

IBM WebSphereApplication Server - Express
バージョン 5.1



マイグレーション・ガイド

注!

本書および本書で紹介する製品をご使用になる前に、71 ページの『特記事項』に記載されている情報をお読みください。

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

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

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

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

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： IBM WebSphere
Application Server - Express Version 5.1
Migration Guide

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2003.12

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, 2003. All rights reserved.

© Copyright IBM Japan 2003

目次

第 1 章 WebSphere Application Server - Expressマイグレーション・ガイドの概要	1
第 2 章 実動サーバーのマイグレーション	3
マイグレーション	3
マイグレーションと共存	4
マイグレーション・ツール	5
WASPreUpgrade コマンド	6
WASPostUpgrade コマンド	7
マイグレーションにおける構成のマッピング	9
構成データの手動マイグレーション	11
V3.5.x から V5.1 へのマイグレーション	12
V3.5.x からリモート V5.1 マシンへのマイグレーション	14
V5.0.x から V5.1 へのマイグレーション	17
V5.0.x からリモート V5.1 マシンへのマイグレーション	17
サポートされていないオペレーティング・システムからのマイグレーション	19
第 3 章 IBM WebSphere Studio Site Developer バージョン 5.1 からのマイグレーション	21
サーバーのターゲット・サポートを使用するための J2EE プロジェクトのマイグレーション	21
使用可能にされたサーバーのターゲット・サポートとの後方互換性	21
ウィザード生成に必要な JDK 1.4 の Java パッケージ	22
第 4 章 IBM WebSphere Studio Site Developer バージョン 5 またはバージョン 5.0.1 からのマイグレーション	23
バージョン 5、バージョン 5.0.1、およびバージョン 5.1 での WebSphere Studio Workbench	23
IBM WebSphere Studio Site Developer バージョン 5.0 ワークスペースでのバージョン 5.1.1 の使用	23
バージョン 5 またはバージョン 5.0.1 からの Java プロジェクトのマイグレーション	24
ソース・コード管理 (SCM) システムを使用したバージョン 5 またはバージョン 5.0.1 とバージョン 5.1.1 のプロジェクトの共用	24
Web プロジェクトのマイグレーション	26
Struts 1.1 への Web プロジェクトの変換	27
Web サービス・ツールの変更	27
プロファイル作成ツール内で行った変更	28
テンプレート・ウィザードに関する既知の互換性の問題	28

第 5 章 IBM WebSphere Studio Site Developer バージョン 4.0.x からのマイグレーション	29
IBM WebSphere Studio Site Developer バージョン 4.0.x とバージョン 5 の間の相違点	29
WebSphere Application Server の変更および Servlet/JSP 変換ツール	30
バージョン 4.0.3 からの内部変更	31
循環プロジェクトの依存関係はデフォルトでビルドしない	31
バージョン 5 Web プロジェクトはバージョン 4.0.3 とソース・ロケーションの互換性がある	31
IBM WebSphere Studio Site Developer の Web プロジェクトの構造	32
静的プロジェクトと動的プロジェクトの対比	32
HTML と JSP の相違点	32
ソフトウェア構成管理 (SCM) システムを使用したプロジェクトのマイグレーション	33
CVS または Rational ClearCase を使用したプロジェクトのマイグレーション	33
EAR およびサーバー構成の絶対パス参照のマイグレーション後の除去	35
その他の SCM を使用したプロジェクトのマイグレーション	35
プロジェクトのエクスポートおよびインポートによるマイグレーション	36
既存のバージョン 4.0.x ワークスペースを使用したプロジェクトのマイグレーション	36
EAR およびサーバー構成の絶対パス参照のマイグレーション後の除去	37
既知の問題および制限事項	37
4.0.3 Web プロジェクトのリレーショナル・データのマイグレーション	38
4.0.x から Web サービス・ファイルをインポートした後の WSDL エラー	39
J2EE プロジェクト構造および/または J2EE 仕様レベルのマイグレーション	39
第 6 章 WebSphere Studio "Classic" から IBM WebSphere Studio Site Developer へのマイグレーション	41
マイグレーションのための新規シングル・サーバー・ステージの作成	42
Web 構成記述子ファイルの作成	42
マイグレーション JAR ファイルのエクスポート	43
マイグレーション JAR ファイルの IBM WebSphere Studio Site Developer へのインポート	43
テスト・サーバー上でマイグレーション済みアプリケーションのテスト	44

第 7 章 VisualAge for Java から IBM WebSphere Studio Site Developer へのマイグレーション 45

VisualAge for Java と IBM WebSphere Studio Site Developer の相違点	45
VisualAge for Java からのマイグレーション	46
VisualAge for Java からの Java ファイルおよびプロジェクト・リソース・ファイルのエクスポート	46
IBM WebSphere Studio Site Developer の始動、およびコードに含める新しいプロジェクトの作成	47
Java ファイルとリソース・ファイルの IBM WebSphere Studio Site Developer へのインポート	47
サーブレットが正しく定義されていることを確認するための web.xml エディターの使用 (Web プロジェクトのみ)	48
プロジェクト設定とワークスペース設定のマイグレーション	48
WebSphere V4 テスト環境のセットアップとマイグレーション済みアプリケーションのテスト	50
IBM WebSphere Studio Site Developer からリモート WebSphere Application Server へのアプリケーションのデプロイ	50
複数の開発者間での IBM WebSphere Studio Site Developer プロジェクト設定の共用 (事後マイグレーション)	51
IBM WebSphere Studio Site Developer のチーム・サポート	51

第 8 章 VisualAge for Java Visual Composition Editor から Visual Editor for Java へのマイグレーション 53

VisualAge for Java からの拡張設計時メタデータの保管	53
マイグレーションの完了 (WebSphere Studio ヘインポート)	54

第 9 章 ビルド・セットアップ (ライブラリー、JAR、従属プロジェクト JAR、Ant ビルド) 55

Java ライブラリー JAR およびサード・パーティー外部 JAR	55
Web プロジェクトでサード・パーティー JAR を使用するための推奨される方法	55

複数の Web プロジェクトでサード・パーティー JAR を使用するための推奨される方法	56
外部 JAR ファイルを使用するための代替方法 (グローバル・ビルドおよびサーバー・クラスパス)	56
従属プロジェクト JAR を使用したマルチプロジェクト・ビルドの最適化	57
Ant を使用した自動化実動ビルド	57

第 10 章 マイグレーションの例 59

VisualAge for Java JSP/Servlet サンプル (LeapYear)	59
VisualAge for Java からのファイルのエクスポート	59
新規 IBM WebSphere Studio Site Developer Web プロジェクトの作成	60
Java ファイルとプロジェクト・リソース・ファイルの IBM WebSphere Studio Site Developer プロジェクトへのインポート	60
サーブレットの定義と再構築したアプリケーションの変更	61
IBM WebSphere Studio Site Developer サーバー・プロジェクトの作成	62
マイグレーション済みの LeapYear アプリケーションのテスト	62
例: WebSphere Studio "Classic" バージョン 4.0 Web アプリケーション (YourCo)(Windows)	63
WebSphere Studio "Classic" バージョン 4.0 の始動および新規マイグレーション・ステージの作成	64
Web 構成記述子ファイルの作成	64
マイグレーション・ファイルの作成	64
IBM WebSphere Studio Site Developer の始動と WAR ファイルのインポート	64
IBM WebSphere Studio Site Developer サーバー・プロジェクトの作成	65
マイグレーション済みの YourCo アプリケーションのテスト	66

第 11 章 その他の資料 69

特記事項	71
プログラミング・インターフェース情報	73
商標	73

第 1 章 WebSphere Application Server - Expressマイグレーション・ガイドの概要

IBM® WebSphere® Application Server - Express バージョン 5.1 では、以下からコードをマイグレーションできます。

- VisualAge for Java
- IBM WebSphere Application Single Server Edition バージョン 3.5
- IBM WebSphere Application Server - Express バージョン 5.0、5.0.1、5.0.2
- WebSphere Studio "Classic"
- IBM WebSphere Studio Site Developer
- IBM WebSphere Studio Site Developer バージョン 5 ベータ、早期出荷版または一般出荷版
- IBM WebSphere Studio Site Developer バージョン 5.0.1
- IBM WebSphere Studio Site Developer バージョン 5.1

WebSphere Application Server - Express 5.1 は、 WebSphere Application Server 5.1 と WebSphere Studio Site Developer 5.1.1 から構成されています。以下の最初の章では、WebSphere Application Server - Express のサーバー・フィーチャーのマイグレーションについて説明します。マイグレーション・ガイドの残りの部分では、異なるバージョンの WebSphere Studio Site Developer からのコードのマイグレーションについて説明します。

サーバーのマイグレーションに関する重要な注記

ご使用のサーバー構成のマイグレーションは、管理コンソールを使用してサーバーを管理している場合のみ意味があります。すなわち、通常は実稼動環境の場合です。このオペレーション・モードでは、サーバー構成とデプロイされたアプリケーションはサーバーの構成ディレクトリに格納されます。マイグレーション・プロセスでは、これらの構成ファイルおよびアプリケーション・ファイルをマイグレーションします。一方、WebSphere Studio Site Developer を使用してアプリケーションを構成し、リモート・サーバーにデプロイしている場合は、サーバー構成ファイルをマイグレーションする必要はありません。その理由は、構成ファイルおよびアプリケーション・ファイルはすべて、Studio Site Developer のワークスペースに維持されているからです。ワークスペースは Studio Site Developer によってマイグレーションされます。このマイグレーションの後に、WebSphere Application Server - Express 5.1 サーバーの新しいインスタスを定義し、Studio Site Developer からのアプリケーションの構成およびデプロイを続行することができます。

本書は、以下の章から構成されています。

- 実動サーバーのマイグレーション
- IBM WebSphere Studio Site Developer バージョン 5.1 からのマイグレーション
- IBM WebSphere Studio Site Developer バージョン 5 またはバージョン 5.0.1 からのマイグレーション

- IBM WebSphere Studio Site Developer バージョン 4.0.x からのマイグレーション
-  WebSphere Studio "Classic" から IBM WebSphere Studio Site Developer へのマイグレーション
- VisualAge for Java から IBM WebSphere Studio Site Developer へのマイグレーション
- VisualAge for Java Visual Composition Editor から Visual Editor for Java へのマイグレーション
- ビルド・セットアップ (ライブラリー JAR、従属プロジェクト JAR、Ant ビルド)
- マイグレーションの例
- その他の資料

WebSphere Application Server - Express の使用に関する情報は、「ご使用になる前に」ガイドおよびオンライン・ヘルプを参照してください。 WebSphere Application Server - Express をインストールする場合は、その前に「インストール・ガイド」をお読みください。 WebSphere Application Server - Express を正しくインストールしたら、「ご使用になる前に」ガイドを参照して、「ご使用になる前に」のチュートリアルに従ってください。このチュートリアルでは、ワークベンチ、Java™ 開発、および Web サービスについて学習します。チュートリアルを終了したら、このガイドを参照してご使用のリソースを WebSphere Application Server - Express にマイグレーションしてください。

このガイドは HTML バージョンと Acrobat PDF の両方のバージョンが /Readme のディレクトリーで使用可能です。両方のバージョンは同一の情報内容です。ユーザーはどのような Web ブラウザーでも migrate.html をオープンすることができます。migrate.pdf をオープンするには、Acrobat Reader ソフトウェアがインストール済みでなければなりません。これは www.adobe.com/products/acrobat/readstep2.html からダウンロードすることができます。

この「マイグレーション・ガイド」では、一貫して Windows® の規則を使用しています。例えば、Windows の「WS_InstallDir¥」は Linux の「WS_InstallDir/」に相当します。

本書の更新については、www.ibm.com/websphere/developer/zones/studio/transition.html を参照してください。

第 2 章 実動サーバーのマイグレーション

マイグレーション

マイグレーションとは、既存の資料を利用する作業です。マイグレーション作業およびツールは、製品とその前提条件をアップグレードし、可能な場合には既存のアプリケーション・コンポーネントを再利用し、以前のバージョンから現行バージョンに管理構成を移行する作業を支援します。

WebSphere Application Server 製品のマイグレーションとは、既存の環境とアプリケーションを利用して、現在の製品バージョンと互換性のある状態にそれを変更することです。

製品のマイグレーション機能は、IBM WebSphere Application Server - Express バージョン 5.1 のマイグレーション・ツールによって提供されます。マイグレーション・ツールは、以下の製品からのマイグレーションをサポートしています。

- IBM WebSphere Application Server - Express バージョン 5.0、5.0.1、5.0.2
- IBM WebSphere Application Single Server Edition バージョン 3.5

製品のインストール・ウィザードは、前のバージョンの IBM WebSphere Application Server - Express を検出し、Version 5.1 のインストールの間に、マイグレーション・ツールを実行できるようにします。IBM WebSphere Application Server Version 3.5 からマイグレーションするには、これらのツールを直接実行する必要があります。

マイグレーションのレッドブック

「*Migrating to WebSphere V5: An End-to-End Migration Guide*」は、レッドブックの Web サイト <http://www.ibm.com/redbooks> から入手できます。レッドブックを探すには、文書番号 SG24-6910-00 を検索してください。レッドブックでは、アプリケーションのマイグレーションの計画に関するさらに詳細な情報や、WebSphere Studio Application Developer のツールとサンプルなど、この記事より広い範囲の情報が提供されています。

バージョン 3.5 のマイグレーション: J2EEモデルへの移行

V3.5.x を V5 にアップグレードすると、J2EE 仕様に基づくプラットフォームに移行します。J2EE 技術では、アプリケーションの開発と作成が、アプリケーションの管理、デプロイメント、および管理から分離されます。V3.5 からのマイグレーションには、アプリケーションの構造、開発、およびデプロイメントの変更が伴います。

マイグレーション・ツールは、システム構成をマイグレーションし、J2EE セキュリティー役割マッピングなどの J2EE 成果物を作成することで、バージョン 3.5.x からバージョン 5 への移行を支援します。マイグレーション・ツールは、バージョン 3.5.x の構成に基づいて、初期の J2EE エンタープライズ・アプリケーションを作成します。ただし、アプリケーションの構造が大きく変更されたため、新しい環境で

アプリケーションがどのように機能するのかを正確に判別するために、開発ツールとデプロイメント・ツールを使用して、マイグレーション済みのアプリケーションを慎重にテストおよび調整する計画を立ててください。

J2EE モデルでは、最終的なデプロイメント環境から独立して、アプリケーションを開発できます。このように作業が分離されることで、初期開発環境から実働環境へのアプリケーションのプロモート、またはサーバー間でのアプリケーションの移動が単純化されます。意図されているのは、アプリケーションのデプロイメント・パラメーターだけを変更し、アプリケーション・コードは変わらないようにすることです。

マイグレーションと共存

始める前に、WebSphere Application Server バージョン 5.1 製品をインストールするマシンに、既存のバージョンがインストールされているかどうかを判別する必要があります。前のバージョンがある場合は、前のバージョンの構成およびアプリケーションを新しいバージョンにコピーするかどうか、つまりマイグレーション するかどうかを計画しなければなりません。マイグレーションでは、前のバージョンはアンインストールされません。以前のリリースは変わらずに機能します。前のバージョンをバージョン 5.1 のインストールと同時に実行する場合、2 つのバージョンは共存 することになります。両方のバージョンを同時に実行するには、ポートを競合しないように構成する必要があります。マイグレーション操作はポートの定義をそのままマイグレーションするので、両方のバージョンでポートの定義が同じになることに注意してください。WebSphere Application Server には、すべてのマイグレーション機能を提供するマイグレーション・ツールが用意されています。マイグレーション・ツールは、インストール・ウィザードから呼び出すことも、後で手動で呼び出すこともできます。

IBM WebSphere Application Server - Express V5.0.x から V5.1 へのマイグレーションは、だいたいにおいてルーチン化されています。インストーラーを使用してマイグレーションを行えば、マイグレーション後のチューニングはほとんど、またはまったく必要ありません。または、マイグレーション・ツールを手動で使用して V5.0.0、V5.0.1、または V5.0.2 の構成データを保存し、V5.0.0、V5.0.1、または V5.0.2 をアンインストールしてから V5.1 をインストールした後、再びマイグレーション・ツールを使用して、構成データを復元することもできます。

全体として、IBM WebSphere Application Server V3.5 から IBM WebSphere Application Server - Express V5.1 へのマイグレーションには、アプリケーションの構造、開発、およびデプロイメントに関する大幅な変更が伴います。マイグレーション・ツールは、システム構成をマイグレーションし、以前のセキュリティー設定から J2EE セキュリティー役割へのマッピングなどの J2EE 成果物を作成することで、この移行を支援します。このようなセキュリティー・マッピングにより、移行の間にマイグレーションされる資産へのアクセスが可能になります。マイグレーション・ツールは、バージョン 3.5.x の構成に基づいて、初期の J2EE エンタープライズ・アプリケーションを作成します。ただし、アプリケーション構造が大きく変更されたため、開発ツールとデプロイメント・ツールを使用して、マイグレーション後のアプリケーションを慎重にテストおよび微調整する必要があります。

マイグレーションでは、以下のファイルが backup ディレクトリーに保管されます。

バージョン 3.5.x の場合

- bin/setupCmdLine.sh (または、Windows プラットフォームの場合は bin/setupCmdLine.bat)
- classes (iSeries では保管されません)
- hosts
- properties
- servlets

Version 5.0.x の場合

- classes (iSeries では保管されません)
- config
- installableApps
- installedApps
- properties

マイグレーション・ツール

このトピックでは、WebSphere Application Server で提供されているマイグレーション・ツールについて説明します。マイグレーション・ツールはすべて、製品 CD-ROM の /migration ディレクトリーに収められています。インストールする Application Server のバージョン用のマイグレーション・ツールを使用することが重要です。ツールには、随時変更が加えられていきます。製品 CD-ROM に収められているツールは、以前のリリースの Application Server を製品 CD-ROM のリリースにマイグレーションするために必要な機能を備えています。CD-ROM のツールは、CD-ROM の製品に対応しています。Application Server の以前のリリースのマイグレーション・ツールを使用すると、マイグレーションで問題が発生する可能性があります。

WASPreUpgrade.sh (および WASPreUpgrade.bat)

WebSphere Application Server の以前のインストールのアプリケーションと構成データを、バックアップ・ディレクトリーに保管します。このディレクトリーから新しいインストールに構成データを復元するには、WASPostUpgrade スクリプトを使用します。インストーラーでマイグレーションを選択すると、インストールの際に WASPreUpgrade.sh スクリプトが呼び出されます。新しいバージョンをインストールした後で、コマンドを使用して手動でマイグレーションを実行することもできます。

WASPostUpgrade.sh (および WASPostUpgrade.bat)

以前のリリースの構成データを復元します。WASPostUpgrade は、WASPreUpgrade スクリプトがデータを保管したバックアップ・ディレクトリーからデータを読み取ります。インストーラーでマイグレーションを選択すると、インストールの際に WASPostUpgrade.sh スクリプトが呼び出されます。新しいバージョンをインストールした後で、コマンドを使用して手動でマイグレーションを実行することもできます。

WASPreUpgrade コマンド

WASPreUpgrade.sh (または WASPreUpgrade.bat) スクリプトは、以前のバージョンまたはリリースから Version 5.1 Application Server - Express に構成とアプリケーションをマイグレーションするための、マイグレーション・ツールです。

コマンド・ファイルは、インストール後のインストール・ルートの AppServer/bin サブディレクトリーにあります。CD の migration サブディレクトリーから直接使用することもできます。

構文

```
WASPreUpgrade backupDirectory currentWASDirectory  
                [adminNodeName]  
                [-nameServiceHost host_name [-nameServicePort port_number ]]  
                [-traceString trace_spec [-traceFile file_name ]]
```

パラメーター

最初の 2 つの引き数は必須であり、位置を変えることはできません。

以下の引き数がサポートされています。

backupDirectory

必須の引き数で、位置を変えることはできません。WASPreUpgrade ツールが保管する構成とファイルを格納するディレクトリーの名前です。後で、WASPostUpgrade ツールが、構成とファイルをこのディレクトリーから読み取ります。ディレクトリーがまだ存在していない場合は、WASPreUpgrade ツールが作成します。

currentWASDirectory

必須の引き数で、位置を変えることはできません。V3.5.x または V5.0.x が現在インストールされているインストール・ルートの名前です。サポートされているのは、WebSphere Application Server Standard Edition V3.5.x または WebSphere Application Server - Express V5.0.x です。

adminNodeName

オプションの引き数で、位置を変更することはできません。現在インストールされている製品に対する管理サーバーが存在するノードの名前です。この引き数の値は、現在インストールされている製品の管理コンソールの「Topology」タブにあるトポロジー・ツリーに表示されるノード名と一致していなければなりません。WASPreUpgrade ツールは、このパラメーターを使用して XMLConfig ツールを呼び出します。このパラメーターは、WebSphere Application Server Standard Edition バージョン 3.5.x からアップグレードするときだけに必要です。

-nameServiceHost -nameServicePort

このパラメーターを指定すると、WASPreUpgrade ツールはこれらのオプション・パラメーターを XMLConfig ツールに渡します。これらのパラメーターは、XMLConfig ツールによって使用されるデフォルトのホスト名とポート番号をオーバーライドするために使用します。

-traceString -traceFile

IBM サービス技術員が使用するトレース情報を収集するためのオプション・パラメーターです。すべてのトレース情報を収集するには、`"*=all=enabled"` (引用符を含む) の `trace_spec` を指定してください。

ロギング

WASPreUpgrade ツールは、実行中のステータスを画面に表示します。また、さらに広範なロギング情報のセットを、`backup` ディレクトリーに保管します。WASPreUpgrade.log ファイルの内容は、テキスト・エディターで見ることができます。

WASPostUpgrade コマンド

WASPostUpgrade.sh (または WASPostUpgrade.bat) スクリプトは、以前のバージョンまたはリリースから Version 5.1 Application Server - Express に構成とアプリケーションをマイグレーションするための、マイグレーション・ツールです。

コマンド・ファイルは、インストール・ルートの `AppServer/bin` ディレクトリーにあります。

WASPostUpgrade ツールは、マイグレーションされたすべてのアプリケーションを、バージョン 5.1 インストールの `AppServer/installedApps` ディレクトリーにインストールします。このツールでは、WASPreUpgrade ツールが作成したバックアップ・ディレクトリーにあるアプリケーションもインストールされます。

WASPreUpgrade ツールは、以前のバージョンまたはリリースの `installedApps` およびその他のディレクトリーからアプリケーションをコピーします。

構文

```
WASPostUpgrade backupDirectory
    [-serverName server_name]
    [-webModuleAdditionalClasspath classpath]
    [-documentRootLimit number]
    [-substitute "key1=value1[;key2=value2;[...]]"]
    [-portBlock port_starting_number]
    [-backupConfig true | false]
    [-replacePorts true | false]
    [[-traceString trace_spec [-traceFile file_name]]
```

パラメーター

最初の引き数は必須です。以下の引き数がサポートされています。

serverName

オプションのサーバー・インスタンス名です。デフォルトは `server1` です。

backupDirectory

必須の引き数です。WASPreUpgrade ツールが保管する構成とファイルを格納するディレクトリーの名前です。WASPostUpgrade ツールは、構成とファイルをこのディレクトリーから読み取ります。ディレクトリーがまだ存在していない場合は、WASPreUpgrade ツールが作成します。

-backupConfig

オプションのパラメーターです。マイグレーション・ツールが構成を変更する前に既存の構成をバックアップするために使用します。デフォルトは `true` で、構成をバックアップします。

-documentRootLimit

オプションのパラメーターです。Web-application の `document-root` フィールドからプログラムがコピーするファイルの数を指定します。バージョン 3.5.x のアップグレードに対してのみ適用されます。指定しない場合のデフォルトは 300 です。

-portBlock

オプションのパラメーターです。ポートを作成するとき使用する開始の値を指定します。

-substitute

XMLConfig ツールに渡されるオプションの引き数です。置き換えるセキュリティ変数の値を指定します (例 : `-substitute "NODE_NAME=admin_node;APP_SERVER=default_server"`)。

入力 XML データ・ファイルでは、置き換えられる各キーは `key` と示されています。上の例の引き数では、入力 XML ファイル中の `$NODE_NAME$` はすべて `admin_node` に置き換えられ、`APP_SERVER` はすべて `default_server` に置き換えられます。

置換ストリングにセミコロンが含まれる場合は、区切り文字の `;` と区別するために、`$semiColon$` を使用します。UNIX プラットフォームでは、置換ストリング内の各ドル記号 (`$`) に対してエスケープ文字を追加します (例 : `\$semiColon\$`)。

このパラメーターは、Advanced Edition バージョン 3.5.x から保管された構成に適用されます。

-traceString -traceFile

IBM サービス技術員が使用するトレース情報を収集するためのオプション・パラメーターです。すべてのトレース情報を収集するには、`"*=all=enabled"` (引用符を含む) の `trace_spec` を指定してください。

-webModuleAdditionalClasspath

オプションのパラメーターです。Web アーカイブ・ファイル (WAR) にコピーしたくない特定のディレクトリーまたはファイルのパスおよびファイル名を指定します。コピーする代わりに、プログラムは、パスとファイルを、Web モジュールの拡張 (`ibm-web-ext.xml`) `additionalClassPath` 属性に追加します。このパラメーターは、バージョン 3.5.x のインストールをマイグレーションするときのみ適用されます。

ロギング

WASPostUpgrade ツールは、実行中のステータスを画面に表示します。また、さらに広範なロギング情報のセットを、`logs` ディレクトリーに保管します。

WASPostUpgrade.log ファイルの内容は、テキスト・エディターで見ることができます。

マイグレーションにおける構成のマッピング

このトピックでは、マイグレーションにおいて変更される部分について説明します。これには、常に、スタンドアロン・マシンでの開発環境のようなシングル・マシンが含まれます。

バージョン 3.5 からバージョン 5.x へのマイグレーション

マイグレーション・ツールは、システム構成をマイグレーションし、J2EE セキュリティ役割マッピングなどの J2EE 成果物を作成することで、バージョン 3.5.x からバージョン 5 への移行を支援します。マイグレーション・ツールは、バージョン 3.5.x の構成に基づいて、初期の J2EE エンタープライズ・アプリケーションを作成します。ただし、アプリケーションの構造が大きく変更されたため、バージョン 5 でアプリケーションがどのように機能するのかを正確に判別するために、開発ツールとデプロイメント・ツールを使用して、マイグレーション済みのアプリケーションを慎重にテストおよび微調整する計画を立ててください。

マイグレーションされた Enterprise Bean に関する詳細な情報については、WASPostUpgrade.log ファイルを分析してください。J2EE プログラミング・モデルでは、アプリケーションを作成およびデプロイする方法についてのアーキテクチャーが指定されています。バージョン 3.5.x のアプリケーションはアーキテクチャーが異なるので、WASPostUpgrade ツールはアプリケーションを再作成します。J2EE アプリケーションにおいてマイグレーションされるすべての Web リソースと Enterprise Bean が作成されます。また、バージョン 3.5.x インストールのすべてのエンタープライズ・アプリケーションが、同じ名前でも J2EE アプリケーションにマップされ、同じサーバーにデプロイされます。

WASPostUpgrade ツールは、エンタープライズ・アプリケーションに組み込まれていない Web リソースを、サーバーの名前を含むデフォルトの J2EE アプリケーションにマップします。このツールは、Web アプリケーションを J2EE WAR ファイルにマップします。このツールは J2EE EAR ファイル内のリソースを結合し、それをバージョン 5 の構成にデプロイします。

V3.5.x からバージョン 5.x へのマイグレーションに関するマッピングの詳細

• datasources.xml

バージョン 3.5.x の datasources.xml ファイルを使用して、データ・ソース構成の設定を追加できます。バージョン 3.5.x では、このファイルは properties ディレクトリーに保管されています。マイグレーション・ツールは、ファイル内のプロパティをデータ・ソースおよび JDBC ドライバーの構成にマージすることで、既存の datasources.xml ファイルをマイグレーションします。

• エンタープライズ・アプリケーション

バージョン 3.5.x の enterprise-application エントリーはオプションです。このエントリーは、セキュリティ定義に対してオブジェクトの集合をまとめて編成するために、最もよく使用されます。enterprise-application の Enterprise Bean と web-application の部分は、XML ファイルの他の部分にある対応するエントリーを指しています。各 enterprise-application は、同じ名前を持つ J2EE アプリケーションを作成するために処理されます。Enterprise Bean および web-application のエントリーは、Enterprise Bean

および Web アプリケーションの定義へのポインターとして使用されます。さらに、これらのエントリーの詳細は J2EE アプリケーションを作成するために使用されます。

Enterprise Bean のファイルの場合、JAR-file の定義は、J2EE アプリケーションを再デプロイしたり、J2EE アプリケーションに追加したりするための JAR ファイルを検出するために使用されます。Web-application の document-root エントリーは、Web-application 内で使用されるリソース (HTML、JSP ページなど) を検出するために使用されます。これらのファイルは、J2EE アプリケーション内の WAR ファイルにコピーされます。Web-application の classpath エントリーは、J2EE アプリケーション内の WAR ファイルにコピーされるサーブレットおよび JAR ファイルを検出するために使用されます。

エンタープライズ・アプリケーションは、バージョン 3.5.x からのマイグレーションの過程で作成されます。これらは J2EE 1.2 互換のエンタープライズ・アプリケーションとして作成され、サーブレット 2.2 レベルおよび JSP 1.1 レベルのモジュールを含みます。これにより、最も直接的な互換性が提供され、前のバージョンの WebSphere Application Server との相互運用が可能になります。

• J2EE セキュリティー

バージョン 3.5.x のセキュリティー許可モデルは、エンタープライズ・アプリケーションおよびメソッド・グループの概念に基づいています。エンタープライズ・アプリケーションとメソッド・グループのクロス積が、WebSphere Application Server の許可になります。J2EE の仕様には、役割に基づく許可モデルが含まれています。

バージョン 3.5.x の WebSphere Application Server 許可モデルをバージョン 5 の役割ベースの許可モデルに変換するため、マイグレーション・ツールは、WebSphere Application Server の許可からそのアプリケーションでの新しい役割に対する 1 対 1 のマッピングを作成します。したがって、V3.5.x の各エンタープライズ・アプリケーションおよび各メソッド・グループに対し、マイグレーション・ツールはバージョン 5 の役割を作成します。これらは、J2EE アプリケーションのデプロイメント・ディスクリプターに含まれています。それぞれの役割で許可される対象は、J2EE アプリケーションのバインディングにある許可テーブルに含まれています。

J2EE の仕様には、役割に基づく許可モデルが含まれています。

WebSphere Application Server は、リソースにアクセスするための一連の許可を意味する役割を解釈します。Enterprise Bean メソッドの呼び出しの場合、特定の Bean のメソッドにアクセスするための許可は、メソッド許可によって指定されます。このメソッド許可は、Bean JAR ファイルのデプロイメント・ディスクリプター内の 1 つ以上の役割に関連付けられます。

Web リソースへのアクセスの場合は、Web URI にアクセスし、その URI の HTTP メソッドを呼び出す許可は、J2EE 仕様の Web リソース・コレクションおよびセキュリティー制約という形で指定されています。セキュリティー制約と Web リソース・コレクションは、Web アプリケーションの WAR ファイルのデプロイメント・ディスクリプターにあります。

- **JSP レベル**

バージョン 5 は、JSP 1.0 と 1.1 のオブジェクトを、JSP 1.2 のオブジェクトとして実行します。バージョン 5 がサポートするレベルは JSP 1.2 だけです。

- **サーブレット・リダイレクター**

バージョン 5 は、以前のバージョンのサーブレット・リダイレクターをサポートしません。マイグレーション・ツールは、これらのオブジェクトを無視します。

- **V3.5.x から V5.x にマイグレーションするときのサーブレット・パッケージ名の変更**

バージョン 3.5.x の構成で SimpleFileServlet サーブレットが定義されている場合、このサーブレットはマイグレーションされません。マイグレーション・ツールは、Web モジュール・ファイル ibm-web-ext.xmi 内の FileServingEnabled 属性を true に設定します。

バージョン 3.5 の構成で InvokerServlet サーブレットが定義されている場合、このサーブレットはマイグレーションされません。マイグレーション・ツールは、Web モジュール・ファイル ibm-web-ext.xml 内の ServeServletsByClassnameEnabled 属性を true に設定します。

バージョン 3.5.x の構成で DefaultErrorReporter サーブレットが定義されている場合、このサーブレットは Web モジュール・ファイル web.xml にマイグレーションされます。マイグレーションでは、クラス名を設定するために新しいパッケージが使用されます。

- **トランスポート**

バージョン 3.5.x のサーブレット・エンジンのデフォルト・トランスポート・タイプは、Open Servlet Engine (OSE) です。バージョン 5 では OSE トランスポートがサポートされなくなったため、マイグレーション・ツールは、同じポート割り当てを使用して、これらのトランスポートを HTTP トランスポートにマップします。ポートごとに VirtualHost エリアスのエントリーを手動で追加する必要があります。

構成データの手動マイグレーション

このタスクで説明するように、インストール・ウィザードを使用して、または手動で、管理構成をマイグレーションできます。マイグレーションを手動で行う場合は、インストール・ウィザードのマイグレーション・パネルでマイグレーション・チェック・ボックスを選択しないでください。

以前のバージョンの WebSphere Application Server を使用している場合は、システム管理者が、アプリケーションやサーバーのさまざまな設定を、環境に合わせて微調整している可能性があります。このような設定を最大限の効率と最小限の損失でマイグレーションする戦略を用意する必要があります。

手動での段階的マイグレーションは、マイグレーション・ツールを繰り返し呼び出し、そのたびに異なる構成ファイルを指定することで実行できます。複数の構成ファイルを使用するには、いくつかの理由があります。どのような理由であっても、一度に 1 つの構成ファイルをマイグレーションすることで、次の構成ファイルに進む前に、アプリケーションを段階的にテストすることができます。

マイグレーション・ツールを使用する前に、V5.x のリリース情報資料を参照して、以前のバージョンに適用する必要がある修正を確認してください。以前のバージョンの修正を適用することで、マイグレーションに使用するファイルにも修正が適用される場合があります。修正を適用することで、構成やアプリケーションを可能な限り最も効率よくマイグレーションすることができます。

手動によるマイグレーションでは、インストール・ウィザードが提供する完全なマイグレーションよりさらに段階的なマイグレーション方法が用意されています。IBM は、V3.5.x または V5.0.x の任意の版から WebSphere Application Server - Express 製品に管理構成をマイグレーションするための、一連のマイグレーション・ツールを提供しています。マイグレーションの全体的なプロセスは、WASPreUpgrade マイグレーション・ツールを使用して現在の構成と必要なファイルをバックアップし、以前のリリースをアンインストールし、自動マイグレーション・オプションを選択しないでバージョン 5 をインストールした後、WASPostUpgrade マイグレーション・ツールを使用して以前のリリースの構成を復元するというものです。

構成データを WebSphere Application Server の基本ノードにマイグレーションする方法については、以下のマイグレーション・シナリオのいずれかを選択してください。

- 『V3.5.x から V5.1 へのマイグレーション』
- 14 ページの 『V3.5.x からリモート V5.1 マシンへのマイグレーション』
- 17 ページの 『V5.0.x から V5.1 へのマイグレーション』
- 17 ページの 『V5.0.x からリモート V5.1 マシンへのマイグレーション』
- 19 ページの 『サポートされていないオペレーティング・システムからのマイグレーション』

V3.5.x から V5.1 へのマイグレーション

マイグレーション・ツールを使用して、バージョン 3.5 の WebSphere Application Server からバージョン 5.1 の WebSphere Application Server - Express に構成データをマイグレーションできます。

通常は、WebSphere Application Server V5.1 のマイグレーション・ツール WASPreUpgrade および WASPostUpgrade を使用して、同じマシン上で V3.5 から V5.1 にアップグレードします。あるマシンの V3.5 構成を別のマシンの WebSphere Application Server - Express V5.1 にマイグレーションするシナリオの場合は、14 ページの 『V3.5.x からリモート V5.1 マシンへのマイグレーション』 で説明されている代替手順を使用してください。

このトピックでは、V5.1 のマイグレーション・ツールを使用してマイグレーションを行う方法について説明します。

- WebSphere Application Server Single Server Edition バージョン 3.5

WASPreUpgrade ツールは、既存の V3.5 の構成を、*migration-specific-backup* ディレクトリに保管します。WASPostUpgrade ツールは、このディレクトリを使用して、古い構成の設定を新しい V5.1 環境に追加します。

この作業のステップ

1. V5.1 製品の CD-ROM を入手する。

この CD には、migration/bin というディレクトリーがあります。このディレクトリーには、V5.1 をインストールしないで WASPreUpgrade ツールを実行するために使用できる特殊な環境が含まれています。

2. V5.1 製品 CD-ROM の /migration/bin ディレクトリーにある WASPreUpgrade スクリプトを使用して、現在の構成を保管する。

構成を migration-specific-backup ディレクトリーに保管します。

```
WASPreUpgrade /usr/tmp/migration-specific-backup /usr/websphere/appserver yourNodeName
```

既存の環境の管理サーバーが稼動していることを確認します。WASPreUpgrade ツールは、状況を画面に表示すると共に、migration-specific-backup ディレクトリーのログ・ファイルに記録します。ASCII 形式のログ・ファイルの名前は、WASPreUpgrade で始まり、日付のタイム・スタンプが含まれています。

WASPreUpgrade ツールは、既存の V3.5.x 構成に存在する以下のディレクトリーに含まれるすべてのファイルを、バックアップ・ディレクトリーに保管します。

バージョン 3.5.x の場合

- bin
- classes
- hosts
- properties
- servlets

WASPreUpgrade ツールは、V3.5.x の /bin ディレクトリーから、選択されたファイルを保管します。また、V3.5.x のリポジトリーから既存の Application Server の構成をエクスポートします。WASPreUpgrade ツールは、XMLConfig を呼び出して、既存の V3.5 のリポジトリーを migration-specific-backup ディレクトリーの websphere_backup.xml ファイルにエクスポートします。

WASPreUpgrade ツールの実行中にエラーが発生する場合、エクスポート・ステップを正常に実行するためには、V3.5 のインストールに修正を適用する必要がある場合があります。該当する可能性のある最新の修正については、IBM のサポート・ページを参照してください。InfoCenter からこの情報を表示するには、IBM サポート・ページにリンクしている「サポート」をクリックして下さい。

3. WebSphere Application Server - Express バージョン 5.1 製品をインストールする。

マイグレーション・オプションが表示されても選択しないでください。

WASPostUpgrade の使用が終了するたびに、2 つのファイルで V5 のポート設定を検査します。

- serverindex.xml ファイルでは、server1 に対する BOOTSTRAP_ADDRESS ポートの割り当てを検査する。

以前のバージョンの BOOTSTRAP_ADDRESS ポートが 900 の場合は、マイグレーションによって 7809 にマップされます。以前のバージョンの BOOTSTRAP_ADDRESS ポートが 900 ではない場合は、マイグレーションに

よって、Advanced Edition マイグレーションでの server1 に、または Advanced Single Server Edition マイグレーションでの実際のサーバー名に、マップされます。

- server.xml ファイルでは、HTTP Transport ポートの割り当てを検査する。WASPostUpgrade の処理では、以前のバージョンの HTTP Transport ポートが、バージョン 5 の server.xml ファイルに追加されます。つまり、server1 には、共存パネルからと、以前のバージョンのデフォルト・サーバーからの、重複する HTTP Transport ポート割り当てが含まれています。

4. V5.1 インストールのルート・ディレクトリーの AppServer/bin ディレクトリーにある WASPostUpgrade ツールを使用して、以前の構成を新しいインストールにマイグレーションする。

WASPostUpgrade ツールは、WASPreUpgrade ツールによって作成された V3.5.x の構成情報を、V5.1 のインストールにマイグレーションします。V5.1 製品は J2EE プログラミング・モデルに準拠し、V3.5.x は準拠しないので、V3.5.x の構成を V5.1 インストールに適用するには、大幅な変更が必要です。

V5.1 にはすでにサンプルと管理コンソール・アプリケーションがあるので、WASPostUpgrade ツールは、サンプルまたは管理コンソール・アプリケーションのマイグレーションを行いません。

WASPostUpgrade ツールは、デプロイされる各 Enterprise Bean に固有の詳細情報を、WASPostUpgrade.log ファイルに、記録します。

5. 以前のバージョンの管理サーバーが稼動している場合は、バージョン 5 のノードを稼動する前に停止する。
6. マイグレーション後に WebSphere Application Server の構成を行って、マイグレーション・ツールの結果を検査します。マイグレーションの途中で構成マッピングを使用して、マイグレーションの結果を検査することもできます。トピックには、マイグレーション・ツールがオブジェクトをマイグレーションする方法と、検査が必要な項目についての詳細な説明があります。

V3.5.x からリモート V5.1 マシンへのマイグレーション

マイグレーション・ツールを使用して、2 つのマシン間の手動マイグレーションを実行できます。

通常は、WebSphere Application Server - Express のマイグレーション・ツール WASPreUpgrade および WASPostUpgrade を使用して、同じマシン上で V3.5 から V5.1 にアップグレードします。

しかし、あるマシンの V3.5 構成を別のマシンの V5.1 にマイグレーションしなければならないシナリオがあります。このようなシナリオの 1 つとしては、新しいマシンに最新の V5.1 環境をインストールし、一方で他のマシンにある既存の V3.5 構成をマイグレーションする必要があるような場合です。

このトピックでは、V5.1 のマイグレーション・ツールを使用してマイグレーションを行う方法について説明します。

- WebSphere Application Server Single Server Edition バージョン 3.5

WASPreUpgrade ツールは、既存の V3.5 の構成を、*migration-specific-backup* ディレクトリーに保管します。WASPostUpgrade ツールは、このディレクトリーを使用して、古い構成の設定を新しい V5.1 環境に追加します。

この作業のステップ

1. V5.1 製品の CD-ROM を入手する。

この CD には、*migration/bin* というディレクトリーがあります。このディレクトリーには、V5.1 をインストールしないで WASPreUpgrade ツールを実行するために使用できる特殊な環境が含まれています。

2. V5.1 製品 CD-ROM の */migration/bin* ディレクトリーにある WASPreUpgrade スクリプトを使用して、現在の構成を保管する。このディレクトリーは、V3.5 マシンに対してマウントする必要があります。

V3.5 マシン上の *migration-specific-backup* ディレクトリーに、構成を保管します。

```
WASPreUpgrade /opt/tmp/migration-specific-backup /opt/websphere/appserver adminNodeName
```

既存の環境の管理サーバーが稼動していることを確認します。

WASPreUpgrade ツールは、状況を画面に表示すると共に、*/migration-specific-backup* ディレクトリーのログ・ファイルに記録します。ASCII 形式のログ・ファイルの名前は、WASPreUpgrade で始まり、日付のタイム・スタンプが含まれています。

WASPreUpgrade ツールは、V3.5.x の */bin* ディレクトリーから、選択されたファイルを保管します。また、V3.5.x のリポジトリーから既存の Application Server の構成をエクスポートします。WASPreUpgrade ツールは、XMLConfig を呼び出して、既存の V3.5 のリポジトリーを *migration-specific-backup* ディレクトリーの *websphere_backup.xml* ファイルにエクスポートします。

WASPreUpgrade ツールの実行中にエラーが発生する場合、エクスポート・ステップを正常に実行するためには、V3.5 のインストールに修正を適用する必要があります。該当する可能性のある最新の修正については、IBM のサポート・ページを参照してください。InfoCenter からこの情報を表示するには、IBM サポート・ページにリンクしている「サポート」をクリックして下さい。

3. V3.5 マシンから V5.1 マシンに、*/migration-specific-backup* ディレクトリーをコピーする。

ファイルを新しいマシンにコピーするには、FTP、共用ストレージ、または他のメカニズムを使用します。

WebSphere Application Server - Express の V5.1 のマシンで、以下のステップを実行します。

4. */migration-specific-backup/websphere_backup.xml* ファイルまたは */migration-specific-backup/config/server-cfg.xml* ファイルをコピーし、コピーをアーカイブとして保存するために選択した場所に格納する。

このファイルをコピーするのは、次のステップで元のファイルを編集するためです。

5. /migration-specific-backup/websphere_backup.xml ファイルまたは /migration-specific-backup/config/server-cfg.xml ファイルを編集し、マシンに依存する設定を修正する。

ファイルを以下のように変更します。

- a. /migration-specific-backup/websphere_backup.xml ファイルで、**ノード名を変更する**。/migration-specific-backup/config/server-cfg.xml ファイルにはノード名はありません。

元の V3.5 構成で使用していたものと同じノード名前を V5.1 マシンでも使用する場合は、名前を変更しません。名前が異なる場合は、V3.5 のノード名が使用されているすべての箇所を、WebSphere Application Server V5.1 で使用するノード名に変更する**必要があります**。ノード名は、ファイル中の複数の XML スタンザで使用されています。すべての箇所が変更されないと、データを完全にマイグレーションすることができません。

- b. /migration-specific-backup/websphere_backup.xml ファイルまたは /migration-specific-backup/config/server-cfg.xml ファイルで、**パス名を変更する**。

構成ファイル内の複数の XML スタンザで、パス名が参照されています。同等のディレクトリを作成しなければならない場合であっても、V3.5 のディレクトリ構造の外部にあるファイルに対する参照を、新しいマシン上の同等のディレクトリに更新します。対応する環境を構成するということは、元のディレクトリを V5.1 マシンにコピーする必要がある場合があることを意味します。あるいは、適切なソフトウェアのインストールが必要になる場合があります。

- c. オペレーティング・システムに依存するパス名の**指定スタイルを修正する**。

パスの指定が、V5.1 を実行するマシン上で動作する形式と異なる場合は、修正する必要があります。たとえば、Windows プラットフォーム上の V3.5 を Linux プラットフォーム上の V5.1 に移行する場合は、構成ファイルに含まれる Windows 固有のパスを、Linux のパス・スタイルを使用するように変更します。たとえば、“c:\mystuff\somepath” は “/opt/mystuff/somepath” に変更します。

- d. **セキュリティ要件に合わせてユーザー ID とパスワードを変更する**。

ユーザー ID とパスワードが V5.1 マシンで使用されているものと異なる場合は、変更する必要があります。

エンコードされたパスワードを平文のパスワードに変更するには、

<password>{xor}LCoxayht</password> を <password>mypassword</password> に変更します。

- e. **他のマシン固有情報を変更する**。

構成では、新しいマシンに存在しない他のソフトウェア製品や構成が参照されている場合があります。たとえば、古いマシンにはデータベースが存在する場合があります。V5.1 の構成では、おそらく、古いマシンのデータベースをまだ指し示す必要があります。このような場合は、V3.5 マシン上のデータベースを指すようにデータ・ソースを変更します。

6. マイグレーション・オプションを選択しないで、WebSphere Application Server バージョン 5.1 をインストールする。
7. V5.1 の構成に、V3.5 の構成を追加する。

V5.1 インストールのルート・ディレクトリーの AppServer/bin ディレクトリーにある WASPostUpgrade ツールを使用して、V5.1 の構成に V3.5 の構成を追加します。

```
WASPostUpgrade /opt/tmp/migration-specific-backup
```

WASPostUpgrade ツールは、デプロイされる各 Enterprise Bean に固有の詳細情報を、/migration-specific-backup/WASPostUpgrade.log ファイルに、記録します。

V5.0.x から V5.1 へのマイグレーション

V5.1 インストール・プログラムを使用して、WebSphere Application Server - Express V5.0.x から V5.1 にマイグレーションできます。

この作業のステップ

1. V5.0.x Application Server を停止する。

インストール・ルートの AppServer/bin ディレクトリーの stopServer.sh (または stopServer.bat) スクリプトを使用します。

```
stopServer.sh server1
```

サーバーを停止しなくても V5.0.x ノードをマイグレーションできます。ただし、構成をマイグレーションするにはノードが稼動している必要はありません。マイグレーション・ツールは、ノードが停止していても、すべての構成データを取得できます。また、インストールしている V5.1 ノードを開始する前には、ノードを停止する必要があります。したがって、現時点でノードを停止してもかまいません。

2. V5.1 製品をインストールする。

マイグレーション・オプションが表示されたら、それを選択します。

3. V5.1 Application Server のインストールを検査する。

製品のインストールの最後で First Steps ツールが表示されたら、それを使用します。または、何らかの理由で First Steps ツールが表示されない場合は、手動でインストール検査テストを実行します。

V5.0.x サーバーのアンインストールは、いつでも都合のよいときに行うことができます。

V5.0.x からリモート V5.1 マシンへのマイグレーション

マイグレーション・ツールを使用して、2 つのマシン間のマイグレーションを行うことができます。

始める前に

通常は、WebSphere Application Server V5.1 のマイグレーション・ツール WASPreUpgrade および WASPostUpgrade を使用して、同じマシン上で V5.0.x から V5.1 にアップグレードします。

しかし、あるマシンの V5.0.x 構成を別のマシンの V5.1 にマイグレーションしなければならないシナリオがあります。このようなシナリオの 1 つとしては、新しいマ

シンに最新の V5.1 環境をインストールし、一方で他のマシンにある既存の V5.0.x 構成をマイグレーションする必要があるような場合です。

このシナリオでは、V5.1 のマイグレーション・ツールを使用してマイグレーションを実行する方法について説明します。

WASPreUpgrade ツールは、既存の V5.0.x の構成を、*migration-specific-backup* ディレクトリーに保管します。WASPostUpgrade ツールは、このディレクトリーを使用して、古い構成の設定を新しい V5.1 環境に追加します。

この作業のステップ

1. V5.1 製品の CD-ROM を入手する。

この CD には、migration/bin というディレクトリーがあります。このディレクトリーには、V5.1 をインストールしないで WASPreUpgrade ツールを実行するために使用できる特殊な環境が含まれています。

2. V5.1 製品 CD-ROM の /migration/bin ディレクトリーにある WASPreUpgrade スクリプトを使用して、現在の構成を保管する。このディレクトリーは、V5.0.x マシンに対してマウントする必要があります。

V5.0.x マシン上の migration-specific-backup ディレクトリーに、構成を保管します。

```
WASPreUpgrade /opt/tmp/migration-specific-backup /opt/websphere/appserver
```

WASPreUpgrade ツールは、状況を画面に表示すると共に、/migration-specific-backup ディレクトリーのログ・ファイルに記録します。ASCII 形式のログ・ファイルの名前は、“WASPreUpgrade” で始まり、日付のタイム・スタンプが含まれています。

3. V5.0.x マシンから V5.1 マシンに、/migration-specific-backup ディレクトリーをコピーする。

ファイルを新しいマシンにコピーするには、FTP、共用ストレージ、または他のメカニズムを使用します。

4. マイグレーション・オプションを選択しないで、WebSphere Application Server バージョン 5.1 をインストールする。

5. V5.0.x の構成を V5.1 の構成に追加する。

V5.1 インストールのルート・ディレクトリーの AppServer/bin ディレクトリーにある WASPostUpgrade ツールを使用して、V5.1 の構成に V5.0.x の構成を追加します。

```
WASPostUpgrade /opt/tmp/migration-specific-backup
```

WASPostUpgrade ツールは、デプロイされる各 Enterprise Bean に固有の詳細情報を、/migration-specific-backup/WASPostUpgrade.log ファイルに、記録します。

6. WebSphere Application Server 5.1 の管理インターフェースを使用して、構成を変更する。

以下の変更を行います。

- a. セキュリティ要件に合わせてユーザー ID とパスワードを変更する。

ユーザー ID とパスワードが V5.0.x マシンで使用されているものと異なる場合は、変更する必要があります。

b. 他のマシン固有情報を変更する。

構成では、新しいマシンに存在しない他のソフトウェア製品や構成が参照されている場合があります。たとえば、古いマシンにはデータベースが存在する場合があります。このような場合は、古いマシン上のデータベースを指すようにデータ・ソースを変更します。

7. V5.0.x サーバーのアンインストールは、いつでも都合のよいときに行うことができます。

サポートされていないオペレーティング・システムからのマイグレーション

バージョン 5.1 がサポートしていないオペレーティング・システム上で稼動する WebSphere Application Server バージョン 3.5.x またはバージョン 5.0.x をマイグレーションできます。

この作業のステップ

1. WebSphere Application Server バージョン 3.5.x または バージョン 5.0.x Administrative Server を開始する。
2. コマンド行マイグレーション・ツール **WASPreUpgrade** を実行する。
2 つのオプションがあります。バージョン 5.1 CD-ROM の *platform_root* の *migration¥bin* (または *migration/bin*) ディレクトリーからコマンドを実行します。または、CD-ROM のディレクトリーのファイルを、ハード・ディスクに作成したディレクトリーにコピーしてもかまいません。

バージョン 3.5.x または 5.0.x のリリース、およびコマンドが構成ファイルおよび以前のバージョンからマイグレーションするアプリケーションを格納したバックアップ・ディレクトリーを確認します。コマンドの構文については、*WASPreUpgrade* のトピックを参照してください。

- a. バージョン 5.1 CD-ROM の *platform_root* の *migration¥bin* (または *migration/bin*) ディレクトリーからコマンドを実行する。

バックアップ・ディレクトリーおよび構成ファイルの場所を確認します。

```
CD_drive:¥WASPreUpgrade backupDirectory filepath¥WebSphere¥AppServer yourNodeName
```

正常に処理が行われた場合はステップ 4 に進みます。何らかの理由で正常に処理が行われなかった場合は、ステップ 2B から 2F を実行します。

- b. ハード・ディスクに **migration** ディレクトリーを作成する。
- c. バージョン 5.1 CD-ROM の *platform_root* の *migration¥bin¥* (または *migration/bin/*) ディレクトリーから、ハード・ディスクに作成したディレクトリーに、*WASPreUpgrade.bat* (または *WASPreUpgrade.sh*) ファイルおよび *setupCmdLine.bat* (または *setupCmdLine.sh*) ファイルをコピーする。
- d. 新しいディレクトリーの *setupCmdLine.bat* (または *setupCmdLine.sh*) ファイルを編集する。

以下の変数を変更します。

- **WAS_HOME** を、作成したマイグレーション・ディレクトリーへの完全修飾パスを指すように変更します。

- **JAVA_HOME** を、IBM Developer Kit または Java ディレクトリーへの完全修飾パスを指すように変更します。
- e. UNIX ベースのインストールをバックアップしている場合は、バージョン 5.1 CD-ROM の *UNIX-based_platform_root* の migration/bin ディレクトリーにある setupCmdLine.sh ファイルおよび WASPreUpgrade.sh ファイルに対して実行可能ビットが設定されていることを確認する。
 - f. 作成した migration ディレクトリーからコマンドを実行する。
バックアップ・ディレクトリーおよび構成ファイルの場所を確認します。

```
yourMigrationDirectory%WASPreUpgrade backupDirectory  
filePath%WebSphere%AppServer yourNodeName
```
3. WebSphere Application Server バージョン 3.5.x またはバージョン 5.0.x をシャットダウンする。
 4. バックアップ・ディレクトリーを tar または zip で圧縮し、別のシステムに FTP で転送する。
 5. ホスト名を変更しないで、新しいオペレーティング・システムをインストールする。
可能であれば、システム名とパスワードも古いシステムと同じにします。マイグレーションしているアプリケーションに関連するすべてのデータベース・ファイルを、以前のシステムと同じパスに配置します。一般に、パスは変更しないようにします。ただし、バージョン 5.1 は、以前のバージョンと同じディレクトリーにインストールしないでください。パスと名前を変更する場合は、それに合わせて XML 構成ファイルを変更します。このような変更は、以下の **WASPostUpgrade** コマンドを実行する前に行う必要があります。
 6. 他のシステムからバックアップ・ディレクトリーを FTP で転送して unzip する。
 7. WebSphere Application Server - Express バージョン 5.1 をインストールする。マイグレーション・オプションが表示されても選択しないでください。
 8. バージョン 5.1 の install_root の bin ディレクトリーから、**WASPostUpgrade** コマンド行マイグレーション・ツールを実行する。
WASPreUpgrade コマンドによって作成されたバックアップ・ディレクトリー (および、そのディレクトリー内にあるすべての非標準構成ファイル名) を指定します。コマンドの正しい構文については、**WASPostUpgrade** のトピックを参照してください。

```
install_root%bin%WASPostUpgrade WAS_HOME%migration
```

第 3 章 IBM WebSphere Studio Site Developer バージョン 5.1 からのマイグレーション

この章では、以下のマイグレーション・トピックを取り上げます。

- サーバーのターゲット・サポートを使用するための J2EE プロジェクトのマイグレーション
- ウィザード生成に必要な JDK 1.4 の Java パッケージ

サーバーのターゲット・サポートを使用するための J2EE プロジェクトのマイグレーション

IBM WebSphere Studio Site Developer バージョン 5.1.1 には、新しいサーバーのターゲット機能が追加されています。この機能はデフォルトでは使用不可になっており、「ウィンドウ (Window)」>「設定 (Preferences)」>「J2EE」を選択することにより、J2EE 設定ページでこの機能を使用可能にする必要があります。サーバーのターゲット機能の機能詳細については、IBM WebSphere Studio Site Developer 製品資料を参照してください。この機能が使用可能になっていれば、特定のアプリケーション・サーバーをターゲットとするオプションが使用できます。このフィーチャーは、JDK 1.4 をサポートするためにインプリメントされたものです。これは、IBM WebSphere Studio Site Developer バージョン 5.1.1 に付属の WebSphere Application Server バージョン 5.1 用の JRE です。サーバーのターゲット・サポートを利用する J2EE プロジェクトは、IBM WebSphere Studio Site Developer の以前のバージョンとは後方互換性がありません。このため、IBM WebSphere Studio Site Developer の以前のバージョン (例えば、IBM WebSphere Studio Site Developer バージョン 5.1、IBM WebSphere Studio Site Developer バージョン 5.0.1) で作業するユーザーとは共用できません。IBM WebSphere Studio Site Developer は、このフィーチャーとの後方互換性を使用可能にする方法を提供します。21 ページの『使用可能にされたサーバーのターゲット・サポートとの後方互換性』を参照してください。この非互換性の理由は、サーバーのターゲット機能が J2EE プロジェクトの .classpath ファイルを変更してしまい、WebSphere Application Server - Express の以前のバージョンは、新しい .classpath ファイルのエントリーを認識できないためです。

使用可能にされたサーバーのターゲット・サポートとの後方互換性

サーバーのターゲット・サポートを使用可能にしておけば、サーバーのターゲットにされる J2EE プロジェクトは、ターゲット・サーバーを「ターゲット・サーバーの変更 (Modify Target Server)」ダイアログで選択可能な「ターゲット・サーバーを指定しない (No target server specified)」オプションに変更することによって、サーバーのターゲット・サポートを使用しないように戻すことができます。「ターゲット・サーバーの変更 (Modify Target Server)」ダイアログは、「リソース・ナビゲーター (Resource Navigator)」または「J2EE パースペクティブ (J2EE Perspective)」ビューの J2EE プロジェクトのポップアップ・メニューから起動します (「ターゲット・サーバー (Target Server)」>「変更 (Modify)」)。ターゲット・サーバーは、EAR、EJB、アプリケーション・クライアントおよびコネクター・

プロジェクトの J2EE プロパティ・ページからでも「ターゲット・サーバーを指定しない (No target server specified)」に変更できます (「**プロパティ (Properties)**」 > 「**J2EE**」)。Web プロジェクトの場合は、ターゲット・サーバー設定は、「Web プロパティ (Web properties)」ページで行います (「**プロパティ (Properties)**」 > 「**WEB**」)。サーバー・ターゲットの変更機能の機能詳細については、IBM WebSphere Studio Site Developer 資料を参照してください。「**ターゲット・サーバーを使用しない (No target server specified)**」オプションが使用されていると、.classpath ファイルは、IBM WebSphere Studio Site Developer の以前のバージョンと互換性のあるスタイルに戻り、.server はプロジェクトから除去されます。

注: J2EE プロジェクトをターゲットとするサーバーのみ、WebSphere Application Server バージョン 5.1 上に配置して、JDK 1.4 サポートを利用することができます。

ウィザード生成に必要な JDK 1.4 の Java パッケージ

JDK 1.4 での変更により、ユーザーは、IBM WebSphere Studio Site Developer バージョン 5.1.1 で実行するページを生成する「データベース Web ページ (Database Web Pages)」ウィザードおよび「Java Bean Web ページ (Java Bean Web Pages)」ウィザードを使用して、Java パッケージを指定する必要があります。この問題は、View Bean テンプレートが「Java Bean Web ページ (Java Bean Web Pages)」ウィザードまたは IBM データベース・アクセス Java Bean マスター詳細パターンに使用されている場合に発生します。また、作成中にパッケージを指定せずにこれらのウィザードで以前に生成されたページおよび .java ファイルが含まれるプロジェクトにも適用されます。以前に生成されたコードの場合は、.java ファイルをパッケージに移動します。次に、.jsp ファイルを更新し、インポート・ステートメントおよびクラス情報を更新します。プロジェクトの web.xml ファイルの場合は、サブレット・クラスのエントリーを更新します。

第 4 章 IBM WebSphere Studio Site Developer バージョン 5 またはバージョン 5.0.1 からのマイグレーション

この章では、以下のマイグレーション・トピックを取り上げます。

- WebSphere Studio Workbench (WSWB) バージョン 5、バージョン 5.0.1、およびバージョン 5.1
- WebSphere Studio バージョン 5.0 のワークスペースでのバージョン 5.1.1 の使用
- バージョン 5 またはバージョン 5.0.1 からの Java プロジェクトのマイグレーション
- ソース・コード管理 (SCM) システムを使用したバージョン 5 またはバージョン 5.0.1 とバージョン 5.1 のプロジェクトの共用
- Web プロジェクトのマイグレーション
- Struts 1.1 への Web プロジェクトの変換
- J2EE プロジェクト・マイグレーションに追加されたマイグレーション機能
- Web サービス・ツールの変更
- プロファイル作成ツール内で行った変更
- テンプレート・ウィザードに関する既知の互換性の問題

バージョン 5、バージョン 5.0.1、およびバージョン 5.1 での WebSphere Studio Workbench

IBM WebSphere Studio Site Developer バージョン 5.1.1 は、新規 Eclipse ベース WebSphere Studio Workbench (WSWB) 2.1.2 に基づいています。バージョン 2.1.2 および 2.0.3 または 2.0.2 では、いくつかの相違点があります。この違いの詳細情報については、`WS_Installdir\%eclipse%readme` ディレクトリー (ここで `WS_Installdir` は、IBM WebSphere Studio Site Developer をインストールしたパス) にある README ファイルを参照してください。

IBM WebSphere Studio Site Developer バージョン 5.0 は、Eclipse ベース WSWB 2.0.2 に基づき、IBM WebSphere Studio Site Developer バージョン 5.0.1 は、Eclipse ベース WSWB 2.0.3 に基づいていました。バージョン 2.0.2 と 2.0.3 には、大きな違いはありません。IBM WebSphere Studio Site Developer バージョン 5.0.1 リリースは、IBM WebSphere Studio Site Developer バージョン 5.0 の先頭にインストールされた更新マネージャーの修正パッケージでした。

IBM WebSphere Studio Site Developer バージョン 5.0 ワークスペースでのバージョン 5.1.1 の使用

既存の IBM WebSphere Studio Site Developer バージョン 5.0 のワークスペースを使用して、IBM WebSphere Studio Site Developer バージョン 5.1.1 を初めて始動すると、バージョン 5.0 からのマイグレーション方法を示すダイアログ・ボックスが表示されます。「OK」をクリックしてバージョン 5.0 のワークスペースをマイグ

レーションするか、または「キャンセル (Cancel)」をクリックして IBM WebSphere Studio Site Developer が開始するのを停止します。

ワークスペースをマイグレーションしても、新しいプロジェクト・フィーチャーのメタデータは無視され、バージョン 5.0 で読み取ることができるため、そのワークスペースはバージョン 5.0 で使用できます。メタデータに影響を与えるか、またはプロジェクトの新しいプロジェクト・フィーチャーのメタデータを上書きするようなワークスペース内のプロジェクトに対して、バージョン 5.0 で変更を行うことはできません。

バージョン 5 またはバージョン 5.0.1 からの Java プロジェクトのマイグレーション

バージョン 5 またはバージョン 5.0.1 からの Java プロジェクトのマイグレーションは単純で自動的です。一度プロジェクトをバージョン 5.1.1 のワークスペースにロードすると、バージョン 5.1.1 の新規フィーチャーを使用しない限り、.classpath ファイルまたは .project ファイルにメタデータの変更は発生しません。

ソース・コード管理 (SCM) システムを使用したバージョン 5 またはバージョン 5.0.1 とバージョン 5.1.1 のプロジェクトの共用

IBM WebSphere Studio Site Developer バージョン 5 および IBM WebSphere Studio Site Developer バージョン 5.1.1 を使用して、チーム・リポジトリ内のプロジェクトが開発者によってロードおよび操作される場合は、特別の管理が必要です。一般的な問題は、ワークスペース内のメタデータの存在、内容、そして解釈が特定のフィーチャーやプラグインに固有なものである場合があります、しかもバージョンごとに異なる点があります。ワークスペースの互換性は、すべての開発者が厳格に管理されたステップでその IBM WebSphere Studio Site Developer ワークスペースをアップグレードする場合にのみ保証されます。そのような場合、共用メタデータで問題は発生しません。しかし、ある開発者が IBM WebSphere Studio Site Developer バージョン 5.1.1 で作業している最中に、別の開発者が IBM WebSphere Studio Site Developer バージョン 5 で作業をした場合は、このような保証はありません。この章は何をして何をすべきでないかのアドバイスを提供します。

標準的な障害モードは、IBM WebSphere Studio Site Developer バージョン 5.1.1 ユーザーに通知されます。バージョン 5.1.1 メタデータは、バージョン 5 ユーザーが変更を保管し、そして更新済みのメタデータ・ファイルをリポジトリにコミットすると失われます。ここには間違いとなる可能性のあるケースを示します。

- IBM WebSphere Studio Site Developer バージョン 5.1.1 で作業するユーザーはプロジェクトの作成や変更を、結果として共用メタデータ・ファイルが 5.1.1 特有の情報に頼って変更される方法で行います。そこでユーザーは、共用のメタデータ・ファイルを含む更新済みのプロジェクト・ファイルを、共用のリポジトリにコミットしてしまいます。
- IBM WebSphere Studio Site Developer バージョン 5 で作業する別のユーザーが、同じリポジトリのこのプロジェクトを共用します。メタデータ・ファイル内の 5.1.1 特有の情報は、IBM WebSphere Studio Site Developer バージョン 5 では理解されず、一般的には警告なしに破棄されるか無視されます。ユーザーはプロジェクトを、結果として共用メタデータ・ファイルが変更されるやり方で変

更するので、共用メタデータ・ファイルは 5.1.1 特有の情報が全くないまま書き直されてしまいます。ユーザーは、共用のメタデータ・ファイルを含む更新済みのプロジェクト・ファイルを共用のリポジトリにコミットします。ユーザーは一般には、そのアクションの結果として共用の情報が失われたばかりだということを知りません。

- IBM WebSphere Studio Site Developer バージョン 5.1.1 で作業する 1 人のユーザーが、共用リポジトリから更新された共用メタデータ・ファイルを含むプロジェクトの変更を取り上げます。このユーザーは、後でこの一連の作業が誤動作を始めるまで、逆のステップを取っていることに気付いていない可能性があります。

ここに、プロジェクトが IBM WebSphere Studio Site Developer バージョン 5.1.1 とバージョン 5 またはバージョン 5.0.1 のユーザーで共用される場合に監視し続ける作業のリストがあります。

- **リンクされたフォルダーとファイル**

このサポートは、IBM WebSphere Studio Site Developer バージョン 5.1.1 に追加されています。リンクされたリソースについての情報は .project ファイルに記録されます。

推奨事項: 使用しない。「ワークベンチ (Workbench)」>「リンクされたリソース (Linked Resources)」の設定ページを使用して、リンクされたリソースを使用不可にすることをお勧めします。

- **外部ツール (Ant) ビルダー**

外部ツール・ビルダーについての情報は .project ファイルに記録されます。バージョン 5 とバージョン 5.1.1 の間で変更された情報のフォーマット。バージョン 5.1.1 で作成または変更されたビルダーは、バージョン 5 ワークスペースでは理解されない新規のフォーマットを使います。バージョン 5 で作成されたビルダーは古いフォーマットを使用し、バージョン 5.1.1 で継続して機能します。

推奨事項: 常にバージョン 5 ワークスペースから外部ツール・ビルダーを作成、または編集する。

- **ビルド・クラスパスにおける Java ソース項目のオプションの排他パターン**

追加されたサポート。この情報は .classpath ファイルに記録されます。

推奨事項: 排他パターンを指定しない。「Java」>「コンパイラ (Compiler)」>「ビルド・パス (Build Path)」の設定ページを使用して排他パターンを使用不可にすることをお勧めします。

- **ビルド・クラスパスにおける Java ソース項目で関連づけられる出力フォルダー**

追加されたサポート。この情報は .classpath ファイルに記録されます。

推奨事項: デフォルト (プロジェクト・ワイド) 出力フォルダー以外は何も指定しない。「Java」>「コンパイラ (Compiler)」>「ビルド・パス (Build Path)」の設定ページを使用して複数出力ロケーションを使用不可にすることをお勧めします。

- **ビルド・クラスパス上の Java ライブラリー項目で関連づけられたソース接続ルート・パス**

ソース ZIP ファイルを Java ビルド・パス上の JAR ライブラリーに接続するとき、ソース・ルート・パスのプレフィックスは自動的に推測されます。これはバージョン 5 から変更された点であり、ユーザー・インターフェースを使用して明

示的に設定され、.classpath ファイルに明示的に記録することができます。このため、5.1.1 ワークスペースで作成された Java プロジェクトは接続されたソースを検出できない可能性があります。

バージョン 5 を使用してソース接続ルート・パスを指定します。追加のソース接続の柔軟性がバージョン 5.1.1 で提供されています。ユーザーはソース接続として JAR ファイルや ZIP ファイルの代わりにフォルダーを提供することができます。また、ソースをクラス・ファイル・フォルダーに接続することもできます。この機能はバージョン 5 では使用できません (バージョン 5.1.1 情報は無視されるため)。

- **従属プラグインのための PDE クラスパス・コンテナ**

クラスパス・コンテナを使用する PDE が追加されています。クラスパス・コンテナは .classpath ファイルに記録されます。もし PDE クラスパス・コンテナが使用されると、そこではバージョン 5 ワークスペースが未解決のクラスパス項目を持つことになり、その結果大部分の Java の能力 (コンパイル、検索、実行、デバッグを含む) が期待通りの結果を生まないことになります。

推奨事項: クラスパス・コンテナを使用するための「**プラグインの開発 (Plug-in Development)**」 > 「**Java ビルド・パス・コントロール (Java Build Path Control)**」の設定ページ上の設定が、どの新規プラグイン (またはフラグメント) プロジェクトを作成する前にも、使用不可になっているようにする。

Web プロジェクトのマイグレーション

フォルダー名は「**Java ソース (Java Source)**」および「**Web コンテンツ (Web Content)**」です。新規 Web プロジェクト用のデフォルトのフォルダー名は、設定ページ (「**ウィンドウ (Window)**」 > 「**設定 (Preferences)**」 > 「**Web ツール (Web Tools)**」 > 「**新規プロジェクト (New Project)**」) を介して構成できます。デフォルトの名前は、**JavaSource** と **WebContent** になりました。デフォルト名は新規の Web プロジェクトのみに使用されます。このリリースより以前のバージョンで作成された Web プロジェクトは、古い名前のみで継続して機能します。同じことが静的 Web プロジェクトについても言えます。

ユーザーが 4.0.x と 5.0 のプロジェクトのソース・フォルダー名をバージョン 5.1.1 で変更したいときは、ナビゲーター・ビューの「**名前変更 (Rename)**」ポップアップ・メニュー・アクションを使用します。「**名前変更 (Rename)**」ポップアップ・メニュー・アクションはフォルダー名を名前変更し、4.0.x と 5.0 Web プロジェクトへの Java ビルド・パスを修正します。

JavaSource フォルダーの場合、「**名前変更 (Rename)**」ポップアップ・メニュー・アクションは「**プロジェクト・ナビゲーター (Project Navigator)**」ビューと「**パッケージ (Packages)**」ビューで作動します。**WebContent** フォルダーの場合は、「**名前変更 (Rename)**」ポップアップ・メニュー・アクションは「**リソース・ナビゲーター (Resource Navigator)**」ビューと「**プロジェクト・ナビゲーター (Project Navigator)**」ビューで作動します。

このリリースより以前のバージョンの Web プロジェクトを SCM リポジトリに保管してからこのリリースにロードすると、**ソース・フォルダー**と **webApplication** フォルダーを持つ古い構造が保持されます。どちらの構造も正しくビルドされます。

注: ユーザーが「**Java ソース (Java Source)**」と「**Web コンテンツ (Web Content)**」のフォルダー名を名前変更すると、変更が必要な自動化されたビルド・スクリプトは、新規のフォルダー名を使用するために手動で更新する必要があります。

Struts 1.1 への Web プロジェクトの変換

Struts ツール・ランタイムが、バージョン 5 内の時のバージョン 1.1 ベータ 2 からバージョン 1.1 にステップアップされました。IBM WebSphere Studio Site Developer バージョン 5 (一般出荷版) では、ユーザーが Web プロジェクトを作成する際に、ユーザーのプロジェクトに Struts のサポートを追加するオプションが加えられました。ユーザーは Struts 1.0.2 または Struts 1.1 ベータ 2 のどちらかを選択できます。IBM WebSphere Studio Site Developer バージョン 5.1.1 では、後者の選択は Struts 1.1 に置き換えられています。IBM WebSphere Studio Site Developer バージョン 5 で Struts 1.1 ベータ 2 Web プロジェクトを作成した場合、そのプロジェクトを Struts 1.1 に変換することをお望みになるかもしれませんが、Struts 1.1 ベータ 2 はまだサポートされているため、その必要はありません。もしユーザーが Struts 1.1 ベータ 2 Web プロジェクトをお持ちで、Struts 1.1 への変換をお望みの場合は以下を行ってください。

1. Struts 1.1 ベータ 2 プロジェクトを IBM WebSphere Studio Site Developer バージョン 5.1.1 のワークスペースにロードする。
2. **Struts11** という名前の新規 Struts 1.1 Web プロジェクトを作成する。これは Struts 1.1 作成物への便利なアクセス方法を提供するもので、本当のプロジェクトを変換する間必要になります。完了したらこのプロジェクトを削除できます。
3. Struts 1.1 ベータ 2 プロジェクトのおおのを、Struts 1.1 へ変換をお望みの場合は以下を行ってください。
 - a. .jar ファイル commons-*.jar および struts.jar を、ユーザーのプロジェクトの Web Content/WEB-INF/lib ディレクトリーから削除する。
 - b. .jar ファイル commons-*.jar および struts.jar を、Struts11/WebContent/WEB-INF/lib ディレクトリーからユーザーのプロジェクトの Web Content/WEB-INF/lib ディレクトリーにコピーする。
 - c. .tld ファイル struts-*.tld をユーザーのプロジェクトの Web Content/WEB-INF ディレクトリーから削除する。
 - d. .tld ファイル struts-*.tld を Struts11/WebContent/WEB-INF ディレクトリーからユーザーのプロジェクトの Web Content/WEB-INF ディレクトリーにコピーする。

上記のすべての情報は、ユーザーが Struts 1.1 Beta 3 Web プロジェクトを IBM WebSphere Studio Site Developer バージョン 5.0.1 内から Struts 1.1 へ移動する場合に適用されます。

Web サービス・ツールの変更

Web サービス・ツールに 2 つの新規 IBM WebSphere Application Server バージョン 5.0.2 ランタイム・プロトコルが追加されました。これは WebSphere Application Server バージョン 5.0.2 のみで実行されます。IBM WebSphere Studio Site Developer バージョン 5.1.1 以降の必須のマイグレーションではなく、WebSphere

Application Server バージョン 5.0.2 は新旧両方のランタイム・プロトコルをサポートします。IBM WebSphere Studio Site Developer は、Web サービス成果物の 3 つのランタイム・プロトコルを生成およびデプロイします。すなわち、WebSphere Application Server バージョン 4.x およびバージョン 5.x で実行される古い "IBM SOAP" ランタイム・プロトコル、WebSphere Application Server バージョン 5.0.2 のみで実行される新しい "IBM WebSphere 5.0.2 runtime" ランタイム・プロトコル、そして WebSphere Application Server バージョン 5.0.2 のみで実行される新しい "Apache Axis 1.0" ランタイム・プロトコルです。

ユーザーはバージョン 5 プロジェクトと Web サービスに関連した EAR および WAR ファイルを、バージョン 5.1.1 で変更なしに再使用できるはずですが、Web サービスとクライアントを新規の IBM WebSphere 5.0.2 ランタイム・プロトコルに変換して、JSR 101、109、WS-I および WS-Security 標準を利用するためには、Web サービス・ウィザードを使用して再生成する必要があります。物理的データ・ファイルは自動的に新規のロケーションに移動しますが、Web サービス・エクスプローラーは、ユーザーの好みも継続して自動的に読みとります。

プロファイル作成ツール内で行った変更

ワークスペースをバージョン 5 からマイグレーションするとき、「ワークベンチの復元で問題が発生しました (Problems occurred restoring workbench)」というポップアップ・エラー・メッセージを受けとります。このメッセージは、プロファイル・パースペクティブがマイグレーションの時点でオープンされていて、ヒープまたはインスタンス統計ビューがプロファイル・パースペクティブで可視の場合に現れます。これは、バージョン 5 で存在していた「ヒープ (Heap)」ビューも「インスタンス統計 (Instance Statistic)」ビューも除去されているからです。このメッセージは、プロファイル・パースペクティブがマイグレーションの時点でワークスペース内にオープンされている場合にも現れます。このエラー・メッセージは「OK」をクリックして無視しても安全です。

テンプレート・ウィザードに関する既知の互換性の問題

バージョン 5 で作成されたカスタマイズしたテンプレートを使用するためには、ユーザーはカスタマイズしたテンプレートをロードし、それをデータベースに再接続して保管する必要があります。次回、ユーザーが保管したカスタマイズされたテンプレートを再ロードすると接続が検査されます。

このリリースで作成された生成済みの J2EE 1.2 成果物は、IBM WebSphere Studio Site Developer バージョン 4.0.3 では読み取れない可能性があります。バージョン 4.0.3 テスト環境では実行されます。バージョン 4.0.3 ではテンプレート・ウィザードが入っていなかったため、そのバージョンへの後方互換性は維持されません。

Web プロジェクト設定で Java ソース・フォルダーが「Java Source」と名付けられ、Web コンテンツ・フォルダーが「Web Content」と名付けられると、このリリースで生成されたテンプレート・アプリケーションをバージョン 5 で実行できません。

第 5 章 IBM WebSphere Studio Site Developer バージョン 4.0.x からのマイグレーション

この章では、IBM WebSphere Studio Site Developer バージョン 4.0.x からバージョン 5 へのマイグレーションについて説明します。

プロジェクトを IBM WebSphere Studio Site Developer バージョン 4.0.x からバージョン 5 へマイグレーションするために使用できるメソッドが、2 つ用意されています。

- Concurrent Versioning System (CVS) または Rational® ClearCase® などのソフトウェア構成管理 (SCM) システムを使用する。これがお勧めするメソッドです。
- プロジェクトをバージョン 4.0.x からエクスポートし、次にバージョン 5 へインポートする。このメソッドは、プロジェクト・ビルド・パス情報以外のすべてのものをマイグレーションします。
- 既存のバージョン 4.0.x ワークスペースを使用する。これは後述の理由によりサポートされません。

バージョン 4 からバージョン 5 へのマイグレーションは、バージョン 5 が WebSphere Application Server バージョン 4 へまだビルドおよびデプロイできるため、プロジェクト J2EE レベルを自動的に変更しません。Web プロジェクトを含むすべての J2EE プロジェクト・タイプは、IBM WebSphere Studio Site Developer で使用可能な J2EE マイグレーション・ウィザードを使ってマイグレーションすることができます。J2EE マイグレーション・ウィザードにアクセスするためには、J2EE タイプ・プロジェクトを右マウス・ボタンでクリックし、次に「**マイグレーション (Migrate)**」 > 「**J2EE マイグレーション・ウィザード (J2EE Migration Wizard)**」を選択します。

IBM WebSphere Studio Site Developer バージョン 4.0.x とバージョン 5 の間の相違点

以下は、バージョン 4.0.x 以降に行われた機能拡張の一部をリストしたものです。

- IBM WebSphere Studio Site Developer バージョン 5 は、WebSphere Application Server バージョン 4.0 またはバージョン 5 のいずれかのコードを生成でき、WebSphere Application Server バージョン 4.0.4 とバージョン 5 の両方のテスト環境を組み込みます。
- エンタープライズ・アプリケーション・アーカイブ (EAR) J2EE レベルが、WebSphere Application Server バージョン 5 プロジェクトの 1.2 から 1.3 に変更されました。
 - J2EE 1.2 EAR は、WebSphere Application Server バージョン 4.0.x または WebSphere Application Server バージョン 5 のいずれかで稼働します。
- Enterprise Java Beans (EJB) 仕様レベルが、WebSphere Application Server バージョン 5 EAR の 1.1 から 2.0 に変更されました。

- EJB 1.1 プロジェクトは引き続きサポートされ、WebSphere Application Server バージョン 4.0.x J2EE 1.2 EAR またはバージョン 5 J2EE 1.3 EAR のいずれかの一部となります。
- Web アプリケーション (WAR) J2EE レベルが、WebSphere Application Server バージョン 5 プロジェクトの 1.2 から 1.3 に変更されました。
 - JSP レベルが 1.1 から 1.2 に、また Servlet レベルが 2.2 から 2.3 に変更されました。
 - Dynamic 1.2 Web プロジェクト (WAR) は引き続きサポートされ、WebSphere Application Server バージョン 4.0.x J2EE 1.2 EAR またはバージョン 5 J2EE 1.3 EAR のいずれかの一部になります。
- バージョン 5 では、動的 Web プロジェクトだけでなく、静的 Web プロジェクトも作成できます。静的 Web プロジェクトでは、従来型の HTTP サーバー (HTML、JavaScript™、イメージ、テキストなど) によって提供されるコンテンツしか作成できません。
- (Eclipse オープン・ソース・プロジェクトをベースにしている) ワークベンチが、バージョン 1.0 からバージョン 2.0 に変更されました。
 - 新しい Java ビルダーは大幅に改善されています。
 - そこにはソフトウェア構成管理 (SCM) ベンダーのための新しい、より進歩したバージョン管理 (VCM) インターフェースが用意されています。

WebSphere Application Server の変更および Servlet/JSP 変換ツール

WebSphere InfoCenter

[www.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/index.html] には、以下の情報があります。

- WebSphere Application Server バージョン 3.5 と 4.0 の違いについての情報 [www.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/was/03.html]
- WebSphere Application Server バージョン 3.5 またはバージョン 4.0 からバージョン 5.0 へのマイグレーションについては、www.ibm.com/software/webservers/appserv/infocenter.html の WebSphere InfoCenter に移って以下のステップに従い、マイグレーション・ページをナビゲートしてください。
 1. 最初のリンク、「分散オペレーティング・システムのための **Application Server (Application Server for distributed operating systems)**」の中で、使用する言語を選択する。
 2. 上段左コーナーにあるリンク「アクティビティーごとのすべてのトピック (**All topics by activity**)」を選択する。
 3. リンク「**マイグレーション (Migrating)**」を選択する。

「*Migrating to WebSphere V5.0 An End-to-End Migration Guide*」はバージョン 3.5 およびバージョン 4.0 からバージョン 5.0 へマイグレーションするための良い情報源です [www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246910.pdf]。

WebSphere Application Server のダウンロード・ページ

[www14.software.ibm.com/webapp/download/product.jsp?s=p&id=TDUN-49EVRT&type=s&dt=DIAGNOSTIC+TOOL] には、ユーザーのアプリケーションの変換を支援するツールがあります。

- MigrateWC。このツールは .91 または 1.0 JSP を 1.1 JSP に変換します。また、2.1 Servlet から 2.2 Servlet への変換も行います。
- XMLconvert。このツールは、XML 構成ファイルをリリース 3.02x または 3.5x のフォーマットから 4.0 のフォーマットに変換します。

バージョン 4.0.3 からの内部変更

循環プロジェクトの依存関係はデフォルトでビルドしない

プロジェクトが循環依存関係を持っていれば、バージョン 5 はビルド・エラーを報告します。「ウィンドウ (Window)」>「設定 (Preferences)」>「Java」>「コンパイラ (Compiler)」とすすみ、「ビルド・パス (Build Path)」タブを選択して、「ビルド・パス・エラーでビルドの打ち切り (Abort building on build path errors)」チェック・ボックスを選択解除できます。これにより以降はビルドが停止されることなく、「タスク (Task)」ビューに 1 つまたは複数のビルドの「循環依存関係 (circular dependency)」エラーが (ビルドがそれ以外の点で正常な状態であっても) 依然として表示されるので、注意してください。この場合、ユーザーは、「その他 (Other)」タブを使ってこれらの「エラー (errors)」を「注意 (warnings)」に変更し、その後「循環依存 (Circular Dependencies)」ドロップダウンの設定を変更することが可能です。

バージョン 5 Web プロジェクトはバージョン 4.0.3 とソース・ロケーションの互換性がある

IBM WebSphere Studio Site Developer バージョン 5 では、バージョン 4.0.3 から内部プロジェクト構造が変更されています。バージョン 5 J2EE 1.2 Web WAR は、Java ソースと一緒にエクスポートされる時に、IBM WebSphere Studio Site Developer バージョン 4 にインポートされ、ソース・コード・フォルダーは自動的に正しい名前に変換され、ビルドも正常に行われます。バージョン 4 プロジェクトがバージョン 5 にインポートされる時、ソース・コード・フォルダーは自動的に正しい名前に変換されるため、Web プロジェクトは依然として WebSphere Application Server バージョン 4 上で正しく同様に実行されます。フォルダー名の変更について詳しくは、32 ページの『IBM WebSphere Studio Site Developer の Web プロジェクトの構造』を参照してください。

注: Web プロジェクトが、ソフトウェア構成管理 (SCM) システムを経由してバージョン 5 とバージョン 4 の間で共有されている場合、これは真実ではありません。バージョン 4 プロジェクトはバージョン 5 プロジェクト構造にマイグレーション済みであることが必要で、いったんマイグレーション済みになると SCM システムからバージョン 4 に戻ってロードされることはできません。

IBM WebSphere Studio Site Developer の Web プロジェクトの構造

IBM WebSphere Studio Site Developer バージョン 5 の Web プロジェクトの内部構造は、以前の IBM WebSphere Studio Site Developer バージョン 4.0.x の場合とは異なります。この違いは、J2EE 1.2 と J2EE 1.3 の違いとは関係なく、むしろツールの使いやすさの違いによるものです。

バージョン 4 では、Web プロジェクトはデフォルトで 動的 Web プロジェクトであって、「ナビゲーター (Navigator)」ビューに「ソース (source)」フォルダーおよび「webApplication」フォルダーを持つ形で表示されました。バージョン 5 では、動的 Web プロジェクトを作成すると、「ソース (source)」フォルダーではなく「Java ソース (Java Source)」フォルダーに、「webApplication」フォルダーではなく「Web コンテンツ (Web Content)」フォルダーに表示されます。

しかし、バージョン 4 の Web プロジェクトを SCM リポジトリに保管してからバージョン 5 にロードすると、「ソース (source)」フォルダーと「webApplication」フォルダーを持つ古い構造のまま表示されます。どちらの構造もバージョン 5 で正しくビルドされます。

静的プロジェクトと動的プロジェクトの対比

バージョン 5 では、動的 Web プロジェクトだけでなく静的プロジェクトも作成することができます。

静的 Web プロジェクトは、HTML、Java スクリプト、イメージ、テキストなどの静的リソースのみを含み、動的コンテンツを含みません。静的 Web プロジェクトは従来の HTTP Web サーバー上で実行され、サービスを受けていて Web アプリケーション・サーバーを必要としません。

動的 Web プロジェクトは静的リソースのほかに、サーブレット、JSP、フィルター、関連付けられたメタデータのような動的 J2EE リソースを含みます。動的 Web プロジェクトを作成する場合、豊富なプロジェクト・リソースのセットを用いて開発を始められるように、カスケード・スタイル・シートと JSP タグ・ライブラリーを組み込むことができます。動的 Web プロジェクトはいつも Enterprise Application プロジェクトに組み込まれていて、Web Application Servers 上でのみ実行されます。

HTML と JSP の相違点

- バージョン 4.0.x では、Page Designer が HTML ファイルと JSP ファイルを同様に処理していました。例えば、JSP タグを HTML ファイルで使用することができました。今後はこのような使用ができなくなります。つまりこのリリースでは、JSP ファイルと HTML ファイルが区別されますので、JSP タグを HTML ファイルに含めることができなくなります。
- 従来の区別では、英語以外の JSP ファイルのエンコードに影響を与えます。このリリースより前のバージョンでは、JSP ファイルの場合でも、ファイル内の指定エンコードを決定するために、HTML エンコード・ルールが使用されていました。つまり、メタ・タグのコンテンツ・タイプ属性が検索されました (<META http-equiv="Content-Type" content="text/html; charset=UTF-8">)。

今回のバージョンでは、これが変更され、JSP エンコード・ルールを使用して、JSP ファイル内の指定エンコードが決定されるようになりました。つまり、JSP ファイルのページ・ディレクティブが検索されます (<%@page contentType="text/html; charset=UTF-8"%>)。HTML ファイルの場合は、エンコード方式は従来のバージョンと変わりありません。

ソフトウェア構成管理 (SCM) システムを使用したプロジェクトのマイグレーション

CVS または Rational ClearCase を使用したプロジェクトのマイグレーション

ワークスペースをバージョン 4.0.x から IBM WebSphere Studio Site Developer バージョン 5 へ移動する場合は、この方法の使用をお勧めします。これがプロジェクト・ビルド・パスの情報を含む、すべての情報をマイグレーションするための唯一の方法です。

1. バックアップのために、すべてのバージョン 4 プロジェクトを SCM リポジトリに保管して、次に、保留中のすべての変更をコミット (リリース) する。
2. ユーザーが IBM WebSphere Studio Site Developer バージョン 4 およびバージョン 5 で作業することを望む場合は、作業結果を再度新規のバージョン 5 ブランチ (ストリーム) に保管する。このブランチは、バージョン 5 で作業する場合に使用します。
3. バージョン 5 をインストールする。
4. IBM WebSphere Studio Site Developer バージョン 4 をクローズし、IBM WebSphere Studio Site Developer バージョン 5 を開始します。

ヒント: バージョン 4 では、ワークスペース・ディレクトリーが、デフォルトで、インストール・ディレクトリーに入っていました。バージョン 5 では、このデフォルトが、「My Documents」ディレクトリー内の、「ワークスペース」と呼ばれるディレクトリーに変わりました。作業を保管するロケーションをオーバーライドしたい場合は、ワークベンチを始動するときに、コマンドの `-data` オプションを使用します。

注: `-data` を使用して既存のバージョン 4 ワークスペースをポイントしないでください。このオプションは、サポートされていない別個のマイグレーション・アプローチです (詳しくは、36 ページの『既存のバージョン 4.0.x ワークスペースを使用したプロジェクトのマイグレーション』を参照してください)。

5. 「ウィンドウ (Windows)」 > 「設定 (Preferences)」 > 「ワークベンチ (Workbench)」 > 「ビルドをリソース変更時に自動で実行 (Perform build automatically on resource modification)」 を使用不可にする (個々の従属プロジェクトをロードするときにビルド・エラーを起こさないようにするため)。
6. **CVS 用:** 処理したいすべてのプロジェクトを SCM リポジトリから IBM WebSphere Studio Site Developer バージョン 5 にロードする。

ClearCase の場合: クリーンなバージョン 5 ワークスペースを使用して、各プロジェクトにロードするため、「ファイル (File)」 > 「インポート (Import)」 >

「現在の WebSphere Studio 4.x ClearCase プロジェクトをワークスペースへ (**Existing WebSphere Studio 4.x ClearCase Project into Workspace**)」を選択する。

7. 「ウィンドウ (**Windows**)」>「設定 (**Preferences**)」>「ワークベンチ (**Workbench**)」>「ビルドをリソース変更時に自動で実行 (**Perform build automatically on resource modification**)」に対する希望の設定を復元する。
8. Web プロジェクトにおいてフル・ビルドが必要な場合、「ソース (**source**)」フォルダー名を「ソース (**source**)」から「Java ソース (**Java Source**)」へ、そして「webApplication」フォルダーから「Web コンテンツ (**Web Content**)」に変更する。そのようにしないと、古いフォルダー構造が保持されて Web プロジェクトが完全に再ビルドされなくなります。
9. フル再ビルド (「プロジェクト (**Project**)」>「すべて再ビルド (**Rebuild all**)」) を実行して、作成されたプロジェクトを新規バージョン 5 ストリームのリポジトリに再保管する (これらのリソースを現行のバージョン 4 ストリームと混合しないでください)。

注: これらのプロジェクトは、現在は、バージョン 5 プロジェクトになっており、IBM WebSphere Studio Site Developer バージョン 4.0.x では使用できません。

マイグレーション後の考慮事項:

- IBM WebSphere Studio Site Developer バージョン 4.0 では、プロジェクト内のファイルはバイナリー・フォーマットで CVS 内に保管されていました。バイナリー・フォーマットでは、ファイル内の復帰/改行は、他のプラットフォームで使用するための変換が行われず、UNIX®/Linux プラットフォーム環境では認識できません。混合プラットフォーム環境 (例えば、DOS/Windows および UNIX/Linux プラットフォーム) で作業している場合は、ソース・ファイルにテキストのマークを付けて (「チーム (**Team**)」>「CVS ASCII/バイナリー・プロパティーの変更 (**Change ASCII/Binary Property**)」を使用)、CVS に再保管することができます。
- CVS リポジトリからのバージョン 4 Web プロジェクトでは、「ウィンドウ (**Window**)」>「設定 (**Preferences**)」>「チーム (**Team**)」>「CVS」>「空ディレクトリーの枝取り (**Prune empty directories**)」の設定を使用不可にする必要があります (デフォルトでは使用可能)。使用不可にせずに、空のソース・フォルダーを持つ Web プロジェクト (Web 例 MyHomePage など) をロードすると、チェックイン時に以下のようなエラーが出ます。

```
The project was not built since it is involved in a cycle or has classpath problems.  
Missing required source folder: /MyHomePageExample403/source.
```
- ClearCase リポジトリに保管され、そこからロードされた Web プロジェクトの場合、ファイルをチェックアウトしないと、エディターで開くことはできません。ファイルをチェックアウトしなかった場合、「このビューの活動化中にエラー (error activating this view)」(Page Designer のログでの NULL ポインター例外) というエラーが示されます。xml エディターを使用して web.xml ファイルを編集する場合、web.xml、ibm-web-bnd.xmi、および ibm.web-ext.xmi をチェックアウトする必要があります。(状況表示行に、これらのファイルが読み取り専用であり、チェックアウトする必要があると示されていますが、見過ごしがちです。)

- プロジェクトが循環依存関係を持っていれば、バージョン 5 はビルド・エラーを報告します。「ウィンドウ (Window)」 > 「設定 (Preferences)」 > 「Java」 > 「コンパイラ (Compiler)」 とすすみ、「ビルド・パス (Build Path)」タブを選択して、「ビルド・パス・エラーでビルドの打ち切り (Abort building on build path errors)」チェック・ボックスを選択解除できます。
- .vcm_meta (または .cc_meta) ファイルはバージョン 5 では使用されない (バージョン 5 では代りに新しい .project ファイルが使用されます) ので、削除できるようになりました。バージョン 5 プロジェクトに関しては、新しいリポジトリ・ブランチ (ストリーム) が使用されています。これらのファイルは、引き続き現行のバージョン 4 ブランチ (ストリーム) に必要です。

EAR およびサーバー構成の絶対パス参照のマイグレーション後の除去

バージョン 4 EAR IBM アプリケーション拡張機能ファイルとサーバー構成ファイルは絶対パス参照を含んでいました。それらをバージョン 5 へマイグレーションしてから、バージョン 5 のエディターで開く必要があります (エディターは絶対パス参照を新しい相対参照へ自動的に変更します)。

1. それぞれの EAR プロジェクトごとに、「ナビゲーター (Navigator)」ビューで「META-INF/application.xml」 > 「アプリケーションを指定して開く (Open with)」 > 「デプロイメント・ディスクリプター・エディター (Deployment Descriptor Editor)」を右マウス・ボタンでクリックする。
 - a. ダイアログ・ウィンドウにメッセージがポップアップされる。

The IBM extensions file contains deprecated absolute paths. This can be auto-corrected and should be saved. This will remove the paths from the file, and only needs to be done once. Would you like to autocorrect?
 - b. 「はい (Yes)」をクリックする。
 - c. 保管して、エディター・ウィンドウを閉じる。

注: 代わりに、J2EE マイグレーション・ウィザードを使って EAR プロジェクトのみのためにプロジェクト構造をマイグレーションすることができます。J2EE マイグレーション・ウィザードにアクセスするためには、EAR プロジェクトを右マウス・ボタンでクリックし、次に「マイグレーション (Migrate)」 > 「J2EE マイグレーション・ウィザード (J2EE Migration Wizard)」を選択します。

2. 「サーバー・パースペクティブ (Server Perspective)」ビュー、「サーバー構成 (Server Configuration)」ビューのそれぞれのサーバー構成ごとに、「サーバー (server)」を右マウス・ボタンでクリックして「開く (Open)」を選択する。
 - a. 同様の自動訂正ダイアログが表示されます。
 - b. 「はい (Yes)」をクリックする。
 - c. 保管して、エディター・ウィンドウを閉じる。

その他の SCM を使用したプロジェクトのマイグレーション

IBM WebSphere Studio Site Developer の SCM プラグインを提供している SCM ベンダーは他にも何社かあります。

www.ibm.com/software/ad/studioappdev/partners/scm.htmlのベンダー・リストを参照し

てください。バージョン 4 プラグインを提供していたすべての SCM ベンダーは、それぞれの *Ready for IBM WebSphere Studio software* [www.developer.ibm.com/websphere/ready.html] 検証の一部として、従来のマイグレーション・ステップ (バージョン 4 から SCM リポジトリへの保管、リポジトリからバージョン 5 へのロード) がそれぞれのシステムでも機能することを確認します。

プロジェクトのエクスポートおよびインポートによるマイグレーション

1. IBM WebSphere Studio Site Developer バージョン 4.0.x では、プロジェクトを WAR ファイル、EAR ファイル、または JAR ファイルにエクスポートする (「ファイル (File)」 > 「エクスポート (Export)」)。
2. IBM WebSphere Studio Site Developer バージョン 5 では、WAR ファイル、EAR ファイル、または JAR ファイルをインポートする (「ファイル (File)」 > 「インポート (Import)」)。

注: プロジェクトのビルド・パス情報が保守されていないので、これは完全なマイグレーションではありません。

既存のバージョン 4.0.x ワークスペースを使用したプロジェクトのマイグレーション

このアプローチは部分的にしかサポートされておらず、マイグレーションは不完全です。ユーザー・インターフェース設定、デバッグ設定、およびほとんどの設定がすべて消失します。プロジェクト名、プロジェクト・ソース・ファイル、およびプロジェクト Java ビルド・パス (クラスパス) は保存されますが、それ以外のはすべて保証されません。このアプローチを使用するのは、SCM をサポートしていないシステムを使用している場合や、プロジェクト・ビルド・パス情報の保存が不可欠な場合、つまり、バージョン 4 からエクスポートしたプロジェクトをインポートするとそれらの情報が消失する場合に限ってください。次の方法により、既存のバージョン 4.0.x のワークスペースを使用することができます。

1. リポジトリに対する保留中のすべての変更をコミット (リリース) する。
2. すべてのパーспекティブをクローズし、IBM WebSphere Studio Site Developer バージョン 4 をシャットダウンする。
3. *workspace_directory* の内容をバックアップする。ここで、*workspace_directory* は、バージョン 4.0.x ワークスペースが入っているディレクトリーの完全修飾名です。デフォルトでは、バージョン 4.0.x ワークスペースのサブディレクトリーは、この製品がインストールされているディレクトリーと同じディレクトリーに入っています。IBM WebSphere Studio Site Developer バージョン 4.0.x を再度処理したい場合に、このバックアップが必要になります。バージョン 5 IDE からバージョン 4.0.x ワークスペースをポイントすると、そのワークスペースは、IBM WebSphere Studio Site Developer バージョン 4.0.x で使用できなくなります。
4. IBM WebSphere Studio Site Developer バージョン 5 をインストールする。
5. バージョン 4.0.x ワークスペースを持つ IBM WebSphere Studio Site Developer バージョン 5 をコマンド・プロンプトから始動する場合 (つまり、コマンド

の `-data` オプションを使用して、バージョン 4.0.x ワークスペース・ディレクトリーに対する完全修飾パスを指定する場合)、`.metadata` 情報のアップグレードの原因になります。

6. 新規のユーザー・インターフェース・フォーマットに変換したいかどうかを確認するプロンプトが出たら、「OK」をクリックする。
7. 再ビルドを行う前、またはワークスペース内のプロジェクトを妥当性検査する前に、「リソース (Resource)」パースペクティブの「ナビゲーター (Navigator)」ビューに入っているすべてのプロジェクトを選択して、ポップアップ・メニューから「最新表示 (Refresh)」を選択する。こうすれば、すべてのファイルがそれぞれのメタデータと同期するようになります。
8. 閉じているすべてのプロジェクトを開く (以下の既知の問題を参照)。
9. クラスパス変数を検査する (以下の既知の問題を参照)。
10. このバージョン 5 では、一部のビルダーやバリデーターが追加、除去、または変更されています。エラーや警告が正しく表示されるようにするには、「プロジェクト (Project)」>「すべて再ビルド (Rebuild All)」を選択してすべてのプロジェクトを再ビルドしてから、各 Java プロジェクトごとに「検証の実行 (Run Validation)」を選択します。
11. ユーザー・プリファレンスは、一部を除きほとんどが維持されません。バージョン 5 のプリファレンス設定が希望どおりの設定になっているか調べてください。

EAR およびサーバー構成の絶対パス参照のマイグレーション後の除去

35 ページの『EAR およびサーバー構成の絶対パス参照のマイグレーション後の除去』で説明したマイグレーション後の指示がここでも適用されます。

既知の問題および制限事項

IBM WebSphere Studio Site Developer バージョン 5 で バージョン 4.0 のワークスペースを開いてマイグレーションしようとする、次のような問題が発生することがあります。

JRE_LIB クラスパス変数が無効な値になる

JRE_LIB クラスパス変数を有効なロケーションに設定し直すには、以下のステップを実行します。「設定 (Preferences)」ウィンドウを最初に開くときは、値が正しいように見える場合であっても、この操作を実行してください。

1. 「ウィンドウ (Window)」>「設定 (Preferences)」>「Java」>「インストール済み JRE (Installed JREs)」を選択する。
2. このリストで、JRE_LIB を設定するデフォルトの JRE ロケーションのチェック・ボックスを選択する。
3. 「編集 (Edit)」を選択し、次に「OK」をクリックして「JREの編集 (Edit JRE)」ダイアログ・ボックスを閉じる。

この操作を行わないと、JRE_LIB の値が正しくなくなり、Java ファイルに多くのビルド・エラーを生じさせる原因になる可能性があります。

一般的な検査として、他のすべてのクラスパス変数の値を調べてください。

以前の SCM 共用プロジェクトの「チーム (Team)」メニューに「共用プロジェクト (Share Project)」が含まれている

Eclipse 1.0 と 2.0 の間で、チーム・サポートが大幅に変更されました。プロジェクトをリポジトリと共用する方式も変更されました。

- プロジェクトを右マウス・ボタンでクリックし、「**チーム (Team)**」>「**プロジェクトの共用 (Share Project)**」オプションを選択すると、ウィザードによりマイグレーション・プロセスがガイドされます。それが終了すると、プロジェクトが共用され、「同期 (Synchronize)」ビューが開きます。矛盾する変更がどのファイルにも見られます。これは、Eclipse 1.0 と 2.0 の間で同期情報の保管の仕方が変わったからです。
- 出力変更がなければ (つまり、上で推奨したアップグレード手順を実行する前に、すべての出力変更をコミットした場合は、出力変更は発生しないはずです)、単に「同期 (Synchronize)」ビューのプロジェクトを選択して、「**オーバーライドおよび更新 (Override and Update)**」を選択するだけで、現在の内容がサーバーからロードされます。
- 出力変更が発生した場合は、「同期 (Synchronize)」ビューの三角メニューをプルダウンして、「**ファイル内容の比較 (Compare File Contents)**」を選択することができます。なんらかの作業を行うと、実際に異なるファイルのみが「同期 (Synchronize)」ビューに表示されます。次に、「同期 (Synchronize)」ビューを使用してこれらの矛盾を解決することができます。

ワークスペース・ディレクトリ外でプロジェクトが作成される

デフォルトでは、プロジェクトはワークスペース・ディレクトリに作成されます。このデフォルトをオーバーライドしてプロジェクトを別の場所に作成する場合は、ワークベンチを閉じる前にすべてのプロジェクトを開く必要があります。これにより、そのプロジェクトの .project ファイルが適切なロケーションに書き込まれます。ディレクトリがワークスペースの外側にある閉じたプロジェクトを開くことができないと、実際のプロジェクトをマスクするプロジェクトが作成され、そこには .project ファイルしか入りません。

JSP ブレークポイントの再設定が必要

現在設定されているすべての JSP ブレークポイントを除去して、それらをマイグレーション済みバージョン 5 ワークスペース内に再設定する必要があります。

4.0.3 Web プロジェクトのリレーショナル・データのマイグレーション

IBM WebSphere Studio Site Developer 4.0.3 のプロジェクトからリレーショナル・データをマイグレーションするには、以下のようにします。

1. IBM WebSphere Studio Site Developer 4.0.3 のワークスペースから、使用可能な各データベースに対する DDL ファイルを生成する。
2. Web プロジェクトのソース/データベース・フォルダーからデータベースを除去する (「データ定義 (Data Definition)」ビューを使用)。
3. IBM WebSphere Studio Site Developer バージョン 5 で 4.0.3 のワークスペースを開く。
4. リレーショナル・データを復元したい Web プロジェクトをマイグレーションする。

5. 「ファイル (File)」 > 「インポート (Import)」 > 「ファイル・システム (File System)」をクリックし、4.0.3 ワークスペースから DDL ファイルを指定する。
6. 「データ (Data)」パースペクティブの「データ定義 (Data Definition)」ビューで、「ローカルに対して実行 (Run against Local)」を選択して、ターゲット Web プロジェクトを指定する。

リレーショナル・データの成果物が復元されます。

4.0.x から Web サービス・ファイルをインポートした後の WSDL エラー

4.0.x から Web サービス・ファイルをインポートすると、以下のエラー・メッセージを受け取る場合があります。

```
Error The part 'result' has an invalid value 'anyElement' defined for its type. Type declarations must refer to valid values defined in a schema.
```

```
Error The part 'return' has an invalid value 'findPatientResult' defined for its element. Element declarations must refer to valid values defined in a schema.
```

```
Error The part 'response' has an invalid value 'findPatientResponse' defined for its element. Element declarations must refer to valid values defined in a schema.
```

このエラーの対応策は、以下のとおりです。

1. WSDL ファイルを削除する。
2. Web サービス・ウィザードを再実行して、Web サービスを再生成する。

J2EE プロジェクト構造および/または J2EE 仕様レベルのマイグレーション

バージョン 5 の J2EE マイグレーション・ウィザードにアクセスするには以下のステップに従ってください。

1. プロジェクトを選択する。
2. そのプロジェクトを右マウス・ボタンでクリックして、次に「マイグレーション (Migrate)」 > 「J2EE マイグレーション・ウィザード (J2EE Migration Wizard)」を選択する。マイグレーションをガイドするウィザードのステップに従ってください。
3. プロジェクトがソース制御の下にある場合は、その再構造化されたプロジェクトを SCM へ保管する。

第 6 章 WebSphere Studio "Classic" から IBM WebSphere Studio Site Developer へのマイグレーション

この章では、WebSphere Studio バージョン 4.0 (アドバンスド版とプロフェッショナル版の両方) から IBM WebSphere Studio Site Developer にマイグレーションする方法を説明します。WebSphere Studio "Classic" バージョン 4.0 から IBM WebSphere Studio Site Developer バージョン 5.0 へのマイグレーションは、以下のステップから成っています。

1. マイグレーション用に新しいシングル・サーバー・ステージを作成する。
2. Web 構成記述子ファイルを作成する。
3. マイグレーション JAR ファイルをエクスポートする。
4. マイグレーション JAR ファイルを IBM WebSphere Studio Site Developer にインポートする。
5. サーバーをセットアップして、マイグレーション済みのアプリケーションをテストする。

注: 以下の説明は、WebSphere Studio バージョン 4.0 からのマイグレーションを対象としています。WebSphere Studio のそれ以前のバージョンからマイグレーションする場合は、まず WebSphere Studio 4.0 にマイグレーションしてから IBM WebSphere Studio Site Developer にマイグレーションしてください。

WebSphere Studio "Classic" の拡張公開フィーチャー (ファイルを公開ステージにマップする機能) および Page Detailer フィーチャー (Web ページの分析) は IBM WebSphere Studio Site Developer では使用できません。バージョン 4.0.x CD メディア・パックのその他のフィーチャーの一部も使用できません。たとえば、Web ページを分析する Page Detailer フィーチャー、リッチ・メディア・タイプ用の HotMedia[®] フィーチャー、Voice XML エディター (WebSphere Everyplace[™] ツールキットとポータル・ツールキットへ移されました)、パーベイスブ・デバイス用の DataBaseWizard がその例です。

WebSphere Studio のデータをマイグレーションする場合は、以下の制限事項に注意してください。

- IBM WebSphere Studio Site Developer は XML ベースの SQL エディターを使用しているため、.sql ファイルは使用できません。
- プロジェクトの公開情報とステージ情報をマイグレーションすることはできません。
- WebSphere Studio のサーバー構成情報をマイグレーションすることはできません。
- バージョン管理情報をマイグレーションすることはできません。

以下に説明するマイグレーション・プロセス中に、WebSphere Studio は、シングル・サーバー用のすべてのプロジェクト・ファイル (公開可能なものとソース) を収めた JAR ファイルを作成します。「公開中 (Publishing)」ビューに表示される、デ

フォルト・サーバーのファイルはすべて、JAR ファイルにパッケージされます。JAR ファイルは IBM WebSphere Studio Site Developer にインポートできます。

既存のプロジェクトをマイグレーションする場合、プロジェクトの公開情報とステージ情報は、すべてマイグレーション時に消失します。ステージが複数のサーバーにまたがる場合は、デフォルト・サーバーに公開されるファイルしか保存されません。したがって、マイグレーション用として、サーバーが 1 つだけの新しいステージを作成します。

マイグレーションのための新規シングル・サーバー・ステージの作成

現行のステージに複数のサーバーがある場合は、以下のステップを実行して、1 つのサーバーのみの「Migration」という新しいステージを作成します。

1. 「プロジェクト (Project)」 > 「公開ステージのカスタマイズ (Customize Publishing Stages)」をクリックする。
2. 「ステージ名 (Stage name)」フィールドに Migration と入力する。
3. 「追加 (Add)」をクリックする。
4. 「OK」をクリックする。
5. 「プロジェクト (Project)」 > 「公開ステージ (Publishing Stage)」をクリックして、選択可能なステージのリストから 「マイグレーション (Migration)」 を選択する。
6. 「公開中 (Publishing)」ビューで、「挿入 (Insert)」 > 「サーバー (Server)」をクリックする。
7. localhost などのサーバー名を入力する。
8. サーバーを変更したり、公開ステージを変更したりしても、WebSphere Application Server バージョン 4.0 のサーブレット・マッピング情報は伝搬されません。「公開中 (Publishing)」ビューにジャンプして、サーブレットごとに 「プロパティ (Properties)」 > 「公開 (Publishing)」 > 「サーブレット・マッピング (Servlet Mapping)」をクリックして、実際のサーブレット・マッピングをコピーする。

Web 構成記述子ファイルの作成

1. プロジェクト・ファイル・ビューで、「プロジェクト (Project)」 > 「Web 構成記述子ファイルの作成 (Create Web Configuration Descriptor File)」をクリックする。
2. 必要なサーブレットをすべて選択する。
3. 必要なタグ・ライブラリー記述子 (TLD) ファイルをすべて選択する。
4. 「作成 (Create)」をクリックする。

デフォルトの Web 構成記述子ファイル名は `serverName_web.xml` です。ここでは、`localhost_web.xml` になります。別のロケーションを指定しない限り、.xml ファイルは WEB-INF ディレクトリーに保管されます。

マイグレーション JAR ファイルのエクスポート

1. 「プロジェクト・ファイル (project file)」ビューで、サーバー **localhost** を選択して「プロパティ (Properties)」>「公開 (Publishing)」>「WebApp Webパス (WebApp Web Path)」をクリックし、myWebPath などの Web パス (コンテキスト・ルート) を入力する。これは、WebSphere Application Server - Express のプロジェクト名としても使用されます。
2. 「プロジェクト・ファイル (project file)」ビューで、「プロジェクト (Project)」>「マイグレーション・ファイルの作成 (Create Migration file)」を選択する。
3. サーバーとして **localhost** が選択されていることを確認する。
4. Web 構成記述子ファイルとして **localhost_web.xml** が選択されていることを確認する。
5. 「OK」をクリックする。
6. デフォルトの JAR ファイル名は **serverName.jar** です。ここでは、localhost.jar になります。必要に応じて、ファイル名を変更します。
7. JAR ファイルを保管する。

マイグレーション JAR ファイルの IBM WebSphere Studio Site Developer へのインポート

1. IBM WebSphere Studio Site Developer を始動する。
2. Web プロジェクトを作成する (「ファイル (File)」>「新規作成 (New)」>「プロジェクト (Project)」>「Web プロジェクト (Web Project)」)。
3. 「プロジェクト名 (Project name)」フィールドに Web プロジェクトの名前を入力する。この名前は、上記の43 ページの『マイグレーション JAR ファイルのエクスポート』のセクションのステップ 1 で指定した名前と同じでなければなりません。
4. デプロイメントを目的とする 新規 Web プロジェクトを含む新規または既存の EAR プロジェクト名を指定する。
5. 「コンテキスト・ルート (Context Root)」フィールドに、WebSphere Studio でマイグレーション JAR ファイルを作成したときに指定した Webapp Web パス名を入力する。「終了 (Finish)」をクリックする。
6. 「ナビゲーター (Navigator)」ビューで、作成したばかりの Web プロジェクトを選択する。
7. JAR ファイルをインポートする。
 - a. 「ファイル (File)」>「インポート (Import)」をクリックする。
 - b. 「WAR ファイル (WAR file)」をクリックして、「次へ (Next)」をクリックする。JAR ファイルは WAR ファイル・オプションを使用してインポートしなければなりません。そうしないと、正しく動作しません。
 - c. 「WAR ファイル (WAR File)」フィールドに localhost.jar のパスを入力するか、「参照 (Browse)」をクリックして検索する (ブラウザできるのは .WAR 名だけで、.JAR 名はできません)。

- d. ユーザーの作成した既存の Web プロジェクトを選択する。「コンテキスト・ルート (Context Root)」フィールドには以前指定した値が自動的に入力されます。
 - e. 「終了 (Finish)」をクリックする。「リソース WEB-INF/web.xml はすでに存在しています。上書きしますか？ (Resource WEB-INF/web.xml already exists. Would you like to overwrite it?)」という質問のダイアログが現れます。
 - f. 「はい (Yes)」を選択し、IBM WebSphere Studio Site Developer に *localhost.jar* を解凍させる。
8. 未解決の参照や欠落したインポート・ファイルがあれば、「タスク (Tasks)」ビューに表示されます。修正するには、Web プロジェクトの Java ビルド・パスを変更しなければなりません。
 - a. プロジェクトを右マウス・ボタンでクリックして、「プロパティ (Properties)」>「Java ビルド・パス (Java Build Path)」をクリックする。
 - b. 「ライブラリー (Libraries)」タブをクリックする。「外部 JAR の追加 (Add External JARs)」をクリックする。
 - c. 次のディレクトリーから必要な JAR をインポートする。
 - WS_Installdir/runtimes/aes_v4/lib および
 - WS_Installdir/runtimes/base_v4/lib
 9. 「ナビゲーター (Navigator)」ビューで、プロジェクトを右マウス・ボタンでクリックして、「プロジェクトの再ビルド (Rebuild Project)」を選択する。

テスト・サーバー上でマイグレーション済みアプリケーションのテスト

アプリケーションをテストする準備が整いました。デフォルトのテスト・サーバーでアプリケーションをテストする場合は、以下のステップを実行してください。


1. EAR プロジェクトを右マウス・ボタンでクリックする。
2. 「サーバーで実行 (Run on Server)」を選択する。

デフォルト以外のサーバーランタイム環境でアプリケーションをテストする場合は、オンライン・ヘルプで「サーバー・ツール (Server Tools)」フィーチャーを参照してください。

第 7 章 VisualAge for Java から IBM WebSphere Studio Site Developer へのマイグレーション

この章では、VisualAge® for Java プロフェッショナル版または VisualAge for Java エンタープライズ版を IBM WebSphere Studio Site Developer にマイグレーションする方法を説明します。

注: この章で示している手順は、Windows 版の VisualAge for Java バージョン 3.5.3 または 4.0 からマイグレーションするためのものです。旧バージョンの VisualAge for Java から IBM WebSphere Studio Site Developer にマイグレーションする場合は、まず旧バージョンの VisualAge for Java から または Windows 版のバージョン 3.5.3 または 4.0 にマイグレーションしてから、IBM WebSphere Studio Site Developer にマイグレーションしてください。

注:  IBM のビジネス・パートナーである Instantiations, Inc. が配布している CodePro Studio という製品には、マイグレーション機能および共存機能があり、VisualAge for Java と WebSphere Application Server - Express の生産性を向上させます。VisualAge for Java のユーザーがマイグレーションを開始できるように、Instantiations は、無料で利用制限のない VisualAge for Java を IBM WebSphere Studio Site Developer エクスポート機能に CodePro Studio の期間限定評価版の一部として提供しています。評価版は、www.instantiations.com/vaj-migrate からダウンロードできます。ファイルの完全双方向エクスポート/インポート、エクスポート/インポート・セットの作成、プロジェクトの同期化、タスクの自動化など、インスタンス化の拡張マイグレーションと共存サポートの詳細については、Instantiations, Inc. の Web ページ www.instantiations.com/codepro/ws を参照してください。

VisualAge for Java と IBM WebSphere Studio Site Developer の相違点

以下に、VisualAge for Java からの変更点の一部をリストします。

- Enterprise Java Beans (EJB) 仕様レベルが、1.0 から 1.1 に変更されました (WebSphere Application Server バージョン 5 にデプロイされるアプリケーションでは、EJB 2.0 もサポートされます)。
- Web アプリケーションの場合、JSP レベルは 1.1 のままです (WebSphere Application Server バージョン 5 アプリケーションの場合は 1.2)。
- Web アプリケーションの場合、サーブレット・レベルは 2.2 のままです (WebSphere Application Server バージョン 5 アプリケーションの場合は 2.3)。
- サポートされる Java 2 プラットフォームのレベルが 1.2 から 1.3 に変更されました (コンパイラーは 1.4 コード生成を行うことができますが、WebSphere Application Server ランタイム環境は 1.3 のままです)。
- Visual Composition Editor が Visual Editor for Java に置き換えられました。

- VisualAge for Java バージョン管理とプロプラエタリー・ソース・コード・リポジトリが、ソフトウェア構成管理 (SCM) プラグインに対するサポートに置き換えられました。
- VisualAge for Java Tools API が、WebSphere Studio Workbench プラグイン・アーキテクチャーに置き換えられました。
- VisualAge for Java XML ツールが、IBM WebSphere Studio Site Developer XML ツールに置き換えられました。
- VisualAge for Java プロジェクト概念が、各種タイプの IBM WebSphere Studio Site Developer プロジェクトに置き換えられました。

VisualAge for Java からのマイグレーション

以下のステップは、VisualAge for Java からマイグレーションする方法を示したものです。その後で、それぞれのステップの実行方法を詳しく説明します。

1. VisualAge for Java からの Java ファイルおよびプロジェクト・リソース・ファイルのエクスポート
2. IBM WebSphere Studio Site Developer の始動、およびコードに含める新しいプロジェクトの作成
3. Java ファイルとプロジェクト・リソース・ファイルの IBM WebSphere Studio Site Developer へのインポート
4. web.xml エディターを使用した、サーブレットが正しく定義されていることの確認 (Web プロジェクトのみ)。
5. プロジェクト設定とワークスペース設定のマイグレーション
6. サーバーのセットアップおよびマイグレーション済みのアプリケーションのテスト
7. IBM WebSphere Studio Site Developer から WebSphere Application Server へのアプリケーションのデプロイ
8. 複数の開発者間での IBM WebSphere Studio Site Developer プロジェクト設定の共用 (事後マイグレーション)

VisualAge for Java からの Java ファイルおよびプロジェクト・リソース・ファイルのエクスポート

VisualAge for Java リポジトリからバージョン管理されたプロジェクトやリソースを「まとめて」マイグレーションする方法はサポートされていません。VisualAge for Java ワークスペースにあるプロジェクトとリソースだけをマイグレーションできます。バージョン管理されたプロジェクトやリソースは、まず VisualAge for Java ワークスペースに移動してから IBM WebSphere Studio Site Developer にマイグレーションする必要があります。

注: プロジェクトに複数の種類のデータ (例えば、エンタープライズ Bean と Java ソース・コード・ファイル) が入っている場合は、タイプに基づいてデータを別々の JAR に分割してください。

プロジェクトを JAR ファイルにエクスポートする場合は、以下のステップを実行してください。

1. エクスポートするプロジェクトが現在 VisualAge for Java ワークスペース内にはない場合は、ワークスペースに追加する。
2. 「VisualAge for Java ワークベンチ (VisualAge for Java Workbench)」ウィンドウで、プロジェクトを選択して右マウス・ボタンでクリックし、「エクスポート (Export)」をクリックする。
3. 「JAR ファイル (Jar file)」ラジオ・ボタンを選択して、「次へ (Next)」をクリックする。
4. JAR ファイルの名前を指定する。
5. 「.java」チェック・ボックスを選択して Java ファイルをエクスポートし、「リソース (resources)」チェック・ボックスを選択してリソース・ファイルをエクスポートする。
6. 必要に応じて、その他のフィールドに入力する。この作業の実行方法の詳細については、VisualAge for Java のオンライン・ヘルプを参照してください。

IBM WebSphere Studio Site Developer の始動、およびコードに含める新しいプロジェクトの作成

IBM WebSphere Studio Site Developer を始動し、所要のプロジェクトを作成します。以下は、ファイルのインポート先となる IBM WebSphere Studio Site Developer プロジェクトの種類を決める場合に役立つ、一般的なマイグレーション・ガイドラインです。

- コードが Web アプリケーションの一部である場合は、そのコードを Web プロジェクトにインポートする。
 - Java ファイルはすべて Web プロジェクトの **Java ソース・ディレクトリ** にインポートする (IBM WebSphere Studio Site Developer により、パッケージのステートメントに基づき適切な階層が自動的に作成されます)。
 - リソース・ファイルはすべて Web プロジェクトの **Web コンテンツ・ディレクトリ** にインポートする。
- コードが純粋な Java (例えば、スタンドアロンで実行されるアプリケーション) の場合は、そのコードを Java プロジェクトにインポートする。

注: 前述の内容は、どの種類の IBM WebSphere Studio Site Developer プロジェクトを使用するかを決める上での一般的なガイドラインに過ぎません。プロジェクトを作成したり、コードをインポートしたりする場合は、事前に IBM WebSphere Studio Site Developer のオンライン・ヘルプを参照し、各種の WebSphere Application Server - Express プロジェクトをよく理解しておくことをお勧めします。

Java ファイルとリソース・ファイルの IBM WebSphere Studio Site Developer へのインポート

1. IBM WebSphere Studio Site Developer を開き、「リソース (Resource)」パースペクティブに切り替える。
2. 「ファイル (File)」>「インポート (Import)」>「ZIP ファイル (Zip file)」をクリックする。「次へ (Next)」をクリックする。

3. 対象の JAR ファイルを参照する。
4. インポートするファイル、およびファイルを入れるプロジェクトまたはフォルダーを選択する。

注:

- ZIP インポートの代わりにファイル・システムのインポートも使用可能ですが、ZIP インポートの方がより一般的です。インポート・ウィザード を使用して ZIP ファイルをインポートするのではなく、フォルダーを指定してファイルをインポートすることについても同じ説明が使用できます。
- ファイルを IBM WebSphere Studio Site Developer にインポートする場合は、適切なディレクトリーに入れるようにしてください。コードをインポートする場合は、事前に IBM WebSphere Studio Site Developer のオンライン・ヘルプを参照し、各種の IBM WebSphere Studio Site Developer プロジェクトをよく理解しておくことをお勧めします。どのフォルダーにどのコードを入れるかを判別する際に役立ちます。

サーブレットが正しく定義されていることを確認するための web.xml エディターの使用 (Web プロジェクトのみ)

アプリケーションでサーブレットを使用している場合は、web.xml ファイルでサーブレットと URL のマッピングを定義する必要があります。以下のステップを実行してください。

1. 「Web」パースペクティブで、web.xml ファイルを開く。このファイルは、Web プロジェクトの Web Content/WEB-INF サブディレクトリーに入っています。
2. 「サーブレット (Servlets)」タブをクリックする。
3. 「追加 (Add)」をクリックして、「サーブレット (Servlet)」ラジオ・ボタンを選択する。
4. サーブレットの名前を入力して、「OK」をクリックする。
5. 「参照 (Browse)」をクリックして、「サーブレット・クラス (Servlet class)」の値を適切なパッケージ名に変更する。
6. (オプション) 表示名は、サーブレットの識別に使用するショート・ネームです。「表示名 (Display name)」フィールドに、サーブレットのショート・ネームを入力する。
7. URL マッピングで、サーブレットと URL パターンを定義する。「URL マッピング (URL mappings)」フィールドの横にある「追加 (Add)」ボタンをクリックし、URL マッピングの名前を入力する。
8. 変更を保管して (「ファイル (File)」> 「web.xml の保管 (Save web.xml)」)、web.xml ファイルを閉じる。

プロジェクト設定とワークスペース設定のマイグレーション

以下の VisualAge for Java 設定を書き留め、IBM WebSphere Studio Site Developer で設定してください。

- プロジェクトのクラスパス
- リソース関連
- コードのフォーマット設定

- WTE の構成
- Java ファイルおよびプロジェクト・リソース・ファイル

プロジェクトのクラスパス

VisualAge for Java では、「オプション (Options)」ウィンドウの「リソース (Resources)」ページ (「ウィンドウ (Window)」 > 「オプション (Options)」 > 「リソース (Resources)」) で、プロジェクトのクラスパスを設定します。プロジェクトを IBM WebSphere Studio Site Developer にマイグレーションした後でプロジェクトのクラスパスを設定するには、プロジェクトの「プロパティ (Properties)」ウィンドウを使用します (プロジェクトを右マウス・ボタンでクリックして、「プロパティ (Properties)」 > 「Java ビルド・パス (Java Build Path)」を選択します)。「ライブラリー (Libraries)」タブをクリックします。また、「設定 (Preferences)」ウィンドウでクラスパスの変数を設定することもできます (「ウィンドウ (Window)」 > 「設定 (Preferences)」 > 「Java」 > 「クラスパス変数 (Classpath Variables)」)。

リソース関連

ファイル・タイプと実行可能プログラムとの関連をセットアップすれば、ワークベンチの外側にあるファイルをワークベンチの中から開くことができます。

VisualAge for Java では、「オプション (Options)」ウィンドウでリソース関連をセットアップします (「ウィンドウ (Window)」 > 「オプション (Options)」 > 「リソース (Resources)」 > 「リソース関連 (Resource Associations)」)。リソース・ファイルを IBM WebSphere Studio Site Developer にマイグレーションした後にリソース関連をセットアップするには、「設定 (Preferences)」ウィンドウを使用します (「ウィンドウ (Window)」 > 「設定 (Preferences)」 > 「ワークベンチ (Workbench)」 > 「ファイル関連 (File Associations)」)。

コード・フォーマッター

VisualAge for Java では、「オプション (Options)」ウィンドウの「フォーマッター (Formatter)」ページで、コードのフォーマット・オプションをセットアップします (「ウィンドウ (Window)」 > 「オプション (Options)」 > 「コーディング (Coding)」 > 「フォーマッター (Formatter)」)。コードを IBM WebSphere Studio Site Developer にマイグレーションした後にコードのフォーマット設定をセットアップするには、「設定 (Preferences)」ウィンドウを使用します (「ウィンドウ (Window)」 > 「設定 (Preferences)」 > 「Java」 > 「コード・フォーマッター (Code Formatter)」)。

WTE の構成

VisualAge for Java では、WebSphere 単体テスト環境と WebSphere Application Server ランタイムの設定は、*VisualAgeInstalldir*\%ide\project_resources\IBM WebSphere Test Environment\properties ディレクトリー (*VisualAgeInstalldir* は、製品のインストール・ディレクトリー) の下のさまざまなプロパティ・ファイルに入っています。

例えば、session.xml プロパティ・ファイルで、

```
<url-rewriting-enabled>true</url-rewriting-enabled>
```

 のようにプロパティを

true に変更して URL 再書き込みを使用可能にした場合、IBM WebSphere Studio Site Developer バージョン 4.0 テスト環境で、このプロパティを構成できます。(「サーバー (Server)」パースペクティブで、「サーバー構成 (Server Configuration)」ビューを開き、利用したいサーバーを右マウス・ボタンでクリックし、「開く(Open)」をクリックする。「Web」タブをクリックして、「URL 再書き込みを使用可能にする (Enable URL rewrite)」チェック・ボックスを選択する。)

Java ファイルおよびプロジェクト・リソース・ファイル

プロパティ・ファイル default.servlet_engine には、VisualAge for Java Web アプリケーションの <root-uri> コンテキスト・ルートが入っています。IBM WebSphere Studio Site Developer で Web プロジェクトを作成する場合、「Web プロジェクトの作成 (Create a Web Project)」ダイアログに、このデータの「コンテキスト・ルート (Context root)」フィールドがあります。

ユーザーが自らカスタマイズした、*VisualAgeInstallDir* ¥ide¥project_resources¥IBM WebSphere Test Environment¥hosts¥default_host¥default_app¥servlets¥default_app.webapp などのファイル内の Web アプリケーション設定は、IBM WebSphere Studio Site Developer 内の *your_Web_project*¥Web Content¥WEB-INF¥web.xml ファイルにマイグレーションする必要があります。例えば、default_app.webapp ファイルでサーブレット名とサーブレット・パスを変更した場合は、web.xml ファイルでも対応する変更を行います。

WebSphere V4 テスト環境のセットアップとマイグレーション済みアプリケーションのテスト

アプリケーションが Java プロジェクトの場合は、Java プロジェクトに対する標準の IBM WebSphere Studio Site Developer の**実行 (Run)** サポートまたは**デバッグ (Debug)** サポートだけを使用してそのアプリケーションをテストします。

アプリケーションが WebSphere Application Server を使用している場合は、組み込み WebSphere Application Server を使用してそのアプリケーションをテストします。このためには、デフォルトのテスト・サーバーを作成して、始動する必要があります。Web プロジェクトの場合は、メイン HTML ページを右マウス・ボタンでクリックし、「サーバーで実行 (Run on Server)」を選択して、Web ブラウザーを起動します。

その他のタイプのプロジェクトのテスト方法については、オンライン・ヘルプを参照してください。

IBM WebSphere Studio Site Developer からリモート WebSphere Application Server へのアプリケーションのデプロイ

WebSphere Application Server をランタイム環境として使用している場合は、IBM WebSphere Studio Site Developerの「サーバー・ツール (Server Tool)」フィーチャーを使用してアプリケーションをデプロイします。

複数の開発者間での IBM WebSphere Studio Site Developer プロジェクト設定の共用 (事後マイグレーション)

IBM WebSphere Studio Site Developer プロジェクト (およびその関連する設定) は、開発者間で共用することができます。このためには、プロジェクトを IBM WebSphere Studio Site Developer のソフトウェア構成管理 (SCM) サーバーに保管してから、IBM WebSphere Studio Site Developer の別のチーム・メンバーに取り出します。

IBM WebSphere Studio Site Developer のチーム・サポート

IBM WebSphere Studio Site Developer バージョン 4.0 のチーム・サポートについては、
www.ibm.com/websphere/developer/library/techarticles/0108_karasiuk/0108_karasiuk.html
を参照してください。

また、IBM WebSphere Studio Site Developer のチーム・サポートについては、「インストール・ガイド」やオンライン・ヘルプも参考になります。

第 8 章 VisualAge for Java Visual Composition Editor から Visual Editor for Java へのマイグレーション

この章では、VisualAge for Java の Visual Composition Editor フィーチャーで作成したアプリケーションを WebSphere Application Server - Express の Visual Editor for Java にマイグレーションする方法を説明します。

- VisualAge for Java から拡張設計時メタデータを保管する。
- マイグレーションを完了する (WebSphere Studio ヘインポートする)。

VisualAge for Java からの拡張設計時メタデータの保管

このステップはオプションですが以下の理由のため、(特に、アプリケーションが接続されている場合は) 強くお勧めします。

- VisualAge for Java は、フリー・フォーム域のトップレベル Bean の配置に関する情報は保管しませんでした (他の Bean 内に組み込まれない Bean があることを意味しています)。これに対して、WebSphere Studio は常にこの情報を Bean を宣言するコード行にコメントとして保管し、Visual Editor を開いたときに Bean をいつもキャンバスの同じ位置に復元します。この VisualAge for Java の設計時情報を、マイグレーションする前に保管するオプションがあります。その情報を保管しない場合は、最初に .java ファイルを新しい Visual Editor for Java に開くと、エディターはトップレベル (フリー・フォーム・パーツとして知られている) Bean のデフォルト位置を計算します。そのトップレベル Bean はドラッグ・アンド・ドロップ操作で簡単に変更できます。
- WebSphere Application Server - Express の新規 Visual Editor for Java の初期バージョンは、接続設計をサポートしませんが、将来、これに関するいくつかの同等機能が追加される可能性があります (注: これは製品コミットメントではなく、単なる将来の可能性を述べているに過ぎません)。

マイグレーションの前に拡張メタデータ情報を保管する方法:

1. VisualAge for Java 開発者ドメイン
[www.software.ibm.com/vad.com/vad/data/document4293] に移動して、*IBM VCE Code Generation and Export Utility* をダウンロードします。
2. ツールの README に従って、ツールを VisualAge for Java に追加し、次に VisualAge for Java を再始動する。
3. 現在のアプリケーション・コードを VisualAge for Java リポジトリへ入れてバージョン管理する (したがって、VisualAge for Java 開発を進めている場合に、このバージョンに戻すことができます)。
4. それぞれの VisualAge for Java 内の図形アプリケーションごとに、図形プログラム (通常、XxxxxView) を選択して、右マウス・ボタンでクリックし、以下のステップを実行する。

- a. 「**VCE コード生成 / エクスポート (VCE Code Generation/Export)**」をクリックして、「**コード生成後にディレクトリへエクスポートする (Export to a directory after code regeneration)**」オプションを選択状態のままにする。
- b. 「**終了 (Finish)**」をクリックする。
- c. 「**ディレクトリ (Directory)**」をエクスポート宛先として選択状態にして、「**次へ (Next)**」をクリックする。
- d. ターゲット・ディレクトリを選択して、**.class** オプションを選択解除し、さらに (ソース・コードが必要なため) **.java** オプションを選択して、次に「**終了 (Finish)**」をクリックする。
- e. オプションとして、前のバージョンの VisualAge for Java コードを再ロードする (右マウス・ボタンをクリックして、「**アプリケーションを指定して置き換え (Replace with)**」 > 「**前のエディション (Previous edition)**」を選択する)。

マイグレーションの完了 (WebSphere Studio ヘインポート)

これで、クラスを WebSphere Application Server - Express にインポートできる状態になりました。前出の45 ページの『第 7 章 VisualAge for Java から IBM WebSphere Studio Site Developer へのマイグレーション』を参照してください。従来の Visual Composition Editor ソース・プログラムを WebSphere Application Server - Express にインポートしておけば、それらを Visual Editor for Java で保守することができます。

第 9 章 ビルド・セットアップ (ライブラリー、JAR、従属プロジェクト JAR、Ant ビルド)

この章では、以下のマイグレーション・トピックを取り上げます。

- Java ライブラリー JAR およびサード・パーティー外部 JAR
- 従属プロジェクト JAR を使用したマルチプロジェクト・ビルドの最適化
- Ant を使用した自動化実動ビルド

Java ライブラリー JAR およびサード・パーティー外部 JAR

関連事項の詳細については、*J2EE* クラス・ロードに関する資料

(www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer.html) (J2EE モジュールとクラスパス)、および J2EE ユーティリティ JAR の開発に関する資料 (www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer2.html) (J2EE モジュールの Java JAR) を参照してください。これらの資料は、詳細な技術的なバックグラウンドとアドバイスを提供します。

Web プロジェクトでサード・パーティー JAR を使用するための推奨される方法

サード・パーティーの JAR ファイルを Web プロジェクト内で使用する場合は、そのファイルを (JAR ファイルのまま) Web プロジェクトのライブラリー・フォルダーにインポートすることをお勧めします。これは、JAR ファイルを手軽に使用する場合の、J2EE で定義された移植可能な唯一の方法であり、別のサーバーにデプロイする場合に変更を加える必要がありません。

単一の Web プロジェクトで外部 JAR ファイルを使用するには、以下のステップを実行します。JAR ファイルを複数の Web プロジェクトで使用する必要がある場合は、代わりに、56 ページの『複数の Web プロジェクトでサード・パーティー JAR を使用するための推奨される方法』で説明しているステップを実行してください。

1. 「ファイル (File)」 > 「インポート (Import)」 > 「ファイル・システム (File System)」を選択し、「次へ (Next)」をクリックする。JAR ファイルがインポート時に展開されないように、「Zip ファイル (Zip file)」ではなく「ファイル・システム (File system)」を選択してください。
2. 「参照 (Browse)」をクリックして、JAR ファイル・ディレクトリーを探す。
3. そのディレクトリーを「WebProject/WebContent/WEB-INF/lib」フォルダーにインポートする。ここで、WebProject は Web プロジェクトの名前です。
4. 「終了 (Finish)」をクリックする。JAR ファイルは、Java ビルド・パスに追加されます。ランタイム時に必要な変更は他にはありません。

複数の Web プロジェクトでサード・パーティー JAR を使用する ための推奨される方法

サード・パーティーの JAR を複数の Web プロジェクトで使用する場合は、そのファイルを (JAR ファイルのまま) エンタープライズ・アプリケーション (EAR) プロジェクトにインポートすることをお勧めします。これは、JAR ファイルを手軽に使用する場合の、J2EE で定義された移植可能な唯一の方法であり、別のサーバーにデプロイする場合に変更を加える必要がありません。

外部 JAR ファイルを複数の Web プロジェクトで使用するには、以下のステップを実行します。JAR ファイルを単一の Web プロジェクトでしか使用する必要がない場合は、前のセクションのステップを実行します。

1. 「**ファイル (File)**」 > 「**インポート (Import)**」 > 「**ファイル・システム (File System)**」を選択し、「**次へ (Next)**」をクリックする。JAR ファイルがインポート時に展開されないように、「**Zip ファイル (Zip file)**」ではなく「**ファイル・システム (File system)**」を選択してください。
2. 「**参照 (Browse)**」をクリックして、JAR ファイル・ディレクトリーを探す。
3. JAR ファイルを、Web プロジェクトが入っているエンタープライズ・アプリケーション・プロジェクトにインポートする。
4. 「**終了 (Finish)**」をクリックする。JAR ファイルは、Java ビルド・パスに自動的に追加されます。ランタイム時に必要な変更は他にはありません。
5. 次のセクションのステップを実行して、JAR を Web プロジェクトのモジュール依存関係に追加する。

外部 JAR ファイルを使用するための代替方法 (グローバル・ビルドおよびサーバー・クラスパス)

JAR ファイルを WebSphere Application Server - Express の外部に残したまま、Java ビルド・パスとサーバー・インスタンスのクラスパスの両方に追加することもできます。この方法を取ると、アプリケーションを簡単に移植できなくなるため、お勧めしません。別のサーバーに移動した場合、そのサーバーのクラスパスを必ず更新しなければなりません。また、クラス・ファイルが、すでにそのサーバーのクラスパスに入っている (かつ、サーバーまたは他のアプリケーションに必要な) 他のバージョンの同様なクラス・ファイルと矛盾していないことも確認しなければなりません。このアプローチをとる場合は、以下のステップを実行します。

1. 外部 JAR ファイルを、JAR ファイルを必要とするプロジェクトの Java ビルド・クラスパスに追加する。
 - a. プロジェクトを選択して右マウス・ボタンでクリックし、ポップアップ・メニューから「**プロパティ (Properties)**」を選択する。
 - b. 「**Java ビルド・パス (Java Build Path)**」をクリックする。
 - c. 「**ライブラリー (Libraries)**」タブをクリックする。
 - d. 「**外部 JAR の追加 (Add External JARs)**」をクリックする。JAR ファイルを選択して、「**開く (Open)**」をクリックする。
 - e. 「**OK**」をクリックする。
2. 外部 JAR ファイルをサーバー・インスタンスのクラスパスに追加する。

- a. 「サーバー構成 (Server Configuration)」ビューを開き、「サーバー (Server)」フォルダーを展開する。
- b. プロジェクトがデプロイされているサーバー・インスタンスを選択する。右マウス・ボタンでクリックして「開く (Open)」をクリックする。
- c. 「パス (Paths)」タブをクリックする。
- d. ws.ext.dirs 内の「外部 JAR の追加 (Add External JARs)」をクリックする。JAR ファイルを選択して、「開く (Open)」をクリックする。
ws.ext.dirs はアプリケーション JAR ファイルに使用され、クラスパス (CLASSPATH) はサーバー JAR ファイルに使用されることに留意してください。
- e. サーバー・インスタンスを閉じ、変更を保管する。

従属プロジェクト JAR を使用したマルチプロジェクト・ビルドの最適化

WebSphere Application Server - Express の強力な自動ビルド・フィーチャーは、複雑なマルチプロジェクト・ビルドでは、ビルド・パフォーマンスを低下させる可能性があります。これらの自動ビルドを制御するにはいくつかの方法がありますが (例えば、従属ファイル、アクティブおよび非アクティブ・プロジェクト、ソースまたは JAR フォーマットのプロジェクトなど)、これらのオプションは非常に複雑なものになる可能性があります。これらのオプションとビルド・パフォーマンスの最適化について説明した資料があります。WebSphere Developer Domain の記事『Optimizing Multi-Project Builds in WebSphere Studio Application Developer』(www.ibm.com/websphere/developer/library/techarticles/0204_searle/searle.html) を参照してください。

Ant を使用した自動化実動ビルド

Ant を WebSphere Application Server - Express で使用して実動ビルドを自動化することができます。以下のことを説明した複数のセクションからなる資料があります。

- Ant とは
- Ant を WebSphere Application Server - Express の内部と外部の両方で実行する方法
- Ant を J2EE エlement (EJB、WAR、EAR、など) の実動ビルドで使用方法
- Ant を新規の WebSphere Application Server - Express ビルド・タスクで拡張する方法

WebSphere Developer Domain の記事『Using Ant with WebSphere Studio Application Developer』(www.ibm.com/websphere/developer/library/techarticles/0203_searle/searle1.html) を参照してください。

第 10 章 マイグレーションの例

この章では、VisualAge for Java および WebSphere Studio "Classic" から WebSphere Application Server - Express IBM WebSphere Studio Site Developer へのマイグレーションの学習に役立つマイグレーションの例を紹介します。

- VisualAge for Java JSP/Servlet サンプル (LeapYear)
- WebSphere Studio "Classic" Web アプリケーション・サンプル (YourCo)

VisualAge for Java JSP/Servlet サンプル (LeapYear)

説明

これは、VisualAge for Java バージョン 4.0 に付属の FindTheLeapYears というサンプルです。詳細については、VisualAge for Java のオンライン・ヘルプ (「**サンプル (Samples)**」 > 「**JSP/Servlet 開発環境 (JSP/Servlet Development Environment)**」) を参照してください。

マイグレーションの概説

以下の手順に従って、VisualAge for Java から IBM WebSphere Studio Site Developer にサンプルをマイグレーションします。各ステップについては以降で詳しく説明します。

1. VisualAge for Java からの Java ファイルおよびプロジェクト・リソース・ファイルをエクスポートする。
2. 新しい IBM WebSphere Studio Site Developer Web プロジェクトを作成する。
3. Java ファイルとプロジェクト・リソース・ファイルを IBM WebSphere Studio Site Developer プロジェクトにインポートする。
4. サーブレットを定義し、再構築したアプリケーションを変更する。
5. IBM WebSphere Studio Site Developer サーバー・プロジェクトを作成する。
6. マイグレーション済みアプリケーションをテストする。

VisualAge for Java からのファイルのエクスポート

1. VisualAge for Java を開く。
2. 「**IBM JSP Examples**」プロジェクトを選択する。
3. プロジェクトを右マウス・ボタンでクリックして、「**エクスポート (Export)**」を選択する。「**ディレクトリー (Directory)**」ラジオ・ボタンを選択して、「**次へ (Next)**」をクリックする。
4. ファイルのエクスポート先のディレクトリー名を入力する。
5. 「**.class**」チェック・ボックスを選択解除する。これらのファイルは、WebSphere Application Server - Express でプロジェクトを再ビルドしてから作成し直すので、エクスポートする必要はありません。
6. 「**.java**」チェック・ボックスを選択して「**詳細 (Details)**」をクリックし、「LeapYear」ファイルのみを選択して「**OK**」をクリックする。

7. 「**resource**」チェック・ボックスを選択して「**詳細 (Details)**」をクリックする。
8. 「**LeapYearInput.html**」および「**LeapYearResults.jsp**」を選択する。これらは、ディレクトリー **IBM WebSphere Test Environment\hosts\default_host\default_app\web\JSP\sample3** に入っています。
9. 「**OK**」をクリックする。
10. 「**マニフェストの作成 (Create Manifest)**」チェック・ボックスを選択解除する (マニフェスト・ファイルを作成する必要はありません)。
11. 「**終了 (Finish)**」をクリックする。
12. VisualAge for Java を閉じる。

新規 IBM WebSphere Studio Site Developer Web プロジェクトの作成

1. IBM WebSphere Studio Site Developer を始動する。
2. LeapYear という名前の新しいプロジェクトを作成する (「**ファイル (File)**」 > 「**新規作成 (New)**」 > 「**プロジェクト (Project)**」 > 「**Web**」 > 「**WebProject**」)。
3. 「**Dynamic プロジェクト (J2EE Web project)**」が選択状態であることを確認して、「**次へ (Next)**」をクリックする。
4. 「**新規作成 (New)**」を選択する。
5. 「**エンタープライズ・アプリケーション・プロジェクト名 (Enterprise Application project name)**」を LeapYearEAR に変更して、「**J2EE レベル 1.2 (J2EE level 1.2)**」をクリックする。Web プロジェクトは、既存のエンタープライズ・アプリケーション (EAR) プロジェクトに入れることができますが、この例では、LeapYearEAR に入れます。
6. 「**コンテキスト・ルート (Context root)**」フィールドに LeapYear と入力する。
7. 「**終了 (Finish)**」をクリックする。

Java ファイルとプロジェクト・リソース・ファイルの IBM WebSphere Studio Site Developer プロジェクトへのインポート

以下のステップを実行して、Java ソース・ファイルを LeapYear プロジェクト・ソース・ディレクトリーにインポートします。

1. 「**Web**」パースペクティブで LeapYear を展開し、「**Javasource**」ディレクトリーを選択する。
2. 「**ファイル (File)**」 > 「**インポート (Import)**」 > 「**ファイル・システム (File system)**」を選択して、「**次へ (Next)**」をクリックする。ファイルをエクスポートしたディレクトリーをブラウズして「**OK**」をクリックする。

3. Java ソース・ファイルのみを「**Javasource**」ディレクトリーにインポートしたいので、「インポート (Import)」ダイアログで、「export」ディレクトリーを展開し、「**com**」サブディレクトリー (Java ソース・ファイルが 3 つ入っているサブディレクトリー) のみを選択する。
4. 「終了 (**Finish**)」をクリックする。これで
LeapYear¥JavaSource¥com¥ibm¥ivj¥wte¥samples¥leapyear¥LeapYearXXXX.java ファイルが作成されます。Java クラスは LeapYear/webContent/WEB-INF/classes に自動的にコンパイルされます。

以下のステップを実行して、リソース・ファイルを「**WebContent**」ディレクトリー下の「LeapYear」プロジェクトにインポートする。

1. 現行の「Web」パースペクティブで、「LeapYear」プロジェクトを展開し、「**WebContent**」ディレクトリーを選択する。
2. 「**ファイル (File)**」>「**インポート (Import)**」>「**ファイル・システム (File system)**」を選択して、「**次へ (Next)**」をクリックする。ファイルをエクスポートしたディレクトリーをブラウズし、エクスポート・ディレクトリーを「**sample3**」サブディレクトリーに展開して、「**OK**」をクリックする。
3. リソース・ファイルだけを「**WebContent**」ディレクトリーにインポートするために、「インポート (Import)」ダイアログで、.jsp ファイルと .html ファイルが入っている「**sample3**」サブディレクトリーを選択する。
4. 「**終了 (Finish)**」をクリックする。ファイルが「**WebContent**」ディレクトリーにインポートされます。

サーブレットの定義と再構築したアプリケーションの変更

1. 次に、サーブレットを作成する。「LeapYear」プロジェクトを選択して、それを web.xml ファイルまで展開する (「**Leap Year**」>「**WebContent**」>「**WEB-INF**」)。web.xml ファイルを開きます。
2. ページの下部にある「**サーブレット (Servlets)**」タブをクリックする。
3. 「**追加 (Add)**」をクリックする。
4. 「**サーブレット (Servlet)**」ラジオ・ボタンが選択されていることを確認する。
5. クラス「**LeapYear**」を選択して、「**OK**」をクリックする。
6. 「**URL マッピング (URL Mapping)**」>「**追加 (Add)**」を選択して、LeapYear と入力する。
7. 変更を保管して (「**ファイル (File)**」>「**web.xml の保管 (Save web.xml)**」)、web.xml ファイルを閉じる。

ソース/アプリケーション構造が若干変更されているので、アプリケーションに変更を加える必要があります。

1. 2 つのエラーが「**タスク (Tasks)**」ビューに表示されます。1 つは LeapYearInput.html 内のエラー、もう 1 つは LeapYearResults.jsp 内のエラーです。
2. LeapYearResults.jsp ファイルを開く。/JSP/index.html を LeapYearInput.html に置き換える。
3. LeapYearInput.html ファイルを開く。
/servlet/com.ibm.ivj.wte.samples.leapyear.LeapYear を LeapYear に置き換える。

4. 変更を保管して、LeapYearResults.jsp ファイルと LeapYearInput.html ファイルを閉じる。
5. ランタイム・エラーが発生しないようにするために、
JavaSource¥com¥ibm¥ivj¥wte¥samples¥leapyear サブディレクトリーにある
LeapYear.java ファイルを開く。
6. 118 行目に移動し、getRequestDispatcher を “/JSP/Sample3/LeapYearResults.jsp” から “LeapYearResults.jsp” に変更する。
7. 変更を保管して、LeapYear.java を閉じる。

これで、サンプルは IBM WebSphere Studio Site Developer にマイグレーションされました。あとは、IBM WebSphere Studio Site Developer サーバー・プロジェクトを作成して、サンプルを WebSphere テスト環境でテストするだけです。

IBM WebSphere Studio Site Developer サーバー・プロジェクトの作成

1. 「ファイル (File)」 > 「新規作成 (New)」 > 「プロジェクト (Project)」 > 「サーバー (Server)」 > 「サーバー・プロジェクト (Server Project)」 をクリックする。「次へ (Next)」 をクリックする。「プロジェクト名 (Project name)」フィールドに newServer と入力して、「終了 (Finish)」 をクリックする。自動的に「サーバー (Server)」 パースペクティブに切り替わります。
2. 「newServer」 を右マウス・ボタンでクリックして「新規 (New)」 > 「サーバーとサーバー構成 (Server and Server Configuration)」 をクリックする。
3. 「サーバー名 (Server name)」 フィールドに WSTestEnv と入力し、「サーバー・インスタンス型 (Server instance type)」 フィールドで「WebSphere v4.0 テスト環境 (WebSphere V4.0 Test Environment)」 を選択して、「終了 (Finish)」 をクリックする。

次に、サーバー構成に対して EAR プロジェクトを指定する必要があります。

1. 「サーバー構成 (Server Configuration)」 ビューで、サーバー項目を展開して、「WSTestEnv」 をクリックする。
2. それを右マウス・ボタンでクリックして、「追加 (Add)」 > 「LeapYearEAR」 をクリックする。

マイグレーション済みの LeapYear アプリケーションのテスト

1. LeapYearInput.html ファイルを選択する。
2. HTML ファイルを右マウス・ボタンでクリックし、ポップアップ・メニューから、「サーバーで実行 (Run on Server)」 をクリックする。
3. サーバーが始動するまで待機する。「コンソール (Console)」 ページ (「サーバー (Servers)」 ビューの「コンソール (Console)」 タブをクリックする) の監視を、「e-business 用にデフォルト・サーバーが開いています」というメッセージが表示されるまで続けます。
4. ブラウザーが開いたら、「開始年 (Start Year)」 フィールドに 2001 と入力し、「実行依頼 (Submit)」 をクリックする。

5. 「コンソール (Console)」ビューに、LeapYear:init というメッセージが表示される。うるう年のリストが表示されるまで待機し、「サーバー (Servers)」ビューで「WSTestEnv」を選択して、右マウス・ボタンをクリックし、「停止 (Stop)」をクリックする。

例: WebSphere Studio "Classic" バージョン 4.0 Web アプリケーション (YourCo)(Windows)

説明

この例では、WebSphere Studio "Classic" バージョン 4.0.x を使用しなければなりません。

これから使用するサンプルは、**YourCo** サンプルです。このサンプルにアクセスするには、オンライン・ヘルプを開きます (「ヘルプ (Help)」 > 「WebSphere Studio 4.0」 > 「方法 (How to)」 > 「サンプルの使用 (Work with the samples)」 > 「概要 (Overview)」)。このサンプルをロードするには、「Studio サンプルを開く (Opening the Studio Samples)」 (WebSphere Application Server 4.0 用) に記載されている指示に従って、YourCo.war をロードします。

注: マイグレーションしたアプリケーションは IBM WebSphere Studio Site Developer で実行しますが、IBM WebSphere Studio Site Developer には、現時点で WebSphere Studio プロフェッショナル版またはアドバンスド版のすべての Web ページ設計機能および開発機能が用意されているわけではありません。

始める前に

- YourCo サンプル・アプリケーションが WebSphere Studio "Classic" にロードされていることを確認する。
- WebSphere Application Server のすべてのインスタンスを定義する (IBM WebSphere Studio Site Developer との競合を防ぐため)。

マイグレーションのステップ

WebSphere Studio "Classic" から IBM WebSphere Studio Site Developer にサンプルをマイグレーションするには、以下のステップを実行してください。各ステップについては以降で詳しく説明します。

1. (オプション) WebSphere Studio "Classic" を始動し、新規マイグレーション・ステージを作成する。
2. Web 構成記述子ファイルを作成する。
3. マイグレーション WAR ファイル (WebPath を含む) を作成する。
4. IBM WebSphere Studio Site Developer を始動し、WAR ファイルをインポートする。
5. IBM WebSphere Studio Site Developer サーバー・プロジェクトを作成する。
6. マイグレーション済みアプリケーションをテストする。

WebSphere Studio "Classic" バージョン 4.0 の始動および新規マイグレーション・ステージの作成

(オプション) 通常は、マイグレーション用の新しいステージを作成しますが、この例では、WebSphere Studio "Classic" に用意されているテスト・ステージを使用します。テスト・ステージを使用すれば、ステップ 8 で多くのサブレット・マッピングを手動で編集しなければならない手間を省くことができます。

マイグレーション用の新しいステージの作成方法については、『WebSphere Studio "Classic" から IBM WebSphere Studio Site Developer へのマイグレーション』を参照してください。

Web 構成記述子ファイルの作成

1. 「プロジェクト・ファイル (project file)」ビューで、「プロジェクト (Project)」>「Web 構成記述子ファイルの作成 (Create Web Configuration Descriptor File)」をクリックして、デフォルト値の WEB-INF\localhost_web.xml を受け入れる。
2. 必要なすべてのサブレット (xxxxBean という名前以外のすべてのファイル) を選択する。
3. このサンプルに、タグ・ライブラリー記述子 (TLD) ファイルはありません。
4. 「作成 (Create)」をクリックする。

マイグレーション・ファイルの作成

1. 「プロジェクト・ファイル (project file)」ビューで、サーバー localhost を選択し、「プロパティ (Properties)」>「公開 (Publishing)」>「WebApp Web パス (WebApp Web Path)」をクリックして、Web パス (コンテキスト・ルート) "newStudioSample" を入力する (Web パスの設定は、最終的な IBM WebSphere Studio Site Developer 製品では推奨アプローチとして残される予定です)。
2. 「プロジェクト・ファイル (project file)」ビューで、「プロジェクト (Project)」>「マイグレーション・ファイルの作成 (Create Migration file)」を選択する。
3. サーバーとして localhost が選択されていることを確認する。
4. Web 構成記述子ファイルとして localhost_web.xml が選択されていることを確認する。
5. 「OK」をクリックする。
6. デフォルトの JAR ファイル名は X:\Studio\projects\YourCo\localhost.jar (ここで、X は WebSphere Studio "Classic" インストール・ディレクトリ)。
7. 「保管 (Save)」をクリックする。
8. WebSphere Studio "Classic" を閉じる。
9. localhost.jar file を localhost.war に名前変更する。

IBM WebSphere Studio Site Developer の始動と WAR ファイルのインポート

1. IBM WebSphere Studio Site Developer を始動する。

2. 「ファイル (File)」 > 「インポート (Import)」 > 「WAR ファイル (WAR file)」 > 「次へ (Next)」をクリックする。

注: JAR ファイルは WAR ファイル・オプションを使用してインポートしなければ、正しく動作しません。

3. 「WAR ファイル (WAR File)」フィールドに localhost.war のパスを入力するか、「参照 (Browse)」をクリックして検索する。
4. 「Web プロジェクト (Web Project)」フィールドで「新規作成 (New)」を選択して、newStudioSample と入力する。
5. 「EAR プロジェクト名 (EAR project name)」フィールドで「新規作成 (New)」を選択して、newStudioSampleEAR と入力する。
6. 「終了 (Finish)」をクリックする。IBM WebSphere Studio Site Developer が localhost.war を解凍する。
7. 多数の未解決の参照やインポート・ファイルの欠落が「タスク (Task)」ビューに表示される。
 - a. com.ibm.db requires databeans.jar,
 - b. com.ibm.webtools.runtime requires webtlsrn.jar,
 - c. com.ibm.ejs.ns.jndi requires ns.jar,
 - d. com.ibm.webshpere.advanced.cm.factory requires cm.jar,
 - e. com.ibm.ejs.models.base.extensions.webappext.ServletExtensions requires ws-base-extensions.jar

これを修正するには、Web プロジェクトの Java ビルド・パスを変更しなければなりません。

- a. プロジェクトを右マウス・ボタンでクリックして、「プロパティ (Properties)」 > 「Java ビルド・パス (Java Build Path)」をクリックする。
 - b. 「ライブラリー (Libraries)」タブをクリックする。「外部 JAR の追加 (Add External JARs)」をクリックする。
 - c. ディレクトリ MyInstall¥runtimes¥aes_v4¥lib から、**databeans.jar**, **webtlsrn.jar**, **ns.jar**, **cm.jar**、および **ws-base-extensions.jar** の各 JAR ファイルをインポートする。
 - d. 未解決の警告が 24 あるが、処理は不要。
8. 「newStudioSample」プロジェクトを右マウス・ボタンでクリックして、「プロジェクトの再ビルド (Rebuild Project)」をクリックする。

これで、サンプルは IBM WebSphere Studio Site Developer にマイグレーションされました。あとは、IBM WebSphere Studio Site Developer サーバー・プロジェクトを作成して、サンプルを WebSphere テスト環境でテストするだけです。

IBM WebSphere Studio Site Developer サーバー・プロジェクトの作成

1. 「サーバー (Server)」パースペクティブに切り替える。
2. 「ファイル (File)」 > 「新規作成 (New)」 > 「プロジェクト (Project)」 > 「サーバー (Server)」 > 「サーバー・プロジェクト (Server Project)」をクリックする。「次へ (Next)」をクリックする。「プロジェクト名 (Project name)」フィールドに newServer と入力して、「終了 (Finish)」をクリックする。

3. 「ナビゲーター (Navigator)」ビューで、**newServer** を右マウス・ボタンでクリックし、「新規 (New)」>「サーバーとサーバー構成 (Server and Server Configuration)」をクリックする。
4. 「サーバー名 (Server name)」フィールドに **WSTestEnv** と入力し、「サーバー・インスタンス型 (Server instance type)」フィールドで「**WebSphere V4.0**」>「**テスト環境 (Test Environment)**」を選択する。「**終了 (Finish)**」をクリックする。

次に、サーバー構成に対して EAR プロジェクトを指定する必要があります。

1. 「サーバー構成 (Server Configuration)」ビューで、「**サーバー (Server)**」>「**WSTestEnv**」をクリックする。
2. それを右マウス・ボタンでクリックして、「**追加 (Add)**」>「**newStudioSampleEAR**」をクリックする。

注: (オプション)「newStudioSample」プロジェクトを右マウス・ボタンでクリックして「**プロパティ (Properties)**」>「**サーバー設定 (Server Preference)**」>「**次のサーバーで常に実行します: (Always run on the following server)**」を選択し、「**WSTestEnv**」を選択し、「**適用 (Apply)**」>「**OK**」をクリックする (この手順は、他にサーバーがある場合のみ必要です)。

マイグレーション済みの YourCo アプリケーションのテスト

1. 「**YourColntro.html**」ファイルを選択する。このファイルは、newStudioSample プロジェクト内のディレクトリ `WebContent\StudioSamples` に入っています。
2. **YourColntro.html** を右マウス・ボタンでクリックし、ポップアップ・メニューから、「**サーバーで実行 (Run on Server)**」をクリックして、「**WSTestEnv**」を選択する。
3. サーバーが始動するまで待機する。「**コンソール (Console)**」ページ (「サーバー (Servers)」ビューの「**コンソール (Console)**」タブをクリックする) の監視を、「e-business 用にデフォルト・サーバーが開いています」というメッセージが表示されるまで続けます。
4. このサンプルをまだ WebSphere Studio "Classic" で実行したことがない場合は、「**データベース構成 (Database Configuration)**」をクリックしてデータベースを構成する必要がある。
5. ブラウザーが開いたら、スクロールダウンして、「**このサンプルを実行 (Run This Sample)**」をクリックする。
6. ブラウザーに「**ウェルカム (Welcome)**」ページが表示されるまで待機し、「**Employee Center**」をクリックする。

注: このアプリケーションを初めて実行したときには、「コンソール (Console)」ページに次のエラーが表示されます。DataSource not found. Try to construct a new datasource name: jdbc/yourco DataSource not found. Try to construct a new datasource name: jdbc/studio. これらのエラーは自己解決型です。これらの警告は無視してかまいません。

7. 終了したら、「ブラウザー」ウィンドウと「Web Browser」ビューを閉じ、「**サーバー・コントロール・パネル (Server Control Panel)**」で、「**WSTestEnv**」を右マウス・ボタンでクリックして「**停止 (Stop)**」をクリックする。

8. (オプション) IBM WebSphere Studio Site Developer を閉じる。

第 11 章 その他の資料

マイグレーションおよびその他のトピックに関する最新の情報が、
www.ibm.com/websphere/developer/zones/studio/transition.html から入手できます。

WebSphere Application Server - Express を利用するとき役に立つ一般情報については、以下の資料および Web ページを参照してください。

- *JSR-000053 Java Servlet 2.3 および JavaServer Page 1.2 の仕様:*
java.sun.com/aboutJava/communityprocess/first/jsr053/index.html
- IBM WebSphere Application Server バージョン 4 InfoCenter:
www.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/index.html
- *WebSphere Version 4 Application Development Handbook:*
www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246134.pdf
- *Programming J2EE APIs with WebSphere Advanced:*
www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246124.pdf
- *WebSphere Application Server Version 3.5 to 4.x - Migration Hints & Tips:*
www7b.software.ibm.com/wsdd/library/techarticles/0208_wright/wright.html
- *WebSphere Studio Application Developer Service Portal:*
www.ibm.com/software/ad/studioappdev/support/
- *WebSphere Studio Application Developer FAQ Frequently Asked Questions:*
www.ibm.com/support/search.wss?rs=457&tc=SSBRLP&dc=D800
- *WebSphere Application Server Service Portal:*
www.ibm.com/software/software/webservers/appserv/support.html
- *WebSphere Application Server FAQ Frequently Asked Questions:*
www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&dc=D800

その他、次のような資料も参考になります。

- WebSphere Studio Application Developer での Ant の使用に関する資料 (J2EE プロジェクト・ビルド/エクスポートを含む):
www.ibm.com/websphere/developer/library/techarticles/0203_searle/searle1.html
- WebSphere Studio Application Developer における複雑なビルドの最適化に関する資料:
www.ibm.com/websphere/developer/library/techarticles/0204_searle/searle.html
- WebSphere Studio Application Developer における J2EE クラス・ロードに関する資料 (J2EE モジュールおよびクラスパス):
www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer.html
- WebSphere Studio Application Developer における J2EE ユーティリティー JAR の開発に関する資料 (J2EE モジュールにおける Java JAR):
www.ibm.com/websphere/developer/library/techarticles/0112_deboer/deboer2.html
- WebSphere Studio Application Developer におけるチーム・サポートに関する資料:
www.ibm.com/websphere/developer/library/techarticles/0108_karasiuk/0108_karasiuk.html

- VisualAge for Java から WebSphere Studio Application Developer への Enterprise Access Builder コンポーネントのマイグレーションに関する資料:
www.ibm.com/websphere/developer/techjournal/0201_minocha/minocha.html
- Session Facade を使用した CMP との対話の EJB アプリケーション設計に関する資料:
www.ibm.com/websphere/developer/library/techarticles/0106_brown/sessionfacades.html
- WebSphere Application Server 最良実例に関する資料:
www.ibm.com/software/webservers/appserv/ws_bestpractices.pdf
- WebSphere 最良実例ゾーンに関する資料:
www.ibm.com/websphere/developer/zones/bp/
- WebSphere Developer Domain メイン Web ページ:
www.ibm.com/websphere/developer
- WebSphere Developer Domain のテクニカル記事:
www.ibm.com/websphere/developer/techjournal/
- WebSphere Studio ファミリー、および WebSphere Studio Application Developer のフィーチャーと利用方法に関する情報:
www.ibm.com/websphere/developer/zones/studio/transition.html
- IBM WebSphere Studio ファミリー開発ツール製品情報:
www.ibm.com/websphere/developer/library/techarticles/0108_studio/studio_beta.html
- 外部 Application Developer ニュースグループ:
[news://news.software.ibm.com/ibm.software.websphere.studio.application-site-developer](http://news.software.ibm.com/ibm.software.websphere.studio.application-site-developer)
- 外部ワークベンチ (Eclipse) ニュースグループ:
[news://news.software.ibm.com/ibm.software.websphere.studio.workbench](http://news.software.ibm.com/ibm.software.websphere.studio.workbench)
- 外部 WebSphere Application Server ニュースグループ:
[news://news.software.ibm.com/ibm.software.websphere.application-server](http://news.software.ibm.com/ibm.software.websphere.application-server)
- WebSphere Studio Application Developer から WebSphere Application Server への J2EE アプリケーションのデプロイに関する資料:
www.software.ibm.com/vad.nsf/Data/Document3584
- Application Developer Software 構成管理 (ソース・コード管理) ベンダー:
www.ibm.com/software/ad/studioappdev/partners/scm.html
- アプリケーションの同業他社開発ツールから Application Developer へのマイグレーション:
www.ibm.com/websphere/developer/zones/studio/migration.html
- VisualCafé WebLogic アプリケーションの Application Developer へのマイグレーション (WebLogic へデプロイ):
www.ibm.com/websphere/developer/library/techarticles/0209_searle/searle1.html
- Eclipse.org: www.eclipse.org
- WebSphere Developer Domain プラグイン・セントラル:
www.ibm.com/websphere/developer/downloads/plugin/
- Eclipse ワークベンチ・プラグイン (Eclipse.org のパーツではない):
www.eclipse-workbench.com/jsp/plugins.jsp
- Eclipse ワークベンチ・プラグイン (Eclipse.org のパーツではない):
www.eclipse-plugins.2y.net/eclipse/plugins.jsp

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

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せん。

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せん。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せん。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書において 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。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

(C) (お客様の会社名) (年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 (C) Copyright IBM Corp. 2000, 2003. All rights reserved.
(C) Copyright IBM Japan 2003.

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

汎用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

以下は、IBM Corporation の米国およびその他の国における商標です。

- AIX
- CICS
- Cloudscape
- DB2
- DB2 Extenders
- DB2 Universal Database
- e-business
- IBM
- iSeries
- OS/390
- S/390
- VisualAge
- WebSphere
- z/OS

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Microsoft、Windows、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または登録商標です。

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などはそれぞれ各社の商標または登録商標です。