

PeopleSoft®

EnterpriseOne 8.9
Web サーバー・インストール
PeopleBook
(UNIX ベース・システムおよび WebSphere)

2003 年 9 月

PeopleSoft EnterpriseOne 8.9
Web サーバー・インストール (UNIX ベース・システムおよび WebSphere) PeopleBook
SKU AC89JWSIU0309

Copyright 2003 PeopleSoft, Inc. All rights reserved.

本書に含まれるすべての内容は、PeopleSoft, Inc. (以下、「ピープルソフト」) が財産権を有する機密情報です。すべての内容は著作権法により保護されており、該当するピープルソフトとの機密保持契約の対象となります。本書のいかなる部分も、ピープルソフトの書面による事前の許可なく複製、コピー、転載することを禁じます。これには電子媒体、画像、複写物、その他あらゆる記録手段を含みます。

本書の内容は予告なく変更される場合があります。ピープルソフトは本書の内容の正確性について責任を負いません。本書で見つかった誤りは書面にてピープルソフトまでお知らせください。

本書に記載されているソフトウェアは著作権によって保護されており、このソフトウェアの使用許諾契約書に基づいてのみ使用が許諾されます。この使用許諾契約書には、開示情報を含むソフトウェアと本書の使用条件が記載されていますのでよくお読みください。

PeopleSoft、PeopleTools、PS/nVision、PeopleCode、PeopleBooks、PeopleTalk、Vantiveはピープルソフトの登録商標です。Pure Internet Architecture、Intelligent Context Manager、The Real-Time Enterpriseはピープルソフトの商標です。その他すべての会社名および製品名は、それぞれの所有者の商標である場合があります。ここに含まれている内容は予告なく変更されることがあります。

オープンソースの開示

この製品には、Apache Software Foundation (<http://www.apache.org/>) が開発したソフトウェアが含まれています。Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved. このソフトウェアは「現状のまま」提供されるものとし、特定の目的に対する商品性および適格性の黙示保証を含む、いかなる明示または黙示の保証も行いません。Apache Software Foundationおよびその供給業者は、損害の発生原因を問わず、責任の根拠が契約、厳格責任、不法行為(過失および故意を含む)のいずれであっても、また損害の可能性が事前に知らされていたとしても、このソフトウェアの使用によって生じたいかなる直接的損害、間接的損害、付随的損害、特別損害、懲罰的損害、結果的損害に関しても一切責任を負いません。これらの損害には、商品またはサービスの代用調達、使用機会の喪失、データまたは利益の損失、事業の中断が含まれますがこれらに限らないものとします。

ピープルソフトは、いかなるオープンソースまたはシェアウェアのソフトウェアおよび文書の使用または頒布に関しても一切責任を負わず、これらのソフトウェアや文書の使用によって生じたいかなる損害についても保証しません。

目次

概要	1
はじめる前に	1
WebSphereコンポーネントの命名規則	2
新機能	3
ERP 9.0	3
サードパーティ・ソフトウェアのインストール	5
WebSphere要件の理解	5
サポートされているWebSphereのバージョン	5
WebSphereの入手	5
はじめる前に	6
必要なデータベース・リポジトリ	6
WebSphereリポジトリ用データベースの構成	6
DB2/UDBの構成	7
▶ DB2/UDBを構成するには	7
Oracle (Oracle 9i)の構成	8
▶ Oracleを構成するには	8
IBM HTTPサーバーとWebSphere (バージョン4.0) のインストール	9
▶ IBM HTTPサーバーとWebSphere (バージョン4.0) をインストールするには	9
WebSphere FixPackの適用	18
▶ WebSphere FixPackを適用するには	18
IBM HTTPサーバーの構成	18
WebSphereをサポートするためのIBM HTTPサーバーの構成	19
▶ WebSphereをサポートするようにIBM HTTPサーバーを構成するには	19
httpサービスの開始	19
▶ httpdサービスを開始 (稼働することを確認) するには	19
WebSphereの起動	20
▶ WebSphere Administrative Consoleを起動するには (AEの場合)	20
WebSphereアプリケーション・サーバーのテスト	21
▶ WebSphereアプリケーション・サーバーをテストするには (AEの場合)	21
Webサーバーのインストールと構成	23
はじめる前に	23
Webサーバーのインストール	23
▶ Webサーバーをインストールするには	23
OneWorld/ERPデータベース接続 (JDBCドライバ) の構成	34
DB2/UDBの構成	34
JDBCドライバの入手	35
OneWorld/ERP用のJDBCドライバの構成	35
db2profileの実行	35
▶ db2profileを実行するには	35
DB2/UDBからWebSphereへの接続	36
▶ DB2/UDBをWebSphereに接続するには	36
Oracleの構成	36
JDBCドライバの入手	36
OneWorld/ERP用のJDBCドライバの構成	36

tnsnames.oraファイルの構成	37
OneWorld/ERPインストール・プランへのWebサーバーの追加	37
▶ OneWorld/ERPインストール・プランにJavaサーバーを追加するには	37
JASをサポートするためのIBM HTTPサーバーの構成	39
▶ JASをサポートするようにIBM HTTPサーバーを構成するには	39
スタティック・ファイルの処理の構成	40
▶ ファイル・サーブレットを無効化するには	41
▶ ファイル・サーブレットを有効化するには	41
JASログの構成	41
サンプルjdelog.propertiesファイル	42
OneWorld/ERPのシリアル化オブジェクトの生成	45
はじめる前に	46
eGeneratorのインストール	46
はじめる前に	46
デフォルトの記憶域パラメータの設定	46
▶ デフォルトの記憶域パラメータを設定するには(Oracleのみ)	46
▶ デフォルトの記憶域パラメータを設定するには(SQLのみ)	47
シリアル化オブジェクト生成のためのWin32クライアントのセットアップ	47
▶ シリアル化オブジェクトを生成するようにOneWorld/ERPクライアントをセットアップするには	47
ジェネレート・マシンへのJASビルドのコピー	47
▶ ジェネレート・マシンにJASビルドをコピーするには	47
シリアル化オブジェクトのためのOneWorld/ERPの準備	48
▶ データ辞書を自動入力するには	48
▶ グローバル・スペックを自動入力するには	49
gen.batの修正	49
▶ gen.batを修正するには	49
ログイン	50
はじめる前に	51
jas.iniの設定	51
jde.iniの設定	51
Webサーバーに生成する方法	52
▶ Webサーバーに生成するには	52
Webサーバーをバイパスしてデータベースに直接生成する方法	53
▶ オブジェクトをデータベースに直接生成するには	54
eGeneratorの構成	55
eGeneratorオプションの設定	55
▶ eGeneratorオプションを設定するには	55
生成モード	57
逐次生成	57
パラレル生成	57
直接生成	58
HTMLOverridesファイルの選択	58
▶ HTMLOverridesファイルを選択するには	58
言語の選択	58
▶ 言語を選択するには	59
genapp.iniファイルの構成	59
すべての標準シリアル化オブジェクトの生成	61
TAM Analyzerの実行	62
▶ TAM Analyzerを実行するには	62
全オブジェクトの生成	63
▶ すべてのOneWorld/ERPオブジェクトを生成するには	63

生成プロセスの検証	66
ログ・ファイルのチェック	66
▶ ステータス・ログを表示するには	67
▶ エラー・ログを表示するには	67
データベース確認のチェック	67
選択したオブジェクトの生成	68
バルク生成ファイルを使用したオブジェクトの生成	69
バルク生成ファイルの作成	69
▶ バルク生成ファイルを作成するには	69
バルク生成ファイルの実行	70
▶ バルク生成ファイルを実行するには	70
コア・オブジェクトの生成	72
▶ コア・オブジェクトを生成するには	72
アプリケーションの生成	74
▶ 単一アプリケーションを生成するには	74
▶ システム・コード別にアプリケーションを生成するには	76
▶ モード別にアプリケーションを生成するには	77
フォームの生成	78
▶ フォームを生成するには	78
レポートの生成	80
▶ 単一レポートを生成するには	80
▶ レポート・バージョンを生成するには	80
▶ システム・コード別にレポートを生成するには	81
NERの生成	83
▶ NERを生成するには	83
データ辞書項目の生成	84
▶ データ辞書項目を生成するには	84
テーブルの生成	85
▶ テーブルを生成するには	85
ビジネス・ビューの生成	86
▶ ビジネス・ビューを生成するには	87
データ構造体の生成	88
▶ データ構造体を生成するには	88
グリッド・タブの変換	89
▶ グリッド・タブを変換するには	89
Webサーバーの実行	91
Webサーバーの起動	91
▶ Webサーバーをインストールするには (WebSphere AEの場合)	91
Webサーバーの停止	91
▶ Webサーバーを停止するには (WebSphere AEの場合)	91
Webサーバーへのアクセス	91
▶ Webサーバーにアクセスするには	92
J.D. EdwardsポータルでのInherited Trustシステムのセットアップ	92
▶ J.D. Edwardsポータル上でInherited Trustシステムをセットアップするには	93
トラブルシューティングのヒント	94
URLがAIX上のIBM HTTPサーバーで機能しない場合	94
Webサーバーのチューニング	97
jas.ini設定のチューニング	97
[JDENET]	97
[LOGS]	97
[PORTALCONFIGURATION]	97

Java仮想マシンのチューニング	98
単一VMのチューニング	98
複数VMが動作するシステムのチューニング	98
エンタープライズ・サーバーのチューニング	98
DB2®/UDBの場合のみ	98
データベース・マネージャ構成	98
データベース構成(OWSH9)	98
シリアル化オブジェクト・データベースのチューニング	99
DB2/UDBのチューニング	99
▶ DB2/UDBをチューニングするには	99
Oracle9iのチューニング	100
最適化設定の確認	100
▶ init.oraファイルのパラメータ設定をチェックするには	100
▶ データベースをクエリーするには	100
統計の更新	100
▶	101
A. jas.iniファイルのパラメータと値	
[SECURITY]	105
[OWWEB]	108
[PORTALCONFIGURATION]	112
[CACHE]	121
[SERVER COMPONENTS]	123
[JDENET]	123
[SERVER]	124
[LOGIN]	125
[LOGS]	126
[TRANSACTIONS]	127
[WEB GUI]	128
B. サンプルjas.iniファイル	
C. jdbj.iniファイルのパラメータと値	
J.D. Edwards OneWorld/ERPデータ固有のセクション	133
[JDBj-BOOTSTRAP SESSION]	133
[JDBj-BOOTSTRAP DATA SOURCE]	134
[JDBj-SPEC DATA SOURCE]	135
汎用セクション	136
[JDBj-JDBC DRIVERS]	137
[JDBj-ORACLE]	137
[JDBj-LOGS]	138
[JDBj-CONNECTION POOL]	138
[JDBj-RUNTIME PROPERTIES]	139
D. サンプルJDBj.iniファイル	
E. WebSphere 4.0への移行	
Javaサーバー構成のバックアップ作成	153
WebSphere 3.5構成のバックアップ作成	154
▶ WebSphere 3.5構成のバックアップを作成するには	154
IBM HTTPサーバー構成のバックアップ作成 (IHSの場合のみ)	154
▶ IBM HTTPサーバー構成のバックアップを作成するには	154
jas.iniファイルのバックアップ作成	155

▶ jas.iniファイルのバックアップを作成するには	155
データベース・ドライバのバックアップ作成	155
▶ データベース・ドライバのバックアップを作成するには	155
WebSphere 3.5のアンインストール	156
WebSphereからの Web Server構成の削除	156
▶ WebSphereからWeb Server構成を削除するには	156
WAS 3.5.xインストールの削除	157
▶ WAS 3.5.xインストールを削除するには	157
IBM HTTPサーバーのインストール削除	158
▶ IBM HTTPサーバーのインストールを削除するには	158
WebSphere 4.0とJASのインストールと構成	159
F. Webサーバーのサービス・パックのアップグレード	
Webサーバー構成のバックアップ作成	161
WebSphere 4.0構成のバックアップ作成	161
▶ WebSphere 4.0構成のバックアップを作成するには	161
jas.iniファイルのバックアップ作成	162
▶ jas.iniファイルのバックアップを作成するには	162
ポータル・コンポーネントのバックアップ作成	162
WebSphereからのWebサーバー構成の削除	162
▶ WebSphereからWebサーバー構成を削除するには	162
キャッシュ・ディレクトリの削除	163
▶ キャッシュ・ディレクトリを削除するには	163
新規サーバーのサービス・パックのインストール	164
▶ 新規Webサーバーのサービス・パックをインストールするには	164
G. IBM HTTPサーバーの構成	
▶ IBM HTTPサーバーを構成するには	165
H. WebSphereリポジトリとしてのiSeries DB2の使用	
iSeries上でのWASデータベースの作成	177
▶ iSeries上でWASデータベースを作成するには	177
WebSphereのインストール	178
▶ WebSphereをインストールするには	178
WebSphereの構成	179
▶ WebSphereを構成するには	179
例: Initial.Setup.Config	181
例:	182
I. Webサーバー上のメディア・オブジェクトの理解	
必要なjas.ini設定	185
Webサーバーでのメディア・オブジェクトの表示	185

概要

このガイドでは、UNIXプラットフォーム上でOneWorld XeまたはERP 8以上（以下、OneWorld/ERP）を実行するJ.D. Edwards HTML Webサーバー（以下、Webサーバー）のインストールと構成について説明します。インストール・プロセスの構成は次のとおりです。

- ・ サードパーティ・ソフトウェアのインストール
- ・ Webサーバーのインストールと構成
- ・ OneWorld/ERPのシリアル化オブジェクトの生成
- ・ Webサーバーの実行
- ・ Webサーバーのチューニング

同じJavaサーバーに複数のサービス・パックをインストールする方法については、Knowledge Gardenで次のドキュメントを参照してください。

https://knowledge.jdedwards.com/jdecontent/WWAT/TipsTechniques/ttdk_JAS_Multi_Foundations.doc

J.D. Edwardsポータルを使用している場合は、この製品の使用ガイドを次のURLのJ.D. Edwards Knowledge Gardenで検索できます。

<https://knowledge.jdedwards.com/JDEContent/documentationcbt/overview/B7333eindex.htm>
[https://knowledge.jdedwards.com/jdecontent/WWAT/TipsTechniques/ttdk_JAS_Multi_Foundations.doc]

注意

Webサーバーのインストールに必要な手順が記載されています。ただし、インストール担当者がUNIXとシステム管理の基本的な知識を持っていることが前提となります。

注意

OneWorld/ERP JavaサーバーとWebSphere 4.0では、SQL Server 7のサポートが終了しています。このガイドでSQL Server 7に言及している箇所は、すべて無視してください。最新情報については、「最低限の技術要件」を参照してください。

はじめる前に

- ・ ポータルを実装するには、SAR 5274158、5897371、および6107095の特殊指示を完了する必要があります。
- ・ システムが、特定のプラットフォーム上で動作するWebサーバーのインストールに必要なハードウェアとソフトウェアの技術的な最小要件(MTR)を満たしているかどうかを確認します。これらの要件のリストについては、次のURLにアクセスし、Knowledge GardenのWebページを参照してください。

<https://knowledge.jdedwards.com/JDEContent/TechMarketingCU/information/bjtechreqs.htm>

- ・ OneWorld/ERP™エンタープライズ・サーバーが正常にインストールされ、機能しているかどうかを確認します。
- ・ 次のリンクを使用して、インストール環境がWebSphereコンポーネントのインストール前提条件を満たしているかどうかを確認します。

<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/index.html>

WebSphereコンポーネントの命名規則

WebSphereコンポーネントの場合は、次の命名規則を使用することをお勧めします。これらの名称を使用すると、各種のWebサーバー・コンポーネントをOneWorld/ERPのさまざまなインスタンスと区別できます。次の表のすべてのコンポーネント一式により、OneWorld/ERPの単一のインスタンスが定義されます。

アプリケーション・サーバー	推奨名
アプリケーション・サーバー	AS_JDEdwards_[instance]
仮想ホスト	VH_JDEdwards_[instance]
Webアプリケーション	EA_JDEdwards_[instance]
標準出力	stdout_[instance].log
標準エラー	stderr_[instance].log

[instance]は、OneWorld/ERPのインスタンス定義であるWebSphereコンポーネント・グループを識別する固有の文字、番号または文字列です。[instance]は、あるグループの全コンポーネントについて同じにする必要があります。

たとえば、OneWorld/ERP™のインスタンスが1つしかない場合は、[instance]を“1”と定義できます。その場合、そのインスタンスのアプリケーション・サーバー名はAS_JDEdwards_1となります。

新機能

ここでは、各サービス・パック・リリースのマニュアルに加えられた変更事項について説明します。

ERP 9.0

SP22以降のリリースB9には、次の変更事項があります。

変更	セクション
WebSphere Advanced Edition Single Serverはサポート対象外となりました。	この変更は、このガイド全体に適用されます。
インストール・プログラムが修正されました。	Webサーバーのインストール Webサーバーのインストールと構成
JASにログインするには、新規構成ファイル <code>jdelog.properties</code> の設定が必要です。	「Webサーバーのインストールと構成」の「JASログの構成」を参照してください。
全オブジェクトの生成の前に、コア・オブジェクトの生成手順が追加されました。	全オブジェクトの生成 OneWorld/ERPのシリアル化オブジェクトの生成
<code>jas.ini</code> ファイルから、多数のパラメータが <code>JDBj.ini</code> ファイルに移動されました。	付録の修正： Appendix A: <code>jas.ini</code> ファイルのパラメータと値
サンプル <code>jas.ini</code> ファイルが新規ファイルに置き換えられました。	付録の修正： Appendix B: サンプル <code>jas.ini</code> ファイル
新規構成ファイル: <code>JDBj.ini</code>	新規の付録： Appendix C: <code>jdbj.ini</code> ファイルのパラメータと値
<code>JDBj.ini</code> サンプル・ファイルに関する新規の付録が追加されました。	新規の付録： Appendix D: サンプル <code>JDBj.ini</code> ファイル

サードパーティ・ソフトウェアのインストール

ここでは、Webサーバーの実行に必要なWebSphereと他のサードパーティ製品をインストールする方法について説明します。

ここでは次の内容について説明します。

- ・ [WebSphere要件の理解 - ページ \[5\]](#)
- ・ [WebSphereリポジトリ用データベースの構成 - ページ \[6\]](#)
- ・ [IBM HTTPサーバーとWebSphere \(バージョン4.0\) のインストール - ページ \[9\]](#)
- ・ [WebSphere FixPackの適用 - ページ \[18\]](#)
- ・ [IBM HTTPサーバーの構成 - ページ \[18\]](#)
- ・ [WebSphereの起動 - ページ \[20\]](#)
- ・ [WebSphereアプリケーション・サーバーのテスト - ページ \[21\]](#)

注意

特に明記しない限り、すべてのインストール画面でデフォルト値を受け入れてください。開発に必要なステップについては、その旨を記載してあります。

WebSphere要件の理解

WebSphereには複数のバージョンがあり、それぞれ要件とインストール・タスクが異なります。

サポートされているWebSphereのバージョン

IBM WebSphereアプリケーション・サーバー(バージョン4.0、FixPack 3以上付き)を購入する必要があります。

J.D. Edwards Javaアプリケーション・サーバーでは、次のバージョンのWebSphere 4.0がサポートされています。

- ・ WebSphere Advanced Edition (AE)

WebSphere Advanced Edition™(AE)では複数のアプリケーション・サーバーを実行できますが、その構成情報を格納するデータベース・リポジトリが必要です。リポジトリ要件については、下記の「[必要なデータベース・リポジトリ - ページ \[6\]](#)」を参照してください。

WebSphereの入手

WebSphereは、CDからロードするか、次のURLからダウンロードできます。

<http://www-4.ibm.com/software/webservers/appserv/>

WebSphere FixPackも、次のURLから入手できます。

<http://www-3.ibm.com/software/webservers/appserv/support.html>.

はじめる前に

WebSphere 3.5からWebSphere 4.0に移行する場合は、次のタスクを行う前に「Appendix E: WebSphere 4.0への移行」の指示に従って操作する必要があります。

必要なデータベース・リポジトリ

WebSphere Advanced Edition (AE)アプリケーションには、構成設定を格納するためのデータベース・リポジトリが必要です。

データベースには、次のいずれかを使用できます。

- ・ IBM DB2汎用データベース。
- ・ JDBC-Thin/100% Javaドライバ組込みのOracle。

J.D. Edwardsでサポートされているデータベースのバージョンについては、J.D. Edwardsの技術的な最小要件を参照してください。

データベースは、Webサーバーと同じマシン上または異なるマシン上で運用できます。各種データベースのインストール事例については、IBMマニュアルを参照してください。次のリンクを使用すると、WebSphereインフォメーション・センターにアクセスして、各種プラットフォーム用のインストール・ガイドを入手できます。

<http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/index.html> (WebSphere AE)

警告

これらのデータベースの1つのみをWebSphereリポジトリとして構成してください。複数のデータベースを構成しようとすると、さまざまな問題が発生する可能性があります。

WebSphereリポジトリ用データベースの構成

次のいずれかのタスクを選択し、WebSphereリポジトリとして使用するデータベースを構成します。

- ・ [DB2/UDBの構成 - ページ \[7\]](#)
- ・ [Oracle \(Oracle 9i\)の構成 - ページ \[8\]](#)

注意

J.D. Edwards WebサーバーをWindowsまたはUNIXマシンにインストールし、iSeriesマシンでWebSphereリポジトリとしてDB2を使用する場合は、「Appendix H: WebSphereリポジトリとしてのiSeries DB2の使用」を参照してください。

警告

これらのタスクのうち1つのみを実行します。複数タイプのデータベースをWebSphere™リポジトリとして構成しないでください。

DB2/UDBの構成

Javaサーバーにデータベースをインストールする方法と、リモート・データベースを使用する方法があります。JavaサーバーからDB2/UDBを実行するリモート・データベース・サーバーに接続する場合は、2台のサーバーが通信できるように、JavaサーバーにDB2/UDBのクライアントをインストールする必要があります。

リモート・データベースまたはローカル・データベースを使用する場合は、どちらを使用する場合もデータベース・サーバー上にリポジトリ・データベースを作成する必要があります。

▶ DB2/UDBを構成するには

1. DB2/UDBデータベースに添付されている指示に従って、Javaサーバーまたはデータベース・サーバーにインストールします。
2. データベースがあるサーバーに、DB2/UDBのインスタンス所有者としてログオンします(たとえば、db2inst1でログオンします)。
3. 次のコマンドを入力して、WebSphereリポジトリ・データベースを作成します。

```
>db2 create database was40
```

```
>db2 update db cfg for was40 using applheapsz 256
```

4. リポジトリ・データベースをリモート・データベース・サーバー上に作成した場合は、次のコマンドを入力してリモート・ノードとデータベースをカタログ化します。

```
>db2 catalog tcpip node <DBServer> remote <DBServerNodeName> server <port>
```

```
>db2 catatlog database was40 at node <DBServer>
```

<DBServer>はDB2/UDBでデータベース・サーバーの認識に使用される名称、<DBServerNodeName>はネットワーク上で認識されるデータベース・サーバー名です。この名称はIPアドレスでもかまいません。<port>はDB2/UDBのサービス・ポートです。

ヒント

ポートが不明な場合は、データベース・サーバー上の次のファイルで指定する必要があります。

```
/etc/services
```

デフォルト値は50000です。

たとえば、サーバー名がhpdev1で、このサーバーがネットワーク上でも同じ名称で認識される場合は、次のコマンドを入力できます。

```
>db2 catalog tcpip node hpdev1 remote hpdev1 server 50000
```

```
>db2 catatlog database was40 at node hpdev1
```

Oracle (Oracle 9i)の構成

Oracleデータベースを使用する場合は、WebSphereアプリケーション・サーバーをインストールする前に、Oracleユーザーを作成する必要があります。

Oracleデータベースは、Javaサーバーにインストールする方法と、リモートOracleデータベースを使用する方法があります。リモートデータベースを使用する場合は、2台のサーバーが通信できるように、JavaサーバーにOracleクライアントをインストールする必要があります。

▶ Oracleを構成するには

1. 次のコマンドを使用し、権限ユーザーとしてOracleにログインします。

```
>sqlplus system/manager@connect_string
```

systemはユーザー名、managerはパスワード、connect_stringはtnsnames.oraファイルに定義されているOracle接続文字列です。

2. 次のユーザーを作成し、同じユーザー名で識別します。

```
"ejsadmin"
```

たとえば、次のように入力します。

```
> create user ejadmin identified by password
```

ユーザーはejadmin、パスワードはejadminまたは他の値です。

3. ユーザーejadminに次の権限を付与し、該当する表領域を割り当てます。

- ・ dba(データベース管理者)
- ・ Connect(接続)
- ・ Resource(リソース)

たとえば、次のように入力します。

```
>grant dba, connect, resource to ejadmin
```

```
>alter user ejadmin default tablespace users;
```

```
>alter user ejadmin temporary tablespace temp;
```

usersとtempを該当する表領域で置き換えてください。

4. splplusの接続を切断します。

IBM HTTPサーバーとWebSphere (バージョン4.0) のインストール

はじめる前に

- ・ Webサーバーに関連付けるすべてのアプリケーションのインストールと構成には、“root” UNIXアカウントを使用することをお勧めします。
- ・ WebSphereの要件を満たしていることを確認する必要があります。これにはオペレーティング・システムの必須パッチが含まれます。

IBM WebSphereアプリケーション・サーバーの前提条件については、次のリンクを使用して確認してください。

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

- ・ インストール・タイプの選択時には、デフォルトの[Typical Installation (標準インストール)]ではなく[Custom Installation (カスタム・インストール)]を選択することをお勧めします。

▶ IBM HTTPサーバーとWebSphere (バージョン4.0) をインストールするには

1. まだ設定していない場合は、ディスプレイを設定します。たとえば、ローカル・ディスプレイのIPアドレスが10.0.151.1:0.0の場合は、次のコマンドを入力するとディスプレイが正しく設定されます。

```
export DISPLAY=10.0.151.1:0.0
```

2. WebSphere™インストールCDを挿入します。
3. 次のコマンドを入力してインストール・プログラムを実行します。

```
mount /cdrom
```

注意

mountコマンドの構文はプラットフォームごとに異なり、Sunプラットフォームの場合は不要です。HP-UXの場合は、pfs_mountコマンドで正しいマウント・オプションを使用してください。

```
cd /cdrom
```

```
cd / platform
```

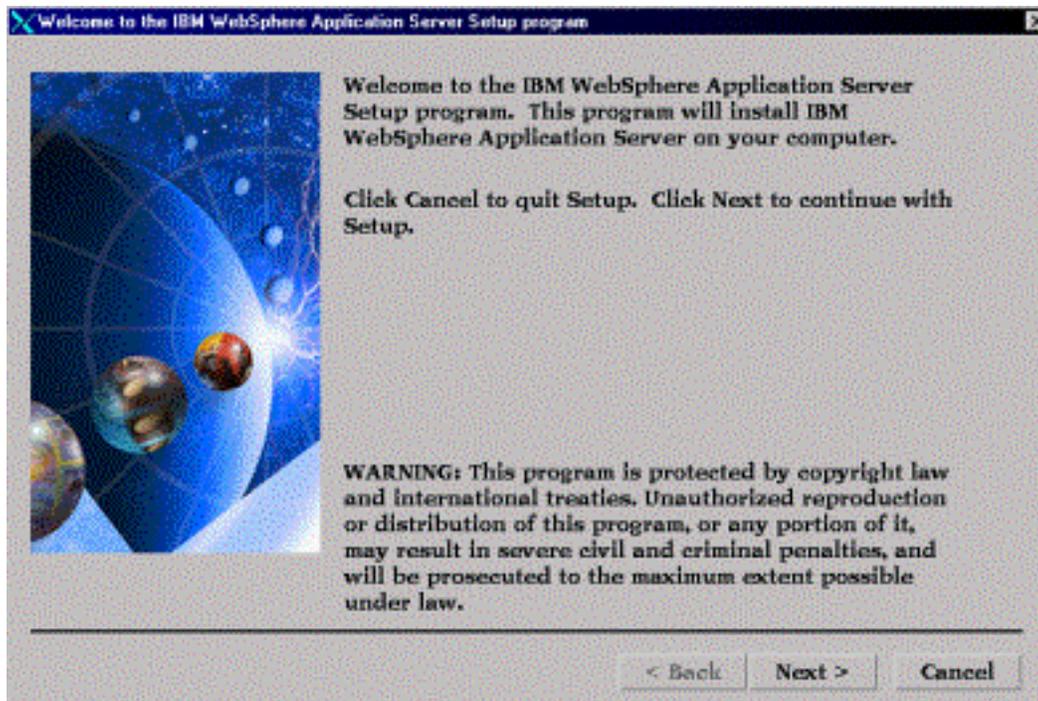
platformはWebサーバーのプラットフォームです。有効な値は次のとおりです。

- ・ aix

- sun
- hpux

たとえば、次のように入力します。

```
cd /aix  
./install.sh
```



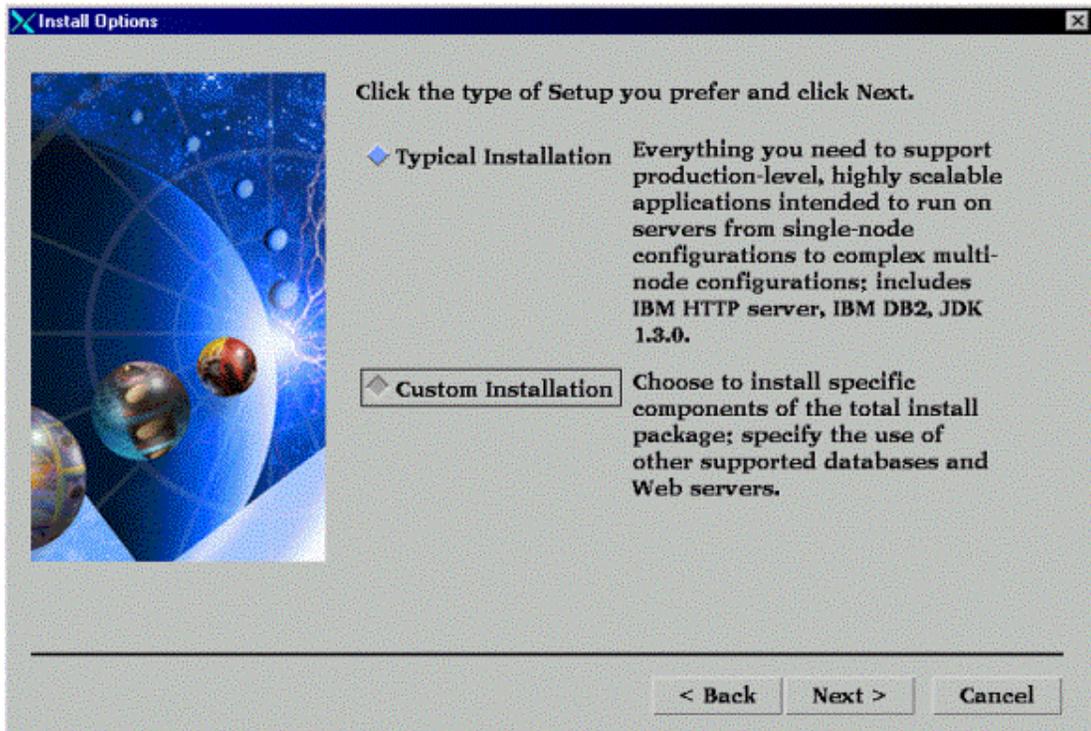
4. [Welcome]画面で[Next]をクリックします。
5. サーバー・マシンがすべての要件を満たしていない場合は、プロンプト画面が表示されます。

[Prerequisites Check(前提条件のチェック)]で、前提条件が満たされていることと、WebサーバーのMTRに対応していることを確認します。

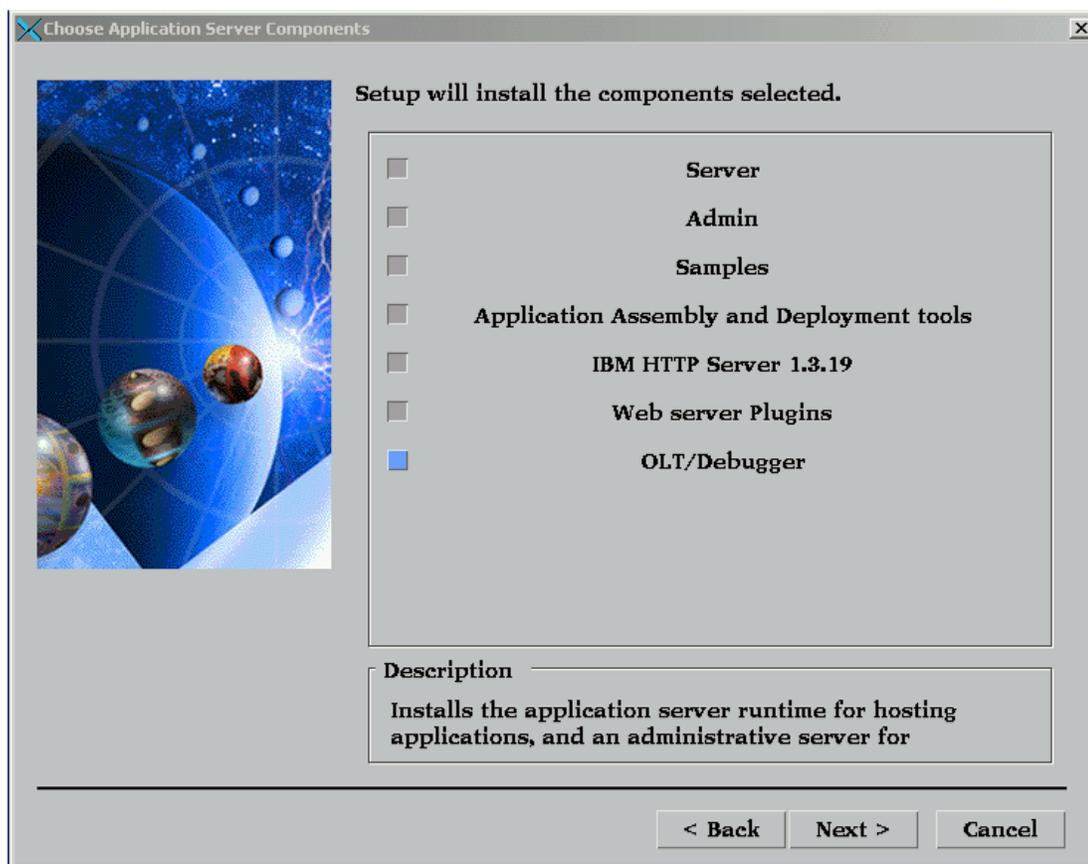
注意

満たされていない要件があることを示すメッセージが表示される場合は、[Cancel(キャンセル)]をクリックし、システムを最低限必要なレベルまでアップグレードします。

6. [Next]をクリックします。



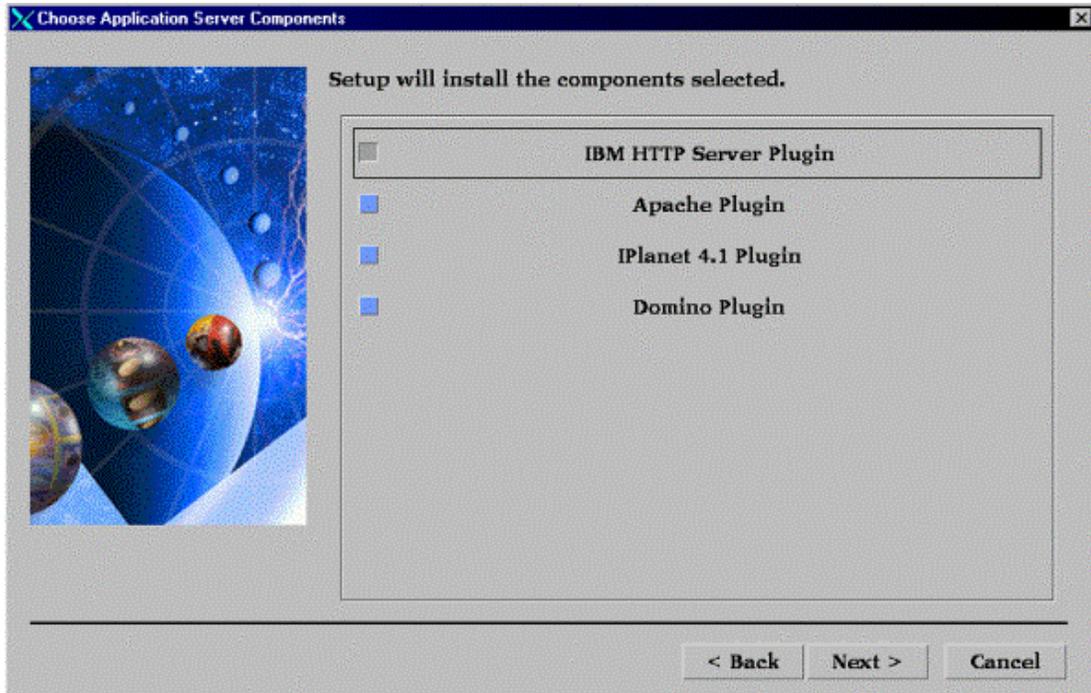
7. [Install Option(インストール・オプション)]で、[Custom Installation]を選択します。
8. [Next]をクリックします。



9. [Choose Application Server Components(アプリケーション・サーバー・コンポーネントの選択)]で、次のオプションをすべて選択します。

- ・ Server(サーバー)
- ・ Admin
- ・ Samples(サンプル)
- ・ Application Assembly and Deployment tools(アプリケーション・アセンブリおよびデプロイメント・ツール)
- ・ IBM HTTP Server(IBM HTTPサーバー)
- ・ Web Server Plugins(Webサーバー・プラグイン)

10. [Next]をクリックします。



11. 2番目の[Choose Application Server Components]画面で、[IBM HTTP Server Plugin]を選択します。
12. [Next]をクリックします。
13. [Database Options (データベース・オプション)]で、WebSphereリポジトリとして選択したデータベース・タイプに関するデータベース情報を入力します。

DB2/UDBの場合： 次のフィールドに値を入力します。

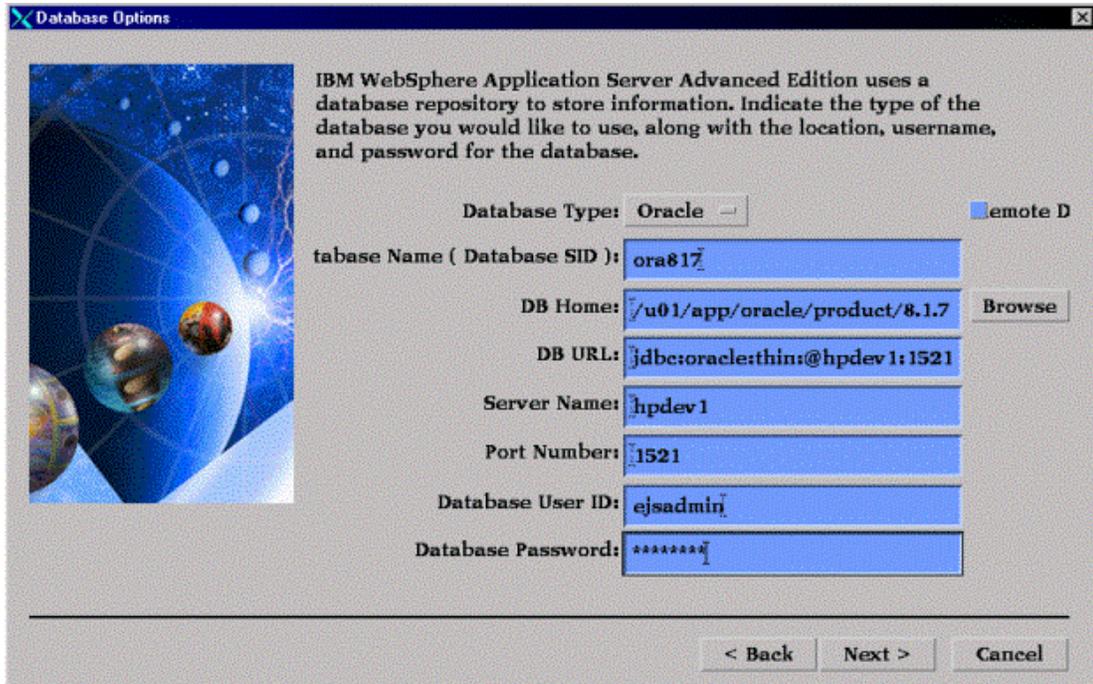
- ・ [Database Type (データベース・タイプ)]：“DB2”
- ・ [Remote Database (リモート・データベース)]：データベース・サーバーが同じシステム上にない場合は、このオプションを有効にします。
- ・ [Database Name (データベース名)]：“was40”
前に作成したWebSphereリポジトリ・データベースの名称。
- ・ [DB Home (DBホーム)]：“/home/db2inst1”
リポジトリ・データベース・ディレクトリへのパス。
- ・ [DB URL]：“jdbc:db2:was40”
- ・ [Server Name (サーバー名)]：DB2/UDBサーバー名を入力します。
- ・ [Port Number (ポート番号)]：デフォルトのポート番号を受け入れます。
この値を変更するのは、DB2ポート番号を変更する場合のみです。
- ・ [Database User ID (データベース・ユーザーID)]：“db2inst1”

データベース・リポジトリのインスタンス所有者。データベースに有効な値を使用してください。

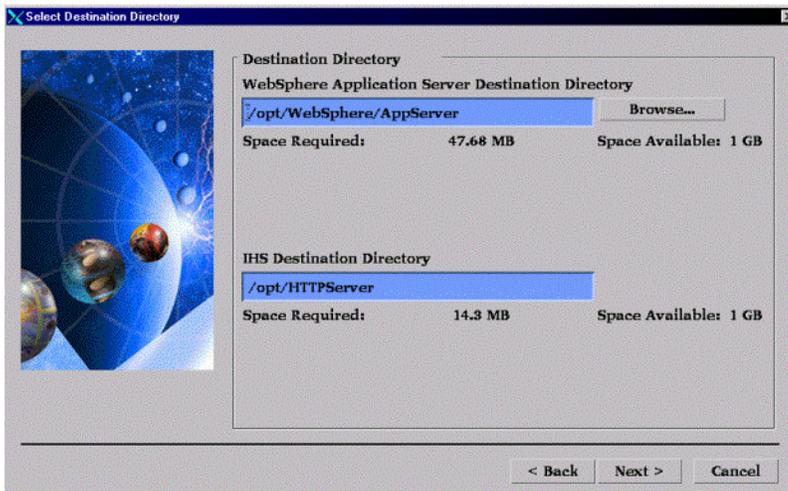
- ・ [Database Password(データベース・パスワード)]: データベース・リポジトリのインスタンス所有者のパスワード。

Oracleの場合 次のフィールドに値を入力します。

- ・ [Database Type]: "Oracle"
- ・ [Remote Database]: データベース・サーバーが同じシステム上にない場合は、このオプションを有効にします。
- ・ [Database Name]: Oracle SID (ora901など)を入力します。
- ・ [DB Home]: Oracle DBホーム・ディレクトリ。
Oracleクライアントまたはサーバーが置かれているディレクトリを入力します。
- ・ [DB URL]: デフォルト値を受け入れます。
このフィールドの値は、関連フィールドを修正すると自動的に更新されます。
- ・ [Server Name]: Oracleサーバー名を入力します。
- ・ [Port Number]: デフォルトのポート番号を受け入れます。
この値を変更するのは、Oracleポート番号を変更する場合のみです。
- ・ [Database User ID]: "ejsadmin"
- ・ [Database Password]: ejadminのパスワードを入力します。

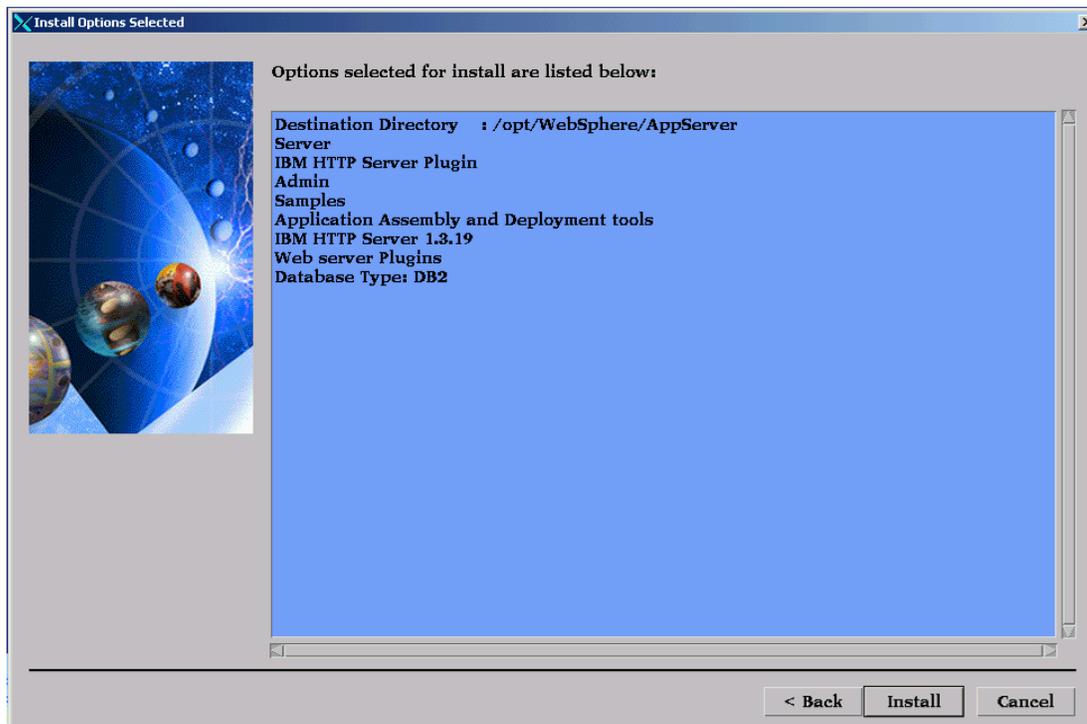


14. [Next]をクリックします。

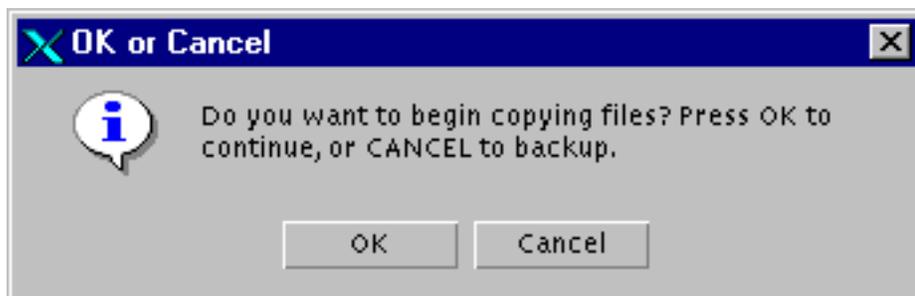


15. [Select Destination Directory(インストール先ディレクトリの選択)]には、デフォルトでHTTPサーバーのIHSインストール先ディレクトリが設定されます。この設定は変更できません。ただし、WebSphereのインストール先ディレクトリは変更できます。WebSphere™とIHSは統合されるので、WebSphere™のインストール先を指定する必要があります。

16. [Next]をクリックします。



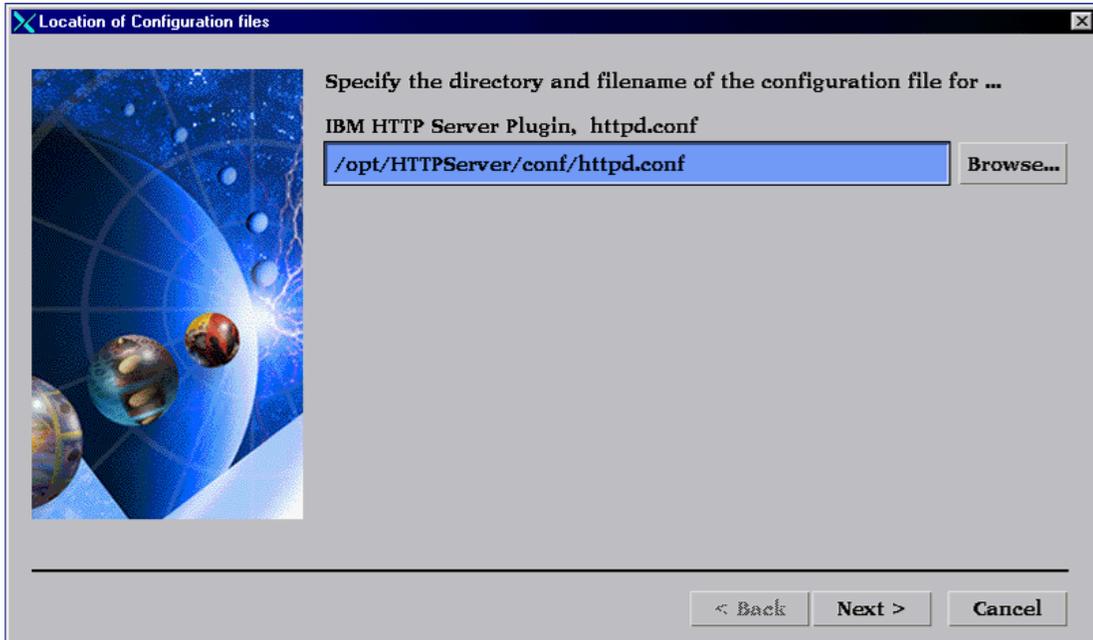
17. [Install Options Selected(選択したインストール・オプション)]で[Next]をクリックします。



18. [OK or Cancel(OKまたは取消)]ダイアログで[OK]をクリックします。

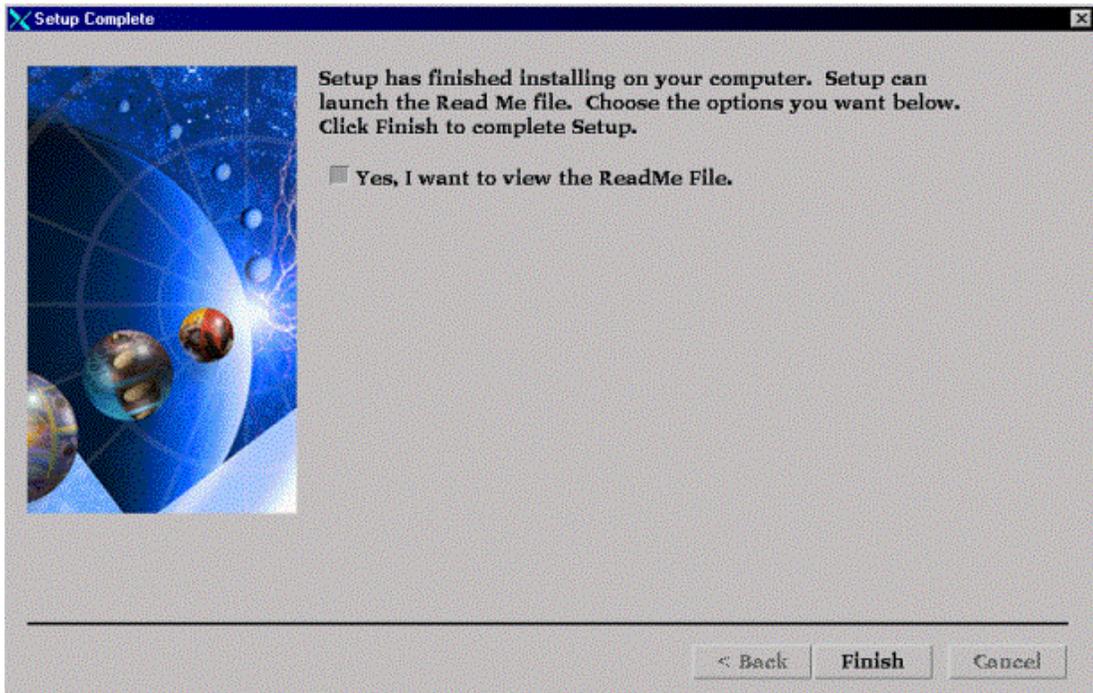
インストール・プロセスにより、CDからWebサーバーへのファイルのコピーが開始されます。

コピー・プロセスが完了すると、http.confファイルの位置指定を求めるプロンプトが表示されます。



19. 構成ファイルのパスとファイル名を入力します。

- ・ HP-UXの場合 : /opt/HTTPServer/conf/httpd.conf
- ・ Solarisの場合 : /opt/IBMHTTPD/conf/httpd.conf
- ・ AIXの場合 : /usr/HTTPServer/conf/httpd.conf



20. [Setup Complete(セットアップの完了)]で[Finish]をクリックします。

WebSphere FixPackの適用

必要なFixPackは、J.D. Edwardsの技術的な最小要件を参照して判断してください。次のタスクを実行してFixPackを適用します。

▶ WebSphere FixPackを適用するには

1. WebSphereをインストール後に初めて起動する場合は、[WebSphere Application Server First Steps]アプリケーションを閉じます。
2. 次のコマンドを入力して、WebSphereとIBM HTTPサーバーが実行されていないことを確認します。

```
>ps -ef | grep java  
  
>ps -ef | grep http
```
3. Javaプロセスとhttpプロセスの出力をチェックします。
4. WebSphereまたはHTTPサーバーを実行している場合は、これらのアプリケーションを停止します。
5. 技術的な最小要件を参照して最新のWebSphere FixPackを判断し、それをサーバーの一時ディレクトリにダウンロードします。

FixPackは、次のIBM WebSphere Application Server Webサイトからダウンロードできます。

<http://www-3.ibm.com/software/webservers/appserv/support.html>

FixPackと共に、リリース・ノートとREADMEファイルもダウンロードしてください。これらのファイルには、FixPackのインストールと構成に関する手順の詳細が含まれています。

6. 通常、FixPackファイルはtar形式になっています。次のコマンドを使用すると、ファイルを復元できます。

```
>tar -xvf filename
```

filenameはtarファイル名(was40_ae_ptf_3_<platform>.tarなど)です。

7. READMEファイルとリリース・ノートにある手順に従って、WebSphere FixPackをインストールします。

このtempディレクトリ内でinstall.shを使用してFixPackをインストールします。

FixPackのインストール時には、WebSphereアプリケーション・サーバー、IBM JDK、IBM HTTPサーバーも同時に更新するように選択します。

IBM HTTPサーバーの構成

httpd.confファイルをサイトに合わせて更新する必要があります。ここでは次の内容について説明します。

- ・ [WebSphereをサポートするためのIBM HTTPサーバーの構成 - ページ \[19\]](#)

- ・ [httpサービスの開始 - ページ \[19\]](#)

WebSphereをサポートするためのIBM HTTPサーバーの構成

▶ WebSphereをサポートするようにIBM HTTPサーバーを構成するには

1. Webサーバー上でWebサーバー構成ファイルを指定します。通常、ディレクトリ・パスは次のとおりです。

- ・ AIXの場合

/usr/HTTPServer/conf/httpd.conf

- ・ Solarisの場合

/opt/IBMHTTPD/conf/httpd.conf

- ・ HP-UXの場合

/opt/HTTPServer/conf/httpd.conf

2. httpd.confのServerNameディレクティブが有効なサーバー名に設定されていることを確認します。存在しない場合は、次の2行を追加します。

```
#ServerName
```

```
ServerName server_name
```

server_nameは有効なサーバー名です。

httpサービスの開始

httpd.confファイルを修正してサイトの有効なサーバーを組み込んだ後、サービスを再開し、稼働することを確認する必要があります。

▶ httpdサービスを開始(稼働することを確認)するには

1. httpdサービス制御用の実行可能スクリプトを含むディレクトリに移動します。このディレクトリはオペレーティング・システムに応じて異なりますが、通常は次のとおりです。

- ・ AIXの場合

/usr/HTTPServer/bin/apachectl

- ・ Solarisの場合

/opt/IBMHTTPD/bin/apachectl

- ・ HP-UXの場合

```
/opt/HTTPServer/bin/apachectl
```

2. 次のコマンドを使用してhttpdサービスを開始します。

```
>./apachectl start
```

3. httpdサービスが開始され、稼働していることを確認します。

注意

httpdサービスが開始されない場合は、/etc/hostsファイルで完全修飾ホスト名(server.domain.com)が指定されているかどうかを確認してください。

WebSphereの起動

これ以降に説明するステップを実行するには、WebSphereを起動する必要があります。このタスクはIBM HTTPサーバーに依存しないため、IBM HTTPサーバーの構成前に実行できます。

使用中のWebSphereのバージョンに合わせてタスクを選択してください。

- ・ [手順 7 \[20\]](#)

▶ WebSphere Administrative Consoleを起動するには(AEの場合)

Webサーバーのコマンド行から、

1. 次のコマンドを使用してhttpdサービスが開始されていることを確認します。

```
>ps -ef | grep httpd
```

httpdサービスの開始手順については、「[httpサービスの開始 - ページ \[19\]](#)」を参照してください。

2. WebSphereアプリケーション・サーバーを起動するには、次のスクリプトを実行します。通常、このスクリプトはWebSphere™基本ディレクトリ直下の/binディレクトリにあります。

```
>./startupServer.sh port_number
```

WebSphereの基本ディレクトリは次のとおりです(本書では、<WAS_HOME>と表記しています)。

- ・ AIXの場合 : /usr/WebSphere/AppServer
- ・ HP-UXの場合 : /opt/WebSphere/AppServer
- ・ Solarisの場合 : /opt/WebSphere/AppServer

ヒント

次のコマンド行を含むshellスクリプトを作成することをお勧めします。

```
>./startupServer.sh port_number &
```

port_numberは任意です。デフォルト・ポート9000が使用中であることを示すエラーが表示される場合は、異なるポートを指定できます。たとえば、次のように入力します。

```
>./startupServer.sh 9001
```

このコマンドで"&"を指定すると、WebSphere™サーバー・プロセスはバックグラウンドで開始されます。このオプションを使用すると、WebSphere™を停止せずに現行のコンソール・セッションを終了できます。

特に初回には、サーバーの起動に時間がかかることがあります。起動処理が完了したかどうかを判断するには、次のように入力してlogsディレクトリに移動します。

```
>cd <WAS_HOME>/logs
```

<WAS_HOME>はWebSphereの基本インストール・ディレクトリです。デフォルトは、SunとHP-UXの場合は/opt/WebSphere/AppServer、AIXの場合は/usr/WebSphere/AppServerです。

次のコマンドを実行してトレースファイルをチェックします。

```
>tail -f tracefile
```

サーバーの起動を示す次のようなメッセージが表示されます。

```
212.366 6005c5e AdminServer A WebSphere Administration server open for e-business.
```

3. 管理サーバーの起動完了後に、次のスクリプトを実行できます。このスクリプトもWebSphereのbinディレクトリにあります。

```
>./adminclient.sh &
```

このコマンドにより[WebSphere Administrative Console]が起動されます。

WebSphereアプリケーション・サーバーのテスト

次のタスクを実行して、WebSphereアプリケーション・サーバーが正しく構成されているかどうかを確認します。

▶ WebSphereアプリケーション・サーバーをテストするには(AEの場合)

Webサーバー上で、

1. [WebSphere Administrative Console]を開きます。
2. [WebSphere Administrative Domain (WebSphere管理ドメイン)]のアイコンを展開します。
3. [Nodes (ノード)]アイコンを展開します。

4. サーバー・ホストのアイコンを展開します。
5. [Applications Servers(アプリケーション・サーバー)]アイコンを展開し、[Default Server(デフォルト・サーバー)]を選択します。
6. 画面上部の[Start(起動)]をクリックし、デフォルトのWebSphereアプリケーション・サーバーを起動します。

サーバーが正常に起動すると、次のメッセージが表示されます。

“Default server.start” completed successfully (“Default server.start”が正常に起動しました。)

7. 上記のメッセージが表示されない場合は、問題を解決してサーバーを再起動します。
8. クライアント・マシンで、ブラウザを開いて次のURLを入力します。

`http://servername/servlet/snoop`

servernameはWebサーバー名です。

要求したURLと、各種ブラウザからサーバーへの通信情報も表示されます。

9. [WebSphere Administrative Console]で、デフォルト・サーバーを停止します。

Webサーバーのインストールと構成

ここでは、J.D. Edwardsインストール・プログラム(JASインストーラ)を使用して、Webサーバーをインストールし、WebSphereでWebサーバーのインスタンスを構成し、jas.iniファイルを構成します。

このインストール用に用意されているJava Server CDを使用して、Webサーバー・コードをインストールします。

次のタスクを実行して、Webサーバーと関連ソフトウェアをインストールします。

- ・ [Webサーバーのインストール - ページ \[23\]](#)
- ・ [OneWorld/ERPデータベース接続\(JDBCドライバ\)の構成 - ページ \[34\]](#)
- ・ [OneWorld/ERPインストール・プランへのWebサーバーの追加 - ページ \[37\]](#)
- ・ [JASをサポートするためのIBM HTTPサーバーの構成 - ページ \[39\]](#)
- ・ [スタティック・ファイルの処理の構成 - ページ \[40\]](#)
- ・ [JASログの構成 - ページ \[41\]](#)

はじめる前に

- ・ WebSphere Adminのポート番号を900(デフォルト)以外の値に変更した場合は、Javaサーバーをインストールする前に元の900に戻す必要があります。元のポート番号に戻さないと、インストール・プロセスが失敗します。このプロセスの完了後に、ポート番号をカスタマイズした値に変更できます。

Webサーバーのインストール

[InstallShield]ウィザードを使用して、J.D. Edwards Javaアプリケーション・サーバーをコンピュータにインストールします。

はじめる前に rootとしてログインしていることと、ローカル・ディスプレイがマシンのIPアドレスに設定されていることを確認します(たとえば、コマンドexport DISPLAY= ip_address:0.0を使用できます)。

▶ Webサーバーをインストールするには

1. WebSphereが実行されていることを確認します。
 - ・ WebSphereを起動するには、「WebSphereの起動」を参照してください。
 - ・ WebSphereが実行されていることを確認するには、「WebSphereアプリケーション・サーバーのテスト」を参照してください。
2. J.D. Edwards Java Installation CDをCD ROMドライブに挿入するか、Knowledge Gardenの[Update

Center(アップデート・センター)]からプラットフォーム固有のWebサーバー・インストール・ソフトウェアをダウンロードします。

注意

ソフトウェアをWindowsマシンにダウンロードした場合は、それをUnix WebサーバーにFTPする必要があります。

-
3. コマンド行から、Java Server Installation CDの基本ディレクトリに移動するか、インストール・ソフトウェアをダウンロードしたディレクトリに移動します。
 4. ソフトウェアを[Update Center]からダウンロードした場合は、次のコマンドを実行し、ダウンロード・ファイルに対する許可を変更します。

```
CHMOD 777 <filename>
```

<filename>はUNIXプラットフォーム用のダウンロード・ファイル名(SPxxJ_RS600など)です。

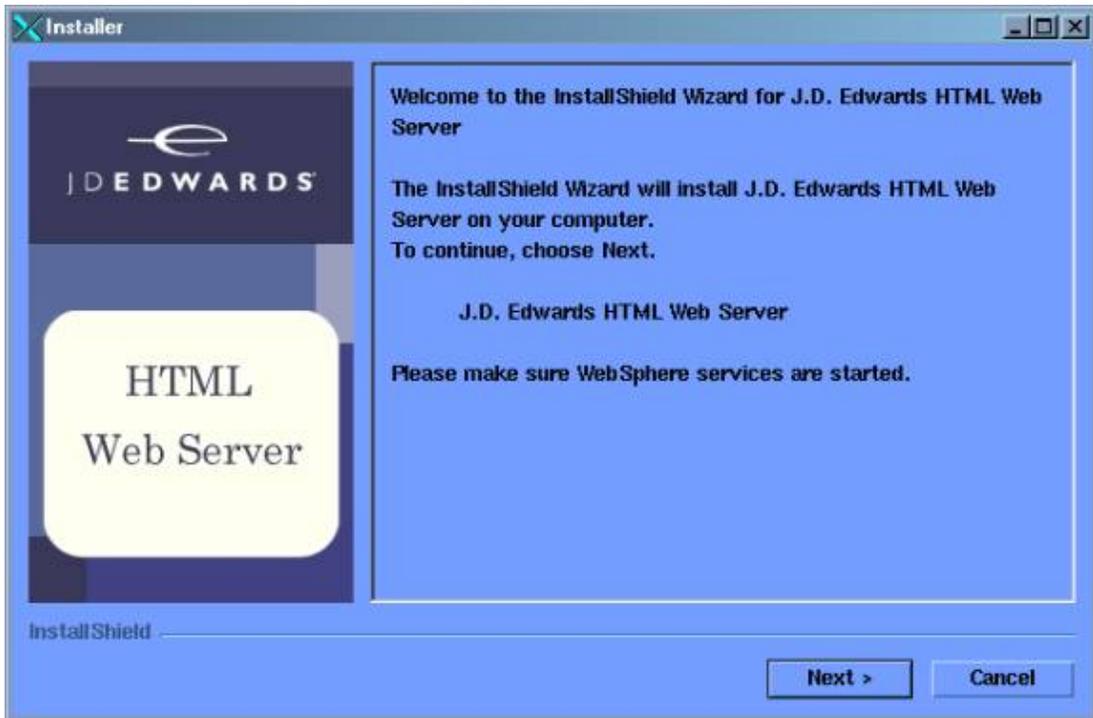
5. 次のオプションのうち1つを選択します。
 - ・ CDからInstallerを実行する場合は、次のいずれかのコマンドを入力します。
 - ・ AIXの場合 : setup_aix
 - ・ HP-UXの場合 : setup_hpux
 - ・ Solarisの場合 : setup_solaris
 - ・ [Update Center]からInstallerを実行する場合は、次のいずれかのコマンドを入力します。
 - ・ AIXの場合 : ./SPxxJ_RS600
 - ・ HP-UXの場合 : ./SPxxJ_HP9000
 - ・ Solarisの場合 : ./SPxxJ_SUN

xxはサービス・パック番号です(SP20J_SUN.exeなど)。

注意

このコマンドを実行するには、UnixのパスにJavaが指定されている必要があります。

J.D. Edwardsインストーラ・プログラムがロードされるまでに数分かかることがあります。



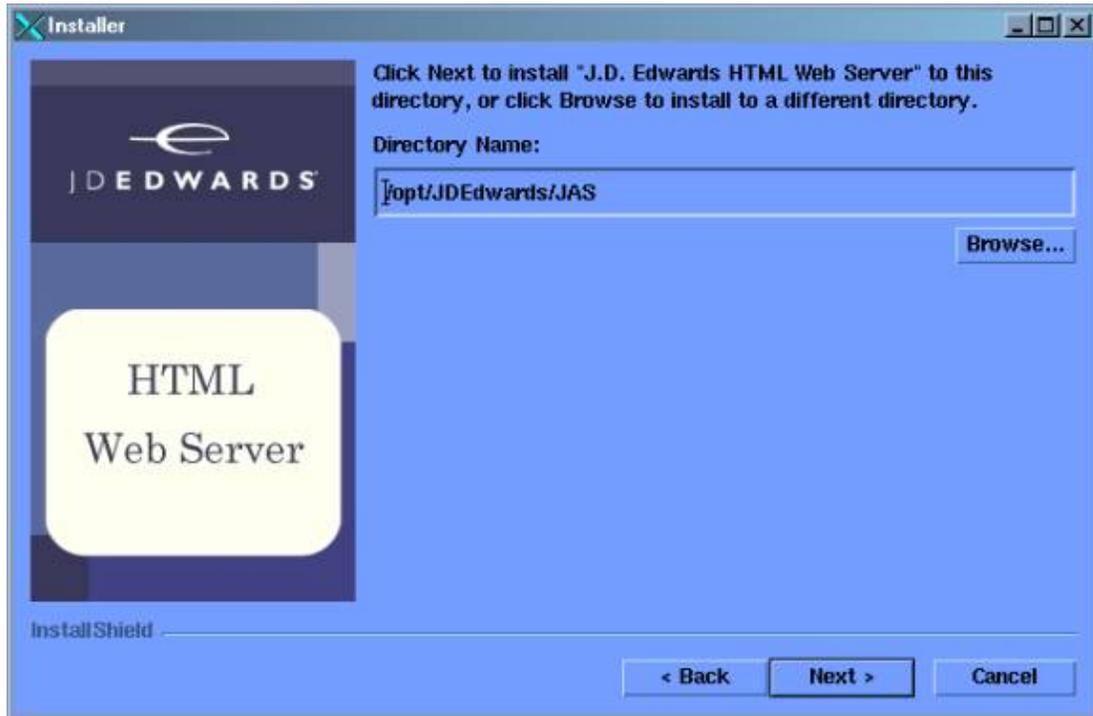
6. [Next]をクリックします。



[WebSphere Setup (WebSphereのセットアップ)]に、WebSphereのデフォルトのインストール・ディレクトリ(<WAS_HOME>)が表示されます。

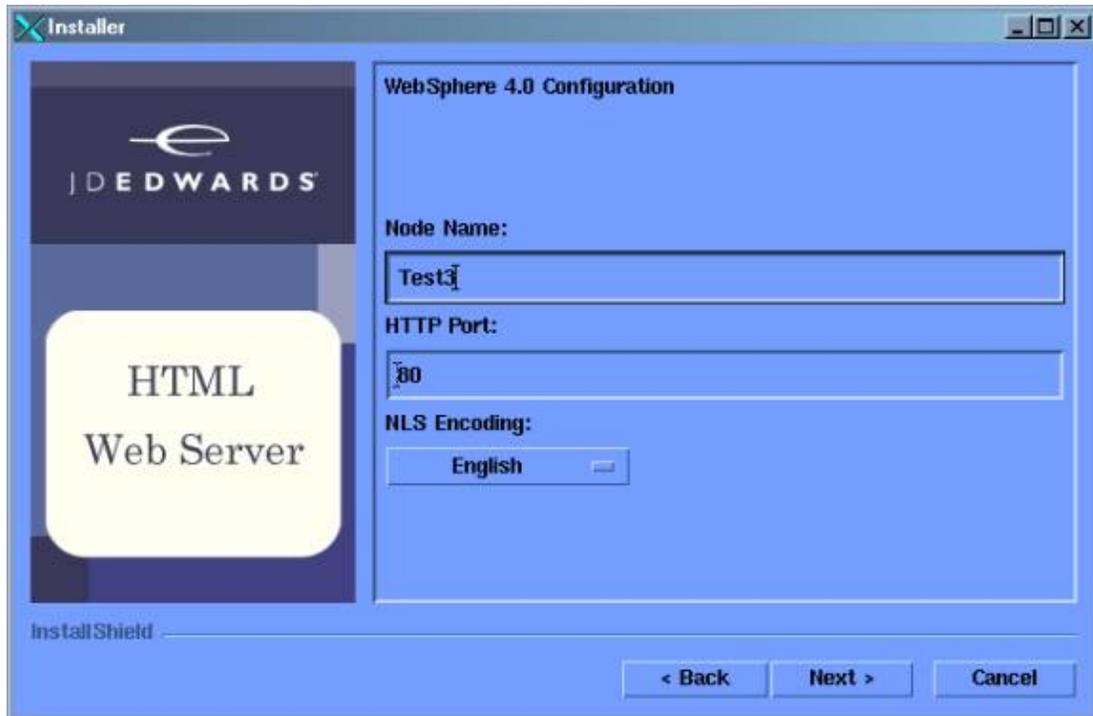
- ・ HP-UXとSolarisの場合 : /opt/WebSphere/AppServer.

- ・ AIXの場合 : /usr/WebSphere/AppServer.
7. デフォルト・パス以外のディレクトリにWebSphereをインストールした場合は、そのパスを[WebSphere Installation Directory (WebSphereインストール・ディレクトリ)]フィールドに入力します。
 8. [Next]をクリックします。



Webサーバーのデフォルトのインストール・パスが表示されます。

9. J.D. Edwards Javaサーバーを別のディレクトリにインストールする場合は、[Browse(参照)]をクリックして別のディレクトリを選択します。
10. [Next]をクリックします。



[WebSphere Configuration (WebSphere構成)]画面に、WebSphereでJavaサーバーをセットアップするために必要なデフォルト設定が表示されます。

11. [WebSphere Configuration]のほとんどの設定フィールドは、デフォルトで正しい値に設定されます。ただし、次のフィールド記述を検討し、必要に応じてフィールド値を確認または変更できます。

- ・ [Node Name (ノード名)]:

[WebSphere Administrative Console]に表示されるとおりのマシン名を使用します。このマシン名が正しいかどうかを確認するには、[WebSphere Administrative Console]を開き、[WebSphere Administrative Domain]を展開して、[Nodes]アイコンの下にあるマシン名を確認します。この値は大文字と小文字が区別されることに注意してください。

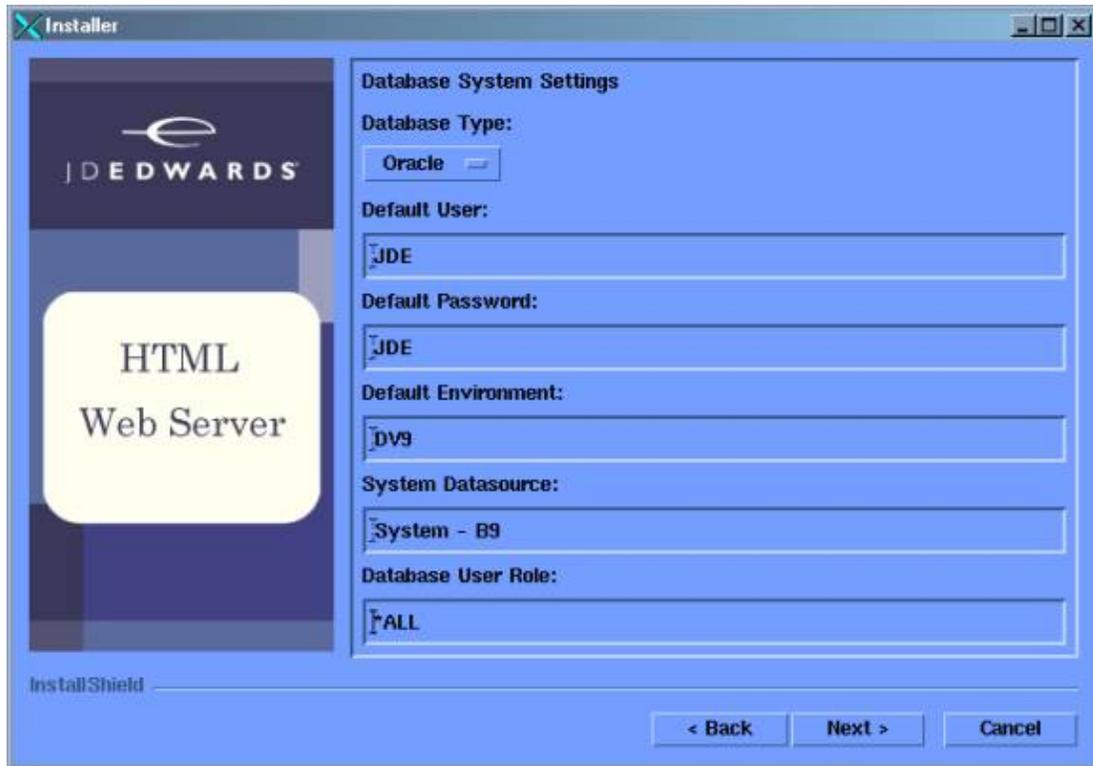
- ・ [HTTP Port (HTTPポート)]: デフォルト・ポート(80)の場合は、J.D. Edwardsアプリケーション・サーバー(AS_JDEdwards)がデフォルトのホストに関連付けられ、J.D. Edwards構成の一部として新規仮想ホストは作成されません。

J.D. Edwards Javaサーバー用に別の仮想ホストをセットアップする場合は、この値を別のポート番号(81または82など)に変更します。値を変更すると、J.D. Edwards WebSphere構成をデフォルト構成から変更できるので、WebSphereのパフォーマンスを向上できる可能性があります。

- ・ [NLS Encoding (NLSエンコード)]: Javaサーバーに英語以外の言語をインストールする場合は、その言語を選択します。

他のWebSphere設定の多くは、ユーザーが入力しなくてもインストーラにより自動的に構成されます。

12. [Next]をクリックします。



13. 残りの画面では、jdbj.iniファイルの重要な設定を構成します。[Database System Settings(データベース・システム設定)]画面には、jdbj.iniファイルの[JDBj BOOTSTRAP SESSION]セクションの重要な設定が表示されます。これらの設定は、OneWorld/ERPデータベース・サーバー上のOneWorld/ERPデータベースへの接続を定義します。詳しくは、「[JDBj-BOOTSTRAP SESSION]」を参照してください。これらの設定については、CNC管理者に問い合わせるか、OneWorld/ERPのインストール・ガイドを参照してください。

これらのフィールドのほとんどは、デフォルトで正しい値に設定されます。

ヒント

このインストール・プロセス中に、jas.iniファイルまたはjdbj.iniファイルの設定のどちらかに間違っ
た値を入力した場合は、該当するファイルを開いて設定を手作業で編集できます。これらのフ
ァイルの各設定については、「Appendix A: jas.iniファイルのパラメータと値」を参照してください。

次のフィールドを確認し、必要に応じて変更します。

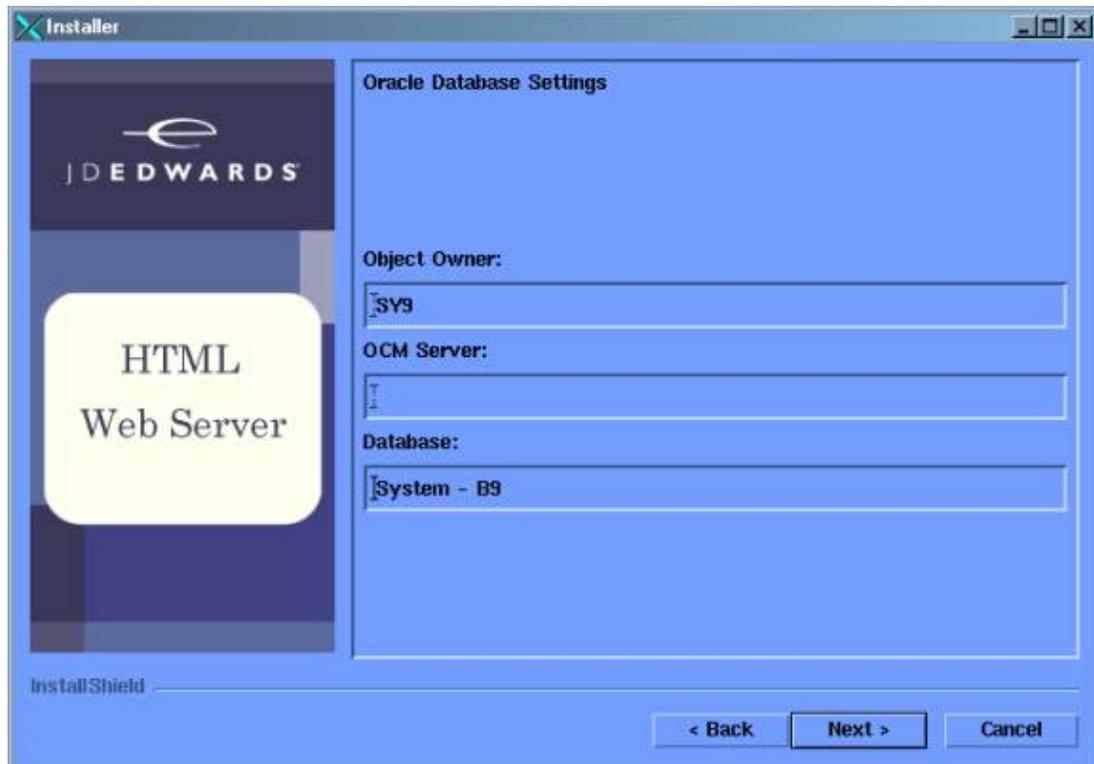
- ・ [Database Type]: "Oracle"、"MS SQL"、"UDB/DB2"、または"DB2" (iSeries)
- ・ [Default User(デフォルト・ユーザー)]: "JDE"
- ・ [Default Password(デフォルト・パスワード)]: "JDE"
- ・ [Default Environment(デフォルト環境)]: デフォルト環境を入力します(DV9など)。
- ・ [System Datasource(システム・データ・ソース)]: "System - B9"

- ・ [Database User Role (データベース・ユーザー・ロール)]: “*ALL”

[JDBj-BOOTSTRAP DATA SOURCE]の値については、「Appendix A: jas.iniファイルのパラメータと値」の「[JDBj-BOOTSTRAP DATA SOURCE]」を参照してください。

14. [Next]をクリックします。
15. 上記の画面で[Database Type]として選択したデータベースに応じて、異なる画面が表示されます。[Database Type]で選択したタイプに対応して表示される画面で操作します。

Oracleの場合 [Oracle Database Settings (Oracleデータベース設定)]画面には、jdbj.iniファイルの[JDBj-BOOTSTRAP DATA SOURCE]セクションの重要な設定が表示されます。これらの設定は、OneWorld/ERPデータベース・サーバー上のOneWorld/ERPデータベースへの接続を定義します。詳しくは、[JDBj-BOOTSTRAP DATA SOURCE]を参照してください。これらの設定については、CNC管理者に問い合わせるか、OneWorld/ERPのインストール・ガイドを参照してください。



次のフィールドを確認し、必要に応じて変更します(ほとんどのフィールドは、デフォルトで正しい値が設定されます)。

- ・ [Object Owner (オブジェクト所有者)]: “SY9”
- ・ [OCM Server (OCMサーバー)]:
OCMテーブルが格納されているサーバーの名称。通常はエンタープライズ・サーバーまたはデータベース・サーバーです。このフィールドには、JDBj.iniファイルの[JDBj-BOOTSTRAP DATA SOURCE]セクションのServer=設定が自動入力されます。
- ・ [Database]: 正しい文字列を入力します。

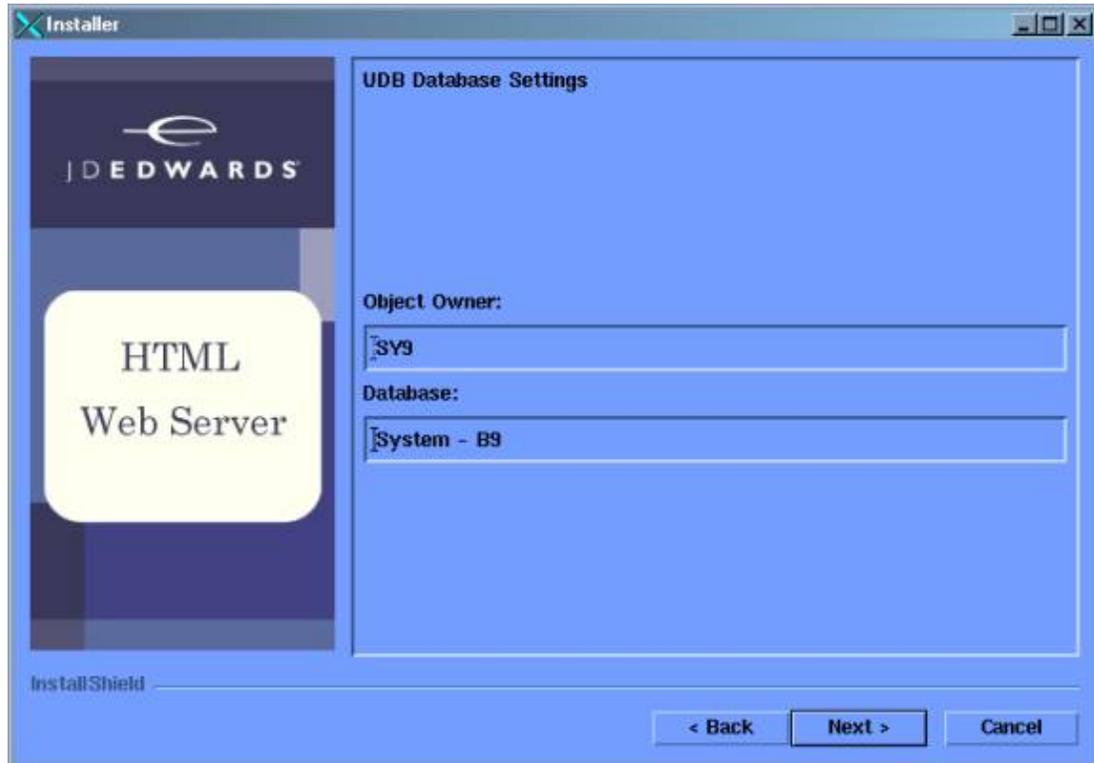
SQLの場合： [SQL Database Settings (SQLデータベース設定)] 画面には、jdbj.iniファイルの [JDBj-BOOTSTRAP DATA SOURCE] セクションの重要な設定が表示されます。これらの設定は、OneWorld/ERPデータベース・サーバー上のOneWorld/ERPデータベースへの接続を定義します。詳しくは、「[JDBj-BOOTSTRAP SESSION]」を参照してください。これらの設定については、CNC管理者に問い合わせるか、OneWorld/ERPのインストール・ガイドを参照してください。

次のフィールドを確認し、必要に応じて変更します（ほとんどのフィールドは、デフォルトで正しい値が設定されます）。

- ・ [OCM Server]:
OCMテーブルが格納されているサーバーの名称。通常はエンタープライズ・サーバーまたはデータベース・サーバーです。このフィールドには、JDBj.iniファイルの[JDBj-BOOTSTRAP DATA SOURCE]セクションのServer=設定が自動入力されます。
- ・ [OCM Server Port (OCMサーバー・ポート)]: OCMが実行されているSQLデータベースのポート。
- ・ [Database Name for System Tables (システム・テーブル用データベース名)]: OCMマッピングを含むライブラリの名前を入力します。通常は“SY9”です。
- ・ [Object Owner]: “SY9”

UDB/DB2の場合 [UDB Database Settings (UDBデータベース設定)] 画面には、jdbj.iniファイルの [JDBj-BOOTSTRAP DATA SOURCE] セクションの重要な設定が表示されます。これらの設定は、OneWorld/ERPデータベース・サーバー上のOneWorld/ERPデータベースへの接続を定義します。詳しくは、「[JDBj-BOOTSTRAP SESSION]」を参照してください。これらの設定については、CNC管理

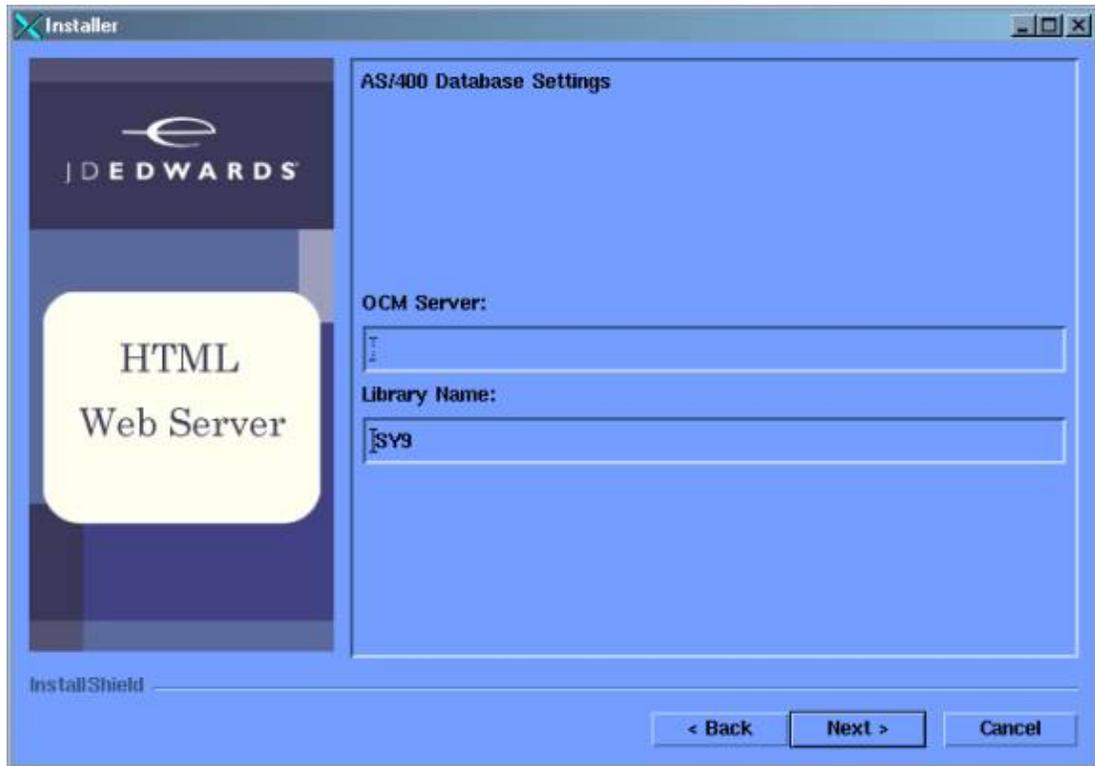
者に問い合わせるか、OneWorld/ERPのインストール・ガイドを参照してください。



次のフィールドを確認し、必要に応じて変更します(ほとんどのフィールドは、デフォルトで正しい値が設定されます)。

- ・ [Object Owner]: "SY9"
- ・ [Database]: 正しい文字列を入力します。

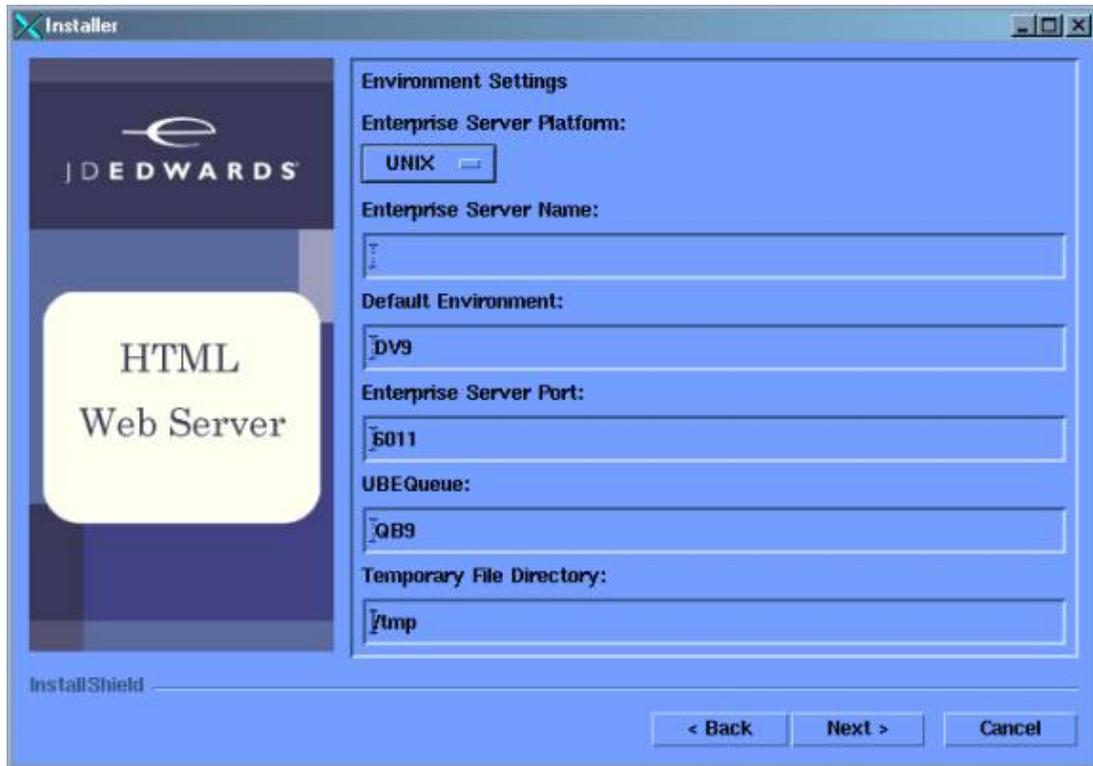
iSeriesの場合 [iSeries Database Settings(iSeriesデータベース設定)]画面には、jdbj.iniファイルの[JDBj-BOOTSTRAP DATA SOURCE]セクションの重要な設定が表示されます。これらの設定は、OneWorld/ERPデータベース・サーバー上のOneWorld/ERPデータベースへの接続を定義します。詳しくは、「[JDBj-BOOTSTRAP SESSION]」を参照してください。これらの設定については、CNC管理者に問い合わせるか、OneWorld/ERPのインストール・ガイドを参照してください。



次のフィールドを確認し、必要に応じて変更します(ほとんどのフィールドは、デフォルトで正しい値が設定されます)。

- ・ [OCM Server]:
OCMテーブルが格納されているサーバーの名称。通常はエンタープライズ・サーバーまたはデータベース・サーバーです。このフィールドには、JDBj.iniファイルの[JDBj-BOOTSTRAP DATA SOURCE]セクションのServer=設定が自動入力されます。
- ・ [Library Name(ライブラリ名)]: AS/400上で実行している場合は、OCMマッピングを含むライブラリの名称を入力します。通常、ライブラリ名はSY9です。

16. [Next]をクリックします。



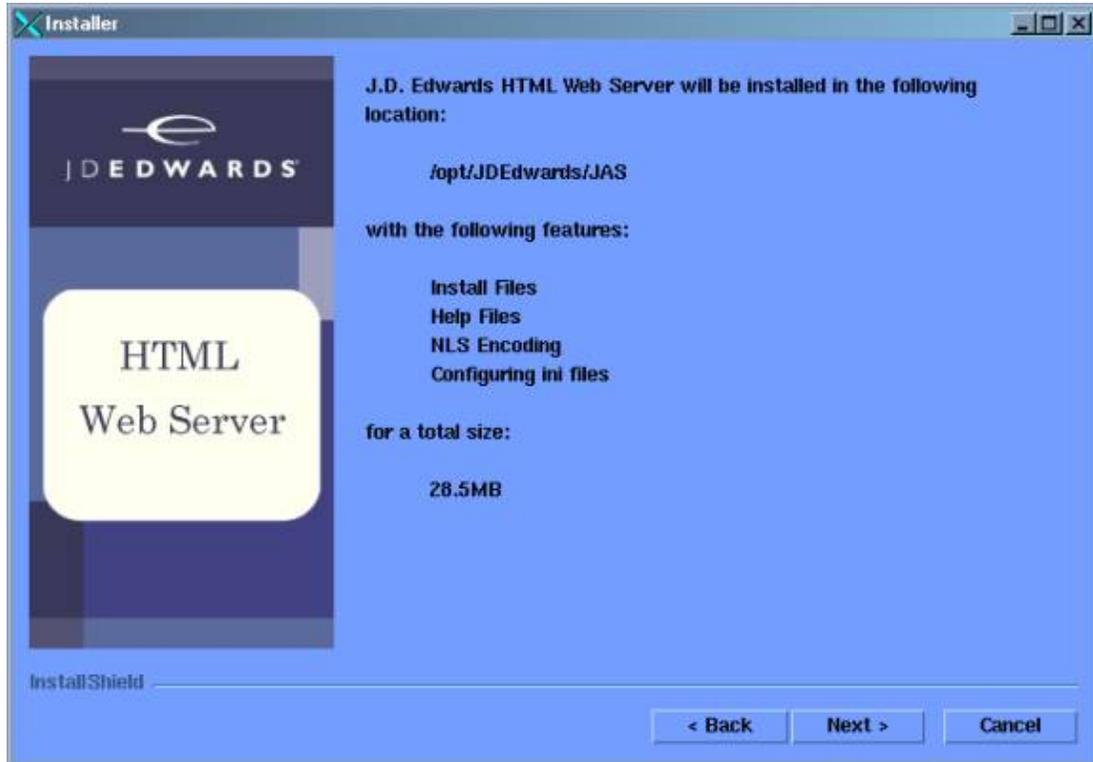
17. [Environment Settings(環境設定)]画面には、jas.iniファイルの各セクションの重要な設定が表示されます。これらの値は、JASサーバーの構成(使用するパス・コード、接続先となるエンタープライズ・サーバーなど)を定義します。

次のフィールドを確認し、必要に応じて変更します。

- ・ [Enterprise Server Platform(エンタープライズ・サーバーのプラットフォーム)]:エンタープライズ・サーバーのプラットフォームを選択します。
 - ・ Windows
 - ・ Unix
 - ・ iSeries
- ・ [Enterprise Server Name(エンタープライズ・サーバー名)]:エンタープライズ・サーバー名を入力します。
- ・ [Default Environment(デフォルト環境)]:ステップ[ステップ 13](#) [28]でデフォルト環境に使用したのと同じ値(DV9など)を使用します。
- ・ [Enterprise Server Port(エンタープライズ・サーバーのポート)]:“6011”
- ・ [UBE Queue(UBE待ち行列)]:“QB9”
- ・ [Temporary File Directory(一時ファイル・ディレクトリ)]:デフォルトでローカル・ドライブの標準一時フォルダに設定されます。

この値を、システム上で有効な一時ディレクトリに変更してください。

18. [Next]をクリックします。



19. 検証画面で情報を慎重にチェックし、入力値が正しいかどうかを確認します。

変更するには、変更する値が表示されるまで[Back(戻る)]をクリックします。

20. 情報の確認後に[Next]をクリックします。

J.D. Edwards Javaアプリケーション・サーバーのインストールが開始されます。このプロセスは、完了までに数分かかることがあります。

OneWorld/ERPデータベース接続(JDBCドライバ)の構成

次のいずれかのタスクを実行して、OneWorld/ERPデータベースをWebサーバーに接続します。

- ・ [DB2/UDBの構成 - ページ \[34\]](#)
- ・ [Oracleの構成 - ページ \[36\]](#)

DB2/UDBの構成

次のタスクを実行し、DB2/UDB用のJDBCドライバを入手して構成します。

JDBCドライバの入手

まだWebサーバーのJavaサーバーにDB2/UDBクライアントまたはサーバーをインストールしていない場合は、次のリンクを使用してインストール手順を参照してください。

<http://www-3.ibm.com/software/data/db2/>

DB2/UDB用のJDBCドライバは、DB2/UDBサーバーまたはクライアントのインストール内容に含まれています。DB2/UDBサーバーまたはクライアントのインストール後に、「WebSphereとWebサーバー用のDB2/UDB JDBCドライバの構成」の手順に従って、正しいバージョンのドライバを入手してください。

OneWorld/ERP用のJDBCドライバの構成

注意

WebSphere用のデータベース・リポジトリとしてDB2/UDBを使用している場合は、WebSphereが起動しなくなる恐れがあるため、このタスクを実行しないでください。

ファイルdb2java.zipを、<DB_HOME>/sqlib/java12から<WAS_HOME>/lib/appにコピーします。

<DB_HOME> はDB2/UDBの基本ディレクトリ、WAS_HOMEは<filesystem>/WebSphere/AppServerです。

db2profileの実行

db2profileを実行すると、WebSphereを起動するユーザー用に環境変数が設定され、DB2/UDBドライバの正しい構成が確保されます。

▶ db2profileを実行するには

1. Webサーバーのコマンド行から、WebSphereを起動するユーザーのプロファイルに次のコマンドを追加します。

```
./ / home/ instance/sqlib/db2profile
```

homeはDB2/UDBの位置、instanceはDB2/UDB 7.2データベース・インスタンス名です。

このコマンドは、上記のとおり正確に入力してください。(ピリオド)と/homeの間をスペース1個で区切っていることに注意してください。

2. ログアウトし、同じユーザーとして再度ログインします。

DB2/UDB 7.2用に環境変数を設定するスクリプトが実行されます。

3. スクリプトが正常に実行されたことを確認するために、次のコマンドを入力します。

```
echo $PATH
```

db2profileが正常に実行された場合は、リプライ・メッセージに次のDB2/UDBパスが含まれます。

```
/home/instance/sqlib/bin:/home/instance/sqlib/adm:/home/ instance/sqlib/misc
```

4. DB2/UDBデータベースがリモート・サーバー上にある場合は(ローカル・サーバーの場合は除く)、次のサブステップを実行して、データベースとその常駐ノードをカタログ化します。

注意

このステップは、リモート・データベースごとに1度実行するだけですみます。

- a. 次のコマンドを入力し、サーバーからの該当する値を使用してノードをカタログ化します。

```
> db2 catalog tcpip node nodename remote nodename server dbport
```

- b. 次のコマンドを入力し、該当するデータベース名とノードの値を指定してデータベースをカタログ化します。

```
> DB2 catalog database dbname at node nodename
```

カタログ化については、『OneWorld/ERPインストール・ガイド』の「クライアント上のDB2-UDBデータベースのカタログ化」(スクリプトを使用してデータベースをカタログ化するには)を参照してください。

DB2/UDBからWebSphereへの接続

DB2/UDBデータベースをWebSphere™に接続するために、次のタスクを実行します。

▶ DB2/UDBをWebSphereに接続するには

- ・ libdb2jdbc. xxファイルを、/ db2home/libディレクトリからWebSphere/AppServer/binディレクトリにコピーします。

db2homeはDB2/UDBのホーム・ディレクトリ(/opt/IBMdb2など)で、xxはAIXとSUNの場合はso、HP-UXの場合はslです。

Oracleの構成

次のタスクを実行し、Oracle用のJDBCドライバを入手して構成します。

JDBCドライバの入手

Oracle用のJDBCドライバは、Oracleクライアントまたはサーバーのインストール時にシステムに自動的にダウンロードされます。classes12.zipファイルがOracleディレクトリ内の/jdbc/lib/ディレクトリにあることを確認してください。このファイルが存在しない場合は、OracleのWebサイトから前述のディレクトリにダウンロードしてください。

OneWorld/ERP用のJDBCドライバの構成

注意

WebSphere用のデータベース・リポジトリとしてOracleを使用する場合は、WebSphereが起動しなくなるため、このタスクを実行しないでください。

ファイルclasses12.zipとnls_charset12.zipを、〈ORACLE_HOME〉/jdbc/libから〈WAS_HOME〉/lib/appにコピーします。

〈ORACLE_HOME〉はORACLEデータベースのホーム・ディレクトリ、WAS_HOMEは/〈filesystem〉/WebSphere/AppServerです。

tnsnames.oraファイルの構成

Webサーバー上でOracleデータベースを使用する場合は、tnsnames.oraファイルを構成する必要があります。このファイルでは、OracleデータベースとJavaサーバー間の接続を定義します。構成でOracleデータベースを使用する場合は、OneWorld/ERPクライアント・インストールからJavaサーバー上のディレクトリにtnsnames.oraファイルをコピーできます。このファイルまたはJavaサーバー上の既存のtnsnames.oraファイルへのパスを、jas.iniファイルの構成時に定義します。

注意

jas.iniファイルでは、tnsnames.oraのデフォルト・ディレクトリは〈JDE_HOME〉/EA_JDEdwards_1.ear/webclient.war/WEB-INFです。このファイルを複数のアプリケーション間で共有する場合は、jas.iniファイルでtnsnames.oraの位置を編集できます。

tnsnames.oraファイルを、〈JDE_HOME〉/EA_JDEdwards_1.ear/webclient.warから〈JDE_HOME〉/EA_JDEdwards_1.ear/webclient.war/WEB-INFディレクトリに移動します。ファイルを移動すると、HTTPサーバーを介して公開されるパスから外れます。これにより、このファイルにエンドユーザーがアクセスするのを防止できます。

OneWorld/ERPインストール・プランへのWebサーバーの追加

WebサーバーでOneWorld/ERPと通信するには、JavaサーバーをOneWorld/ERPインストール・プランに追加し、OneWorld/ERP環境の一部として定義する必要があります。次の手順に従って、OneWorld/ERPのインストール・ガイドまたはアップグレード・ガイドの2つの章の作業を完了してください。

► OneWorld/ERPインストール・プランにJavaサーバーを追加するには

デプロイメント・サーバー上で、次の手順で操作します。

1. OneWorld/ERPインストール(アップグレード)・ガイドの「既存構成へのサーバーの追加」の章に記載されているタスクを開始します。

OneWorld/ERP™のプランナ環境にユーザーJDEとしてサインオンし、〈システム・インストール・ツール〉メニュー(GH961)で[アドオン・サーバー・プラン]をダブルクリックします。OneWorld/ERPのインストール/アップグレード・ガイドに記載されている指示に従ってプランを定義します。

2. 〈アドオン・サーバー・プラン - [サーバーの追加]〉画面で、[JASサーバー]オプションを選択して[OK]をクリックします。
3. [OK]をクリックします。
4. 〈アドオン・サーバー・プラン - [JASサーバーの改訂]〉で、次のフィールドに値を入力します。

・ [マシン名]:

- JASサーバー名
- ・ 説明
 - JASサーバーの記述
 - ・ [リリースNo.]: “B9”
 - ・ [ホスト・タイプ]:
50
 - ・ [基本ユーザー]:
“JDE”
 - ・ [インストール・パス]:
JASサーバー上の¥jasディレクトリへのパス。たとえば、x:¥Program Files¥JDEdwards¥JASです。
。
- 5. [OK]をクリックします。
- 6. 〈アドオン・サーバー・プラン - [環境の選択]〉画面が表示されるまで、各画面に値を入力します。
- 7. 〈アドオン・サーバー・プラン - [環境の選択]〉画面で、J環境を選択します(たとえば、JTS9 xまたはJPY9を使用します)。
- 8. デフォルトのデータ・ソースを受け入れます。
- 9. プランに追加する他のすべてのJ環境について、上記のステップを繰り返します。
- 10. 「インストール・プランの確定」の章の次の手順を完了します。
- 11. 〈インストール計画の処理〉で[検索]をクリックし、新規プランを展開してエンタープライズ・サーバーをクリックします。
- 12. [追加]をクリックします。
- 13. 〈Enterprise Server(エンタープライズ・サーバー)〉で、[選択]をクリックします。
- 14. 〈アドオン・サーバー・プラン - [マシンの検索]〉で、Javaオブジェクトを格納するエンタープライズ・サーバーを選択して[選択]をクリックします。
- 15. 〈アドオン・サーバー・プラン - [エンタープライズ・サーバーの改訂]〉で、[フォーム]メニューの環境をクリックします。
- 16. 〈アドオン・サーバー・プラン - [マシン環境の改訂]〉で、空のローの[環境]フィールドをクリックし、ビジュアル・アシスト・ボタンをクリックします。
- 17. 〈アドオン・サーバー・プラン - [環境の検索]〉で、環境を選択して[選択]をクリックします。
- 18. インストールしたすべての環境について、上記のステップを繰り返します。

19. 〈アドオン・サーバー・プラン - [マシン環境の改訂]〉で[OK]をクリックします。
20. 〈アドオン・サーバー・プラン - [エンタープライズ・サーバーの改訂]〉で[OK]をクリックします。
21. デフォルトのデータ・ソースを受け入れて、残りの画面に情報を入力します。
プランへの追加が確定されます。
22. OneWorld/ERPインストール(またはアップグレード)・ガイドの「インストール・ワークベンチ」セクションの操作を実行し、OneWorld/ERPでJavaサーバーを構成します。

JASをサポートするためのIBM HTTPサーバーの構成

次のタスクを実行して、JASをサポートするようにHTTPサーバーを構成します。

▶ JASをサポートするようにIBM HTTPサーバーを構成するには

1. Webサーバーで複数のポートをリスニングできるように、IBM HTTPサーバー構成を修正します。
通常、ディレクトリ・パスは次のとおりです。

- ・ AIXの場合

```
/usr/HTTPServer/conf/httpd.conf
```

- ・ Solarisの場合

```
/opt/IBMHTTPD/conf/httpd.conf
```

- ・ HP-UXの場合

```
/opt/HTTPServer/conf/httpd.conf
```

IBM HTTPサーバー構成ファイルで、Listenディレクティブを追加し、対応するVirtualHostディレクティブを指定します。VirtualHostディレクティブを使用すると、第2のインスタンスを含む代替Aliasディレクティブを指定できます。

WebSphere 4.0 Advanced Edition Single Serverの場合：

JavaサーバーでデフォルトのIBM HTTPサーバー・ポート(80)をリスニングする場合は、httpd.confファイルの最後に次のステートメントを追加します。

```
Alias /jde "/u01/jas/EA_JDEdwards_1/webclient.war"
```

WebSphere 4.0 Advanced Editionの場合：

```
Listen 80 Listen 81 NameVirtualHost myserver:81
```

```
<VirtualHost myserver:81>
```

```
Alias /jde /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/
```

```
</VirtualHost>

<VirtualHost myserver.jdedwards.com:81> Alias /jde /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/

</VirtualHost>
```

ヒント

- Alias /jdeを使用すると、HTTPサーバーでJASディレクトリにあるスタティック・ファイルを処理できます。JASでリスニングするすべてのポート番号に対して、Alias /jdeが宣言されていることを確認してください。Alias /jdeは、owhtmlディレクトリを含むディレクトリにマップする必要があります。JASインストール・プログラムによりJASディレクトリ構造が作成された後に、Aliasを宣言すると便利です。

2. httpd.confに次のディレクティブを追加します。

WEB-INFディレクトリに異なるパスを使用した場合は、次のディレクティブを正しいパスで置き換えてください。このディレクティブにより、jas.iniファイルや他の重要な情報を格納するWEB-INFディレクトリがプロテクトされます。

```
<Directory "/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/WEB-INF">

Order Deny,Allow Deny from All </Directory>
```

注意

WebSphere 4.0 Advanced Edition Single Serverを実行している場合は、“EA_JDEdwards_1.ear”を“EA_JDEdwards_1”に置き換えてください。

3. IBM HTTPサーバーを再起動して変更内容を有効にします。

スタティック・ファイルの処理の構成

デフォルトでは、Webサーバー用のスタティック・ファイルはすべてHTTPサーバーで処理されます。このため、HTTPサーバーからすべてのスタティック・ファイルにアクセスできる必要があります。この設定を構成するには、IBM HTTPサーバーのhttpd.confファイルのAliasディレクティブを使用するか、またはVirtual Directory for IISを使用します。この構成は、ほとんどの場合に適合します。これにより通信量が減少し、水平複製処理についてはHTTPサーバーとWebSphere™間のネットワーク・トラフィックが減少します。また、一般にHTTPサーバーの場合、スタティック・ファイルの処理効率が向上します。

2番目のオプションは、WebSphereファイル処理サブルーチンを有効化してスタティック・ファイルを処理することです。Webサーバーのスタティック・ファイルをWebサーバー・マシンに格納できず、HTTPサーバーからネットワーク経由でスタティック・ファイルにアクセスしたくない場合は、この方法が必要になることがあります。

次のどちらかのタスクを実行して、スタティック・ファイルを構成します。

・ [手順 6](#) [41]

このタスクは、WebSphereファイル処理サーブレットを無効化し、すべてのスタティック・ファイルをHTTPサーバーで処理できるようにします。

・ [手順 7](#) [41]

このタスクは、WebSphereファイル処理サーブレットを有効化し、HTTPサーバーに代わってスタティック・ファイルを処理できるようにします。

▶ **ファイル・サーブレットを無効化するには**

1. テキスト・エディタでibm-web-ext.xmiファイルを開きます。

このファイルは次のディレクトリにあります。

/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war¥WEB-INF

2. fileServingEnabledプロパティをfileServingEnabled="false"に変更して、ファイル・サーブレット・オプションを無効にします。
- 3.
4. HTTPサーバー上でプラグインを再生成し、HTTPサーバーを再起動します。

▶ **ファイル・サーブレットを有効化するには**

1. テキスト・エディタでibm-web-ext.xmiファイルを開きます。

このファイルは次のディレクトリにあります。

/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war¥WEB-INF

2. fileServingEnabledプロパティをfileServingEnabled="true"に変更して、ファイル・サーブレット・オプションを有効にします。
3. デフォルト・ページindex.htmlをセットアップするには、webclient.war¥WEB-INFディレクトリにあるweb.xmlファイルを開き、最終行(</web-app> タグを含む行)の真上に次のセクションを追加します。

```
<welcome-file-list> <welcome-file>index.html</welcome-file> </welcome-file-list>
```

注意

これらのタグを追加するのは、ファイルに存在しない場合のみです。

4. HTTPサーバー上でプラグインを再生成し、HTTPサーバーを再起動します。

JASログの構成

B9以降のJASでは、新規のログAPIを使用してJASプロセスがモニタリングされ、情報がログ・ファイルに書き込まれます。この新規ログ・プロセスでは、jdelog.propertiesファイルを使用してログ・ファイルの位置が指定され、ログ・パラメータが設定されます。新規ログ・プロセスを有効にするには、jdelog.propertiesを含むディレクトリをclasspathに指定します。

注意

このディレクトリは、J.D. Edwards ERPソフトウェアのインストール時にclasspathに自動的に指定されます。

次にサンプルjdelog.propertiesファイルを示します。

サンプルjdelog.propertiesファイル

```
# This is a sample configuration file for root configuration

jdelog.rootLogger=DEBUG,JDELOG,JASLOG
jdelog.loggerFactory=com.jdedwards.base.logging.log4j.JdeLoggerFactory
jdelog.reloadInterval=60❶

# These are the settings for file handlers.
jdelog.handler.JDELOG=com.jdedwards.base.logging.log4j.FileHandler❷
jdelog.handler.JDELOG.File=¥¥jderoot.log❸
jdelog.handler.JDELOG.Level=ERROR❹
jdelog.handler.JDELOG.Append=TRUE❺
jdelog.handler.JDELOG.MaxBackupIndex=1❻
jdelog.handler.JDELOG.MaxFileSize=10MB❼
jdelog.handler.JDELOG.format=com.jdedwards.base.logging.log4j.DefaultFormat❽

# These are the settings for file handlers.
jdelog.handler.JASLOG=com.jdedwards.base.logging.log4j.FileHandler
jdelog.handler.JASLOG.File=¥¥jas.log
jdelog.handler.JASLOG.Level=ERROR
jdelog.handler.JASLOG.Append=TRUE
jdelog.handler.JASLOG.MaxBackupIndex=1
jdelog.handler.JASLOG.MaxFileSize=10MB
jdelog.handler.JASLOG.format=com.jdedwards.base.logging.log4j.DefaultFormat

#This is added for debug logging.Please uncomment following to start debug.
#jdelog.Debug=DEBUG, jasdebug
#jdelog.handler.jasdebug=com.jdedwards.base.logging.log4j.FileHandler
#jdelog.handler.jasdebug.File=¥¥jasdebug.log
#jdelog.handler.jasdebug.Level=DEBUG
```

上記のサンプルは、3つのログ・ファイル、jde.log、jas.log、およびjasdebug.logのパラメータ設定を示します。次のリストに、jde.logファイルの各パラメータを示します。JDELOGはハンドラ名です。この名前はどのような名前でもかまいませんが、ロガー、ルート、またはユーザーに与えられたハンドラ名と一致する必要があります。

この説明を参考にして各ログ・ファイルのパラメータを修正してください。

❶ Webサーバーによりjdelog.propertiesファイルが再ロードされる間隔を定義します。

jdelog.propertiesファイルの修正後にシステムで構成設定が更新される際の所要時間を決定するグローバル・パラメータです。デフォルト値は60(秒)です。この値を修正して異なる間隔を指定できません。

ヒント

値を0(ゼロ)に変更すると、更新後のパラメータは再ロードされません。

- ❷ ルートのファイル・ハンドラ設定を定義します。
- ❸ ファイルの位置と名前を定義します。

このパスを修正してログ・ファイルのディレクトリを指定する必要があります。現行のWebSphere基本ディレクトリへの相対パスを指定してください(/WebSphere/AppServerなど)。

注意

パスの入力時には、ファイル・パスの指定に2つのバックスラッシュ¥¥を使用します。または、代わりに1つのスラッシュ/を使用してもかまいません。ファイル・パスの指定には1つのみのバックスラッシュ¥¥を使用しないでください。

- ❹ ハンドラのログ・レベルを定義します。
- ❺ アプリケーションの再起動時に、新規ログ情報を既存のファイルに追加するかどうかを定義します。
- ❻ ログ・ファイルのサイズが次に指定するMaxFileSizeよりも大きくなった場合に、システムで作成されるバックアップ・ファイルの数を定義します。
- ❼ ログ・ファイルの最大サイズを定義します。
- ❽ メッセージ・ログのフォーマットを定義します。

2) このコンソール・ハンドラ構成を次に示します。

注意

jdelog.propertiesファイルの最後にステートメントを追加すると、ログの出力先をWebSphereエラー・ログ・ファイル(stderr.txt)または[WebSphere Administrative Console]の最下部にあるコンソールに変更できます(この場合も、出力はstd.outファイルに送られます)。

- ・ 出力先をWebSphereコンソールにするには、jdelog.propertiesファイルに次のステートメントを追加します。
- ・ 出力先をWebSphereのstderr.txtファイルに変更するには、太字で示したSystem.outのテキストをSystem.errに置き換えます。

```
#These are the settings for console handler (or System.err)
jdelog.handler.JDELOG=com.jdedwards.base.logging.log4j.ConsoleHandler
#This specifies where your output goes.
jdelog.handler.JDELOG.Out=System.out
#This specifies the level for the handler.
jdelog.handler.JDELOG.Level=ERROR
```


OneWorld/ERPのシリアル化オブジェクトの生成

Webサーバーを実行するには、サーバーにOneWorld/ERP™の一連のシリアル化JASオブジェクトへのアクセス権が必要です。これらのオブジェクトは、該当するOneWorld/ERP™スペック・セットを使用してOneWorld/ERP™オブジェクトから直接生成する方法と、OneWorld/ERP™のインストール時にテーブル(F989998とF989999)に自動的にロードされる事前生成済みの一連のプリスティン(JDEオリジナル)オブジェクトから派生させる方法があります。

OneWorld/ERP™のシリアル化オブジェクトを生成するには、特定のマシン構成が必要です。OneWorld/ERP™のシリアル化オブジェクトを生成するようにWebサーバーを構成することもできますが、別のジェネレート・マシンをこの処理専用を使用することをお勧めします。ジェネレート・マシンに関する全要件のリストについては、Knowledge Gardenの次のリンクを参照してください。

[Minimum Technical Requirements \(最低限の技術要件\)](#)

[<https://knowledge.jdedwards.com/JDEContent/TechMarketingCU/information/bjtechreqs.htm>]

旧リリースからOneWorld/ERP™にアップグレードし、OneWorld/ERP™オブジェクトをカスタマイズした場合は、最初カスタム修正をテストした後、アップグレード済みのパス・コードからシリアル化JASオブジェクトを生成する必要があります。

ヒント

JASがOneWorld/ERP™ Windowsクライアントと同じスペックで動作するようにすべてのオブジェクトを生成することをお勧めします。すべてのプラットフォーム用に事前生成済みのオブジェクトが用意されていますが、オブジェクトを正常に生成できるかどうか、Webサーバー導入の成否を左右します。JASオブジェクトを生成するのは、OneWorld/ERP™クライアント用パッケージをビルドするのと同じことなので、導入作業を完全かつ順調に進めるには、オブジェクトを生成する必要があります。

次のタスクを実行してeGeneratorをインストールし、一連のOneWorld/ERP™オブジェクトからOneWorld/ERP™ JASオブジェクトを生成します。

Javaサーバー用のシリアル化オブジェクトを生成するには、開発者はeGeneratorをインストールして使用する必要があります。eGeneratorは、OneWorld/ERP™アプリケーションをJava、HTMLまたはその両方で生成できるように、OneWorld/ERP™のスペックをJavaコードに変換するJ.D. Edwardsツールです。このツールを使用して生成するOneWorld/ERP™のフォームとアプリケーションは、JavaまたはHTMLオブジェクトとなります。OneWorld/ERP™では、オブジェクトはデータベースに格納され、実行時に取り込まれます。

次の手順を実行し、eGeneratorをインストールして実行します。

- ・ [eGeneratorのインストール - ページ](#) [46]
- ・ [ログイン - ページ](#) [50]
- ・ [すべての標準シリアル化オブジェクトの生成 - ページ](#) [61]
- ・ [生成プロセスの検証 - ページ](#) [66]
- ・ [選択したオブジェクトの生成 - ページ](#) [68]

はじめる前に

Webサーバーと同じサービス・パック・レベルで動作するOneWorld/ERPクライアントをインストールし、実行しておきます。このタスクでは、OneWorld/ERPクライアントをジェネレート・マシンとして構成します。OneWorld/ERPクライアントのインストール手順については、OneWorld/ERPのインストール・ガイド(Windows NTベース・システム)を参照してください。

eGeneratorのインストール

ここでは次の内容について説明します。

- ・ [デフォルトの記憶域パラメータの設定 - ページ \[46\]](#)
- ・ [シリアル化オブジェクト生成のためのWin32クライアントのセットアップ - ページ \[47\]](#)
- ・ [ジェネレート・マシンへのJASビルドのコピー - ページ \[47\]](#)
- ・ [シリアル化オブジェクトのためのOneWorld/ERPの準備 - ページ \[48\]](#)
- ・ [gen.batの修正 - ページ \[49\]](#)

はじめる前に

- ・ SP20の場合は、SAR 5897371に含まれるESU JD14797をインストールする必要があります。このESUにより、WBOIDとWBLNGPREFに基づいてシリアル化オブジェクト・テーブル(F989999)に新規のインデックスが作成されます。
- ・ ジェネレート・マシン上のgen.batファイルが示すパスが、構成に有効であることを確認します。必要な場合は、ジェネレート・マシン上の正しいローカルjarファイルを指すように変更してください。
- ・ エンタープライズ・サーバー上でDB2 UDBを使用している場合は、次のタスクを実行します。ジェネレート・マシン上でdb2cli.iniファイルを開き、OneWorld/ERPに関連するデータベース・エイリアスの下のLobCachSize/パラメータをコメント化するか削除します。

通常、このファイルはDB_HOME¥SQLLIB¥にあります。

デフォルトの記憶域パラメータの設定

OneWorld/ERP™データベース上のデフォルトの記憶域パラメータでは、空き領域不足のために標準的なOneWorld/ERP™ Javaオブジェクトをすべて転送できない場合があります。この問題を回避するために、次のデータベース関連タスクを実行します。

- ・ デフォルトの記憶域パラメータの設定 (Oracleのみ)
 - ・ デフォルトの記憶域パラメータの設定 (SQLのみ)
-
- ▶ デフォルトの記憶域パラメータを設定するには(Oracleのみ)

- ・ Oracleデータベースにログインし、次のコマンドを入力します。

Alter tablespace xx9T default storage (maxextents unlimited);

xxは環境、9はJavaオブジェクトをインストールするJ.D. Edwards ERPソフトウェアのリリースです(たとえば、B9プロトタイプ環境の場合はPY9Tを使用します)。

▶ デフォルトの記憶域パラメータを設定するには(SQLのみ)

- SQLデータベースを開き、データベースのMaximum File Sizeパラメータを“Unrestricted filegrowth”に設定します。

このステップは、Javaオブジェクトをインストールする環境(PY9など)ごとに実行してください。

シリアル化オブジェクト生成のためのWin32クライアントのセットアップ

OneWorld/ERP™クライアントのインストール後に、このマシンを使用してシリアル化オブジェクトを生成できます。

▶ シリアル化オブジェクトを生成するようにOneWorld/ERPクライアントをセットアップするには

- Windows NTエクスプローラで、jde.iniファイルが格納されているディレクトリ(通常はc:\winnt)を検索します。
- jde.iniファイルの[Interactive Runtime]セクションで、次のパラメータと値を追加します。

WebAdmin=1

このパラメータの値を“1”に設定すると、eGeneratorで8つのタブを表示して使用できるようになります。これにより、デフォルト・ユーザー用にすべてのシリアル化オブジェクトを生成できます。

- [Save(保存)]をクリックします。

ジェネレート・マシンへのJASビルドのコピー

OneWorld/ERPクライアントのセットアップ後に、シリアル化オブジェクトの生成に使用するJASビルドを、このマシンにコピーします。ほとんどの場合、JASビルドのSystemディレクトリには、OneWorld/ERPの正常動作に必要なファイルがすべて含まれています。

▶ ジェネレート・マシンにJASビルドをコピーするには

- ジェネレート・マシン上で、¥B9フォルダの下に次の新規フォルダを作成します。

¥B9¥Generator

- Webサーバー上で、WebSphere™ドキュメントのルート・フォルダを検索します。

<filesystem>/JDEdwards/JAS/EA_JDEdwards_1.ear/webclient.war

- WebSphere™ドキュメントのルート・フォルダとサブディレクトリにある全ファイルを、Webサーバーか

らジェネレート・マシン上の¥Generatorディレクトリにコピーします。

4. Webサーバー上で次のファイルを検索します。

genapp.ini

このファイルは、gen.batと同じディレクトリ(通常は次のディレクトリ)にあります。

<filesystem>/JDEdwards/JAS/EA_JDEdwards_1.ear/webclient.war

Webサーバーからジェネレート・マシン上の次のディレクトリに、genapp.iniファイルをコピーします。

C:¥WINNT

シリアル化オブジェクトのためのOneWorld/ERPの準備

OneWorld/ERPでシリアル化Javaオブジェクトを生成する前に、次のタスクを実行してJavaジェネレート・マシンにデータ辞書とグローバル・スペックを自動入力します。

- ・ データ辞書の自動入力
- ・ グローバル・スペックの自動入力

▶ データ辞書を自動入力するには

1. Javaジェネレート・マシン上で、OneWorld/ERP™の有効なローカル環境にログオンします。
2. [略式コマンド]に“BV”と入力して[Enter]キーを押します。
3. <バッチ・アプリケーション - [バッチ・バージョンの処理 - 使用可能なバージョン]>で、[バッチ・プログラムID]フィールドに“R92TAM”と入力します。
4. [検索]をクリックします。
5. バージョンXJDE001を選択し、[選択]をクリックします。
6. <バッチ・バージョン - [バージョン・プロンプト]>で、[フォーム]メニューの[上級]をクリックします。
7. <バッチ・バージョン - [上級バージョン・プロンプト]>で、[ロケーション一時変更]をクリックします。
8. [OK]をクリックします。
9. <バッチ・バージョン - [バージョン・プロンプト]>で、[投入]をクリックします。
10. <バッチ・バージョン - [JDEデータ・ソース]>で、[LOCAL(ローカル)]を選択して[選択]をクリックします。

アプリケーションにより、x:¥B9ディレクトリに格納されているデータベースへの自動入力UBEが起動されます。この処理には1時間程度かかります。

11. 処理の完了後に、次のファイルをx:¥B9ディレクトリからJavaシリアル化オブジェクトの生成に使用する環境のパス・コードの下の¥specディレクトリ(x:¥b9¥PY9¥specなど)にコピーします。

- ・ dddict.xdb
- ・ ddttext.xdb
- ・ dddict.ddb
- ・ ddttext.ddb

▶ グローバル・スペックを自動入力するには

1. [略式コマンド]に“BV”と入力して[Enter]キーを押します。
2. 〈バッチ・アプリケーション - [バッチ・バージョンの処理 - 使用可能なバージョン]〉で、[バッチ・プログラムID]フィールドに“R98CRTGL”と入力します。
3. [検索]をクリックします。
4. バージョンXJDE001を選択し、[選択]をクリックします。
5. 〈バッチ・バージョン - [バージョン・プロンプト]〉で、[フォーム]メニューの[上級]をクリックします。
6. 〈バッチ・バージョン - [上級バージョン・プロンプト]〉で、[ロケーション一時変更]をクリックします。
7. [OK]をクリックします。
8. 〈バッチ・バージョン - [バージョン・プロンプト]〉で、[投入]をクリックします。
9. 〈バッチ・バージョン - [JDEデータ・ソース]〉で、[LOCAL(ローカル)]を選択して[選択]をクリックします。

アプリケーションによりUBEが起動されます。完了までに1時間程度かかります。

gen.batの修正

eGeneratorには、標準的な構成設定のリストを含むbatファイル(gen.bat)が用意されています。これらの設定は、環境に合わせて修正する必要があります。

▶ gen.batを修正するには

1. Windowsエクスプローラから、ジェネレート・マシンの¥B9¥Generatorディレクトリに移動してgen.batをコピーします。
2. gen.batを右クリックして[Edit(編集)]を選択します。

注意

Webサーバーの旧バージョンがある場合は、gen.batファイルを旧バージョンで置き換えないでください。旧バージョンのファイルは、最初は正常に実行されますが、時間が経過すると異常終了します。

3. gen.bat構成ファイルの次の設定を変更します。

パラメータ	設定
set JDK 1.3=	<p>WebSphereで使用するJDKアプリケーションのディレクトリへのパス。通常、このパスはx:¥WebSphere¥AppServer¥javaです。</p> <hr/> <p>注意</p> <p>このパスはx:¥WebSphere¥AppServer¥JDK(WebSphere 3.5)から変更されています。</p> <hr/>
set JAS INI=	<p>jas.iniファイルのディレクトリへのパス(直接生成の場合にのみ必須)。通常、このパスは次のとおりです。</p> <p>x:¥B9¥Generator¥WEB-INF</p>
set DBDRIVERS=	<p>データベースのJDBCドライバのパスとファイル名</p> <p>(直接生成の場合にのみ必須)。</p> <p>特定のデータベース用のJDBCドライバへのパスを入力します。</p> <p>Oracleの場合: "classes12.zip"</p> <p>SQL 2000の場合: msbase.jar、mssqlserver.jar、msutil.jar</p> <p>DB2/UDBの場合: db2java.zip</p> <p>DB2(iSeries)の場合: jt400.jar</p>

注意

jas.iniファイルまたはデータベース・ドライバへのパス名にスペースが含まれている場合は、パス全体を二重引用符で囲む必要があります。たとえば、Windowsの場合は、JAS INI="c:¥program files¥JDEdwards¥JAS¥EA_JDEdwards_1.ear¥webclient.war¥WEB-INF"に設定します。

4. gen.batファイルの次の行を検索します。

```
%JDK 1.3%¥bin¥java -ms128m -512m...
```

この行を次のように変更します。

```
%JDK 1.3%¥bin¥java -classic -ms128m -512m...
```

ログイン

eGeneratorを使用すると、次のどちらかの方法でOneWorld/ERPオブジェクトを生成できます。

- ・ Webサーバーに生成する方法

- ・ Webサーバーをバイパスしてデータベースに直接生成する方法

この2つの生成方法では、eGeneratorへのログイン方法が異なります。オブジェクトの生成方法に対応するタスクを選択してください。

はじめる前に

次のファイルとフォルダがワークステーション上にあることを確認します。

- ・ gen.bat
- ・ OWResource.jar
- ・ xalan.jar
- ・ xerces.jar
- ・ images.jar
- ・ classesフォルダ

データベースに直接生成する場合は、さらに次のファイルも必要です。

- ・ jas.ini
- ・ データベース・ドライバ
- ・ tnsnames.ora

jas.iniの設定

jas.iniファイルの次のセクションが正しく構成されていることを確認します。必要な場合は、構成に合わせて変更します。

- ・ [LOGS]
ログ・ファイルへのパスが有効なディレクトリを指していることを確認します。
- ・ [JDBC URL]
DEFAULT= この設定が、シリアル化オブジェクトが生成される正しいデータベースを指していることを確認します。
- ・ [OWWEB]
OracleTNS= Oracleデータベースを使用している場合は、この設定がtnsnames.oraファイルのディレクトリを指していることを確認します。

jde.iniの設定

jde.iniファイルの[Interactive Runtime]セクションで、次のパラメータが設定されていることを確認します。

注意

jde.iniファイルの変更内容は、ジェネレート・マシンにフル・パッケージをデプロイするときに上書きされます。これらの設定を保持するために、jde.iniファイルをバックアップ・ディレクトリにコピーできます。新規パッケージをデプロイした後、Java生成用にカスタマイズした設定（下記を参照）を、バックアップ・ファイルの[Interactive Runtime]セクションから更新済みのjde.iniファイルにコピーできます。パッケージ・ビルド後は、更新済みjde.iniファイル全体をバックアップ・ファイルで上書きしないでください。全体を上書きすると、新規パッケージに必要な更新済みの設定が削除される可能性があります。

- WebAdmin

このパラメータを、eGeneratorの管理者権限を表す"1"に設定します。管理者権限がない場合は、一部のアクションとオプションを使用できません。

Webサーバーに生成する方法

オブジェクトをWebサーバーに生成する場合は、このタスクを実行します。Webサーバーをバイパスしてオブジェクトをデータベースに直接生成する場合は、このタスクをスキップして次のタスクを実行してください。

▶ Webサーバーに生成するには

ジェネレート・マシン上で、

1. ジェネレート・マシン上の¥B9¥Generatorディレクトリの下でgen.batを実行します。このバッチ・ファイルにより初期化ファイルgenapp.iniが呼び出されます。このファイルを使用して、シリアル化オブジェクト生成用に特定のアプリケーションを構成できます。

ヒント

ジェネレート・マシン上のWINNTディレクトリに次の初期化ファイルがあることを確認します。

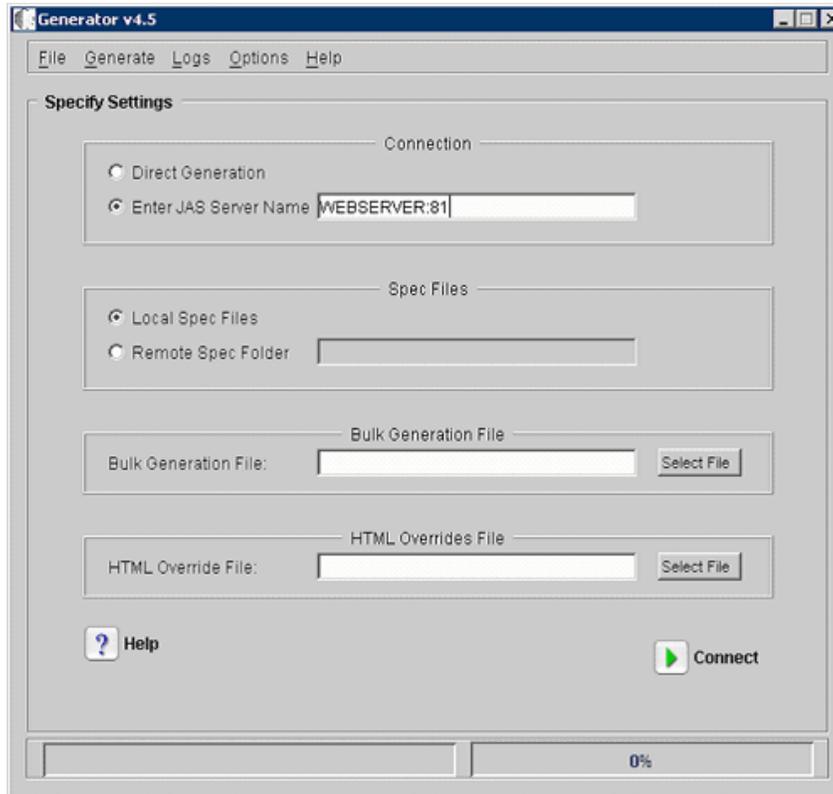
genapp.ini

サンプルのgenapp.iniファイルを次に示します。

2. 〈OneWorld/ERP Sign On (OneWorld/ERPのサインオン画面)〉で、OneWorld/ERPのJava環境（JPY9など）にログオンします。

ヒント

OneWorld/ERP™環境にログオンするために[OK]をクリックしてからログオン処理が完了するまで最大で2分かかる場合があります。



3. eGeneratorで、次のフィールドに値を入力します。

- ・ JAS Server Name (JASサーバー名)

ログインするJASサーバーの名称を入力します。

システムによりデータベース・パスがjas.iniファイルから取得されます。eGeneratorにより、このWebサーバーに全オブジェクトが生成されます。

注意

ジェネレート・マシン上で複数のWebサーバー・ポートが動作している場合は、machine name:portを入力します。machine name はジェネレート・マシン名、portはポート番号です。

4. [OK]をクリックします。

指定したWebサーバーにシステムが接続します。ログインが完了すると、ステータス・バーにメッセージが表示されます。

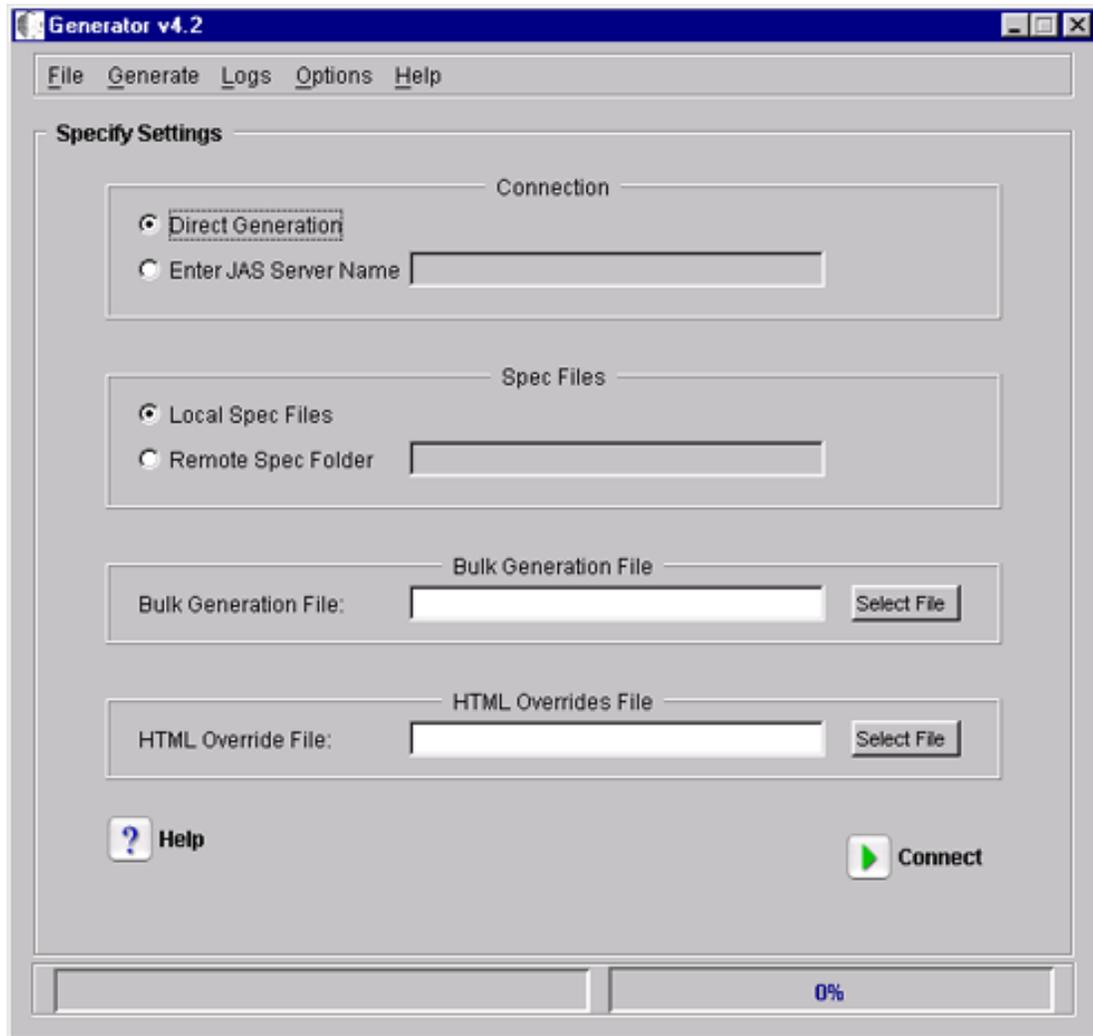
Webサーバーへのログイン後は、次のタスクをスキップし、その後のオブジェクト生成方法に関するセクションを参照してください。

Webサーバーをバイパスしてデータベースに直接生成する方法

オブジェクトをデータベースに直接生成するには、次のタスクを実行します。

▶ オブジェクトをデータベースに直接生成するには

1. Gen.batを起動します。
2. 〈OneWorld/ERP™ Sign On〉で、OneWorld/ERPにログインします。
3. [Direct Generation(直接生成)]ラジオ・ボタンをクリックします。



直接生成モードでは、ローカル・ディスク上のjas.iniファイルの[JDBC URL]セクションでサーバーを検索することでデータベースが検索されます。ただし、Javaサーバーを使用してオブジェクトを生成する場合、ローカル・ドライブ上のjas.ini ファイルは不要です。

4. [Connect(接続)]をクリックします。

システムはWebサーバーをバイパスし、jas.iniファイルで指定されているデータベースに直接接続します。

eGeneratorの構成

eGeneratorには、特定の環境に合わせて設定できる多数のオプションが用意されています。構成プロセスは次のタスクからなります。

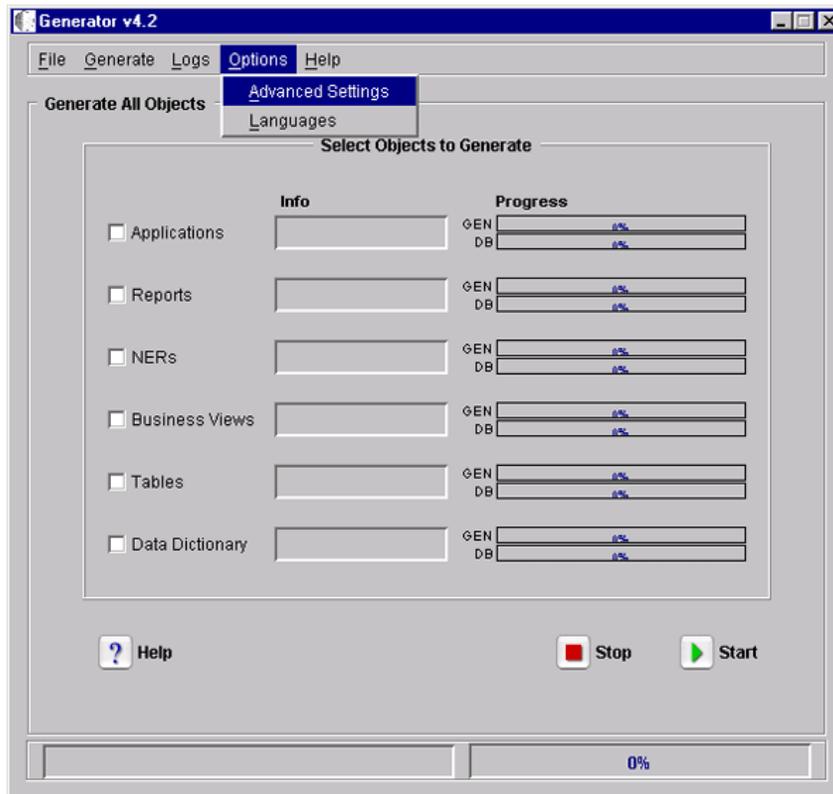
- ・ [eGeneratorオプションの設定 - ページ \[55\]](#)
- ・ [HTMLOverridesファイルの選択 - ページ \[58\]](#)
- ・ [言語の選択 - ページ \[58\]](#)
- ・ [genapp.iniファイルの構成 - ページ \[59\]](#)

eGeneratorオプションの設定

オブジェクトを生成する前に、eGeneratorの動作に影響するさまざまなパラメータを定義できます。

▶ eGeneratorオプションを設定するには

1. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。



2. 〈Advanced Settings〉で、生成要件に合ったオプションを選択します。



- ・ Logging(ログ)

[Status Log(ステータス・ログ)]または[Error Log(エラー・ログ)]の左のチェックボックスをクリックし、生成プロセスのログを記録するかどうかを指定します。各ログ・ファイルの位置を入力することもできます。ログは、生成プロセスのトラブルシューティングには不可欠です。
- ・ 生成モード

該当するラジオ・ボタンを有効にして、全オブジェクトの生成をパラレル・モードで実行するか逐次モードで実行するかを指定します

生成モードについては、「[生成モード - ページ \[57\]](#)」を参照してください。
- ・ Report Licensing(レポート・ライセンス)

各レポートの後にレポート・ライセンス情報を生成するには、このチェックボックスをクリックします。
- ・ User Option(ユーザー・オプション)

関連オプションを選択することで、アプリケーションのパブリック・バージョンまたは個人用バージョンを生成できます。個人用バージョンを使用できるのは、そのバージョンの作成者のみです。
- ・ Application Entry Point(アプリケーション・エントリ・ポイント)

このオプションを有効にすると、各アプリケーションの後にアプリケーション・エントリ・ポイント情報が生成されます。

生成モード

eGeneratorには、次のWebオブジェクト生成モードが用意されています。

- ・ [逐次生成](#)
- ・ [パラレル生成](#)
- ・ [直接生成](#)

逐次生成

これはデフォルトの生成モードです。このモードでは、[Generate-All Objects(全オブジェクトの生成)]オプションを有効にすると、Webオブジェクトが1つずつ逐次生成されます。

メリット:

- ・ 最も安定的な生成モード
- ・ メモリなどのシステム・リソースの所要量が最小
- ・ JITI(ジャストインタイム・インストール)の実行時にもエラーなしで使用可能

デメリット:

- ・ 最も低速の生成モード

パラレル生成

この生成モードを選択するには、[Advanced Settings]、[Parallel Generation(パラレル生成)]の順にクリックします。

この生成モードでは、[Generate-All Objects]オプションを有効にすると、Webオブジェクトが同時に生成されます。

メリット:

- ・ 逐次生成より高速(特に、マルチプロセッサ・ワークステーションの場合)

デメリット:

- ・ JITIが発生する可能性があるため、ユーザーがスペック・ファイル・セット全体を持っていない場合は

使用不可

- ・ パラレル生成中にJITIが発生するとスペック・ファイル破損の可能性あり

直接生成

この生成モードを選択するには、eGeneratorの〈Connect to Web Server (Webサーバーへの接続)〉画面で、下部にある[Direct Generation (直接生成)]チェックボックスを有効にします

直接生成は、前述の各生成タイプと併用できます。このモードでは、ジェネレート・マシン上で小型のJASサーバー・インスタンスが初期化されます。小型JASサーバーが初期化された後、前述のどちらかの方法で生成処理を実行できます。生成中には、JASサーバーはバイパスされ、オブジェクトはデータベースに直接格納されます。

メリット:

- ・ Javaサーバーのセットアップ前にオブジェクトを生成可能
- ・ どの生成モードでも最も高速
- ・ 管理が容易

デメリット:

- ・ 初期セットアップが困難

HTMLOverridesファイルの選択

HTMLOverridesファイルを選択し、複数行編集などの上級機能を使用してHTMLオブジェクトを生成します。

▶ HTMLOverridesファイルを選択するには

1. Gen.batを起動します。
2. 〈OneWorld/ERP Sign On〉で、OneWorld/ERP™にログインします。
3. eGeneratorで、次のフィールドに値を入力します。
 - ・ [HTMLOverrides File (HTMLOverrides ファイル)] : htmloverrides.iniファイルのパスと名称を入力するか、[Select File (ファイルの選択)]ボタンをクリックしてポップアップ・ウィンドウでファイルを参照します。

eGeneratorを実行してアプリケーションを生成すると、htmloverrides.iniファイルが処理されます。

言語の選択

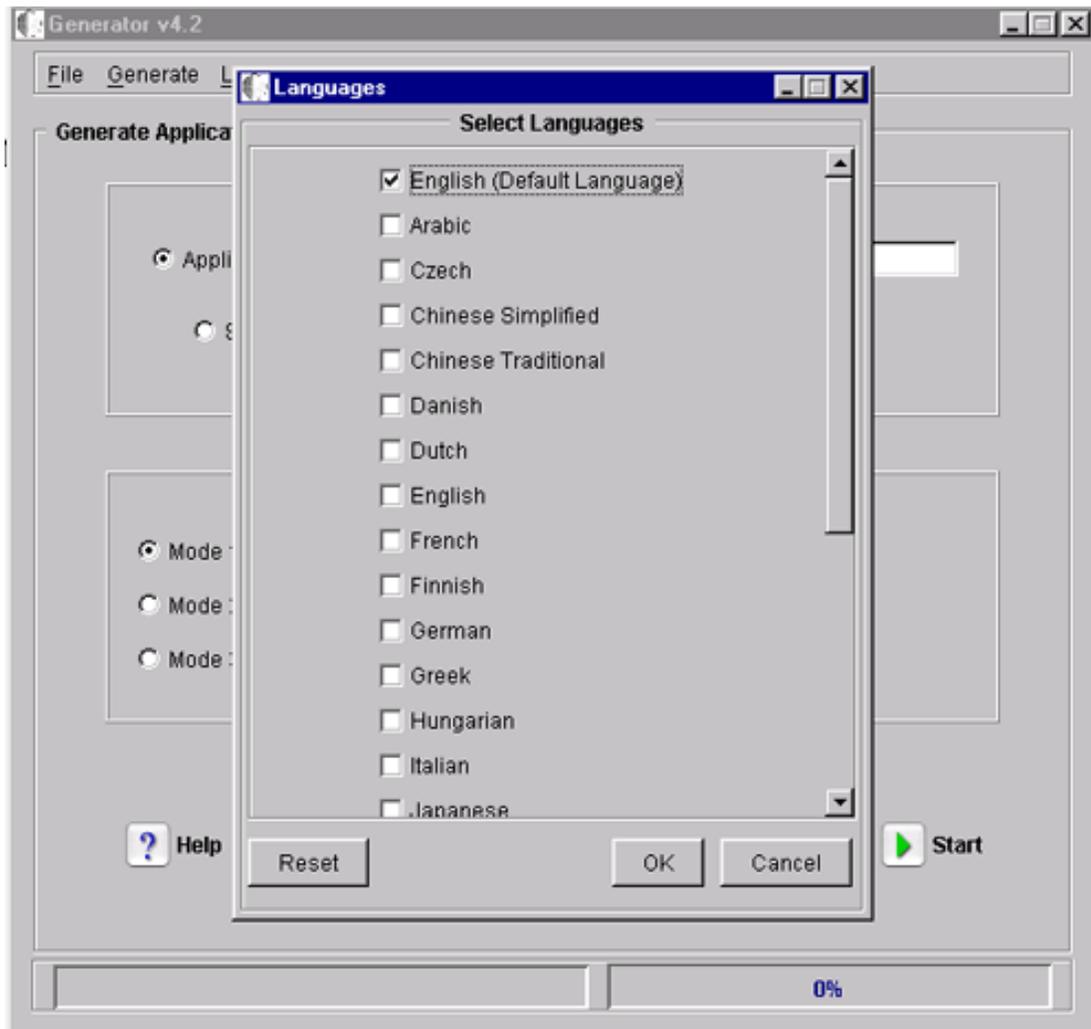
eGeneratorアプリケーションでサポート言語を選択することで、アプリケーションとレポートを希望する言語で生成できます。

注意

使用する言語ごとにLanguage Service Packをインストールする必要があります。

▶ 言語を選択するには

1. eGeneratorで、プルダウン・メニューから[Options]->[Languages(言語)]を選択します。



2. 使用する各言語の左のチェックボックスを有効にします。
3. [OK]をクリックします。

genapp.iniファイルの構成

genapp.iniファイルには、デフォルト以外のモードで生成するアプリケーションのリストが含まれています。デフォルト・モードは1で、Windowsの外観と操作性が得られます。genapp.iniファイルで、他のいずれかのモードで生成するアプリケーションをリスト形式で指定できます。Javaの場合はモード2、HTMLの場合はモード3です。

次のファイルは、サンプルのgenapp.iniファイルです。eGeneratorでは、このファイルが読み取られます。このファイルを使用して、特定のアプリケーションを生成用に構成できます。

```
# genapp.ini Last Revised 3/15/00
# Application is the name of the application to generate
# Mode is the FDA mode used to create the new "version" of the application.
# Attach a mode to a menu from menu design.This determines which mode of
# the application is displayed.The mode has nothing to do with whether or
# not it gets generated in HTML or Java - they are always generated for both.
# An application can only be specified for a mode once.
# Footer is the name of the menu you want displayed at the bottom of the
# application.Footer menus are optional.
```

```
P4015:2:G42314
P4015:3:G42314
P4210:3:G4231
P4210:2:G42314
P4101H:3:G42314
P41202:2:G42314
P41202:3:G42314
P03B2002:2:G42314
P03B2002:3:G42314
P42050:2:G42314
P42050:3:G42314
P4015W:2:G42314
P4015W:3:G42314
P40215:2:G42314
P40215:3:G42314
P41829W:2:G42314
P41829W:3:G42314
P42232:2:G42314
P42232:3:G42314
P4006:2:G42314
P4006:3:G43S11
P42045:2:G42314
P42045:3:G42314
P01012:2:G42314
P01012:3:G43S11
P4310:2:P4310:3:G43S11
P43214:2:P43214:3:P4334:2:P4334:3:G43S11
P0411:2:P0411:3:G43S11
P34301:2:P34301:3:G43S11
P40ITM3:2:G42314
P40ITM3:3:G42314
P4960:2:P01013:2:G42314
P01013:3:G42314
```

P3460:2:G42314
 P3460:3:G42314
 P43214:3:G43S11
 P41201:3:G43S11
 P43230:3:G43S11
 P44200:3:G43S11
 P430114:3:G43S11
 P43100:2:G43S11
 P43100:3:G43S11
 P0150R:3:G42314
 P0150S:3:G42314
 P055011:2:G05BESS1
 P055011:3:G05BESS1
 P4915:3:G43S11
 P4960:3:G43S11
 P17500:2:G1715
 P17500:3:G1715
 P17501:2:G1715
 P17501:3:G1715
 P48201:2:G1715
 P48201:3:G1715
 P17714:2:G1715
 P17714:3:G1715
 P1723:2:G1715
 P1723:3:G1715
 P4947:2:G1705
 P4947:3:G1705
 P32942:3:P053020:3:

P986116 and P986110B do not use modes in B733.3
 # They are being kept in genapp.ini to prevent any
 # potential issues with previous releases.

P986116:2:P986116:3:P986110B:2:P986110B:3:

#These apps were added for the project DONUT (6/22/01)
 P0092SS:3:P01012SS:3:P4334SS:3:P4311S:3:P4312S:3:P4310SS:3:P41204:3:P04111:3:
 P3462:3:P4210SS:3:P4947S:3:P03B2003:3:P43230SA:3:P42230:3:P42235:3:P42240:3:P34301:3:

#SAR
 5336769
 P0005S:3:P0006S:3:P0101S:3:

すべての標準シリアル化オブジェクトの生成

eGeneratorとJASビルドを併用して、Javaシリアル化オブジェクトのフル・セットを生成できます。これらのシリアル化オブジェクトを使用すると、Webサーバーの実行時にすべてのOneWorld/ERP™オブジェクトにアクセスできます。Webサーバーをインストールするか、新規サービス・パックにアップグレードする場合は、シリアル化オブジェクトのフル・セットを生成します。ただし、オブジェクトの部分セットのみを生成する場合は、このタスクをスキップし、次の「[選択したオブジェクトの生成 - ページ \[68\]](#)」の関連タスクを実行してください。

ここでは次の内容について説明します。

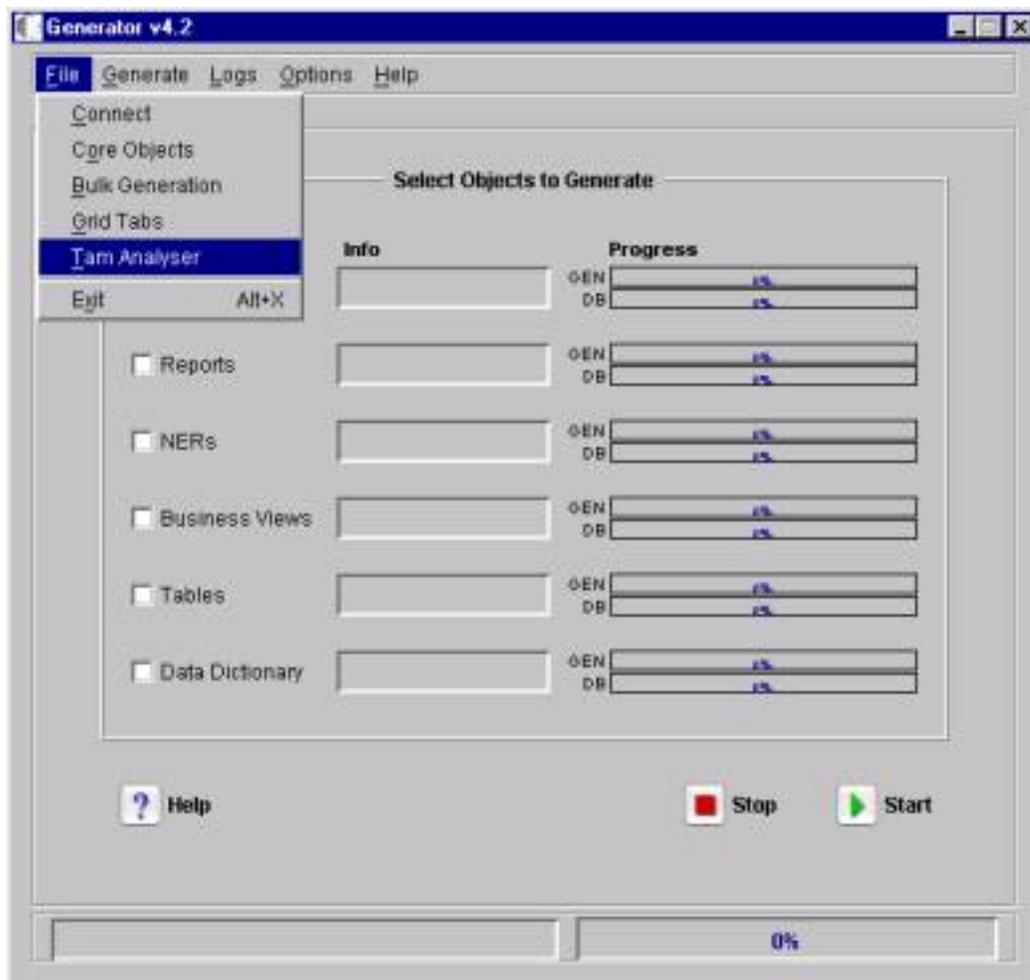
- ・ [TAM Analyzerの実行 - ページ \[62\]](#)
- ・ [全オブジェクトの生成 - ページ \[63\]](#)

TAM Analyzerの実行

TAM Analyzerを実行すると、eGeneratorでデータベースに保存せずに全オブジェクトを生成できます。これにより、生成プロセスをすばやくテストし、ログ・ファイルをチェックして破損スペックを検出できます。TAM Analyzerを使用するには、ステータス・ログとエラー・ログの両方をアクティブ化する必要があります。

▶ TAM Analyzerを実行するには

1. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
2. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
3. プルダウン・メニューから[File(ファイル)]->[Tam Analyzer]を選択します。



全オブジェクトの生成が開始されます。

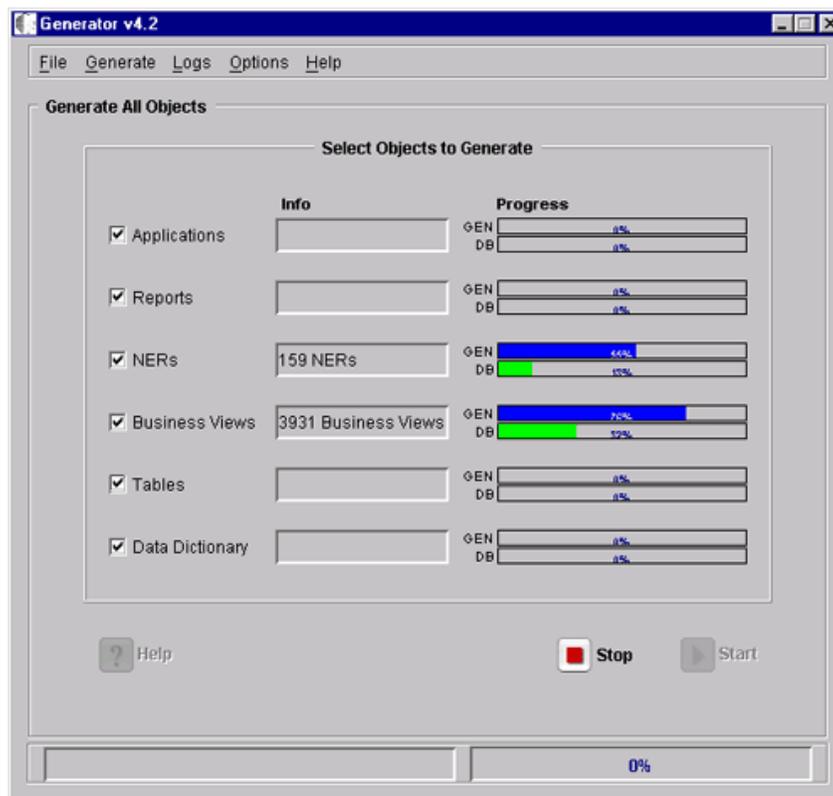
4. 生成完了後に、Status LogとError Logで破損スペックの有無をチェックします

全オブジェクトの生成

次のタスクを実行して、OneWorld/ERP™オブジェクトのフル・セットを生成します。

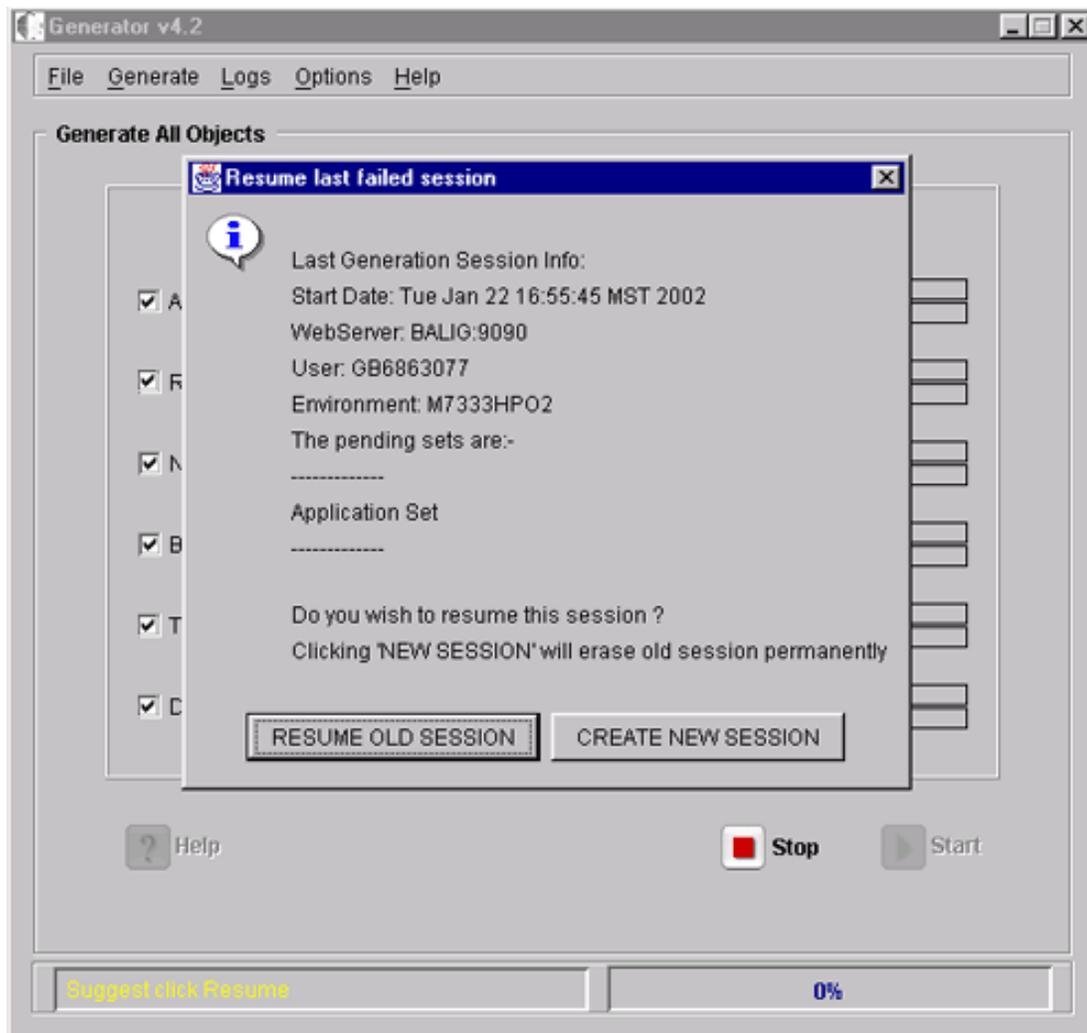
▶ すべてのOneWorld/ERP™オブジェクトを生成するには

1. プルダウン・メニューから[Generate(生成)]->[Core Objects(コア・オブジェクト)]を選択します。
コア・オブジェクトの生成については、「[コア・オブジェクトの生成 - ページ \[72\]](#)」を参照してください。
2. コア・オブジェクトが正常に生成された場合は、プルダウン・メニューから[Generate(生成)]->[All Objects(全オブジェクト)]を選択します。



3. <Generate All Objects(全オブジェクトの生成)>で、表示されるオブジェクトをすべて有効にします。
4. [Start(開始)]ボタンをクリックします。

前の<Generate-All Objects>のセッションが正常終了しなかった場合は、次の画面が表示されます。

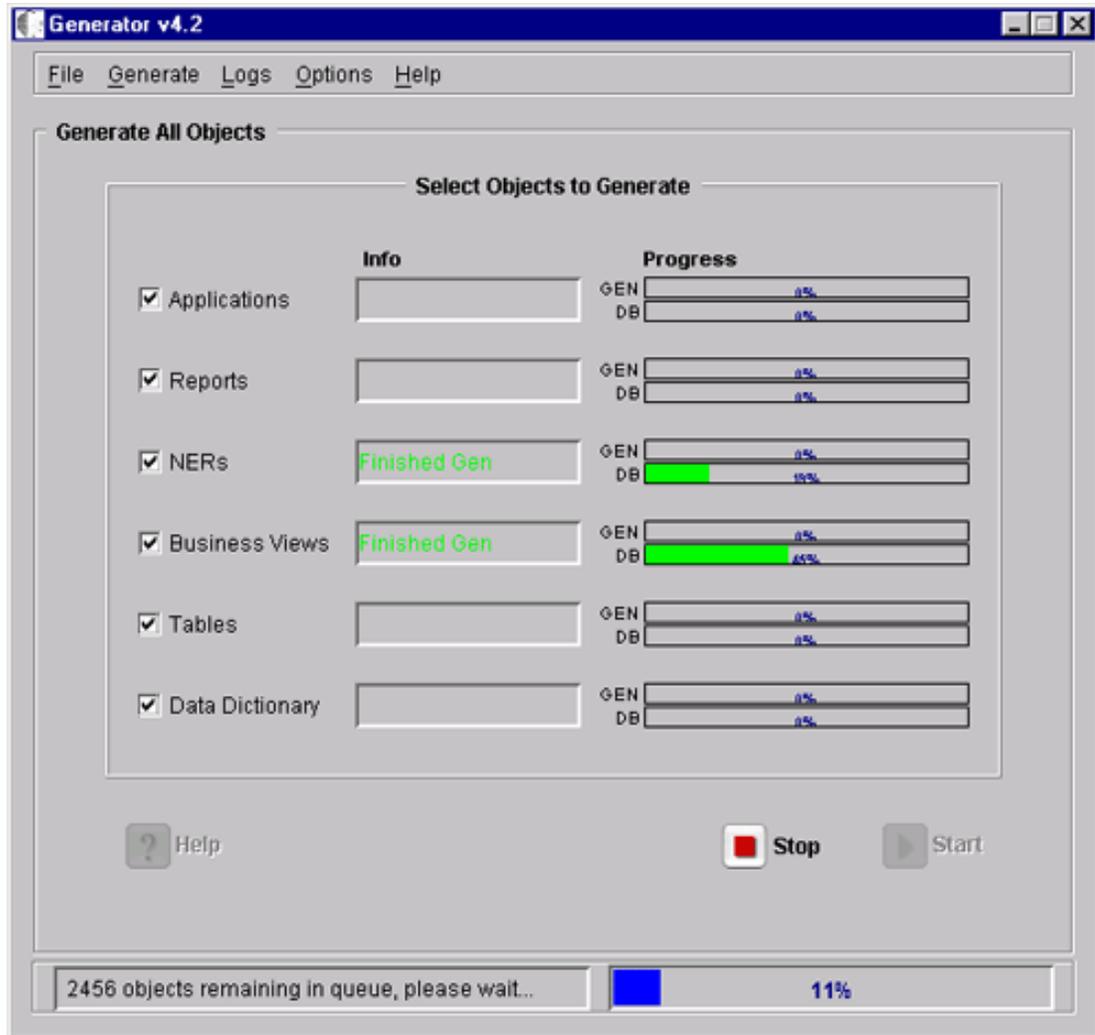


5. 次のオプションのうち1つを選択します。

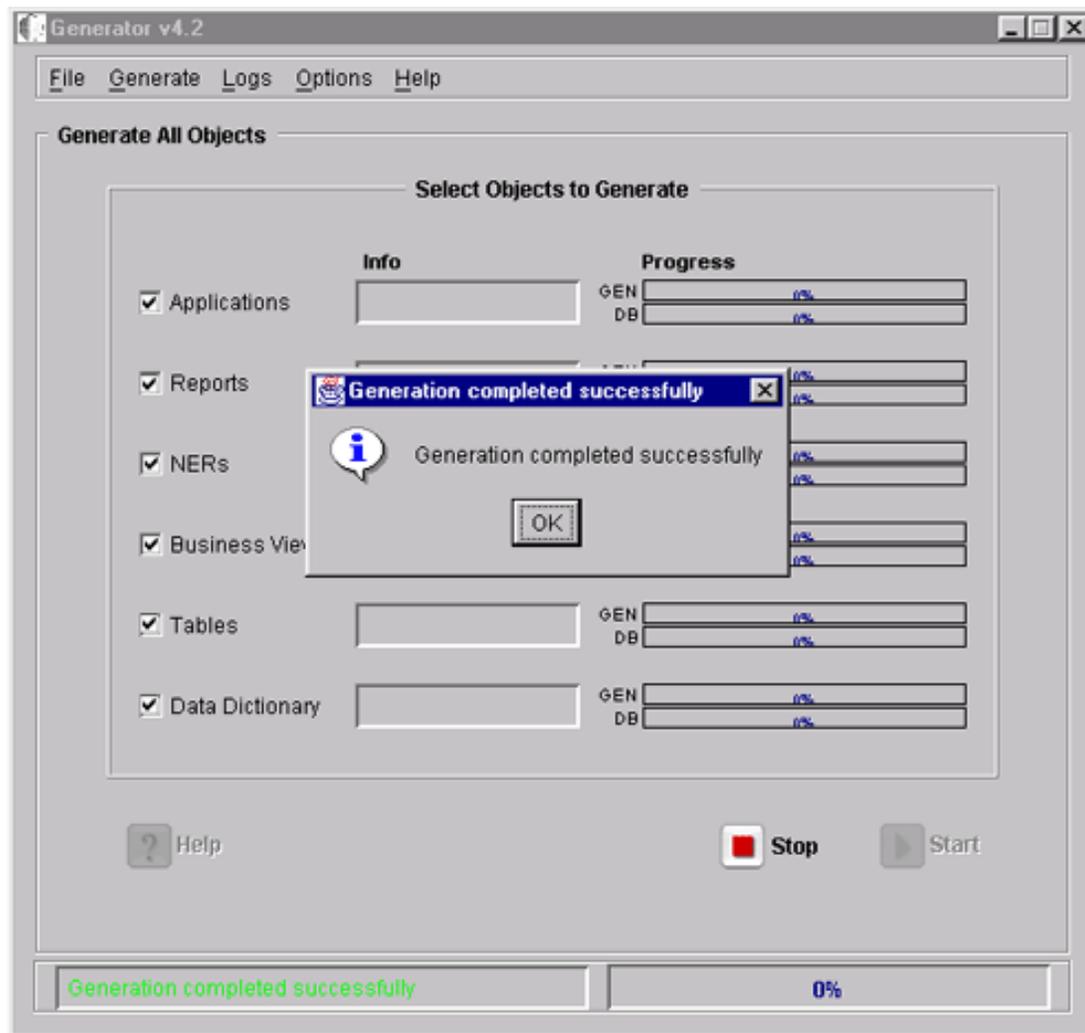
- ・ 前の生成プロセスを再開するには、[RESUME OLD SESSION(前セッションの再開)]をクリックにします。前のプロセスがエラー発生ポイントから再開されるので、このオプションを有効にすることをお勧めします。そのプロセスが完了した後、このタスクを繰り返して新規オブジェクト・セットを生成できます。
- ・ 前の生成プロセスを取り消して新規オブジェクトを生成するには、[CREATE NEW SESSION(新規セッションの作成)]をクリックします。このオプションを選択するのは、すべてのオブジェクト・タイプの新規フル・セットを生成する場合です。

どちらのオプションをクリックした場合も、オブジェクト・セットの生成が開始されます。オブジェクト生成の進行状況が画面に表示されます。青の進行状況バーは生成プロセスの進行状況を示し、緑の進行状況バーはオブジェクトをデータベースに挿入する処理の進行状況を示します。

生成が完了した時点でデータベースへのオブジェクト挿入が完了しない場合は、待ち行列に残っているオブジェクトの数が表示されます。



すべてのオブジェクトがデータベースに挿入されると、次のメッセージが表示されます。



生成プロセスの検証

次のタスクを実行して、生成プロセスが正常終了したかどうかを検証します。

- ・ [ログ・ファイルのチェック - ページ \[66\]](#)
- ・ [データベース確認のチェック - ページ \[67\]](#)

ログ・ファイルのチェック

eGeneratorでは、メニューから[Status Log]と[Error Log]を選択すると、これらのログ・ファイルを自動的に表示できます。

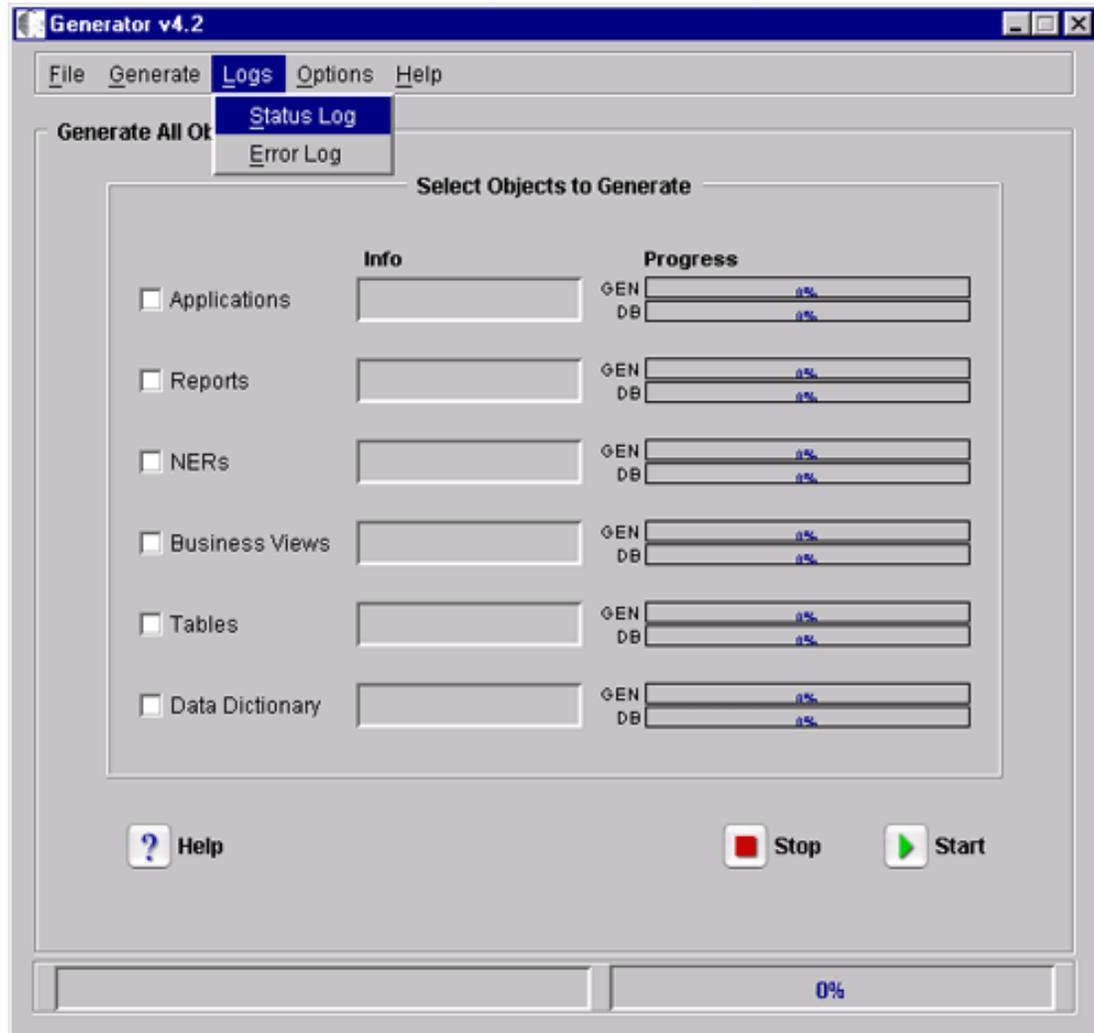
注意

これらのログは、[Generate All Objects]オプションを有効にするたびに、選択したオブジェクト・

タイプに関係なく生成されます。

▶ ステータス・ログを表示するには

- ・ eGeneratorで、プルダウン・メニューから[Logs(ログ)]->[Status Log]を選択します。



▶ エラー・ログを表示するには

- ・ eGeneratorで、プルダウン・メニューから[Logs(ログ)]->[Error Log]を選択します。

データベース確認のチェック

単一オブジェクトの生成時には、オブジェクトがデータベースに正常に挿入されたことを示す確認メッセージが[コマンド プロンプト]ウィンドウに送られます。これにより、生成プロセスをモニタリングし、オブジェクトがデータベースに正常に挿入されていることを確認できます。

ただし、データ辞書項目またはデータ構造体の場合、[コマンド プロンプト]画面に出力するには多すぎ

るので、これらの確認は出力されません。

次の例は、DOSコマンド行への出力を示します。

```
Generating application P01012 ...
-->Database Ack:ER_P01012_W01012D successfully inserted into database
-->Database Ack:P01012_HTML_W01012D successfully inserted into database
-->Database Ack:ER_P01012_W01012B successfully inserted into database
-->Database Ack:P01012_HTML_W01012B successfully inserted into database
-->Database Ack:ER_P01012_W01012A successfully inserted into database
-->Database Ack:P01012_HTML_W01012A successfully inserted into database
App generation finished
Generating 2 NERs
-->Database Ack:ER_P01012_W01012C successfully inserted into database
-->Database Ack:P01012_HTML_W01012C successfully inserted into database
-->Database Ack:NER_PlugAndPlay_P0101 successfully inserted into database
NER generation finished
Generating 1 PO Data Structure Items
PO Data Structure generation finished
Generating 2 Business Views
Business View generation finished
Generating 5 Tables
Table generation finished
Generating 294 Data Dictionary Items
Data Dictionary generation finished
Generating 30 Data Structure Items
Data Structure generation finished
-->Database Ack:NER_UpdateChangedGeoCode successfully inserted into database
-->Database Ack:PODATAP01012 successfully inserted into database
-->Database Ack:VIEW_V0101E successfully inserted into database
-->Database Ack:TABLE_F0101 successfully inserted into database
-->Database Ack:TABLE_F0111 successfully inserted into database
-->Database Ack:TABLE_F0116 successfully inserted into database
-->Database Ack:TABLE_F03012 successfully inserted into database
-->Database Ack:TABLE_F0401 successfully inserted into database
```

選択したオブジェクトの生成

最初にOneWorld/ERPオブジェクトを生成した後、OneWorld/ERP™オブジェクト(UBEやアプリケーションなど)を随時作成またはカスタマイズする場合は、各オブジェクトを個別に生成する必要があります。

eGeneratorには、シリアル化オブジェクトの部分セットを生成できるように、複数の方法が用意されています。「バルク生成ファイルを使用したオブジェクトの生成」を除き、すべての方法はeGenerator画面からオプションを選択することにより実行されます。バルク生成ファイルを使用すると、選択したオブジェクトをテキストファイルに直接入力し、このファイルをeGeneratorで実行できます。

次のいずれかのタスクを実行して、シリアル化オブジェクトの部分セットを生成します。

- ・ [バルク生成ファイルを使用したオブジェクトの生成 - ページ \[69\]](#)
- ・ [コア・オブジェクトの生成 - ページ \[72\]](#)
- ・ [アプリケーションの生成 - ページ \[74\]](#)

- ・ [フォームの生成 - ページ \[78\]](#)
- ・ [レポートの生成 - ページ \[80\]](#)
- ・ [NERの生成 - ページ \[83\]](#)
- ・ [データ辞書項目の生成 - ページ \[84\]](#)
- ・ [テーブルの生成 - ページ \[85\]](#)
- ・ [ビジネス・ビューの生成 - ページ \[86\]](#)
- ・ [データ構造体の生成 - ページ \[88\]](#)
- ・ [グリッド・タブの変換 - ページ \[89\]](#)

バルク生成ファイルを使用したオブジェクトの生成

バルク生成ファイルは、一連の生成パラメータを含むテキスト・ファイルです。バルク生成を使用すると、1つまたは複数の生成パラメータ・セットを簡単に構成してファイルに格納できます。その後は、該当するバルク生成ファイルを選択すると、生成パラメータ・セットをeGeneratorに自動的にロードできます。

ここでは次の内容について説明します。

- ・ [バルク生成ファイルの作成 - ページ \[69\]](#)
- ・ [バルク生成ファイルの実行 - ページ \[70\]](#)

バルク生成ファイルの作成

gen.batと共に、バルク生成ファイルの作成用テンプレートが用意されています。新規バルク生成ファイルを作成するには、このテンプレートを修正し、同じファイルとして保存する必要があります。

▶ バルク生成ファイルを作成するには

1. WindowsエクスプローラでBulkGen.txtファイルを開きます。通常、このファイルは次のディレクトリにあります。

¥B9¥Generator¥WebClient



```

MAGIC_NUMBER=0x00C8FFEE
#Do Not modify the Magic Number or its position in any case!
#This file contains the objects that will be generated when this file is chosen
#and menu item BulkGen is clicked.
#Please add the desired objects in the proper section
#Invalid object names or objects in incorrect section can not be generated.

#Applications
APP:P01012
APP:P4210
APP:P0911

#Reports
REPORT:R0008P
REPORT:R0006P

#Ners (Ners are case-sensitive)
NER:FetchUBEMapping
NER:GetObjectDesc

#Views
VIEW:V9001B
VIEW:V0101E

#Tables
TABLE:F9000
TABLE:F9210

#DataDictionary
DD:OBHM
DD:SY

```

- このファイル内の既存のオブジェクトをサンプルとして使用し、生成するオブジェクトを入力します(このサンプルは、完了後に必ず削除してください)。

注意

ファイルの先頭にあるMagic_Numberは削除または修正しないでください。この番号により、このファイルがバルク生成ファイルとして識別されます。

- ファイルを保存します。

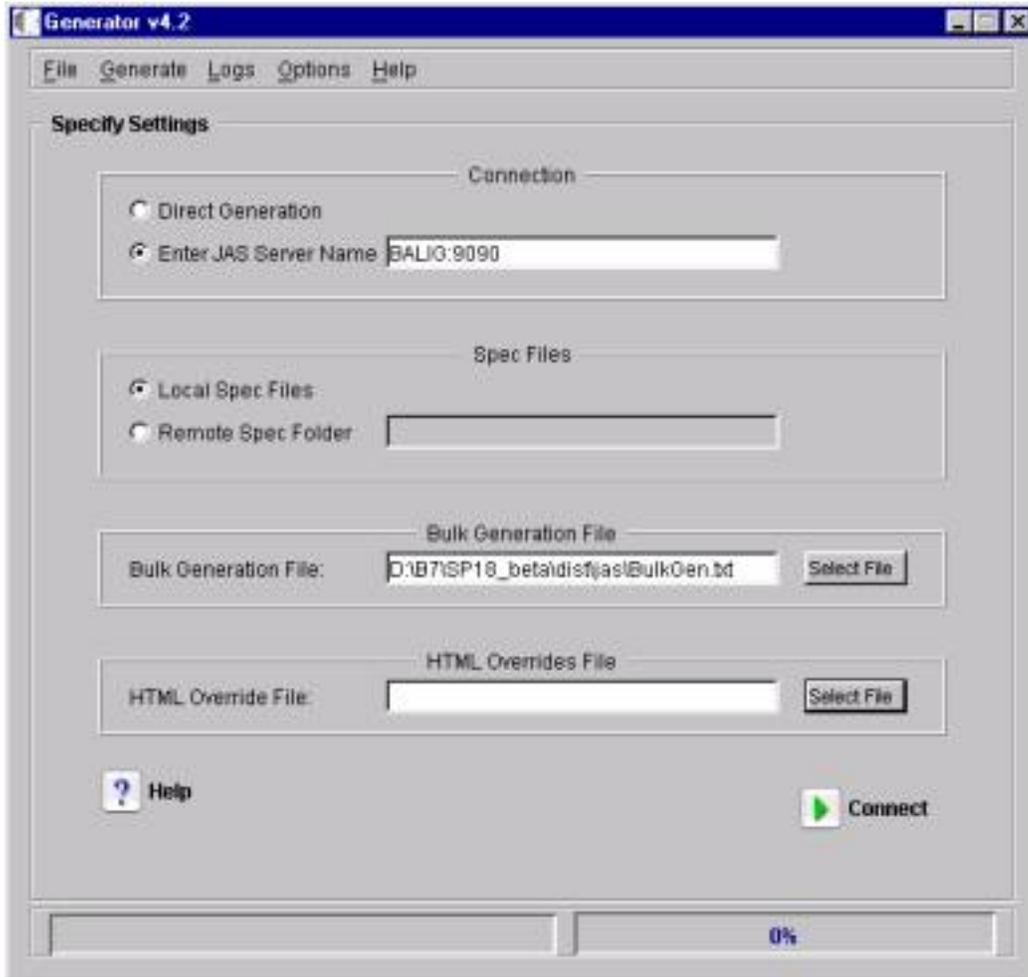
バルク生成ファイルの実行

▶ バルク生成ファイルを実行するには

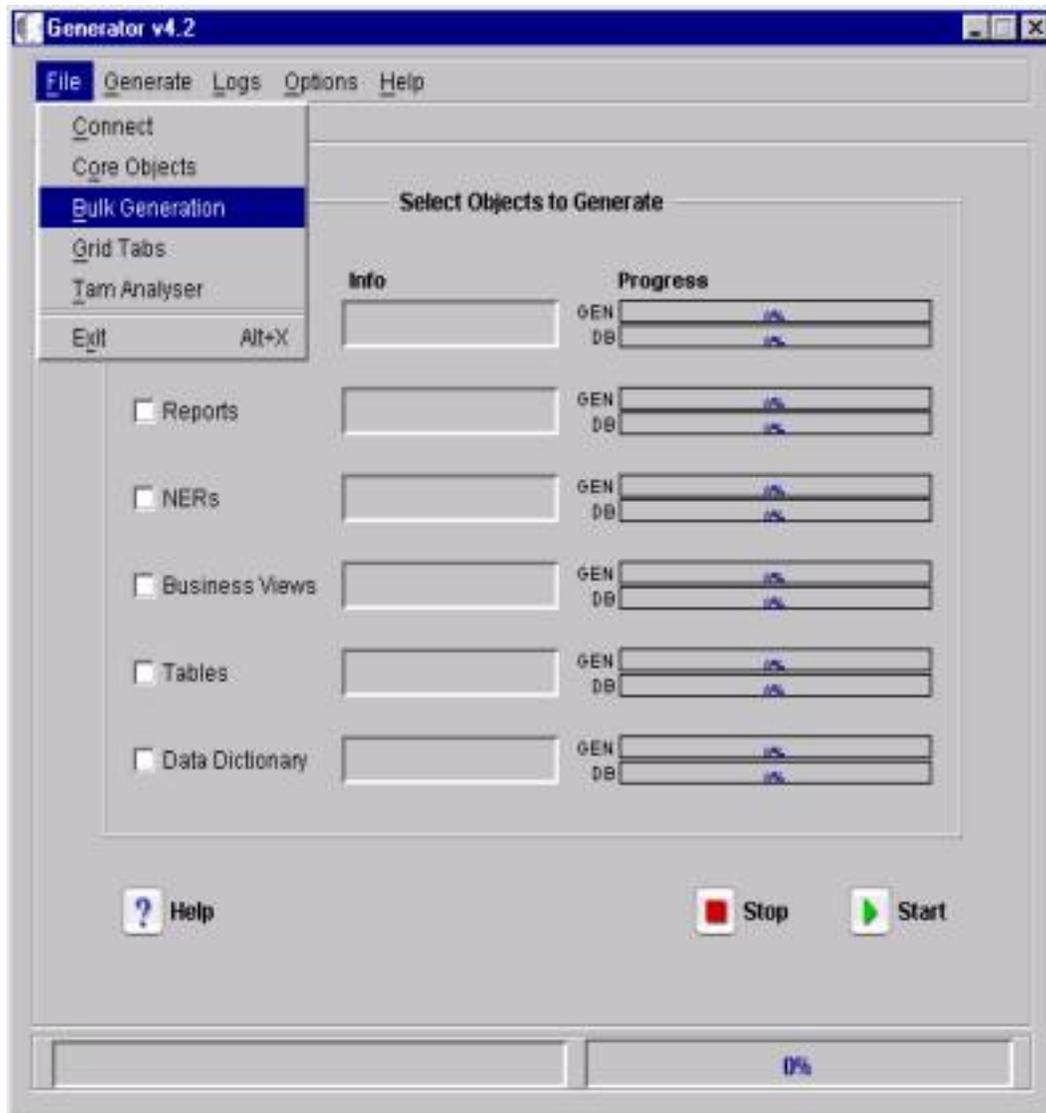
- Gen.batを起動します。
- 〈OneWorld Sign On(OneWorldへのサインオン)〉で、OneWorld/ERPにログインします。
- eGeneratorで、次のフィールドに値を入力します。

- ・ [Bulk Generation File(バルク生成ファイル)]: ファイル名を入力するか、[Select File]をクリックしてバルク生成ファイルを選択します。

システムがファイルに接続するまでに少し時間がかかることがあります。



4. 画面上部のプルダウン・メニューから、[File]->[Bulk Generation(バルク生成)]を選択します。



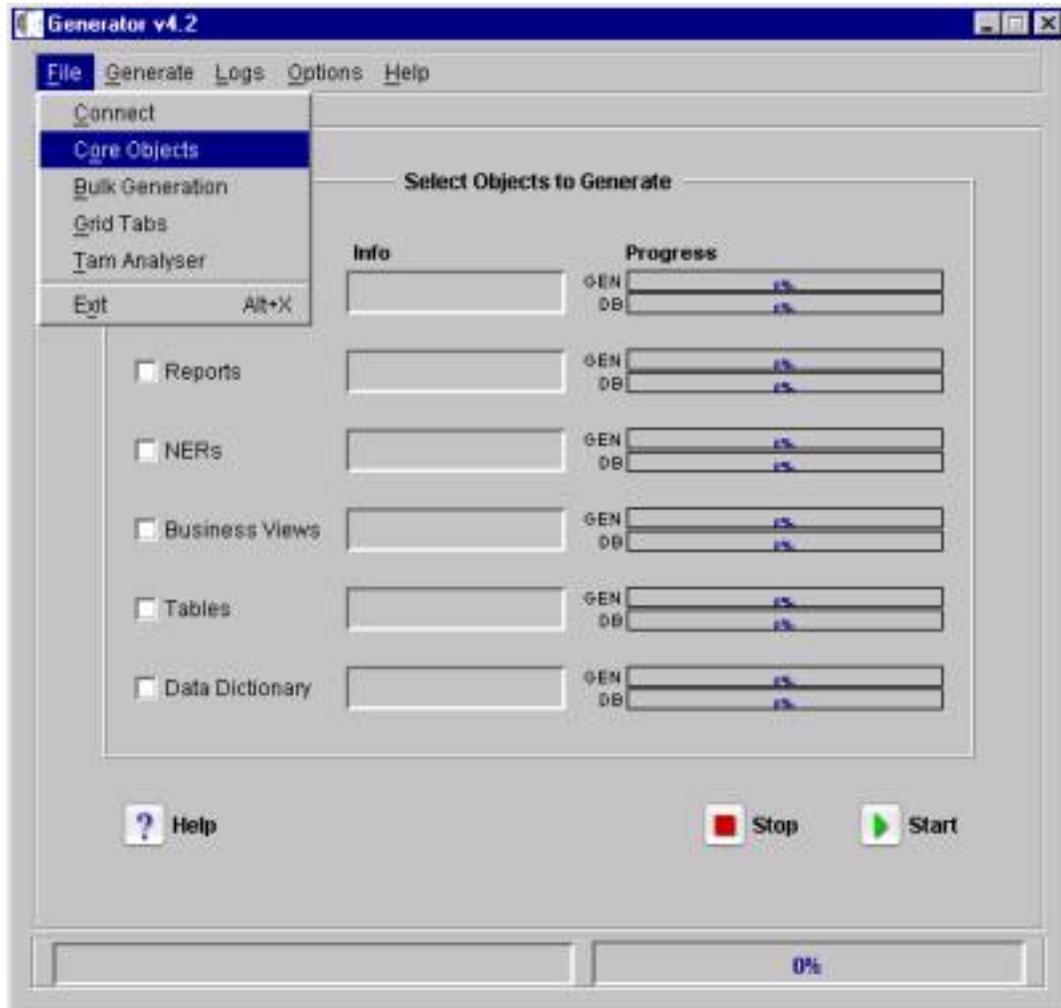
選択したファイルに指定されているオブジェクトがすべて生成されます。

コア・オブジェクトの生成

コア・オブジェクトは、他のOWコンポーネントに必要なファンデーション・オブジェクトです。これらのオブジェクトを生成すると、Javaサーバーでは特にアプリケーションをインストールしなくてもTask Explorer(タスク・エクスプローラ)やポータルを実行できます。

▶ コア・オブジェクトを生成するには

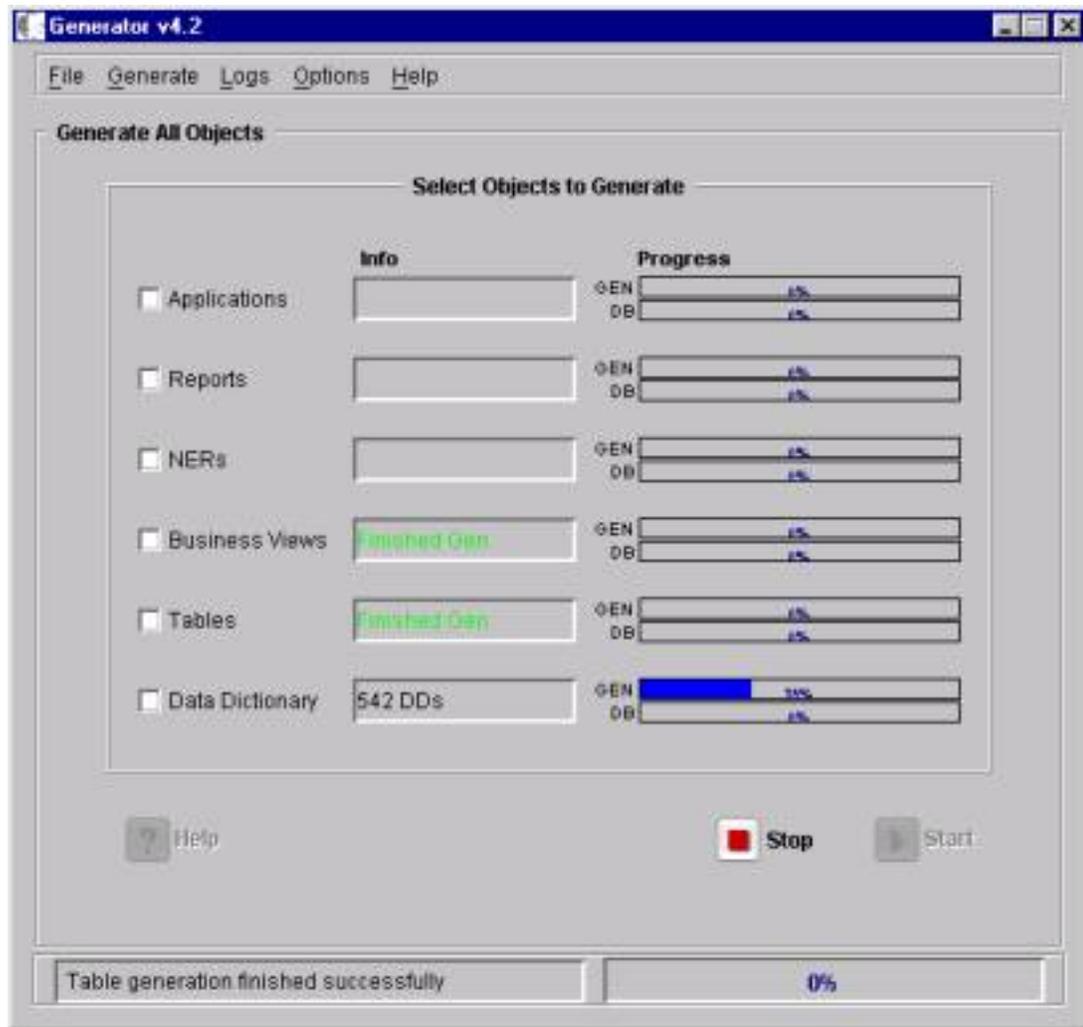
1. eGeneratorで、プルダウン・メニューから[File]->[CoreObjects]を選択します。



コア・オブジェクトとそれに従属関係を持つデータ辞書項目の生成が開始されます。

- ステータス・ログとデータベースをチェックして、次のオブジェクトが生成されたことを確認します。

テーブル	F9860、F9865、F0005、F983051、F9210、F98980、F98950、F0092、F0101、F00950、F95921、F9060、F9060D、F9061、F9062、F9064、F0010、F0093、F986101、F00921、F00922、F0004、F00941、F0094、F95921、F9065、F0082、F00821、F0083、F0084、F9000、F9001、F9002、F9006
ビュー	V9001B



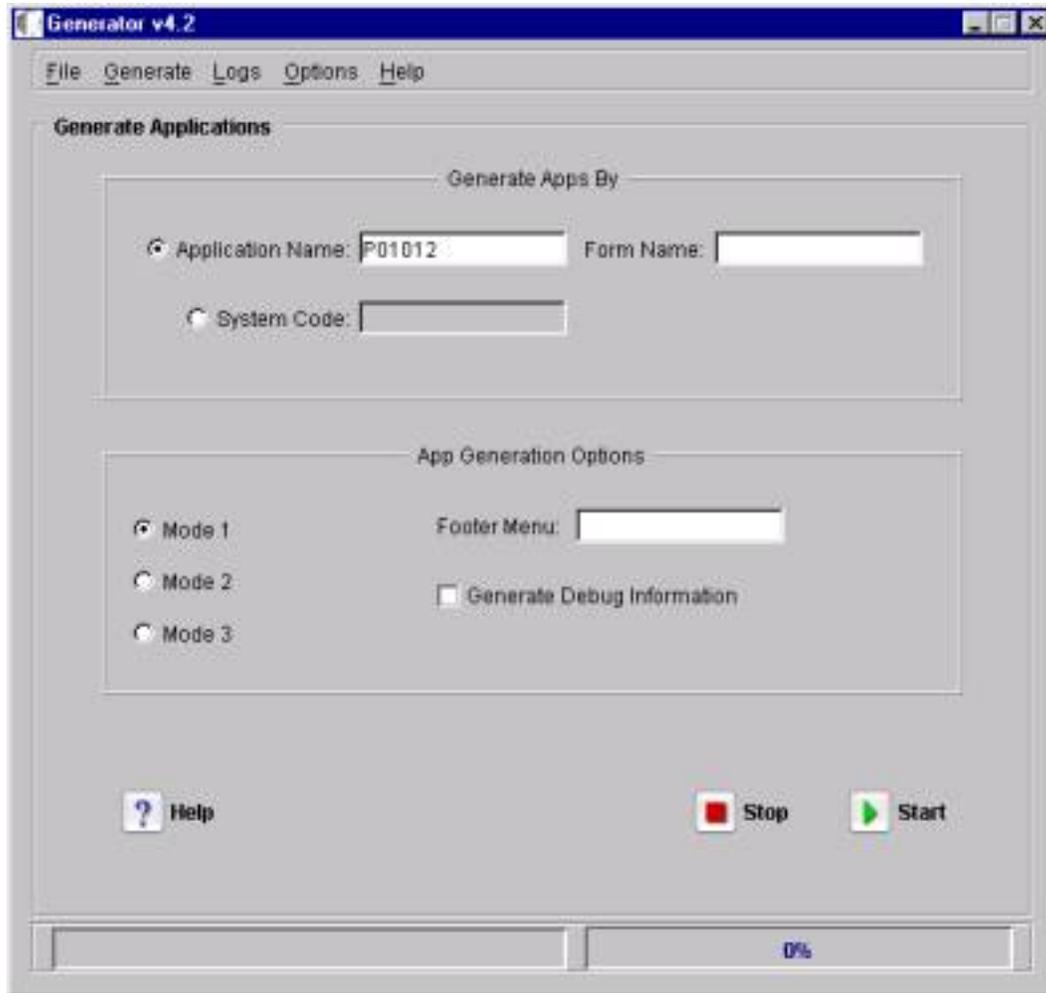
アプリケーションの生成

次のいずれかのタスクを実行して、1つまたは複数のアプリケーションを生成します。

- ・ [手順 20](#) [74]
- ・ [手順 21](#) [76]
- ・ [手順 22](#) [77]

▶ 単一アプリケーションを生成するには

1. プルダウン・メニューから[Generate]->[Applications]を選択します。



2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. [Application Name(アプリケーション名)]を有効にして、アプリケーション名(P01012など)を入力します。
5. [Start]をクリックします。

最初にオブジェクト・ライブラリアン・マスター(F9860)と38のデータ辞書項目が生成されます。

6. ステータス・ログで生成状況をチェックします。

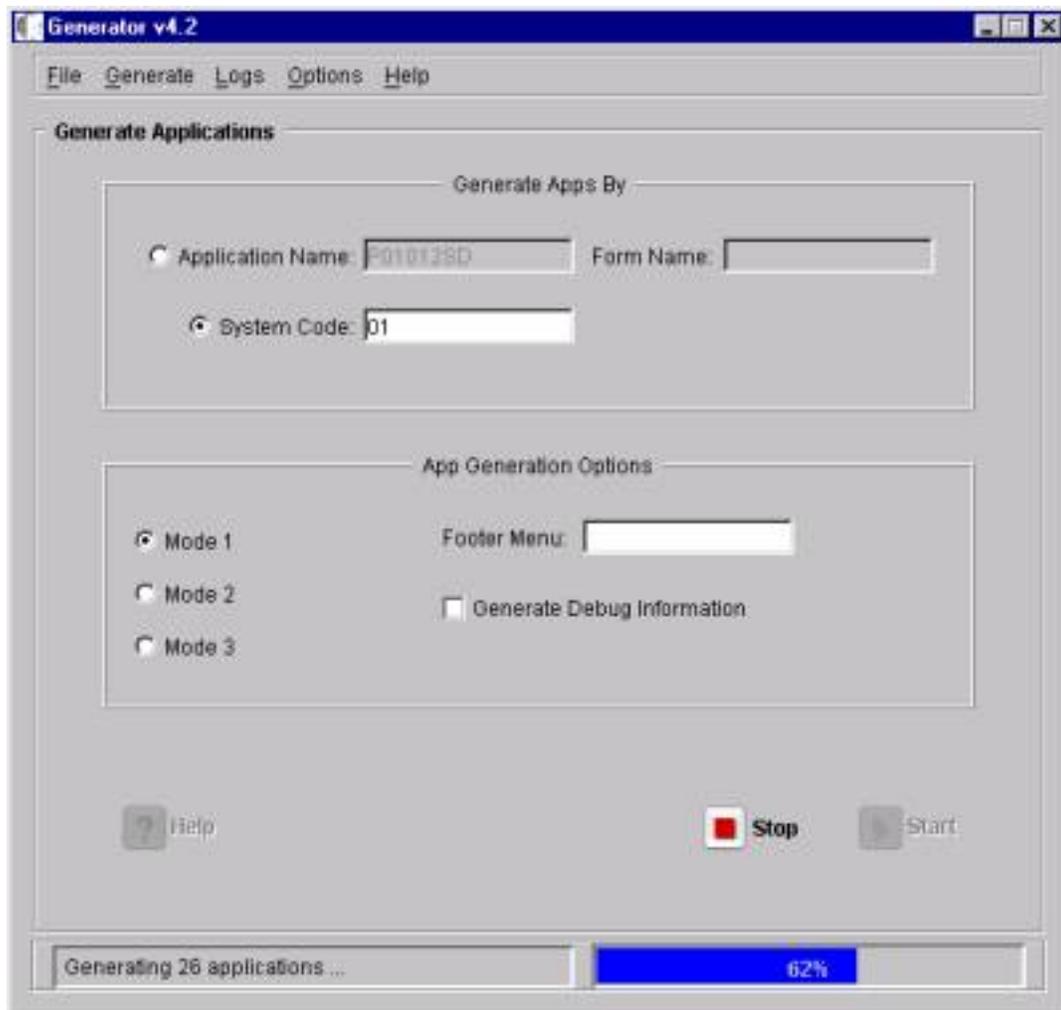
P01012の場合は、次のオブジェクトが生成されます。

- ・ 3つのフォームP01012_W01012A、P01012_W01012B、P01012_W01012Cと、それぞれに対応するERオブジェクト
- ・ アプリケーションに含まれるビュー、テーブル、データ辞書項目

7. ステータス・ログをチェックして、これらのオブジェクトが生成されたことを確認します。

▶ システム・コード別にアプリケーションを生成するには

1. プルダウン・メニューから[Generate]->[Applications]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. [Generate by System Code(システム・コード別に生成)]を選択します。
6. [Start]をクリックします。



指定したシステム・コードを持つアプリケーションがすべて取り込まれます(たとえば、マスター環境では26のアプリケーションが取り込まれます)。

7. ステータス・ログとエラー・ログをチェックして、各アプリケーションが生成されたことを確認します。

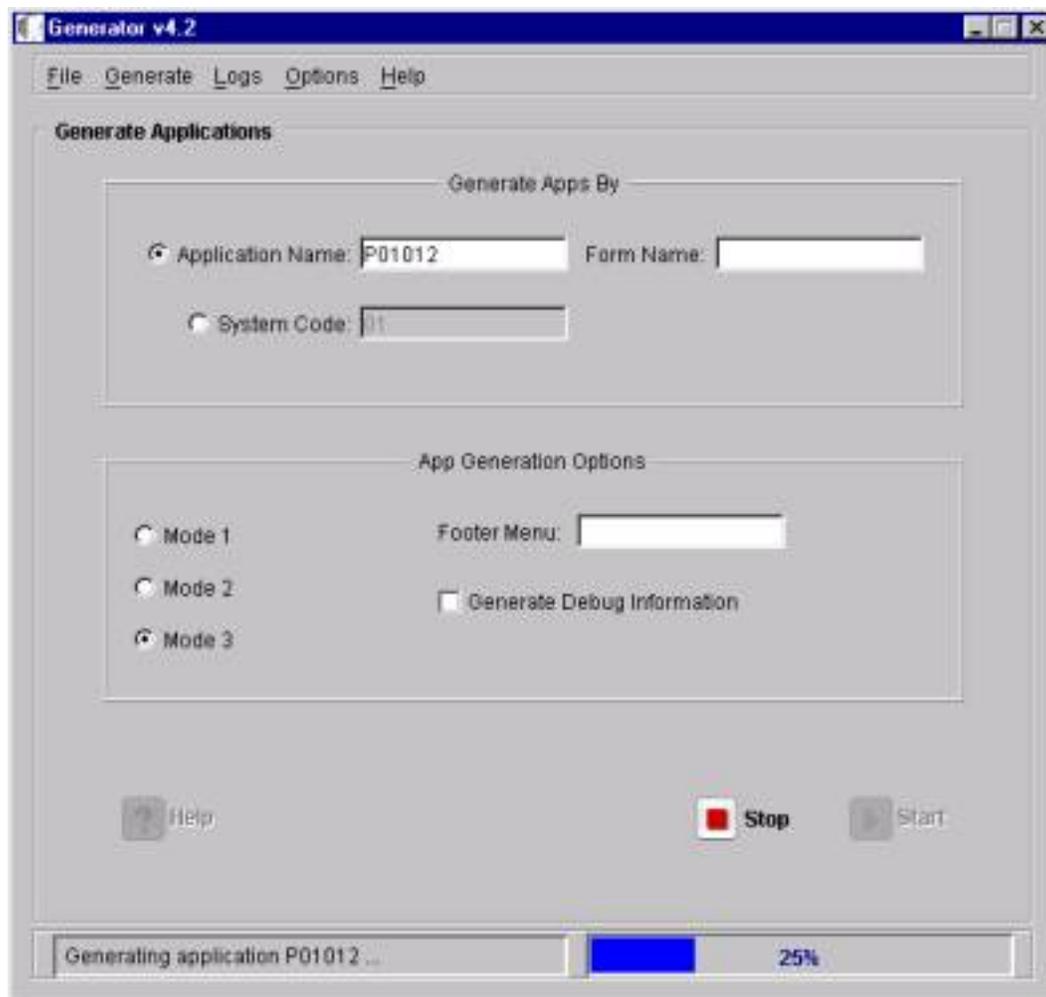
8. 次のSQLコマンドを使用して、各アプリケーションがデータベース上にあることを確認します。

```
SQL> select wboid from f989999 where wboid like 'P%W%';
```

▶ モード別にアプリケーションを生成するには

アプリケーションをモード1、2、または3で生成できます。これらのモードにより、アプリケーションの外観と操作性が確定されます。生成モードについては、「OneWorld/ERP Webアプリケーションの設計」の「オプションの生成」を参照してください。

1. プルダウン・メニューから[Generate]->[Applications]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. [Mode 3(モード3)]を選択します。



6. [Application Name]を有効にして、アプリケーション名 (P01012など)を入力します。
7. [Start]をクリックします。
8. 次のSQLコマンドを使用して、データベースにアプリケーションのHTML版が格納されていることを確認します。

```
SQL> select wboid from f989999 where wboid like 'P01012_HTML%';
```

フォームの生成

▶ フォームを生成するには

1. プルダウン・メニューから[Generate]->[Applications]を選択します。



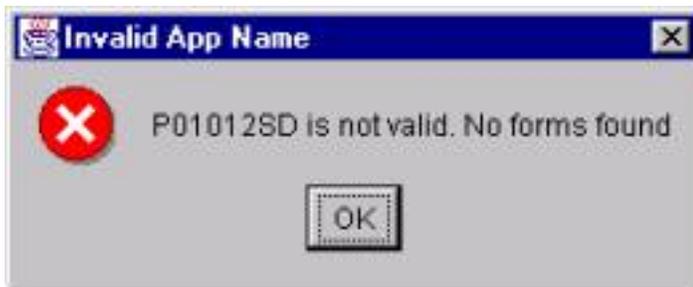
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。

5. [Form Name(フォーム名)]フィールドにフォーム名(P01012_W01012Aなど)を入力します。
6. [Start]をクリックします。

フォーム名が無効な場合は、次のエラー・メッセージが表示されます。



アプリケーション名が無効な場合は、次のエラー・メッセージが表示されます。



7. 最初にオブジェクト・ライブラリアン・マスター(F9860)と38のデータ辞書項目が生成されます。
8. ステータス・ログで生成状況をチェックします。

W01012Aの場合は、次のオブジェクトが生成されます。

- ・ フォームP01012_W01012Aとそれに対応するERオブジェクト
- ・ アプリケーションに含まれるビュー、テーブル、データ辞書項目
- ・ たとえば、P01012の場合は、次のオブジェクトと294のデータ辞書項目が作成されます。
- ・ NER_UpdateChangedGeoCode
- ・ NER_PlugAndPlay_P0101
- ・ F0101
- ・ F03012
- ・ F0116
- ・ F0111
- ・ F0401
- ・ V0101E

9. これらのオブジェクトがステータス・ログに含まれていることを確認するか、次のクエリーを使用してデータベースをチェックします。

```
SQL> select wboid from f989999
```

レポートの生成

次のいずれかのタスクを実行して、1つまたは複数のレポートを生成します。

- ・ [手順 24](#) [80]
- ・ [手順 25](#) [80]
- ・ [手順 26](#) [81]

▶ 単一レポートを生成するには

1. プルダウン・メニューから[Generate]->[Reports]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. レポート名 (R0006Pなど)を入力します。
6. [Start]をクリックします。

最初に、バージョン・リスト・テーブル(F983051)と28のデータ辞書項目が生成されます。

7. ステータス・ログで生成状況をチェックします。

レポートの全バージョンとそれに対応するDSEL、DSEQ、PRINTINFOの各オブジェクトが生成されます。

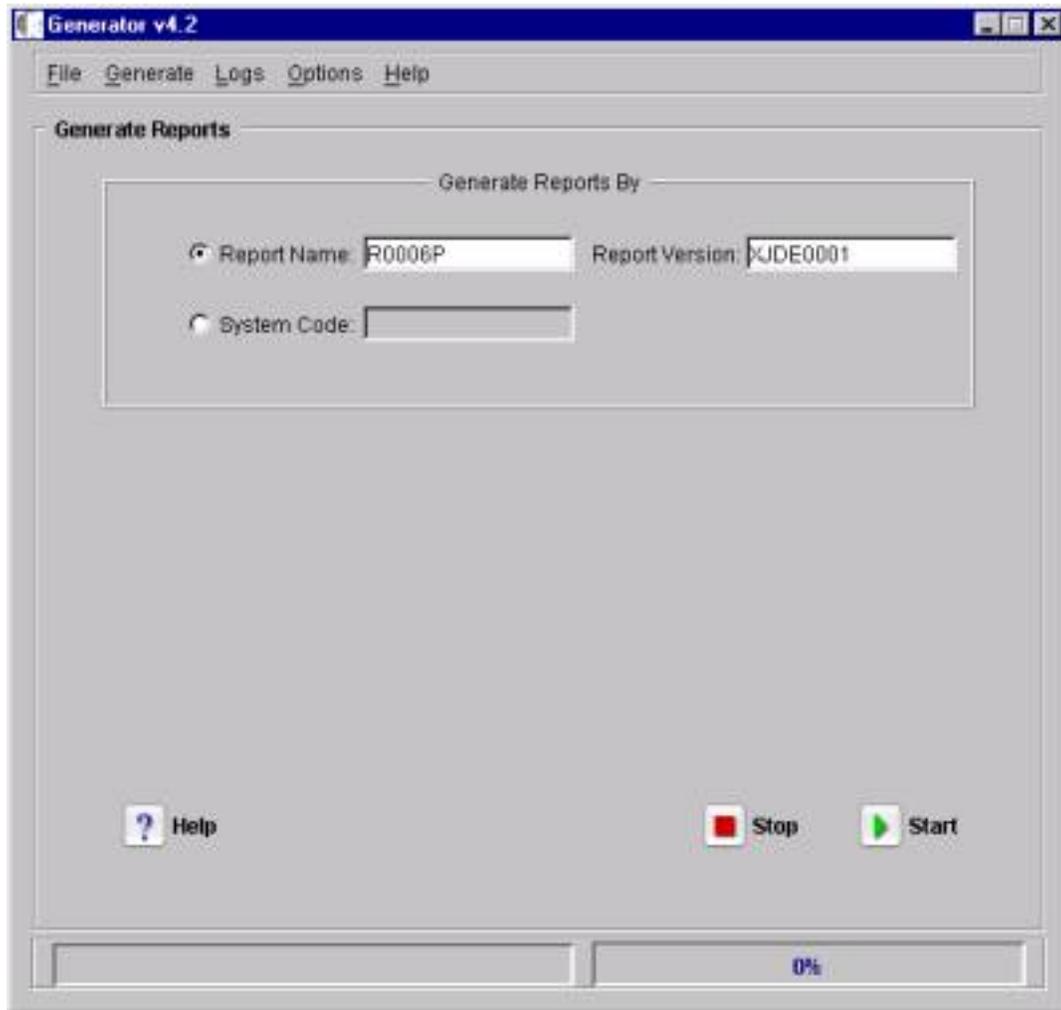
8. ステータス・ログを調べてオブジェクトの存在を確認するか、次のクエリーを使用してデータベースをチェックします。

```
SQL> select wboid from f989999 where wboid like '%R0006P%';
```

▶ レポート・バージョンを生成するには

1. プルダウン・メニューから[Generate]->[Reports]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。

5. レポート名 (R0006Pなど)とレポート・バージョン(XJDE0001など)を入力します。



6. [Start]をクリックします。

最初にバージョン・リスト(F983051)と28のデータ辞書項目が生成されます。

7. ステータス・ログで生成状況をチェックします。

レポートの全バージョンとそれに対応するDSEL、DSEQ、PRINTINFOの各オブジェクトが生成されます。

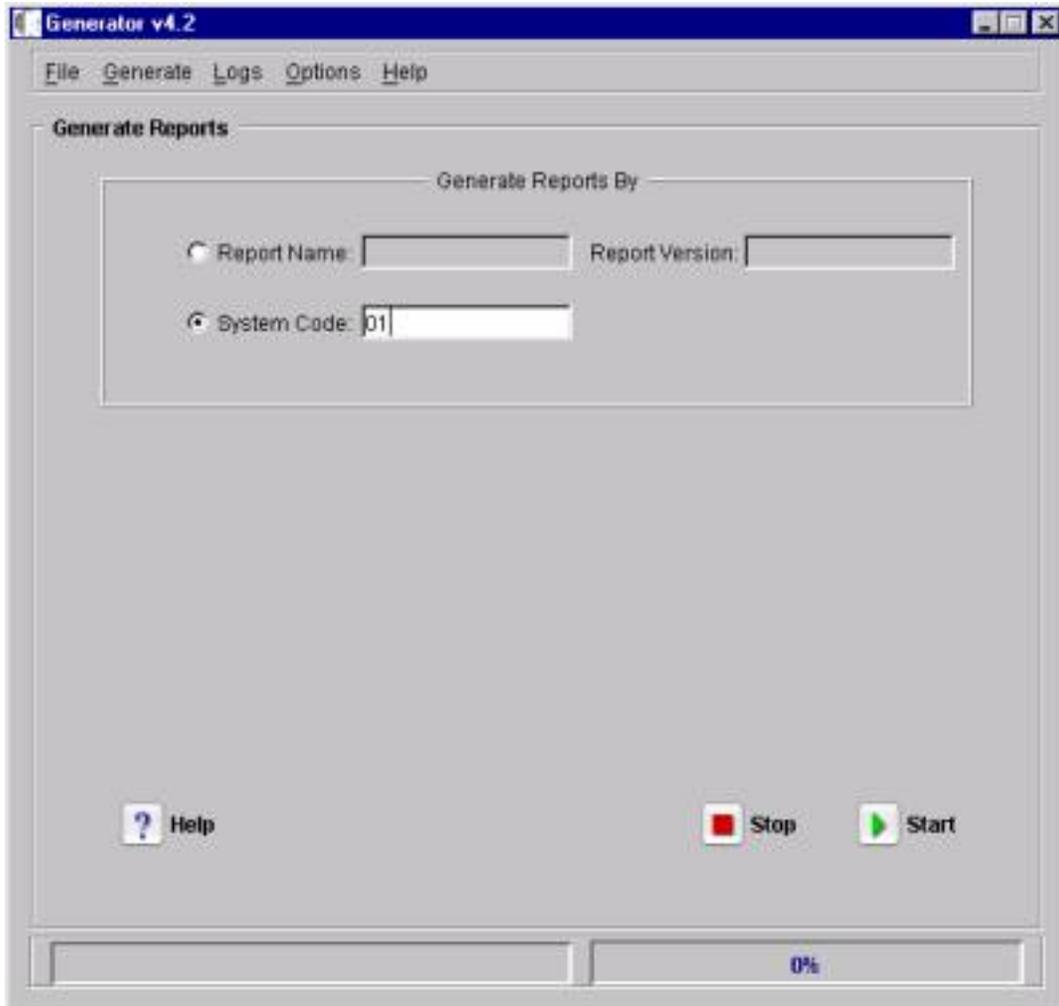
8. ステータス・ログを調べてオブジェクトの存在を確認するか、次のクエリーを使用してデータベースをチェックします。

```
SQL> select wboid from f989999 where wboid like '%R0006P%';
```

▶ システム・コード別にレポートを生成するには

1. プルダウン・メニューから[Generate]->[Reports]を選択します。

2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. [Generate by System Code]を選択し、コード番号を入力します。
6. [Start]をクリックします。



指定したシステム・コードを持つアプリケーションがすべて取り込まれます(たとえば、マスター環境では26のアプリケーションが取り込まれます)。

入力したバージョン名が間違っている場合は、次のエラー・メッセージが表示されます。



レポート名が間違っている場合は、次のエラー・メッセージが表示されます。



7. ステータス・ログとエラー・ログをチェックして、そのシステム・コードに関するレポートがすべて生成されたことを確認します。
8. 次のSQLコマンドを使用して、各レポートがデータベース上にあることを確認します。

```
SQL> select wboid from f989999 where wboid like 'P%W%';
```

NERの生成

▶ NERを生成するには

1. プルダウン・メニューから[Generate]->[NERs]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. NER名(GetObjectDescなど)を入力します。

注意

NER名では、大文字と小文字が区別されます。



6. [Start]をクリックします。
7. ステータス・ログで生成状況をチェックします。
8. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。

```
SQL> select wboid from f989999 where wboid like 'NER_ GetObjectDesc%';
```

データ辞書項目の生成

▶ データ辞書項目を生成するには

1. プルダウン・メニューから[Generate]->[Data Dictionary(データ辞書)]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. データ辞書項目(OBNMなど)が生成されます。



6. ステータス・ログで生成状況をチェックします。
7. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。

```
SQL> select wboid from f989999 where wboid like 'DICT_OBNM%';
```

テーブルの生成

▶ テーブルを生成するには

1. プルダウン・メニューから[Generate]->[Tables(テーブル)]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. テーブル名(F9860など)を入力します。



6. ステータス・ログで生成状況をチェックします。
7. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。
SQL> select wboid from f989999 where wboid like 'TABLE_F9860%';
8. [UTBrowse(UT参照)]を開き、テーブルにデータ辞書項目があることを確認します。
9. ログをチェックして、データ辞書項目が生成されたことを確認します。
10. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。
SQL> select wboid from f989999 where wboid like 'DICT%';
オブジェクト・ライブラリアン・マスター(F9860)には、38のデータ辞書項目が必要です。

ビジネス・ビューの生成

▶ ビジネス・ビューを生成するには

1. プルダウン・メニューから[Generate]->[Business Views(ビジネス・ビュー)]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. ビジネス・ビュー名(V9001Bなど)を入力します。
6. [Start]をクリックします。
7. ステータス・ログで生成状況をチェックします。



8. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。
SQL> select wboid from f989999 where wboid like 'VIEW_V9001B%';
9. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。

```
SQL> select wboid from f989999 where wboid like 'TABLE_%';
```

たとえば、ビジネス・ビューV9001Bには、テーブルF9000とF9001、および78のデータ辞書項目が含まれています。

データ構造体の生成

▶ データ構造体を生成するには

1. プルダウン・メニューから[Generate]->[Data Structures(データ構造体)]を選択します。
2. プルダウン・メニューから[Options]->[Advanced Settings]を選択します。
3. [Error Log]と[Status Log]の左のチェックボックスを有効にして[OK]をクリックします。
4. データベースを切り捨てます。
5. データ構造体名(D9800150Cなど)を入力します。



6. [Start]をクリックします。
7. ステータス・ログで生成状況をチェックします。
8. 次のSQLコマンドを使用して、各オブジェクトがデータベース上にあることを確認します。

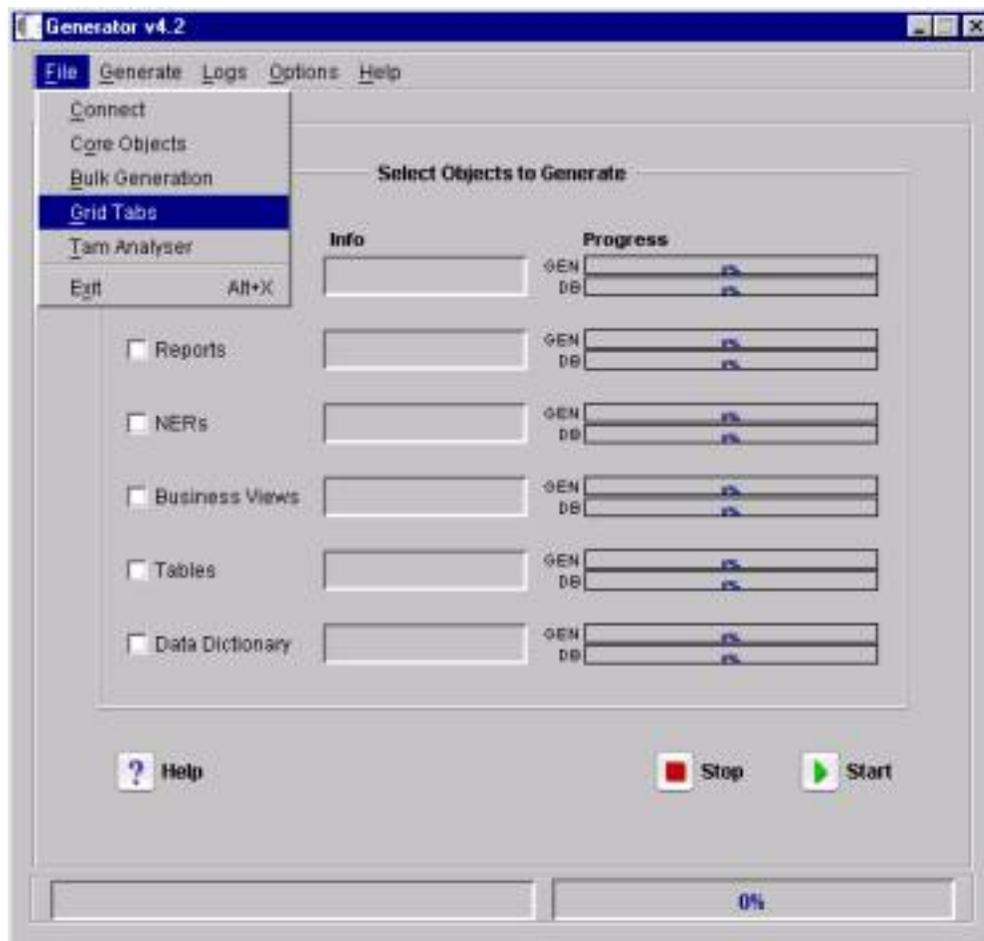
```
SQL> select wboid from f989999 where wboid like 'DSTR_ D9800150C%';
```

グリッド・タブの変換

グリッド・タブでは、ユーザー一時変更テーブル(F98950)内のGDおよびGFレコード・タイプが選択され、バイナリBLOBデータがXML BLOBデータに変換されます。次のタスクを実行して、グリッド・タブを変換します。

▶ グリッド・タブを変換するには

1. プルダウン・メニューから[File]->[Grid Tabs(グリッド・タブ)]を選択します。



[Grid Tabs Conversion(グリッド・タブの変換)]ウィンドウが表示されます。

2. jde.logをチェックして、変換が正常終了したことを確認します。

3. グリッド・タブがアクティブになっていることを確認します。そのためには、アプリケーションを開き、
User Overrides(ユーザー一時変更)がアクティブになっていることを確認します。

この検証が関係するのは、プリスティン(JDEオリジナル)環境で変換を実行する場合のみです。他の環境を変換する場合は、Web上のグリッド・フォーマットがWindowsクライアントと同じであることを確認してください。

Webサーバーの実行

最初にJ.D. Edwards HTML Webサーバーをインストールした後、WebSphereを停止して再起動する必要があります。

ここでは次の内容について説明します。

- ・ [Webサーバーの起動 - ページ \[91\]](#)
- ・ [Webサーバーの停止 - ページ \[91\]](#)
- ・ [Webサーバーへのアクセス - ページ \[91\]](#)
- ・ [J.D. EdwardsポータルでのInherited Trustシステムのセットアップ - ページ \[92\]](#)

ヒント

修正内容を有効にするには、構成を修正するたびに、必ずWebSphereアプリケーション・サーバーを停止して再起動する必要があります。

Webサーバーの起動

次の関連タスクを実行してWebサーバーを起動します。

▶ [Webサーバーをインストールするには \(WebSphere AEの場合\)](#)

1. <WAS_HOME>/binディレクトリにある./adminclient.shユーティリティを使用して[WebSphere Administrative Console]を起動します。
2. [WebSphere Administrative Console]で、[Node(ノード)]ツリーを展開します。
3. アプリケーション・サーバー(AS_JDEdwards_1など)を選択し、[Start]ボタンをクリックします。

Webサーバーの停止

▶ [Webサーバーを停止するには \(WebSphere AEの場合\)](#)

- ・ [WebSphere Administrative Console]で、J.D. Edwardsアプリケーション・サーバー(AS_JDEdwards_1など)を選択して[Stop(停止)]をクリックします。

Webサーバーへのアクセス

Webサーバーには、ネットワークに接続している任意のWebサーバーからアクセスできます。

ヒント

デフォルト以外のホストを使用してWebサーバーにアクセスするには、ポート番号を使用する必要がありますが、index.htmlを指定する必要はありません。

▶ Webサーバーにアクセスするには

1. ブラウザを開き、次のURLを入力してJ.D. Edwardsポータル・クライアントを呼び出します。

`http://<web_server_name>:<port_number>/jde/owportal/`

たとえば、次のように入力します。

`http://jdewebs1.jdedwards.com:81/jde/owportal/index.html`

2. ブラウザを開き、次のURLを入力してJ.D. Edwards HTMLクライアントを呼び出します。

`http://<web_server_name>:<port_number>/jde/owhtml/`

たとえば、次のように入力します。

`http://jdewebs1.jdedwards.com:81/jde/owhtml/`

3. 次のURLを入力してJ.D. Edwards SAWを呼び出します。

`http://<web_server_name>:<port_number>/jde/saw/`

たとえば、次のように入力します。

`http://jdewebs1.jdedwards.com:81/jde/saw/`

エラーは、jas.log、jasdebug.log、stderr.txt、stdout.txtに記録されます。

J.D. EdwardsポータルでのInherited Trustシステムのセットアップ

次のタスクを実行して、Inherited Trustサーブレット用のシークレット・エンタープライズ・キーを設定します。これらのサーブレットは、J.D. Edwardsポータルのログインを、ビジネス・インテリジェンス(BI)やカスタマー・リレーションシップ・マネジメント(CRM)など、他のJ.D. Edwards製品の認証システムとリンクするために使用されます。Inherited Trustシステムを使用すると、ユーザーはJ.D. Edwardsポータルにログインするだけで、このシステムを使用する他の製品に対して自動的に認証され、何度もログインしなくても製品間を移動できます。各システムは、同じシークレット・エンタープライズ・キーおよびユーザーIDを使用して構成する必要があります(ユーザー・パスワードは、異なってもかまいません)。BIとCRMの構成方法については、該当製品の構成ガイドを参照してください。

下記の手順に従い、サンプル・ファイルを使用して、次の2つの新規サーブレットをweb.xmlファイルに追加します。

- ・ oneworld.owportal
- ・ portal

この2つのサーブレットでは、Inherited Trustの構成が定義されます。サーブレットは1つで済む場合もありますが、最大限の互換性を確保するには両方を指定することをお勧めします。

▶ J.D. Edwardsポータル上でInherited Trustシステムをセットアップするには

1. /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/Web-INFディレクトリに移動します。
JASクラス・ファイルとリソース・ファイルがインストール時にデプロイされたディレクトリです。
2. テキスト・エディタで、このディレクトリにあるweb.xmlファイルを開きます。

注意

このファイル内の最初のサーブレットServlet Proxy Servletは修正しないでください。このサーブレットは、インストール・プロセス中にJ.D. Edwardsインストーラにより作成されており、重要な構成設定が含まれます。

3. 下記のサンプル・ファイルをテンプレートとして使用し、web.xmlファイルにoneworld.owportal servletを追加します。このサーブレットは、ファイルの最後の/servlet>終了タグと</web-app>終了タグの間に挿入してください。

次の設定を使用します。

- ・ Servlet ID: Servlet_2

このサーブレットIDが既に存在する場合は、次の連番を選択します。

- ・ servlet-name、display-name、servlet-class:
com.jdedwards.oneworld.owportal.InheritedTrustManagerServlet
- ・ init-parameter id: 排他IDを選択します。
- ・ param-name: secretEnterpriseKey
- ・ param-value: 任意エンタープライズ・キー値を設定します。
- ・ load-on-startup: 0

次のコードは、Servlet_2の例を示します。

```
<servlet id="Servlet_2">
<servlet-name>com.jdedwards.oneworld.owportal.InheritedTrustManagerServlet<
/servlet-name>
<display-name>com.jdedwards.oneworld.owportal.InheritedTrustManagerServlet</display-name>
<servlet-class>com.jdedwards.oneworld.owportal.InheritedTrustManagerServlet</servlet-class>
<init-param id="InitParam_2"> <param-name>secretEnterpriseKey</param-name>
<param-value>key</param-value> <description>For Inherited Trust</description>
</init-param> <load-on-startup>0</load-on-startup> </servlet>
```

4. 次のサンプル設定を使用して、web.xmlにportalサーブレットを追加します。このサーブレットは、ファイルの最後の/servlet>終了タグと</web-app>終了タグの間に挿入してください。

次の設定を使用します。

- Servlet ID: Servlet_3

このサーブレットIDが既に存在する場合は、次の連番を選択します。

- servlet-name、display-name、servlet-class:
com.jdedwards.portal.InheritedTrustManagerServlet
- init-parameter id: 排他IDを選択します。
- param-name: secretEnterpriseKey
- param-value: 任意エンタープライズ・キー値を設定します。
- load-on-startup: 0

次のコードは、Servlet_3の例を示します。

```
<servlet id="Servlet_3">  
<servlet-name>com.jdedwards.portal.InheritedTrustManagerServlet</servlet-name>  
<display-name>com.jdedwards.portal.InheritedTrustManagerServlet</display-name>  
<servlet-class>com.jdedwards.portal.InheritedTrustManagerServlet</servlet-class>  
<init-param id="InitParam_3"> <param-name>secretEnterpriseKey</param-name>  
<param-value>key</param-value> <description>For Inherited Trust</description>  
</init-param> <load-on-startup>0</load-on-startup> </servlet>
```

5. 変更内容をweb.xmlに保存します。
6. Webアプリケーション・サーバーを再起動します。

トラブルシューティングのヒント

URLがAIX上のIBM HTTPサーバーで機能しない場合

現象 WebブラウザがWebSphereのリソースを提供するURLに接続されません。

Webブラウザに接続できない場合は、次のことをチェックします。

- HTTPサーバーが正常に動作し、デフォルト・ページが正常に表示されるかどうかを確認します。
- WebSphereアプリケーションがエラーなしで起動するかどうかを確認します。
- WebSphereのデフォルト・アプリケーション・サーバーが稼働しており、servlet snoopにアクセスできるかどうかを確認します。

- ・ `ibm-web-ext.xmi`ファイルで`fileServingEnable=false`に設定されており、プラグインの再生成が完了しているかどうかを確認します。
- ・ `httpd.conf`ファイルですべての設定が正しく定義されているかどうかを確認します。

上記の項目をすべてチェックしてもURLにアクセスできない場合は、次の処置を行ってください。

処置

1. インデックス・ファイルにURLを明示的に入力します。たとえば、HTMLサーバーの場合は、`"http://your_server_name:your_port/jde/owhtml/index.html"`と入力します。
2. 次のURLを入力します。
 - ・ FRMの場合：`"http://your_server_name:your_port/yourleate/"`
 - ・ HTMLサーバーの場合：`"http://your_server_name:your_port/jde/owhtml/"`

注意

URLの最後に必ず/`(スラッシュ)`を付けてください。

3. これで問題が解決した場合は、次の手順で解決結果を確定します。
4. `httpd.conf`ファイルを開き、エントリ`UseCanonicalName on`をチェックします。
5. このエントリがある場合は、次のように変更します。
`Use CanonicalName off`
これで問題が解決します。
6. `index.html`ファイルを明示的に指定しないか、URLの最後にスラッシュを付けずに、ブラウザにURLを入力します。

Webサーバーのチューニング

ここでは、下記のパラメータをチューニングすることで、Webサーバーのパフォーマンスを最大限まで高める方法について説明します。

- ・ [jas.ini設定のチューニング - ページ \[97\]](#)
- ・ [Java仮想マシンのチューニング - ページ \[98\]](#)
- ・ [エンタープライズ・サーバーのチューニング - ページ \[98\]](#)
- ・ [シリアル化オブジェクト・データベースのチューニング - ページ \[99\]](#)

jas.ini設定のチューニング

パフォーマンスを最適化するために、jas.iniファイルを開いて次の設定を変更します。

[JDENET]

- ・ `maxPoolSize= 50`

この値はデフォルト値の30から変更してください。変更すると、エンタープライズ・サーバーへの接続数が増加します。

- ・ `maxConnection= 150`

作成できるJDBCデータベース接続の最大数です。

[LOGS]

- ・ `Debug= FALSE`
- ・ `Log`および`Debuglog`パラメータをコメント化します。

[PORTALCONFIGURATION]

- ・ `MaxComponentThreadPool=`

ユーザー側にOneWorld/ERPとURIの多数のコンポーネントがあり(ワークスペースごとに10以上)、100人以上の同時ユーザーが予想される場合は、この数値をデフォルトの100から増やします。この値を101以上に増やすと、最初は効率が向上しますが、ある時点を超えると低下します。

Java仮想マシンのチューニング

仮想マシン(VM)の最適なメモリ構成は、実行するVMが単一か複数か、HTMLクライアントがスタンドアロン構成で動作するかポータル経由で動作するかに応じて異なります。

単一VMのチューニング

単一VMが動作するシステム上で、次の操作を行います。

- HTMLクライアントをスタンドアロン構成で実行している場合は、最小および最大ヒープ・サイズを512MBに設定します。
- 1.5GB以上のRAMを搭載したマシン上でポータルを介してHTMLクライアントを実行している場合は、最小および最大ヒープ・サイズを1024MBに設定します。RAMの小さいマシンでは、512MBに設定してください。

複数VMが動作するシステムのチューニング

複数VMが動作しているシステムの場合は、VMごとに1GB以上のRAMを確保し、各VMを次のメモリ・サイズで構成することをお勧めします。

- スタンドアロンHTMLクライアント: 512M
- ポータル経由のHTMLクライアント: 768M

4方向プロセッサの場合、ポータルではオペレーティング・システム用に1GB、HTMLクライアントではオペレーティング・システム用に2Gが残ります。経験上、システム上のすべてのVMIに、システム・メモリの%以上を割り当ててください。

HTMLクライアントの動作はVMごとに512MBで最適になり、HTMLクライアント付きポータルの動作はVMごとに1GBで最適になります。

エンタープライズ・サーバーのチューニング

エンタープライズ・サーバー上で、3~5人のユーザーを予期するように各カーネルを構成します。

DB2®/UDBの場合のみ

初期インストール後にOneWorld/ERP™エンタープライズ・サーバーの構成を修正している場合は、次の構成パラメータをチェックして、OneWorld/ERP™の要件を満たしていることを確認してください。

データベース・マネージャ構成

パラメータ	OneWorld/ERP™のデフォルト設定
query_heap_sz	8000

データベース構成(OWSH9)

パラメータ	OneWorld/ERP™のデフォルト設定
applheapsz	6000
locklist	2000
locktimeout	60

シリアル化オブジェクト・データベースのチューニング

すべてのシリアル化オブジェクトを最初に生成するときには、F989999テーブルとF989998テーブルに多数の行が挿入されます。コストを考慮してデータベースを最適化している場合は、アクセス・パスを最適化するためにテーブル統計を更新する必要があります。

シリアル化オブジェクト・データベースをチューニングするには、データベース管理者は次のいずれかのセクションの手順を実行する必要があります。

- ・ [DB2/UDBのチューニング - ページ \[99\]](#)
- ・ [Oracle9iのチューニング - ページ \[100\]](#)

DB2/UDBのチューニング

DB2/UDBでは、デフォルトでコストベース最適化が使用されるため、統計を更新する必要があります。次のタスクをエンタープライズ・サーバー上で実行します。

▶ DB2/UDBをチューニングするには

1. DB2/UDBのインスタンス所有者 (db2inst1など)としてログインします。
2. [コマンド プロンプト]から、次のコマンドを入力してデータベース・サーバーに接続します。
 - > `db2 connect to JDE9 user username using password`
3. 次のコマンドを入力して統計を更新します。
 - > `db2 "runstats on table PD9.f989999"`
 - > `db2 "runstats on table PD9.f989998"`

注意

上記のコマンドは、環境がPD9の場合です。シリアル化オブジェクトが別の環境にある場合は、PD9を正しい環境で置き換えてください。

このプロセスは、完了までに数分かかることがあります。

Oracle9iのチューニング

Oracleでは、デフォルトでコスト・ベースの最適化が使用されます。ただし、OneWorld/ERPエンタープライズ・サーバーをインストールすると、インストール・プロセスによりデータベースがルール・ベースの最適化に自動的に変更され、チューニングする必要はありません。

次のタスクを実行して最適化の設定をチェックし、コスト・ベースの最適化を使用する場合はデータベースのチューニングを行います。

- ・ [最適化設定の確認 - ページ](#) [100]
- ・ [統計の更新 - ページ](#) [100]

最適化設定の確認

次のどちらかの方法で設定を確認します。

▶ init.oraファイルのパラメータ設定をチェックするには

1. iinit.oraファイルを開き、パラメータ名optimizer_modeを検索します。
値がruleに設定されている場合、最適化モードはルールベースであり、次の統計更新ステップを実行する必要はありません。
2. このパラメータが見つからない場合や、異なる値に設定されている場合は、「[統計の更新 - ページ](#) [100]」のタスクを実行します。

▶ データベースをクエリーするには

1. sqlplusを使用してデータベース・サーバーにログインし、次のステートメントを入力します。

```
select name, value from v$parameter where name = 'optimizer_mode';
```
2. 値がruleの場合、統計を更新する必要はありません。他の値に設定されている場合は、次のタスクを実行して統計を更新します。

統計の更新

1. 次のコマンドを入力してデータベース・サーバーにログオンします。
> sqlplus username@connection_string/password
2. 次のコマンドを入力して統計を更新します。
> analyze table PD9.f989999 compute statistics;
> analyze table PD9.f989998 compute statistics;

注意

上記のコマンドは、環境がPD9の場合です。他の環境の場合は、PD9を正しい環境で置き換えてください。

このプロセスは、完了までに数分かかることがあります。

付録 A. jas.iniファイルのパラメータと値

jas.iniファイルでは、Webサーバーからエンタープライズ環境に通信できるように、COMおよび他のWebサーバー固有のプロパティが定義されています。また、このファイルはインストールの成否を左右するものです。重要な設定が正しくなかったり空白になっていると、Webサーバーは動作しません。ここでは、重要な設定をセクション別に表形式で説明します。

Javaサーバーのインストーラを使用してWebサーバーをインストールするときには、jas.iniファイルが自動的に更新されます。ただし、インストール後に、テキスト・エディタを使用してjas.ini設定を修正できます。jas.iniファイルの機能は、OneWorld/ERPエンタープライズ・サーバーのjde.iniファイルに似ています。

注意

ERP以降は、jas.iniファイルの多数の設定が新規のファイルJDBj.iniに移動しています。この新規ファイルについては、「Appendix C: jdbj.iniファイルのパラメータと値」を参照してください。

jas.iniファイルを構成するには、次の操作を実行する必要があります。

- ・ インストール環境に固有のサーバー名を[SECURITY]および[SERVER]セクションのパラメータに追加します。
- ・ パス名固有の各種設定のパス名を確認します。J.D. Edwardsの推奨に従っていれば、これらのパスを大幅に変更する必要はありません。
- ・ TCP/IP接続用のポート番号を確認します。

ヒント

通常は、次の情報が含まれるようにiniファイルにコマンド行を追加してください。

- ・ 現行のJASパッケージのバージョン(CDIに貼付されているラベルを参照)
- ・ ファイルの手動更新日
- ・ 更新者名

次の表は、Webサーバーで使用されるjas.ini設定をまとめたものです。

セクション	説明
[DB SYSTEM SETTINGS]	ERP9以降、これらの設定は新規のファイルJDBj.iniの次のセクションにあります。 [JDBj-BOOTSTRAP SESSION]

	[JDBj-BOOTSTRAP DATA SOURCE]
[SECURITY] - ページ [105]	OneWorld ERPセキュリティ・サーバー情報が含まれています。
[OWWEB] - ページ [108]	Webサーバー固有の設定。
[PORTALCONFIGURATION] - ページ [112]	<p>このオプション・セクションは、標準jas.iniファイルには含まれていません。</p> <p>J.D. Edwardsポータルオプション設定が含まれています。このセクションに示されている設定はデフォルト設定であり、jas.iniの一部として組み込む必要はありません。つまり、jas.iniで値を指定しなければ、ポータルではこのセクションに示されているデフォルト設定が自動的に使用されます。</p> <p>ただし、jas.iniから読み取れる値を修正するサイトでは、そのカスタム・インストールに合わせて正しい値を入力および修正する必要があります。また、将来のインストールではカスタム修正が保存されない可能性があるため、アップグレード前にjas.iniのコピーを保存しておく必要があるので注意してください。</p>
[CACHE] - ページ [121]	HTTPセッションの設定
[DB CONNECTION POOL]	<p>DB接続プールでは、データ・ソース、ユーザーID、およびパスワードに基づいて接続グループが管理されます。この設定を使用すると、管理者は使用可能にする接続数を調整できます。使用されない接続を維持する期間は、Connectionプロパティを使用してCACHEセクションで指定します。また、このセクション内で構成されるプール・クリーナー・プロセスにより、プールの接続が指定した間隔でスケジュールに従ってチェックされます。プール・クリーナーにより、適用可能なSQLステートメントにより無効と判断された接続が置換されます。デフォルトでは、プール・クリーナーは無効になっています。有効化するには、ValidationString.xプロパティを少なくとも1つは指定する必要があります。</p> <p>ERP9以降は、このセクションはJDBj.iniファイルに移動され、名前が変更されています。この新規セクションについては、「[JDBj-CONNECTION POOL]」を参照してください。</p>
[JDBC Drivers]	ERP9以降は、このセクションはJDBj.iniファイルに移動され、名前が変更されています。この新規セクションについては、「[JDBj-JDBC DRIVERS]」を参照してください。
[JDBC URL]	ERP9以降は、このセクションはJDBj.iniファイルに移動され、名前が変更されています。この新規セクションについては、「[JDBj-SPEC DATA SOURCE]」を参照してください。
[SERVER COMPONENTS] - ページ [123]	Webサーバー・パッケージをロードします。このセクションは変更しないでください。
[JDENET] - ページ [123]	WebサーバーでOneWorld/ERPエンタープライズ・サーバーと通信するための値を指定します。

[SERVER] - ページ [124]	WebサーバーでOneWorld/ERPエンタープライズ・サーバーからデータ辞書エラーの記述を取り込むための値を指定します。
[LOGIN] - ページ [125]	ログインとローカル・ディレクトリの設定が含まれています。
[LOGS] - ページ [126]	デバッグ・ログを有効化して、ログ・ファイルのロケーションを指定します。
[TRANSACTIONS] - ページ [127]	JASでトランザクション処理をカスタマイズするための構成設定が含まれています。
[JAS Instance]	このオプション・セクションは、標準jas.iniファイルには含まれていません。 リダイレクタ機能を使用して複数のJASインスタンスを定義するための設定が含まれています。複数のインスタンスを定義して、パフォーマンスを改善し、多数のユーザー向けにインストール環境を拡張できます。複数のJavaインスタンス間でユーザーを割り当てるためのデフォルト・メカニズムは「ラウンド・ロビン」、つまり順次選択方式です。
[JAS Weight]	このオプション・セクションは、標準jas.iniファイルには含まれていません。 リダイレクタ機能を使用して複数のJASインスタンスのロード・バランシングを有効化するための設定が含まれています。
[Redirector]	このオプション・セクションは、標準jas.iniファイルには含まれていません。 使用可能なWebサーバーがない場合の、リダイレクタによる転送先を指定します。通常、これは説明を提供するHTMLページです。
[WEB GUI] - ページ [128]	このセクションではWebのGUIインターフェイスのカスタマイズ方法とサービス・パック22で新しく加わった内容について説明します。

[SECURITY]

次の表は、Webサーバーの基本機能を提供する重要な設定を示します。

[SECURITY]		
パラメータ	推奨設定	説明
DefaultEnvironment=	JPD9	デフォルトのOneWorld/ERP環境を指定します。
SecurityServer=	サイト依存の変数	OneWorld/ERPエンタープライズ・サーバーのインストール用に定義したセキュリティ・サーバーの名称

[SECURITY]		
パラメータ	推奨設定	説明
		を指定します。
Factory=	<ul style="list-style-type: none"> ・ J.D. Edwards ERPと統合されているポータルの場合: com.jdedwards.system.security.SecurityServerProxyFactory ・ ポータル単体の場合: com.jdedwards.system.security.SecurityServerDBFactory ・ セキュリティ・サーバーとしてiSeriesを使用するポータル単体の場合: com.jdedwards.system.security.SecurityServerAS400Factory 	<p>ユーザー名とパスワードの検証に使用するメソッドを指定します。</p> <p>メソッドはポータル構成に依存するので注意してください。Factoryの値を指定しないと、デフォルトでJ.D. Edwards ERPと統合されているポータルの値に設定されます。</p>

次の表は、クリティカルではない設定を示します。

[SECURITY]		
パラメータ	推奨設定	説明
NumServers=	1	<p>このWebサーバーにサインオンするユーザーに使用可能として定義する、OneWorld/ERPセキュリティ・サーバーの合計数を指定します。</p> <p>このパラメータを指定しないか、ブランクにした場合、デフォルト値は1で、サインオンはjas.iniの[SECURITY]セクションのSecurityServer=パラメータで定義されているプライマリ・セキュリティ・サーバーによって処理されます。</p>
SecurityServerN=	サイト依存の変数	<p>セカンダリ・セキュリティ・サーバー名を指定します。ユーザーがプライマリ・サーバーにサインオンできない場合に、有効なセカンダリ・サーバーへのフェールオーバーにサインオンする場合は、複数のセカンダリ・サーバーを定義できます。</p>

[SECURITY]		
パラメータ	推奨設定	説明
		<p>Nに有効な値は1～nです。</p> <ol style="list-style-type: none"> 最初のセカンダリ・セキュリティ・サーバーを定義します。 第2のセカンダリ・セキュリティ・サーバーを定義します。 <p>n 順次アクセスされるセキュリティ・サーバーを必要な数だけ定義します。</p>
UserLogonCookie=	FALSE	<p>ユーザー・サインオン情報がHTMLクライアント・マシン上の暗号化Cookieに保存されるかどうかを定義します。この情報には、ユーザー名、パスワード、および環境が含まれます。</p> <p>有効な値は次のとおりです。</p> <p>TRUE ユーザー情報は、ログイン画面に自動入力される暗号化Cookieに保存されます。</p> <p>DIRECT ユーザーはCookie内のログイン情報にアクセスし、ログイン画面をバイパスできます。</p> <p>FALSE ユーザー情報は暗号化Cookieに保存されません。JASリダイレクタ使用時には、この設定を使用する必要があります。</p>
CookieLifeTime unit is	"day"	CookieLifeTime=パラメータで使用する時間の単位を指定します。
CookieLifeTime=	7	Cookieが失効するまでの期間を指定します。この値の単位には、CookieLifeTime unit isパラメータの値が使用されます。
Data Source=	サイト依存の変数	<p>セキュリティ・テーブルへの接続に使用するデータ・ソースを指定します。</p> <p>この設定は、ポータル単体の場合</p>

[SECURITY]		
パラメータ	推奨設定	説明
		にのみ使用されます。Data Sourceを指定した場合は、UserとPasswordも指定する必要があります。
User=	"JDE"	データベース・ユーザーIDを指定します。 この設定は、ポータル単体の場合にのみ使用されます。
Password=	"JDE"	データベース・パスワードを指定します。 この設定は、ポータル単体の場合にのみ使用されます。

[Security]セクションの構成は、ユーザー認証に使用するセキュリティ・サーバーのタイプに応じて異なります。次に基本的な構成の例を示します。

OneWorldセキュリティ・サーバーを使用する場合

```
[Security] Factory=com.jdedwards.system.security.SecurityServerProxyFactory
NumServers=3
SecurityServer=JDED
SecurityServer1=JDEC
SecurityServer2=corowhp
```

データベース・セキュリティ・サーバーを使用する場合 (ポータル単体)

```
[Security] Factory=com.jdedwards.system.security.SecurityServerDBFactory
DataSource=System-Security
User=JDE
Password=JDE
```

ポータル単体でデータベース・セキュリティ・サーバーを指定すると、システムではデータベース・プロキシ・ユーザーのJava実装が使用されます。

iSeriesセキュリティ・サーバーを使用する場合 (ポータル単体)

```
[Security] Factory=com.jdedwards.system.security.SecurityServerAS400Factory
SecurityServer=JDEC
```

SecurityServer設定でiSeriesセキュリティ・サーバー名を指定します。

[OWWEB]

次の表は、Webサーバーの基本機能を提供する重要な設定を示します。

[OWWEB]		
パラメータ	推奨設定	説明
PathCodes=	('JD9','DV9', 'PY9','PD9')	このキーでは、標準J.D. Edwardsパス・コードをリストします。標準パス・コードを使用する場合、これらの値を修正する必要はありません。このキーを修正する場合は、OneWorld/ERPに有効なパス・コードを指定する必要があります。 デフォルト環境はパス・コードで示されます。この値が空の場合、使用可能なすべての環境が特定のユーザーまたはグループに使用可能になります。各パス・コードは、一重引用符で囲んでカンマで区切ります。
MO QUEUE=	サイト依存のパス	Webサーバー上のメディア・オブジェクト・ディレクトリの位置を識別します。このパスは、Webブラウザ用の仮想パス/jde/moqueue/に変換する必要があります。 Webサーバー・プロセスには、このパスの読取り権限と作成権限が必要です。 注意 このディレクトリの作成を必要とする場合があります。 デフォルト値は次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/moqueue
FtpPort=	21	FTP用のデフォルト・ポートを指定します。
FtpUsr=	anonymous	メディア・オブジェクト・ファイル・サーバーへのFTPアクセスに使用するユーザーIDを指定します。
FtpPwd=	anonymous	メディア・オブジェクト・ファイル・サーバーへのFTPアクセスに使用するパスワードを指定します。
UseMOWinNT Share=	FALSE	WebサーバーでWindowsのファイル共有メカニズムの代わりにFTPアクセスを使用するように指定します。
InYourFace Error=	TRUE	Webサーバーで、Webインターフェイス関連のエラー・メッセージを現行フォームの上部に表示するように指定します。 この設定をFALSEに変更すると、代わりに右上隅に小さい赤アイコンが表示されます。このアイコンをクリックすると、エラー・メッセージのテキストが表示されます。

次の表は、クリティカルではない設定を示します。

パラメータ	推奨設定	説明
LargeLoginImage	/jde/share/images/ jdelogin_large.gif	<p>HTTPサーバーのドキュメント・ルートへの相対URLを指定します。標準デスクトップ・ブラウザ (IE 5、Netscape 6) を使用している場合は、このイメージがログイン・ページに表示されます。</p> <hr/> <p>ヒント</p> <p>ポータル単体の場合は、別のグラフィックを指定できます。次に例を示します。</p> <p>/jde/share/images/ portallogin_large.gif</p>
SmallLoginImage	/jde/share/images/ jdelogin_small.gif	<p>HTTPサーバーのドキュメント・ルートへの相対URLを指定します。普及型デバイス (PocketPC、Windows CE) を使用している場合は、このイメージがログイン・ページに表示されます。</p> <hr/> <p>ヒント</p> <p>ポータル単体の場合は、別のグラフィックを指定できます。次に例を示します。</p> <p>/jde/share/images /portallogin_small.gif</p>
F0005Prefix=	DR	ユーザー定義コード・テーブル (F0005) のカラム・プレフィックス。
F0004Prefix=	DT	ユーザー定義コード・タイプ・テーブル (F0004) のカラム・プレフィックス。
MaxUser=	100	OneWorld/ERP (インターネット) ユーザーの最大数。
PrintImmediate=	FALSE	FALSE に設定すると、エンタープライズ・サーバーでは PDF ファイルが生成されるだけです。TRUE に設定すると、エンタープライズ・サーバーでは PDF ファイルが生成され、PostScript に変換されます。UBE ジョブの PDL または ライン 出力です。
KeepUBE=	TRUE	FALSE に設定すると、UBE ジョブの完了時に

パラメータ	推奨設定	説明
		JDE.LOGおよびJDEDEBUG.LOGファイルが削除されます。TRUEに設定すると、UBEジョブの完了時にJDE.LOGおよびJDEDEBUG.LOGファイルが保存されます。
UBEQueue=	"QB9"	UBEジョブが投入されるバッチ待ち行列。
SystemDateFormat=	MDE	システム日付の形式を指定します。
SystemDateSeparator=	/	システム日付区切り文字を指定します。
HelpPath=	/jde/owhelp/e	<p>ヘルプ・ファイルのディレクトリ・パスを指定します。この設定の構文は次のとおりです。</p> <p>http://[machine]/[path]/</p> <p>HelpPathの設定を指定しないか、デフォルト設定の/jde/owhelp/eを使用すると、WebサーバーではJASサーバーの現行インスタンスに基づいてヘルプ・パスが作成されます。</p> <p>たとえば、WebサーバーのURLが次の場合、</p> <p>http://machinename:82/</p> <p>Webサーバーではヘルプ・パスのURLが次のように想定されます。</p> <p>http://machinename:82/jde/owhelp/e</p> <hr/> <p>注意</p> <p>eは英語版ヘルプ・ファイルのデフォルト・パスです。他の言語を使用する場合は、"e"を使用言語を表す正しい記号で置き換えて、このディレクトリにその言語のヘルプ・ファイルをインストールしていることを確認してください。</p> <hr/>
OWJRNL=	OWJRNL	<p>この設定は、iSeriesプラットフォーム上で実行中のWebサーバーにのみ適用可能です。AS/400のDB2データベース・レコードを更新するために手動コミットを有効化している場合は、この設定を使用して、Webサーバーで該当するiSeriesテーブルのJOURNALを有効化できます。</p> <p>この設定には、STRJOURNALストアド・プロシージャ用のAS/400ライブラリの名称を指</p>

パラメータ	推奨設定	説明
		定する必要があります。
AutoPilotIDs=	FALSE	この設定では、JavaサーバーでJ.D. Edwards AutoPilotツールを使用してHTMLクライアント・アクションのスクリプトを作成する場合に役立つIDが生成されるかどうかを制御します。有効な値は次のとおりです。 FALSE JavaサーバーではAutoPilot IDは生成されません。 TRUE JavaサーバーでAutoPilot IDが生成されます。
AnonAccess=	TRUE	匿名ユーザー・アクセスを有効にします。
DefaultEnvironment=	サイト依存の変数	デフォルトのログイン環境。基本認証と併用されます。
InitialLanguageCode=	EN	初期ユーザー言語のISO言語コード。
LogoutProcessTimeout=	90	ユーザーがアクティブ化したビジネス関数が、ログアウト後も引き続き実行される秒数。
PPCSetDefaultFocus=	TRUE	この設定により、PocketPCデバイスの画面に仮想キーボードが表示されるかどうかは確定されます。デフォルト値はFALSEです。 この設定がFALSEの場合、PocketPCデバイスのフォームにはデフォルトのフォーカスが設定されず、画面にキーボードは表示されません。 この設定がTRUEの場合は、PocketPCにデフォルトのフォーカスが設定され、画面に仮想キーボードが表示されます。キーボードでPocketPCデバイスの貴重な領域が使用されますが、一部のカスタマーはスキャナーを使用して画面にデータを入力する際にデフォルトのフォーカスを必要とします。PocketPCデバイスでデフォルトのフォーカスが必要な場合は、この値をTRUEに設定してください。

[PORTALCONFIGURATION]

このセクションには、J.D. Edwardsポータルオプション設定が含まれています。

Cisco LocalDirectorなどのロード・バランサを使用している場合は、localhost設定を指定する必要があります。

Portal Component Importerを使用している場合は、backup、jde、およびservlet設定を指定する必要があります。

これらの設定を除き、このセクションに示されるのはすべてデフォルト設定であり、jas.iniの一部として組み込む必要はありません。つまり、jas.iniで値を指定しなければ、ポータルではこのセクションに示されているデフォルト設定が自動的に使用されます。

ただし、jas.iniから読み取れる値を修正するサイトでは、そのカスタム・インストールに合わせて正しい値を入力および修正する必要があります。また、将来のインストールではカスタムな変更内容が保存されない可能性があるため、アップグレード前にjas.iniのコピーを保存しておく必要があるので注意してください。

次の設定は、J.D. Edwardsポータルのインポート/エクスポート・コンポーネントを構成するには必須です。

[PORTALCONFIGURATION]		
パラメータ	推奨設定	説明
jde	/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war	JASインストール時に作成されるWebディレクトリ。Component Importer/Exporterは、このディレクトリで.html、.gif、.jpgファイルなどのHTMLリソースの検索と書き込みを行います。
servlet	/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/classes	ポータル・サーブレットが常駐し、アップデート・センターからのコンポーネントが格納されるディレクトリ。Component Importer/Exporterは、このディレクトリでサーブレットの検索と書き込みを行います。このディレクトリは、JAS管理者が作成し、JDE WebアプリケーションのCLASSPATHに組み込まれるように構成する必要があります。
backup	/backup	Component Importerによって上書きされるファイルの保存場所。これにより、上書きされたファイルのバックアップが提供されます。 このディレクトリはCLASSPATHに指定せず、Webサーバーからは参照可能にしないでください。
OneWorld	TRUE	ポータルに対して、J.D.

[PORTALCONFIGURATION]		
パラメータ	推奨設定	説明
Present=		Edwards ERP製品がWebサーバーと統合されているかどうかを指示します。この設定をjas.iniファイルで指定しなければ、値TRUEが使用されます。この値をFALSEに設定すると、OneWorld/ERPシステムが必要であることを示すフラグが付いたポータル・コンポーネントは実行されません。

次の表は、クリティカルではない設定を示します。

パラメータ	推奨設定	説明
Admin (SP 13.1以上の場合)	サイト依存の変数、デフォルト値なし	ユーザーIDを 文字で区切ったリストを指定します。これらのユーザーは、リレーションシップに関係なくすべてのコンポーネントとワークスペースを管理できます。
cache_workspace_purge	3600000	ワークスペースが削除される前にアクセスされないままキャッシュに残っているミリ秒数。0(ゼロ)に設定すると、ワークスペースはキャッシュから除去されません。 この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。
cache_workspace_expire	900000	期限付きコンポーネントについて、ワークスペースが削除されるまでの、最後にロードされたタイムスタンプに加算されるミリ秒数。コンポーネントを期限付きにするには、 public long getLastLoadedTimestamp();を実装します。ユーザーがアクセスしない限り、ワークスペースは期限切れになりません。0(ゼロ)に設定すると、ワークスペースはキャッシュ内で期限切れになりません。

		この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。
cache_workspace_timeout	300000	項目が除去対象か期限切れかをキャッシュ内でチェックするまでシステムが待機するミリ秒数。この値が小さいほど、メモリ使用効率は向上しますが、キャッシュが低速になります。0(ゼロ)に設定すると、ワークスペースはキャッシュから除去されず、期限切れにもなりません。 この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。
cache_workspace_refresh	0	すべてのワークスペースがキャッシュから削除されるまでのミリ秒数。 この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。
cache_component_purge	3600000	コンポーネントが削除される前にアクセスされないままキャッシュに残っているミリ秒数。0(ゼロ)に設定すると、コンポーネントはキャッシュから除去されません。 この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。
cache_component_expire	900000	期限付きコンポーネントについて、コンポーネントが削除されるまでの、最後にロードされたタイムスタンプに加算されるミリ秒数。コンポーネントを期限付きにするには、 public long getLastLoadedTimestamp();を実装します。ユーザーがアクセスしない限り、コンポーネントは期限切れになりません。0(ゼロ)に設定すると、コンポーネントはキャッシュ内で期限切れになりません。 この値を変更すると、ポータルのパフォーマンスが大幅に変化する可

		<p>能性があります。この値を変更する場合は注意してください。</p>
cache_component_timeout	300000	<p>項目が除去対象か期限切れかをキャッシュ内でチェックするまでシステムが待機するミリ秒数。この値が小さいほど、メモリ使用効率は向上しますが、キャッシュが低速になります。0(ゼロ)に設定すると、コンポーネントはキャッシュから除去されず、期限切れにもなりません。</p> <p>この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。</p>
cache_component_refresh	0	<p>すべてのコンポーネントがキャッシュから削除されるまでのミリ秒数。</p> <p>この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。</p>
cache_itrust_purge	60000	<p>継承によるトラスト・セッションが削除される前にアクセスされないままキャッシュに残っているミリ秒数。0(ゼロ)に設定すると、継承によるトラスト・セッションはキャッシュから除去されず、期限切れにもなりません。</p> <p>この値を変更すると、ポータルのパフォーマンスとセキュリティが大幅に変化する可能性があります。この値を変更する場合は注意してください。</p>
cache_itrust_expire	0	<p>期限付きコンポーネントについて、継承によるトラスト・セッションが削除されるまでの、最後にロードされたタイムスタンプに加算されるミリ秒数。コンポーネントを期限付きにするには、public long getLastLoadedTimestamp();を実装します。ユーザーがアクセスしない限り、継承によるトラスト・セッションは期限切れになりません。</p> <p>このセクションは、J.D. Edwardsから特に指示されない限り修正しないでください。</p>

cache_itrust_timeout=	30000	項目が除去対象か期限切れかをキャッシュ内でチェックするまでシステムが待機するミリ秒数。この値が小さいほど、メモリ使用効率は向上しますが、キャッシュが低速になります。0(ゼロ)に設定すると、継承によるトラスト・セッションはキャッシュから除去されず、期限切れにもなりません。
cache_itrust_refresh	0	すべての継承によるトラスト・セッションがキャッシュから削除されるまでのミリ秒数。
cache_entbutton_purge	3600000	<p>エンタープライズ・ナビゲーション・バーのボタンが削除される前にアクセスされないままキャッシュに残っているミリ秒数。0(ゼロ)に設定すると、エンタープライズ・ナビゲーション・バーのボタンはキャッシュから除去されません。</p> <p>この値を変更すると、ポータルのパフォーマンスが大幅に変化する可能性があります。この値を変更する場合は注意してください。</p>
cache_entbutton_expire	0	<p>期限付きコンポーネントについて、エンタープライズ・ナビゲーション・バーのボタンが削除されるまでの、最後にロードされたタイムスタンプに加算されるミリ秒数。コンポーネントを期限付きにするには、public long getLastLoadedTimestamp();を実装します。0(ゼロ)に設定すると、エンタープライズ・ナビゲーション・バーのボタンはキャッシュ内で期限切れになりません。</p> <p>0(ゼロ)に設定すると、エンタープライズ・ナビゲーション・バーのボタンはキャッシュ内で期限切れになりません。</p>
cache_entbutton_timeout	900000	項目が除去対象か期限切れかをキャッシュ内でチェックするまでシステムが待機するミリ秒数。この値が小さいほど、メモリ使用効率は向上しますが、キャッシュが低速になります。0(ゼロ)に設定すると、エンタープライズ・ナビゲーション・バーのボタンはキャッシュ内で期限切れに

		ならず、除去もされません。
cache_entbutton_refresh	0	エンタープライズ・ナビゲーション・バーのボタンがすべてキャッシュから削除されるまでのミリ秒数。
pagegreeting	Welcome to your Portal	新規ワークスペースを追加するときのデフォルト・ページに表示される挨拶メッセージ。この設定のデフォルト値はブランクです。
localhost	サイト依存の変数	Cisco LocalDirectorルーターを使用している場合は、この設定を組み込む必要があります。Webサーバーのポートが80以外の場合は、ポートを指定する必要があります。構文は次のとおりです。構文は ip_address:port で、ip_address:port はローカルの OneWorld/ERP Java サーバー・マシン (Cisco LocalDirector マシンではなく) を指します。ポータル・コードでマシン要求の出所を判別し、それに従って応答をルーティングできるようにするには、この設定が必須です。 たとえば、次のように入力します。 10.0.110.79:85
styleurl	/jde/owportal/portal.css	ポータル・スタイルシートのURI。 相対URLと完全修飾URLの両方が有効です。
hlpimg	/jde/owportal/images/ help2.gif	コンポーネント・ツールバーのヘルプ・イメージ名。 相対URLと完全修飾URLの両方が有効です。
perimg	/jde/owportal/images/ edit2.gif	/jde/images/edit2.gif など、デフォルトのカスタマイズ・アイコン・ファイルのパスと名称。 相対URLと完全修飾URLの両方が有効です。
maximg	/jde/owportal/images/ maximize2.gif	/jde/images/maximize2.gif など、デフォルトの拡張アイコン・ファイルのパスと名称。 相対URLと完全修飾URLの両方が

		有効です。
minimg	/jde/owportal/images/ minimize2.gif	/jde/images/minimize2.gifなど、デフォルトの連絡先アイコン・ファイルのパスと名称。 相対URLと完全修飾URLの両方が有効です。
resimg	/jde/owportal/images/ restore2.gif.	/jde/images/restore2.gifなど、デフォルトの復元アイコン・ファイルのパスと名称。 相対URLと完全修飾URLの両方が有効です。
retimg	/jde/owportal/images/ return.gif.	/jde/images/return.gifなど、デフォルトのリターン・アイコン・ファイルのパスと名称。 相対URLと完全修飾URLの両方が有効です。
ShowCurrentEnvironmentRole	FALSE	環境表示。TRUEに設定すると、ワークスペース・ナビゲーション・バーに現行の環境が表示されます。
ShowSignin	TRUE	ユーザーが匿名でログインしたときに、ワークスペース・ナビゲーション・バーに標準ログインのハイパーリンクを表示します。
corplogourl	/jde/owportal/images/jdelogo.gif	デフォルトの会社ロゴのURL。このイメージは、現行のワークスペースでイメージが指定されていない場合に使用されます。 相対URLと完全修飾URLの両方が有効です。
corplogolinkurl		デフォルトの会社ロゴのハイパーリンクのURL。このリンクは、現行のワークスペースでリンクが指定されていない場合に使用されます。 この設定にはデフォルト値がありません。 相対URLと完全修飾URLの両方が有効です。
DataMigrationHasOccurred	これはシステム設定です。この設	B9のプリスティン(JDEオリジナル)

	定の修正やjas.iniファイルへの追加は行わないでください。	コンポーネントとワークスペースのデータが更新されている場合はTRUE。
DefaultWorkspace		他のワークスペースが指定されていない場合や、DefaultWorkspaceOnlyがTRUEに設定されている場合に表示されるワークスペース。 この値は、ワークスペース名ではなくワークスペースIDに設定してください。 このパラメータの設定には大文字を使用します。 この設定にはデフォルト値がありません。
DefaultWorkspaceOnly	FALSE	デフォルトのワークスペースへのアクセスのみを許可します。
ForceDefaultWorkspace	TRUE	DefaultWorkspaceを入力してForceDefaultWorkspaceをTRUEに設定すると、ユーザーにはログイン時に最初のワークスペースとして常にデフォルト・ワークスペースが表示されます。
edting	/jde/owportal/images/ edit2.gif	デフォルトの編集アイコンのURL。編集アイコンは、ユーザーがオブジェクトを変更するためのボタンに使用されます。 相対URLと完全修飾URLの両方が有効です。
NumberOfIcons	34	エンタープライズ・ナビゲーション・バーのアイコンの一時変更数。
colorscheme1	Default #FFFFFF #00009C #0063CE #080029 #CECECE #FFFFFF #636363 #FFFFFF background.jpg /jde/owportal/owportal.css	カラー・コードを示す区切り付き文字列。文字列は、名称、背景色、ツールバーの色、ツールバーのツールの色、固定領域の色、枠の色、テキストの色、メニューの色、メッセージの色、一番手前の背景イメージの色、およびスタイルシートのURLからなっています。
colorscheme2	Springtime #FFFFFF #218C7B #84BDB5 #006B63 #CECECE #FFFFFF #FF6B29 #FFFFFF	カラー・コードを示す区切り付き文字列。文字列は、名称、背景色、ツールバーの色、ツールバーのツ

	springtimebkgd.gif /jde/owportal/portal.css	ルの色、固定領域の色、枠の色、テキストの色、メニューの色、メッセージの色、一番手前の背景イメージの色、およびスタイルシートのURLからなっています。
colorscheme3	Bluedot #FFFFFF #4A5A9C #849CC6 #001873 #CECECE #FFFFFF #737BB5 #FFFFFF bluedotbkgd.gif /jde/owportal/portal.css	カラー・コードを示す区切り付き文字列。文字列は、名称、背景色、ツールバーの色、ツールバーのツールの色、固定領域の色、枠の色、テキストの色、メニューの色、メッセージの色、一番手前の背景イメージの色、およびスタイルシートのURLからなっています。
colorscheme4	Techno #FFFFFF #006363 #739C9C #004242 #CECECE #FFFFFF #B5C6C6 #FFFFFF technobkgd.gif /jde/owportal/portal.css	カラー・コードを示す区切り付き文字列。文字列は、名称、背景色、ツールバーの色、ツールバーのツールの色、固定領域の色、枠の色、テキストの色、メニューの色、メッセージの色、一番手前の背景イメージの色、およびスタイルシートのURLからなっています。
colorscheme5	Cityscape #FFFFFF #FF9C00 #FFC66B #000063 #CECECE #FFFFFF #636363 #FFFFFF cityscapebkgd.gif /jde/owportal/portal.css	カラー・コードを示す区切り付き文字列。文字列は、名称、背景色、ツールバーの色、ツールバーのツールの色、固定領域の色、枠の色、テキストの色、メニューの色、メッセージの色、一番手前の背景イメージの色、およびスタイルシートのURLからなっています。

[CACHE]

各キーで指定する値はミリ秒単位なので注意してください。たとえば、60000ミリ秒は1分です。

次の設定は、Webサーバーの基本機能には影響しません。

注意

ERP9までこのセクションにあった設定の多くは、JDBj.iniファイルの次のセクションに移動しています。

- ・ [JDBj-CONNECTION POOL]
 - ・ [JDBj-RUNTIME PROPERTIES]
-

パラメータ	推奨設定	説明
UserSession=	1200000	<p>非アクティブなユーザー・セッションがキャッシュからクリーンアップされるまでのミリ秒数。このユーザー・セッションはJ.D. Edwardsコードのサーバー側で管理され、WebSphereのHTTPセッションとは異なります。UserSessionの設定値を、WebSphere HTTPセッションのタイムアウト(Invalidate timeout)設定値の2倍にすることをお勧めします。</p> <hr/> <p>注意</p> <p>この値は、JavaサーバーがWebSphereより先にタイムアウトにならないように、WebSphereの[Invalidate Time]設定より大きい値に設定する必要があります。WebSphereでの[Invalidate Time]設定をチェックするには、[WebSphere Administrative Console]を開き、アプリケーション・サーバー(AS_JDEdwards_1)までノードを展開します。次に、Session Manager Servicesノードをクリックして[Advance]タブをクリックします。[Invalidate Time]が有効かされ、正しい値(分単位)に設定されていることを確認してください。</p> <hr/> <p>デフォルト値: 1200000</p> <p>UserSessionには、常にWebSphereでのHTTPセッション・タイムアウトの[Invalidate Timeout]設定より大きい値を設定する必要があります。</p> <p>変更の効果: [HTTP Session timeout]の値に達すると、JASはキャッシュからUserSessionを削除するように指示する通知を受け取ります。UserSession値を変更しても、この値がWebSphereでの[HTTP session timeout]の値よりも大きければ影響はありません。</p>
CacheCheck=	300000	<p>キャッシュに格納されている全オブジェクトの状況をチェックするミリ秒数。デフォルト値を使用することをお勧めします。</p> <p>デフォルト値: 300000</p> <p>最小値: 60000。60000未満の値を指定すると、CacheManagerがキャッシュに格納されている値をチェックする頻度が高くなりすぎて、JASサーバーのオーバーヘッドが大きくなります。</p> <p>最大値: 600000。600000を超える値を指定しないでください。CacheManager はキャッシュ内のオブジェクトの状況を頻繁にチェックできなくなり、無効なキャッシュが期限切れ後もキャッシュに残ることになります。</p> <p>リレーションシップ: この値は、JAS.INIファイルの他のパラメータには依存しません。</p> <p>変更の効果: この値を小さくすると、CacheManager はキャッシュに格納されている値を頻繁にチェックするため、JASサーバーにオーバーヘッドが発生します。この値を極端に大きくすると、CacheManagerはキャッシュ内のオブジェクトの状況を頻繁にチェックしなくなり、無効</p>

		なオブジェクトが期限切れ後もキャッシュに残ることになります。
--	--	--------------------------------

[SERVER COMPONENTS]

[SECURITY]セクションでは次の設定を確認してください。jas.iniのこのセクションは、Webサーバーで環境変数の設定に使用されます。このセクションは、J.D. Edwardsから特に指示されない限り修正しないでください。

[SERVER COMPONENTS]

com.jdedwards.jas.UserManager
 com.jdedwards.jas.JDBCProxy
 com.jdedwards.jas.JDEORB
 com.jdedwards.jas.DDValidation
 com.jdedwards.jas.security.SecurityBroker
 com.jdedwards.jas.UJCJDBC
 com.jdedwards.jas.JDEUDCText
 com.jdedwards.jas.JDEUpdates
 com.jdedwards.jas.JDEQueries
 com.jdedwards.jas.JDEOWDirect
 com.jdedwards.jas.MenuServer
 com.jdedwards.jas.ServerQuery
 com.jdedwards.jas.JDESignon

[JDENET]

このセクションを使用するのは、ビジネス関数の実行とUBEの起動にJOWProxyサービスを使用しない場合のみです。

次の表は、Webサーバーの基本機能を提供する重要な設定を示します。

[JDENET]		
パラメータ	推奨設定	説明
serviceNameConnect=	6011	J.D. Edwardsエンタープライズ・サーバーがリスニングするTCPポート。この設定は環境に合わせて修正する必要があります。 デフォルトでは、このデフォルト・ポートの番号はOneWorld/ERPのリリースごとに1ずつ

[JDENET]		
パラメータ	推奨設定	説明
		大きくなります。OneWorld/ERPの場合、デフォルト値は6011です。

次の表は、クリティカルではない設定を示します。

パラメータ	推奨設定	説明
enterpriseServerTimeout=	90000	タイムアウト条件が発生するまでのミリ秒数。
MaxPoolSize=	50	エンタープライズ・サーバーへの最大接続数。
TempFileDir=		JDENET用の一時ディレクトリ。システム上の有効なディレクトリを入力する必要があります。

[SERVER]

このセクションを使用するのは、ビジネス関数の実行とUBEの起動にJOWProxyサービスを使用しない場合のみです。

次の表は、Webサーバーの基本機能を提供する重要な設定を示します。

[SERVER]		
パラメータ	推奨設定	説明
GlossaryText Server=	サイト依存の変数	このエントリでは、OneWorld/ERPがWebサーバー用の用語解説テキスト情報を提供するためにリスニングする、エンタープライズ・サーバーとポート番号を指定します。 構文は次のとおりです。 machine_name:6011 machine_nameはサイトの有効なマシン名です。

次の表は、クリティカルではない設定を示します。

パラメータ	推奨設定	説明
codePage=	1252	用語解説テキスト情報の表示に使用するコード・ページ。サポートされるエンコード体系は次のとおりです。 <ul style="list-style-type: none"> ・ 1252 英語と西ヨーロッパ言語 ・ 932 日本語 ・ 950 繁体字中国語 ・ 936 簡体字中国語 ・ 949 韓国語

[LOGIN]

このセクションでは、ログイン・オプションとLocal Directorオプションを設定します。

パラメータ	推奨設定	説明
PassKey=	サイト依存の変数	Cookieの暗号化に使用されるキーの英数字の一時変更。
externalhost=	machine:port	ブラウザからのリソース要求について完全修飾URLの作成に使用されます。
DisplayEnvironment=	Administrator preference (管理者の作業環境)	DefaultEnvironmentでデフォルト環境を設定している場合の環境表示ルール。 Show: ユーザーはログイン時にデフォルト環境を一時変更できます。 Hidden: ユーザーはログイン時に[Environment (環境)]ボックスを表示できません。 ReadOnly: ユーザーはログイン時に環境設定を変更できません。 UseDefault: ユーザーはログイン時に異なる環境を選択できますが、システムでは他のユーザー選択が一時変更され、最終的にはデフォルト環境にログインします。

DisablePasswordAboutToExpire	TRUE	「password about to expire (パスワード失効間近)」の通知を抑止する一時変更。
------------------------------	------	---

[LOGS]

このセクションでは、Webサーバーに関連する各種ロギング機能のシステム依存値を設定します。

[LOGS]		
パラメータ	推奨設定	説明
Log=	サイト依存の変数	jas.logファイルの位置と名称。パスには、システム上に存在するディレクトリを指定する必要があります。このパスをサイトのインストール環境と対照して確認してください。デフォルト値は次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war /log/jas.log
Debuglog=	サイト依存の変数	jasdebug.logファイルの位置と名称。パスには、システム上に存在するディレクトリを指定する必要があります。このパスをサイトのインストール環境と対照して確認してください。デフォルト・パスは次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/log /jasdebug.log
Debug=	FALSE	TRUE デバッグ・ロギングが有効です。 FALSE デバッグ・ロギングが無効です。
jdbcTrace=	FALSE	TRUE JDBCステートメントのトレース・ログが、標準出力ログ・ファイルに書き込まれます。 FALSE トレース・ログは書き込まれません。
JasDBLogging=	1	JAS DBのロギングなし。1 = ステートメント、UserSession。2 = ステートメント、UserSession、DataSource、パラメータ値。3 = 2 およびSELECTステートメント。
stderr=	サイト依存の変数	入力エラー用のWebSphereログ・ディレクトリを指定します。 STDERRキーとSTDOUTキーは、WebSphereアプリケーション・サーバーの定義時にSTDERRおよびSTDOUTフィールドで指定したのと同じディレクトリを指す必要があります。通常、これはB9のサブディレクトリです。デフォルト・パスは次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/log /stderr.log

[LOGS]		
パラメータ	推奨設定	説明
stdout=	サイト依存の変数	出力エラー用のWebSphereログ・ディレクトリを指定します。 STDERRキーとSTDOUTキーは、WebSphereアプリケーション・サーバーの定義時にSTDERRおよびSTDOUTフィールドで指定したのと同じディレクトリを指す必要があります。通常、これはB9のサブディレクトリです。デフォルト・パスは次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/log/stdout.log
rtlog=	サイト依存の変数	ランタイム・ログ・ファイルの位置と名称。デフォルト値は次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/log/rt.log
rtdebug=	サイト依存の変数	ランタイム・ログ・デバッグ・ファイルの位置と名称。デフォルト値は次のとおりです。 /u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/log/rtdebug.log
rtdebugTrace=	TRUE	ランタイム・ログ・ファイルを有効化するかどうかを指定します。
rtlogType=	ALL	ランタイム・デバッグ・ログ・ファイルに記録できるイベントのタイプを指定します。“ALL”の代わりに、次の1つまたは複数の設定を入力できます。 BSFN TableIO SYSFUNC MISC ER EVENT

[TRANSACTIONS]

このセクションでは、JASでトランザクション処理をカスタマイズするための構成を設定します。

[TRANSACTIONS]		
パラメータ	推奨設定	説明
ReapInterval=	5	Transaction Reaperで破棄されたトランザクションをチェックする間隔(分単位)。有効な値:1~nの整数。デフォルト:5分。
MaxAge=	10	トランザクションの最大経過時間(分単位)。この時間より古いトランザクションは、TransactionReaperによりロールバックされます。有効な値:1~nの整数。デフォルト:10分

[WEB GUI]

このセクションではWebのGUIインターフェイスのカスタマイズ方法とサービス・パック22で新しく加わった内容について説明します。

[WEB GUI]		
パラメータ	推奨設定	説明
bandwidth=	narrowまたはwide	<p>bandwidth設定を使用して、帯域幅の低いネットワークを考慮してGUIインターフェイスをカスタマイズします。デフォルトでは、WebサーバーはHTML形式のすべてのOneWorld/ERPフォームにタブ修飾を使用しています。タブ修飾により隅の丸いタブが表示されますが、このフォーマットでの表示には大きい帯域幅が必要になります。</p> <p>bandwidthを“narrow”に設定するとタブ修飾が無効化され、タブ表示に使用される帯域幅が小さくなります。この設定では、HTMLフォームに表示されるタブは単純な長方形のボックス状になります。</p> <p>bandwidthを“wide”に設定すると、タブ修飾が有効化されます。</p> <p>この設定は任意です。設定しなければ、Web修飾はデフォルトでwideとなります。</p>

付録 B. サンプルjas.iniファイル

次のコードは、Unix Webサーバー用のjas.iniファイルのサンプルです。

注意

次のサンプルjas.iniは、当初はB7333 OneWorldシステム用に構成されていました。そのため、多数のパスとパラメータ値をERP 9.0に合わせて更新する必要があります。ERP 9.0で有効な値については、「Appendix A: jas.iniファイルのパラメータと値」を参照してください。

```
# Please refer to the JAS Installation Guide for detailed information on
# this file.
#
#

[SECURITY] DataSource=System - B9
SecurityServer=
UseLogonCookie=FALSE
CookieLifeTime=7
SSOEnabled=FALSE
SSOUserIDOnly=FALSE
# set password only if SSOUserIDOnly is set to TRUE
Password=

[OWWEB] PathCodes=('JD9','DV9','PY9','PD9')
MO QUEUE=/u01/PeopleSoft/B7333/port8000/webclient/web/moqueue
MAXUser=100
UseMOWinNTShare=FALSE
SystemDateFormat=MDE
SystemDateSeparator=/

# The HelpPath key determines where JAS looks for help files.
# Syntax: "http://[machine]/[path]/".
# Note, if you specify "/jde/owhelp/", this machine is assumed.
HelpPath=/jde/owhelp/

[CACHE] # These intervals are in milliseconds.For example, 60000ms = 1 minute
UserSession=2400000
ResultSet=60000
CacheCheck=60000

[SERVER COMPONENTS]
#
# Do not modify this section
# com.jdedwards.jas.UserManager
```

```

com.jdedwards.jas.JDBCProxy
com.jdedwards.jas.JDEORB
com.jdedwards.jas.DDValidation
com.jdedwards.jas.security.SecurityBroker
com.jdedwards.jas.UDCJDBC
com.jdedwards.jas.JDEUDCText
com.jdedwards.jas.JDEUpdates
com.jdedwards.jas.JDEQueries
com.jdedwards.jas.JDEOWDirect
com.jdedwards.jas.ServerQuery
com.jdedwards.jas.JDESignon

[JDENET] # Enterprise Server JDENET port serviceNameConnect=6009

# JAS Server JDENET Listening port serviceNameListen=6009

# Maximum number of dedicated kernel types maxKernelRanges=1

# Time out value for requests sent to the Enterprise Server enterpriseServerTimeout=90000

# Directory for JDENET temporary file tempFileDir=/tmp

# Clustered logical datasource retry interval ClusterRetryInterval=60000

# Dedicated Kernel Definitions
[JDENET_KERNEL_DEF1]
krnlName=XAPI
Kernel process
ClassName=com.jdedwards.runtime.virtual.xapi.PublishedMessageProcessor
startMessageRange=15001
endMessageRange=15250
maxNumberOfThread=3

[SERVER] # Enterprise Server and port providing Glossary Text information
glossaryTextServer=SERVERNAME:6009

# Encoding scheme
# 1252- English and Western European
# 932- Japanese
# 950- Traditional Chinese
# 936- Simplified Chinese
# 949- Korean codePage=1252

[LOGS]
log=/u01/PeopleSoft/B7333/port8000/webclient/logs/jas.log
debuglog=/u01/PeopleSoft/B7333/port8000/webclient/logs/jasdebug.log Debug=FALSE

rtdebugTrace=FALSE
rtlog=/u01/PeopleSoft/B7333/port8000/webclient/logs/rt.log

```

```
rtdebug=/u01/PeopleSoft/B7333/port8000/webclient/logs/rtdebug.log
rtlogType=All
#or =BSFN|TableIO|SYSFUNC|MISC|ER|EVENT|THREAD (any combination).

# The STDERR and STDOUT keys must point to the same directory that was supplied for the
# STDERR and STDOUT fields when defining the WebSphere Application Server.
# This is usually the directory under B7333.
#
#stderr=/u01/PeopleSoft/B7333/port8000/webclient/logs/stderr.log
#stdout=/u01/PeopleSoft/B7333/port8000/webclient/logs/stdout.log

[JAS INSTANCE]
# This is for JAS redirector only
# RoundRobin will be used as default if there is no entry in [JAS WEIGHT].
# If redirect based on JAS instance weight, then each instance's weight must be specified,
# otherwise, weight 0 will be assumed, which means the server will not get
# any redirected request
# example:JAS1=http://127.0.0.1/jde
# Note:in this example, "JAS1" is the key, if weight will be given,
# this key must be used in [JAS WEIGHT] section to match this JAS instance

[JAS WEIGHT]
# The key must match the key in [JAS INSTANCE] section
# example:JAS1=2
# The weight is ratio based, for example:
# JAS1=2
# JAS2=1
# That means JAS1 will take as twice load as JAS2

[REDIRECTOR]
# interval unit is milliseconds Interval=5000
# This is the location that Redirector will redirect to, if no server is available.
# If it is empty, redirect will display a simple page telling user that no server is available
# example:NoServerAvailableURL=http://servername/path/noserver.html NoServerAvailableURL=

[TRANSACTIONS]
# Configuration settings to customize transaction processing within JAS.

# ReapInterval      :Interval in minutes of how often the TransactionReaper
#                   should check for abandoned transactions.
#                   Valid values:Integer from 1 to n
#                   Default:5 minutes
# MaxAge            :Maximum age in minutes of a transaction.Transactions
#                   older than this will be rolled back by the
#                   TransactionReaper.
#                   Valid values:Integer from 1 to n
#                   Default:10 minutes
#

ReapInterval=5
MaxAge=10
```


付録 C. jdbj.iniファイルのパラメータと値

jdbj.iniファイルには、JDBjの構成情報が含まれます。JDBjにより、J.D. Edwards WebサーバーからOneWorld/ERPデータベースにアクセスできます。このファイルのパラメータは、OneWorld/ERPデータへのアクセスにのみ使用され、スタンドアロン・データベースへのアクセス時には無視されます。

jdbj.iniファイルの重要なパラメータは、Javaサーバーのインストーラを使用してWebサーバーをインストールする間に自動的に更新されます。ただし、インストール後に、テキスト・エディタを使用してこれらの設定を修正できます。jdbj.iniファイルの機能は、OneWorld/ERPエンタープライズ・サーバーのjde.iniファイルに似ています。

ヒント

通常は、次の情報が含まれるように.iniファイルにコマンド行を追加してください。

- ・ 現行のJASパッケージのバージョン(CDに貼付されているラベルを参照)
- ・ ファイルの手動更新日
- ・ 更新者名

jdbj.iniファイルの各セクションは、次のカテゴリに分かれています。

- ・ [J.D. Edwards OneWorld/ERPデータ固有のセクション - ページ \[133\]](#)
- ・ [汎用セクション - ページ \[136\]](#)

J.D. Edwards OneWorld/ERPデータ固有のセクション

jdbj.iniの次のセクションは、OneWorld/ERPデータへのアクセス時にのみ適用されます。スタンドアロン・データベースへのアクセス時には無視されます。

設定	説明
[JDBj-BOOTSTRAP SESSION] - ページ [133]	システム・テーブルへのアクセスを提供するサインオン情報が含まれます。
[JDBj-BOOTSTRAP DATA SOURCE] - ページ [134]	OCMと他の一部のシステム・テーブルがあるデータ・ソースを定義します。
[JDBj-SPEC DATA SOURCE] - ページ [135]	シリアル化スペック・テーブルがあるデータ・ソースを定義します。

[JDBj-BOOTSTRAP SESSION]

この情報は、システム・テーブルにアクセスするためのサインオンに使用されます。JDBjは、ブートストラップ処理中にセキュリティ・サーバーを介してこのユーザーをサインオンします。サインオン情報が無効な場合は、OneWorld/ERPデータベースにアクセスできません。

注意

このセクションの情報は、旧インストール(ERP 9より前)のJDE.INIファイルの[DB SYSTEM SETTINGS]セクションに指定していた情報と一致する必要があります。

[JDBj-BOOTSTRAP SESSION]		
パラメータ	推奨設定	説明
user=	サイト依存の変数	OneWorld/ERPセキュリティ・サーバーへのログインに使用するユーザー名。
password=	サイト依存の変数	OneWorld/ERPセキュリティ・サーバーへのログインに使用するパスワード。
environment	サイト依存の変数	システム・テーブルが格納されているOneWorld/ERP環境。
role=	"*ALL"	すべてのロールにアクセスするには"*ALL"と入力します。

[JDBj-BOOTSTRAP DATA SOURCE]

このセクションでは、OCMと他の一部のシステム・テーブルがあるデータ・ソースを定義します。JDBjは、このデータ・ソースをブートストラップ時と後で必要に応じてOCMエントリを参照する際に使用します。データ・ソースが有効でなければ、OneWorld/ERPデータベースにはアクセスできません。

注意

このセクションの情報は、旧インストール(ERP 9より前)のJDE.INIファイルの[DB SYSTEM SETTINGS]セクションに指定していた情報と一致する必要があります。

[JDBj-BOOTSTRAP DATA SOURCE]		
パラメータ	推奨設定	説明
name=	サイト依存の変数	データ・ソース名。ブートストラップ接続には重要ではありませんが、エラー・メッセージとログ・ファイルに示されます。 (全サーバー)

[JDBj-BOOTSTRAP DATA SOURCE]		
パラメータ	推奨設定	説明
databaseType=	I = AS/400 O = Oracle S = SQL Server W = UDB	OneWorld/ERPシステムで使用するデータベースのタイプを選択します。 (全サーバー)
database=	サイト依存の変数	データベース名。 (Oracle、UDB)
owner	サイト依存の変数	データベース所有者。 (Oracle、SQL Server、UDB)
lob=	trueまたはfalse	LOBのサポート。 (OracleとAS/400)

[JDBj-SPEC DATA SOURCE]

これは任意のセクションです。シリアル化スペック・テーブルが存在するデータ・ソースを定義します。JDBjは、これらのパラメータをブートストラップ時およびその後で必要に応じてシリアル化スペック・オブジェクトを参照する際に使用します。このセクションを指定しなければ、JDBjはOCMを使用してシリアル化スペック・テーブルを検索します。有効でなければ、OneWorld/ERPデータベースにはアクセスできません。

特定のデータベース・プラットフォーム用に入力する必要があるフィールドについては、上記の「[\[JDBj-BOOTSTRAP DATA SOURCE\] - ページ \[134\]](#)」のパラメータの説明を参照してください。

[JDBj-SPEC DATA SOURCE]		
パラメータ	推奨設定	説明
name=	サイト依存の変数	データ・ソース名。ブートストラップ接続には重要ではありませんが、エラー・メッセージとログ・ファイルに示されます。 (全サーバー)
databaseType=	I = AS/400 O = Oracle S = SQL Server W = UDB	OneWorld/ERPシステムで使用するデータベースのタイプを選択します。 (全サーバー)

[JDBj-SPEC DATA SOURCE]		
パラメータ	推奨設定	説明
server=	サイト依存の変数	サーバー名。 (AS/400、SQL Server)
serverPort=	サイト依存の変数	サーバーのポート番号。 (SQL Server)
database=	サイト依存の変数	データベース名。 (Oracle、UDB)
physicalDatabase=	サイト依存の変数	物理データベース(AS/400のライブラリ修飾子として使用)。 (AS/400、SQL Server)
owner=	サイト依存の変数	データベース所有者。 (Oracle、SQL Server、UDB)
lob=	trueまたはfalse	LOBのサポート。 (OracleとAS/400)
unicode=	trueまたはfalse	Unicode変換を実行します。 (SQL Server)
user=	サイト依存の変数	OneWorld/ERPサーバーへのログインに使用するユーザー名。
password=	サイト依存の変数	OneWorld/ERPサーバーへのログインに使用するパスワード。

汎用セクション

上記以外のセクションは、OneWorld/ERPとスタンドアロンの両方のデータへのアクセス時に適用されます。

設定	説明
[JDBj-JDBC DRIVERS] - ページ [?]	JDBCドライバとJDBC固有の設定を定義します。
[JDBj-ORACLE] - ページ [137]	Webサーバー上のtnsnames.oraファイルの位置を定義します。
[JDBj-LOGS] - ページ [138]	JDBCドライバからのJDBCトレースを有効化します。

設定	説明
[JDBj-CONNECTION POOL] - ページ [138]	現在、これらの設定はJDBC接続のプーリングに常に使用されます。
[JDBj-RUNTIME PROPERTIES] - ページ [139]	これらの設定は、JDBj論理接続に適用可能なシステム・ランタイム・プロパティに対応しています。

[JDBj-JDBC DRIVERS]

このセクションの値を使用して、Webサーバーでデータベース・タイプへのアクセスに使用されるJDBCドライバを指定します。OneWorld/ERPの旧リリースでは、OracleとiSeriesのドライバがWebサーバーのデフォルトだったため、これらのドライバを設定する必要はありませんでした。ただし、OneWorld/ERPでは、JASのCLASSPATH例外を回避するために、デフォルト・ドライバは想定されません。Webサーバーでは、このセクションで指定したドライバのみがロードされます。

[JDBC DRIVERS]		
パラメータ	推奨設定	説明
Oracle=	oracle.jdbc.driver.OracleDriver	このドライバがOracleデータベースに使用されます。
UDB=	com.ibm.db2.jdbc.app.DB2Driver	このドライバはWindows用DB2 UDBに使用されます。

[JDBj-ORACLE]

このセクションでは、Webサーバー上のtnsnames.oraファイルの位置を定義します。このパスが必要なのは、Oracleデータベースへのデータ接続のみです。

パラメータ	推奨設定	説明
tns=	サイト依存の変数	<p>Oracle tnsnames.oraファイルの位置を指定します。Oracleを使用しない場合、このエントリは空白でかまいません。Oracleデータベースを使用する場合は、このステートメントのコメント化を解除する必要があります。</p> <p>デフォルト値は次のとおりです。</p> <pre>/u01/Oracle/Ora9/ network/ADMIN/tnsnames.ora</pre> <p>値“Ora9”では、使用中のOracleのバージョンを識別します。異なるバージョン</p>

パラメータ	推奨設定	説明
		ヨンのOracleをインストールすると、この値が変更されます。

[JDBj-LOGS]

このセクションでは、JDBCドライバからのJDBCトレースを有効化します。

パラメータ	推奨設定	説明
jdbcTrace=	false	このセクションでは、JDBCドライバからのJDBCトレースを有効化できます。通常はデバッグにのみ使用します。本稼働用環境でJDBCトレースを有効化すると、パフォーマンスが低下します。

[JDBj-CONNECTION POOL]

これらの設定はJDBC接続のプーリングに使用されます。

パラメータ	推奨設定	説明
jdbcPooling=	false	現在、JDBCプーリングはサポートされていないため、この設定を有効化してもプーリングはJDBjにより実行されます。
MaxConnection=	50	データ・ソースへの最大接続数。この数を越えた接続要求は、次に使用可能な接続の待ち行列に入れられます。
MinConnection=	0	データ・ソースへの最小接続数。古い接続をクローズしても、使用中かどうかに関係なくプールにはこの数の接続が残ります。
PoolGrowth=	5	接続要求を現在の割当てで満たせない場合に、システムによって作成される接続数。
InitialConnection=	5	プール作成時に作成される接続数。

connectionTimeout	1800000	非アクティブな接続キャッシュがプールから削除されるまでの時間(ミリ秒数)。 ERP9までは、この設定は[cache]セクションのConnectionパラメータでした。
CleanPoolInterval=	300000	プール・クリーナーが検証文字列を持つデータ・ソースをクリーニングするように設定される間隔を指定します。各キーで指定する値はミリ秒単位なので注意してください。たとえば、60000ミリ秒は1分です。 検証文字列の書式は次のとおりです。 ValidationString_x=<statement> xは、DB2/400の場合はI、4またはR、SQL Serverの場合はS、Oracleの場合はOです。 statementは、任意のユーザーが実行できるSQLステートメントです。 たとえば、次のように入力します。 ValidationString_I=Select * from QSYS2.SYSCOLUMNS WHERE 1=2 ValidationString_S=Select @@connections ValidationString_O=Select sysdate from dual
maxSize	50	各JDBC接続でキャッシュされるJDBC準備ステートメントの最大数。
cachePurgeSize	5	maxSize制限に達した場合にJDBC接続から除去されるJDBCステートメントの数。

[JDBj-RUNTIME PROPERTIES]

このセクションの設定は、JDBj論理接続に適用可能なシステム・ランタイム・プロパティに対応しています。

注意

示されている値はそれぞれのデフォルトです。テンプレートのデフォルト・エントリはコメント化されています。これらの値を変更する場合は、シャープ記号(#)を削除してください。

パラメータ	推奨設定	説明
dataCacheEntrySizeThreshold=	500	<p>データ・キャッシュ内の単一エントリの最大サイズ(ロー数)を設定します。入力がこのしきい値を超える可能性がある場合は、キャッシュされません。値-1は、該当するエントリをすべてキャッシュする必要があることを示します。</p> <p>有効な値:-1以上。</p>
dataCacheEntryThreshold=	100	<p>データ・キャッシュの最大サイズ(エントリ数)を設定します。データ・キャッシュがこのサイズを超えると、新規エントリ用の領域を確保するために最後に使用されたエントリが削除されます。値-1は、最大サイズがないことを示します。</p>
forceSingleDataSource=	false	<p>JDBjが、複数のデータベース・オブジェクトが関連する操作を単一データ・ソースに対して強制的に実行する必要があるかどうかを指定します。値がtrueの場合、JDBjは操作に関連する最初のデータベース・オブジェクトの物理データ・ソースのみの解決します。値がfalseの場合、JDBjは全データベース・オブジェクトの物理データ・ソースを解決し、単一操作が複数のデータ・ソースにまたがる場合は例外を発生させます。</p> <p>有効な値:true、false。</p>
resultSetTimeout=	60000	<p>操作が実行されなかった場合に、結果セットがタイムアウトになるまでの時間(ミリ秒数)。</p> <p>デフォルト設定を使用することをお勧めします。</p> <p>有効な値:-1以上。</p> <p>ERP9までは、この設定は[cache]セ</p>

		クシヨンのResultSet/パラメータでした。
retryInterval=	0	再試行間隔のプロパティ。べき等のデータベース操作を再試行するまでの待機間隔(ミリ秒数)を設定します。-1は再試行なし、0は即時再試行を意味します。 有効な値:-1以上。
retryMaximum=	5	べき等データベース操作の再試行回数を設定します。-1は無限再試行を意味します。 有効な値:-1以上。
ocmCachePurge=	3600000	非アクティブなOCMキャッシュ・エントリが除去されるまでのミリ秒数を設定します。 0はOCMキャッシュ・エントリが除去されないことを意味します。
personalSpecVersions=	false	ユーザーの個人用バージョンを読み取りかどうかを指定します。この設定は読み取り時のみ適用され、スペック生成時には適用されません。trueに設定すると、存在する場合は個人用バージョンが戻され、それ以外の場合はパブリック・バージョンが戻されます。このプロパティをfalseに設定すると、個人用バージョンは読み取られず、デフォルトでパブリック・バージョンに設定されます。 有効な値:true, false
securityCachePurge=	3600000	非アクティブ・サービス・キャッシュ・エントリが除去されるまでの時間(ミリ秒数)。0はサービス・キャッシュ・エントリが除去されないことを意味します。 デフォルト値を使用することをお勧めします。 有効な値:0以上。 ERP9までは、この設定は[cache]セクションのSecurity/パラメータでした。

serviceCachePurge=	3600000	<p>非アクティブ・サービス・キャッシュ・エントリが除去されるまでの時間(ミリ秒数)。0はサービス・キャッシュ・エントリが除去されないことを意味します。</p> <p>有効な値:0以上。</p> <p>ERP9までは、この設定は[cache]セクションのUDCInfoパラメータでした。</p>
specCachePurge=	3600000	<p>非アクティブ・スペック・キャッシュ・エントリが除去されるまでの時間(ミリ秒数)。0はスペック・キャッシュ・エントリが除去されないことを意味します。</p> <p>有効な値:0以上。</p> <p>ERP9までは、この設定は[cache]セクションのViewTableパラメータでした。</p>
specConsistencyCheck=	minimal	<p>スペックの一貫性チェック・レベル。このプロパティを上位レベルに設定すると、スペックの一貫性問題に関連する例外メッセージが改善されます。パフォーマンスが低下するため、不要な場合は下位レベルに設定してください。</p> <p>有効な値: full, minimal, none</p>
transactionIsolation=	default	<p>トランザクションに使用する分離レベルを設定します。</p> <p>有効な値: default、none、read uncommitted、read committed、repeatable read、serializable</p>
transactionTimeout=	120000	<p>トランザクションがタイムアウトになるまでの時間(ミリ秒数)。指定した時間内に操作が実行されなかった場合は、トランザクションがタイムアウトになります。このタイムアウトを-1に設定すると、トランザクションのタイムアウト機能が無効化されます。現在、この状況が発生するのはJDBjがJASTランザクションの一部として実行される場合のみです。</p> <p>有効な値:-1以上。</p>

triggerAutoFetch=	none	JDBjがOneWorldトリガーに渡すために、変更があったローを自動的に取り込む必要があるかどうかを指定します。特定のトリガーを正常に動作させるには、この操作が必要です。 有効な値: none、single、all。
updatableResultSetTimeout=	-1	操作が実行されなかった場合に、更新可能な結果セットがタイムアウトになるまでの時間(ミリ秒数)。-1は、更新可能な結果セットがRESULT_SET_TIMEOUTと同じ値になる必要があることを示します。 有効な値: -1以上。
usageExecutionThreshold=	20000	単一のデータベース操作の最大実行時間(ミリ秒数)。単一データベース操作の実行時間がこのしきい値を超えると、使用状況トラッキングの一部としてステートメントと時間がログに記録されます。 有効な値: 0以上。
usageFetchSizeThreshold=	500	1回の取込みで戻される予想最大取込みサイズ(ロー数)。1回の取込で戻されるロー数がこのしきい値を超えると、使用状況トラッキングの一部とし実際の取込みサイズがログに記録されます。-1は、任意の取込みサイズが有効であることを示します。 有効な値: -1以上。
usageResultSetOpenThreshold=	60000	結果セットがオープン状態になっている最大時間(ミリ秒数)。結果セットがオープン状態になっている時間がこのしきい値を超えると、使用状況トラッキングの一部として結果セットと時間がログに記録されます。この設定値に達しても、結果セットが自動的にクローズされることはありません。値-1は、この種の使用状況がトラッキングされないことを示します。 有効な値: -1以上。
usageTracking=	false	使用状況トラッキングを有効または

		<p>無効にしますこの設定により、テスト中に使用するトラッキングとロギングを追加できます。使用状況をトラッキングするとパフォーマンスが低下するため、本稼働用コードでは無効化してください。</p> <p>有効な値: true、false。</p>
usageTrackingOutput=	log	<p>使用状況トラッキングの出力先を指定します。</p> <p>有効な値: log、exception。</p>
usageTransactionActiveThreshold=	120000	<p>有効な値: -1以上。</p>

付録 D. サンプルJDBj.iniファイル

次のコードは、JDBj.iniファイルのサンプルです。このサンプルはすべてのプラットフォームに有効です。

```
=====
#
#           jdbj.ini - JDBj configuration
#
# This file contains configuration information for JDBj, which provides
# general database access capabilities on behalf of J.D. Edwards Java
# solutions.## IMPORTANT:All keys in this file are case sensitive.
#
=====

#
#           OneWorld specific information
#
# These sections apply only when accessing OneWorld data.These
# sections are ignored for standalone database access.
#
=====

#-----
# Bootstrap session
#
# This information is used to sign on to provide access to system
# tables.JDBj signs this user on via the security server at bootstrap-
# time, so if it is not valid, then no OneWorld database access is
# possible.
#
# NOTES:* This information in this section should match the
#         corresponding information specified in the
#         [DB SYSTEM SETTINGS] section of this installation's JDE.INI
#         file.
#-----

[JDBj-BOOTSTRAP SESSION]
user=xxxxxxx
password=xxxxxxx
environment=xxxxxxx
role=*ALL

#-----
# Bootstrap physical data source
#
# This defines the data source where the OCM and some other system
# tables reside.JDBj uses this at bootstrap time and later to look up
# OCM entries on demand.If it is not valid, then no OneWorld database
# access is possible.## Here is a description of each field in this section:
#
#-----
# Field           Description           Applies to
```

```

# -----
# name          The data source name.This is not that All
#               important for bootstrap connections,
#               but it does show up in error messages
#               and the log.
# -----
# databaseType  The database type.One of the      All
#               following:
#
#               I = AS/400
#               O = Oracle
#               S = SQL Server
#               W = UDB
# -----
# server        The server (i.e. the host).AS/400,
#               SQL Server
# -----
# database      The database.Oracle, UDB
# -----
# serverPort    The server port.SQL Server
# -----
# physicalDatabase The physical database (used as library AS/400,
#               qualifier for AS/400).SQL Server
# -----
# owner         The owner.Oracle,
#               SQL Server,
#               UDB
# -----
# lob          Supports lobs.Oracle,
#
#               AS/400
# -----
# unicode       Performs unicode conversion.SQL Server
# -----
#
# NOTES:* This information in this section should match the
#        corresponding information specified in the
#        [DB SYSTEM SETTINGS] section of this installation's JDE.INI
#        file.
# -----

[JDBj-BOOTSTRAP DATA SOURCE]
name=yyyyyyyyy
databaseType=y
server=
serverPort=
database=
physicalDatabase=
owner=
lob=false
unicode=true
# -----
# Bootstrap spec data source (Optional)

```

```
#
# This defines the data source where the serialized spec tables reside.
# JDBj uses this at bootstrap time and after to look up serialized spec
# objects on demand.If this section is not specified, then JDBj will
# use the OCM to find the serialized spec table.If it is not valid, no
# OneWorld database access is possible.
#
# See the description for "Bootstrap physical data source" above for
# information about which fields must be filled in for a given database
# platform.
#-----

[JDBj-SPEC DATA SOURCE]
name=zzzzzzzzz
databaseType=z
server=
serverPort=
database=
physicalDatabase=
owner=
lob=false
unicode=true
user=zzzzzzzzz
password=zzzzzzzzz

#=====
#           END - OneWorld specific information
#
# The rest of the sections apply when accessing OneWorld and standalone
# data.
#=====

#-----
# JDBC drivers
#
# This defines JDBC drivers and JDBC specific settings.
#-----

[JDBj-JDBC DRIVERS]
ORACLE=oracle.jdbc.driver.OracleDriver
AS400=com.ibm.as400.access.AS400JDBCdriver
SQLSERVER=com.microsoft.jdbc.sqlserver.SQLServerDriver
UDB=COM.ibm.db2.jdbc.app.DB2Driver

[JDBj-ORACLE]
tns=c:¥oracle¥ora9¥network¥admin¥tnsnames.ora

[JDBj-LOGS]
jdbcTrace=false

[JDBj-SERVER]
dbcsConversionTolerant=true
```

```

#-----
# Connection pool settings
#
# These are used only when J2EE connection pooling is not available.
#-----

[JDBj-CONNECTION POOL]
jdbcPooling=false
minConnection=5
maxConnection=50
initialConnection=5
poolGrowth=5
connectionTimeout=1800000
cleanPoolInterval=600000
maxSize=50
cachePurgeSize=5

#-----
# JDBj Runtime properties
#
# These are all optional.They correspond to system runtime properties
# applicable to JDBj logical connections.
#
# NOTES:* In each case, the values listed are the defaults.The
#       template default entries are commented out.Make sure
#       and remove the pound sign (#) when changing these values.
#-----

[JDBj-RUNTIME PROPERTIES]

; Data cache entry size threshold property.Sets the maximum
; size (in rows) for a single entry in the data cache.If a
; potential entry exceeds this threshold, then it will not be
; cached.-1 indicates that all eligible entries should be
; cached.
;
; Valid values:-1 or greater

# dataCacheEntrySizeThreshold=500

; Data cache entry threshold property.Sets the maximum size (in
; entries) for the data cache.If the data cache exceeds this
; size, then the least recently used entries are removed to
; make room for the new entry.-1 indicates that there is no
; maximum.
;
; Valid values:-1 or greater

# dataCacheEntryThreshold=100

; Force single data source property.Indicates whether JDBj
; should force operations involving multiple database objects to
; run against a single data source.If this is true, JDBj will

```

```
; only resolve the physical data source for the first database
; object involved in a given operation.Otherwise, JDBj will
; resolve physical data sources for all database objects and
; throw an exception if a single operation spans multiple data
; sources.
;
; Valid values:true, false

# forceSingleDataSource=false

; Result set timeout property.Sets the the number of
; milliseconds for the result set timeout.A result set will
; timeout if it has not had any operations for a period of time.
;
; Valid values:-1 or greater

# resultSetTimeout=60000

; Retry interval property.Sets the interval (in milliseconds)
; to wait before retrying idempotent database operations.-1
; means not to retry, 0 means retry immediately.
;
; Valid values:-1 or greater

# retryInterval=0

; Retry maximum property.Sets the number of times to retry
; idempotent database operations.-1 means to retry
; indefinitely.
;
; Valid values:-1 or greater

# retryMaximum=5

; OCM cache purge property.Sets the number of
; milliseconds after which inactive OCM cache entries are
; purged.0はOCMキャッシュ・エントリが除去されないことを意味します。
;
; Valid values:0 or greater

# ocmCachePurge=3600000

; Personal Spec Versions property.Indicates if user's personal
; version can be read.This is only enforced during a read and not
; during a spec generation.If it set to a true, the personal version
; will be returned if it exists, otherwise the public version will be
; returned.If the property is set to a false, then the personal
; version will not even be read and we will default to the public
; version
;
; Valid values:true, false

# personalSpecVersions=false
```

```

; Service cache purge property.Sets the number of
; milliseconds after which inactive service cache entries are
; purged.0 means to never purge service cache entries.
;
; Valid values:0 or greater

# securityCachePurge=3600000

; Service cache purge property.Sets the number of
; milliseconds after which inactive service cache entries are
; purged.0 means to never purge service cache entries.
;
; Valid values:0 or greater

# serviceCachePurge=3600000

; Spec cache purge property.Sets the number of
; milliseconds after which inactive spec cache entries are
; purged.0 means to never purge spec cache entries.
;
; Valid values:0 or greater

# specCachePurge=3600000

; Spec consistency check property.Sets the level of spec
; consistency checking.Set this property to a higher level to
; improve exception messages relating to spec inconsistency
; problems.Set this to a lower level when it is not needed
; since it may degrade performance.
;
; Valid values:full, minimal, none

# specConsistencyCheck=minimal

; Transaction isolation property.Sets the isolation level to be
; used for transactions.
;
; Valid values:default, none, read uncommitted, read committed,
;             repeatable read, serializable

# transactionIsolation=default

; Transaction timeout property.Sets the number of milliseconds
; for the transaction timeout.A transaction will timeout if it
; has not had any operations for a period of time.If this
; timeout is set to a -1 then the transaction timeout feature
; will be disabled.(This currently only happens when JDBj is
; run as part of a JAS transaction.)
;
; Valid values:-1 or greater

```

```
# transactionTimeout=120000

; Trigger auto fetch property.Indicates whether JDBj should
; automatically fetch rows to be changed in order to pass them
; to OneWorld triggers.This is needed for certain triggers to
; work properly.
;
; Valid values:none, single, all

# triggerAutoFetch=none

; Updatable result set timeout property.Sets the the number of
; milliseconds for the updatable result set timeout.An
; updatable result set will timeout if it has not had any
; operations for a period of time.-1 indicates that updatable
; result sets should have the same value as RESULT_SET_TIMEOUT.
;
; Valid values:-1 or greater

# updatableResultSetTimeout=-1

; Usage execution threshold property.Sets the expected maximum
; time (in milliseconds) for a single database operation to
; execute.If a single database operation takes longer than this
; threshold, then the statement and time will be logged as part
; of usage tracking.
;
; Valid values:0 or greater

# usageExecutionThreshold=20000

; Usage fetch size threshold property.Sets the expected maximum
; fetch size (in rows) for a single fetch to return.If a single
; fetch returns more than than this threshold, then the actual
; fetch size will be logged as part of usage tracking.-1
; indicates that any fetch size is valid.
;
; Valid values:-1 or greater

# usageFetchSizeThreshold=500

; Usage result set open threshold property.Sets the expected
; maximum time (in milliseconds) for a result set to remain
; open.If a result set is left open longer than this threshold,
; then the result set and time will be logged as part of usage
; tracking.This does not automatically close the result set.A
; usage result set open threshold value of -1 indicates that no
; such usage tracking will occur.
;
; Valid values:-1 or greater

# usageResultSetOpenThreshold=60000
```

```

; Usage tracking property.Enables or disables usage tracking.
; This provides additional tracking and logging to use during
; testing.Usage tracking does incur a performance penalty so it
; should be disabled for production code.
;
; Valid values:true, false

# usageTracking=false

; Usage tracking output property.Specifies the destination of
; usage tracking output.
;
; Valid values:log, exception

# usageTrackingOutput=log

; Usage transaction active threshold property.Sets the expected
; maximum time (in milliseconds) for a transaction to remain
; active.If a transaction is left active longer than this
; threshold, then the transaction and time will be logged as
; part of usage tracking.This does not automatically rollback
; the transaction.A usage transaction active threshold value of
; -1 indicates that no such usage tracking will occur.
;
; Valid values:-1 or greater

# usageTransactionActiveThreshold=120000

#=====
#                               END OF FILE
#=====

```

付録 E. WebSphere 4.0への移行

ここでは、Webサーバー構成をWebSphereアプリケーション・サーバー(WAS) 3.5からWAS 4.0に移行する方法について説明します。

WebSphere™ 4.0のアップグレードに関する技術的な最小要件は、次のとおりです。

- ・ JAS SP20以上
- ・ WAS 4.0、FixPack 3以上付き
- ・ IIS 5.0、またはIBM HTTPサーバー、バージョン1.3.19以上 (WAS4.0インストール・ソフトウェア上で使用可能な最新バージョンを使用)
- ・ JDK 1.3、最新のPTFまたはFixPack付き (WAS4.0とFixPackインストール・ソフトウェアに付属)
- ・ WebSphere™ 4.0リポジトリ用データベース・サーバー
次のいずれか1つを使用します。
 - ・ DB2/UDB
 - ・ Oracle

移行プロセスは次のタスクからなります。

- ・ [Javaサーバー構成のバックアップ作成 - ページ \[153\]](#)
- ・ [WebSphere 3.5のアンインストール - ページ \[156\]](#)
- ・ [WebSphere 4.0とJASのインストールと構成 - ページ \[159\]](#)

Javaサーバー構成のバックアップ作成

WAS 4.0に問題が生じた場合にシステムを復元できるように、現行のWebSphere™ 3.5構成と、IBM HTTPサーバー構成について、バックアップを作成することをお勧めします。移行プロセス中に、バックアップ・ディレクトリからWAS 4.0構成にデータベース・ドライバをコピーします。

構成ファイルを、現行のJASインストール環境で作成したバックアップ・ディレクトリに保存することをお勧めします。たとえば、現行のWebサーバーが/u01/PeopleSoft/B9にインストールされている場合は、バックアップ・ディレクトリとして/u01/PeopleSoft/B9/backupを作成し、このディレクトリに構成ファイルをすべて格納します。

バックアップ・プロセスは次のタスクからなります。

- ・ [WebSphere 3.5構成のバックアップ作成 - ページ \[154\]](#)

- ・ [IBM HTTPサーバー構成のバックアップ作成 \(IHSの場合のみ\) - ページ \[154\]](#)
- ・ [jas.iniファイルのバックアップ作成 - ページ \[155\]](#)
- ・ [データベース・ドライバのバックアップ作成 - ページ \[155\]](#)

WebSphere 3.5構成のバックアップ作成

ここでは、WebSphere™のbinディレクトリにあるXMLConfigユーティリティを使用して、WebSphere™ 3.5構成ファイルのバックアップを作成します。後でこのファイルを使用して、WebSphere™構成を復元できます。次の手順をWebサーバー上で実行して、XML構成ファイルを生成します。

▶ WebSphere 3.5構成のバックアップを作成するには

1. Webサーバー上で、IBM WS AdminServerサービスが実行されていることを確認します。
実行されていない場合は、WebSphereのstartupServer.shユーティリティを使用して 次のコマンドで開始します。
2. コマンド行で、<WAS_HOME>/binディレクトリに移動します。<WAS_HOME>はWebSphere™のルート・ディレクトリです。
たとえば、Sunサーバーの場合、このディレクトリは/opt/WebSphere/AppServer/binです。
3. 次のコマンドを実行してXML構成ファイルを生成します。

```
./XMLConfig.sh -export WebConf35.xml -adminNodeName <nodename>
```


<node name>はWebSphere™を実行中のサーバーの名称、WebConf35.xmlはバックアップ・ファイル名です。
4. 生成されたWebConf35.xmlを、WebSphereインストール・ディレクトリ外部のバックアップ・ディレクトリにコピーします。

IBM HTTPサーバー構成のバックアップ作成 (IHSの場合のみ)

IBM HTTPサーバーを使用している場合は、次のタスクを実行してIBM HTTPサーバー構成ファイル (httpd.conf)を保存します。

▶ IBM HTTPサーバー構成のバックアップを作成するには

1. Webサーバー上で、次のディレクトリに移動します。
<IBM HTTP Server>/conf/
<IBM HTTP Server>はIBM HTTPサーバーがインストールされているディレクトリです。
このディレクトリのパスは、UNIXサーバーごとに異なります。
 - ・ AIXの場合 : /usr/HTTPServer

- ・ HPの場合 : /opt/HTTPServer
 - ・ Sunの場合 : /opt/IBMHTTPD
2. httpd.confファイルを、上記のディレクトリからバックアップ・ディレクトリにコピーします。

jas.iniファイルのバックアップ作成

▶ jas.iniファイルのバックアップを作成するには

1. 基本のJASインストール内容が含まれているディレクトリに移動します。
たとえば、/u01/JDEdwards/JASに移動します。
2. jas.iniファイルを、JASディレクトリから前に作成したバックアップ・ディレクトリにコピーします。

データベース・ドライバのバックアップ作成

▶ データベース・ドライバのバックアップを作成するには

1. 現行のWebサーバー構成に含まれているデータベース・ドライバを、すべてバックアップ・ディレクトリにコピーします。

システム構成に応じて、次のファイルを含めることができます。

データベース	ファイル
"Oracle"	classes12.zip
DB2/UDB	db2java.zip
MS SQL Server 7	weblogic.zip
MS SQL Server 2000	msbase.jar、mssqlserver.jar、msutil.jar
iSeries (AS/400)	jt400.jar

デフォルトでは、これらのファイルはJASの基本インストール・ディレクトリ(/u01/JDEdwards/JASなど)にあります。

2. OracleまたはSQLデータベースを使用している場合は、次のtnsnameファイルを/u01/JDEdwards/JASからバックアップ・ディレクトリに移動します。

データベース	ファイル
“Oracle”	tnsnames.ora

WebSphere 3.5のアンインストール

WebSphere™ 3.5.xをアンインストールして、新たにWebSphere™ 4.0をインストールすることをお勧めします。このプロセスの場合は、「[WebSphereからの Web Server構成の削除 - ページ \[156\]](#)」をスキップし、「[WAS 3.5.xインストールの削除 - ページ \[157\]](#)」に進んでください。WebSphere™ 3.5をアンインストールせずに、WebSphere™ 3.5.xから4.0に直接アップグレードするように選択した場合、またはWebSphere™ 3.5.xと4.0を並行して実行する場合は、次の点に注意してください。

- ・ WAS 4.0.x Advanced Editionに直接アップグレードまたは移行できるのは、WebSphere™ 3.5.x Advanced Edition Application Serverのみです。
- ・ WebSphere 3.5.x Standard Application ServerからWebSphere™ 4.0 Advanced Editionには、直接移行できません。WebSphere 3.5.x Standard Application Serverを使用している場合は、最初のタスクをスキップし、「[WAS 3.5.xインストールの削除 - ページ \[157\]](#)」のタスクを実行して、WebSphere™ 4.0 Advanced Editionにアップグレードしてください。
- ・ WebSphere 3.5.x Standard Editionをインストールして、WebSphere™ 4.0.x Advanced Editionと並行して実行できますが、このアプローチはお勧めしません。この構成をセットアップするには、IBMのWebサイトにアクセスし、移行手順に従ってください。
- ・ WebSphere 3.5.x Advanced Edition Application Serverを使用しており、WebSphere™ 4.0.x Advanced Editionに直接移行するように選択した場合は、Webサーバーをアンインストールする必要があります。このアプローチはお勧めしませんが、セクション「[WebSphereからの Web Server構成の削除 - ページ \[156\]](#)」のタスクを実行した後、IBMのWebサイトにアクセスし、その手順に従ってWebSphere™ 4.0にアップグレードできます。

次のタスクを実行して、現行のWebサーバーをアンインストールしてください。

- ・ [WebSphereからの Web Server構成の削除 - ページ \[156\]](#)
- ・ [WAS 3.5.xインストールの削除 - ページ \[157\]](#)
- ・ [IBM HTTPサーバーのインストール削除 - ページ \[158\]](#)

WebSphereからの Web Server構成の削除

他のWebアプリケーションをWebSphere™で管理しているか、WebSphere™ 3.5.x Advanced Application Serverを使用しており、WebSphere™ 4.0 Advanced Editionに直接アップグレードするように選択したために、J.D. Edwards推奨のアップグレード・ルートに従えない場合は、次のタスクを実行して、WebSphere™ 3.5.xから現行のJAS構成を削除します。

▶ WebSphere™からWeb Server構成を削除するには

1. <WAS_HOME>/binディレクトリから次のコマンドを実行します。

```
./adminclient.sh
```
2. [WebSphere Administrative Console]を開きます。
3. J.D. Edwardsアプリケーション・サーバーが実行中でないことを確認します。
4. J.D. Edwardsアプリケーション・サーバー (AS_JDEdwards_1など)を右クリックし、[Remove]を選択します。
5. 削除したアプリケーションに関連付けられている仮想ホスト (VH_JDEdwards_1など)を選択し、アイコンを展開してみます。

アイコンを展開できる場合は、他のアプリケーションが関連付けられています。
6. 仮想ホストに関連付けられているアプリケーションを削除します。
7. すべての関連アプリケーションを削除した後、[Virtual Host]アイコンを右クリックして[Remove]を選択します。
8. J.D. Edwardsアプリケーション・サーバーと仮想ホストが他にもある場合は、上記のステップを繰り返して削除します。
9. 下記の残りのタスクをスキップします。特定のアップグレード・ルートを実行する手順については、IBMのWebサイト上で関連するWebSphere™マニュアルを参照してください。WebSphere™ 4.0へのアップグレード後に、「Webサーバーのインストールと構成」のタスクを実行します。

WAS 3.5.xインストールの削除

ここでは、推奨アップグレード・パスについて説明します。このアプローチでは、WebSphere Application Server 3.5.xインストールを削除し、その後のタスクで、新たにWebSphere Application Server 4.0をインストールします。

▶ WAS 3.5.xインストールを削除するには

1. コマンド行から、次のコマンドを入力して<WAS_HOME>ディレクトリに移動します。

```
> cd <WAS_HOME>
```


<WAS_HOME>はWebSphere™の基本またはルート・ディレクトリです。
2. 次のコマンドを入力してDISPLAYを設定します。

```
> export DISPLAY=<IPAddress>:0.0
```


<IPAddress>はマシンのIPアドレスです。
3. WebSphere™WebSphereが実行されていることを確認します。WebSphere™が実行されている場合は、次の手順に従ってWebSphereアプリケーション・サーバーを停止します。
 - a. [WebSphere Administrative Console]を開きます。

- b. サーバーのノード(サーバー名)を右クリックして、[Stop]を選択します。

この管理コンソールを実行中のノードまたは接続先ノードを停止しようとしていることを示すメッセージが表示されます。

- c. [Yes]をクリックしてサーバーを停止します。
- d. コマンド行から、次のコマンドを入力してサーバーが停止していることを確認します。

```
> ps -ef | grep java
```

WebSphere™実行中のプロセスがないことが表示されます。

4. 次のコマンドを実行して、このディレクトリにあるuninstall.shスクリプトを実行します。

```
> ./uninstall.sh
```

このスクリプトでは、Websphere™インストール削除の確認を求めるプロンプト・ウィンドウが表示されます。

5. [Uninstall(アンインストール)]をクリックし、WebSphere™インストールを完全に削除します。
6. サーバーからWebSphere™ディレクトリを削除します。

IBM HTTPサーバーのインストール削除

WebSphere 4.0には、IBM HTTPサーバーのバージョン1.3.19以上が必要です。HTTPサーバーの旧バージョンを使用している場合は、次のタスクを実行し、現行のIBM HTTPサーバーのインストールを削除してから最新のIBM HTTPサーバーをインストールします。

► IBM HTTPサーバーのインストールを削除するには

1. HTTPサーバーが実行中でないことを確認します。
2. サーバーが実行されている場合は、次のコマンドを入力して停止します。
 - a. IBM HTTPサーバーの基本ディレクトリで、/binに移動します。
 - b. <IBM HTTP SERVER>/binディレクトリから次のコマンドを実行します。

```
> ./apachectl stop
```

3. インストールされているソフトウェアの削除に使用するユーティリティは、UNIXプラットフォームごとに異なります。IBM HTTPサーバー・ソフトウェアを削除するには、UNIXサーバー固有のプラットフォーム・ユーティリティを使用する必要があります。このソフトウェアの削除には、必ずroot UNIXアカウントを使用します。

該当するユーティリティを使用してIBM HTTPサーバーを削除してください。

プラットフォーム	ユーティリティ
AIX™	SMIT
HP-UX	SAM
SUN	admintool

4. インストール削除プロセスの完了後に、サーバーからIBM HTTP Serverディレクトリを削除します。

WebSphere 4.0とJASのインストールと構成

J.D. Edwardsの推奨パスに従ってWebSphere™ 3.5からWebSphere™ 4.0にアップグレードした場合は、このガイドに従ってWebSphere™ 4.0をインストールします。これ以後の各付録の説明に従って、WebサーバーでWebSphere™を構成してください。

他の移行パスを選択した場合は、IBMのWebサイトで該当するガイドを検索して、アップグレード・プロセスを実行してください

付録 F. Webサーバーのサービス・パックのアップグレード

ここでは、Webサーバーのサービス・パックを現行リリースにアップグレードする方法について説明します。この手順の途中で、現行の構成のバックアップを作成した後、WebSphere構成からJ.D. Edwards関連のコンポーネントを削除します。

はじめる前に

- ・ MTRを参照し、WebSphereの有効なFixPackがインストールされていることを確認します。

アップグレード・プロセスは次のタスクからなります。

- ・ [Webサーバー構成のバックアップ作成 - ページ \[161\]](#)
- ・ [WebSphereからのWebサーバー構成の削除 - ページ \[162\]](#)
- ・ [キャッシュ・ディレクトリの削除 - ページ \[163\]](#)
- ・ [新規サーバーのサービス・パックのインストール - ページ \[164\]](#)

Webサーバー構成のバックアップ作成

問題が生じた場合にシステムを復元できるように、現行のWebSphere™ 4.0構成と、IBM HTTPサーバー構成について、バックアップを作成することをお勧めします。

構成ファイルを、現行のJASインストール環境で作成したバックアップ・ディレクトリに保存することをお勧めします。たとえば、現行のWebサーバーが/u01/JDEdwards/JASにインストールされている場合は、バックアップ・ディレクトリとして/u01/JDEdwards/JAS/backupを作成し、このディレクトリに構成ファイルをすべて格納します。

バックアップ・プロセスは次のタスクからなります。

- ・ [WebSphere 4.0構成のバックアップ作成 - ページ \[161\]](#)
- ・ [jas.iniファイルのバックアップ作成 - ページ \[162\]](#)
- ・ [ポータル・コンポーネントのバックアップ作成 - ページ \[162\]](#)

WebSphere 4.0構成のバックアップ作成

ここでは、WebSphere™のbinディレクトリにあるXMLConfigユーティリティを使用して、WebSphere™ 4.0構成ファイルのバックアップを作成します。後でこのファイルを使用して、WebSphere™構成を復元できます。次の手順をWebサーバー上で実行して、XML構成ファイルを生成します。

- ▶ WebSphere 4.0構成のバックアップを作成するには
-

1. Webサーバー上で、IBM WS AdminServerサービスが実行されていることを確認します。
実行されていない場合は、WebSphereのstartupServer.shユーティリティを使用して開始します。
2. コマンド行で、<WAS_HOME>/binディレクトリに移動します。<WAS_HOME>はWebSphere™のルート・ディレクトリです。
たとえば、Sunサーバーの場合、<WAS_HOME>/のデフォルト・ディレクトリは /opt/WebSphere/AppServer/binです。
3. 次のコマンドを実行してXML構成ファイルを生成します。

```
./XMLConfig.sh -export WebConf40.xml -adminNodeName <nodename>
```


<node name>はWebSphere™を実行中のサーバーの名称、WebConf40.xmlはバックアップ・ファイル名です。
4. 生成されたWebConf40.xmlを、WebSphereインストール・ディレクトリ外部のバックアップ・ディレクトリにコピーします。
たとえば、このファイルを/u01/JDEdwards/JAS/backupにコピーします。

jas.iniファイルのバックアップ作成

▶ jas.iniファイルのバックアップを作成するには

1. 基本のJASインストール内容が含まれているディレクトリに移動します。
たとえば、/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/WEB-INFに移動します。
2. このディレクトリからjas.iniファイルを前に作成したバックアップ・ディレクトリにコピーします。
たとえば、このファイルを/u01/JDEdwards/JAS/backupにコピーします。

ポータル・コンポーネントのバックアップ作成

カスタマイズされたポータル・コンポーネントを作成しており、それをアップグレード後の構成に移行する場合は、『Portal (ポータル)』ガイドの指示に従って、カスタマイズされたコンポーネントをバックアップ・ディレクトリにエクスポートします。アップグレードの完了後に、『Portal (ポータル)』ガイドの指示に従って、カスタマイズされたコンポーネントを新規構成にインポートしてください。

WebSphereからのWebサーバー構成の削除

▶ WebSphere™からWebサーバー構成を削除するには

1. <WAS_HOME>/binディレクトリから次のコマンドを実行します。

```
./adminclient.sh
```


[WebSphere Administrative Console]が開きます。

2. J.D. Edwardsアプリケーション・サーバー(AS_JDEdwards_1)が実行中でないことを確認します。
3. アプリケーション・サーバー・ノード(AS_JDEdwards_1)を展開し、J.D. Edwardsエンタープライズ・アプリケーション(EA_JDEdwards)を右クリックして[削除]を選択します。

注意

アプリケーション・サーバーを削除する前にエンタープライズ・アプリケーションを削除しないと、エラーが生成されます。

エンタープライズ・アプリケーションのエクスポートを確認するプロンプトが表示されます。

4. エンタープライズ・アプリケーションのバックアップを作成する場合は、バックアップ・ディレクトリ名を入力します。

このディレクトリにファイル(EA_JDEdwards_1.ear)がエクスポートされるので、後で問題が発生した場合に使用できます。

ファイルノ生成後に、エンタープライズ・アプリケーションの削除を確認するプロンプトが表示されません。
5. [はい]をクリックします。
6. J.D. Edwardsアプリケーション・サーバー(AS_JDEdwards_1など)を右クリックし、[Remove]を選択します。

アプリケーション・サーバーを削除すると、/u01/JDEdwards/JASの下のEA_JDEdwards_1.earが自動的に削除されます。
7. 削除したアプリケーションに関連付けられている仮想ホストを選択します(VH_JDEdwards_1など)。
8. 仮想ホストに関連付けられているアプリケーションを削除します。(アプリケーション・サーバーは1つのみです)。
9. すべての関連アプリケーションを削除した後、[Virtual Host]アイコンを右クリックして[Remove]を選択します。
10. J.D. Edwardsアプリケーション・サーバーと仮想ホストが他にもある場合は、上記のステップを繰り返して削除します。
11. 下記の残りのタスクをスキップします。代わりに、前の「Webサーバーのインストールと構成」のタスクを実行します。これらのタスクは、新規Webサーバーをインストールする場合と同様に実行してください。

キャッシュ・ディレクトリの削除

WebSphereインストールには、Webサーバーの旧バージョンに関連付けられているアプリケーション・サーバー用の1つまたは複数のキャッシュ・ディレクトリがあります。これらのサブディレクトリにキャッシュされた情報は、新規構成には無効であり、削除する必要があります。

▶ キャッシュ・ディレクトリを削除するには

1. OneWorld Javaサーバー上で、WAS_HOME/temp/nodeディレクトリに移動します。WAS_HOMEはWebSphereのルート・ディレクトリ、nodeはWebSphereのノード名(通常はシステム名)です。
たとえば、Sunサーバーの場合、このディレクトリは/opt/WebSphere/AppServer/temp/nodeです。
2. temp/nodeディレクトリで、対応するアプリケーション・サーバー・ディレクトリ(AS_JDEdwards_1)およびすべてのサブディレクトリを削除します。

これにより、古いWebサーバー・インストール環境からのキャッシュがクリアされます。アプリケーション・サーバーを再起動すると、更新後の情報を使用して新規キャッシュが作成されます。

新規サーバーのサービス・パックのインストール

上記のタスクを実行した後、次のタスクを実行して新規サービス・パック・ソフトウェアをインストールします。

▶ 新規Webサーバーのサービス・パックをインストールするには

1. 「Webサーバーのインストールと構成」に戻り、新規サービス・パックのインストールを実行します。
2. 新規サービス・パックのインストール後に、「OneWorld/ERPのシリアル化オブジェクトの生成」に進んで「ジェネレート・マシンへのJASビルドのコピー」のタスクを実行します。
3. Webサーバーの新規シリアル化オブジェクト・セットを生成します。
4. バックアップ・ディレクトリに保存したデータベース・ドライバを、新規に構成したアプリケーション・サーバーにコピーします。

付録 G. IBM HTTPサーバーの構成

IBM HTTPサーバーには、IBM WebSphereアプリケーション・サーバー用のWebサーバー・コンポーネントが用意されており、通常はWebSphereと共にインストールされ、構成されます。ただし、HTTPサーバーを別にインストールすることもできます。この場合は、Webベースの管理ツールを使用して、IBM HTTPサーバーをJAS用に構成できます。最初に、Webサーバーの構成権限を持つHTTPサーバーの管理ユーザーを設定した後、ブラウザからHTTPサーバーを構成します。

注意

インストール後のIBM HTTPサーバーは、UNIXプラットフォームに応じて次の位置に常駐します。

- ・ AIXの場合 : /usr/HTTPServer
- ・ HP-UXの場合 : /opt/HTTPServer
- ・ Solarisの場合 : /opt/IBMHTTPD

次のタスクでは、このディレクトリを<IHS_HOME>ディレクトリと表記しています。

▶ IBM HTTPサーバーを構成するには

1. コマンド・プロンプトを開き、<IHS_HOME>/binディレクトリに移動します。
2. 次のコマンドを入力して、管理ユーザーとパスワードを設定します。

```
>./htpasswd -m <admin.passwd> <loginname>
```

<admin.passwd>は<IHS_HOME>/conf/admin.passwdファイルへのパス、<loginname>はパスワードを設定するユーザーの名前です。

注意

任意のユーザー名を使用できます。サーバー上で有効なユーザーでなくてもかまいません。

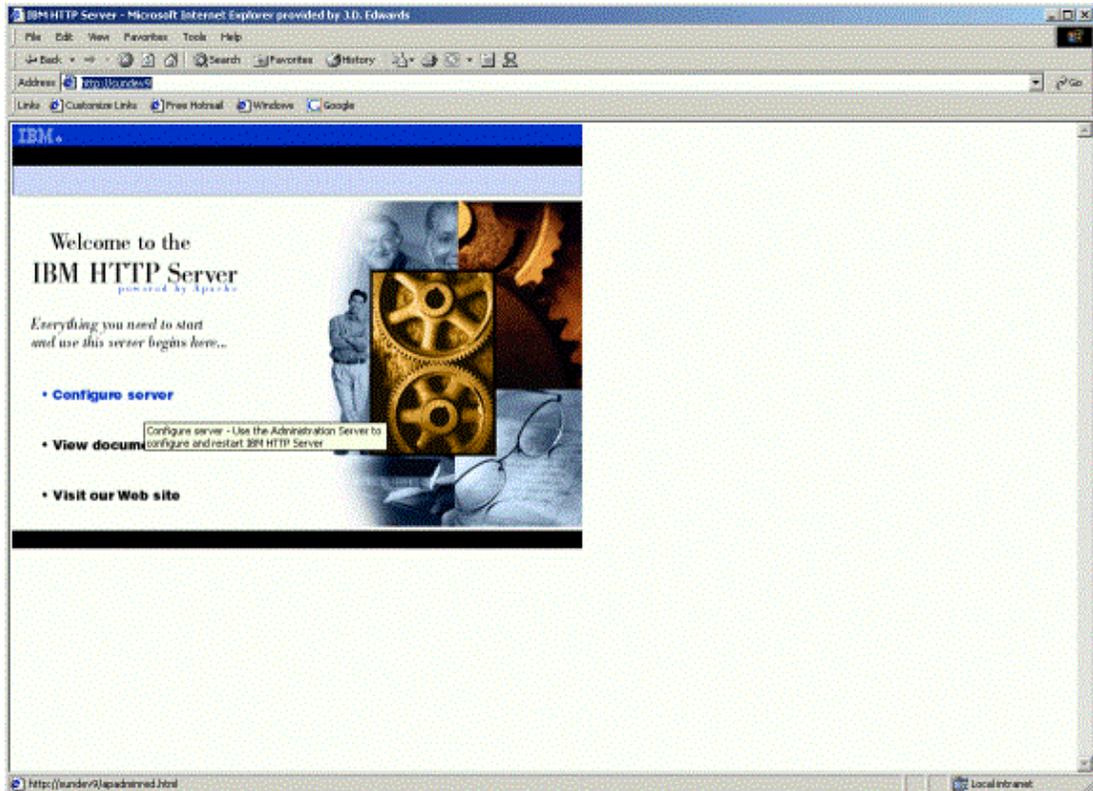
3. <IHS_HOME>/binディレクトリから、次のコマンドを入力してIHS adminサーバーとhttpサーバーを起動します。

```
>./adminctl start
```

4. ブラウザを開いて次のURLを入力します。

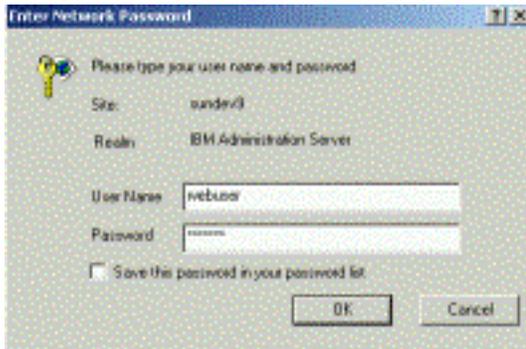
```
http://servername
```

ブラウザに[IBM HTTP Server Welcome (IBM HTTPサーバーへようこそ)]ページが表示されます。

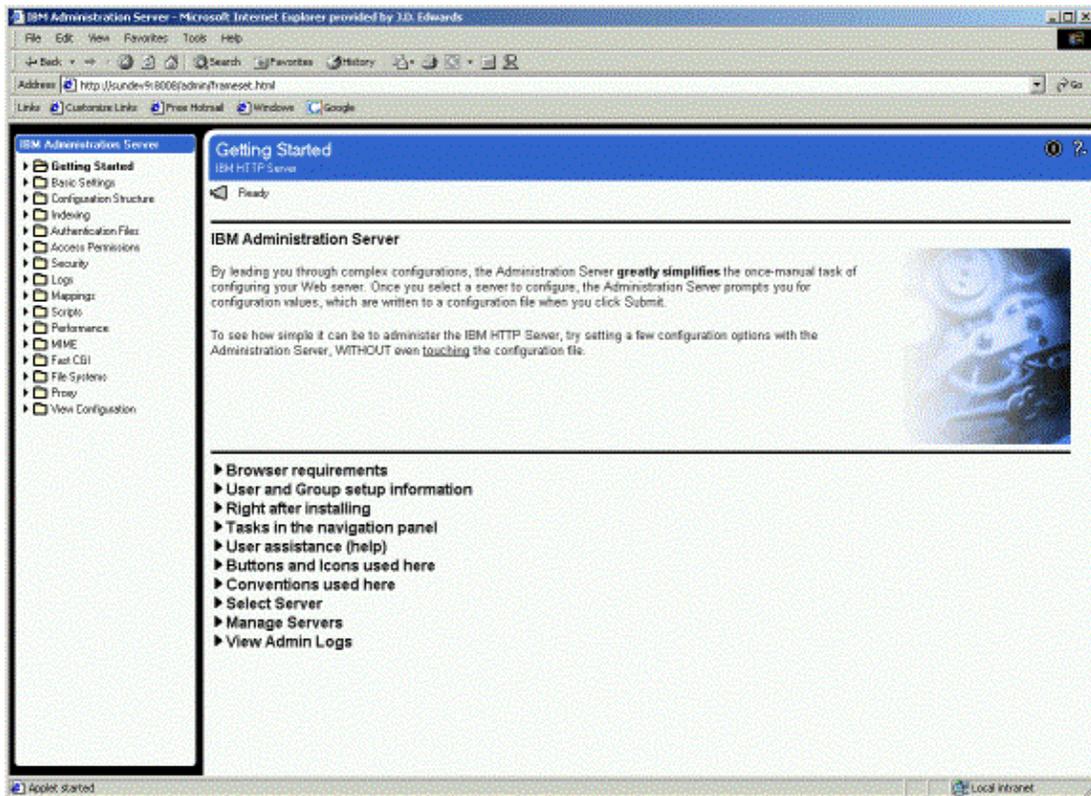


5. [configure server(サーバーの構成)]をクリックします。

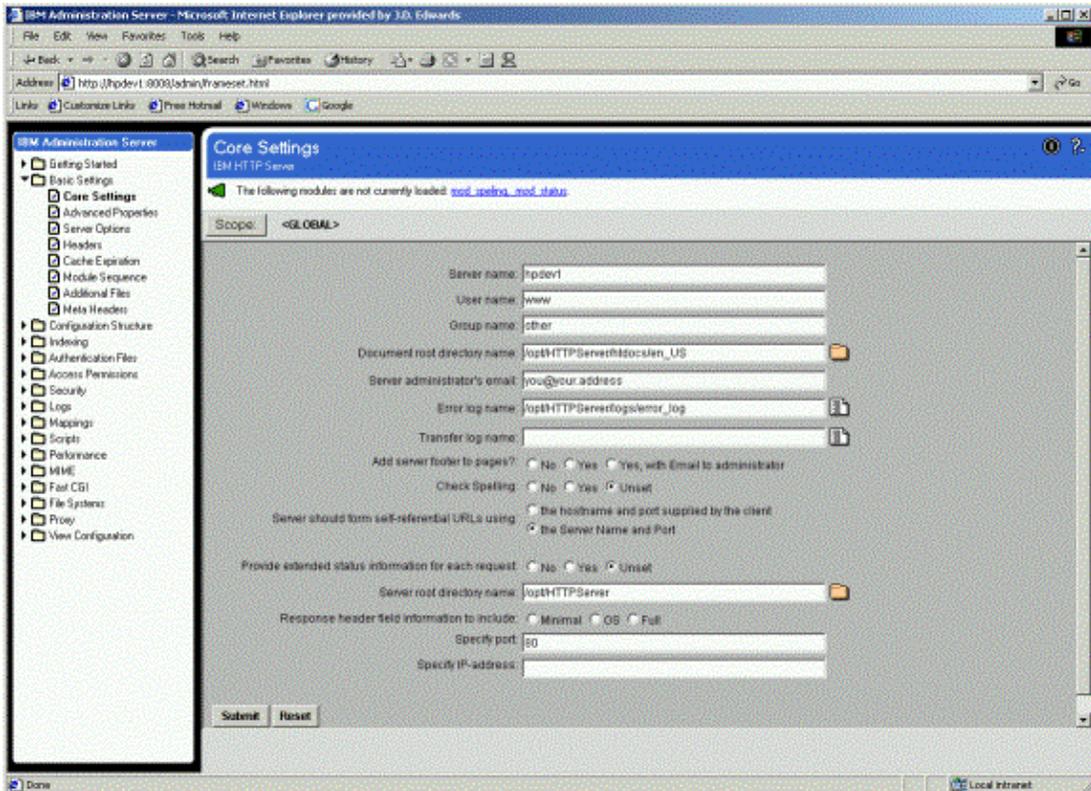
ログオン・プロンプトから、上記のステップで作成したユーザーIDとパスワードを入力します。



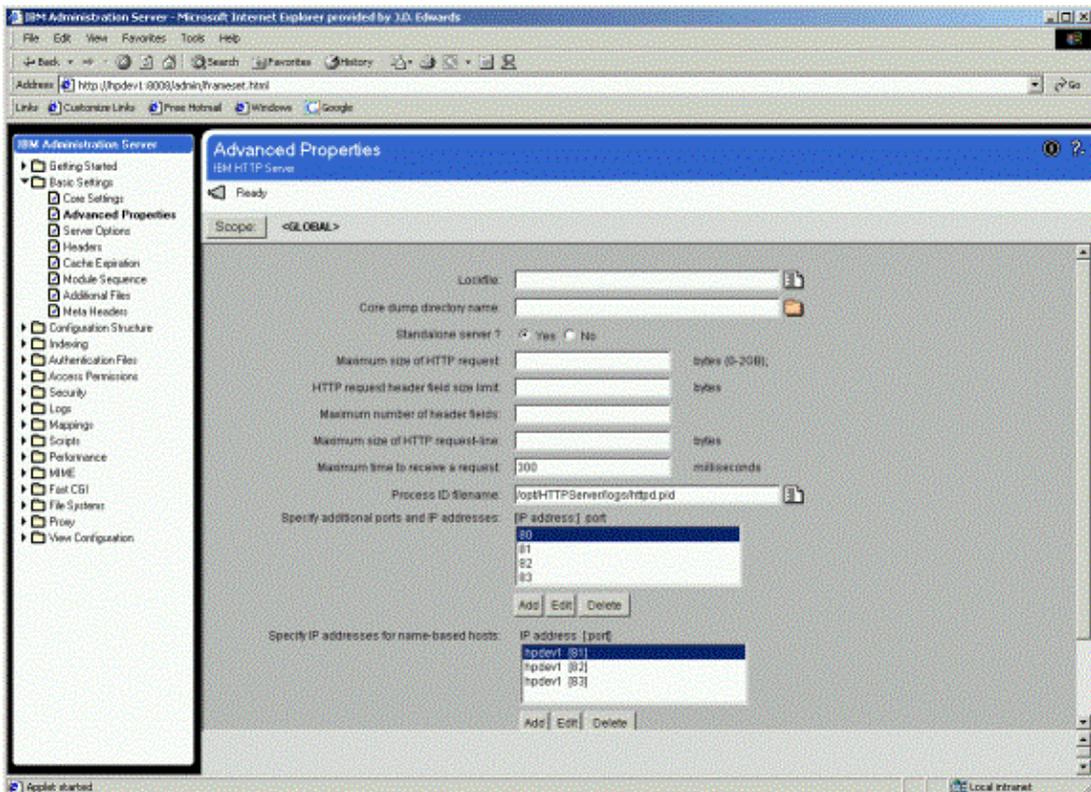
6. [OK]をクリックします。



7. [Getting Started(スタート・ガイド)]の左ペインで、矢印をクリックして[Basic Settings(基本設定)]アイコンを展開します。
8. [Core Settings(コア設定)]をクリックし、[Server name(サーバー名)]の設定が正しいことを確認します。

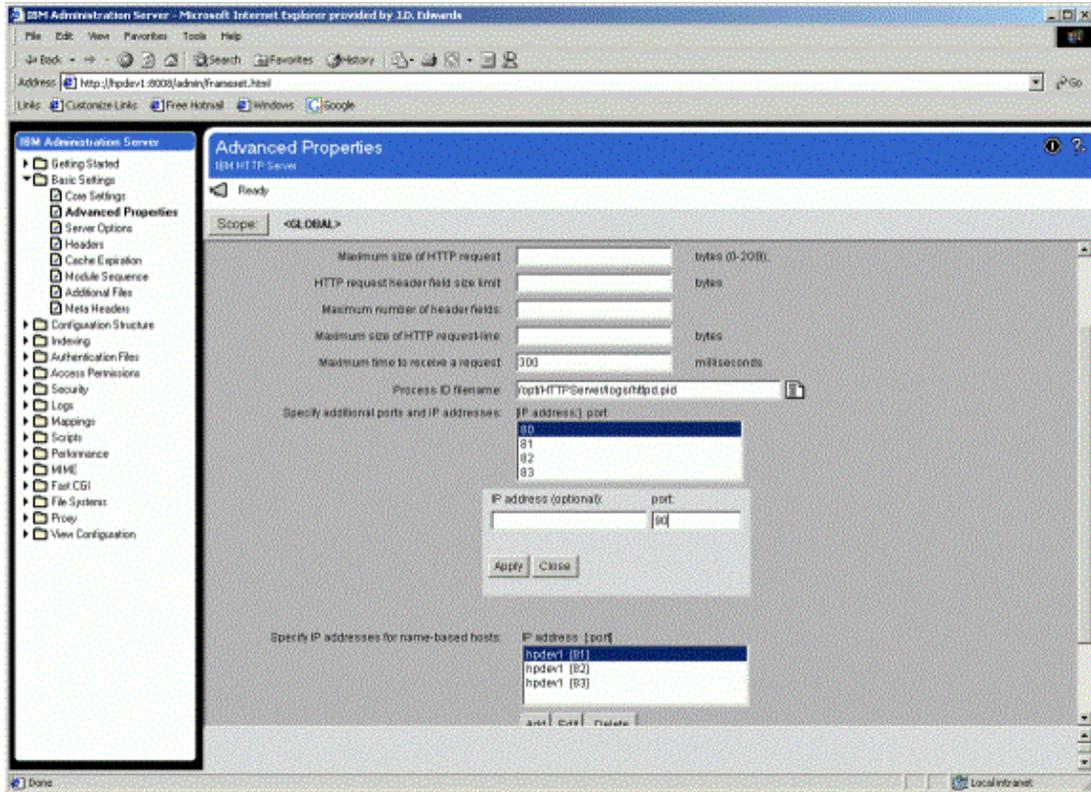


9. 左ペインで、[Core Settings]の下の[Advance Properties (拡張プロパティ)]をクリックします。

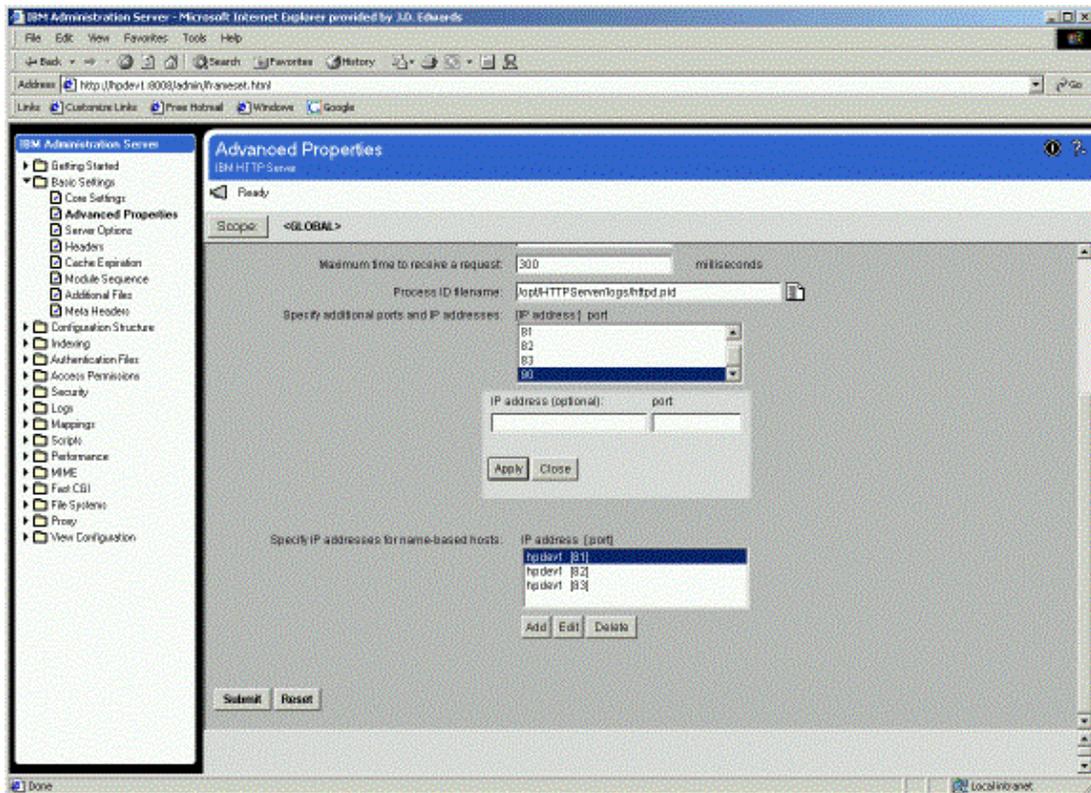


10. [(ポートとIPアドレスの追加指定)]フィールドで、Webサーバーでリスニングするポートを指定できます。

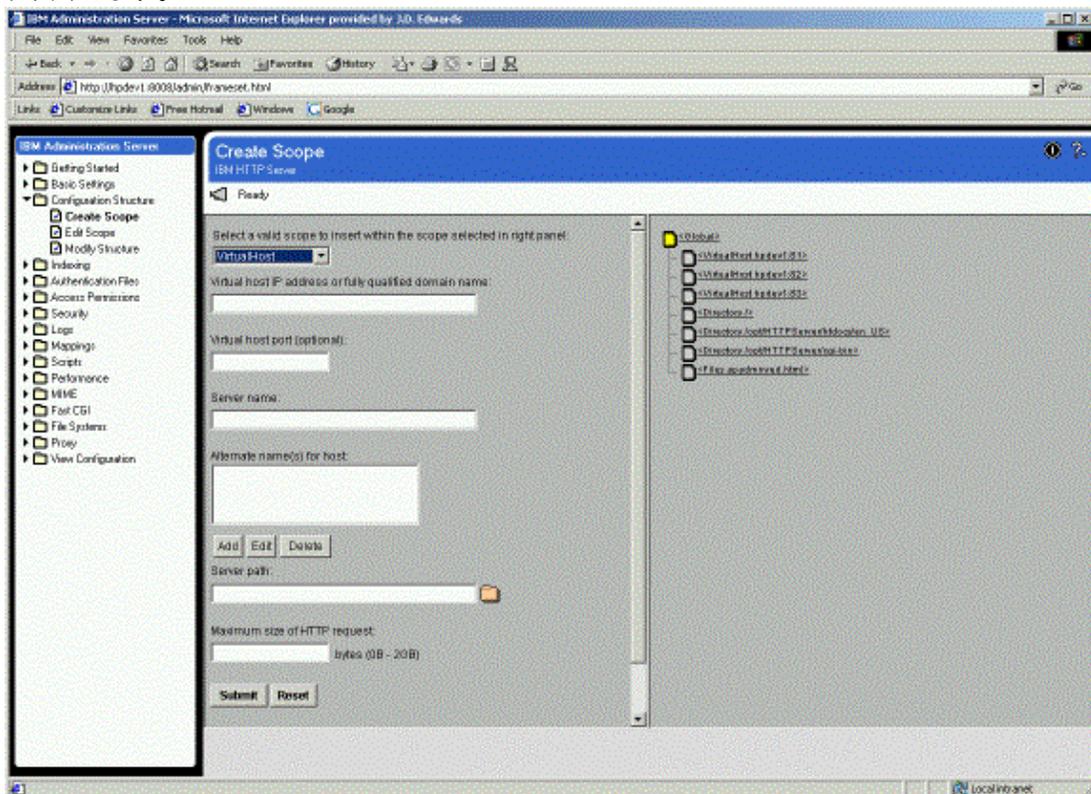
たとえば、ポート90を追加する場合は、このフィールドの下の[Add]をクリックし、[port(ポート)]フィールドに“90”と入力します。[IP Address(IPアドレス)]フィールドは空白にしておきます。



11. [Apply]をクリックしてポート・リストに新規ポートを追加し、スクロール・ダウンして[Submit(発行)]をクリックします。



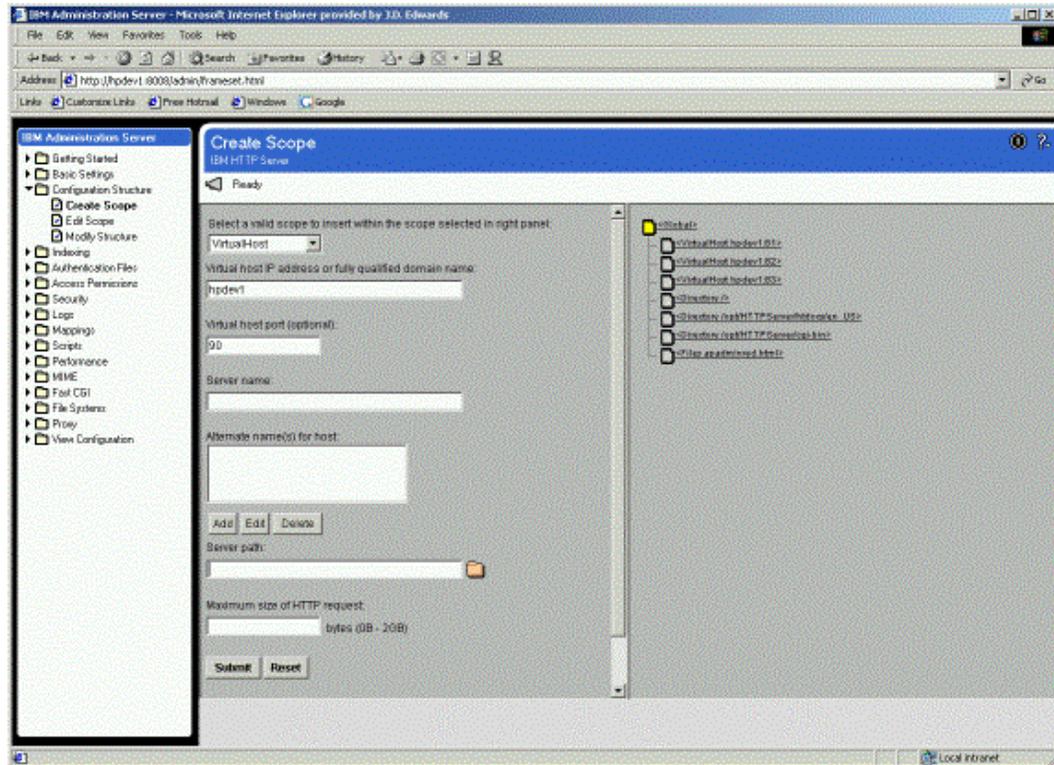
12. 左ペインで[Configuration Structure(構成の体系)]を展開し、[Create Scope(スコープの作成)]をクリックします。



13. [Create Scope]で、次のフィールドに値を入力して、アプリケーション・サーバーに関連付ける仮想ホストを作成します。

- ・ [IP Address]または[Server Name]
- ・ [Port number(ポート番号)]
- ・ [Alternate names(代替名)]:このセクションでは、ポートのエイリアスを作成できます。

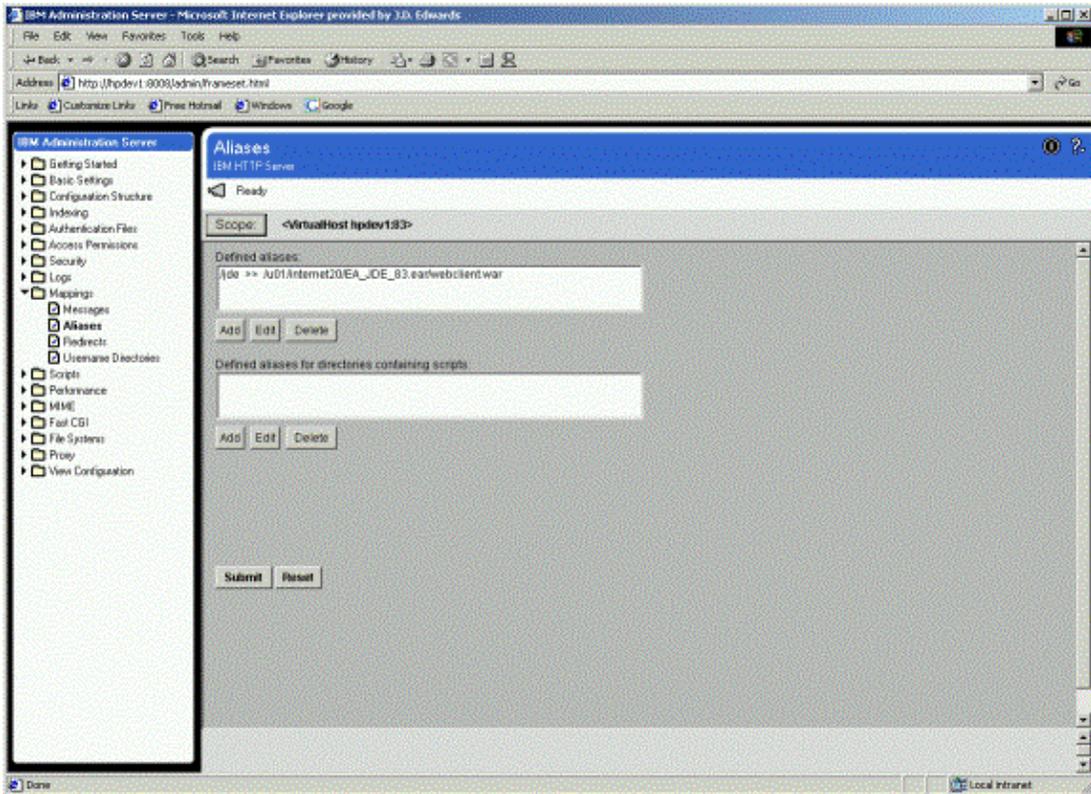
たとえば、このセクションにサーバー名とサーバー名ドメインを入力できます。



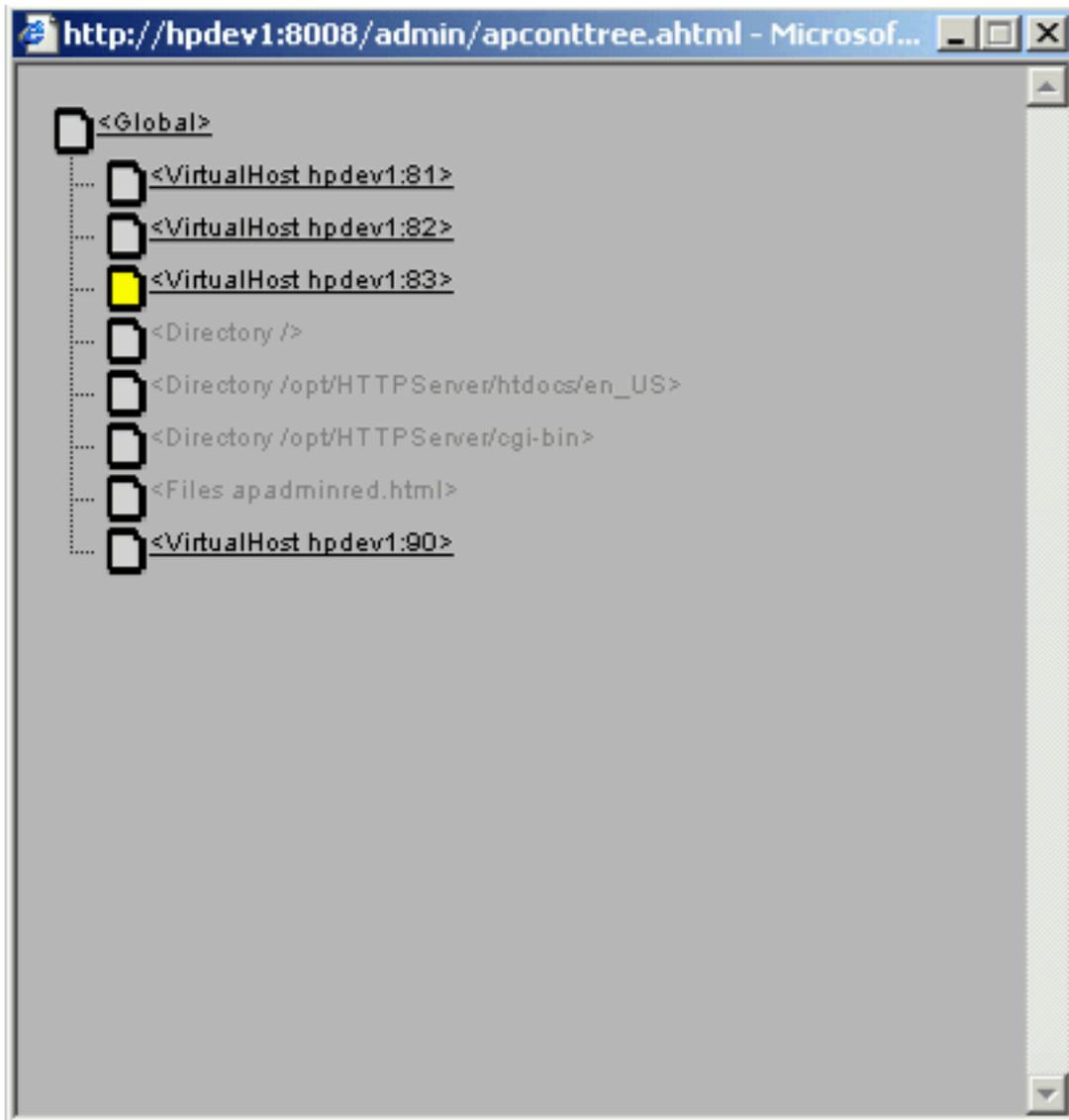
14. [Submit]をクリックします。

仮想ホストが作成されます。

15. エイリアスを設定するには、左ペインで[Mappings(マッピング)]を展開し、[Aliases(エイリアス)]をクリックします。

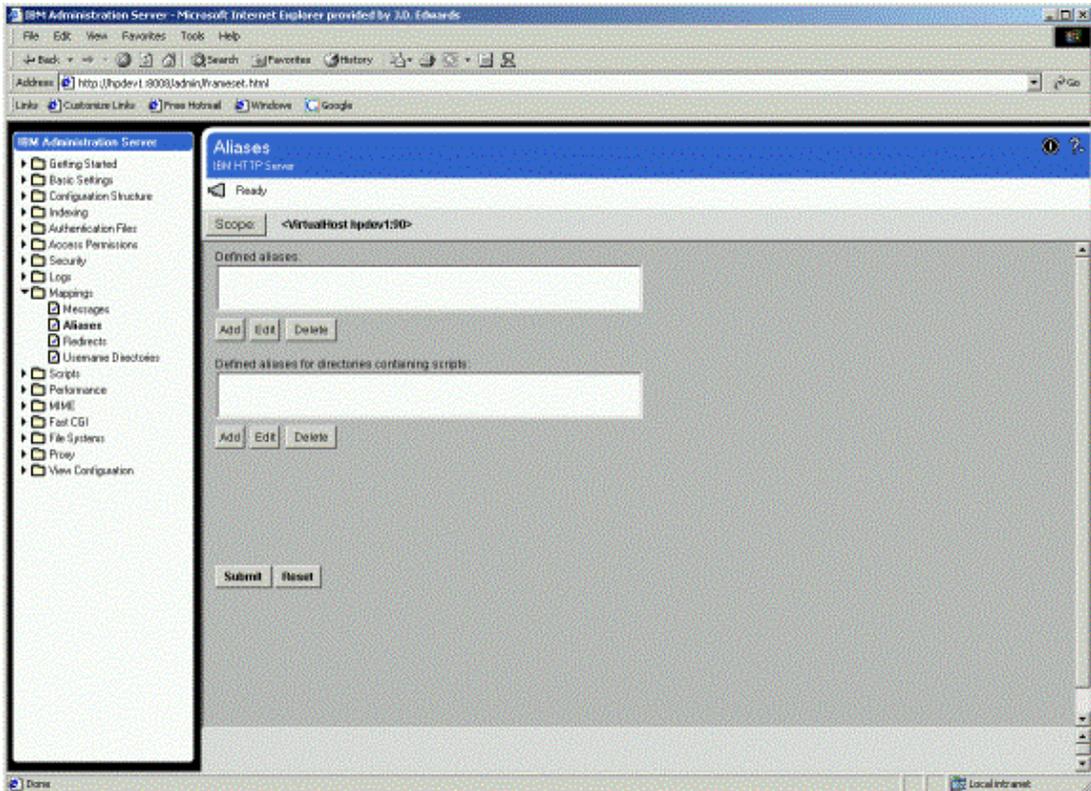


16. [Aliases]で、画面上部の[Scope(スコープ)]をクリックします。

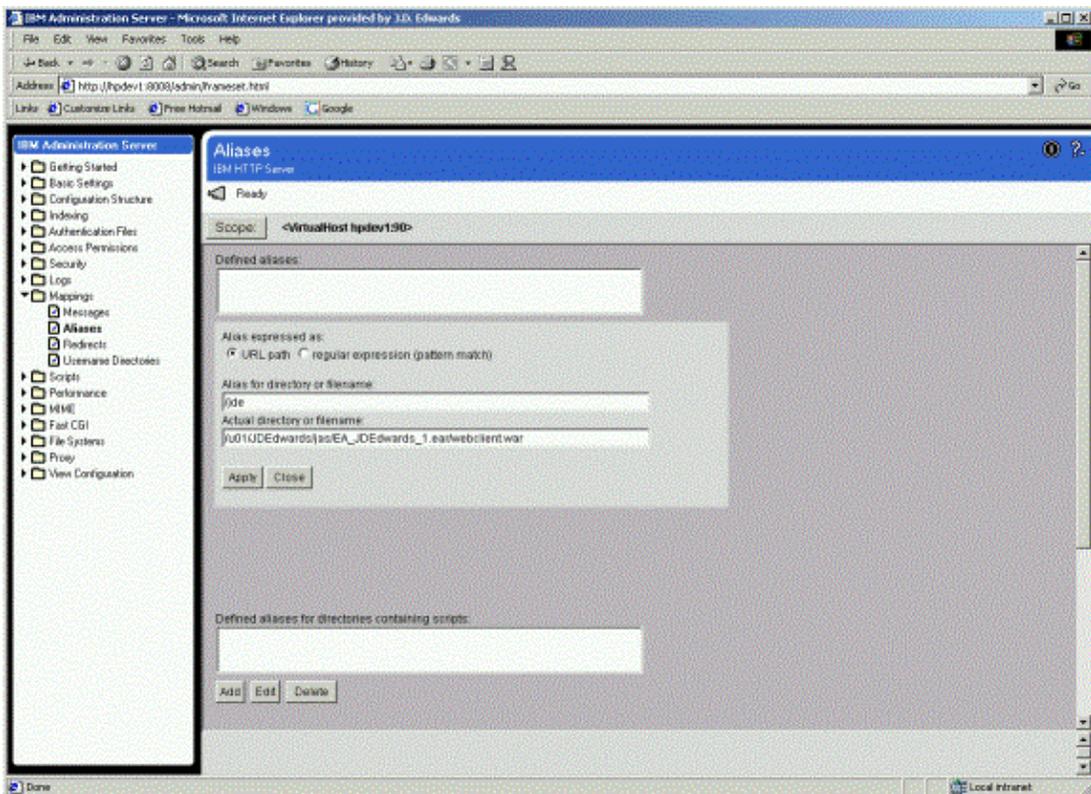


17. エイリアスを作成する仮想ホストを選択します。

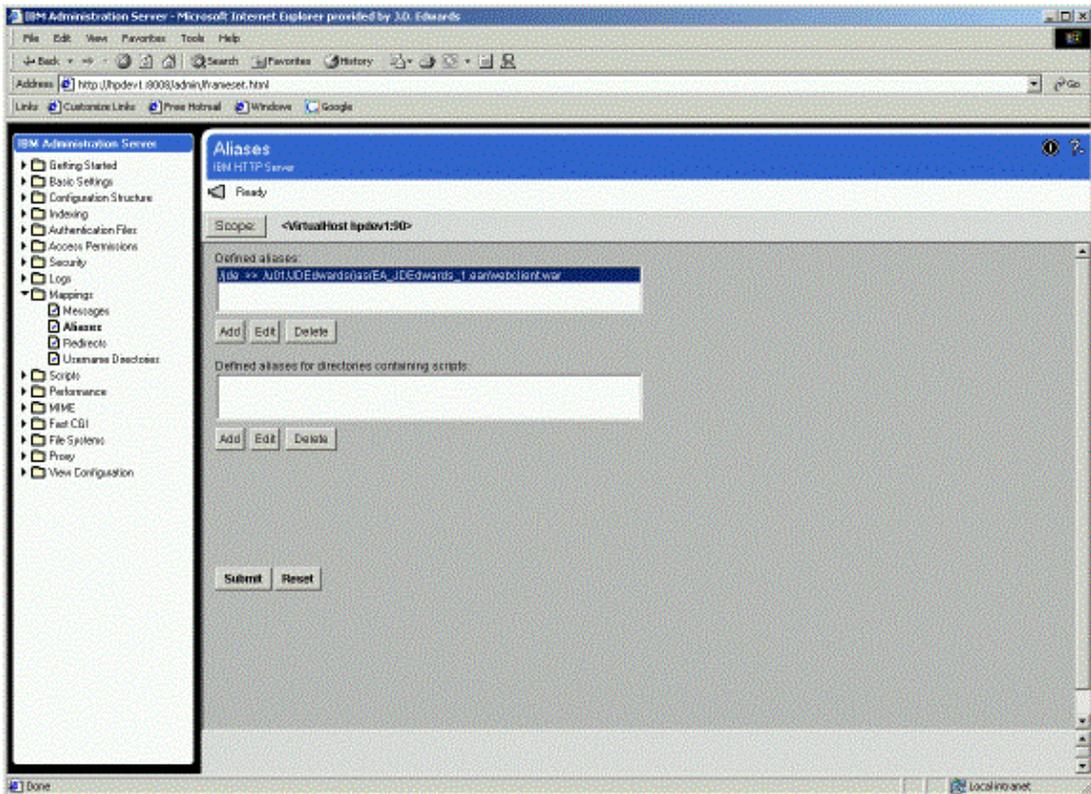
たとえば、[hpdev1:90]を選択します。



18. [Aliases]で、[Add]をクリックしてエイリアスを追加します。

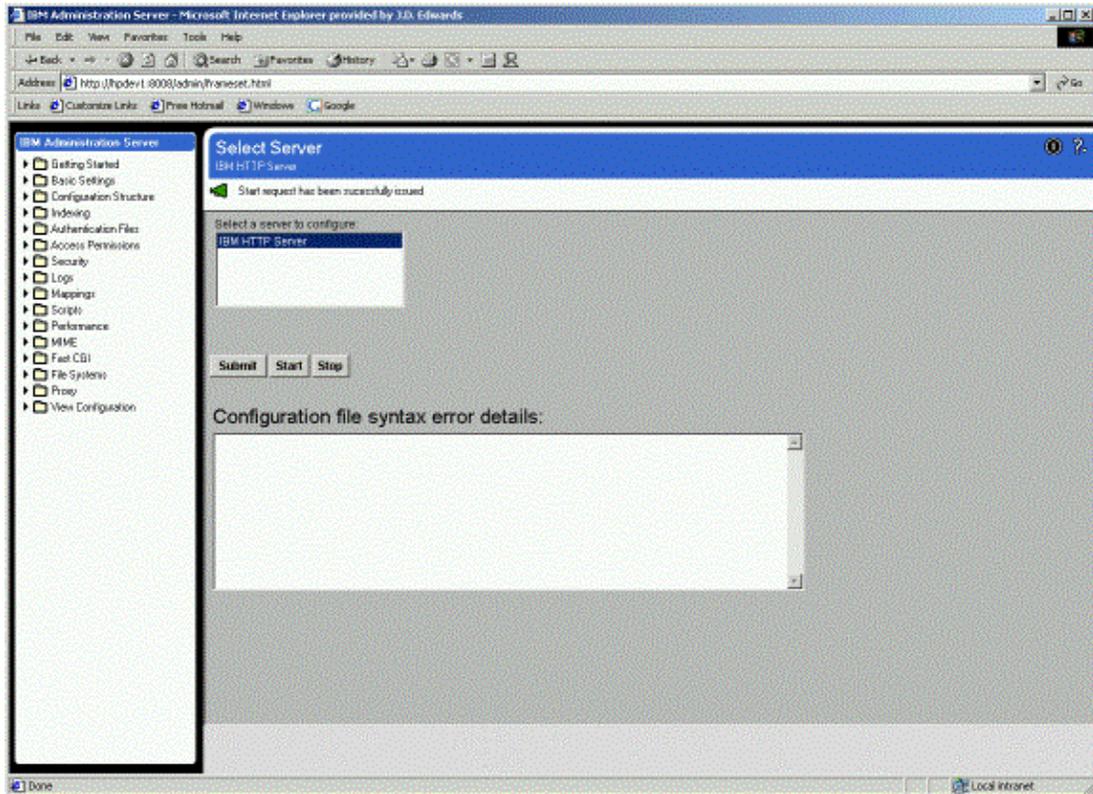


19. [Apply]、[Close(閉じる)]の順にクリックします。
20. スクロール・ダウンして[Submit]をクリックします。



これでHTTPサーバーが構成されました。

21. HTTPサーバーを起動するには、左ペインで[Getting Started]->[Select Server(サーバーの選択)]を選択します。[IBM HTTP Server]を選択して[Start(起動)]をクリックし、サーバーを起動できます。



22. ブラウザから“http://servername”と入力します。

[IBM Welcome (IBMへようこそ)]ページが表示されます。

付録 H. WebSphereリポジトリとしてのiSeries DB2の使用

ここでは、iSeries DB2データベース上でWebSphereリポジトリをWindowsおよびUNIX Webサーバー用にセットアップする方法について説明します。iSeriesサーバー上と同様に、リポジトリを中央に配置すると、複数のリポジトリを中央にまとめて配置して定期的にバックアップを作成できます。この構成では、リポジトリを安定的なプラットフォームで管理することで、可用性を高めてフェイルオーバーを1箇所にまとめることができます。

ここでは、WebSphereリポジトリを別のマシンに配置することによる応答時間の増大は考慮していませんが、増大するとしても最小限であると予想されます。

WebSphereアプリケーション・サーバーを別のプラットフォーム（WindowsまたはUNIX）で実行している場合に、iSeries DB2上のWebSphereリポジトリを作成して利用するには、次のタスクを実行する必要があります。

- ・ [iSeries上でのWASデータベースの作成 - ページ \[177\]](#)
- ・ [WebSphereのインストール - ページ \[178\]](#)
- ・ [WebSphereの構成 - ページ \[179\]](#)

注意

次の各タスクでは、Windowsプラットフォーム上で実行中のWebSphereアプリケーション・サーバーを例として使用していますが、どのUNIXプラットフォームの場合もプロセスはほぼ同じです。

各タスクの後に、Initial.Setup.Configおよびadmin.configファイルの例を示します。

- ・ [例:Initial.Setup.Config - ページ \[181\]](#)
- ・ [例: - ページ \[182\]](#)

iSeries上でのWASデータベースの作成

WebSphereアプリケーション・サーバーをWindowsまたはUNIXプラットフォームにインストールする前に、次のステップを実行します。

▶ iSeries上でWASデータベースを作成するには

1. iSeriesサーバー上で、DB2にアクセスしてSQLを実行できる有効なユーザーIDを使用してログインします。
2. iSeriesのコマンド行からSTRSQLコマンドを実行してSQLを起動します。

SQLセッションで次のコマンドを入力します。

CREATE COLLECTION WAS40

WAS40は、WebSphereリポジトリの格納用に作成する新規ライブラリ/コレクションです。

3. SQLセッションを終了します。
4. iSeriesセッションを終了します。

WebSphereのインストール

WebサーバーにWebSphereをインストールする間に、次の手順を実行する必要があります。

▶ WebSphereをインストールするには

1. 「IBM HTTPサーバーとWebSphere (バージョン4.0) のインストール」の説明に従ってWebSphereアプリケーション・サーバーのインストールを開始します。
2. [Install Options]画面で[Custom Installation]を選択します。
3. プラグインを更新するWebサーバーを含め、通常選択するオプションをすべて選択します。
4. [Database Options]画面で、次のフィールドを選択または自動入力します。
 - ・ [Database Type]: "DB2"
 - ・ [Remote Database]: 有効化
 - ・ [Database Name]: "WAS40" (iSeriesの最初のセクションで設定したコレクション)
 - ・ [Database User ID]: "JDE"
(iSeriesのユーザーID)
 - ・ [Password]:
(iSeriesのユーザーIDのパスワード)
 - ・ [Path]: "C:\¥WebSphere¥Appserver¥lib¥jt400.jar"
iSeries DB2 JDBCドライバ\jt400.jarのパスとファイル。

Database Options

IBM WebSphere Application Server uses a database repository to store information. Indicate the type and name of the database you would like to use, along with the location, user name, and password for the database.

Database Type: DB2 Remote Database

Database Name: WAS40

Database User ID: JDE

Password: ****

Path: \\WebSphere\\Appserver\\lib\\jt400.jar

URL:

Server:

Port:

< Back Next > Cancel

5. インストール手順を完了します。
6. インストール後にサーバーを再起動する前に、次のセクションのタスクを完了してください。

WebSphereの構成

WebSphereアプリケーション・サーバーをWindowsまたはUNIXプラットフォームにインストールした後に、次のステップを実行します。

► WebSphereを構成するには

1. jt400.jarファイルをiSeries(/qibm/proddata/http/public/jt400/lib/jt400.jar)から C:¥WebSphere¥Appserver¥libディレクトリにコピーします。

このファイルはIBM.com [http://www-1.ibm.com/servers/eserver/iseriess/toolbox/downloads.htm] で利用できるjtopenダウンロードから入手できますが、そのソースをリポジトリとして使用することはお勧めしません。

2. C:¥WebSphere¥Appserver¥propertiesディレクトリにあるinitial_setup.configファイルを開きます。

注意

このファイルの例については、「例: Initial.Setup.Config - ページ [181]」を参照してください。

3. このファイルを次のように編集します。

初期設定	変更後の設定
<pre><name>jdbcDriver</name> <value>C:/WebSphere/Appserver/lib/jt400.jar/ java/db2java.zip< /value></pre>	<pre><name>jdbcDriver</name> <value>C:/WebSphere/Appserver/lib/jt400.jar; </value></pre>

4. C:\WebSphere\Appserver\bin directoryディレクトリにあるadmin.configファイルを開きます。

注意

このファイルの例については、「例: - ページ [182]」を参照してください。

5. このファイルを次のように編集します。

初期設定	変更後の設定
<pre>com.ibm.ejs.sm.util.process.Nanny. adminServerJvmArgs=... C:/WebSphere/Appserver/lib/jt400.jar/java/ db2java.zip</pre>	<pre>com.ibm.ejs.sm.util.process.Nanny. adminServerJvmArgs=... C:/WebSphere/Appserver/lib/jt400.jar</pre>
<pre>com.ibm.ejs.sm.adminServer. dbdataSourceClassName= COM.ibm.db2.jdbc.DB2ConnectionPoolData Source</pre>	<pre>com.ibm.ejs.sm.adminServer. dbdataSourceClassName= com.ibm.as400.access. AS400JDBCCONNECTION PoolDataSource</pre>
<pre>com.ibm.ejs.sm.adminServer.dbserverName=</pre>	<pre>com.ibm.ejs.sm.adminServer.dbserverName= servername servernameはWebサーバー名です。</pre>
<pre>com.ibm.ejs.sm.adminServer.dbportNumber=</pre>	<pre>com.ibm.ejs.sm.adminServer.dbportNumber= portnumber portnumberはポート番号です。</pre>
<pre># URLはOracle com.ibm.ejs.sm.adminServer.db URL=</pre>	<pre># URLはOracle com.ibm.ejs.sm.adminServer.db URL= jdbc:as400: //iSeriesServer;prompt=false; user=jde;password=jde</pre>

初期設定	変更後の設定
にのみ有効です。	にのみ有効です。 iSeriesServerはiSeriesサーバー名で、user idとpasswordもiSeriesのユーザーIDとパスワードです。
# Create AdminServer database tables com.ibm.ejs.sm.adminServer.createTables=true	# Create AdminServer database tables com.ibm.ejs.sm.adminServer.createTables=false この構成が複数のノードを対象としている(各テーブルは既にコレクションで作成されている)場合は、このパラメータがfalseに設定されていることを確認します。
com.ibm.ejs.sm.adminServer.dbSchema= EJSADMIN.	com.ibm.ejs.sm.adminServer.dbSchema=WAS40 最初のセクションでiSeries上で作成されたコレクション。

6. サーバーを再起動します。
7. サーバーのブート後に、WebSphere Administrationサービスを開始します。

このサービスを開始できない場合は、WebSphereのログを調べてください。
8. Administrationサービスが開始 (Javaプロセスが実行)された場合は、Administrationコンソールを起動します。

コンソールに新規ノードが表示されます。
9. プロセスが正常終了した後、必要なWebSphere FixPackを適用します。

例: Initial.Setup.Config

```
<?xml version="1.0"?>
<!DOCTYPE initial-setup [ <!ELEMENT config-file (#PCDATA)>
<!ELEMENT variable (name,value)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT value (#PCDATA)>
<!ELEMENT admin-config-file (#PCDATA)>
<!ELEMENT initial-setup (config-file+,variable*,admin-config-file)>]]>
<initial-setup>
<config-file>C:/WebSphere/AppServer/properties/Default_Server.wascml</config-file>
<config-file>C:/WebSphere/AppServer/properties/ResourceProviders.xml</config-file>
<config-file>C:/WebSphere/AppServer/bin/ImportSamplesConfigDB2.xml</config-file>
<!-- sampleOracle
<config-file>C:/WebSphere/AppServer/bin/ImportSamplesConfigOracle.xml
</config-file> sampleOracle -->
<!-- sampleSybase
```

```

<config-file>C:/WebSphere/AppServer/bin/ImportSamplesConfigSybase.xml</config-file>
sampleSybase -->
<!-- sampleMerant
<config-file>C:/WebSphere/AppServer/bin/ImportSamplesConfigMerant.xml</config-file> sampleMerant -->
<!-- sampleInformix
<config-file>C:/WebSphere/AppServer/bin/ImportSamplesConfigInformix.xml</config-file>
sampleInformix -->
<!-- xml file for SE4.0 to AE 4.0 Migration -->
<!-- SEAEigrate
<config-file>C:/WebSphere/AppServer/properties/migratemof.xml</config-file> SEAEigrate -->
<!-- xml file for 3.x to 4.0 Migration --> <!-- WASPostUpgrade
<config-file>C:/WebSphere/AppServer/properties/websphere_4x_restore.xml</config-file>
WASPostUpgrade -->
<variable>
<name>XMLConfigDTDLocation</name>
<value>C:/WebSphere/AppServer/bin</value>
</variable>
<variable>
<name>serverName</name>
<value>Default Server</value>
</variable>
<variable>
<name>dbPrefix</name>
<value></value>
</variable>
<variable>
<name>jdbcDriver</name>
<value>C:/WebSphere/Appserver/lib/jt400.jar/java/db2java.zip</value>
</variable>
<variable>
<name>repositoryDBName</name>
<value>WAS40</value>
</variable>
<variable>
<name>pslash</name>
<value>%</value>
</variable>
<admin-config-file>C:/WebSphere/AppServer/bin/admin.config</admin-config-file>
</initial-setup>

```

例:

```

com.ibm.ejs.sm.adminServer.earFile=
C:/WebSphere/AppServer/config/admin.ear
com.ibm.ejs.sm.adminserver.classpath=
C:/WebSphere/AppServer/properties;
C:/WebSphere/AppServer/lib/bootstrap.jar
com.ibm.ejs.sm.util.process.Nanny.path=
C:%WebSphere%AppServer%bin;
C:%WebSphere%Appserver%lib%jt400.jar%bin;
C:%WebSphere%Appserver%lib%jt400.jar%function;
C:%WebSphere%AppServer%java%bin

```

```
com.ibm.ejs.sm.util.process.Nanny.adminServerJvmArgs=
-Xmx128m -Xminf0.15 -Xmaxf0.25 -Xms17M -Djavax.rmi.CORBA.UtilClass=com.ibm.CORBA.iiop.Util
-Dcom.ibm.CORBA.iiop.noLocalCopies=true
-Dws.ext.dirs=C:/WebSphere/AppServer/java/lib;C:/WebSphere/AppServer/classes;
C:/WebSphere/AppServer/lib;C:/WebSphere/AppServer/lib/ext;
C:/WebSphere/Appserver/lib/jt400.jar
com.ibm.ws.jdk.path=C:/WebSphere/AppServer/java
com.ibm.ejs.sm.util.process.Nanny.maxtries=3
com.ibm.ejs.sm.adminServer.traceFile=C:/WebSphere/AppServer/logs/tracefile
com.ibm.ejs.sm.util.process.Nanny.traceFile=C:/WebSphere/AppServer/logs/nanny.trace
com.ibm.ejs.sm.adminServer.logFile=C:/WebSphere/AppServer/tranlog/DEN-CS5830089_tranlog1,
C:/WebSphere/AppServer/tranlog/DEN-CS5830089_tranlog2
server.root=C:¥¥WebSphere¥¥AppServer was.install.root=C:/WebSphere/AppServer
com.ibm.CORBA.ConfigURL=file:/C:/WebSphere/AppServer/properties/sas.server.props
com.ibm.ejs.sm.adminServer.disablePMI=true
com.ibm.ejs.sm.util.process.Nanny.errtraceFile=C:/WebSphere/AppServer/logs/adminserver_stderr.log
install.initial.config=true
install.initial.config.file=C:/WebSphere/AppServer/properties/initial_setup.config
com.ibm.ejs.sm.adminServer.seriousEventLogSize=1000
com.ibm.itp.location=C:/WebSphere/AppServer/bin
# Trace settings
#com.ibm.ejs.sm.adminServer.traceString=com.ibm.ejs.*=all=enabled
#com.ibm.ejs.sm.adminServer.traceOutput=C:/WebSphere/AppServer/logs/admin.trace
# Repository database settings
com.ibm.ejs.sm.adminServer.dbdataSourceClassName=com.ibm.as400.access.AS400JDBCConnectionPoolDataSource
com.ibm.ejs.sm.adminServer.dbserverName=DENAS10 com.ibm.ejs.sm.adminServer.dbportNumber=8417
com.ibm.ejs.sm.adminServer.dbdatabaseName=WAS40 com.ibm.ejs.sm.adminServer.dbuser=JDE
com.ibm.ejs.sm.adminServer.dbpassword=JDE
com.ibm.ejs.sm.adminServer.dbdisable2Phase=true
# URL is only valid for Oracle com.ibm.ejs.sm.adminServer.dbURL=jdbc:as400://DENAS10;prompt=false;user=jde;password=jde
# informixLockModeWait is only valid for Informix
com.ibm.ejs.sm.adminServer.dbinformixLockModeWait=
# Connection pool size
#com.ibm.ejs.sm.adminServer.dbmaximumPoolSize=10
# Create AdminServer database tables and exit
#com.ibm.ejs.sm.adminServer.initializeDb=true
# Create AdminServer database tables
com.ibm.ejs.sm.adminServer.createTables=false
# AdminServer's DrAdmin port
#com.ibm.ejs.sm.adminServer.diagThreadPort=-1
# Disable restart of application servers on AdminServer startup
#com.ibm.ejs.sm.adminServer.disableAutoServerStart=true
# Enable WLM of AdminServers com.ibm.ejs.sm.adminServer.wlm=true
# Custom Services
# A list of the files containing configurations for Custom Services
# to be added to all application servers defined in the domain
com.ibm.websphere.preconfiguredCustomServices=C:/WebSphere/AppServer/config/plugin-cfg-service.xmi
com.ibm.ejs.sm.adminServer.qualifyHomeName=true
com.ibm.ejs.sm.adminServer.dbSchema=WAS40
```


付録 I. Webサーバー上のメディア・オブジェクトの理解

このセクションでは、Webサーバー上のメディア・オブジェクトへのアクセスに必要なjas.ini設定の概要と、Webサーバーがこれらのオブジェクトにネットワークからアクセスするプロセスについて説明します。

必要なjas.ini設定

jas.iniファイルの[OWWEB]セクションで、次のパラメータが設定されていることを確認してください。

[OWWEB]		
パラメータ	推奨設定	説明
MO QUEUE=	サイト依存のパス	<p>Webサーバー上のメディア・オブジェクト・ディレクトリの位置を識別します。このパスは、Webブラウザ用の仮想パス/jde/moqueue/に変換する必要があります。</p> <p>Webサーバー・プロセスには、このパスの読取り権限と作成権限が必要です。</p> <hr/> <p>注意</p> <p>このディレクトリの作成を必要とする場合があります。</p> <hr/> <p>デフォルト値は次のとおりです。</p> <p>/u01/JDEdwards/JAS/EA_JAS_80.ear/webclient.war/moqueue</p>
FtpPort=	21	FTP用のデフォルト・ポートを指定します。
FtpUsr=	anonymous	メディア・オブジェクト・ファイル・サーバーへのFTPアクセスに使用するユーザーIDを指定します。
FtpPwd=	anonymous	メディア・オブジェクト・ファイル・サーバーへのFTPアクセスに使用するパスワードを指定します。
UseMOWinNT Share=	FALSE	WebサーバーでWindowsのファイル共有メカニズムの代わりにFTPアクセスを使用するように指定します。

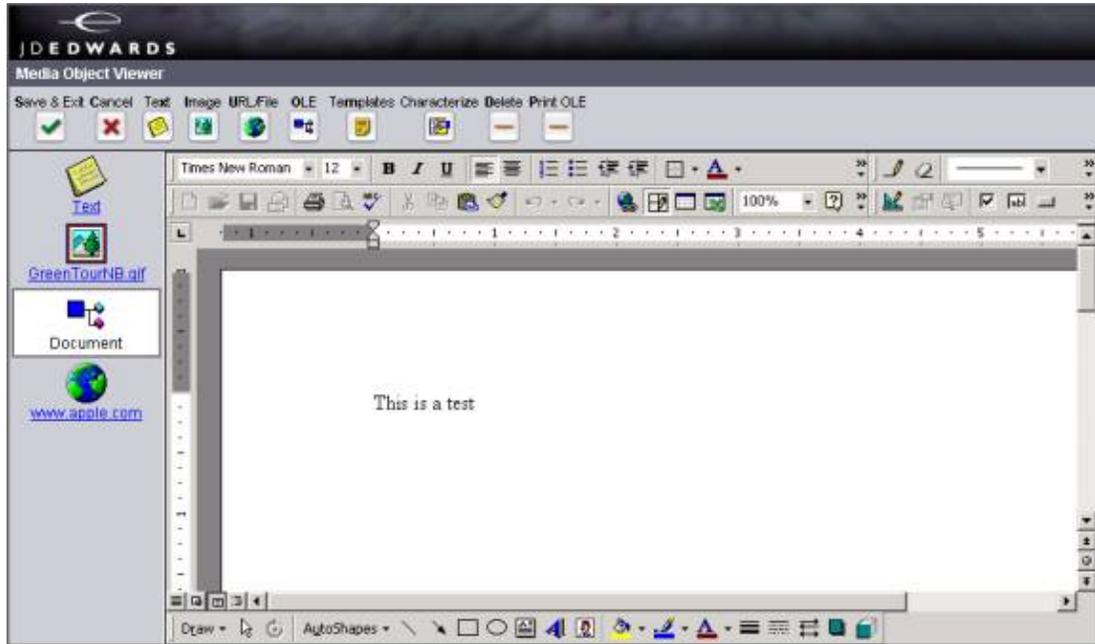
Webサーバーでのメディア・オブジェクトの表示

ここでは、WebサーバーからHTMLクライアントに、メディア・オブジェクトがどのように送信されるかについて説明します。

1. ユーザーが、メディア・オブジェクトのサポート機能を持つアプリケーションで添付ファイルを選択しま

す。

2. [Media Object(メディア・オブジェクト)]ウィンドウにイメージ、ファイルまたはOLEオブジェクトが表示され、ユーザーがそこから1つ選択します。



3. Webサーバーに要求が送信されます。
4. Webサーバーでは、メディア・オブジェクト待ち行列テーブル(F98MOQUEUE)からオブジェクトの位置が読み取られ、そのファイルが検索され、MO QUEUEのパスで指定された位置にキャッシュされます。
ファイルはFTPプロトコルを使用して転送されます。

次の図は、このプロセスを示します。

