK) V1.3 <p>アセンブリー用: WebSphere for z/OS アプリケーション組み立てツール</p> <p>J2EE サーバーでのインストール用: WebSphere for z/OS 管理アプリケーション</p>

2. 必要に応じて、適切なアプリケーション開発ソフトウェアをワークステーションにインストールするかアップグレードしてください。インストールまたはマイグレーションの説明については、以下を参照してください。

表3. アプリケーション開発ソフトウェアのインストールまたはマイグレーション情報を得るための参照先

ソフトウェア	インストールまたはマイグレーションの情報源
<ul style="list-style-type: none"> VisualAge for Java WebSphere Studio 分散プラットフォーム用の WebSphere Application Server アドバンスド版 	<p>ハードウェア要件とインストールの説明については、ブラウザーを使用して次の Web ページを表示してください。</p> <p>http://www.ibm.com/jp/software/webservers/appserv/library.html</p> <p>この Web ページから次のリンクをクリックしてください。</p> <ul style="list-style-type: none"> ハードウェアとソフトウェアの完全なリスト表示については、WebSphere Application Server V3.5 (分散プラットフォーム用のすべての版) についてサポートされているハードウェア、ソフトウェア、および API 計画とインストールの説明に関する情報およびこの説明へのリンクについては、分散プラットフォーム用の WebSphere Application Server バージョン 3.5 スタンダード および アドバンスド版に対する インフォセンター

表 3. アプリケーション開発ソフトウェアのインストールまたはマイグレーション情報を
得るための参照先 (続き)

ソフトウェア	インストールまたはマイグレーションの情報源
JBuilder や Visual Cafe などの IBM 以外のツ ール	インストールとマイグレーションの説明について、これらの 製品の資料を利用してください。
DB2 ユニバーサル・デー タベース バージョン 7.1	DB2 のセットアップ方法およびアプリケーション・プログ ラムに対する影響についての詳細は、ご使用のワークステー ション・プラットフォーム用の資料 <i>DB2 ユニバーサル・デー タベース... 概説</i> およびインストールをまず参照してくだ さい。

3. 適切なアセンブリーおよび配置ソフトウェアをワークステーションにインス
トールするかアップグレードします。インストールまたはマイグレーション
の説明については、以下を参照してください。

表 4. アセンブリーおよび展開ソフトウェアのインストールまたはマイグレーション情報
を得るための参照先

ソフトウェア	インストールまたはマイグレーションの情報源
WebSphere for z/OS アプ リケーション組み立てツ ール	ワークステーションの要件とインストールの説明につい ては、40ページの『アプリケーション組み立てツールをインス トールするためのステップ』を参照してください。
WebSphere for z/OS 管理 アプリケーション	ワークステーションの要件とインストールの説明につい ては、 <i>WebSphere Application Server V4.0 for z/OS and OS/390:</i> インストールおよびカスタマイズ, GA88-8652 の管理および 運用アプリケーションのインストールに関するトピックを参 照してください。

4. ワークステーションの環境変数が正しく設定されていることを確認します。
検査する変数は、CLASSPATH、JAVA_HOME、LIBPATH、PATH などで
す。インストールした製品によっては、これらの変数が自動的に設定される
場合もあります。詳細については、各製品のインストール資料を参照してく
ださい。
5. パーシスタント・データ・ストアが必要とされる bean の実行とテストのため
に VisualAge for Java の IBM WebSphere Test Environment フィーチャ

ーを使用しようとする場合、必ず、正しい DB2 JDBC ドライバーを指定してください。JDBC V2.1 が DB2 製品のインストールのためのデフォルトではない場合があります。詳細については、DB2 のインストール資料を参照してください。

-
6. **推奨:** VisualAge for Java を使用しようとする場合、VisualAge for Java のインストールまたはアップグレードを終了したあとでワークスペースをバックアップしてください。そのためには、ide.icx ファイルと ide.ini ファイルをバックアップします。

-
7. WebSphere Studio を使用しようとする場合、このツールのインストール後に、必ず、以下のステップを完了してください。

- a. WebSphere Studio を始動し、次のステップを完了します。

- 1) 「ファイル (File)」 → 「新規プロジェクト (New Project)」を選択し、自分のプロジェクトの名前を指定しします。

- 2) VisualAge で「プロジェクト (Project)」 → 「VisualAge for Java」 → 「Studio Tool をインストール (Install Studio Tools)」を選択します。

- 3) 「プロジェクト (Project)」 → 「発行ステージをカスタマイズ (Customize Publishing Stages)」を選択します。

表示されるダイアログ・ボックスで、「ステージ名 (Stage Name)」に VAJ と入力し、「追加 (Add)」をクリックしたあと、「OK」をクリックします。

- 4) 右側のペインで VAJ ステージ・アイコンを右クリックし、次に、「挿入 (Insert)」を選択し、localhost:8080 という名前のサーバーを挿入します。

結果: デフォルト・サーバーのサーバー・アイコンが表示されます。プロジェクト内の発行可能なすべてのリソースは、このサーバーに自動的に追加されます。

- 5) 「発行 (Publishing)」ビューで、http://localhost:8080 サーバーを右クリックし、「プロパティ (Properties)」を選択してから「ターゲットの発行を定義 (Define Publishing Targets)」をクリックします。このあと、WebSphere Test Environment 文書とサブレット・ディレクトリーへのパスを、次のように設定します。

```
html=D:\Program Files\IBM\VisualAge for Java\ide\project_resources\
  IBM WebSphere Test Environment\hosts\default_host\default_app\web
```

```
servlet=D:\Program Files\IBM\VisualAge for Java\ide\project_resources\
  IBM WebSphere Test Environment\hosts\default_host\default_app\servlets
```

- b. VisualAge for Java を始動し、次のステップを完了します。
- 「ウィンドウ」 → 「オプション」 → 「ツール API へのリモート・アクセス」を選択します。
 - VisualAge の始動時のリモート・アクセスをアクティブにするために「ツール API へのリモート・アクセス」チェック・ボックスを選択します。
 - リモート・アクセスが現在実行中でない場合、「ツール API へのリモート・アクセスの開始」をクリックし、リモート・アクセスを開始します。
 - 「ウィンドウ」 → 「オプション」 → 「リソース」を選択します。このあと、ワークスペース・クラス・パスに下記を追加します。

```
D:¥Program Files¥IBM¥VisualAge for Java¥ide¥project_resources¥  
IBM WebSphere Test Environment¥hosts¥default_host¥default_app¥servlets
```

- c. コマンド・プロンプトから `SERunner.properties` ファイルを検索します。このプロパティ・ファイルを開き、`docRoot` パスと `serverRoot` パスが正しいことを確認します。

```
docRoot=D:¥¥Program Files¥¥IBM¥¥VisualAge for Java¥¥ide¥¥project_resources¥¥  
IBM WebSphere Test Environment¥¥hosts¥¥default_host¥¥default_app¥¥web
```

```
serverRoot=D:¥¥Program Files¥¥IBM¥¥VisualAge for Java¥¥ide¥¥project_resources¥¥  
IBM WebSphere Test Environment
```

-
8. パーシスタント・データ・ストアが必要とされる bean を開発しようとする場合、その bean で必要となる DB2 表を作成します。DB2 表のセットアップ方法の詳細については、ご使用のワークステーション・プラットフォーム用の資料 *DB2 ユニバーサル・データベース... 概説およびインストール* をまず参照してください。
-

z/OS または OS/390 をセットアップするためのステップ

ほとんどすべてのアプリケーション開発、アセンブリー、およびインストール作業はワークステーション上で行われますが、一部の作業では、J2EE アプリケーションを配置する予定の z/OS または OS/390 システムに接続することが必要となります。ユーザーのサイトのシステム・プログラマーが WebSphere for z/OS をインストールするとき、UNIX アプリケーション開発環境を z/OS または OS/390 にセットアップすることもできます。このトピックの説明で最小限の要件がリストされているので、環境が正しいことをユーザー自身で検証できます。詳細について、必要であれば次の資料を参照してください。

- UNIX 環境のセットアップ方法については、*z/OS UNIX システム・サービス計画*, GA88-8639 を参照してください。
- FTP サーバーのセットアップ方法については、*z/OS Communications Server: IP Configuration Guide*, SC31-8775 を参照してください。WebSphere for z/OS 管理アプリケーションに関してあとで使用するのと同じユーザー ID およびパスワードを使用してください (必要に応じて *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください)。

z/OS または OS/390 をセットアップするために以下のステップを実行してください。

1. ワークステーションと、WebSphere for z/OS が常駐する z/OS または OS/390 システムとの間に TCP/IP 接続が確立されていることを確認します。検査方法の 1 つとして、コマンド・プロンプト・ウィンドウを開き、TCP/IP ホスト名を指定して PING コマンドを入力するという方法があります。

2. z/OS または OS/390 UNIX 上で、各アプリケーション開発者の領域サイズ (BPXPRMxx の MAXASSIZE、または RACF ADDUSER コマンドか ALTUSER コマンドの ASSIZEMAX) を検査してください。経験的に言えるのは、可能な限り最大の領域サイズで実行することです。しかし、最小サイズの 256 MB で開始してください。サイズは IEFUSI 出口、JES2 EXIT06、JES3 IATUX03、または TSO セグメントのデフォルトで制限される可能性があります。コンパイラーのメモリーが不足している場合、アプリケーション開発者の領域サイズを増やす必要がある場合があります。

3. z/OS または OS/390 上で、HFS への書き込みアクセス権をもつ ftp サーバーをセットアップしてください。

第5章 J2EE サーバーにインストールされる新規アプリケーション・コンポーネントの作成

21ページの『第4章 アプリケーション開発環境の設定』に記載されているように適切なワークステーション・ソフトウェアのインストールまたはアップグレードを行ったあとでは、J2EE アプリケーション・コンポーネントを作成する準備ができています。J2EE アプリケーション・コンポーネントの開発に関する資料は、他の WebSphere Application Server、Sun Microsystems、およびサード・パーティーの文書から得ることができるので、以下のトピックでは、WebSphere for z/OS J2EE サーバーの EJB コンテナまたは Web コンテナにインストールするためにアプリケーション・コンポーネントをコーディングするときに考慮すべき規則やガイドラインだけを示します。

Enterprise bean または Web コンポーネントを開発するとき、コンポーネントごとのビジネス・ロジックとアプリケーション・ロジックだけに専念することができます。全社的情報システム (たとえば、DB2 データベース、エンタープライズ・リソース・プランニング・システム、CICS や IMS のようなメインフレーム・システム、RDBMS、レガシー・アプリケーションなど) へのアクセスに関連したトランザクション、セキュリティ、およびスケーラビリティを処理するよう、J2EE サーバーの Web コンテナと EJB コンテナは設計されています。

このトピックでは、J2EE サーバーにインストールされる以下のコンポーネントのタイプのための開発とテストに関するガイドラインを示します。

- Sun Microsystems Enterprise JavaBeans (EJB) 1.1 および 1.0 仕様に対して作成される Enterprise bean
- Sun Microsystems Java Servlet Specification, V2.2 に対して作成される Web アプリケーション

J2EE アプリケーション・クライアントの開発については、67ページの『第8章 J2EE アプリケーション・クライアントの作成と実行』を参照してください。

Enterprise bean の作成

WebSphere for z/OS J2EE サーバーでは、以下のすべてのタイプの Enterprise JavaBeans がサポートされています。

- ステートレス・セッション
- ステートフル・セッション
- コンテナ管理 (CMP) エンティティー
- bean 管理 (BMP) エンティティー

これらの用語について、または Sun Microsystems EJB 仕様に対する Enterprise bean のコーディングの概念と要件についてよく知らない場合は、使用する bean プログラミング・モデル、アプリケーション・プログラミング・インターフェース、およびサービスに関する情報を得るために、IBM または IBM 以外の情報源を利用できます。32ページの『Enterprise bean の開発』で説明している z/OS and OS/390 に特有の一時的制限事項にも、必ず注意してください。

bean の開発方法としては、IBM VisualAge for Java EJB Development Environment を使用するのが最も簡単です。これを使用すると、EJB 1.0 仕様に定義されている分散コンポーネント・アーキテクチャーに準拠する Enterprise bean のコード化とテストを行うことができます。VisualAge for Java のこのフィーチャーでは、bean 配置環境をシミュレートするために使用できるコード生成テスト・クライアントおよび WebSphere Application Server ランタイムも提供されます。IBM EJB Development Environment を使用すると、開発者は、JNDI ルックアップ、リモート・メソッドの呼び出し、およびホーム・インターフェース上のメソッドの呼び出しを含めて、エンティティーおよびセッション bean を十分にテストすることができます。

推奨: ワークステーション上で Enterprise bean をテストできますが、z/OS または OS/390 プラットフォーム上でもテストする必要があります。そのわけは、ランタイム環境が微妙に違っている場合があるからです。たとえば、bean が DB2 を使用する場合、ワークステーション上と z/OS または OS/390 上の DB2 サポートの違いが明らかになることがあります。ワークステーションと z/OS または OS/390 のランタイム環境の潜在的な違いを確認するために、この章のガイドラインを利用してください。

Enterprise bean の開発のためのチェックリスト

以下のチェックリストを利用して、Enterprise bean の作成に関連した必要な作業すべてを完了していることを確認してください。

始める前に:

- この開発過程の詳細については、以下のリソースのいずれかを利用してください。
 - WebSphere for z/OS 用の予防サービス計画 (PSP) バケット
 - Sun Microsystems EJB 1.1 または 1.0 の仕様書、あるいはこれに相当する情報源
 - 選択したアプリケーション開発ツールの使用方法に関する説明。この説明は、製品資料の中にあります。製品資料は、ソフトウェア製品のリンクをクリックして IBM ホーム・ページ (www.ibm.com) から入手できます。たとえば、VisualAge for Java IBM EJB Development Environment の使用方法に関する資料を見つけるには、IBM ホーム・ページをご使用のブラウザにロードしたあとで、以下のステップに従ってください。
 1. 次のリンクをクリックします。「Products」→「Software」
→「WebSphere」→「Products」→「VisualAge for Java」
 2. 左側のフレーム内の「Library」をクリックしたあと、スクロールダウンして「Product Documentation」をクリックします。
 3. 「IBM VisualAge for Java 3.5 PDF Documentation」のリンクをクリックします。
 4. カテゴリー「Developing EJB components」までスクロールダウンし、[ejb.pdf](#) をクリックします。

表 5. Enterprise bean の開発のためのチェックリスト

✓	作業
<input type="checkbox"/>	作成しようとする bean について新規プロジェクトおよびパッケージを追加する。 注: プロジェクトとは、VisualAge for Java でのみ使用される概念です。
<input type="checkbox"/>	名前と型を指定して Enterprise bean を追加する。
<input type="checkbox"/>	bean のクラス属性とインターフェースを定義する。
<input type="checkbox"/>	必要に応じてインポート・ステートメントを追加する (たとえば、特定の機能についての関連 bean クラスおよび Java クラス)。
<input type="checkbox"/>	bean クラスのビジネス・ロジック・メソッドをコード化する。

表 5. Enterprise bean の開発のためのチェックリスト (続き)

✓	作業
□	<p>CMP bean を作成する場合のヒント:</p> <ul style="list-style-type: none"> • ファインダー・メソッドをサポートするファインダー・ヘルパー・インターフェースをツールが生成できるようにする。 • どの bean 属性でコンテナ管理パーシスタンスが必要とされるかを定義する。 • データ・アクセスに必要なクラス (たとえば、DB2 JDBC クラス) をインポートする。 • 接続タイプとデータ・ソースを指定してデータベース・スキーマをインポートする。 • データベース・スキーマから適切なテーブルのマップを生成する。 • テーブル・マップ内の関連列名に bean 属性をマップする。
□	<p>BMP bean を作成する場合のヒント:</p> <ul style="list-style-type: none"> • bean のライフ・サイクルを管理するメソッドをコード化する。 • 動的 SQL を使用している場合、DB2 表の作成時に使用されるものと一致する高位修飾子が #sql ステートメントに含まれていることを確認する。
□	<p>次のようなプロパティを追加する。</p> <ul style="list-style-type: none"> • BeanHome の JNDI 名 • トランザクション属性
□	<p>bean について配置されるコードを生成する。 注: CMP bean の場合、スタブ、タイ、およびパーシスタンスを生成する必要があります。</p>
□	<p>アプリケーション・コンポーネントをテストする。</p>
□	<p>bean を JAR ファイルにパッケージする。</p> <p>ヒント: 34ページの『JAR ファイルへの bean のパッケージ化』を参照してください。</p>

Enterprise bean の開発

WebSphere for z/OS J2EE サーバーにインストールする Enterprise bean を開発するとき、以下の点に留意してください。

規則:

- コンテナ管理パーシスタンス (CMP) は、VisualAge for Java を使用して作成された Enterprise bean についてのみサポートされています。

- bean がトランザクションを管理する (つまり、「bean 管理トランザクション」属性をもっている) 場合、WebSphere for z/OS が非コミット・トランザクションを解決する際の以下の規則と方式を認識する必要があります。
 - bean がもっている非コミット・ローカル・トランザクションまたはリソースを bean が解決するまでは、グローバル・トランザクションを開始できません。
 - グローバル・トランザクションでは、bean がそのトランザクションを解決しないでその処理を終了した場合、WebSphere for z/OS は、bean のトランザクションをロールバックします。
 - bean がリソース・マネージャーのローカル・トランザクションを解決しないでその処理を終了した場合、WebSphere for z/OS は、そのトランザクションをロールバックします。
- WebSphere for z/OS は、Enterprise bean について役割に基づいたセキュリティをサポートしています。セキュリティ検査を実行するよう bean をコード化したい場合、isCallerInRole メソッドと getCallerPrincipal メソッドを使用できます。
規則: 役割名には空白を入れることはできません。また、役割名は、245 文字を超えることができません。ただし、役割名は大文字小文字混合にすることができます。

ガイドライン:

- DB2 データにアクセスするために JDBC アプリケーション・プログラミング・インターフェース (API) を使用するよう、bean をコード化することができます。追加情報を得るには、以下を参照してください。
 - z/OS または OS/390 JDBC ドライバーで DB2 を使用方法の説明については、*DB2 適用業務プログラミングおよび SQL の手引き*, SC88-7377
 - JDBC API の詳細については、<http://java.sun.com/products/jdbc>
- パーシスタント・データを保管するために、コード化する bean で DB2 が必要とされる場合、DB2 for OS/390 バージョン 7.1 と一緒に提供されるタイプ 2 JDBC ドライバーを使用する必要があります。このドライバを使用する場合、以下のことを実行するように bean を設計できます。
 - グローバル・トランザクションまたはローカル・トランザクションに参加する。
 - DB2 との接続を複数もつ。(ただし、接続が 1 つだけのほうがパフォーマンスは良好です。)

- それぞれの DB2 接続ごとに、異なる結果を割り当てる。この場合、bean は、グローバル・トランザクションで実行されません。

JAR ファイルへの bean のパッケージ化

VisualAge for Java を使用し、Enterprise bean を開発してテストする場合、「エクスポート」→「Jar ファイル」オプションを選択して bean をパッケージします。JAR ファイルに組み込まれる bean、クラス、およびリソースを、この過程で選択します。

推奨: z/OS または OS/390 上で必要なアプリケーションのパーツだけに JAR ファイルの内容を制限する (これにより、そのアプリケーションの将来の保守が比較的簡単になる) ために、以下のことを実行してください。

- VisualAge for Java がエクスポートするクラスの詳細を検討します。左側にチェックマークの付いたクラスは、そのチェックマークをクリックしてクラスを選択解除しない限りはエクスポートされます。たとえば、この bean がパーシスタント・データ・ストアとして DB2 を使用する場合、VisualAge for Java は、bean を作成するためにインポートした DB2 JDBC クラスを自動的に組み込みます。これらのクラスは z/OS または OS/390 上にすでにあるので、bean の JAR ファイルにこれらを組み込む必要はありません。
- VisualAge for Java の「参照される型およびリソースの選択」オプションを使用しないで JAR ファイルをエクスポートすることを考慮してください。アプリケーションをパッケージすると (39ページの『第6章 J2EE アプリケーションのアセンブル』で説明しているように)、その結果の EAR ファイルには、J2EE サーバーに必要なクラスが含まれることになります。

Web アプリケーションの作成

Web アプリケーションには、以下の Web コンポーネントが 1 つ以上含まれることがあります。

- サブレット
- JavaServer Pages (JSP)
- ユーティリティ・クラス
- 静的文書

各 Web コンポーネントが Web アプリケーションで果たすべき役割は、下記で参照できる Java™ Servlet Specification, V2.2 に定義されています。

<http://www.javasoft.com>

Web コンポーネントの開発

Web アプリケーションのインスタンスは、単一の WebSphere for z/OS J2EE サーバーの中に存在します。これは、サーブレットのコンテキストを使用して、他のローカル・オブジェクトへの参照を取得し、データを他のアプリケーションと共有します。配布可能とマークされていれば、サーブレットを複製することができます。ただし、この場合、次の条件の 1 つが該当するよう保証する必要があります。

- サーブレットへの書き込みが行われることなく、あとで処理するためにオブジェクトがそのままにしておくられる。
- シスプレックス内で複製できない J2EE サーバーにサーブレットがインストールされる。

Web アプリケーションを J2EE サーバーにインストールするには、その前に、以下のことを行う必要があります。

1. Web アプリケーションのコンポーネントのすべてが Web アプリケーション (WAR) ファイルにパッケージされる必要があります。このファイルにはファイル拡張子 `.war` が付いています。また、`web.xml` ファイルがこのファイルに組み込まれる必要があります (`web.xml` ファイルは、クライアントに要求されたときにアプリケーションがどのように提供されるかについて指示します)。
2. この WAR ファイルは、次に、Enterprise アーカイブ (EAR) ファイルの一部としてパッケージされる必要があります (このパッケージ化を実行する方法については、41ページの『新規 J2EE アプリケーションをアSEMBルするためのステップ』を参照してください)。EAR ファイルは、基本的には、特定のディレクトリー構造とフォーマットをもつ JAR ファイルであり、拡張子 `.ear` が付いています。EAR ファイルにパッケージされた WAR ファイルまたは EJB JAR ファイルのすべてを結合する記述的メタ情報を含む `application.xml` ファイルが、この EAR ファイルに組み込まれます。

アプリケーション・コンポーネント提供者のアプリケーションに関するビジネス・ロジックとアプリケーション・ロジックの作成は、アプリケーション・コンポーネント提供者が担当する必要があります。アプリケーション・コンポーネント提供者は、エンタープライズ情報システム (EIS) アクセスに関連したトランザクション、セキュリティ、およびスケーラビリティを処理するために、Web コンテナと EJB コンテナを信頼できます (EIS には、DB2 データベース、エンタープライズ・リソース・プランニング・システム、CICS や IMS などのメインフレーム・システム、RDBMS、およびレガシー・アプリケーションが含まれます)。

WebSphere for z/OS では、サーブレットについて役割に基づいたセキュリティーがサポートされています。セキュリティー検査を実行するようサーブレットをコード化したい場合、isCallerInRole メソッドと getCallerPrincipal メソッドを使用できます。**規則:** 役割名にはブランクを入れることはできません。また、役割名は、238 文字を超えることができません。ただし、役割名は大文字小文字混合にすることができます。

WAR ファイルへの Web コンポーネントのパッケージ化

結果的に作成される WAR ファイルが以下の条件を満たす限り、JAR ファイルの作成のために使用するツール、たとえば、IBM WebSphere Studio 製品などを使用して、WAR ファイルを作成できます。

- web.xml ファイルが web-inf ディレクトリーに含まれる。
- 組み込まれるすべての JAR ファイルが web-inf/lib ディレクトリーにある
- 組み込まれるファイルおよび JSP は、jar の最上部をルートとしている。

サーブレット、サーブレット・クラス・ファイル、JSP および静的ファイルが組み込まれる war ファイルの例を以下に示します。

```
/mywebApplication
  index.html
  other.html
  CatalogDisplay.jsp
  /gifs
    logo.gif
  /web-inf
    web.xml
    /classes
      /com
        /mycorp
          /CatalogServlet.class
  /lib
    myItems.jar
```

この WAR ファイルに含まれる web.xml ファイルは、以下のようなものです。

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc... ..dtd"><
web-app><
display-name>A Simple Application</display-name><
context-param><
param-name>Webmaster</param-name><
param-value>webmaster@mycorp.com</param-value><
/context-param>

<servlet><
servlet-name> catalog</servlet-name><
servlet-class>com.mycorp.CatalogServlet</servlet-class><
init-param><
```



```
param-name>catalog</param-name><
param-value>Spring</param-value><
/init-param><
/servlet><
servlet mapping><
servlet-name>catalog</servlet-name><
url-pattern>/catalog</url-pattern><
/servlet mapping>
```

第6章 J2EE アプリケーションのアセンブル

J2EE アプリケーションを WebSphere for z/OS J2EE サーバーにインストールするには、その前に、このアプリケーションのインストールの準備を行う必要があります。つまり、個々のコンポーネントの JAR ファイルまたは WAR ファイルを Enterprise アーカイブ (EAR) ファイルにパッケージし、必ずすべてのコンポーネント参照を解決できるようにする必要があります。EAR ファイルは、基本的には、特定のディレクトリー構造とフォーマットをもつ JAR ファイルであり、拡張子 `.ear` が付きます。ユーザーのアプリケーション用の EAR ファイルを作成するには、WebSphere for z/OS アプリケーション組み立てツールを使用してください。

このアセンブリー処理中に、単一アプリケーションの個々のコンポーネントを定義するだけでなく、WebSphere for z/OS 環境のコンポーネントの配置も行います。アプリケーション組み立てツールは、`application.xml` ファイル (EAR ファイルの内容の一部) を生成します。このファイルには、単一アプリケーションにパッケージする JAR ファイルまたは WAR ファイルのすべてを結合する記述的メタ情報が入っています。このメタデータによって、J2EE サーバーは、インストールされる J2EE アプリケーションの内容と個々のコンポーネントの要件を理解できます。

新規アプリケーションをパッケージする以外に、アプリケーション組み立てツールを使用して、既存の J2EE アプリケーションの内容とこれに関連した EAR ファイルを変更することもできます。

アプリケーション組み立てツールを使用して J2EE アプリケーションをアセンブルするためのサブタスクおよび関連した手順を、次の表に示します。

サブタスク	関連した手順 (参照項目)
ワークステーションへの アプリケーション組み立てツールのインストール	40ページの『アプリケーション組み立てツールをインストールするためのステップ』
新規 J2EE アプリケーションのアセンブル	41ページの『新規 J2EE アプリケーションをアセンブルするためのステップ』

アプリケーション組み立てツールをインストールするためのステップ

始める前に:

- ご使用のシステムで アプリケーション組み立てツールの最新コピーを入手する方法を調べてください。IBM では、このツールを WebSphere Application Server Web サイト (<http://www.ibm.com/software/webservers/appserv/>) を介して提供します。
- 最新のコードに適用される一時的制限や指示について理解するために、必ず、アプリケーション組み立てツールの Readme ファイルを読んでください。
- WebSphere for z/OS アプリケーション組み立てツールのコピーがご使用のワークステーションにすでにインストールされている場合、それを除去してから最新版をインストールしてください。

以下のステップを実行し、アプリケーション組み立てツールをワークステーションにインストールしてください。

1. アプリケーション組み立てツールのコードをワークステーションの一時ディレクトリーにダウンロードするかコピーします。

-
2. ワークステーションの一時ディレクトリーで アプリケーション組み立てツールの `aatxxxx.exe` ファイルを見つけて選択したあと、左マウス・ボタンをダブルクリックします。

結果: 「InstallShield ウィザード (InstallShield Wizard)」で言語設定ウィンドウが表示されます。

-
3. 「OK」 をクリックし、デフォルト設定の言語 (「英語 (English)」) を選択します。

結果: 初期の準備が終わると、「ウェルカム (Welcome)」ウィンドウが画面に表示されます。

-
4. ツールの「ウェルカム (Welcome)」ウィンドウで「次へ (Next)」をクリックします。

結果: ライセンス承諾ウィンドウが表示されます。

-
5. 該当のラジオ・ボタンを選択してライセンス承諾条件を受け入れ、「次へ (Next)」をクリックします。

結果: カスタマー情報ウィンドウが表示されます。

-
6. デフォルトのユーザー名（「IBM ユーザー (IBM user)」）を受け入れ、自分専用ツールをインストールするラジオ・ボタンを選択し、「次へ (Next)」をクリックします。

結果: タイプ設定ウィンドウが表示されます。

7. 完全なツールのインストールを行うラジオ・ボタンを選択し、「次へ (Next)」をクリックします。

結果: 「インストールの準備完了 (Ready to install)」ウィンドウが表示されます。

8. 「インストール (Install)」ボタンをクリックします。

結果: 「InstallShield ウィザード完了 (InstallShield Wizard Completed)」ウィンドウが表示されるまで、進行状況バーに状況が示されます。

9. 「終了 (Finish)」ボタンをクリックします。

結果: 「InstallShield ウィザード (InstallShield Wizard)」ウィンドウがクローズします。

この時点で、C:\Program Files\IBM の下に WebSphere 390 Application Assembly というサブディレクトリーが示されています。

新規 J2EE アプリケーションをアセンブルするためのステップ

新規 Java 2 エンタープライズ版 (J2EE) アプリケーションを J2EE サーバーにインストールするには、その前に、このアプリケーションのインストールの準備を行う必要があります。この準備作業として、アプリケーション組み立てツールを使用して以下のことを実行します。

1. 適切なアプリケーション開発ツールで作成した J2EE アプリケーション・コンポーネントをインポートし、1 つのアプリケーションにそのコンポーネントをパッケージします。
2. それぞれのアプリケーション・コンポーネントごとの、またはアプリケーション自体の配置記述子内の属性の定義、検査、訂正を行います。
3. アプリケーションをエクスポートします。その結果、アプリケーション組み立てツールによって Enterprise アーカイブ (EAR) ファイルが作成されます。このファイルは、アプリケーションをインストールするために管理アプ

リケーションで必要とされる入力フォーマットです。コンポーネント・コード (JAR ファイルおよび WAR ファイル) のすべてと J2EE アプリケーションの配置記述子が、EAR ファイルによって一緒にパッケージされます。

これ以降の説明では、アプリケーション組み立てツールをはじめて使用するという状況を想定しています。この説明で、新規アプリケーションをインポートして配置するための主要な作業について手引きします。このツールに習熟するために、「ヘルプ (Help)」 → 「目次 (Contents)」を選択し、「クイック・スタート (Quick Start)」トピックを表示することによって、アプリケーション組み立てツールのヘルプ情報を利用できます。

ユーザー独自のコンポーネントをアSEMBルするためにこのツールの使用を開始するときには、コンポーネント開発のために推奨ツールおよび推奨仕様レベルを使用していることを確認してください。これらについては、次の章で説明しています。

- 21ページの『第4章 アプリケーション開発環境の設定』
- 29ページの『第5章 J2EE サーバーにインストールされる新規アプリケーション・コンポーネントの作成』

このツールをはじめて始動したとき、ウィンドウの左側のフレームには、Applications というフォルダーだけが入っています。アプリケーション名の定義および J2EE アプリケーション・コンポーネントのインポートを開始すると、左側のフレームには、そのアプリケーションとコンポーネントが階層ツリー構造で表示されます。そのコンポーネントに関連したタスクを実行するために、次の方法のいずれかによってコンポーネントと作業を選択できます。

- 左マウス・ボタンを使用して、左側のフレームでアプリケーションまたはコンポーネントのラベルを選択し、次に、右マウス・ボタンを使用して、作業のポップアップ・メニューを表示します。

ヒント: 代わりに、右マウス・ボタンを使用してオブジェクトをクリックするという方法もあります。こうすると、そのオブジェクトが自動的に選択され、ポップアップ・メニューが表示されます。

- 左マウス・ボタンを使用して、左側のフレームでアプリケーションまたはコンポーネントのラベルを選択し、次に、左ボタンを使用して、プルダウン・メニューのラベルまたはツールバー・アイコンをクリックします。

始める前に:

- 40ページの『アプリケーション組み立てツールをインストールするためのステップ』で説明したように、ワークステーション環境をすでに正しく更新し、アプリケーション組み立てツールをインストールしたことを確認してください。
- 最新のコードに適用される一時的制限や指示について理解するために、アプリケーション組み立てツールの **Readme** ファイルを読んでください。
- **bean** やサーブレットの属性またはその他の情報を調べる必要がある場合には、Sun Microsystems の該当の仕様書 (あるいはこれに相当する解説書) を入手することを考慮してください。アプリケーション組み立てツールのヘルプ情報にはこの種の情報も記載されているので、他の解説書は必要がないこともあります。

以下のステップを実行し、J2EE サーバーにインストールするための新規 J2EE アプリケーションをアSEMBルします。

1. アプリケーション組み立てツールを始動するには、「スタート」 → 「プログラム」 → 「IBM WebSphere for z/OS」 → 「アプリケーション組み立て」の順にクリックします。

結果: アプリケーション組み立てツール・ウィンドウが表示され、Applications フォルダが選択されています。

-
2. 新規アプリケーションを定義するには、「選択 (Selected)」 → 「追加 (Add)」の順にクリックします。右側のフレームに、次の情報の値を入力します。
 - a. アプリケーション表示名。これは、左側のフレームに現れるフォルダの名前です。
 - b. (オプション) アプリケーション記述。これは、この新規アプリケーションについて簡単に説明する任意のテキストです。

-
3. これらの値を保管するディスク・アイコンをクリックします。Applications フォルダの下に、入力したばかりのアプリケーション表示名と一致するラベルのある新規ノードが現れます。
 4. アプリケーション・ノードを展開し、インポートするアプリケーション・コンポーネントのタイプのフォルダを表示します。

-
5. アプリケーション・コンポーネントをインポートするために、インポートしたいアプリケーション・コンポーネントのタイプのフォルダを強調表

示し、「インポート (Import)」を選択します。「インポート (Import)」ダイアログから以下のことを行います。

- a. 「選択 (Choose)」ボタンをクリックし、当該アプリケーションにインポートしたい JAR ファイルまたは WAR ファイルを選択します。
- b. 「選択 (Select)」をクリックし、JAR ファイルまたは WAR ファイルの絶対パス名を入力します。
- c. 「OK」をクリックし、インポート・プロセスを開始します。

ガイドライン: 一部のアプリケーション・コンポーネントでは、追加ファイルを EAR ファイル内にパッケージすることが必要となる場合があります。EAR ファイルに対して行われる可能性のある追加について以下のガイドラインで示し、アプリケーションのパーツの数を制限したりファイルの重複を回避するための推奨を提供します。「ファイル (Files)」プロパティの「インポート (Import)」ボタンを使用して、その追加ファイルをインポートできます。

- Web コンポーネントの場合、web.inf ファイルを必ずインポートしてください。
- ファインダー・ヘルパーのある CMP bean の場合、関連 VAP.jar ファイルを組み込んでください。
- 特定のユーティリティまたは関数を必要とするコンポーネントの場合、その関数に対する JAR ファイルを組み込んでください。アプリケーション・コンポーネントやその z/OS または OS/390 クライアントが同じユーティリティや機能を共有する場合、その JAR ファイルを EAR ファイルから除外することを考慮してください。代わりに、その JAR ファイルを z/OS または OS/390 に転送し、以下に対する CLASSPATH 変数にそのファイルを組み込むことができます。
 - その関数を必要とする z/OS または OS/390 クライアント
 - アプリケーション・コンポーネントがインストールされる J2EE サーバー

-
6. ビューを展開し、インポートしたばかりの JAR ファイルまたは WAR ファイル内のコンポーネントをリストします。

アプリケーション・ツリーを展開し、インポートされたファイル内のコンポーネントを表示するとき、アプリケーション組み立てツールは、配置記述子の中にエラーを検出すると、次のメッセージを表示します。

```
BB094030E Error(s) detected in application_name deployment descriptors.  
See message log for details.
```


ヒント: このようなエラーの詳細情報を表示するには、「ファイル (File)」 → 「メッセージ・ログ (Message log)」を選択するか、またはメッセージ・ログのツールバー・アイコンをクリックします。この場合、訂正する必要のあるエラーのチェックリストとして、このメッセージ・ログを利用できます。

推奨: アプリケーション組み立てツールを使用してコンポーネント、JAR ファイル、または WAR ファイル内のエラーを訂正できますが、アプリケーション・コンポーネントを作成するときに使用したアプリケーション開発ツール (VisualAge for Java など) を使用して訂正を行ってください。

-
7. 以下のプロセスによって、各アプリケーション・コンポーネントに関連したプロパティーを変更するか更新します。そのプロパティーは、それぞれのアプリケーション・コンポーネントごとの配置記述子の中に現れる属性です。

J2EE アプリケーションを構成するそれぞれのコンポーネントごとに、このプロセスを反復してください。コンポーネント・プロパティーのいずれかについてヘルプが必要な場合、右マウス・ボタンをクリックしてそのプロパティーを選択したあと、「ヘルプ (Help)」をポップアップ・メニューから選択してください。

- a. 変更したいアプリケーション・コンポーネントについて「選択 (Selected)」 → 「変更 (Modify)」の順にクリックします。コンポーネントのプロパティーが アプリケーション組み立てツールウィンドウの右側のフレームに表示されます。
- b. 各種プロパティーをナビゲートするには、このウィンドウの右側のフレームにあるタブを使用します。このコンポーネント・プロパティーは、Sun Microsystems の該当の仕様に対応します。
- c. 選択したアプリケーション・コンポーネントに対する変更を完了したあとで、変更内容を保管します。変更内容を保管するときに、アプリケーション組み立てツールは、変更されたプロパティー値にエラーがあれば検出し、メッセージ・ログに詳細情報を表示します。

規則:

- 現在選択しているコンポーネントに対する変更を保管するか取り消すまでは、別のコンポーネントを選択することはできません。
- アプリケーション組み立てツールが検出したエラーは、訂正することが必要です。訂正しないと、ランタイムの結果は予測できないものとなります。

ガイドライン: アプリケーションまたはコンポーネントをインポートしたときに、アプリケーション組み立てツールがエラーをまったく検出しなかった場合、アプリケーションを J2EE サーバーにインストールするためのアプリケーション・プロパティまたはコンポーネント・プロパティを何も変更する必要はありません。しかし、場合により、以下の事項を実行することが必要になります。

- z/OS または OS/390 上にインストールされるアプリケーションについてご使用のシステムで推奨されている命名規則に一致するよう、アプリケーションまたはコンポーネントの名前を変更する。

規則: bean 名は、特定の JAR ファイル内で固有のものにする必要があります。

- アプリケーションで次のものが必要とされるかどうかを決定する。
 - コンテナ・トランザクション・エレメント
 - セキュリティー役割とメソッド許可

Enterprise bean またはサブレットについてセキュリティ役割を使用する場合は、個々のコンポーネントか J2EE アプリケーションのいずれかに関する配置記述子に、役割名を指定してください。アプリケーション・レベルの役割によって、コンポーネント・レベルの役割は変更されます。

規則:

- bean またはサブレットが `isCallerInRole` メソッドか `getCallerPrincipal` メソッドを使用する場合、コード内に指定された役割が、個々のコンポーネントのセキュリティ役割名プロパティの中に現れます。この役割名を変更することはできません。しかし、アプリケーション・レベルのセキュリティ役割名をリンクとして指定すると、その役割をアプリケーション・レベルのセキュリティ役割にリンクすることができます。
- 役割名には空白を入れることはできません。また、役割名は、245 文字を超えることはできません。ただし、役割名は大文字小文字混合にすることができます。
- 役割名は、EJBROLE または GEJBROLE クラスに指定されたプロファイルにマップする必要があります。このクラスは、セキュリティ管理者によってご使用のシステムに定義されます (これらのクラスの定義方法については、50ページの『OS/390 の手動作業を完了するためのステップ』で説明しています)。
- 個々のコンポーネントについて定義するそれぞれのセキュリティ役割ごとに、それぞれの役割がアクセスできるコンポーネント・メ

ソッドを必ず選択してください。その選択を行うために、メソッド許可プロパティ・フォームを使用してください。

-
8. 単一の J2EE アプリケーションにアセンブルするコンポーネントすべてについて、ステップ 5 ~ 7 までを反復します。
-
9. アプリケーションの内容の妥当性検査を行うために、ツリー内のアプリケーションを選択したあと、「妥当性検査 (Validate)」を選択します。
ヒント: ステップ 5 ~ 7 までを終了したあとで各コンポーネントの個別の妥当性検査を行うほうが、アプリケーション全体の妥当性検査を行うよりも速い場合があります。
-
10. アプリケーションを配置するために、ツリー内のアプリケーションを選択したあと、「配置 (Deploy)」を選択します。配置プロセスが完了すると、メッセージ BBO94009I がステータス・バーに表示されます。
-
11. 配置されるアプリケーションをエクスポートするために、ツリー内のアプリケーションを選択したあと、「エクスポート (Export)」を選択します。「アプリケーションのエクスポート (Export application)」ウィンドウが開きます。
-
12. 「アプリケーションのエクスポート (Export application)」ウィンドウで、新規または既存 EAR ファイルの絶対パス名を入力するか、あるいは「選択 (Choose)」をクリックして、既存 EAR ファイルをブラウズしたり新規 EAR ファイルに適した場所をブラウズすることができます。
推奨: ユーザー独自のアプリケーションをエクスポートするとき、z/OS または OS/390 上で J2EE サーバーにインストールする準備ができているアプリケーションを簡単に見つけることができる特定のフォルダーまたはサブディレクトリを設定して使用することを考慮してください。
結果: アプリケーション組み立てツールは、ユーザーのアプリケーションについて、およびコンポーネントの JAR ファイルか WAR ファイルのそれぞれについて、XML ファイルを新たに作成するか、または既存の XML ファイルを更新します。これらの XML ファイルには、コンポーネントのプロパティについて入力した値が含まれ (その値がある場合)、これらの

XML ファイルによって、J2EE サーバーは、インストールされた J2EE アプリケーションの内容を理解し、このアプリケーションを正確に管理できるようにになります。

サンプル:

- 2 つの WAR ファイルと 1 つの EJB JAR ファイルを含むアプリケーションの EAR ファイルの内容:

```
/usr/MyApp

    EJB123.jar
    webappABC.war
    myItems.war

    /meta-inf
        application.xml
        manifest.mf
```

- 同じ EAR ファイルの application.xml ファイルの内容:

```
<?xml version="1.0 encoding="ISO-8859-1"?>
<application>
<display-name>MyApp</display-name>

<module><
web><
web-uri>webappABC.war</web-uri><
context-root>/Payroll</context-root><
/web><
/module>

<module><
web><
web-uri>myItems.war</web-uri><
context-root>/MyTools</context-root><
/web><
/module>

<module><
ejb>EJB123.jar</ejb><
/module>

</application>
```

ユーザーのアプリケーションが EAR ファイルにエクスポートされたことを示すメッセージ BBO94010I がステータス・バーに表示され、処理が完了したことが分かります。

第7章 J2EE サーバーのランタイム環境の作成

J2EE アプリケーションをインストールするために使用する EAR ファイルを作成したので、今度はユーザーのアプリケーションが稼働する WebSphere for z/OS J2EE サーバーを作成する必要があります。J2EE サーバー、つまり、ランタイム環境は、以下のエレメントで構成されます。

- アプリケーション環境を表す汎用サーバー。サーバーは、ある種の作業の役割を担うエンティティです。
- ユーザーのアプリケーションが実行するサーバー・インスタンス。サーバー・インスタンスは、1 つの制御領域と少なくとも 1 つのサーバー領域から構成されています。制御領域は、作業要求を受け入れ、サーバー領域は、アプリケーションの実際のランタイム環境です。これには、Java 仮想マシン (JVM) と EJB コンテナが含まれ、おそらく、Web コンテナも含まれます。(サプレットまたは JSP を含む J2EE アプリケーションをインストールしようとする場合、追加作業をいくつか完了して J2EE サーバー用の Web コンテナを構成する必要があります。)
- J2EE リソースと J2EE リソース・インスタンス。これらは、汎用タイプのサブシステムと特定のサブシステムをそれぞれ識別します。このリソースは、たとえば、J2EE サーバーにインストールされたコンポーネントのパーススタント・データ・ストアを管理するサブシステムである場合もあります。
- J2EE サーバー内のコンポーネントがリソースにアクセスできるようにする J2EE リソース接続。J2EE アプリケーションをインストールすると、管理アプリケーションは、この接続を自動的に作成します。

J2EE サーバーの作成に関する以下の説明には、エレメント用に使用する次の名前サンプルが含まれています。

- 汎用 J2EE サーバー用の **J2SERV**
- J2EE サーバー・インスタンス用の **J2SERV1**

サンプル名と同じにする必要はありません。ただし、別の名前を選択する場合は、50ページの『OS/390 の手動作業を完了するためのステップ』にリストされている規則に従い、アプリケーション環境を正しくセットアップしなければなりません。

この説明では、J2EE サーバー用の OS/390 ART ファイルを作成するためにコピーし、変更できるサンプル・ファイルがある場所も示してあります。

ユーザーのアプリケーション用の J2EE サーバーを作成するためのサブタスクおよび関連した手順を、次の表に示します。

サブタスク	関連した手順 (参照項目)
OS/390 の手動作業の完了	『OS/390 の手動作業を完了するためのステップ』
JCL プロシージャの作成	53ページの『制御およびサーバー領域用の JCL プロシージャを作成するためのステップ』
JVM プロパティの設定	53ページの『JVM のプロパティを設定するためのステップ』
J2EE サーバーが Web アプリケーションのホストになることを可能化 (オプション)	54ページの『Web アプリケーションに対する J2EE サーバーのサポートの使用可能化』
管理アプリケーションによる J2EE サーバーの定義と活動化	55ページの『サーバー構成の定義』

OS/390 の手動作業を完了するためのステップ

ご使用のシステムの規約によっては、WebSphere for z/OS 製品自体のインストールと検証または WebSphere for z/OS のテスト環境か実稼働環境のセットアップの一環として、手動作業の多くがすでに完了している場合もあります。その手動作業の資料はすでに利用可能になっているので、以下の手順では、作業の要約だけを示し、より詳細な説明がある情報源について言及しています。アプリケーション・コンポーネントをテストするために新規 J2EE サーバーの定義を始める前に、正しい環境がセットアップされていることを確認するためのチェックリストとして、この手順を利用してください。

以下のステップを実行し、新規 J2EE サーバーの定義に関連した z/OS または OS/390 の手動作業を完了してください。

1. J2EE アプリケーション・コンポーネント、J2EE サーバーの要素、および DB2 などの z/OS または OS/390 サブシステムに関する命名規則を決定します。J2EE サーバーの作成に関する以下の説明には、これらの要素用にサンプル名が含まれていますが、このサンプル名は、ご使用のシステムですでにセットアップしたかまたは使用する名前に置き換える必要があります。

命名規則の推奨事項については、*WebSphere Application Server V4.0 for z/OS and OS/390: 操作および管理*, SA88-8653 の付録を参照してください。

- 新規 J2EE サーバーおよびこのサーバーがホストとなるアプリケーションについて、ワークロード管理 (WLM) アプリケーション環境、サービス・クラス、および種別規則を定義します。アプリケーション環境を定義する (つまり、J2EE サーバーを WLM に対して定義する) には、IWMARIN0 ダイアログを使用して以下の値を入力します。

IWMARIN0 ダイアログのフィールド	使用する値
ランタイム・サーバー	EJB-DB2 application server のような J2EE サーバーの簡略説明を使用します。
アプリケーション環境名	J2SERV
サブシステム・タイプ	CB
プロシージャ名	J2SERV1
開始パラメーター	IWMSSNM=&IWMSSNM
サブシステム・インスタンス用のサーバー・アドレス・スペースの始動の制限	制限なし

IWMARIN0 ダイアログの使用法およびサービス・クラスと種別規則の定義方法についての詳細は、*z/OS MVS 計画: ワークロード管理, SA88-8574* を参照してください。

- データ・アクセスのためのデータベース・リソースまたはコネクタをセットアップします。システム・プログラマーまたはデータベース管理者が DB2 などの必要な z/OS または OS/390 サブシステムのインストールと構成をすでに行っていて、ユーザーのアプリケーションで使用されるデータベースや表をすでに作成している場合もあります。したがって、実行する必要があるのは次の作業だけの場合があります。

- J2EE サーバーを定義するとき、どのような DB2 サブシステム名を指定する必要があるかについて調べます。
- J2EE アプリケーション・コンポーネントが使用する必要のあるデータベース表を作成します。

必要であれば、データベース表の作成に関する説明について *DB2 管理の手引き, SC88-7376* を参照してください。

- ご使用のシステムのセキュリティー製品を使用して、セキュリティー・プロファイルおよび許可を定義します。ご使用のシステムのセキュリティー管理

者と共同でこの作業を遂行することが必要な場合もあります。セキュリティー・プロファイルおよび許可は、テスト・システムまたは実動システムに対するご使用のシステムのガイドラインによって、ある程度異なったものとなります。たとえば、テスト環境では、セキュリティー・メカニズムを何も使用しないで、J2EE アプリケーション・クライアントがテスト・システムおよびデータにアクセスすることを認める場合があります。クライアント・プログラムが J2EE サーバーと同じ z/OS または OS/390 システムで実行される時、この方法が特に適していることがあります。

ガイドライン:

- リソースへのクライアント・アクセスのためにセットアップするセキュリティーとは関係なく、特定の権限が J2EE サーバーには必要です。たとえば、J2EE アプリケーションで DB2 を使用する必要がある場合、J2EE サーバーは、DB2 計画 DSNJDBC へのアクセスが認可される必要があります。J2EE サーバーおよび J2EE アプリケーション・クライアントに対するセキュリティーをセットアップする方法の推奨事項と説明については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ* を参照してください。
- 役割に基づいたセキュリティーを必要とする J2EE アプリケーションをインストールする場合、プロファイルを EJBROLE または GEJBROLE クラスに定義する必要があります。そのあと、ユーザーまたはグループにこのプロファイルへの読み取りアクセスを許可する必要があります。

規則:

- EJBROLE または GEJBROLE クラスに指定するプロファイルは、次のフォーマットに従います。

`role_name`

この `role_name` は、次のいずれかに指定したセキュリティー役割属性と一致します。

- J2EE アプリケーション配置記述子
- 個々のアプリケーション・コンポーネントの配置記述子
- 役割名には空白を入れることはできません。また、役割名は、246 文字を超えることができません。ただし、役割名は大文字小文字混合にすることができます。

ご使用のシステムで z/OS または OS/390 SecureWay Security Server (RACF) を使用する時、次のコマンドの使用方法について z/OS *SecureWay Security Server (RACF) コマンド言語 解説書*, SA88-8617 を参照してください。

- プロファイルを EJBROLE または GEJBROLE クラスに定義する RDEFINE コマンド
 - このプロファイルへの読み取りアクセスをユーザーに認可する PERMIT コマンド
-

制御およびサーバー領域用の JCL プロシージャーを作成するためのステップ

環境変数を設定して、アプリケーション制御領域およびサーバー領域用の JCL プロシージャーを作成するには、TSO で以下のステップを実行します。

1. 作業用 PROCLIB データ・セットに J2SERV (汎用サーバー名) という名前の新規メンバーを作成します。BBOASR2 サンプル・メンバーを BBO.SBBOJCL からこの新規メンバーにコピーし、ファイル内のコメントに従って適切な更新を行います。WebSphere for z/OS 管理アプリケーションで指定するサーバー・インスタンス名を使用するように PROC ステートメントを変更します。たとえば、PROC ステートメントで次のような記述を行う必要があります。

```
//J2SERV PROC SRVNAME='J2SERV1'
```

2. また、PROCLIB でも、J2SERV1 (後で WLM に指定する JCL プロシージャー名) という名前の新規メンバーを作成します。BBOASR2S サンプル・メンバーを BBO.SBBOJCL からこの新規メンバーにコピーし、ファイル内のコメントに従って適切な更新を行います。たとえば、IWMSSNM パラメーターを編集し、WebSphere for z/OS 管理アプリケーションで指定するサーバー・インスタンス名を使用するよう次のように指定します:

```
IWMSSNM='J2SERV1'
```

JVM のプロパティーを設定するためのステップ

J2EE サーバーで稼働する Java 仮想マシン (JVM) 用に WebSphere for z/OS が使用するデフォルトの設定値を変更する場合にのみ、次の手順を実行してください。そのデフォルトを変更するには、使用するプロパティーと値を指定して JVM プロパティー・ファイルを作成します。

始める前に:

- サポートされている JVM プロパティ (155ページの『JVM プロパティおよびプロパティ・ファイル』にリストされている) を検討してください。このトピックには、JVM プロパティ・ファイルの位置と内容についての情報も記載されています。
- 既存の JVM プロパティ・ファイルを編集したり新規ファイルを適切なディレクトリーに保管するために、特別な許可が必要な場合があります。WebSphere for z/OS をご使用のテスト・システムにインストールしたシステム・プログラマーに問い合わせてください。

以下のステップを実行して、JVM プロパティ・ファイルをセットアップします。

1. 既存の JVM プロパティ・ファイルを編集するか、またはユーザー独自の JVM プロパティ・ファイルを作成し、この J2EE サーバーの環境変数を含む `current.env` ファイルを WebSphere for z/OS が入れるディレクトリーと同じ HFS ディレクトリーの中に、その JVM プロパティ・ファイルを入れてください。

規則: この JVM プロパティ・ファイルには `jvm.properties` と命名する必要があります。

-
2. 使用するプロパティ・キーとプロパティ値を追加するか設定するように JVM プロパティ・ファイルを編集します。
-
3. JVM プロパティ・ファイルに対する変更内容を保管します。
-

WebSphere for z/OS は、提供されたプロパティ・ファイルを見つけられないか使用できない場合には、デフォルトの JVM プロパティ値を使用して、サーバーをアクティブにするプロセスを継続します。

Web アプリケーションに対する J2EE サーバーのサポートの使用可能化

サーブレットまたは JSP を含む J2EE アプリケーションを WebSphere for z/OS J2EE サーバーにインストールする場合、J2EE サーバー構成を設定するために、以下の追加作業を完了する必要があります。

1. J2EE サーバーがサーブレットまたは JSP を管理するために使用する Web コンテナを構成します。
2. J2EE サーバーと Web ブラウザー間の通信を確立する HTTP サーバーを設定します。

これらの作業については、WebSphere for z/OS のWeb アプリケーション・サポートの使用可能化に関する資料を参照してください。その資料は、<http://www.ibm.com/jp/software/websphere/appserv/> で利用できます。

サーバー構成の定義

ユーザーのアプリケーションのランタイム環境を定義するには、システム管理エンド・ユーザー・インターフェース (SM EU) とも呼ばれる WebSphere for z/OS 管理アプリケーションを使用します。このランタイム環境を定義する作業、またはサーバー構成とは、J2EE サーバー、サーバー・インスタンス、およびデータ・ソースを定義する作業、ユーザーの J2EE アプリケーション用の EAR ファイルをインストールする作業などです。

環境変数の設定は、サーバー・レベルで定義することをお勧めします。そうすると、これらの設定はすべてのサーバー・インスタンスに適用されます。

始める前に: 以下のことが分かっている必要があります。

- 管理アプリケーションの使用方法に関する他のヘルプがどこにあるか。ヘルプは、管理アプリケーション自体、および *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 で得ることができます。
- どの環境変数をランタイム環境のために設定する必要があるか。この過程で設定できるランタイム変数と値の全リストについては、127ページの『付録A. 環境ファイルおよび JVM プロパティ・ファイル』を参照してください。

以下の表は、WebSphere for z/OS 管理アプリケーションを使用して、サーバー構成を定義するためのサブタスクおよび関連した手順を示しています。

サブタスク	関連した手順 (参照項目)
管理アプリケーションの始動	56ページの『管理アプリケーションを始動するためのステップ』
新規会話の開始	56ページの『会話を開始するためのステップ』
サーバーの追加	57ページの『J2SERV サーバーを追加するためのステップ』
サーバー・インスタンスの追加	59ページの『J2SERV1 サーバー・インスタンスを追加するためのステップ』
J2EE リソースの追加	59ページの『J2EE リソースを追加するためのステップ』
J2EE リソース・インスタンスの追加	60ページの『J2EE リソース・インスタンスを追加するためのステップ』

サブタスク	関連した手順 (参照項目)
J2EE アプリケーションのインストール	61ページの『J2EE アプリケーションをインストールするためのステップ』
新規会話の妥当性検査	64ページの『新規会話モデルの妥当性検査のためのステップ』
会話のコミット	64ページの『会話をコミットするためのステップ』

管理アプリケーションを始動するためのステップ

始める前に: WebSphere for z/OS を実行するマシンのネーミング・サーバー IP 名およびポート番号を知っておく必要があります。ネーミング・サーバー IP 名は、ドメイン・ネーム・サーバー (DNS) またはワークステーション HOSTS ファイルのいずれかに設定されます。デフォルトのポート番号は 900 です。

管理アプリケーションを始動するには、以下のステップを実行します。

1. ワークステーションで「スタート」、「プログラム」、「IBM WebSphere for z/OS Administration」の順にクリックします。
2. ネーミング・サーバーの IP 名、ポート 900、ユーザー ID CBADMIN、およびパスワード cbadmin でダイアログを埋めます。「OK」をクリックします。初期化が完了したことを示すメッセージが表示されるのを待ちます。

メイン・ダイアログ・ウィンドウが表示されると、処理を完了したことが確認されます。

会話を開始するためのステップ

新規会話を始動するには、以下のステップを実行します。

1. 左マウス・ボタンで「会話 (Conversations)」フォルダーを選択します。次に右マウス・ボタンを使用して、「会話 (Conversations)」フォルダーをクリックしてから「追加 (Add)」を選択します。
2. プロパティ・フォーム (右側のパネル) で、新規会話に名前を入力します。
3. 保管 (ディスク) アイコンをクリックします。ツリーに「Adding... Conversation (会話の追加中)」というメッセージがひょうじされます。

新規会話が追加されたことを示すメッセージ **BBON0515I** がステータス・バー (ダイアログ・ウィンドウの下部にある) に表示されると、作業が完了していません。

J2SERV サーバーを追加するためのステップ

新規サーバーを追加するには、以下のステップを実行します。

1. 会話名の左のノードをクリックして、新規会話ツリーを展開します。
2. 「シスプレックス (Sysplexes)」を展開して、次に自分のシスプレックスを展開します。
3. 左マウス・ボタンで J2EE サーバー・フォルダーを選択します。次に、右マウス・ボタンを使用して J2EE サーバー・フォルダーをクリックしたあと、「追加 (Add)」を選択します。
4. プロパティ・フォームで、初期化のために、適宜、値を入力するか、選択を行います。

通常、ほとんどのプロパティにデフォルト値を使用できます。ただし、少なくとも以下の表に示されたプロパティは、必ず確認してください。サーバー・プロパティの完全なリストおよび説明については、管理アプリケーションにより使用可能なヘルプを使用するか、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

サーバー名	J2SERV
制御領域 ID	制御領域が稼働するユーザー ID。このユーザー ID は、RACF STARTED クラスのエントティティーと一致し、制御領域に対して適切な RACF 許可を得ています。
サーバー領域 ID	サーバー領域が稼働するユーザー ID。このユーザー ID は、RACF STARTED クラスのエントティティーと一致し、サーバー領域に対して適切な RACF 許可を得ています。
ローカル ID	非認証クライアントを許可する場合にのみ使用します。
リモート ID	非認証クライアントを許可する場合にのみ使用します。
制御領域開始プロシージャ名	J2SERV

アプリケーション・サーバーに対して以下のキー環境変数の値を提供します。
注: 必ず、ランタイム環境についてすべての**必須**環境変数を設定してください。アプリケーション・サーバーのランタイム変数とその値の全リストについては、127ページの『付録A. 環境ファイルおよび JVM プロパティ・ファイル』を参照してください。current.env ファイルをブラウザで現行値を検索することもできます。その場合、必要に応じて、既存の値をパネルに切り貼りして追加します。切り取り / コピーおよび貼り付けにはクイック・キー (COPY は [ctrl]+c、CUT は [ctrl]+x、PASTE は [ctrl]+v) を使用してください。これらの機能は、環境変数テーブル内でポップアップ・メニューから使用することはできません。

- **LIBPATH。** LIBPATH 変数は、階層ファイル・システム (HFS) で Java の DLL 検索パスを指定します。システム、WebSphere for z/OS、Java、および DB2 JDBC DLL を指定します。

例:

```
LIBPATH=/db2_install_path/lib
:/usr/lpp/java/IBM/J1.3/bin
:/usr/lpp/java/IBM/J1.3/bin/classic
:/usr/lpp/WebSphere/
```

ここで、db2_install_path は、DB2 for OS/390 をインストールした HFS です。

- **CLASSPATH。** CLASSPATH ステートメントは、Java アプリケーションがサーバー領域で使用する Java クラス・ファイル (JAR ファイルと classes.zip) を指定します。

CLASSPATH 変数に値がまだ含まれていない場合は、この会話でシスプレックスに対する値セットをコピーし、必要なファイルを付加します。アプリケーション・コンポーネントが DB2 データにアクセスする場合は、JDBC ドライバーの ZIP ファイルの絶対パスを追加します。

規則: CLASSPATH の内容全体が 1 行に収まる必要があります。

例:

```
CLASSPATH=:/usr/lpp/ldap/lib/ibmjndi.jar
:/db2_install_path/classes/db2j2classes.zip
```

注: この会話がアクティブになったあと、WebSphere for z/OS は、J2EE サーバーの CLASSPATH の前に次のファイルを自動的に付加します。

- ws390srt.jar
- waswebc.jar
- xerces.jar

5. 保管 (ディスクett) アイコンをクリックします。ツリー内に、「Adding... J2EE serverJ (2EE サーバーの追加中)」というメッセージが表示されます。

新規サーバー定義が追加されたことを示すメッセージ BBON0515I がステータス・バーに表示されると、作業が完了しています。

J2SERV1 サーバー・インスタンスを追加するためのステップ

サーバー・インスタンスを追加するには、以下のステップを実行します。

1. 必要であれば、フォルダー・アイコンの左のノードをクリックして、J2SERV フォルダーを展開します。
2. 左マウス・ボタンで「サーバー・インスタンス (Server Instances)」を選択します。次に右マウス・ボタンを使用して、「サーバー・インスタンス (Server Instances)」をクリックしてから「追加 (Add)」を選択します。
3. プロパティ・フォームで、サーバー・インスタンス名として J2SERV1 と入力します。
4. オプションで、サーバー・インスタンスの説明を入力します。
5. オプションで、ログ・ストリーム名を入力します。ログ・ストリーム名を提供しない場合は、デフォルトは、J2SERV サーバー用に選択したログ・ストリーム名になります。
6. 保管 (ディスケット) アイコンをクリックします。ツリーに「Adding... Server Instance (サーバー・インスタンスの追加中)」というメッセージが表示されます。

新規サーバー・インスタンスが追加されたことを示すメッセージ BBON0515I がステータス・バーに表示されると、作業は完了しています。

J2EE リソースを追加するためのステップ

以下のステップを実行して、J2EE リソースを追加します。

1. 左マウス・ボタンで「J2EE リソース (J2EE Resource)」を選択します。次に右マウス・ボタンを使用して、「追加 (Add)」を選択します。
2. プロパティ・フォームで、J2EE リソースに名前を入力します。

3. オプションで、J2EE リソース記述を入力します。

4. データ・ソース・タイプと記されたプロパティータを見つけて DB2 を選択します。

管理アプリケーションによって、DB2 データ・ソースに該当する情報が上記のフィールドに入れられます。

5. 保管 (ディスクett) アイコンをクリックします。ツリー内に、「Adding... J2EE resource (J2EE リソースの追加中)」というメッセージが表示されま
す。

J2EE リソースが追加されたことを示すメッセージ BBON0515I がステータス・バーに表示されると、作業は完了しています。

J2EE リソース・インスタンスを追加するためのステップ

以下のステップを実行して、データ・ソース・インスタンスを追加します。

1. リソース名の左側のノードをクリックし、新たに作成された J2EE リソースのツリーを必要に応じて展開します。

2. 左マウス・ボタンで J2EE リソース・インスタンスを選択します。次に、右マウス・ボタンを使用して J2EE リソース・インスタンスをクリックし、その後、「追加 (Add)」を選択します。

3. プロパティ・フォームで以下について適切な値を入力します。

- J2EE リソース・インスタンス名
- J2EE リソース・インスタンスの記述
- データベース名。DB2 for OS/390 ロケーション名を提供します。

4. 保管 (ディスクett) アイコンをクリックします。ツリー内に、「Adding... J2EE resource instance (J2EE リソース・インスタンスの追加中)」というメッセージが表示されます。

J2EE リソース・インスタンスが追加されたことを示すメッセージ BBON0515I がステータス・バーに表示されると、作業は完了しています。

J2EE アプリケーションをインストールするためのステップ

始める前に:

- z/OS または OS/390 上の FTP サーバーが稼働中であることを確認します。
- FTP サーバーが次のディレクトリーに書き込みアクセスできることを確認します。

```
targetdir/sysplex/temp/administrator_ID/PolicyIVP.ear
```

targetdir は WebSphere for z/OS のマウント・ポイント、*sysplex* はシスプレックスの名前、*administrator_ID* は管理者のユーザー ID (通常は CBADMIN) です。

以下のステップを実行し、管理アプリケーションを使用してユーザーのアプリケーション用の EAR ファイルをインストールします。

1. アプリケーションをインストールしようとする J2EE サーバーをツリーで選択します。

2. 「J2EE アプリケーションのインストール... (Install J2EE Application...)」を「選択 (Selected)」メニュー・バーから選択します。「J2EE アプリケーションのインストール (Install J2EE Application)」ダイアログ・ボックスが表示されます。

3. このダイアログ・ボックスで以下の値を入力します。
 - ユーザーの J2EE アプリケーションを含む EAR ファイルの完全修飾パス名
 - アプリケーションをインストールするシスプレックスに対する FTP サーバーの名前。この名前は、通常、56ページの『管理アプリケーションを始動するためのステップ』の説明に従って指定したサーバー IP 名です。
 - 「OK」をクリックします。

結果: 「参照およびリソース解決 (Reference and Resource Resolution)」ウィンドウが表示され、EAR ファイル内のアプリケーションの内容が表示されます。

4. 「参照およびリソース解決 (Reference and Resource Resolution)」ウィンドウにリストされた Enterprise bean ごとに bean 名をクリックすると、このウィンドウの右側にその bean の詳細が表示されます。次に、bean ごとに必要に応じて以下のステップを完了します。

- a. EJB タブをクリックし、次に、「デフォルト JNDI 名を設定 (Set Default JNDI Name)」ボタンをクリックします。
- b. 「参照 (Reference)」タブをクリックし、当該 bean が参照する bean をリストします。ラベル「JNDI 名 (JNDI Name)」の下の ↓ 記号をクリックし、参照される bean ごとに選択可能な JNDI 名のリストを表示し、次に、適切な JNDI 名を選択します。

「参照 (Reference)」リスト内の bean ごとにこのステップを反復します。

- c. 「参照 (Reference)」タブをクリックし、当該 bean のデータ・ソースを表示します。ラベル「データ・ソース (Datasource)」の下の ↓ 記号をクリックし、データ・ソースについて選択可能な JNDI 名のリストを表示し、次に、適切な JNDI 名を選択します。この名前は、通常、`db2os390:ssn` であり、この `ssn` は、データ・ソース・インスタンスを追加するときに指定した DB2 for OS/390 サブシステム名です。

ヒント: 「参照およびリソース解決 (Reference and Resource Resolution)」ウィンドウのデータは、`application_name_resolved.ear` という EAR ファイルの新規コピーに保管されてから、サーバーに転送されて配置されます。ファイルのそのコピーをあとで再オープンした場合、その情報をもう一度入力する必要はありません。

-
5. 残りの bean すべてについて JNDI 選択処理を反復します。bean 名の左側の bean 記号にチェックマークが付き、bean ごとにこの処理が終了したことが分かります。

-
6. 「参照およびリソース解決 (Reference and Resource Resolution)」ウィンドウにリストされたサーブレットごとにサーブレット名をクリックすると、このウィンドウの右側にそのサーブレットの詳細が表示されます。次に、サーブレットごとに必要に応じて以下のステップを完了します。
 - a. 参照されるコンポーネントにデフォルトの JNDI 名を設定します。
 - b. サーブレットが必要とする J2EE リソースに JNDI 名を設定します。

-
7. 残りのサーブレットすべてについて JNDI 選択処理を反復します。サーブレット名の左側の記号にチェックマークが付き、サーブレットごとにこの処理が終了したことが分かります。
-

8. すべてのアプリケーション・コンポーネントについて JNDI 選択処理が完了すると、「OK」ボタンが選択可能になります。「OK」をクリックします。

結果: このアクションにより、ユーザーのワークステーションから z/OS または OS/390 への EAR ファイルの内容の自動 FTP 転送が開始されます。「Deploying... *application_name* (*application_name* の展開中)」というメッセージが画面に表示されます。FTP 転送は、以下の段階をたどって進行します。

段階	説明
1	<p>ear ファイルがインポートされると、下記にこのファイルが転送されます。</p> <p><i>targetdir/sysplex/temp/administrator_ID/application_name.ear</i></p> <p><i>targetdir</i> はマウント・ポイント、<i>sysplex</i> はシスプレックスの名前、<i>administrator_ID</i> は管理者のユーザー ID (通常は CBADMIN) です。</p>
2	<p>ear ファイルは下記にコピーされます。</p> <p><i>targetdir/apps/J2SERV/Ln/application_name.ear</i></p> <p><i>n</i> はレベル番号です。</p>
3	<p>ear ファイルが処理されます。ear ファイルの処理中に、次のディレクトリーに ear ファイルが展開されます。</p> <p><i>targetdir/apps/J2SERV/Ln/app_name/</i></p> <p><i>app_name</i> はアプリケーションの名前です (必ずしも ear ファイル名と同じではありません)。</p>
4	<p>次の枠組的ディレクトリーが作成されます。</p> <p><i>targetdir/apps/J2SERV/Ln/A/</i></p> <p>このディレクトリーの下に、配置情報のすべてが保管されます。</p>
<p>注: 会話がアクティブになったとき、</p> <p><i>targetdir/apps/J2SERV/Ln/</i></p> <p>このディレクトリーの下にあるすべてのものは、1 つ上のレベルの下記へ移動します。</p> <p><i>targetdir/apps/J2SERV/</i></p>	

また、ユーザーのアプリケーションに対するこの配置処理中に、以下のことが行われます。

- 適切な所有権とファイル許可が、ユーザーのアプリケーション・ファイルに設定されます。

- ユーザーの J2EE アプリケーションの JAR ファイルと WAR ファイルが、当該 J2EE サーバーの CLASSPATH 変数に追加されます。
- クライアントが要求したときに HTTP プロトコルを介して WAR ファイルと EJB JAR ファイルにアクセスできるようにする完全修飾 URI で、Web アプリケーションが提供されます。

`application_name_resolved.ear` ファイルが正常にインストールされたことを示すメッセージ BBON0470I がステータス・バーに表示され、作業が完了したことが分かります。

新規会話モデルの妥当性検査のためのステップ

会話の妥当性検査を行うには、以下のステップを実行します。

1. 必要に応じて、定義した会話までツリーをスクロールアップします。
-
2. 左マウス・ボタンで、会話を選択します。次に右マウス・ボタンを使用して、「EJBDBConfig 会話 (EJBDBConfig conversation)」をクリックしてから「検査 (Validate)」を選択します。

新規会話が有効であることを示す、メッセージ BBON0442I がステータス・バーに表示されると、処理を完了したことが確認されます。

会話をコミットするためのステップ

会話をコミットするには、以下のステップを実行します。

1. 必要に応じて、妥当性検査を行った会話までツリーをスクロールアップします。
-
2. 左マウス・ボタンで、会話を選択します。次に右マウス・ボタンを使用して、「EJBDBConfig 会話 (EJBDBConfig conversation)」をクリックしてから「コミット (Commit)」を選択します。「Do you still want to commit? (本当にコミットしますか?)」という質問に、「はい (Yes)」と答えます。「Committing... `conversation_name (conversation_name のコミット中)`」というメッセージがツリーに表示されます。
-

新規会話および J2EE サーバー定義がコミットされたことを示す、メッセージ BBON0444I がステータス・バーに表示されると、処理を完了したことが確認されます。

z/OS または OS/390 タスクの完了のマーク付けをするためのステップ

1. 左マウス・ボタンで新規会話を選択します。次に右マウス・ボタンでその会話をクリックした後、「指示 (Instructions)」を選択します。

2. 管理アプリケーションによって提供される説明をダブルクリックし、ワークロードの定義やセキュリティーの設定などの必須の z/OS または OS/390 タスクすべてを完了したかどうかを判断します。これらの必須の z/OS または OS/390 タスクのチェックリストは、50ページの『OS/390 の手動作業を完了するためのステップ』に記載されています。

3. z/OS または OS/390 タスクを検証したかまたは終了したとき、以下のステップに従って管理アプリケーションのすべてのタスクに完了のマークを付けてください。
 - a. 左マウス・ボタンで、会話を選択します。次に、右マウス・ボタンでその会話をクリックして「完了 (Complete)」を選択した後、「全タスク (All tasks)」を選択します。
 - b. 「Are you sure that all tasks have been completed? (すべてのタスクが完了したかどうか確認しますか。)」という質問には、「はい (Yes)」と答えます。

すべてのタスクが完了したことを示す、メッセージ BBON0484I がステータス・バーに表示されると、処理を完了したことが確認されます。

サーバー構成を活動化するためのステップ

1. 左マウス・ボタンで、会話を選択します。次に、右マウス・ボタンで会話をクリックした後、「活動化 (Activate)」を選択します。

2. 「Do you want to activate conversation *conversation_name*? (会話 *conversation_name* を活動化しますか。)」という質問に、「はい (Yes)」と答えます。ダイアログの下部に表示されるメッセージが、いつサーバー定義が活動化したのかを示します。

新規会話が活動化されたことを示す、メッセージ BBON0449I がステータス・バーに表示されると、処理を完了したことが確認されます。以上で、活動化した J2EE サーバーは、インストールしたアプリケーションのホストになる準備ができました。

第8章 J2EE アプリケーション・クライアントの作成と実行

アプリケーション・コンポーネントを J2EE サーバーにインストールしたあとでは、そのコンポーネントを使用する J2EE アプリケーション・クライアントを作成する準備ができています。6ページの図2 には、J2EE サーバーで稼働するコンポーネントについて最も可能性の高いタイプのクライアントが示されています。この章の情報は、主に z/OS または OS/390 クライアントに該当する情報ですが、非 z/OS または非 OS/390 クライアントの開発と実行に関する詳細な情報源についても言及しています。

クライアント・アプリケーションのタイプ、およびその作成と実行に関する情報が見つかる場所を、次の表に示します。どのタイプのクライアントについても、必ず、74ページの『J2EE アプリケーションのセキュリティー上の考慮事項』の情報を読んでください。

クライアントのタイプ	情報源
z/OS または OS/390 上の WebSphere Application Server スタンド版 で稼働するクライアント	『z/OS または OS/390 クライアント用の WebSphere Application Server スタンド版』
z/OS または OS/390 で稼働するクライアント	71ページの『ネイティブ z/OS または OS/390 Java クライアント』
分散プラットフォーム上の WebSphere Application Server で稼働するクライアント	72ページの『非 z/OS および非 OS/390 プラットフォーム上の WebSphere Application Server アドバンスド版およびスタンド版クライアント』

z/OS または OS/390 クライアント用の WebSphere Application Server スタンド版

7ページの図3 には、WebSphere Application Server スタンド版 の Web アプリケーションについて可能な構成が示されています。以下の情報が該当するのは、Web アプリケーションが スタンド版 環境で稼働し、WebSphere for z/OS J2EE サーバーで実行される Enterprise bean を駆動する場合だけです。

このタイプの J2EE アプリケーション・クライアントの開発と稼働に関する資料は *WebSphere Application Server for OS/390* スタンド版 計画、インストール

ールおよび使用, GC88-8970 で得ることができるので、以下の手順では、開発およびインストール過程の要約だけを提供し、より詳細な説明がある情報源について言及しています。 **始める前に:** WebSphere Application Server スタンダード版 バージョン 4.0 for OS/390 をインストールし、カスタマイズしてあることが必要です。この作業は、一般に、システム・プログラマーによって行われるので、この作業をご使用のシステムで完了するために、このような専門書の助けが必要となる場合があります。必要に応じて、以下の情報源のいずれかまたは両方を参照してください。

- *WebSphere Application Server for OS/390* バージョン 4.0 スタンダード版 プログラム・ディレクトリー。これは、スタンダード版の製品と同梱になっています。
- *WebSphere Application Server for OS/390* スタンダード版 計画、インストールおよび使用, GC88-8970

以下のステップを実行して、J2EE アプリケーション・クライアントを作成し、実行してください。

1. 適切なアプリケーション開発ソフトウェアをすでにインストールし、そのツールの関連資料を手元に用意してあることを確認します。詳細については、21ページの『第4章 アプリケーション開発環境の設定』を参照してください。

推奨:

- VisualAge for Java または WebSphere Studio を使用し、サーブレットと JSP を開発してテストしてください。VisualAge for Java WebSphere Test Environment フィーチャーを使用すると、WebSphere Application Server 環境をワークステーションにインストールしなくても、アプリケーション・クライアントのテストを行うことができます。
- アプリケーション開発ツールの追加的考慮事項について、必ず、次の資料の最新版を調べてください。*WebSphere Application Server for OS/390* スタンダード版 計画、インストールおよび使用, GC88-8970

-
2. J2EE アプリケーションのサーブレットと JSP コンポーネントをコード化し、テストしてください。

規則: J2EE アプリケーション・クライアントは、Java プロパティー・キー `java.naming.factory.initial` を `com.ibm.ws.naming.WsnInitialContextFactory` に設定する必要があります。

推奨:

- クライアント・アプリケーション自体でプロパティ値をハードコーディングする代わりに、WebSphere Application Server の `was.conf` ファイルによってプロパティを設定してください。70ページの6 ステップを参照してください。
- アプリケーション・クライアントの設計に影響することがあるプログラミング・モデルの制約事項について、必ず、*WebSphere Application Server for OS/390* スタンダード版 計画、インストールおよび使用 の最新版を調べてください。

-
3. これらのコンポーネントの単体テストの結果に納得した場合は、コンポーネントの `ART` ファイル (JAR ファイルなど) を、OS/390 上の階層ファイル・システム (HFS) の作業ディレクトリーに転送してください。

推奨: 作業ディレクトリーについて次の命名規則を適用してください。

```
/webapp/servlet_or_JSP_name/
```

注: ワークステーション (ASCII ベースのシステム) から OS/390 (EBCDIC ベースのシステム) にファイルを転送するとき、ファイル転送作業の一環として必要な変換を行う必要があります。

-
4. HTTP サーバー構成ファイル (`httpd.conf` と呼ばれる) の中の WebSphere Application Server ディレクティブ・セクションに、`webapp/servlet_or_JSP_name` 相対ディレクトリーに対する次の Service ディレクティブを追加してください。

```
Service /webapp/servlet_or_JSP_name/*  
    /usr/lpp/WebSphere/AppServer/bin/was302plugin.so:service_exit
```

`httpd.conf` ファイルは、通常、`/etc` ディレクトリー内にあります。

-
5. 次に示すように、`was.conf` ファイル内の設定値を変更して WebSphere Application Server を構成してください。

```
appserver.libpath=/usr/lpp/WebSphere390/CB390/lib  
appserver.  
classpath=path/ws390crt.jar
```

ここで、`path` は、JAR ファイルがあるディレクトリーです。
`/usr/lpp/WebSphere390/CB390/lib/` はデフォルト・パスです。

was.conf ファイルは、通常、/usr/lpp/WebSphere/AppServer/properties ディレクトリー内にあります。このファイルによって設定できるすべてのアプリケーション・サーバー・プロパティーについては、*WebSphere Application Server for OS/390* スタンダード版 計画、インストールおよび使用 で説明しています。

6. Web アプリケーション・プロパティーを was.conf ファイル内に設定することによって、J2EE アプリケーション・クライアントを WebSphere Application Server に対して定義してください。

規則: 以下のプロパティー設定値は、サーブレットと JSP の両方にとって必須です。

- `deployedwebapp.<webapp_name>.classpath=`
/usr/lpp/WebSphere390/CB390/wsz0Sclt.jar およびクライアントが使用する Enterprise bean に対する JAR ファイルの場所と名前
- `deployedwebapp.<webapp_name>.java.naming.factory.initial=`
com.ibm.ws.naming.WsnInitialContextFactory
- `deployedwebapp.<webapp_name>.javax.naming.provider.url=`
"iiop://x.x.x.x:ppp。この x.x.x.x:ppp は、WebSphere for z/OS システム管理サーバーの IP アドレスおよびポートです。

推奨: 追加 Web アプリケーション・プロパティーを was.conf ファイルによって設定する必要があるかどうかを判断するために、必ず、*WebSphere Application Server for OS/390* スタンダード版 計画、インストールおよび使用 を調べてください。

7. HTTP サーバーを停止してから再始動してください。
-

8. `http://<host>/webapp/<webapp_name>` の Web サイトに場所を設定し、J2EE アプリケーション・クライアントに Web ブラウザーからアクセスしてください。ここで、<host> は、アプリケーション・クライアントのホスト・システムです。
-

ネイティブ z/OS または OS/390 Java クライアント

ご使用のシステムのシステム・プログラマーが WebSphere for z/OS をインストールしてカスタマイズしたとき、システム・プログラマーは、正しくインストールされたことを検証するために、サンプル J2EE アプリケーション (製品の一部として提供されている) を実行しました。同じアプリケーションを使用すると、z/OS または OS/390 上の J2EE アプリケーションを z/OS または OS/390 で稼働中の Java クライアントから駆動する方法を知ることができます。 *WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 には、WebSphere for z/OS インストール検査プログラム (IVP) を実行する手順が記載されています。UNIX システム・サービス (USS) 環境で実行される Java プロセスを含む場合があるユーザー独自の z/OS または OS/390 クライアントを稼働するためのモデルとして、この手順および関連ファイルを使用してください。 `ejbivp.sh` ファイルの内容と命令には、特に注意を払ってください。このファイルは、Policy Enterprise bean IVP を駆動するためのサンプル `PolicyClient` を実行する USS シェル・スクリプトです。このシェル・スクリプトは、WebSphere for z/OS がインストールされる HFS の位置の `/samples/PolicyIVP/ejb` サブディレクトリーで使用可能です。

z/OS または OS/390 で稼働する Java クライアントを設計し、コード化する方法については、まず、*z/OS UNIX システム・サービス ユーザーズ・ガイド*, SA88-8640 および以下の規則とガイドラインを参照してください。

• 規則:

- クライアントは、その CLASSPATH に次のものを含む必要があります。`/usr/lpp/WebSphere390/CB390/ws390crt.jar` およびクライアントが使用する Enterprise bean に対する JAR ファイルの位置と名前
- WebSphere for z/OS はクライアント・コンテナーを提供しないので、クライアントは、Enterprise bean およびホームに対して明示的な名前を使用する必要があります。つまり、クライアントは、`java:comp` 名に JNDI ルックアップを使用することはできません。
- WebSphere for z/OS ネーミング・サービスを利用するために、クライアントは、次のプロパティーを設定する必要があります。

```
javax.naming.provider.url="iiop://x.x.x.x:ppp"
```

この `x.x.x.x:ppp` は、WebSphere for z/OS システム管理サーバーの IP アドレスおよびポートです。

- **ガイドライン:** 必ずセキュリティー要件を考慮してください。z/OS または OS/390 クライアントの場合、このクライアントを実動システムで稼働している場合でも、ご使用のシステムにセキュリティー・メカニズムが必要でない場合もあります。

非 z/OS および非 OS/390 プラットフォーム上の WebSphere Application Server アドバンスド版およびスタンダード版クライアント

ご使用のシステムのシステム・プログラマーが WebSphere for z/OS をインストールしてカスタマイズしたとき、システム・プログラマーは、正しくインストールされたことを検証するために、サンプル J2EE アプリケーション (製品の一部分として提供されている) を実行しました。同じアプリケーションを使用すると、z/OS または OS/390 上の J2EE アプリケーションを Windows NT などのリモート・プラットフォームから駆動する方法を知ることができます。次の手順では、サンプル・アプリケーションを駆動するために実行する必要のあるステップを示します。WebSphere for z/OS J2EE サーバーにインストールされた Enterprise bean を駆動するためにユーザー独自のクライアントを Windows 上でセットアップするためのモデルとして、この手順を使用してください。

規則:

- クライアントは、その CLASSPATH に次のものを含む必要があります。
/usr/lpp/WebSphere390/CB390/wsZOSclt.jar およびクライアントが使用する Enterprise bean に対する JAR ファイルの位置と名前
- クライアントは、Enterprise bean およびホームに対して明示的な名前を使用する必要があります。つまり、クライアントは、java.comp 名に JNDI ルックアップを使用することはできません。
- WebSphere for z/OS ネーミング・サービスを利用するために、クライアントは、次のプロパティを設定する必要があります。

```
javax.naming.provider.url="iiop://x.x.x.x:ppp"
```

この *x.x.x.x:ppp* は、WebSphere for z/OS システム管理サーバーの IP アドレスおよびポートです。

ガイドライン: 必ずセキュリティ要件を考慮してください。z/OS または OS/390 クライアントの場合、このクライアントを実動システムで稼働している場合でも、ご使用のシステムにセキュリティ・メカニズムが必要でない場合もあります。考慮事項を検討し、セキュリティ管理者と相談してください。

J2EE クライアント・アプリケーションを z/OS and OS/390 以外のプラットフォームで稼働するために設計し、コード化する方法の詳細については、使用している WebSphere Application Server 製品に関する **インフォセンター** の情報を参照してください。インフォセンターは、<http://www.ibm.com/jp/software/websphere/appserv/> で利用できます。

Enterprise bean に非 z/OS または非 OS/390 プラットフォームからアクセスするためのステップ

以下の手順では、Enterprise bean インストール検査プログラム (IVP) (WebSphere for z/OS で提供される) を Windows にインストールされたクライアント・アプリケーション (PolicyClient) から実行する手順を示します。WebSphere for z/OS J2EE サーバーにインストールされた Enterprise bean を駆動するためにユーザー独自のクライアントを Windows 上でセットアップするためのモデルとして、この手順を使用してください。

始める前に: 以下のことを行う必要があります。

- Policy IVP ファイルがすでにインストールされている WebSphere for z/OS J2EE サーバーにアクセスできることを確認するか、または *WebSphere Application Server V4.0 for z/OS and OS/390*: インストールおよびカスタマイズ, GA88-8652 の説明に従って、J2EE サーバーを定義し、PolicyIVP.ear ファイルをインストールしてください。
- WebSphere for z/OS J2EE サーバーがアクティブになっていることを確認してください。

以下のステップを実行して PolicyClient アプリケーションを Windows 上で稼働し、Enterprise bean IVP を駆動します。

1. ワークステーションの作業ディレクトリーに WebSphere for z/OS でインストールされたサンプルから PolicyClient.zip ファイルをダウンロードします。

WebSphere for z/OS のサンプルに対するデフォルトの HFS の位置は、`/usr/lpp/WebSphere/samples/` です。

2. PolicyClient.zip ファイルを解凍します。

結果: 以下のファイルが示されます。

ファイル名	記述
ejbivp.bat	クライアントを実行する Windows バッチ・ファイル
ejbivp シェル・スクリプト	クライアントを実行する Korn シェル・スクリプト
PolicyClient.jar	クライアント Java メイン
policysession_deploy.jar	Policy Enterprise bean に対する RMI スタブ

ファイル名	記述
Policy.Property.asc	クライアント Java メインで使用されるデータ・ファイル

-
3. WebSphere for z/OS ホスト IP アドレス:set ws390Server_IP=x.x.x.x を指定するように、ejbivp.bat ファイル内の行を編集します。

注: Enterprise bean を WebSphere for z/OS 上で駆動するための Windows クライアント・アプリケーションを稼働するモデル・バッチ・ファイルとして、ejbivp.bat ファイルの内容を使用できます。

-
4. Windows コマンド・プロンプトから ejbivp.bat を実行します。
-

IVP が正常終了したことを示すメッセージがコマンド・プロンプト・ウィンドウに表示されます。

J2EE アプリケーションのセキュリティー上の考慮事項

クライアントのセキュリティー・プロファイルおよび許可は、テスト・システムまたは実動システムに対するご使用のシステムのガイドラインによって、ある程度異なったものとなります。たとえば、テスト環境では、セキュリティー機構を何も使用しないで、J2EE アプリケーション・クライアントがテスト・システムおよびデータにアクセスすることを認める場合があります。クライアント・プログラムが J2EE サーバーと同じ z/OS または OS/390 システムで実行されるとき、この方法が特に適していることがあります。しかし、ご使用のシステムでクライアントの認証を必要とする場合は、セキュリティー要件を満たすためにご使用のシステムのセキュリティー管理者と連携することが必要な場合があります。

ガイドライン:

- J2EE サーバーおよび J2EE アプリケーション・クライアントに対するセキュリティーをセットアップする方法の推奨事項と説明については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ* を参照してください。この文書のセキュリティーのトピックでは、以下のことを取り上げています。
 - クライアント認証

- J2EE サーバーへのクライアント・アクセス
- サーバー内のオブジェクトへのクライアント・アクセス
- DB2 へのクライアント・アクセス (オプション)
- 役割に基づいたセキュリティーで保護される Enterprise bean またはサーブレット・メソッドを使用するために、クライアント・アプリケーションがアクセス権を必要とする場合、プロファイルを EJBROLE または GEJBROLE クラスに定義する必要がある、そのあと、ユーザーまたはグループにそのプロファイルへの読み取りアクセスを許可する必要があります。

規則:

- EJBROLE または GEJBROLE クラスに指定するプロファイルは、次のフォーマットに従います。

role_name

この *role_name* は、次のいずれかに指定したセキュリティー役割属性と一致します。

- J2EE アプリケーション配置記述子
- 個々のアプリケーション・コンポーネントの配置記述子
- 役割名には空白を入れることはできません。また、役割名は、246 文字を超えることができません。ただし、役割名は大文字小文字混合にすることができます。

ご使用のシステムで z/OS または OS/390 SecureWay Security Server (RACF) を使用する場合、次のコマンドの使用方法について *z/OS SecureWay Security Server (RACF) コマンド言語 解説書, SA88-8617* を参照してください。

- プロファイルを EJBROLE または GEJBROLE クラスに定義する RDEFINE コマンド
- このプロファイルへの読み取りアクセスをユーザーに認可する PERMIT コマンド

第3部 ランタイム環境での J2EE アプリケーションに関する作業

第9章 WebSphere for z/OS サーバーへのアプリケーションのインストール

WebSphere for z/OS 管理アプリケーションを使用して段階的にインストールするほかに、次の方法によってアプリケーションを WebSphere for z/OS にインストールすることもできます。

インストール方法	参照先
管理アプリケーションのエクスポート / インポート機能を使用する方法。	『管理アプリケーションを介してエクスポート / インポート・プロセスを使用するためのステップ』
システム管理スクリプト API を使用する方法。	81ページの『スクリプトによるアプリケーションのインストール』

管理アプリケーションを介してエクスポート / インポート・プロセスを使用するためのステップ

J2EE アプリケーションのテストを終了したあと、WebSphere for z/OS 管理アプリケーションを使用して、テスト・システムで使用した J2EE サーバー構成をエクスポートし、実動システムにそのモデル構成をインポートできます。このエクスポート・プロセスおよびインポート・プロセスを介して、サーバー定義を含む HFS ファイルを構成し、実動システムに転送します。このプロセスは、スクラッチからのサーバー構成の定義より高速で、エラーが発生しにくくなります。

エクスポート / インポート・プロセスを使用するには、以下のステップを実行してください。

1. 管理アプリケーションで、ユーザーのアプリケーションが配置される J2EE サーバーのサーバー・モデルをエクスポートします。
 - a. アクティブ・イメージでサーバーを選択する。
 - b. 「選択 (Selected)」メニュー・バー選択項目の「サーバーのエクスポート ... (export server...)」アクションを選択する。「サーバーのエクスポート (Export server)」ダイアログ・ボックスが表示されます。
 - c. エクスポート・プロセスの出力を含める HFS ファイルの完全修飾名を入力する。

d. 「**OK**」をクリックする。

結果: アクション「サーバーのエクスポート... (Export server...)」によって、ホストのサーバー用の HFS ファイルが作成されます。この HFS ファイルには、サーバーのプロパティのほとんどすべてによるサーバーとそのサブツリーの定義、参照されるが定義されていない J2EE リソースが含まれます。

-
2. サーバーを稼働させたい z/OS または OS/390 実動システムに出力 HFS ファイルをコピーまたは移動します。ファイルを移動またはコピーするための方法と説明については、*z/OS UNIX システム・サービス ユーザーズ・ガイド*、SA88-8640 を参照してください。

警告: 出力 HFS ファイルは編集しないでください。

-
3. 管理アプリケーションで、J2EE サーバー・モデルを以下のステップに従ってインポートします。
- 必要に応じて、会話を追加する。
 - 「**サーバー (Servers)**」フォルダーを選択する。
 - 「選択 (Selected)」メニュー・バー選択項目の「**サーバーのインポート ... (import server...)**」アクションを選択する。「**サーバーのインポート (Import server)**」ダイアログ・ボックスが表示されます。
 - 「**サーバー名 (Server name)**」に、この WebSphere for z/OS 構成に対する固有の名前を入力する。
 - 「**入力ファイル (Input file)**」に、実動システムに移動またはコピーした HFS ファイルの完全修飾名を入力する。
 - 「**OK**」をクリックする。
 - 「**制御領域 proc 名 (Control region proc name)**」および「**デバッガー許可 (Debugger allowed)**」を含め、サーバーのプロパティを変更する。
 - 実動システムに対するサーバー・インスタンスを適宜追加する。
 - 実動システムに対する J2EE リソース・インスタンスを適宜追加する。
-
4. さらに、管理アプリケーションで、以下のことを実行します。
- 会話を選択してから「**検査 (Validate)**」を選択して、インポートされたモデルの妥当性検査を行う。ステータス・バーにメッセージ BBON0442I が表示されると、新規会話は有効です。

- b. 会話を選択してから「**コミット (Commit)**」を選択して、会話をコミットする。「Do you still want to commit? (本当にコミットしますか。)」という質問に、「はい (Yes)」と答えます。ステータス・バーにメッセージ BBON0444I が表示されると、新規会話はコミットされています。
 - c. OS/390 作業を適宜完了する。
 - d. 会話を選択してから「**活動化 (Activate)**」を選択して、会話を活動化する。「Do you want to activate conversation? (会話を活動化しますか。)」という質問に、「はい (Yes)」と答えます。サーバー定義が活動化されると、ダイアログの下部にメッセージが表示されます。
-

スクリプトによるアプリケーションのインストール

WebSphere for z/OS 管理アプリケーションを使用しないでアプリケーションを J2EE サーバーにインストールするために、システム管理スクリプト API を使用できます。これによって、管理アプリケーションとまったく同じ機能が提供されます。スクリプトを使用することにより、実動サーバーへのアプリケーションのインストールなど、より高速でエラーの可能性が低いメソッドが提供される場合があります。システム管理スクリプト API の使用方法の詳細については、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理スクリプト API*, SA88-8657 を参照してください。

第10章 J2EE アプリケーションの活動に関するデータの収集

WebSphere for z/OS には、J2EE サーバーで稼働中のアプリケーションについて情報を収集する方法が数通りあります。

情報の種類	参照項目
SMF レコードを使用してアカウントिंग情報を収集する方法	『SMF レコードによる J2EE アプリケーション情報の収集』
JRas サポートを使用してアプリケーションがメッセージ発行と項目のトレースを使用可能にする方法	『Java アプリケーションのロギング・メッセージおよびトレース・データ』

SMF レコードによる J2EE アプリケーション情報の収集

サーバー・アプリケーションに関連した統計を収集して記録する場合、z/OS または OS/390 システム管理機能 (SMF) を使用するために J2EE サーバーを定義できます。SMF の活動およびインターバル・レコードによって、J2EE サーバーは、アプリケーションのプロファイル作成に使用するアプリケーション詳細を記録します。SMF の記録を可能にするには、SMF レコードを作成するための J2EE サーバーの定義、およびその他の管理作業を行う必要があります。詳細については、*WebSphere Application Server V4.0 for z/OS and OS/390: 操作および管理*, SA88-8653の SMF のトピックを参照してください。

Java アプリケーションのロギング・メッセージおよびトレース・データ

WebSphere for z/OS ランタイムは、Ras Toolkit for Java をサポートします。これを使用すると、WebSphere for z/OS J2EE または MOFW サーバーで稼働する Java サーバー・アプリケーションからメッセージを発行することができます。このアプリケーションのトレース・データを収集することができます。JRas サポートと呼ばれるこのツールキットに対する WebSphere for z/OS 拡張機能を介して、Java アプリケーションのメッセージを、MVS マスター・コンソール上で表示したり、WebSphere for z/OS のエラー・ログ・ストリームやコンポーネント・トレース (CTRACE) データ・セットの中書き込むことができます。また、ユーザーのアプリケーションのトレース項目も同じ CTRACE データ・セットに表示できます。

MVS マスター・コンソールにメッセージを発行して、主幹業務アプリケーションの重大なエラー状態を報告することができます。マスター・コンソールを介して、オペレーターはメッセージを受け取り、ユーザーのアプリケーションの状態を示すメッセージに応答して処置を講じます。さらに、メッセージをマスター・コンソールに送信することによって、自動化パッケージを起動して、ユーザーのアプリケーションの処理に関連した特定の条件またはイベントに対して処置を講じることができます。

ユーザーのアプリケーションからコンソールに対して発行されるすべてのメッセージは、WebSphere for z/OS の CTRACE データ・セットの中にも現れ、また、メッセージ・タイプによってはエラー・ログ・ストリームの中にも現れます。これらのシステム・リソース内のメッセージをログに記録すると、ユーザーのアプリケーションの処理に関連したエラーを容易に診断することができます。

同様に、CTRACE データ・セット内のログ・トレース・データに要求を発行することは、エラー状態の記録、すなわち診断目的でのアプリケーション・データの収集のもう 1 つのメソッドです。収集するトレース・データの量とタイプは、選択することができます。したがって、パフォーマンスが優先される場合には最少のトレースでアプリケーションを実行したり、あるいは、問題を再作成して追加の診断情報を収集する場合は詳細にトレースしてアプリケーションを実行することができます。

推奨: エラー・ログ・ストリームである WebSphere for z/OS の CTRACE データ・セットとマスター・コンソールの主な目的は、システム・コンポーネントと重要なアプリケーションの診断データをモニターするか記録することです。インストールの構成によって、アプリケーション・メッセージとデータをこれらのリソースに送信することは、システム・パフォーマンスに悪影響を及ぼす可能性があります。たとえば、アプリケーション・データを CTRACE データ・セットに送信すると、そのデータ・セット内のトレース項目がさらに迅速にラップされる可能性があります。これは、ラップが発生したときにシステムが既存の項目に新規のものを書き込むために、重要な診断データを失う可能性があります。このロギング・サポートの使用は慎重に行ってください。

注:

1. ロギング・メッセージとトレース・データの WebSphere for z/OS サポートを、Java アプリケーション (Java アプレットではなく) のみに使用することができます。
2. Ras Toolkit に対する WebSphere for z/OS のサポートは、エンタープライズ版 V3.02 で提供された JRas のサポートと同じではありません。新しい JRas のサポートについて、以下に説明します。

- ユーザーのアプリケーションが発行するメッセージを、常に、ログに記録します。この変更は、いったん、メッセージを発行するようにアプリケーションをコード化し、そのアプリケーションを実行すると、そのメッセージが常に収集されてログに記録されることを意味します。エンタープライズ版 V3.02 の場合は、メッセージ収集をオフにする機能がありました。
- トレース・データの収集を使用可能にするために、異なる機構が必要です。エンタープライズ版 V3.02 の場合は、MOFW アプリケーション・サーバーに対する環境変数によって、トレース・データの収集を制御しました。WebSphere for z/OS V4.0 の場合は、ユーザー提供のトレース設定ファイルによって、トレース・データの収集を使用可能または使用不可にすることができます。
- メッセージ・ロガーまたはトレース・ロガーを取得するためのクラスは異なるクラスを使用しますが、同じメソッド、つまり、`createRASTraceLogger` メソッドと `createRASMessageLogger` メソッドを使用します。ただし、WebSphere for z/OS V4.0 のメソッドのシグニチャーは、エンタープライズ版 V3.02 のものとは多少異なっています。
エンタープライズ版 V3.02 の `createRASTraceLogger` メソッドと `createRASMessageLogger` メソッドは使用すべきではありませんが、これらのメソッドを使用するためにコード化したどのプログラムも、変更する必要はありません。ただし、それらのプログラムを別のプラットフォーム上、および OS/390 上で実行しなければならない場合は、変更する必要があります。WebSphere for z/OS V4.0 の場合、`createRASTraceLogger` または `createRASMessageLogger` の呼び出しは、WebSphere for z/OS V4.0 の新しいクラス内の同じメソッドに委任されます。Windows NT のような別のプラットフォーム上でユーザーのアプリケーションを実行するには、新しいメソッドを使用するようプログラムを再コード化する必要があります。

メッセージを発行したりトレース項目をログに記録するためにサーバー・アプリケーションから発行できるメソッドの説明については、*Using the WebSphere J2EE Message Logging and Trace Facility* を参照してください。これには、z/OS または OS/390 などのサポートされているすべてのプラットフォームに適用される、`com.ibm.ras` パッケージ内のメソッドについての説明があります。

以下の表は、Java アプリケーション用のメッセージとトレース・データをログに記録するためのサブタスクおよび関連した手順を示しています。

サブタスク	関連した手順 (参照項目)
発行または収集するメッセージおよびトレース・データのタイプの決定	<ul style="list-style-type: none"> 『アプリケーション・メッセージを MVS マスター・コンソールに発行するときのバックグラウンド』 89ページの『ユーザーのアプリケーションに対してトレース要求を発行するときのバックグラウンド』
メッセージとトレース要求を発行するための Java サーバー・アプリケーションの作成	91ページの『Java アプリケーションをコード化してメッセージとトレース要求を発行するためのステップ』
メッセージをログに記録しトレース・データを収集するための z/OS または OS/390 ランタイム環境の作成	98ページの『Java アプリケーションのメッセージとトレース要求をログに記録するための z/OS または OS/390 環境を作成するステップ』
Java サーバー・アプリケーション用に収集されたメッセージまたはトレース・データの表示	<ul style="list-style-type: none"> 102ページの『メッセージとトレース・データを表示するときのバックグラウンド』 104ページの『IPCS をバッチ・モードで使用して、アプリケーション・トレース・データをフォーマットするためのステップ』

アプリケーション・メッセージを MVS マスター・コンソールに発行するときのバックグラウンド

Ras Toolkit に対する WebSphere for z/OS ランタイム・サポート (JRas サポート) によって、ユーザーの Java アプリケーションから MVS マスター・コンソールにメッセージを発行できます。MVS マスター・コンソールにメッセージを発行して、主幹業務アプリケーションの重大なエラー状態を報告したり、自動化パッケージを起動することができます。ユーザーのアプリケーションが発行するメッセージも、WebSphere for z/OS で使用するコンポーネント・トレース (CTRACE) データ・セットの中に表示され、メッセージがエラー・メッセージとして分類された場合には、そのエラー・ログ・ストリームの中に表示されます。メッセージをログに記録することは、エラー状態の記録、すなわち、診断目的でのアプリケーション・データの収集のもう 1 つのメソッドです。

WebSphere for z/OS は、メッセージ・ロガーを作成し管理するコードを提供します。メッセージ・ロガーは、アプリケーションのメッセージを処理します。

メッセージ・ロガーは、ユーザーの Java アプリケーションが稼動する WebSphere for z/OS J2EE サーバーまたは MOFW サーバーの Java 仮想マシン (JVM) で稼動します。メッセージ・ロガーを使用するには、Java アプリケーションで以下のことを行う必要があります。

1. メッセージ・ロガーを定義する。
2. メッセージ・ロガーを作成するように WebSphere for z/OS に指示するメソッドを駆動する。
3. アプリケーションの適切なポイントでメッセージをコード化する。

アプリケーションを更新して JRAS サポートを使用するための特別な指示は、91ページの『Java アプリケーションをコード化してメッセージとトレース要求を発行するためのステップ』にあります。ただし、これらの指示を使用してメッセージを適切にコード化できるようになるには、以下のトピックにある概念について理解しておく必要があります。

- 『インライン・メソッドの呼び出しまたはメッセージ・プロパティー・ファイルによるメッセージの定義』
- 88ページの『メッセージ・タイプがメッセージ宛先に及ぼす影響について』

インライン・メソッドの呼び出しまたはメッセージ・プロパティー・ファイルによるメッセージの定義

ユーザーの Java アプリケーションからメッセージを発行する場合は、メッセージをインラインで定義するか、あるいは別々のファイルを使用してメッセージを含めることができます。一般的には、メッセージをインラインで定義する方が速く、完了に必要なステップは少なくなります。さまざまな言語でメッセージ・テキストを提供する予定の場合は、使用可能度とテキスト変換のためには、別々のメッセージ・プロパティー・ファイルを使用する方が適しています。メッセージの定義にファイルを使用するかあるいはインライン方法を使用するかに関係なく、Java アプリケーション内のメソッドをコード化して、処理中の適切なポイントでメッセージを発行しなければなりません。適切なポイントで、RASIMessageLogger インターフェースで定義したメソッドを使用して、メッセージを発行します。

メッセージをインラインで定義する場合は、`textMessage` メソッドを使用して、ユーザーのアプリケーションからメッセージを発行します。メソッドの呼び出しで指定する文字列は、メッセージ・ロガーがマスター・コンソール、エラー・ログ・ストリーム、または CTRACE データ・セットに送信する内容です。

メッセージ・プロパティー・ファイルを使用する予定の場合は、以下のことを行う必要があります。

1. メッセージ・プロパティ・ファイルを作成する。
2. キーとテキストの組みを使用してすべてのメッセージを定義する。
キーを使用すると、メッセージ・ロガーは適切なメッセージをメッセージ・ファイルで見つけることができます。テキストは、メッセージ・ロガーがマスター・コンソール、エラー・ログ・ストリーム、または CTRACE データ・セットに送信する内容です。
3. 適切なメソッドを使用して、アプリケーションのメッセージ用のメッセージ・テキストがある場所をメッセージ・ロガーに知らせます。
以下の 2 つの機構によって、メッセージ・ロガーのためのメッセージ・ファイルを識別することができます。
 - `setMessageFile` メソッド。このメソッドで、メッセージ・テキストを検索するためのデフォルト・ファイルとして機能する 1 つのメッセージ・プロパティ・ファイルを登録します。
 - `message` メソッドまたは `msg` メソッド。これらのメソッドでは、メッセージ・プロパティ・ファイルの名前を指定することができます。

メッセージ・ファイルを作成するための特別な指示、このファイルでメッセージを定義するための規則、および例については、91ページの『Java アプリケーションをコード化してメッセージとトレース要求を発行するためのステップ』を参照してください。

メッセージ・タイプがメッセージ宛先に及ぼす影響について

メッセージを発行するためのメソッドをコード化するときに、メッセージにタイプを割り当てて、エラー・メッセージ、警告メッセージ、または情報メッセージとしてメッセージの特性を示します。RASIMessageEvent インターフェースは、メッセージのタイプを定義します。3 タイプのメッセージすべてがマスター・コンソールと CTRACE データ・セットの両方に送信されます。エラー・メッセージは、エラー・ログ・ストリームにも表示されます。

注: メッセージは、常にログに記録されます。つまり、いったん、メッセージを出すようにアプリケーションをコード化し、そのアプリケーションを z/OS または OS/390 上で実行すると、そのメッセージは、常に収集されてログに記録されます。

処理時に、メッセージ・ロガーは、アプリケーションが発行する各メッセージに記述子コードと宛先コードを割り当てます。ロガーは、以下の表に示すように、メッセージ・タイプを基にして特定のコードを割り当てます。

ガイドライン: ユーザーのアプリケーション用にメッセージを開発する場合、メッセージの内容とタイプは、以下の表の記述子と宛先コードの説明に従いま

す。各メッセージには 1 つのメッセージ・タイプだけを割り当てます。また、インストール時に、マスター・コンソールに送信されるメッセージのガイドラインを定義しているかどうか調べてください。

RASIMessageEvent メッセージ・タイプ	記述子コード	宛先コード
TYPE_ERROR (または TYPE_ERR)	12: 重要な情報 メッセージには、コンソールには表示しなければならないが、応答のアクションを必要としない重要な情報が含まれています。	2: オペレーター情報 このメッセージは、システム状態の変化を示します。アクションを要求しません。逆に、アクションを必要とする可能性がある条件に応じて、マスター・コンソールのオペレーターに警告します。
TYPE_WARNING (または TYPE_WARN)	12: 重要な情報 (上記の説明のとおり)	
TYPE_INFORMATION (ま たは TYPE_INFO)	4: システム状況 メッセージは、システム・タスクの状況またはハードウェア装置の状況を示します。	この宛先コードは、状況がオペレーターの問い合わせによって特に要求されていないときに、ジョブ状況を示す任意のメッセージに使用します。

ユーザーのアプリケーションに対してトレース要求を発行するときのバックグラウンド

トレース・データ収集の目的は、十分な情報を提供して、アプリケーションでの問題を診断することです。Ras Toolkit に対する WebSphere for z/OS ランタイム・サポート (JRas サポート) によって、ユーザーの Java アプリケーションからトレース要求を発行することができ、その結果生成されたトレース・データを、WebSphere for z/OS が使用するコンポーネント・トレース (CTRACE) データ・セットに記録させることができます。ユーザーのアプリケーションが稼動する WebSphere for z/OS J2EE サーバーまたは MOFW サーバーの CTRACE データ・セットに、ユーザーのアプリケーションのトレース・データが表示されます。

WebSphere for z/OS は、トレース・ロガーを作成し管理するコードを提供します。トレース・ロガーは、アプリケーションのトレース要求を処理します。トレース・ロガーは、ユーザーの Java アプリケーションが稼動する WebSphere

for z/OS J2EE サーバーまたは MOFW サーバーの Java 仮想マシン (JVM) で稼働します。トレース・ロガーを使用するには、Java アプリケーションで以下のことを行う必要があります。

1. トレース・ロガーを定義する。
2. トレース・ロガーを作成するように WebSphere for z/OS に指示するメソッドを駆動する。
3. アプリケーションの適切なトレース・ポイントでトレース要求をコード化する。

アプリケーションを更新して JRAS サポートを使用するための特別な指示は、91ページの『Java アプリケーションをコード化してメッセージとトレース要求を発行するためのステップ』にあります。ただし、これらの指示を使用してトレース要求を適切にコード化できるようになるには、以下のトピックにある概念について理解しておく必要があります。

- 『トレース・ポイントを配置する場所と要求するデータの決定』
- 91ページの『トレース・ポイントへのトレース・タイプの割り当て』

トレース・ポイントを配置する場所と要求するデータの決定

WebSphere for z/OS J2EE または MOFW サーバーで稼働中の Java アプリケーション用のトレース・データを収集するには、アプリケーションのコード内でトレース・ポイントを配置する場所を決定する必要があります。このトレース・ポイントでは、RASTraceLogger クラス・インターフェースを使用してトレース項目を要求することができます。一般的なトレース・ポイントには、以下が含まれます。

- メソッド入り口
- メソッド出口
- 機能要求の開始
- 要求完了のプロセスの主なチェックポイント
- 機能要求の完了
- 他のシステム機能へのインターフェース
- 検出された入出力エラーまたは予期しない例外などの、異常なイベント

トレース項目に記録する情報の内容も決定しなければなりません。このトレース項目は可変データ量を保持できます。WebSphere for z/OS は、作業単位またはトランザクションのアドレス・スペース ID (ASID) とタスク制御ブロック (TCB)、またはスレッドの Java 名を自動的に収集します。以下は、WebSphere for z/OS J2EE サーバーまたは MOFW サーバーで稼働中の Java アプリケーションのトレース項目に入れる可能性がある追加のデータ・タイプを示しています。

- アプリケーションがサービスを提供している作業単位またはトランザクションの識別。これは、JOBNAME、USERID、またはトランザクション ID でもかまいません。
- 機能要求の開始をトレースする項目の場合は、入力パラメーター。
- 内部チェックポイントの場合は、このトレース項目を元の要求に結合する識別、およびプロセスの現在の状況に関する情報。
- 異常なイベントの場合は、問題の原因と任意の追加のデータ。たとえば、あらゆる例外とスタック・トレースを記録することができます。
- サービスからの戻りでは、戻りコードと理由コード。
- デバッグ補助機能ではなく、分析に使用するトレース項目の場合は、アプリケーションのユーザーが必要とするあらゆる情報。

トレース・ポイントへのトレース・タイプの割り当て

Java アプリケーションで定義するトレース・ポイントごとに、RASITraceLogger インターフェースで定義したメソッドを使用して、トレース項目を要求します。トレース要求の一部として、この特定の要求にトレース・タイプを割り当てます。RASITraceEvent インターフェースは、使用するタイプを定義します。

注: エンタープライズ版 V3.02 JRas サポートでは、ユーザーのアプリケーション内のトレース・ポイントにトレース・レベルを割り当てる必要がありました。このような割り当ては依然としてサポートされているので、トレース・レベルを使用するアプリケーションを再コード化する必要はありません。

トレース要求をコード化すると、Java アプリケーションは稼動中にトレース要求を発行することができるようになります。ただし、要求されたトレース・データを実際に記録するには、アプリケーションが稼動する WebSphere for z/OS J2EE サーバーまたは MOFW サーバーでトレースが使用可能でなければなりません。98ページの『Java アプリケーションのメッセージとトレース要求をログに記録するための z/OS または OS/390 環境を作成するステップ』には、特定のトレース・タイプについてトレースを使用可能にする方法の詳細が記載されています。

Java アプリケーションをコード化してメッセージとトレース要求を発行するためのステップ

メッセージを発行するためとトレース項目をログに記録するための指示をコード化することによって、ユーザーの Java サーバー・アプリケーションの信頼性、可用性、保守性 (RAS) を改善することができます。Java アプリケーシ

ンが WebSphere for z/OS J2EE サーバーまたは MOFW サーバーで稼働する場合、そのメッセージは MVS マスター・コンソールと、WebSphere for z/OS で使用するコンポーネント・トレース (CTRACE) データ・セットに表示されません。アプリケーションのトレース項目は、同じ CTRACE データ・セットに表示されます。エラー・メッセージは、WebSphere for z/OS で使用されるエラー・ログ・ストリームにも表示されます。

始める前に:

- ユーザーの Java アプリケーションからメッセージを発行したい場合は、メッセージをインラインで定義するか、あるいは別々のファイルを使用してメッセージを含めることができます。コード化を開始する前に、使用したい方法を決定します。必要な場合は、これらの 2 つの方法に関する 87 ページの『インライン・メソッドの呼び出しまたはメッセージ・プロパティ・ファイルによるメッセージの定義』を参照してください。
- メッセージを発行したりトレース項目をログに記録したりするために使用できる JRas インターフェースおよびメソッドの説明については、*Using the WebSphere JRas Message Logging and Trace Facility* を参照してください。これには、z/OS または OS/390 などのサポートされているすべてのプラットフォームに適用される、com.ibm.ras パッケージ内のメソッドについての説明があります。

以下のステップを実行して、ユーザーのサーバー・アプリケーションにコードを追加し、メッセージとトレース記入項目要求を z/OS または OS/390 メッセージ・ロギング機能とトレース・データ・ロギング機能に送信します。

1. (オプション) ユーザーのアプリケーションからのメッセージをログに記録する場合、およびメッセージをインラインで定義していない場合は、メッセージ・プロパティ・ファイルを作成します。Java アプリケーションが発行するそれぞれのメッセージごとに、キーとテキストを対にしてメッセージを定義します。
 - マスター・コンソールまたはエラー・ログ・ストリームに表示される内容を示すには、テキスト部分を使用します。
 - メッセージ・プロパティ・ファイルと Java アプリケーション・コードの両方に表示される内容を示すには、キーを使用します。これによって、ランタイム・コードで適切なメッセージ・テキストを検出することができます。

規則:

- キーをテキストから分離するには、いつでも等号を使用します。たとえば、次のようになります。


```
BBOJ0001=BBOJ0001 Java B0 created.  
BBOJ0002=BBOJ0002 Policy number {0} obtained.
```

- 変数データを含むメッセージ・テキストには、配置と内容を示すための特別なコード化が必要です。変数テキストをもつメッセージを正しく定義するには、中括弧 {} を使用して、変数はテキスト内の特別の位置に表示されることを示します。中括弧内では、数字を使用して、この位置に属する変数を示します。

たとえば、ユーザーのコードには、以下の指示が含まれていると想定します。

```
String day = "Monday";  
Integer temp = new Integer(75);  
msgLogger.message(RASIMessageEvent.TYPE_INFO,  
                 this,  
                 "methodName",  
                 "APPL061I",  
                 day,  
                 temp);
```

このメッセージを正しく定義するには、メッセージ・プロパティ・ファイルで以下をコード化します。

```
APPL061I=APPL061I On {0}, it is {1} degrees.
```

-
2. ユーザーのアプリケーションに適したアプリケーション開発ツールを使用して、ユーザーの Java アプリケーションのソース・コードを以下のように編集します。

- `com.ibm.ras` パッケージと `com.ibm.WebSphere` パッケージのインポート・ステートメントを追加する。たとえば、以下を入力します。

```
import com.ibm.ras.*;  
import com.ibm.websphere.ras.*;
```

- メッセージ・ロガーとトレース・ロガーの定義ステートメントを追加する。たとえば、以下を入力します。

```
private RASIMessageLogger msgLogger = null;  
private RASITraceLogger trcLogger = null;
```

-
3. ユーザーの Java アプリケーションのコンストラクターを編集して、メッセージ・ロガーかトレース・ロガーまたはこの両方を作成します。

ロガーのタイプ	完了するステップ
メッセージ	<ul style="list-style-type: none"> • createRASMessageLogger メソッドを使用して、メッセージ・ロガーを要求する。 • (オプション) アプリケーションからメッセージを発行するための方法に、インラインではなくファイルを使用している場合には、メッセージ・プロパティ・ファイルを定義する。
トレース	createRASTraceLogger メソッドを使用して、トレース・ロガーを要求する。

規則:

- アプリケーションは、RASIMessageLogger オブジェクト型として createRASMessageLogger メソッドが戻すオブジェクトを参照する必要があります。
- アプリケーションは、RASITraceLogger オブジェクト型として createRASTraceLogger メソッドが戻すオブジェクトを参照する必要があります。

ヒント: WebSphere for z/OS 用に予約されている接頭部 `com.ibm.` で始まるロガー名の使用は避けてください。

-
4. Java アプリケーションからメッセージを発行する場合は、アプリケーションのソース・コードの適切なポイントでメッセージを追加します。

規則:

- メッセージをインラインで定義している場合は、`textMessage` メソッドを RASIMessageLogger インターフェイスで使用して、メソッドの呼び出し時に完全なメッセージを文字列で指定します。
- メッセージ・プロパティ・ファイルを使用している場合は、`message` メソッドまたは `msg` メソッドを RASIMessageLogger インターフェイスで使用して、メソッドの呼び出し時にメッセージ・キーを指定します。たとえば、次のようになります。

```
msgLogger.message(RASIMessageEvent.TYPE_INFO,
                  "com.myCompany.JRasSample",
                  "doSomething",
                  "BBOJ0001");
```

- それぞれのメッセージごとに、88ページの『メッセージ・タイプがメッセージ宛先に及ぼす影響について』のガイドラインに説明されているように、適切なタイプを割り当てます。

注:

- a. 各メッセージには 1 つのメッセージ・タイプだけを割り当てる。
 - b. メッセージのタイプを割り当てないか、あるいはタイプに "null" を指定すると、Java コンパイラーはエラー・メッセージを発行する。
 - c. 無効なタイプを割り当てると、メッセージ・ロガーは、そのメッセージを監査タイプとして処理する。
- メッセージに使用する各文字は、EBCDIC 文字にマップしなければならない。
 - メッセージをマスター・コンソールにルーティングする場合、WebSphere for z/OS はメッセージ・テキストの最初の 700 文字しか送信しない。

制限: エラー・ログ・ストリームにメッセージを書き込む場合、WebSphere for z/OS は、識別のためにメッセージ・テキストに追加する情報を含めて、データの 512 文字しか使用しません (この追加情報は、日付、時刻、組織名、などで構成されます)。アプリケーション・メッセージ用のエラー・ログ・ストリーム項目の形式と内容については、*WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断, GA88-8655* を参照してください。

-
5. Java アプリケーション用のトレース・データを収集する場合は、アプリケーションのソース・コードの適切なポイントでトレース要求を追加します。

規則:

- それぞれのトレース要求ごとに、RASITraceEvent インターフェースで定義されている適切なタイプを割り当てます。

注: トレース要求にタイプを割り当てないと、トレース・ロガーはそのトレース要求を無視します。

- トレース・データに使用する各文字は、EBCDIC 文字にマップする必要があります。

制限: トレース・データの処理時に、WebSphere for z/OS は 16 進データまたは文字データの限定量しか使用しません。

- 16 進のトレース・データ (Java バイト配列からの) の場合、WebSphere for z/OS は 1024 バイトより後のデータを切り捨てます。

- 文字のトレース・データの場合は、そのトレース・データが 16384 文字を超えると、WebSphere for z/OS はリテラル `***BUFFER OVERFLOW***` で置き換えます。この累積制限により、それぞれの文字ストリングごとに 1 バイトのストリング終了文字が組み込まれます。

ヒント: アプリケーションのパフォーマンスを改善するために、以下の代替のいずれかを使用することができます。

- `RASTraceLogger.isLogging` 変数のテストでのトレース呼び出しをラップする。トレース・ロギングがアクティブでない場合、この変数は `false` に設定されます。
- `if` 文で `isLogging` メソッドを使用して、トレース・ロギングがトレースのどのレベルでもアクティブであるかどうかをテストする。
- `isLoggable` メソッドを使用して、指定されたトレース・タイプでロギングがアクティブであるかどうかを判別する。

最初の 2 つの方法を使用すると、トレース・ロギングがアクティブでなくても、トレース項目の作成がオーバーヘッドになることはありません。対照的に、`isLoggable` メソッドでは、オーバーヘッドを多く必要としますが、特にトレースの一部のレベルがいつでもアクティブである場合に適したオプションとなる場合があります。

-
6. ユーザーの Java アプリケーションに適したアプリケーション開発ツールを使用して、ユーザーのアプリケーションのコードを生成し、コンパイルします。
-

ユーザーの Java アプリケーションの実行可能コードがあれば、98ページの『Java アプリケーションのメッセージとトレース要求をログに記録するための z/OS または OS/390 環境を作成するステップ』に示されているステップを完了する準備ができています。

例: これまでに説明したコーディング要件を、以下に例示します。この例では、次のメッセージ定義を含む `com/myCompany/JRasSample.properties` という名前のメッセージ・プロパティ・ファイルを使用することを想定していません。

```
BB0J0001=BB0J0001 Java B0 created.  
BB0J0002=BB0J0002 Policy number {0} obtained.  
BB0J0003=BB0J0003 Java B0 destroyed.
```

```

package com.myCompany;

// Import JRas and Websphere packages
import com.ibm.ras.*;
import com.ibm.websphere.ras.*;

public class JRasSample
{
    // Loggers
    private RASIMessageLogger msgLogger = null;
    private RASITraceLogger trcLogger = null;
    // Message file
    private static final String MSG_FILE = "com.myCompany.JRasSample";
    // Array of trace objects
    Object[] objs = new Object[3];

    // Constructor
    public JRasSample()
    {
        // Get logger manager object
        Manager manager = Manager.getManager();
        // Get logger
        trcLogger = manager.createRAsTraceLogger("com.myCompany","myProduct",
                                                "myComponent","myLogger.COM");
        msgLogger = manager.createRAsMessageLogger("com.myCompany","myProduct",
                                                  "myComponent","myLogger.COM");
        msgLogger.setMessageFile(MSG_FILE);
    }

    // Example of JRas trace events and messages
    public int doSomething(String parm1,String parm2,String parm3)
    {
        int returnValue = 0;
        byte[] byteArray = {1,2,3,4,5};

        // Trace input parameters
        objs[0] = parm1;
        objs[1] = parm2;
        objs[2] = parm3;
        if (trcLogger.isLoggable(RASITraceEvent.TYPE_ENTRY_EXIT))
            trcLogger.entry(RASITraceEvent.TYPE_ENTRY_EXIT,
                           "com.myCompany.JRasSample",
                           "doSomething",
                           objs);
        if (trcLogger.isLoggable(RASITraceEvent.TYPE_MISC_DATA))
        {
            // Trace a text string
            trcLogger.trace(RASITraceEvent.TYPE_MISC_DATA,
                           "com.myCompany.JRasSample",
                           "doSomething",
                           "Text data to be traced");
            // Trace binary data
            trcLogger.trace(RASITraceEvent.TYPE_MISC_DATA,
                           "com.myCompany.JRasSample",
                           "doSomething",

```

```

        byteArray);
    // Trace the current stack
    trcLogger.stackTrace(RASITraceEvent.TYPE_MISC_DATA,
        "com.myCompany.JRasSample",
        "doSomething");
}
// Issue message to WTO
msgLogger.message(RASIMessageEvent.TYPE_INFO,
    "com.myCompany.JRasSample",
    "doSomething",
    "BBOJ0001");
// Issue warning message to WTO
msgLogger.message(RASIMessageEvent.TYPE_WARN,
    "com.myCompany.JRasSample",
    "doSomething",
    "BBOJ0002",
    "123");
// Issue error message to WTO and error log
msgLogger.message(RASIMessageEvent.TYPE_ERR,
    "com.myCompany.JRasSample",
    "doSomething",
    "BBOJ0003");
// Trace return value
if (trcLogger.isLoggable(RASITraceEvent.TYPE_ENTRY_EXIT))
    trcLogger.exit(RASITraceEvent.TYPE_ENTRY_EXIT,
        "com.myCompany.JRasSample",
        "doSomething",
        returnValue);
return returnValue;
}

// This method is invoked when a JRasSample object is traced
public String toString()
{
    String traceString = "This is the JRasSample object trace data";
    return traceString;
}

public static void main(String[] args)
{
    JRasSample sample = new JRasSample();
    sample.doSomething("parm1", "parm2", "parm3");
}
}

```

Java アプリケーションのメッセージとトレース要求をログに記録するための z/OS または OS/390 環境を作成するステップ

始める前に:

- WebSphere for z/OS のインストール過程で、エラー・ログ・ストリームとコンポーネント・トレース・データ・セットがセットアップされたかどうか、適切なインストール担当者にお問い合わせます。エラー・ログと CTRACE デ

ータ・セットがすでに使用可能になっている場合がありますが、その一方で、インストール担当者は、他の WebSphere for z/OS サーバーとアプリケーションからの現行データだけでなく、ユーザーのアプリケーション・データを処理するために、変更する必要があると決定する場合があります。たとえば、インストール時に、すべての WebSphere for z/OS サーバーに共通のエラー・ログ・ストリームか、またはそれぞれの個々のサーバーごとに別々のログ・ストリームのいずれかをセットアップすることができます。インストール時に、共通のログの使用から別々のログの使用に切り替え、ユーザーの Java アプリケーションからの追加の診断データに適用した場合があります。

- J2EE サーバーまたは MOFW サーバーでアプリケーションのトレースをオンにするために、JVM プロパティ・ファイルを編集するか作成する必要があります。この作業を行うには、このファイルを編集したり適切なディレクトリに保管するための特別な権限が必要な場合があります。WebSphere for z/OS をご使用のシステムにインストールしたシステム・プログラマーに問い合わせてください。

注:

1. エラー・ログ・ストリームのセットアップに関する説明は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 にあります。
2. CTRACE のセットアップと実行に関する説明は、*WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断*, GA88-8655 にあります。

以下のステップを実行して、ロギング・サポートのための z/OS または OS/390 環境をセットアップします。

1. WebSphere for z/OS J2EE サーバーまたは MOFW サーバーがユーザーのアプリケーションのトレース・データを収集したりログに記録したりできるようにするには、z/OS または OS/390 上でトレース設定ファイルを階層ファイル・システム (HFS) 内に作成します。このファイルの中に、必要なトレース設定値を次の形式で入力してください。

```
logger_name=type=[enabled|disabled]
```

例:

```
myLogger.COM=all=enabled
```

`logger_name` は、トレース・ロガーを取得するよう create メソッドをコード化したときにユーザーのアプリケーションのソース・コード内に指定したロガー名に対応します。複数のロガー名についてロギング・サポートを使用

可能にするとき、各ロガー名の全体を略さずに指定する代わりに、共通接頭部にアスタリスクを付けて指定する (たとえば、a.b.c.*) ことができます。a.b.c.* のように指定すると、a.b.c.d や a.b.c.e という名前のロガーについてロギングが可能になります。

ヒント: WebSphere for z/OS 用に予約されている接頭部 `com.ibm.` で始まるロガー名の使用は避けてください。

`type` は、次の表のプロパティー値の 1 つに対応します。プロパティー・タイプは、大文字と小文字の区別がなされます。

表 6. トレース設定プロパティー・タイプとこれに対応する JRas トレース・タイプ

指定するプロパティー・タイプ	トレースを使用可能にする JRas トレース・タイプ
all	サポートされているすべての RASITraceEvent タイプ
event	<ul style="list-style-type: none"> • RASITraceEvent.TYPE_ERROR_EXC • RASITraceEvent.TYPE_SVC • RASITraceEvent.TYPE_OBJ_CREATE • RASITraceEvent.TYPE_OBJ_DELETE • RASITraceEvent.TYPE_LEVEL1
entryExit	<ul style="list-style-type: none"> • RASITraceEvent.TYPE_ENTRY_EXIT • RASITraceEvent.TYPE_API • RASITraceEvent.TYPE_CALLBACK • RASITraceEvent.TYPE_PRIVATE • RASITraceEvent.TYPE_PUBLIC • RASITraceEvent.TYPE_STATIC • RASITraceEvent.TYPE_LEVEL1 • RASITraceEvent.TYPE_LEVEL2
debug	<ul style="list-style-type: none"> • RASITraceEvent.TYPE_MISC_DATA • RASITraceEvent.TYPE_LEVEL1 • RASITraceEvent.TYPE_LEVEL2 • RASITraceEvent.TYPE_LEVEL3

規則:

- 同じトレース・プロパティー・ファイルを使用して、特定のロガーについての異なるトレース・タイプを使用可能にすることができます。各ロガーのトレース・タイプを同じ行に定義する場合、単一のコロン (:) を使用して各ロガーのトレース設定を区別する必要があります。

例 (それぞれのロガーを別々の行に定義):


```
com.aCompany.*=all=enabled
com.anotherCompany.*=event=enabled
```

例 (複数のロガーを同じ行に定義):

```
com.aCompany.*=all=enabled:com.anotherCompany.*=event=enabled
```

- 1 つのロガーに複数のトレース・タイプを指定するときは、それぞれのトレース・タイプをコンマ (,) で区切ってください。

例:

```
com.aCompany.aComponent=debug=enabled,event=enabled
```

-
2. 作成したトレース設定ファイルを指すように、新規の Java 仮想マシン (JVM) プロパティ・ファイルを作成するか、既存の Java 仮想マシン (JVM) プロパティ・ファイルを編集します。jvm.properties という名前のこのプロパティ・ファイルによって、WebSphere for z/OS J2EE サーバーまたは MOFW サーバーで稼働する JVM のデフォルト設定が変更されます。

規則:

- トレース設定ファイルの完全修飾ディレクトリー・パスおよびファイル名に、com.ibm.ws390.trace.settings システム・プロパティを設定する必要があります。このシステム・プロパティを指定しない場合、またはパスとファイル名を誤って指定した場合には、すべてのトレース・タイプが使用不可 (デフォルト設定) になります。
- WebSphere for z/OS から jvm.properties ファイルにアクセスできるようにして、サーバーをアクティブにしたときにプロパティ設定値を検出して使用できるようにする必要があります。ユーザーの Java アプリケーションが稼働するサーバーの環境変数設定値を含む current.env ファイルを WebSphere for z/OS が入れるディレクトリーと同じ HFS ディレクトリーの中に、この jvm.properties ファイルを入れてください。このディレクトリーについての詳細は、127ページの『付録A. 環境ファイルおよび JVM プロパティ・ファイル』を参照してください。
- トレース・ロギングを動的に開始したり、停止したりすることはできません。

-
3. J2EE サーバーまたは MOFW サーバーによるコンポーネント・トレースの使用に関連した環境変数の設定値を検査します。値の一部を変更して、CTRACE データ・セット内の追加のトレース項目に適用することができます。特に、以下の環境変数の設定値を検査してください。

- TRACEBUFFCOUNT
- TRACEBUFFSIZE

-
4. ユーザーのアプリケーションが稼働する WebSphere for z/OS J2EE サーバーまたは MOFW サーバーを始動します。
 - サーバーにすでにインストールした既存のアプリケーションに対して JRas サポートを設定した場合、以下の確認を行う必要があります。
 - a. 新たにコンパイルしたコードで既存のコードが置換されていることを確認する。
 - b. `jvm.properties` ファイルまたは環境変数に加えた変更内容を WebSphere for z/OS サーバーがピックアップしていることを確認する。サーバーを一度停止してから再始動して、これらの変更内容をピックアップする必要があります。
 - 新しいアプリケーションに対して JRas サポートを設定した場合は、適切なプロセスに従ってユーザーの Java アプリケーションをアセンブルし、WebSphere for z/OS サーバーにインストールしてください。J2EE サーバーにインストールするアプリケーションについては、19ページの『第2部 J2EE サーバー・アプリケーションの作成、アセンブル、および配置』に記載されている情報を参照してください。
-

メッセージとトレース・データを表示するときのバックグラウンド

Java アプリケーションが稼働を開始すると、以下のように、そのメッセージとトレース・データを表示することができます。

表示に使用する出力のタイプ	使用する指示
MVS マスター・コンソール上のメッセージ	メッセージ・ロガーは、メッセージを読み取り可能な形式で MVS マスター・コンソールに自動的に経路を定めます。メッセージの外観と期間は、インストール時のコンソール構成のセットアップの仕方によって異なります。必要な場合には、メッセージ表示の制御、スクロール、および削除を含めた、コンソールの構成方法に関する説明について、 <i>z/OS MVS 計画: 操作</i> , SA88-8573 を参照してください。

表示に使用する出力のタイプ	使用する指示
エラー・ログ・ストリーム内のメッセージ	<p>エラー・ログ・ストリームにメッセージを表示するには、ログ・ブラウザ・ユーティリティ (BBORBLOG) を使用します。ログ・ブラウザ・ユーティリティの使用法およびメッセージ出力の例については、<i>WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断</i>, GA88-8655 を参照してください。</p>
コンポーネント・トレース内のメッセージまたはトレース・データ	<p>コンポーネント・トレースにメッセージまたはアプリケーション・トレース・データを表示するには、以下の方法のいずれかで、対話式問題制御システム (IPCS) を使用しなければなりません。</p> <ul style="list-style-type: none"> • 端末でのライン・モード (IPCS CTRACE コマンド) • 端末でのフルスクリーン・モード (IPCS ダイアログ) • 端末モニター・プログラムを使用したバッチ・モード <p>推奨: IPCS、TSO/E および ISPF に精通していない場合は、104 ページの『IPCS をバッチ・モードで使用して、アプリケーション・トレース・データをフォーマットするためのステップ』で説明するように、バッチ・モードの IPCS を使用して、トレース・データのフォーマットと表示を行います。</p> <p>IPCS ダイアログの使用法およびメッセージとトレース・データ出力の例については、<i>WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断</i>, GA88-8655 を参照してください。</p>

注: ユーザーの Java アプリケーションのトレース・データを表示する場合、メッセージと CTRACE レコードは、ユーザーのアプリケーションがメッセージとトレース要求を発行した順序で表示されない可能性があります。すべてのメッセージ要求は、お互いに関連した順次配列で表示されます。同様に、すべての CTRACE レコードも、お互いに関連した順序で表示されます。ただし、トレース・データのタイプが異なると、順次に表示されない可能性があります。たとえば、トレース要求の後に発行されたメッセージが、トレース要求の前のトレース出力に表示される可能性があります。

IPCS をバッチ・モードで使用して、アプリケーション・トレース・データをフォーマットするためのステップ

コンポーネント・トレースからのメッセージまたはアプリケーション・トレース・データを表示するには、対話式問題制御システム (IPCS) を使用してデータをフォーマットしなければなりません。IPCS、TSO/E、および ISPF の使用経験が少ない場合には特に、バッチ・モードで IPCS を使用することは、データをフォーマットする最も簡単な方法です。バッチ・モードによって、IPCS を使用してトレース・データをフォーマットしたり、このトレース・データを MVS データ・セットに書き込むことができます。オプションで、そのデータ・セットの内容を HFS ファイルにコピーして、表示することができます。

始める前に: IPCS をバッチ・モードで使用できるようにする前に、IPCS ダンプ・ディレクトリーを作成しなければなりません。IPCS のセットアップ時に、インストールにより、IPCS をユーザーのためにカスタマイズすることができます。このカスタマイズには、IPCS ダンプ・ディレクトリーを作成するために、IBM 提供の BLSCDDIR CLIST のデフォルト値を変更することを組み込むことができます。

インストールにより BLSCDDIR CLIST が変更されている場合は、以下のステップを実行して、IPCS ダンプ・ディレクトリーを作成します。

1. このディレクトリー用の完全修飾データ・セット名を決定する。
2. TSO/E コマンド・プロンプトから、BLSCDDIR コマンドを入力して、データ・セット名を指定する。たとえば、IBMUSER.DDIR という名前のダンプ・ディレクトリーを作成するには、以下を入力します。

```
%blscddir dsn('ibmuser.ddir')
```

インストール時に、IPCS をカスタマイズしていない場合は、他の BLSCDDIR CLIST パラメーターを更新する必要がある場合があります。BLSCDDIR CLIST を使用してダンプ・ディレクトリーを作成することの詳細については、*z/OS MVS 対話式問題管理システム (IPCS) ユーザーズ・ガイド*、SA88-8568 および *z/OS MVS 対話式問題管理システム (IPCS) コマンド*、SA88-8566 を参照してください。

IPCS をバッチ・モードで使用して、アプリケーション・トレース・データをフォーマットするには、以下のステップを実行します。

1. ファイルを作成し、以下のサンプル JCL をそのファイルにコピーする。この JCL が IPCS を呼び出し、JRAS トレース・データを抽出してフォーマットし、そのトレース・データを MVS データ・セットに書き込んだあと、TSO/E OPUT コマンドを使用してフォーマットされたデータを MVS データ・セットから HFS ファイルに書き込みます。

```

//IBMUSERX JOB ,
// CLASS=J,NOTIFY=&SYSUID,MSGCLASS=H
//IPCS EXEC PGM=IKJEFT01,REGION=4096K,DYNAMNBR=50
//IPCSDDIR DD DSN=IBMUSER.DDIR,DISP=SHR
//IPCSDOC DD SYSOUT=H
//JRASTRC DD DSN=IBMUSER.CB390.CTRACE,DISP=SHR
//IPCSPRNT DD DSN=IBMUSER.IPCS.OUT,DISP=OLD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
IPCS
DROPDUMP DDNAME(JRASTRC)
PROFILE LINESIZE(80)PAGESIZE(99999999)
SETDEF NOCONFIRM
CTRACE COMP(SYSBBOSS) DDNAME(JRASTRC) FULL PRINT +
NOTERMINAL
DROPDUMP DDNAME(JRASTRC)
END
/*
//OPUT EXEC PGM=IKJEFT01,REGION=4096K,DYNAMNBR=50
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
oput 'ibmuser.ipcs.out' '/u/ibmuser/ipcs/jrastrace.txt' TEXT
/*

```

-
2. サンプル JCL を編集し、IBMUSER.DDIR を、作成した IPCS ダンプ・ディレクトリーに使用したデータ・セット名と置き換える。

注:

- a. 出力データ・セットを印刷しない場合にのみ、PROFILE ステートメントで PAGESIZE パラメーターを使用します。
- b. HFS ファイル名を既存の HFS ファイルの名前と置き換えることができますが、そのようにする必要はありません。指定したファイルがない場合には、OPUT コマンド処理により新規 HFS ファイルが作成され、読み取りアクセスと書き込みアクセスがユーザーのユーザー ID にのみ認可されます。

既存の HFS ファイルを指定する場合は、OPUT コマンド処理により、そのファイル内にすでにあるあらゆるデータが上書きされます。OPUT コマンドについて詳しく知りたい場合は、*z/OS UNIX システム・サービス コマンド解説書, SA88-8641* を参照してください。
- c. 例の JRASTRC DD で指定されたデータ・セット名を、CTRACE データが入っているデータ・セットの名前に変更してください。
- d. 必要な場合には、SYSTSIN ストリーム内の JRASTRC DD ステートメントと OPUT コマンドの両方にある MVS データ・セットの名前を変更してください。まず、JRAS CTRACE データのフォーマット済み出力が

IPCSPRNT DD ステートメントで指定された MVS データ・セットに書き込まれたあと、(オプションで) HFS データ・セットにコピーされます。このデータ・セットを事前に割り当てるか、あるいは、サンプル JCL を変更してデータ・セットを割り当てるか、いずれかを行わなければなりません。このデータ・セットのレコード・フォーマットは VBA であり、レコード長は 133 でなければなりません。

3. JCL に IPCS バッチ・ジョブを開始するよう実行依頼する。

完了すると、vi などの UNIX エディターを使用して、HFS ファイル内のトレース・データを表示できるようになります。UNIX エディターについて詳しく知りたい場合は、*z/OS UNIX システム・サービス ユーザーズ・ガイド*, SA88-8640 を参照してください。

IPCS ダイアログを使用して CTRACE データをフォーマットすることについての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断*, GA88-8655 を参照してください。

第4部 J2EE サーバーへのアプリケーションのマイグレーション

これ以降の章には、WebSphere for z/OS J2EE サーバー環境へのアプリケーションのマイグレーションに関する情報が記載されています。WebSphere for z/OS V4.0 は、最新の J2EE プログラミングおよびパッケージ化仕様に準拠する必要があります。特定の開発、アセンブリー、およびインストール・ツールを使用する必要がありますので、説明に従ってアプリケーション・コンポーネントをマイグレーションする前に、次の概説的情報を読むことを検討してください。

- WebSphere for z/OS J2EE サーバーで J2EE アプリケーションの開発、アセンブル、およびインストールを行うためのツールとプロセスの簡単な概要については、9ページの『第2章 アプリケーション・ツールの概要』
- マイグレーションの概念、作業、および推奨事項の簡単な概要については、109ページの『第11章 マイグレーションのバックグラウンド』

WebSphere for z/OS J2EE サーバー用の J2EE アプリケーションを開発した経験がある場合、表7 を利用して、特定のアプリケーション・コンポーネントをマイグレーションする手順へ直接進んでください。たとえば追加情報が必要な場合のツールに関する詳細情報の参照先を、手順ごとに示してあります。

表7. アプリケーションのマイグレーション・パス一覧

マイグレーションするアプリケーション・コンポーネントのタイプ	マイグレーション元の WebSphere 製品	参照項目
Enterprise bean	分散プラットフォーム上のアドバンスド版 V3.5	119ページの『第13章 WebSphere for z/OS プラットフォームへのアプリケーションのマイグレーション』

表7. アプリケーションのマイグレーション・パス一覧 (続き)

マイグレーションする アプリケーション・ コンポーネントのタイプ	マイグレーション 元の WebSphere 製品	参照項目
サーブレットと JavaServer Pages	分散プラットフォーム上の スタンダード版 または アドバンスド版 V3.5	119ページの『第13章 WebSphere for z/OS プラットフォームへのアプリケーションのマイグレーション』
	z/OS または OS/390 上の スタンダード版	114ページの『z/OS または OS/390 スタンダード版の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ』
Java ビジネス・オブジェクト (CORBA)	z/OS または OS/390 上の エンタープライズ版 V3.02	114ページの『OS/390 エンタープライズ版 V3.02 の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ』

第11章 マイグレーションのバックグラウンド

一般的には、既存のアプリケーションを IBM 製品の最新バージョンまたはリリースで引き続き正常に機能させるために実行するアクションが、マイグレーション作業には含まれます。しかし、J2EE サーバー環境を WebSphere for z/OS または OS/390 で利用するために、WebSphere ファミリー環境から WebSphere for z/OS または OS/390 へ既存の J2EE アプリケーションをマイグレーションする場合があります。

表8 は、マイグレーション・アクションのチェックリストであり、このアクションは、マイグレーションしようとするアプリケーションのタイプおよびその現行ランタイム環境によって、必須である場合もオプションである場合もあります。このチェックリストには、Sun Microsystems J2EE の仕様に基づいた、各アクションまたは一連のアクションを実行するための該当するほとんどの要員の役割も示されています。多くのアクションは、19ページの『第2部 J2EE サーバー・アプリケーションの作成、アSEMBル、および配置』に説明されている新規アプリケーションの開発、アSEMBル、配置、およびインストールを行うステップと同じです。これらの手順を熟知していると、マイグレーション作業をより効率的に完了するのに役立ちます。

特定の状況でどのマイグレーション・アクションが必要かを判断するために、以下の章の詳細情報を参照してください。

- 113ページの『第12章 WebSphere for z/OS の新規リリースへのアプリケーションのマイグレーション』
- 119ページの『第13章 WebSphere for z/OS プラットフォームへのアプリケーションのマイグレーション』
- 123ページの『第14章 WebSphere for z/OS J2EE サーバーにすでにインストールされているアプリケーションのアップグレード』

表8. 役割および実行する可能性のあるマイグレーション・アクションのチェックリスト

<input checked="" type="checkbox"/>	実行する可能性のあるマイグレーション・アクション	条件 / コメント
システム・プログラマーまたは管理者の場合:		
<input type="checkbox"/>	製品、システム、データベース、または構成の変更がユーザーのアプリケーションにどのように影響するかを理解します。	すべてのマイグレーション・パスについて必須

表 8. 役割および実行する可能性のあるマイグレーション・アクションのチェックリスト
(続き)

✓	実行する可能性のあるマイグレーション・アクション	条件 / コメント
<input type="checkbox"/>	DB2 などの J2EE リソースを事前構成しません。	オプション 。インストールされるアプリケーションの要件によります (接続プーリングの場合は必須)。
アプリケーション・コンポーネント提供者の場合:		
<input type="checkbox"/>	ツールの変更がユーザーのアプリケーションにどのように影響するかを理解します。	WebSphere for z/OS は特定のツールを必要とするので 必須
<input type="checkbox"/>	インターフェースの変更がユーザーのアプリケーションにどのように影響するかを理解します。	WebSphere for z/OS は、特定レベルのプログラミング仕様に準拠する必要があるので 必須
<input type="checkbox"/>	インターフェース変更のためにコンポーネントの設計を変更するかソース・コードを再作成します。	アプリケーション・コンポーネントのタイプ、および必須レベルのプログラミング仕様へのアプリケーション・コンポーネントの準拠によっては 必須
<input type="checkbox"/>	インターフェースの拡張、または開発ツールに対する拡張やメンテナンスを検出するコードを再生成します。	オプション 。上記と同様
アプリケーション・アセンブルまたは配置の担当者の場合:		
<input type="checkbox"/>	アプリケーション・アセンブリーおよび配置ツールの拡張やメンテナンスのためにアプリケーションを再パッケージし、メタデータを再生成します。	WebSphere for z/OS J2EE サーバーにインストールされるすべてのアプリケーションについて 必須
アプリケーション・インストール担当者の場合:		
<input type="checkbox"/>	再アセンブルされたすべてのアプリケーションを再インストールして既存のアプリケーションを置換するか、または新規 EAR ファイルとメタデータを追加します。	WebSphere for z/OS J2EE サーバーで稼働するすべてのアプリケーションについて 必須
<input type="checkbox"/>	z/OS または OS/390 にインストールされた個々のコンポーネント・ファイルを手動で置換します (EAR ファイルを再パッケージしたりアプリケーションのメタデータを再生成したりすることが、アプリケーションのメンテナンスで必要とされないとき)。	オプション

表 8. 役割および実行する可能性のあるマイグレーション・アクションのチェックリスト
(続き)

✓	実行する可能性のあるマイグレーション・アクション	条件 / コメント
□	WebSphere for z/OS 管理アプリケーションに対する拡張やメンテナンスのために、会話エレメントを変更します。	WebSphere for z/OS J2EE サーバーとこれに関連した J2EE リソースを定義してアクティブにするため、および J2EE アプリケーションをインストールするために 必須
□	製品、システム、または構成の変更のために、z/OS または OS/390 の構成を変更します。このような変更には、アプリケーション・データベースの移動や再構成、WLM サービスの目標の変更などが含まれます。	次のアプリケーションについて 必須 : <ul style="list-style-type: none"> • 役割に基づいたセキュリティーを必要とするアプリケーション • 以前に WebSphere for z/OS スタンダード版 環境で稼働していた Web アプリケーション

第12章 WebSphere for z/OS の新規リリースへのアプリケーションのマイグレーション

WebSphere for z/OS または OS/390 製品の新規リリース、あるいはこれに関連したツールでは、前のリリースにすでにインストールされて稼働しているアプリケーションをマイグレーションすることが必要な場合があります。新規バージョンまたはリリースの拡張機能やサービスのアップデートに適合するか、これらを活用するためにそのアプリケーションをマイグレーションしなければならない場合もあります (別の WebSphere ファミリー環境で現在稼働しているアプリケーションのマイグレーション方法を知るには、119ページの『第13章 WebSphere for z/OS プラットフォームへのアプリケーションのマイグレーション』を参照してください)。

WebSphere for z/OS の新規リリースについて理解するため、および前のリリースで現在稼働しているアプリケーションをマイグレーションするためのサブタスクおよび関連情報を、次の表に示します。

サブタスク	関連情報 (参照項目)
サポートされるマイグレーション・パスを判別する。	<i>WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ</i> , GA88-8652 内のマイグレーション・ロードマップ
あるバージョンまたはリリースによってどのような変更が発生しているかを判別する。	新規 WebSphere for z/OS 製品のリリースの要約。リリースの要約は、 <i>WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ</i> にリストされています。
それぞれの変更がご使用のシステムおよびその既存アプリケーションにどのように影響するかを理解する。	リリースの要約にリストされた変更の説明 (<i>WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ</i> 内)。機能更新の説明では、変更、影響を受ける場合がある処理の領域、マイグレーション作業、新規インターフェース、変更されたインターフェース、および追加詳細情報源を要約しています。
既存アプリケーションを新規リリースにマイグレーションする方法を判別する。	<ul style="list-style-type: none">• 114ページの『z/OS または OS/390 スタンダード版の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ』• 114ページの『OS/390 エンタープライズ版 V3.02 の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ』

z/OS または OS/390 スタンダード版の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ

ご使用のシステムで Web サービス提供のために使用している WebSphere 製品の 1 つに z/OS または OS/390 スタンダード版用の WebSphere Application Server があります。現在使用しているスタンダード版製品のバージョンによっては、z/OS または OS/390 システムのエレメントをアップグレードするとともに、Web アプリケーションのマイグレーション処理を実行しなければならない場合があります。スタンダード版 Web アプリケーションを WebSphere for z/OS J2EE サーバー環境にマイグレーションする方法の追加情報については、<http://www.ibm.com/jp/software/websphere/appserv/> で得られるスタンダード版の文書を参照してください。

OS/390 エンタープライズ版 V3.02 の WebSphere Application Server で稼働しているアプリケーションのマイグレーション・シナリオ

115ページの表9 には、OS/390 エンタープライズ版 V3.02 用の WebSphere Application Server で現在稼働しているアプリケーションについて実行可能なマイグレーションのシナリオが示されています。詳細な情報が必要な場合は、以下の情報源を利用してください。

- Software Development Kit (SDK) のレベル間の違い、および Enterprise bean の仕様間の違いの詳細については、Sun Microsystems Web サイト (java.sun.com)
- WebSphere for z/OS J2EE サーバーでのアプリケーションの作成、アセンブル、およびインストールに関連した作業について、説明やガイドラインを参照する場合は、19ページの『第2部 J2EE サーバー・アプリケーションの作成、アセンブル、および配置』
- WebSphere for z/OS MOFW サーバーでの CORBA アプリケーションの作成、アセンブル、およびインストールに関連した作業について、説明やガイドラインを参照する場合は、*WebSphere Application Server V4.0 for z/OS and OS/390: CORBA アプリケーションのアセンブル*, SA88-8658

注: WebSphere Application Server エンタープライズ版 for OS/390 V3.02 を現在稼働している z/OS または OS/390 のシステム・イメージと同じシステム・イメージまたは同じシスプレックスに WebSphere for z/OS J2EE サーバーをインストールすることはできません。

表9. WebSphere Application Server for OS/390 エンタープライズ版 V3.02 アプリケーションのマイグレーション作業の要約

マイグレーションのシナリオ	説明
V4.0 への エンタープライズ版 V3.02 Enterprise bean のマイグレーション	<p>推奨: Enterprise bean に対する V3.02 のサポートは限定的であったので、V4.0 J2EE サーバーで提供される新規サポートに合うように、Enterprise bean を新たに設計してコーディングすることを検討してください。</p> <p>規則: WebSphere for z/OS V4.0 では、次の 2 種類のサーバー、すなわち新しい J2EE サーバー、および CORBA アプリケーション用の V3.02 マネージド・オブジェクト・フレームワーク (MOFW) を基にしたサーバーがサポートされます。Enterprise bean を V4.0 (MOFW) サーバーで実行することはできません。</p> <p>V3.02 の bean を保存し、これを V4.0 J2EE サーバーのサポートに合うようにアップグレードする必要がある場合は、以下の手順を実行してください。</p> <ol style="list-style-type: none"> 適切な開発ツールに bean のソースをインポートします。 注: ソース・コードにアクセスできない場合、bean JAR ファイルをインポートして、使用すべきでないインターフェースについて調べることができます。該当するものがない場合は、ステップ 4 にスキップしてください。 推奨: 新しい SDK レベルまたは EJB 仕様のために使用すべきでない関数があれば、その関数を除去するようコードを再作成します。 エンティティー bean について、コンテナ管理または bean 管理のパーシスタンスの必要性を検討します。 bean コードを再生成し、その結果の JAR ファイルをエクスポートします。 WebSphere for z/OS アプリケーション組み立てツールを使用してその JAR ファイルをインポートし、bean を Enterprise archive (EAR) ファイルにパッケージします。 WebSphere for z/OS 管理アプリケーションを使用して、EAR ファイルをインストールします。

表9. WebSphere Application Server for OS/390 エンタープライズ版 V3.02 アプリケーションのマイグレーション作業の要約 (続き)

マイグレーション 説明
のシナリオ

V4.0 Enterprise bean へのエンタープライズ版 V3.02 Java ビジネス・オブジェクトのマイグレーション

推奨: Java BO を Enterprise bean にあとで変換する意図で、*WebSphere Application Server エンタープライズ版 for OS/390 Component Broker アプリケーション・アセンブル・ガイド バージョン 3.02*, GA88-7007 に記載されているガイドラインに従うようにアプリケーションを設計し、コーディングした場合にのみ、以下のマイグレーション手順を実行してください。そうでない場合には、その移行作業は、スクラッチから Enterprise bean を新たにコーディングするよりも複雑で困難になる可能性があります。

Java BO を Enterprise bean に変換するには、以下の手順を実行してください。

1. CORBA および EJB プログラミング・モデルと新規の SDK レベルとの違いを理解します。
 2. 適切な開発ツール内に Enterprise bean を新たに作成します。Java BO のビジネス・ロジックをコピーし、必要に応じてそのロジックを変更します。
 - **推奨:** 新しい SDK レベルまたは EJB 仕様のために使用すべきでない関数があれば、その関数を除去するようコードを再作成します。
 - エンティティー bean について、コンテナ管理または bean 管理のパーシスタンスの必要性を検討します。
 - 以下の事項に特に注意して、その他のコードを必要に応じて再作成します。
 - インプリメンテーションの継承
 - その他のオブジェクトまたは CORBA サービスを扱うヘルパー・クラス
 - 環境に基づいて変化することがあるプロパティへの参照を含むコード
 - その他のオブジェクトに関連しているオブジェクトの関連コード
 3. bean コードを生成し、その結果の JAR ファイルをエクスポートします。
 4. WebSphere for z/OS アプリケーション組み立てツールを使用してその JAR ファイルをインポートし、bean を Enterprise archive (EAR) ファイルにパッケージじます。
 5. WebSphere for z/OS 管理アプリケーションを使用して、EAR ファイルをインストールします。
-

表9. WebSphere Application Server for OS/390 エンタープライズ版 V3.02 アプリケーションのマイグレーション作業の要約 (続き)

マイグレーション のシナリオ	説明
V4.0 (MOFW) サー バーで実行するた めの エンタープラ イズ版 V3.02 Java ビジネス・オブジ ェクトのマイグレ ーション	<p>V4.0 サーバーで実行するために Java BO をアップグレードするには、以下の手順を実行してください。</p> <ol style="list-style-type: none"> 適切な開発ツールにソースをインポートします。 <p>規則: Java BO にはオブジェクト・ビルダー 3.5 を使用してください。</p> <ol style="list-style-type: none"> 推奨: 新しい SDK のレベルのために使用すべきでない関数があれば、その関数を除去するようコードを再作成します。 Enterprise bean を エンタープライズ版 V3.02 MOFW サーバーで駆動するために使用されたコードがこのオブジェクトに含まれている場合、以下の機能を提供するコードを変更する必要があります。Enterprise bean は現在 V4.0 J2EE サーバーに常駐している必要があるため、bean を使用するためのプログラミング・モデルは変更されています。 <ul style="list-style-type: none"> 初期コンテキストの設定 ホームの検索と使用 オブジェクト参照子の限定 コードを再生成し、その結果の JAR ファイルをエクスポートします。 アプリケーション・ファイルを z/OS または OS/390 に転送し、all.mak ファイルを実行してそのコードを再コンパイルします。 WebSphere for z/OS 管理アプリケーションを使用して V4.0 (MOFW) サーバーにアプリケーションをインストールします。
V4.0 (MOFW) サー バーで実行するた めの エンタープラ イズ版 V3.02 C++ ビジネス・オブジ ェクトのマイグレ ーション	<p>既存の V3.02 C++ ビジネス・オブジェクトのソース・コードを変更する必要はまったくありません。しかし、コンパイラーまたはオブジェクト・ビルダーに対するサービスの更新や拡張を利用するために、アプリケーション開発ツールによってそのビジネス・オブジェクトを再度利用することを検討してください。</p> <p>規則:</p> <ul style="list-style-type: none"> C++ ビジネス・オブジェクトにはオブジェクト・ビルダー 3.5 を使用してください。 z/OS または OS/390 で C++ ビジネス・オブジェクトを再コンパイルする必要があります。 <p>V4.0 サーバーで実行するために C++ ビジネス・オブジェクトをアップグレードするには、以下の手順を実行してください。</p> <ul style="list-style-type: none"> all.mak ファイルを再実行し、コードを z/OS または OS/390 上で再コンパイルします。 管理アプリケーションを使用して V4.0 (MOFW) サーバーにアプリケーションをインストールします。

表 9. WebSphere Application Server for OS/390 エンタープライズ版 V3.02 アプリケーションのマイグレーション作業の要約 (続き)

マイグレーション のシナリオ	説明
CORBA サーバ ー・アプリケー ションを駆動する 「ネイティブ」 z/OS または OS/390 アプリケー ションのマイグレ ーション	CORBA サーバ・アプリケーションを使用するために CORBA プログラミング・モデルを使用する既存の C++ または Java クライアントを変更する必要はありません。しかし、コンパイラまたはツールに対するサービスのアップデートや拡張を利用するために、アプリケーション開発ツールによってそのクライアントを再度利用することを検討してください。
Enterprise bean を 駆動する「ネイテ ィブ」 z/OS また は OS/390 クライ アント・アプリケー ションのマイグレ ーション	Enterprise bean が J2EE サーバに常駐している必要があるため、bean を使用するためのプログラミング・モデルは変更されています。bean にアクセスするために「ネイティブ」 z/OS または OS/390 クライアント・アプリケーションで CORBA プログラミング・モデルが使用されている場合、ソース・コードを変更する必要があります。その変更を行うには、以下の手順を実行してください。 <ol style="list-style-type: none"> 1. 適切な開発ツールにソースをインポートします。 2. 推奨: SDK の新規レベルのために使用すべきでない関数があれば、その関数を除去するようコードを再作成します。 3. 以下の機能を提供するコードを変更します。 <ul style="list-style-type: none"> • 初期コンテキストの設定 • ホームの検索使用 • オブジェクト参照子の限定 4. コードを再生成し、その結果の JAR ファイルをエクスポートします。 5. コードを z/OS または OS/390 上で再コンパイルします。
分散クライアン ト・アプリケー ションのマイグレ ーション	次の基準の両方に該当するクライアントを変更する必要はありません。 <ul style="list-style-type: none"> • MOFW サーバがインストールされている C++ または Java ビジネス・オブジェクトがクライアントで使用される。 • ワークステーション・プラットフォーム (Windows NT など) 上の WebSphere Application Server エンタープライズ版 Component Broker バージョン 3.0、V3.5、または V3.6 でクライアントが稼働する。

第13章 WebSphere for z/OS プラットフォームへのアプリケーションのマイグレーション

WebSphere ファミリー製品の前のリリースでは、プログラミング・モデルが異なるために分散プラットフォームから OS/390 へのアプリケーションの移植が阻止されていました。しかし、WebSphere for z/OS J2EE サーバーを介して利用できるサポートによって、他の WebSphere 環境から WebSphere for z/OS または OS/390 への J2EE アプリケーション・コンポーネントの移動が比較的簡単になっています。J2EE アプリケーション・コンポーネントのマイグレーションに関する実際上の主な考慮事項は、以下のとおりです。

1. サポートされている仕様レベル間の違い

たとえば、WebSphere for z/OS J2EE サーバーは、Sun Microsystems Java Servlet 2.2 仕様をサポートします。別の WebSphere 環境で稼働している V2.2 サブレットがある場合、コードを変更しないでそのサブレットをマイグレーションすることができます。しかし、V2.1 サブレットを使用している場合は、V2.1 と V2.2 の仕様の違いを理解し、使用すべきでない関数をサブレット・コードから除去する必要があります。

適切な仕様レベルへのコンポーネントのマイグレーションについては、<http://www.ibm.com/jp/software/websphere/appserv/library.html> の **インフォセンター** で得られる説明を利用してください。

2. データベース要件またはアクセス間の違い

たとえば、コンテナ管理パーシスタンスを使用する Enterprise bean をマイグレーションしようとする場合、DB2 をパーシスタント・データ・ストアとして使用していれば、何も変更する必要はありません。しかし、ユーザーのアプリケーションが DB2 に直接アクセスする場合は、分散プラットフォーム上の DB2 と z/OS または OS/390 上の DB2 の違いについて責任を負うことが必要になります。その違いは、JDBC ドライバー名と一部の SQL 構文を変更する必要があるということです。

詳しくは、*z/OS and OS/390 DB2 Application Development Guide* を参照してください。これは、

<http://www.ibm.com/software/db2os390/v7books.html> にあります。

3. アセンブリー、配置、およびインストールのためのプラットフォーム特有のツール

現在、WebSphere for z/OS では、インストールのための J2EE アプリケーションの準備を行うアプリケーション組み立てツールを使用する必要があります。

り、J2EE アプリケーションをインストールする管理アプリケーションを使用する必要があります。マイグレーション作業の一部でこれら 2 つのツールを使用することが必要になります。

表10 には、他の WebSphere ファミリー環境で現在稼働しているアプリケーションについて実行可能なマイグレーションのシナリオが示されています。詳細な情報が必要な場合は、以下の情報源を利用してください。

- Software Development Kit (SDK) のレベル間の違い、および J2EE アプリケーション・コンポーネントの仕様レベル間の違いの詳細については、Sun Microsystems の Web サイト (java.sun.com) を参照してください。
- WebSphere for z/OS J2EE サーバーでのアプリケーションの作成、アセンブル、およびインストールに関連した作業について説明やガイドラインを得る必要がある場合は、19ページの『第2部 J2EE サーバー・アプリケーションの作成、アセンブル、および配置』を参照してください。

表 10. 他の WebSphere ファミリー環境でのアプリケーションのマイグレーション・シナリオの要約

**マイグレーション・説明
シナリオ**

WebSphere Application Server スタンダード版 V3.5 と JSP の移動	<ol style="list-style-type: none"> 1. 必要に応じて、ご使用のワークステーションのアプリケーション開発ソフトウェアをアップグレードします。 2. アプリケーション・コンポーネントを適切な開発ツールにインポートします。次に、 <ul style="list-style-type: none"> • 新しい SDK レベルまたはコンポーネント仕様レベルのために使用すべきでない関数を除去するよう、コードを再作成します。 • Web コンポーネントで DB2 などの J2EE リソースが使用される場合、コードを検討して適切な変更を加えます。 3. コンポーネント・コードを再生成し、Web アプリケーションの Web アーカイブ (WAR) ファイルを作成します。 4. 結果的に作成された WAR ファイルをアプリケーション開発ツールからエクスポートします。 5. WebSphere for z/OS アプリケーション組み立てツールを使用してコンポーネント・ファイルをインポートし、コンポーネントを Enterprise archive (EAR) ファイルにパッケージします。 6. WebSphere for z/OS 管理アプリケーションを使用して、EAR ファイルをインストールします。
---	---

表 10. 他の WebSphere ファミリー環境でのアプリケーションのマイグレーション・シナリオの要約 (続き)

マイグレーション・シナリオ	説明
WebSphere Application Server アドバンスド版 V3.5 からの Enterprise bean、サーブレット、および JSP の移動	<ol style="list-style-type: none"> 1. 必要に応じて、ご使用のワークステーションのアプリケーション開発ソフトウェアをアップグレードします。 2. アプリケーション・コンポーネントを適切な開発ツールにインポートします。次に、 <ul style="list-style-type: none"> • 新しい SDK レベルまたはコンポーネント仕様レベルのために使用すべきでない関数を除去するよう、コードを再作成します。 • エンティティー bean について、コンテナ管理または bean 管理パーシスタンスの要件を検討します。 • Web コンポーネントで DB2 などの J2EE リソースが使用される場合、コードを検討して適切な変更を加えます。 3. コンポーネント・コードを再生成し、Web アプリケーション・コンポーネントの Web アーカイブ (WAR) ファイルまたは Enterprise bean の JAR ファイルを作成します。 4. 結果的に作成された WAR ファイルまたは JAR ファイルをアプリケーション開発ツールからエクスポートします。 5. WebSphere for z/OS アプリケーション組み立てツールを使用してコンポーネント・ファイルをインポートし、コンポーネントを Enterprise archive (EAR) ファイルにパッケージします。 6. WebSphere for z/OS 管理アプリケーションを使用して、EAR ファイルをインストールします。 7. コンポーネントで使用される Enterprise bean ホームの JNDI ルックアップ名を検討し、必須コンポーネントが z/OS で登録されるときの名前とそのルックアップ名が一致することを確認します。

第14章 WebSphere for z/OS J2EE サーバーにすでにインストールされているアプリケーションのアップグレード

WebSphere for z/OS J2EE サーバーに以前にインストールしたアプリケーションに関する作業に該当する注記を、以下に示します。

注:

1. アプリケーションの各バージョンに固有の名前が付けられ、各バージョンが WebSphere for z/OS アプリケーション組み立てツールによって個別にパッケージされてインストールされる限り、同じアプリケーションの複数のバージョンを同じ J2EE サーバーにインストールできます。異なるデータベース・スキーマがアプリケーションの各バージョンで必要とされる場合、アプリケーションごとに別の J2EE リソース接続を定義できます。
2. すでにインストールされているアプリケーションに変更を加える場合、該当のツールを使用してそのアプリケーションを再アセンブルして再インストールする必要がないほど、その変更が小さいものであれば、以下のようにすることができます。
 - a. z/OS または OS/390 HFS 内の JAR ファイルまたは WAR ファイルを置換します。
 - b. z/OS または OS/390 ワークロード管理コマンドを発行し、現行作業をドレーンします。
 - c. 新規サーバー領域を開始します。

スタンダード版サブレット再ロード機能に置き換わるものとして、このプロセスを利用してください。

第5部 付録

付録A. 環境ファイルおよび JVM プロパティー・ファイル

この付録では、以下の参照情報を説明しています。

- 環境ファイルおよび環境変数
- JVM プロパティー・ファイルおよびプロパティー

環境ファイルおよび環境変数

このセクションでは、以下のトピックについて説明します。

- WebSphere for z/OS による環境変数と環境ファイルの管理方法
- ランタイム・サーバー開始プロシージャーによる環境ファイルの指示方法
- OS/390 クライアント向け環境変数
- ランタイム環境変数の構文および意味

注: アプリケーション開発環境に追加の環境変数を設定することが必要になることがあります。

WebSphere for z/OS による環境変数と環境ファイルの管理方法

インストールおよびカスタマイズ中のブートストラップ・プロセスのあとで、WebSphere for z/OS は管理アプリケーションを介して環境データを管理し、システム管理データベースにその環境データを書き込みます。環境変数を追加または変更するには、シスプレックス、サーバー、またはインスタンス・プロパティー形式に環境データのペア (環境変数名とその値) を入力する必要があります。会話を活動化したり、コールド・スタートのための準備を行うと、環境変数データが HFS ファイルに書き込まれます。WebSphere for z/OS によって、環境ファイルを最も明確に指定する値が決定されます。たとえば、サーバー・インスタンスに対する設定は、そのサーバーに対する同じ変数の設定に優先し、サーバーに対する設定は、そのシスプレックスに対する同じ変数の設定に優先します。

管理アプリケーションを介さずに環境ファイルを直接変更すると、行われた変更は、会話を活動化したり、コールド・スタートのために準備したりするときに、すべて上書きされます。

会話を活動化したり、コールド・スタートのための準備を行うと、WebSphere for z/OS は環境データを各サーバー・インスタンスの HFS ファイルに書き込みます。各環境ファイルのパスと名前は、次のようになります。

```
CBCONFIG/controlinfo/envfile/SYSPLEX/SRVNAME/current.env
```

この場合、次のようになります。

CBCONFIG

インストール時に、WebSphere for z/OS による構成データと環境ファイルの書き込み先ディレクトリーとして指定する、読み取りおよび書き込みディレクトリーです。デフォルトは、/WebSphere390/CB390 です。

推奨: システム管理サーバー領域のユーザー ID (BBOCBRAC サンプルでは CBSYMSR1) /WebSphere390/CB390 ディレクトリーの所有者でなければなりません。システム管理サーバー領域によって、このディレクトリーにファイルが書き込まれます。他のサーバー領域のユーザー ID がそのディレクトリーへの読み取りアクセス権を持つように、許可ビットを 775 にする必要があります。

SYSPLEX

ユーザーのシस्पлексの名前です。WebSphere for z/OS は、事前定義された &SYSPLEX JCL 変数からこの名前を派生させます。

SRVNAME

サーバー・インスタンス名です。

環境変数は、WebSphere for z/OS の初期インストールの場合を除き、管理アプリケーションを介して管理する必要があります。初期インストール時には、ブートストラップ・ジョブが使用する初期環境ファイルを変更する必要があります。環境変数ファイルを直接変更しなければならないのは、このときだけです。

したがって、ユーザーのサーバー向けに環境データを定義する明確な状態は、2 つあります。これらの状態に一致するのは、以下のようにして環境データを作成する、2 つの明確な方法です。

1. ブートストラップ・プロセスより前に環境変数をコーディングすることによって、環境データを定義する。この状態では、ここで与えられるサンプルを変更します。ブートストラップ・ジョブによって、ファイルが読み取られ、環境データがシステム管理データベース内に置かれます。HFS で環境ファイルを直接変更するのは、このときだけです。

環境変数の構文については、130ページの『環境変数構文』を参照してください。

2. 管理アプリケーションを介して環境データを定義し、管理する。この状態では、管理アプリケーション内のパネルを介して環境データのペア (環境名とその値 - no "=") を入力します。

ランタイム・サーバー開始プロシージャによる環境ファイルの指示方法

WebSphere for z/OS ランタイム・サーバー開始プロシージャは、構成情報に対する環境ファイルを示す必要があります。開始プロシージャは、HFS ファイルを示す PATH パラメーターを持つ BBOENV DD ステートメントを使用します。BBOENV DD ステートメントは、次のようになります。

```
//BBOENV DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
```

この場合、次のようになります。

&CBCONFIG

開始プロシージャの中に設定する変数です。インストール時に、WebSphere for z/OS による構成データと環境ファイルの書き込み先ディレクトリーとして指定する、読み取りおよび書き込みディレクトリーと一致していなければなりません。デフォルトは、WebSphere390/CB390 です。

&RELPATH

サブディレクトリー (controlinfo/envfile) です。その値は、変更する必要はありません。

&SYSPLEX

ユーザーのシस्पレックスの名前です。事前定義された JCL 変数であるため、開始プロシージャの中で設定する必要はありません。

&SRVNAME

サーバー・インスタンス名です。プロシージャの開始時にサーバー・インスタンス名を指定することによって、他のサーバー・インスタンスに対しても同じ開始プロシージャを使用することができます。

例: サーバー・インスタンス名 BBOASRIA をその開始プロシージャに渡すには、次のように指定します。

```
s bboasr1.bboasr1a,srvname='BBOASRIA'
```

サーバー・インスタンス BBOASR1B に対して同じ開始プロシージャを使用するには、次のように指定します。

```
s bboasr1.bboasr1b,srvname='BBOASR1B'
```

OS/390 クライアント向け環境変数

管理アプリケーションは、OS/390 クライアント向けの環境変数を管理しません。OS/390 クライアント環境ファイルを作成、管理し、クライアント・プログ

ラムからそれらを示す必要があります。132ページの表11 は、OS/390 クライアントに対する必須、またはオプションの環境変数を示しています。

置換変数使用上の注意事項

環境ステートメント内で変数置換 (\$ 変数) を使用することはできません。

UNIX シェル環境で使用される変数置換は、言語環境プログラム (LE) に実装されていません。WebSphere for z/OS は、言語環境プログラム内で環境変数を処理するため、パス環境変数内で \$PATH などの変数を使用すると、失敗します。

例:

UNIX シェル環境は、多くの場合、次のように、既存のパスに新しいパスを追加することによってパスを設定します。

```
PATH=yourdir  
PATH=$PATH/mydir
```

\$PATH 変数を置き換えた後、結果として設定されたパスは、PATH=yourdir/mydir です。しかし、WebSphere for z/OS は言語環境プログラムで環境変数を処理するため、変数が割り当てられていないと、結果として設定されるパスは PATH=\$PATH/mydir になります。

環境変数構文

この構文に従うのは、ブートストラップ・プロセスの前に初期環境ファイルを定義する場合だけです。

規則: 構文規則は以下のとおりです。

- 環境変数の構文は、次のようなパターンになっています。

```
VARIABLE=VALUE
```

この場合、次のようになります。

VARIABLE

環境変数です。

VALUE

変数に対する設定です。説明によって、変数ごとの可能な値が定義されています。

- 変数と値の両方に関して、前後の空白文字 (ブランクまたはタブ) は無視されます。例: 次の 2 行は、結果的に同じになります。

```
VARIABLE1=VALUE1
```

および

```
VARIABLE1 = VALUE1
```

- 変数は空ではないこと、および英字で始まる必要があります。
- 『=』 は必須です。
- 値を空にすることは**できません**。非空白文字を少なくとも 1 つは指定する必要があります。そうでない場合は、環境変数は無視されます。
- ブランク行は無視されます。
- このトピックでの説明に従って、大文字と小文字をコーディングしてください。
- 環境変数をコメント化するには、変数に '#' などの文字を追加するだけです。たとえば、TRACEALL=0 を #TRACEALL=0 に変更することができます。この変数は英字で始まっていないので、このようなコーディングは無視されます。

環境変数の使用

サーバーまたはクライアントごとに、すべての環境変数を使用する必要があるというわけではありません。132ページの表11 は、与えられた環境変数をどこで使用するかを示しています。以下は、各列に表示される内容の意味です。

- 『R』 は必須であることを意味します。
- 『O』 はオプションであることを意味します。
- 『F』 は今後のリリースにおいて必須であることを意味します。
- 「デフォルト」列のブランクは、変数が設定されていないことを意味します。
- その他の列のブランクは、変数が使用されていないことを意味します。

脚注は、表の終わりに示します。

注: デフォルト設定および例では、標準 `_CEE_ENVFILE` 構文を使用します。この構文は、管理アプリケーションで環境データを定義するときには使用しません。

表 11. 環境変数を使用する場所

環境変数 = <デフォルト>	デーモン・サーバー・インスタンス		システム管理サーバー・インスタンス		ネームミング・サーバー・インスタンス		インターフェース・リポジトリ・インスタンス		J2EE サーバー・インスタンス		OS/390 クライアント
	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	
BBOLANG=ENUS	O		O		O		O		O		O
CBCONFIG= /WebSphere390/CB390	R		R		R		R		R		R
CLASSPATH=		O		O		O		O		O ¹	
CLIENT_DCE_QOP= NO_PROTECTION											O
CLIENT_HOSTNAME=											O
CLIENTLOGSTREAMNAME=											O
CLIENT_RESOLVE_IPNAME=<RESOLVE_IPNAMEの値>		O		O		O		O		O	O
CLIENT_TIMEOUT=											
com.ibm.ws.naming.ldap.containerdn=				O		O					
ibm-wsnTree=(1,0=WASNaming,c=us											
com.ibm.ws.naming.ldap.domainname=syplex				O		O					
com.ibm.ws.naming.ldap.masterurl=ldap://localhost:1389				O		O					
DAEMON_IPNAME=	R	O									
DAEMON_PORT=5555	O ²	O ²									
DATA.CTRLHOST=											O
DATA.CTRLPORT=5000											O
DEFAULT_CLIENT_XML_PATH=											O ³
DM_GENERIC_SERVER_NAME= CBDAEMON	O ²	O ²									
DM_SPECIFIC_SERVER_NAME= DAEMON01	O ⁴	O ⁴			O ⁴		O ⁴		O ⁴		
HOME=											O
IBM_OMGSSL=0											O
ICU_DATA= /usr/lpp/WebSphere/bin/		R		R						O	
IR_GENERIC_SERVER_NAME= CBINTFRP		O									
IR_SPECIFIC_SERVER_NAME= INTFRP01	O ⁴	O ⁴			O ⁴		O ⁴		O ⁴		O ⁴
IRPROC=BBOIR	O	O									

表 11. 環境変数を使用する場所 (続き)

環境変数 = <デフォルト>	デーモン・サーバー・インスタンス		システム管理サーバー・インスタンス		ネーミング・サーバー・インスタンス		インターフェース・リポジトリ・インスタンス		J2EE サーバー・インスタンス		OS/390 クライアント
	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	
IVB_DEBUG_ENABLED=											0
IVB_DRIVER_PATH= /usr/lpp/WebSphere		R									
IVB_HOME=											0
java.naming.factory.initial= com.ibm.ws.naming.ldap.WsnLdapInitialContextFactory					0	0					
java.naming.security.credentials= secret					0	0					
java.naming.security.principal= cn=WASAdmin, o=WASNaming, c=us					0	0					
JAVA_COMPILER=											0
JAVA_JEE754=											0 ¹¹
JVM_DEBUG=				0		0		0			0
JVM_HEAPSIZE=256											0
JVM_LOGFILE=											0
LDAPBINDPW=		F				R ⁵					
LDAPCONF=		F				R ⁵					
LDAPHOSTNAME=		F				R ⁵					
LDAPIRBINDPW=		F								R ⁶	
LDAPIRCONF=		F								R ⁶	
LDAPIRHOSTNAME=		F								R ⁶	
LDAPIRNAME=		F								R ⁶	
LDAPIRROOT=		F								R	
LDAPNAME=		F								R ⁵	
LDAPPROOT=		F								R	
LIBPATH=											0 ¹
LOGSTREAMNAME=		0									
MIN_SRS=[MOFW の場合は 0, J2EE の場合は 1]										0	

表 11. 環境変数を使用する場所 (続き)

環境変数 = <デフォルト>	デーモン・サーバー・インスタンス		システム管理サーバー・インスタンス		ネームミニング・サーバー・インスタンス		インターフェース・リポジトリ・インスタンス		J2EE サーバー・インスタンス		OS/390 クライアント
	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	
NM_GENERIC_SERVER_NAME= CBNAMING		0									
NM_SPECIFIC_SERVER_NAME= NAMING01	0 ⁴		0 ⁴		0 ⁴		0 ⁴		0 ⁴		
NMPROC=BBONM	0		0								
OTIS_DEFAULT_TIMEOUT=30	0	0	0	0	0	0	0	0	0	0	
OTIS_MAXIMUM_TIMEOUT=60	0	0	0	0	0	0	0	0	0	0	
PATH=										0	0
RAS_MINORCODEDEFAULT=											
NODIAGNOSTICDATA											
REM_DCEPASSWORD=											0
REM_DCEPRINCIPAL=											0
REM_PASSWORD=	0 ⁷		0 ⁷		0 ⁷		0 ⁷		0 ⁷		0
REM_USERID=	0 ⁷		0 ⁷		0 ⁷		0 ⁷		0 ⁷		0
RESOLVE_IPNAME=	0 ⁸	0 ⁹	0 ⁹	0 ⁹	0 ⁹	0 ⁹	0 ⁹	0 ⁹	0 ⁹	0 ⁹	R ¹⁰
RESOLVE_PORT=900	0	0	0	0	0	0	0	0	0	0	0
SM_DEFAULT_ADMIN= CBADMIN	0		0								
SM_GENERIC_SERVER_NAME= CBSYSMGT	0		0								
SM_SPECIFIC_SERVER_NAME= SYSMGT01	0 ⁴		0 ⁴		0 ⁴		0 ⁴		0 ⁴		
SMPROC=BBOSMS	0		0								
SOMOOSQL=										0	
SRVIPADDR=	0	0	0	0	0	0	0	0	0		
SSL_KEYRING=											0
SYS_DB2_SUB_SYSTEM_NAME=DB2	R	R	R	R	R	R	R	R	R	R	
TRACEALL=1	0	0	0	0	0	0	0	0	0	0	0
TRACEBASIC=	0	0	0	0	0	0	0	0	0	0	0
TRACEBUFFERCOUNT=4	0	0	0	0	0	0	0	0	0	0	0

表 11. 環境変数を使用する場所 (続き)

環境変数 = <デフォルト>	デーモン・サーバー・インスタンス		システム管理サーバー・インスタンス		ネームミニング・サーバー・インスタンス		インフォエース・リポジトリ・インスタンス		J2EE サーバー・インスタンス		OS/390 クライアント
	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	制御領域	サーバー領域	
TRACEBUFFLOC=(Server: BUFFER Client: SYSPRINT)	0	0	0	0	0	0	0	0	0	0	0
TRACEBUFSIZE=IM	0	0	0	0	0	0	0	0	0	0	
TRACEDetail=	0	0	0	0	0	0	0	0	0	0	0
TRACEMINORCODE=											
TRACEPARAM=00	0										

注:

- IMS PAA および CICS PAA を含め、Java を使用するサーバー領域に必要です。
- デーモン・サーバーに対する値を指定する場合は、システム管理サーバー制御領域に対するのと同じ値を指定する必要があります。
- クライアントがシステム管理スクリプト API を使用するときが必要です。
- この値は、シンプレックス内の 2 番目およびそれ以降のシステムに対して指定する必要があります。
- LDAPCONF は、LDAPBINDPW、LDAPHOSTNAME、および LDAPNAME と一緒に使用することはできません。LDAPCONF か、あるいは LDAPBINDPW、LDAPHOSTNAME、および LDAPNAME が必要です。
- LDAPIRCONF は、LDAPIRBINDPW、LDAPIRHOSTNAME、および LDAPIRNAME と一緒に使用することはできません。LDAPIRCONF か、あるいは LDAPIRBINDPW、LDAPIRHOSTNAME、および LDAPIRNAME が必要です。
- サーバーが別のサーバーのリモート・クライアントになるときに使用されます。
- デフォルトは、ブートストラップ中の DAEMON_IPNAME の値です。
- デフォルトは、ローカル・システム IP 名です。通常は、コーディングしないでください。
- デーモン・サーバーがクライアントとして同じシステム上にある場合は、オプションです。この場合、デフォルトはローカル・システム IP 名です。
- OS/390 上で稼働する Java クライアントに必要です。

環境変数の説明

BBOLANG=LANGUAGE

使用される WebSphere for z/OS メッセージ・カタログの名前。デフォルトは、ENUS です。

CBCONFIG=path

会話が活動化されるときに WebSphere for z/OS による構成および環境ファイルの書き込み先となる、HFS ファイルの読み取りおよび書き込みディレクトリーを指定します。制御およびサーバー領域開始プロシーチャーの &CBCONFIG 変数は、この値と一致していなければなりません。この方法で、WebSphere for z/OS は、これらの開始プロシーチャーが実行されたときに、サーバーに対する適切な環境ファイルを検出することができます。デフォルトは、/WebSphere390/CB390 です。

例: CBCONFIG=/WebSphere390/CB390

CLASSPATH=path1:[path2]:...

サーバー領域内の Java ビジネス・オブジェクトが使用するための、Java クラス・ファイル (.jar ファイルおよび classes.zip ファイル) を指定します。ユーザーが Java ビジネス・オブジェクトを使用するとき、Java ビジネス・オブジェクトの .jar ファイルを指定します。CLASSPATH ステートメント全体は、1 行にしなければなりません。

例:

CLASSPATH=/usr/lpp/WebSphere/lib/xerces.jar:...

CLIENT_DCE_QOP= value

ローカル OS/390 クライアントが現行のトランザクション・フローに適用するために使用する、DCE メッセージ・オプションのレベル。通常は、リモート・システム上のサーバーにアクセスする OS/390 クライアントに対する DCE セキュリティーを設定します。サーバーに対する DCE レベルは、管理アプリケーションを介して設定されることに注意してください。

クライアントおよびサーバー上で DCE 認証が使用可能になると、DCE の第三者認証方式を使用したハンドシェーク・メッセージ交換が行われ、それによって、双方の正当性が証明されます。いったん上記の交換が行われると、3 つの保護レベルのうちの 1 つにメッセージが割り振られます。これらのメッセージは、以下に示すように、この環境変数の値です。

NO_PROTECTION

DCE は、メッセージとその応答が正当な送信側からのものであることだけを保証します。これはデフォルトです。

INTEGRITY

DCE は、メッセージが正当な送信側からのものであり、送信側による送信後、一切変更されていないことを保証します。

CONFIDENTIALITY

DCE は、正当な受信側だけが読めるように、メッセージを暗号化します。

CLIENT_HOSTNAME=

同じシステムでデーモンが実行中ではないとき、OS/390 クライアントがそのホスト IP 名を判別できるようにします。クライアント・プログラムが `CBSeriesGlobal::hostName()` メソッドを発行すると、システムは、まず、`CLIENT_HOSTNAME` 環境変数を検査し、この値が設定されている場合この値を戻します。この値が設定されていない場合は、システムは、デーモンが実行中であればそのシステムで実行中のデーモンの IP 名を戻します。デフォルト値は、ヌルです。

例: `CLIENT_HOSTNAME=MYSYS.SYS.COM`

CLIENTLOGSTREAMNAME=*LOG_STREAM_NAME*

OS/390 クライアント ORB がエラー情報を書き込む WebSphere for z/OS エラー・ログ・ストリーム。

例: `CLIENTLOGSTREAMNAME=MY.CLIENT.ERROR.LOG`

CLIENT_RESOLVE_IPNAME=*IP_NAME*

OS/390 クライアント、またはクライアントとして機能しているサーバー領域が、ブートストラップ・サーバーにアクセスするため (つまり、クライアントまたはサーバー領域が解決初期参照メソッドを呼び出すとき) に使用する、インターネット・プロトコル名。デフォルトは、`RESOLVE_IPNAME` 環境変数によって指定された値です。この値は、システム管理サーバー (デフォルト・ブートストラップ・サーバー) に関連したインターネット・プロトコル名です。`RESOLVE_IPNAME` が設定されていない場合、値はクライアントまたはサーバー領域が実行されているシステムになります。

`CLIENT_RESOLVE_IPNAME` 環境変数を使用すると、リモート・システム上で稼働中のブートストラップ・サーバーを指定することができます。これに対して、その他のクライアントは、`RESOLVE_IPNAME` 環境変数で定義されたローカル・ブートストラップ・サーバーを使用します。

注: `CLIENT_RESOLVE_IPNAME` に対する TCP/IP ポート番号は、`RESOLVE_PORT` 環境変数で定義されます。

`CLIENT_RESOLVE_IPNAME` の値は、最大 255 文字までです。

例: CLIENT_RESOLVE_IPNAME=REMHOST

CLIENT_TIMEOUT=*n*

クライアント・メソッドの呼び出しからの応答に対するタイムアウト値を設定します。値は整数で、秒数は 10 倍 (したがって、値 10 は 1 秒) に表示されます。デフォルト値は 0 ですが、これは、タイムアウト値が設定されていないことを表します。

例: CLIENT_TIMEOUT=20

com.ibm.ws.naming.ldap.containerdn=*dn*

WsnName ツリーの開始点。ネーミング・サーバーだけがこの環境変数を使用します。デフォルトは、以下のとおりです。

```
ibm-wsnTree=t1,o=WASNaming,c=us
```

この値は、LDAP 初期設定ファイル (サンプルでは `bboldif.cb`) に指定した値と一致する必要があります。環境変数に関しては大文字と小文字の違いが問題になっても LDAP では問題にならないことに注目してください。`"o=c="` の部分も接尾部として `bboslapd.conf` 内に指定する必要があります。たとえば、次のようになります。

```
suffix "o=WASNaming,c=US"
```

ヒント: 接尾部のステートメントは、提供されているサンプル `bboslapd.conf` 内で次のようになっています。

```
suffix "<ws_rdn>"
```

例:

```
com.ibm.ws.naming.ldap.containerdn=ibm-wsnTree=t1,o=WASNaming,c=us
```

com.ibm.ws.naming.ldap.domainname=*sysplex*

ホスト・ルートを一意的に識別します。また、これは JNDI グローバル・ネーム・スペースを区分化するための基礎となります。ネーミング・サーバーだけがこの環境変数を使用します。デフォルトは、シスプレックス名です。例:

```
com.ibm.ws.naming.ldap.domainname=plex1
```

com.ibm.ws.naming.ldap.masterurl=ldap://*IP_name*:*port*

LDAP サーバー IP 名およびポート番号。ネーミング・サーバーだけがこの環境変数を使用します。デフォルトは、`ldap://localhost:1389` です。例:

```
com.ibm.ws.naming.ldap.masterurl=ldap://wsldap:1389
```

DAEMON_IPNAME=*IP_NAME*

デーモン・サーバーがドメイン・ネーム・サービス (DNS) に登録するイン

ターネット・プロトコル名。WebSphere for z/OS と通信するどの CORBA クライアントにも、この IP 名が必要です。

インストール時には、デーモン・ブートストラップ・プロセスを開始する前に、必ず DAEMON_IPNAME 環境変数を定義する必要があります。定義をしないと、WebSphere for z/OS はエラー・メッセージを発行し、デーモンを終了してしまいます。

ブートストラップ・プロセスは、処理中に、システム管理データベース内のデーモン IP 名を設定します。ブートストラップの後で、WebSphere for z/OS はシステム管理データベース内の値を使用します。ブートストラップの後で、DAEMON_IPNAME 環境変数の値を、システム管理データベース内にある値以外の値に変更することもできます。このような変更を行うと、エラー・メッセージが出されますが、デーモンはシステム管理データベースからのデーモン IP 名を使って初期化を行います。

デーモン・サーバー・インスタンスを同じホスト・クラスターに配置するには、サーバー・インスタンスごとに同じ DAEMON_IPNAME 値をコーディングする必要があります。

規則:

- DAEMON_IPNAME の値は、完全修飾された長い名前にする必要があります。
- 第 1 レベルの修飾子は、1 ~ 18 文字です。
- いったん選択したら、デーモンに対するポートおよび IP 名は変更しないでください。各オブジェクト参照にポートと IP 名が含まれているため、それらを変更すると、既存のオブジェクトにアクセスできなくなってしまうからです。

例: DAEMON_IPNAME=CBQ091.PDL.POK.IBM.COM

DAEMON_PORT=*n*

デーモン・サーバーが要求に対して listen するポート番号。デフォルトは、5555 です。値を指定する場合は、システム管理サーバー制御領域に対する値と同じにする必要があります。

例: DAEMON_PORT=5555

DATA.CTRLHOST=*IP_ADDRESS*

オブジェクト・レベル・トレース・クライアント・コントローラーが実行されているワークステーションの IP アドレスを指定します。IBM 分散デバッグを使ってクライアントおよびサーバー・コンポーネントをデバッグしているときは、このアドレスを使用します。

例: DATA.CTRLHOST=MYHOST.IBM.COM

DATA.CTRLPORT=*n*

オブジェクト・レベル・トレース・クライアント・コントローラーが listen するポートを指定します。IBM 分散デバッガーを使ってクライアントおよびサーバー・コンポーネントをデバッグしているときは、このアドレスを使用します。デフォルトは、5000 です。

例: DATA.CTRLPORT=5000

DEFAULT_CLIENT_XML_PATH=*path*

システム管理スクリプト API で使用されるデフォルトのパラメーター・リストを保持する一連の XML ファイルの位置を指定します。システム管理スクリプト API を使用するクライアントに関して、この環境変数を設定する必要があります。

デフォルトのパラメーター・リストが入っている一連のサンプル XML ファイルが用意されています。インストール後、これらのサンプル・ファイルは、/usr/lpp/WebSphere/samples/smapi に入ります。XML ファイルとパラメーター・リストについては、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理スクリプト API*, SA88-8657 を参照してください。

システム管理スクリプト API のデフォルトの振る舞いは、次の 2 とおりの方法で変更できます。

1. システム管理スクリプト API を呼び出す REXX スクリプト内にパラメーターを明示的に指定する。パラメーターを明示的に指定することにより、IBM 提供の XML サンプルを変更する必要がなくなります。クライアントの環境ファイルの中に次のようにコーディングするだけで済みます。

```
DEFAULT_CLIENT_XML_PATH=/usr/lpp/WebSphere/samples/smapi
```

2. XML ファイルを別のディレクトリーにコピーし (IBM 提供のサンプルは読み取り専用です)、パラメーター・リストに変更を加えたあと、新しいディレクトリーを指すように DEFAULT_CLIENT_XML_PATH を変更する。これらの変更を行うことが必要なのは、システム管理スクリプト API のデフォルトの振る舞いを永続的に変更したい場合だけです。

例: DEFAULT_CLIENT_XML_PATH=/usr/lpp/WebSphere/samples/smapi

DM_GENERIC_SERVER_NAME=SERVER_NAME

デーモン・サーバーのサーバー名。デフォルトは、CBDAEMON です。値を指定する場合は、システム管理サーバー制御領域に対する値と同じにする必要があります。

例: DM_GENERIC_SERVER_NAME=CBDAEMON

DM_SPECIFIC_SERVER_NAME=SERVER_INSTANCE_NAME

デーモン・サーバーのサーバー・インスタンス名。デフォルトは、DAEMON01 です。この環境変数は、シスプレックス内の 2 番目およびそれ以降のシステムのすべてのサーバー・インスタンスに対して指定する必要があります。

例: DM_SPECIFIC_SERVER_NAME=DAEMON01

IBM_OMGSSL=[0 | 1]

CORBA 準拠のセキュリティー・タグのみをサーバーによってエクスポートするかどうかを指定します。値 1 は、CORBA 準拠のタグのみをエクスポートすることを意味します。値 0 (デフォルト) は、CORBA 準拠のタグも準拠しないタグもエクスポートすることを意味します。

サーバーがそのセキュリティーに関して SSL 基本認証のみを使用し、クライアント (CICS またはその他の OEM ORB など) が CORBA 準拠のタグを使用するときは、値 1 を使用してください。これは、サーバーが SSL 基本認証を使用する場合に限ります。サーバーが SSL クライアント認証もサポートする場合は、この変数を設定する必要はありません。

サーバーが SSL 基本認証を使用し、分散プラットフォーム上の WebSphere クライアントまたは WebSphere Application Server エンタープライズ版 for OS/390 V3.02 と相互協調処理を行うときは、値 0 を使用 (または、デフォルトを採用する) してください。

例: IBM_OMGSSL=1

HOME=path

ホーム・ディレクトリーを指定します。この変数は、ユーザーが UNIX シェルにログインすると、セキュリティー・プロダクト・ユーザー・プロファイルから自動的に設定されます。OS/390 上で実行されている C++ または Java クライアントの場合は、IBM 分散デバッガーを使ってビジネス・オブジェクトをデバッグするときに、この変数を /tmp に設定します。

例: HOME=/tmp

ICU_DATA=path

ブートストラップおよびインポート・サーバー・プロセッシング中にシステム管理サーバーが使用する、XML パーサーに必要なバイナリー・ファイ

ルへのパス。デフォルト・ディレクトリーに WebSphere for z/OS コードをインストールした場合は、このパスを変更する必要はありません。デフォルトは、/usr/lpp/WebSphere/bin/ です

例: ICU_DATA=/usr/lpp/WebSphere/bin/

IR_GENERIC_SERVER_NAME=SERVER_NAME

インターフェース・リポジトリー・サーバーのサーバー名。デフォルトは、CBINTFRP です。インターフェース・リポジトリー・サーバーのサーバー領域を機能させるためには、この名前を使ってワークロード管理 (WLM) アプリケーション環境を定義する必要があります。

IR_SPECIFIC_SERVER_NAME=SERVER_INSTANCE_NAME

インターフェース・リポジトリー・サーバーのサーバー・インスタンス名。デフォルトは、INTFRP01 です。この環境変数は、シスプレックス内の 2 番目およびそれ以降のシステムのすべてのサーバー・インスタンスに対して指定する必要があります。

IRPROC=PROC_NAME

インターフェース・リポジトリー・サーバーを開始するためにデーモン・サーバーが使用する開始プロシージャー。デフォルトは、BBOIR です。ユーザーは、独自の開始プロシージャーの名前を指定することができます。その場合は、デフォルトの開始プロシージャーから新規開始プロシージャーに情報をコピーします。

例: IRPROC=BBOIR

IVB_DEBUG_ENABLED=1

この OS/390 クライアントがオブジェクト・レベル・トレース・ランタイムをロードし、オブジェクト・レベル・トレースを使用することを指定します。IBM 分散デバッガーを使ってビジネス・オブジェクトをデバッグしている場合、アプリケーション・サーバーと、OS/390 上で実行されている C++ または Java クライアントに対しては、値 1 を指定する必要があります。

IVB_DRIVER_PATH=path

SMP/E のインストール後に WebSphere for z/OS ファイルが入るディレクトリーの名前。デフォルトは、/usr/lpp/WebSphere です。

例: IVB_DRIVER_PATH=/usr/lpp/WebSphere

IVB_HOME=path

IBM 分散デバッガーがアプリケーション・ソース・コードを検索できる位置を指定します。この環境変数は、オプションです。

JAVA_COMPILER=

ジャストインタイム (JIT) コンパイラーを使用することを指定します。

環境変数を使用する場合、ヌル値 (JAVA_COMPILER=) では JIT コンパイラーがオンになります。その他の値では、JIT コンパイラーはオフになります。

デフォルトでは、OS/390 で稼働している Java 仮想マシン (JVM) は JIT コンパイラーを使用するので、この環境変数を明示的に設定する必要はありません。ただし、Java ビジネス・オブジェクトをデバッグしている場合は、非ヌル値を指定して JIT コンパイラーをオフにしてください。

例: JAVA_COMPILER=

JAVA_IIEE754=EMULATION

OS/390 上の Java クライアントが稼働する Java 仮想マシン (JVM) 用にシステムがロードする正しい実行可能コードを指定します。この環境変数を設定する必要があるのは、OS/390 上で稼働する Java クライアントについてのみです。

java.naming.factory.initial=context

クライアントが使用する初期ネーミング・ファクトリー・コンテキスト。デフォルト値は、com.ibm.ws.naming.ldap.WsnLdapInitialContextFactory です。例:

```
java.naming.factory.initial=com.ibm.ws.naming.ldap.WsnLdapInitialContextFactory
```

java.naming.security.credentials=password

java.naming.security.principal で指定した識別名によって使用されるパスワード。このパスワードは、初期システム・カスタマイズ時に LDAP 初期設定ファイルによってアドミニストレーター・アクセス ID (デフォルトでは WASAdmin) に対して定義されたパスワードと一致する必要があります。bboldif.cb というサンプル LDIF ファイルの中には、IBM 提供の WASAdmin アクセス ID が入っています。デフォルト値は、secret です。例:

```
java.naming.security.credentials=secret
```

推奨: IBM 提供のこのパスワードは、変更する必要があります。

java.naming.security.principal=distinguished_name

WsnName ディレクトリーに書き込みアクセスするために定義する識別名 (ユーザー ID)。読み取り / 書き込みアクセスをすべての JNDI ユーザーに提供する場合にのみ、この識別名を指定してください。この識別名は、初期システム・カスタマイズ時に LDAP LDIF ファイルによってアドミニストレーター・アクセス ID (デフォルトでは WASAdmin) に対して定義され

た識別名と一致する必要があります。bboldif.cb というサンプル LDAP 初期設定ファイルの中には、IBM 提供の WASAdmin アクセス ID が入っています。デフォルト値は cn=WASAdmin,o=WASNaming,c=us です。例:

```
java.naming.security.principal=cn=WASAdmin,o=WASNaming,c=us
```

推奨: WASAdmin アクセス ID を保持することをお勧めします。

JVM_DEBUG=1

OLT for Java オブジェクトをオンまたはオフにして、デバッグを設定します。IBM 分散デバッガーを使って Java オブジェクトをデバッグしている場合、OS/390 上で実行されているアプリケーション・サーバーおよび Java クライアントに対しては、値 1 が必要です。

この変数は、デバッグのために JVM メッセージを SYSOUT に転送する場合にも必要です。JVM_DEBUG=1 に設定して、JVM メッセージングを呼び出します。

JVM_HEAPSIZE=*n*

JVM ヒープの最大サイズ (メガバイト) を設定します。デフォルトは、256 MB です。

例: JVM_HEAPSIZE=256 # specifies a 256 MB heap

JVM_LOGFILE=*filename*

JVM からのメッセージを記録する HFS ファイルを指定します。

推奨: この変数は、単一サーバー環境の場合のみ使用します。JVM_LOGFILE を複数サーバー環境で使用すると、すべてのサーバーが同じファイルに書き込みを行うので、診断目的でこのファイルを使用していて問題が起きることがあります。複数サーバー環境では JVM_DEBUG=1 を使用し、JVM メッセージを特定のサーバーの SYSOUT に送信してください。

LDAPBINDPW=*password*

LDAP サーバーにバインドするためにネーミング・サーバーが使用するパスワード。LDAPNAME と結合して使用されます。

LDAPCONF=*filename*

WebSphere for z/OS が使用する LDAP 構成ファイル。HFS 内のファイルを指定する場合は、引用符を使用しないでください。MVS データ・セットを指定する場合は、そのデータ・セットを単一引用符で囲んでください。

例: LDAPCONF='bbo.s21s1apd.conf'

LDAPHOSTNAME=*name:port*

インターフェース・リポジトリ・サーバーがそのデータ・ストアとして使用する LDAP サーバーのホスト名。

LDAPIRBINDPW=*password*

LDAP サーバーにバインドするためにインターフェース・リポジトリ・サーバーが使用するパスワード。LDAPIRNAME と結合して使用されます。

LDAPIRCONF=*filename*

インターフェース・リポジトリ・サーバーがそのデータ・ストアとして使用する LDAP サーバーによって使用される LDAP 構成ファイル。HFS 内のファイルを指定する場合は、引用符を使用しないでください。MVS データ・セットを指定する場合は、そのデータ・セットを単一引用符で囲んでください。

LDAPIRHOSTNAME=*name:port*

インターフェース・リポジトリ・サーバーがそのデータ・ストアとして使用する LDAP サーバーのホスト名。

LDAPIRNAME

インターフェース・リポジトリ・サーバーが使用する LDAP 入り口名。インターフェース・リポジトリ・サーバーは、この入り口名を使って、データ・ストアとして使用する LDAP サーバーに対してそれ自体を認証します。

LDAPIRROOT=*root*

インターフェース・リポジトリがそのデータをアンカーする LDAP 入り口名。

例: LDAPIRROOT=o=BOSS,c=U

LDAPNAME

ネーミング・サーバーが使用する LDAP 入り口名。ネーミング・サーバーは、この入り口名を使って、データ・ストアとして使用する LDAP サーバーに対してそれ自体を認証します。

LDAPROOT=*root*

ネーミング・サーバーがそのデータをアンカーする LDAP 入り口名。

例: LDAPROOT=o=BOSS,c=US

LIBPATH=*path1:[path2]:...*

階層ファイルシステム (HFS) 内の Java に対する DLL 検索パスを指定します。システム、WebSphere for z/OS、および Java DLL を指定します。

例:

LIBPATH=/db2_install_path/lib:/usr/lpp/java/J1.3/bin:/usr/lpp/java/J1.3/bin/classic:/usr/lpp/WebSphere/lib

ここで、`db2_install_path` は、DB2 for OS/390 をインストールした HFS です。

LOGSTREAMNAME=LOG_STREAM_NAME

ブートストラップ中にデーモンおよびシステム管理サーバーが使用する WebSphere for z/OS エラー・ログ・ストリーム名。システムは、ブートストラップ中に、デーモンおよびシステム管理サーバーに対する環境ファイル内に指定されていないと、次のアルゴリズムを使用してエラー・ログ・ストリーム名を作成します。

1. デーモン・サーバー内の IP 名の最初の修飾子を使用します。
2. 最初の修飾子が 9 文字以上である場合は、修飾子を 8 文字のストリングに分割し、ピリオドで区切ります。
3. 上位修飾子「BBO」を追加します。

たとえば、デーモン IP 名が MYDAEMONSERVER.IBM.COM である場合、アルゴリズムによって、エラー・ストリーム名 BBO.MYDAEMON.SERVER が作成されます。

ブートストラップの後で、管理アプリケーションを介して、シスプレックス全体、サーバー、またはサーバー・インスタンスに対するエラー・ログ・ストリーム名を作成または変更することができます。サーバー・エラー・ログ設定によって一般的な WebSphere for z/OS 設定が、サーバー・インスタンス設定によってサーバー設定が、それぞれ上書きされます。このようにして、サーバーやサーバー・インスタンスに対する一般的なエラー・ロギングを、特定のログ・ストリームに設定することができます。

処理中、指定されたログ・ストリームが見つからなかったり、アクセス不能な場合は、メッセージが表示され、サーバーのジョブ・ログにエラーが書き込まれます。

例: LOGSTREAMNAME=MY.CB.ERROR.LOG

ヒント: ログ・ストリーム名は引用符で囲まないでください。ログ・ストリーム名はデータ・セット名ではありません。

MIN_SRS=nn

サーバー領域がいったん初期化されたあとで実行状態に保たれるサーバー領域の数。つまり、サーバー領域が非アクティブになっても、そのサーバー領域は、シャットダウンするようワークロード管理によって指示されることがなくなります。ワークロードに関する応答時間のためにいくつかのサーバー領域が常に作業を処理できる状態になっていなければならない場合に、この環境変数を使用してしてください。

J2EE サーバーの場合は、デフォルトは 1 です。MOFW サーバーの場合は、デフォルトは 0 です。最大値は 20 です。20 より大きい値を指定した場合、この変数は 20 に設定されます。

WebSphere for z/OS ガーベッジ・コレクションはサーバー領域をリフレッシュする場合があります。しかし、サーバー領域の最小数は、この環境変数に指定した値未満にはなりません。

例: MIN_SRS=2

NM_GENERIC_SERVER_NAME=SERVER_NAME

ネーミング・サーバーのサーバー名。デフォルトは、CBNAMING です。ネーミング・サーバーのサーバー領域を機能させるためには、この名前を使ってワークロード管理 (WLM) アプリケーション環境を定義する必要があります。

例: NM_GENERIC_SERVER_NAME=CBNAMING

NM_SPECIFIC_SERVER_NAME=SERVER_INSTANCE_NAME

ネーミング・サーバーのサーバー・インスタンス名。デフォルトは、NAMING01 です。この環境変数は、シスプレックス内の 2 番目およびそれ以降のシステムのすべてのサーバー・インスタンスに対して指定する必要があります。

例: NM_SPECIFIC_SERVER_NAME=NAMING01

NMPROC=PROC_NAME

ネーミング・サーバーを開始するためにデーモン・サーバーが使用する開始プロシージャー。デフォルトは、BBONM です。ユーザーは、独自の開始プロシージャーの名前を指定することができます。その場合は、デフォルトの開始プロシージャーから新規開始プロシージャーに情報をコピーします。

例: NMPROC=BBONM

OTS_DEFAULT_TIMEOUT=n

デフォルトで、アプリケーション・トランザクションが完了するまでの所要時間 (秒数)。この時間は、current → set_timeout メソッドを介してアプリケーション・トランザクションに独自のタイムアウト値が指定されていない場合に、アプリケーション・トランザクションに対して設定されます。

デフォルトは 30 秒で、最大値は 2147483 秒 (24.85 日) です。ヌルや 0 値は使用しないでください。

注: 会話がアクティブになると、システム管理のサーバー・インスタンスについてのみ特別な処理が実行されます。

- OTS_DEFAULT_TIMEOUT 変数を設定しない場合、これが追加されます。
- OTS_DEFAULT_TIMEOUT の値が 3600 (秒) 未満であれば、3600 に設定されます。

システム管理サーバー・インスタンスは、場合により実行時間の長いトランザクションを実行するので、この特別な処理がシステム管理サーバー・インスタンスについて実行されます。その他のサーバー・インスタンスでは、このような長時間のトランザクションのデフォルトは必要ありません。

例: OTS_DEFAULT_TIMEOUT=30

OTS_MAXIMUM_TIMEOUT=*n*

アプリケーション・トランザクションが完了するまでの最大許容時間 (秒数)。アプリケーションによってより長い時間が割り当てられると、システムはそれを OTS_MAXIMUM_TIMEOUT 値に制限します。

デフォルトは 60 秒で、最大値は 2147483 秒 (24.85 日) です。ヌルや 0 値は使用しないでください。

注: 会話がアクティブになると、システム管理サーバー・インスタンスについてだけ特別な処理が実行されます。

- OTS_MAXIMUM_TIMEOUT 変数を設定しない場合、これが追加されます。
- OTS_MAXIMUM_TIMEOUT の値が 3600 (秒) 未満であれば、3600 に設定されます。

システム管理サーバー・インスタンスは、場合により実行時間の長いトランザクションを実行するので、この特別な処理がシステム管理サーバー・インスタンスについて実行されます。その他のサーバー・インスタンスでは、このような長時間のトランザクションのデフォルトは必要ありません。

例: OTS_MAXIMUM_TIMEOUT=60

PATH=*path*

パスを指定します。OS/390 上で Java をトレースしたり、デバッグするときには、アプリケーション・サーバーの場合だけ、irmtdbgj という実行可能プログラムをパスに含めてください。

RAS_MINORCODEDEFAULT=value

システム例外マイナー・コードについてのドキュメンテーションを集めるためのデフォルトの動作を決定します。IBM サービスを受けている場合にだけ使用してください。

CEEDUMP

コールバックおよびオフセットを取り込みます。

ヒント: システムが CEEDUMP を取るには時間がかかるため、トランザクション・タイムアウトが起こることがあります。たとえば、OTS_DEFAULT_TIMEOUT が 30 秒に設定されていても、CEEDUMP を取るときに 30 秒より長くかかるため、アプリケーション・トランザクションがタイムアウトになることがあります。これを回避するためには、以下のいずれかを行います。

- トランザクション・タイムアウト値を増やす。
- RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA をコーディングする。TRACEMINORCODE が環境ファイル内にないことを確認してください。

TRACEBACK

言語環境プログラムおよび OS/390 UNIX トレースバック・データを取り込みます。

SVCDUMP

MVS ダンプを取り込みます (クライアント内にダンプは作成しません)。

NODIAGNOSTICDATA

デフォルト。CEEDUMP、TRACEBACK、SVCDUMP の設定は収集されません。

注: 結果は、別の環境変数、TRACEMINORCODE の設定によって異なることがあります。TRACEMINORCODE=(ヌル値) および RAS_MINORCODEDEFAULT=TRACEBACK をコーディングすると、トレースバックが得られます。しかし、RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA および TRACEMINORCODE=ALL をコーディングしても、トレースバックが得られます。したがって、RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA を指定しても、TRACEBACK はキャンセルされず、TRACEBACK が収集されないだけです。

REM_DCEPASSWORD=*password*

OS/390 クライアントがシスプレックス外のシステムに要求を出し、SSL タイプ 1 認証が使用されているときに、セキュリティー・コンテキスト内で渡されるリモート DCE プリンシパルのパスワード。パスワードは、パスワードに対する DCE 要件を満たしていなければなりません。

例: REM_DCEPASSWORD=mydcePW

REM_DCEPRINCIPAL=*principal*

クライアントがシスプレックス外のシステムに要求を出し、SSL タイプ 1 認証が使用されているときに、セキュリティー・コンテキスト内で渡されるプリンシパル。このプリンシパルは、ターゲット・サーバー上で定義されていなければなりません。値は、プリンシパルに対する DCE 要件を満たしていなければなりません。

例: REM_DCEPRINCIPAL=myDCEprin

REM_PASSWORD=*password*

クライアントがリモート OS/390 システムに要求を出し、ユーザー ID / パスワード・セキュリティーまたは SSL セキュリティーが使用されているときに、セキュリティー・コンテキスト内で使用されるパスワード。

例: REM_PASSWORD=MYPASSW

REM_USERID=*USER_ID*

クライアントがリモート OS/390 システムに要求を出し、ユーザー ID / パスワード・セキュリティーまたは SSL セキュリティーが使用されているときに、セキュリティー・コンテキスト内で使用されるユーザー ID。

例: REM_USERID=MCOX

RESOLVE_IPNAME=*IP_NAME*

システム管理サーバーがドメイン・ネーム・サービス (DNS) に登録するインターネット・プロトコル。WebSphere for z/OS と通信するどの CORBA クライアントにも、この IP 名が必要です。未設定の場合、解決 IP 名は、プログラムが実行されているシステムです。

規則: RESOLVE_IPNAME の値は完全修飾名にする必要がありますが、255 文字を超えてはいけません。

例: RESOLVE_IPNAME=CBQ091.COMPANY.NY.COM

RESOLVE_PORT=*n*

システム管理サーバーが要求に対して listen するポート番号。デフォルトは、900 です。これは、オブジェクト・リクエスト・ブローカーに対する予約済ポートなので、弊社は、この値を変更しないことをお勧めします。

すでにこのポートを使用しているアプリケーションがある場合は、TCP/IP バインド固有サポートと SRVIPADDR 環境変数を使用することを検討してください。

例: RESOLVE_PORT=900

SM_DEFAULT_ADMIN=USER_ID

管理アプリケーションおよび操作アプリケーションを使用する管理者のユーザー ID。この環境変数は、インストール中にシステム管理ブートストラップによって使用されます。システム管理ブートストラップの実行後にこの環境変数を設定しても、有効になりません。この環境変数を定義しない場合、デフォルトのユーザー ID は CBADMIN になります。このユーザー ID を OS/390 に定義して、適切なセキュリティー権限 (たとえば、RACF 許可および LDAP 許可) を与える必要があります。

注: ブートストラップの実行後は、管理アプリケーションを介してのみ追加の管理者ユーザー ID を定義することができます。これらのユーザー ID は、SM_DEFAULT_ADMIN によって定義されたユーザー ID には置き換わりません。

例: SM_DEFAULT_ADMIN=DUDE

SM_GENERIC_SERVER_NAME=SERVER_NAME

システム管理サーバーのサーバー名。デフォルトは、CBSYSMGT です。システム管理サーバーのサーバー領域を機能させるためには、この名前を使ってワークロード管理 (WLM) アプリケーション環境を定義する必要があります。

例: SM_GENERIC_SERVER_NAME=CBSYSMGT

SM_SPECIFIC_SERVER_NAME=SERVER_INSTANCE_NAME

システム管理サーバーのサーバー・インスタンス名。デフォルトは、SYSMGT01 です。この環境変数は、シスプレックス内の 2 番目およびそれ以降のシステムのすべてのサーバー・インスタンスに対して指定する必要があります。

例: SM_SPECIFIC_SERVER_NAME=SYSMGT01

SMPROC=PROC_NAME

システム管理サーバーを開始するためにデーモン・サーバーが使用する開始プロシージャ。デフォルトは、BBOSMS です。ユーザーは、独自の開始プロシージャの名前を指定することができます。その場合は、デフォルトの開始手順から新規開始プロシージャに情報をコピーします。

例: SMPROC=BBOSMS

SOMOOSQL=value

文字列属性についてオブジェクト指向 SQL 照会を使用するクライアント・アプリケーションのパフォーマンスを向上させます。SOMOOSQL=1 を使用することによって、文字列比較がデータベースにプッシュ・ダウンされます。

デフォルト値は、ヌル (SOMOOSQL=) です。

規則: SOMOOSQL=1 を使用できるのは、データベースとサーバー領域アドレス・スペースが同じロケール内で実行するように宣言されている場合だけです。

SRVIPADDR=IP_ADDRESS

クライアント接続要求を listen するために WebSphere for z/OS サーバーが使用する、ドット付き 10 進形式の IP アドレス。

この IP アドレスは、サーバーが TCP/IP にバインドするときに使用します。通常、サーバーは、ローカル TCP/IP スタックに構成されたすべての IP アドレスで listen します。しかし、作業を分離したり、同じポート上で複数のサーバーが listen できるようにしたいときは、SRVIPADDR を使用することができます。指定された IP アドレスは、WebSphere for z/OS がインバウンド要求を受信するときに使用する唯一の IP アドレスになります。通常、デーモン IP 名、解決 IP 名、またはユーザーが使用しているサーバーのホスト名も、この特定の SRVIPADDR にマップする必要があります。

SSL_KEYRING=keyring

SSL 処理において使用される OS/390 クライアントの鍵リングの名前。鍵リングは、RACF 内に置く必要があります。

例: SSL_KEYRING=IVPRING

SYS_DB2_SUB_SYSTEM_NAME=NAME

データベースに接続するために、デーモンおよびシステム管理サーバーが使用する DB2 for OS/390 名。DB2 for OS/390 サブシステム名かグループ接続名のいずれかを使用してください。デフォルトは、DB2 です。デフォルトがユーザーのインストールに適さない場合は、正しい値に合わせて環境変数を変更してください。

例: SYS_DB2_SUB_SYSTEM_NAME=DB21

TRACEALL=n

WebSphere for z/OS に対するトレース・レベルを指定します。有効な値とその意味は、以下のとおりです。

0 トレースなし。

- 1 例外トレース。デフォルトです。
- 2 基本および例外トレース。
- 3 基本および例外トレースを含めた、詳細トレース。

WebSphere for z/OS サブコンポーネントに対するトレース・レベルを設定するために、この変数を TRACEBASIC および TRACEDETAIL 環境変数と結合して使用します。この変数は、IBM サービス技術員の指示があるまで変更しないでください。

例: TRACEALL=1

TRACEBASIC=*n* | (*n*,...)

特定の WebSphere for z/OS サブコンポーネントに対するトレース・オーバーライドを指定します。番号で指定されたサブコンポーネントは、基本および例外トレースを受け取ります。複数のサブコンポーネントを指定する場合は、括弧を使って番号とコンマを区切ってください。サブコンポーネント番号とその意味については、IBM サービスにお問い合わせください。WebSphere for z/OS のその他のパーツは、TRACEALL 環境変数で指定されたとおりにトレースを受け取ります。TRACEBASIC は、IBM サービス技術員の指示があるまで変更しないでください。

例: TRACEBASIC=3

TRACEBUFFCOUNT=*n*

割り振るトレース・バッファの番号を指定します。有効な値は、4 ~ 8 です。デフォルトは、4 です。

TRACEBUFFLOC=SYSPRINT | BUFFER

トレース・レコードの移動先を、sysprint (SYSPRINT) またはメモリー・バッファ (BUFFER) のいずれかに指定し、その後、CTRACE データ・セットに指定します。デフォルトは、トレース・レコードをクライアントの sysprint に送信し、さらに他のすべての WebSphere for z/OS プロセスのためのバッファに送信します。サーバーの場合は、片方または両方の値を、スペースで区切って指定することができます。クライアントの場合は、TRACEBUFFLOC=SYSPRINT しか指定できません。

例: TRACEBUFFLOC=SYSPRINT BUFFER

TRACEBUFFSIZE=*n*

単一トレース・バッファのサイズ (バイト) を指定します。文字「K」(K バイト) または「M」(M バイト) を使用することができます。有効な値は、128K~4M です。デフォルトは、1M です。

TRACEDETAIL=*n* | (*n*,...)

特定の WebSphere for z/OS サブコンポーネントに対するトレース・オーバーライドを指定します。番号で指定されたサブコンポーネントは、詳細トレースを受け取ります。複数のサブコンポーネントを指定する場合は、括弧を使って番号とコンマを区切ってください。サブコンポーネント番号とその意味については、IBM サービスにお問い合わせください。

WebSphere for z/OS のその他のパーツは、TRACEALL 環境変数で指定されたとおりにトレースを受け取ります。TRACEDETAIL は、IBM サービス技術員の指示があるまで変更しないでください。

例:

```
TRACEDETAIL=3  
TRACEDETAIL=(3,4)
```

TRACEMINORCODE=*value*

システム例外マイナー・コードのトレースバックを使用可能にします。IBM サービスから指示があった場合にだけ使用してください。値は、以下のとおりです。

ALL|all

すべてのシステム例外マイナー・コードに対するトレースバックを使用可能にします。

minor_code

特定のマイナー・コードに対するトレースバックを使用可能にします。コードは、X'C9C21234' などの 16 進数で指定してください。

(null value)

デフォルト。この設定では、トレースバックは収集されません。

注: 結果は、別の環境変数 RAS_MINORCODEDEFAULT の設定によって異なることがあります。TRACEMINORCODE=ALL と RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA をコーディングすると、トレースバックが得られます。しかし、TRACEMINORCODE=(ヌル値) と RAS_MINORCODEDEFAULT=TRACEBACK をコーディングしても、トレースバックが得られます。したがって、TRACEMINORCODE=(ヌル値) を指定しても、TRACEBACK はキャンセルされず、TRACEBACK が収集されないだけです。

TRACEPARAM=*SUFFIX* | *MEMBER_NAME*

CTRACE PARMLIB メンバーを識別します。値は、2 文字の接尾部 (ストリング CTIBBO に追加されて、PARMLIB メンバーの名前を作成する) であるか、PARMLIB メンバーの完全指定名のいずれかです。たとえば、接

尾部「01」を使用することができますが、システムはこれを「CTIBBO01」に解決されます。完全指定名は、CTRACE PARMLIB メンバーに対する命名要件を満たしていなければなりません。詳細については、*z/OS MVS 診断: ツールおよび保守援助プログラム*, GA88-8561 を参照してください。

デフォルト値は、00 です。

この環境変数が指定されていて、PARMLIB メンバーが見つからないときは、デフォルトの PARMLIB メンバーである CTIBBO00 が使用されます。指定された変数も、デフォルトの PARMLIB メンバーも見つからないときは、トレースが CTRACE に定義されますが、CTRACE 外部書き出しプログラムには接続されません。PARMLIB メンバーと CTRACE 外部書き出しプログラムの詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断*, GA88-8655 を参照してください。

デーモン・サーバーは、この環境変数を認識する唯一のサーバーであることに注意してください。

例: TRACEPARAM=01

JVM プロパティーおよびプロパティー・ファイル

プロパティー・ファイルを使用するのは、サーバーで稼働する Java 仮想マシン (JVM) 用に WebSphere for z/OS が使用するデフォルトの設定値を変更したい場合だけにしてください。WebSphere for z/OS 管理アプリケーションを介して設定する環境変数が、このファイルに定義するプロパティーの設定値によって変更されないよう、注意してください。

JVM プロパティーは環境変数に似ており、構文も似ています。しかし、大文字小文字の区別は有効です。

JVM プロパティーの構文は、次のようなパターンになっています。

```
property=value
```

この場合、次のようになります。

property

JVM プロパティー

value

プロパティーの設定値。この記述によって、プロパティーごとに設定可能な値を定義します。

注: この値は、引用符で囲まないでください。

JVM プロパティの管理方法

あるサーバーの JVM のデフォルト・プロパティを変更するには、そのサーバーの環境変数設定値が含まれている `current.env` ファイルを WebSphere for z/OS が入れるディレクトリーと同じ HFS ディレクトリーの中に、ファイルを作成してください。

`CBCONFIG/controlinfo/envfile/SYSPLEX/SRVNAME/`

この場合、次のようになります。

CBCONFIG

WebSphere for z/OS が構成データおよび環境ファイルを書き込むディレクトリーとしてインストール時に指定する読み取り / 書き込みディレクトリーです。

SYSPLEX

ユーザーのシスプレックスの名前です。

SRVNAME

サーバー・インスタンス名です。

注: このサーバーを含む会話がはじめてアクティブになるときまでは、このサブディレクトリーは存在しません。

規則:

- ファイルには `jvm.properties` と命名する必要があります。
- この HFS ディレクトリーの許可ビットは、サーバー領域のユーザー ID でこのディレクトリーに読み取りアクセスできるよう、775 にする必要があります。

JVM プロパティの使用

157ページの表12 には、WebSphere for z/OS サーバーについてサポートされている JVM プロパティが示されています。表の内容について、次のリストで説明します。

- 『O』 はオプションであることを意味します。
- 「デフォルト」列のブランクは、変数が設定されていないことを意味します。
- その他の列のブランクは、変数が使用されていないことを意味します。

表 12. JVM プロパティーが使用される場所

JVM property=<デフォルト>	J2EE サーバー・ インスタンス	MOFW サーバー・ インスタンス
	サーバー領域	サーバー領域
com.ibm.ws390.trace.settings=	O	O
com.ibm.ws390.webcontainer.configfile=	O	

プロパティーの説明

com.ibm.ws390.trace.settings=*path/file*

トレース設定ファイルの完全修飾ディレクトリー・パスおよびファイル名。

トレース設定の詳細については、以下を参照してください。

- 98ページの『Java アプリケーションのメッセージとトレース要求をログに記録するための z/OS または OS/390 環境を作成するステップ』
- *WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断*, GA88-8655

例:

```
com.ibm.ws390.trace.settings=/mydir/trace.settings
```

com.ibm.ws390.webcontainer.configfile=*path/file*

Web コンテナ構成情報用のオプション・ファイルの完全修飾ディレクトリー・パスおよびファイル名。デフォルトの `webcontainer.conf` ファイルを変更したい場合にのみ、このプロパティーを指定してください。

ユーザー独自の Web コンテナ構成ファイルを使用する方法については、Web アプリケーション・サポートを WebSphere for z/OS で使用可能にする方法に関する資料を参照してください。その資料は、<http://www.ibm.com/jp/software/websphere/appserv/> から入手できます。

例:

```
com.ibm.ws390.webcontainer.configfile=/usr/lpp/WebSphere/AppServer/bin/WebCon2.conf
```

付録B. 特記事項

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミングまたはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品のみが使用可能であることを意味するものではありません。IBM 製品、プログラム、またはサービスに代えて、IBM の有効な知的所有権またはその他の法的に保護された権利を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM によって明示的に指定されたものを除き、他社の製品と組み合わせた場合の操作の評価と検証はお客様の責任で行っていただきます。

IBM は、本書で解説されている主題について特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書の提供は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用権等を許諾することを意味するものではありません。

〒 106-0032 東京都港区六本木 3 丁目 2-31

AP 事業所

IBM World Trade Asia Corporation

Intellectual Property Law & Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書に対して、周期的に変更が行われ、これらの変更は、文書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するもので

はありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。また、IBM 以外の製品に関するパフォーマンスの正確性、互換性、またはその他の要求は確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書で使用している例について

本書で使用している例は、IBM が作成した単なるサンプルです。これらの例は、標準または IBM 製品の一部ではなく、単に、お客様のアプリケーション開発を支援することを目的として提供されています。例は、「現存するままの」状態で提供されています。IBM は、これらの例の機能またはパフォーマンスに関して、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含む明示もしくは黙示の保証責任を負わないものとします。IBM は、これらの例の使用によって生じたいかなる損害に対しても、たとえ、そのような損害の可能性を通知している場合であっても、その責任を負いません。

これらの例は、上記の免責条項をそのまま適用することを条件として、無料で配布し、複製し、改変し、他のソフトウェアに取り込むことができます。

プログラミング・インターフェース情報

本書には、お客様が WebSphere for z/OS のサービスを受けるためのプログラムを作成することを目的とした、プログラミング・インターフェースについての情報が含まれています。

商標

以下は、米国およびその他の国における IBM Corporation の商標または登録商標です。

AIX
CICS
DB2
IBM
IMS
IMS/ESA
Language Environment
Open Class
OS/390
RACF
VisualAge

VTAM
WebSphere
z/OS

Java およびすべての Java 関連の商標は、Sun Microsystems, Inc. の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT、および Windows ロゴは Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group が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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

用語集

本書で使用される用語の詳細については、以下のソースのいずれかを参照してください。

- *WebSphere Application Server* エンタープライズ版 *Component Broker* 用語集, SD88-7380。これは、インターネット上の以下のサイトにありません。

<http://www.ibm.com/jp/software/websphere/appserv/>

- Sun Microsystems Glossary of Java Technology-Related Terms。これは、インターネット上の以下のサイトにあります。

<http://java.sun.com/docs/glossary.html>

探している用語が見つからない場合、*IBM Glossary of Computing Terms* を参照してください。これは、インターネット上の以下のサイトにあります。

<http://www.ibm.com/ibm/terminology/>

またはインターネット上の以下の Sun Web サイトを参照してください。

<http://www.sun.com/>

索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

[ア行]

インターフェース・リポジトリ・サーバー
開始プロシージャ 142
サーバー名 142
サーバー・インスタンス名 142
エクスポート / インポート・プロセス
サーバー・アプリケーションを実動システムに移動するための 79
エラー・ログ・ストリーム
環境変数 133, 137, 146
クライアント 132, 137

[カ行]

会話
管理アプリケーションによる開始のステップ 56
管理アプリケーションによるコミット
のステップ 64
管理アプリケーションによる妥当性検査
のステップ 64
解決ポート 150
環境変数
ランタイム環境変数
参照 127
DB2 for OS/390 134, 152
OS/390 クライアント向けの
参照 127

管理アプリケーション タスク
会話の開始
のステップ 56
新規会話のコミット
のステップ 64
新規会話の妥当性検査
のステップ 64
新規サーバー構成の活動化
のステップ 65
J2EE サーバー インスタンスの追加
のステップ 59
J2EE サーバー の追加
のステップ 57
J2EE サーバーへのサーバー・アプリケーションのインストール
のステップ 61
J2EE リソースの追加
のステップ 59
J2EE リソース・インスタンスの追加
のステップ 60
z/OS または OS/390 タスクの完了のマーク付け
のステップ 65

管理アプリケーション・タスク
管理アプリケーションの始動
のステップ 56
管理および操作アプリケーション
CBADMIN 151

[サ行]

サーバー構成
管理アプリケーションによる活動化
のステップ 65
サーバー・アプリケーション
実動システムへの移動
エクスポート / インポート・プロセス 79

システム管理サーバー
開始プロシージャ 151
サーバー名 151
サーバー・インスタンス名 151
ポート 150
IP 名 150
システム管理スクリプト API
DEFAULT_CLIENT_XML_PATH 140
システム・ロガー 132, 133, 137, 146
シスプレックス・システム
環境変数 128
セキュリティ
環境変数 134, 150
クライアントの設定 136
リモート DCE パスワード 150
リモート DCE プリンシパル 150
リモート・パスワード 150
リモート・ユーザー ID 150
Lightweight Directory Access Protocol (LDAP) 144
セキュリティ・サーバー (RACF)
リモート・パスワード 134, 150
リモート・ユーザー ID 134, 150

[タ行]

デーモン
サーバー名 140
サーバー・インスタンス名 141
ポート 139
IP 名 138
トレース・データ
Java アプリケーションのロギング 83

[ナ行]

ネーミング・サーバー
開始プロシージャ 147

ネーミング・サーバ (続き)

サーバ名 147

サーバ・インスタンス名 147

ルート・ネーミング・コンテキスト
133, 145

[ハ行]

分散コンピューティング環境 (DCE)

クライアントの設定 136

[マ行]

メッセージ

Java アプリケーションのログイン
83

[ラ行]

ランタイム環境

環境変数 127

ルート・ネーミング・コンテキスト
133, 145

D

DB2 for OS/390

環境変数 134, 152

H

HFS ディレクトリー 128

J

J2EE アプリケーション

J2EE サーバーへのインストール
のステップ 61

J2EE サーバー

管理アプリケーションによる追加
のステップ 57

J2EE サーバー インスタンス

管理アプリケーションによる追加
のステップ 59

J2EE リソース

管理アプリケーションによる追加
のステップ 59, 60

Java アプリケーション

ログイン・メッセージおよびトレ
ース・データ 83

L

Lightweight Directory Access Protocol
(LDAP)

環境変数 133, 144

O

OS/390 作業

新規アプリケーション・サーバ
の

セキュリティー・プロファイ
ルと許可の定義 50

データベース・リソースの設
定 50

命名規則の使用 50

WLM アプリケーション環境
の定義 50

OS/390 タスク

完了のマーク付け
のステップ 65

S

Secure Sockets Layer (SSL)

環境変数 134, 150

T

TCP/IP

解決 IP 名 150

解決ポート 150

クライアント解決 IP 名 132

サーバ IP アドレス 152

Z

z/OS タスク

完了のマーク付け
のステップ 65



プログラム番号: 5655-F31

Printed in Japan

SA88-8654-00



日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12