

Cosminexus アプリケーション開発ガイド

手引・操作書

3020-3-M41-60

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参
照ください。

対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) , Windows Server 2003 R2 (x64)

P-2443-7D74 uCosminexus Application Server Standard 07-60

P-2443-7K74 uCosminexus Application Server Enterprise 07-60

P-2443-7M74 uCosminexus Web Redirector 07-60

P-2443-7S74 uCosminexus Service Platform 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Vista , Windows XP

P-2443-7E74 uCosminexus Developer Standard 07-60

P-2443-7F74 uCosminexus Developer Professional 07-60

P-2443-7T74 uCosminexus Service Architect 07-60

P-2443-7U74 uCosminexus Operator 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) , Windows Server 2003 R2 (x64) , Windows Vista , Windows XP

P-2443-7H74 uCosminexus Client 07-60

適用 OS : AIX 5L V5.2 , AIX 5L V5.3

P-1M43-7D71 uCosminexus Application Server Standard 07-60

P-1M43-7K71 uCosminexus Application Server Enterprise 07-60

P-1M43-7M71 uCosminexus Web Redirector 07-60

P-1M43-7S71 uCosminexus Service Platform 07-60

適用 OS : HP-UX 11i V2 (IPF) , HP-UX 11i V3 (IPF)

P-1J43-7D71 uCosminexus Application Server Standard 07-60

P-1J43-7K71 uCosminexus Application Server Enterprise 07-60

P-1J43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (x86) , Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux ES 3 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T)

P-9S43-7D71 uCosminexus Application Server Standard 07-60

P-9S43-7K71 uCosminexus Application Server Enterprise 07-60

P-9S43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (x86) , Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux ES 3 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 (AMD/Intel 64)

P-9S43-7S71 uCosminexus Service Platform 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (IPF) , Red Hat Enterprise Linux AS 4 (IPF) , Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium)

P-9V43-7D71 uCosminexus Application Server Standard 07-60

P-9V43-7K71 uCosminexus Application Server Enterprise 07-60

P-9V43-7M71 uCosminexus Web Redirector 07-60

適用 OS : Solaris 9 , Solaris 10

P-9D43-7D71 uCosminexus Application Server Standard 07-60

P-9D43-7K71 uCosminexus Application Server Enterprise 07-60

P-9D43-7M71 uCosminexus Web Redirector 07-60

P-9D43-7S71 uCosminexus Service Platform 07-60

印の製品については、サポート時期をご確認ください。

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

本製品では日立トレース共通ライブラリをインストールします。

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

AMD は、Advanced Micro Devices, Inc. の商標です。

Borland のブランド名および製品名はすべて、米国 Borland Software Corporation の米国およびその他の国における商標または登録商標です。

cBank, cContentsTemplate, cFramework, cInputCheck, cLogWrapper, cPersistence, cSQLBuilder, cValidator, Eclipse Plugin For cFramework は、株式会社イーシー・ワンの日本における商標または登録商標です。

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

GIF は、米国 CompuServe Inc. が開発したフォーマットの名称です。

gzip は、米国 FSF(Free Software Foundation) が配布しているソフトウェアです。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

IIOP は、OMG 仕様による ORB(Object Request Broker) 間通信のネットワークプロトコルの名称です。

Intel は、Intel Corporation の会社名です。

Internet Explorer は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

Itanium は、アメリカ合衆国および他の国におけるインテル コーポレーションまたはその子会社の登録商標です。

Java 及びすべての Java 関連の商標及びロゴは、米国及びその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

JavaScript は、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

JBoss および Hibernate は、Red Hat, Inc. の登録商標です。

JDK は、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

JSTL は、The Jakarta Project のタグライブラリ名称です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標あるいは商標です。

Matisse は、Sun Microsystems, Inc. の GUI ビルダ機能名称です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Internet Information Services は、米国 Microsoft Corp. の商品名称です。
Microsoft SQL Server は、米国 Microsoft Corp. の商品名称です。
Mozilla は、Mozilla Foundation の、米国およびその他の国における商標です。
MyEclipse は、米国 Genuitec 社の商品名称です。
OMG , CORBA , IIOP , UML , Unified Modeling Language , MDA , Model Driven Architecture は、Object Management Group, Inc. の米国及びその他の国における登録商標または商標です。
ORACLE は、米国 Oracle Corporation の登録商標です。
Oracle は、米国 Oracle Corporation 及びその子会社、関連会社の登録商標です。
Oracle9i は、米国 Oracle Corporation の商標です。
Oracle 10g は、米国 Oracle Corporation の商標です。
PA-RISC は、米国 Hewlett-Packard Company の商標です。
Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標若しくは商標です。
Solaris は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。
Spring Framework は、Interface21 のアプリケーションフレームワーク名称です。
SQL Server は、米国法人 Sybase, Inc. の商標です。
Sun , Sun Microsystems , Java は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。
Sun Microsystems は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。
Tapestry は、Apache Software Foundation のオープンソースフレームワーク名称です。
UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。
Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。
Windows Server は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。
Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。
X/Open は、X/Open Company Limited の英国ならびに他の国における登録商標です。
Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc. により構築された開発ツール統合のためのオープンプラットフォームです。
Struts は Apache Software Foundation が運営する Apache Struts Project が公開するサーブレット・JSP の Web アプリケーション構築用のフレームワークです。
This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).
プログラムプロダクト「P-9D43-7D71, P-9D43-7K71, P-9D43-7M71, P-9D43-7S71」には、米国 Sun Microsystems, Inc. が著作権を有している部分が含まれています。
プログラムプロダクト「P-9D43-7D71, P-9D43-7K71, P-9D43-7M71, P-9D43-7S71」には、UNIX System Laboratories, Inc. が著作権を有している部分が含まれています。

発行

2006 年 4 月（第 1 版）3020-3-M41

2007 年 12 月（第 4 版）3020-3-M41-60

著作権

All Rights Reserved. Copyright (C) 2006, 2007, Hitachi, Ltd.

変更内容

変更内容 (3020-3-M41-60) uCosminexus Application Server Standard 07-60 , uCosminexus Developer Standard 07-60 , uCosminexus Developer Professional 07-60 , uCosminexus Client 07-60 , uCosminexus Application Server Enterprise 07-60 , uCosminexus Web Redirector 07-60 , uCosminexus Service Platform 07-60 , uCosminexus Service Architect 07-60 , uCosminexus Operator 07-60

追加・変更内容	変更箇所
開発環境インスタントセットアップ機能に対応した。	1.1.4 , 1.2.2 , 1.4.1 , 1.4.2 , 1.5.2 , 1.6.1 , 2 章 , 3 章 , 7.1 , 7.2 , 7.8.2 , 付録 A , 付録 B , 付録 C.2 , 付録 E , 付録 I , 付録 K
エンタープライズアプリケーションプロジェクト作成時にモジュールプロジェクトが一つ以上含まれる必要があることを追加した。	4.1.1
プロジェクト作成時に指定するディレクトリの注意事項を追加した。	4.2.1 , 4.3.1 , 4.4.1
MyEclipse の GUI の変更に対応した。	4.2.1 , 4.3.1 , 4.4.1 , 5.2 , 5.3 , 6.5 , 7.3.1 , 7.4.1 , 7.4.2 , 7.5.1 , 付録 A , 付録 D
エンタープライズアプリケーションプロジェクト作成時の注意事項に、リソースの移動および名前変更を禁止することを追加した。	4.4.2
スタートメニューのショートカットを使用した、組み込みデータベースの開始および停止の手順を追加した。	7.2
MyEclipse がサポートする UML 図のエディタからコンポーネント図を削除し、ステートチャート図を追加した。	9.2.2
Eclipse を使用しない開発環境の適用 OS に Solaris を追加した。	9.3.1
Developer Standard と Developer Professional の機能差に、サーバ管理コマンドと J2EE サーバの通信に使用するポートの固定方法を追加した。	付録 C.2
次の製品の適用 OS に、Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , および Red Hat Enterprise Linux 5 (AMD/Intel 64) を追加した。 • uCosminexus Service Platform	-
次の製品の適用 OS に Linux (IPF) を追加した。 • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector	-

追加・変更内容	変更箇所
次の製品の適用 OS に Solaris を追加した。 <ul style="list-style-type: none"> • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector • uCosminexus Service Platform 	-
HP-UX (PA-RISC) のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。 <ul style="list-style-type: none"> • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector 	-
次の製品の適用 OS から HP-UX (IPF) を削除した。 <ul style="list-style-type: none"> • uCosminexus Service Platform 	-

単なる誤字・脱字などはお断りなく訂正しました。

第 4 版では第 3 版の内容の一部を移動し、目次の構成を変更しました。第 3 版との対応は次のようになっています。

第 3 版	第 4 版
1. J2EE アプリケーション開発の概要	1. J2EE アプリケーション開発の概要
2. インストール	2. インストールとセットアップ
2.1 インストールの流れ	2.1 インストールとセットアップの流れ
2.2 Developer のインストール	2.2 Developer のインストール
2.2.1 インストール後のディレクトリ構成	2.2.1 インストール後のディレクトリ構成
2.2.2 Cosminexus を再インストールしたときの注意事項	2.2.3 Cosminexus を再インストールしたときの注意事項
2.3 Eclipse のインストール	付録 B.1(1) Eclipse のインストール
2.4 MyEclipse のインストール	付録 B.1(2) MyEclipse のインストール
2.5 プラグインの組み込み	付録 B.1(4) プラグインの組み込み
2.6 アンインストール	付録 B.5 アンインストール
2.6.1 プラグインのアンインストール	付録 B.5(1) プラグインのアンインストール
2.6.2 MyEclipse のアンインストール	付録 B.5(2) MyEclipse のアンインストール
2.6.3 Developer のアンインストール	2.8 Developer のアンインストール
3. 開発環境の構築	削除
3.1 開発環境の構築の流れ	削除
3.2 環境変数の設定	2.2.2 環境変数の設定
3.3 Eclipse の設定	2.4.1 Eclipse の設定
3.4 MyEclipse の設定	削除
3.4.1 ライブラリパスの設定	付録 B.2(1) MyEclipse を利用するための設定

第 3 版	第 4 版
3.4.2 プロキシの設定	2.5.1 MyEclipse のプロキシの設定
3.5 テスト用実行環境のシステム構築	付録 B.2(2) テスト用実行環境のシステム構築
3.5.1 システム構築の概要	付録 B.2(2) テスト用実行環境のシステム構築
3.5.2 Management Server の設定	削除
3.5.3 簡易構築定義ファイルの作成	削除
3.5.4 システムの構築	削除
3.5.5 JavaVM のセキュリティポリシーの設定	付録 B.2(2)(b) JavaVM のセキュリティポリシーの設定
3.6 プラグインを使用するための設定	付録 B.2(3) プラグインを使用するための設定
3.6.1 Management Server のリモート管理機能の設定	付録 B.2(3)(a) Management Server リモート管理機能の設定
3.6.2 コンソールへの情報出力の設定	付録 B.2(3)(b) コンソールへの情報出力の設定
3.6.3 プラグイン使用不可の設定	削除
3.7 Cosminexus コネクタの設定	付録 B.2(4) Cosminexus コネクタの設定
3.7.1 デプロイ作業ディレクトリの設定	2.5.2 デプロイ作業ディレクトリの設定変更
3.7.2 ソケット操作のブロックのタイムアウト設定	2.5.3 ソケット操作のブロックのタイムアウト設定変更
3.7.3 接続先ホストの設定変更	付録 B.2(4)(a) 接続先ホストの設定変更
3.7.4 Management Server のリモート管理機能へのログインおよびログアウト (MyEclipse)	2.4.2 Management Server リモート管理機能へのログインおよびログアウト (MyEclipse)
3.7.5 J2EE サーバの選択	2.4.3 J2EE サーバの選択
3.7.6 J2EE サーバ起動時の動作の設定	付録 B.2(4)(b) J2EE サーバ起動時の動作の設定
3.7.7 コンソールの設定 (Eclipse)	2.5.4(1) Eclipse のコンソールの設定
3.8 Server Plug-in の設定	2.4.4 Server Plug-in の設定
3.8.1 接続先ホストの設定	2.4.4(1) 接続先ホストの設定
3.8.2 Management Server のリモート管理機能へのログインおよびログアウト (Server Plug-in)	2.4.4(2) Management Server リモート管理機能へのログインおよびログアウト (Server Plug-in)
3.8.3 コンソールの設定 (Server Plug-in)	2.5.4(2) Server Plug-in のコンソールの設定
3.8.4 アプリケーション統合属性ファイルのエディタの設定	2.5.5 アプリケーション統合属性ファイルのエディタの設定
4. テスト用実行環境で使用するデータベースの構築	3. テスト用実行環境で使用するデータベースのテーブルの作成
4.1 組み込みデータベース構築の流れ	3.1 組み込みデータベースのテーブルの作成の流れ
4.2 HiRDB SQL Executer のインストール	3.2 HiRDB SQL Executer のインストール
4.3 組み込みデータベースの構築	付録 B.3 組み込みデータベースの構築

第 3 版	第 4 版
4.4 組み込みデータベースの操作	3.3 組み込みデータベースの操作
4.5 RD エリアの追加と削除	3.4 RD エリアの追加と削除
4.6 組み込みデータベースの動作制御の設定	3.5 組み込みデータベースの動作制御の設定
4.7 組み込みデータベースの削除	付録 B.3(3) 組み込みデータベースの削除
5. プロジェクトの作成	4. プロジェクトの作成
6. ユーザインタフェースの作成	5. ユーザインタフェースの作成
7. 業務処理プログラムの作成	6. 業務処理プログラムの作成
8. J2EE アプリケーションのテスト	7. J2EE アプリケーションのテスト
9. 実行環境への J2EE アプリケーションの配布	8. 実行環境への J2EE アプリケーションの配布
10. そのほかの開発	9. そのほかの開発
11. JSP 事前コンパイルとチェック	10. JSP 事前コンパイルとチェック
付録 A MyEclipse の動作確認用サンプルプログラム	付録 A MyEclipse の動作確認用サンプルプログラム
付録 A.1 サンプルプログラムの概要	付録 A.1 サンプルプログラムの概要
付録 A.2 組み込みデータベースの構築	削除
付録 A.3 組み込みデータベースの開始	付録 A.2 組み込みデータベースの開始
付録 A.4 サンプルプログラムの実行手順	付録 A.3 サンプルプログラムの実行手順
付録 B Developer Standard 使用時の注意事項	付録 C Developer Standard 使用時の注意事項
付録 C Matisse4MyEclipse フィーチャのインストール	付録 D Matisse4MyEclipse フィーチャのインストール
付録 D テスト環境のシステムのチューニング	付録 E テスト用実行環境のシステムのチューニング
付録 E Cosminexus Version 6 から Cosminexus Version 7 への移行	付録 F Cosminexus Version 6 から Cosminexus Version 7 への移行
付録 F JavaMail の使用例	付録 G JavaMail の使用例
付録 G web.xml 編集時の注意事項（互換用の動作モードの場合）	付録 H web.xml 編集時の注意事項（互換用の動作モードの場合）
付録 H データベース構築・削除時のエラーメッセージ一覧	付録 J データベース構築・削除時のエラーメッセージ一覧
付録 I 用語解説	付録 K 用語解説

変更内容（3020-3-M41-40）uCosminexus Application Server Standard 07-50，uCosminexus Developer Standard 07-50，uCosminexus Developer Professional 07-50，uCosminexus Client 07-50，uCosminexus Application Server Enterprise 07-50，uCosminexus Web Redirector 07-50，

uCosminexus Service Platform 07-50 , uCosminexus Service Architect 07-50 , uCosminexus Operator 07-50

追加・変更内容
J2EE アプリケーションのテスト用実行環境の構築方法を追加した。
リソースアダプタを J2EE アプリケーションに含めてデプロイする方式をサポートした。
開発環境に Windows Vista を追加した。
Windows Vista 使用時の注意事項を追加した。
mngsvrctl コマンドで管理者アカウントの設定ができるようになった。
簡易構築定義ファイルのスキーマのバージョンが 2.5 になった。これに伴い、タグの記述形式が次のとおり変更になった。 <ul style="list-style-type: none">簡易構築定義ファイルの <host-ref> タグに @myhost が設定できるようになった。<host> タグ下の <name> タグの名称が <host-name> タグに変更になった。 <operational-ip> タグの名称が <agent-host> タグに変更になった。
cmx_build_system コマンドでモデル登録およびシステム構築ができるようになった。
テーブル定義ファイルのサンプルを変更した。
Connector1.5 に対応した。
RMI-IIOP 通信を使用した Enterprise Bean のメソッド呼び出しに Unicode の補助文字を送受信できるようになった。
Java ソースファイルにアノテーションを記載する方法を追加した。
EJB ホームオブジェクトへの再接続機能について追加した。
Matisse4MyEclipse フィーチャのインストールについて追加した。
テスト環境のシステムのチューニングについて追加した。
AIX 5L V5.1 のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。 <ul style="list-style-type: none">uCosminexus Application Server EnterpriseuCosminexus Application Server StandarduCosminexus Service PlatformuCosminexus Web Redirector
Microsoft(R) Windows(R) 2000 Advanced Server Operating System , Microsoft(R) Windows(R) 2000 Datacenter Server Operating System , および Microsoft(R) Windows(R) 2000 Server Operating System のサポート中止に伴い、これらの OS に適応していた次の製品の記述を削除した。 <ul style="list-style-type: none">uCosminexus Application Server EnterpriseuCosminexus Application Server StandarduCosminexus ClientuCosminexus Developer ProfessionaluCosminexus Developer StandarduCosminexus OperatoruCosminexus Service ArchitectuCosminexus Service PlatformuCosminexus Web Redirector

追加・変更内容

Microsoft(R) Windows(R) 2000 Professional Operating System のサポート中止に伴い、この OS に適応していた次の製品の記述を削除した。

- uCosminexus Client
 - uCosminexus Developer Professional
 - uCosminexus Developer Standard
 - uCosminexus Operator
 - uCosminexus Service Architect
-

次の製品の適用 OS に、HP-UX (IPF) を追加した。

- uCosminexus Service Platform
-

次の製品の適用 OS に、HP-UX 11i V3 (IPF) を追加した。

- uCosminexus Application Server Enterprise
 - uCosminexus Application Server Standard
 - uCosminexus Web Redirector
-

次の製品の適用 OS に、Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T), および Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) を追加した。

- uCosminexus Application Server Enterprise
 - uCosminexus Application Server Standard
 - uCosminexus Service Platform
 - uCosminexus Web Redirector
-

次の製品の適用 OS に、Windows Vista を追加した。

- uCosminexus Client
 - uCosminexus Developer Professional
 - uCosminexus Developer Standard
 - uCosminexus Operator
 - uCosminexus Service Architect
-

uCosminexus Application Server Standard 07-20, uCosminexus Application Server Enterprise 07-20, uCosminexus Developer Standard 07-20, uCosminexus Developer Professional 07-20, uCosminexus Client 07-20, uCosminexus Service Platform 07-20, uCosminexus Service Architect 07-20

追加・変更内容

MyEclipse を使用したアプリケーション開発に対応した。

第 3 版では第 2 版の内容の一部を、マニュアル「Cosminexus 機能解説」に移動し、目次の構成を変更しました。第 2 版との対応は次のようになっています。

第 2 版	第 3 版
1. アプリケーション開発の概要	1. J2EE アプリケーション開発の概要
2. アプリケーションを開発する前に 2.1 Developer を使用したアプリケーション開発の流れ	1. J2EE アプリケーション開発の概要
2.2 展開ディレクトリ形式のアプリケーションの利用方法 (サーバ管理コマンド使用時)	10.3 Eclipse を使用しない J2EE アプリケーションの開発

第 2 版	第 3 版
3. 開発環境のインストールと環境設定 3.1 開発環境のインストールと環境設定の流れ 3.2 Developer のインストールとアンインストール 3.3 Eclipse のインストール 3.4 プラグインの組み込み	2. インストール
3.5 Eclipse の設定 3.6 プラグインの設定	3. 開発環境の構築
3.7 HiRDB SQL Executer のインストール	4.2 HiRDB SQL Executer のインストール
4. プロジェクトの作成	5. プロジェクトの作成
5. ユーザインタフェースの開発	6. ユーザインタフェースの作成
6. 業務処理プログラムの作成	7. 業務処理プログラムの作成
7. プロジェクトのビルド, アーカイブ 7.1 EAR プロジェクトのビルド 7.2 EJB-JAR プロジェクトのビルド 7.3 WAR プロジェクトのビルド 7.4 エクスポートしたビルドファイルの使い方	9.2 EAR ファイルの作成
7.5 JSP の事前コンパイル	11.1 JSP ファイルの事前コンパイル
7.6 JSP ソースのチェック (cjjsp2java コマンド)	11.2 JSP ソースのチェック
7.7 アーカイブおよび DD 作成時の注意事項	5.6 DD の編集
8. 組み込みデータベースの構築	4. テスト用実行環境で使用するデータベースの構築
9. アプリケーションの実行	8. J2EE アプリケーションのテスト
10. IDE を使用しないアプリケーションの開発	10.3 Eclipse を使用しない J2EE アプリケーションの開発
11. EJB クライアントアプリケーションの開発	マニュアル「Cosminexus 機能解説」へ移動
12. リソース接続とトランザクション管理機能の実装	マニュアル「Cosminexus 機能解説」へ移動
13. コンテナ拡張ライブラリの利用と実装	マニュアル「Cosminexus 機能解説」へ移動
14. セッションフェールオーバー機能の実装	マニュアル「Cosminexus 機能解説」へ移動
15. HTTP レスポンス圧縮機能の実装	マニュアル「Cosminexus 機能解説」へ移動
16. セキュリティフィルタの実装	削除
17. 統合ユーザ管理フレームワークを使用したユーザ認証の実装	マニュアル「Cosminexus 機能解説」へ移動
18. ユーザログ機能の実装	マニュアル「Cosminexus 機能解説」へ移動
19. 運用機能の実装	マニュアル「Cosminexus 機能解説」へ移動
20. Timer Service の利用	マニュアル「Cosminexus 機能解説」へ移動
21. JavaBeans リソースの実装	マニュアル「Cosminexus 機能解説」へ移動

第 2 版	第 3 版
22. Cosminexus DABroker Library を使用したデータベース接続の実装	マニュアル「Cosminexus 機能解説」へ移動
付録 A サンプルプログラム	付録 A MyEclipse の動作確認用サンプルプログラム
付録 A.1 Studio Plug-in の動作確認用サンプルプログラム	削除
付録 A.2 サーバ管理コマンドによるサンプル実行環境の構築，運用手順 付録 A.3 Enterprise Bean (EJB2.0) のサンプルプログラム 付録 A.4 Enterprise Bean (アノテーション) のサンプルプログラム 付録 A.5 Enterprise Bean (Timer Service) のサンプルプログラム 付録 A.6 JavaBeans リソースのサンプルプログラム 付録 A.7 ユーザログ機能のサンプルプログラム 付録 A.8 Web アプリケーションのサンプルプログラム	マニュアル「Cosminexus システム構築ガイド」へ移動
付録 B 文字コードの扱い	マニュアル「Cosminexus 機能解説」へ移動
付録 C クラスローダの構成	マニュアル「Cosminexus 機能解説」へ移動
付録 D Developer Standard 使用時の注意事項	付録 B Developer Standard 使用時の注意事項
付録 E TCC 用 Server Plug-in の設定と操作	削除
付録 F JavaMail の使用例	付録 F JavaMail の使用例
付録 G Cosminexus Version 6 から Cosminexus Version 7 への開発環境の移行	付録 E Cosminexus Version 6 から Cosminexus Version 7 への移行
付録 H データベース構築・削除時のエラーメッセージ一覧	付録 H データベース構築・削除時のエラーメッセージ一覧
付録 I 用語解説	付録 I 用語解説

変更内容 (3020-3-M41-20) uCosminexus Application Server Standard 07-10 , uCosminexus Application Server Enterprise 07-10 , uCosminexus Developer Standard 07-10 , uCosminexus Developer Professional 07-10 , uCosminexus Client 07-10 , uCosminexus Web Redirector 07-10 , uCosminexus Service Platform 07-10 , uCosminexus Service Architect 07-10 , uCosminexus Operator 07-10

追加・変更内容
適用 OS に，HP-UX (PA-RISC) を追加した。
適用 OS に，Linux (IPF) を追加した。
HiRDB Version 8 に対応した。
プラグインで RMI - IIOP 通信によるアプリケーション開発ができるようになった。

追加・変更内容

Management Server リモート管理機能からのログアウトを追加した。

EAR プロジェクトの表示名の変更がアプリケーション名に反映される機能を追加した。

JSP 事前コンパイル時にデフォルトエンコーディングを設定できるようになった。

JavaMail 1.3 の機能を追加した。

第 2 版では第 1 版の内容の一部を，マニュアル「Cosminexus 機能解説」に移動し，目次の構成を変更しました。初版との対応は次のようになっています。

第 1 版	第 2 版
22. Enterprise Bean 実装時の注意事項	マニュアル「Cosminexus 機能解説」へ移動
23. サブレット，JSP 実装時の注意事項	マニュアル「Cosminexus 機能解説」へ移動

はじめに

このマニュアルは，Cosminexus（コズミネクス）上で動作するアプリケーションの開発の流れと，アプリケーションの開発方法について説明します。

Cosminexus では，次に示すプログラムプロダクトを使用してアプリケーションを開発，実行します。

- P-1J43-7D71 uCosminexus Application Server Standard
- P-1J43-7K71 uCosminexus Application Server Enterprise
- P-1J43-7M71 uCosminexus Web Redirector
- P-1M43-7D71 uCosminexus Application Server Standard
- P-1M43-7K71 uCosminexus Application Server Enterprise
- P-1M43-7M71 uCosminexus Web Redirector
- P-1M43-7S71 uCosminexus Service Platform
- P-2443-7D74 uCosminexus Application Server Standard
- P-2443-7E74 uCosminexus Developer Standard
- P-2443-7F74 uCosminexus Developer Professional
- P-2443-7H74 uCosminexus Client
- P-2443-7K74 uCosminexus Application Server Enterprise
- P-2443-7M74 uCosminexus Web Redirector
- P-2443-7S74 uCosminexus Service Platform
- P-2443-7T74 uCosminexus Service Architect
- P-2443-7U74 uCosminexus Operator
- P-9D43-7D71 uCosminexus Application Server Standard
- P-9D43-7K71 uCosminexus Application Server Enterprise
- P-9D43-7M71 uCosminexus Web Redirector
- P-9D43-7S71 uCosminexus Service Platform
- P-9S43-7D71 uCosminexus Application Server Standard
- P-9S43-7K71 uCosminexus Application Server Enterprise
- P-9S43-7M71 uCosminexus Web Redirector
- P-9S43-7S71 uCosminexus Service Platform
- P-9V43-7D71 uCosminexus Application Server Standard
- P-9V43-7K71 uCosminexus Application Server Enterprise
- P-9V43-7M71 uCosminexus Web Redirector

このマニュアルでは，これらのプログラムプロダクトの構成ソフトウェアのうち，次に示す構成ソフトウェアについて説明しています。

- Cosminexus Application Development Plug-in
- Cosminexus Component Container
- Cosminexus Component Container - Client
- Cosminexus Operator Plug-in

はじめに

- Cosminexus Component Container - Redirector
- Cosminexus Component Transaction Monitor
- Cosminexus DABroker Library
- Cosminexus Developer's Kit for Java
- Cosminexus Performance Tracer
- Cosminexus TPBroker
- HiRDB Embedded Server Version 8
- MyEclipse for Cosminexus

なお、オペレーティングシステム（OS）の種類によって、機能が異なる場合があります。OSごとの違いがある場合の表記方法については、「適用 OS の違いによる機能相違点の表記」を参照してください。

対象読者

このマニュアルは、Cosminexus で動作するアプリケーションを開発する方を対象にしています。

なお、次の内容を理解されていることを前提としています。

- OS（Windows または UNIX）のシステム構築および運用に関する知識
- Windows または UNIX の基本操作に関する知識
- Java によるプログラム開発に関する基本的な知識
- Eclipse に関する基本的な知識

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 J2EE アプリケーション開発の概要

Developer を使用した J2EE アプリケーション開発の概要、Cosminexus システムで動作する J2EE アプリケーションの形式などについて説明しています。また、J2EE アプリケーション開発の流れについて説明しています。

第 2 章 インストールとセットアップ

Cosminexus の開発環境で使用する製品のインストール方法およびインスタントセットアップ機能を使用したセットアップ方法について説明しています。

第 3 章 テスト用実行環境で使用するデータベースのテーブルの作成

J2EE アプリケーションのテスト用実行環境で使用するデータベースのテーブルを作成する手順について説明しています。

第 4 章 プロジェクトの作成

MyEclipse を使用して J2EE アプリケーションのプロジェクトを作成する方法について説明しています。

第 5 章 ユーザインタフェースの作成

MyEclipse を使用してユーザインタフェースを作成する方法について説明しています。

第 6 章 業務処理プログラムの作成

Eclipse を使用して Enterprise Bean に業務処理を実装する方法について説明しています。

第 7 章 J2EE アプリケーションのテスト

MyEclipse を使用して、作成した J2EE アプリケーションをテストする方法について説明しています。

第 8 章 実行環境への J2EE アプリケーションの配布

テストした J2EE アプリケーションを、実行環境に配布する方法について説明しています。

第 9 章 そのほかの開発

フレームワークや UML を利用した J2EE アプリケーションの開発について説明しています。また、Eclipse を使用しない J2EE アプリケーションの開発についても説明しています。

第 10 章 JSP 事前コンパイルとチェック

JSP 事前コンパイルの実施方法と JSP ソースのチェックについて説明しています。

付録 A MyEclipse の動作確認用サンプルプログラム

Cosminexus が提供する MyEclipse を使用したサンプルプログラムの概要と実行方法について説明しています。

付録 B インスタントセットアップ機能を使用しない環境の構築手順

インスタントセットアップ機能を使用しないで、開発環境およびテスト用実行環境を構築する手順について説明しています。

付録 C Developer Standard 使用時の注意事項

Developer Standard を使用する上での制限事項および注意事項について説明しています。

付録 D Matisse4MyEclipse フィーチャのインストール

Matisse4MyEclipse フィーチャのインストール手順とインストールの確認方法について説明しています。

付録 E テスト用実行環境のシステムのチューニング

J2EE アプリケーションのテスト用実行環境用として構築したシステムのチューニングについて説明しています。

付録 F Cosminexus Version 6 から Cosminexus Version 7 への移行

Cosminexus Version 6 で構築した開発環境を Cosminexus Version 7 に移行する手順と、Cosminexus Version 7 に移行したあとの運用手順について説明しています。

付録 G JavaMail の使用例

JavaMail の使用例として、メールコンフィグレーションの取得からメッセージの送信までの手順について説明しています。

付録 H web.xml 編集時の注意事項（互換用の動作モードの場合）

互換用の動作モードのシステムで J2EE アプリケーションを使用する場合の、web.xml 編集時の注意事項について説明しています。

はじめに

付録 I インスタントセットアップ機能実行時のエラーメッセージ一覧

インスタントセットアップ機能を実行したときにコンソールに出力されるメッセージについて説明しています。

付録 J データベース構築・削除時のエラーメッセージ一覧

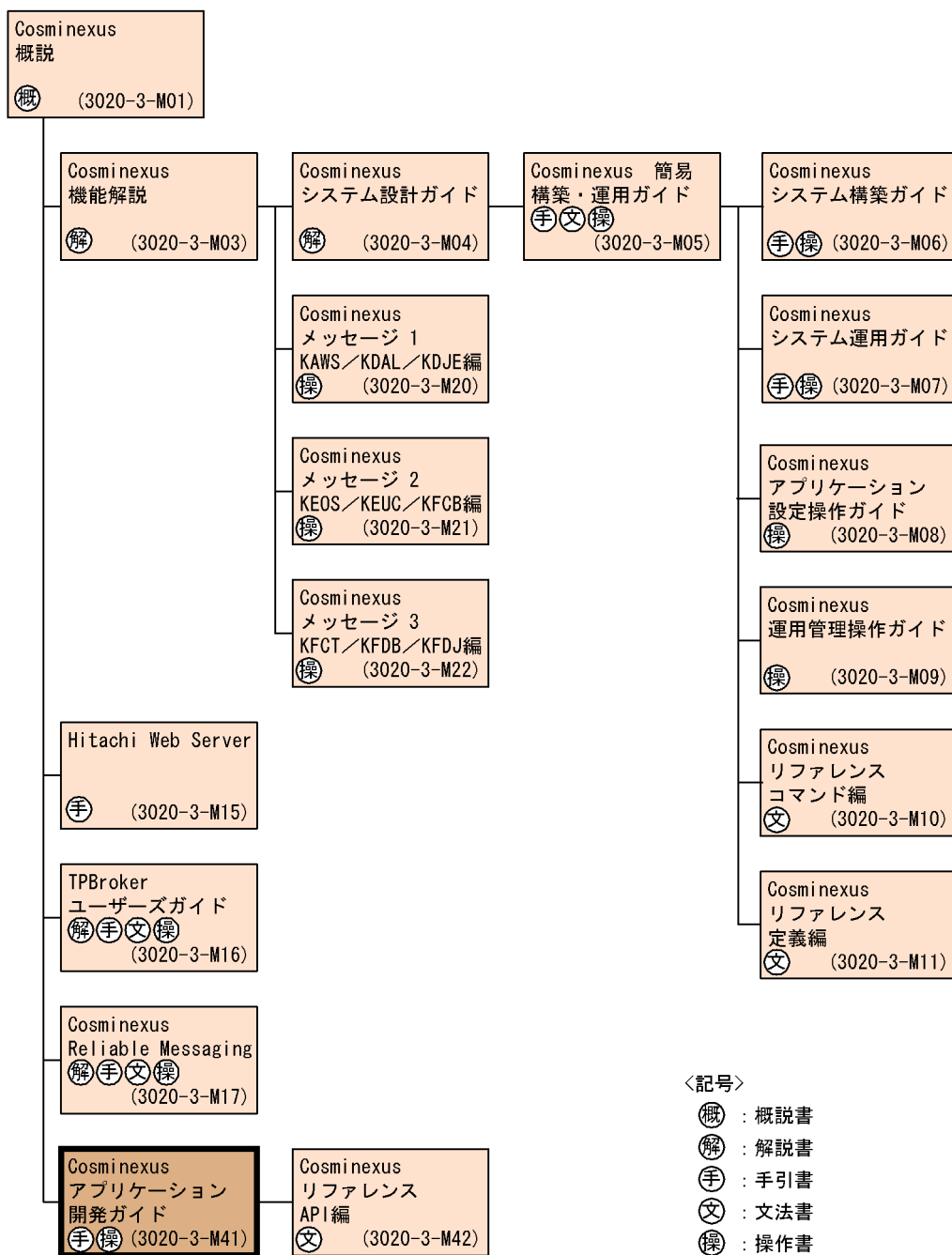
データベースの構築・削除時にエラーファイルに出力されるメッセージについて説明しています。

付録 K 用語解説

このマニュアルで使用している用語の意味について説明しています。

関連マニュアル

Cosminexus のマニュアル体系について、次の図に示します。



マニュアル体系図で示した関連マニュアルについて、それぞれの位置づけを次に示します。

Cosminexus 概説

Cosminexus の製品概要について説明しています。

はじめに

Cosminexus 機能解説

Cosminexus Component Container の機能を中心に、uCosminexus Application Server の概要と提供する機能について説明しています。

Cosminexus システム設計ガイド

システム設計時に、システムの目的に応じたシステム構成や運用方法を検討するための指針について説明しています。また、チューニングの方法についても説明しています。

Cosminexus 簡易構築・運用ガイド

セットアップウィザードおよび Smart Composer 機能を使用して、システムを構築・運用する手順について説明しています。また、セットアップウィザードおよび Smart Composer 機能が提供するコマンドやファイルについても説明しています。

Cosminexus システム構築ガイド

システム構築時に必要な機能の設定方法について説明しています。

Cosminexus システム運用ガイド

Cosminexus を使用したシステムの運用方法のうち、Smart Composer 機能を使用する運用以外の方法について説明しています。

Cosminexus アプリケーション設定操作ガイド

Cosminexus Component Container のサーバ管理コマンド、および Server Plug-in を使用した操作について説明しています。

Cosminexus 運用管理操作ガイド

Cosminexus Component Container の運用管理ポータルの使用方法について説明しています。

Cosminexus リファレンス コマンド編

Cosminexus のシステムを構築・運用するときに使用するコマンドについて説明しています。

Cosminexus リファレンス 定義編

Cosminexus のシステムを構築・運用するとき、またはアプリケーションを開発するときに、使用するファイルの形式について説明しています。

Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編, Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編, Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編

Cosminexus で出力されるメッセージについて説明しています。

Hitachi Web Server

Hitachi Web Server (Web サーバ) の構築、管理方法について説明しています。

TPBroker ユーザーズガイド

Cosminexus TPBroker の概要、機能、運用方法について説明しています。

Cosminexus Reliable Messaging

Cosminexus RM を使用したメッセージの非同期通信によるアプリケーションの連携方法に

ついて説明しています。

Cosminexus リファレンス API 編

アプリケーションの開発で使用する API およびタグについて説明しています。

また、マニュアル体系図に示したマニュアル以外で、このマニュアルと関連するマニュアルを次に示します。必要に応じてお読みください。

- DABroker (3020-6-031)
- スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (UNIX(R) 用)(3000-6-272)
- スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (UNIX(R) 用)(3000-6-275)
- スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (UNIX(R)/Windows(R) 用)(3000-6-276)
- スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (UNIX(R)/Windows(R) 用)(3000-6-277)
- スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (UNIX(R)/Windows(R) 用)(3000-6-278)
- スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (UNIX(R) 用)(3000-6-352)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (UNIX(R) 用)(3000-6-355)
- スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (Windows(R) 用)(3020-6-272)
- スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (Windows(R) 用)(3020-6-275)
- スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (Windows(R) 用)(3020-6-276)
- スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (Windows(R) 用)(3020-6-277)
- スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (Windows(R) 用)(3020-6-278)
- スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (Windows(R) 用)(3020-6-352)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (Windows(R) 用)(3020-6-355)
- スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド (3020-6-356)
- スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス (3020-6-357)
- スケーラブルデータベースサーバ HiRDB Version 8 メッセージ (3020-6-358)

なお、このマニュアルでは、次のマニュアルについてバージョン番号を省略して表記しています。マニュアルの正式名称とこのマニュアルでの表記を次の表に示します。

正式名称	このマニュアルでの表記
スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (Windows(R) 用)	HiRDB システム導入・設計ガイド
スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (UNIX(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (UNIX(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (Windows(R) 用)	HiRDB コマンドリファレンス
スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (UNIX(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 コマンドリファレンス (UNIX(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド	HiRDB UAP 開発ガイド
スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (UNIX(R)/Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス	HiRDB SQL リファレンス
スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (UNIX(R)/Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 8 メッセージ	HiRDB メッセージ
スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (UNIX(R)/Windows(R) 用)	
スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (Windows(R) 用)	

読書手順

このマニュアルは、利用目的に合わせて章を選択して読むことができます。利用目的別にお読みいただくことをお勧めします。

マニュアルを読む目的		記述箇所
アプリケーション開発の概要について知りたい。	Cosminexus が提供する開発環境 (Developer) を使用した開発の特長について知りたい。	1.1

マニュアルを読む目的		記述箇所
	Developer で提供する機能について知りたい。	1.2
	開発するアプリケーションの形式について知りたい。	1.3
	開発環境の前提となる構成ソフトウェアについて知りたい。	1.4
	アプリケーション開発の流れについて知りたい。	1.5
インストール方法について知りたい。	Developer をインストールする方法について知りたい。	2.2
	開発環境インスタントセットアップを使用して、Eclipse および MyEclipse をインストールする方法について知りたい。	2 章
	開発環境インスタントセットアップを使用しないで、Eclipse および MyEclipse をインストールする方法について知りたい。	付録 B.1
開発環境およびテスト用実行環境の構築方法について知りたい。	開発環境インスタントセットアップを使用して、開発環境およびテスト用実行環境を構築する方法について知りたい。	2 章
	開発環境インスタントセットアップを使用しないで、開発環境およびテスト用実行環境を構築する方法について知りたい。	付録 B.2
	構築したシステムのチューニング方法について知りたい。	付録 E
組み込みデータベースを使用する方法が知りたい。	開発環境インスタントセットアップを使用しないで、組み込みデータベースを構築する方法について知りたい。	付録 B.3
	組み込みデータベースを操作する方法について知りたい。	3 章
	組み込みデータベースを構築・削除するときに表示されるメッセージについて知りたい。	付録 J
アプリケーションの開発手順について知りたい。	MyEclipse を使用して、プロジェクトを作成する手順について知りたい。	4 章
	MyEclipse を使用して、ユーザインタフェースを作成する手順について知りたい。	5 章
	Eclipse を使用して、業務処理プログラムを作成する手順について知りたい。	6 章
	MyEclipse を使用してアプリケーションをテストする手順について知りたい。	7 章
	フレームワークを利用した開発について知りたい。	9.1
	UML を利用した開発について知りたい。	9.2
	Eclipse を使用しないでアプリケーションを開発する方法について知りたい。	9.3
	JSP 事前コンパイルの実行方法について知りたい。	10.1

マニュアルを読む目的		記述箇所
	JSP ソースのチェック方法について知りたい。	10.2
アプリケーション開発時の注意事項について知りたい。	Windows Vista の場合の J2EE アプリケーション開発時の注意事項について知りたい。	1.6
	Developer Standard 使用時の制限事項と注意事項について知りたい。	付録 C
	互換用の動作モードの場合の web.xml 編集時の注意事項について知りたい。	付録 H
実行環境への配布の手順について知りたい。	開発環境で作成したアプリケーションを、実行環境に配布する手順について知りたい。	8 章
サンプルプログラムについて知りたい。	サンプルプログラムの実行方法について知りたい。	付録 A
Matisse4MyEclipse について知りたい。	Matisse4MyEclipse フィーチャのインストール手順について知りたい。	付録 D
開発環境やアプリケーションの移行方法が知りたい。	Cosminexus Version 6 の環境で構築した開発環境を Cosminexus Version 7 の環境で使用方法について知りたい。	付録 F
JavaMail の送信方法が知りたい。	JavaMail を使用してメッセージを送信する手順が知りたい。	付録 G
用語について知りたい。	マニュアル本文で使用されている用語について知りたい。	付録 K

ご利用製品ごとの用語の読み替えについて

uCosminexus Service Platform をご利用の場合は、このマニュアルで使用している用語のうち、Application Server および Application Server Enterprise を、uCosminexus Service Platform に読み替えてください。

uCosminexus Service Architect をご利用の場合は、このマニュアルで使用している用語のうち、Developer および Developer Professional を uCosminexus Service Architect に読み替えてください。

このマニュアルでの表記

このマニュアルで使用している表記と、対応する製品名を次に示します。

表記		製品名
Application Server	Application Server Enterprise	uCosminexus Application Server Enterprise
	Application Server Standard	uCosminexus Application Server Standard
Developer	Developer Professional	uCosminexus Developer Professional
	Developer Standard	uCosminexus Developer Standard

表記		製品名	
HiRDB または HiRDB サーバ	HiRDB/Single Server	HiRDB/Single Server Version 7	
		HiRDB/Single Server Version 8	
	HiRDB/Parallel Server	HiRDB/Parallel Server Version 7	
		HiRDB/Parallel Server Version 8	
HiRDB Run Time または HiRDB クライアント		HiRDB/Run Time Version 7	
		HiRDB/Run Time Version 8	
Internet Explorer		Microsoft(R) Internet Explorer(R)	
IPF		Itanium(R) Processor Family	
JSF		JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS	
Microsoft Cluster Service		Microsoft(R) Cluster Service	
Microsoft IIS		Microsoft(R) Internet Information Services 6.0	
Oracle	Oracle9i	Oracle9i	
		Oracle9i R2	
	Oracle10g	Oracle 10g	
		Oracle 10g R2	
SQL Server		Microsoft(R) SQL Server(TM) 2000	
		Microsoft(R) SQL Server(TM) 2005	
Struts		Jakarta Struts 1.1	
UNIX	AIX	AIX 5L V5.2	
		AIX 5L V5.3	
	HP-UX または HP-UX (IPF)	HP-UX 11i V2 (IPF)	
		HP-UX 11i V3 (IPF)	
	Linux	Linux (IPF)	Red Hat Enterprise Linux AS 3 (IPF)
			Red Hat Enterprise Linux AS 4 (IPF)
			Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium)
		Linux (x86 / AMD64 & Intel EM64T)	Red Hat Enterprise Linux AS 3 (x86)
			Red Hat Enterprise Linux AS 4 (x86)
			Red Hat Enterprise Linux 5 Advanced Platform (x86)
			Red Hat Enterprise Linux ES 3 (x86)
			Red Hat Enterprise Linux ES 4 (x86)

表記			製品名
			Red Hat Enterprise Linux 5 (x86)
			Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)
			Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux 5 (AMD/Intel 64)
	Solaris		Solaris 9
			Solaris 10
Web Redirector			uCosminexus Web Redirector
Windows Server 2003			Microsoft(R) Windows Server(R) 2003 , Enterprise Edition Operating System (x86)
			Microsoft(R) Windows Server(R) 2003 , Standard Edition Operating System (x86)
Windows Server 2003 R2			Microsoft(R) Windows Server(R) 2003 R2 , Enterprise Edition Operating System (x86)
			Microsoft(R) Windows Server(R) 2003 R2 , Standard Edition Operating System (x86)
Windows Server 2003 (x64)			Microsoft(R) Windows Server(R) 2003 , Enterprise x64 Edition Operating System
			Microsoft(R) Windows Server(R) 2003 , Standard x64 Edition Operating System
Windows Server 2003 R2 (x64)			Microsoft(R) Windows Server(R) 2003 R2 , Enterprise x64 Edition Operating System
			Microsoft(R) Windows Server(R) 2003 R2 , Standard x64 Edition Operating System
Windows Vista			Microsoft(R) Windows Vista(R) Business
			Microsoft(R) Windows Vista(R) Enterprise
			Microsoft(R) Windows Vista(R) Ultimate
Windows XP			Microsoft(R) Windows(R) XP Professional Operating System

なお , Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) , Windows Server 2003 R2 (x64) , Windows Vista , および Windows XP を総称して Windows

と表記することがあります。

このマニュアルで使用している表記と、対応する Cosminexus の機能名を次に示します。

表記	Cosminexus の機能名
Cosminexus Developer's Kit for Java	Cosminexus Developer's Kit for Java TM
CTM	Cosminexus Component Transaction Monitor
Management Server	Cosminexus Management Server
MyEclipse	MyEclipse for Cosminexus
PRF	Cosminexus Performance Tracer
Server Plug-in	Cosminexus Server Plug-in
Smart Composer	Cosminexus Smart Composer
Studio Plug-in	Cosminexus Studio Plug-in
TPBroker	Cosminexus TPBroker

このマニュアルで使用している表記と、対応する Java 関連用語を次に示します。

表記	Java 関連用語
DI	Dependency Injection
EAR	Enterprise ARchive
EJB または Enterprise JavaBeans	Enterprise JavaBeans TM
EJB QL	EJB TM Query Language
J2EE または Java 2 Platform, Enterprise Edition	Java TM 2 Platform, Enterprise Edition
J2SE	Java TM 2 Platform, Standard Edition
JAAS	Java TM Authentication and Authorization Service
JAR	Java TM Archive
Java	Java TM
Java 2 Runtime Environment, Standard Edition	Java TM 2 Runtime Environment, Standard Edition
Java 2 SDK, Standard Edition	Java TM 2 Software Development Kit, Standard Edition
JavaBeans	JavaBeans TM
JavaMail	JavaMail TM
JAXP	Java TM API for XML Processing
JCA	J2EE TM Connector Architecture
JCE	Java TM Cryptography Extension

表記	Java 関連用語
JDBC	JDBC™
	Java™ Database Connectivity
JDK	Java™ Development Kit
JMS	Java™ Message Service
JNDI	Java Naming and Directory Interface™
JNI	Java™ Native Interface
JSF	JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS
JSP	JavaServer Pages™
JSTL	JavaServer Pages™ Standard Tag Library
JTA	Java™ Transaction API
JTS	Java™ Transaction Service
Servlet またはサーブレット	Java™ Servlet
WAR	Web ARchive

適用 OS の違いによる機能相違点の表記

このマニュアルは、適用 OS が Windows、AIX、HP-UX、および Linux、および Solaris の製品に対応します。OS によって記述を書き分ける場合、次に示す表記を使用して、それぞれの説明に OS 名を明記しています。

表記	意味
Windows の場合	Windows に該当する表記です。
Solaris の場合	Solaris に該当する表記です。
UNIX の場合	UNIX (AIX、HP-UX、Linux、Solaris) に該当する表記です。

このマニュアルで使用している略語

このマニュアルで使用している英略語を次に示します。

英略語	英字での表記
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
ASCII	<u>A</u> merican <u>S</u> tandard <u>C</u> ode for <u>I</u> nformation <u>I</u> nterchange
BLOB	<u>B</u> inary <u>L</u> arge <u>O</u> bject
BMP	<u>B</u> ean- <u>M</u> anaged <u>P</u> ersistence
BMT	<u>B</u> ean- <u>M</u> anaged <u>T</u> ransaction
CMP	<u>C</u> ontainer- <u>M</u> anaged <u>P</u> ersistence

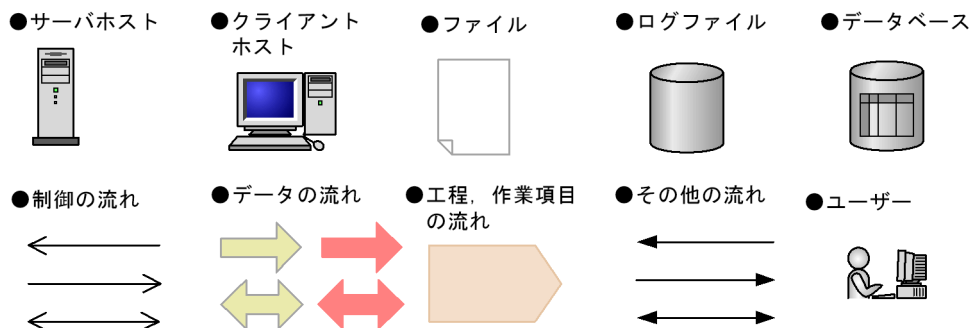
英略語	英字での表記
CMR	<u>C</u> ontainer- <u>M</u> anaged <u>R</u> elationship
CMT	<u>C</u> ontainer- <u>M</u> anaged <u>T</u> ransaction
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CSS	<u>C</u> ascading <u>S</u> tyle <u>S</u> heets
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alue
CUI	<u>C</u> haracter <u>U</u> ser <u>I</u> nterface
CVS	<u>C</u> oncurrent <u>V</u> ersions <u>S</u> ystem
DB	<u>D</u> atabase
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DD	<u>D</u> eployment <u>D</u> escriptor
DII	<u>D</u> ynamic <u>I</u> nvocation <u>I</u> nterface
DNS	<u>D</u> omain <u>N</u> ame <u>S</u> ystem
DTD	<u>D</u> ocument <u>T</u> ype <u>D</u> efinition
EAR	<u>E</u> nterprise <u>A</u> rchive
EIS	<u>E</u> nterprise <u>I</u> nformation <u>S</u> ystem
EJB QL	<u>E</u> JB <u>Q</u> uery <u>L</u> anguage
EPS	<u>E</u> ncapsulated <u>P</u> ost <u>S</u> cript
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
GIF	<u>G</u> raphic <u>I</u> nterchange <u>F</u> ormat
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HA	<u>H</u> igh <u>A</u> vailability
HTML	<u>H</u> yper <u>T</u> ext <u>M</u> arkup <u>L</u> anguage
HTTP	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol
HTTPS	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol <u>S</u> ecurity
IDE	<u>I</u> ntegrated <u>D</u> evelopment <u>E</u> nvironment
IDL	<u>I</u> nterface <u>D</u> efinition <u>L</u> anguage
IIOP	<u>I</u> nternet <u>I</u> nter- <u>O</u> rb <u>P</u> rotocol
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandards
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
LDAP	<u>L</u> ightweight <u>D</u> irectory <u>A</u> ccess <u>P</u> rotocol
LDIF	<u>L</u> LDAP <u>D</u> ata <u>I</u> nterchange <u>F</u> ormat
MDA	<u>M</u> odel <u>D</u> riven <u>A</u> rchitecture
MIB	<u>M</u> anagement <u>I</u> nformation <u>B</u> ase

英略語	英字での表記
MVC	<u>M</u> odel <u>V</u> iew <u>C</u> ontroller
OID	<u>O</u> bject <u>I</u> dentifier
OMG	<u>O</u> bject <u>M</u> anagement <u>G</u> roup
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
OS	<u>O</u> perating <u>S</u> ystem
OTS	<u>O</u> bject <u>T</u> ransaction <u>S</u> ervice
OTM	<u>O</u> bject <u>T</u> ransaction <u>M</u> onitor
PNG	<u>P</u> ortable <u>N</u> etwork <u>G</u> raphics
POA	<u>P</u> ortable <u>O</u> bject <u>A</u> dapter
PS	<u>P</u> ost <u>S</u> cript
RAC	<u>R</u> eal <u>A</u> pplication <u>C</u> lusters
RAR	<u>R</u> oshal <u>A</u> rchive
RD	<u>R</u> elational <u>D</u> atabase
RDB	<u>R</u> elational <u>D</u> atabase
RMI	<u>R</u> emote <u>M</u> ethod <u>I</u> nvocation
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SCCS	<u>S</u> ource <u>C</u> ode <u>C</u> ontrol <u>S</u> ystem
SDK	<u>S</u> oftware <u>D</u> evelopment <u>K</u> it
SFO	<u>S</u> ession <u>F</u> ail <u>O</u> ver
SHA	<u>S</u> ecure <u>H</u> ash <u>A</u> lgorithm
SNMP	<u>S</u> imple <u>N</u> etwork <u>M</u> anagement <u>P</u> rotocol
SPI	<u>S</u> ervice <u>P</u> rovider <u>I</u> nterface
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
SSL	<u>S</u> ecure <u>S</u> ockets <u>L</u> ayer
SVG	<u>S</u> calable <u>V</u> ector <u>G</u> raphics
TCS	<u>T</u> ransaction <u>C</u> ontext <u>S</u> erver
UCS	<u>U</u> niversal multi-octet coded <u>C</u> haracter <u>S</u> et
UML	<u>U</u> nified <u>M</u> odeling <u>L</u> anguage
UNC	<u>U</u> niversal <u>N</u> aming <u>C</u> onvention
URI	<u>U</u> niform <u>R</u> esource <u>I</u> dentifier
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator
UTF	<u>U</u> CS <u>T</u> ransformation <u>F</u> ormat
VM	<u>V</u> irtual <u>M</u> achine

英略語	英字での表記
XML	Extensible Markup Language

このマニュアルの図中で使用している記号

このマニュアルの図中で使用している記号を、次のように定義します。



このマニュアルで使用している記号

！ 注意事項

間違いやすい点、および間違うとトラブルが発生するような項目について説明しています。

ポイント

その説明の要点について説明しています。

参考

補足的な情報を説明しています。

このマニュアルの GUI の説明で使用している記号

このマニュアルでは、次に示す記号を使用して GUI を説明しています。

記号	意味
[]	画面の名称および画面に表示されている項目を表します。
[] - []	- の前に示したメニューから、- の後ろのメニューを選択することを表します。
[] + []	+ の前のキーを押したまま、後ろのキーを押すことを表します。
< >	<> 内の名称がユーザの環境によって異なることを表します。
「 」	入力値、可変値、またはメッセージなどを表します。

このマニュアルのコマンドの説明で使用している記号

このマニュアルでは、次に示す記号を使用してコマンドを説明しています。

記号	意味
	横に並べられた複数の項目に対する項目間の区切りを示し、「または」を意味します。 (例) A B A または B を指定することを示します。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号 で区切られている場合は、そのうちの一つを選択します。 (例) {A B C} A, B または C のどれかを指定することを示します。
[] []	この記号で囲まれている項目は省略してもよいことを示します。複数の項目が横に並べて記述されている場合には、すべてを省略するか、どれか一つを選択します。 (例 1) [A] 「何も指定しない」か「A を指定する」ことを示します。 (例 2) [B C] 「何も指定しない」か「B または C を指定する」ことを示します。

このマニュアルのコーディング例で使用している記号

このマニュアルでは、次に示す記号を使用してコーディング例を説明しています。

記号	意味
: ...	記述が省略されていることを示します。
/* */ //	コメント文であることを示します。

フォルダとパスの表記について

このマニュアルでは、IDE や Java コマンドでの表示に従い、フォルダ / ディレクトリの表現、およびパスの表記 (「¥」または「/」) を変更している場合があります。ご使用の環境に合わせて、フォルダ / ディレクトリの表現、およびパスの表記を置き換えてお読みください。また、UNIX 環境で使用する場合は、本文中の「<Cosminexus のインストールディレクトリ>¥」の表記を次のとおり読み替えてください。

/opt/Cosminexus/

常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語については、常用漢字以外の漢字を使用しています。

宛先 (あてさき) 個所 (かしょ) 括る (くくる) 桁 (けた) 汎用 (はんよう) 必須 (ひつす) 閉塞 (へいそく)

KB（キロバイト）などの単位表記について

1KB（キロバイト）, 1MB（メガバイト）, 1GB（ギガバイト）, 1TB（テラバイト）は、それぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

目次

1	J2EE アプリケーション開発の概要	1
1.1	Developer による J2EE アプリケーション開発の特長	2
1.1.1	MyEclipse との連携によるスムーズな J2EE アプリケーション開発	2
1.1.2	展開ディレクトリ形式の利用による開発効率の向上	3
1.1.3	開発用データベースの標準提供	4
1.1.4	開発環境の簡易構築	5
1.2	Developer で提供する機能	6
1.2.1	J2EE アプリケーション開発で使用する機能	6
1.2.2	テスト用実行環境で使用する機能	9
1.3	Developer で開発する J2EE アプリケーションの形式	10
1.3.1	展開ディレクトリ形式の J2EE アプリケーション	10
1.3.2	アーカイブ形式の J2EE アプリケーション	13
1.4	開発環境のマシン構成	15
1.4.1	1 台のマシンで開発・テストする場合の構成	15
1.4.2	異なるマシンで開発・テストする場合の構成	16
1.5	J2EE アプリケーション開発の流れ	19
1.5.1	J2EE アプリケーションの開発サイクル	19
1.5.2	J2EE アプリケーションの開発手順	20
1.5.3	そのほかの開発の特長	22
1.6	Windows Vista 使用時の注意事項	23
1.6.1	管理者特権で実行する必要がある操作	23
1.6.2	JIS X0213:2004 に含まれる Unicode の補助文字を使用する場合の注意事項	25
2	インストールとセットアップ	27
2.1	インストールとセットアップの流れ	28
2.2	Developer のインストール	30
2.2.1	インストール後のディレクトリ構成	30
2.2.2	環境変数の設定	31
2.2.3	Cosminexus を再インストールしたときの注意事項	32
2.3	インスタントセットアップ機能を使用したセットアップ	34
2.3.1	セットアップの前提条件	34
2.3.2	セットアップする環境の設定内容	35
2.3.3	標準セットアップ	39

2.3.4	カスタムセットアップ	47
2.4	開発環境の設定	55
2.4.1	Eclipse の設定	55
2.4.2	Management Server リモート管理機能へのログインおよびログアウト (MyEclipse)	58
2.4.3	J2EE サーバの選択	60
2.4.4	Server Plug-in の設定	61
2.5	開発環境の設定変更 (任意の作業)	67
2.5.1	MyEclipse のプロキシの設定	67
2.5.2	デプロイ作業ディレクトリの設定変更	69
2.5.3	ソケット操作のブロックのタイムアウト設定変更	70
2.5.4	コンソールの設定変更	72
2.5.5	アプリケーション統合属性ファイルのエディタの設定変更	75
2.6	インスタントセットアップ機能で構築した J2EE サーバの設定変更	77
2.7	インスタントセットアップ機能で構築した環境のアンセットアップ	82
2.8	Developer のアンインストール	87

3

	テスト用実行環境で使用するデータベースのテーブルの作成	89
3.1	組み込みデータベースのテーブルの作成の流れ	90
3.2	HiRDB SQL Executer のインストール	92
3.3	組み込みデータベースの操作	93
3.3.1	組み込みデータベースへの接続	93
3.3.2	テーブルの作成	95
3.3.3	テーブルの参照	96
3.4	RD エリアの追加と削除	97
3.4.1	RD エリアの追加の流れ	97
3.4.2	RD エリアの削除の流れ	98
3.5	組み込みデータベースの動作制御の設定	100

4

	プロジェクトの作成	107
4.1	MyEclipse でのプロジェクトの作成	108
4.1.1	MyEclipse で作成するプロジェクトの構成	108
4.1.2	プロジェクト作成の流れ	109
4.2	EJB プロジェクトの作成	111
4.2.1	EJB プロジェクトの作成手順	111

4.2.2	EJB プロジェクト作成時の注意事項	113
4.2.3	ejb-jar.xml ファイルのひな型の作成	114
4.3	Web プロジェクトの作成	121
4.3.1	Web プロジェクトの作成手順	121
4.3.2	Web プロジェクト作成時の注意事項	124
4.3.3	コンテキストルートの変更手順	124
4.4	エンタープライズアプリケーションプロジェクトの作成	127
4.4.1	エンタープライズアプリケーションプロジェクトの作成手順	127
4.4.2	エンタープライズアプリケーションプロジェクト作成時の注意事項	133
4.4.3	エンタープライズアプリケーションプロジェクトのモジュールの変更手順	134
4.5	リソースアダプタのインポート	136
4.6	DD の編集	142
4.6.1	Cosminexus でサポートする DD について	142
4.6.2	[MyEclipse XML エディター] の操作方法	142
4.6.3	application.xml 編集時の注意事項	146
4.6.4	ejb-jar.xml 編集時の注意事項	147
4.6.5	web.xml 編集時の注意事項	148
4.6.6	Servlet 2.4 仕様で追加, 変更された仕様についての注意事項 (web.xml)	154

5

ユーザインタフェースの作成	157
5.1 ユーザインタフェースの開発の流れ	158
5.2 JSP ファイルの作成	159
5.3 JSP ファイルの編集	161
5.3.1 スニペットを使用した JSP ファイルの編集	164
5.4 サープレットの作成・編集	175

6

業務処理プログラムの作成	179
6.1 業務処理プログラム作成の流れ	180
6.2 パッケージの作成	181
6.3 Enterprise Bean クラスの作成	182
6.4 業務処理プログラムおよびインタフェースの実装	184
6.4.1 業務処理プログラムの実装	184
6.4.2 リモートインタフェースの実装	184
6.4.3 ホームインタフェースの実装	185
6.5 RMI-IIOP インタフェースを利用した J2EE アプリケーションの呼び出し	188

6.5.1 Web アプリケーションから呼び出す場合	188
6.5.2 Enterprise Bean から呼び出す場合	192
6.5.3 EJB クライアントアプリケーションからの呼び出す場合	194
6.6 アノテーションの記述方法	197

7

J2EE アプリケーションのテスト	199
7.1 J2EE アプリケーションのテストの流れ	200
7.2 組み込みデータベースの開始および停止	202
7.3 J2EE サーバの開始および停止	203
7.3.1 J2EE サーバの開始	203
7.3.2 J2EE サーバの停止	204
7.3.3 J2EE サーバの再起動	204
7.4 J2EE アプリケーションのデプロイおよびアンデプロイ	205
7.4.1 J2EE アプリケーションのデプロイ	205
7.4.2 J2EE アプリケーションのアンデプロイ	208
7.5 プロジェクトの入れ替え	209
7.5.1 プロジェクトの入れ替え手順	209
7.5.2 プロジェクトの入れ替え時の注意事項	210
7.5.3 バージョン 07-20 とバージョン 07-50 以降の機能差	210
7.6 J2EE アプリケーションのデバッグ	212
7.7 J2EE アプリケーションのエクスポート	213
7.8 MyEclipse 使用時の注意事項	215
7.8.1 MyEclipse と Server Plug-in を併用する場合	215
7.8.2 MyEclipse および Server Plug-in で表示される J2EE サーバ名	216
7.8.3 デプロイまたはアンデプロイできない場合の対処方法	216
7.9 Server Plug-in から起動した J2EE サーバでのデバッグ	218

8

実行環境への J2EE アプリケーションの配布	223
8.1 J2EE アプリケーション配布の流れ	224
8.2 EAR ファイルの作成	226
8.2.1 ビルドファイルの作成	226
8.2.2 ビルドファイルの編集・実行	227
8.3 Connector 属性ファイルのエクスポート	234
8.4 実行環境への J2EE アプリケーションのインポート	235
8.5 実行環境への Connector 属性ファイルのインポート	236

9	そのほかの開発	237
9.1	フレームワークを利用した開発	238
9.1.1	JSF を利用した開発	238
9.1.2	Struts を利用した開発	238
9.2	UML を利用した開発	240
9.2.1	UML とは	240
9.2.2	MyEclipse での UML を利用した設計	240
9.3	Eclipse を使用しない J2EE アプリケーションの開発	242
9.3.1	Eclipse を使用しない J2EE アプリケーションの開発の概要	242
9.3.2	アプリケーションディレクトリの作成 (展開ディレクトリ)	244
9.3.3	Java プログラムのコンパイル (javac コマンド)	247
9.3.4	アーカイブ形式の J2EE アプリケーションの作成	248
9.3.5	実行環境への配布	251
9.3.6	RMI-IIOP スタブの用意	254

10	JSP 事前コンパイルとチェック	257
10.1	JSP ファイルの事前コンパイル	258
10.1.1	JSP 事前コンパイル機能の前提条件	258
10.1.2	JSP 事前コンパイルの対象ファイルと出力ファイル	259
10.1.3	cjjspc コマンドを使用した JSP の事前コンパイル	260
10.1.4	コンパイル不要な JSP ファイルの指定	263
10.1.5	実行結果リストファイルの出力	265
10.1.6	Java ソースファイルの保存	267
10.1.7	JSP コンパイル時の Java 言語仕様のバージョン指定	267
10.1.8	JSP ワークディレクトリ名の変更	268
10.1.9	JavaVM 起動オプションの変更	269
10.1.10	コマンド実行時のログの出力	269
10.1.11	JSP ファイルの文字エンコーディングの指定	269
10.1.12	MyEclipse で J2EE アプリケーションを開発する場合	270
10.2	JSP ソースのチェック	272
10.2.1	cjjs2java コマンド実行時に出力されるメッセージ	272
10.2.2	cjjs2java コマンドの使用例	273
10.2.3	cjjs2java コマンドで生成したソースファイルのコンパイル	276

付録	277
付録 A MyEclipse の動作確認用サンプルプログラム	278
付録 A.1 サンプルプログラムの概要	278
付録 A.2 組み込みデータベースの開始	282
付録 A.3 サンプルプログラムの実行手順	282
付録 B インスタントセットアップ機能を使用しない環境の構築手順	285
付録 B.1 インストール	286
付録 B.2 開発環境の構築	295
付録 B.3 組み込みデータベースの構築	306
付録 B.4 リソースアダプタのプロパティ設定	317
付録 B.5 アンインストール	320
付録 C Developer Standard 使用時の注意事項	323
付録 C.1 Developer Standard が提供するテスト用実行環境	323
付録 C.2 Developer Professional との機能差	324
付録 C.3 Developer Standard での J2EE アプリケーションのインポート手順	336
付録 D Matisse4MyEclipse フィーチャのインストール	338
付録 D.1 Matisse4MyEclipse フィーチャのインストール手順	338
付録 D.2 Matisse4MyEclipse フィーチャのインストール確認	343
付録 E テスト用実行環境のシステムのチューニング	345
付録 F Cosminexus Version 6 から Cosminexus Version 7 への移行	346
付録 F.1 アプリケーションサーバの移行	346
付録 G JavaMail の使用例	358
付録 G.1 Session オブジェクトの取得	358
付録 G.2 メッセージの作成	360
付録 G.3 メッセージの送信	361
付録 G.4 電子メールアドレス指定時の注意事項	362
付録 H web.xml 編集時の注意事項（互換用の動作モードの場合）	363
付録 I インスタントセットアップ機能実行時のエラーメッセージ一覧	366
付録 J データベース構築・削除時のエラーメッセージ一覧	370
付録 K 用語解説	379

索引	391
-----------	------------

1

J2EE アプリケーション開発の概要

Cosminexus システムで動作する J2EE アプリケーションを開発するには、Developer で開発する J2EE アプリケーションの形式や構成を理解しておく必要があります。

この章では、Developer を使用した J2EE アプリケーション開発の特長や開発の流れを説明します。また、Developer が提供する機能と開発環境の概要と前提プログラムについても説明します。

1.1 Developer による J2EE アプリケーション開発の特長

1.2 Developer で提供する機能

1.3 Developer で開発する J2EE アプリケーションの形式

1.4 開発環境のマシン構成

1.5 J2EE アプリケーション開発の流れ

1.6 Windows Vista 使用時の注意事項

1.1 Developer による J2EE アプリケーション開発の特長

Cosminexus システムで動作する J2EE アプリケーションの開発では、Developer および MyEclipse を使用します。

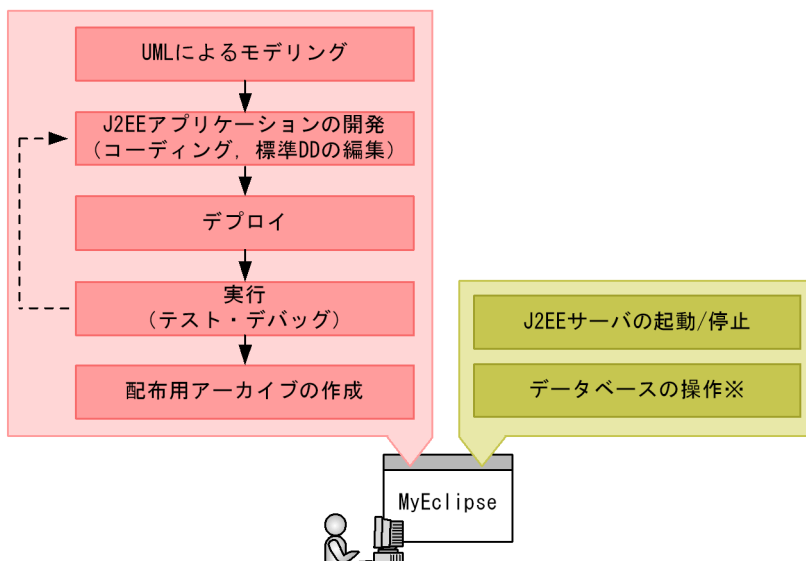
ここでは、Developer および MyEclipse を使用した J2EE アプリケーションの特長を説明します。

1.1.1 MyEclipse との連携によるスムーズな J2EE アプリケーション開発

Developer では、MyEclipse を使用して J2EE アプリケーションを開発できます。MyEclipse とは、J2EE アプリケーション開発に必要な Eclipse のプラグインが一通りそろっている Java 開発環境です。

MyEclipse を使用すると、J2EE アプリケーション開発の一連の作業を開発環境 (Eclipse) からシームレスに実行できます。MyEclipse を使用した場合に開発環境から実行できる作業を次の図に示します。

図 1-1 MyEclipse 使用時に開発環境から実行できる作業



注※ OracleおよびSQL Serverを操作できます。
なお、組み込みデータベース、またはHiRDBを操作する場合は、Developerが提供するHiRDB SQL Executerを使用してください。

図に示すように、MyEclipse では、UML によるモデリングから J2EE アプリケーションの開発、デプロイ、テスト、デバッグ、配布用アーカイブの作成までの J2EE アプリケーション開発の一連の作業を実行できます。また、J2EE サーバやデータベースも開発環境から操作できます。開発時に作成したプログラムは、すべて MyEclipse のプロジェクトとして管理されます。

次に、MyEclipse を利用した J2EE アプリケーション開発の特長を説明します。

(1) UML の利用による開発効率の向上

MyEclipse では UML モデリング機能を提供しているため、開発環境で上流設計ができます。UML を利用するとユースケース図、クラス図などの各種ダイアグラムを作成できます。また、UML モデルからの Java ソースコードの作成、および Java ソースコードからクラス図の作成ができます。これによって、上流設計から開発までの作業をスムーズに進められます。

(2) フレームワークの利用による生産性の向上

フレームワークとは、J2EE アプリケーションで実装する機能やサービスを規定するための枠組みのことです。フレームワークを利用すると、J2EE アプリケーションの開発効率が向上するとともに、プログラムコードの質のばらつきを防止できます。さらに、ソースコードが標準化されているため、メンテナンス効率も向上します。

MyEclipse では、Struts、JSF など、さまざまなフレームワークに対応しています。このため、ユーザインタフェース開発時のほとんどの作業を開発環境で実行できます。

(3) J2EE サーバへのデプロイ・デバッグ

Eclipse のビュー上で、開発した J2EE アプリケーションをデプロイできます。MyEclipse では、作成したプロジェクトを J2EE サーバにデプロイします。アーカイブの作成またはアプリケーションディレクトリへのコピーは、デプロイ時に実施されるため、ユーザ側でのコピーやアーカイブのインポート作業は不要になります。また、Eclipse で開発したソースは、デプロイ時に自動的にビルドされます。

また、デプロイした J2EE アプリケーションは、Eclipse の機能を使用してデバッグできます。

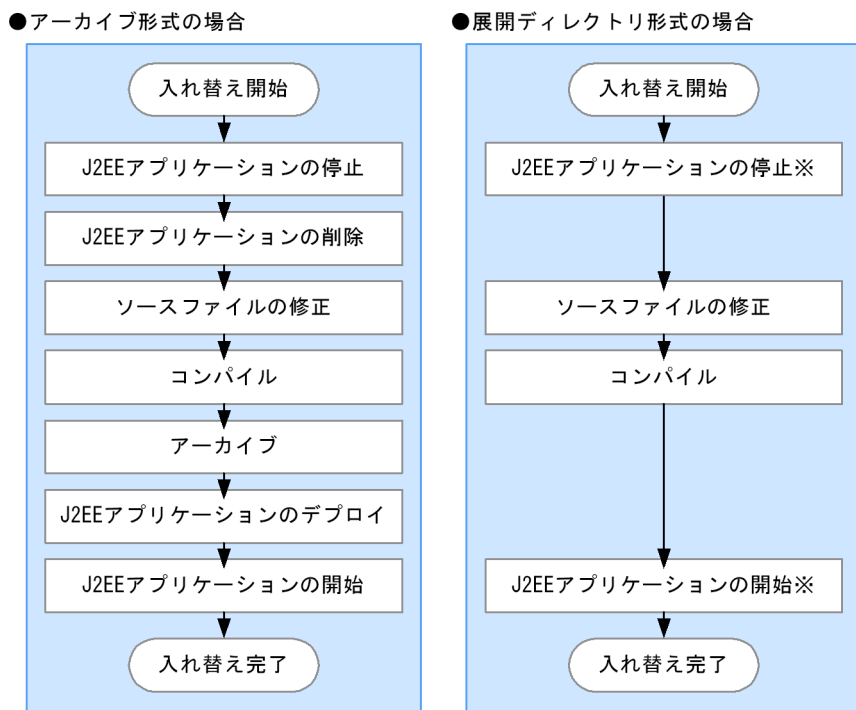
1.1.2 展開ディレクトリ形式の利用による開発効率の向上

展開ディレクトリ形式とは、J2EE アプリケーションを構成するプログラムやファイルを、アーカイブしないでディレクトリの状態のままデプロイする形式です。

展開ディレクトリ形式の J2EE アプリケーションを利用すると、J2EE アプリケーションの入れ替えが容易になります。次の図に、アーカイブ形式と展開ディレクトリ形式の J2EE アプリケーションの入れ替え手順を示します。

1. J2EE アプリケーション開発の概要

図 1-2 J2EE アプリケーション入れ替え手順の比較（アーカイブ形式・展開ディレクトリ形式）



注※ リロード機能利用時は不要となります。

デプロイ済みの J2EE アプリケーションを入れ替える場合、アーカイブ形式の J2EE アプリケーションでは、J2EE アプリケーションを停止、削除してからソースファイルを修正します。また、修正したファイルを含めてアーカイブし直してから、再度デプロイする必要があります。

展開ディレクトリ形式の J2EE アプリケーションを入れ替える場合、J2EE アプリケーションの削除、アーカイブ、およびデプロイが不要となります。さらにリロード機能を使用すれば、J2EE アプリケーションを停止および開始しなくても、ソースファイルを自動で入れ替えることができます。

このように、展開ディレクトリ形式を利用すれば入れ替え手順が容易になるため、修正作業が多くなる J2EE アプリケーション開発の効率が向上します。

展開ディレクトリ形式の J2EE アプリケーションについては、「1.3.1 展開ディレクトリ形式の J2EE アプリケーション」を参照してください。

1.1.3 開発用データベースの標準提供

Developer では、開発環境で使用できるデータベース（組み込みデータベース）を標準

で提供しています。また、組み込みデータベースを操作するために、HiRDB SQL Executer を提供しています。HiRDB SQL Executer を使用すると、組み込みデータベースに対して対話形式で SQL を実行できます。これによって、開発環境でデータベース接続のテストも実施できるため、テスト・デバッグの作業効率が向上します。

1.1.4 開発環境の簡易構築

Developer では、開発環境インスタントセットアップ機能を使用して、開発環境を構築できます。開発環境インスタントセットアップ機能とは、GUI を利用して開発環境およびテスト用実行環境を構築できる機能です。GUI を利用できるため、定義ファイルやコマンドを使用して構築する場合に比べて、短時間で開発環境を構築できるようになります。以降この機能をインスタントセットアップ機能といいます。

インスタントセットアップ機能では、次に示す作業を実施できます。

- Eclipse および MyEclipse のインストール
- テスト用のサーバ環境の構築
- 組み込みデータベースの環境構築

また、インスタントセットアップ機能は、構築した環境の設定変更およびアンセットアップも実行できます。

1.2 Developer で提供する機能

Developer では、J2EE アプリケーションを開発するための機能を提供しています。また、開発した J2EE アプリケーションをテストするために、アプリケーションサーバを構築する機能や、テスト用実行環境で使用するデータベースも提供しています。ここでは、Developer で提供する機能について説明します。

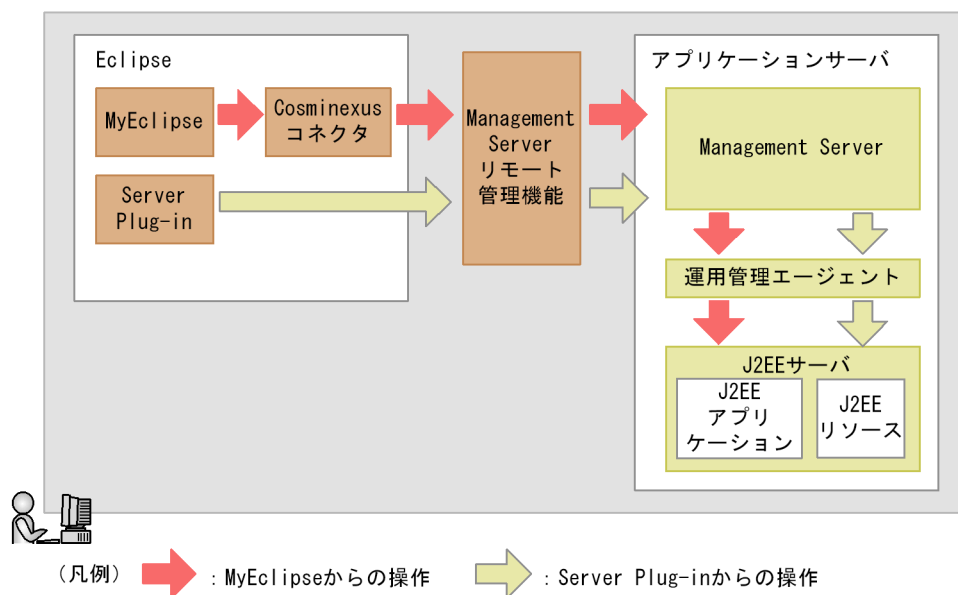
1.2.1 J2EE アプリケーション開発で使用する機能

Developer では、J2EE アプリケーションを開発するために次の機能を提供しています。

- MyEclipse
- Cosminexus コネクタ
- Management Server リモート管理機能
- Server Plug-in

それぞれの機能の関係を次の図に示します。

図 1-3 Developer で提供する機能の関係



MyEclipse は、J2EE アプリケーションの開発に使用します。MyEclipse を使用すると、開発した J2EE アプリケーションを J2EE サーバにデプロイしたり、J2EE サーバを起動したりできます。また、Server Plug-in は、J2EE サーバにデプロイされた J2EE リソースのプロパティ設定に使用します。

MyEclipse から J2EE サーバ上の J2EE アプリケーションを操作するには、Cosminexus

コネクタおよび Management Server リモート管理機能を使用します。また、Server Plug-in から J2EE サーバ上の J2EE リソースを操作するには、Management Server リモート管理機能を使用します。

Cosminexus コネクタおよび Management Server リモート管理機能は、MyEclipse や Server Plug-in からの要求を受け取り、アプリケーションサーバ内の Management Server に処理を渡します。Management Server は、運用管理エージェントを通して、J2EE アプリケーションをデプロイしたり、J2EE サーバを起動したりします。

次に、Developer の機能について紹介します。

(1) MyEclipse

MyEclipse は、J2EE アプリケーションの開発で使います。また、開発した J2EE アプリケーションのテストおよびデバッグにも使います。MyEclipse が提供する機能の一覧を次の表に示します。

表 1-1 MyEclipse が提供する機能の一覧

MyEclipse の機能	説明
JavaScript デバッガ	JavaScript をデバッグするときに使用する機能です。
Matisse4MyEclipse	Java Swing クライアントアプリケーションを開発するときに使用する機能です。
MyEclipse UML	UML で上流設計をするときに使用する機能です。
拡張データベースサポート • Oracle コネクタ • SQL Server	データベースエクスプローラを Oracle または SQL Server に対応させる拡張機能です。
Java イメージエディタ	イメージを編集するときに使用する機能です。
Visual Web デザイナ	HTML や JSP を開発するときに使用する機能です。
XML エディタ	XML を編集するときに使用する機能です。
OR ツール Hibernate サポート	Hibernate を使用する J2EE アプリケーションの開発をサポートする機能です。
MyEclipse Spring ツール	Spring を使用する J2EE アプリケーションの開発をサポートする機能です。
データベースエクスプローラ	JDBC 準拠の任意のデータベースに接続して操作するときに使用する機能です。
Web サービス	Web サービスの開発をサポートする機能です。
Struts デザイナ	Struts を使用する J2EE アプリケーションの開発で、アプリケーションフローをビジュアルに表示・編集する機能です。
JSF デザイナ	JSF を使用する J2EE アプリケーションの開発で、アプリケーションフローをビジュアルに表示・編集する機能です。

1. J2EE アプリケーション開発の概要

MyEclipse の機能	説明
Cosminexus コネクタ	J2EE サーバや J2EE アプリケーションを操作するときに使用する機能です。 なお、Cosminexus コネクタは、Developer が提供する機能です。
JSP 開発	JSP の開発をサポートする機能です。
Sync-on-Demand 開発	J2EE アプリケーションの開発で、Java ソースや JSP の修正を J2EE サーバにデプロイ済みの J2EE アプリケーションに自動的に反映する機能です。 この機能は、Cosminexus で提供するリロード機能と併用して使用します。
CSS エディタ	CSS を編集するときに使用する機能です。
JSTL サポート	JSTL を使用した JSP の開発をサポートする機能です。
ウィザード	プロジェクトの作成、JSP の作成などの操作ができる機能です。
テンプレート	JSP や HTML などを新規に作成するときに使用できる機能です。テンプレートはカスタマイズできます。
ワークベンチ構成	MyEclipse 用に Eclipse ワークベンチウィンドウを再構成する機能です。

注

この表では、Developer Professional または Service Architect をインストールした場合に使用できる MyEclipse の機能を示しています。Developer Standard をインストールした場合に使用できる MyEclipse の機能および注意事項については、「付録 C Developer Standard 使用時の注意事項」を参照してください。

なお、Developer では、MyEclipse の機能のうち、Cosminexus がサポートする範囲の機能を使用できます。

(2) Cosminexus コネクタ

Cosminexus コネクタは、MyEclipse から J2EE サーバを起動したり、J2EE サーバに J2EE アプリケーションをデプロイしたりするときに使用します。Cosminexus コネクタは Eclipse のプラグインです。

(3) Management Server リモート管理機能

Cosminexus コネクタおよび Server Plug-in から J2EE サーバにアクセスできるようにするためのライブラリプラグインです。

MyEclipse や Server Plug-in からの操作は、Management Server リモート管理機能に要求として渡されます。Management Server リモート管理機能が受け取った要求は、さらに Management Server に渡され、J2EE サーバに対して要求された操作が実行されます。MyEclipse や Server Plug-in から J2EE サーバを操作するためには、Management Server リモート管理機能の設定が必要になります。

(4) Server Plug-in

Server Plug-in は Eclipse のプラグインです。Server Plug-in では、次の作業を実行できます。

- J2EE アプリケーションや J2EE リソースのプロパティの設定
- アプリケーションサーバの起動
- 作成した J2EE アプリケーションを実行環境へ配布
- 実行環境での J2EE アプリケーションの実行時属性の設定

参考

MyEclipse を使用して J2EE アプリケーションを開発している場合、J2EE アプリケーションのデプロイ、J2EE サーバの開始などの操作は MyEclipse 上から実施します。

1.2.2 テスト用実行環境で使用する機能

Developer では、テスト用実行環境を構築するために、インスタントセットアップ機能を提供しています。また、開発用データベースとして、組み込みデータベースを提供しています。ここでは、テスト用実行環境で使用する機能について紹介します。

(1) インスタントセットアップ機能

Developer では、J2EE アプリケーション開発のための環境を構築する機能として、インスタントセットアップ機能を提供しています。インスタントセットアップ機能では、GUI を使用して、開発環境およびテスト用実行環境の構築を実行できます。

なお、Developer では、Smart Composer 機能を使用して、テスト用実行環境を構築することもできます。Smart Composer 機能については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

(2) 組み込みデータベース

Developer では、開発環境で利用できるデータベース (HiRDB) を標準提供しています。このデータベースを、組み込みデータベースと呼びます。組み込みデータベースは、MyEclipse で開発した J2EE アプリケーションをテストするときに使用します。

ただし、組み込みデータベースは、実行環境で HiRDB を使用する J2EE アプリケーションをテストするときだけ使用できます。実行環境で Oracle などのデータベースを使用する場合は、テスト用実行環境で組み込みデータベースを使用できません。

また、組み込みデータベースへの接続で使用する JDBC ドライバには、Cosminexus で提供する JDBC ドライバを使用します。なお、Connector 1.5 には対応していません。

1.3 Developer で開発する J2EE アプリケーションの形式

Cosminexus のアプリケーションサーバで実行できる J2EE アプリケーションの形式には、展開ディレクトリ形式とアーカイブ形式があります。Developer では、両方の形式の J2EE アプリケーションを開発できます。なお、Developer で J2EE アプリケーションを開発する場合、J2EE アプリケーションの形式を意識する必要はありません。J2EE アプリケーションの形式は、J2EE アプリケーションを J2EE サーバにデプロイするときに選択します。

ここでは、展開ディレクトリ形式およびアーカイブ形式それぞれの J2EE アプリケーションの概要について説明します。

参考

展開ディレクトリ形式を利用すると、修正したソースファイルを自動検知して更新するリロード機能を使用できます。リロード機能を使用すると、修正したソースファイルを、少ない手順で動的に入れ替えられるようになるため、作業効率が向上します。このマニュアルで説明する J2EE アプリケーションの開発手順では、展開ディレクトリ形式で開発することを前提としています。

1.3.1 展開ディレクトリ形式の J2EE アプリケーション

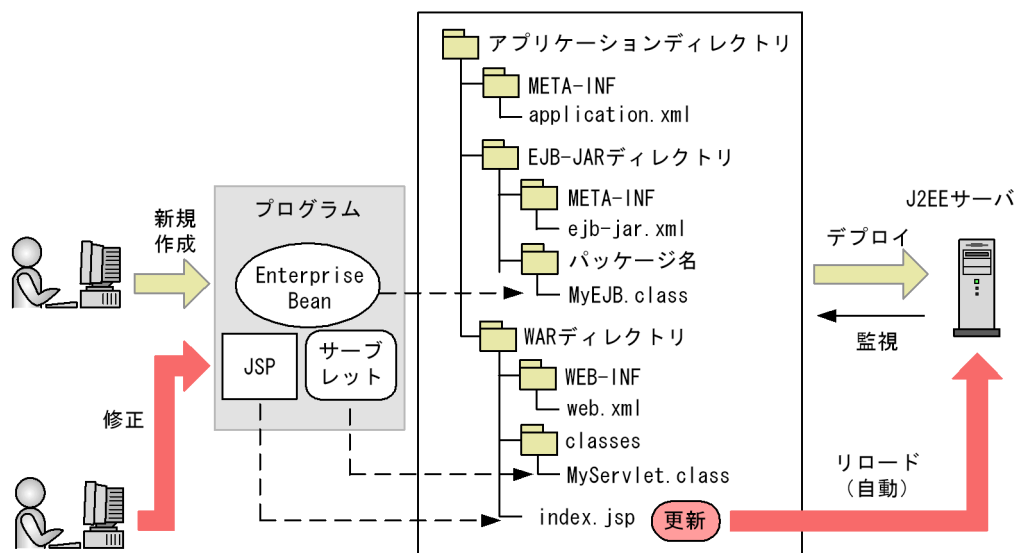
ここでは、展開ディレクトリ形式の J2EE アプリケーションの概要および構成を説明します。

(1) 展開ディレクトリ形式の J2EE アプリケーションの概要

展開ディレクトリ形式の J2EE アプリケーションとは、EJB やサーブレットなどの J2EE アプリケーションの実体を、J2EE サーバの外部に持つ J2EE アプリケーションです。

次の図に展開ディレクトリ形式の J2EE アプリケーションの概要を示します。

図 1-4 展開ディレクトリ形式の J2EE アプリケーションの概要



展開ディレクトリ形式の J2EE アプリケーションを実行するためには、アプリケーションディレクトリと呼ばれる、ある一定のルールに従ったディレクトリとファイルの構造に、作成したプログラムを含め、そのままデプロイします。

展開ディレクトリ形式の場合、デプロイした J2EE アプリケーションを J2EE サーバに監視させて、ファイルを更新したときなどに自動でリロードさせることができます（リロード機能）。このとき、J2EE アプリケーションの停止や削除、インポートなどの作業は不要です。

また、展開ディレクトリ形式の J2EE アプリケーションは、複数のクラスタで共有できます。J2EE アプリケーションの実体を J2EE サーバの外部に持ち、各クラスタから参照する形を取るため、プログラムを修正したときも、一回の更新作業ですべてのクラスタに反映できます。

なお、J2EE サーバにデプロイ済みの展開ディレクトリ形式の J2EE アプリケーションに対して、次に示す変更はできません。

- EJB-JAR および WAR の追加
- EJB-JAR および WAR の削除
- ライブラリ JAR の追加
- ライブラリ JAR の削除

注

J2EE アプリケーションの DD (application.xml) の <module> タグ以下に定義されているファイル以外で、拡張子が小文字の JAR ファイル (.jar) です。J2EE アプリケーション内の J2EE コンポーネントから共通に使用できます。

1. J2EE アプリケーション開発の概要

これらの変更をする場合、一度 J2EE サーバから J2EE アプリケーションを削除してから、再度デプロイする必要があります。

(2) 展開ディレクトリ形式の J2EE アプリケーションの構成

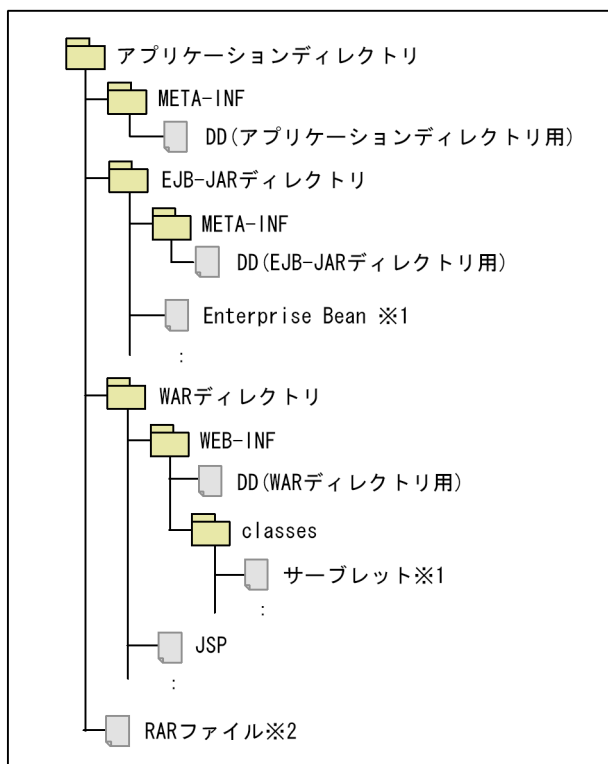
展開ディレクトリ形式では、アプリケーションディレクトリを作成し、アプリケーションディレクトリ以下に EJB-JAR ディレクトリと WAR ディレクトリを含めます。

ここでは、展開ディレクトリ形式でデプロイした場合の J2EE アプリケーションの構成を説明します。

なお、MyEclipse を使用して J2EE アプリケーションを開発した場合は、MyEclipse が自動的に展開ディレクトリ形式で作成するので、構成を意識する必要はありません。

アプリケーションディレクトリの構成例を次の図に示します。

図 1-5 アプリケーションディレクトリの構成例



注※1 コンパイル済みのクラスファイルを格納します。

注※2 リソースアダプタをJ2EEアプリケーションに含めて使用する場合だけ格納します。

アプリケーションディレクトリに含まれる EJB-JAR ディレクトリ、および WAR ディレクトリの構成を示します。

EJB-JAR ディレクトリ

EJB-JAR ディレクトリには、次のファイルなどが含まれます。

- Enterprise Bean
- DD (EJB-JAR ディレクトリ用)

WAR ディレクトリ

WAR ディレクトリには、次のファイルなどが含まれます。

- JSP
- サブレット
- DD (WAR ディレクトリ用)

1.3.2 アーカイブ形式の J2EE アプリケーション

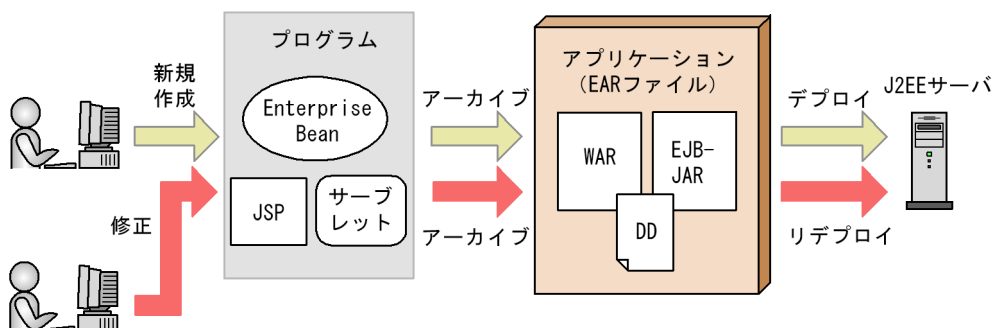
ここでは、アーカイブ形式の J2EE アプリケーションの概要および構成を説明します。

(1) アーカイブ形式の J2EE アプリケーションの概要

アーカイブ形式の J2EE アプリケーションとは、EJB やサブレットなどの J2EE アプリケーションの実体を J2EE サーバの作業ディレクトリに持つ J2EE アプリケーションです。

次の図にアーカイブ形式の J2EE アプリケーションの概要を示します。

図 1-6 アーカイブ形式の J2EE アプリケーションの概要



アーカイブ形式の J2EE アプリケーションを実行するには、作成したプログラムを EJB-JAR ファイルおよび WAR ファイルにアーカイブし、さらに EJB-JAR ファイルおよび WAR ファイルを EAR ファイルにアーカイブしてから、J2EE サーバにデプロイします。

プログラムを修正した場合は、アーカイブとデプロイをやり直して J2EE アプリケーションを入れ替えます。なお、リデプロイ機能を使用すると、J2EE アプリケーションが持つすべての属性情報を引き継いで、J2EE アプリケーションを入れ替えられます。

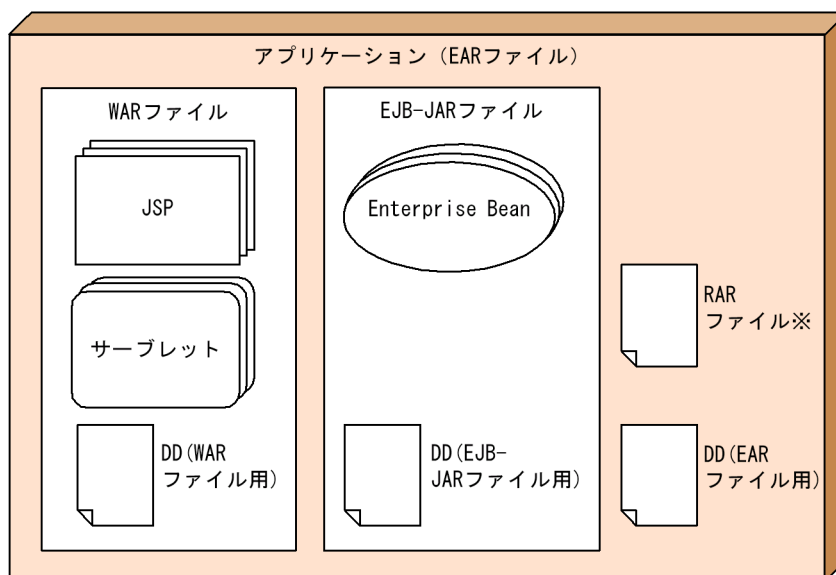
(2) アーカイブ形式の J2EE アプリケーションの構成

アーカイブ形式では、J2EE アプリケーションを EAR ファイルとして作成し、EAR ファイルに WAR ファイルおよび EJB-JAR ファイルを含めます。

ここでは、アーカイブ形式でデプロイした場合の J2EE アプリケーションの構成を説明します。

EAR ファイルの構成例を次の図に示します。

図 1-7 EAR ファイルの構成例



注※ リソースアダプタをJ2EEアプリケーションに含めて使用する場合だけ格納します。

EAR ファイルに含まれる WAR ファイル、および EJB-JAR ファイルの構成を示します。

WAR ファイル

WAR ファイルには、次のファイルなどが含まれます。

- JSP
- サーブレット
- DD (WAR ファイル用)

EJB-JAR ファイル

EJB-JAR ファイルには、次のファイルなどが含まれます。

- Enterprise Bean
- DD (EJB-JAR ファイル用)

なお、Enterprise Bean は JavaBeans に実装することもできます。その場合、EJB-JAR ファイルは作成しません。

1.4 開発環境のマシン構成

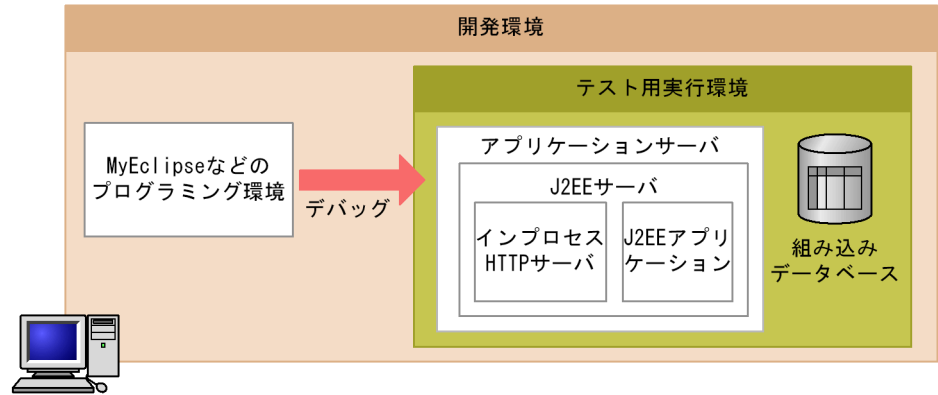
Developer と MyEclipse を使用して J2EE アプリケーションを開発，テストするときには前提となる OS や必要となる構成ソフトウェアについて示します。ここでは，J2EE アプリケーションを，1 台のマシンで開発・テストする場合の構成と，異なるマシンで開発・テストする場合の構成に分けて説明します。

なお，このマニュアルでは 1 台のマシンで開発およびテストをする場合の手順を中心に説明します。

1.4.1 1 台のマシンで開発・テストする場合の構成

J2EE アプリケーションの開発環境とテスト用実行環境を 1 台のマシン上に構築する場合の構成は次のとおりです。

図 1-8 開発環境とテスト用実行環境を 1 台のマシンに構築する場合の構成



開発環境とテスト用実行環境を 1 台のマシン上に構築する場合は，次の環境が必要となります。

表 1-2 開発環境とテスト用実行環境を 1 台のマシンに構築する場合に必要な環境

項番	分類	プログラム名称 / 構成ソフトウェア名称
1	OS	次に示す OS に対応しています。 <ul style="list-style-type: none">• Windows Server 2003• Windows XP• Windows Vista
2	JDK	Cosminexus Developer's Kit for Java
3	J2EE サーバ	Cosminexus Component Container
4	Web サーバ	Hitachi Web Server または Microsoft IIS

1. J2EE アプリケーション開発の概要

項番	分類	プログラム名称 / 構成ソフトウェア名称
5	通信基盤	Cosminexus TPBroker
6	運用管理機能および統合ユーザ管理	Cosminexus Component Container
7	XML プロセッサ	Cosminexus XML Processor
8	フレームワーク	フレームワークを使用した J2EE アプリケーションの場合、実行環境にフレームワークが必要になります。JSF ¹ , Struts ¹ , Cosminexus cFramework などを使用できます。
9	統合開発環境 (IDE)	Eclipse ²
10	プラグイン	プラグインを使用するには次の構成ソフトウェアが必要です。 <ul style="list-style-type: none">• MyEclipse• Cosminexus Application Development Plug-in• Cosminexus Component Container

注 1

MyEclipse を使用すると、アプリケーションフローをビジュアルに表示・編集できます。

注 2

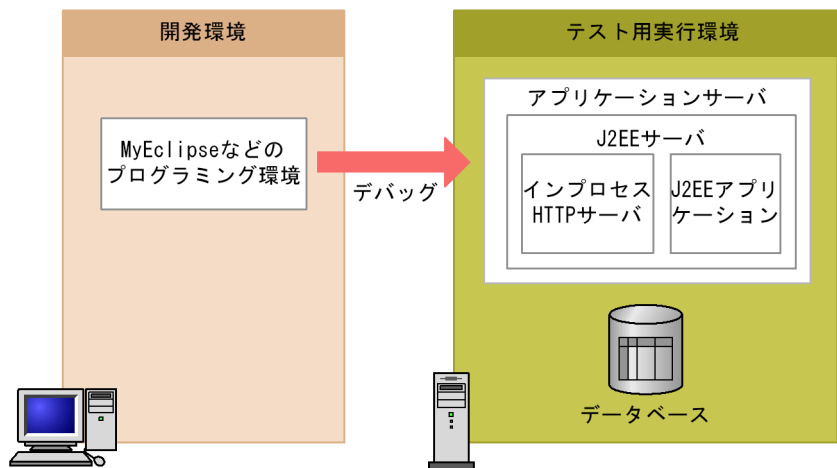
Eclipse は次に示すプログラムが前提になります。

- Eclipse Platform 3.2.2 および Eclipse Platform 3.2.1 NLPack1
- Eclipse Java Development Tools 3.2.2 および Eclipse Java Development Tools 3.2.1 NLPack1

1.4.2 異なるマシンで開発・テストする場合の構成

J2EE アプリケーションを開発する環境とテストする環境を異なるマシンに構築する場合は、次のような構成になります。なお、インスタントセットアップ機能では、上記の環境は構築できません。開発環境とテスト用実行環境を異なるマシンに構築する場合は、「付録 B インスタントセットアップ機能を使用しない環境の構築手順」を参照してください。

図 1-9 開発環境とテスト用実行環境を異なるマシンに構築する場合の構成



開発環境とテスト用実行環境を異なるマシンに構築する場合は、次の環境が必要となります。

表 1-3 開発環境とテスト用実行環境を別のマシンに構築する場合に必要な環境（開発環境）

項番	分類	プログラム名称 / 構成ソフトウェア名称
1	OS	次に示す OS に対応しています。 <ul style="list-style-type: none">• Windows Server 2003• Windows XP• Windows Vista
2	JDK	Cosminexus Developer's Kit for Java
3	J2EE アプリケーション開発環境	Cosminexus Component Container
4	通信基盤	Cosminexus TPBroker
5	統合開発環境（IDE）	Eclipse
6	プラグイン	プラグインを使用するには次の構成ソフトウェアが必要です。 <ul style="list-style-type: none">• MyEclipse• Cosminexus Application Development Plug-in• Cosminexus Component Container
7	XML プロセッサ	Cosminexus XML Processor

注

- Eclipse は次に示すプログラムが前提になります。
- Eclipse Platform 3.2.2 および Eclipse Platform 3.2.1 NLPack1
 - Eclipse Java Development Tools 3.2.2 および Eclipse Java Development Tools 3.2.1 NLPack1

1. J2EE アプリケーション開発の概要

なお、テスト用実行環境には、Application Server をインストールします。Application Server の構成ソフトウェアについては、マニュアル「Cosminexus 概説」を参照してください。

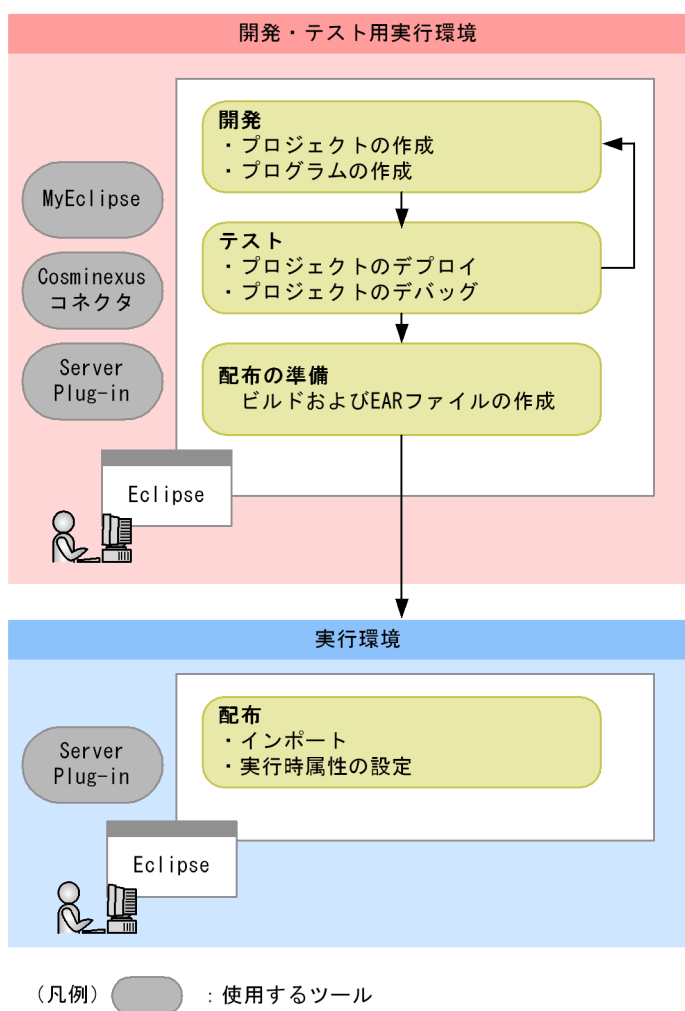
1.5 J2EE アプリケーション開発の流れ

Developer での J2EE アプリケーションの開発をスムーズに進めるために、J2EE アプリケーションの開発サイクルと、開発の手順を説明します。

1.5.1 J2EE アプリケーションの開発サイクル

J2EE アプリケーションの開発サイクルと、開発・テスト用実行環境および実行環境で使用するツールを次の図に示します。

図 1-10 J2EE アプリケーションの開発サイクルと開発・テスト用実行環境および実行環境で使用するツール



開発・テスト用実行環境では、MyEclipse、Cosminexus コネクタ、および Server

1. J2EE アプリケーション開発の概要

Plug-in を使用して、J2EE アプリケーションの開発およびテストを実施します。J2EE アプリケーションのテストで問題があった場合は、J2EE アプリケーションを修正します。

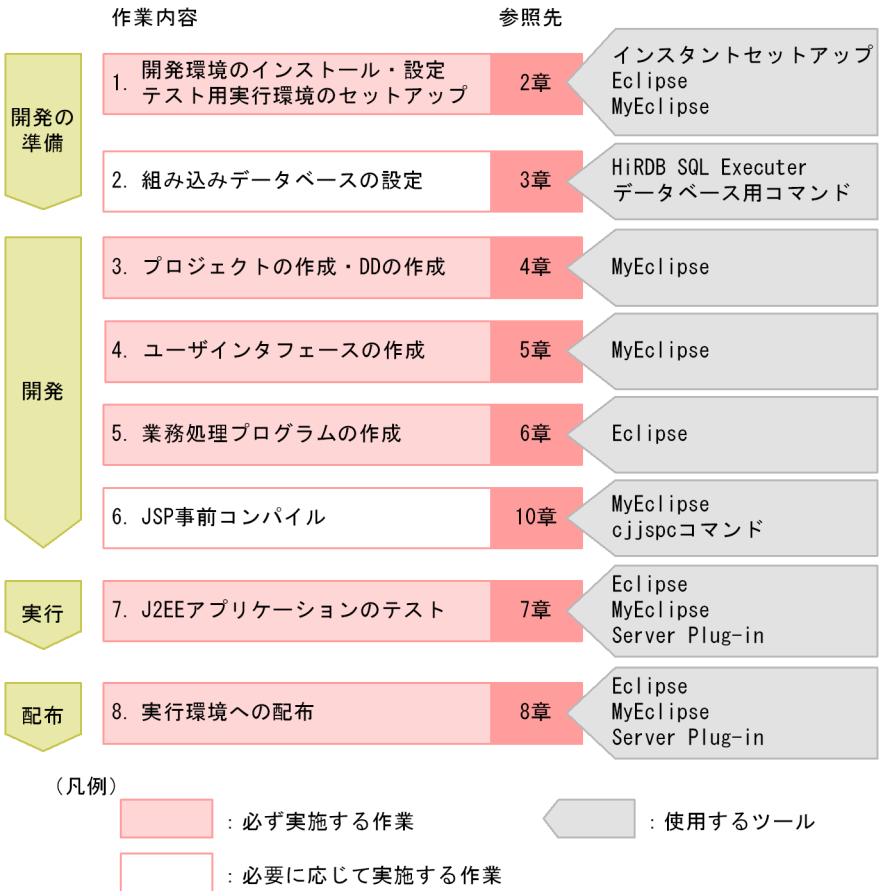
完成した J2EE アプリケーションは、Server Plug-in を使用して実行環境に配布します。実行環境では、Server Plug-in を使用して J2EE アプリケーションや J2EE リソースに実行時属性を設定します。

1.5.2 J2EE アプリケーションの開発手順

ここでは、J2EE アプリケーションの開発手順について説明します。J2EE アプリケーションの開発の流れ、参照先、および各工程で必要となるツールについて、次の図に示します。

なお、このマニュアルで説明する J2EE アプリケーションは、テストで組み込みデータベースを利用するため、組み込みデータベースも構築します。

図 1-11 組み込みデータベースと EJB を利用した J2EE アプリケーション開発の流れ



各工程の概要を説明します。

1. 開発環境のインストール・設定およびテスト用実行環境のセットアップ

J2EE アプリケーション開発に必要な開発環境およびテスト用実行環境を準備します。インスタントセットアップ機能を使用して、次の環境をインストールおよびセットアップします。

- Eclipse および MyEclipse のインストール
- テスト用実行環境のセットアップ
- 組み込みデータベースのセットアップ

インスタントセットアップ機能を実行したあと、開発環境を設定する必要があります。開発環境およびテスト用実行環境のインストール、セットアップ、および設定については、「2. インストールとセットアップ」を参照してください。

2. 組み込みデータベースの設定

インスタントセットアップ機能では、組み込みデータベースの構築とテーブルの作成を実行します。インスタントセットアップ機能で作成されたテーブル以外を使用したい場合は、HiRDB SQL Executer を使用してテーブルを作成する必要があります。組み込みデータベースのテーブルの作成については、「3. テスト用実行環境で使用するデータベースのテーブルの作成」を参照してください。

3. プロジェクトの作成・DD の作成

MyEclipse のプロジェクトである、EJB プロジェクト、Web プロジェクト、エンタープライズアプリケーションプロジェクトを作成します。また、MyEclipse を使用して DD も作成します。プロジェクトの作成、および DD の作成については、「4. プロジェクトの作成」を参照してください。

4. ユーザインタフェースの作成

MyEclipse を使用して、ユーザインタフェースとなる JSP ファイルおよびサーブレットを作成します。ユーザインタフェースの作成については、「5. ユーザインタフェースの作成」を参照してください。

5. 業務処理プログラムの作成

Eclipse の標準機能を使って、業務処理プログラムである Enterprise Bean を実装します。業務処理プログラムの作成については、「6. 業務処理プログラムの作成」を参照してください。

6. JSP 事前コンパイル

JSP ファイルに対する初回リクエスト時のレスポンスタイムを短縮したい場合、cjjspc コマンドを使って、J2EE サーバにデプロイする前に JSP ファイルをコンパイルできます。また、コンパイルした JSP ファイルは、MyEclipse でデバッグできます。JSP 事前コンパイルについては、「10. JSP 事前コンパイルとチェック」を参照してください。

7. J2EE アプリケーションのテスト

Eclipse、MyEclipse、および Server Plug-in の機能を使って、開発環境で J2EE アプ

1. J2EE アプリケーション開発の概要

리케이션のデプロイ, テスト, およびデバッグを実施します。J2EE アプリケーションのテストについては, 「7. J2EE アプリケーションのテスト」を参照してください。

8. 実行環境への配布

開発した J2EE アプリケーションを MyEclipse でビルドして, EAR ファイルを生成します。生成した EAR ファイルは, Server Plug-in またはサーバ管理コマンドを利用して, 実行環境へ配布します。J2EE アプリケーションの実行環境への配布については, 「8. 実行環境への J2EE アプリケーションの配布」を参照してください。

1.5.3 そのほかの開発の特長

MyEclipse では, フレームワークや UML を利用して J2EE アプリケーションを開発することもできます。また, Developer では, Eclipse を利用しないで J2EE アプリケーションを開発することもできます。それぞれの開発の特長を説明します。

(1) フレームワークを利用した開発

MyEclipse では, フレームワークを利用した J2EE アプリケーションを開発できます。フレームワークを利用する場合は, J2EE アプリケーションの設計時に使用するフレームワークを検討します。

なお, MyEclipse では, J2EE アプリケーションを開発するフレームワークとして Struts および JSF を使用できます。MyEclipse の機能を使用すると, ユーザインタフェースおよびアプリケーションフローをビジュアルに編集できます。

フレームワークを利用した開発については, 「9.1 フレームワークを利用した開発」を参照してください。

(2) UML を利用した開発

MyEclipse では, J2EE アプリケーションの設計で UML を使用することもできます。設計工程で UML を使用することで, オブジェクト指向スタイルでの J2EE アプリケーション開発ができます。UML を利用した J2EE アプリケーション開発については, 「9.2 UML を利用した開発」を参照してください。

(3) Eclipse を使用しない開発

Eclipse を使用しないで J2EE アプリケーションを開発できます。Eclipse を使用しない場合, 任意のエディタでプログラムを作成し, コマンドを実行してコンパイルおよびアーカイブの作成をします。Eclipse を使用しない開発については, 「9.3 Eclipse を使用しない J2EE アプリケーションの開発」を参照してください。

1.6 Windows Vista 使用時の注意事項

ここでは、OS が Windows Vista の場合に、Cosminexus を使用して J2EE アプリケーションを開発したり、テスト用のアプリケーションサーバを構築したりするときの注意事項について説明します。

1.6.1 管理者特権で実行する必要がある操作

Windows Vista を使用している場合で、管理者特権で実行する必要がある操作について説明します。なお、ここではシステムドライブを C ドライブとして説明します。

(1) プログラムのインストール

Developer や Eclipse などのプログラムを、C:\Program Files 以下のディレクトリにインストールする場合、管理者特権でインストールする必要があります。インストール時に、Windows Vista の [UAC] ダイアログが表示される場合は、[UAC] ダイアログで管理者アカウントのパスワードを入力してください。

(2) プラグインまたは MyEclipse のアンインストール

プラグインまたは MyEclipse をアンインストールする場合は、管理者特権でインスタントセットアップ機能のアンセットアップ、またはアンインストール用のコマンドを実行します。管理者特権のないユーザが、Developer で提供するプラグインまたは MyEclipse をアンインストールする場合は、-configuration を指定してアンインストール用のコマンドを実行してください。コマンドの実行については、「付録 B.5 アンインストール」を参照してください。

(3) インスタントセットアップ機能の実行

インスタントセットアップ機能は、必ず管理者特権で実行してください。管理者特権がないユーザは、インスタントセットアップ機能を起動できません。

管理者特権でインスタントセットアップ機能を実行する場合は、次に示すスタートメニューのショートカットを右クリックして、[管理者として実行] を選択してください。

[Cosminexus] - [Application Development Plug-in] - [開発環境インスタントセットアップ]

(4) Cosminexus が提供するコマンドの使用

Cosminexus が提供するコマンドは、管理者特権で実行する必要があります。

Cosminexus が提供するコマンドは、「管理者：コマンドプロンプト」で実行してください。

「管理者：コマンドプロンプト」は、Windows Vista で提供されている機能を使用して起動してください。起動方法の例を次に示します。

1. J2EE アプリケーション開発の概要

1. [スタート] ボタンをクリックします。
2. [プログラム] - [アクセサリ] を選択します。
3. [コマンド プロンプト] を右クリックして, [管理者として実行] をクリックします。
管理者のパスワードまたは確認を求められた場合は, 画面の指示に従って, パスワードを入力するか, または確認情報を提供してください。

(5) Cosminexus が提供する定義ファイルの更新

Cosminexus が提供する定義ファイルは, 管理者特権で更新する必要があります。管理者特権のないユーザが定義ファイルを更新しても, C:\Program Files 以下のディレクトリにある定義ファイルは更新されません。管理者特権のないユーザが更新したファイルは, 次に示すディレクトリ以下に保存されます。

C:\Users\< ユーザ名 >\AppData\Local\VirtualStore

なお, アプリケーションサーバは管理者特権で起動されるため, 管理者特権のないユーザが更新した定義ファイルの内容は無視されます。

(6) Eclipse または MyEclipse での操作

Eclipse または MyEclipse での次の操作では, 管理者特権が必要になります。

(a) Eclipse の起動

Eclipse を C:\Program Files 以下のディレクトリにインストールしている場合, Eclipse は管理者特権で起動する必要があります。管理者特権のないユーザが, Eclipse を起動するとエラーダイアログが表示されます。

これを回避するためには, 次のどちらかの方法で eclipse.exe を実行してください。

- 「管理者: コマンドプロンプト」で eclipse.exe を実行する。
「管理者: コマンドプロンプト」の起動方法については, 「(4) Cosminexus が提供するコマンドの使用」を参照してください。
- eclipse.exe に -configuration を指定する。
指定のしかたを次に示します。

`<Eclipseのインストールディレクトリ>\eclipse.exe -configuration <任意のフォルダ>`

なお, 管理者特権でインスタントセットアップ機能を実行したときに, [開発環境インストールセットアップ] ダイアログの [セットアップの完了] ページで [ウィザード終了後に MyEclipse を起動する] にチェックしている場合は, インスタントセットアップ機能を終了したあと, Eclipse が自動的に起動します。

(b) 制限されたフォルダに対する操作

C:\Program Files 以下のディレクトリまたは C:\windows 以下のディレクトリに対する

操作には管理者特権が必要です。これらの制限されたディレクトリに対して、管理者特権のないユーザがファイルをエクスポートしたり、Eclipse のワークスペースを保存したりすると、リダイレクション機能によって、操作対象のファイルはリダイレクトされ、次の場所に格納されます。

C:\¥Users¥< ユーザ名 >¥AppData¥Local¥VirtualStore

Eclipse 上からは、C:\¥Program Files 以下に実ファイルがあるかのように、ファイルを読み込んだり書き込んだりできますが、エクスポートまたは保存したファイルの実体は上記の場所に格納されます。

(c) デプロイ作業ディレクトリの指定

管理者特権のないユーザが、MyEclipse でデプロイ作業ディレクトリとして C:\¥Program Files 以下のディレクトリを指定していると、J2EE アプリケーションのデプロイ時にエラーダイアログが表示されます。対処方法については、「2.5.2 デプロイ作業ディレクトリの設定変更」を参照してください。

(7) ショートカットを使用した組み込みデータベースの開始および停止

スタートメニューのショートカットを使用して組み込みデータベースを開始および停止する場合は、管理者特権で実行する必要があります。

次に示すスタートメニューのショートカットを右クリックして、[管理者として実行] を選択してください。

- 組み込みデータベースを開始する場合
[Cosminexus] - [Application Development Plug-in] - [組み込みデータベースの開始]
- 組み込みデータベースを停止する場合
[Cosminexus] - [Application Development Plug-in] - [組み込みデータベースの停止]

1.6.2 JIS X0213:2004 に含まれる Unicode の補助文字を使用する場合の注意事項

Windows Vista でサポートされた JIS X0213:2004 の第三水準および第四水準の文字の一部には、Unicode の補助文字が含まれます。Unicode の補助文字とは、基本多言語面以外の文字（Unicode のコードポイントが U+10000 ~ U+10FFFF の範囲の文字）のことです。UTF-16 エンコーディングでは、サロゲートペアとして表されます。

Unicode の補助文字を使用する場合の注意事項を次に示します。

(1) リクエストで使用する場合の注意事項

アプリケーションサーバに対して、Internet Explorer 7 などのクライアントから

1. J2EE アプリケーション開発の概要

Unicode の補助文字を含むリクエストを送信した場合、Unicode の補助文字は、ログや PRF トレースに正しく出力されません。ただし、その場合も、Unicode の補助文字以外の文字は、ログや PRF トレースに正しく出力されます。

また、リクエストに Unicode の補助文字が含まれる場合も、J2EE アプリケーションは正しく動作します。

リクエストでの Unicode の補助文字の使用を制限したい場合には、J2EE アプリケーションでの対応などを検討してください。

(2) 環境構築 / 運用時の注意事項

アプリケーションサーバを構築、運用する場合、および J2EE アプリケーションやリソースアダプタをデプロイする場合に使用する定義に、Unicode の補助文字は使用できません。

Unicode の補助文字を使用できない定義の例を示します。

- EAR, WAR, JAR, EJB-JAR, サブレット, JSP, クラス, メソッド, 引数, または変数の名称
- DD 内の各種定義
- システムのインストール先の指定値
- 各種定義ファイルの設定値
- そのほか、Eclipse および MyEclipse で扱うファイル・ディレクトリ全般
(Eclipse および MyEclipse のインストール先, デプロイ作業ディレクトリなど)

2

インストールとセットアップ

この章では、J2EE アプリケーション開発で使用するプログラムのインストール、および開発環境とテスト用実行環境のセットアップの手順について説明します。また、プログラムのアンインストール、および開発環境とテスト用実行環境のアンセットアップの手順についても説明します。

この章では、インスタントセットアップ機能を使用したセットアップ手順を中心に説明します。インスタントセットアップ機能を使用しない場合は、「付録 B インスタントセットアップ機能を使用しない環境の構築手順」を参照してください。

2.1 インストールとセットアップの流れ

2.2 Developer のインストール

2.3 インスタントセットアップ機能を使用したセットアップ

2.4 開発環境の設定

2.5 開発環境の設定変更（任意の作業）

2.6 インスタントセットアップ機能で構築した J2EE サーバの設定変更

2.7 インスタントセットアップ機能で構築した環境のアンセットアップ

2.8 Developer のアンインストール

2.1 インストールとセットアップの流れ

J2EE アプリケーション開発環境で使用するプログラムをインストールおよびセットアップします。

インスタントセットアップ機能を使用したプログラムのインストールとセットアップの流れを次に示します。

図 2-1 インストールとセットアップの流れ

作業内容	参照先
1. Developerのインストール	2. 2
↓	
2. Eclipseのダウンロード MyEclipseの入手	2. 3
↓	
3. インスタントセットアップ機能を利用したセットアップ	2. 3
↓	
4. 開発環境の設定	2. 4

それぞれの作業の概要を説明します。

1. Developer のインストール

HITACHI 総合インストーラを使用して、Developer をインストールします。

2. Eclipse のダウンロードおよび MyEclipse の入手

Eclipse のアーカイブファイルのダウンロードおよび MyEclipse のアーカイブファイルの入手をします。

3. 開発環境・テスト用実行環境のセットアップ

インスタントセットアップ機能を使用して、開発環境およびテスト用実行環境をセットアップします。

4. 開発環境の設定

MyEclipse を使用して J2EE アプリケーションを開発するための設定をします。

以降の節で、これらの手順について説明します。

また、この章ではセットアップした環境の設定変更とアンセットアップ、およびプログラムのアンインストールについても説明します。それぞれの手順については、次の個所を参照してください。

- MyEclipse 使用時に必要となる設定を変更したい場合
「2.5 開発環境の設定変更（任意の作業）」
- インスタントセットアップ機能で構築した環境の設定を変更したい場合

- 「2.6 インスタントセットアップ機能で構築した J2EE サーバの設定変更」
- インスタントセットアップ機能で構築した環境をアンセットアップしたい場合
 - 「2.7 インスタントセットアップ機能で構築した環境のアンセットアップ」
- Developer をアンインストールしたい場合
 - 「2.8 Developer のアンインストール」

2.2 Developer のインストール

Developer のインストールには、HITACHI 総合インストーラを使用します。

インストールを実行する前に、構成ソフトウェアのすべてのプロセスが実行中でないことを確認してください。実行中の場合は、構成ソフトウェアのすべてのプロセスを停止してから、インストールしてください。

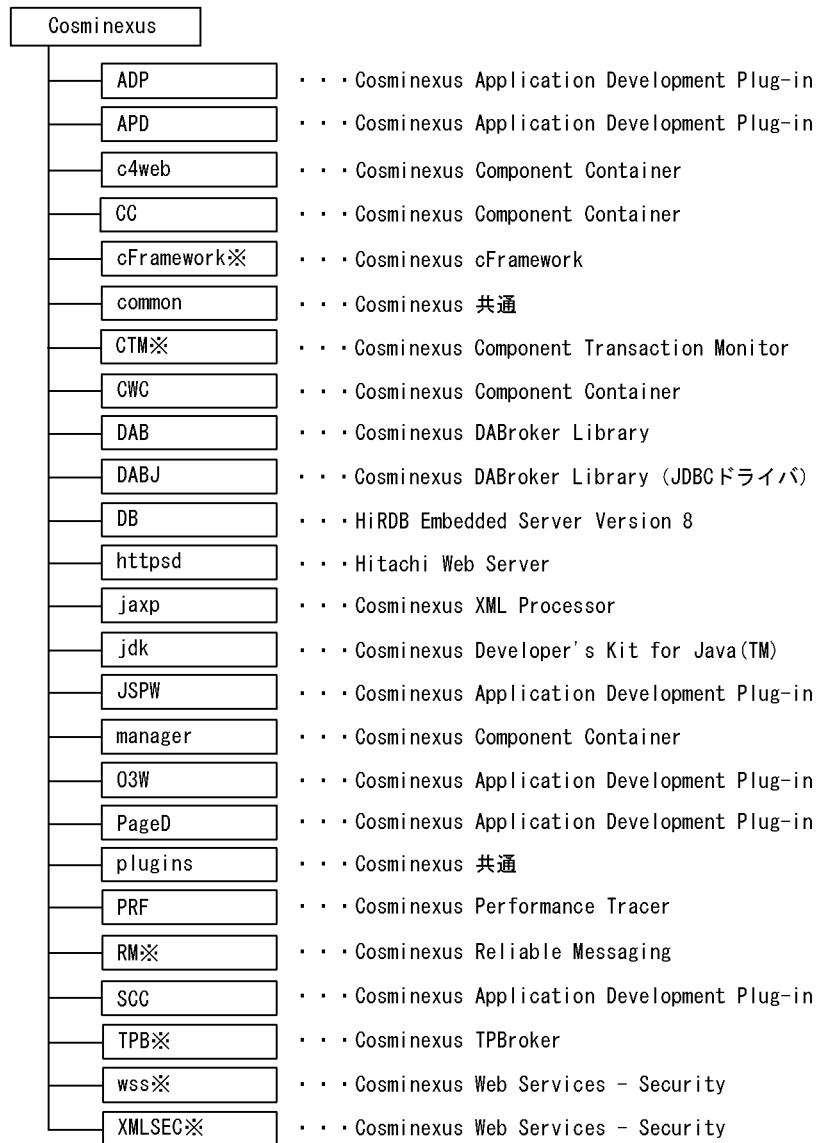
インストールするには、Administrator 権限または管理者特権が必要です。なお、Developer のインストール手順については、リリースノートを参照してください。

ここでは、インストール後のディレクトリ構成、環境変数の設定、および Cosminexus を再インストールしたときの注意事項について説明します。

2.2.1 インストール後のディレクトリ構成

Developer を C:\Program Files\Hitachi の下にインストールすると「Cosminexus」というディレクトリが作成されます。「Cosminexus」ディレクトリ以下の構成を次の図に示します。

図 2-2 Developer のディレクトリ構成



(凡例)

 : 構成ソフトウェアのインストールディレクトリ

注※ Developer Professionalの場合だけインストールされるディレクトリです。

2.2.2 環境変数の設定

Developer では、次の表に示す環境変数をシステムの環境変数に設定します。環境変数の設定が終わったら、OS を再起動してください。

2. インストールとセットアップ

表 2-1 Developer で設定する環境変数

環境変数名	設定値
PATH	<Cosminexus のインストールディレクトリ>%jdk%bin ¹
	<Cosminexus のインストールディレクトリ>%PRF%bin
	<Cosminexus のインストールディレクトリ>%TPB%bin ²
	<Cosminexus のインストールディレクトリ>%DAB%lib
	<Cosminexus のインストールディレクトリ>%DB%BIN
	<Cosminexus のインストールディレクトリ>%DB%CLIENT%UTL
TPDIR	<Cosminexus のインストールディレクトリ>%TPB ²
TZ	JST-9 など
VBROKER_ADM	<Cosminexus のインストールディレクトリ>%TPB%adm ²

注 1

環境変数 PATH の先頭に設定してください。

注 2

Developer Professional の場合にだけ指定します。

環境変数は、Windows の [コントロールパネル] 中の [システム] を選択し、[詳細設定] タブの中の [環境変数] ボタンをクリックすることで設定します。

環境変数に設定済みの定義内容がある場合は、';' (セミコロン) で区切って設定してください。環境変数が正しく設定されていない場合はエラーが発生します。

参考

テスト用実行環境のシステムとして、CTM を使用したシステムを構築する場合は、次の環境変数も設定してください。

環境変数名	設定値
OSAGENT_PORT	14000 など
CTMDIR	<Cosminexus のインストールディレクトリ>%CTM
CTMSPOOL	%CTMDIR%spool

注

必要に応じて設定してください。

2.2.3 Cosminexus を再インストールしたときの注意事項

ここでは、Cosminexus を再インストールしたときの注意事項と必要な作業について説

明します。バージョンアップによって Cosminexus を再インストールした場合、次に示すどちらかの操作が必要になります。

- 拡張ロケーションの削除

拡張ロケーションを削除します。Eclipse のインストール先 (eclipse.exe があるフォルダ) で次のコマンドを実行して、拡張ロケーションを削除します。

なお、拡張ロケーションを削除したあとは、以降の節に従って、Eclipse やプラグインをインストールしてください。

```
java -cp startup.jar org.eclipse.core.launcher.Main
-application org.eclipse.update.core.standaloneUpdate
-command removeSite
-to "<Cosminexusのインストールディレクトリ>%plugins%eclipse"
```

注 改行を入れないでコマンドを実行してください。

- Eclipse の起動引数の指定

Eclipse の起動引数に -clean を指定して再起動します。

```
<Eclipseのインストールディレクトリ>%eclipse.exe -clean
```

また、インスタントセットアップ機能で開発環境およびテスト用実行環境をセットアップしたあとで Cosminexus を再インストールする場合は、インスタントセットアップ機能を使用して事前に開発環境およびテスト用実行環境をアンセットアップしておいてください。事前にアンセットアップしなかった場合は、Cosminexus の再インストール後に手動で Management Server, J2EE サーバ, パフォーマンストレーサ, および組み込みデータベースを削除してください。

2.3 インスタントセットアップ機能を使用したセットアップ

ここでは、インスタントセットアップ機能を使用して、開発環境およびテスト用実行環境をセットアップする方法について説明します。

2.3.1 セットアップの前提条件

インスタントセットアップ機能を使用したセットアップを実行するためには、次の条件を満たしている必要があります。

(1) Eclipse および MyEclipse のアーカイブファイルの入手

インスタントセットアップ機能を実行する前に、次に示す Eclipse および MyEclipse のアーカイブファイルを入手して、任意のフォルダに格納してください。

なお、Eclipse のアーカイブファイルの入手元 URL については、リリースノートを参照してください。MyEclipse のアーカイブファイルについては、提供されている媒体から入手してください。

また、入手したアーカイブファイルをアーカイブファイル格納用ディレクトリ（<Cosminexus のインストールディレクトリ>\ADP\archives）に格納しておくとし、インスタントセットアップ機能実行時に自動的にアーカイブファイルを取り込むことができます。アーカイブファイル格納用ディレクトリに Eclipse SDK および Eclipse Platform のアーカイブファイルが両方存在する場合は、Eclipse SDK のアーカイブファイルを自動的に取り込みます。なお、Eclipse SDK には Eclipse Platform および Eclipse Java Development Tools が含まれます。

表 2-2 セットアップに必要なアーカイブファイル

アーカイブファイル	必須 / 任意
Eclipse SDK	1
Eclipse SDK のランゲージパック	2
Eclipse Platform	1
Eclipse Platform のランゲージパック	2
Eclipse Java Development Tools	3
Eclipse Java Development Tools のランゲージパック	2
MyEclipse for Cosminexus	

(凡例)

：必須 ：任意

注

Eclipse のバージョン 3.2.2 以上に対応するアーカイブファイルを入手してください。

注 1

必要に応じて、どちらかを入手してください。

注 2

ランゲージパックを使用する場合は、対応するアーカイブファイルを入手してください。なお、サンプルプログラム（Bank）を実行する場合は、必ず入手してください。

注 3

Eclipse SDK をダウンロードする場合は、必要ありません。

（2）インスタントセットアップ機能を実行する環境

次の条件に当てはまる場合、インスタントセットアップ機能で開発環境およびテスト用実行環境をセットアップできません。次に示す対応をしてください。

- インスタントセットアップ機能以外の方法で Management Server をセットアップしている場合は、Management Server をアンセットアップする。
- インスタントセットアップ機能以外の方法でホストやサーバを構築している場合は、ホストおよびサーバを削除する。
- インスタントセットアップ機能以外の方法で組み込みデータベースをセットアップしている場合は、組み込みデータベースをアンセットアップする。
- コマンド拡張機能が無効の場合は、有効にする。
- 次のサービスのスタートアップの状態が「無効」になっている場合は、「自動」または「手動」に設定してください。
 - Cosminexus Management Server
 - Cosminexus Management Server - Administration Agent
 - HiRDB/EmbeddedEdition_CSO

2.3.2 セットアップする環境の設定内容

インスタントセットアップ機能でセットアップできる内容は、次のとおりです。

- Eclipse および MyEclipse のインストール
- アプリケーションサーバ の構築
- 組み込みデータベースの構築

注

Management Server、ホスト、パフォーマンストレーサ、および J2EE サーバを含みます。

HiRDB SQL Executer については、セットアップ終了後にインストールしてください。HiRDB SQL Executer のインストール手順は、「3.2 HiRDB SQL Executer のインストール」を参照してください。

セットアップの実行方法は、標準セットアップとカスタムセットアップのどちらかを選

2. インストールとセットアップ

択できます。標準セットアップを実行した場合は、表 2-3 に示す値で環境が構築されます。

カスタムセットアップを実行した場合は、次の内容を任意の値に変更できます。

- Eclipse および MyEclipse のインストールディレクトリ
- J2EE サーバのポート番号
- 組み込みデータベース構築ディレクトリ（設定ファイルおよび RD エリアのファイル用ディレクトリ）

インスタントセットアップ機能がセットアップする環境の設定値を次の表に示します。構築したい環境に合わせて、標準セットアップまたはカスタムセットアップを選択してください。

表 2-3 インスタントセットアップ機能でセットアップする環境の設定値

環境	設定項目	標準セットアップで設定される値	カスタム セット アップで の変更
Management Server	サーバ名	cosmi_m	×
	Server Plug-in との連携	連携する	×
	コンソールへの情報出力	出力する	×
	運用管理エージェントの 起動	自動	×
	Management Server の起 動	自動	×
	接続ポート番号	28099 ¹	×
	接続 HTTP ポート番号	28080	×
	終了要求受信ポート番号	28005	×
	内部通信用ポート番号	28009	×
	運用管理エージェントの ポート番号	20295	×
	ユーザ ID ²	admin	×
	パスワード ²	admin	×
ホスト	構築できるホストの数	1	×
	ホスト名	Smart Composer 機能が生成するホ スト名	×
	表示名	InstantHost	×
J2EE サーバ	構築できるサーバの数	1	×
	論理サーバ名	Smart Composer 機能が生成する論 理サーバ名	×

環境	設定項目	標準セットアップで設定される値	カスタム セット アップで の変更
	表示名	InstantJ2EEServer	×
	デバッグ接続のための ポート番号	3999 ³	
	インプロセス HTTP サー バ	使用する	×
	HTTP のポート番号	80	
	簡易 Web サーバのポート 番号	8080	
	アクセスを許可するホス トの定義	すべてのホストからのアクセスを許 可する	×
	RMI レジストリのポート 番号	23152	
	ネーミングサービスの ポート番号 (インプロセ スネーミングサービスの ポート番号)	900	
	ライトランザクション	有効	×
	JTA リカバリの固定ポ ート番号	20302	×
	リロード機能	リロード対象と設定されるファイル は次のとおりです。 • EJB-JAR • サンプル • JSP	×
	リロードの更新検知のイ ンターバル	1 秒	×
	セキュリティマネージャ	使用しない	×
	クラスパス	add.class.path=<Cosminexus のイ ンストールディレクトリ >¥DB¥CLIENT¥UTL¥pdjdbc2.ja r	×
パフォーマンス ストレサ	構築できるサーバの数	1	×
	論理サーバ名	Smart Composer 機能が生成する論 理サーバ名	×
	表示名	InstantPRF	×
組み込みデー タベース	ポート番号	22200	×
	サイズ	L (660MB)	×

2. インストールとセットアップ

環境	設定項目	標準セットアップで設定される値	カスタム セット アップで の変更
	ユーザ ID	USER1	×
	パスワード	PSWD	×
	開始・停止	ユーザの操作 ⁴	×
	テーブル ⁵	<ul style="list-style-type: none"> • checking • saving 	×
	組み込みデータベース構築ディレクトリ（設定ファイルおよび RD エリアのファイル用ディレクトリ）	<Cosminexus のインストールディレクトリ>¥ADP¥DB¥	
リソースアダプタ	インポート	DBConnector_HiRDB_Type4_CP.rar をインポートする	×
	表示名 ⁶	DB_Connector_for_Cosminexus_Driver ⁷	×
	ユーザ ID ⁶	USER1 ⁸	×
	パスワード ⁶	PSWD ⁸	×
	description（コンフィグレーションプロパティ） ⁶	22200 ⁸	×
	DBHostName（コンフィグレーションプロパティ） ⁶	localhost	×
	コネクション・プールの最小値 ⁶	1	×
	コネクション・プールの最大値 ⁶	8	×
	デプロイ	デプロイする	×
開発環境	Eclipse のインストールディレクトリ	<環境変数 USERPROFILE が指すディレクトリ>¥Eclipse	
	MyEclipse のインストールディレクトリ	<環境変数 USERPROFILE が指すディレクトリ>¥MyEclipse	

（凡例）

：カスタムセットアップで変更できます。

×：カスタムセットアップで変更できません。標準セットアップと同じ値が設定されます。

注 1

Cosminexus コネクタおよび Server Plug-in から Management Server に接続するために使用します。

注 2

セットアップしたあとに変更しないでください。変更すると、インスタントセットアップ機能の設定変更およびアンセットアップが実行できなくなります。

注 3

Eclipse のデバッグ構成で使します。

注 4

スタートメニューのショートカットまたはコマンドを使用して、組み込みデータベースの開始および停止ができます。詳しい手順については、「7.2 組み込みデータベースの開始および停止」を参照してください。

注 5

Developer が提供する MyEclipse プロジェクトのサンプル (Bank) 用のテーブルが作成されます。作成されるテーブルの内容は、次のとおりです。

```
CREATE SCHEMA;
create table checking
(id varchar(3) primary key,
balance double precision);
create table saving
(id varchar(3) primary key,
balance double precision);
```

また、セットアップ終了後、必要に応じてテーブルを作成できます。テーブルの作成方法については、「3.3.2 テーブルの作成」を参照してください。

注 6

インポートしたリソースアダプタの Connector 属性として設定されます。

注 7

サンプルプログラム (Bank) が前提とするリソースアダプタ名が設定されます。

注 8

組み込みデータベースの設定値と同じ値が設定されます。

2.3.3 標準セットアップ

標準セットアップでは、あらかじめ設定された値で環境をセットアップします。標準セットアップを実行するためには、次の条件を満たしている必要があります。

Eclipse および MyEclipse のインストールディレクトリ

- ディレクトリのパスの長さが全角文字または半角文字で 50 文字以内であること。
- 同じパスのファイルが存在しないこと。
- ディレクトリの直下に eclipse ディレクトリ、または eclipse ファイルが存在しないこと。

組み込みデータベース構築ディレクトリ (設定ファイルおよび RD エリアのファイル用ディレクトリ)

- ディレクトリのパスの長さが半角文字で 43 バイト以内であること。
- 同じパスのファイルが存在しないこと。
- 次に示すディレクトリおよびファイルがディレクトリの直下に存在しないこと。

2. インストールとセットアップ

- ・ area ディレクトリ, area ファイル
- ・ bats ディレクトリ, bats ファイル
- ・ conf ディレクトリ, conf ファイル
- ・ ini ディレクトリ, ini ファイル

ここでは、標準セットアップを実行する場合の手順を説明します。

! 注意事項

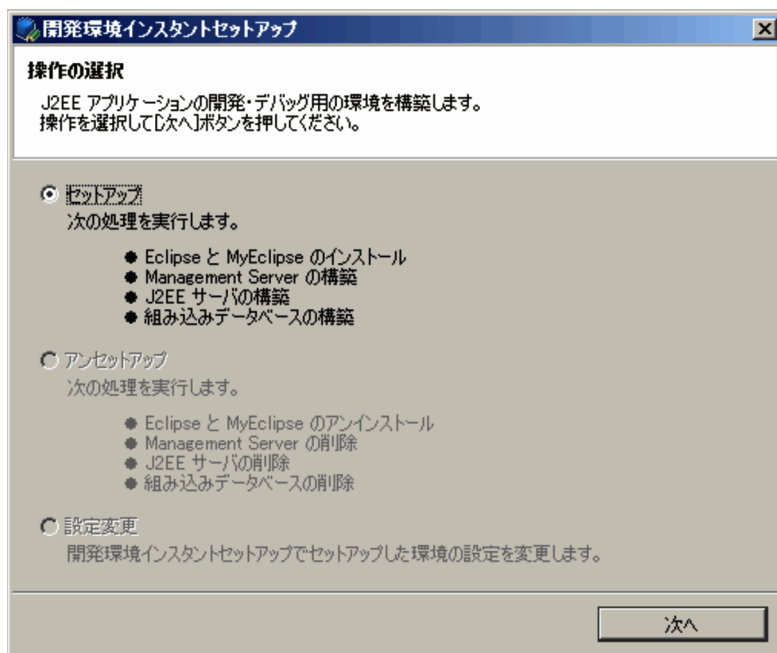
Windows セキュリティの重要な警告について

インスタントセットアップ機能の実行中に、[Windows セキュリティの重要な警告] ダイアログが表示された場合は、直ちに [ブロックを解除する] ボタンをクリックする必要があります。

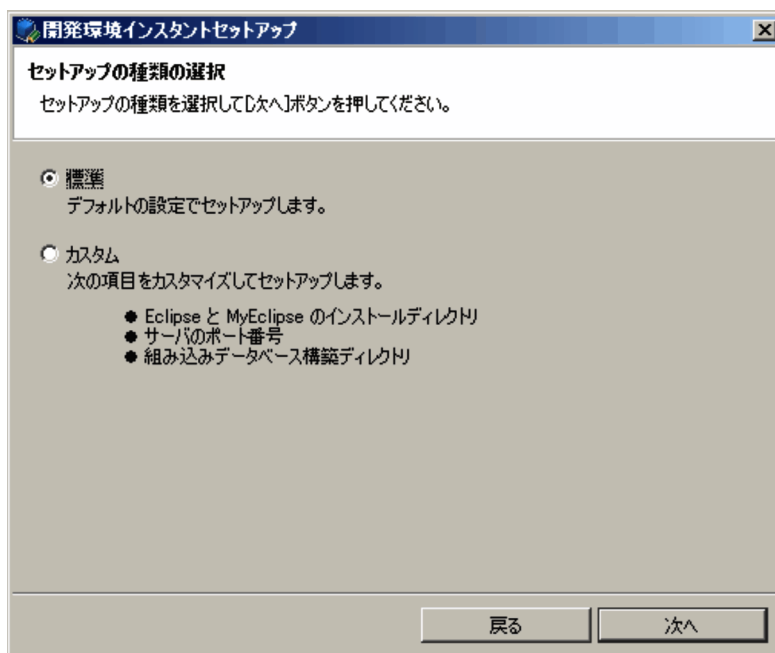
指定したアーカイブファイルについて

インスタントセットアップ機能で指定したアーカイブファイルは、削除または名称変更しないでください。

1. スタートメニューから、[プログラム] - [Cosminexus] - [Application Development Plug-in] - [開発環境インスタントセットアップ] を選択します。
インスタントセットアップ機能が起動して、[開発環境インスタントセットアップ] ダイアログの [操作の選択] ページが表示されます。

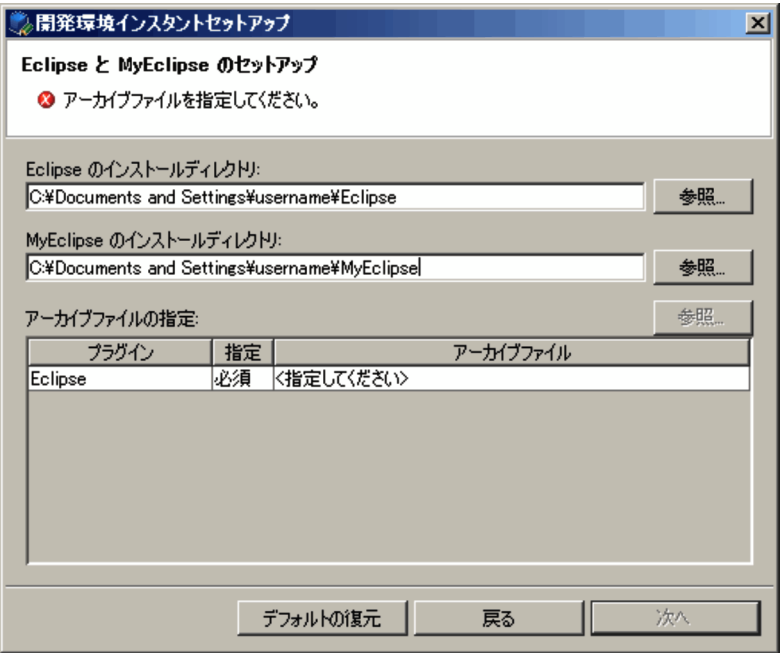


2. [セットアップ] を選択して、[次へ] ボタンをクリックします。
[セットアップの種類の選択] ページが表示されます。



3. [標準] を選択して , [次へ] ボタンをクリックします。
 次のアーカイブファイルがアーカイブファイル格納用ディレクトリ (<Cosminexus
 のインストールディレクトリ >¥ADP¥archives) に格納されていない場合は ,
 [Eclipse と MyEclipse のセットアップ] ページが表示されます。
 - Eclipse Platform および Eclipse Java Development Tools のアーカイブファイル ,
 または Eclipse SDK のアーカイブファイル
 - MyEclipse のアーカイブファイル
 上記のアーカイブファイルがアーカイブファイル格納用ディレクトリに格納されてい
 る場合は , [セットアップの確認] ページが表示されます。手順 6. に進んでください。
4. Eclipse および MyEclipse のインストールディレクトリと , アーカイブファイルを設
 定します。

2. インストールとセットアップ



設定が必要な項目を次に示します。

項目名		設定値
アーカイブファイルの指定	Eclipse	[参照] ボタンから Eclipse SDK または Eclipse Platform のアーカイブファイルを指定します。
	Java Development Tools	Java Development Tools のアーカイブファイルを指定します。 なお、Eclipse SDK を指定した場合は、指定しません。
	MyEclipse	MyEclipse のアーカイブファイルを指定します。

注

Eclipse のアーカイブファイルを指定すると、[プラグイン] 列に項目名が表示されます。なお、各アーカイブファイルがすでに指定してあるアーカイブファイルと同じディレクトリに格納してある場合は、[アーカイブファイル] 列も自動で設定されます。
必要に応じて設定する項目を次に示します。

項目名	設定値
Eclipse のインストールディレクトリ	Eclipse のインストールディレクトリを 50 文字以内で指定します。なお、文字列は全角、半角どちらでもかまいません。 デフォルトの設定から変更したい場合は、[参照] ボタンをクリックして、[フォルダの参照] ダイアログから指定してください。

項目名		設定値
MyEclipse のインストールディレクトリ		MyEclipse のインストールディレクトリを 50 文字以内で指定します。なお、文字列は全角、半角どちらでもかまいません。デフォルトの設定から変更したい場合は、[参照] ボタンをクリックして、[フォルダの参照] ダイアログから指定してください。
アーカイブファイルの指定	Eclipse ランゲージパック	Eclipse ランゲージパックのアーカイブファイルを指定します。必要に応じて設定してください。 なお、サンプルプログラム (Bank) を実行する場合は、必ず設定してください。
	Java Development Tools ランゲージパック	Java Development Tools ランゲージパックのアーカイブファイルを指定します。必要に応じて設定してください。 なお、サンプルプログラム (Bank) を実行する場合は、必ず設定してください。

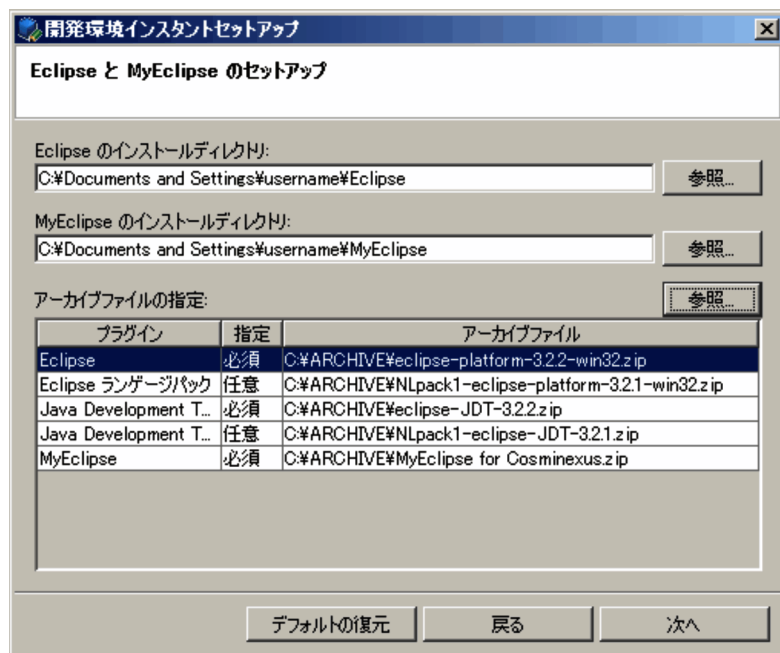
注

Eclipse のアーカイブファイルを指定すると、[プラグイン] 列に項目名が表示されます。なお、各アーカイブファイルがすでに指定してあるアーカイブファイルと同じディレクトリに格納してある場合は、[アーカイブファイル] 列も自動で設定されます。

注

[アーカイブファイルの指定] テーブルで指定したアーカイブファイルは、削除またはリネームしないでください。

すべての項目を指定した場合は、次の図のようになります。



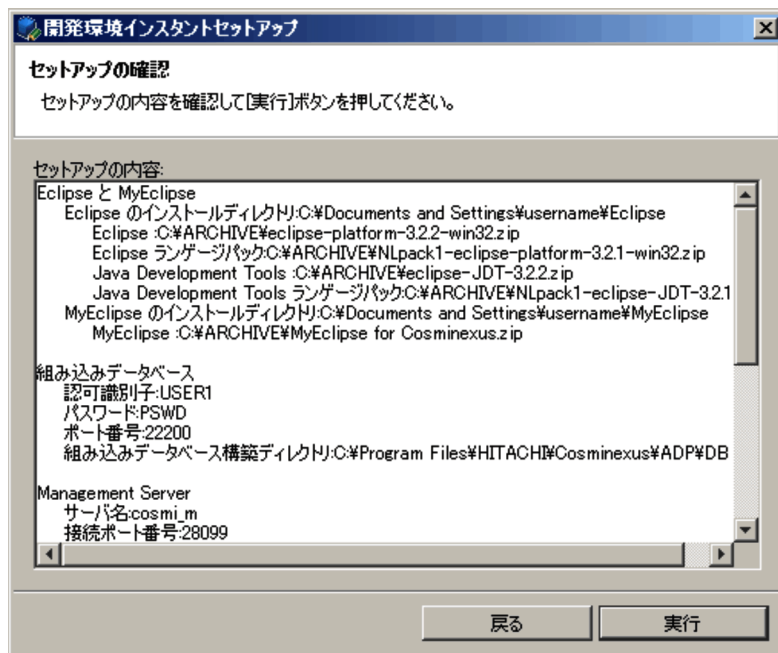
2. インストールとセットアップ

5. [次へ] ボタンをクリックします。

[セットアップの確認] ページが表示されます。

6. 設定値を確認します。

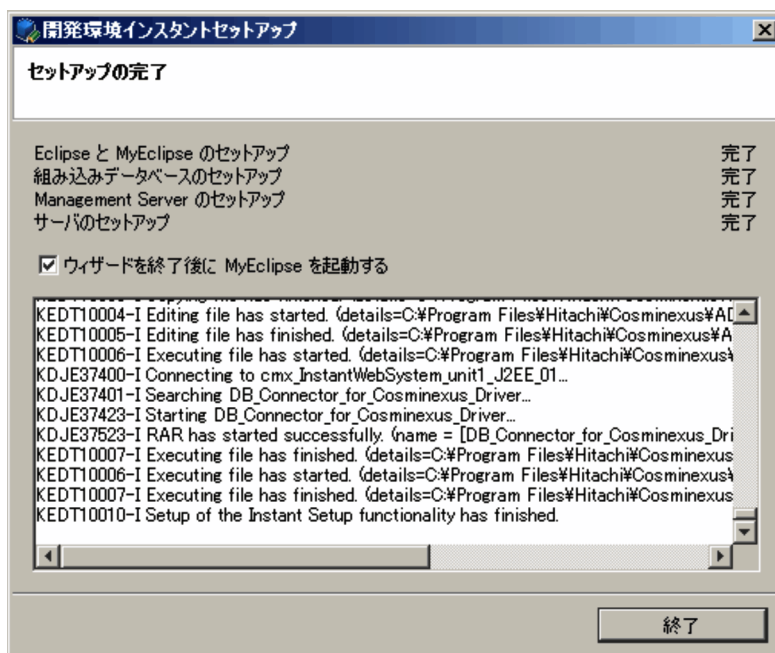
[セットアップの内容] エリアにセットアップされる内容が表示されるので、設定値を確認してください。標準セットアップで構築される環境の設定内容の詳細は、「2.3.2 セットアップする環境の設定内容」を参照してください。



7. [実行] ボタンをクリックします。

[進行状況] ページが表示されます。

セットアップが終了すると、[セットアップの完了] ページが表示されます。



8. [終了] ボタンをクリックします。

[開発環境インスタントセットアップ] ダイアログが閉じます。

[開発環境インスタントセットアップ] ダイアログ終了後に MyEclipse を起動させたい場合は、[ウィザードを終了後に MyEclipse を起動する] をチェックしてから、[終了] ボタンをクリックしてください。

参考

インスタントセットアップ機能の起動オプションの使用

注意事項ダイアログで設定した [次回からこのダイアログを表示しない] の設定を取り消して起動したい場合は、起動オプション -resetdialog を指定して、インスタントセットアップ機能を起動します。

次の起動オプションを指定して、実行してください。

```
<Cosminexusのインストールディレクトリ>%ADP%\InstantSetup\bin\InstantSetup  
-resetdialog
```

エラーが発生した場合の対処

セットアップの実行中にエラーが発生した場合は、コンソールにエラーメッセージが表示されます。

Eclipse, MyEclipse, Management Server, および J2EE サーバのセットアップ時にエラーが発生した場合は、次の表に示す付録またはマニュアルを参照して、対処してください。

2. インストールとセットアップ

表 2-4 エラーメッセージの参照先（Eclipse，MyEclipse，Management Server，J2EE サーバのセットアップ時）

プリフィックス	参照先
KDAL KDJE	マニュアル「Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編」
KEOS KEUC KFCB	マニュアル「Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編」
KFCT KFDB KFDJ	マニュアル「Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編」
KEDT	付録 I インスタントセットアップ機能実行時のエラーメッセージ一覧

組み込みデータベースのセットアップ時にエラーが発生した場合は、マニュアル「HiRDB メッセージ」を参照してください。

エラーの対処が終わったら、アンセットアップを実行して、再度セットアップしてください。アンセットアップの手順については、「2.7 インスタントセットアップ機能で構築した環境のアンセットアップ」を参照してください。

セットアップ後の各サーバの状態

インスタントセットアップ機能を使用してセットアップすると、各サーバは次の状態になります。

表 2-5 セットアップ後の各サーバの状態

サーバ名	状態
運用管理エージェント	
Management Server	
J2EE サーバ	×
パフォーマンストレーサ	
組み込みデータベース	

（凡例）

：開始 ×：停止

MyEclipse の起動方法

セットアップが終了したあとに手動で MyEclipse を起動する場合は、起動オプション -clean を指定して、MyEclipse を起動してください。

```
<Eclipseのインストールディレクトリ>%eclipse%eclipse.exe -clean
```

2.3.4 カスタムセットアップ

カスタムセットアップでは、次の内容を指定して、環境を構築できます。

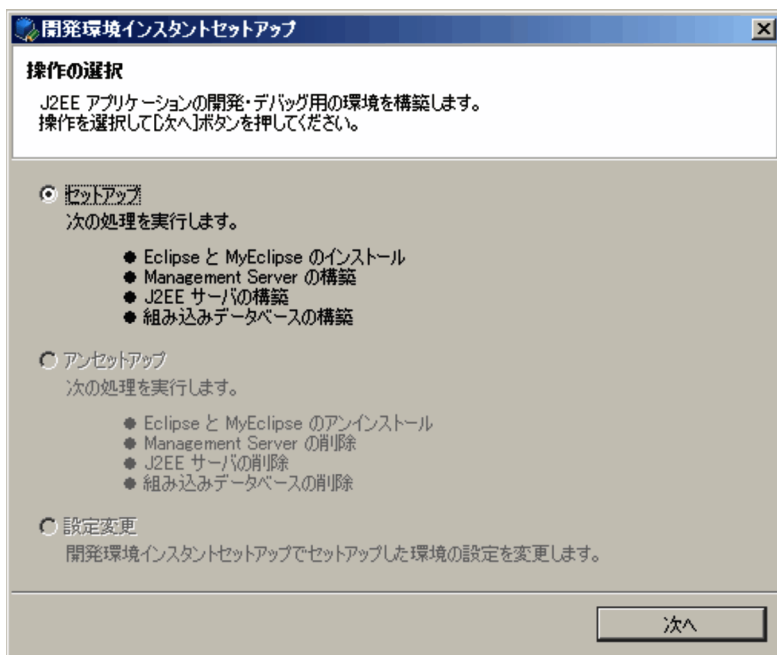
- Eclipse および MyEclipse のインストールディレクトリ
- J2EE サーバのポート番号
- 組み込みデータベース構築ディレクトリ（設定ファイルおよび RD エリアのファイル用ディレクトリ）

ここでは、カスタムセットアップを実行する場合の手順を説明します。

！ 注意事項

インスタントセットアップ機能の実行中の注意事項については、「2.3.3 標準セットアップ」の注意事項を参照してください。

1. スタートメニューから、[プログラム] - [Cosminexus] - [Application Development Plug-in] - [開発環境インスタントセットアップ] を選択します。
インスタントセットアップ機能が起動して、[開発環境インスタントセットアップ] ダイアログの [操作の選択] ページが表示されます。



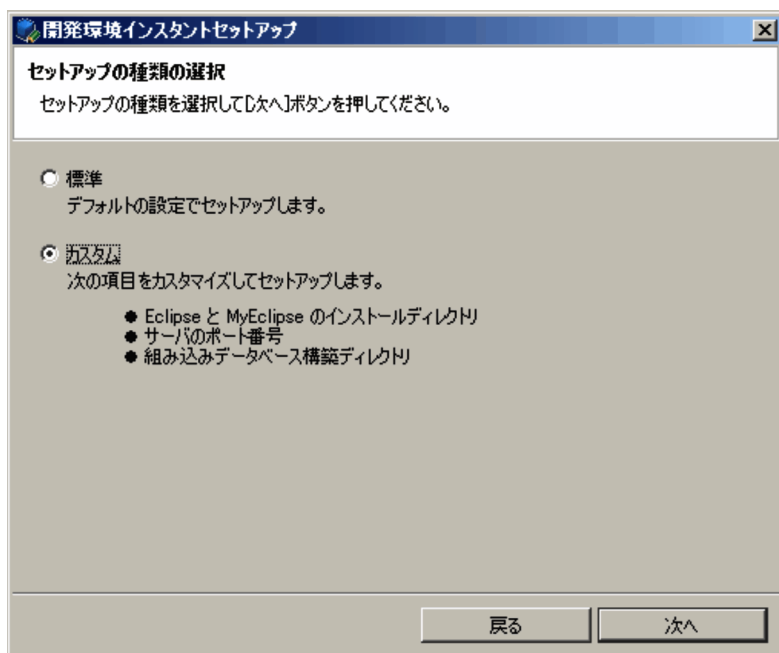
参考

起動オプションを使用して、インスタントセットアップ機能を起動したい場合は、「2.3.3 標準セットアップ」の参考事項を参照してください。

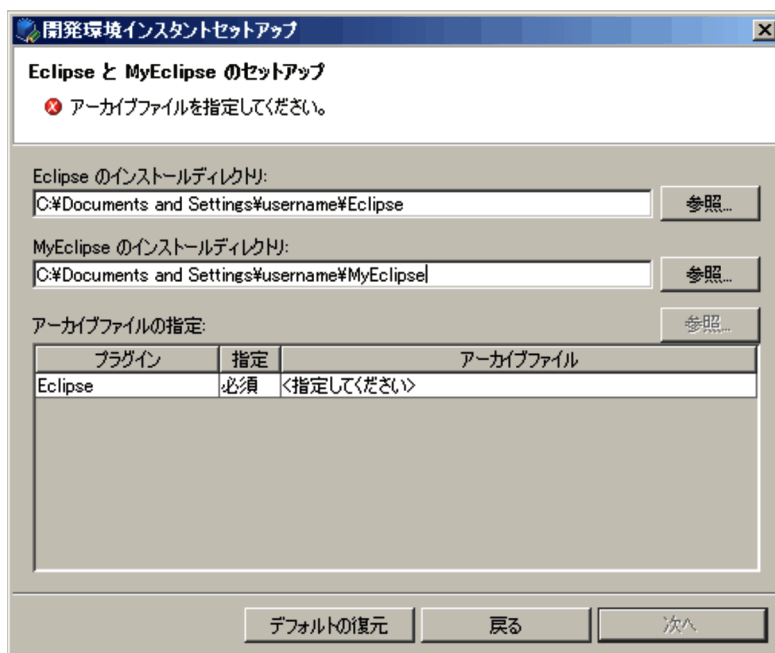
2. [セットアップ] を選択して、[次へ] ボタンをクリックします。

2. インストールとセットアップ

[セットアップの種類を選択] ページが表示されます。



3. [カスタム] を選択して , [次へ] ボタンをクリックします。
[Eclipse と MyEclipse のセットアップ] ページが表示されます。
4. Eclipse および MyEclipse のインストールディレクトリと , アーカイブファイルを設定します。



設定が必要な項目を次に示します。

項目名		設定値
アーカイブファイルの指定	Eclipse	[参照] ボタンから Eclipse SDK または Eclipse Platform のアーカイブファイルを指定します。
	Java Development Tools	Java Development Tools のアーカイブファイルを指定します。 なお、Eclipse SDK を指定した場合は、指定しません。
	MyEclipse	MyEclipse のアーカイブファイルを指定します。

注

Eclipse のアーカイブファイルを指定すると、[プラグイン] 列に項目名が表示されます。なお、各アーカイブファイルがすでに指定してあるアーカイブファイルと同じディレクトリに格納してある場合は、[アーカイブファイル] 列も自動で設定されます。

必要に応じて設定する項目を次に示します。

項目名	設定値
Eclipse のインストールディレクトリ	Eclipse のインストールディレクトリを 50 文字以内で指定します。なお、文字列は全角、半角どちらでもかまいません。 デフォルトの設定から変更したい場合は、[参照] ボタンをクリックして、[フォルダの参照] ダイアログから指定してください。

2. インストールとセットアップ

項目名		設定値
MyEclipse のインストールディレクトリ		MyEclipse のインストールディレクトリを 50 文字以内で指定します。なお、文字列は全角、半角どちらでもかまいません。デフォルトの設定から変更したい場合は、[参照] ボタンをクリックして、[フォルダの参照] ダイアログから指定してください。
アーカイブファイルの指定	Eclipse ランゲージパック	Eclipse ランゲージパックのアーカイブファイルを指定します。必要に応じて設定してください。なお、サンプルプログラム (Bank) を実行する場合は、必ず設定してください。
	Java Development Tools ランゲージパック	Java Development Tools ランゲージパックのアーカイブファイルを指定します。必要に応じて設定してください。なお、サンプルプログラム (Bank) を実行する場合は、必ず設定してください。

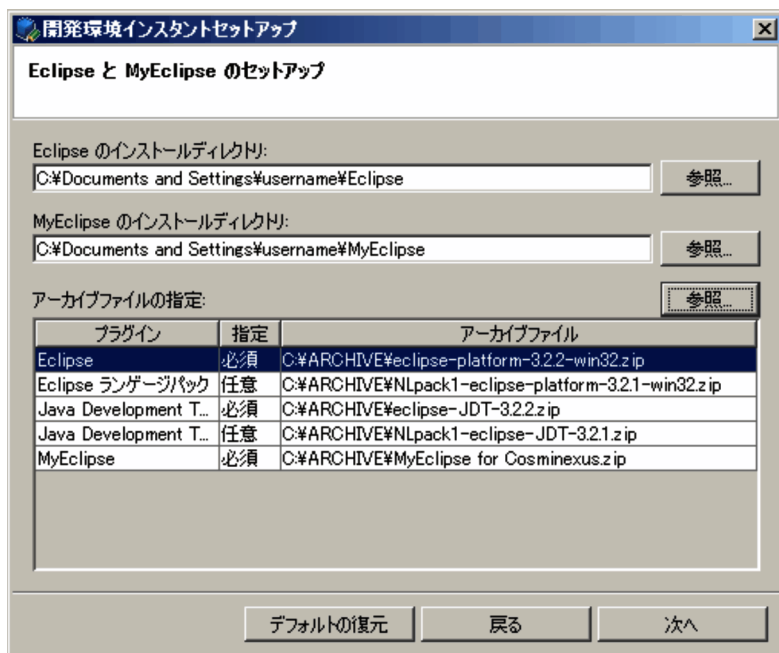
注

Eclipse のアーカイブファイルを指定すると、[プラグイン] 列に項目名が表示されます。なお、各アーカイブファイルがすでに指定してあるアーカイブファイルと同じディレクトリに格納してある場合は、[アーカイブファイル] 列も自動で設定されます。

注

[アーカイブファイルの指定] テーブルで指定したアーカイブファイルは、削除またはリネームしないでください。

すべての項目を指定した場合は、次の図のようになります。



5. [次へ] ボタンをクリックします。
[サーバのセットアップ] ページが表示されます。

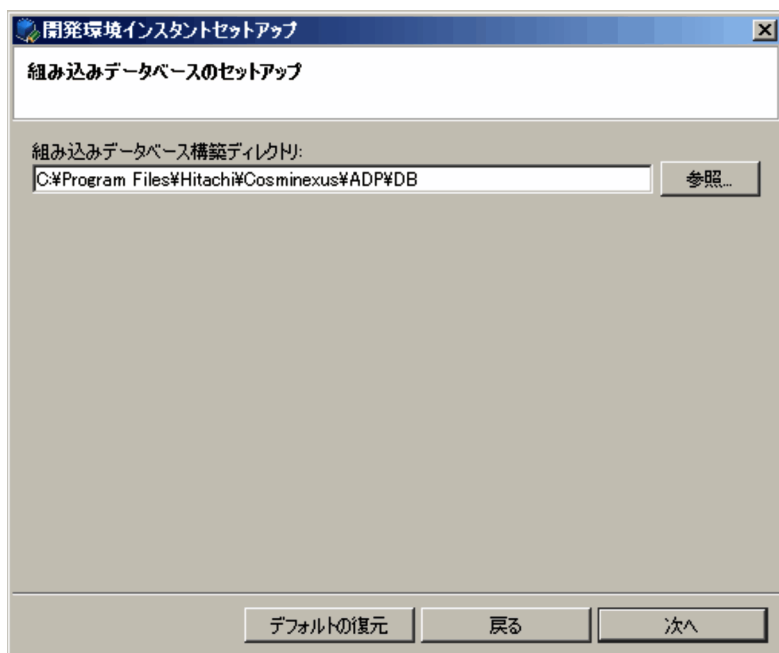
6. J2EE サーバで使用するポート番号を設定します。
[] および [] ボタン, [] および [] キー, または直接数値を入力して設定してください。

項目名	設定値
デバッグ接続のためのポート番号	J2EE サーバにデバッグ接続するために使用するポート番号を半角数字で設定します。 設定できる範囲は 0 ~ 65535 です。
HTTP のポート番号	J2EE サーバが利用する HTTP のポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。
簡易 Web サーバのポート番号	J2EE サーバが利用する簡易 Web サーバのポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。
RMI レジストリのポート番号	J2EE サーバが利用する RMI レジストリのポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。
ネーミングサービスのポート番号	J2EE サーバがネーミングサービスとして利用する CORBA ネーミングサービスのポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。

7. [次へ] ボタンをクリックします。

2. インストールとセットアップ

[組み込みデータベースのセットアップ] ページが表示されます。



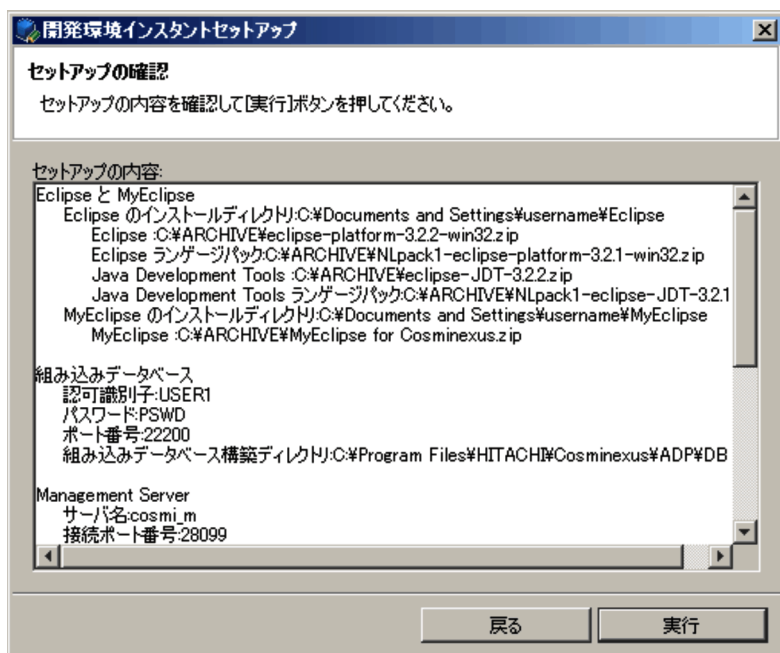
8. 必要に応じて、[組み込みデータベース構築ディレクトリ] を設定します。
組み込みデータベースの設定ファイルおよび RD エリアのファイルを格納するディレクトリを 43 バイト以内で指定します。ただし、半角文字で指定してください。
デフォルトの設定 (<Cosminexus のインストールディレクトリ>¥ADP¥DB¥) から変更したい場合は、[参照] ボタンから格納するディレクトリを指定できます。

注意

[組み込みデータベース構築ディレクトリ] には、次のディレクトリは指定できません。

- ・ドライブ直下 (C:¥ など)
- ・パスに日本語 (半角かたかなを含む) などのマルチバイト文字を含むディレクトリ

9. [次へ] ボタンをクリックします。
[セットアップの確認] ページが表示されます。



10. 設定値を確認します。

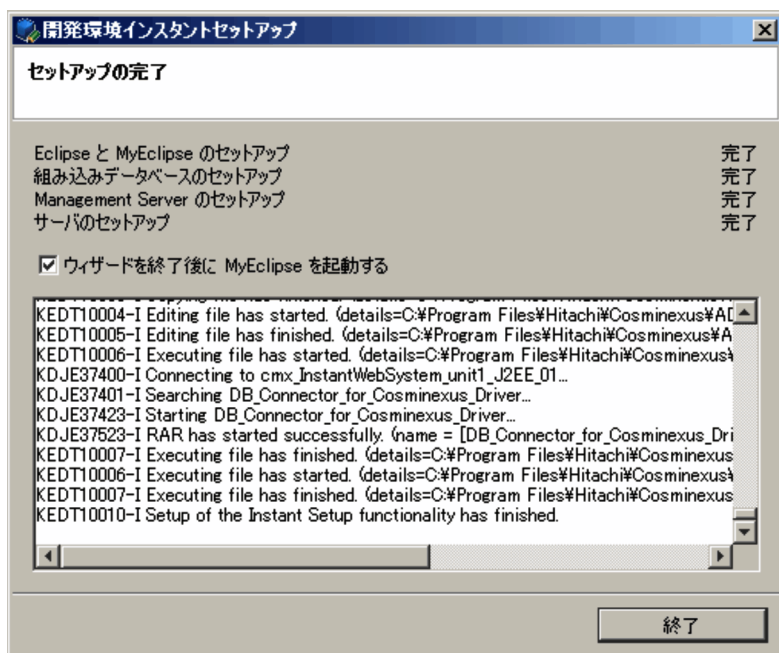
[セットアップの内容] エリアにセットアップされる内容が表示されるので、設定値を確認してください。カスタムセットアップで指定しなかった設定値については、「2.3.2 セットアップする環境の設定内容」を参照してください。

11. [実行] ボタンをクリックします。

[進行状況] ページが表示されます。

セットアップが終了すると、[セットアップの完了] ページが表示されます。

2. インストールとセットアップ



12[終了] ボタンをクリックします。

[開発環境インスタントセットアップ] ダイアログが閉じます。

[開発環境インスタントセットアップ] ダイアログ終了後に MyEclipse を起動させたい場合は、[ウィザードを終了後に MyEclipse を起動する] をチェックしてから、[終了] ボタンをクリックしてください。

参考

エラーが発生した場合の対処，セットアップ後の各サーバの状態，MyEclipse の起動方法については、「2.3.3 標準セットアップ」の参考事項を参照してください。

2.4 開発環境の設定

ここでは、J2EE アプリケーションを開発するために必要な設定について説明します。設定の流れを次の図に示します。

作業内容	参照先
1. Eclipseの設定	2.4.1
↓	
2. Management Server リモート管理機能へのログイン	2.4.2
↓	
3. J2EEサーバの選択	2.4.3
↓	
4. Server Plug-inの設定	2.4.4

それぞれの作業の概要を説明します。

1. Eclipse の設定

Eclipse で使用する JDK , およびローカル変数情報の出力について設定します。

2. Management Server リモート管理機能へのログイン

3. 以降の作業を実施するために、MyEclipse から Management Server リモート管理機能へログインします。

3. J2EE サーバの選択

MyEclipse から操作する J2EE サーバを選択します。選択した J2EE サーバは、MyEclipse から起動および停止できます。また、J2EE アプリケーションをデプロイする J2EE サーバとして使用されます。

4. Server Plug-in の設定

Server Plug-in を設定します。Server Plug-in は、J2EE リソースのプロパティ設定などで使用します。

以降の項では、この流れに沿って開発環境の設定の手順を説明します。

2.4.1 Eclipse の設定

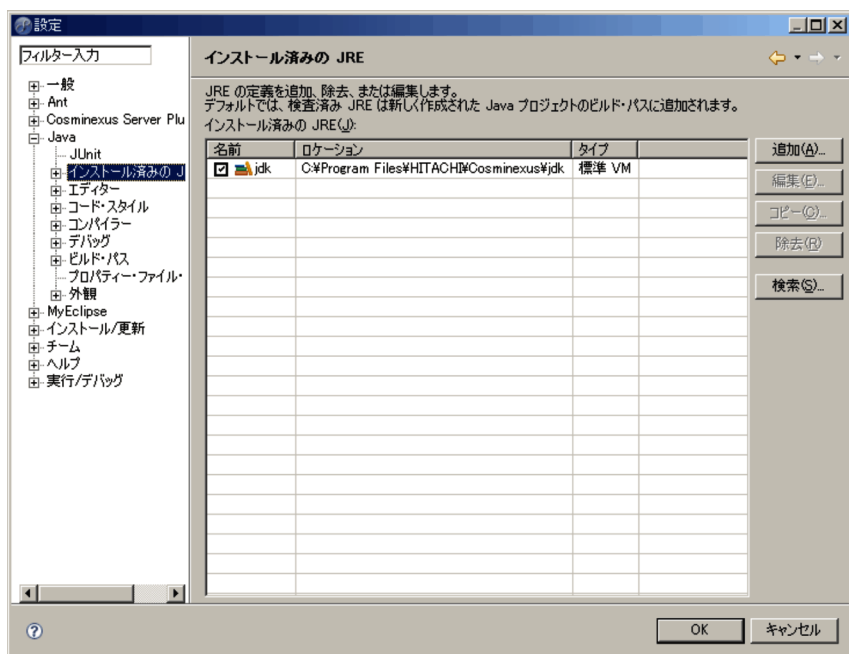
ここでは、Eclipse 上で JDK の設定を確認する手順を示します。また、J2EE アプリケーションが持つローカル変数情報を出力する場合の設定手順を示します。

2. インストールとセットアップ

(1) JDK の確認

Eclipse 上で開発するとき使用する JDK が Cosminexus で提供されている JDK かどうかを確認します。確認手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインで [Java] - [インストール済みの JRE] を選択します。
右ペインに [インストール済みの JRE] ページが表示されます。



3. [インストール済みの JRE] が Cosminexus で提供されている JDK かどうかを確認します。
[ロケーション] に次のパスが表示されているかどうかを確認します。
<Cosminexus のインストールディレクトリ>\jdk
 - パスが表示されていない場合
[追加] ボタンをクリックして、上記のパスを設定します。パスの設定後、[名前] のチェックボックスにチェックします。
 - パスが表示されている場合
[名前] のチェックボックスにチェックしているかどうかを確認します。チェックしていない場合はチェックしてください。
特に、複数の JDK がインストールされている場合、<Cosminexus のインストールディレクトリ>\jdk にチェックしていないことがあります。チェックしていない場合は、チェックしてください。
4. [OK] ボタンをクリックします。

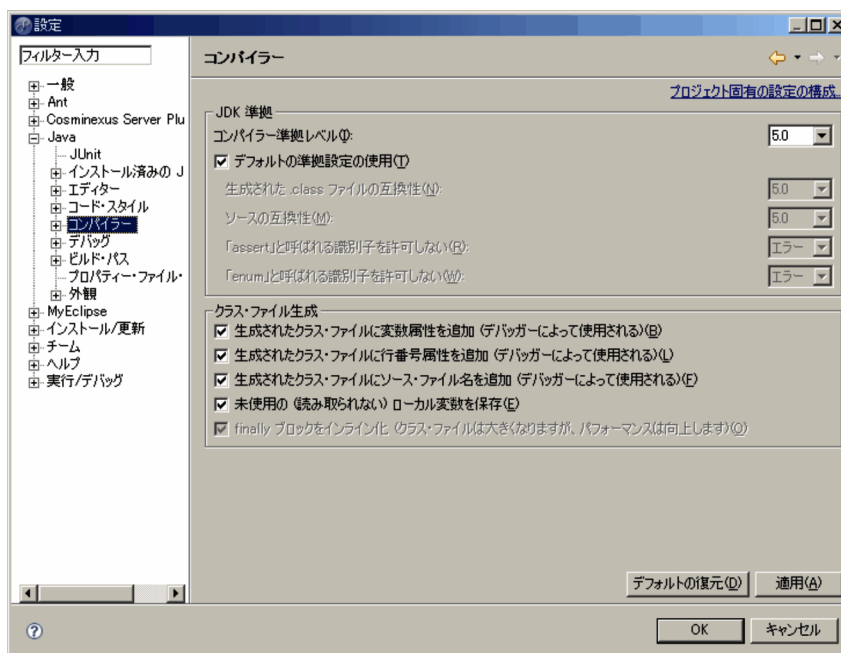
設定が保存されます。

(2) ローカル変数情報の出力の設定

Eclipse のコンパイラの設定によって、例外発生時に J2EE アプリケーションが持つローカル変数の情報をスタックトレース上に出力できます。ローカル変数情報の出力については、マニュアル「Cosminexus システム運用ガイド」に記載されているスタックトレースの調査に関する説明を参照してください。

ローカル変数情報を出力する場合のコンパイラの設定手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインで [Java] - [コンパイラー] を選択します。
[コンパイラー] ページが表示されます。



設定が必要な項目を次に示します。

項目名	設定値
コンパイラー準拠レベル	「5.0」を選択します。
クラス・ファイル生成	[生成されたクラス・ファイルに変数属性を追加 (デバッガーによって使用される)] をチェックします。 なお、それ以外の項目については必要に応じてチェックしてください。

必要に応じて、次の項目を設定してください。

2. インストールとセットアップ

項目名	設定値
デフォルトの準拠設定の使用	必要に応じて、コンパイラに使用する設定を指定します。 <ul style="list-style-type: none">• チェックする [コンパイラ準拠レベル] で指定したレベルに沿った設定が適用されます。• チェックしない 次の項目を手動で設定します。 [生成された .class ファイルの互換性] [ソースの互換性] [「assert」と呼ばれる識別子を許可しない] [「enum」と呼ばれる識別子を許可しない]

3. [適用] ボタンまたは [OK] ボタンをクリックします。

設定が保存されます。

プロジェクト単位で設定を変更する場合は、Eclipse のメニューの [プロジェクト] - [プロパティ] - [Java コンパイラ] を選択することで、同様の設定ができます。

2.4.2 Management Server リモート管理機能へのログインおよびログアウト (MyEclipse)

Management Server リモート管理機能にログインします。MyEclipse から Management Server リモート管理機能にログインまたはログアウトするには、[Cosminexus 7] ページで操作します。[Cosminexus 7] ページの表示方法を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。
[Cosminexus 7] ページが表示されます。

次に、[Cosminexus 7] ページでの Management Server リモート管理機能へのログインおよびログアウトの操作方法を説明します。

注意事項

Server Plug-in で Management Server リモート管理機能にログインしていても、MyEclipse の機能は使用できません。MyEclipse を使用する場合、Server Plug-in とは別に Management Server リモート管理機能にログインする必要があります。

参考

次の設定をデフォルトの値から変更したい場合は、Management Server リモート管理機能にログインする前に変更してください。

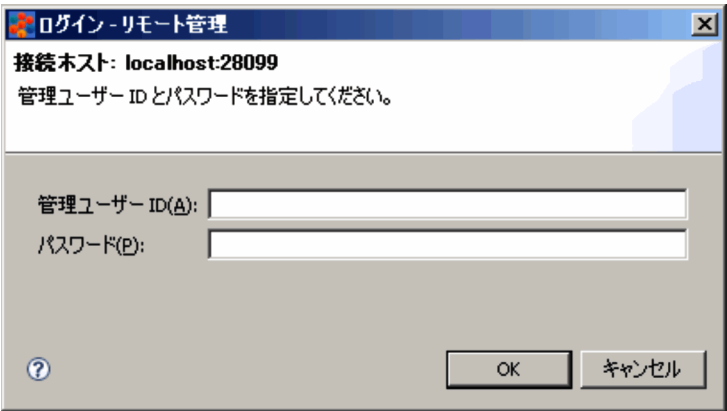
- デプロイ作業ディレクトリ
- ソケット操作のブロックのタイムアウト設定

変更手順については、「2.5.2 デプロイ作業ディレクトリの設定変更」または「2.5.3 ソケット操作のブロックのタイムアウト設定変更」を参照してください。

(1) Management Server リモート管理機能へのログイン (MyEclipse)

Management Server リモート管理機能へのログイン手順を次に示します。

1. [Cosminexus 7] ページの [Cosminexus サーバー] ツリービューから接続先ホストを選択します。
Management Server があるホストを選択します。MyEclipse とは別のマシンに Management Server がある場合は、「付録 B.2(4)(a) 接続先ホストの設定変更」で設定した接続先を選択してください。
2. [ログイン] ボタンをクリックします。
[ログイン - リモート管理] ダイアログが表示されます。



設定が必要な項目を次に示します。

項目名	設定値
管理ユーザー ID	Management Server にログインするときの管理ユーザ ID を指定します。 インスタントセットアップ機能で構築した場合は、「admin」を指定します。 インスタントセットアップ機能で構築していない場合は、Management Server 設定時に使用した値を指定します。

2. インストールとセットアップ

項目名	設定値
パスワード	Management Server にログインするときの管理ユーザパスワードを指定します。 インスタントセットアップ機能で構築した場合は、「admin」を指定します。インスタントセットアップ機能で構築していない場合は、Management Server 設定時に使用した値を指定します。

3. [OK] ボタンをクリックします。

Management Server リモート管理機能に接続します。[Cosminexus サーバー] ツリービューに Management Server の運用管理ドメインが表示されます。

(2) Management Server リモート管理機能からのログアウト (MyEclipse)

Management Server リモート管理機能からのログアウト手順を次に示します。

1. [Cosminexus 7] ページの [Cosminexus サーバー] ツリービューから接続済みのホストを選択します。

Management Server があるホストを選択します。MyEclipse とは別のマシンに Management Server がある場合は、「付録 B.2(4)(a) 接続先ホストの設定変更」で設定した接続先を選択してください。

2. [ログアウト] ボタンをクリックします。

[ログアウト - リモート管理] ダイアログが表示されます。

3. [はい] ボタンをクリックします。

Management Server リモート管理機能からログアウトします。

2.4.3 J2EE サーバの選択

MyEclipse から操作する J2EE サーバを選択します。ここで選択した J2EE サーバに対して、MyEclipse から起動および停止をしたり、J2EE アプリケーションをデプロイしたりします。

なお、J2EE サーバを選択するには、Management Server リモート管理機能にログインする必要があります。Management Server リモート管理機能にログインしていない場合は、「2.4.2(1) Management Server リモート管理機能へのログイン (MyEclipse)」に示す手順に従って、Management Server リモート管理機能にログインしてください。

設定手順を次に示します。

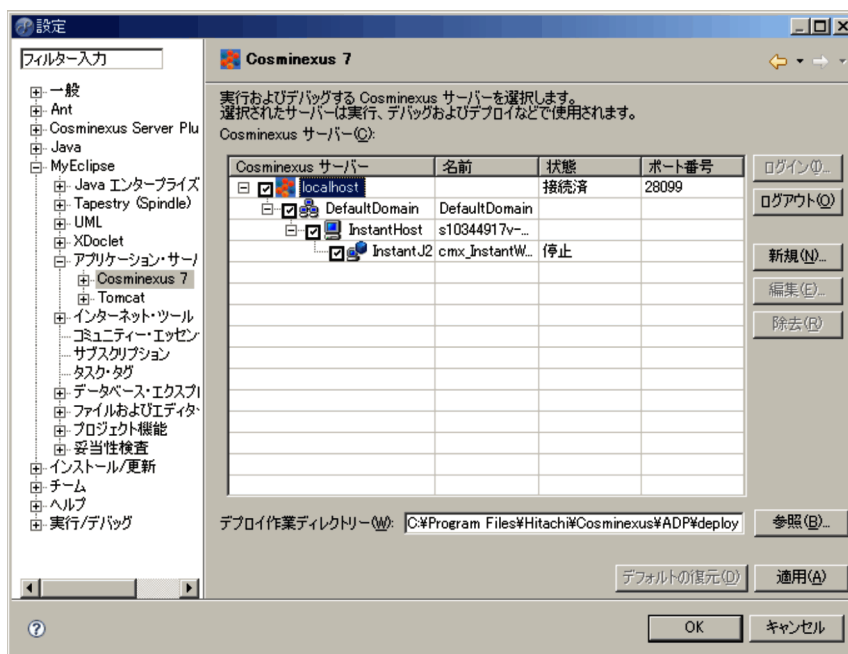
1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。

[設定] ダイアログが表示されます。

2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。

[Cosminexus 7] ページが表示されます。

3. [Cosminexus サーバー] ツリービューで、使用する J2EE サーバのチェックをオンにします。



4. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

2.4.4 Server Plug-in の設定

データベースなどのリソースと接続する J2EE アプリケーションの場合、リソースアダプタが必要になります。MyEclipse で J2EE アプリケーションを開発する場合、Server Plug-in は、リソースアダプタのプロパティを設定するときに使用します。

ここでは、Server Plug-in を使用するための設定について説明します。なお、Server Plug-in の概要については、マニュアル「Cosminexus アプリケーション設定操作ガイド」も参照してください。

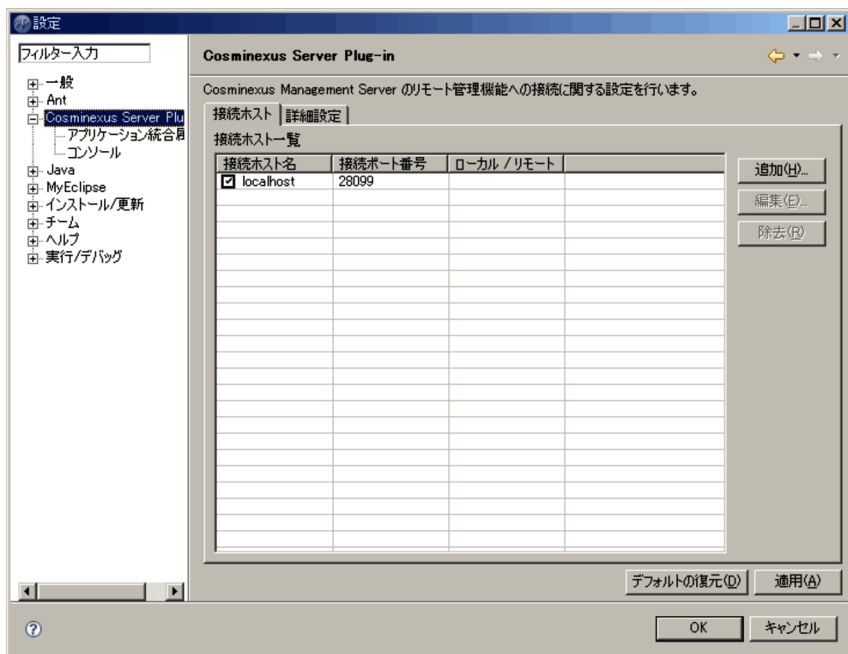
(1) 接続先ホストの設定

Server Plug-in で Management Server リモート管理機能に接続するために、接続先ホストを設定します。

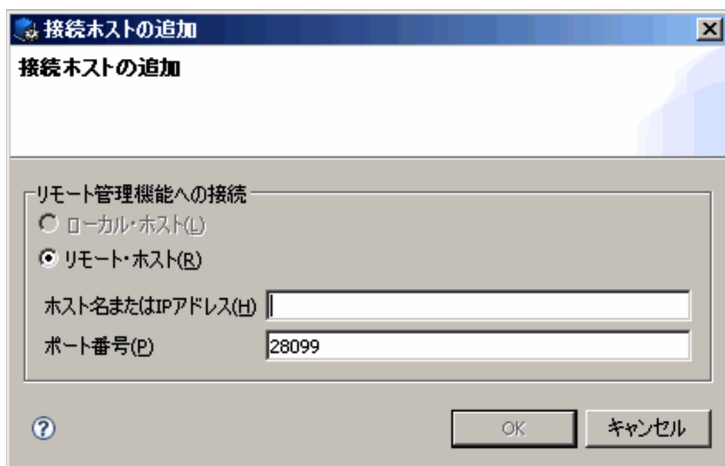
Server Plug-in の設定では、Management Server の接続先に関する設定をします。Server Plug-in の設定手順を次に示します。

2. インストールとセットアップ

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [Cosminexus Server Plug-in] を選択します。
[Cosminexus Server Plug-in] ページが表示されます。



3. [接続ホスト] タブで [追加] ボタンをクリックします。
[接続ホストの追加] ダイアログが表示されます。



設定が必要な項目を次に示します。

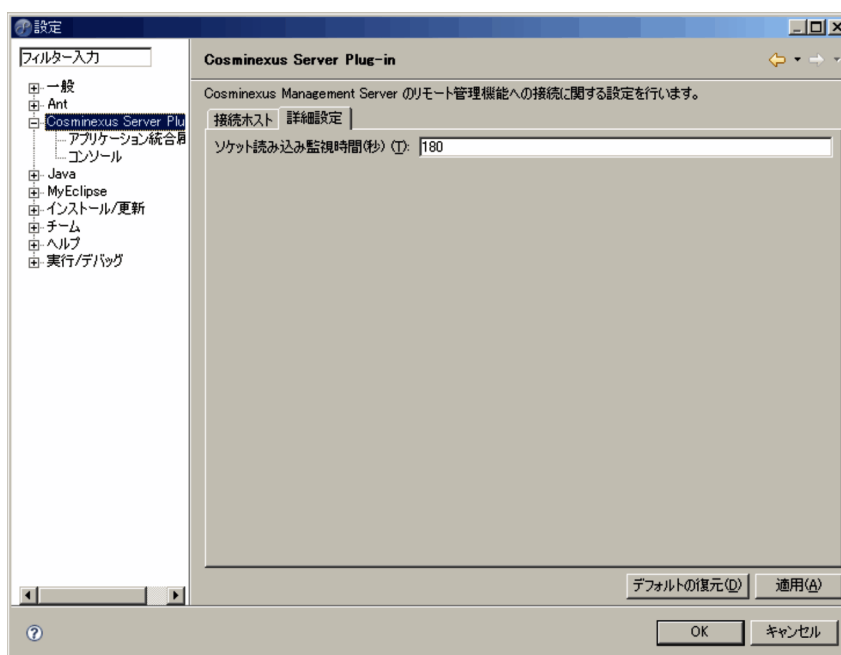
項目名	設定値
ローカル・ホスト / リモート・ホスト	Management Server の接続先がローカルホストかリモートホストかを選択します。インスタントセットアップ機能で構築した場合は、[ローカル・ホスト] を選択します。 <ul style="list-style-type: none"> ローカル・ホスト Management Server の接続先がローカルホストの場合に選択します。 リモート・ホスト Management Server の接続先がリモートホストの場合に選択します。
ホスト名または IP アドレス	接続先がリモートホストの場合は、接続先のホスト名または IP アドレスを指定します。
ポート番号	接続先のポート番号を指定します。 接続先ホストの mserver.properties の com.cosminexus.mngsvr.management.port に指定したポート番号を指定します。5 文字以内の半角数字で指定してください。

4. [OK] ボタンをクリックします。

接続先のホストが追加されます。

5. [詳細設定] タブをクリックします。

[詳細設定] タブが表示されます。



必要に応じて、次の項目を設定してください。

2. インストールとセットアップ

項目名	設定値
ソケット読み込み監視時間 (秒)	ソケット読み込み監視時間とは、Management Server リモート管理機能との接続で、ソケットの読み込みを監視する時間です。指定値の有効範囲は 0 または 180 ~ 99999 (秒) です。0 を指定した場合は、監視しません。

注

ソケット読み込み監視時間は、Management Server リモート管理機能にログインしているときだけ有効です。Management Server リモート管理機能にログインしているとき（接続中）に変更した監視時間は、すぐには有効になりません。次回、Management Server リモート管理機能にログインしてから有効になります。

また、ソケット読み込み監視時間を設定している場合で、Management Server リモート管理機能を使用した処理を実行しているときに、「Read time out」のメッセージが表示されることがあります。このメッセージが表示されたあとも、Management Server リモート管理機能の処理が続行されていることがあります。

6. [適用] ボタンまたは [OK] ボタンをクリックします。

設定が保存されます。

(2) Management Server リモート管理機能へのログインおよびログアウト (Server Plug-in)

Server Plug-in から、Management Server リモート管理機能にログインまたはログアウトするには、[Cosminexus Server Plug-in] パースペクティブの[サーバー・エクスプローラー]ビューで操作します。なお、[サーバー・エクスプローラー]ビューが表示されていない場合は、次の手順で表示してください。

1. Eclipse のメニューから [ウィンドウ] - [ビューの表示] - [その他] を選択します。
[ビューの表示] ダイアログが表示されます。
2. [Cosminexus Server Plug-in] - [サーバー・エクスプローラー] を選択して、[OK] ボタンをクリックします。
[サーバー・エクスプローラー] ビューが表示されます。



ログインおよびログアウト方法を次に示します。

! 注意事項

MyEclipse で Management Server リモート管理機能にログインしていても、Server Plug-in の機能は使用できません。Server Plug-in を使用する場合、MyEclipse とは別に Management Server リモート管理機能にログインする必要があります。

(a) Management Server リモート管理機能へのログイン (Server Plug-in)

Management Server リモート管理機能へのログイン手順を次に示します。


1. ツールバーの [ (Cosminexus Management Server リモート管理機能)] をクリックして表示されるサブメニューから [ ログイン] を選択します。
[Cosminexus Management Server リモート管理機能へログイン] ダイアログが表示されます。



設定が必要な項目を次に示します。



項目名	設定値
管理ユーザー ID	Management Server にログインするときの管理ユーザ ID を指定します。 インスタントセットアップ機能で構築した場合は、「admin」を指定します。インスタントセットアップ機能で構築していない場合は、Management Server 設定時に使用した値を指定します。
パスワード	Management Server にログインするときの管理ユーザパスワードを指定します。 インスタントセットアップ機能で構築した場合は、「admin」を指定します。インスタントセットアップ機能で構築していない場合は、Management Server 設定時に使用した値を指定します。

2. [OK] ボタンをクリックします。

Management Server リモート管理機能にログインします。ログインすると、ツリービューにある [ (Management Server 接続先ホストノード)] の状態表示が、「未接続」から「接続済」に変化します。


(b) Management Server リモート管理機能からのログアウト (Server Plug-in)

Management Server リモート管理機能からのログアウト手順を次に示します。

1. ツールバーの [ (Cosminexus Management Server リモート管理機能)] をクリックして表示されるサブメニューから [ ログアウト] を選択します。
[Cosminexus Management Server リモート管理機能からログアウト] ダイアログが表示されます。

2. インストールとセットアップ

2. [はい] ボタンをクリックします。

ツリービューにある [ (Management Server 接続先ホストノード)] の状態表示が、「接続済」から「未接続」に変化します。

2.5 開発環境の設定変更（任意の作業）

開発環境の設定は、必要に応じて変更できます。次に示す項目は、J2EE アプリケーション開発時に使用される設定です。これらの設定は、デフォルト値が設定されていますが、必要に応じて設定を変更できます。

- MyEclipse のプロキシの設定
- デプロイ作業ディレクトリの設定
- ソケット操作のブロックのタイムアウト設定
- コンソールの設定
- アプリケーション統合属性ファイルのエディタの設定

注

インターネット接続時にプロキシを使用している場合は、使用している Internet Explorer のプロキシの設定に必ず合わせてください。

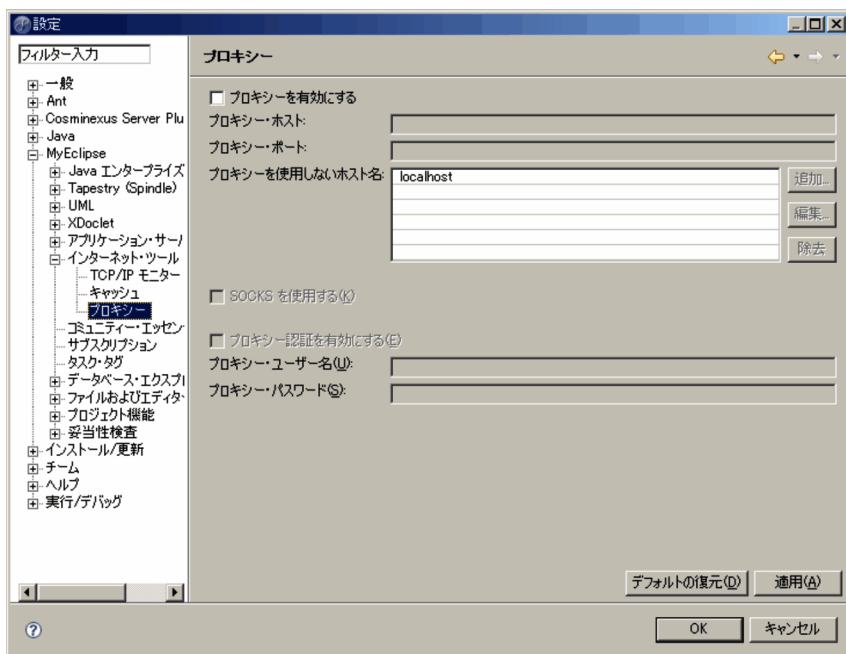
ここでは、これらの設定を変更する手順を説明します。

2.5.1 MyEclipse のプロキシの設定

インターネット接続時にプロキシを使用している環境では、MyEclipse でプロキシを設定します。プロキシの設定は、使用している Internet Explorer のプロキシの設定に合わせてください。プロキシの設定手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [インターネット・ツール] - [プロキシ] を選択します。
[プロキシ] ページが表示されます。

2. インストールとセットアップ



設定が必要な項目を次に示します。

項目名	設定値
プロキシを有効にする	<p>プロキシを有効にするかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする プロキシを有効にします。 • チェックしない プロキシを無効にします。 <p>プロキシを有効にした場合は、[プロキシ・ホスト] および [プロキシ・ポート] も指定してください。</p>
プロキシ・ホスト	Internet Explorer で設定しているプロキシのホストを指定します。
プロキシ・ポート	Internet Explorer で設定しているプロキシのポート番号を指定します。

必要に応じて、次の項目を設定してください。

項目名	設定値
プロキシを使用しないホスト名	必要に応じて、プロキシを利用しないホストを追加します。
SOCKS を使用する	Internet Explorer で SOCKS を使用している場合は、チェックしてください。

項目名	設定値
プロキシー認証を有効にする	<p>プロキシの認証を有効にするかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする プロキシの認証を有効にします。ネットワーク接続時の認証ダイアログが省略されます。 • チェックしない プロキシの認証を無効にします。ネットワーク接続時の認証ダイアログが表示されます。 <p>プロキシの認証を有効にした場合は、[プロキシー・ユーザー名] および [プロキシー・パスワード] も指定してください。</p>
プロキシー・ユーザー名	プロキシの認証を有効にする場合は、プロキシのユーザー名を指定します。
プロキシー・パスワード	プロキシの認証を有効にする場合は、プロキシのパスワードを指定します。

参考

プロキシを有効にして、ネットワークに接続すると認証ダイアログが表示されます。あらかじめ、[プロキシー] ページで次の設定をすると、認証ダイアログは表示されません。

- [プロキシー認証を有効にする] をチェックする。
- [プロキシー・ユーザー名] と [プロキシー・パスワード] を指定する。

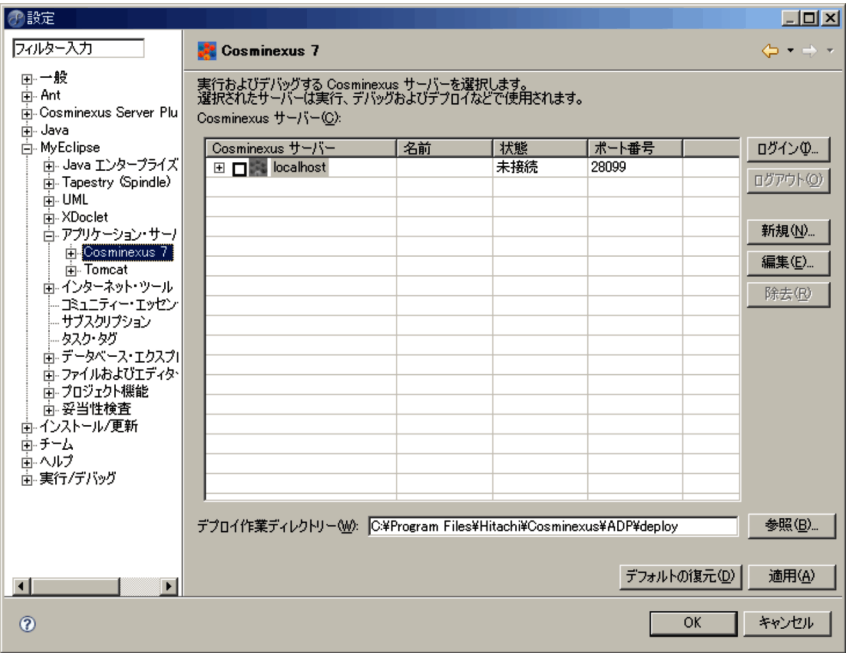
3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

2.5.2 デプロイ作業ディレクトリの設定変更

必要に応じて、デプロイ作業ディレクトリの設定を変更します。デプロイ作業ディレクトリとは、Cosminexus コネクタが MyEclipse から J2EE アプリケーションを受け取るための開発環境用のデプロイメントエリアです。デプロイ作業ディレクトリの設定手順を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。
[Cosminexus 7] ページが表示されます。

2. インストールとセットアップ



必要に応じて、次の項目の設定を変更します。

項目名	設定値
デプロイ作業ディレクトリ	MyEclipse から J2EE サーバに J2EE アプリケーションをデプロイするときに使用する作業用のディレクトリを指定します。 [参照] ボタンをクリックして表示される [フォルダの参照] ダイアログからも指定できます。

Windows Vista の場合

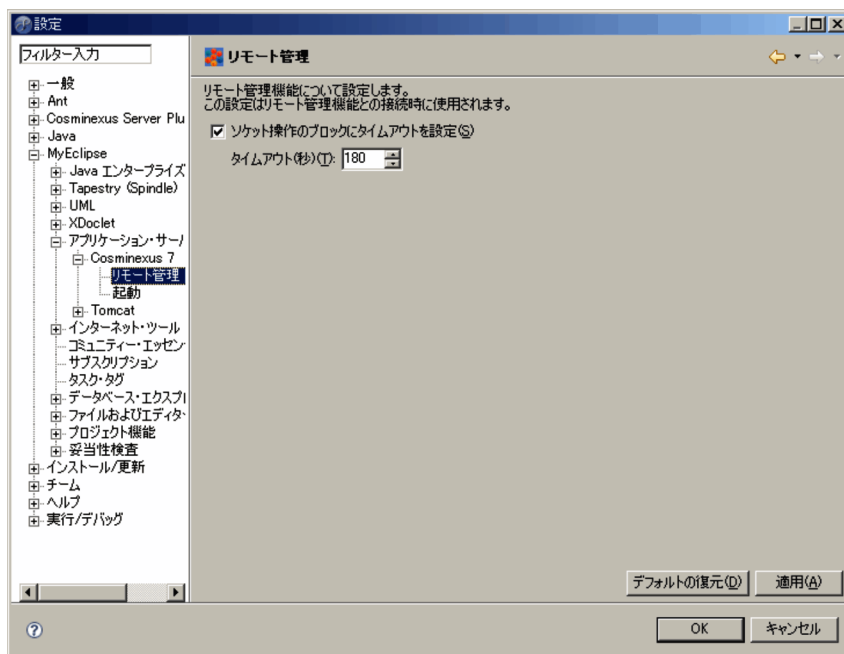
デプロイ作業ディレクトリに C:\Program Files 以下のディレクトリを指定している場合で、管理者特権のないユーザが MyEclipse で J2EE アプリケーションをデプロイしようとする、エラーダイアログが表示されます。このため、デプロイ作業ディレクトリには、C:\Program Files 以下のディレクトリを指定しないでください。

3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

2.5.3 ソケット操作のブロックのタイムアウト設定変更

必要に応じて、[リモート管理] ページで、Management Server リモート管理機能と接続するときに使用するソケット操作のブロックのタイムアウト設定を変更します。
Management Server リモート管理機能にログインしている場合は設定できません。
Management Server リモート管理機能からログアウトして、設定してください。設定手順を次に示します。

1. Eclipse のメニューから , [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] -
[Cosminexus 7] - [リモート管理] を選択します。
[リモート管理] ページが表示されます。



必要に応じて、次の項目の設定を変更します。

項目名	設定値
ソケット操作のブロックにタイムアウトを設定	ソケット操作のブロックにタイムアウトを設定するかどうかを指定します。 <ul style="list-style-type: none"> • チェックする タイムアウトを設定します。 • チェックしない タイムアウトを設定しません。
タイムアウト (秒)	ソケット操作のブロックでタイムアウトする時間を秒で設定します。

注

ソケット操作のブロックにタイムアウトを設定している場合で、Management Server リモート管理機能を使用した処理を実行しているときに、「Read time out」のメッセージが表示されることがあります。このメッセージが表示されたあとも、Management Server リモート管理機能の処理が続行されていることがあります。

3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

2. インストールとセットアップ

注

タイムアウトの値を変更した場合も、[適用] ボタンをクリックして値を保存してください。[適用] ボタンをクリックしないで Management Server リモート管理機能にログインすると、変更した値は破棄され、変更前の値で動作します。

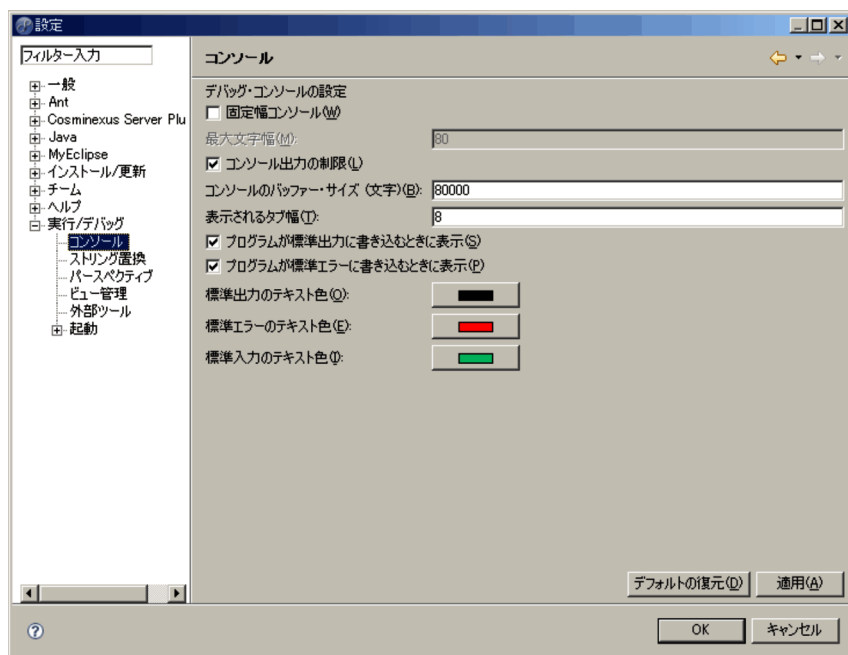
2.5.4 コンソールの設定変更

必要に応じて、コンソールの出力形式の設定を変更します。Eclipse および Server Plug-in のコンソールの設定を変更する手順を次に示します。

(1) Eclipse のコンソールの設定

J2EE サーバのプロセスの標準出力や標準エラー出力は、Eclipse が提供するコンソールに表示されます。Eclipse のコンソールの設定を変更する手順を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [実行 / デバッグ] - [コンソール] を選択します。
[コンソール] ページが表示されます。



必要に応じて、次の項目の設定を変更してください。

項目名	設定値
固定幅コンソール	<p>コンソールビューで表示する 1 行の文字数を固定するかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする 1 行の文字数を固定します。 • チェックしない 1 行の文字数を固定しません。 <p>1 行の文字数を固定する場合、[最大文字幅] も指定します。 なお、1 行の文字数が [最大文字幅] に指定した数を超える場合は、改行して表示されます。</p>
最大文字幅	<p>コンソールビューに表示する 1 行の文字数を指定します。80 ~ 1000 の範囲で指定します。 [固定幅コンソール] をチェックしている場合は必ず入力します。</p>
コンソール出力の制限	<p>コンソールの出力を制限するかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする コンソールの出力を制限します。 • チェックしない コンソールの出力を制限しません。 <p>なお、コンソールの出力を制限している場合に、出力された行までの文字数が [コンソールのバッファサイズ (文字)] で指定されているバッファサイズを超えると、先頭から超えた分の出力行が消去されます。消去されたメッセージは次のファイルで確認できます。 <Eclipse のワークスペースディレクトリ>¥.metadata¥.log</p>
コンソールのバッファサイズ (文字)	<p>コンソールのバッファサイズ (文字数) を、1000 ~ 1000000 の範囲で指定します。必要に応じて調節してください。 [コンソール出力の制限] をチェックしている場合に必ず入力します。</p>
表示されるタブ幅	<p>コンソールビューで表示するタブ幅を文字数で指定します。1 ~ 100 の範囲で指定します。</p>
プログラムが標準出力に書き込むときに表示	<p>J2EE サーバや J2EE アプリケーションが標準出力にメッセージを出力したときに、コンソールビューを前面に表示するかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする コンソールビューを前面に表示します。 • チェックしない コンソールビューを前面に表示しません。
プログラムが標準エラーに書き込むときに表示	<p>J2EE サーバや J2EE アプリケーションが標準エラー出力にメッセージを出力したときに、コンソールビューを前面に表示するかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする コンソールビューを前面に表示します。 • チェックしない コンソールビューを前面に表示しません。
標準出力のテキスト色	<p>標準出力のテキストの色を指定します。</p>
標準エラーのテキスト色	<p>標準エラー出力のテキストの色を指定します。</p>
標準入力 of テキスト色	<p>標準入力 of テキストの色を指定します。</p>

2. インストールとセットアップ

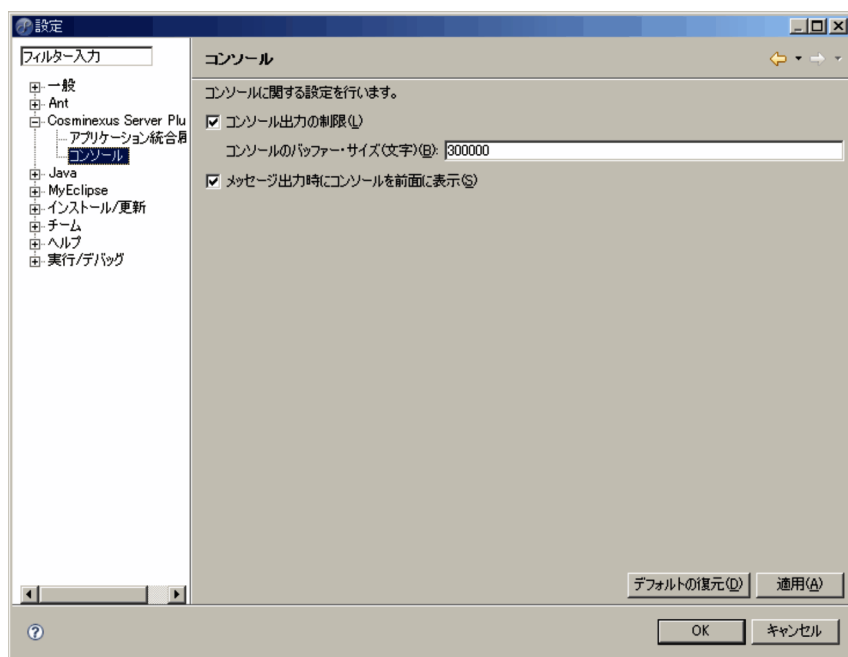
なお、デフォルト値を復元させる場合は、[デフォルトの復元] ボタンをクリックしてください。

3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

(2) Server Plug-in のコンソールの設定

Server Plug-in では、複数のコンソールを使用します。各種コンソールの設定は、Server Plug-in から変更できます。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [Cosminexus Server Plug-in] - [コンソール] を選択します。
[コンソール] ページが表示されます。



必要に応じて、次の項目の設定を変更してください。

項目名	設定値
コンソール出力の制限	<p>コンソールの出力を制限するかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする コンソールの出力を制限します。 • チェックしない コンソールの出力を制限しません。 <p>なお、コンソールの出力を制限している場合、出力された行までの文字数が [コンソールのバッファ・サイズ (文字)] で指定されているバッファサイズを超えると、先頭から超えた分の出力行が消去されます。消去されたメッセージは次のファイルで確認できます。 <Eclipse のワークスペースディレクトリ>¥.metadata¥.log</p>
コンソールのバッファ・サイズ (文字)	<p>一つのコンソールのバッファサイズ (文字数) を、20000 ~ 1000000 の半角数字で指定できます。必要に応じて調節してください。 ただし、[コンソール出力の制限] チェックボックスをチェックしている場合は、必ず入力してください。</p>
メッセージ出力時にコンソールを前面に表示	<p>メッセージ出力時に出力対象のコンソールを前面に表示するかどうかを指定します。</p> <ul style="list-style-type: none"> • チェックする コンソールを前面に表示します。 • チェックしない コンソールを前面に表示しません。 <p>ただし、コンソールに初めてメッセージが出力された場合は、そのコンソールは前面表示されます。</p>

なお、デフォルト値を復元させる場合は、[デフォルトの復元] ボタンをクリックしてください。

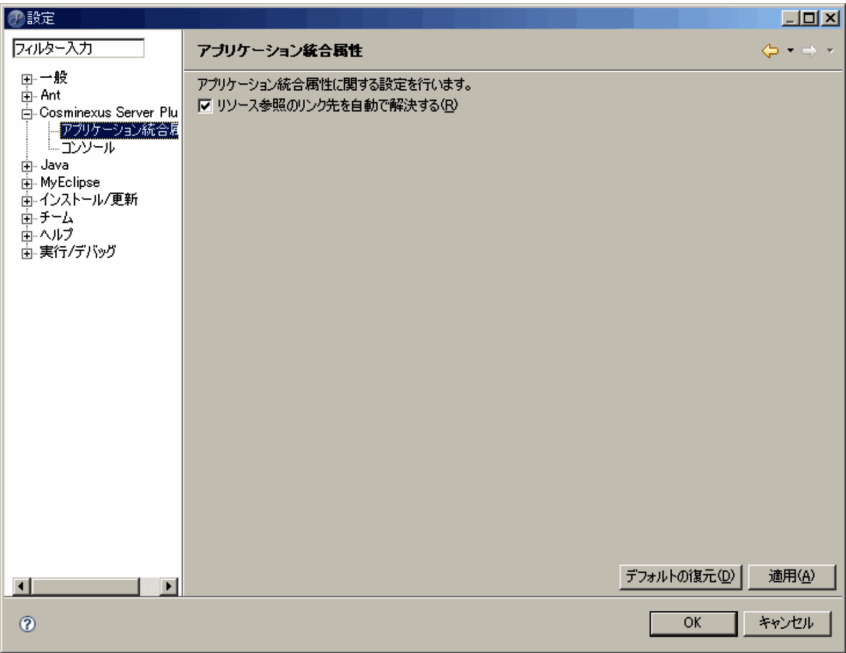
3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

2.5.5 アプリケーション統合属性ファイルのエディタの設定変更

必要に応じて、Server Plug-in のアプリケーション統合属性ファイルの編集に使用する属性ファイル編集エディタの設定を変更します。属性ファイル編集エディタについては、マニュアル「Cosminexus アプリケーション設定操作ガイド」の属性ファイル編集エディタに関する説明を参照してください。設定変更の手順を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [Cosminexus Server Plug-in] - [アプリケーション統合属性] を選択します。
[アプリケーション統合属性] ページが表示されます。

2. インストールとセットアップ



必要に応じて、次の項目の設定を変更します。

項目名	設定値
リソース参照のリンク先を自動で解決する	アプリケーション統合属性ファイルの属性編集エディタを起動したときに、J2EE アプリケーションのリソース参照を自動で解決するかどうかを指定します。 <ul style="list-style-type: none">• チェックする リソース参照を自動で解決します。• チェックしない リソース参照を自動で解決しません。

なお、リソース参照の自動解決については、マニュアル「Cosminexus アプリケーション設定操作ガイド」のリソース参照の自動解決に関する説明を参照してください。

3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

2.6 インスタントセットアップ機能で構築した J2EE サーバの設定変更

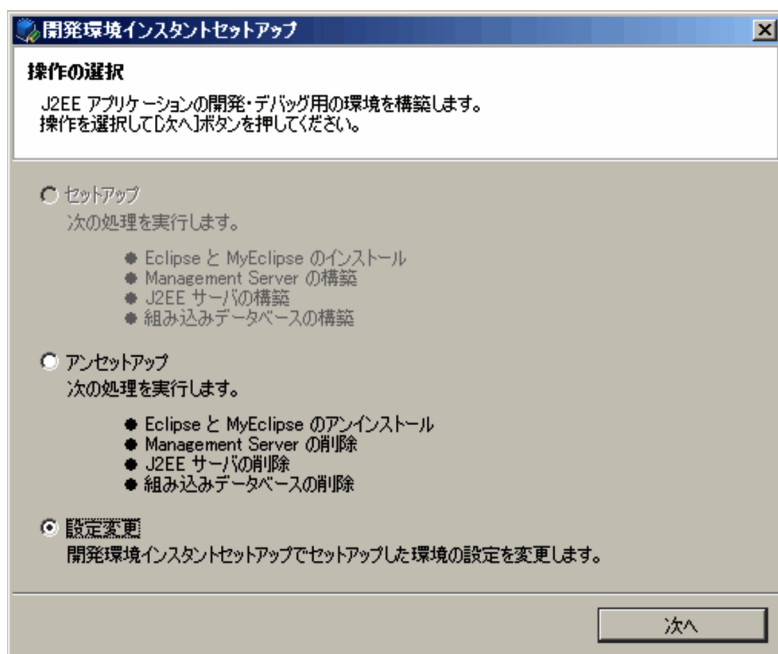
インスタントセットアップ機能で構築した環境は、インスタントセットアップ機能を使用して J2EE サーバの設定内容を変更できます。

ここでは、インスタントセットアップ機能で J2EE サーバの設定内容を変更する手順を次に説明します。

！ 注意事項

インスタントセットアップ機能の実行中に、[Windows セキュリティの重要な警告] ダイアログが表示された場合の対処方法については、「2.3.3 標準セットアップ」の注意事項を参照してください。

1. スタートメニューから、[プログラム] - [Cosminexus] - [Application Development Plug-in] - [開発環境インスタントセットアップ] を選択します。
インスタントセットアップ機能が起動して、[開発環境インスタントセットアップ] ダイアログの [操作の選択] ページが表示されます。



参考

起動オプションを使用して、インスタントセットアップ機能を起動したい場合は、「2.3.3 標準セットアップ」の参考事項を参照してください。

2. インストールとセットアップ

2. [設定変更] を選択して, [次へ] ボタンをクリックします。

[設定変更] ページが表示されます。

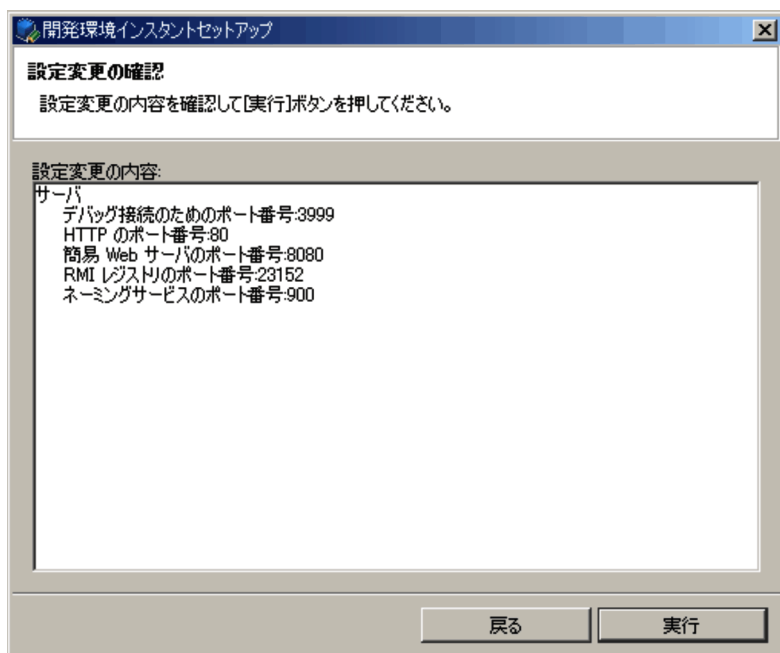
3. 必要に応じて, J2EE サーバの設定を変更します。

[] および [] ボタン, [] および [] キー, または直接数値を入力して設定してください。

項目名	設定値
デバッグ接続のためのポート番号	J2EE サーバにデバッグ接続するために使用するポート番号を半角数字で設定します。 設定できる範囲は 0 ~ 65535 です。
HTTP のポート番号	J2EE サーバが利用する HTTP のポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。
簡易 Web サーバのポート番号	J2EE サーバが利用する簡易 Web サーバのポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。
RMI レジストリのポート番号	J2EE サーバが利用する RMI レジストリのポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。
ネーミングサービスのポート番号	J2EE サーバがネーミングサービスとして利用する CORBA ネーミングサービスのポート番号を半角数字で設定します。 設定できる範囲は 1 ~ 65535 です。

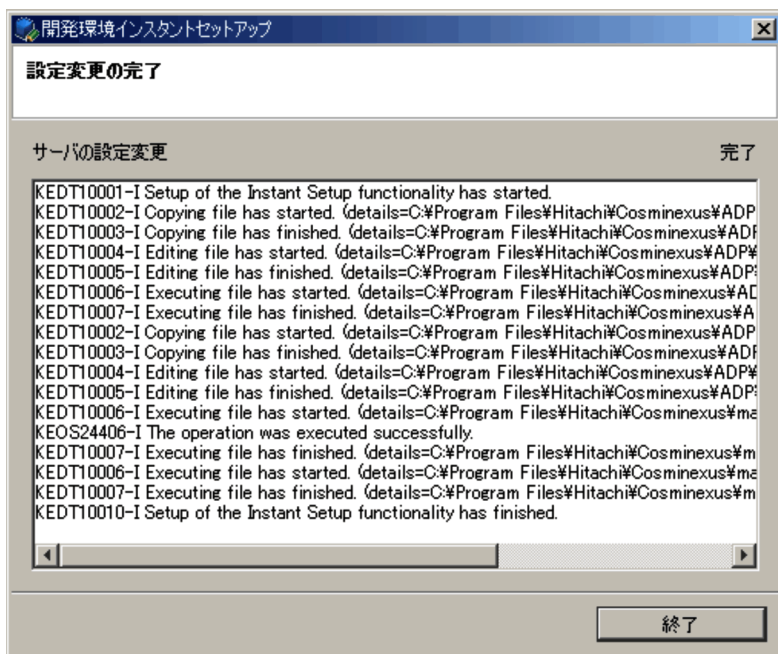
4. [次へ] ボタンをクリックします。

[設定変更の確認] ページが表示されます。



5. [設定変更の内容] エリアに表示された設定値を確認します。
6. [実行] ボタンをクリックします。
[進行状況] ページが表示されます。
設定変更が終了すると , [設定変更の完了] ページが表示されます。

2. インストールとセットアップ



7. [終了] ボタンをクリックします。

[開発環境インスタントセットアップ] ダイアログが閉じます。

参考

インスタントセットアップ機能以外から環境を変更した場合

インスタントセットアップ機能で構築した J2EE サーバの設定は、インスタントセットアップ機能以外を使用しても変更できます。

ただし、インスタントセットアップ機能以外で設定変更した J2EE サーバを再度インスタントセットアップ機能で設定を変更しようとする、エラーが発生する場合があります。

エラーが発生した場合の対処

初めてエラーが発生した場合は、「2.3.3 標準セットアップ」の参考事項にあるエラーの対処方法のとおりに対処したあと、再度設定変更を実行してください。

2 回以上エラーが発生した場合は、アンセットアップを実行して、環境を構築し直してください。

設定変更後の各サーバの状態

インスタントセットアップ機能を使用して設定変更すると、各サーバは次の状態になります。

表 2-6 設定変更後の各サーバの状態

サーバ名	状態
運用管理エージェント	

サーバ名	状態
Management Server	
J2EE サーバ	×
パフォーマンスストレサ	
組み込みデータベース	

(凡例)

：開始 ×：停止 ：設定変更前と同じ状態

注

J2EE サーバが開始されたときに，変更した内容が有効になります。

2.7 インスタントセットアップ機能で構築した環境のアンセットアップ

インスタントセットアップ機能で構築した開発環境およびテスト用実行環境をアンセットアップしたい場合は、必ずインスタントセットアップ機能のアンセットアップを実行してください。

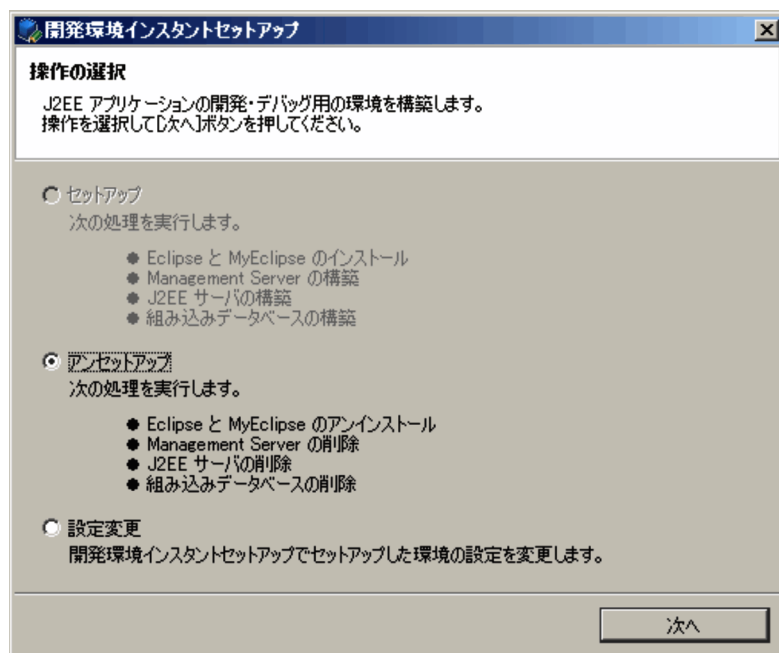
アンセットアップを実行すると、インスタントセットアップ機能で構築した環境がすべて削除され、Cosminexus を新規でインストールした直後の状態になります。Eclipse のインストールディレクトリ、MyEclipse のインストールディレクトリ、および組み込みデータベース構築ディレクトリの下に手動で追加したファイルやディレクトリも削除されます。Eclipse のワークスペースやプロジェクトなどのデータは、これらのディレクトリ以外のディレクトリに作成してください。

インスタントセットアップ機能でアンセットアップする手順を次に説明します。

注意事項

インスタントセットアップ機能の実行中に、[Windows セキュリティの重要な警告] ダイアログが表示された場合の対処方法については、「2.3.3 標準セットアップ」の注意事項を参照してください。

1. スタートメニューから、[プログラム] - [Cosminexus] - [Application Development Plug-in] - [開発環境インスタントセットアップ] を選択します。
インスタントセットアップ機能が起動して、[開発環境インスタントセットアップ] ダイアログの [操作の選択] ページが表示されます。

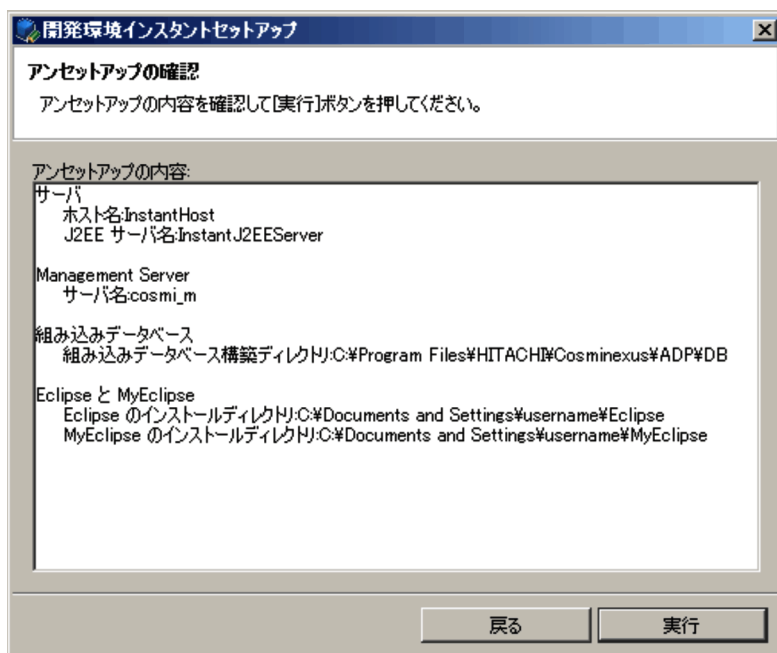


参考

起動オプションを使用して、インスタントセットアップ機能を起動したい場合は、「2.3.3 標準セットアップ」の参考事項を参照してください。

2. [アンセットアップ]を選択して,[次へ]ボタンをクリックします。
[アンセットアップの確認]ページが表示されます。

2. インストールとセットアップ

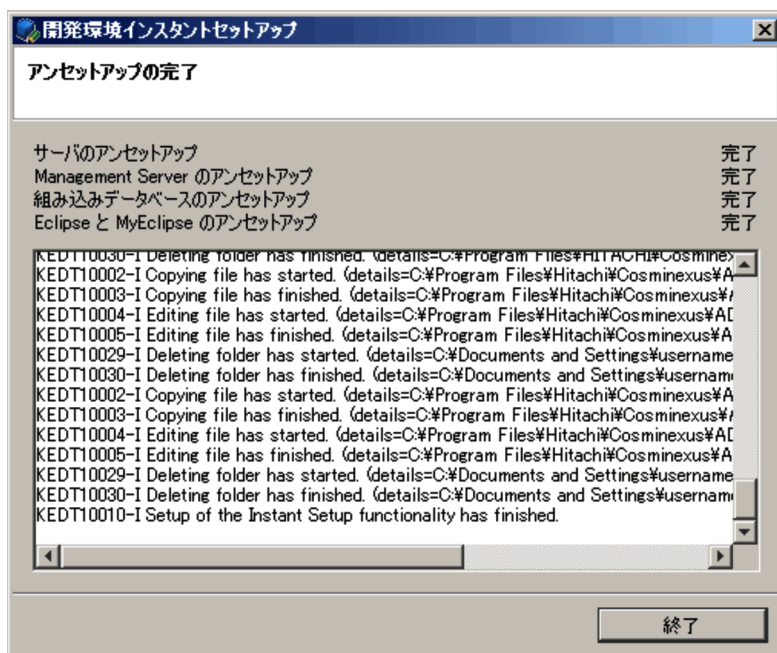


3. [アンセットアップの内容] エリアに表示された内容を確認します。

4. [実行] ボタンをクリックします。

[進行状況] ページが表示されます。

アンセットアップが終了すると,[アンセットアップの完了] ページが表示されます。



5. [終了] ボタンをクリックします。
 [開発環境インスタントセットアップ] ダイアログが閉じます。

参考

エラーが発生した場合の対処

エラーが発生した場合は、コンソールに出力されたエラーを参照して、手動でアンセットアップしてください。なお、各サーバが構築されていない場合も、コンソールにエラーメッセージが出力されます。この場合は、手動でアンセットアップする必要はありません。また、アンセットアップ中にプロセスを落としたり、PC の電源が落ちたりした場合は、アンセットアップが中断されます。この場合は、アンセットアップを再度実行してください。

アンセットアップ後の各サーバの状態

インスタントセットアップ機能を使用してアンセットアップすると、各サーバは次の状態になります。

表 2-7 アンセットアップ後の各サーバの状態

サーバ名	状態
運用管理エージェント	×
Management Server	-
J2EE サーバ	-
パフォーマンストレーサ	-
組み込みデータベース	-

(凡例)

× : 停止 - : 削除されたため、該当しない。

手動でアンセットアップする手順

インスタントセットアップ機能で構築した環境をアンセットアップする前に、Cosminexus をアンインストールおよび再インストールした場合は、次の手順で構築した環境を手動でアンセットアップしてください。

なお、インスタントセットアップ機能のアンセットアップに失敗した場合は、アンセットアップに失敗した環境だけ手動でアンセットアップしてください。

1. J2EE サーバおよびパフォーマンストレーサをアンセットアップします。

次に示す Smart Composer のコマンドおよびサーバ管理コマンド (CUI) を実行して、J2EE サーバおよびパフォーマンストレーサをアンセットアップください。なお、Smart Composer のコマンドの詳細は「Cosminexus 簡易構築・運用ガイド」を、サーバ管理コマンド (CUI) の詳細は「Cosminexus リファレンス コマンド編」を参照してください。

2. インストールとセットアップ

```
<Cosminexusのインストールディレクトリ>%manager%bin%cmx_stop_target -m  
localhost:28080 -u admin -p admin -mode ALL -s InstantWebSystem
```

```
<Cosminexusのインストールディレクトリ>%CC%server%bin%cjsetup -d  
cmx_InstantWebSystem_unit1_J2EE_01
```

```
<Cosminexusのインストールディレクトリ>%manager%bin%cmx_delete_system -m  
localhost:28080 -u admin -p admin -s InstantWebSystem
```

2. Management Server をアンセットアップします。

次の手順で Management Server をアンセットアップしてください。

1. 次のファイルをコピーします。

コピー元のファイル

- ・ <Cosminexus のインストールディレクトリ>
>%manager%config%templates%adminagent.properties
- ・ <Cosminexus のインストールディレクトリ>
>%manager%config%templates%msserver.properties
- ・ <Cosminexus のインストールディレクトリ>
>%manager%config%templates%msserver.xml

コピー先のディレクトリ

<Cosminexus のインストールディレクトリ> %manager%config

2. 次のコマンドを実行します。

```
<Cosminexusのインストールディレクトリ>%CC%web%bin%cjwebsetup -d cosmi_m
```

3. 組み込みデータベースをアンセットアップします。

組み込みデータベースのアンセットアップについては、「付録 B.3(3) 組み込みデータベースの削除」を参照してください。また、次に示すディレクトリを手動で削除してください。

- ・ <組み込みデータベース構築ディレクトリ>%area
- ・ <組み込みデータベース構築ディレクトリ>%bats
- ・ <組み込みデータベース構築ディレクトリ>%conf
- ・ <組み込みデータベース構築ディレクトリ>%ini

4. Eclipse および MyEclipse をアンセットアップします。

次に示すディレクトリを手動で削除してください。

- ・ <Eclipse のインストールディレクトリ>%eclipse
- ・ <MyEclipse のインストールディレクトリ>%eclipse

2.8 Developer のアンインストール

Developer をアンインストールします。アンインストールには、Administrator 権限または管理者特権が必要です。なお、Developer のアンインストール方法については、リリースノートを参照してください。

3

テスト用実行環境で使用するデータベースのテーブルの作成

インスタントセットアップ機能で組み込みデータベースをセットアップすると、サンプルプログラム（Bank）で使用するテーブルが作成されます。これ以外のテーブルを使用したい場合は、組み込みデータベースを操作して、新しくテーブルを作成する必要があります。この章では、HiRDB SQL Executerを使用したテスト用実行環境で使用するデータベースのテーブルの作成について説明します。

3.1 組み込みデータベースのテーブルの作成の流れ

3.2 HiRDB SQL Executer のインストール

3.3 組み込みデータベースの操作

3.4 RD エリアの追加と削除

3.5 組み込みデータベースの動作制御の設定

3.1 組み込みデータベースのテーブルの作成の流れ

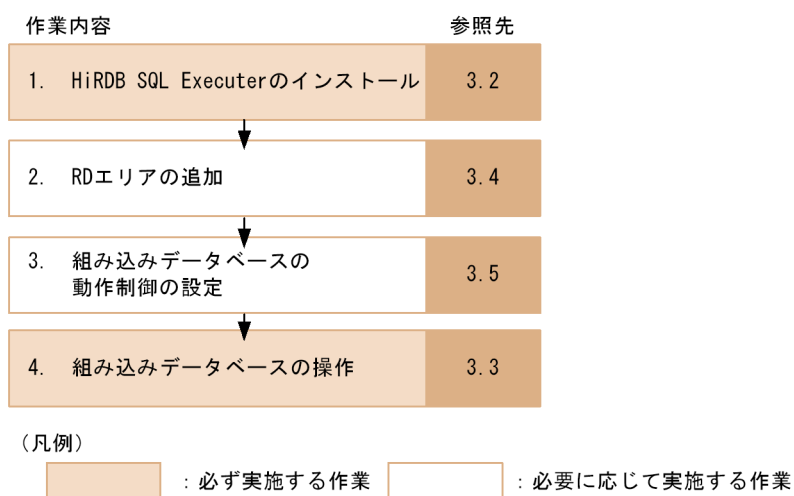
Developer では、組み込みデータベースを提供しています。組み込みデータベースは、開発した J2EE アプリケーションのテストおよびデバッグで使します。

組み込みデータベースは、Developer が提供するインスタントセットアップ機能によって構築されます。また、構築された組み込みデータベースには、サンプルプログラム (Bank) で使用するテーブルも作成されます。

インスタントセットアップ機能が構築したテーブルを使用しない場合は、HiRDB SQL Executer で組み込みデータベースのテーブルを作成する必要があります。

組み込みデータベースのテーブルを作成する流れを次の図に示します。

図 3-1 組み込みデータベースのテーブルを作成する流れ



それぞれの作業の概要を説明します。

1. HiRDB SQL Executer のインストール

組み込みデータベースを操作するために、HiRDB SQL Executer をインストールします。

2. RD エリアの追加

必要に応じて、データベースに RD エリアを追加します。

3. 組み込みデータベースの動作制御の設定

必要に応じて、組み込みデータベースの動作制御について設定します。

4. 組み込みデータベースの操作

HiRDB SQL Executer を使用して、構築した組み込みデータベースを操作して、テーブルの作成または参照をします。

以降の節では、この流れに沿って組み込みデータベースの設定の手順を説明します。

なお、インスタントセットアップ機能で構築した組み込みデータベースは、インスタントセットアップ機能からアンセットアップできます。

3.2 HiRDB SQL Executer のインストール

組み込みデータベースの操作には、HiRDB SQL Executer を使用します。

HiRDB SQL Executer をインストールするには、次のフォルダ下にある Setup.exe を実行してください。なお、異なるバージョンおよびリビジョンの HiRDB SQL Executer がインストールされている場合は、事前にアンインストールしてから、インストールしてください。

<Cosminexusのインストールディレクトリ>¥DB¥utility¥Executer

これで、HiRDB SQL Executer のインストールは完了です。

なお、HiRDB SQL Executer をインストールすると、GUI 版 HiRDB SQL Executer および簡易 GUI 版 HiRDB SQL Executer がインストールされます。Windows Vista 以外の OS で GUI 版 HiRDB SQL Executer を使用するためには、あらかじめ「Microsoft .NET Framework」の 2.0 以上および「Microsoft .NET Framework 日本語 Language Pack」の 2.0 以上をインストールしておく必要があります。

3.3 組み込みデータベースの操作

ここでは、組み込みデータベースのテーブルを作成または参照する方法について説明します。なお、HiRDB SQL Executer を使用して組み込みデータベースを操作する場合は、次の条件を満たしている必要があります。インスタントセットアップ機能で構築した環境は次の条件を満たしているので、HiRDB SQL Executer で操作できます。

- 組み込みデータベースが構築されていること。
- ユーザ定義が完了していること。
- 組み込みデータベースに RD エリアが追加されていること。
- データを参照または更新する場合は、参照または更新の対象となるテーブルが組み込みデータベースに作成されていること。

なお、組み込みデータベースを操作するには、組み込みデータベースに接続する必要があります。

ここでは、組み込みデータベースの操作方法について説明します。

3.3.1 組み込みデータベースへの接続

HiRDB SQL Executer を使用した組み込みデータベースへの接続方法を示します。なお、ここでは、GUI 版 HiRDB SQL Executer を使用した手順を説明します。

1. Windows の [スタート] メニューの [プログラム] - [HiRDB SQL Executer] - [GUI 版 HiRDB SQL Executer] を選択します。
[CONNECT] ダイアログが表示されます。



設定が必要な項目を次に示します。

3. テスト用実行環境で使用するデータベースのテーブルの作成

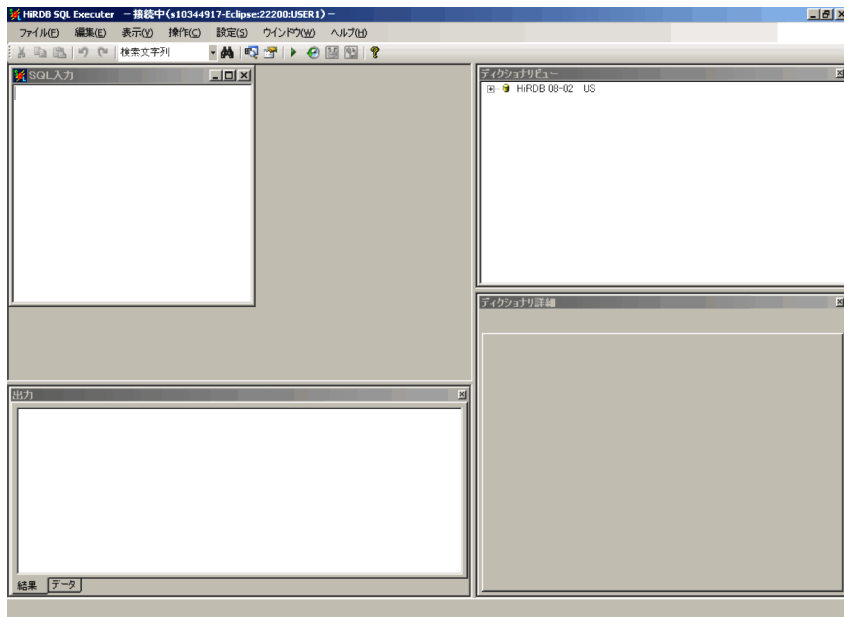
項目名	設定値
認可識別子	セットアップ時に設定されたテーブル所有者の認可識別子を指定します。 インスタントセットアップ機能では「USER1」を設定しています。 なお、デフォルトのユーザ ID とパスワードは、「root」/「root」です。
パスワード	セットアップ時に設定されたテーブル所有者のパスワードを指定します。 インスタントセットアップ機能では「PSWD」を設定しています。
ホスト名	セットアップ時に構築された組み込みデータベースのホストの名称を指定します。 インスタントセットアップ機能では「localhost」を設定しています。
ポート番号	セットアップ時に設定された DB_PORT パラメタの指定値を入力します。 インスタントセットアップ機能では「22200」を設定しています。

必要に応じて、次の項目を設定してください。

項目名	設定値
認可識別子 / パスワードを記憶する	認可識別子およびパスワードを記憶するかどうかを指定します。 <ul style="list-style-type: none">• チェックする 認可識別子およびパスワードを記憶します。• チェックしない 認可識別子およびパスワードを記憶しません。

2. [CONNECT] ダイアログの [OK] ボタンをクリックします。

組み込みデータベースに接続します。組み込みデータベースへの接続が成功すると、GUI 版 HiRDB SQL Executer のタイトルバーに [接続中] と表示されます。



3.3.2 テーブルの作成

ここでは、GUI 版 HiRDB SQL Executer を使用してテーブルを作成する手順を説明します。なお、テーブルを作成する前に、「3.3.1 組み込みデータベースへの接続」の手順に従って、GUI 版 HiRDB SQL Executer に接続してください。

1. GUI 版 HiRDB SQL Executer の [SQL 入力] ウィンドウに SQL 入力して、スキーマの定義およびテーブルの作成をします。
スキーマの定義例を次に示します。

```
CREATE SCHEMA;
```


テーブルの作成例を次に示します。

```
create table checking
(id varchar(3) primary key,
balance double precision);

insert into checking
values ('001', 10000.00);

create table saving
(id varchar(3) primary key,
balance double precision);

insert into saving
values ('001', 500.00);
```

2. GUI 版 HiRDB SQL Executer のツールバーの [ (SQL 実行)] をクリックします。

3. テスト用実行環境で使用するデータベースのテーブルの作成

SQL が実行されて、テーブルが作成されます。

3.3.3 テーブルの参照

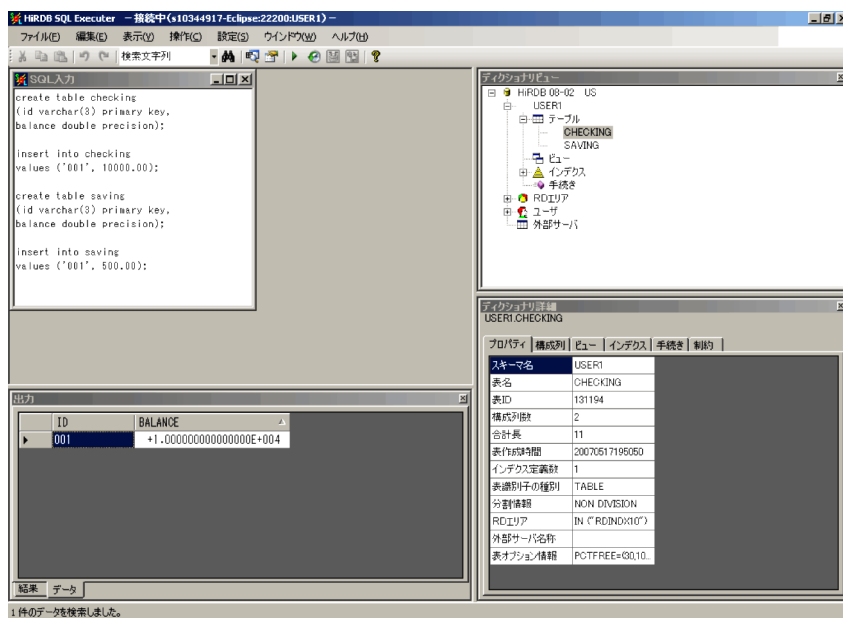
ここでは、GUI 版 HiRDB SQL Executer を使用してテーブルの内容を自動で参照する手順を説明します。

1. GUI 版 HiRDB SQL Executer の [ディクショナリビュー] ウィンドウの [< 接続データベース名 >] - [< スキーマ名 >] - [テーブル] を選択します。

テーブルが展開されます。

2. 参照したいテーブルを選択して、コンテキストメニューから [データを開く] を選択します。

[出力] ビューに参照結果が出力されます。



3.4 RD エリアの追加と削除

ここでは、RD エリアの追加方法および削除方法について説明します。画像などのバイナリデータを扱うために BLOB を使用する場合は、BLOB 型の列数分の RD エリアを追加する必要があります。

3.4.1 RD エリアの追加の流れ

コマンドを使用して RD エリアを追加します。RD エリアを追加するには、RD エリア用のファイル領域を初期化しておく必要があります。

(1) 組み込み DB プロンプトの起動

Windows の [スタート] メニューから [Cosminexus] - [Application Development Plug-in] - [組み込み DB プロンプト] から、組み込み DB プロンプトを起動します。

(2) RD エリア用ファイル領域の初期化

RD エリア用ファイル領域は、次のコマンドを組み込み DB プロンプトに入力することで初期化します。

```
pdfmkfs -k DB -n 物理ファイルサイズ -l 物理ファイル内に含む最大論理ファイル数
-e 論理ファイルの最大容量拡張回数 物理ファイル絶対パス
```

組み込みデータベースでは一つの物理ファイル内を複数の論理ファイルに分けて管理しています。ここでは、物理ファイルと論理ファイルの各種パラメタを指定します。

- -n 物理ファイルサイズ
ファイルサイズを MB 単位で記述します。1 ~ 2047 の符号なし整数を指定します。
- -l 物理ファイル内に含む最大論理ファイル数
一つの物理ファイル内に作成できる論理ファイルの個数の上限値を指定します。1 ~ 4096 の符号なし整数を指定します。
- -e 論理ファイルの最大容量拡張回数
論理ファイルの容量拡張回数の上限値を指定します。0 ~ 60000 の符号なし整数を指定します。デフォルト値は 0 です。
組み込みデータベースでは指定した回数に従って、論理ファイルの容量を必要に応じて増分していきます。なお、1 個の論理ファイルの拡張回数は最大 23 回です。最大論理ファイル数 × 23 以上の値を指定した場合は指定を無効とし、最大論理ファイル数 × 23 が仮定されます。
- 物理ファイル絶対パス
物理ファイルを絶対パスで指定します。

(3) RD エリアの追加

RD エリアは、次のコマンドを組み込み DB プロンプトに入力することで追加します。

```
pdmod -a パラメタファイルパス
```

パラメタファイルには次の内容を記載します。なお、RD エリアはセグメントで管理されています。1 セグメントは 8,192 バイトです。

```
create rdarea "RDエリア名"  
for LOB used by PUBLIC  
extension use 増分セグメント数 segments  
file name "物理ファイル絶対パス¥論理ファイル名"  
initial 初期セグメント数 segments;
```

- RD エリア名
RD エリアの名称を 1 ~ 30 文字の識別子で指定します。RD エリアを指定したテーブルを作成するときに、create table ステートメントのパラメタとして指定する名称を指定します。
- 増分セグメント数
このパラメタでは 1 回の容量拡張で増加させるセグメント数を指定します。1 ~ 64000 の符号なし整数を指定します。
RD エリアの容量が不足した場合、物理ファイル内に空きがあれば、自動的に RD エリアを拡張できます（容量拡張は、pdfmkfs の -e オプションで指定した回数以上は実施できません）。
- 物理ファイル絶対パス¥論理ファイル名
RD エリアに対して割り当てる、物理ファイルの絶対パスと論理ファイルの名称を 131 文字以内で指定します。物理ファイルの絶対パスは、pdfmkfs で指定した値と一致させてください。論理ファイル名は、物理ファイル内で一意な文字列を指定してください。
- 初期セグメント数
あらかじめ確保しておくセグメント数を指定します。符号なし整数を指定します。

(4) RD エリアを指定したテーブル作成

追加した RD エリアを指定してテーブルを作成します。テーブルの作成方法については、マニュアル「HiRDB SQL リファレンス」を参照してください。

3.4.2 RD エリアの削除の流れ

コマンドを使用して RD エリアを削除します。RD エリアを削除するには、RD エリアを閉塞しておく必要があります。

(1) 組み込み DB プロンプトの起動

Windows の [スタート] メニューから [Cosminexus] - [Application Development Plug-in] - [組み込み DB プロンプト] から , 組み込み DB プロンプトを起動します。

(2) RD エリアの閉塞

RD エリアは , 次のコマンドを組み込み DB プロンプトに入力することで閉塞します。
RD エリアは閉塞したあとでないと , 削除できません。

```
pdhold -r RDエリア名
```

- RD エリア名

閉塞する RD エリアの名称を指定します。RD エリアの名称は , 1 ~ 30 文字の識別子を指定します。

(3) RD エリアの削除

RD エリアを削除します。RD エリアは , 次のコマンドを組み込み DB プロンプトに入力することで削除します。

```
pdmod -a パラメタファイルパス
```

パラメタファイルには , 次の内容を記載しておきます。

```
remove rdarea "RDエリア名"
```

- RD エリア名

RD エリアの名称を指定します。RD エリアの名称は , 1 ~ 30 文字の識別子を指定します。

3.5 組み込みデータベースの動作制御の設定

環境変数を設定することで組み込みデータベースの動作を制御できます。環境変数は、システムの環境変数で設定します。システムの環境変数は、Windows の [コントロールパネル] の中の [システム] を選択し、[詳細設定] タブの中の [環境変数] ボタンから表示される画面で設定します。

設定する環境変数は次の表を参照してください。また、環境設定の指定値および動作については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

表 3-1 組み込みデータベースの動作を制御する環境変数

環境変数	機能	指定値 (デフォルト値)	機能説明
PDTMID	OLTP 識別子	4 文字	複数の OLTP から一つの組み込みデータベースにアクセスする場合、それぞれの OLTP にユニークな識別子を指定します。
PDXAM ODE ¹	トランザクションの移行機能	{0 1}	OLTP システムと連携する場合に、トランザクションの移行機能を使用するかどうかを指定します。必ず 1 を指定してください。
PDTXAC ANUM ¹	トランザクション最大同時実行数	1 ~ 2147483647 (20)	XA インタフェースを使用した UAP から同時実行する、最大トランザクション数を指定します。符号なし整数を指定します。 次の値を指定します。 (組み込みデータベースにアクセスする J2EE コンポーネントの同時実行数) × (J2EE コンポーネントでアクセスする可能性があるコネクション数)
PDXARC VWTIME	トランザクションの待ち合わせ時間	0 ~ 270 (2)	トランザクションが回復できない場合の待ち合わせ時間を指定します。符号なし整数を指定します。単位は秒です。
PDUSER ¹	ユーザ名・パスワード	パスワードなしのカレントユーザ名	ユーザ名、およびパスワードを指定します。
PDCLTA PNAME	実行する UAP の識別名称	30 文字以内 (unknown)	組み込みデータベースに対してアクセスする、UAP の識別情報 (UAP 識別子) を指定します。ここで指定した名称は、SQL トレースファイルなどの UAP 名称として表示されます。文字列を指定します。
PDDBLOG	ログの取得	{ALL NO}	UAP を実行するときに、データベースの更新ログを取得するかどうかを指定します。 ALL ログ取得モードで UAP を実行します。 NO ログレスモードで UAP を実行します。 NO を選択すると、トランザクションの回復ができなくなります。必ず ALL を指定してください。

環境変数	機能	指定値 (デフォルト値)	機能説明
PDAUTO RECON NECT 1	自動再接続 機能	{YES NO}	自動再接続機能を使用するかどうかを指定します。 J2EE サーバでコネクションの障害検知機能を使用する場合は、NO を指定してください。
PDRCCO UNT	CONNECT のリトライ 回数	1 ~ 200 (5)	自動再接続機能での CONNECT のリトライ回数を指定します。符号なし整数を指定します。
PDRCIN TERVAL	CONNECT のリトライ 間隔	1 ~ 600 (5)	自動再接続機能での CONNECT のリトライ間隔を指定します。単位は秒です。
PDCWAI TTIME 1	J2EE サー バの最大待 ち時間	0 ~ 65535 (0)	J2EE サーバから組み込みデータベースへ要求をしてから、応答が戻ってくるまでの J2EE サーバの最大待ち時間を指定します。符号なし整数を指定します。単位は秒です。 トランザクションタイムアウトの値よりも大きな値を指定します。
PDSWAI TTIME 1	データベ ースの最大待 ち時間 (ト ランザク ション処理 中)	0 ~ 65535 (600)	組み込みデータベースが J2EE サーバからの要求に対する応答を返してから、次に J2EE サーバから要求が来るまでの組み込みデータベースの最大待ち時間を指定します。この時間監視は、トランザクション処理中の時間です。 符号なし整数を指定します。単位は秒です。 トランザクションタイムアウトの値よりも大きな値を指定します。
PDSWAT CHTIME 1	データベ ースの最大待 ち時間 (ト ランザク ション処理 以外)	0 ~ 65535	組み込みデータベースが J2EE サーバからの要求に対する応答を返してから、次に J2EE サーバから要求が来るまでの組み込みデータベースの最大待ち時間を指定します。この時間監視は、トランザクション処理以外の時間です。 符号なし整数を指定します。単位は秒です。 0 を指定した場合、組み込みデータベースは J2EE サーバからの要求があるまで待ち続けます。コネクションプーリング機能を使用する場合は、0 を指定してください。
PDTIME DOUTRE TRY	システム コールのリ トライ回数	0 ~ 32767 (2)	J2EE サーバが HiRDB サーバと接続する場合に実行する、connect() システムコールでエラーが発生したときに、connect() システムコールをリトライする回数を指定します。符号なし整数を指定します。
PDCLTP ATH	SQL トレ ースファイ ル・エラー ログファイ ルの格納先 ディレクト リ	カレント ディレク トリのバ ス名	J2EE サーバが作成する SQL トレースファイルおよびエラーログファイルの格納先ディレクトリを指定します。 パス名を指定します。

3. テスト用実行環境で使用するデータベースのテーブルの作成

環境変数	機能	指定値 (デフォルト値)	機能説明
PDSQLTRACE	SQL トレースファイルのサイズ	0 または 4096 ~ 2000000 000	UAP の SQL トレースを出力する SQL トレースファイルのサイズを指定します。単位はバイトです。 0 を指定した場合は、ファイルの最大のサイズとなります。省略した場合は、SQL トレースを出力しません。
PDUAPERLOG	エラーログファイルのサイズ	0 または 4096 ~ 65536 (4096)	UAP のエラーログを出力するエラーログファイルのサイズを指定します。符号なし整数を指定します。単位は、バイトです。
PDERRSKIPCODE	エラーログへのメッセージ出力の抑止	SQLCODE[,SQLCODE...]	特定のエラーログを出力しない場合に指定します。
PDPRMTRC	パラメタ情報・検索データの出力	{YES NO IN OUT INOUT}	SQL トレースにパラメタ情報および検索データを出力するかどうかを指定します。 YES SQL トレースに入力用パラメタ情報を出力します。検索データ情報と入力パラメタを出力します。 NO SQL トレースにパラメタ情報を出力しません。 IN SQL トレースに入力用パラメタ情報を出力します。CALL 文の IN パラメタと INOUT パラメタも該当します。 OUT SQL トレースに出力用パラメタ情報、および検索データ情報を出力します。CALL 文の OUT パラメタと INOUT パラメタも該当します。 INOUT SQL トレースに入力用パラメタ情報、出力用パラメタ情報、および検索データ情報を出力します。CALL 文の INOUT パラメタは 2 回出力します。
PDPRMTRCSIZE	パラメタ情報・検索データの最大データ長	4 ~ 32008 (256)	SQL トレースに出力するパラメタ情報および検索データの最大データ長を指定します。符号なし整数を指定します。単位はバイトです。
PDTRCMODE	トラブルシュート情報の出力	{ERR NONE}	SQL トレース以外のトラブルシュート情報 (pderr*.trc の情報) を出力するかどうかを指定します。 ERR pderr*.trc の情報を出力します。 NONE pderr*.trc の情報を出力しません。

環境変数	機能	指定値 (デフォルト値)	機能説明
PDUAPR EPLVL	UAP 統計 レポートの 出力情報	{[s][u][p] [r][a]}	UAP 統計レポートの出力情報を指定します。 s SQL 単位の情報が出力されます。また、SQL ト レース情報も出力します。 u UAP 単位の情報が出力されます。 p アクセスパス情報が出力されます。 r SQL 実行時の中間結果情報を出力します。 a すべての情報が出力されます。
PDREPP ATH	UAP 統計 レポートの 出力先	パス名	PDCLTPATH で指定したディレクトリとは別のディレ クトリに、UAP 統計レポートを出力する場合に指定し ます。 パス名を指定します。
PDSQLT EXTSIZE	SQL 文のサ イズ	4096 ~ 2000000 (4096)	SQL トレースに出力する SQL 文のサイズを指定しま す。符号なし整数を指定します。単位はバイトです。
PDSQLE XECTIME	SQL 実行時 間の出力	{YES N O}	SQL トレースに SQL 実行時間を出力するかどうかを 指定します。 YES SQL 実行時間を出力します。出力される SQL 実 行時間の単位はマイクロ秒となります。SQL ト レースに出力される値は、実行時間が 24 時間以 上のものは正常に出力されません。 NO SQL 実行時間を出力しません。
PDRCTR ACE	UAP の再 接続トレ ースを出力 するファイ ルのサイ ズ	0 または 4096 ~ 2000000 000	UAP の再接続トレースを出力するファイルのサイ ズを指定します。符号なし整数を指定します。単位はバ イトです。 0 を指定した場合はファイルの最大サイズとなります。 省略した場合、UAP の再接続トレースは出力されませ ん。

3. テスト用実行環境で使用するデータベースのテーブルの作成

環境変数	機能	指定値 (デフォルト値)	機能説明
PDVWO PTMODE	アクセスパス情報ファイルの取得	{0 1 2}	<p>アクセスパス情報ファイルを取得するかどうかを指定します。</p> <p>0 アクセスパス情報を取得しません。</p> <p>1 アクセスパス情報を取得し、アクセスパス情報ファイルに出力します。このとき、SQL オブジェクトがバッファ中にある SQL については、情報を出力しません。</p> <p>2 アクセスパス情報を取得し、アクセスパス情報ファイルに出力します。このとき、SQL オブジェクトがバッファ中にある SQL についても SQL オブジェクトを再作成し、情報を出力します。</p>
PDSTJT RNOUT 2	統計ログファイルの出力	{YES NO}	<p>UAP に関する統計情報を、トランザクションごとに統計ログファイルに出力するかどうかを指定します。</p> <p>YES UAP に関する統計情報を、トランザクションごとに統計ログファイルに出力します。</p> <p>NO UAP に関する統計情報を、コネクションごとに統計ログファイルに出力します。</p>
PDLOCK SKIP	無排他条件判定	{YES NO}	<p>無排他条件判定をするかどうかを指定します。</p> <p>YES 無排他条件判定をします。</p> <p>NO 無排他条件判定をしません。</p>
PDFORU PDATEE XLOCK	WITH EXCLUSIVE LOCK の適用	{YES NO}	<p>FOR UPDATE 句を指定した (または仮定された) SQL の排他オプションに、WITH EXCLUSIVE LOCK を適用するかどうかを指定します。</p> <p>YES FOR UPDATE 句を指定した SQL の排他オプションに、WITH EXCLUSIVE LOCK を適用します。</p> <p>NO FOR UPDATE 句を指定した SQL の排他オプションに、PDISLLVL の指定値を適用します。</p>
PDISLLVL	SQL 文のデータ保証レベル	0 ~ 2 (2)	SQL 文のデータ保証レベルを指定します。符号なし整数を指定します。
PDSQLO PTLVL 3	SQL 最適化オプション	SQL 最適化オプションの指定値	データベースの状態を考慮して、最も効率的なアクセスパスを決定するための最適化の方法 (SQL 最適化オプション) を指定します。識別子、または符号なし整数を指定します。

環境変数	機能	指定値 (デフォルト値)	機能説明
PDADDITIONAL OPTLVL 3	SQL 拡張最適化オプション	SQL 拡張最適化オプションの指定値	データベースの状態を考慮して、最も効率的なアクセスパスを決定するための最適化の方法 (SQL 拡張最適化オプション) を指定します。識別子、または符号なし整数を指定します。
PDHASH TBLSIZE	ハッシュ表サイズ	128 ~ 524288	SQL の最適化で、ハッシュジョイン、副問合せのハッシュ実行を適用する場合、ハッシュ表サイズを指定します。符号なし整数を指定します。単位はバイトです。
PDDFLN VAL	埋め込み変数の既定値	{USE NOUSE}	表中のデータを埋め込み変数に取り出す場合、取り出した値が NULL 値のときに、埋め込み変数に既定値を設定するかどうかを指定します。 USE NULL 値に既定値を設定します。 NOUSE NULL 値に既定値を設定しません。
PDAGGR 4	グループ数の最大値	0 ~ 3000000 0 (1024)	GROUP BY 処理に使用するメモリ量を決定するため、サーバごとに発生するグループ数の最大値を指定します。符号なし整数を指定します。
PDDSQL OBJCACHE	SQL オブジェクト用バッファの再利用	{YES NO}	動的 SQL の SQL オブジェクトを、SQL オブジェクト用バッファで再利用するかどうかを指定します。 YES 再利用します。 NO 再利用しません。
PDCMM TBFDDL	定義系 SQL の実行	{YES NO}	操作系 SQL を実行していたトランザクションで定義系 SQL を実行する場合、自動的にコミットしてから定義系 SQL を実行するかどうかを指定します。 YES 操作系 SQL のトランザクションを自動的にコミットしてから、定義系 SQL を実行します。 NO 操作系 SQL のトランザクションを自動的にコミットしないで、定義系 SQL を実行します。
PDBLKF	1 回の転送処理で送られる行数	1 ~ 4096 (1)	HiRDB サーバから HiRDB クライアントに検索結果を転送するときの、1 回の転送処理で送られる行数を指定します。符号なし整数を指定します。
PDBINARYBLKF	ブロック転送機能の適用	{YES NO}	定義長が 32,001 バイト以上の BINARY 型の選択式がある表を検索する場合、ブロック転送機能を適用するかどうかを指定します。 YES ブロック転送機能を適用します。 NO ブロック転送機能を適用しません。

3. テスト用実行環境で使用するデータベースのテーブルの作成

環境変数	機能	指定値 (デフォルト値)	機能説明
PDBLKB UFFSIZE	通信バッファのサイズ	0 ~ 2000000 (0)	ブロック転送機能で使用する、サーバおよびクライアント間の通信バッファのサイズを指定します。 符号なし整数を指定します。単位はバイトです。
PDSPAC ELVL	空白変換レベル	{0 1 3}	データの格納、比較、および検索時の、空白変換レベルを指定します。 0 空白は変換しません。 1 操作系 SQL での定数、埋め込み変数、またはパラメタのデータの空白を変換します。 3 空白変換レベル 1 に加えて、各国文字列型の値式のデータを検索するときに、全角空白 1 文字を半角空白 2 バイトに変換します。
PDCNST RNTNAME	制約名定義の位置	{LEADING TRAILING}	参照制約、および検査制約を定義する場合、制約名定義の位置を指定します。 LEADING 制約名定義を、制約定義の前に指定します。 TRAILING 制約名定義を、制約定義の後に指定します。

注

指定値で使用している記号の意味を次に示します。

{A|B} : A または B を指定することを示します。

[C] : C の指定は任意であることを示します。

注 1

環境変数については、マニュアル「Cosminexus システム構築ガイド」を参照してください。

注 2

環境変数については、マニュアル「HiRDB コマンドリファレンス」を参照してください。

注 3

環境変数については、マニュアル「HiRDB SQL リファレンス」を参照してください。

注 4

環境変数については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

4

プロジェクトの作成

MyEclipse で J2EE アプリケーションを開発する場合は、最初にプロジェクトを作成します。

この章では、MyEclipse のプロジェクトの作成手順、リソースアダプタのインポート手順、および DD の作成手順について説明します。

4.1 MyEclipse でのプロジェクトの作成

4.2 EJB プロジェクトの作成

4.3 Web プロジェクトの作成

4.4 エンタープライズアプリケーションプロジェクトの作成

4.5 リソースアダプタのインポート

4.6 DD の編集

4.1 MyEclipse でのプロジェクトの作成

MyEclipse を使用して J2EE アプリケーションを開発するには、最初に MyEclipse のプロジェクトを作成します。MyEclipse のプロジェクトを作成したあと、作成したプロジェクト単位でプログラムおよびファイルを作成します。

ここでは、MyEclipse で作成するプロジェクトの構成およびプロジェクト作成の流れを説明します。

4.1.1 MyEclipse で作成するプロジェクトの構成

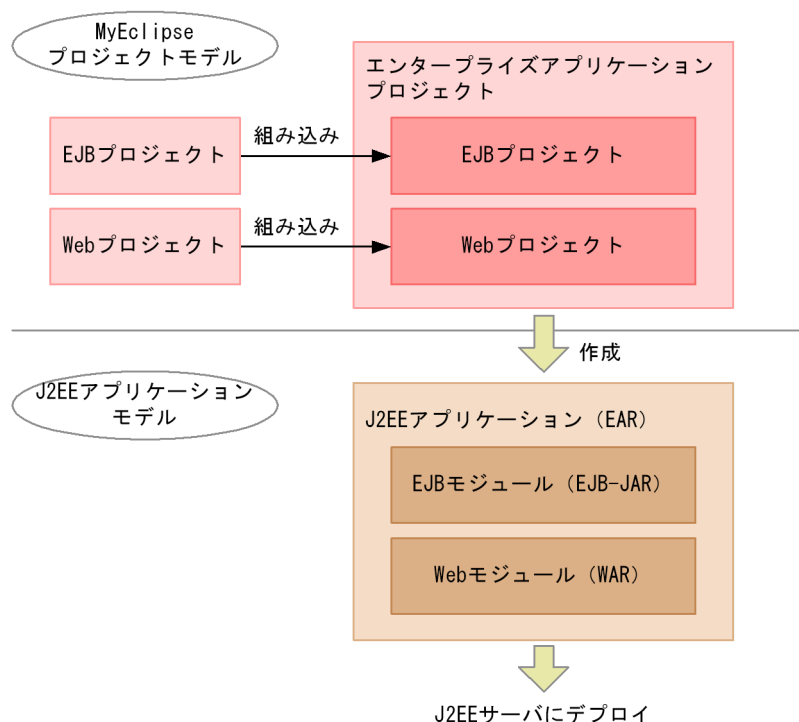
MyEclipse では、EJB プロジェクト、Web プロジェクト、エンタープライズアプリケーションプロジェクトを作成します。

それぞれのプロジェクトについて説明します。

- EJB プロジェクト
Session Bean、Entity Bean の EJB、ejb-jar.xml など構成されるプロジェクトです。Session Bean、Entity Bean の EJB は、業務処理プログラムとなります。
- Web プロジェクト
JSP ファイル、サーブレット、HTML、web.xml など構成されるプロジェクトです。JSP ファイル、サーブレット、HTML は、入力画面となります。
- エンタープライズアプリケーションプロジェクト
Web プロジェクト、EJB プロジェクト、および application.xml を一つの J2EE アプリケーションとしてまとめるためのプロジェクトです。

MyEclipse で作成する各プロジェクトの構成とそれに対応する J2EE アプリケーションの構成を次の図に示します。MyEclipse ではプロジェクトのデプロイ時にビルドが実行されます。

図 4-1 MyEclipse で作成するプロジェクトの構成



MyEclipse ではエンタープライズアプリケーションプロジェクトに EJB プロジェクトおよび Web プロジェクトをモジュールプロジェクトとして組み込みます。エンタープライズアプリケーションプロジェクトには、モジュールプロジェクトを一つ以上組み込む必要があります。

J2EE サーバにデプロイするために、エンタープライズアプリケーションプロジェクトから J2EE アプリケーション (EAR)、EJB プロジェクトから EJB モジュール (EJB-JAR)、Web プロジェクトから Web モジュール (WAR) が作成されます。

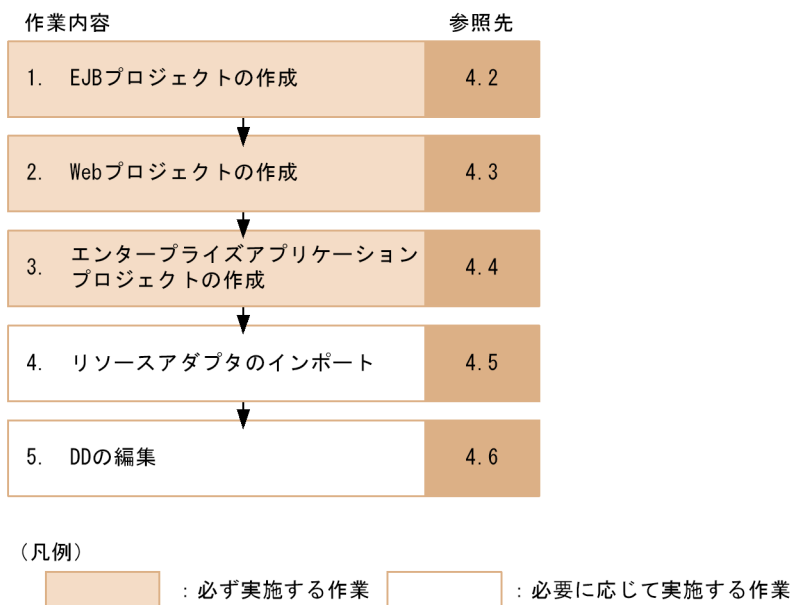
4.1.2 プロジェクト作成の流れ

MyEclipse では、EJB プロジェクト、Web プロジェクト、およびエンタープライズアプリケーションプロジェクトを作成します。

ここでは、MyEclipse のプロジェクトを作成する流れを説明します。

4. プロジェクトの作成

図 4-2 プロジェクト作成の流れ



それぞれの作業の概要を説明します。

1. EJB プロジェクトの作成

MyEclipse の機能を使用して、EJB プロジェクトを作成します。

2. Web プロジェクトの作成

MyEclipse の機能を使用して、Web プロジェクトを作成します。

3. エンタープライズアプリケーションプロジェクトの作成

事前に作成した EJB プロジェクトと Web プロジェクトを組み込んで、エンタープライズアプリケーションプロジェクトを作成します。

4. リソースアダプタのインポート

エンタープライズアプリケーションプロジェクトにリソースアダプタをインポートします。

5. DD の編集

[MyEclipse XML エディター] を使用して各プロジェクトの DD を編集します。

以降の節では、この流れに沿って MyEclipse のプロジェクトの作成手順を説明します。

4.2 EJB プロジェクトの作成

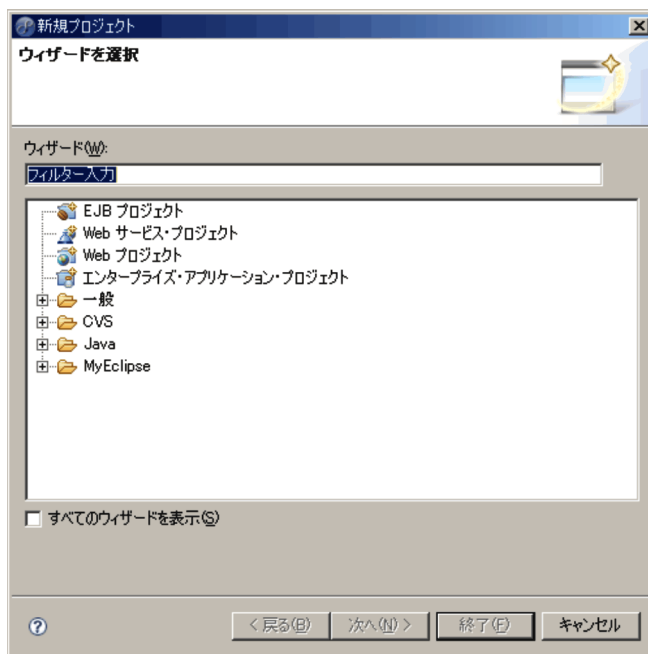
EJB プロジェクトは、業務処理プログラムとなる Session Bean や Entity Bean の EJB ,
ejb-jar.xml などで構成されます。

ここでは、MyEclipse の EJB プロジェクトを作成する手順と作成時の注意事項を説明します。また、EJB プロジェクトはプロジェクトの作成と同時に DD を作成しません。このため、ejb-jar.xml の作成手順についても説明します。

4.2.1 EJB プロジェクトの作成手順

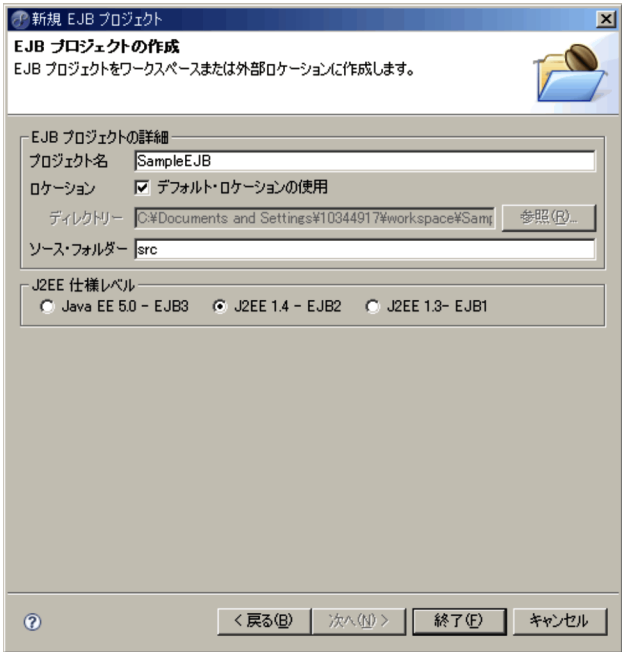
MyEclipse で EJB プロジェクトを作成する手順を説明します。

1. Eclipse のメニューから [ファイル] - [新規] - [プロジェクト] を選択します。
[新規プロジェクト] ダイアログが表示されます。



2. [EJB プロジェクト] を選択して , [次へ] ボタンをクリックします。
[EJB プロジェクトの作成] ページが表示されます。

4. プロジェクトの作成



設定が必要な項目を次に示します。

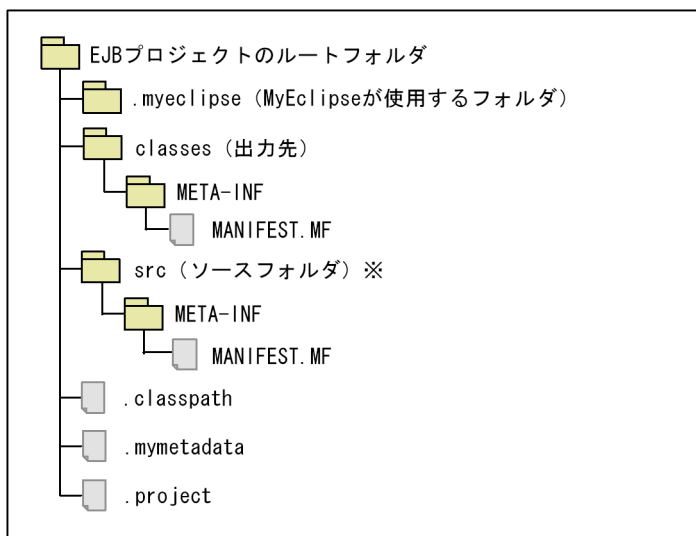
項目名	設定値
プロジェクト名	作成する EJB プロジェクトのプロジェクト名を指定します。
ソース・フォルダー	Java ソースフォルダを指定します。
J2EE 仕様レベル	[J2EE1.4 - EJB2] を指定します。

必要に応じて、次の項目を設定してください。

項目名	設定値
ロケーション	プロジェクトルートに使用するロケーションを指定します。 <ul style="list-style-type: none">デフォルト・ロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックします。任意のロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックしないで、[ディレクトリー] で任意のロケーションを指定します。
ディレクトリー	[デフォルト・ロケーションの使用] をチェックしていない場合に、プロジェクトルートに使用するロケーションのパスを指定します。 [参照] ボタンをクリックすると、[フォルダの参照] ダイアログから指定できます。 ただし、デフォルトと同じワークスペースにあるほかのディレクトリは選択できません。

3. [終了] ボタンをクリックします。
EJB プロジェクトが作成されます。

作成される EJB プロジェクトの構成を次に示します。



注※ ソースフォルダは「EJB プロジェクトの作成」ページで指定した任意のフォルダになります。

EJB プロジェクトを作成した時点では ejb-jar.xml は作成されません。ejb-jar.xml の作成については、「4.2.3 ejb-jar.xml ファイルのひな型の作成」を参照してください。

4.2.2 EJB プロジェクト作成時の注意事項

ここでは、EJB プロジェクト作成時の注意事項について説明します。

(1) EJB プロジェクト名の注意事項

EJB プロジェクト名は、作業ディレクトリ中のディレクトリ名として使用されます。作業ディレクトリのパス長が OS の上限に達しないように EJB プロジェクト名を指定してください。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(2) 別の EJB-JAR ファイルに含まれるクラスを参照する場合の注意事項

EJB-JAR ファイルに含まれているコンポーネントインタフェースのメソッドのリターン値または引数に、別の EJB-JAR ファイルに定義してあるクラスを指定した場合、指定したクラスは EJB-JAR ファイル内に含めるようにしてください。含めない場合、EJB-JAR ファイルに関して、属性ファイルでの設定変更ができません。

(3) アノテーション有効時の EJB-JAR ファイルの作成について

アノテーションが有効となる (DD なし、または EJB 2.1 の DD を使用) EJB-JAR ファイルを作成する場合、EJB-JAR ファイルに含まれるクラスのフィールド、およびメソッドのリターン値や引数として使用するユーザ定義のクラスは、インポートする単位に含めるようにしてください。EAR としてインポートする場合は EAR の中に、EJB-JAR としてインポートする場合は EJB-JAR の中にクラス (またはインターフェース) 定義が含まれる必要があります。含まれない場合は、インポート時にエラーが発生します。

(4) J2EE Application Client のデプロイと実行について

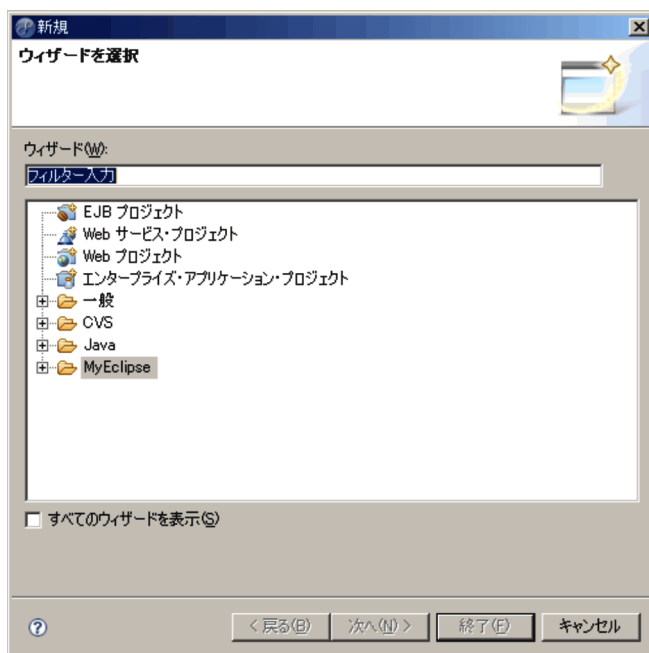
J2EE Application Client をデプロイし、実行する機能はサポートしていません。代わりに EJB クライアントアプリケーションを使用してください。なお、EAR ファイルに Application Client の JAR ファイルを含めて、application.xml に指定している場合は、該当する JAR ファイルおよび DD の指定を削除してからインポートしてください。

4.2.3 ejb-jar.xml ファイルのひな型の作成

EJB プロジェクト作成時には、EJB-JAR の DD は作成されません。XML ファイルの作成ウィザードから ejb-jar.xml ファイルのひな型を作成します。

EJB2.1 の DD を例に、作成手順を説明します。

1. [パッケージ・エクスプローラー] ビューでプロジェクトを選択します。
2. Eclipse のメニューから [ファイル] - [新規] - [その他] を選択します。
[新規] ダイアログが表示されます。

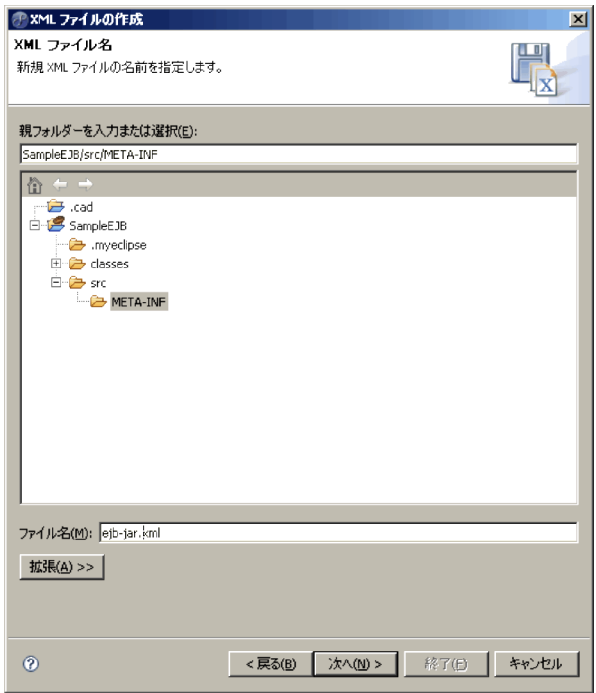


3. [MyEclipse] - [XML] - [XML(基本テンプレート)] を選択して , [次へ] ボタンをクリックします。
[XML ファイルの作成] ページが表示されます。



4. プロジェクトの作成

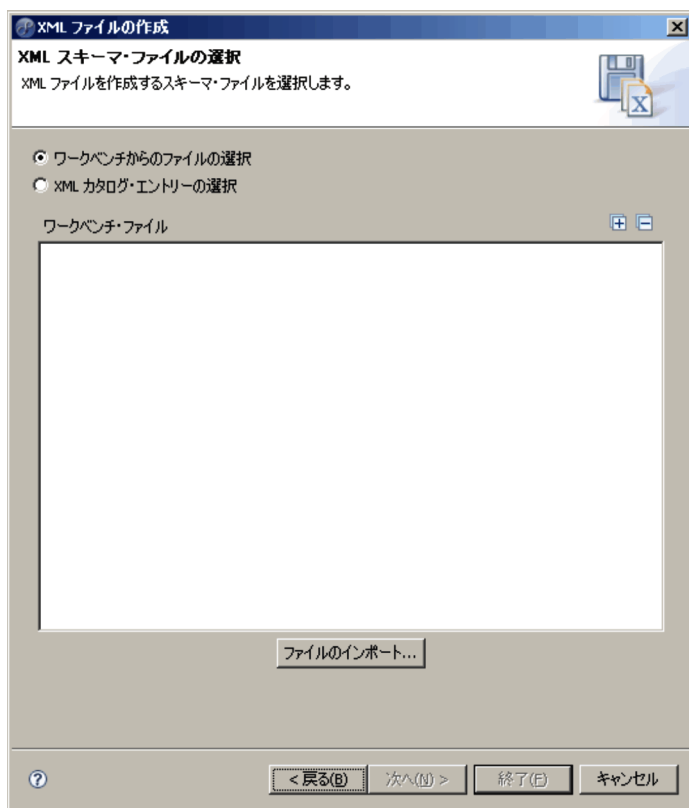
4. [XML スキーマ・ファイルから XML ファイルを作成] を選択して , [次へ] ボタンをクリックします。
- [XML ファイル名] ページが表示されます。



設定が必要な項目を次に示します。

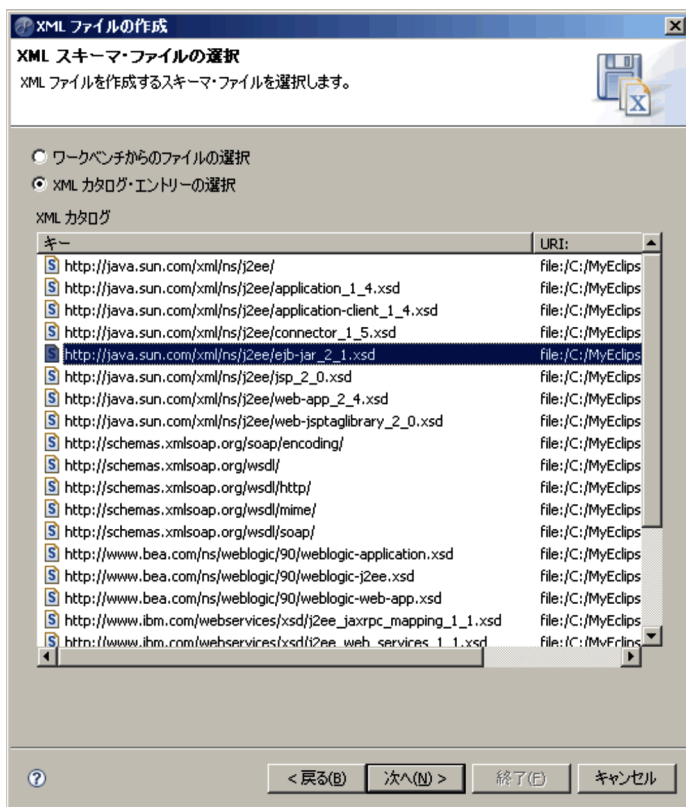
項目名	設定値
親フォルダーを入力または選択	「<EJB プロジェクト>/<ソースフォルダ>/META-INF」を指定します。
ファイル名	「ejb-jar.xml」を指定します。

5. [次へ] ボタンをクリックします。
- [XML スキーマ・ファイルの選択] ページが表示されます。

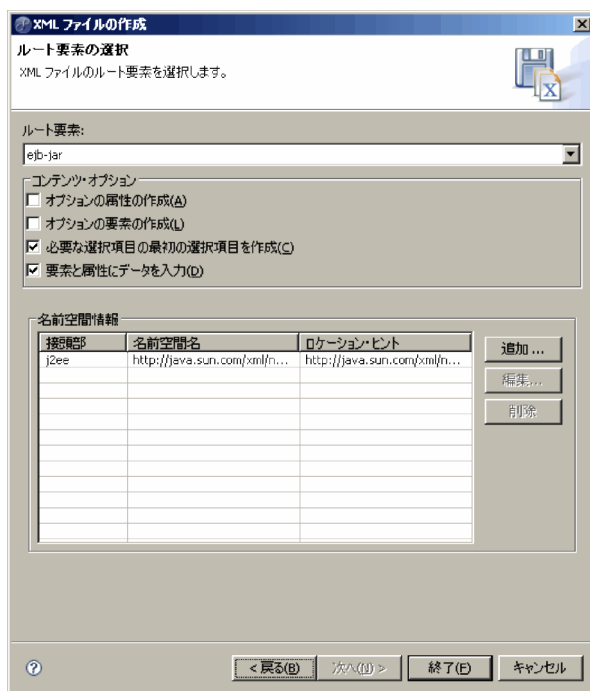


6. [XML カタログ・エントリーの選択] を選択して、一覧から [キー] 列の値が「`http://java.sun.com/xml/ns/j2ee/ejb-jar_2_1.xsd`」の XML カタログを選択します。

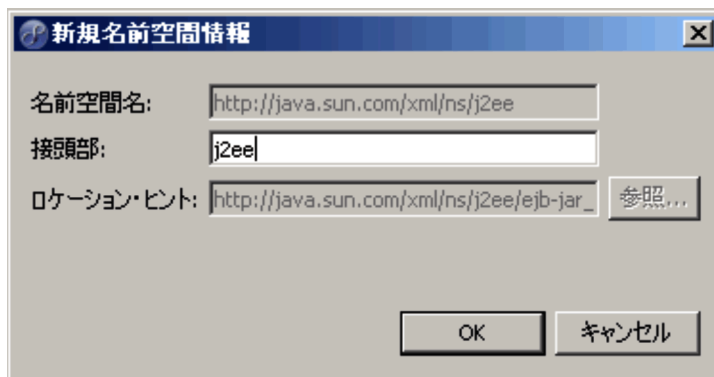
4. プロジェクトの作成



7. [次へ] ボタンをクリックします。
[ルート要素の選択] ページが表示されます。

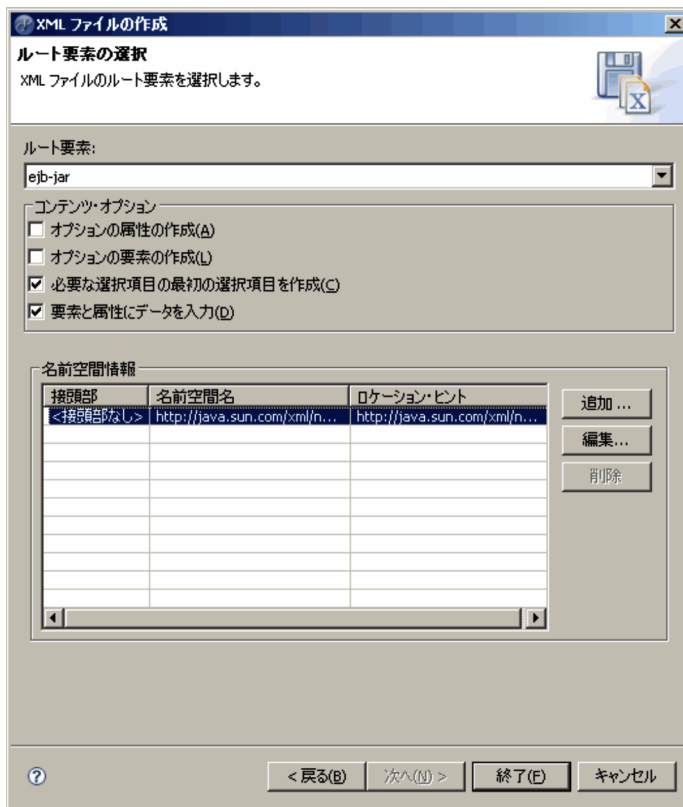


8. [名前空間情報] リストで [接頭部] 列に「j2ee」が表示されている行を選択して、[編集] ボタンをクリックします。
[新規名前空間情報] ダイアログが表示されます。



9. [接頭部] に入力されている「j2ee」を削除して、[OK] ボタンをクリックします。

4. プロジェクトの作成



10[ルート要素の選択] ページの [名前空間情報] リストで、編集した行の [接頭部] 列が「<接頭部なし>」に変更されていることを確認して、[終了] ボタンをクリックします。

ejb-jar.xml が作成されます。

作成される DD の例を次に示します。DD を編集する場合は、「4.6.4 ejb-jar.xml 編集時の注意事項」を参照してください。なお、<ejb-name> タグ、<ejb-class> タグ、<session-type> タグ、<transaction-type> タグには、仮の値が設定されているので、作成した EJB プロジェクトに合わせて変更してください。

```
<?xml version="1.0"encoding="UTF-8"?>
<ejb-jar version="2.1" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemalocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/
xml/ns/j2ee/ejb-jar_2_1.xsd">
  <enterprise-beans>
    <session>
      <ejb-name>NMTOKEN</ejb-name>
      <ejb-class>token</ejb-class>
      <session-type>token</session-type>
      <transaction-type>token</transaction-type>
    </session>
  </enterprise-beans>
</ejb-jar>
```

4.3 Web プロジェクトの作成

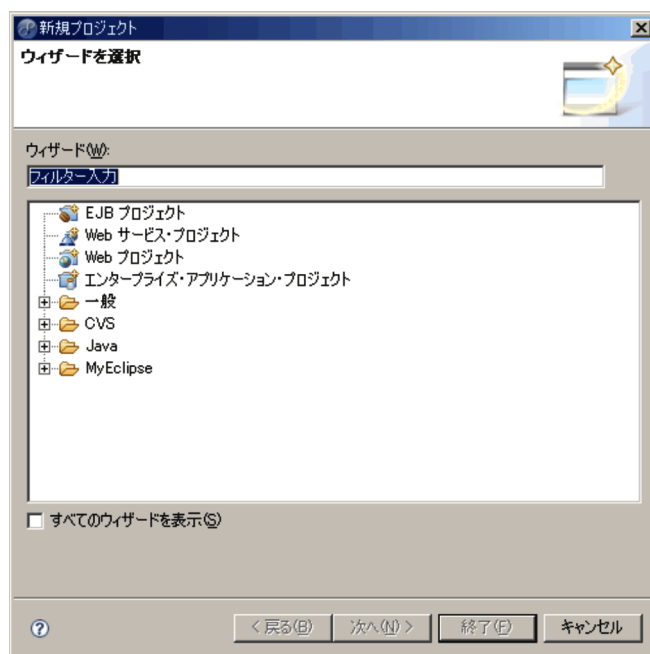
Web プロジェクトは、JSP ファイル、Servlet、HTML、web.xml など構成されます。なお、JSP ファイル、Servlet、および HTML は入力画面となります。

ここでは、MyEclipse の Web プロジェクトを作成する手順と作成時の注意事項を説明します。また、Web プロジェクトのコンテキストルートを変更する手順も説明します。

4.3.1 Web プロジェクトの作成手順

MyEclipse の Web プロジェクトを作成する手順を説明します。

1. Eclipse のメニューから [ファイル] - [新規] - [プロジェクト] を選択します。
[新規プロジェクト] ダイアログが表示されます。



2. [Web プロジェクト] を選択して [次へ] ボタンをクリックします。
[Web プロジェクトの作成] ページが表示されます。

4. プロジェクトの作成

設定が必要な項目を次に示します。

項目名	設定値
プロジェクト名	作成する Web プロジェクトのプロジェクト名を指定します。
ソース・フォルダー	Java ソースフォルダを指定します。
Web ルート・フォルダー	Web アプリケーションのルートとなるフォルダパスをプロジェクトルートからの相対パスで指定します。
コンテキスト・ルート URL	Web アプリケーションのコンテキストルートを指定します。ただし、URI (RFC3986) で使用できる文字を入力してください。
J2EE 仕様レベル	[J2EE1.4] を指定します。

必要に応じて、次の項目を設定してください。

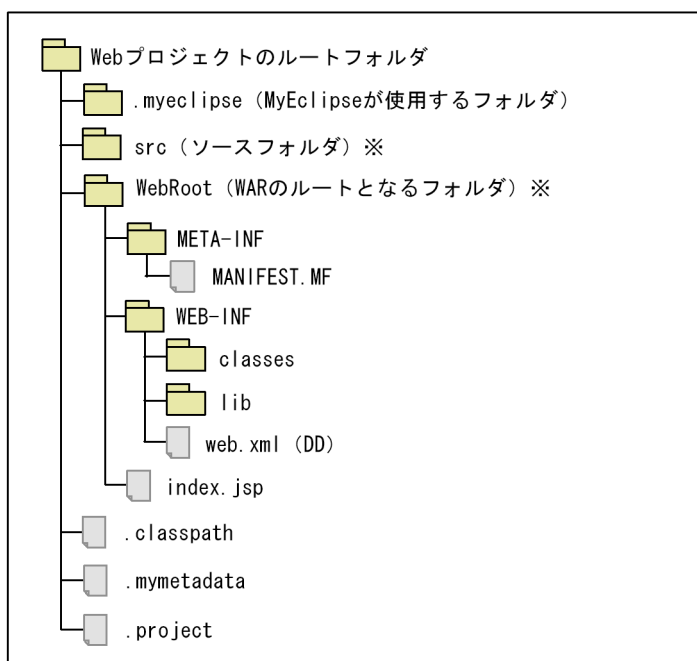
項目名	設定値
ロケーション	プロジェクトルートに使用するロケーションを指定します。 <ul style="list-style-type: none"> デフォルト・ロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックします。 任意のロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックしないで、 [ディレクトリ] で任意のロケーションを指定します。

項目名	設定値
ディレクトリ	[デフォルト・ロケーションの使用] をチェックしていない場合に、プロジェクトルートに使用するロケーションのパスを指定します。 [参照] ボタンをクリックすると、[フォルダの参照] ダイアログから指定できます。 ただし、デフォルトと同じワークスペースにあるほかのディレクトリは選択できません。
JSTL サポート	JSTL のライブラリを <Web アプリケーションのルート>¥WEB-INF¥lib にコピーする場合は、[JSTL ライブラリーを WEB-INF/lib フォルダーに追加する] をチェックします。さらに、コピーする JSTL のバージョンを選択します。

3. [終了] ボタンをクリックします。

Web プロジェクトが作成されます。

作成される Web プロジェクトの構成を次に示します。



注※ [Webプロジェクトの作成] ページで指定した任意のフォルダになります。

生成される DD を次に示します。DD の編集時の注意事項については、「4.6.5 web.xml 編集時の注意事項」を参照してください。

4. プロジェクトの作成

```
<?xml version="1.0"encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemalocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

4.3.2 Web プロジェクト作成時の注意事項

ここでは、Web プロジェクト作成時の注意事項について説明します。

(1) Web プロジェクト名の注意事項

Web プロジェクト名は、作業ディレクトリ中のディレクトリ名として使用されます。作業ディレクトリのパス長が OS の上限に達しないように Web プロジェクト名を指定してください。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(2) コンテキストルートの URL の指定について

Web プロジェクト作成時に Web アプリケーションのコンテキストルートを指定する場合は、URI (RFC3986) で使用できる文字を使用してください。

(3) アノテーション有効時の WAR ファイルの作成について

アノテーションが有効となる (Servlet2.4 の DD 使用) WAR ファイルを作成する場合、WAR ファイルに含まれるクラスのフィールドおよびメソッドのリターン値や引数として使用するユーザ定義のクラスは、インポートする単位に含めるようにしてください。EAR としてインポートする場合は EAR の中に、WAR としてインポートする場合は WAR の中にクラス (またはインタフェース) 定義が含まれる必要があります。含まれない場合は、インポート時にエラーが発生します。

4.3.3 コンテキストルートの変更手順

Web プロジェクト作成時に指定したコンテキストルートは、Web プロジェクトのプロパティと、その Web プロジェクトを含むエンタープライズアプリケーションプロジェクトの application.xml に保持されます。

コンテキストルートの変更と反映の関係を次の表に示します。

表 4-1 コンテキストルートの変更の反映

コンテキストルートを変更したダイアログ	Web プロジェクトのプロパティ	application.xml
Web プロジェクトのプロパティ ([プロパティ : <Web プロジェクト名>] ダイアログ)		x
エンタープライズアプリケーションプロジェクトのモジュール構成 ([モジュール・プロジェクト] ダイアログ)	x	

(凡例) : 変更が反映されます x : 変更が反映されません

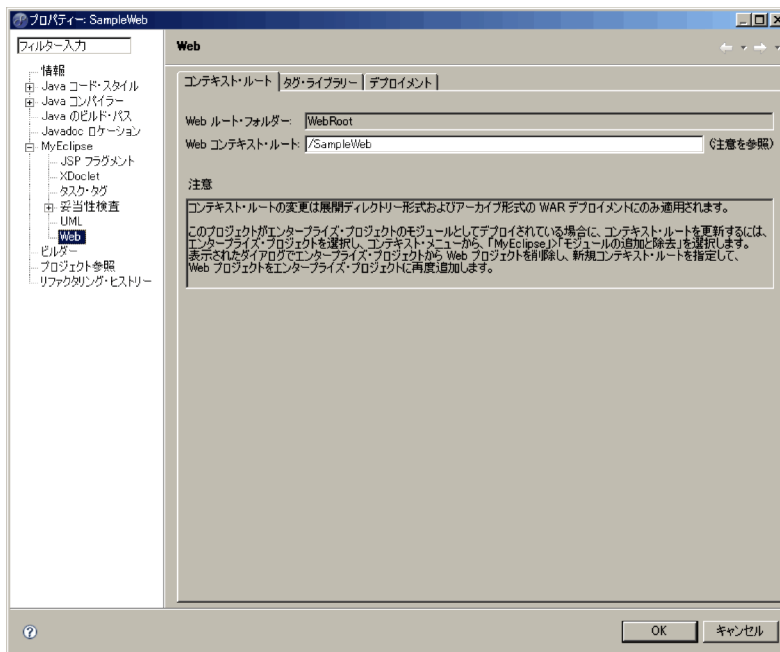
注

application.xml にコンテキストルートの変更を反映させるためには , [モジュール・プロジェクト] ダイアログで該当する Web プロジェクトを削除して , 再度追加する必要があります。
[モジュール・プロジェクト] ダイアログでの操作については , 「 4.4.3 エンタープライズアプリケーションプロジェクトのモジュールの変更手順 」を参照してください。

Web プロジェクトのプロパティでコンテキストルートを変更する手順を説明します。

1. [パッケージ・エクスプローラー] ビューで Web プロジェクトを選択します。
2. Eclipse のメニューから [プロジェクト] - [プロパティ] を選択します。
[プロパティ : <プロジェクト名>] ダイアログが表示されます。
3. [プロパティ : <プロジェクト名>] ダイアログの左ペインで [MyEclipse] - [Web] を選択します。
[Web] ページが表示されます。

4. プロジェクトの作成



4. [Web コンテキスト・ルート] を変更して, [OK] ボタンをクリックします。
コンテキストルートが変更されます。

4.4 エンタープライズアプリケーションプロジェクトの作成

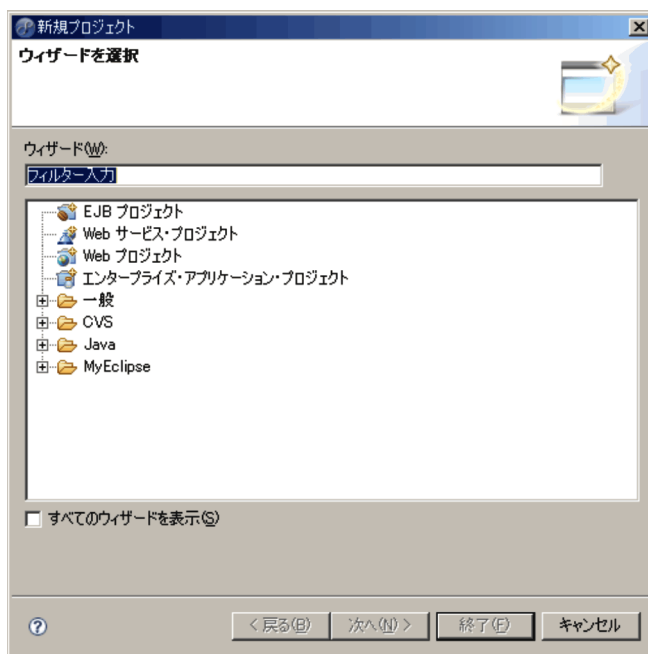
エンタープライズアプリケーションプロジェクトは、Web プロジェクトと EJB プロジェクトを一つの J2EE アプリケーションとしてまとめるプロジェクトです。また、エンタープライズアプリケーションプロジェクトはデプロイする単位になります。

ここでは、エンタープライズアプリケーションプロジェクトを作成する手順と作成時の注意事項を説明します。また、エンタープライズアプリケーションプロジェクトに組み込まれる EJB プロジェクトおよび Web プロジェクトの構成を変更する手順も説明します。

4.4.1 エンタープライズアプリケーションプロジェクトの作成手順

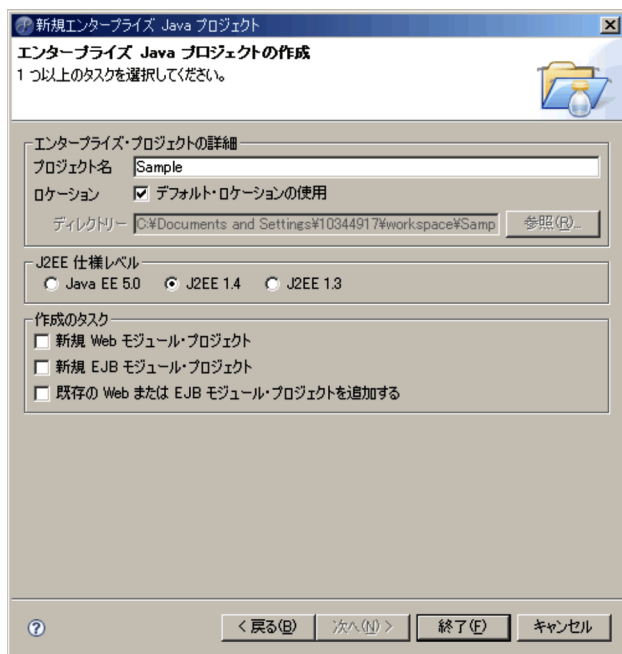
MyEclipse のエンタープライズアプリケーションプロジェクトを作成する手順を説明します。

1. Eclipse のメニューから [ファイル] - [新規] - [プロジェクト] を選択します。
[新規プロジェクト] ダイアログが表示されます。



2. [エンタープライズ・アプリケーション・プロジェクト] を選択し、[次へ] ボタンをクリックします。
[エンタープライズ Java プロジェクトの作成] ページが表示されます。

4. プロジェクトの作成



設定が必要な項目を次に示します。

項目名	設定値
プロジェクト名	作成するエンタープライズアプリケーションプロジェクトのプロ ジェクト名を指定します。
J2EE 仕様レベル	[J2EE1.4] を指定します。
作成のタスク	J2EE アプリケーションを構成するモジュールプロジェクトを作成 または選択します。作業内容に合わせて、次の項目にチェックを入 れます。 <ul style="list-style-type: none">• [新規 Web モジュール・プロジェクト] J2EE アプリケーションに含める Web プロジェクトを新規作成す る場合にチェックします。• [新規 EJB モジュール・プロジェクト] J2EE アプリケーションに含める EJB プロジェクトを新規作成す る場合にチェックします。• [既存の Web または EJB モジュール・プロジェクトを追加する] 既存のモジュールプロジェクトから、J2EE アプリケーションに 含めるプロジェクトを選択する場合にチェックします。

必要に応じて、次の項目を設定してください。

項目名	設定値
ロケーション	プロジェクトルートに使用するロケーションを指定します。 <ul style="list-style-type: none"> デフォルト・ロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックします。 任意のロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックしないで、[ディレクトリー] で任意のロケーションを指定します。
ディレクトリー	[デフォルト・ロケーションの使用] をチェックしていない場合に、プロジェクトルートに使用するロケーションのパスを指定します。 [参照] ボタンをクリックすると、[フォルダの参照] ダイアログから指定できます。 ただし、デフォルトと同じワークスペースにあるほかのディレクトリーは選択できません。

3. [次へ] ボタンをクリックします。

手順 2. の [エンタープライズ Java プロジェクトの作成] ページで、[作成のタスク] の [新規 Web モジュール・プロジェクト] をチェックした場合は、[Web プロジェクト・モジュールの定義] ページが表示されます。チェックしていない場合は、手順 6. に移ります。

4. Web プロジェクトのモジュールを定義します。

設定が必要な項目を次に示します。

項目名	設定値
プロジェクト名	新規作成する Web プロジェクト名を指定します。

4. プロジェクトの作成

項目名	設定値
ソース・フォルダー	Java ソースフォルダを指定します。
Web ルート・フォルダー	Web アプリケーションのルートとなるフォルダパスをプロジェクトルートからの相対パスで指定します。
コンテキスト・ルート URL	WAR のコンテキストルートを指定します。 ただし、URI (RFC3986) で使用できる文字を入力してください。

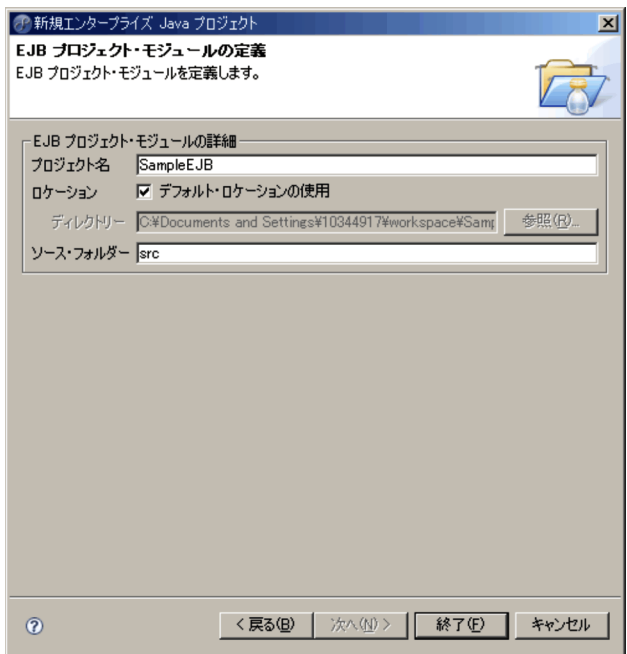
必要に応じて、次の項目を設定してください。

項目名	設定値
ロケーション	プロジェクトルートに使用するロケーションを指定します。 <ul style="list-style-type: none">デフォルト・ロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックします。任意のロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックしないで、[ディレクトリー] で任意のロケーションを指定します。
ディレクトリー	[デフォルト・ロケーションの使用] をチェックしていない場合に、プロジェクトルートに使用するロケーションのパスを指定します。 [参照] ボタンをクリックすると、[フォルダの参照] ダイアログから指定できます。 ただし、デフォルトと同じワークスペースにあるほかのディレクトリは選択できません。

5. [次へ] ボタンをクリックします。

手順 2. の [エンタープライズ Java プロジェクトの作成] ページで、[作成のタスク] の [新規 EJB モジュール・プロジェクト] をチェックした場合は、[EJB プロジェクト・モジュールの定義] ページが表示されます。チェックしていない場合は、手順 8. に移ります。

6. EJB プロジェクトのモジュールを定義します。



設定が必要な項目を次に示します。

項目名	設定値
プロジェクト名	新規作成する EJB プロジェクト名を指定します。
ソース・フォルダー	Java ソースフォルダを指定します。

必要に応じて、次の項目を設定してください。

項目名	設定値
ロケーション	プロジェクトのルートに使用するロケーションを指定します。 ・ デフォルト・ロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックします。 ・ 任意のロケーションに作成する場合 [デフォルト・ロケーションの使用] をチェックしないで、 [ディレクトリ] で任意のロケーションを指定します。
ディレクトリ	[デフォルト・ロケーションの使用] をチェックしていない場合に、プロジェクトのルートに使用するロケーションのパスを指定します。 [参照] ボタンをクリックすると、[フォルダの参照] ダイアログから指定できます。 ただし、デフォルトと同じワークスペースにあるほかのディレクトリは選択できません。

7. [次へ] ボタンをクリックします。

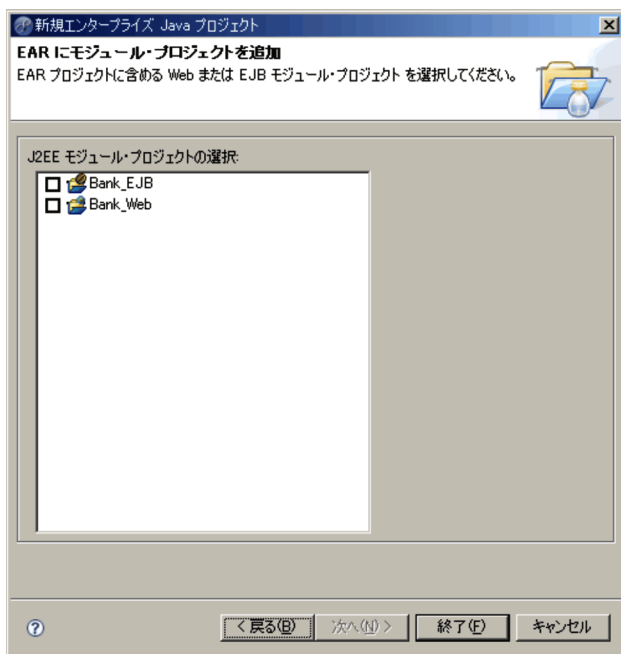
手順 2. の [エンタープライズ Java プロジェクトの作成] ページで、[作成のタスク] の [既存の Web または EJB モジュール・プロジェクトを追加する] をチェックした

4. プロジェクトの作成

場合は、[EAR にモジュール・プロジェクトを追加] ページが表示されます。

8. J2EE アプリケーションに含めるモジュールプロジェクトを [J2EE モジュール・プロジェクトの選択] から選択します。

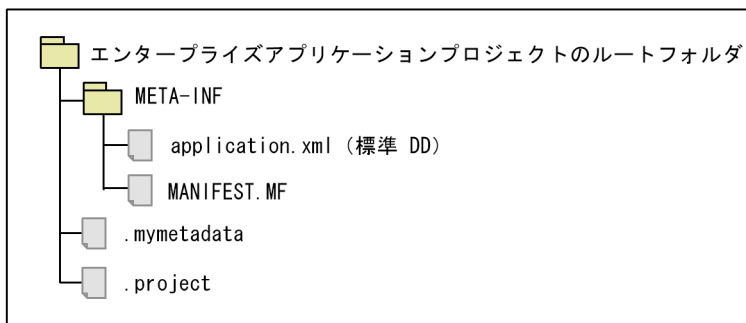
必要なモジュールプロジェクトをチェックします。



9. [終了] ボタンをクリックします。

エンタープライズアプリケーションプロジェクトが作成されます。

作成したプロジェクト構造を次に示します。



生成される DD の例を、次に示します。DD の編集時の注意事項については、「4.6.3 application.xml 編集時の注意事項」を参照してください。


```
<?xml version="1.0"encoding="UTF-8"?>
<application
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="1.4"
xsi:schemalocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/application_1_4.xsd">
  <display-name>Sample</display-name>
  <module id="myeclipse.1159250865328">
    <web>
      <web-uri>SampleWeb.war</web-uri>
      <context-root>/SampleWeb</context-root>
    </web>
  </module>
  <module id="myeclipse.1159250865781">
    <ejb>SampleEJB.jar</ejb>
  </module>
</application>
```

4.4.2 エンタープライズアプリケーションプロジェクト作成時の注意事項

ここでは、エンタープライズアプリケーションプロジェクト作成時の注意事項について説明します。

(1) エンタープライズアプリケーションプロジェクト名についての注意事項

エンタープライズアプリケーションプロジェクト名は、作業ディレクトリ中のディレクトリ名として使用されます。作業ディレクトリのパス長が OS で規定されているパス長の上限に達しないようにエンタープライズアプリケーションプロジェクト名を指定してください。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(2) エンタープライズアプリケーションプロジェクトの注意事項

エンタープライズアプリケーションプロジェクトのルートフォルダ下はデブロイの対象となります。このため、J2EE アプリケーションの動作に必要なファイルまたはフォルダがエンタープライズアプリケーションプロジェクトのルートフォルダ下にある場合、それらもデブロイされます。また、EAR ファイルをエクスポートしたときに、J2EE アプリケーションの動作に必要なファイルまたはフォルダが EAR ファイルに含まれます。エンタープライズアプリケーションプロジェクトには、不要なファイルを置かないようにしてください。

また、エンタープライズアプリケーションプロジェクトで作成された /META-INF ディレクトリおよび application.xml は、移動または名前変更をしないでください。移動または名前変更をすると、デブロイできません。

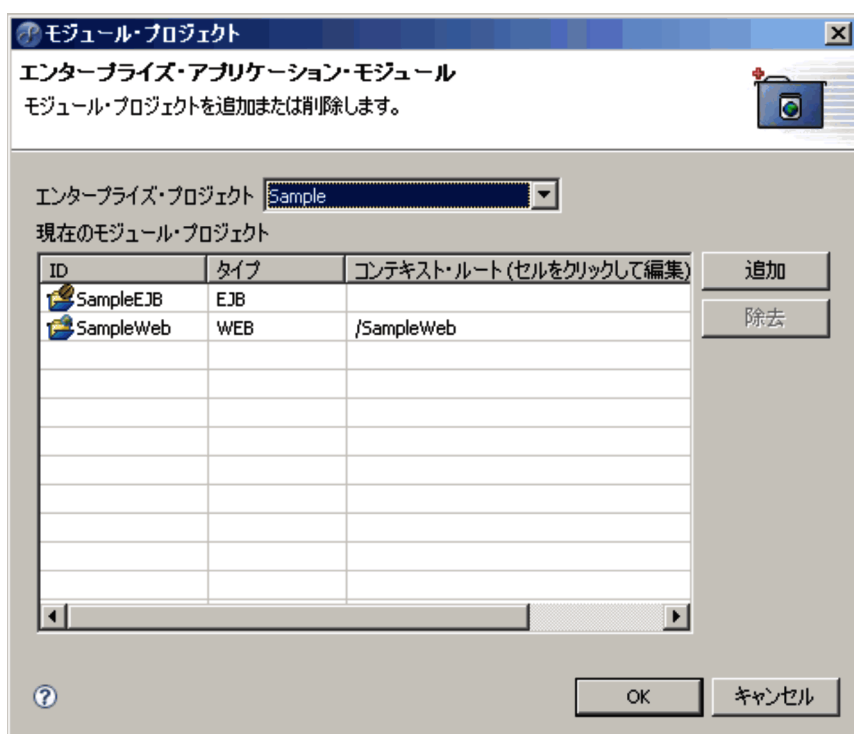
4.4.3 エンタープライズアプリケーションプロジェクトのモジュールの変更手順

エンタープライズアプリケーションプロジェクトを作成したあとで、J2EE アプリケーションのモジュール構成を変更できます。

[モジュール・プロジェクト] ダイアログでモジュールを追加および削除すると、application.xml の <module> タグが自動で追加および削除されます。このとき、application.xml 内のコメント、[MyEclipse XML エディター] などを使用してユーザが手動で追加および編集したタグが削除されます。

次に、J2EE アプリケーションのモジュール構成を変更する手順を示します。

1. [パッケージ・エクスプローラー] ビューで変更対象のエンタープライズアプリケーションプロジェクトを選択して、コンテキストメニューから [MyEclipse] - [モジュールの追加と除去] を選択します。
[モジュール・プロジェクト] ダイアログが表示されます。



2. モジュールプロジェクト (EJB プロジェクトまたは Web プロジェクト) を追加および削除する場合は、次の操作をします。
 - モジュールプロジェクトを追加する場合
[追加] ボタンをクリックして、[新規アプリケーション・モジュール] ダイアログで追加するモジュールプロジェクト (EJB プロジェクトまたは Web プロジェクト)

を選択します。

- モジュールプロジェクトを削除する場合
削除するモジュールプロジェクト（EJB プロジェクトまたは Web プロジェクト）
を選択して，[除去] ボタンをクリックします。

3. [モジュール・プロジェクト] ダイアログの [OK] ボタンをクリックします。
[モジュール・プロジェクト] ダイアログが閉じて，変更内容が反映されます。

参考

[モジュール・プロジェクト] ダイアログで Web プロジェクトのコンテキストルートも変更
できます。

Web プロジェクトのコンテキストルートを変更する場合は，変更対象の Web プロジェクト
の [コンテキスト・ルート] 列を選択して編集します。

ただし，[モジュール・プロジェクト] ダイアログでコンテキストルートを変更した内容は，
エンタープライズアプリケーションプロジェクトの application.xml に反映されますが，
Web プロジェクトのプロパティには反映されません。

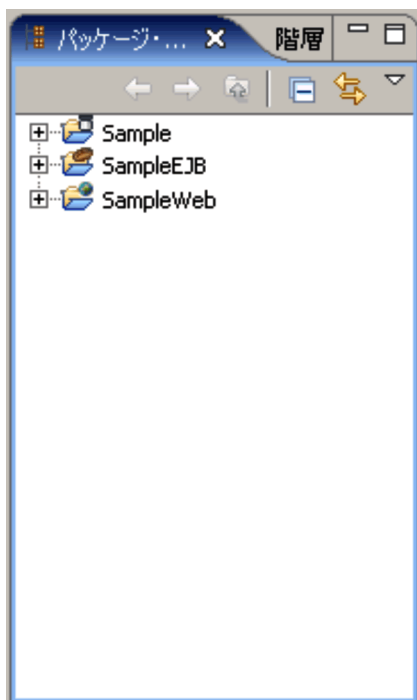
Web プロジェクトのプロパティのコンテキストルートを変更する手順は，「4.3.3 コンテキ
ストルートの変更手順」を参照してください。

4.5 リソースアダプタのインポート

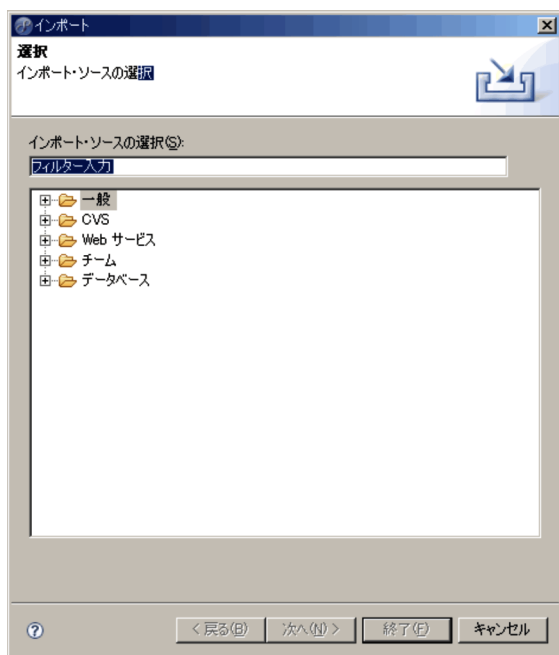
MyEclipse を使った開発では、Cosminexus の提供するリソースアダプタをエンタープライズアプリケーションプロジェクトにインポートできます。ここでは、MyEclipse で作成したエンタープライズアプリケーションプロジェクトにリソースアダプタをインポートする手順を説明します。

なお、リソースアダプタをインポートする前に、必ずエンタープライズアプリケーションプロジェクトを作成してください。エンタープライズアプリケーションプロジェクトの作成手順については、「4.4 エンタープライズアプリケーションプロジェクトの作成」を参照してください。

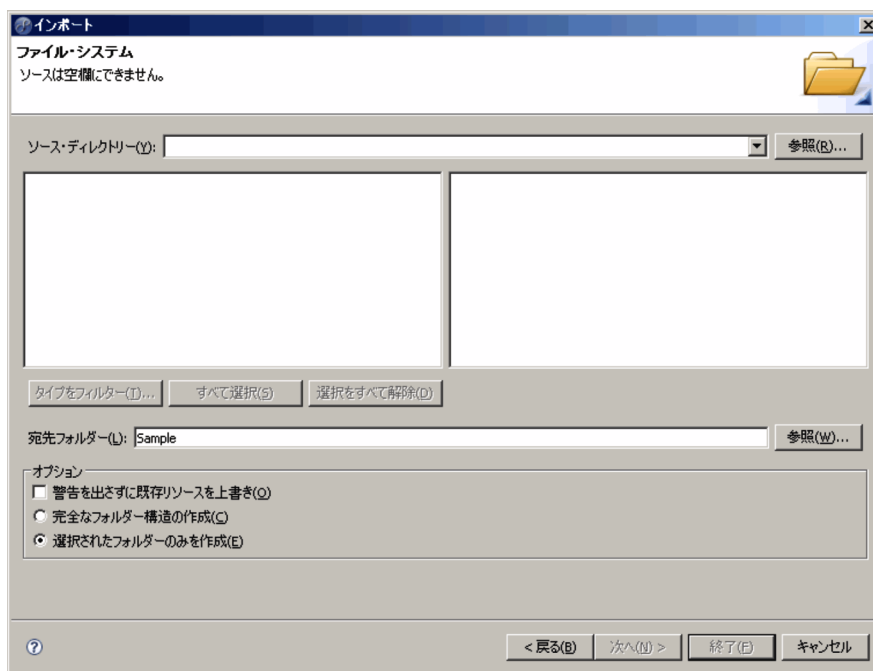
1. Eclipse のメニューから [ウィンドウ] - [ビューの表示] - [パッケージ・エクスプローラー] ビューを選択します。
[パッケージ・エクスプローラー] ビューが表示されます。



2. [パッケージ・エクスプローラー] ビューでインポート先のエンタープライズアプリケーションプロジェクトを選択して、コンテキストメニューから [インポート] を選択します。
[インポート] ダイアログが表示されます。



3. [一般] - [ファイル・システム] を選択して, [次へ] ボタンをクリックします。
[ファイル・システム] ページが表示されます。



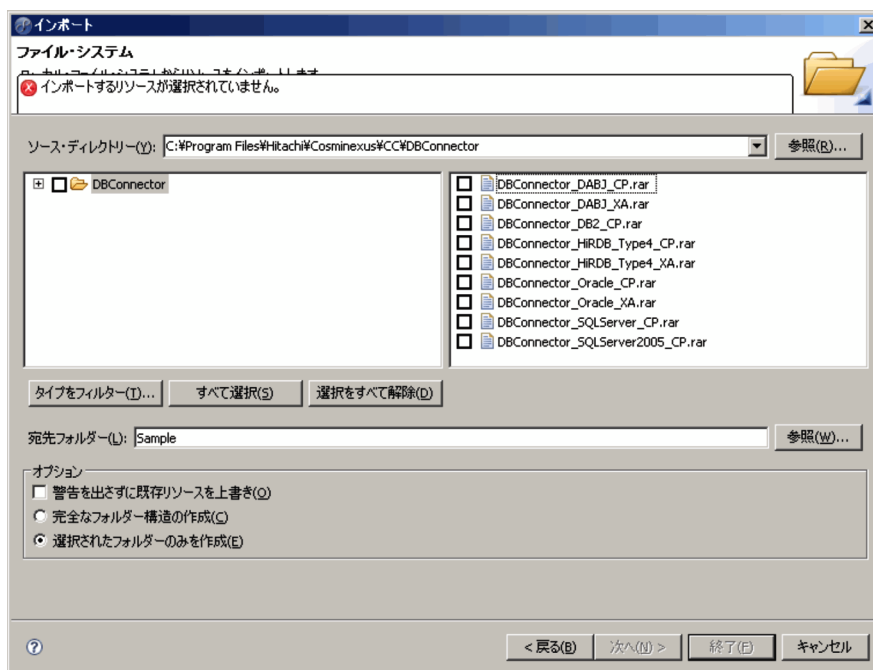
4. [ソース・ディレクトリー] の [参照] ボタンをクリックします。
[ディレクトリーからインポート] ダイアログが表示されます。

4. プロジェクトの作成

5. インポートするリソースアダプタが含まれるディレクトリを選択します。
リソースアダプタは次のディレクトリに格納されています。
<Cosminexus のインストールディレクトリ>\CC\DBConnector\

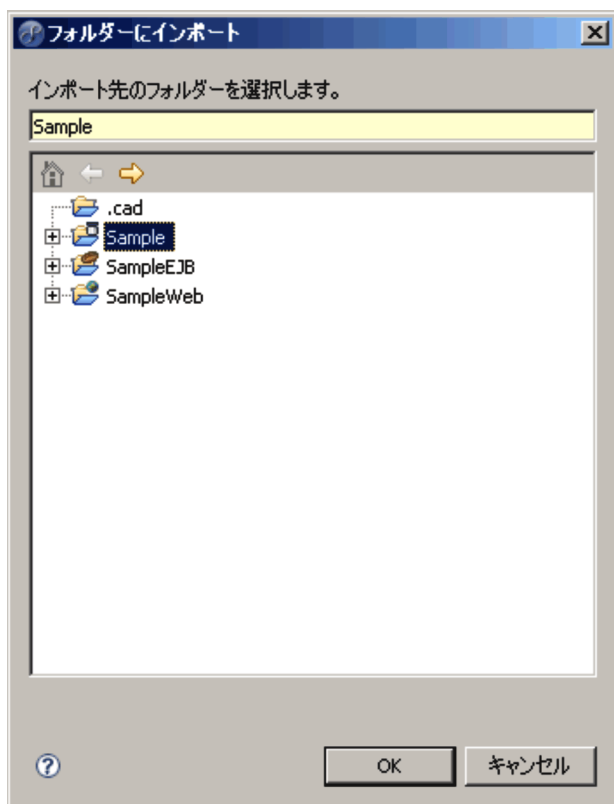


6. [OK] ボタンをクリックします。
[ディレクトリからインポート] ダイアログが閉じて、[ファイル・システム] ページの左のリストボックスに選択したディレクトリが表示されます。右のリストボックスに選択したディレクトリに含まれるファイルの一覧が表示されます。

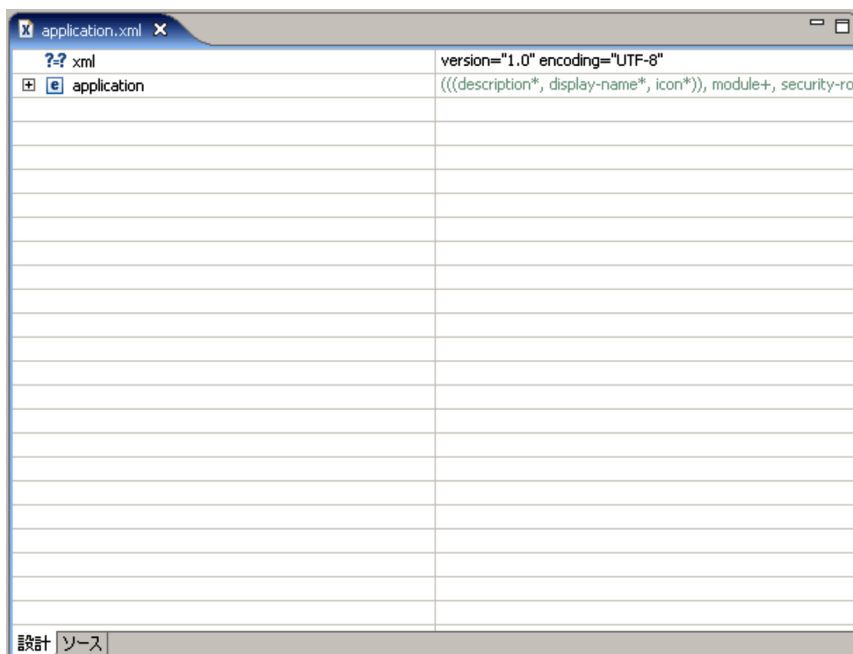


7. インポートするファイルのチェックボックスにチェックを入れます。
8. [宛先フォルダー] の [参照] ボタンをクリックします。
[フォルダーにインポート] ダイアログが表示されます。

4. プロジェクトの作成



9. インポート先のエンタープライズアプリケーションプロジェクトのルートフォルダを選択して、[OK] ボタンをクリックします。
[フォルダーにインポート] ダイアログが閉じます。
10. [ファイル・システム] ページの [終了] ボタンをクリックします。
手順 8. で選択したフォルダに、リソースアダプタがインポートされます。
11. [パッケージ・エクスプローラー] ビューで [< インポート先のエンタープライズアプリケーションプロジェクト >] - [META-INF] - [application.xml] を選択して、コンテキストメニューから [開く] を選択します。
[MyEclipse XML エディター] に application.xml が表示されます。
12. [設計] タブを選択します。
[設計] タブの画面が表示されます。



13. 画面上に表示されている <application> タグを展開します。
14. <module> タグを選択して、コンテキストメニューから [後に追加] - [module] を選択します。
選択した <module> タグの後ろに、新しい <module> タグが追加されます。
15. 追加された <module> タグを展開して、<connector> タグの値をインポートしたりソースアダプタのファイル名に変更します。
16. Eclipse のメニューから、[ファイル] - [保管] を選択します。
application.xml の変更内容が保管されて、エンタープライズアプリケーションプロジェクトにリソースアダプタがインポートされます。

リソースアダプタを J2EE アプリケーションに含めないでインポートする場合は、Server Plug-in を使用する必要があります。Server Plug-in を使用したインポート手順については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

4.6 DD の編集

[MyEclipse XML エディター] を使用して DD を編集します。

ここでは, [MyEclipse XML エディター] の操作方法と各 DD 編集時の注意事項を説明します。また, Cosminexus でサポートする各 DD のバージョンも説明します。

4.6.1 Cosminexus でサポートする DD について

Cosminexus では, 次を示す範囲の DD をサポートします。

表 4-2 Cosminexus でサポートする DD の範囲

XML 種別	バージョン	備考
application.xml	1.2	インポート時に 1.4 にバージョンアップする。
	1.3	インポート時に 1.4 にバージョンアップする。
	1.4	-
ejb-jar.xml	1.1	インポート時に 2.0 にバージョンアップする。
	2.0	-
	2.1	-
web.xml	2.2	インポート時に 2.3 にバージョンアップする。
	2.3	-
	2.4	-
ra.xml	1.0	-
	1.5	-

4.6.2 [MyEclipse XML エディター] の操作方法

DD の編集には [MyEclipse XML エディター] を使用します。

ここでは, [MyEclipse XML エディター] の表示方法, 構成, 基本操作について説明します。

(1) [MyEclipse XML エディター] の表示方法

[MyEclipse XML エディター] の表示方法を説明します。

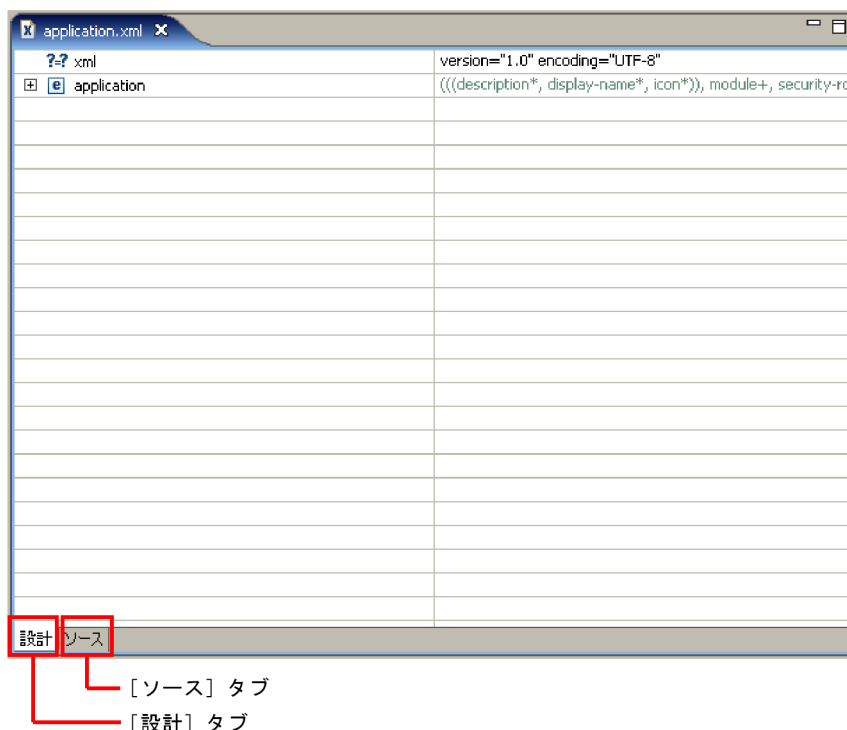
1. [パッケージ・エクスプローラー] ビューで編集対象の DD を選択して, コンテキストメニューから [アプリケーションから開く] - [MyEclipse XML エディター] を選択します。

エディタエリアに [MyEclipse XML エディター] が表示されます。

(2) [MyEclipse XML エディター] の構成

[MyEclipse XML エディター] の構成を説明します。

[MyEclipse XML エディター] は , [設計] タブと [ソース] タブから構成されます。



[設計] タブと [ソース] タブに表示される情報は次のとおりです。

[設計] タブ

XML のタグがテーブルビューで階層表示されます。左列には、タグまたは属性が表示されます。右列には、値または属性値が表示されます。

[ソース] タブ

XML のソースが表示されます。

(3) [MyEclipse XML エディター] の [設計] タブの基本操作

[MyEclipse XML エディター] の基本操作を説明します。ここでは、[MyEclipse XML エディター] の [設計] タブでの XML タグの設定方法を説明します。

1. テーブルビューの左列で、操作対象のタグおよび属性を選択します。
2. コンテキストメニューから、タグおよび属性の操作を選択します。
コンテキストメニューは操作対象によって表示項目が異なります。

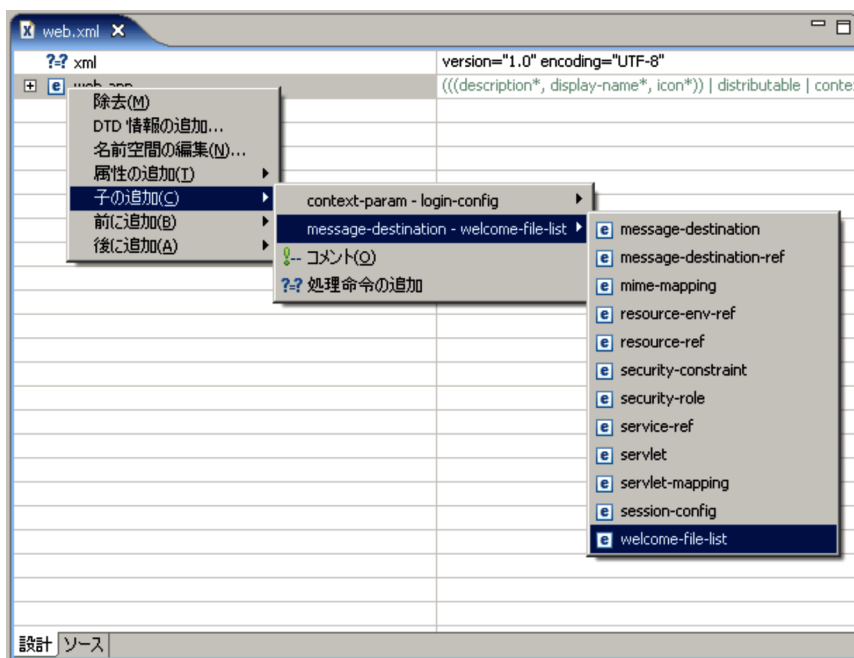
4. プロジェクトの作成

3. テーブルビューの右の列をクリックし、値を入力します。

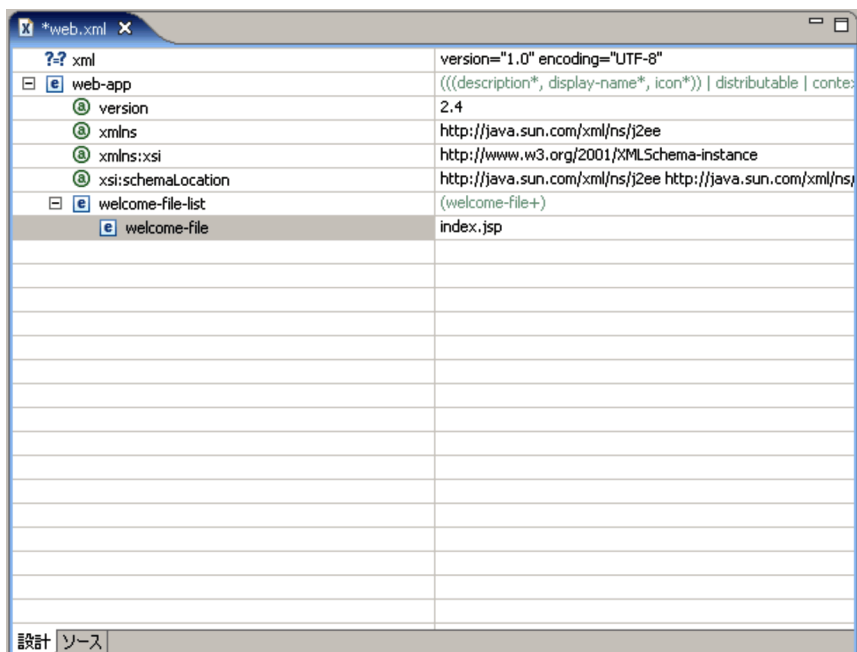
XML が文法的に誤っている場合やスキーマの定義に合わない場合は、Eclipse の [問題] ビューにエラーメッセージが表示されます。[問題] ビューでエラーが表示された行をダブルクリックすると、XML ソースの該当個所が表示されます。

web.xml に <welcome-file-list> タグを挿入する場合の操作例を示します。

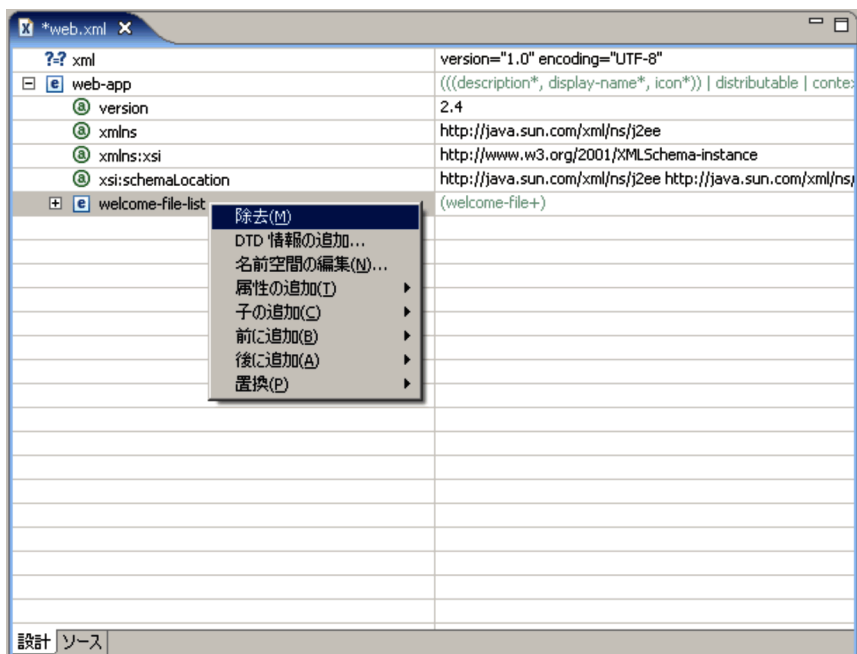
1. [パッケージ・エクスプローラー] ビューで web.xml を選択し、コンテキストメニューから [アプリケーションから開く] - [MyEclipse XML エディター] を選択します。
選択したファイルがエディタエリアに表示されます。
2. 挿入タグの親タグとなる <web-app> タグを選択します。
3. コンテキストメニューから [子の追加] - [message-destination - welcome-file-list] - [welcome-file-list] を選択します。
<welcome-file-list> タグと、その子要素である <welcome-file> タグが挿入されます。



4. <welcome-file> タグの右の列をクリックして、設定する値を入力します。



なお、要素や属性値の削除は、コンテキストメニューの「除去」からできます。



4.6.3 application.xml 編集時の注意事項

ここでは、application.xml 編集時の注意事項を説明します。

(1) <display-name> タグの設定

エンタープライズアプリケーションプロジェクトの作成時に、[新規プロジェクト] ダイアログの [エンタープライズ Java プロジェクトの作成] ページで指定した [プロジェクト名] が入ります。

値を指定しないと Cosminexus コネクタでのデプロイ、およびデプロイ時にエラーが発生するので、必ず値を指定してください。また、デプロイ後に変更すると、J2EE サーバ上の J2EE アプリケーションの入れ替え、およびアンデプロイが適切に実行されません。

また、<display-name> タグの設定値は、作業ディレクトリ中のディレクトリ名として使用されます。作業ディレクトリのパス長が OS の上限に達しないように <display-name> タグを指定してください。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(2) <module> タグの id 属性の設定

<module> タグの id 属性には、MyEclipse で使用する値が自動で設定されます。変更または削除すると誤動作する場合がありますので、設定を変更しないでください。

(3) <ejb> タグの設定

<module> タグ内の <ejb> タグには、エンタープライズアプリケーションプロジェクトに組み込まれている EJB プロジェクトの <プロジェクト名>.jar が設定されます。変更すると誤動作するおそれがあるので、注意してください。

(4) <web-uri> タグの設定

<module> タグ内にある、<web> タグ下の <web-uri> タグには、エンタープライズアプリケーションプロジェクトに組み込まれている Web プロジェクトの <プロジェクト名>.war が設定されます。変更すると誤動作するおそれがあるので、注意してください。

(5) <context-root> タグの設定

<context-root> タグにはコンテキストルートが設定されます。コンテキストルートは、エンタープライズアプリケーションプロジェクトの作成時に、[新規プロジェクト] ダイアログの [Web プロジェクト・モジュールの定義] ページで指定します。コンテキストルートの変更については、「4.4.3 エンタープライズアプリケーションプロジェクトのモジュールの変更手順」を参照してください。

また、<context-root> タグの設定値は、作業ディレクトリ中のディレクトリ名として使

用されます。作業ディレクトリのパス長が OS の上限に達しないように <context-root> タグを指定してください。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(6) DD 作成時の注意事項

各プロジェクトの DD では Processing Instruction, XInclusion, および名前空間接頭辞を記述できません。記述しても、値が正しく読み込まれません。

4.6.4 ejb-jar.xml 編集時の注意事項

ここでは、ejb-jar.xml 編集時の注意事項を説明します。

(1) DTD に従っていない ejb-jar.xml の取り扱い

必須項目の指定がない、または指定順序の不正などで、EJB-JAR の DTD (ejb-jar_1_1.dtd, ejb-jar_2_0.dtd および ejb-jar_2_1.xsd) に従っていない DD は、EJB-JAR のインポート時にエラーとなり、インポートできません。インポートできない場合は、ejb-jar.xml の設定を見直してください。

(2) CMP フィールドおよび CMR フィールドの命名規則

Entity Bean の CMP では、アンダースコア () で始まる CMP フィールド名および CMR フィールド名を指定できません。

(3) EJB QL での 2 バイトコードの使用について

EJB QL では 2 バイトコードを使用しないでください。

(4) セキュリティロールリファレンスの設定について

<security-role-ref> タグに含まれる role-link には、<security-role> タグに含まれる role-name で指定したロール名を指定します。

(5) クエリメソッドのタグを記述するときの注意事項

ejb-jar.xml の <query-method> タグに含まれる <method-name> タグでは、「* (アスタリスク)」を記述できません。<query-method> タグに「*」を指定した ejb-jar.xml を含む EJB-JAR は、インポートできません。

(6) アブストラクトスキーマ名の指定時の注意事項

<abstract-schema-name> タグは、同じファイルにあるほかの <abstract-schema-name> タグで指定する名称や、cmp-field, cmr-field で指定する名称と重複できません。また、<abstract-schema-name> タグには EJB QL の予約語を使用できません。

(7) <display-name> タグ編集時の注意事項

<ejb-jar> タグに含まれる <display-name> タグは作業ディレクトリ中のファイル名として使用されます。作業ディレクトリのパス長が OS の上限に達しないように <display-name> タグを指定してください (デフォルトは EJB-JAR ファイル名を基に付けられます)。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(8) <ejb-client-jar> タグの使用について

Developer では、ejb-jar.xml に記載する DD の <ejb-client-jar> タグの機能をサポートしていません。

(9) Relationship の設定の注意事項

ejb-jar.xml に <relationships> タグを設定する場合は、二つの <ejb-relationship-role> タグのどちらか一方に <cmr-field> タグを設定します。

(10) DD 作成時の注意事項

各プロジェクトの DD では Processing Instruction, XInclusion, および名前空間接頭辞を記述できません。記述しても、値が正しく読み込まれません。

4.6.5 web.xml 編集時の注意事項

ここでは、web.xml 編集時の注意事項を説明します。

(1) セッションタイムアウトの設定時の注意事項

セッションタイムアウトを設定するときの注意事項を示します。

- セッションタイムアウトの設定値
セッションタイムアウトを無期限に設定しないでください。無期限に設定した場合、セッション情報を保持する領域が解放されないため、メモリを消費し続けます。
- web.xml の <session-timeout> タグの指定
web.xml の <session-timeout> タグに指定する値の有効範囲は、-35,791,394 ~ 35,791,394 です。この範囲内の値を指定してください。

(2) アクセスする URL パターンの定義の注意事項

web.xml 上の <servlet> タグに含まれる <init-param>, <load-on-startup>, および <security-role-ref> タグの指定は、<servlet-mapping> タグに定義した URL パターンに該当するサーブレットまたは JSP ファイルにアクセスした場合にだけ有効になります。このため、マッピング定義なしで直接 JSP ファイルのパスを URL 指定した場合、または /servlet/ のプリフィックスでサーブレットクラスを URL に指定して実行した場合は有効になりません。

(3) セキュリティロール使用時の設定について

web.xml の <security-role> タグを使用する場合は、J2EE サーバモードの実行環境で、サーバ管理コマンドでセキュリティロールのリファレンスを解決する必要があります。

(4) エラーページ設定時の注意事項

Servlet 2.2 仕様および Servlet 2.3 仕様の場合、web.xml で <error-page> タグを指定すると、そのエラーページが表示されるときに返されるステータスコードが 200 になります。ステータスコード 401 のエラーページを指定した場合、401 のステータスコードが 200 になってしまうため、Basic 認証と併用できません。Basic 認証を使用する場合には、ステータスコード 401 用のエラーページを指定しないでください。

(5) <run-as> タグと Web コンテナの認証の関連について

web.xml の <servlet> タグ要素に指定する <run-as> タグは、指定されたサーブレットまたは JSP から Enterprise Bean を呼び出すときに使用されるものであり、Web コンテナでの認証とは無関係です。

このため、Web コンテナでの認証結果を参照するための

javax.servlet.http.HttpServletRequest クラスの isUserInRole メソッドや
getUserPrincipal メソッドには影響しません。例えば、サーブレットまたは JSP ファイルから <run-as> タグに記述したロール名を引数に指定した
javax.servlet.http.HttpServletRequest の isUserInRole メソッドを呼び出しても、戻り値は false となります。

(6) <load-on-startup> タグ指定時の注意事項

<load-on-startup> タグに空文字を指定 (<load-on-startup></load-on-startup> または <load-on-startup/> と指定) したサーブレットおよび JSP は、<load-on-startup> タグに 2,147,483,647 を指定されたものとして Web モジュールのデプロイ時にロードされます。

(7) <display-name> タグ編集時の注意事項

<web-app> タグ内に含まれる <display-name> タグは、作業ディレクトリ中のファイル名として使用されます。作業ディレクトリのパス長がプラットフォームで規定されているパス長の上限に達しないように display-name を指定してください (デフォルトは WAR ファイル名を基に付けられます)。作業ディレクトリのパス長の見積もりについては、マニュアル「Cosminexus システム構築ガイド」を参照してください。

(8) web.xml の記述内容とサーバの動作

Servlet 2.3 に対応した J2EE アプリケーションの場合の、web.xml の記述内容とサーバの動作を次の表に示します。

4. プロジェクトの作成

表 4-3 web.xml の記述内容とサーバの動作（1.4 モード）

項番	web.xml の記述内容	サーバの動作
1	プロパティに webserver.xml.validate=false を設定した場合に、DTD の仕様で必須となっているタグを記述しないとき	WAR ファイルのインポート時にエラーとなります。
2	<filter> タグに含まれる <filter-name> タグ要素を空要素にして、<filter-mapping> タグに含まれる <filter-name> タグ要素も空要素にした場合	実行されません。
3	<listener> タグに含まれる <listener-class> タグ要素を空要素にした場合	<listener> タグの指定を無視して正常に実行されます。
4	プロパティに webserver.xml.validate=false を設定した場合に、DTD の仕様で一つだけ指定が許されているタグを複数記述したとき	WAR ファイルのインポート時にエラーになります。
5	要素が同じ <filter-mapping> タグを複数記述した場合	doFilter メソッドは 1 回だけ呼び出されます。
6	親タグ、その下に要素としてキーとなるタグ、およびそのほかの情報を持つタグを複数記述した場合に、キーとなるタグの要素が同じでそのほかの情報が異なるとき	最初に記述された親タグを有効にします。
7	プロパティに webserver.xml.validate=false を設定している場合に、DTD の仕様で規定されていない順序でタグを記述したとき	WAR ファイルのインポート時にエラーになります。
8	<jsp-file> タグに「/」で始まらない文字列を記述した場合	先頭に「/」を付けて、正常に実行されます。
9	次に示す <url-pattern> タグに、「*.」以外の「/」で始まらない文字列を記述した場合 <ul style="list-style-type: none"> <servlet-mapping>-<url-pattern> <security-constraint>-<web-resource-collection>-<url-pattern> 	先頭に「/」を付けて、正常に実行されます。
10	<mime-mapping> タグに含まれる <extension> タグ要素に空文字を記述した場合	WAR ファイルのインポート時にエラーになります。
11	<mime-mapping> タグに含まれる <mime-type> タグ要素に空文字を記述した場合	WAR ファイルのインポート時にエラーになります。
12	<error-page> タグに含まれる <error-code> タグ要素に空文字を記述した場合	WAR ファイルのインポート時にエラーになります。
13	<transport-guarantee> タグに空文字を記述した場合	WAR ファイルのインポート時にエラーになります。
14	<form-login-page> タグまたは <form-error-page> タグに空文字を記述した場合	WAR ファイルのインポート時に設定が無視されます。

項番	web.xml の記述内容	サーバの動作
1 5	<env-entry-type> タグ, <ejb-ref-type> タグ, または <ejb-ref-type> タグに空文字を設定した場合	WAR ファイルのインポート時にエラーになります。
1 6	<security-constraint> を設定して <auth-method> を省略した場合	Basic 認証としてインポートされます。

(9) ゲートウェイ指定機能を使用する場合の注意事項

ゲートウェイ指定機能でスキームを https とみなすように設定した場合, Web サーバへのリクエストが http であっても https とみなされます。したがって, web.xml の <transport-guarantee> タグで INTEGRAL や CONFIDENTIAL を指定しても, https の URL へリダイレクトされません。

(10) 静的コンテンツキャッシュ機能を使用する場合の DD の定義

Cosminexus の Web コンテナでは, 静的コンテンツをメモリにキャッシュする静的コンテンツキャッシュ機能を使用できます。一度アクセスした静的コンテンツの内容をメモリにキャッシュすることで, 二度目以降のアクセス時にファイルシステムへのアクセス回数を減らし, 応答速度を向上できます。

静的コンテンツキャッシュ機能を有効にするためには, DD (web.xml) または Web アプリケーションの属性ファイルに定義を追加します。ここでは, DD に定義する方法を説明します。

DD の定義では, DD の <web-app> タグ内に <context-param> タグを追加して, <context-param> タグ内に <param-name> タグと <param-value> タグを追加します。DD で定義する項目および指定する内容を次に示します。

表 4-4 静的コンテンツキャッシュ機能使用時の DD の定義内容

定義する項目	<param-name> に指定する要素	<param-value> に指定する内容	デフォルト値
静的コンテンツキャッシュ機能の有効 / 無効	com.hitachi.software.web.static_content.cache.enabled	静的コンテンツキャッシュ機能の有効 / 無効を指定します。 true 静的コンテンツキャッシュ機能を有効にします。 false 静的コンテンツキャッシュ機能を無効にします。	false

4. プロジェクトの作成

定義する項目	<param-name> に指定する要素	<param-value> に指定する内容	デフォルト値
Web アプリケーション単位のメモリサイズ	com.hitachi.software.web.static_content.cache.size	<p>静的コンテンツキャッシュ機能を有効にした場合、メモリにキャッシュできるサイズを byte 単位で指定します。</p> <ul style="list-style-type: none"> Web アプリケーション単位で、キャッシュの合計サイズが指定した値を超えた場合は、アクセスされていない時間が最も長いキャッシュから削除されて、キャッシュの合計サイズが設定した値以下になるまでキャッシュの削除を繰り返します。 メモリサイズが設定されていない Web アプリケーションでは、そのプロパティに指定した値が用いられます。しかし、メモリサイズが設定されている Web アプリケーションでは、そのプロパティに指定した値は用いられません。 0 ~ 2147483647 までの整数値で指定します。0 を指定した場合は、Web アプリケーション単位でメモリにキャッシュできるサイズに制限を設けません。 このプロパティに無効な値が設定された場合、およびキャッシュを許可するファイルサイズで指定した値よりも小さい値の場合、デフォルト値が使用されます。 このプロパティに空文字列、または空白文字が設定された場合は、デフォルト値が使用されます。 	10485760
キャッシュを許可するファイルサイズ	com.hitachi.software.web.static_content.cache.filesize.threshold	<p>静的コンテンツキャッシュ機能を有効にした場合、キャッシュできるファイルサイズを byte 単位で指定します。</p> <ul style="list-style-type: none"> 指定した値を超えるサイズのファイルはキャッシュされません。 ファイルサイズが設定されていない Web アプリケーションでは、そのプロパティに指定した値が用いられます。しかし、ファイルサイズが設定されている Web アプリケーションでは、そのプロパティに指定した値は用いられません。 0 ~ 2147483647 までの整数値で指定します。0 を指定した場合は、キャッシュできるファイルのサイズに制限を設けません。 プロパティに無効な値が設定された場合、および Web アプリケーション単位のメモリサイズで指定した値より大きい場合は、デフォルト値が使用されます。 このプロパティに空文字列、または空白文字が設定された場合は、デフォルト値が使用されます。 	524288

！ 注意事項

次に示すパラメタは、静的コンテンツキャッシュ機能で使用するため、DD の
<context-param> タグ内で任意に使用できません。

- com.hitachi.software.web.static_content.cache.enabled
- com.hitachi.software.web.static_content.cache.size
- com.hitachi.software.web.static_content.cache.filesize.threshold

DD の定義例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <context-param>
    <param-name>
      com.hitachi.software.web.static_content.cache.enabled
    </param-name>-
    <param-value>true</param-value>
  </context-param>

  <context-param>
    <param-name>
      com.hitachi.software.web.static_content.cache.size
    </param-name>
    <param-value>5242880</param-value>
  </context-param>

  <context-param>
    <param-name>
      com.hitachi.software.web.static_content.cache.filesize.threshold
    </param-name>
    <param-value>102400</param-value>
  </context-param>
</web-app>
```

定義例では、次の内容が定義されています。

- 静的コンテンツキャッシュ機能を有効にします。
- Web アプリケーション単位のメモリサイズを 5MB にします。
- キャッシュを許可するファイルサイズの上限値を 100KB にします。

(11) <taglib-location> タグに指定したパスの大文字、小文字が異なる場合の動作

web.xml の <taglib-location> タグまたは JSP の taglib ディレクティブに指定したタグライブラリ・ディスクリプタ (TLD ファイル) のパスが、実際のパスと大文字、小文字が異なっている場合、Windows 上では正常に動作しますが、UNIX 上ではエラーになります。

Windows 上で動作していたユーザプログラムを UNIX に移行する場合は、web.xml の <taglib-location> タグまたは JSP の taglib ディレクティブに指定したタグライブラリ・ディスクリプタ (TLD ファイル) のパスが、実際のパスと大文字・小文字が異なっていないか確認してください。

(12) web.xml の DOCTYPE 宣言の注意事項

web.xml で DOCTYPE 宣言に内部サブセットを記述しないでください。J2EE 仕様で定義された DTD / XML スキーマだけを使用してください。

(13) DD 作成時の注意事項

各プロジェクトの DD では Processing Instruction, XInclusion, および名前空間接頭辞を記述できません。記述しても、値が正しく読み込まれません。

4.6.6 Servlet 2.4 仕様で追加, 変更された仕様についての注意事項 (web.xml)

Servlet 2.4 仕様で追加および変更された仕様を持つ web.xml を Cosminexus 上で使用する際の注意事項を示します。Servlet 2.4 仕様については, Servlet 2.4 仕様書を参照してください。

(1) Servlet 2.4 仕様でサポートされない web.xml の要素

Cosminexus では, 次を示す Servlet 2.4 仕様の web.xml の要素は定義できません。定義した場合, デプロイ時にエラーになります。

- message-destination
- message-destination-ref
- service-ref

(2) <security-constraint> タグの設定

Servlet 2.4 仕様では, web.xml の <security-constraint> タグに特別な指定をした場合の動作について追記されています。

Cosminexus では, Web アプリケーションのバージョンに関係なく, 次の動作をします。

- <auth-constraint> タグのサブ要素, <role-name> タグに「*」(アスタリスク)を指定した場合, すべてのロールを許可します。
- <auth-constraint> タグを指定しない場合, Web コンテナは認証しないでリクエストを許可します。
- <transport-guarantee> タグを指定しない場合, Web コンテナはどのような接続も受け付けます。

(3) <security-constraint> タグの複数定義

Servlet 2.4 仕様では, web.xml の <security-constraint> タグを複数定義した場合の動作について追記されています。

Cosminexus 上で <security-constraint> タグを複数定義した場合の動作を, Servlet 2.4

および Servlet 2.3 に分けて示します。

Servlet 2.4

web.xml に `<security-constraint>` タグを複数定義した場合、アクセス制御の対象となる `<security-constraint>` タグは、Servlet 2.4 仕様書に記述された規則に従い選択されます。

Servlet 2.3

web.xml に定義された `<security-constraint>` タグを、ファイルの上部に記述されたものから順に確認し、リクエストの URI、および HTTP メソッドとマッチする `<security-constraint>` タグを使用してアクセス制御します。`<http-method>` タグを定義していない場合、すべての HTTP メソッドを対象とします。

(4) `<url-pattern>` の改行コード

Servlet 2.4 仕様では、web.xml に記述する URL パターンに改行コードを含む場合の動作について追記されています。

Cosminexus では、Web アプリケーションのバージョンに関係なく、KDJE39304-W の警告メッセージが出力されます。ただし、エラーにはならないでアプリケーションは開始されます。また、該当するマッピングは無視されます。

なお、web.xml でこのような動作をするタグは、`<servlet-mapping>` タグ、`<filter-mapping>` タグ、`<jsp-property-group>` タグ、および `<web-resource-collection>` タグと各タグに含まれるサブクラスの `<url-pattern>` タグです。

(5) 指定したエラーページが表示されたレスポンスのステータスコード

Servlet 2.4 仕様に対応した Web アプリケーションでは、web.xml によって表示するエラーページを指定した場合でも、エラーが発生した時点のレスポンスのステータスコードがクライアントに送信されます。

また、Servlet 2.2 仕様および Servlet 2.3 仕様に対応したアプリケーションでは、ステータスコード 200 がクライアントに送信されます。

(6) Web コンテナ単位でのエラーページのカスタマイズ

Servlet 2.4 仕様に対応したアプリケーションでは、web.xml で指定したエラーページを表示させる場合、エラーページはカスタマイズされません。また、同じステータスコードに対してカスタマイズしたエラーページを指定した場合、Web アプリケーションで問題が発生し、web.xml で指定したエラーページの出力に失敗したときだけ、カスタマイズが有効となります。

エラーページのカスタマイズについては、マニュアル「Cosminexus 機能解説」を参照してください。

(7) リダイレクトによるエラーページの生成

Servlet 2.4 仕様に対応したアプリケーションでは、web.xml で指定したエラーページが表示されたあとのステータスコードが、エラーページの生成を委任するエラーステータスコードと一致すると、Web サーバによって生成されるエラーページが有効となります。このとき、web.xml で指定したエラーページの内容はリダイレクトに転送されたあとに破棄されます。

web.xml で指定したエラーページでレスポンスのステータスコードを 200 に変更することで、web.xml で指定したエラーページを有効にできます。

ステータスコードに対応するリダイレクトについては、マニュアル「Cosminexus 機能解説」を参照してください。

(8) フィルタ機能を使用する場合の定義

Servlet 2.4 に対応した Web アプリケーションでは、web.xml の <filter-mapping> タグを定義するとき、サブ要素として <dispatcher> タグを記述することで、リクエストのフォワード時、インクルード時、および web.xml に記述したエラーページへの転送時にフィルタを動作させることができます。

JSP では、page ディレクティブの errorPage 属性を使用することで、JSP での例外発生時にエラーページを出力できます。ただし、このときに実行されるリクエストの転送はフォワードです。フィルタを適用する場合に <dispatcher> タグへ必要となる定義は「ERROR」ではなく、「FORWARD」となります。

5

ユーザインタフェースの作成

MyEclipse では、テンプレートを使用してユーザインタフェースを開発できます。この章では、MyEclipse のテンプレートから JSP ファイルおよびサーブレットを作成する手順を説明します。また、JSP ファイルの編集方法、サーブレットの情報を web.xml に反映する手順についても説明します。

5.1 ユーザインタフェースの開発の流れ

5.2 JSP ファイルの作成

5.3 JSP ファイルの編集

5.4 サーブレットの作成・編集

5.1 ユーザインタフェースの開発の流れ

MyEclipse では、テンプレートを使用してユーザインタフェースを開発できます。テンプレートを使用して作成できるファイルは次のとおりです。

- JSP ファイル
- サーブレット

MyEclipse を使用した JSP ファイル、およびサーブレットの作成の流れを説明します。

図 5-1 JSP ファイルおよびサーブレットの作成の流れ

作業内容	参照先
1. JSPファイルの作成	5.2
↓	
2. JSPファイルの編集	5.3
↓	
3. サーブレットの作成・編集	5.4

それぞれの作業の概要を説明します。

1. JSP ファイルの作成

MyEclipse のテンプレートを使用して、JSP ファイルを作成します。

2. JSP ファイルの編集

[MyEclipse ビジュアル JSP デザイナー] を使用して、JSP ファイルを編集します。

3. サーブレットの作成・編集

MyEclipse のテンプレートからサーブレット作成して、サーブレットの情報を web.xml に反映します。また、作成したサーブレットに必要な処理を実装します。

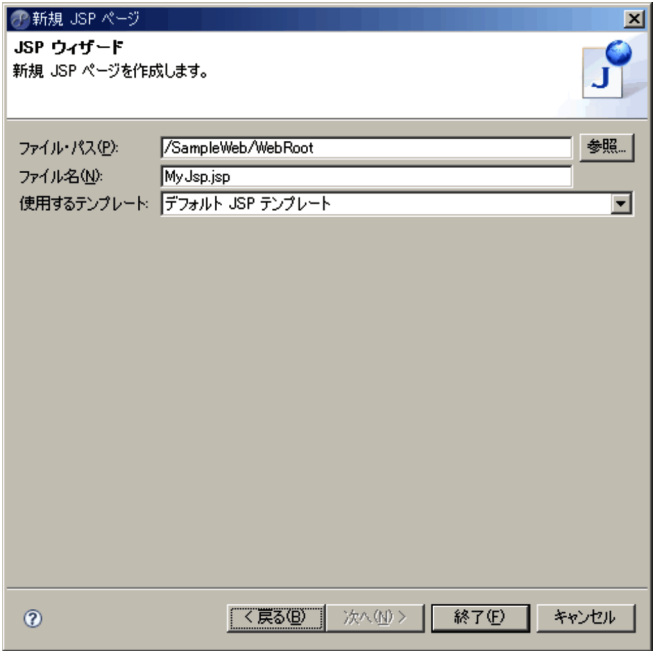
以降の節では、この流れに沿ってユーザインタフェースの作成手順を説明します。

5.2 JSP ファイルの作成

MyEclipse では、テンプレートを使用して JSP ファイルを作成できます。

ここでは、MyEclipse のテンプレートを使用した JSP ファイルの作成手順を説明します。

1. Eclipse のメニューから [ファイル] - [新規] - [その他] を選択します。
[新規] ダイアログが表示されます。
2. [MyEclipse] - [Web] - [JSP (拡張テンプレート)] を選択して、[次へ] ボタンをクリックします。
[新規 JSP ページ] ダイアログが表示されます。



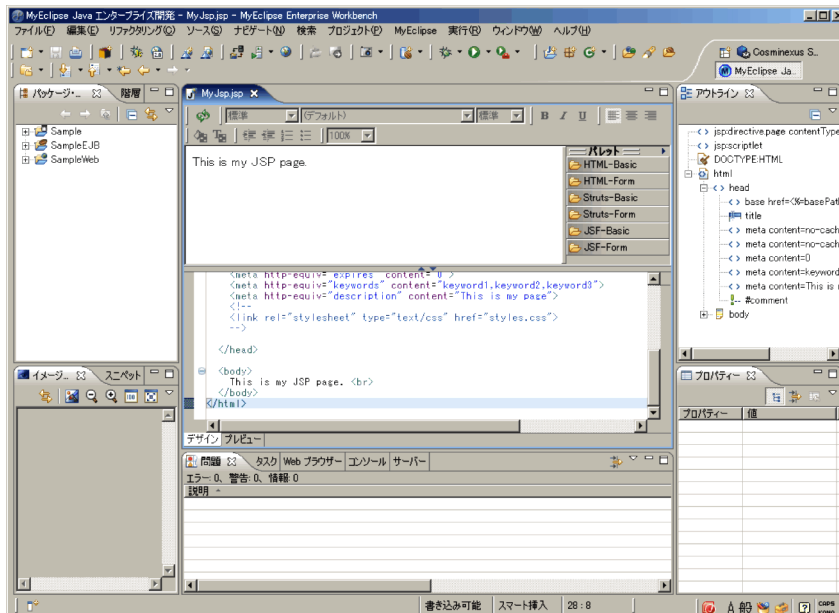
設定が必要な項目を次に示します。

項目名	設定値
ファイル・パス	作成する JSP ファイルの保存先のパスを指定します。プロジェクトのルートを含むパスを指定してください。
ファイル名	作成する JSP ファイルのファイル名を入力します。 ただし、JSP ファイルの名前は重複しないようにしてください。同じプロジェクトに同名の JSP ファイルがある場合、JSP 事前コンパイルを実施したあとにデバッグできません。
使用するテンプレート	[デフォルト JSP テンプレート] を選択します。

3. [終了] ボタンをクリックします。

5. ユーザインタフェースの作成

作成した JSP ファイルがエディタエリアに表示されます。



5.3 JSP ファイルの編集

テンプレートを使用して作成した JSP ファイルを編集します。JSP ファイルの編集には MyEclipse の [MyEclipse ビジュアル JSP デザイナー] を使用できます。[MyEclipse ビジュアル JSP デザイナー] を使用すると、ブラウザで表示イメージを確認しながら JSP ファイルを編集できます。

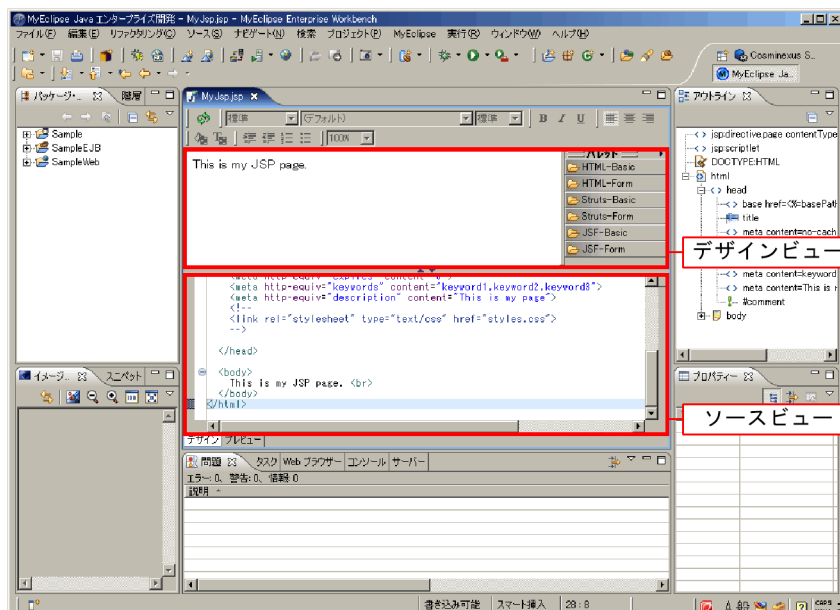
ここでは、[MyEclipse ビジュアル JSP デザイナー] を使用した JSP ファイルの編集手順を説明します。

1. [パッケージ・エクスプローラー] ビューで JSP ファイルを選択して、コンテキストメニューから [アプリケーションから開く] - [MyEclipse ビジュアル JSP デザイナー] を選択します。

JSP ファイルがエディタエリアに表示されます。

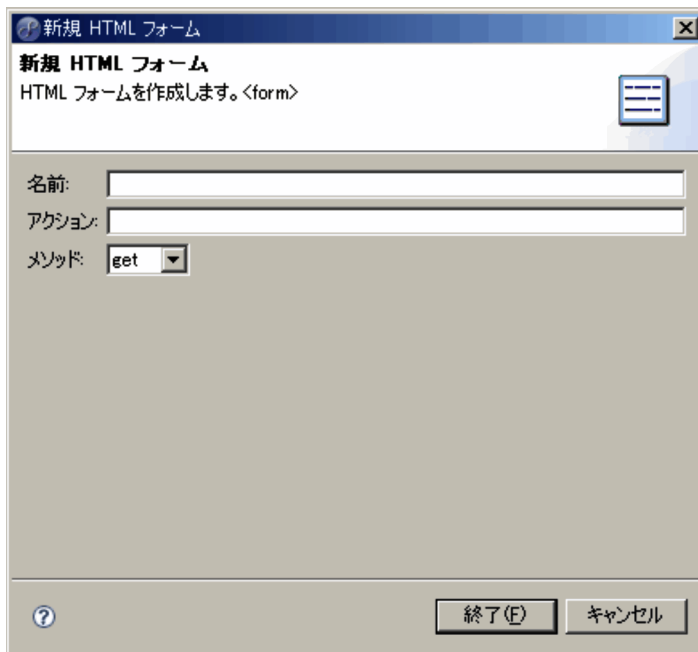
2. [デザイン] タブを選択します。

[デザイン] タブでは、デザインビュー、ソースビュー、および [パレット] が表示されます。

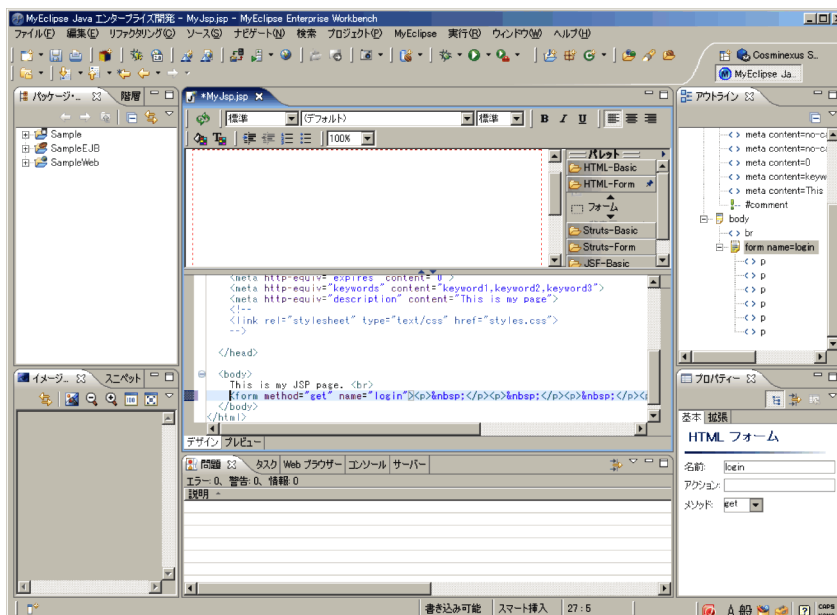


3. [パレット] で追加するタグのパレットを展開します。
 4. デザインビューまたはソースビューでタグの挿入位置をクリックして、[パレット] から挿入するタグを選択します。
- 選択したタグの値を設定するダイアログが表示されます。ここでは、[HTML-FORM] - [フォーム] を選択しています。

5. ユーザインタフェースの作成

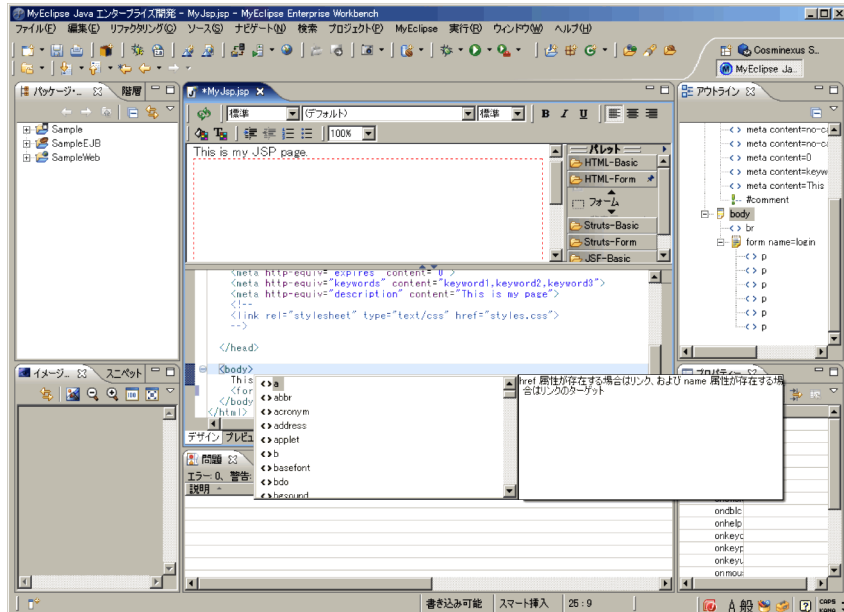


5. 表示されたダイアログで値を指定して、[終了] ボタンをクリックします。
指定した位置にタグが追加されます。



6. タグの属性値を指定する場合は、ソースビューで属性値を指定する個所にカーソルを合わせて、[ctrl] + [space] キーを押します。
属性の一覧が表示されます。

7. 指定する属性をダブルクリックします。
カーソル位置に選択した属性が挿入されます。挿入された属性に値を設定します。



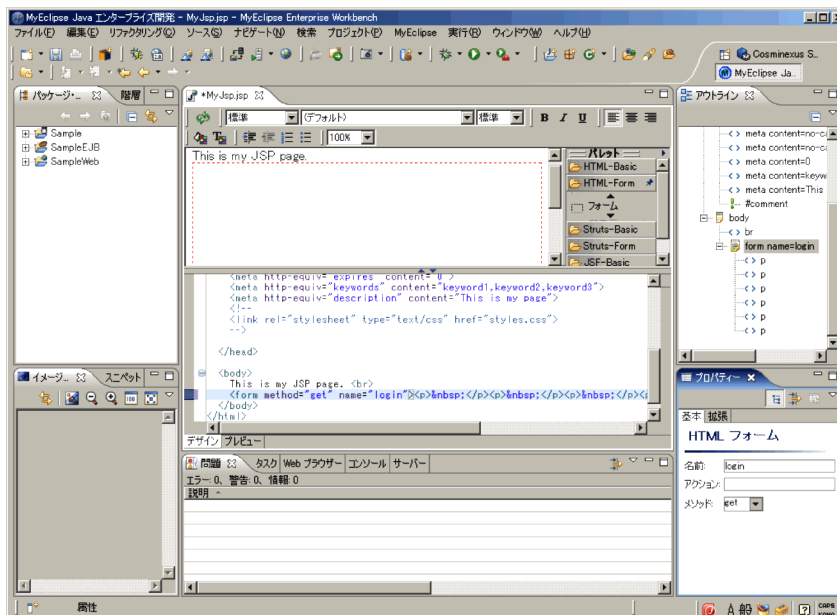
8. 手順 3. ~ 7. を繰り返して、JSP ファイルを編集します。
9. JSP ファイルの編集が終了したら、[プレビュー] タブを選択して、ブラウザでの表示イメージを確認します。

参考

手順 6. で実施しているタグの属性値の指定は、[MyEclipse ビジュアル JSP デザイナー] の [プロパティ] ビューでもできます。[プロパティ] ビューでタグの属性値を指定する手順を次に説明します。

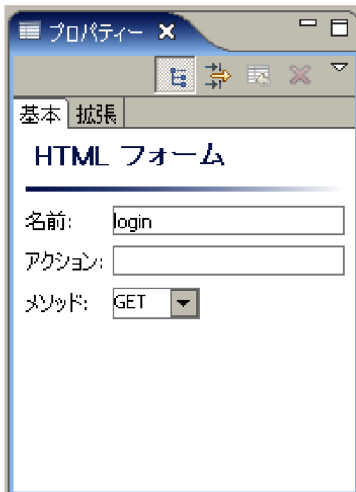
1. Eclipse のメニューから [ウィンドウ] - [ビューの表示] - [プロパティ] を選択します。
[プロパティ] ビューが表示されます。

5. ユーザインタフェースの作成

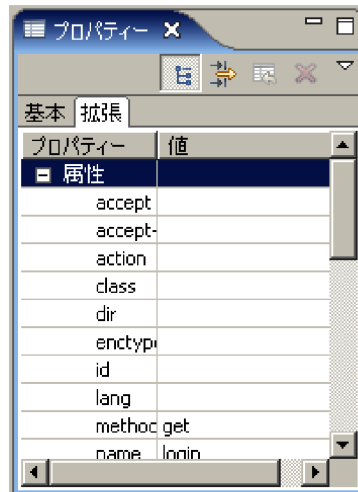


2. 編集対象のタグを選択します。

[プロパティ] ビューに選択したタグのプロパティが表示されます。なお，
[プロパティ] ビューには，[基本] タブと [拡張] タブがあります。



〔基本〕タブ



〔拡張〕タブ

3. [基本] タブの入力フォームが [拡張] タブで [値] 列に値を指定します。

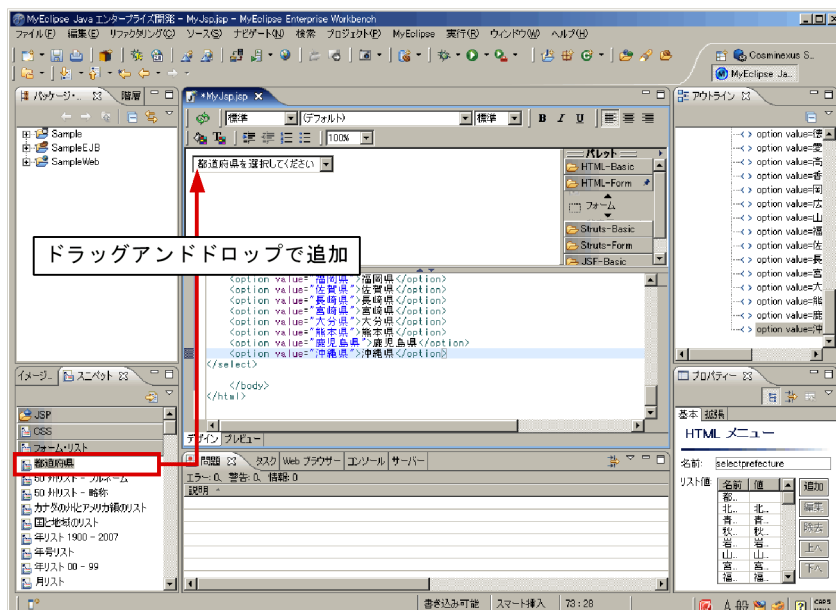
5.3.1 スニペットを使用した JSP ファイルの編集

スニペットは，汎用的なソースコードベースの部品です。MyEclipse では，JSP に配置できるスニペットを提供します。スニペットを使用すると効率良く，JSP ファイルを編

集めます。

ここでは、スニペットを使用した JSP ファイルの編集を説明します。

スニペットは、[MyEclipse Java エンタープライズ開発] パースペクティブの [スニペット] ビューから [MyEclipse ビジュアル JSP デザイナー] にドラッグアンドドロップで追加できます。



[スニペット] ビューは、スニペットを次のカテゴリに分類して表示します。

表 5-1 MyEclipse のスニペットカテゴリ

カテゴリ	説明
JSP	JSP コンポーネント（コメント、ディレクティブ、スクリプトレットなど）のスニペットを提供します。
CSS	カスケードスタイルシートのスニペットを提供します。
フォーム・リスト	都道府県、年月日などの各種リストのスニペットを提供します。
フォーム・エレメント	送信画面フォームを作成するのに便利なスニペットを提供します。
HTML	Web 画面をデザインするのに便利なスニペットを提供します。
JavaScript	JavaScript のスニペットを提供します。
JSF-F	JSF コアタグライブラリのスニペットを提供します。
JSF-H	JSF HTML タグライブラリのスニペットを提供します。
Struts-Bean	Struts Bean タグライブラリのスニペットを提供します。

5. ユーザインタフェースの作成

カテゴリ	説明
Struts-HTML	Struts HTML タグライブラリのスニペットを提供します。
Struts-Logic	Struts Logic タグライブラリのスニペットを提供します。
Struts-Tiles	Struts Tiles タグライブラリのスニペットを提供します。

MyEclipse は、カテゴリごとに次のスニペットを提供します。

表 5-2 MyEclipse が提供するスニペット

カテゴリ	スニペット	説明
JSP	<%--...--%> comment	JSP コメントです。
	<%..%> scriptlet	JSP スクリプトレットです。
	<%=..%> expression	JSP 式です。
	<%!..%> declaration	JSP 宣言です。
	include directive	インクルードディレクティブです。
	page directive	ページディレクティブです。
	taglib directive	taglib ディレクティブです。
CSS	コメントの追加	複数行の CSS コメントです。
	セルの背景色の変更	マウスをあてた時にテーブルのセルの背景色を変更します。
	スタイル・クラスの追加	スタイルのクラスです。

カテゴリ	スニペット	説明
フォーム・リスト	都道府県	都道府県のリスト・メニューです。
	50 州リスト・フルネーム	アメリカ 50 州のリスト・メニューです。 州のフルネームを選択すると、郵便コードの略称が値になります。
	50 州リスト・略称	アメリカ 50 州のリスト・メニューです。 州名も値も郵便コードの略称です。
	カナダの州とアメリカ領のリスト	カナダの州とアメリカの領土のリスト・メニューです。
	国と地域のリスト	65 の国 / 地域のリスト・メニューです。
	年リスト 1900 - 2007	4 桁の年を返す 1900 年から 2007 年までのリスト・メニューです。
	年号リスト	年号のリスト・メニューです。
	年リスト 00 - 99	2 桁の年を返す 00 年から 99 年までのリスト・メニューです。
	月リスト	対応する数値を返す月のリスト・メニューです。
	曜日リスト	対応する数値を返す曜日のリスト・メニューです。
	タイムゾーン・リスト	グリニッジ標準時タイムゾーンのリスト・メニューです。
	時間リスト・AM/PM	真夜中から 11:45 (AM/PM) までの 15 分毎の時刻のリスト・メニューです。
	24 時間制の時刻	24 時間制の時刻を秒まで設定する三つのリスト・メニューのグループです。
	通貨リスト	各国の通貨を表示するリスト・メニューです。値はその国の通貨記号です。
フォーム・エレメント	1 フィールド・フォーム	一つのフィールドがあるフォームです。
	2 フィールド・フォーム	二つのフィールドとボタンを含むフォームです。
	自動クリアされるテキスト・フィールド	デフォルト・メッセージが表示されるフォーム・テキスト・フィールドです。
	キャンセル・ボタン	一つ前のページに戻ります。
	ファイル参照ボタン	ビジターが指定したローカル・ファイルのアップロードを許可するファイル名フィールドです。

5. ユーザインタフェースの作成

カテゴリ	スニペット	説明
HTML	メタタグ	ヘッダーと完全なメタタグ・リストです。
	コメントの追加	複数行のコメントです。
	単一表	1 行 1 列の表です。
	リストのある表	リストを持つ 2 列のテーブル (2 列 3 行) です。
	ページ・ヘッダー	左にロゴ, 右にリンクが表示されるページ・ヘッダーです。
	ダイナミック・ページ・ヘッダー	日付, 時刻などの情報を使用した動的なヘッダーです。次の表示項目を挿入して, カスタマイズできます。 <ul style="list-style-type: none"> • 時刻に合わせたあいさつ • 特別な日のメッセージ • クイック・ジャンプ・ナビゲーション・メニュー
	フッター・ページ番号	ページ番号ナビゲーションを提供します。
	フッター・ナビゲーションと著作権	リンクと著作権を持つフッターです。

カテゴリ	スニペット	説明
JavaScript	お気に入り追加	指定されたページをお気に入りに自動追加します。サイトとタイトルを変更して使用してください。
	前に戻る	キャンセルして、前のページへ戻るボタンと同様です。
	次に進む	1 ページ先へ進みます。
	すべてのチェックボックスを選択	1 クリックでチェックボックス・グループを選択します（同じページ上に複数グループを持てます）。
	曜日リダイレクト	曜日に応じて異なるページへ送ります。例えば、今日が火曜日の場合、Tuesday.html にリダイレクトします。
	一定時間後に URL へ移動	一定時間後に URL へ移動します。
	リダイレクト	指定された URL にリダイレクトします。
	Cookie の読み取り	指定した名前の Cookie の値を読み取る関数です。
	Cookie の書き込み	Cookie に値を書き込む関数です。
	現在の日付	YYYY/MM/DD 形式で現在の日付を表示します。
	GMT ライブ日付 / 時刻	秒単位で変化するライブ日付 / 時刻を表示します。
	メッセージ・ウィンドウ	指定したメッセージとタイトルを含む新規ウィンドウを表示します。
	ポップアップ・ウィンドウ	指定したウィンドウの位置（左および上部）、幅、高さ、URL で新規ウィンドウを開きます。
	ウィンドウを閉じる	現在のウィンドウを閉じるフォーム・ボタンです。

5. ユーザインタフェースの作成

カテゴリ	スニペット	説明
JSF-F	<f:actionListener ../> タグ	JSF の f:actionListener タグです。
	<f:convertDateTime ../> タグ	JSF の f:convertDateTime タグです。
	<f:convertNumber ../> タグ	JSF の f:convertNumber タグです。
	<f:facet ../> タグ	JSF の f:facet タグです。
	<f:selectItems ../> タグ	JSF の f:selectItems タグです。
	<f:validateDoubleRange ../> タグ	JSF の f:validateDoubleRange タグです。
	<f:validateLength ../> タグ	JSF の f:validateLength タグです。
	<f:validateLongRange ../> タグ	JSF の f:validateLongRange タグです。
	<f:valueChangeListener ../> タグ	JSF の f:valueChangeListener タグです。
	<f:verbatim> タグ	JSF の f:verbatim タグです。

カテゴリ	スニペット	説明
JSF-H	<h:commandButton ../> タグ	JSF の h:commandButton タグです。
	<h:commandLink ../> タグ	JSF の h:commandLink タグです。
	<h:dataTable ../> タグ	JSF の h:dataTable タグです。
	<h:form> タグ	JSF の h:form タグです。
	<h:graphicImage ../> タグ	JSF の h:graphicImage タグです。
	<h:inputHidden ../> タグ	JSF の h:inputHidden タグです。
	<h:inputSecret ../> タグ	JSF の h:inputSecret タグです。
	<h:inputText ../> タグ	JSF の h:inputText タグです。
	<h:inputTextarea ../> タグ	JSF の h:inputTextarea タグです。
	<h:message ../> タグ	JSF の h:message タグです。
	<h:outputFormat ../> タグ	JSF の h:outputFormat タグです。
	<h:outputLabel ../> タグ	JSF の h:outputLabel タグです。
	<h:outputLink ../> タグ	JSF の h:outputLink タグです。
	<h:panelGrid ../> タグ	JSF の h:panelGrid タグです。
	<h:selectBooleanCheckbox ../> タグ	JSF の h:selectBooleanCheckbox タグです。
	<h:selectManyCheckbox ../> タグ	JSF の h:selectManyCheckbox タグです。
	<h:selectManyListbox ../> タグ	JSF の h:selectManyListbox タグです。
	<h:selectManyMenu ../> タグ	JSF の h:selectManyMenu タグです。
	<h:selectOneListbox ../> タグ	JSF の h:selectOneListbox タグです。
	<h:selectOneMenu ../> タグ	JSF の h:selectOneMenu タグです。
	<h:selectOneRadio ../> タグ	JSF の h:selectOneRadio タグです。
	<h:column> タグ	JSF の h:column タグです。
Struts-Bean	<bean:write ../> タグ	Struts の bean:write タグです。
	<bean:parameter ../> タグ	Struts の bean:parameter タグです。
	<bean:message ../> タグ	Struts の bean:message タグです。
	<bean:define ../> タグ	Struts の bean:define タグです。

5. ユーザインタフェースの作成

カテゴリ	スニペット	説明
Struts-HTML	<html:base /> タグ	Struts の html:base タグです。
	<html:cancel /> タグ	Struts の html:cancel タグです。
	<html:checkbox ../> タグ	Struts の html:checkbox タグです。
	<html:errors ../> タグ	Struts の html:errors タグです。
	<html:file ../> タグ	Struts の html:file タグです。
	<html:form ../> タグ	Struts の html:form タグです。
	<html:hidden ../> タグ	Struts の html:hidden タグです。
	<html:html> タグ	Struts の html:html タグです。
	<html:image ../> タグ	Struts の html:image タグです。
	<html:img ../> タグ	Struts の html:img タグです。
	<html:link ../> タグ	Struts の html:link タグです。
	<html:messages ../> タグ	Struts の html:messages タグです。
	<html:multibox ../> タグ	Struts の html:multibox タグです。
	<html:option ../> タグ	Struts の html:option タグです。
	<html:optionsCollection ../> タグ	Struts の html:optionsCollection タグです。
	<html:password ../> タグ	Struts の html:password タグです。
	<html:radio ../> タグ	Struts の html:radio タグです。
	<html:reset /> タグ	Struts の html:reset タグです。
	<html:select ../> タグ	Struts の html:select タグです。
	<html:submit /> タグ	Struts の html:submit タグです。
	<html:text ../> タグ	Struts の html:text タグです。
	<html:textarea.../> タグ	Struts の html:textarea タグです。

カテゴリ	スニペット	説明
Struts-Logic	<logic:iterate ../> タグ	Struts の logic:iterate タグです。
	<logic:redirect ../> タグ	Struts の logic:redirect タグです。
	<logic:empty ../> タグ	Struts の logic:empty タグです。
	<logic:notEmpty ../> タグ	Struts の logic:notEmpty タグです。
	<logic:present /> タグ	Struts の logic:present タグ
	<logic:notPresent /> タグ	Struts の logic:notPresent タグ
	<logic:equal ../> タグ	Struts の logic:equal タグです。
	<logic:notEqual ../> タグ	Struts の logic:notEqual タグです。
	<logic:lessEqual ../> タグ	Struts の logic:lessEqual タグです。
	<logic:lessThan ../> タグ	Struts の logic:lessThan タグです。
	<logic:greaterEqual ../> タグ	Struts の logic:greaterEqual タグです。
	<logic:greaterThan ../> タグ	Struts の logic:greaterThan タグです。
Struts-Tiles	<tiles:insert../>tag	Struts の tiles:insert タグです。
	<tiles:put../>tag	Struts の tiles:put タグです。
	<tiles:get../>tag	Struts の tiles:get タグです。
	<tiles:getAsString../>tag	Struts の tiles:getAsString タグです。
	<tiles:useAttribute../>tag	Struts の tiles:useAttribute タグです。

注意

スニペットは、Eclipse のワークスペースごとに保持されます。スニペットは、ワークスペース作成時に、次のファイルに保存されます。

```
<Eclipseのワークスペースディレクトリ>/ .metadata/ .plugins/
org.eclipse.wst.common.snippets/user.xml
```

なお、MyEclipse のバージョンアップで MyEclipse が提供するスニペットに変更があった場合は、既存のワークスペースに反映されません。既存のワークスペースにスニペットの変更を反映する場合は、新しいワークスペースを指定して MyEclipse を起動したあとに新しいワークスペースのスニペットをエクスポートして、既存のワークスペースにインポートする必要があります。

既存のワークスペースでスニペットをカスタマイズしている場合は、インポート時に表示される [既存ドロワー・エントリーの上書き] ダイアログで [キャンセル] ボタンをクリックすると、カスタマイズしたスニペットを残せません。

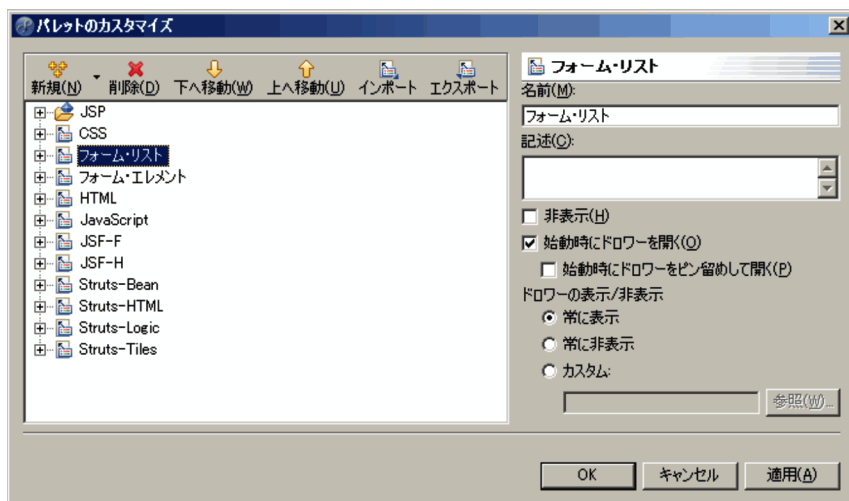
インポートの手順を次に示します。

1. 日本語リソースのスニペットが追加されているワークスペースで [スニペット] ビューの項目を選択して、コンテキストメニューから [カスタマイズ] を選択します。

[パレットのカスタマイズ] ダイアログが表示されます。

5. ユーザインタフェースの作成

2. [パレットのカスタマイズ] ダイアログでエクスポートしたいスニペットのカテゴリを選択します。



3. [パレットのカスタマイズ] ダイアログのツールバーにある [エクスポート] をクリックして、xml ファイルにエクスポートします。
4. インポート先のワークスペースを開いて、[スニペット] ビューの項目を選択して、コンテキストメニューから [カスタマイズ] を選択します。
[パレットのカスタマイズ] ダイアログが表示されます。
5. [パレットのカスタマイズ] ダイアログのツールバーにある [インポート] をクリックして、手順 3. でエクスポートした xml ファイルを指定します。
6. [既存ドロワー・エントリーの上書き] ダイアログで [OK] ボタンをクリックします。
7. [パレットのカスタマイズ] ダイアログの [OK] ボタンをクリックします。

5.4 サブレットの作成・編集

MyEclipse では、テンプレートを使用してサブレットを作成できます。また、サブレットの作成とあわせて web.xml にサブレットの情報を反映できます。

ここでは、テンプレートを使用したサブレットの作成手順を説明します。なお、この手順には、web.xml に作成したサブレットの情報を反映する手順も含まれています。

1. Eclipse のメニューから [ファイル] - [新規] - [その他] を選択します。
[新規] ダイアログが表示されます。
2. [MyEclipse] - [Web] - [サブレット] を選択して、[次へ] ボタンをクリックします。
[新規サブレット] ダイアログが表示されます。

新規サブレット

サーブレット・ウィザード
新規サブレット・クラスを作成します。

ソース・フォルダー(D): 参照(O)...

パッケージ(K): (デフォルト) 参照(W)...

☐ エンクロージング型(Y): 参照(W)...

名前(M):

修飾子: ☒ public(P) ☐ default(U) ☐ private(V) ☐ protected(T)
☐ abstract(I) ☐ final(L) ☐ static(S)

スーパークラス(S): javax.servlet.http.HttpServlet 参照(E)...

インターフェース(I): 追加(A)...

使用するテンプレート: [1] サブレット用デフォルト・テンプレート

オプション:

<input checked="" type="checkbox"/> 継承メソッドの作成	<input checked="" type="checkbox"/> doGet の作成
<input checked="" type="checkbox"/> コンストラクターの作成	<input checked="" type="checkbox"/> doPost の作成
<input checked="" type="checkbox"/> init と destroy の作成	<input type="checkbox"/> doPut の作成
<input type="checkbox"/> doDelete の作成	<input type="checkbox"/> getServletInfo の作成

< 戻る(B) 次へ(N) > 終了(F) キャンセル

設定が必要な項目を次に示します。

項目名	設定値
ソース・フォルダー	サブレットを作成するソースフォルダを指定します。

5. ユーザインタフェースの作成

項目名	設定値
パッケージ	パッケージを指定します。
名前	作成するサブレットのサブレットクラス名を指定します。
スーパークラス	作成するサブレットのスーパークラスを指定します。
使用するテンプレート	[1] サブレット用デフォルト・テンプレート] を指定します。

必要に応じて、次の項目を設定してください。

項目名	設定値
エンクロージング型	サブレットクラスをインナークラスとして作成する場合は、チェックボックスをチェックします。また、外包するクラスを指定します。
修飾子	サブレットクラスの修飾子を指定します。
インターフェース	作成するサブレットに任意のインターフェースを実装する場合に指定します。
オプション	サブレット内に埋め込むメソッドを指定します。

3. [次へ] ボタンをクリックします。
[XML ウィザード] ページが表示されます。

[XML ウィザード] ページで設定する項目を示します。

なお、作成するサーブレットの情報を web.xml に反映する場合は、[web.xml ファイルの生成 / マップ] をチェックします。[web.xml ファイルの生成 / マップ] をチェックすると、[XML ウィザード] ページで指定する情報が web.xml に追加されます。

[XML ウィザード] ページに設定する内容を次に示します。必要に応じて、設定してください。

項目名	設定値	web.xml に追加される記述
サーブレット / JSP クラス名	[新規サーブレット] ダイアログの [名前] で指定した名称が表示されます。	<pre><servlet> <servlet-class> サーブレット /JSP クラス名 </servlet-class> <servlet></pre>
サーブレット / JSP 名	作成するサーブレットのサーブレット名を指定します。	<pre><servlet> <servlet-name> サーブレット /JSP 名 </servlet-name> <servlet></pre>

5. ユーザインタフェースの作成

項目名	設定値	web.xml に追加される記述
サーブレット / JSP マッピング URL	作成するサーブレットをマッピングする URL を指定します。	<code><servlet-mapping> <url-pattern> サーブレット /JSP マッピング URL</url-pattern> </servlet-mapping></code>
web.xml の ファイル・パス	web.xml のパスを指定します。	-
表示名	作成するサーブレットの表示名を指定します。	<code><servlet> <display-name> 表示名 </display-name> </servlet></code>
説明	作成するサーブレットの説明を指定します。	<code><servlet> <description> 説明 </description> </servlet></code>

4. [終了] ボタンをクリックします。

サーブレットが作成されて、作成したサーブレットおよび web.xml がエディタエリアに表示されます。

5. サーブレットに必要な処理を実装して保管します。

6

業務処理プログラムの作成

業務処理プログラム（Enterprise Bean）は，Eclipse の標準機能を使用して実装します。

この章では，業務処理プログラムの作成方法として，リモートホームインタフェース，およびリモートコンポーネントインタフェースを使用して実行する Enterprise Bean の作成手順について説明します。また，ほかの J2EE アプリケーションを呼び出す J2EE アプリケーションを開発する手順，およびアノテーションの記述方法についても説明します。

-
- 6.1 業務処理プログラム作成の流れ
 - 6.2 パッケージの作成
 - 6.3 Enterprise Bean クラスの作成
 - 6.4 業務処理プログラムおよびインタフェースの実装
 - 6.5 RMI-IIOP インタフェースを利用した J2EE アプリケーションの呼び出し
 - 6.6 アノテーションの記述方法
-

6.1 業務処理プログラム作成の流れ

Enterprise Bean に業務処理を実装し、リモートインタフェースやローカルインタフェースを使用して業務処理を実行できます。また、ビジネスインタフェースを使用して、業務処理を実行することもできます。

Enterprise Bean クラスの作成や、業務処理およびインタフェースの実装には、Eclipse の標準機能を使用できます。ここでは、リモートホームインタフェース、およびリモートコンポーネントインタフェースを使用する場合を例に、業務処理プログラムの作成手順を示します。

リモートホームインタフェース、およびリモートコンポーネントインタフェースを使用する場合、業務処理プログラムは次に示す流れで作成します。

作業内容	参照先
1. パッケージの作成	6.2
↓	
2. Enterprise Beanクラスの作成	6.3
↓	
3. 業務処理プログラム およびインタフェースの実装	6.4

それぞれの手順の概要を説明します。

1. パッケージの作成

関連のあるクラスやインタフェースを管理するためのパッケージを作成します。

2. Enterprise Bean クラスの作成

作成したパッケージに実装クラスを作成します。このときに使用する Enterprise Bean の種類を、Session Bean または Entity Bean に指定します。

3. 業務処理プログラムおよびインタフェースの実装

作成した Enterprise Bean に業務処理を実装します。また、リモートインタフェースおよびホームインタフェースを作成し、必要な記述を追加します。

ほかの J2EE アプリケーションを呼び出す Web アプリケーション、J2EE アプリケーション、および EJB クライアントアプリケーションを開発する場合は、「6.5 RMI-IIOP インタフェースを利用した J2EE アプリケーションの呼び出し」を参照してください。

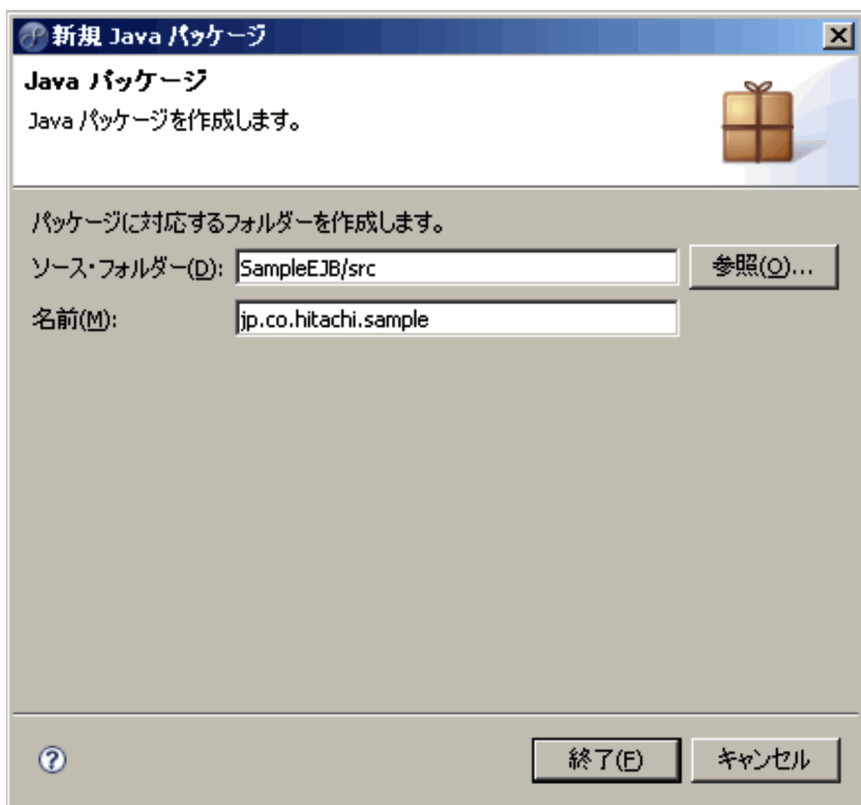
また、アノテーションの記述方法については、「6.6 アノテーションの記述方法」を参照してください。

以降の節では、この流れに沿って業務処理プログラムの作成方法を説明します。

6.2 パッケージの作成

EJB プロジェクトに、パッケージを作成します。

1. [パッケージ・エクスプローラー] ビューで EJB プロジェクトを選択して、コンテキストメニューから [新規] - [パッケージ] を選択します。
[新規 Java パッケージ] ダイアログが表示されます。
2. [名前] に任意のパッケージ名を入力します。



3. [終了] ボタンをクリックします。
パッケージが作成されます。

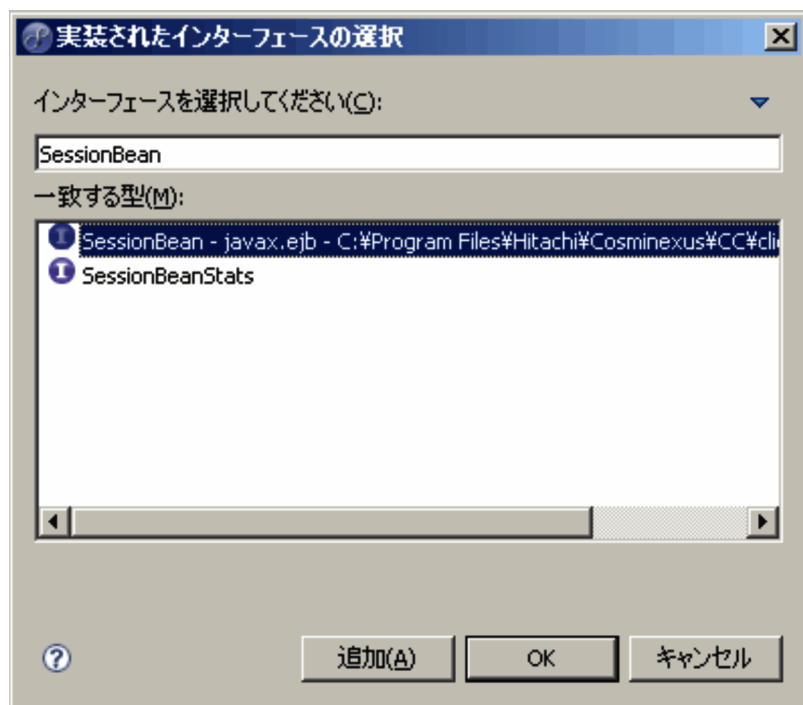
6.3 Enterprise Bean クラスの作成

作成したパッケージに実装クラスとなる Enterprise Bean を作成します。

1. [パッケージ・エクスプローラー] ビューで作成したパッケージを選択して、コンテキストメニューから [新規] - [クラス] を選択します。
[新規 Java クラス] ダイアログが表示されます。
2. [名前] に任意の Enterprise Bean のクラス名を入力します。



3. [インターフェース] の [追加] ボタンをクリックします。
[実装されたインターフェースの選択] ダイアログが表示されます。
4. [インターフェースを選択してください] に実装する Bean の種類を入力します。
「SessionBean」または「EntityBean」を指定します。



5. [一致する型] から指定する Bean を選択して , [OK] ボタンをクリックします。
[実装されたインターフェースの選択] ダイアログが閉じます。
6. [新規 Java クラス] ダイアログで [終了] ボタンをクリックします。
＜ 指定したクラス名 >.java のひな型が生成されます。

6.4 業務処理プログラムおよびインタフェースの実装

作成した Enterprise Bean に業務処理を実装します。また、リモートインタフェースおよびホームインタフェースを作成して、記述を追加します。

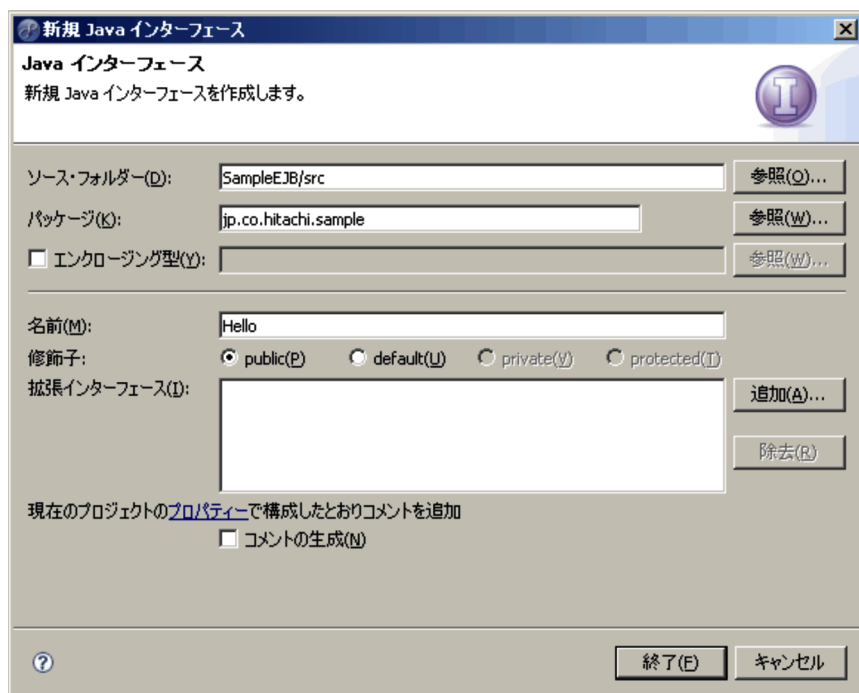
6.4.1 業務処理プログラムの実装

作成した Enterprise Bean に業務処理を実装します。業務処理プログラムは、マニュアル「Cosminexus 機能解説」に記載されている Enterprise Bean 実装時の注意事項の内容に注意して実装してください。

6.4.2 リモートインタフェースの実装

リモートインタフェースを作成して、実装します。

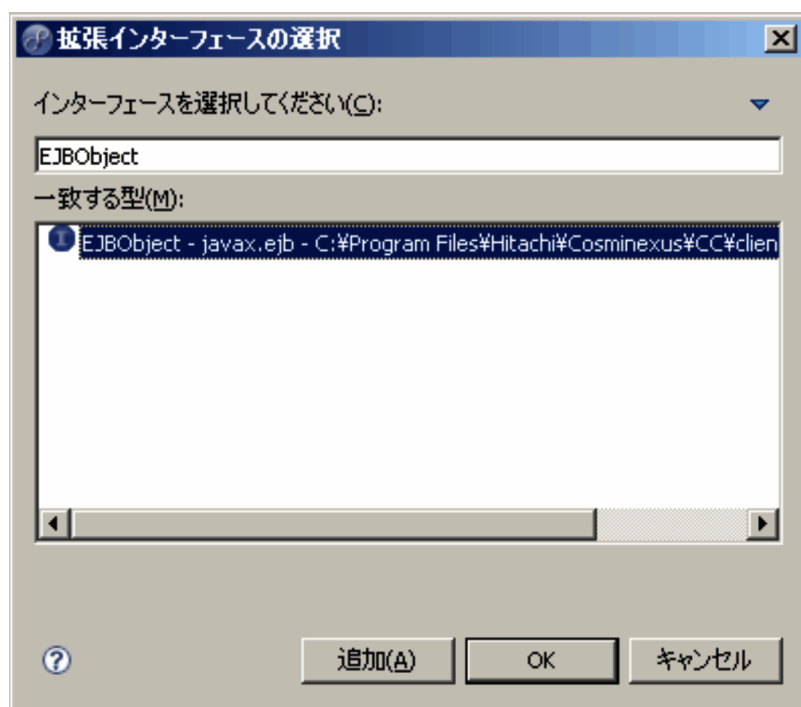
1. [パッケージ・エクスプローラー] ビューで作成したパッケージを選択して、コンテキストメニューから [新規] - [インターフェース] を選択します。
[新規 Java インターフェース] ダイアログが表示されます。
2. [名前] に任意のリモートインタフェース名を入力します。



3. [拡張インターフェース] の [追加] ボタンをクリックします。

[拡張インターフェースの選択] ダイアログが表示されます。

4. [インターフェースを選択してください] に「EJLObject」を入力します。



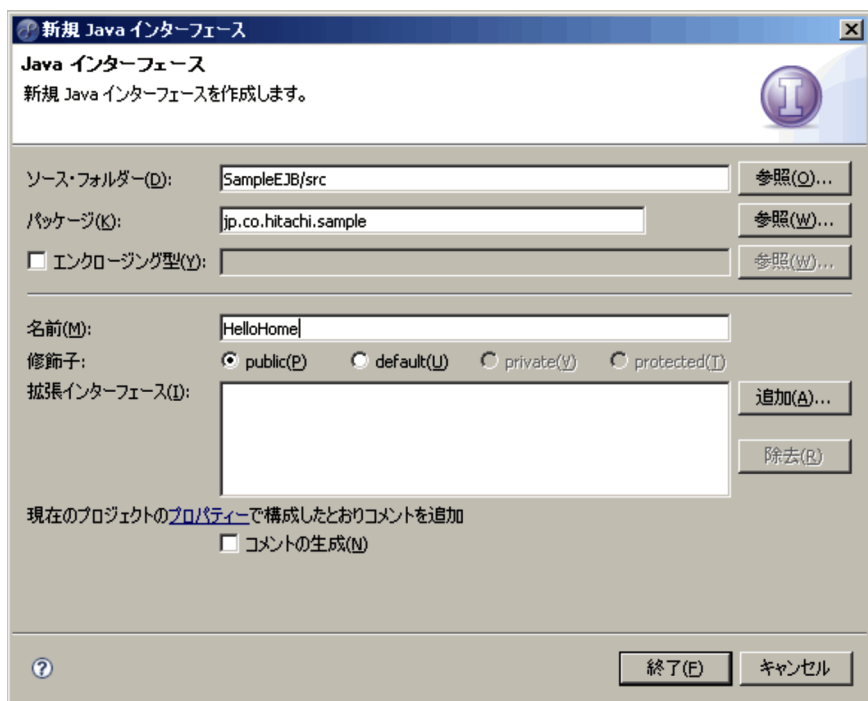
5. [一致する型] から指定する EJLObject を選択して、[OK] ボタンをクリックします。
[拡張インターフェースの選択] ダイアログが閉じます。
6. [新規 Java インタフェース] ダイアログで [終了] ボタンをクリックします。
< 指定したリモートインタフェース名 >.java のひな型が生成されます。
7. 生成されたリモートインタフェースに、Enterprise Bean として呼び出すビジネスメソッドを定義します。

6.4.3 ホームインタフェースの実装

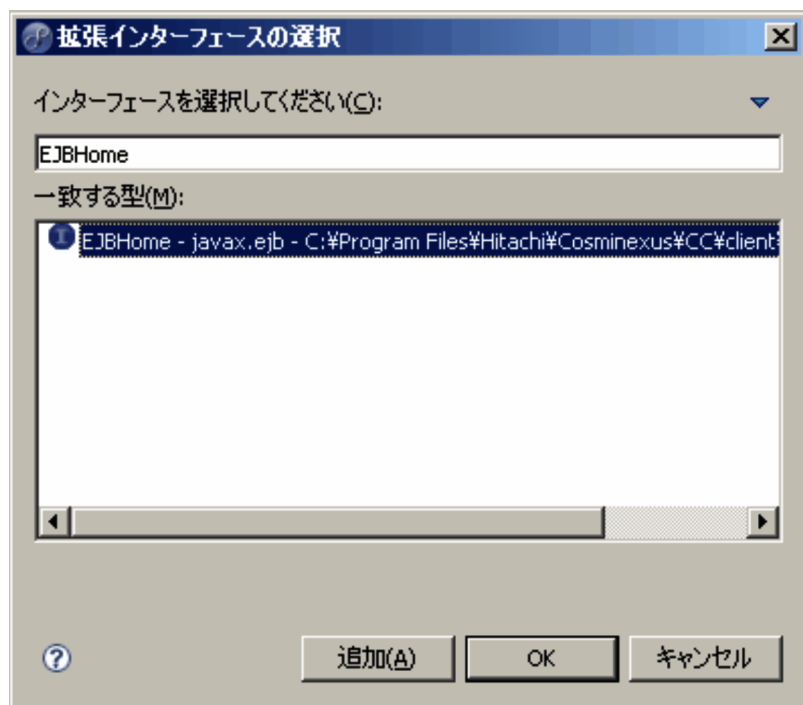
ホームインタフェースを作成し、実装します。

1. [パッケージ・エクスプローラー] ビューで作成したパッケージを選択して、コンテキストメニューから [新規] - [インターフェース] を選択します。
[新規 Java インタフェース] ダイアログが表示されます。
2. [名前] に任意のホームインタフェース名を入力します。

6. 業務処理プログラムの作成



3. [拡張インターフェイス] の [追加] ボタンをクリックします。
[拡張インターフェイスの選択] ダイアログが表示されます。
4. [インターフェイスを選択してください] に「EJBHome」を入力します。



5. [一致する型] から指定する EJBHome を選択して , [OK] ボタンをクリックします。
[拡張インターフェースの選択] ダイアログが閉じます。
6. [新規 Java インタフェース] ダイアログで [終了] ボタンをクリックします。
< 指定したホームインタフェース名 >.java のひな型が生成されます。
7. 生成されたホームインタフェースに , create メソッドを定義します。

6.5 RMI-IIOP インタフェースを利用した J2EE アプリケーションの呼び出し

Developer では、RMI-IIOP インタフェースを利用して、J2EE アプリケーションからほかの J2EE アプリケーションを呼び出したり、クライアントアプリケーションから J2EE アプリケーションを呼び出したりするアプリケーションを開発できます。

RMI-IIOP インタフェースは、Developer が提供する Eclipse のプラグインで、J2EE サーバ上の J2EE アプリケーションから取得できます。

ここでは、RMI-IIOP インタフェースの取得方法、および RMI-IIOP インタフェースを利用するアプリケーションの開発の流れを説明します。

ポイント

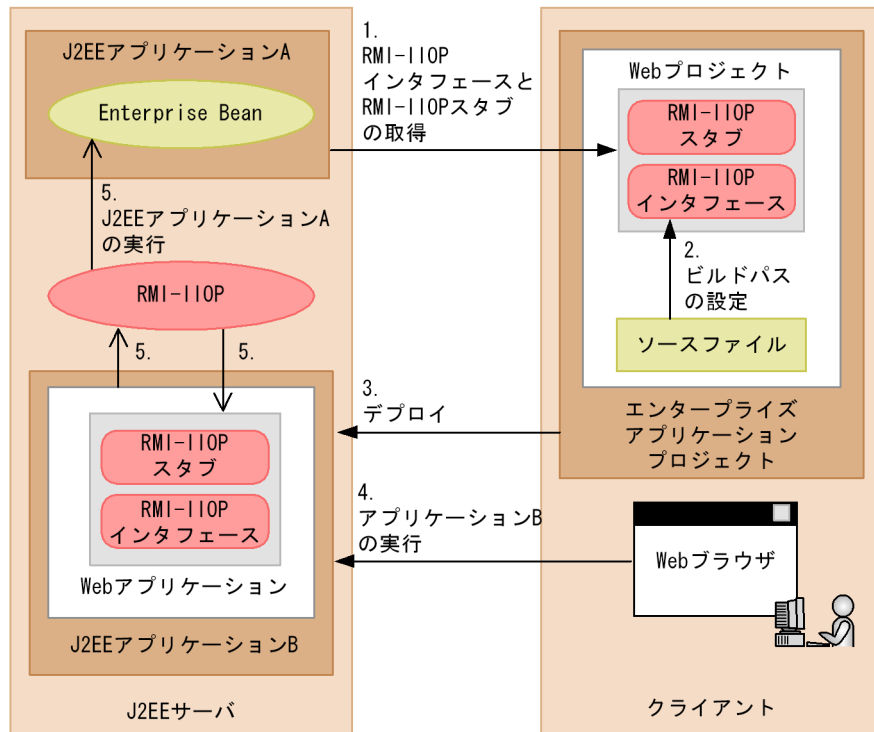
RMI-IIOP 通信を使用した Enterprise Bean のメソッド呼び出しでは、Unicode の補助文字を送受信できます。詳細については、マニュアル「Cosminexus 機能解説」の Enterprise Bean 共通の注意事項に関する説明を参照してください。

6.5.1 Web アプリケーションから呼び出す場合

同じ J2EE サーバ上にある J2EE アプリケーションを呼び出す Web アプリケーションを作成します。このとき、呼び出し先の J2EE アプリケーションの RMI-IIOP インタフェースを Web プロジェクトに取得する必要があります。

RMI-IIOP インタフェースの取得から J2EE アプリケーションの実行までの仕組みを次の図に示します。

図 6-1 Web プロジェクトで RMI-IIOP インタフェースを使用するときの仕組み



(凡例)

→ プラグインでサポートする動作

→ J2EEサーバ内部の動作

実行の流れを次に示します。

1. RMI-IIOP インタフェースと RMI-IIOP スタブの取得

RMI-IIOP インタフェースおよび RMI-IIOP スタブを Web プロジェクトに取得します。取得先に、次のフォルダを指定します。

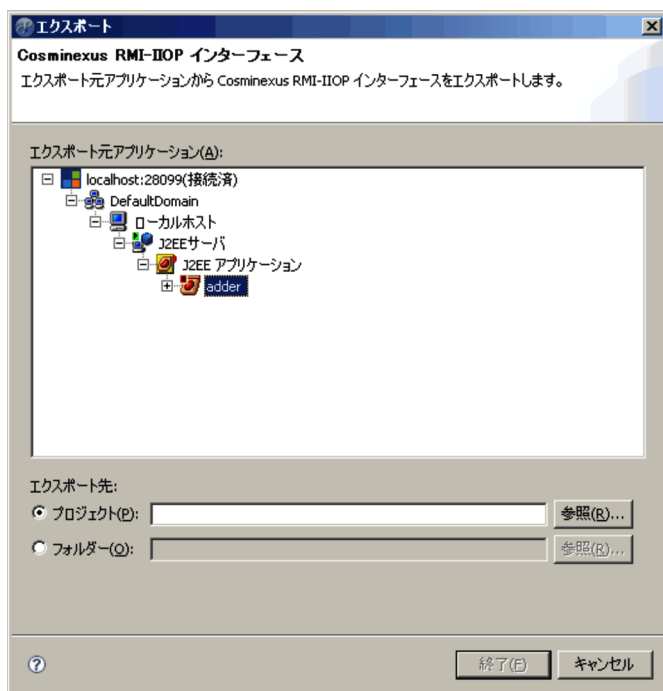
<Webプロジェクトのルート>¥<Webルート・フォルダ>¥WEB-INF¥lib

参考

RMI-IIOP インタフェースを Web プロジェクトに取得する手順は次のとおりです。

1. [Cosminexus Server Plug-in] パースペクティブの [サーバー・エクスプローラー] ビューで、RMI-IIOP インタフェースのエクスポート対象の J2EE アプリケーションを選択します。
2. コンテキストメニューから [Cosminexus RMI-IIOP インタフェースのエクスポート] を選択します。
[エクスポート] ダイアログが表示されます。

6. 業務処理プログラムの作成



3. [エクスポート] ダイアログで , [エクスポート先] の [フォルダ] にチェックして , [参照] ボタンをクリックします。
[フォルダの参照] ダイアログが表示されます。エクスポート先のフォルダを指定してください。ここでは , 「 lib 」 フォルダを指定します。

<Web プロジェクトのルート >¥<Web ルート・フォルダ >¥WEB-INF¥lib



4. [エクスポート] ダイアログの [終了] ボタンをクリックします。
[エクスポート] ダイアログが閉じて , RMI-IIOP インタフェースが取得されます。

2. ビルドパスの設定

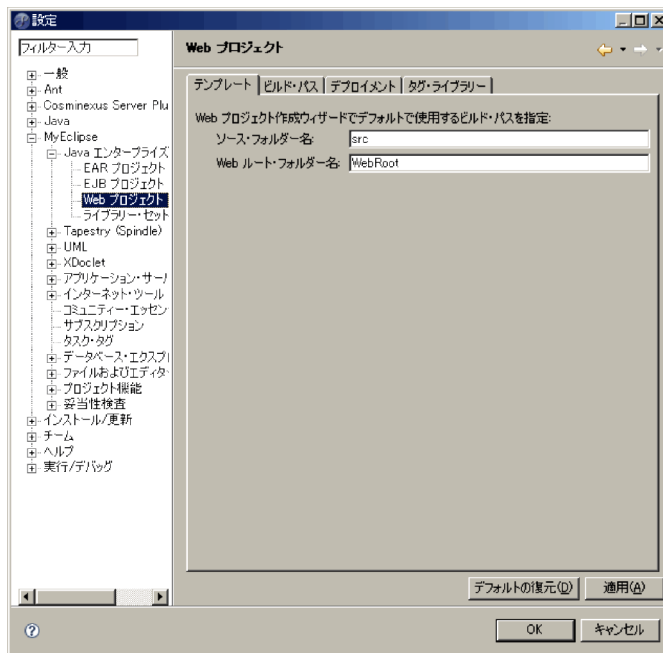
取得した RMI-IIOP インタフェースに , 自動でビルドパスが設定されます。ビルドパ

スが設定されると、Web プロジェクトの開発で Enterprise Bean のクラスを参照できます。

！ 注意事項

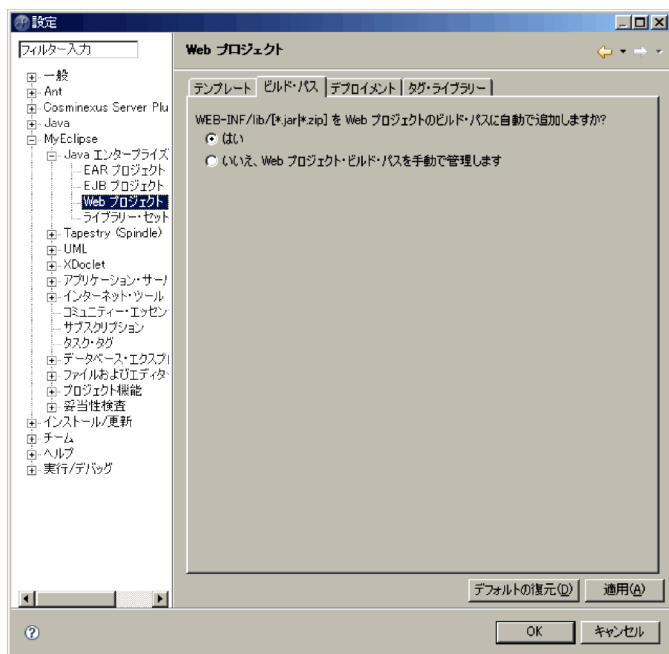
MyEclipse の設定によって、自動でビルドパスを設定する機能が無効になる場合があります。RMI-IIOP インタフェースを取得する前に、ビルドパスを自動で設定する機能が有効になっていることを次の手順で確認してください。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. [設定] ダイアログの左ペインで [MyEclipse] - [Java エンタープライズ・プロジェクト] - [Web プロジェクト] を選択します。
[Web プロジェクト] ページが表示されます。



3. [Web プロジェクト] ページの [ビルド・パス] タブを選択します。
[WEB-INF/lib/*.*jar|*.zip] を Web プロジェクトのビルドパスに自動で追加しますか？] が [はい] にチェックされていることを確認します。

6. 業務処理プログラムの作成



4. [OK] ボタンをクリックする。
[設定] ダイアログが閉じます。

なお、RMI-IIOP インタフェースおよび RMI-IIOP スタブを <Web プロジェクトのルート>¥<Web ルート・フォルダ>¥WEB-INF¥lib 以外のフォルダに取得しないでください。

3. デプロイ

RMI-IIOP インタフェースおよび RMI-IIOP スタブを取得した Web プロジェクトが含まれるエンタープライズアプリケーションプロジェクトを J2EE サーバにデプロイします。

4. J2EE アプリケーション B の実行

Web ブラウザを起動して、J2EE アプリケーション B を実行します。

5. J2EE アプリケーション A の実行

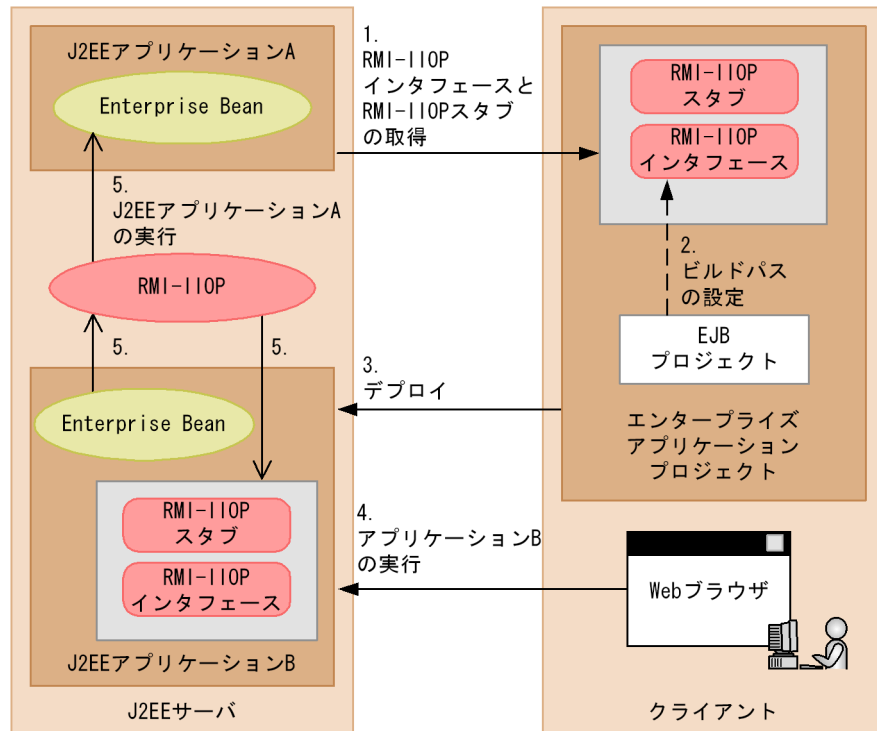
J2EE アプリケーション B の Web アプリケーションによって、J2EE アプリケーション A の Enterprise Bean が実行されます。J2EE アプリケーション A の Enterprise Bean を実行するために、RMI-IIOP インタフェースおよび RMI-IIOP スタブが参照されます。

6.5.2 Enterprise Bean から呼び出す場合

同じ J2EE サーバ上にある J2EE アプリケーションを呼び出す Enterprise Bean を作成します。このとき、呼び出し先の J2EE アプリケーションの RMI-IIOP インタフェースを、呼び出し元となる EJB プロジェクトをモジュールプロジェクトに含むエンタープライズアプリケーションプロジェクトに取得する必要があります。

RMI-IIOP インタフェースの取得から J2EE アプリケーションの実行までの仕組みを次の図に示します。

図 6-2 EJB プロジェクトで RMI-IIOP インタフェースを使用するときの仕組み



- (凡例)
- プラグインでサポートする動作
 - J2EEサーバ内部の動作
 - ユーザの操作

実行の流れを次に示します。

- 1. RMI-IIOP インタフェースと RMI-IIOP スタブの取得**
RMI-IIOP インタフェースおよび RMI-IIOP スタブをエンタープライズアプリケーションプロジェクトに取得します。取得先は、エンタープライズアプリケーションプロジェクトのルートを指定します。
RMI-IIOP インタフェースの取得方法については、「6.5.1 Web アプリケーションから呼び出す場合」を参照してください。
- 2. ビルドパスの設定**
取得した RMI-IIOP インタフェースに、手動でビルドパスを設定します。ビルドパスを設定すると、EJB プロジェクトの開発で Enterprise Bean のクラスを参照できます。

参考

EJB プロジェクトから、エンタープライズアプリケーションプロジェクトに取得した RMI-IIOP インタフェースへビルドパスを設定する手順は、次のとおりです。

1. [MyEclipse Java エンタープライズ開発] パースペクティブの [パッケージ・エクスプローラー] ビューで EJB プロジェクトを選択して、コンテキストメニューから [ビルド・パス] - [外部アーカイブの追加] を選択します。
[JAR の選択] ダイアログが表示されます。
 2. [JAR の選択] ダイアログで、RMI-IIOP インタフェースを取得したエンタープライズアプリケーションプロジェクトを選択して、[開く] ボタンをクリックします。
選択したエンタープライズアプリケーションプロジェクトの中身が表示されます。
 3. 「< 整数の連番 >.jar」および「stubs.jar」を選択して、[開く] ボタンをクリックします。
手順 1 で選択した EJB プロジェクトから「< 整数の連番 >.jar」および「stubs.jar」へのビルドパスが設定されます。
-

3. デプロイ

RMI-IIOP インタフェースおよび RMI-IIOP スタブを取得したエンタープライズアプリケーションプロジェクトを J2EE サーバにデプロイします。

4. J2EE アプリケーション B の実行

Web ブラウザを起動して、J2EE アプリケーション B を実行します。

5. J2EE アプリケーション A の実行

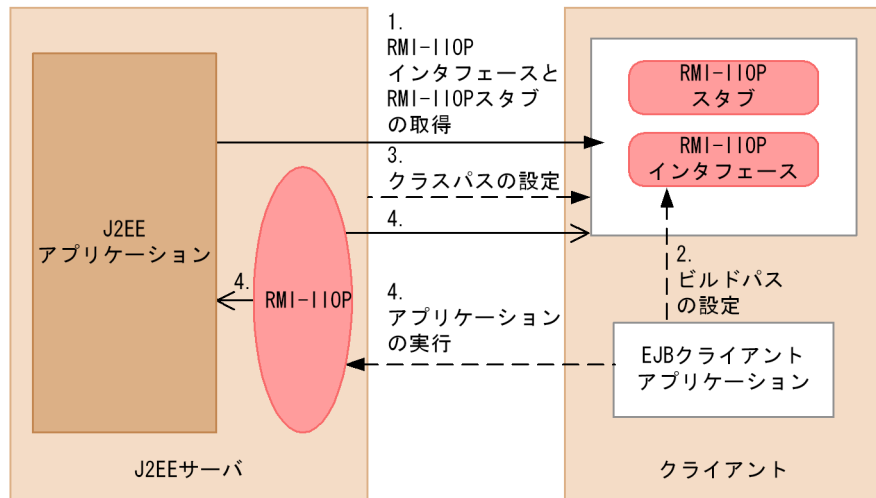
J2EE アプリケーション B の Enterprise Bean によって、J2EE アプリケーション A の Enterprise Bean が実行されます。J2EE アプリケーション A の Enterprise Bean を実行するために、RMI-IIOP インタフェースおよび RMI-IIOP スタブが参照されます。

6.5.3 EJB クライアントアプリケーションからの呼び出す場合

J2EE アプリケーションを呼び出す EJB クライアントアプリケーションを作成します。このとき、呼び出し先の J2EE アプリケーションの RMI-IIOP インタフェースを任意のフォルダに取得する必要があります。

RMI-IIOP インタフェースの取得から J2EE アプリケーションの実行までの仕組みを次の図に示します。ただし、Developer がサポートするのは、RMI-IIOP インタフェースの取得だけです。

図 6-3 EJB クライアントアプリケーションで RMI-IIOP インタフェースを使用するときの仕組み



(凡例)

——→ プラグインでサポートする動作

——→ J2EEサーバ内部の動作

---→ ユーザの操作

実行の流れを次に示します。

1. RMI-IIOP インタフェースと RMI-IIOP スタブの取得

RMI-IIOP インタフェースおよび RMI-IIOP スタブを任意のフォルダに取得します。RMI-IIOP インタフェースの取得方法については、「6.5.1 Web アプリケーションから呼び出す場合」を参照してください。

2. ビルドパスの設定

取得した RMI-IIOP インタフェースに、手動でビルドパスを設定します。ビルドパスを設定すると、EJB クライアントアプリケーションの開発で Enterprise Bean のクラスを参照できます。

参考

Eclipse で EJB クライアントアプリケーションを開発する場合は、任意のディレクトリに取得した RMI-IIOP インタフェースに、次の手順でビルドパスを設定します。

1. [MyEclipse Java エンタープライズ開発] パースペクティブの [パッケージ・エクスプローラー] ビューで EJB クライアントアプリケーションの開発に使用しているプロジェクトを選択して、コンテキストメニューから [ビルド・パス] - [外部アーカイブの追加] を選択します。
[JAR の選択] ダイアログが表示されます。
 2. [JAR の選択] ダイアログで、RMI-IIOP インタフェースを取得したディレクトリを選択して、[開く] ボタンをクリックします。
選択したディレクトリの中身が表示されます。
 3. 「< 整数の連番 >.jar」および「stubs.jar」を選択して、[開く] ボタンをクリックします。
手順 1 で選択したプロジェクトから「< 整数の連番 >.jar」および「stubs.jar」へのビルドパスが設定されます。
-

3. クラスパスの設定

取得した RMI-IIOP インタフェースおよび RMI-IIOP スタブに、手動でクラスパスを設定します。クラスパスを設定すると、EJB クライアントアプリケーションの実行で RMI-IIOP インタフェースおよび RMI-IIOP スタブを参照できます。

4. EJB クライアントアプリケーションの実行

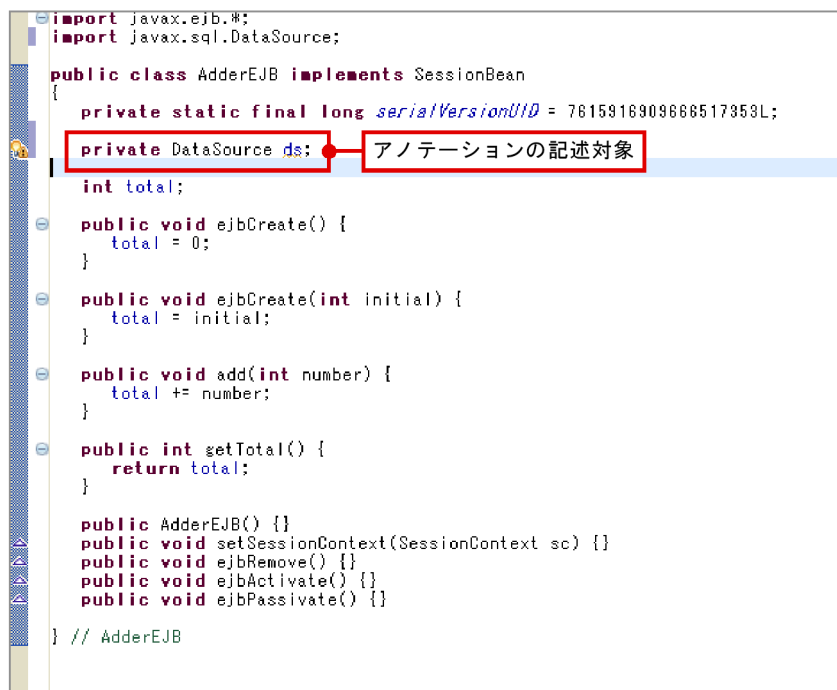
EJB クライアントアプリケーションを実行すると、RMI-IIOP を通じて J2EE サーバ上の J2EE アプリケーションが実行されます。J2EE アプリケーションの Enterprise Bean を EJB クライアントアプリケーション側で実行するために、RMI-IIOP インタフェースおよび RMI-IIOP スタブが参照されます。

6.6 アノテーションの記述方法

MyEclipse を使用した J2EE アプリケーション開発では、Java ソースにアノテーションを使用できます。しかし、Eclipse ではアノテーションのコード補完をサポートしていないので、Java ソースファイルに手動でアノテーションを記述する必要があります。

この節では、Java ソースファイルへのアノテーションの記述例として「@Resource」を記述する場合の手順を説明します。

記述前の Java ソースファイルを次に示します。



```
import javax.ejb.*;
import javax.sql.DataSource;

public class AdderEJB implements SessionBean
{
    private static final long serialVersionUID = 7615916909666517353L;
    private DataSource ds;
    int total;

    public void ejbCreate() {
        total = 0;
    }

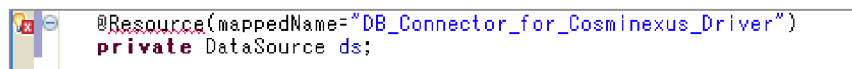
    public void ejbCreate(int initial) {
        total = initial;
    }

    public void add(int number) {
        total += number;
    }

    public int getTotal() {
        return total;
    }

    public AdderEJB() {}
    public void setSessionContext(SessionContext sc) {}
    public void ejbRemove() {}
    public void ejbActivate() {}
    public void ejbPassivate() {}
} // AdderEJB
```

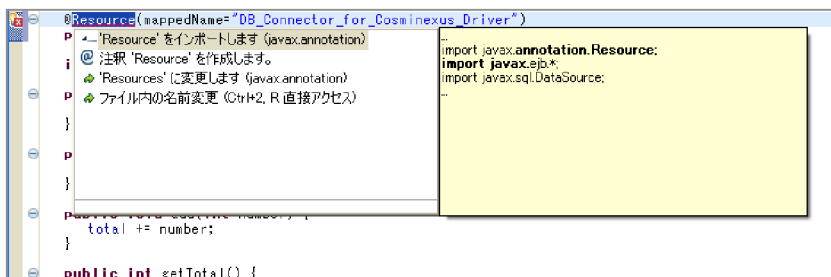
1. 変数の宣言「private DataSource ds;」の上の行に、アノテーション「@Resource」を直接記述します。



```
@Resource(mappedName="DB_Connector_for_Cosminexus_Driver")
private DataSource ds;
```

2. アノテーション「@Resource」を記述した行のエラーアイコンをクリックして、コンテキストメニューから「Resource」をインポートします。

6. 業務処理プログラムの作成



3. 「javax.annotation.Resource」のimport文が自動で追加されて、エラーアイコンが消えます。



なお、アノテーションに存在しない属性を指定した場合、および属性に値を指定しない場合はエラーが表示されます。

7

J2EE アプリケーションのテスト

作成した J2EE アプリケーションを実行環境に配布する前に、開発環境でテストして、デバッグします。ここでは、J2EE アプリケーションのテストに必要な準備、およびテストの手順を説明します。

7.1 J2EE アプリケーションのテストの流れ

7.2 組み込みデータベースの開始および停止

7.3 J2EE サーバの開始および停止

7.4 J2EE アプリケーションのデプロイおよびアンデプロイ

7.5 プロジェクトの入れ替え

7.6 J2EE アプリケーションのデバッグ

7.7 J2EE アプリケーションのエクスポート

7.8 MyEclipse 使用時の注意事項

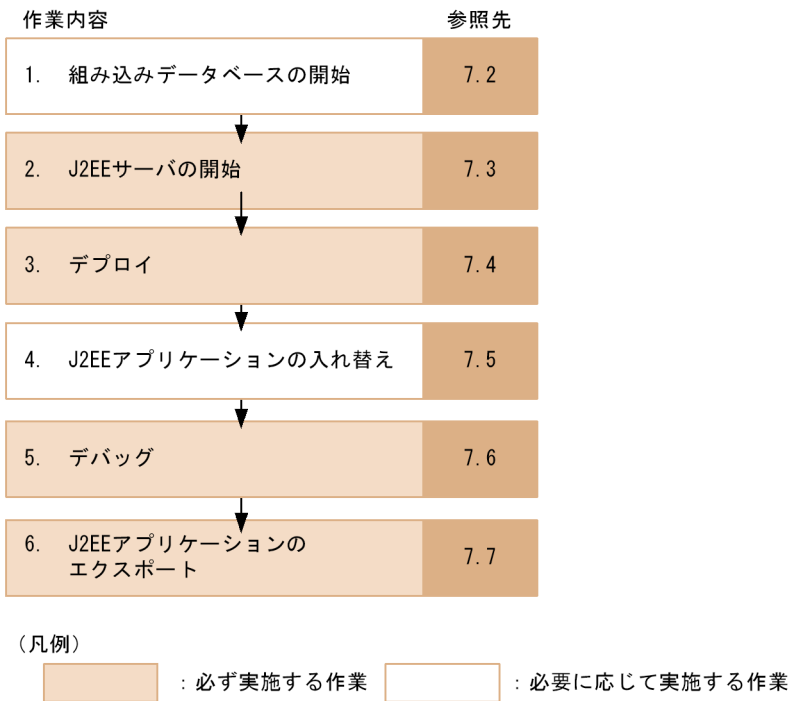
7.9 Server Plug-in から起動した J2EE サーバでのデバッグ

7.1 J2EE アプリケーションのテストの流れ

作成した MyEclipse のプロジェクトを J2EE サーバにデプロイして、J2EE アプリケーションをテスト用実行環境で実行します。

J2EE アプリケーションのテストの流れは、次の図のとおりです。

図 7-1 J2EE アプリケーションのテストの流れ図



それぞれの手順の概要を説明します。

- 1. 組み込みデータベースの開始
組み込みデータベースを使用する場合は、インスタントセットアップ機能が提供するショートカットを使用して組み込みデータベースを開始します。
- 2. J2EE サーバの開始
J2EE サーバをデバッグモードで開始します。ただし、J2EE サーバを開始する前に、Management Server リモート管理機能へログインしてください。
- 3. デプロイ
MyEclipse で作成したプロジェクトを J2EE サーバにデプロイします。プロジェクトを J2EE サーバにデプロイすると、J2EE サーバ上に J2EE アプリケーションが作成されます。

4. J2EE アプリケーションの入れ替え

J2EE アプリケーションに含まれるリソースアダプタに対してプロパティを設定した場合、J2EE アプリケーションを入れ替えます。なお、エンタープライズアプリケーションプロジェクトの内容に変更があった場合も J2EE アプリケーションを入れ替えます。

5. デバッグ

作成した J2EE アプリケーションをデバッグします。デバッグしたあとに、必要に応じて J2EE アプリケーションを入れ替えてください。

6. J2EE アプリケーションのエクスポート

MyEclipse で開発した J2EE アプリケーションを EAR ファイルとしてエクスポートします。エクスポートには MyEclipse を使用します。

以降の節では、この流れに沿って J2EE アプリケーションのテストの手順を説明します。

7.2 組み込みデータベースの開始および停止

J2EE アプリケーションのテストで組み込みデータベースを使用する場合は、組み込みデータベースを開始する必要があります。

Developer では、スタートメニューのショートカットまたはコマンドから、組み込みデータベースの開始と停止を実行できます。なお、どちらの方法でも実行するためには Administrator 権限が必要です。

それぞれの実行方法を次に説明します。

スタートメニューのショートカットを使用する場合

組み込みデータベースの開始

[Cosminexus] - [Application Development Plug-in] - [組み込みデータベースの開始] を選択する。

組み込みデータベースの停止

[Cosminexus] - [Application Development Plug-in] - [組み込みデータベースの停止] を選択する。

組み込みデータベース用コマンドを使用する場合

組み込みデータベースの開始

次のコマンドを実行してください。

```
"<Cosminexusのインストールディレクトリ>%SCC%DB%bats%cddbstart.bat"
```

組み込みデータベースの停止

次のコマンドを実行してください。

```
"<Cosminexusのインストールディレクトリ>%SCC%DB%bats%cddbstop.bat" [-f [-d ]] | [-d ]]
```

-f

強制終了する場合に指定します。

-d

終了と同時に共用メモリダンプを出力する場合に指定します。

共用メモリダンプは、<Cosminexus のインストールディレクトリ>%DB%spool%pdshmdump%shmdump に出力されます。

7.3 J2EE サーバの開始および停止

「2.4.3 J2EE サーバの選択」で選択した J2EE サーバを開始、または停止します。ここでは、MyEclipse での J2EE サーバの開始および停止の手順を説明します。

7.3.1 J2EE サーバの開始

MyEclipse を使用して、J2EE サーバを開始します。

[Cosminexus 7] ページで Management Server リモート管理機能にログインしてから、J2EE サーバを開始してください。Management Server リモート管理機能のログイン方法については、「2.4.2 Management Server リモート管理機能へのログインおよびログアウト (MyEclipse)」を参照してください。


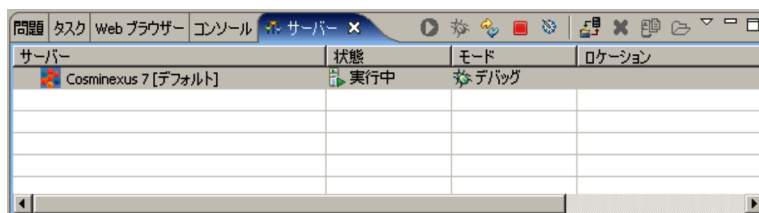
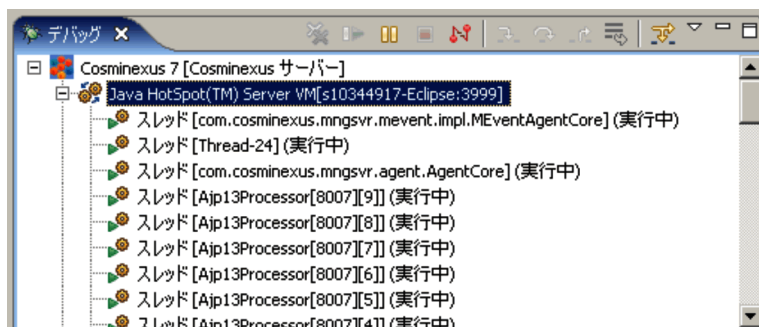
J2EE サーバをデバッグモードで開始する場合は、[MyEclipse Java エンタープライズ開発] パースペクティブの [サーバー] ビューで [Cosminexus 7] を選択して、ビューツールバーから  (デバッグ・モードで開始) をクリックします。Management Server リモート管理機能に未接続の場合は、[ログイン] ダイアログが表示されるので、管理ユーザ ID とパスワードを入力して、[OK] ボタンをクリックします。

図 7-2 J2EE サーバを開始したときの [サーバー] ビュー



J2EE サーバの起動構成を確認したい場合は、[デバッグ] パースペクティブの [デバッグ] ビューを確認してください。

図 7-3 J2EE サーバの起動構成 (デバッグモード)




参考

実行モードでの開始

MyEclipse では、J2EE サーバを実行モードでも開始できます。ただし、実行モードで開始した場合は、デバッグを実行できません。

実行モードで開始する場合は、[MyEclipse Java エンタープライズ開発] パースペクティブの [サーバー] ビューで [Cosminexus 7] を選択して、ビューツールバーの


[ (実行モードで開始)] をクリックします。

(1) 実行中の J2EE サーバに関する注意事項

実行中の J2EE サーバを削除しないでください。削除すると、J2EE サーバを正常に停止できなくなったり、[デバッグ] パースペクティブの [デバッグ] ビューに表示される J2EE サーバの起動構成が終了しなくなったりします。

7.3.2 J2EE サーバの停止


ここでは、J2EE サーバを停止する手順を説明します。

J2EE サーバを停止させる場合は、[サーバー] ビューで [Cosminexus 7] を選択して、ビューツールバーの [ (停止)] をクリックします。

なお、J2EE サーバを停止しないで Eclipse を終了した場合は、自動的に J2EE サーバが停止され、Management Server リモート管理機能からログアウトされます。

7.3.3 J2EE サーバの再起動

ここでは、J2EE サーバを再起動する手順を説明します。

J2EE サーバを再起動させる場合は、[サーバー] ビューで [Cosminexus 7] を選択して、ビューツールバーの [ (リスタート)] をクリックします。

なお、Eclipse のツールバーから J2EE サーバを再起動した場合は、Cosminexus 起動モードの設定に従うため、リスタート前と異なる起動モードになる場合があります。Cosminexus 起動モードの設定については、「付録 B.2(4)(b) J2EE サーバ起動時の動作の設定」を参照してください。

7.4 J2EE アプリケーションのデプロイおよびアンデプロイ

MyEclipse で作成したプロジェクトを J2EE サーバにデプロイします。デプロイを実行すると、デプロイ作業ディレクトリに J2EE アプリケーションが作成されます。作成された J2EE アプリケーションは J2EE サーバへインポートされて、開始されます。

プロジェクトをデプロイして作成した J2EE アプリケーションを J2EE サーバから削除することをアンデプロイといいます。必要に応じて J2EE サーバから J2EE アプリケーションをアンデプロイしてください。

ここでは、J2EE アプリケーションのデプロイおよびアンデプロイの手順を説明します。


7.4.1 J2EE アプリケーションのデプロイ

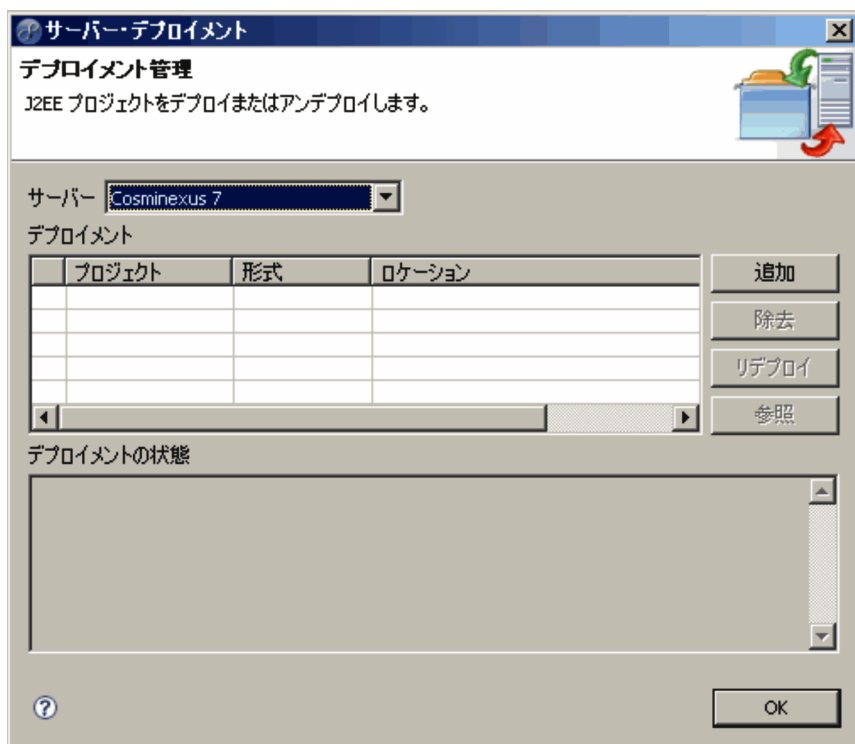
MyEclipse で作成したプロジェクトを J2EE サーバにデプロイします。ここでは、プロジェクトをデプロイする手順、およびデプロイしたプロジェクトに対する操作の注意事項を説明します。

ただし、閉じた状態のプロジェクトをデプロイ、アンデプロイ、および入れ替えをすると誤作動する場合があります。MyEclipse から操作する場合は、エンタープライズアプリケーションプロジェクトおよびそれを構成するモジュールプロジェクト（EJB プロジェクトまたは Web プロジェクト）をすべて開いた状態にしてください。

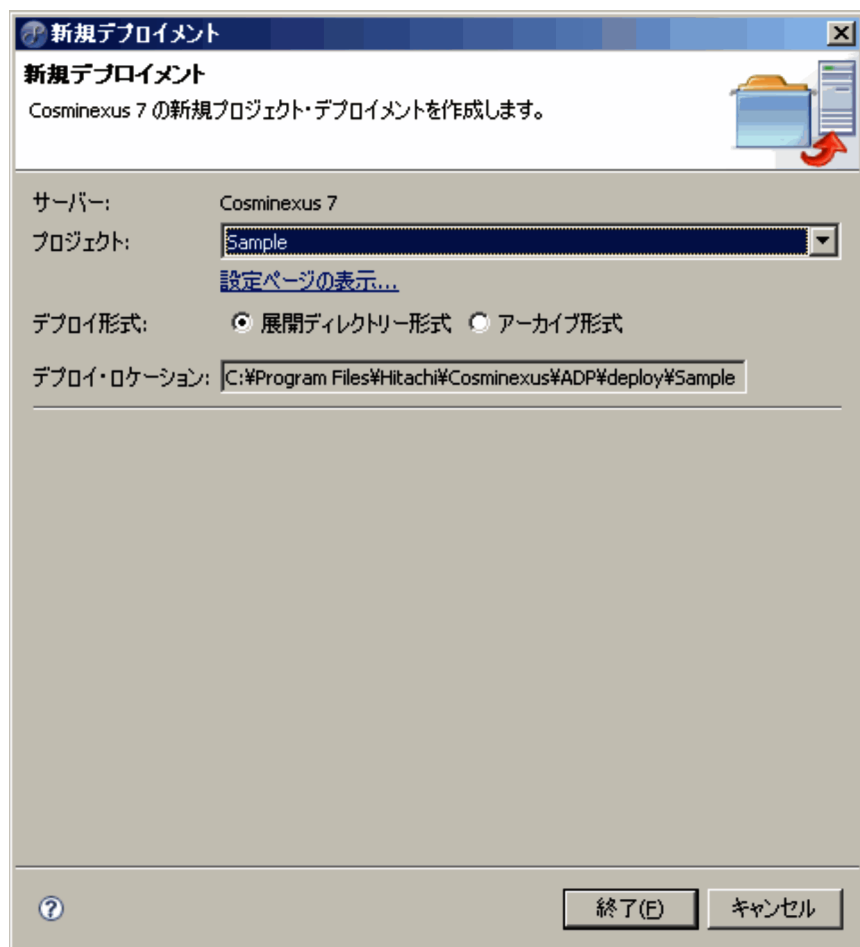
（１）プロジェクトのデプロイ

エンタープライズアプリケーションプロジェクトを J2EE サーバにデプロイする手順を次に示します。

1. [MyEclipse Java エンタープライズ開発] パースペクティブの [サーバー] ビューで [Cosminexus 7] を選択して、ビューツールバーから  (デプロイメント管理) をクリックします。
[サーバー・デプロイメント] ダイアログが表示されます。



2. [サーバー・デプロイメント] ダイアログの [追加] ボタンをクリックします。
[新規デプロイメント] ダイアログが表示されます。



設定が必要な項目を次に示します。

項目名	設定値
プロジェクト	デプロイするエンタープライズアプリケーションプロジェクトを選択します。
デプロイ形式	J2EE アプリケーションのデプロイ形式を選択します。 ここでは、J2EE アプリケーションの入れ替えに自動リロードを使用するので、[展開ディレクトリ形式] を選択します。

3. [新規デプロイメント] ダイアログの [終了] ボタンをクリックします。
エンタープライズアプリケーションプロジェクトがデプロイされて、[サーバー・デプロイメント] ダイアログに戻ります。デプロイの結果が [サーバー・デプロイメント] ダイアログの下部に表示されます。
4. [サーバー・デプロイメント] ダイアログで [OK] ボタンをクリックします。
[サーバー・デプロイメント] ダイアログが閉じます。

(2) デプロイしたプロジェクトに対する操作の注意


Cosminexus コネクタからデプロイしたプロジェクトに対して、次の操作をしないでください。MyEclipse のプロジェクトと J2EE サーバにデプロイした J2EE アプリケーションの関係が一致しくなくなります。

- エンタープライズアプリケーションプロジェクトをデプロイしたあとに、application.xml の <display-name> タグの内容を変更する。
- 閉じた状態のプロジェクトをデプロイ、アンデプロイ、および入れ替えをする。
- デプロイ作業ディレクトリにあるファイル・ディレクトリを削除または変更する。

これらの操作を実施すると、プロジェクトのアンデプロイ、入れ替えができなくなるおそれがあります。アンデプロイ、入れ替えができない場合は、「7.8.3 デプロイまたはアンデプロイできない場合の対処方法」を参照してください。

7.4.2 J2EE アプリケーションのアンデプロイ

次の手順でアンデプロイします。

1. [MyEclipse Java エンタープライズ開発] パースペクティブの [サーバー] ビューで、削除対象のエンタープライズアプリケーションプロジェクトを選択します。
2. [サーバー] ビューのツールバーから [ (アンデプロイ)] をクリックします。

7.5 プロジェクトの入れ替え

MyEclipse の入れ替え機能は、次の場合に使用します。

- J2EE アプリケーションのプロパティの変更を反映する場合
- J2EE アプリケーション内のリソースアダプタのプロパティの変更を反映する場合
- ソースコードの修正など、プロジェクトの変更を J2EE アプリケーションに反映する場合


プロジェクトの入れ替えには、MyEclipse のリデプロイ機能を使用します。変更したソースコードの内容をすぐに反映させたい場合は、MyEclipse のリデプロイを実行して、エンタープライズアプリケーションプロジェクトを入れ替えてください。ただし、展開ディレクトリ形式でデプロイした場合は、更新検知によるリロードを設定しておくことで変更が自動で反映されます。更新検知によるリロードの設定については、インスタントセットアップ機能で構築した場合は、「2.3.2 セットアップする環境の設定内容」を参照してください。インスタントセットアップ機能で構築していない場合は、「付録 B.2(2)(a) 簡易構築定義ファイルの作成」で設定した値を指定してください。

ここでは、プロジェクトの入れ替え手順、プロジェクトの入れ替え時の注意事項を説明します。

7.5.1 プロジェクトの入れ替え手順

更新検知によるリロードを有効にしていない J2EE サーバをデプロイ先にした展開ディレクトリ形式のエンタープライズアプリケーションプロジェクトと、アーカイブ形式のエンタープライズアプリケーションプロジェクトは、MyEclipse のリデプロイ機能を使用して、プロジェクトを入れ替えます。

MyEclipse からプロジェクトを入れ替える場合は、MyEclipse のリデプロイ機能を使用します。次の手順で実行してください。

1. [MyEclipse Java エンタープライズ開発] パースペクティブの [サーバー] ビューで入れ替える対象のエンタープライズアプリケーションプロジェクトを選択します。
2. [サーバー] ビューのツールバーから [ (リデプロイ)] をクリックします。
選択したエンタープライズアプリケーションプロジェクトが入れ替えられます。

参考

更新検知によるリロード機能を有効にしている場合に、MyEclipse でプロジェクトの入れ替えを実行すると、変更されたファイルがないことを示すメッセージ (KDJE42285-I) がコンソールに出力されることがあります。

これは、MyEclipse からプロジェクトの入れ替えを実行する前に、更新検知によって変更したファイルがリロードされていることを示します。

7.5.2 プロジェクトの入れ替え時の注意事項

MyEclipse からプロジェクトを入れ替える場合の注意事項を、J2EE アプリケーションの形式ごとに説明します。

(1) 展開ディレクトリ形式の注意事項

次の条件に当てはまる展開ディレクトリ形式のエンタープライズアプリケーションプロジェクトは、リロードできません。また、更新検知によるリロードもできません。

MyEclipse からアンデプロイして、再度デプロイしてください。

- プロジェクト（エンタープライズアプリケーションプロジェクト、EJB プロジェクト、Web プロジェクト）の DD を変更した場合
- EJB プロジェクトで EJB を構成するクラスを追加および削除した場合
- Web プロジェクトにリソース（ファイル）を追加した場合
- エンタープライズアプリケーションプロジェクトへの RAR ファイルの追加および削除した場合
- リデプロイ対象がモジュールプロジェクト（EJB プロジェクト、Web プロジェクト）だけの場合

なお、更新検知の対象となるファイルについては、マニュアル「Cosminexus 機能解説」を参照してください。

(2) アーカイブ形式の注意事項

次の条件に当てはまるアーカイブ形式のエンタープライズアプリケーションプロジェクトは、リデプロイできません。MyEclipse からアンデプロイして、再度デプロイしてください。

- J2EE アプリケーション内の EJB-JAR、リソースアダプタ、WAR の数が異なる場合
- J2EE アプリケーション内の EJB-JAR ファイル、RAR ファイル、WAR ファイルの名称が異なる場合
- EJB-JAR ファイル内のホームインタフェース（ローカル、リモート）、コンポーネントインタフェース（ローカル、リモート）、ビジネスインタフェース（ローカル、リモート）のメソッド定義が異なる場合
- 設定しているアノテーションの値を変更した場合
- リデプロイ対象がモジュールプロジェクト（EJB プロジェクト、Web プロジェクト）だけの場合

7.5.3 バージョン 07-20 とバージョン 07-50 以降の機能差

バージョン 07-50 以降では、MyEclipse のプロジェクトの入れ替え機能が改善されました。バージョン 07-20 と 07-50 以降の機能差について説明します。

(1) プロジェクトの入れ替え機能の違い

バージョン 07-50 以降では、デプロイ形式でプロジェクトの入れ替え機能が異なります。次の表にバージョン 07-20 と 07-50 以降の入れ替え機能の違いを示します。

表 7-1 07-20 と 07-50 以降の入れ替え機能

バージョン	プロジェクトの入れ替え機能
07-20	プロジェクトをアンデプロイしてからデプロイします。
07-50 以降	デプロイ形式で入れ替え機能が異なります。 ・ 展開ディレクトリ形式 リロード機能でプロジェクトを入れ替えます。 ・ アーカイブ形式 リデプロイ機能でプロジェクトを入れ替えます。

(2) 展開ディレクトリ形式で反映されるファイルの範囲

バージョン 07-20 と 07-50 以降では、展開ディレクトリ形式でプロジェクトを入れ替えるときに変更内容が反映されるファイルの範囲が変更されています。

変更内容が反映されるファイルの範囲については、マニュアル「Cosminexus 機能解説」にある更新検知の対象となるファイルについての説明を参照してください。

(3) 展開ディレクトリ形式でのモジュールプロジェクトの入れ替え

バージョン 07-50 以降のリロード機能は、エンタープライズアプリケーションプロジェクト単位で使用できます。しかし、展開ディレクトリ形式のモジュールプロジェクト (EJB プロジェクトまたは Web プロジェクト) はリロード機能に対応していないため、プロジェクトの入れ替えが正常に実行されません。

入れ替え機能で変更が反映されないファイルは、次の操作をして変更を反映してください。

DD を変更した場合

エンタープライズアプリケーションプロジェクトをアンデプロイしたあとに、デプロイしてください。

JSP ファイル、クラスファイルなどのファイルを追加、削除した場合


Server Plug-in やサーバ管理コマンドで J2EE アプリケーションを停止したあとに、J2EE アプリケーションを開始してください。

7.6 J2EE アプリケーションのデバッグ

作成した J2EE アプリケーションをデバッグします。次の手順に従ってデバッグしてください。


! 注意事項

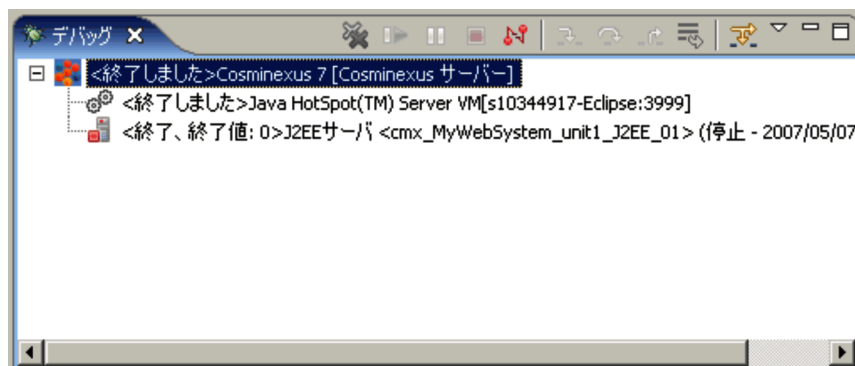
ブレークポイントによって J2EE アプリケーションのプログラム・スレッドの実行が中断されている場合、中断されている J2EE アプリケーションに対する操作は失敗するおそれがあります。J2EE アプリケーションに対する操作を実行する場合は、デバッグを終了させてから実行してください。

1. [パッケージ・エクスプローラー] ビューでデバッグ対象であるプロジェクトの Java ソースファイルを開いて、ブレークポイントを設定します。
2. [サーバ] ビューで [Cosminexus 7] を選択して、ビューツールバーから  (デバッグ・モードで開始)) をクリックします。
3. [Web ブラウザー] ビューで起動 URL を指定して、J2EE アプリケーションを実行します。
ブレークポイントに達すると、[デバッグ] パースペクティブが表示されて、デバッグが実行されます。

参考

リロード機能が有効になっている場合は、デバッグ後にプログラムを修正したときに J2EE アプリケーションの入れ替えが自動的に実行されます。明示的に J2EE アプリケーションを入れ替えたい場合は、「7.5.1 プロジェクトの入れ替え手順」を参考にして、プロジェクトを入れ替えてください。

4. デバッグが終了したら、[デバッグ] ビューで [Cosminexus 7[Cosminexus サーバー]] を選択して、ツールバーの  (終了)) をクリックします。
デバッグが終了します。



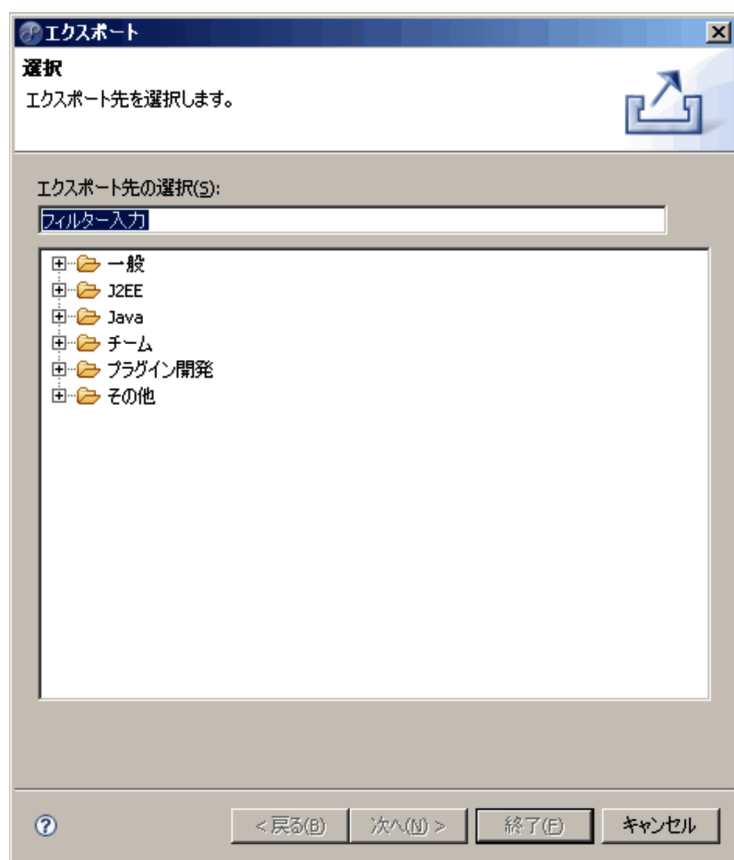
7.7 J2EE アプリケーションのエクスポート

MyEclipse のエクスポート機能を使用すると、J2EE アプリケーションを EAR ファイルとしてエクスポートできます。テスト用実行環境のアプリケーションサーバに J2EE アプリケーションを配布する場合に、J2EE アプリケーションをエクスポートして EAR ファイルを生成してください。

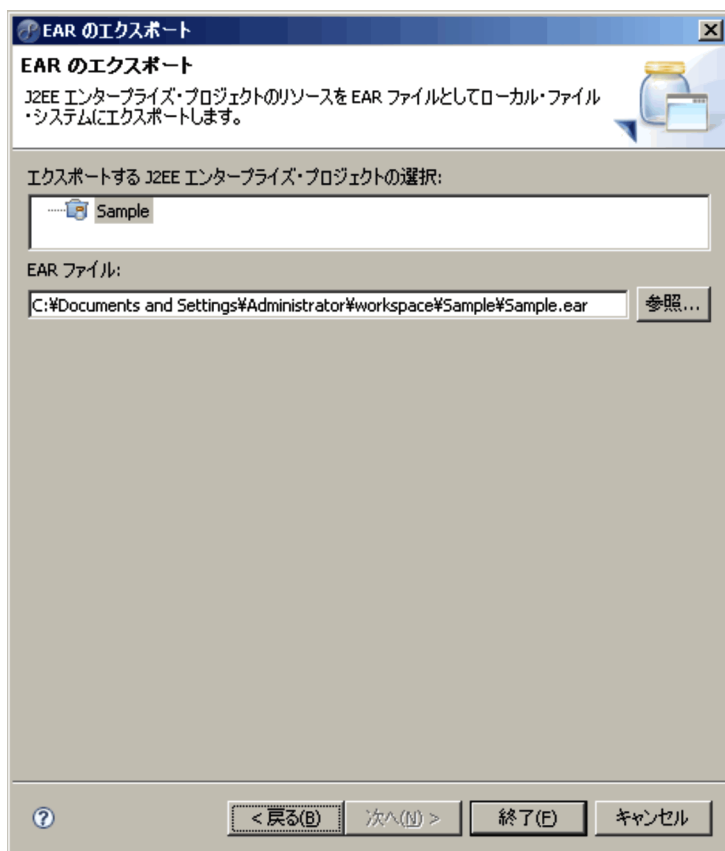
なお、J2EE アプリケーションを実行環境に配布する場合は、「8.2 EAR ファイルの作成」の手順に従って EAR ファイルを作成してください。

エクスポートの手順を次に示します。

1. Eclipse のメニューから [ファイル] - [エクスポート] を選択します。
[エクスポート] ダイアログが表示されます。



2. [J2EE] - [EAR ファイル (MyEclipse)] を選択し、[次へ] ボタンをクリックします。
[EAR のエクスポート] ページが表示されます。



設定が必要な項目を次に示します。

項目名	設定値
エクスポートする J2EE エンタープライズ・プロジェクトの選択	エクスポート対象となるエンタープライズアプリケーションプロジェクトを選択します。
EAR ファイル	エクスポート先のファイルパスを指定します。

3. [終了] ボタンをクリックします。

エンタープライズアプリケーションプロジェクトを、EAR ファイルとしてエクスポートします。

参考

J2EE アプリケーション開発時に、Server Plug-in を使用して J2EE アプリケーションの属性や動作を定義している場合は、Server Plug-in を使用してアプリケーション統合属性ファイルもエクスポートしてください。Server Plug-in での操作については、マニュアル「Cosminexus アプリケーション設定操作ガイド」のアプリケーション統合属性ファイルのエクスポートに関する説明を参照してください。

7.8 MyEclipse 使用時の注意事項

ここでは、MyEclipse 使用時の注意事項として次の内容を説明します。

- MyEclipse と Server Plug-in を併用する場合
- MyEclipse および Server Plug-in で表示される J2EE サーバ名
- MyEclipse で J2EE アプリケーションのデプロイおよびアンデプロイができなくなった場合の対処方法

7.8.1 MyEclipse と Server Plug-in を併用する場合

MyEclipse と Server Plug-in の両方を使用する場合は、次の内容に注意してください。

(1) Server Plug-in で J2EE サーバを起動している場合

MyEclipse から J2EE サーバを起動するときは、J2EE サーバが停止していることを確認してください。

Server Plug-in で起動した J2EE サーバを、MyEclipse から操作することはできません。MyEclipse で J2EE アプリケーションを操作する場合は MyEclipse で J2EE サーバを起動してください。

! 注意事項

サーバ管理コマンド、または Smart Composer 機能のコマンドで J2EE サーバを起動している場合、MyEclipse から操作できません。

(2) MyEclipse 以外のツールで同じ名称の J2EE アプリケーションをインポートしている場合

Server Plug-in やサーバ管理コマンドで、同じ名称・同じ形式の J2EE アプリケーションがすでにインポートされている場合に、MyEclipse で J2EE サーバにプロジェクトをデプロイすると、J2EE アプリケーションの入れ替えが実施されます。インポート済みの J2EE アプリケーションのデプロイ形式がアーカイブ形式の場合は、J2EE アプリケーションはリデプロイされます。また、インポート済みの J2EE アプリケーションのデプロイ形式が展開ディレクトリ形式の場合は、J2EE アプリケーションはリロードされます。このとき、J2EE アプリケーションの構成が異なる場合は、正しく入れ替えできません。入れ替えができなかった場合は、MyEclipse でいったん J2EE アプリケーションをアンデプロイしてから、再度デプロイしてください。

7.8.2 MyEclipse および Server Plug-in で表示される J2EE サーバ名

MyEclipse の [設定] ダイアログの [Cosminexus 7] ページにある [Cosminexus サーバー] ツリービュー, および Server Plug-in の [サーバー・エクスプローラー] ビューに表示される J2EE サーバ名には, 論理サーバ名が表示されます。

インスタントセットアップ機能で J2EE サーバを構築した場合は, 次の名称が表示されます。

```
cmx_InstantWebSystem_unit1_j2ee_01
```

参考

MyEclipse および Server Plug-in で表示される J2EE サーバ名は, 簡易構築定義ファイルの `<logical-server-name>` タグで定義します。`<logical-server-name>` タグを省略した場合は次に示す名称が設定されます。

```
cmx_<Webシステム名>_<サービスユニット名>_<サーバの種別>_<通番>
```

なお, Smart Composer 機能で構築した J2EE サーバの名称を確認するには, Smart Composer 機能で提供する `cmx_list_model` コマンドを使用します。J2EE サーバ名の確認方法については, マニュアル「Cosminexus システム運用ガイド」のコマンド実行時に指定する Web システム名, サービスユニット名, および J2EE サーバ名に関する注意事項に関する説明を参照してください。また, `cmx_list_model` コマンドの使用方法については, マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

7.8.3 デプロイまたはアンデプロイできない場合の対処方法

MyEclipse のプロジェクトと J2EE アプリケーションの関係が一致なくなると, MyEclipse で, J2EE アプリケーションのデプロイ, アンデプロイ, および入れ替えができなくなります。MyEclipse のプロジェクトのデプロイ, アンデプロイ, および入れ替えができなくなった場合は, 次の手順で対象となる J2EE アプリケーションをアンデプロイしてください。

1. サーバ管理コマンド (`cjstopapp`) で J2EE アプリケーションを停止します。
2. サーバ管理コマンド (`cjdeleteapp`) で J2EE アプリケーションを削除します。
3. デプロイ作業ディレクトリのディレクトリまたは EAR ファイルを削除します。
エンタープライズアプリケーションプロジェクトのディレクトリまたは EAR ファイルをデプロイ作業ディレクトリから削除します。
なお, デプロイ作業ディレクトリは, 「2.5.2 デプロイ作業ディレクトリの設定変更」で設定したディレクトリになります。

4. MyEclipse の [サーバー] ビューから対象となるエンタープライズアプリケーションプロジェクトをアンデプロイします。

なお、サーバ管理コマンドについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

7.9 Server Plug-in から起動した J2EE サーバでのデバッグ

MyEclipse では、Server Plug-in またはサーバ管理コマンドを使って、デバッグ起動している J2EE サーバにある J2EE アプリケーションをデバッグできます。

ここでは、Server Plug-in またはサーバ管理コマンドを使用した J2EE サーバのデバッグ起動から MyEclipse を使用したデバッグまでの手順を説明します。

(1) J2EE サーバのデバッグ起動

Server Plug-in またはサーバ管理コマンドを使用して J2EE サーバをデバッグ起動します。

1. <Cosminexus インストールディレクトリ>%CC%server%usrconf%ejb%< デバッグ対象のサーバ名>%usrconf.cfg ファイルに次の内容を記述します。

```
add.jvm.arg=-agentlib:jdwp=transport=dt_socket,server=y,address=<デバッグ・ポート番号>,suspend=n
```

インスタントセットアップ機能で構築した環境を使用する場合は、デバッグポート番号 3999 で設定されています。

2. Server Plug-in またはサーバ管理コマンドで J2EE サーバを起動します。

Server Plug-in を使用した J2EE サーバの起動方法については、マニュアル

「Cosminexus アプリケーション設定操作ガイド」を参照してください。

サーバ管理コマンドで J2EE サーバを起動する場合は、次のコマンドを実行します。

コマンドについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

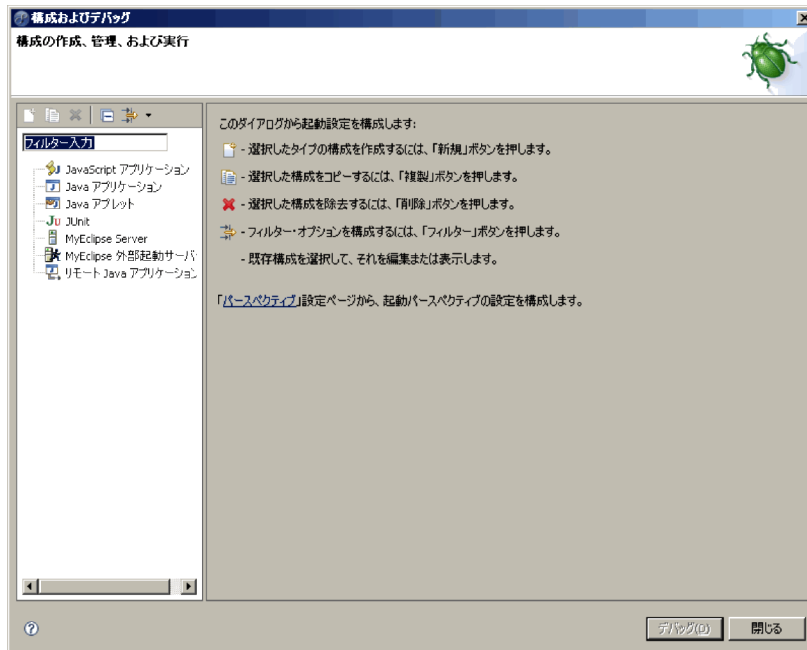
```
cjstartsv [<サーバ名称>]
```


インスタントセットアップ機能で構築した環境を使用する場合は、サーバ名称に「InstantJ2EEServer」を指定します。

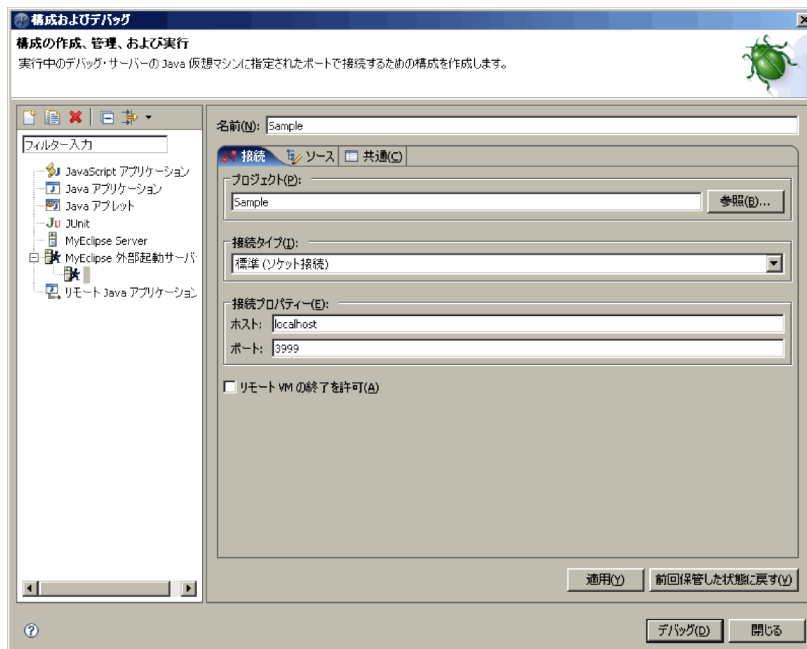
(2) プロジェクトのデバッグの設定

デバッグに必要な起動構成を作成します。

1. [パッケージ・エクスプローラー] ビューでデバッグするプロジェクトを選択して、Eclipse のメニューから [実行] - [構成およびデバッグ] を選択します。
[構成およびデバッグ] ダイアログが表示されます。



2. 左ペインのツリーで [MyEclipse 外部起動サーバー] を選択して、メニューバーの [ (新規の起動構成)] をクリックします。
右ペインに起動構成を作成するページが表示されます。



設定が必要な項目を次に示します。

項目名	設定値
名前	任意の名称を指定します。
プロジェクト	デバッグするプロジェクトのプロジェクト名を指定します。
接続タイプ	「標準（ソケット接続）」を選択します。
接続プロパティ のホスト	接続する J2EE サーバのホスト名を指定します。 インスタントセットアップ機能で構築した環境を使用する場合は、 「localhost」を指定します。
接続プロパティ のポート	usrconf.cfg ファイルに指定したデバッグ・ポート番号を指定します。 インスタントセットアップ機能で構築した環境を使用する場合は、 「3999」を指定します。

注

エンタープライズアプリケーションプロジェクトをデバッグする場合は、プロジェクトを空白にして、[ソース] タブの [ソース・ルックアップ・パス] に使用するソースを追加します。

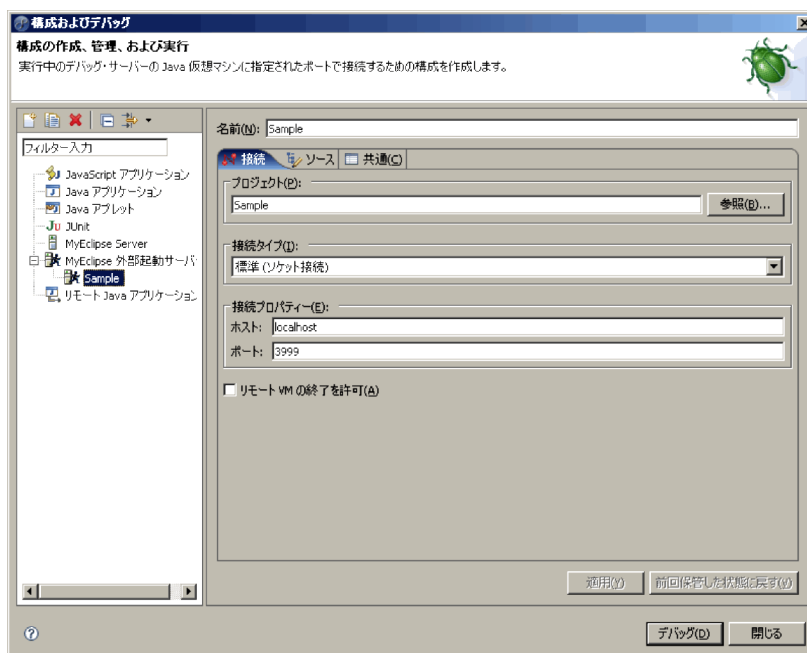
3. [閉じる] ボタンをクリックします。

[変更を保管しますか?] ダイアログが表示された場合は、[はい] ボタンをクリックしてください。[構成およびデバッグ] ダイアログが閉じて、デバッグのための起動構成が作成されます。

(3) デバッグの実行


MyEclipse からデバッグを実行します。


1. [パッケージ・エクスプローラー] ビューでデバッグするプロジェクトの Java ソース ファイルを開いて、ブレークポイントを設定します。
2. Eclipse のメニューから [実行] - [構成およびデバッグ] を選択します。
[構成およびデバッグ] ダイアログが表示されます。



- 作成した起動構成を選択して、[デバッグ] ボタンをクリックします。
デバッグが開始されます。J2EE アプリケーションを実行してブレークポイントに達すると、[デバッグ] パースペクティブが表示されます。[デバッグ] パースペクティブの [デバッグ] ビューには、MyEclipse 外部起動サーバの起動構成が表示されます。

参考

[構成およびデバッグ] ダイアログからデバッグしたあとに、再度デバッグする場合は Eclipse のメニューバーにある [ (デバッグ)] のプルダウンメニューから起動構成を選択することで開始することもできます。

- デバッグが終わったら、[デバッグ] ビューのツールバーにある [ (切断)] をクリックします。
デバッグが終了します。

8

実行環境への J2EE アプリケーションの配布

この章では、開発した J2EE アプリケーションを実行環境に配布する方法について説明します。

8.1 J2EE アプリケーション配布の流れ

8.2 EAR ファイルの作成

8.3 Connector 属性ファイルのエクスポート

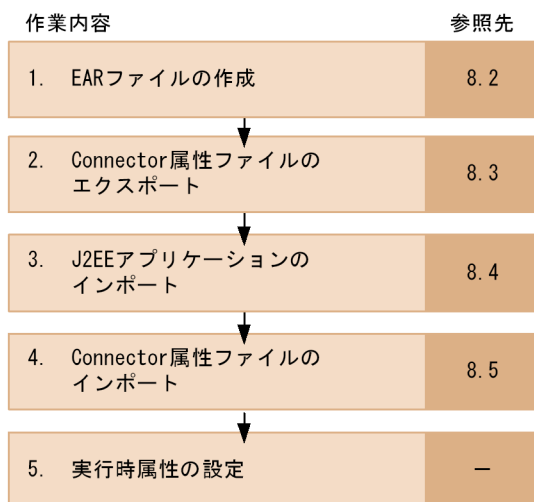
8.4 実行環境への J2EE アプリケーションのインポート

8.5 実行環境への Connector 属性ファイルのインポート

8.1 J2EE アプリケーション配布の流れ

開発環境にある J2EE アプリケーションをエクスポートして、実行環境に配布します。J2EE アプリケーション配布の流れを次に示します。

図 8-1 J2EE アプリケーション配布の流れ



それぞれの作業の概要を説明します。

1. EAR ファイルの作成

ビルドファイルを使用してプロジェクトをビルドして、EAR ファイルを作成します。

2. Connector 属性ファイルのエクスポート

開発環境で設定したリソースアダプタの属性を取得するために、Connector 属性ファイルをエクスポートします。エクスポートには Server Plug-in を使用します。

3. J2EE アプリケーションのインポート

ビルドファイルを使用して作成した J2EE アプリケーション（EAR ファイル）を実行環境にインポートします。インポートには、Server Plug-in またはサーバ管理コマンドを使用します。

4. Connector 属性ファイルのインポート

Connector 属性ファイルを、実行環境にインポートします。インポートには、Server Plug-in を使用します。

5. 実行時属性の設定

実行環境にインポートした J2EE アプリケーションに実行時属性を設定します。また、必要に応じて、J2EE リソースにも実行時属性を設定します。J2EE アプリケーションおよび J2EE リソースの実行時属性の設定には、Server Plug-in またはサーバ

管理コマンドを使用します。実行時属性の設定方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

以降の節では、この流れに沿って J2EE アプリケーションを配布する手順を説明します。

8.2 EAR ファイルの作成

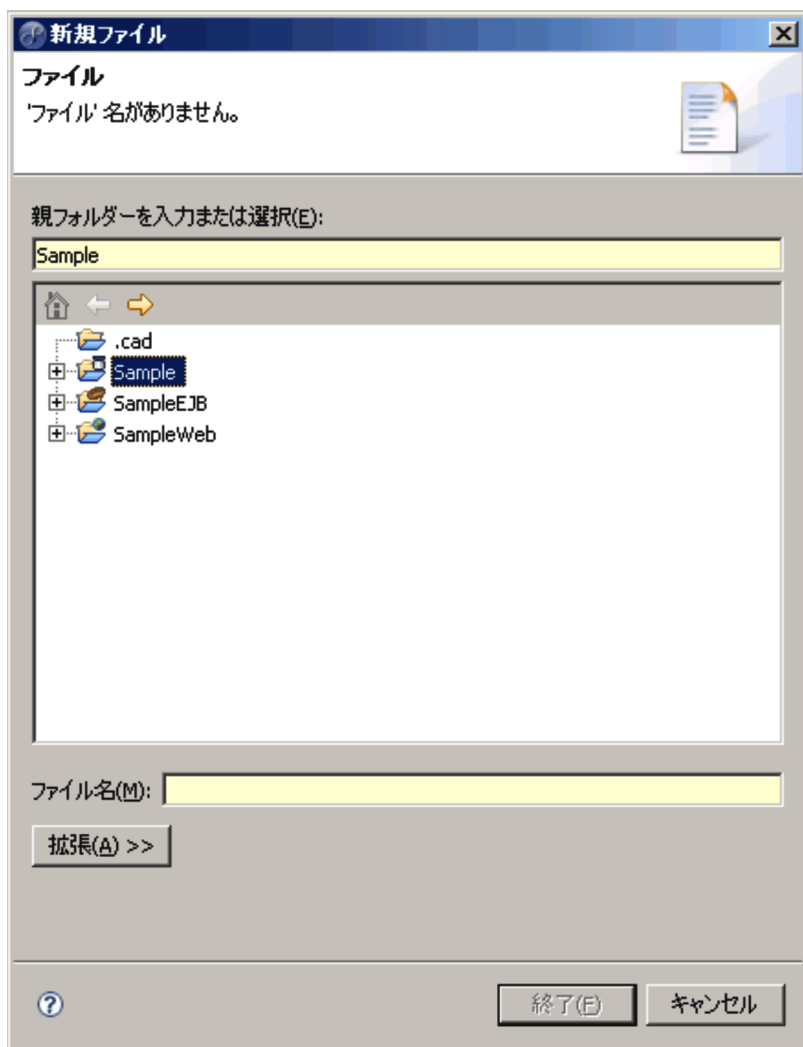
開発した J2EE アプリケーションを実行環境に配布する場合には、ビルドファイル (build.xml) を使用して MyEclipse のプロジェクトをビルドし、EAR ファイルを作成します。ここで作成した EAR ファイルを実行環境に配布します。なお、Java ソースファイルのコンパイルには、Cosminexus の javac コマンドおよび J2EE ライブラリが使用されます。

ここでは、build.xml の作成手順および EAR ファイルの作成手順を説明します。

8.2.1 ビルドファイルの作成

ビルドファイル (build.xml) の作成手順を説明します。

1. [パッケージ・エクスプローラー] ビューで build.xml を追加するプロジェクトを選択して、コンテキストメニューから [新規] - [ファイル] を選択します。
[新規ファイル] ダイアログが表示されます。



2. build.xml を追加するプロジェクトを選択した状態で [ファイル名] に「build.xml」を入力して、[終了] ボタンをクリックします。
選択したプロジェクトに build.xml が追加されます。

8.2.2 ビルドファイルの編集・実行

プロジェクトの種類によって、ビルドファイル (build.xml) に記述する内容が異なります。それぞれのプロジェクトの種類に合わせて、ビルドファイルを編集してください。ビルドファイルを作成する順序は次のとおりです。

1. EJB プロジェクトのビルド
2. Web プロジェクトのビルド
3. エンタープライズアプリケーションプロジェクトのビルド

この流れに沿って、各プロジェクトのビルドファイルの編集と実行について説明します。

(1) EJB プロジェクトのビルド

EJB プロジェクトのビルドファイルに次の内容を記述します。ビルドファイルを実行すると、EJB-JAR ファイルが生成されます。

表 8-1 EJB プロジェクトのビルドファイルの例

行番号	記述例
1	<?xml version="1.0" encoding="UTF-8"?>
2	<project name="Bank_EJB" default="create" basedir="."/>
3	<property environment="myEnv"/>
4	<property name="classPath" value="\${myEnv.COSMINEXUS_HOME}/CC/ client/lib/j2ee-javax.jar"/>
5	<property name="ejbArchiveName" value="Bank_EJB.jar"/>
6	<property name="javacPath" value="\${myEnv.COSMINEXUS_HOME}/jdk/ bin/javac.exe"/>
7	<property name="tempFolder" value="ant"/>
8	<target name="create">
9	<delete file="./\${ejbArchiveName}"/>
10	<mkdir dir="./\${tempFolder}"/>
11	<copy todir="./\${tempFolder}">
12	<fileset dir="./src" excludes="**/*.java"/>
13	</copy>
14	<javac srcdir="./src" destdir="./\${tempFolder}" executable="\${javacPath}" classpath="\${classPath}"/>
15	<jar destfile="./\${ejbArchiveName}" basedir="./\${tempFolder}"/>
16	<delete dir="./\${tempFolder}"/>
17	</target>
18	</project>

ビルドファイルの内容の意味は、次のとおりです。

表 8-2 EJB プロジェクトのビルドファイルの記述内容

行番号	記述内容の意味
1	XML 宣言です。
2	ビルドファイルのルートを指定します。
3	プロパティに、ビルドファイルからシステム環境変数を参照する設定を追加します。
4	プロパティに、Cosminexus の J2EE ライブラリのパスを指定します。

行番号	記述内容の意味
5	プロパティに、EJB プロジェクトで生成するアーカイブファイル名の指定を追加します。
6	プロパティに、Cosminexus の javac コマンドのパスを指定します。
7	プロパティに、ビルドで使用する一時ディレクトリ名を指定します。ディレクトリ名は任意の名称を指定します。
8	ターゲット「create」の開始タグです。
9	既存のアーカイブファイルを削除します。
10	ビルド用の一時ディレクトリを作成します。
11	ビルド用の一時ディレクトリに、アーカイブに含めるファイルをコピーします。
12	Java ソースをコピー対象外にします。
13	11 行目のコピータスクの終了タグです。
14	Java ソースをコンパイルします。コンパイルには Cosminexus の javac コマンドを使用します。
15	ビルド用の一時ディレクトリ以下のフォルダおよびファイルをアーカイブして、jar ファイルを生成します。
16	ビルド用の一時ディレクトリを削除します。
17	8 行目のターゲット「create」の終了タグです。
18	ビルドファイルの終了タグです。

(2) Web プロジェクトのビルド

Web プロジェクトのビルドファイルに、次の内容を記述します。ビルドファイルを実行すると、WAR ファイルが生成されます。

なお、Web プロジェクトのビルドファイルは EJB プロジェクトのアーカイブファイルを参照しています。Web プロジェクトをビルドする前に、必ず EJB プロジェクトをビルドしてください。

表 8-3 Web プロジェクトのビルドファイルの例

行番号	記述例
1	<?xml version="1.0" encoding="UTF-8"?>
2	<project name="Bank_Web" default="create" basedir="."/>
3	<property environment="myEnv"/>
4	<property name="cjjspcdir" value="\${myEnv.COSMINEXUS_HOME}/CC/web/bin"/>
5	<property name="classPath" value="\${myEnv.COSMINEXUS_HOME}/CC/client/lib/j2ee-javax.jar;../Bank_EJB/Bank_EJB.jar"/>
6	<property name="javacPath" value="\${myEnv.COSMINEXUS_HOME}/jdk/bin/javac.exe"/>

8. 実行環境への J2EE アプリケーションの配布

行番号	記述例
7	<property name="tempFolder" value="ant"/>
8	<property name="webArchiveName" value="Bank_Web.war"/>
9	<property name="webProjectPath" value="E:/eclipse/workspace/ Bank_Web/" />
10	<property name="webRoot" value="WebRoot"/>
11	<target name="compile">
12	<exec executable="\${cjjspcdir}/cjjspc.bat" newenvironment="true">
13	<arg line='-source 1.4' />
14	<arg line='-pageencoding UTF-8' />
15	<arg line='-root "\${webProjectPath}/\${tempFolder}"' />
16	</exec>
17	</target>
18	<target name="create">
19	<delete file="./\${webArchiveName}" />
20	<mkdir dir="./\${tempFolder}" />
21	<copy todir="./\${tempFolder}">
22	<fileset dir="./\${webRoot}" excludes="**/classes/**/*.*.class" />
23	</copy>
24	<javac srcdir="./src" destdir="./\${tempFolder}/WEB-INF/classes" executable="\${javacPath}" classpath="\${classpath}" />
25	<antcall target="compile" />
26	<war destfile="./\${webArchiveName}" basedir="./\${tempFolder}" webxml="./\${tempFolder}/WEB-INF/web.xml" excludes="WEB-INF/ web.xml" />
27	<delete dir="./\${tempFolder}" />
28	</target>
29	</project>

ビルドファイルの内容の意味は、次のとおりです。

表 8-4 Web プロジェクトのビルドファイルの記述内容

行番号	記述内容の意味
1	XML 宣言です。
2	ビルドファイルのルートを指定します。
3	プロパティに、ビルドファイルからシステム環境変数を参照する指定を追加します。
4	プロパティに、Cosminexus の JSP 事前コンパイル用コマンドのパスを指定します。

行番号	記述内容の意味
5	プロパティに、Cosminexus の J2EE ライブラリ、および EJB プロジェクトのアーカイブのパスを指定します。
6	プロパティに、Cosminexus の javac コマンドのパスを指定します。
7	プロパティに、ビルドで使用する一時ディレクトリ名を指定します。ディレクトリ名は任意の名称を指定します。
8	プロパティに、Web プロジェクトで生成するアーカイブファイル名を指定します。
9	プロパティに、Web プロジェクトの絶対パスを指定します。
10	プロパティに、Web プロジェクトのルート指定します。
11	JSP 事前コンパイル用のターゲット「compile」の開始タグです。
12	Cosminexus の JSP 事前コンパイル用のコマンドを実行します。
13	コマンドの引数を指定します。指定したバージョンの Java 言語を使用して事前コンパイルします。
14	コマンドの引数として、JSP のデフォルトの文字エンコーディングを指定します。
15	コマンドの引数として、Web プロジェクトのルートに一時ディレクトリを指定します。
16	12 行目のコマンド実行の終了タグです。
17	11 行目のターゲット「compile」の終了タグです。
18	ターゲット「create」のルート指定します。
19	既存のアーカイブファイルを削除します。
20	ビルド用の一時ディレクトリを作成します。
21	アーカイブに含めるファイルを一時ディレクトリにコピーします。
22	Java クラスをコピー対象外にします。
23	21 行目のコピータスクの終了タグです。
24	Java ソースをコンパイルします。コンパイルには Cosminexus の javac コマンドを使用します。
25	JSP 事前コンパイル用のターゲット「compile」を呼び出します。JSP 事前コンパイルをしない場合は記述しません。
26	一時ディレクトリ以下のディレクトリおよびファイルをアーカイブして、WAR ファイルを生成します。
27	一時ディレクトリを削除します。
28	18 行目のターゲット「create」の終了タグです。
29	ビルドファイルの終了タグです。

(3) エンタープライズアプリケーションプロジェクトのビルド

エンタープライズアプリケーションプロジェクトのビルドファイルに、次の内容を記述します。ビルドファイルを実行すると、EAR ファイルが生成されます。

なお、エンタープライズアプリケーションプロジェクトをビルドすると、EJB プロジェクト、Web プロジェクト、エンタープライズアプリケーションプロジェクトの順にビルドされて、EJB-JAR ファイルと WAR ファイルを含む EAR ファイルが生成されます。

エンタープライズアプリケーションプロジェクトをビルドする前に、必ず EJB プロジェクトおよび Web プロジェクトのビルドファイルを作成してください。

表 8-5 エンタープライズアプリケーションプロジェクトのビルドファイルの例

行番号	記述例
1	<?xml version="1.0" encoding="UTF-8"?>
2	<project name="Bank" default="create" basedir="."/>
3	<property name="earArchiveName" value="bank.ear"/>
4	<property name="ejbArchiveName" value="Bank_EJB.jar"/>
5	<property name="ejbProjectName" value="Bank_EJB"/>
6	<property name="webArchiveName" value="Bank_Web.war"/>
7	<property name="webProjectName" value="Bank_Web"/>
8	<property name="excludes" value=".classpath,.mymetadata,.project,build-user.xml"/>
9	<target name="create">
10	<ant antfile="../\${ejbProjectName}/build.xml" dir="../\${ejbProjectName}" inheritall="false"/>
11	<ant antfile="../\${webProjectName}/build.xml" dir="../\${webProjectName}" inheritall="false"/>
12	<delete file="../\${earArchiveName}"/>
13	<ear destfile="../\${earArchiveName}" basedir="." appxml="./META-INF/application.xml" excludes="\${excludes}">
14	<fileset file="../\${ejbProjectName}/\${ejbArchiveName}"/>
15	<fileset file="../\${webProjectName}/\${webArchiveName}"/>
16	</ear>
17	</target>
18	</project>

ビルドファイルの内容の意味は、次のとおりです。

表 8-6 エンタープライズアプリケーションプロジェクトのビルドファイルの記述内容

行番号	記述例
1	XML 宣言です。
2	ビルドファイルのルートを指定します。
3	プロパティに、エンタープライズアプリケーションプロジェクトで生成するアーカイブファイル名を指定します。

行番号	記述例
4	プロパティに、EJB プロジェクトで生成するアーカイブファイル名を指定します。
5	プロパティに、EJB プロジェクトの名称を指定します。
6	プロパティに、Web プロジェクトで生成するアーカイブファイル名を指定します。
7	プロパティに、Web プロジェクトの名称を指定します。
8	プロパティに、EAR ファイル生成時に除くファイルを指定します。
9	ターゲット「create」の開始タグです。
10	EJB プロジェクトのビルドファイルを実行します。
11	Web プロジェクトのビルドファイルを実行します。
12	既存のアーカイブファイルを削除します。
13	EAR ファイルを生成します。
14	EAR ファイルに含める EJB-JAR ファイルを指定します。
15	EAR ファイルに含める WAR ファイルを指定します。
16	13 行目の ear タスクの終了タグです。
17	9 行目のターゲット「create」の終了タグです。
18	ビルドファイルの終了タグです。

8.3 Connector 属性ファイルのエクスポート

開発環境で Server Plug-in を使用してリソースアダプタの属性を設定している場合、設定した属性情報を取得するために、Connector 属性ファイルのエクスポートします。エクスポートした Connector 属性ファイルは、実行環境にインポートします。

Connector 属性ファイルのエクスポートには Server Plug-in を使用します。Server Plug-in から Management Server リモート管理機能にログインして、Server Plug-in の [サーバー・エクスプローラー] ビューで操作します。

Server Plug-in での Connector 属性ファイルのエクスポートについては、マニュアル「Cosminexus アプリケーション設定操作ガイド」の Connector 属性のエクスポートに関する説明を参照してください。

8.4 実行環境への J2EE アプリケーションのインポート

ビルドファイルを使用して作成した J2EE アプリケーション (EAR ファイル) を実行環境にインポートします。EAR ファイルを実行環境にインポートするには、Server Plug-in またはサーバ管理コマンドを使用します。

なお、EAR ファイルを実行環境にインポートするときに、J2EE アプリケーションの形式を選択できます。EAR ファイルを、アーカイブ形式としてインポートするか、展開ディレクトリ形式としてインポートするかによって、インポートの方法が異なります。J2EE アプリケーションの形式ごとに、インポート方法を説明します。

アーカイブ形式の J2EE アプリケーションとしてインポートする

EAR ファイルをアーカイブ形式の J2EE アプリケーションとしてインポートするには、Server Plug-in またはサーバ管理コマンドを使用します。

- Server Plug-in を使用する場合
Server Plug-in の [サーバー・エクスプローラー] ビューで操作します。
- サーバ管理コマンドを使用する場合
cjimportapp コマンドに -f オプションを指定して、コマンドを実行します。-f オプションには EAR ファイルを指定します。

インポート手順については、マニュアル「Cosminexus アプリケーション設定操作ガイド」のアーカイブ形式の J2EE アプリケーションのインポートに関する説明を参照してください。

展開ディレクトリ形式の J2EE アプリケーションとしてインポートする

EAR ファイルを展開ディレクトリ形式の J2EE アプリケーションとしてインポートするには、サーバ管理コマンドを使用します。

cjimportapp コマンドに -f オプションおよび -d オプションを指定して、コマンドを実行します。-f オプションには EAR ファイルを、-d オプションには EAR ファイルの展開先ディレクトリを指定します。

なお、Server Plug-in では、EAR ファイルを展開ディレクトリ形式の J2EE アプリケーションとしてインポートすることはできません。

インポート手順については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の展開ディレクトリ形式の J2EE アプリケーションのインポートに関する説明を参照してください。

8.5 実行環境への Connector 属性ファイルのインポート

開発環境でエクスポートした Connector 属性ファイルを実行環境にインポートします。Connector 属性ファイルを実行環境にインポートするには、Server Plug-in を使用します。Server Plug-in から Management Server リモート管理機能にログインして、Server Plug-in の [サーバー・エクスプローラー] ビューで操作します。

Connector 属性ファイルのインポートについては、マニュアル「Cosminexus アプリケーション設定操作ガイド」の Connector 属性のインポートに関する説明を参照してください。

9

その他の開発

MyEclipse を使用した J2EE アプリケーション開発では，フレームワークを利用したり，UML を利用して J2EE アプリケーションの設計をしたりすることもできます。この章では，MyEclipse でのフレームワークを利用した開発，および UML を利用した開発の概要について説明します。

また，Eclipse を使用しない J2EE アプリケーション開発についても説明します。

9.1 フレームワークを利用した開発

9.2 UML を利用した開発

9.3 Eclipse を使用しない J2EE アプリケーションの開発

9.1 フレームワークを利用した開発

MyEclipse では、フレームワークを利用して J2EE アプリケーションを開発することもできます。ここでは、JSF を利用した開発および Struts を利用した開発の概要を説明します。

9.1.1 JSF を利用した開発

JSF は、Web アプリケーションのユーザインタフェースを開発するためのフレームワークです。MVC アーキテクチャの View であるユーザインタフェースの構成要素、Swing が持つようなイベント処理とコンポーネントのレンダリング要素を部品として持っています。それらの部品をフォームに配置することでユーザインタフェースを作成できます。また、Controller であるサーブレットで Web アプリケーションを管理する機能も持っています。

JSF を利用し、MyEclipse で J2EE アプリケーションを開発する場合の流れを次に示します。

1. プロジェクトの作成

JSF を利用する新規の Web プロジェクトを作成します。

2. メッセージバンドルの作成

J2EE アプリケーションで使用するメッセージバンドル（メッセージとその関連キーを格納するプロパティファイル）を作成します。

3. マネージド Bean の作成

JSP ページにユーザが入力したデータを格納したり、JSP ページから要求された処理を実行したりするマネージド Bean を作成します。

4. JSP ページの作成

J2EE アプリケーションの JSP ページを作成します。

5. J2EE アプリケーションの実行

JSF を利用した J2EE アプリケーションをデプロイして実行します。

JSF を利用した J2EE アプリケーション開発方法については、MyEclipse のオンラインヘルプを参照してください。

9.1.2 Struts を利用した開発

Struts は、Web アプリケーションを開発するためのフレームワークです。MVC アーキテクチャの Controller であるサーブレットで Web アプリケーションを管理する機能があります。また、このほかに、ユーザインタフェースの構成要素を部品として持っています。それらの部品をフォームに配置することでユーザインタフェースを作成できます。

Struts を利用し，MyEclipse で J2EE アプリケーションを開発する場合の流れを次に示します。

1. プロジェクトの作成

Struts を利用する新規の Web プロジェクトを作成します。

2. Struts コンポーネントの作成

Struts アプリケーションは，開発成果物のカテゴリから構成されます。次に示すコンポーネントを作成します。

- JSP
- Action
- ActionForward
- ActionForm

3. Struts アプリケーションの構築

作成した Struts コンポーネントを利用して，Struts アプリケーションのフローをレイアウトします。

4. J2EE アプリケーションの実行

Struts を利用した J2EE アプリケーションをデプロイして実行します。

Struts を利用した J2EE アプリケーション開発方法については，MyEclipse のオンラインヘルプを参照してください。

9.2 UML を利用した開発

MyEclipse では、UML を利用した設計をサポートしています。ここでは、UML を利用した開発について説明します。

9.2.1 UML とは

UML はグラフィカルな記法の一つで、ソフトウェアシステム、特にオブジェクト指向スタイルを使用して構築するシステムの記述や設計に利用されます。UML では次の表に示すダイアグラムが規定されています。

表 9-1 UML のダイアグラムと目的

ダイアグラム	目的
アクティビティ図	手続き的な振る舞いを記述します。
クラス図	クラス、特性、および関係を記述します。
コラボレーション図	リンクを重視したオブジェクト間の相互作用を記述します。
シーケンス図	シーケンスを重視したオブジェクト間の相互作用を記述します。
ステートチャート図	オブジェクトの状態の変化を記述します。
配置図	成果物の配置を記述します。
ユースケース図	ユーザがシステムとどう対話するかを記述します。

これらのダイアグラムを使用して、開発する J2EE アプリケーションの要求定義と分析をします。

9.2.2 MyEclipse での UML を利用した設計

MyEclipse では、UML を利用した J2EE アプリケーション開発をするために、MyEclipse UML という機能を提供しています。MyEclipse UML では、次に示す機能を提供しています。

UML 図のエディタ

次に示す UML 図を作成するためのエディタを提供しています。

- アクティビティ図
- クラス図
- コラボレーション図
- シーケンス図
- ステートチャート図
- 配置図
- ユースケース図

モデルとソースコードの相互変換

エディタで作成した UML モデルから Java ソースコードを生成したり、Java ソースコードからクラス図を生成したりする機能を提供しています。この機能を、ラウンドトリップ・エンジニアリングといいます。

リポジトリ・ファイルでの UML モデルの管理

UML モデルのリポジトリ・ファイルとは、UML 図を管理するためのファイルです。リポジトリ・ファイルを使用することで、複数の UML 図を一つのファイルで管理できます。

XMI 1.0 フォーマットでの UML モデルのエクスポート

XMI 1.0 フォーマットで UML モデルをエクスポートできます。

イメージ・フォーマットでの UML 図のエクスポート

エディタを使用して作成した UML 図を、GIF、PNG、PS、EPS、SVG 形式でエクスポートできます。

なお、MyEclipse UML の用法については、MyEclipse のオンラインヘルプを参照してください。

9.3 Eclipse を使用しない J2EE アプリケーションの開発

ここでは、Eclipse を使用しない J2EE アプリケーションの開発方法として、Java コマンドによるコンパイル方法とアーカイブ方法について説明します。また、RMI-IIOP を利用する場合に必要なスタブを用意する方法、および JSP ソースのチェック方法について説明します。

9.3.1 Eclipse を使用しない J2EE アプリケーションの開発の概要

Eclipse を使用しない J2EE アプリケーションの開発環境、および J2EE アプリケーション開発の流れを説明します。

(1) Eclipse を使用しない場合の開発環境

Cosminexus システムで動作する J2EE アプリケーションは、Eclipse を使用しないで開発することもできます。Eclipse を使用しないで J2EE アプリケーションを開発するときに必要な環境を、次の表に示します。

表 9-2 J2EE アプリケーションの開発環境（Eclipse を使用しない場合）

項番	分類	プログラム名称 / 構成ソフトウェア名称
1	OS	次に示す OS に対応しています。 Windows の場合 <ul style="list-style-type: none"> • Windows Server 2003 • Windows XP • Windows Vista UNIX の場合 <ul style="list-style-type: none"> • AIX • HP-UX • Linux • Solaris
2	JDK	Cosminexus Developer's Kit for Java
3	J2EE アプリケーション開発環境	Cosminexus Component Container

(2) J2EE アプリケーション開発の流れ（Eclipse を使用しない場合）

Eclipse を使用しない場合の J2EE アプリケーション開発の流れは、J2EE アプリケーションの形式によって異なります。J2EE アプリケーションの形式ごとに開発の流れを説明します。

なお、J2EE アプリケーションからほかの J2EE アプリケーションに含まれる Enterprise Bean や、ほかのサーバ上で動作する Enterprise Bean を呼び出す場合には、「9.3.6 RMI-IIOP スタブの用意」も参照してください。

展開ディレクトリ形式の J2EE アプリケーション開発の流れ

展開ディレクトリ形式の場合の J2EE アプリケーション開発の流れを次の図に示します。

図 9-1 展開ディレクトリ形式の J2EE アプリケーション開発の流れ（Eclipse を使用しない場合）

作業内容	参照先
1. アプリケーションディレクトリの作成	9.3.2
↓	
2. プログラムの作成	—
↓	
3. プログラムのコンパイル	9.3.3
↓	
4. 実行環境への配布	9.3.5

（凡例）—：なし

1. アプリケーションディレクトリの作成
展開ディレクトリ形式の J2EE アプリケーションを開発する場合、決められたディレクトリ構成を準備する必要があります。展開ディレクトリ形式のアプリケーションディレクトリを作成します。
2. プログラムの作成
任意のエディタを使用して、プログラムを作成します。
3. プログラムのコンパイル
javac コマンドを使用して、プログラムをコンパイルします。
4. 実行環境への配布
作成したプログラムを実行環境へ配布します。

アーカイブ形式の J2EE アプリケーション開発の流れ

アーカイブ形式の場合の J2EE アプリケーション開発の流れを次の図に示します。

図 9-2 アーカイブ形式の J2EE アプリケーション開発の流れ (Eclipse を使用しない場合)

作業内容	参照先
1. プログラムの作成	—
↓	
2. プログラムのコンパイル	9.3.3
↓	
3. アーカイブの作成	9.3.4
↓	
4. 実行環境への配布	9.3.5

(凡例) — : なし

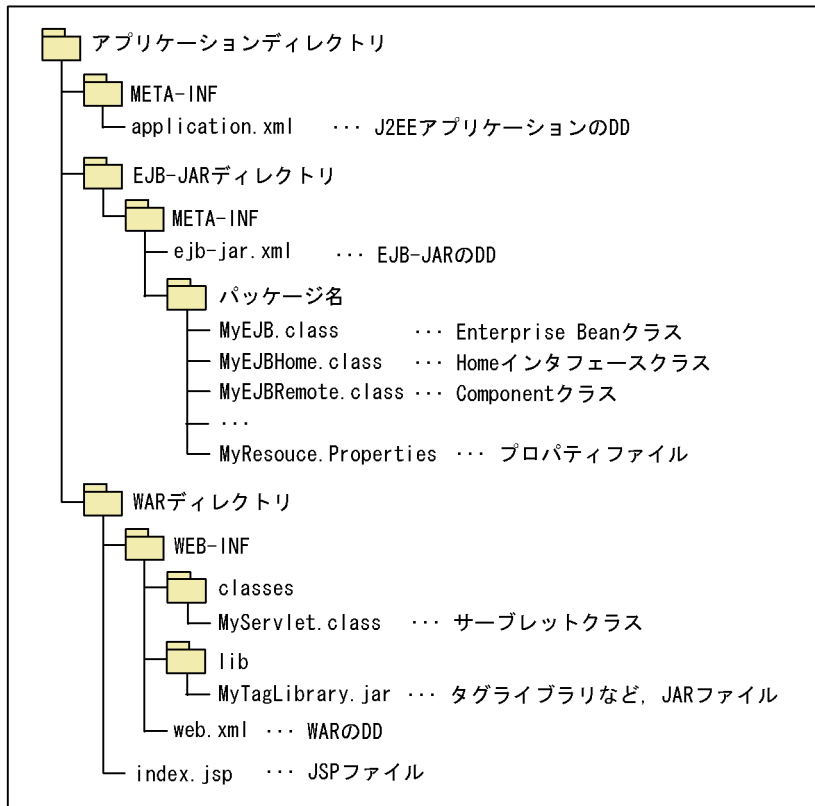
1. プログラムの作成
任意のエディタを使用して、プログラムを作成します。
2. プログラムのコンパイル
javac コマンドを使用して、プログラムをコンパイルします。
3. アーカイブの作成
jar コマンドを使用して、WAR ファイル、EJB-JAR ファイル、EAR ファイルを作成します。
4. 実行環境への配布
作成したプログラムを実行環境へ配布します。

9.3.2 アプリケーションディレクトリの作成 (展開ディレクトリ)

Developer が提供するプラグインを使用しないで展開ディレクトリ形式の J2EE アプリケーションを開発する場合、最初にアプリケーションディレクトリを作成します。

作成するアプリケーションディレクトリの構成例を、次の図に示します。

図 9-3 アプリケーションディレクトリの構成例



アプリケーションディレクトリ名は任意です。EJB-JAR ディレクトリと WAR ディレクトリは、アプリケーションディレクトリの直下に作成する必要はありません。

EJB-JAR ディレクトリおよび WAR ディレクトリは、それぞれ application.xml の <module> タグ以下に指定して作成します。application.xml の記述例を次に示します。

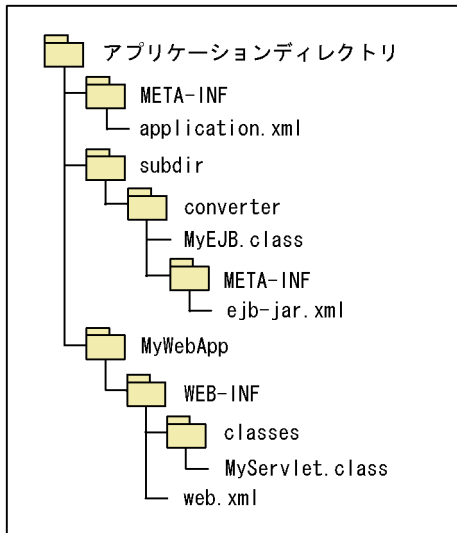
```

<?xml version="1.0" encoding="UTF-8" ?>
<application xmlns="http://java.sun.com/xml/ns/j2ee"
  version="1.4" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/application_1_4.xsd">
  <display-name>converter</display-name>
  <module>
    <ejb>subdir/converter.jar</ejb>
  </module>
  <module>
    <web>
      <web-uri>MyWebApp.war</web-uri>
      <context-root>/</context-root>
    </web>
  </module>
</application>

```

この application.xml に対応するアプリケーションディレクトリの構成を、次に示します。

図 9-4 アプリケーションディレクトリの構成



application.xml の定義内容とアプリケーションディレクトリの対応

- EJB-JAR ディレクトリおよび WAR ディレクトリは、それぞれ <module> タグ以下に指定した EAR ファイル上の相対パスに作成します。
- <module> タグ以下に定義する EJB-JAR および WAR のモジュールは、拡張子（.jar または .war）を付けて宣言します。
- 作成される EJB-JAR ディレクトリおよび WAR ディレクトリの名称は、拡張子（.jar または .war）を取り除いた名称となります。
- アプリケーションディレクトリ以下に置かれている拡張子が小文字の JAR ファイル（.jar）のうち、<module> タグ以下に定義していない JAR ファイルは、ライブラリ JAR として扱われます。

注意

- アプリケーションディレクトリとして UNC パスは指定できません。指定した場合は、サーバ管理コマンドでインポートするときにエラーとなります。
- J2EE サーバ起動処理中やサーバ管理コマンド実行中に、アプリケーションディレクトリ以下のファイルおよびディレクトリを追加、削除、上書きしないでください。
- application.xml に <alt-dd> タグが指定されている J2EE アプリケーションは、展開ディレクトリ形式で使用できません。展開ディレクトリ形式の J2EE アプリケーションをインポートする場合に、J2EE アプリケーションの application.xml に <alt-dd> タグが指定されているときは、コマンド実行時にエラーとなります。
- <module> タグ以下に定義する EJB-JAR および WAR のモジュールで、異なる拡張子

- 張子を付けた場合、J2EE アプリケーションのインポートに失敗します。
- アプリケーションディレクトリ以下に Java ソースファイルを置くような構成は推奨しません。同じディレクトリに置かれているクラスファイルと Java ソースファイルの同期が取れていない場合、J2EE アプリケーションの開始やリロードに失敗する場合があります。
 - 半角記号「!」「#」「%」「+」はエスケープ文字として扱われるため、アプリケーションディレクトリ名およびモジュール名に指定しないでください。
 - アプリケーションディレクトリ以下に置かれている JAR ファイルはライブラリ JAR として扱われるため、アプリケーションディレクトリ以下には、`cjgetstubsjar` コマンドで取得したスタブおよびインタフェースを置かないでください。

9.3.3 Java プログラムのコンパイル (javac コマンド)

Eclipse を使用しないで J2EE アプリケーションを開発する場合、J2EE アプリケーションを構成するプログラム (JSP、サーブレットなど) をテキストエディタなどで実装し、JDK の `javac` コマンドによってコンパイルします。使用する JDK については、「9.3.1(1) Eclipse を使用しない場合の開発環境」を参照してください。

次に、コンパイル時に指定する `javac` コマンドのコンパイルオプションを示します。

`-classpath` (クラスパス)

- ソースファイル中に J2EE 標準 API だけを使用している場合
`<Cosminexus のインストールディレクトリ>/CC/client/lib/j2ee-javax.jar` を追加します。
- パッケージ名が `com.hitachi` で始まる Cosminexus 独自の API (タイムアウト設定 API など) を使用している場合
`<Cosminexus のインストールディレクトリ>/CC/client/lib/HiEJBClientStatic.jar` と `<Cosminexus のインストールディレクトリ>/TPB/lib/vbjorb.jar` を追加します。
- 統合ユーザ管理が提供する API を使用している場合
`<Cosminexus のインストールディレクトリ>/manager/lib/usradmin.jar` を指定します。

サーバ起動・停止フック機能を使用して処理を実装する場合は、クラスパスに `ejbserver.jar` を追加する必要があります。`ejbserver.jar` の指定については、マニュアル「Cosminexus 機能解説」のコンテナ拡張ライブラリ利用の流れに関する説明を参照してください。

！ 注意事項

サーバ起動・停止フック機能は、サーブレット、JSP、および Enterprise Bean などのプログラムからは使用できません。

-encoding (ソースファイルのエンコーディング)

javac 実行時のデフォルトエンコーディングとソースファイルのエンコーディングが異なるときに指定します。

-g (デバッグオプション)

デバッグ情報を生成します。「-g:none」は推奨しません。「-g:none」を指定すると例外発生時に行番号が取得できないため、デバッグの効率が低下します。

9.3.4 アーカイブ形式の J2EE アプリケーションの作成

Developer が提供するプラグインを使用しないでアーカイブ形式の J2EE アプリケーションを開発する場合、jar コマンドを使用してアーカイブを作成します。ここでは、WAR ファイル、EJB-JAR ファイル、および EAR ファイルの作成について説明します。

(1) WAR ファイルの作成 (jar コマンド)

jar コマンドを使用して、Web アプリケーションとして使用するプログラムやファイルを WAR ファイルにアーカイブします。WAR ファイルにアーカイブする場合のディレクトリ構成、および jar コマンドの指定例を示します。

(a) WAR ファイルのディレクトリ構成

表 9-3 jar コマンドによるアーカイブ時のディレクトリ構成 (WAR)

ディレクトリ名またはファイル名	説明	必須
/	アーカイブ内のルートディレクトリです。	
任意	JSP や HTML、そのほかの web クライアントからアクセスされるファイルです。WEB-INF 以外のサブディレクトリに格納することもできます。	-
/WEB-INF/	web クライアントから直接アクセスできないファイルを格納するディレクトリです。	
web.xml	servlet 仕様で規定された DD です。	
*.tld	JavaServer Pages(TM) 仕様で規定されたタグライブラリ・ディスクリプタのファイルです。サブディレクトリの下に格納してもかまいません。	-
/WEB-INF/classes/	サーブレットやそのほかのクラスファイルを格納するディレクトリです。格納するファイルがないときは不要です。	-
.class	パッケージ名称に従って、ディレクトリ階層にサーブレットやそのほかのクラスファイル (.class) を格納します。	-

ディレクトリ名またはファイル名	説明	必須
/WEB-INF/lib/	サーブレットやその他のクラスを含む JAR ファイル (*.jar) を格納するディレクトリです。格納するファイルがないときは不要です。	-
.jar	サーブレットやその他のクラスを含む JAR ファイル (.jar) を格納します。	-
/META-INF/	管理情報を格納するディレクトリです。jar コマンドによって自動的に作成されます。	-
MANIFEST.MF	jar コマンドの m オプションで指定されたファイルが格納されます。用意しなくてもアーカイブ内に自動的に作成されます。	-

(凡例)

：アーカイブするときに必須であることを示します

- ：該当しません

(b) WAR ファイル作成時の jar コマンド指定例

カレントディレクトリおよびサブディレクトリに次のようにファイルが用意されているとします。

```
index.html
howto.jsp
feedback.jsp
images/banner.gif
images/jumping.gif
WEB-INF/web.xml
WEB-INF/lib/jspbean.jar
WEB-INF/classes/com/mycorp/servlets/MyServlet.class
WEB-INF/classes/com/mycorp/util/MyUtils.class
```

このとき、次のようにコマンドを実行すると、「MyApp.war」という名称の WAR ファイルが作成されます。

```
jar cf ../MyApp.war .
```

(2) EJB-JAR ファイルの作成 (jar コマンド)

jar コマンドを使用して、Enterprise Bean や DD を EJB-JAR ファイルにアーカイブします。EJB-JAR ファイルにアーカイブする場合のディレクトリ構成、および jar コマンドの指定例を示します。

(a) EJB-JAR ファイルのディレクトリ構成

表 9-4 jar コマンドによるアーカイブ時のディレクトリ構成 (EJB-JAR)

ディレクトリ名またはファイル名	説明	必須
/	アーカイブ内のルートディレクトリです。	
.class	Enterprise Bean のクラス, Enterprise Bean のホームインタフェース, コンポーネントインタフェース, ビジネスインタフェース, およびそれらが依存する J2SE や J2EE 規定外のクラス (インタフェース) のクラスファイル (.class) をパッケージ名に従ったディレクトリ階層で格納します。	
/META-INF/	管理情報を格納するディレクトリです。jar コマンドによって自動的に作成されます。	-
ejb-jar.xml	EJB 仕様で規定された DD です。	
MANIFEST.MF	マニフェストファイルです。jar コマンドの m オプションを指定することで, アーカイブ内に自動的に作成されます。	-

(凡例)

: アーカイブするときに必須であることを示します

- : 該当しません

注

EJB2.0 仕様の Enterprise Bean では必須です。アノテーションに対応した Enterprise Bean では不要です。

(b) EJB-JAR ファイル作成時の jar コマンド指定例

カレントディレクトリおよびサブディレクトリに次のようにファイルが用意されているとします。

```
com/mycorp/account/Account.class
com/mycorp/account/AccountEJB.class
com/mycorp/account/AccountHome.class
com/mycorp/account/InsufficientBalanceException.class
META-INF/ejb-jar.xml
```

このとき, 次のようにコマンドを実行すると, 「MyEJB.jar」という名称の EJB-JAR ファイルが作成されます。

```
jar cf ../MyEJB.jar .
```

(3) EAR ファイルの作成 (jar コマンド)

jar コマンドを使用して, WAR ファイル, EJB-JAR ファイル, ライブラリ JAR ファイル, および DD を EAR ファイルにアーカイブします。EAR ファイルにアーカイブする場合のディレクトリ構成, および jar コマンドの指定例を示します。

(a) EAR ファイルのディレクトリ構成

表 9-5 jar コマンドによるアーカイブ時のディレクトリ構成 (EAR)

ディレクトリ名またはファイル名	説明	必須
/	アーカイブ内のルートディレクトリです。	
*.jar , *.war	J2EE アプリケーションに含まれる EJB-JAR ファイル, WAR ファイル, およびそのほかの JAR ファイルです。 hitachi-runtime.jar という名称は使用できません。 application.xml に J2EE アプリケーションの構成モジュールとして定義されていない拡張子が小文字の JAR ファイル (.jar) は, ライブラリ JAR として扱われ, J2EE アプリケーションのクラスパスに追加されます。	
/META-INF/	管理情報を格納するディレクトリです。jar コマンドによって自動的に作成されます。	-
application.xml	J2EE 仕様で規定された DD です。	
MANIFEST.MF	マニフェストファイルです。jar コマンドの m オプションを指定することで, アーカイブ内に自動的に作成されます。	-

(凡例)

: アーカイブするときに必須であることを示します

- : 該当しません

(b) EAR ファイル作成時の jar コマンド指定例

カレントディレクトリおよびサブディレクトリに次のようにファイルが用意されているとします。

```
MyEJB.jar
MyUtil.jar
MyApp.war
META-INF/application.xml
```

このとき, 次のようにコマンドを実行すると, 「MyApp.ear」という名称の EAR ファイルが作成されます。

```
jar cf ../MyApp.ear .
```

9.3.5 実行環境への配布

Eclipse のプラグインを使用しない場合, 実行環境へ J2EE アプリケーションを配布するには, サーバ管理コマンドを使用して, J2EE アプリケーションのインポートや属性の設定をします。

(1) J2EE アプリケーションのインポート

J2EE アプリケーションをインポートします。J2EE アプリケーションの形式ごとにインポート方法を説明します。

(a) 展開ディレクトリ形式の J2EE アプリケーションの場合

J2EE アプリケーションを J2EE サーバにインポートするには、`cjimportapp` コマンドを使用します。展開ディレクトリ形式の J2EE アプリケーションをインポートするには、アプリケーションディレクトリをインポートする方法と、EAR または ZIP 形式の J2EE アプリケーションを展開ディレクトリ形式の J2EE アプリケーションとしてインポートする方法があります。それぞれのインポート方法について説明します。

なお、`cjimportapp` コマンドの使用方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の展開ディレクトリ形式の J2EE アプリケーションのインポートに関する説明を参照してください。

アプリケーションディレクトリのインポート

作成したアプリケーションディレクトリを J2EE サーバにインポートするには、`cjimportapp` コマンドに `-a` オプションを指定します。

すでに J2EE サーバ上に同じアプリケーションディレクトリを持つ J2EE アプリケーションがある場合、インポートできません。

EAR ファイル / ZIP ファイルのインポート

EAR ファイルとして作成済みの J2EE アプリケーションまたは Cosminexus からエクスポートした実行時情報付き ZIP ファイルを、展開ディレクトリ形式にしてインポートするには、`cjimportapp` コマンドに `-d` オプションを指定します。

インポートした EAR ファイル / ZIP ファイル内のディレクトリ、EJB-JAR ファイル、および WAR ファイルの名称によって、作成されるディレクトリ名に衝突が起こることがあります。ディレクトリ名の衝突については、マニュアル「Cosminexus 機能解説」に記載されているディレクトリ名の生成規則の説明を参照してください。

(b) アーカイブ形式の J2EE アプリケーションの場合

EAR ファイルとして作成済みの J2EE アプリケーションまたは Cosminexus からエクスポートした実行時情報付き ZIP ファイルを、アーカイブ形式でインポートするには、`cjimportapp` コマンドを使用します。

なお、`cjimportapp` コマンドの使用方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」のアーカイブ形式の J2EE アプリケーションのインポートに関する説明を参照してください。

(2) J2EE アプリケーションの属性設定

J2EE アプリケーションの属性は、次に示すどちらかの方法で設定します。アノテーションを使用していない場合は、サーバ管理コマンドを使用して DD を修正します。アノ

テーションを使用している場合は、クラスファイルのアノテーションを修正します。それぞれの方法を説明します。

サーバ管理コマンドを使用して DD を修正する

`cjsetappprop` コマンドを使用して、J2EE アプリケーションの DD を書き換えます。`cjsetappprop` コマンドの使用方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の J2EE アプリケーションのプロパティ設定に関する説明を参照してください。

クラスファイルのアノテーションを修正する

Java ソースコードに記述したアノテーションを編集して、コンパイルします。J2EE アプリケーションの開始時に変更した情報が読み込まれます。なお、アプリケーションディレクトリ以下の DD を修正しても、J2EE サーバは検知しません。

(3) J2EE アプリケーションの開始

`cjstartapp` コマンドを使用して J2EE アプリケーションを開始します。

`cjstartapp` コマンドの使用方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の J2EE アプリケーションの開始に関する説明を参照してください。

(4) クラスファイルの変更

Java ソースファイルを修正した場合は、その Java ソースファイルをコンパイルしてクラスファイルを生成します。

(5) J2EE アプリケーションのリロード (展開ディレクトリ形式)

展開ディレクトリ形式の J2EE アプリケーションの場合、リロード機能を使用できます。リロード機能を使用していると、J2EE サーバはクラスファイルの更新を検知し、更新後のクラスファイルを自動的に再読み込みします。リロード機能を使用するには、アプリケーションサーバにあらかじめリロードの設定をしておく必要があります。また、`cjreloadapp` コマンドを使用すると、任意のタイミングでリロードを実行できます。

J2EE アプリケーションの更新検知とリロードを有効にするための設定については、マニュアル「Cosminexus システム構築ガイド」の J2EE アプリケーションの更新検知とリロードの設定に関する説明を参照してください。`cjreloadapp` コマンドでのリロードについては、マニュアル「Cosminexus アプリケーション設定操作ガイド」の展開ディレクトリ形式の J2EE アプリケーションに関する説明を参照してください。

(6) J2EE アプリケーションの停止

`cjstopapp` コマンドを使用して J2EE アプリケーションを停止します。

`cjstopapp` コマンドの使用方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の J2EE アプリケーションの停止に関する説明を参照してください。

(7) J2EE アプリケーションの削除

`cjdeleteapp` コマンドを使用して J2EE アプリケーションを削除します。このとき、削除対象の J2EE アプリケーションのアプリケーションディレクトリは削除されません。

J2EE サーバへの登録は解除されます。

`cjdeleteapp` コマンドの使用方法については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の J2EE アプリケーションの削除に関する説明を参照してください。

9.3.6 RMI-IIOP スタブの用意

J2EE アプリケーションからほかの J2EE アプリケーションに含まれる Enterprise Bean や、ほかのサーバ上で動作する Enterprise Bean を呼び出す場合には、呼び出し側に RMI-IIOP スタブを用意する必要があります。RMI-IIOP スタブを用意する方法は、次の 3 とおりです。

J2EE サーバによってデプロイ時に自動的に作成

呼び出し先 Enterprise Bean のリモートインタフェースを呼び出し元の J2EE アプリケーションに含めます。呼び出し元サーバの定義ファイル (`usrconf.properties`) の指定を変更することで、J2EE アプリケーションの開始時にスタブクラスが自動生成されます。呼び出し元サーバの定義ファイルでは、次を指定します。

```
ejbserver.deploy.stub.generation.scope=app
```

J2EE サーバによって呼び出し先サーバから自動的に取得

呼び出し先サーバの定義ファイル (`usrconf.properties`) の指定で、実行時にスタブクラスが自動的にダウンロードされるようにします。呼び出し先サーバの定義ファイルでは、次を指定します。

```
ejbserver.DynamicStubLoading.Enabled=true
```

手動で取得して J2EE アプリケーション内に含める

呼び出す Enterprise Bean をデプロイし、デプロイしたサーバからスタブクラス (`stubs.jar`) を取得して呼び出し元の J2EE アプリケーションに含めます。

RMI-IIOP スタブが自動的に作成、取得される場合、および手動で取得する場合の、呼び出し先サーバと呼び出し元サーバの J2EE アプリケーションのディレクトリ構成を示します。

(1) RMI-IIOP スタブが自動的に作成、取得される場合の J2EE アプリケーションのディレクトリ構成

デプロイ時に RMI-IIOP スタブが自動的に作成される場合、または実行時に RMI-IIOP スタブが自動的にダウンロードされる場合の、呼び出し先の Enterprise Bean を含む J2EE アプリケーションの構成を次に示します。

表 9-6 自動で作成，取得する場合の J2EE アプリケーションの構成（呼び出し先）

ディレクトリ名またはファイル名	説明
/	アーカイブ内のルートディレクトリです。
myejb.jar	呼び出し先の Enterprise Bean を含む EJB-JAR ファイルです。
/META-INF/	管理情報を格納するディレクトリです。jar コマンドによって自動的に作成されます。
application.xml	J2EE 仕様で規定された DD です。

デプロイ時に RMI-IIOP スタブが自動的に作成される場合，または実行時に RMI-IIOP スタブが自動的にダウンロードされる場合の，呼び出し元のサーブレットを含む J2EE アプリケーションの構成を次に示します。

表 9-7 自動で作成，取得する場合の J2EE アプリケーションの構成（呼び出し元）

ディレクトリ名またはファイル名	説明
/	アーカイブ内のルートディレクトリです。
myapp.war	呼び出し元のサーブレットが含まれる Web アプリケーションです。WAR ファイル内のディレクトリ構成例を示します。 /（ルートディレクトリ） /WEB-INF/ web.xml /WEB-INF/lib/ *.jar（呼び出し先 Enterprise Bean のホームインタフェース，コンポーネントインタフェースを含む JAR ファイル） /WEB-INF/classes/ *.class（呼び出し元のサーブレットのクラスファイル）
/META-INF/	管理情報を格納するディレクトリです。jar コマンドによって自動的に作成されます。
application.xml	J2EE 仕様で規定された DD です。

（2）RMI-IIOP スタブを手動で取得する場合の J2EE アプリケーションのディレクトリ構成

手動でスタブクラスを取得して，呼び出し元の J2EE アプリケーションに含ませる場合の，J2EE アプリケーションのディレクトリ構成を示します。呼び出し先の J2EE アプリケーションの構成は，「（1）RMI-IIOP スタブが自動的に作成，取得される場合の J2EE アプリケーションのディレクトリ構成」で示す呼び出し先の構成と同じです。

表 9-8 手動で含ませる場合の J2EE アプリケーションの構成 (呼び出し元)

ディレクトリ名またはファイル名	説明
/	アーカイブ内のルートディレクトリです。
myapp.war	<p>呼び出し元のサーブレットが含まれる Web アプリケーションです。WAR ファイル内のディレクトリ構成例を示します。</p> <p>/ (ルートディレクトリ)</p> <p>/WEB-INF/</p> <p> web.xml</p> <p>/WEB-INF/lib/</p> <p> *.jar (呼び出し先 Enterprise Bean のホームインタフェース, コンポーネントインタフェースを含む JAR ファイル)</p> <p> stubs.jar (手動で取得した RMI-IIOP スタブを含む JAR ファイル)</p> <p>/WEB-INF/classes/</p> <p> *.class (呼び出し元のサーブレットのクラスファイル)</p>
/META-INF/	管理情報を格納するディレクトリです。jar コマンドによって自動的に作成されます。
application.xml	J2EE 仕様で規定された DD です。

注

クラスローダの構成上, スタブクラスとリモートインタフェースは両方とも同じ場所 (/WEB-INF/lib) に配置する必要があります。

10 JSP 事前コンパイルと チェック

J2EE アプリケーション開発で JSP ファイルを事前にコンパイルしておくことで、実行環境で J2EE アプリケーションを実行したときの、初回リクエストのレスポンスタイムを短縮できます。この章では、`cjjspc` コマンドを使用した JSP ファイルの事前コンパイルについて説明します。また、Cosminexus が提供する `cjjsp2java` コマンドを使用して JSP ファイルから Java ソースファイルを生成し、JSP ファイルに記述ミスがないか確認する方法について説明します。

10.1 JSP ファイルの事前コンパイル

10.2 JSP ソースのチェック

10.1 JSP ファイルの事前コンパイル

J2EE アプリケーションをデプロイおよび実行する前に、Web アプリケーションに含まれる JSP ファイルをコンパイルできます。この機能を JSP 事前コンパイル機能といいます。JSP ファイルを事前にコンパイルしておくことで、JSP ファイルに対する初回リクエスト時のレスポンスタイムを短縮できます。

JSP 事前コンパイルは次に示す方法で実行できます。

cjjspc コマンドを使用して実行する

cjjspc コマンドを使用して任意のタイミングでコンパイルします。事前コンパイルの対象は、Web アプリケーションに含まれるすべての JSP ファイル、または個々の JSP ファイルです。

また、MyEclipse で J2EE アプリケーションを開発する場合も、cjjspc コマンドを使用して JSP 事前コンパイルを実行できます。ただし、JSP 事前コンパイルを実行した場合は、JSP 事前コンパイルで出力された Java ソースファイルを MyEclipse のワークスペースに反映する必要があります。MyEclipse で J2EE アプリケーションを開発する場合の JSP 事前コンパイルについては、「10.1.12 MyEclipse で J2EE アプリケーションを開発する場合」を参照してください。

J2EE アプリケーション開始時に実行する

J2EE アプリケーション開始コマンド (cjstartapp) に、-jspc オプションを指定してコンパイルします。事前コンパイル対象は、Web アプリケーションに含まれるすべての JSP ファイルです。

このマニュアルでは、上記二つの方法のうち cjjspc コマンドを使用して実行する方法について説明します。

J2EE アプリケーション開始時に JSP ファイルの事前コンパイルを実行する方法については、マニュアル「Cosminexus システム運用ガイド」を参照してください。

10.1.1 JSP 事前コンパイル機能の前提条件

JSP 事前コンパイル機能を利用するときの前提条件を示します。次に示す条件を満たさない場合、正しくコンパイルされません。

- コンパイルする JSP ファイルが、Web アプリケーションのルート (Web ルートフォルダ) 下、およびそのサブディレクトリ下に格納されていること。
- Web アプリケーションのルート (Web ルートフォルダ) 下に、WEB-INF ディレクトリがあること。
- WEB-INF ディレクトリ直下に DD があること。

10.1.2 JSP 事前コンパイルの対象ファイルと出力ファイル

JSP 事前コンパイルを実行すると、JSP ファイルからクラスファイルが生成されます。ここでは、JSP 事前コンパイルを実行できる JSP ファイルの条件と、JSP 事前コンパイル時に生成される出力ファイルについて説明します。

(1) JSP 事前コンパイルの対象ファイル

次の条件に当てはまる JSP ファイルは、JSP 事前コンパイルを実行できます。

- JSP1.1, JSP1.2, JSP2.0 仕様のどれかに準拠した JSP ファイル ¹
- JSP2.0 仕様に準拠したタグファイル
- JSP ファイルおよびタグファイルから静的にインクルードされるファイル
- TLD ファイル (タグライブラリ・ディスクリプタ) ²
- DD
- コンパイルに必要なクラスライブラリ

注 1

JSP ファイルとみなされる条件を次に示します。詳細については、JSP 仕様を参照してください。

- 拡張子が「.jsp」のファイル
- 拡張子が「.jspx」のファイル (JSP2.0 以降の場合)
- DD の <jsp-file> タグに指定されたファイル
- DD の <jsp-property-group><url-pattern> タグに合致するファイル (JSP2.0 以降の場合)

注 2

タグファイルや静的インクルードされたファイルまたは TLD ファイルを更新した場合、更新したファイルを参照するすべての JSP ファイルをコンパイルしてください。

(2) JSP 事前コンパイルで出力されるファイル

JSP 事前コンパイルを実施すると、JSP ワークディレクトリが作成され、JSP ファイルから生成されたクラスファイルが JSP ワークディレクトリに出力されます。ただし、JSP 事前コンパイルの対象となる JSP ファイルが存在しない場合は、JSP ワークディレクトリは作成されません。

JSP 事前コンパイルの実行時に出力されるファイルを次に示します。

- JSP ファイルから生成された Java ソースファイルおよびクラスファイル
- タグファイルから生成された Java ソースファイルおよびクラスファイル

JSP 事前コンパイル機能を実行した Web アプリケーション内の WEB-INF ディレクトリに生成されます。

10.1.3 cjjspc コマンドを使用した JSP の事前コンパイル

cjjspc コマンドでは、Web アプリケーション単位または JSP ファイル単位で JSP ファイルをコンパイルできます。

それぞれの場合のコンパイル方法について説明します。また、JSP ファイルの格納先となる JSP ファイルパスの指定方法と、コマンド実行時のバージョンチェックについて説明します。

(1) Web アプリケーション単位の JSP 事前コンパイル

cjjspc コマンドに `-root` オプションを指定して実行することで、Web アプリケーションに含まれるすべての JSP ファイルを事前コンパイルできます。`-root` オプションには、事前コンパイル対象の Web アプリケーションのルートを、カレントディレクトリからの相対パスまたは絶対パスで指定します。

Web アプリケーション単位で事前コンパイルする場合の指定形式を示します。

```
> cjjspc -root <Webアプリケーションのルートディレクトリ>
```

Web アプリケーションのルートディレクトリが「D:¥app¥webapp1」の場合の、cjjspc コマンドの指定例（絶対パス）を示します。

```
> cjjspc -root d:¥app¥webapp1
```

オプションの指定によって、コンパイル対象外ファイルを指定できます。コンパイル対象外ファイルの指定方法については、「10.1.4 コンパイル不要な JSP ファイルの指定」を参照してください。コンパイル対象外ファイルを指定した場合は、JSP コンパイル結果のバージョンチェックが行われます。

コンパイル対象外ファイルを指定しないで Web アプリケーション単位の JSP 事前コンパイルを実行した場合、JSP コンパイル結果のバージョンチェックは行われません。Web アプリケーションに含まれるすべての JSP ファイルがコンパイルされます。

JSP コンパイル結果のバージョンチェックについては、「(4)(a) JSP 事前コンパイル結果のバージョンチェック」を参照してください。

(2) JSP ファイル単位の JSP 事前コンパイル

cjjspc コマンドに `-root` オプションおよび JSP ファイルパスを指定して実行することで、指定した JSP ファイルをコンパイルできます。

JSP ファイル単位で事前コンパイルする場合の指定形式を示します。

```
> cjjspc -root <Webアプリケーションのルートディレクトリ> <JSPファイルパス1> <JSPファイルパス2> ...
```


JSP ファイルパスの指定方法については、「(3) JSP ファイルパスの指定方法」を参照してください。

Web アプリケーションのルートディレクトリが「D:¥app¥webappl」, コンパイルする JSP ファイルが「D:¥app¥webappl¥jsp¥index.jsp」の場合の, cjjspc コマンドの指定例を示します。

```
> cjjspc -root d:¥app¥webappl /jsp/index.jsp
```

複数の JSP ファイルをコンパイルする場合は, スペースで区切って指定します。

```
> cjjspc -root d:¥app¥webappl /jsp/index.jsp /jsp/title.jsp
```

JSP ファイル単位の JSP 事前コンパイルを実行した場合, JSP コンパイル結果のバージョンチェックが行われます。DD で指定された Web アプリケーションのバージョンと, JSP コンパイル時の Web アプリケーションのバージョンが合致していなければ処理が中断されます。JSP コンパイル結果のバージョンチェックについては、「(4)(a) JSP 事前コンパイル結果のバージョンチェック」を参照してください。

(3) JSP ファイルパスの指定方法

JSP ファイルパスは, Web アプリケーションのルートディレクトリからの相対パスを指定します。パス区切り文字には「/」を使用し, パスの先頭には必ず「/」を付けます。例えば, Web アプリケーションのルートディレクトリ「D:¥app¥webappl」に含まれる JSP ファイル (D:¥app¥webappl¥jsp¥index.jsp) を指定する場合, JSP ファイルパスは「/jsp/index.jsp」となります。指定例を次に示します。

```
> cjjspc -root d:¥app¥webappl /jsp/index.jsp
```

また, JSP ファイルパスには, ワイルドカード文字 (*) を使用できます。ワイルドカードを使用した場合の指定例を示します。

```
> cjjspc -root d:¥app¥webappl "/jsp/*.jsp"
```

<ワイルドカードの使用方法>

ワイルドカードは,「/*」で終わる場合と「/*.拡張子」で終わる場合の二とおりが指定できます。それぞれの場合の指定例を示します。

- 「/*」で終わる場合

```
/aaa/bbb/*
```

/aaa/bbb/ 直下にあるすべての JSP ファイルがコンパイルされます。

- 「/*.拡張子」で終わる場合

```
/aaa/bbb/*.jsp
```

この場合、/aaa/bbb/ 下にあるファイルのうち、「.jsp」という拡張子を持つすべての JSP ファイルがコンパイルされます。

注意

- ワイルドカード文字を含むファイルパスを指定する場合、ファイルパスを " " (ダブルクォーテーション) で括ってください。
- 「/aaa/bbb/c*c」のように、* をワイルドカード文字として、任意の文字列を指定することはできません。

(4) cjjspc コマンドを使用した JSP 事前コンパイルの注意事項

cjjspc コマンドを使用して JSP 事前コンパイルを実行する場合の注意事項を示します。

(a) JSP 事前コンパイル結果のバージョンチェック

JSP ファイルから生成されるクラスファイルは、DD で指定された Web アプリケーションのバージョンに依存します。したがって、Web アプリケーションと異なるバージョンの JSP ファイルは、Web アプリケーションで使用できません。

cjjspc コマンドでは、コンパイル実行時に、Web アプリケーションと JSP ファイルのバージョンが合致するかをチェックします。バージョンチェックは次の場合に実行されます。

- コンパイル対象外ファイルを指定して、Web アプリケーション単位の JSP 事前コンパイルを実行したとき
- JSP ファイル単位の JSP 事前コンパイルを実行したとき

Web アプリケーションと JSP ファイルのバージョンが異なる場合は、cjjspc コマンドの処理が中断され、エラーが出力されます。エラーが出力された場合は、Web アプリケーション単位で、すべての JSP ファイルに対してコンパイルし直してください。

(b) DD および TLD ファイルの検証

cjjspc コマンドを実行するときに、DTD または XML スキーマに従っているかが検証されます。TLD ファイルがスキーマに従っていない場合は、コンパイル時にエラーとなります。DD がスキーマに従っていない場合は、エラーとなり、コマンドが終了します。

(c) JSP ファイルまたはタグファイルから生成された Java ソースのコンパイルオプション

cjjspc コマンドを使用して生成されたクラスファイルは、J2EE サーバでの実行時に使用されます。cjjspc コマンドでは、J2EE サーバで生成されたクラスファイルと同じクラスファイルを生成します。このため、JSP ファイルまたはタグファイルから生成された Java ソースをコンパイルするときのコンパイルオプションは、クラスパスおよび Java

言語仕様のバージョン以外は指定できません。Java 言語仕様のバージョンの指定方法については「10.1.7 JSP コンパイル時の Java 言語仕様のバージョン指定」を参照してください。

10.1.4 コンパイル不要な JSP ファイルの指定

Web アプリケーションに、コンパイル不要な JSP ファイルが含まれる場合、次のどちらかの方法で、コンパイル対象外ファイルに指定できます。

コンパイル対象外リストファイルを指定する方法

コンパイル対象外ファイルをコンパイル対象外リストファイルに記述し、コマンド実行時に指定します。

コンパイル対象外ファイルを直接指定する方法

コンパイル対象外ファイルをコマンド実行時に直接指定します。

それぞれの場合の指定方法について説明します。

(1) コンパイル対象外ファイルに指定するための前提条件

コンパイル対象外ファイルに指定する場合は、次の条件のうち、どちらかを満たす必要があります。

次のどれかに該当するファイルでコンパイルが不要なファイル

- 拡張子が「.jsp」または「.jspx」のファイル
- DD の <jsp-file> に指定されたファイル
- DD の <jsp-property-group><url-pattern> タグに合致するファイル
- cjspsc コマンドの引数に指定したファイル

コンパイル済みであり、再度コンパイルする必要のないファイル

(2) コンパイル対象外リストファイルを指定する方法

コンパイル対象外リストファイルの作成方法、およびコマンド実行時の指定方法について説明します。

(a) コンパイル対象外リストファイルの作成

テキストファイルにコンパイル対象外の JSP ファイルパスを記述します。コンパイル対象外リストファイルの記述形式を示します。

```
# コメント
<コンパイル対象外とするJSPファイルパス1>
<コンパイル対象外とするJSPファイルパス2>
...
```

コンパイル対象外とする JSP ファイルパスを 1 行に一つ記述します。# で始まる行はコメント行とみなされます。

10. JSP 事前コンパイルとチェック

次に、コンパイル対象外リストファイルの記述例を示します。JSP ファイルパスの指定方法については、「10.1.3(3) JSP ファイルパスの指定方法」を参照してください。

JSP ファイルを単体で指定する場合

```
/jsp/title.jsp  
/jsp/pieces/pageA.jsp
```

ディレクトリに含まれるすべてのファイルを指定する場合

```
/jsp/pieces/*
```

コンパイル対象外とするファイルの拡張子を指定する場合

```
/jsp/pieces/*.jsp
```

コンパイル対象外リストファイル名に使用できる文字は、A ~ Z, a ~ z, 0 ~ 9 の半角英数字, 「_」(半角のアンダースコア), および 「.」(ピリオド) です。

(b) コンパイル対象外リストファイルの指定

コンパイル対象外リストファイルを指定するには、cjjspc コマンドのオプション -excludelist に、コンパイル対象外リストファイルの絶対パスまたはカレントディレクトリからの相対パスを指定します。

コンパイル対象外リストファイルの指定形式を示します。

```
> cjjspc -excludelist <コンパイル対象外リストファイルのパス> -root <Webアプリケーションのルートディレクトリ>
```

(3) コンパイル対象外ファイルを直接指定する方法

cjjspc コマンド実行時に、コンパイル対象外の JSP ファイルを指定するには、cjjspc のオプション -excludefile にコンパイル対象外とする JSP ファイルパスを指定します。対象外の JSP ファイルを複数指定するには、Windows の場合は「;」(セミコロン)、UNIX の場合は「:」(コロン) でファイルパスを区切ります。

注意

- ファイルパスに区切り文字 (Windows の場合は「;」、UNIX の場合は「:」) を含めることはできません。ファイルパスに区切り文字を含める必要がある場合は、コンパイル対象外リストファイルを使用してください。
- ファイルパスにワイルドカード文字「*」(アスタリスク) や空白文字列を含める場合は、ファイルパスを必ず "" (ダブルクォーテーション) で括ってください。

コンパイル対象外ファイルの指定形式を、Windows および UNIX の場合に分けて示します。

Windows の場合の指定形式

```
> cjjspc -excludefile <JSPファイルパス1>;<JSPファイルパス2>;... -root <Webアプリケーションのルートディレクトリ>
```

UNIX の場合の指定形式

```
> cjjspc -excludefile <JSPファイルパス1>:<JSPファイルパス2>:... -root <Webアプリケーションのルートディレクトリ>
```

JSP ファイルパスの指定方法については、「10.1.3(3) JSP ファイルパスの指定方法」を参照してください。

参考

コンパイル不要な JSP ファイル

JSP ファイルには、単独ではコンパイル不要な JSP ファイルがあります。次に、単独ではコンパイル不要な JSP ファイルの例を示します。

- output.jsp

```
<%@ page contentType="text/html; charset=Shift_JIS" %>
<html>
<head><title>tの値まで数値を出力</title></head>
<body>

<% int t = 10; %>

<%@ include file="display.jsp" %>

</body>
</html>
```

- display.jsp

```
<%
  for(int i=0; i<t; i++){
      out.print(i + " ");
  }
%>
```

この例の場合、output.jsp に対してコンパイルを実行すると、display.jsp を静的にインクルードしてコンパイルされます。そのため、display.jsp を単独でコンパイルする必要はありません。display.jsp をコンパイルすると、int 型変数 t が display.jsp 内で宣言されていないためコンパイルエラーとなります。

このように、コンパイル不要なファイルがある場合、-excludelist または -excludefile オプションを使用することでコンパイル対象外に指定します。

10.1.5 実行結果リストファイルの出力

cjjspc コマンドに -resultlist オプションを指定して実行すると、次に示すファイルを実行

10. JSP 事前コンパイルとチェック

結果リストファイルに出力できます。

- コンパイル対象外の JSP ファイルパスの一覧
- コンパイルに成功した JSP ファイル
- コンパイルに失敗した JSP ファイルのファイルパスの一覧

-resultlist オプションには、実行結果リストファイルの絶対パスまたはカレントディレクトリからの相対パスを指定します。実行結果リストファイルを出力する場合の cjjspc コマンドの指定形式を示します。

```
> cjjspc -resultlist <実行結果リストファイルのパス> -root <Webアプリケーションのルートディレクトリ>
```

実行結果リストファイル名に記述できる文字は、A ~ Z, a ~ z, 0 ~ 9 の半角英数字, 「_」(半角のアンダースコア), および 「.」(ピリオド) です。

実行結果リストファイルの出力形式を示します。

```
# Excluded JSP files.
<コンパイル対象外JSPファイルの一覧>
# Compiled JSP files.
<コンパイルに成功したJSPファイル>
# <コンパイルに失敗したJSPファイル>
:
```

実行結果リストファイルには、コンパイル対象外の JSP ファイルパスおよびコンパイルに成功した JSP ファイルパスが出力され、コンパイルに失敗した JSP ファイルパスはコメント行として出力されます。

実行結果リストファイルの出力例を示します。

```
# Excluded JSP files.
/jsp/title.jsp
/jsp/pieces/*.jsp
# Compiled JSP files.
/jsp/confirm.jsp
#/jsp/discorrecrct.jsp
#/jsp/fail.jsp
/jsp/index.jsp
/jsp/order.jsp
#/jsp/uncomfortable.jsp
```

この例では、次に示すファイルパスが出力されています。

コンパイル対象外の JSP ファイルパス

- /jsp/title.jsp
- /jsp/pieces/*.jsp

コンパイルに成功した JSP ファイルパス

- /jsp/confirm.jsp

- /jsp/index.jsp
- /jsp/order.jsp

コンパイルに失敗した JSP ファイルパス

- /jsp/discorrecr.jsp
- /jsp/fail.jsp
- /jsp/uncomfortable.jsp

ポイント

実行結果リストファイルとコンパイル対象外リストファイルの共有

-resultlist オプションと -excludelist オプションに同じファイルを指定することで、一度コンパイルに成功した JSP ファイルをコンパイル対象外ファイルに指定できます。また、コンパイルに失敗した JSP ファイルがコンパイル不要な場合は、コンパイルに失敗した JSP ファイルパスの行頭の「#」を削除することで、コンパイル対象外に指定できます。

10.1.6 Java ソースファイルの保存

cjjspc コマンドに -keepgenerated オプションを指定すると、JSP コンパイル時に生成された Java ソースファイルを残すことができます。このオプションを指定しない場合は、Java ソースファイルは JSP コンパイル実行後に削除されます。

作成された Java ソースファイルは、コンパイルされたクラスファイルと同じディレクトリに作成されます。

Java ソースファイルを保存する場合の cjjspc コマンドの指定形式を示します。

```
> cjjspc -keepgenerated -root <Webアプリケーションのルートディレクトリ>
```

10.1.7 JSP コンパイル時の Java 言語仕様のバージョン指定

cjjspc コマンドに -source オプションを指定すると、JSP コンパイル時に生成された Java ソースファイルを指定された Java 言語仕様のバージョンでコンパイルできます。このオプションを指定しない場合は、JavaSE 5.0 の Java 言語仕様に従ってコンパイルされます。

Java 言語仕様のバージョン指定する場合の cjjspc コマンドの指定形式を示します。

```
> cjjspc -source <バージョン> -root <Webアプリケーションのルートディレクトリ>
```

指定できるバージョンを示します。

1.3

J2SE1.3 の Java 言語仕様に従って、JSP ファイルから生成された Java のソース

ファイルがコンパイルされます。

1.4

J2SE1.4 の Java 言語仕様に従って、JSP ファイルから生成された Java のソースファイルがコンパイルされます。

1.5 または 5

JavaSE 5.0 の Java 言語仕様に従って、JSP ファイルから生成された Java のソースファイルがコンパイルされます。

10.1.8 JSP ワークディレクトリ名の変更

cjjspc コマンドに `-jspworkdir` オプションを使用すると、JSP ワークディレクトリ名を変更できます。

JSP ワークディレクトリを変更する場合の `cjjspc` コマンドの指定形式を示します。

```
> cjjspc -jspworkdir <変更後のJSPワークディレクトリ名> -root <Webアプリケーションのルートディレクトリ>
```

JSP ワークディレクトリ名を「`hitachi_jsp_directory`」に変更する場合の指定例を示します。

```
> cjjspc -jspworkdir hitachi_jsp_directory -root d:¥app¥webappl
```

JSP ワークディレクトリ名に使用できる文字列は、`A ~ Z`、`a ~ z`、`0 ~ 9`の半角英数字、「`_`」(半角のアンダースコア)です。また、次のディレクトリ名は、大文字、小文字の区別なく JSP ワークディレクトリ名として使用できません。

- `lib`
- `tags`
- `classes`

また、JSP ワークディレクトリ名を変更した場合は、J2EE サーバのユーザ定義ファイル (`usrconf.properties`) の `webserver.jsp.precompile.jsp_work_dir` キーに、変更した JSP ワークディレクトリ名を設定する必要があります。

`webserver.jsp.precompile.jsp_work_dir` キーについては、マニュアル「Cosminexus リファレンス 定義編」を参照してください。

ユーザ定義ファイル (`usrconf.properties`) の設定例を次に示します。

```
webserver.jsp.precompile.jsp_work_dir=hitachi_jsp_directory
```


10.1.9 JavaVM 起動オプションの変更

環境変数「COSMINEXUS_CMD_JAVA_ARGS」を設定すると、cjjspc コマンドが動作する JavaVM の起動オプションを変更できます。

デフォルトでは JavaVM の起動オプションに「-Xmx512m」(Java ヒープメモリ領域の最大値が 512MB) が設定されています。cjjspc コマンドで大規模な Web アプリケーションをコンパイルする場合、Java ヒープメモリ領域の最大値を超え、java.lang.OutOfMemoryError が発生するおそれがあります。したがって、大規模な Web アプリケーションをコンパイルする場合は、あらかじめ環境変数 COSMINEXUS_CMD_JAVA_ARGS に、適切な Java ヒープメモリ領域を指定する必要があります。

cjjspc コマンドの Java ヒープメモリ領域の初期値 (512MB) と、最大値 (1,024MB) を設定する場合の環境変数 COSMINEXUS_CMD_JAVA_ARGS の指定例を示します。

Windows の場合の指定例

```
>set COSMINEXUS_CMD_JAVA_ARGS=-Xms512m -Xmx1024m
```

UNIX の場合の指定例

```
$ export COSMINEXUS_CMD_JAVA_ARGS="-Xms512m -Xmx1024m"
```

10.1.10 コマンド実行時のログの出力

コマンド実行時に標準出力または標準エラー出力にログを出力します。ログ出力の結果をファイルに残す場合は、コマンドの出力をファイルにリダイレクトしてください。

ログ出力の結果をファイルに残す場合の指定例を示します。

Windows の場合の指定例

```
> cjjspc -root D:\app\webappl 1> .\stdout.log 2> .\stderr.log
```

UNIX の場合の指定例

```
# cjjspc -root /app/webappl 1> ./stdout.log 2> ./stderr.log
```

10.1.11 JSP ファイルの文字エンコーディングの指定

cjjspc コマンドに -pageencoding オプションを指定することで、JSP 事前コンパイル実行時に JSP 事前コンパイルの実行対象となる JSP ファイルにデフォルトの文字エンコーディングを設定できます。

このオプションを指定しない場合は、Servlet 仕様で規定されている文字エンコーディン

グが適用されます。Servlet 仕様で規定されている文字エンコーディングについては、マニュアル「Cosminexus 機能解説」を参照してください。

JSP ファイルの文字エンコーディングを指定する場合の `cjjspc` コマンドの指定形式を示します。

```
cjjspc -pageencoding <文字エンコーディング> -root <Webアプリケーションのルートディレクトリ>
```

指定できる文字エンコーディングは、JavaVM がサポートしている文字エンコーディングです。JavaVM がサポートしている文字エンコーディングについては、JDK のドキュメントを参照してください。また、指定できる文字列は、`java.nio` API 用の正準名と `java.lang` API 用の正準名に記載されている文字エンコーディング、およびそれらの別名になります。指定できない文字エンコーディングを指定した場合は、メッセージ (KDJE53441-E) が出力されます。

10.1.12 MyEclipse で J2EE アプリケーションを開発する場合

MyEclipse で J2EE アプリケーションを開発する場合は、JSP 事前コンパイル時に出力された Java ソースファイルを MyEclipse のワークスペースに反映する必要があります。また、MyEclipse では、JSP 事前コンパイル時に出力された Java ソースファイルを利用して、JSP ファイルをデバッグできます。

ここでは MyEclipse のワークスペースに Java ソースファイルを反映する手順、JSP ファイルをデバッグする手順を説明します。

(1) MyEclipse のワークスペースに Java ソースファイルを反映する

JSP 事前コンパイル時に出力された Java ソースファイルを MyEclipse のワークスペースに反映する必要があります。次の操作をして、MyEclipse のワークスペースに出力された Java ソースファイルを反映してください。


1. [パッケージ・エクスプローラー] ビューで、JSP 事前コンパイルをした Web プロジェクトを選択します。
2. コンテキストメニューから [更新] を選択します。
MyEclipse のワークスペースに Java ソースファイルが反映されます。

(2) Java ソースファイルを利用して JSP ファイルをデバッグする

MyEclipse では、JSP 事前コンパイル時に出力された Java ソースファイルを利用して JSP ファイルをデバッグできます。次の手順に従って、デバッグしてください。

1. Web プロジェクトに次のディレクトリを作成してください。

<Web ルート・フォルダ>%WEB-INF%cosminexus_jsp_work%org%apache%jsp

2. JSP 事前コンパイル時に出力された Java ソースファイルを「<Web ルート・フォルダ>%WEB-INF%cosminexus_jsp_work%org%apache%jsp」に移動します。
3. [パッケージ・エクスプローラー] ビューから Java ソースファイルを開いて、ブレークポイントを設定します。
4. [サーバー] ビューのツールバーから [ (デバッグ・モードで開始)] を選択して、Cosminexus サーバを開始します。
J2EE アプリケーションを実行してブレークポイントに達すると、[デバッグ] パースペクティブが表示されます。J2EE サーバの開始については「7.3 J2EE サーバの開始および停止」を参照ください。
5. エディタに表示される [ソース・ルックアップ・パスの編集] ボタンをクリックします。
[ソース・ルックアップ・パスの編集] ダイアログが表示されます。
6. [追加] ボタンをクリックします。
[ソースの追加] ダイアログが表示されます。
7. [ソースの追加] ダイアログの一覧から [ワークスペース・フォルダー] を選択して、[OK] ボタンをクリックします。
[フォルダの選択] ダイアログが表示されます。
8. 「<Web ルート・フォルダ>%WEB-INF%cosminexus_jsp_work」を指定して、[OK] ボタンをクリックします。
9. [ソース・ルックアップ・パスの編集] ダイアログに指定したパスが反映されていることを確認して、[OK] ボタンをクリックします。
デバッグが実行されます。[デバッグ] パースペクティブでブレークポイントを設定したソースがエディタに表示されて、処理が中断します。

! 注意事項

ソース・ルックアップ・パスの設定は、J2EE サーバのデバッグ起動中の間だけ有効です。J2EE サーバを停止すると、ソース・ルックアップ・パスの設定は失われます。

10.2 JSP ソースのチェック

Cosminexus が提供する `cjjsp2java` コマンドを使用することで、JSP1.1 仕様および JSP1.2 仕様に準拠した JSP ファイル から Java ソースファイルを生成し、JSP ファイルに記述ミスがないかを確認できます。

注

JSP2.0 仕様に準拠した JSP ファイルのチェックをする場合は、`cjjspc` コマンドを使用してください。

`cjjsp2java` コマンドのインストール先を示します。

```
<Cosminexusのインストールディレクトリ>%CC%\web\bin\cjjsp2java
```

`cjjsp2java` コマンドの形式およびオプションについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

ここでは、次の内容について説明します。

- `cjjsp2java` コマンド実行時に出力されるメッセージ
- `cjjsp2java` コマンドの使用例
- `cjjsp2java` コマンドで生成したソースファイルのコンパイル

10.2.1 `cjjsp2java` コマンド実行時に出力されるメッセージ

`cjjsp2java` コマンド実行時のメッセージは、コマンドを実行したコンソールの標準出力、および標準エラー出力に出力されます。出力されるメッセージは、`cjjsp2java` コマンドの実行結果、および JSP ファイル解析時のエラー通知です。

`cjjsp2java` コマンドの実行結果

`cjjsp2java` コマンドを実行し、正常に Java ソースファイルが生成された場合、次のフォーマットでメッセージが出力されます。

```
KDJE53400-I A JSP file was successfully translated. (date = YYYY/MM/DD
hh:mm:ss, JSP file = <JSP ファイル>, java file = <生成した Java ソースファイル>)
```

YYYY/MM/DD hh:mm:ss

Java ソースファイルの生成時刻です。`-d:non` オプションを指定して Java ソースファイルを生成しない場合は、JSP ファイルの解析完了時刻になります。

<JSP ファイル>

指定した JSP ファイルの絶対パスです。

<生成した Java ソースファイル>

生成した Java ソースファイルの絶対パスです。`-d:non` オプションを指定して

Java ソースファイルを生成しない場合は、空文字 ("") になります。

複数の JSP ファイルを指定した場合、cjsp2java コマンドの実行結果は、JSP ファイル単位に出力されます。

JSP ファイル解析時のエラー通知

JSP ファイルの解析時に JSP ファイルのエラー（シンタックスエラーなど）を検出した場合に出力されます。cjsp2java コマンドで複数の JSP ファイルを指定した場合、エラーを検出した時点で、Java ソースファイルの生成が中断されます。

10.2.2 cjsp2java コマンドの使用例

cjsp2java コマンドの使用例を示します。基本的な使用例だけでなく、タグライブラリを使用する場合、include ディレクティブを使用する場合、および複数の JSP ファイルを指定する場合の使用例についても示します。

(1) 基本的な使用例

タグライブラリと include ディレクティブを使用していない、一つの JSP ファイルから Java ソースファイルを生成する場合の使用例を示します。

実行環境の構成

```
D:¥dev
    webapp1
        jsp
            index.jsp
```

コマンド実行例

index.jsp に対して cjsp2java コマンドを実行し、index.jsp と同一ディレクトリ下に Java ソースファイルを生成します。

```
> D:
> cd dev¥webapp1¥jsp
> <Cosminexusインストールディレクトリ>¥CC¥web¥bin¥cjsp2java index.jsp
KDJE53400-I A JSP file was successfully translated. (date = YYYY/MM/DD
hh:mm:ss, JSP file = D:¥dev¥webapp1¥jsp¥index.jsp, java file =
D:¥dev¥webapp1¥jsp¥index$.java)
```

(2) タグライブラリを使用する場合の使用例

タグライブラリを使用する場合の使用例を示します。

実行環境の構成

```

D:¥dev
    webapp1
      |      jsp
      |      index.jsp
      |      home.jsp
      tmp

```

- D:¥dev¥webapp1 は、Web アプリケーションのルートディレクトリとして、-root オプションに指定します。
- D:¥dev¥webapp1¥jsp¥index.jsp ファイルの内容は次のとおりです。index.jsp では 3 行目で include ディレクティブを使用しています。

```

<HTML>
<BODY>
<jsp:include page="/jsp/home.jsp" flush="true"/>
</BODY>
</HTML>

```

コマンド実行例

index.jsp に対し、cjjsp2java コマンドを実行し、D:¥dev¥tmp ディレクトリ下に Java ソースファイルを生成します。

```

> D:
> cd dev
> <Cosminexusインストールディレクトリ>¥CC¥web¥bin¥cjjsp2java -d tmp -root
webapp1 webapp1¥jsp¥index.jsp
KDJE53400-I A JSP file was successfully translated. (date = YYYY/MM/DD
hh:mm:ss, JSP file = D:¥dev¥webapp1¥jsp¥index.jsp, java file =
D:¥dev¥tmp¥jsp¥index$jsp.java)

```

(4) 複数の JSP ファイル指定する場合の使用例

複数の JSP ファイルから Java ソースファイルを生成する場合の使用例を示します。

実行環境の構成

```

D:¥dev
    webapp1
      jsp
        index.jsp
        index01.jsp
        index02.jsp

```

コマンド実行例

index.jsp、index01.jsp、index02.jsp に対して cjjsp2java コマンドを実行し、index.jsp と同一ディレクトリ下に Java ソースファイルを生成します。

```

> D:
> cd dev¥webapp1¥jsp
> <Cosminexusインストールディレクトリ>¥CC¥web¥bin¥cjjsp2java index.jsp
index01.jsp index02.jsp
KDJE53400-I A JSP file was successfully translated. (date = YYYY/MM/DD
hh:mm:ss, JSP file = D:¥dev¥webapp1¥jsp¥index.jsp, java file =
D:¥dev¥webapp1¥jsp¥index$jsp.java)
KDJE53400-I A JSP file was successfully translated. (date = YYYY/MM/DD
hh:mm:ss, JSP file = D:¥dev¥webapp1¥jsp¥index01.jsp, java file =
D:¥dev¥webapp1¥jsp¥index01$jsp.java)
KDJE53400-I A JSP file was successfully translated. (date = YYYY/MM/DD
hh:mm:ss, JSP file = D:¥dev¥webapp1¥jsp¥index02.jsp, java file =
D:¥dev¥webapp1¥jsp¥index02$jsp.java)

```

10.2.3 cjjsp2java コマンドで生成したソースファイルのコンパイル

生成された Java ソースコードに対して javac コマンドを実行することで、JSP ファイルに記述した `<%! ... %>` タグ、`<% ... %>` タグ、または `<%= ... %>` タグ部分が Java のコードとして正しくコンパイルされるかを確認できます。

cjjsp2java コマンドで生成したソースファイルをコンパイルするには、次に示す JAR ファイルをクラスパスに指定する必要があります。

- `<Cosminexus のインストールディレクトリ>¥CC¥lib¥ejbserver.jar`
- `<Cosminexus のインストールディレクトリ>¥CC¥lib¥hit2jee.jar`

! 注意事項

生成された Java ソースコード、およびそのあとに javac コマンドで作成した class ファイルは、実行環境では使用できません。

付録

付録 A	MyEclipse の動作確認用サンプルプログラム
付録 B	インスタントセットアップ機能を使用しない環境の構築手順
付録 C	Developer Standard 使用時の注意事項
付録 D	Matisse4MyEclipse フィーチャのインストール
付録 E	テスト用実行環境のシステムのチューニング
付録 F	Cosminexus Version 6 から Cosminexus Version 7 への移行
付録 G	JavaMail の使用例
付録 H	web.xml 編集時の注意事項（互換用の動作モードの場合）
付録 I	インスタントセットアップ機能実行時のエラーメッセージ一覧
付録 J	データベース構築・削除時のエラーメッセージ一覧
付録 K	用語解説

付録 A MyEclipse の動作確認用サンプルプログラム

Developer では、MyEclipse の動作確認用のサンプルプログラムとして、組み込みデータベースを使用するサンプルプログラムを提供しています。サンプルプログラムは、MyEclipse が提供する開発支援機能、組み込みデータベースの構築および接続のチュートリアルとして使用できます。また、サンプルプログラムを使用することで、MyEclipse によるデータベースを使用した J2EE アプリケーション開発手順の理解を深められます。

ここでは、サンプルプログラムの概要、組み込みデータベースの構築と開始、およびサンプルプログラムの実行手順を説明します。

参考

なお、サンプルプログラムは、Application Server でも提供しています。Application Server で提供するサンプルプログラムについては、マニュアル「Cosminexus システム構築ガイド」の Cosminexus で提供するサンプルプログラムを参照してください。

付録 A.1 サンプルプログラムの概要

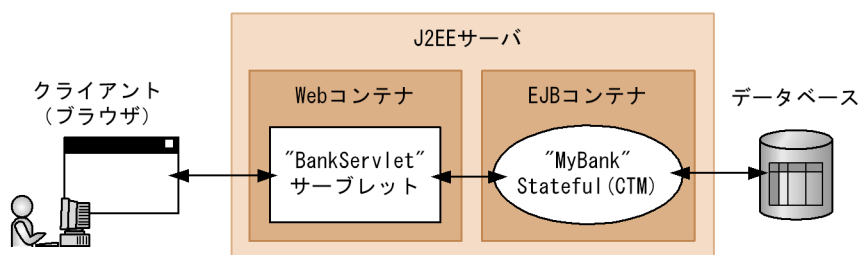
ここでは、Developer が提供するサンプルプログラムの概要を説明します。

(1) サンプルプログラムの内容

Developer が提供しているサンプルプログラム（Bank）は、Web ブラウザから入力されたユーザ ID を持つユーザの資金を当座預金口座から普通預金口座へ移動させるプログラムです。資金はデータベースで管理され、資金移動時に当座預金口座テーブルと普通預金口座テーブルが更新されます。

このサンプルプログラムの構成を次に示します。

図 A-1 サンプルプログラム（Bank）の概要



このサンプルプログラムでは、アノテーションを使用してデータベースに接続する方法を示します。

アノテーション対応 EJB で @Resource アノテーションを使用して、リソースを取得し

ます。

```
@Resource(mappedName="DB_Connector_for_Cosminexus_Driver")
private DataSource ds;
```

サーブレット側からビジネスインタフェースを使用した Stateful Session Bean を lookup するには、JNDI を使用します。

```
Context ic = new InitialContext();
Object objref = ic.lookup("java:comp/env/ejb/Bank");
BankHome home = (BankHome) PortableRemoteObject.narrow(objref,
BankHome.class);
```

指定したユーザ ID、金額に対して処理を実行します。「Bank」で指定できるユーザ ID は「001」、初期残高は「10000 円」です。

(2) サンプルプログラムの前提環境

付録 A では、次の環境を前提として、サンプルプログラムの使用手順を説明します。

インスタントセットアップ機能で構築した環境

付録 A では、インスタントセットアップ機能で構築した開発環境を前提としています。このため、次の条件に当てはまる論理サーバを使用します。

- Cosminexus J2EE サーバがローカルにインストールされている。
- 環境構築が正しく実施されている。
- J2EE サーバを開始する前に、J2EE サーバに関係する論理サーバをあらかじめ開始している。

また、次に示す MyEclipse を使用してください。なお、Eclipse のランゲージパックが適用されていることを前提とします。

- MyEclipse for Cosminexus

JRE の設定

次に示す JDK を使用してください。

- Cosminexus の JDK (<Cosminexus のインストールディレクトリ>%jdk)
JDK は、Eclipse の [設定] ダイアログの [インストール済みの JRE] ページで設定できます。JDK の確認方法については、「2.4.1(1) JDK の確認」を参照してください。

Management Server リモート管理機能の接続先

次に示すホストを指定してください。

- localhost

Management Server リモート管理機能の接続先は、Server Plug-in および MyEclipse (Cosminexus コネクタ) で設定します。Server Plug-in での設定方法は「2.4.4(1) 接続先ホストの設定」を、MyEclipse (Cosminexus コネクタ) での設定方法は「付録 B.2(4)(a) 接続先ホストの設定変更」を参照してください。

使用するパースペクティブ

次に示すパースペクティブを使用します。ただし、パースペクティブの設定は、デフォルトから変更していないことを前提としています。

- Server Plug-in を使用する場合 : [Cosminexus Server Plug-in] パースペクティブ
- デバッグを行う場合 : [デバッグ] パースペクティブ
- それ以外の場合 : [MyEclipse Java エンタープライズ開発] パースペクティブ

プロキシの設定

インターネットの接続にプロキシを使用している場合は、MyEclipse のプロキシを設定してください。MyEclipse のプロキシは、[設定] ダイアログの [プロキシ] ページで設定できます。プロキシの設定方法については、「2.5.1 MyEclipse のプロキシの設定」を参照してください。

リソースアダプタ

サンプルプログラム (Bank) では、インスタントセットアップ機能がデPLOYしたリソースアダプタを使用します。なお、このリソースアダプタは、J2EE リソースアダプタとしてデPLOYされます。

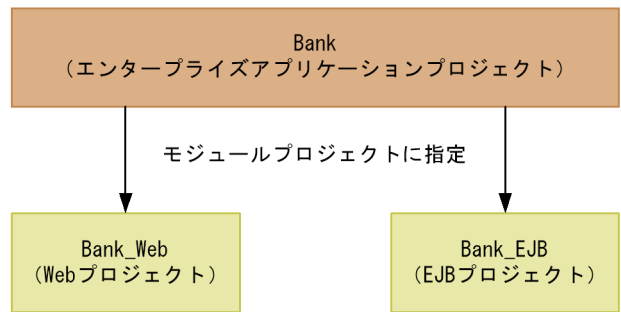
(3) サンプルプログラムのプロジェクトの構成

サンプルプログラムでは、次のプロジェクトを提供しています。

表 A-1 サンプルプログラムが提供するプロジェクト

プロジェクト名	概要
Bank	エンタープライズアプリケーションプロジェクトです。
Bank_EJB	EJB プロジェクトです。Bank のモジュールプロジェクトに指定します。
Bank_Web	Web プロジェクトです。Bank のモジュールプロジェクトに指定します。

サンプルプログラムが提供するプロジェクトの構成は、次のとおりです。



(4) サンプルプログラムの提供形態と格納先

Developer は、サンプルプログラムを zip フォーマットのアーカイブファイル「Bank.zip」として提供しています。Bank.zip は、次のディレクトリに格納されます。

<Cosminexusのインストールディレクトリ>\ADP\MyEclipse\Samples

(5) サンプルプログラムのディレクトリ構成

Bank.zip を任意の場所に解凍すると、次のディレクトリ構成になります。

表 A-2 サンプルプログラムのディレクトリ構成

ディレクトリー / ファイル	説明
Bank¥	エンタープライズアプリケーションプロジェクトのルート
META-INF¥	管理情報を格納するディレクトリ
application.xml	DD ファイル (J2EE アプリケーション (EAR) の配備記述子)
MANIFEST.MF	マニフェストファイル
.mymetadata	MyEclipse プロジェクトファイル (MyEclipse のプロジェクトの情報を保存)
.project	プロジェクト記述ファイル (Eclipse のプロジェクトの情報を保存)
Bank_EJB¥	EJB プロジェクトのルート
src¥	ソースファイル格納用のディレクトリ
connectdb¥	パッケージ・フォルダ
Bank.java	Java ソースファイル (EJB プロジェクト用のソースコード)
BankEJB.java	Java ソースファイル (EJB プロジェクト用のソースコード)
BankHome.java	Java ソースファイル (EJB プロジェクト用のソースコード)
InsufficientBalanceException.java	Java ソースファイル (EJB プロジェクト用のソースコード)
META-INF	管理情報を格納するディレクトリ
ejb-jar.xml	DD ファイル (EJB の配備記述子)
MANIFEST.MF	マニフェストファイル
.classpath	クラスパスファイル (プロジェクトのクラスパスを保存)
.mymetadata	MyEclipse プロジェクトファイル (MyEclipse のプロジェクトの情報を保存)
.project	プロジェクト記述ファイル (Eclipse のプロジェクトの情報を保存)
Bank_Web¥	Web プロジェクトのルート

ディレクトリー / ファイル	説明
src¥	ソースファイル格納用のディレクトリ
connectdb¥	パッケージ・フォルダ
BankBean.java	Java ソースファイル
BankServlet.java	Java ソースファイル
WebRoot¥	Web ルートフォルダ
WEB-INF¥	Web クライアントから直接アクセスできない ファイルを格納するディレクトリ
web.xml	DD ファイル (Web アプリケーションの配備記述子)
META-INF	管理情報を格納するディレクトリ
MANIFEST.MF	マニフェストファイル
bank.jsp	JSP ファイル (ID と数値を入力する画面)
forward.jsp	JSP ファイル (ID と数値を Servlet に送信)
.classpath	クラスパスファイル (プロジェクトのクラスパスを保存)
.mymetadadata	MyEclipse プロジェクトファイル (MyEclipse のプロジェクトの情報を保存)
.project	プロジェクト記述ファイル (Eclipse のプロジェクトの情報を保存)

付録 A.2 組み込みデータベースの開始

次のショートカットまたはコマンドを実行して、組み込みデータベースを開始してください。なお、どちらとも実行するには、Administrator 権限が必要です。

スタートメニューのショートカットを使用する場合

[Cosminexus] - [Application Development Plug-in] - [組み込みデータベースの開始] を選択してください。

組み込みデータベース用コマンドを使用する場合

次のコマンドを実行してください。

```
"<Cosminexusのインストールディレクトリ>¥DB¥bats¥cdstart.bat"
```

付録 A.3 サンプルプログラムの実行手順

Developer が提供するサンプルプログラムは、MyEclipse の J2EE アプリケーション開発のチュートリアルとして使用できます。チュートリアルが対象とする機能は、次のとおりです。

1. MyEclipse の起動

2. J2EE サーバの設定
3. エンタープライズアプリケーションプロジェクトのデプロイ
4. J2EE アプリケーションの実行

(1) MyEclipse の起動

次のファイルを実行して、MyEclipse を起動します。

```
<Eclipseのインストールディレクトリ>%eclipse%eclipse.exe
```

注

インスタントセットアップ機能の [Eclipse と MyEclipse のセットアップ] ページの [Eclipse のインストールディレクトリ] で指定した値です。

(2) J2EE サーバの設定

MyEclipse から J2EE サーバを使用するための設定をします。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. [設定] ダイアログの左ペインで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。
3. 右ペインの [Cosminexus 7] ページで [ログイン] ボタンをクリックします。
[ログイン・リモート管理] ダイアログが表示されます。
4. 次の値を入力して、[OK] ボタンをクリックします。

項目名	設定値
管理ユーザー ID	admin
パスワード	admin

5. [Cosminexus 7] ページの [Cosminexus サーバー] ツリービューで、[localhost] - [DefaultDomain] - [InstantHost] - [InstantJ2EEServer] をチェックして [OK] ボタンをクリックします。
MyEclipse の [サーバー] ビューから J2EE サーバを使用できるようになります。
[サーバー] ビューは、[MyEclipse Java エンタープライズ開発] パースペクティブで表示されます。

(3) エンタープライズアプリケーションプロジェクトのデプロイ

サンプルプログラム (Bank) を Eclipse のワークスペースにインポートします。

1. Eclipse のメニューから [ファイル] - [インポート] を選択します。
[インポート] ダイアログが表示されます。




2. [インポート] ダイアログの [インポート・リソースの選択] で, [既存プロジェクトをワークスペースへ] を選択し, [次へ] ボタンをクリックします。
[プロジェクトのインポート] ページが表示されます。
3. [アーカイブ・ファイルの選択] をチェックし, [参照] ボタンをクリックします。
[インポートするプロジェクトを含むアーカイブの選択] ダイアログが表示されます。
4. [インポートするプロジェクトを含むアーカイブの選択] ダイアログで, 次のファイルを選択して, [開く] ボタンをクリックします。

<Cosminexusのインストールディレクトリ>¥ADP¥MyEclipse¥Samples¥Bank.zip

5. [プロジェクトのインポート] ページの [プロジェクト] エリア内にある項目をすべてチェックして, [終了] ボタンをクリックします。
サンプルプログラム (Bank) が Eclipse のワークスペースにインポートされます。

(4) サンプルプロジェクト (Bank) のデプロイ

J2EE サーバにサンプルプログラム (Bank) をデプロイします。

1. [サーバー] ビューで [Cosminexus 7] を選択し, [ (実行モードで開始)] または [ (デバッグ・モードで開始)] をクリックします。
J2EE サーバが開始します。J2EE サーバが開始すると, [Cosminexus 7] の [状態] が [実行中] になります。
2. [ (デプロイメント管理)] をクリックします。
[サーバー・デプロイメント] ダイアログが表示されます。
3. [サーバー・デプロイメント] ダイアログの [追加] ボタンをクリックします。
[新規デプロイメント] ダイアログが表示されます。
4. [新規デプロイメント] ダイアログの [プロジェクト] で [Bank] を選択し, [終了] ボタンをクリックします。
[サーバー・デプロイメント] ダイアログの [デプロイメント] に [Bank] が追加されます。
5. [サーバー・デプロイメント] ダイアログの [OK] ボタンをクリックします。
J2EE サーバにサンプルプログラム (Bank) がデプロイされます。

(5) サンプルプロジェクト (Bank) の実行

Web ブラウザから次のアドレスに接続すると, サンプルプログラム (Bank) が実行されます。

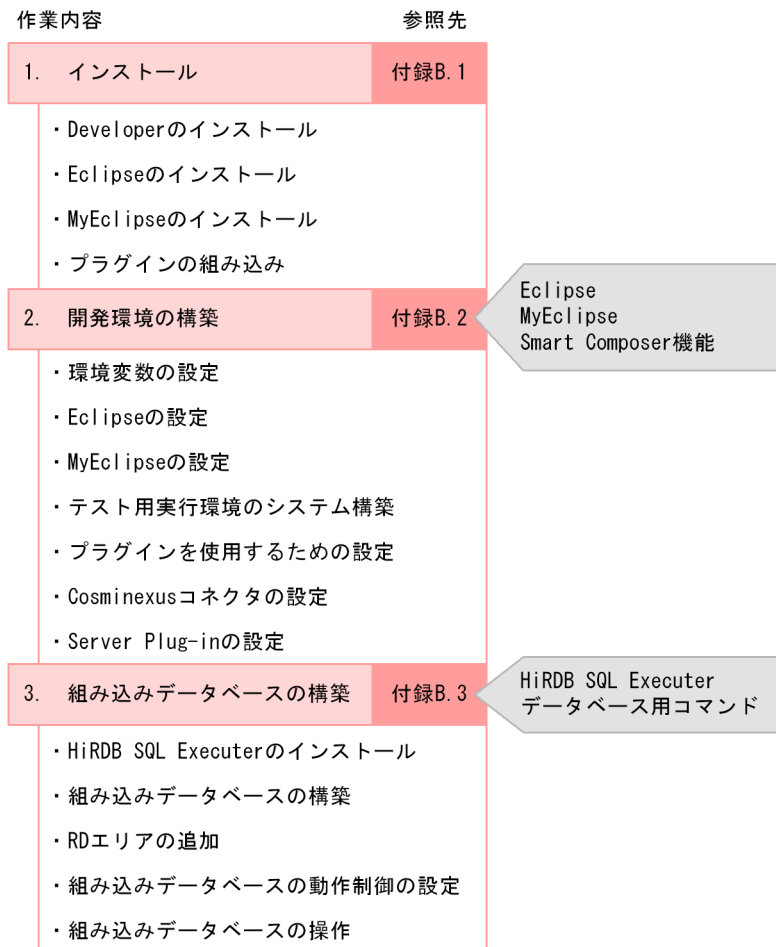
サンプルプログラム (Bank) を実行するための接続先アドレス

<http://localhost/BankWeb/bank.jsp>

付録 B インスタントセットアップ機能を使用しない環境の構築手順

ここでは、Developer が提供するインスタントセットアップ機能を使用しないで開発環境およびテスト用実行環境を構築する手順を説明します。構築の流れを次の図に示します。

図 B-1 開発環境およびテスト用実行環境の構築手順（インスタントセットアップ機能を使用しない場合）



（凡例）  : 使用するツール

環境構築後の作業の流れについては、「1.5.2 J2EE アプリケーションの開発手順」を参照してください。なお、J2EE アプリケーションにリソースアダプタを含めている場合は、リソースアダプタのプロパティを設定します。リソースアダプタのプロパティの設定

定については、「付録 B.4 リソースアダプタのプロパティ設定」を参照してください。

付録 B.1 インストール

J2EE アプリケーション開発環境で使用するプログラムをインストールします。プログラムのインストール順序を次に示します。

1. Developer のインストール

HITACHI 総合インストーラを使用して、Developer をインストールします。インストール手順については、「2.2 Developer のインストール」を参照してください。

2. Eclipse のインストール

Eclipse のプログラムをダウンロードして、インストールします。Eclipse のインストール手順については、「(1) Eclipse のインストール」を参照してください。

3. MyEclipse のインストール

Eclipse に MyEclipse を組み込みます。MyEclipse のインストール手順については、「(2) MyEclipse のインストール」を参照してください。

4. プラグインの組み込み

Developer が提供しているプラグインを Eclipse に組み込みます。プラグインの組み込み手順については、「(4) プラグインの組み込み」を参照してください。

ここでは、Eclipse と MyEclipse のインストール手順、およびプラグインの組み込み方法について説明します。また、プログラムのアンインストールについても説明します。アンインストールの順序については、「付録 B.5 アンインストール」を参照してください。

(1) Eclipse のインストール

ここでは、Eclipse のインストールについて説明します。なお、Cosminexus をアップグレードインストールした場合で、以前のバージョンで使用していた Eclipse を引き続き使用するときの手順についても説明します。

(a) Eclipse インストールの手順

J2EE アプリケーションの開発、実行時に使用する Eclipse をインストールします。インストール手順を次に示します。なお、Eclipse のアーカイブファイルの URL については、リリースノートを参照してください。

1. Eclipse をインストールするためのディレクトリを作成します。

Eclipse をインストールするディレクトリ、および Eclipse のインストールのために一時的に使用するディレクトリ（作業用ディレクトリ）を作成します。

2. Eclipse のアーカイブファイルをダウンロードします。

Eclipse のダウンロードサイトから Eclipse のアーカイブファイルをダウンロードします。

3. Eclipse のアーカイブファイルを解凍します。
ダウンロードした Eclipse のアーカイブファイルを 1. で作成した作業用ディレクトリで解凍します。
4. 解凍した Eclipse のファイルをコピーします。
3. で解凍した eclipse フォルダを 1. で作成した Eclipse のインストールディレクトリにコピーします。
5. 作業用ディレクトリを削除します。
次のコマンドを実行して、作業用ディレクトリを削除します。

```
rd /s <作業用ディレクトリ>
```

(b) 旧バージョンで使用していた Eclipse を引き続き使用する場合 (Eclipse 3.2.2 以上)

07-60 で提供している MyEclipse を使用する場合、Eclipse 3.2.2 以上のバージョンの Eclipse を使用します。

07-60 をインストールしたあとに、旧バージョンの Eclipse で使用していた Eclipse 3.2.2 を引き続き使用する場合、拡張ロケーションを削除してください。

拡張ロケーションの削除方法については、「付録 B.5 アンインストール」を参照してください。

(c) 旧バージョンで使用していた Eclipse を引き続き使用する場合 (Eclipse 3.2.1 以前)

旧バージョンで使用していた Eclipse 3.2.1 以前バージョンを引き続き使用したい場合は、Eclipse から Developer をアンインストールする必要があります。使用していた Cosminexus のバージョンに合わせて、次の操作を実施してください。

Cosminexus Version 6 で Eclipse を使用していた場合

Eclipse に組み込んでいる Cosminexus のプラグインを削除してください。

次のディレクトリを削除することで、Cosminexus のプラグインを削除できます。

- <Eclipse のインストールディレクトリ

```
>%plugins%\jp.co.Hitachi.soft.cede.ec.server_x.x.0
```

- <Eclipse のインストールディレクトリ

```
>%plugins%\jp.co.Hitachi.soft.cede.ec.studio_x.x.0
```

注

「x.x.0」は 6.0.0 または 6.5.0 です。

Cosminexus Version 7 で Eclipse を使用していた場合

拡張ロケーションを削除してください。

拡張ロケーションの削除方法については、「付録 B.5(1) プラグインのアンインストール」を参照してください。また、MyEclipse を使用していた場合は、「付録 B.5(2) MyEclipse のアンインストール」を参照してください。

(2) MyEclipse のインストール

MyEclipse をインストールします。MyEclipse は Developer とは別の媒体で提供されています。なお、MyEclipse をインストールする前に、Developer および Eclipse のインストールが完了していることを確認してください。また、Eclipse を実行している場合は、Eclipse を終了させてから MyEclipse をインストールしてください。

次に、MyEclipse のインストール準備と、Eclipse への組み込み方法を説明します。

(a) MyEclipse のインストール準備

MyEclipse のインストールディレクトリを作成して、Cosminexus で提供する MyEclipse のファイルを解凍します。手順を説明します。

1. ディレクトリを準備します。

MyEclipse をインストールする前に、次のディレクトリを用意してください。

- MyEclipse のインストールディレクトリ

MyEclipse をインストールするディレクトリです。

- 作業用ディレクトリ

MyEclipse のインストールのために一時的に使用するディレクトリです。

2. MyEclipse for Cosminexus.zip を解凍します。

MyEclipse for Cosminexus.zip は MyEclipse の CD-ROM に格納されています。

1. で作成した作業用ディレクトリで次のコマンドを実行し、MyEclipse for Cosminexus.zip を解凍します。

```
"<Cosminexusのインストールディレクトリ>%jdk%bin%jar.exe" xvf "<CD-ROMドライブ>:%MyEclipse for Cosminexus.zip"
```

コマンドを実行すると、作業用ディレクトリ下に eclipse フォルダが作成されます。

3. eclipse フォルダをコピーします。

2. で作成された eclipse フォルダを 1. で作成した MyEclipse のインストールディレクトリにコピーします。コピーには、エクスプローラなどを使用します。

4. 作業用ディレクトリを削除します。

次のコマンドを実行して、作業用ディレクトリを削除します。

```
rd /s <作業用ディレクトリ>
```

(3) MyEclipse の組み込み

MyEclipse を Eclipse に組み込み、eclipse.ini を編集します。それぞれの手順を説明します。

(a) Eclipse への組み込み

Eclipse に MyEclipse を組み込みます。組み込みの手順を次に示します。

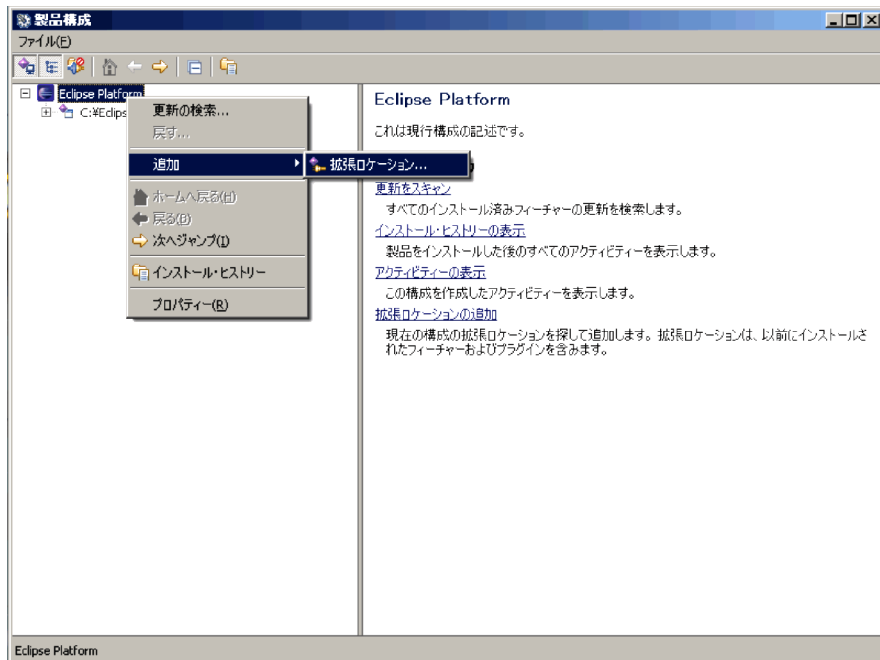
1. Eclipse を起動します。

次のコマンドを実行し、Eclipse を起動します。ここでは、必ず -vm オプションを指定して実行してください。

```
<Eclipseのインストールディレクトリ>%eclipse.exe -vm "<Cosminexusのインストールディレクトリ>%jdk%bin%javaw.exe"
```

2. メニューから [ヘルプ] - [ソフトウェア更新] - [構成の管理] を選択します。
[製品構成] ダイアログが表示されます。

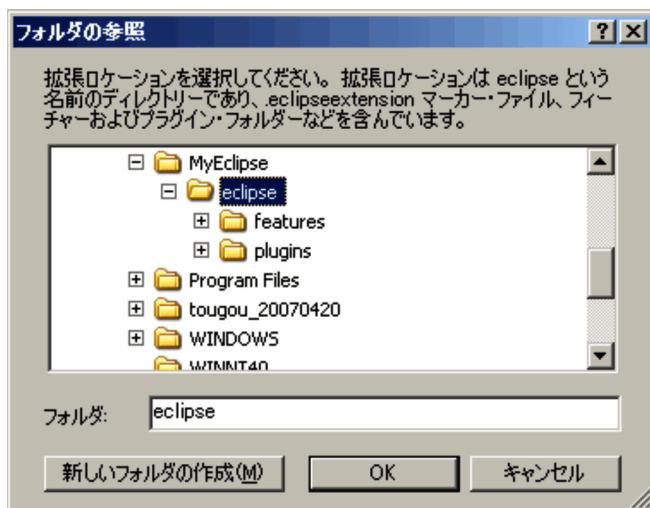
3. 左ペインのツリービューでルートノードを選択し、コンテキストメニューから [追加] - [拡張ロケーション] を選択します。



[フォルダの参照] ダイアログが表示されます。

4. 次のフォルダを指定します。

```
<MyEclipse のインストールディレクトリ >%eclipse
```



5. [OK] ボタンをクリックします。
[インストール / 更新] ダイアログが表示されます。
6. [いいえ] ボタンをクリックします。
ここではワークベンチを再起動しないため、[いいえ] ボタンをクリックしてください。[インストール / 更新] ダイアログに [変更を適用] ボタンがある場合も、[いいえ] ボタンをクリックしてください。

(b) eclipse.ini の編集

eclipse.ini を編集します。Eclipse を起動している場合は、Eclipse を終了してください。
eclipse.ini の編集手順を次に示します。

1. eclipse.ini を退避します。
Eclipse のインストールディレクトリ直下にある eclipse.ini を任意の場所に退避します。退避した eclipse.ini は、Cosminexus のアンインストール後、Eclipse を MyEclipse インストール前の状態に戻すために使用します。
2. Cosminexus で提供している eclipse.ini を Eclipse のインストールディレクトリ直下にコピーします。
Cosminexus で提供している eclipse.ini は次の場所に格納されています。
<Cosminexusのインストールディレクトリ>¥ADP¥MyEclipse
3. eclipse.ini を編集します。
Cosminexus で提供している eclipse.ini の内容を次に示します。下線部分を MyEclipse のインストールディレクトリに修正してください。

```
-vm
<Cosminexusのインストールディレクトリ>%jdk%bin%javaw.exe
-product
com.genuitec.myeclipse.product.ide
-vmargs
-Xms128M
-Xmx512M
-XX:PermSize=64M
-XX:MaxPermSize=128M
-Dosgi.splashLocation=C:%Program
Files%MyEclipse%eclipse%MyEclipseSplash.bmp
-Djava.endorsed.dirs=<Cosminexusのインストールディレクトリ>%jaxp%lib
```

注 改行コードは削除しないでください。

(c) MyEclipse のインストール確認

MyEclipse が正しくインストールされているか確認します。確認手順を次に示します。

1. Eclipse を起動します。

-clean を指定して Eclipse の起動コマンドを実行してください。

```
<Eclipseのインストールディレクトリ>%eclipse.exe -clean
```

注 コマンドの引数に -clean を指定しないと、Eclipse が強制的に再起動されることがあります。

2. メニューから [ヘルプ] - [MyEclipse Enterprise Workbench について] を選択します。

[MyEclipse Enterprise Workbench について] ダイアログが表示されることを確認してください。



(4) プラグインの組み込み

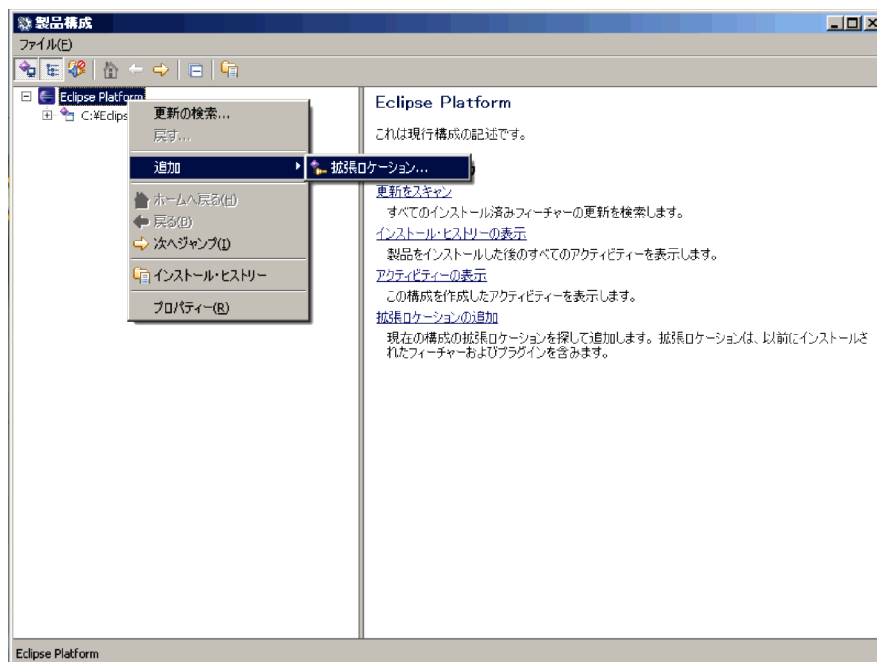
Developer が提供しているプラグイン（Cosminexus コネクタおよび Server Plug-in）を Eclipse に組み込みます。なお、プラグインを組み込む前に、Developer、Eclipse、および MyEclipse のインストールが完了していることを確認してください。

プラグインの組み込みについて説明します。

(a) Eclipse へのプラグインの組み込み

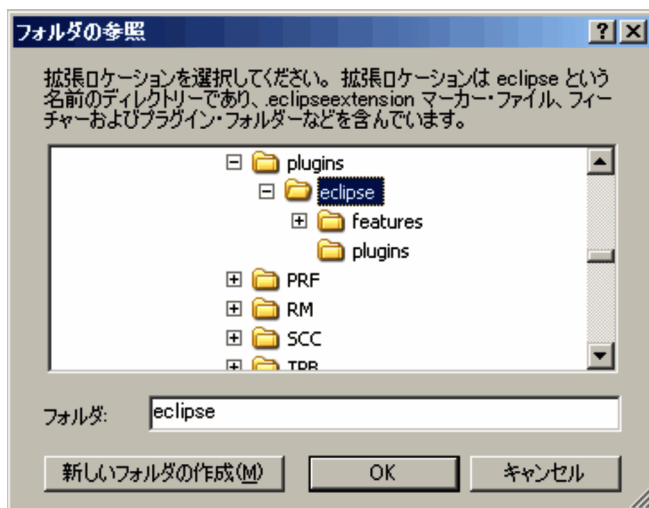
Eclipse へのプラグインの組み込み手順を次に示します。

1. Eclipse を起動し、メニューから [ヘルプ] - [ソフトウェア更新] - [構成の管理] を選択します。
[製品構成] ダイアログが表示されます。
2. 左ペインのツリービューでルートノードを選択し、コンテキストメニューから [追加] - [拡張ロケーション] を選択します。



[フォルダの参照] ダイアログが表示されます。

3. 次のフォルダを指定します。
<Cosminexus のインストールディレクトリ >¥plugins¥eclipse



4. [OK] ボタンをクリックします。
[インストール / 更新] ダイアログが表示されます。
5. [はい] ボタンをクリックします。
[インストール / 更新] ダイアログに [変更を適用] ボタンがある場合も, [はい] ボタンをクリックしてください。Eclipse が再起動されます。

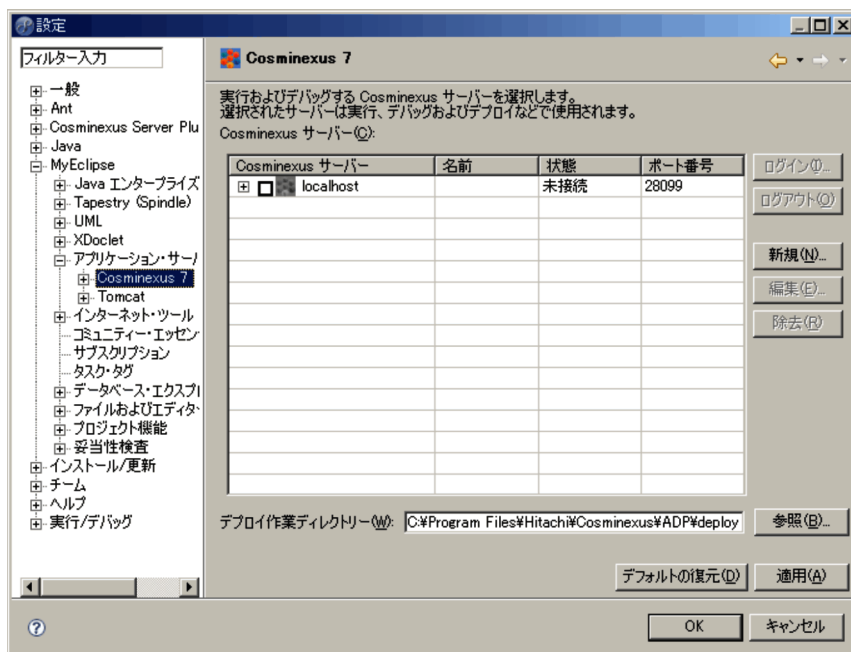
(b) プラグインの組み込み確認

プラグインの組み込み後, Eclipse の再起動が完了したら, Eclipse にプラグインが組み込まれているかを確認します。Cosminexus コネクタの組み込み確認と, Server Plug-in の組み込み確認の手順をそれぞれ示します。

Cosminexus コネクタの組み込み確認

Cosminexus コネクタの組み込み確認の手順を説明します。

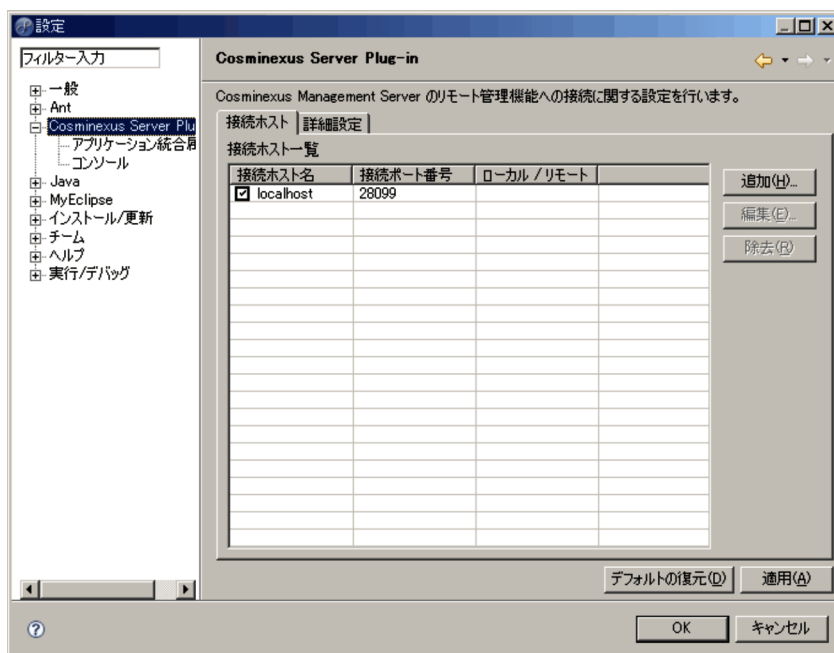
1. メニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューから [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。
右ペインに [Cosminexus 7] ページが表示されることを確認してください。



Server Plug-in の組み込み確認

Server Plug-in の組み込み確認の手順を説明します。

1. メニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューから [Cosminexus Server Plug-in] を選択します。
右ペインに [Cosminexus Server Plug-in] ページが表示されることを確認してください。



付録 B.2 開発環境の構築

インストール完了後，J2EE アプリケーションの開発環境を構築します。J2EE アプリケーションの開発環境の構築の流れを次に示します。

1. 環境変数の設定

Developer および Eclipse で必要となる環境変数を設定します。環境変数の設定方法については、「2.2.2 環境変数の設定」を参照してください。

2. MyEclipse を利用するための設定

J2EE アプリケーション開発で MyEclipse を利用するための設定をします。設定方法については、「(1) MyEclipse を利用するための設定」を参照してください。

3. テスト用実行環境のシステム構築

J2EE アプリケーションのテストで使用するシステムを構築します。システムの構築には，Smart Composer 機能を使用します。Smart Composer 機能とは，システムの構築を支援するための機能です。システムの情報を定義したファイルを作成して Smart Composer 機能のコマンドを実行すると，簡単にシステムを構築できます。構築手順については、「(2) テスト用実行環境のシステム構築」を参照してください。

4. プラグインを使用するための設定

Cosminexus コネクタおよび Server Plug-in は，Management Server リモート管理機能に接続します。Cosminexus コネクタおよび Server Plug-in を使用するために，Management Server リモート管理機能への接続の設定をします。設定方法について

は、「(3) プラグインを使用するための設定」を参照してください。

5. Cosminexus コネクタの設定

Cosminexus コネクタを設定します。Cosminexus コネクタは、MyEclipse から J2EE サーバや J2EE アプリケーションを操作するときに使用します。設定方法については、「(4) Cosminexus コネクタの設定」を参照してください。

6. Server Plug-in の設定

Server Plug-in を設定します。Server Plug-in は、J2EE リソースのプロパティ設定などで使用します。設定方法については、「2.4.4 Server Plug-in の設定」を参照してください。

ここでは、テスト用実行環境のシステム構築およびプラグインを使用するための設定について説明します。

(1) MyEclipse を利用するための設定

J2EE アプリケーション開発で MyEclipse を利用するためには、次の設定が必要です。

- Eclipse で使用する JDK の設定
- ローカル変数情報の出力の設定
- ライブラリパスの設定

ここでは、ライブラリパスの設定方法について説明します。Eclipse で使用する JDK の設定方法およびローカル変数情報の出力の設定方法については、「2.4.1 Eclipse の設定」を参照してください。なお、インターネット接続時にプロキシを使用している環境では、MyEclipse でプロキシを設定する必要があります。MyEclipse のプロキシの設定方法については、「2.5.1 MyEclipse のプロキシの設定」を参照してください。

MyEclipse では、プロジェクト作成時にビルドパスが追加されます。ビルドパスの J2EE 1.4 ライブラリには、次に示す Cosminexus のライブラリパスが設定されています。

```
C:/Program Files/Hitachi/Cosminexus/CC/client/lib/j2ee-javax.jar  
C:/Program Files/Hitachi/Cosminexus/jaxp/lib/csmjaxp.jar
```

Cosminexus のライブラリパスの「C:/Program Files/Hitachi/Cosminexus」は、Developer のデフォルトのインストールディレクトリです。Developer のインストール時に、インストールディレクトリを変更している場合は、Cosminexus のライブラリパスを変更します。

Cosminexus のライブラリパスの確認手順と変更手順を説明します。

(a) ライブラリパスの確認手順

ライブラリパスの確認手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。

[設定] ダイアログが表示されます。

2. 左ペインのツリービューで [MyEclipse] - [Java エンタープライズ・プロジェクト] - [ライブラリー・セット] を選択します。

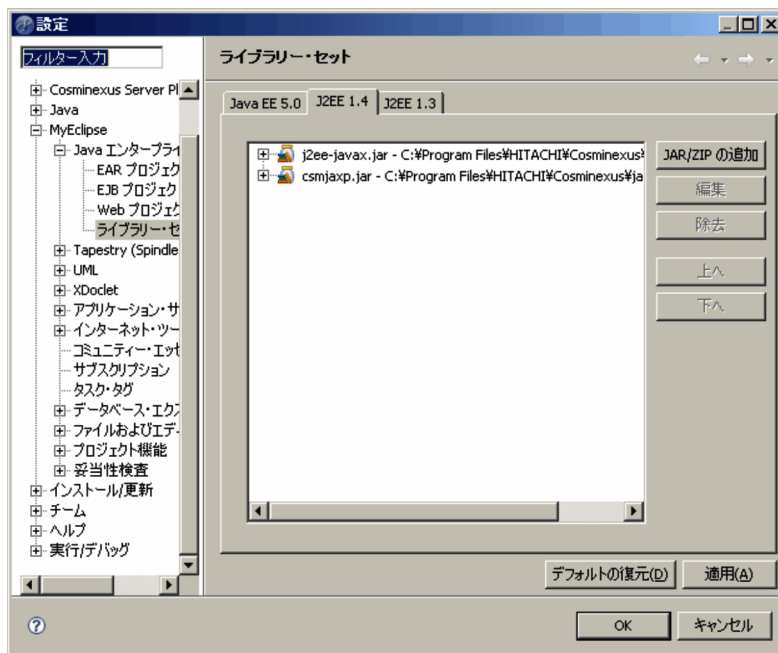
[ライブラリー・セット] ページが表示されます。

3. [ライブラリー・セット] ページの [J2EE 1.4] タブを選択します。

ライブラリパスが表示されます。ライブラリパスと Developer のインストールディレクトリが一致しているかどうかを確認してください。一致していない場合は、(2) の手順に従って、ライブラリパスを変更してください。

注

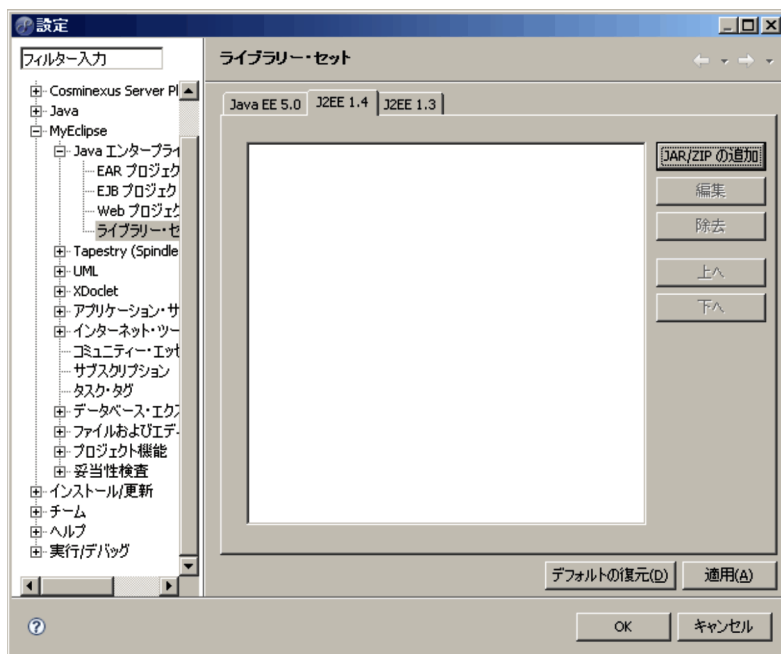
ライブラリの優先順位が変わるため、j2ee-javax.jar と csmjxap.jar の順序は変更しないでください。



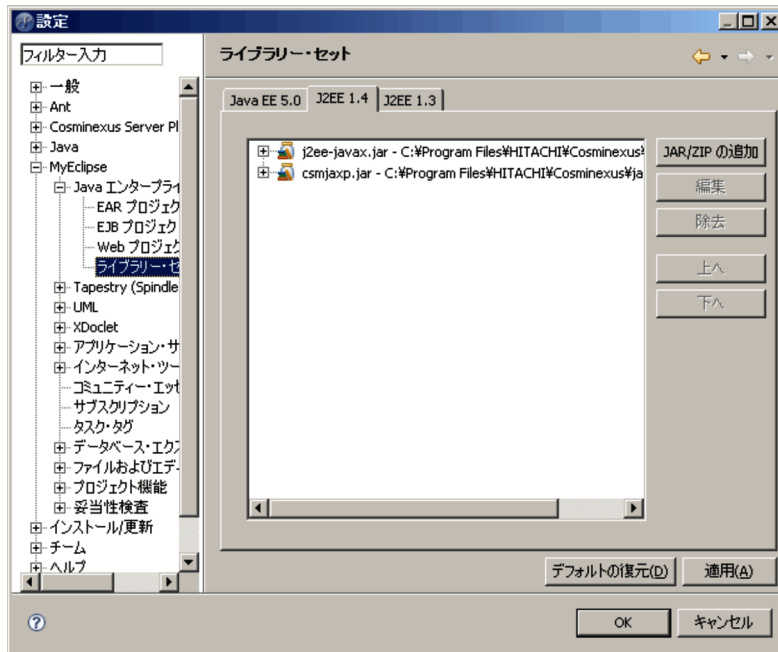
(b) ライブラリパスの変更手順

ライブラリパスの変更手順を次に示します。

1. [ライブラリー・セット] ページの [J2EE 1.4] タブで表示されている Cosminexus のライブラリパスをすべて選択し、[除去] ボタンをクリックします。
表示されている Cosminexus のライブラリパスが削除されます。



2. [JAR/ZIP の追加] ボタンをクリックします。
[ファイルを開く] ダイアログが表示されます。
3. <Cosminexus のインストールディレクトリ>%jaxp%lib%csmjaxp.jar を選択し、[開く] ボタンをクリックします。
csmjaxp.jar がライブラリパスに追加されます。
4. [JAR/ZIP の追加] ボタンをクリックします。
[ファイルを開く] ダイアログが表示されます。
5. <Cosminexus のインストールディレクトリ>%CC%client%lib%j2ee-javax.jar を選択し、[開く] ボタンをクリックします。
j2ee-javax.jar が csmjaxp.jar の上位に追加されます。追加する jar ファイルの順序は、j2ee-javax.jar、csmjaxp.jar の順になるようにしてください。
6. [ライブラリー・セット] ページの [J2EE 1.4] タブに追加されたライブラリパスを確認し、[適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。



(2) テスト用実行環境のシステム構築

Smart Composer 機能を使用したシステム構築の流れを次に示します。

1. Management Server の設定

Management Server とは、アプリケーションサーバを一括管理・運用するための運用管理プロセスです。

Management Server をセットアップして、Management Server の管理ユーザを設定します。また、Management Server および運用管理エージェントを自動起動するための設定もします。設定方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

なお、Management Server および運用管理エージェントの概要については、マニュアル「Cosminexus 機能解説」の Smart Composer 機能で構築したシステムの概要に関する説明を参照してください。

2. 簡易構築定義ファイルの作成

構築するシステムの情報を、簡易構築定義ファイルに定義します。簡易構築定義ファイルには、構築するシステムの名称やシステムに配置するサーバの種類などを定義します。また、J2EE アプリケーションのテスト用実行環境ではリロード機能を使用します。リロード機能を使用するための設定も簡易構築定義ファイルに定義します。作成方法については、「(a) 簡易構築定義ファイルの作成」を参照してください。

3. システムの構築

Smart Composer 機能で提供するコマンドを使用して、システムを構築します。構築

方法については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

4. JavaVM のセキュリティポリシーの設定

セキュリティマネージャを使用する場合は、必要に応じて JavaVM のセキュリティポリシーをあらかじめ設定しておきます。セキュリティマネージャを使用しない場合はこの設定は不要です。設定方法については、「(b) JavaVM のセキュリティポリシーの設定」を参照してください。

(a) 簡易構築定義ファイルの作成

構築するシステムの情報を定義した簡易構築定義ファイルを作成します。

ここでは、Cosminexus で提供しているテンプレートファイル (cmxdefcombinedmodel.xml) を使用して、簡易構築定義ファイルを作成します。簡易構築定義ファイルのテンプレートは次の場所に格納されています。テンプレートファイルを任意の場所にコピーして、簡易構築定義ファイルを作成してください。

テンプレートファイル (cmxdefcombinedmodel.xml) の格納先

<Cosminexusのインストールディレクトリ>%manager%config%templates%

簡易構築定義ファイルで定義する内容については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

なお、テスト用実行環境を開発環境とは別のマシンに構築している場合は、簡易 Web サーバにアクセスするホストを許可する設定が必要になります。簡易構築定義ファイルの <param> タグの下に次のタグを設定します。

- <param-name> タグ
「webserver.connector.http.permitted.hosts」を指定します。
- <param-value> タグ

簡易 Web サーバへのアクセスを許可するホスト名または IP アドレス (MyEclipse がインストールされている開発環境のホスト名または IP アドレス) を指定します。

(b) JavaVM のセキュリティポリシーの設定

JavaVM のセキュリティポリシーを設定するには、server.policy を編集します。server.policy の格納先を次に示します。

server.policy の格納先

<Cosminexus のインストールディレクトリ>%CC%server%usrconf%ejb%<サーバ名称>%server.policy

ポイント

server.policy の格納先である <サーバ名称> フォルダ、および server.policy は、
cmx_build_system コマンドを実行すると作成されます。

server.policy に設定する内容については、マニュアル「Cosminexus リファレンス 定義編」の server.policy (J2EE サーバ用セキュリティポリシーファイル) に関する説明を参照してください。

(3) プラグインを使用するための設定

Developer が提供するプラグイン (Cosminexus コネクタおよび Server Plug-in) は、Management Server リモート管理機能を使用して J2EE サーバに接続します。このため、Management Server リモート管理機能の設定が必要になります。また、MyEclipse および Server Plug-in でコンソールに情報を出力するためには設定が必要になります。ここでは、Management Server リモート管理機能の設定と、コンソールへの情報出力の設定について説明します。

(a) Management Server リモート管理機能の設定

Cosminexus コネクタおよび Server Plug-in を使用するためには、Management Server リモート管理機能の設定が必要です。Management Server リモート管理機能を使用するためには、mserver.properties (Management Server 環境設定ファイル) にプロパティを設定します。mserver.properties の格納先と追加するプロパティを次に示します。

格納先

<Cosminexus のインストールディレクトリ>¥manager¥config¥mserver.properties

追加するプロパティ

```
com.cosminexus.mngsvr.management.enabled=true
com.cosminexus.mngsvr.management.connector.enabled=true
com.cosminexus.mngsvr.management.port=28099
```

注 任意のポート番号を指定してください。デフォルトは 28099 です。

なお、Management Server 実行中にプロパティを設定しても反映されません。設定を反映するには、Management Server を再起動する必要があります。

com.cosminexus.mngsvr.management.port キーに設定するポート番号について

このキーに指定したポート番号は、Cosminexus コネクタおよび Server Plug-in で Management Server リモート管理機能にログインするときに使用します。ここで指定したポート番号は、[Cosminexus 7] ページ (Cosminexus コネクタの場合)、および [Cosminexus Server Plug-in] ページ (Server Plug-in の場合) のポート番号にも設定してください。Cosminexus コネクタの場合および Server Plug-in の場合のポート番号の設定について説明します。

- Cosminexus コネクタの場合

[Cosminexus 7] ページでは、デフォルトのポート番号として 28099 が設定されています。このため、`mserver.properties` でポート番号を変更した場合は、[Cosminexus 7] ページのポート番号も変更してください。[Cosminexus 7] ページでのポート番号の変更方法については、「(4)(a) 接続先ホストの設定変更」を参照してください。

なお、デフォルトのポート番号を使用する場合は [Cosminexus 7] ページでのポート番号の変更は不要です。

- Server Plug-in の場合

「2.4.4(1) 接続先ホストの設定」に示す手順に従ってポート番号を設定します。

(b) コンソールへの情報出力の設定

コンソールに情報を出力する場合は、`adminagent.properties` (運用管理エージェントプロパティファイル) に次の設定を追加してください。

格納先

```
<Cosminexus のインストールディレクトリ  
>%manager%\config\adminagent.properties
```

追加するプロパティ

```
adminagent.j2ee.process.console_event.enabled=true
```

注

`adminagent.properties` には、上記のプロパティがすでに記載されています。
`adminagent.properties` に記載されているプロパティを使用する場合は、次の内容を実施してください。

- コメントを削除します (# を削除します)。
- プロパティの値を、`false` から `true` に変更します。

なお、Management Server 実行中にプロパティを設定しても反映されません。設定を反映するには、Management Server および運用管理エージェントを再起動する必要があります。

ポイント

ここ設定しているのは、コンソールに情報を出力するかどうかです。コンソールに表示する文字数などの設定については、「2.5.4 コンソールの設定変更」を参照してください。

(4) Cosminexus コネクタの設定

Cosminexus コネクタは、MyEclipse から J2EE アプリケーションをデプロイしたり、J2EE サーバを起動したりするときに使用されます。ここでは、Cosminexus コネクタを使用するための設定について説明します。

デブロイ作業ディレクトリの設定およびソケット操作のブロックのタイムアウトの設定を変更したい場合は、Management Server リモート管理機能へログインする前に設定を変更する必要があります。設定の変更方法については、「2.5.2 デブロイ作業ディレクトリの設定変更」または「2.5.3 ソケット操作のブロックのタイムアウト設定変更」を参照してください。

また、Cosminexus コネクタから Management Server リモート管理機能へのログインおよびログアウト方法については、「2.4.2 Management Server リモート管理機能へのログインおよびログアウト (MyEclipse)」を参照してください。

(a) 接続先ホストの設定変更

Cosminexus コネクタの場合、Management Server リモート管理機能への接続情報がデフォルトで設定されています。このため、次に示す場合には、接続先ホストの設定を Management Server リモート管理機能にログインする前に変更する必要があります。

- 接続先が localhost で、msserver.properties (Management Server 環境設定ファイル) でポート番号をデフォルトから変更している場合
Cosminexus コネクタが接続するポート番号を変更します。
- 接続先が localhost 以外の場合
接続先が localhost 以外の場合 (MyEclipse と異なるマシンに Management Server がある場合)、接続ホストを追加する必要があります。

それぞれの場合の設定方法を次に説明します。なお、接続先が localhost で、msserver.properties に設定するポート番号をデフォルトのまま使用している場合、次に示す設定は不要です。

接続先のポート番号の変更

「(3)(a) Management Server リモート管理機能の設定」で接続先ポート番号をデフォルトの 28099 から変更している場合、[Cosminexus 7] ページでポート番号を変更する必要があります。ポート番号の設定手順を次に示します。

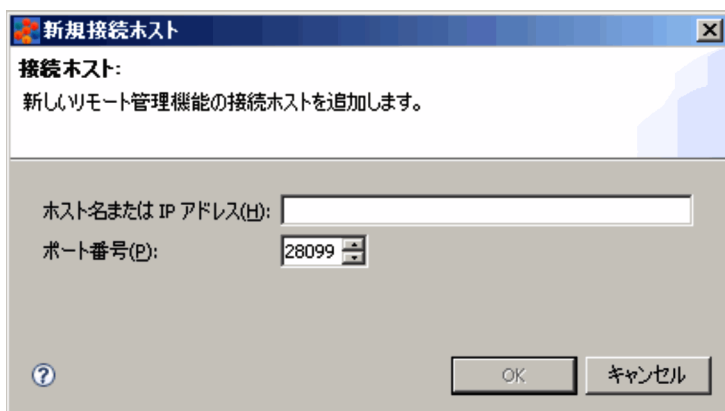
1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。
[Cosminexus 7] ページが表示されます。
3. [Cosminexus サーバー] ツリービューの [localhost] を選択し、[編集] ボタンをクリックします。
[接続ホストの編集] ダイアログが表示されます。
4. 変更後のポート番号を設定します。
[接続ホストの編集] ダイアログの [ポート番号] に、msserver.properties に設定したポート番号を設定します。

Management Server リモート管理機能への接続ホストの追加

MyEclipse とは別のマシンに Management Server がある場合、あらかじめ Management Server があるマシンを Management Server リモート管理機能の接続ホストに追加しておく必要があります。Management Server リモート管理機能の接続先が localhost の場合、この設定は不要です。

接続先ホストの追加手順を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] を選択します。
[Cosminexus 7] ページが表示されます。
3. [新規] ボタンをクリックします。
[新規接続ホスト] ダイアログが表示されます。



設定が必要な項目を次に示します。

項目名	設定値
ホスト名または IP アドレス	Management Server リモート管理機能の接続先となる Management Server のホスト名または IP アドレスを指定します。
ポート番号	Management Server リモート管理機能の接続先となる Management Server のポート番号を指定します。

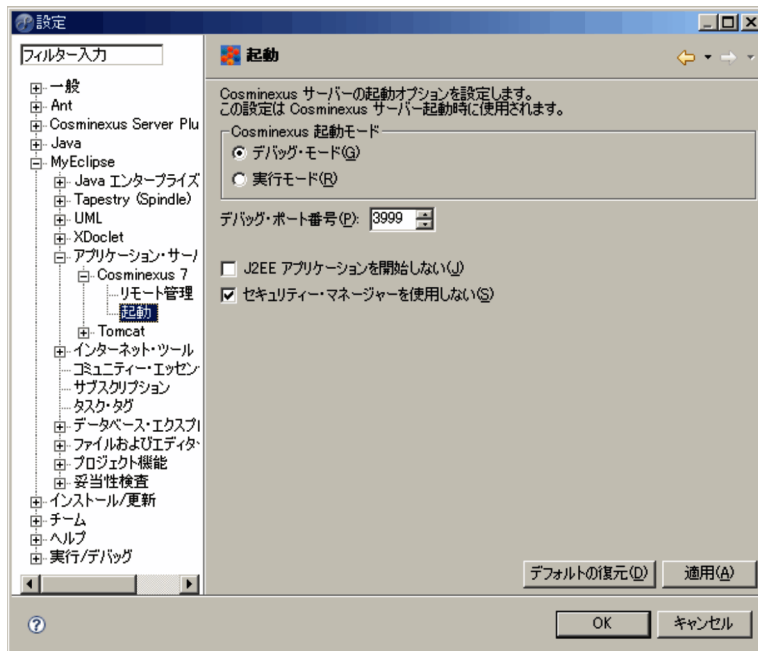
4. [OK] ボタンをクリックします。
[Cosminexus サーバー] ツリービューに追加したホストが表示されます。
5. [適用] ボタンまたは [OK] ボタンをクリックします。
追加したホストの設定が保存されます。

(b) J2EE サーバ起動時の動作の設定

J2EE サーバの起動時の動作を設定します。Management Server リモート管理機能にログインしてから設定してください。Management Server リモート管理機能へのログイン方法については、「2.4.2 Management Server リモート管理機能へのログインおよびログアウト (MyEclipse)」を参照してください。

設定手順を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [アプリケーション・サーバー] - [Cosminexus 7] - [起動] を選択します。
[起動] ページが表示されます。



必要に応じて、次の項目を設定してください。

項目名	設定値
Cosminexus 起動モード	<p>J2EE サーバの起動モードを選択します。</p> <ul style="list-style-type: none"> • デバッグ・モード デバッグモードで J2EE サーバを開始します。 • 実行モード 実行モードで J2EE サーバを開始します。 <p>Eclipse のツールバーから J2EE サーバを起動すると、ここで設定した起動モードで J2EE サーバが起動します。</p>

項目名	設定値
デバッグ・ポート番号	デバッグモードで J2EE サーバを起動するときに、デバッガ・アプリケーション（Eclipse）が接続する任意のポート番号を指定します。 未使用のポート番号を指定してください。
J2EE アプリケーションを開始しない	J2EE サーバ起動時に、J2EE アプリケーションを開始するかどうかを設定します。 <ul style="list-style-type: none"> • チェックする J2EE アプリケーションを開始しません。 • チェックしない J2EE アプリケーションを開始します。 MyEclipse の [サーバー] ビューからデプロイした J2EE アプリケーションだけをテスト・デバッグする場合などに、チェックします。
セキュリティ・マネージャを使用しない	J2EE サーバ起動時にセキュリティマネージャを使用するかどうかを設定します。 <ul style="list-style-type: none"> • チェックする セキュリティマネージャを使用しません。 • チェックしない セキュリティマネージャを使用します。

3. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

参考

J2EE サーバを追加したあとに、[Cosminexus 7] ページの [Cosminexus サーバー] ツリービューの表示を更新するには、再度、Management Server リモート管理機能にログインし直してください。

付録 B.3 組み込みデータベースの構築

組み込みデータベースは、Developer が提供するコマンドと HiRDB SQL Executer を使用して構築します。また、構築した組み込みデータベースを操作するには、HiRDB SQL Executer を使用します。

組み込みデータベースの構築および実装の流れを次に示します。

1. HiRDB SQL Executer のインストール

HiRDB SQL Executer をインストールします。インストール方法については、「3.2 HiRDB SQL Executer のインストール」を参照してください。

2. 組み込みデータベースの構築

コマンドと HiRDB SQL Executer を使用して、組み込みデータベースを構築します。組み込みデータベースのセットアップにはコマンドを、ユーザ定義やテーブル定義には HiRDB SQL Executer を使用します。構築手順については、「(1) 組み込みデータベースの構築手順」を参照してください。

3. RD エリアの追加

必要に応じて、データベースに RD エリアを追加します。追加方法については、「3.4.1 RD エリアの追加の流れ」を参照してください。

4. 組み込みデータベースの動作制御の設定

必要に応じて、組み込みデータベースの動作制御について設定します。設定方法については、「3.5 組み込みデータベースの動作制御の設定」を参照してください。

5. 組み込みデータベースの操作

HiRDB SQL Executer を使用して、構築した組み込みデータベースを操作します。操作方法については、「3.3 組み込みデータベースの操作」を参照してください。

この節では、HiRDB SQL Executer を使用した組み込みデータベースの構築の手順を説明します。

なお、構築した組み込みデータベースを削除する場合は、「(3) 組み込みデータベースの削除」を参照してください。また、構築した組み込みデータベースの開始および停止の手順については、「7.2 組み込みデータベースの開始および停止」を参照してください。

(1) 組み込みデータベースの構築手順

組み込みデータベースの構築方法には、次の二つがあります。

- データベースを構築したあとにユーザ定義およびテーブル定義を行う方法
- ユーザ定義およびテーブル定義を行ったあとにデータベースを構築する方法

それぞれの場合に分けて、組み込みデータベースの構築方法を説明します。

なお、組み込みデータベースを利用して、データベースにアクセスする J2EE アプリケーションを開発する場合、ここで示す手順のほかに、次のマニュアルを参照してください。

- 「HiRDB コマンドリファレンス」
- 「HiRDB UAP 開発ガイド」
- 「HiRDB SQL リファレンス」
- 「HiRDB メッセージ」

！ 注意事項

- ・ 組み込みデータベースを構築したあとに HiRDB のコマンドを実行する場合は、Windows の [スタート] メニューから [Cosminexus] - [Application Development Plug-in] - [組み込み DB プロンプト] から起動できる専用の DB コマンドプロンプトで、コマンドを入力してください。
- ・ 組み込みデータベースは、開始された状態で構築されます。
- ・ BINARY 型および BLOB 型のカラムのアンロードおよびロードはサポートしていません。これらの型のデータは、それぞれのアプリケーションで移行してください。
- ・ 組み込みデータベースの最大同時接続数は 8 接続です。
接続に使用するリソースアダプタの「コネクションの最小値と最大値」のプロパティ定義は「8」以下を設定してください。リソースアダプタのコネクションの最小値と最大値のデフォルト値は「10」です。「8」よりも大きい値を設定している状態でリソースアダプタを開始させると、開始時に KDJE49513-W のメッセージが出力されます。
なお、コネクションプールのウォーミングアップ機能を使用していない場合は、「8」よりも大きい値を設定していても開始時に KDJE49513-W のメッセージは出力されません。

(a) データベース構築後にユーザ定義およびテーブル定義を行う方法

データベースを構築したあとに、ユーザ定義およびテーブル定義を行う場合の手順を次に示します。

1. pdi_s.ini ファイルをコピーします。

組み込みデータベースを構築する前に、pdi_s.ini ファイルを所定のフォルダに上書きコピーする必要があります。コピー元ファイルとコピー先フォルダを次に示します。

・ コピー元ファイル

<Cosminexus のインストールディレクトリ >%SCC%\DB\bats%pdi_s.ini

・ コピー先フォルダ

<Cosminexus のインストールディレクトリ >%SCC%\DB\ini

2. 次のディレクトリを任意の場所にコピーします。

<Cosminexus のインストールディレクトリ >%SCC%\DB

ここでは、次のディレクトリにコピーしたものとして説明します。

C:%Setup_Input%\DB

3. 組み込みデータベース用の設定ファイルに記述を追加します。

C:%Setup_Input%\DB\bats\cdsetupconfig.bat ファイルに、次の項目を設定します。

- ・ データベースの領域サイズ (DB_SIZE)
- ・ RD エリアおよびシステムファイルを作成するディレクトリ (RDAREA_DIR)
- ・ 組み込みデータベースで使用するポート番号 (DB_PORT)
- ・ 手順 2. でコピーしたディレクトリのパス (INPUT_FILE_DIR)

この方法では、次の項目の記述は不要です。

- ・ テーブル所有者の認可識別子 (USRID)
- ・ テーブル所有者のパスワード (PSWD)

設定ファイルの記述例を次に示します。

```
@rem Environment settings file used by HiRDB Embedded Server
@rem Please do not use double quotation marks (") in the file path of this
file.

@rem Table owner's authorization identifier
set USRID=

@rem Table owner's password
set PSWD=

@rem Area size of the database to be constructed (required)
set DB_SIZE=S

@rem Directories where the RD area and system files are to be created
(required)
set RDAREA_DIR=C:\Program Files\Hitachi\Cosminexus\SCC\DB\area

@rem TCP/IP port number used by HiRDB Embedded Server
set DB_PORT=22200

@rem Directory where the input files used for HiRDB Embedded Server
execution exist
set INPUT_FILE_DIR=C:\Setup_Input\DB
```

空白を含むパスでも、() ダブルクォーテーションを使用しないで記述してください。

各設定項目については、「(2) 組み込みデータベース用の設定ファイルの設定項目」を参照してください。

4. ユーザ定義ファイルおよびテーブル定義ファイルを削除します。
次のディレクトリにある grantuser ファイルと tablecreate ファイルを削除するか、
ファイル名を変更してください。
C:\Setup_Input\DB\bats\
5. 組み込みデータベース構築用のバッチファイルを実行します。
コマンドプロンプトのカレントディレクトリを C:\Setup_Input\DB\bats\ に変更し
て、次のコマンドを実行してください。

```
cddbconstruct.bat
```

エラーとなった場合は、バッチファイルを実行したプロンプト上に、構築に失敗したというメッセージが表示されます。この場合、次に示す内容に従って対処してください。

- pdi_err.txt を参照してエラーの原因を特定する。
- エラーの原因を取り除き、組み込みデータベースを再構築する。

組み込みデータベースを再構築する前に、組み込みデータベースを削除してくださ

い。削除する手順については、「(3) 組み込みデータベースの削除」を参照してください。

6. HiRDB SQL Executer を起動して、データベースに接続します。

HiRDB SQL Executer の [CONNECT] ダイアログでは、次に示す情報を入力してください。

- 認可識別子: "root"
- パスワード: "root"
- ホスト名: localhost
- ポート番号: 22200 (設定ファイルの DB_PORT で指定した値)

注 認可識別子およびパスワードには " も含めて指定してください。

7. ユーザ定義をします。

データベースに接続後、データベースにアクセスするユーザを定義します。認可識別子を USER1、パスワードを PSWD とする場合の定義例を示します。

```
GRANT DBA TO USER1 IDENTIFIED BY PSWD;
```

8. 定義したユーザで HiRDB SQL Executer に接続し、テーブル定義をします。

CREATE SCHEMA 文や CREATE TABLE 文などで定義します。また、insert 文でデータをロードできます。SQL 文の文法については、マニュアル「HiRDB SQL リファレンス」を参照してください。

テーブル定義とデータのロードの例を次に示します。

```
CREATE SCHEMA;
CREATE TABLE CUSTOM
(
CUSTOMCD      CHAR(5) ,
CUSTOMNAME    CHAR(30),
TELNO         CHAR(12),
ZIPCD         CHAR(3) ,
ADDRESS       CHAR(30))
IN RDDATA10 PCTFREE=(10,0);
CREATE INDEX CUSTOMX ON CUSTOM
(
CUSTOMNAME ASC)
IN RDINDX10 PCTFREE=10;
insert into CUSTOM values('TK001', ' ミキダデンキ ', '011-434-1998', '091',
' トツカ ');
insert into CUSTOM values('TK002', ' シンコウデパート ', '012-402-5055',
'082', ' ヨコハマ ');
insert into CUSTOM values('TK003', ' キセツヤ ', '014-252-1555', '043', ' カ
ナザワ ');
```

データのロードは SQL 文を使用する方法のほかに、pload コマンドを使用する方法もあります。pload コマンドについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。

(b) ユーザ定義およびテーブル定義後にデータベースを構築する方法

ユーザ定義およびテーブル定義を行ったあとに、データベースを構築する場合の手順を次に示します。

1. pdi_s.ini ファイルをコピーします。

組み込みデータベースを構築する前に、pdi_s.ini ファイルを所定のフォルダに上書きコピーする必要があります。コピー元ファイルとコピー先フォルダを次に示します。

• コピー元ファイル

<Cosminexus のインストールディレクトリ >%SCC%\DB\bats\pdi_s.ini

• コピー先フォルダ

<Cosminexus のインストールディレクトリ >%SCC%\DB\ini

2. 次のディレクトリを任意の場所にコピーします。

<Cosminexus のインストールディレクトリ >%SCC%\DB

ここでは、次のディレクトリにコピーしたものとして説明します。

C:\Setup_Input\DB

3. 組み込みデータベース用の設定ファイルに記述を追加します。

C:\Setup_Input\DB\bats\cdsetupconfig.bat ファイルに、次の項目を設定します。

- テーブル所有者の認可識別子 (USRID)
- テーブル所有者のパスワード (PSWD)
- データベースの領域サイズ (DB_SIZE)
- RD エリアおよびシステムファイルを作成するディレクトリ (RDAREA_DIR)
- 組み込みデータベースで使用するポート番号 (DB_PORT)
- 手順 2. でコピーしたディレクトリのパス (INPUT_FILE_DIR)

設定ファイルの記述例を次に示します。

```

@rem Environment settings file used by HiRDB Embedded Server
@rem Please do not use double quotation marks (") in the file path of this
file.

@rem Table owner's authorization identifier
set USRID=USER1

@rem Table owner's password
set PSWD=PSWD

@rem Area size of the database to be constructed (required)
set DB_SIZE=S

@rem Directories where the RD area and system files are to be created
(required)
set RDAREA_DIR=C:\Program Files\Hitachi\Cosminexus\SCC\DB\area

@rem TCP/IP port number used by HiRDB Embedded Server
set DB_PORT=22200

@rem Directory where the input files used for HiRDB Embedded Server
execution exist
set INPUT_FILE_DIR=C:\Setup_Input\DB

```

空白を含むパスでも、(") ダブルクォーテーションを使用しないで記述してください。

記述例では、認可識別子を USER1、パスワードを PSWD としています。

各設定項目については、「(2) 組み込みデータベース用の設定ファイルの設定項目」を参照してください。

4. ユーザ定義ファイルを作成します。

C:\Setup_Input\DB\bats に格納されているユーザ定義ファイルの grantuser ファイルは、SQL の GRANT 文を記述したテキストファイルです。次に作成規則を示します。

- ファイル名は「grantuser」で固定です。
- このファイルは、C:\Setup_Input\DB\bats に作成してください。
- grantuser ファイルで指定する認可識別子およびパスワードは、C:\Setup_Input\DB\bats\cdsetupconfig.bat ファイルで指定した USRID パラメタおよび PSWD パラメタと同じ値にしてください。
- ユーザ定義ファイルがない場合は、ユーザ定義を実行しません。また、ユーザ定義を実行しないでテーブルを作成した場合、テーブル所有者の認可識別子とパスワードとともに「root」になります。

次に、認可識別子を USER1、パスワードを PSWD とした場合の定義例を示します。

```

/* grantuser */
GRANT DBA TO USER1 IDENTIFIED BY PSWD;
GRANT SCHEMA TO USER1;

```

5. テーブル定義ファイルを作成します。

C:\¥Setup_Input¥DB¥bats¥ に格納されているテーブル定義ファイルの tablecreate ファイルは、CREATE SCHEMA 文や CREATE TABLE 文などの定義系 SQL (DDL 文) を記述したテキストファイルです。次に作成規則を示します。

- このファイルは C:\¥Setup_Input¥DB¥bats¥ に作成します。
- テーブル定義ファイル名は「tablecreate」で固定です。
- テーブル定義ファイルがない場合には、テーブル定義を実行しません。

SQL 文の文法については、マニュアル「HiRDB SQL リファレンス」を参照してください。テーブルを作成する例を次に示します。

```
CREATE SCHEMA;
CREATE FIX TABLE CUSTOM
(
  CUSTOMCD      CHAR(5) ,
  CUSTOMNAME    CHAR(30) ,
  TELNO         CHAR(12) ,
  ZIPCD         CHAR(3)  ,
  ADDRESS       CHAR(30) )
IN RDDATA10 PCTFREE=(10,0);
CREATE INDEX CUSTOMX ON CUSTOM
(
  CUSTOMNAME    ASC)
IN RDINDX10 PCTFREE=10;
```

テーブルデータをロードする場合は、データベースを構築したあとに pdload コマンドを使用してください。pdload コマンドについては、マニュアル「HiRDB コマンドリファレンス」を参照してください。

6. 組み込みデータベース構築用のバッチファイルを実行します。

コマンドプロンプトのカレントディレクトリを C:\¥Setup_Input¥DB¥bats¥ に変更して、次のコマンドを実行してください。

```
cd dbconstruct.bat
```

エラーとなった場合は、バッチファイルを実行したプロンプト上に、構築に失敗したというメッセージが表示されます。この場合、次に示す内容に従って対処してください。

- pdi_err.txt を参照してエラーの原因を特定する。
- エラーの原因を取り除き、組み込みデータベースを再構築する。

組み込みデータベースを再構築する前に、組み込みデータベースを削除してください。削除する手順については、「(3) 組み込みデータベースの削除」を参照してください。

(2) 組み込みデータベース用の設定ファイルの設定項目

ここでは、次に示す組み込みデータベース用の設定ファイル (cdsetupconfig.bat) の設

定項目について説明します。

```
@rem ALL RIGHTS RESERVED, COPYRIGHT (C) 2004, HITACHI, LTD.
@rem LICENSED MATERIAL OF HITACHI, LTD.
@title HiRDB Embedded Server
@echo off

@rem Environment settings file used by HiRDB Embedded Server
@rem Please do not use double quotation marks (") in the file path of this
file.

@rem Table owner's authorization identifier
set USRID=USER1                                . . . (1)

@rem Table owner's password
set PSWD=PSWD                                  . . . (2)

@rem Area size of the database to be constructed (required)
set DB_SIZE=S                                  . . . (3)

@rem Directories where the RD area and system files are to be created
(required)
set RDAREA_DIR=C:\Program Files\Hitachi\Cosminexus\SCC\DB\area    . . . (4)

@rem TCP/IP port number used by HiRDB Embedded Server
set DB_PORT=22200                                . . . (5)

@rem Directory where the input files used for HiRDB Embedded Server
execution exist
set INPUT_FILE_DIR=C:\Program Files\Hitachi\Cosminexus\SCC\DB    . . . (6)
```

設定ファイルに記述を追加するときは、次の点に注意してください。

- 空白を含むパスでも、(") ダブルクォーテーションを使用しないでください。
- 空白も文字として認識します。イコール (=) のあとに空白を入れないでください。

(1) ~ (6) の各設定項目について説明します。

1. USRID (認可識別子)

テーブル所有者の認可識別子を設定します。この設定は、cddbconstruct.bat コマンドでシステム定義と、ユーザ定義・テーブル定義を同時に行う場合に必要です。grantuser ファイルで指定した認可識別子と同じものを指定してください。

2. PSWD (パスワード)

テーブル所有者のパスワードを設定します。この設定は、cddbconstruct.bat コマンドでシステム定義と、ユーザ定義・テーブル定義を同時に行う場合に必要です。grantuser ファイルで指定したパスワードと同じものを指定してください。

3. DB_SIZE (データベースの領域サイズ)

構築するデータベースの領域サイズを設定します。この設定は必須です。提供しているひな型のデータベース領域サイズは、システムファイルと RD エリアを合わせたサ

イズで、S、M、L のどれかを指定します。

- S : 200MB
- M : 350MB
- L : 660MB

4. RDAREA_DIR (RD エリアを作成するディレクトリ)

RD エリアを作成するディレクトリおよびシステムファイルを作成するディレクトリを絶対パスで設定します。この設定は必須です。指定する形式を示します。

```
RDAREA_DIR=rdarea_directory[,systemA_directory[,systemB_directory]]
```

- rdarea_directory
RD エリアを作成するディレクトリを指定します。
- systemA_directory
A 系システムファイルを作成するディレクトリを指定します。
- systemB_directory
B 系システムファイルを作成するディレクトリを指定します。

指定時は、次の点に注意してください。

- RDAREA_DIR で指定するディレクトリは 48 バイト以内の半角英数字で指定してください。
- systemA_directory を省略した場合、rdarea_directory にシステムファイルが作成されます。
- systemB_directory を省略した場合、systemA_directory に B 系システムファイルが作成されます。
- systemB_directory を指定した場合、初期値ファイルで指定するシステムファイル作成先の領域は、A 系と B 系で分ける必要があります。
- 同じファイルシステム領域が使用されている場合、コマンド実行時にエラーとなります。
- 存在しないディレクトリを指定した場合は、新規にディレクトリが作成されます。
- 指定したディレクトリのディスク容量がセットアップするファイル容量よりも小さい場合はエラーとなります。

5. DB_PORT (ポート番号)

組み込みデータベースで使用する TCP/IP ポート番号を設定します (pd_name_port に対応します)。指定時は、次の点に注意してください。

- 指定値は 5001 ~ 65535 の範囲で指定します。
- 省略時は 22200 を仮定します。
- すでに存在するポート番号を指定した場合はエラーとなります。存在するポート番号とは services ファイルに記述されているポート番号を示します。なお、ほかの HiRDB ポート番号との重複チェックはされません。

6. INPUT_FILE_DIR (入力ファイルのディレクトリ)

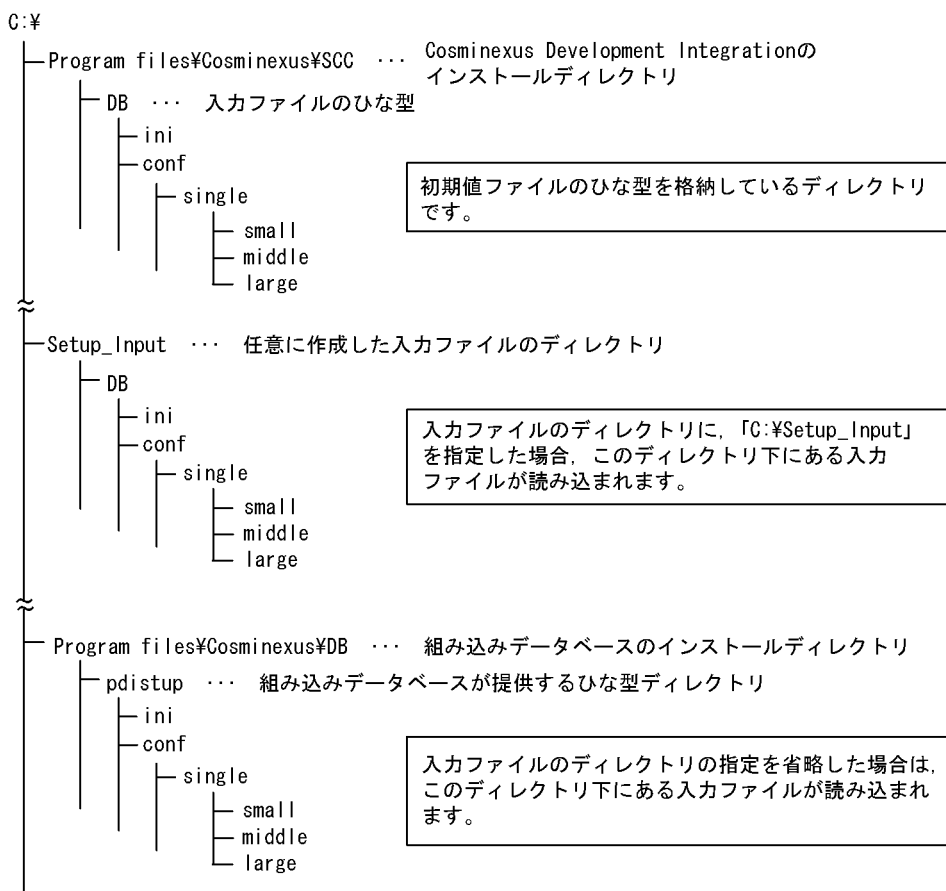
入力ファイルの存在するディレクトリを絶対パスで設定します。入力ファイルとは、

初期値ファイルおよびシステム定義入力ファイルを指します。指定時は、次の点に注意してください。

- 指定したディレクトリの下に、ひな型として提供されているディレクトリ構成と同様に conf および ini 以下のディレクトリを作成する必要があります。
- 指定を省略した場合は、環境変数 COSMINEXUS_HOME¥DB¥PDISTUP ディレクトリが仮定されます。

ひな型として提供されているディレクトリ、任意に作成するディレクトリ、および指定省略時に読み込まれるディレクトリ構成を次の図に示します。

図 B-2 組み込みデータベースの入力ファイルのディレクトリ構成



(3) 組み込みデータベースの削除

組み込みデータベースは、次に示すバッチファイルを実行して削除します。

```
cddbdelete.bat
```


エラーとなった場合は、バッチファイルを実行したプロンプト上に、構築に失敗したというメッセージが表示されます。この場合、pdirst_err.txt を参照してエラーの原因を取り除き、再度 cddbdelete.bat を実行してください。

付録 B.4 リソースアダプタのプロパティ設定

インポートしたリソースアダプタを J2EE アプリケーションで使用するためには、リソースアダプタのプロパティを設定する必要があります。リソースアダプタのプロパティ設定には、Server Plug-in を使用します。

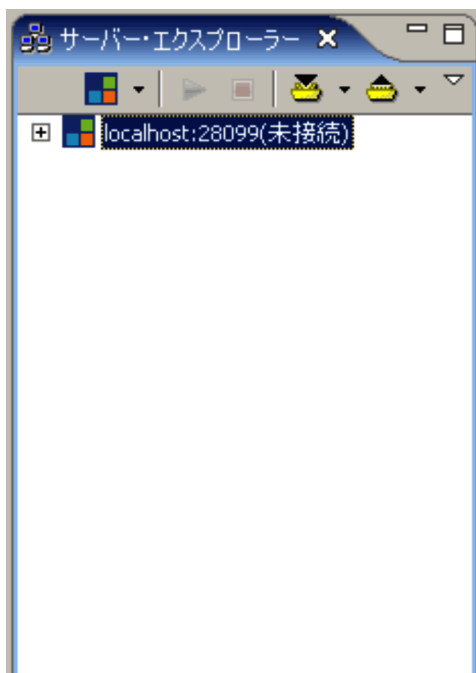
！ 注意事項

リソースアダプタのプロパティを編集する前に、必ず J2EE アプリケーションを停止してください。

参考

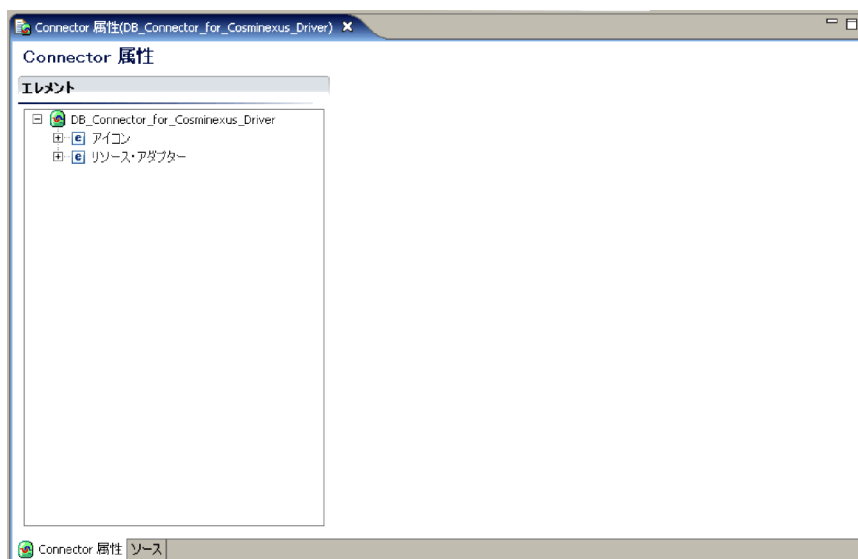
リソースアダプタのプロパティ設定は、[Cosminexus Server Plug-in] パースペクティブの [サーバー・エクスプローラー] ビューを使用します。[サーバー・エクスプローラー] ビューが表示されていない場合は、次の手順で表示させてください。

1. Eclipse のメニューから [ウィンドウ] - [ビューの表示] - [その他] を選択します。
[ビューの表示] ダイアログが表示されます。
2. [Cosminexus Server Plug-in] - [サーバー・エクスプローラー] を選択して、[OK] ボタンをクリックします。
[サーバー・エクスプローラー] ビューが表示されます。



次の手順でリソースアダプタのプロパティを設定してください。

1. [サーバー・エクスプローラー] ビューの [<Management Server リモート管理機能のホスト> (接続済)] - [DefaultDomain] - [<J2EE サーバのホスト>] - [<J2EE サーバ名>] - [J2EE アプリケーション] - [<リソースアダプタを含む J2EE アプリケーションの表示名>] - [<インポートしたリソースアダプタの表示名>] を選択して、コンテキストメニューから [プロパティ] を選択します。
[Connector 属性] エディタが表示されます。



2. [<インポートしたリソースアダプタの表示名>] - [リソースアダプター] - [アウトバウンド・リソース・アダプター] - [コネクション定義] - [<コネクション定義識別子>] を選択します。
3. 設定するプロパティを , [<コネクション定義識別子>] - [コンフィグレーション・プロパティー] 下から選択して , [コンフィグレーション・プロパティーの値] テキストボックスに設定を記述します。
設定する項目と設定例を次に示します。

項目名	設定内容
networkProtocol	Cosminexus DABroker Library との接続種別を設定します。 ここでは , 「 lib 」 を指定します。
databaseName	接続するデータベースの種別を設定します。ここでは , 組み込みデータベースに接続するため , 「 HIRDB 」 を指定します。
description	組み込みデータベースのポート番号を指定します。「付録 B.3(1) 組み込みデータベースの構築手順」で設定したポート番号「22200」を指定します。
DBHostName	接続する組み込みデータベースのホスト名を設定します。接続先がローカルホストの場合は , 「 localhost 」 を指定します。

4. [<コネクション定義識別子>] - [コネクター・ランタイム] - [プロパティー] を選択して , 次の項目を入力します。

項目名	設定例
ユーザー名	USER1 ¹
パスワード	PSWD ¹

項目名	設定例
コネクション・プールの最小値 ²	1
コネクション・プールの最大値 ²	8

注 1

「付録 B.3(1) 組み込みデータベースの構築手順」で設定したユーザ名とパスワードを入力してください。

注 2

リソースアダプタの開始時にコネクションプールの最小値および最大値を指定していない場合は、メッセージ (KDJE49513-W) が [コンソール] ビューに出力されます。

- 設定終了後、Eclipse のメニューから [ファイル] - [保管] を選択します。
プロパティに設定した内容が保管されます。
- [サーバー・エクスプローラー] ビューの [<Management Server リモート管理機能のホスト> (接続済)] - [DefaultDomain] - [<J2EE サーバのホスト>] - [<J2EE サーバ名>] - [J2EE アプリケーション] - [<リソースアダプタを含む J2EE アプリケーションの表示名>] - [<インポートしたリソースアダプタの表示名>] を選択して、コンテキストメニューから [接続テスト] を選択します。
接続できるかどうか確認します。接続できない場合は、手順 2 からやり直してください。
- J2EE アプリケーションをデプロイしたあとに編集した場合は、MyEclipse から J2EE アプリケーションを開始します。
MyEclipse からの J2EE アプリケーションの開始は、プロジェクトの入れ替え機能を使用します。
MyEclipse を使用した入れ替えの手順については、「7.5 プロジェクトの入れ替え」を参照してください。MyEclipse のリデプロイでは、プロジェクトの入れ替えおよび J2EE アプリケーションの開始を実施します。

付録 B.5 アンインストール

J2EE アプリケーション開発環境で使用したプログラムのアンインストールについて説明します。アンインストールは次の順序で実施してください。

- プラグインのアンインストール
- MyEclipse のアンインストール
- Developer のアンインストール

ここでは、プラグインおよび MyEclipse のアンインストールについて説明します。

Developer のアンインストールについては、「2.8 Developer のアンインストール」を参照してください。

(1) プラグインのアンインストール

Eclipse に組み込んだ Developer のプラグインをアンインストールします。アンインストールをするには、Eclipse のインストールディレクトリで、次のコマンドを実行してください。

```
java -cp startup.jar org.eclipse.core.launcher.Main
-application org.eclipse.update.core.standaloneUpdate
-command removeSite
-to "<Cosminexusのインストールディレクトリ>%plugins%eclipse"
```

注

改行を入れないでコマンドを実行してください。

なお、このコマンドでは、<Cosminexus のインストールディレクトリ>%plugins%eclipse 下のファイルおよびフォルダは削除されません。

! 注意事項

次に示す条件にすべて該当する場合、アンインストールするときのコマンドの指定が異なります。

- 使用する OS が Windows Vista の場合
- 管理者特権のないユーザがアンインストールする場合
- Program Files ディレクトリに Eclipse をインストールしている場合

条件にすべて該当する場合、Eclipse のインストールディレクトリで、次のコマンドを実行してください。

```
java -cp startup.jar org.eclipse.core.launcher.Main
-application org.eclipse.update.core.standaloneUpdate
-command removeSite
-to "<Cosminexusのインストールディレクトリ>%plugins%eclipse"
-configuration <起動時に指定したディレクトリ>
```

注

改行を入れないでコマンドを実行してください。

注

Eclipse 起動時の eclipse.exe で、-configuration に指定したディレクトリを指定します。

(2) MyEclipse のアンインストール

MyEclipse をアンインストールします。MyEclipse のアンインストール手順を次に示します。

1. Eclipse から MyEclipse をアンインストールします。
Eclipse のインストールディレクトリで、次のコマンドを実行します。

```
java -cp startup.jar org.eclipse.core.launcher.Main
    -application org.eclipse.update.core.standaloneUpdate
    -command removeSite
    -to "<MyEclipseのインストールディレクトリ>%eclipse"
```

注 改行を入れないでコマンドを実行してください。

なお、このコマンドでは、<MyEclipse のインストールディレクトリ >%eclipse 下のファイルおよびフォルダは削除されません。<MyEclipse のインストールディレクトリ >%eclipse 下のファイルおよびフォルダが不要な場合は、エクスプローラなどを使用して削除してください。

2. eclipse.ini をコピーします。

MyEclipse のインストール時に退避しておいた eclipse.ini を、Eclipse のインストールディレクトリ直下にコピーします。これによって、Eclipse を、MyEclipse をインストールする前の状態に戻します。

! 注意事項

次に示す条件にすべて該当する場合、アンインストールするときのコマンドの指定が異なります。

- 使用する OS が Windows Vista の場合
- 管理者特権のないユーザがアンインストールする場合
- Program Files ディレクトリに Eclipse をインストールしている場合

条件にすべて該当する場合、Eclipse のインストールディレクトリで、次のコマンドを実行してください。

```
java -cp startup.jarorg.eclipse.core.launcher.Main
    -application org.eclipse.update.core.standaloneUpdate
    -command removeSite
    -to "<MyEclipseのインストールディレクトリ>%eclipse"
    -configuration <起動時に指定したディレクトリ>
```

注

改行を入れないでコマンドを実行してください。

注

Eclipse 起動時の eclipse.exe で、-configuration に指定したディレクトリを指定します。

付録 C Developer Standard 使用時の注意事項

Developer では、開発した J2EE アプリケーションをテストするための環境を提供しています。ただし、Developer Standard が提供するテスト用実行環境では、使用できる機能に一部制限があります。

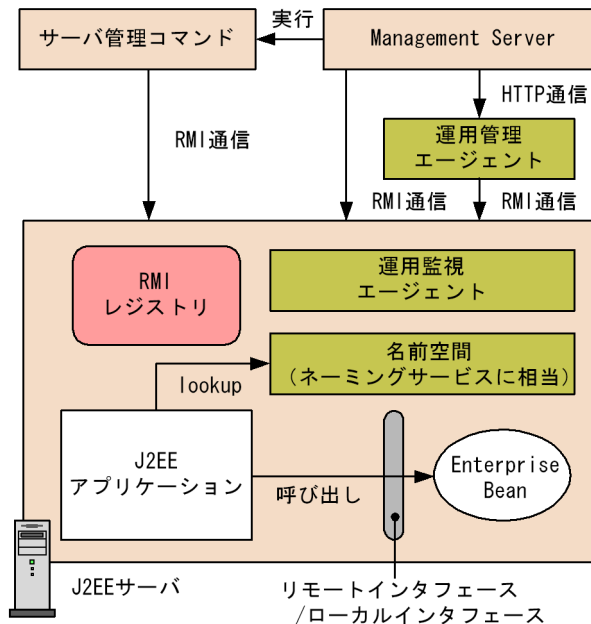
ここでは、Developer Standard のテスト用実行環境を説明したあとに、Developer Professional との機能差を示します。また、Developer Standard で開発した J2EE アプリケーションを実行環境にインポートする方法について説明します。

付録 C.1 Developer Standard が提供するテスト用実行環境

Developer Standard では、同一 J2EE サーバ上に Web アプリケーションと Enterprise Bean を配置する構成で使用します。

Developer Standard が提供するテスト用実行環境の構成を次に示します。

図 C-1 Developer Standard が提供するテスト用実行環境の構成



Developer Standard では、サーバ管理コマンドと J2EE サーバ間の通信に RMI レジストリを使用します。

また、J2EE アプリケーションの実行機能では、次の制限があります。

- JNDI の検索や Enterprise Bean に対するリクエストを送信できる範囲は、同じ J2EE サーバ内に制限されます。

- ラウンドロビン検索機能，CTM 連携機能，セッションフェイルオーバー機能などプロセ
ス間通信を前提とした機能は使用できません。
- グローバルトランザクションは使用できません。

なお，サーバ管理コマンドが使用する RMI レジストリのポート番号を変更する場合，次
のどちらかの方法で指定します。

サーバ管理コマンドの `-nameserver` オプションの指定

サーバ管理コマンド実行時に，`-nameserver` オプションを指定することで，RMI レジ
ストリのポート番号を指定します。オプションの指定形式を示します。

`-nameserver rmi://localhost:<RMI レジストリのポート番号>`

`-nameserver` オプションを指定しない場合は，デフォルト値の 23152 が使用されま
す。ただし，ユーザ定義ファイル（`usrconf.properties`）に RMI レジストリのポート
番号を定義している場合は，その値が使用されます。

ユーザ定義ファイル（`usrconf.properties`）の定義の追加

ユーザ定義ファイル（`usrconf.properties`）に，RMI レジストリのポート番号の定義
を追加します。ユーザ定義ファイルの格納先と指定するプロパティを示します。

格納先

`<Cosminexus インストールディレクトリ
>%CC%\admin\usrconf\usrconf.properties`

指定するプロパティ

`ejbserver.rmi.naming.port`

ポート番号は 1 ～ 65535 の範囲で指定します。デフォルト値は 23152 です。指定し
ない場合は，デフォルト値が使用されます。

付録 C.2 Developer Professional との機能差

Developer Standard と Developer Professional を比較した場合の機能差，および
Developer Standard の制限事項を示します。

（１）システム構築時の機能差

システム構築には，Smart Composer 機能を使用します。Developer Standard でシステ
ムを構築する場合，次に示す制限事項があります。

（a）Smart Composer 機能で使用できないパラメタ

Smart Composer 機能でシステムを構築するときに，簡易構築定義ファイルを作成しま
す。Developer Standard の場合，簡易構築定義ファイルに，次に示すパラメタを使用で
きません。なお，構成変更定義ファイルでシステムの設定を変更する場合も，次に示す
パラメタを使用できません。

- `inprocess.ns.port`
- `webserver.sfo.sfo_servers`

- webserver.sfo.sfo_server.<SFO フィルタのサーバ定義名>.serverName
- webserver.sfo.sfo_server.<SFO フィルタのサーバ定義名>.connectionTimeout
- ejbserver.ctm.CTMID
- ejbserver.distributedtx.XATransaction.enabled
- ejbserver.distributedtx.recovery.port
- vbj.java2iop.jvm.minHeapSize
- vbj.java2iop.jvm.maxHeapSize
- ejbserver.ctm.ActivateTimeOut
- ejbserver.ctm.DeactivateTimeOut
- ejbserver.ctm.QueueLength
- ejbserver.client.ctm.RequestPriority
- ejbserver.rmi.request.timeout
- ejbserver.DynamicStubLoading.Enabled
- vbroker.se.iiop_tp.host.value
- vbroker.se.iiop_tp.scm.iiop_tp.listener.port
- inprocess.ns.waitTime
- inprocess.ns.retryCount
- ejbserver.jndi.request.timeout
- ejbserver.jndi.cache
- ejbserver.jndi.cache.interval
- ejbserver.jndi.cache.interval.clear.option
- java.naming.factory.initial
- jndi.rr.ns.list
- ejbserver.jdbc.propertyInfo.BUF_SIZE
- ejbserver.jdbc.propertyInfo.ENCODELANG
- ejbserver.distributedtx.ots.status.directory1
- ejbserver.distributedtx.ots.status.directory2
- ejbserver.distributedtx.enableXidReuseOptimization
- ejbserver.distributedtx.Synchronization.port
- ejbserver.ctm.CTMDomain
- ejbserver.ctm.CTMMHost
- ejbserver.ctm.enabled
- ejbserver.distributedtx.recovery.completionCheckOnStopping.timeout
- ejbserver.jta.TransactionManager.enlist.compatibleMode.enabled
- ejbserver.rmi.stateless.unique_id.enabled
- ejbserver.container.ejbhome.sessionbean.reconnect.enabled

(b) ホスト定義の制限

Developer Standard を使用する場合、簡易構築定義ファイルおよび構成変更定義ファイル中で定義するホストの運用 IP と管理 IP は、Management Server のマシンの IP アド

レスで指定してください。ほかのマシンの IP アドレスを指定した場合は、使用できないことを示すエラーメッセージが出力されます。

(c) ティア定義の制限

Developer Standard では、combined-tier, j2ee-tier, および free-tier を利用できます。ほかのティアを指定した場合は、使用できないことを示すエラーメッセージが出力されます。

(d) cmx コマンドで接続できる Management Server の制限

Smart Composer 機能で提供するコマンドを実行すると、Management Server に接続します。Smart Composer 機能のコマンドで接続できる Management Server は、コマンドを実行しているローカルマシンだけです。ほかのマシンを指定した場合は、使用できないことを示すエラーメッセージが出力されます。

注

Smart Composer 機能のコマンドが接続する Management Server とは、Smart Composer 機能のコマンドの `-m` オプションに指定する Management Server のホストです。`-m` オプションについては、マニュアル「Cosminexus 簡易構築・運用ガイド」のコマンドに関する説明を参照してください。

(2) J2EE サーバの運用での機能差

Developer Standard の運用時に使用するサーバ管理コマンドのうち、次に示すコマンドは使用できません。

- `cjgetstubsjar` (RMI-IIOP スタブとインタフェースの取得)
- `cjlisttrn` (稼働中の J2EE サーバのトランザクション情報の表示)
- `cjlisttrnfile` (停止中の J2EE サーバのトランザクション情報の表示)
- `cjstartrecover` (J2EE サーバのトランザクション回復)

(3) J2EE アプリケーションの実行での機能差

Developer Standard では、JNDI 機能と EJB 機能で制限があります。JNDI 機能および EJB 機能のサポート範囲を示します。

(a) JNDI 機能のサポート範囲

JNDI の名前空間を利用したオブジェクトの登録、削除、検索機能はサポートされます。ただし、使用する名前空間は、ネーミングサービスではなく、J2EE サーバ内で管理する名前空間を使用します。

次の表に、JNDI 機能のサポート範囲を示します。

表 C-1 Developer Standard での JNDI 機能のサポート範囲

機能	サポート	備考
ネーミングサービス接続機能		CORBA ネーミングサービスには登録しないで、J2EE サーバ内に名前空間を作成して利用します。
ネーミングサービス切替機能	×	CORBA ネーミングサービスを前提としているため、サポートされません。
キャッシング機能	×	CORBA ネーミングサービスで利用する機能のため、サポートされません。
ユーザ指定名前空間管理機能		サポートされます。
ラウンドロビン検索	×	CORBA ネーミングサービスを前提としているため、サポートされません。

(凡例)

: サポートされます。

× : サポートされません。

(b) EJB 機能のサポート範囲

EJB 機能では、RMI-IIOP 通信やそのほかの CORBA 機能が使用できないため、リモート通信や CTM 連携が使用できません。

EJB 機能のサポート範囲を示します。

表 C-2 Developer Standard での EJB 機能のサポート範囲

EJB 機能			サポート	備考
Enterprise Bean	Session Bean	Stateless		リモートインタフェースの remove 処理に制限があります。「(4)(b) Stateless Session Bean の EJBObject に対する remove メソッドの制限」を参照してください。
		Stateful		-
	Entity Bean	BMP		-
		CMP		-
	Message Driven Bean		×	-
インタフェース	リモートインタフェース			プロセス間の呼び出しはできません。
	ローカルインタフェース			-
	ビジネスインタフェース	リモート		プロセス間の呼び出しはできません。
		ローカル		-

EJB 機能		サポート	備考
EJB3.0 機能	Timer Service 機能		-
	DI 機能		org.omg.CORBA.ORB , および org.omg.CORBA_2_3.ORB のタイプについては使用できません。
	その他機能 (interceptor 機能 など)		-
ダイナミッククラスローディング機能		×	-
トランザクション実行機能	BMT		-
	CMT		-
	ローカルトランザクション		-
	グローバルトランザクション	×	-
	トランザクションタイムアウト機能		-
javax.ejb.spi.HandleDelegate , および javax.ejb.HomeHandle , javax.ejb.Handle の取得		×	-
リソース利用機能	データベース		-
	Mail		-
	JMS		JMS 機能に対応している MessageDrivenBean は使用 できません。
	リソースアダプタ		-
	JavaBeans		-
ネーミングサービス利用機能	Enterprise Bean の環境エントリの取得		-
	EJB ホームのオブジェクトリファレンス取得		オブジェクトリファレンスではなく、オブジェクト自体を取得します。
	コネクションファクトリの取得		-
	リソース環境リファレンスの取得		-
	UserTransaction の取得		-
	ORB インスタンスの取得	×	-

EJB 機能		サポート	備考
RMI-IIOP 通信拡張機能	ポート固定 / IIOP アドレス固定機能	×	-
	RMI-IIOP 通信タイムアウト	×	-
	EJB ホームオブジェクトへの再接続機能	×	-
ローカル呼び出し最適化機能	none	×	-
	app		-
	all		-
負荷分散機能 (CTM)		×	-
PRF トレース出力機能			OTS, CTM 部分のトレースは出力されません。

(凡例)

- : サポートされます。
- ×
- × : サポートされません。
- △ : Developer Professional が提供するテスト用実行環境と異なります。

Developer Standard を使用している場合に、ビジネスメソッドで呼び出せる範囲を次に示します。

表 C-3 ビジネスメソッドで呼び出せる範囲

インタフェース	呼び出し範囲		
	J2EE アプリケーション内	J2EE アプリケーション間	J2EE サーバ間
リモートインタフェース			×
ローカルインタフェース		×	×

(凡例)

- : 呼び出せます。
- ×
- × : 呼び出せません。

注

ejbserver.rmi.localinvocation.scope プロパティに all を指定した場合に呼び出せます。

(4) J2EE アプリケーションの実装および設定の注意事項

Developer Standard を使用して J2EE アプリケーションを実装する場合、およびプロパティを設定する場合の注意事項を示します。

(a) 実装で発生する例外・プロパティ設定で発生するエラー

Developer Standard を使用して実装するときに、RMI-IIOP 通信やその他の CORBA 機能を使用すると、実行時に例外が発生します。また、プロパティで CTM 連携の設定を

するとエラーとなります。

Developer Standard を使用して実装する場合に例外となる処理，およびプロパティを設定する場合にエラーとなる処理について次の表に示します。

表 C-4 実装およびプロパティ設定で問題となる処理

問題となる処理	実行結果
CORBA ネーミングサービスを使用して ORB インスタンスを取得する。	javax.naming.NameNotFoundException 例外が発生します。
CORBA ネーミングサービスの切り替え機能を使用する。	java.lang.IllegalStateException 例外が発生します。
java.naming.factory.initial プロパティをキーとして，値に ORB が必要なクラスを生成するファクトリ (com.hitachi.software.ejb.jndi.InsContextFactory など) を設定した Hashtable クラスを引数にして InitialContext を生成する。	javax.naming.ConfigurationException 例外が発生します。
通信タイムアウト機能の API (RequestTimeoutConfigFactory クラスおよび RequestTimeoutConfig クラスのメソッド) を使用する。	java.lang.IllegalStateException 例外が発生します。
J2EE アプリケーションに対して，CTM 連携をするようにプロパティを設定する。	J2EE アプリケーションの起動に失敗します。Developer Professional でも J2EE アプリケーションの起動に失敗しますが，出力するメッセージが異なります。 Developer Professional : KDJE43087-E Developer Standard : KDJE43108-E
javax.ejb.EJBObject の getHandle メソッドおよび javax.ejb.EJBHome の getHomeHandle メソッドを呼び出す。	java.lang.IllegalStateException 例外が発生します。
javax.ejb.EJBHome の remove(Handle) メソッドを呼び出す。	java.rmi.RemoteException 例外が発生します。
JNDI の lookup メソッドを使用して javax.ejb.spi.HandleDelegate を取得する。	javax.naming.NameNotFoundException 例外が発生します。

(b) Stateless Session Bean の EJBObject に対する remove メソッドの制限

Developer Standard の Stateless Session Bean では，remove メソッドで EJB オブジェクトインスタンスのビジネスメソッドの呼び出しを抑止することはできません。remove メソッドを呼び出した EJB オブジェクトのインスタンスに対してビジネスメソッドを呼び出した場合，次に示す動作をします。

Developer Standard

ビジネスメソッドが呼び出されます。

Developer Professional

java.rmi.NoSuchObjectException 例外が発生します。

remove 後にビジネスメソッドを呼び出した場合の動作は保証されません。

(c) EJB ホームおよび EJB オブジェクトのインスタンスに対するシリアル化処理の制限

Developer Standard では、EJB ホームおよび EJB オブジェクトのインスタンスはシリアル化に対応しません。そのため、次に示すインスタンスを create メソッドの引数、ホームメソッド、ビジネスメソッドの引数、および戻り値に指定した場合、メソッドを呼び出したときに java.io.NotSerializableException 例外が発生します。

- EJB ホームおよび EJB オブジェクトを要素に持つコレクションクラスのインスタンス
- EJB ホームおよび EJB オブジェクトをメンバ変数に持ち、かつシリアル化に対応したクラスのインスタンス

これらの条件に当てはまるインスタンスを create メソッドの引数、ホームメソッド、ビジネスメソッドの引数、および戻り値に指定したい場合は、そのメソッドを持つ Bean に対して、参照渡し (pass by reference) を設定します。

(5) プロパティのサポート範囲

Developer Standard でのアプリケーションサーバの設定で、簡易構築定義ファイルまたは構成変更定義ファイルに指定できないパラメータを示します。パラメータの概要および指定値については、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

表 C-5 Developer Standard で指定できない簡易構築定義ファイルのパラメータ一覧

パラメータ名の先頭	パラメータ名
ejbserver.client	ejbserver.client.ctm.RequestPriority
ejbserver.distributedtx	ejbserver.distributedtx.enableXidReuseOptimization ejbserver.distributedtx.ots.status.directory1 ejbserver.distributedtx.ots.status.directory2 ejbserver.distributedtx.recovery.port ejbserver.distributedtx.XATransaction.enabled
ejbserver.DynamicStubLoading	ejbserver.DynamicStubLoading.Enabled
ejbserver.jdbc	ejbserver.jdbc.propertyInfo.BUF_SIZE ejbserver.jdbc.propertyInfo.ENCODELANG
ejbserver.jndi	ejbserver.jndi.cache ejbserver.jndi.cache.interval ejbserver.jndi.cache.interval.clear.option ejbserver.jndi.request.timeout
webserver.sfo	webserver.sfo.sfo_servers

パラメタ名の先頭	パラメタ名
java.naming	java.naming.factory.initial
vbj.java2iio	vbj.java2iio.jvm.maxHeapSize vbj.java2iio.jvm.minHeapSize
vbroker.se	vbroker.se.iio_tp.scm.iio_tp.listener.port

表 C-6 Developer Standard で指定できない usrconf.properties (J2EE サーバ用ユーザプロパティファイル) のプロパティ一覧

プロパティ名の先頭	プロパティ名
ejbserver.deploy	ejbserver.deploy.stub.generation.scope
ejbserver.distributedtx	ejbserver.distributedtx.recovery.completionCheckOnStopping.timeout
ejbserver.jndi	ejbserver.jndi.namingservice.group.<グループ名>.providerurls ejbserver.jndi.namingservice.group.list
ejbserver.jta	ejbserver.jta.TransactionManager.enlist.compatibleMode.enabled
ejbserver.naming	ejbserver.naming.exec.args ejbserver.naming.host ejbserver.naming.nameroot ejbserver.naming.port ejbserver.naming.protocol ejbserver.naming.startupMode ejbserver.naming.startupRetryCount ejbserver.naming.startupWaitTime
webserver.sfo	webserver.sfo.sfo_server.<SFO サーバ名>.connectionTimeout webserver.sfo.sfo_server.<SFO サーバ名>.naming
vbroker.agent	vbroker.agent.port
vbroker.se	vbroker.se.iio_tp.host

表 C-7 Developer Standard で指定できない usrconf.properties (サーバ管理コマンド用システムプロパティファイル) のプロパティ一覧

プロパティ名の先頭	プロパティ名
ejbserver.cui	ejbserver.cui.exitcode.compatible ejbserver.cui.optionalname.enabled
ejbserver.naming	ejbserver.naming.host ejbserver.naming.port ejbserver.naming.protocol

表 C-8 Developer Standard で指定できない usrconf.properties (EJB クライアントアプリケーション用ユーザプロパティファイル) のプロパティ一覧

プロパティ名の先頭	プロパティ名
ejbserver.client	ejbserver.client.transaction.clientName ejbserver.client.transaction.enabled
ejbserver.jndi	ejbserver.jndi.namingservice.group.<グループ名>.providerurls ejbserver.jndi.namingservice.group.list

表 C-9 Developer Standard で指定できない EJB クライアントアプリケーションのシステムプロパティ一覧

プロパティ名の先頭	プロパティ名
ejbserver.container	ejbserver.container.rebindpolicy
java.endorsed	java.endorsed.dirs
java.naming	java.naming.factory.object java.naming.factory.state java.naming.provider.url
javax.rmi	javax.rmi.CORBA.PortableRemoteObjectClass javax.rmi.CORBA.StubClass javax.rmi.CORBA.UtilClass
org.omg	org.omg.CORBA.ORBClass org.omg.CORBA.ORBSingletonClass org.omg.PortableInterceptor.ORBInitializerClass.com.hitachi.software.ejb.security.std.interceptor.SecurityClientInterceptorInit
vbroker.agent	vbroker.agent.enableLocator
vbroker.orb	vbroker.orb.htc.comt.entryCount vbroker.orb.htc.comt.fileCount vbroker.orb.htc.tracePath

(6) MyEclipse の機能差

Developer Standard で提供する MyEclipse と Developer Professional で提供する MyEclipse では、機能差があります。

MyEclipse の機能差を次の表に示します。

表 C-10 MyEclipse の機能差

Developer Professional で提供する MyEclipse の機能	Developer Standard での使用
JavaScript デバッガ	×
Matisse4MyEclipse	×
MyEclipse UML	×

Developer Professional で提供する MyEclipse の機能	Developer Standard での使用
拡張データベースサポート • Oracle コネクタ • SQL Server	×
Java イメージエディタ	×
Visual Web デザイナ	
XML エディタ	
OR ツール Hibernate サポート	
MyEclipse Spring ツール	
データベースエクスプローラ	
Web サービス	
Struts デザイナ	
JSF デザイナ	
Cosminexus コネクタ	
JSP 開発	
EJB サポート (2.0)	
Sync-on-Demand 開発	
XDoclet サポート	
CSS エディタ	
JSTL サポート	
ウィザード	
カスタマイズ可能なテンプレート	
ワークベンチ構成	

(凡例)

：使用できます。

×：使用できません。

Developer Standard の MyEclipse で使用できない機能を使用しようとすると、エラーダイアログが出力されます。

(7) サーバ管理コマンドと J2EE サーバの通信に使用するポートの固定方法の機能差

Developer Standard と Developer Professional では、サーバ管理コマンドが J2EE サーバと通信するためのポートを固定する方法に機能差があります。Developer Standard および Developer Professional の機能差を次の表に示します。

表 C-11 サーバ管理コマンド / J2EE サーバ間の通信で使用するポート番号を固定するためのパラメタおよびプロパティ

Developer の Edition	ポート番号を固定するパラメタおよびプロパティ	
	ファイル名	プロパティ名・パラメタ名
Developer Standard	簡易構築定義ファイル	ejbserver.rmi.remote.listener.port パラメタ
	usrconf.properties (サーバ管理コマンド用システム プロパティファイル)	ejbserver.rmi.remote.listener.port プロパティ
Developer Professional	簡易構築定義ファイル	vbroker.se.iiop_tp.scm.iiop_tp.listener.port パラメタ
	usrconf.properties (サーバ管理コマンド用システム プロパティファイル)	vbroker.se.iiop_tp.scm.iiop_tp.listener.port プロパティ

なお、Developer Standard で指定できる値は 0 ~ 65535 の整数です。すでにほかのアプリケーションで使用されているポート番号は指定できません。パラメタまたはプロパティを設定しない場合、または値に 0 を指定した場合は、ランダムな値が設定されます。

(8) インスタントセットアップ機能使用時の注意事項と機能差

Developer Standard でインスタントセットアップ機能を使用する場合の注意事項と機能差について説明します。

(a) Developer Standard インストール時の注意事項

Developer Standard では、Developer Professional または Service Architect に上書きインストールしたり、アンインストールした Developer Professional または Service Architect と同じディレクトリにインストールしたりすると、インスタントセットアップ機能を実行できません。

Developer Professional または Service Architect がインストールされている環境に Developer Standard をインストールする場合は、必ず Cosminexus Developer をアンインストールしてからインストールしてください。また、インストール先ディレクトリには、アンインストールした Developer Professional または Service Architect が使用していたディレクトリとは異なるディレクトリを指定してください。

(b) Developer Professional との機能差

Developer Standard と Developer Professional では、インスタントセットアップ機能で設定できる J2EE サーバのポート番号に違いがあります。

Developer Standard では、J2EE サーバのネーミングサービスのポート番号を使用しないため、設定できません。

付録 C.3 Developer Standard での J2EE アプリケーションのインポート手順

Developer Standard で開発した J2EE アプリケーションのインポートについて、次の制限があります。

Developer Standard の J2EE サーバ（テスト用実行環境）にインポートされた実行時情報付き EAR ファイルをエクスポートし、そのまま Developer Standard 以外の J2EE サーバ（実行環境）にインポートすることはできません。

Developer Standard 以外の J2EE サーバ（実行環境）にインポートされた実行時情報付き EAR ファイルをエクスポートし、そのまま Developer Standard の J2EE サーバ（テスト用実行環境）にインポートすることはできません。

ここでは、上記の制限を回避し、Developer Standard の J2EE サーバから Developer Standard 以外の J2EE サーバへ実行時情報付き EAR ファイルをインポートする手順を説明します。また、Developer Standard 以外の J2EE サーバから Developer Standard の J2EE サーバへ実行時情報付き EAR ファイルをインポートする手順を示します。

（１）Developer Standard の J2EE サーバから Developer Standard 以外の J2EE サーバへインポートする手順

Developer Standard で開発した J2EE アプリケーションの実行時情報を、Developer Standard 以外の環境で使用するには、次の手順でインポートします。

1. Developer Standard の J2EE サーバから実行時情報なしの EAR ファイルをエクスポートします。
2. Developer Standard の J2EE サーバから J2EE アプリケーションの属性ファイルを取得します（サーバ管理コマンドを使用）。
3. 実行時情報なしの EAR ファイルをインポート先（Developer Standard 以外）の J2EE サーバにインポートします。
4. J2EE アプリケーションの属性ファイルをインポート先のマシンに転送します。
5. サーバ管理コマンドを使用して、インポート先のマシンに属性ファイルを設定します。

（２）Developer Standard 以外の J2EE サーバから Developer Standard の J2EE サーバへインポートする手順

Developer Standard 以外の環境にインポートされた J2EE アプリケーションの実行時情報を、Developer Standard の環境で使用するには、次の手順でインポートします。

1. Developer Standard 以外の J2EE サーバから実行時情報なしの EAR ファイルをエクスポートします。

2. Developer Standard 以外の J2EE サーバから J2EE アプリケーションの属性ファイルを取得します (サーバ管理コマンドを使用)。
3. 実行時情報なしの EAR ファイルをインポート先 (Developer Standard) の J2EE サーバにインポートします。
4. J2EE アプリケーションの属性ファイルをインポート先のマシンに転送します。
5. サーバ管理コマンドを使用して、インポート先のマシンに属性ファイルを設定します。

付録 D Matisse4MyEclipse フィーチャのインストール

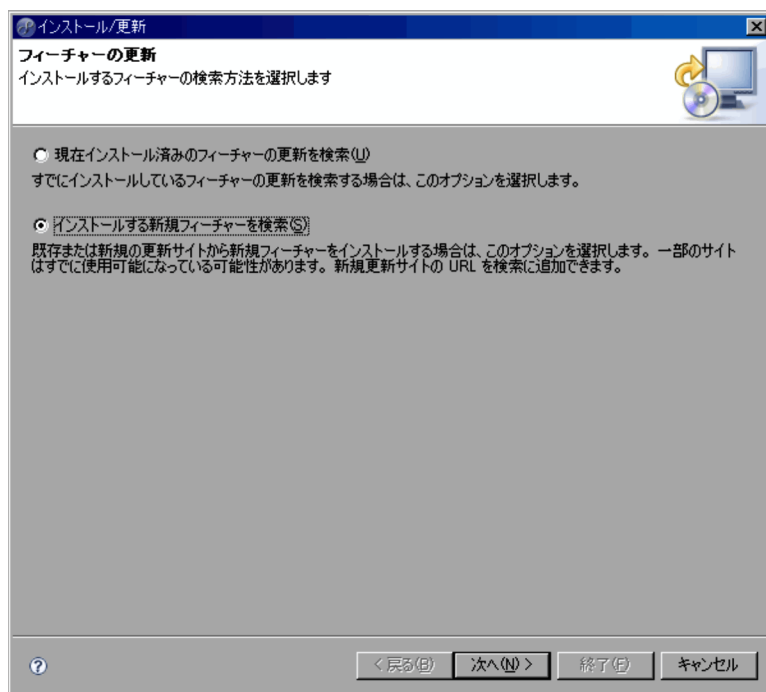
Matisse4MyEclipse フィーチャをインストールすると、MyEclipse で Java Swing クラ イアントアプリケーションを開発できます。

ここでは、Matisse4MyEclipse のインストール手順およびインストールの確認方法につ いて説明します。

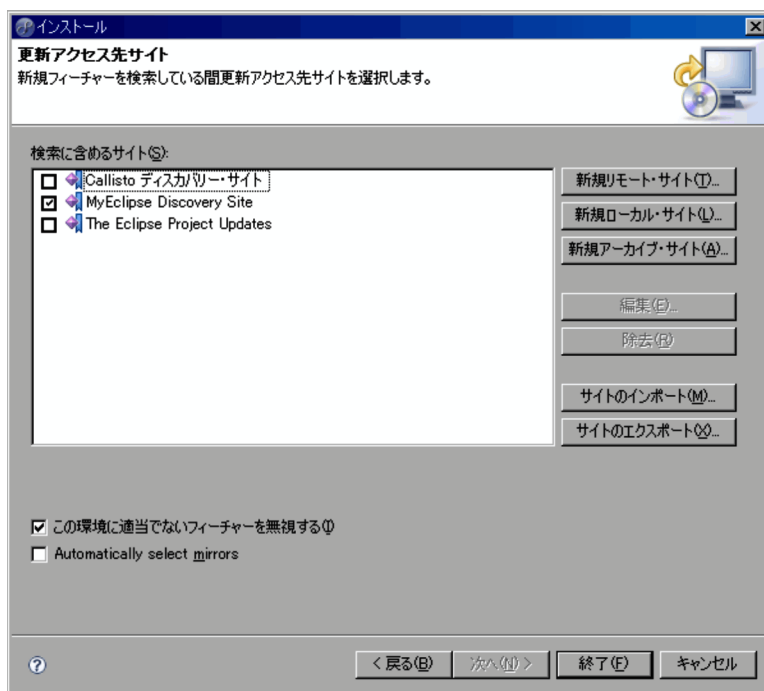
付録 D.1 Matisse4MyEclipse フィーチャのインストール手順

Eclipse の更新マネージャを使用して、Matisse4MyEclipse をインストールします。次の 手順でインストールしてください。

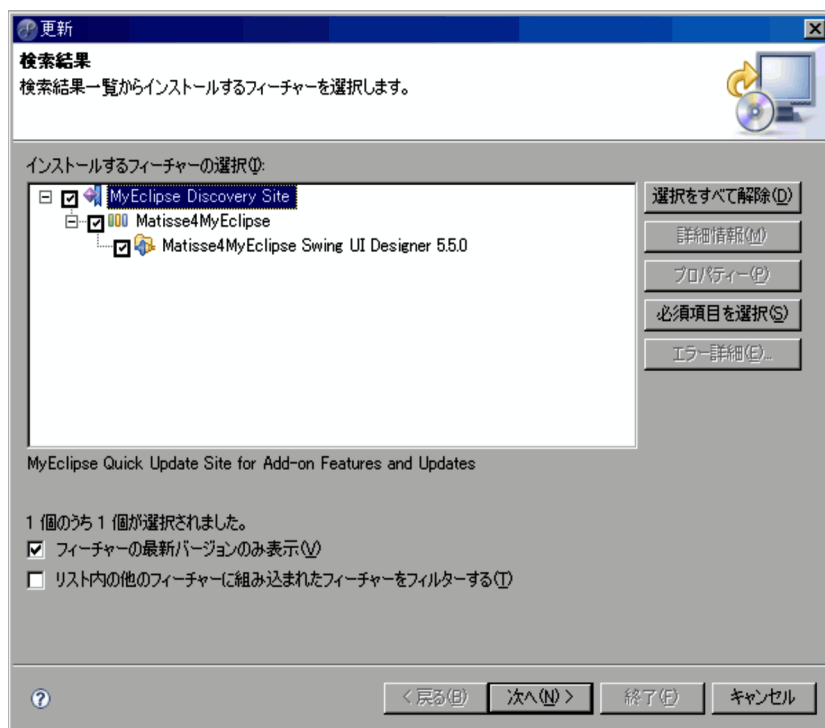
1. Eclipse のメニューから [ヘルプ] - [ソフトウェア更新] - [検索およびインス トール] を選択します。
[インストール / 更新] ダイアログの [フィーチャーの更新] ページが表示されます。



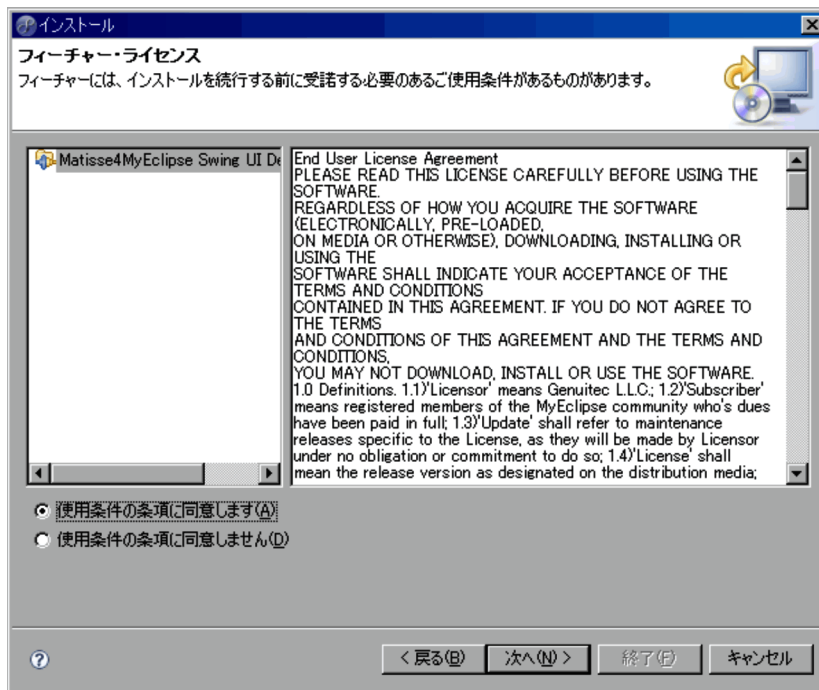
2. [フィーチャーの更新] ページで、[インストールする新規フィーチャーを検索] を選 択して、[次へ] ボタンをクリックします。
ダイアログ名が [インストール] に変わり、[更新アクセス先サイト] ページが表示 されます。



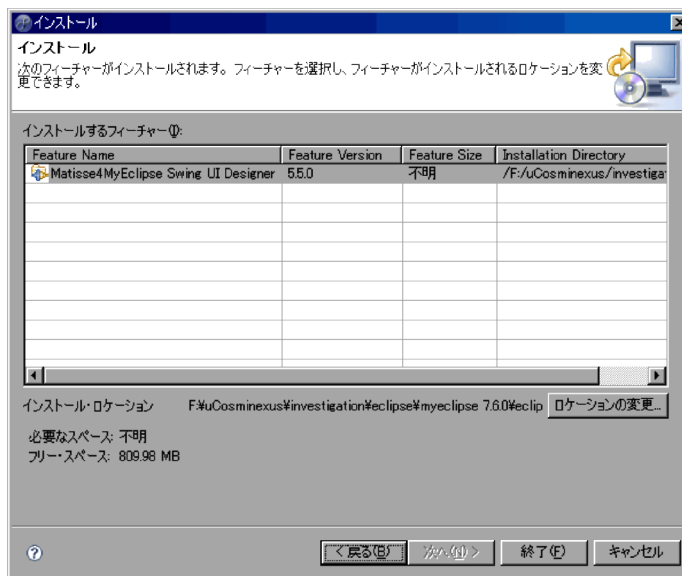
3. [検索に含めるサイト] リストにある [MyEclipse Discovery Site] をチェックして ,
[終了] ボタンをクリックします。
ダイアログ名が [更新] に変わり , [検索結果] ページが表示されます。



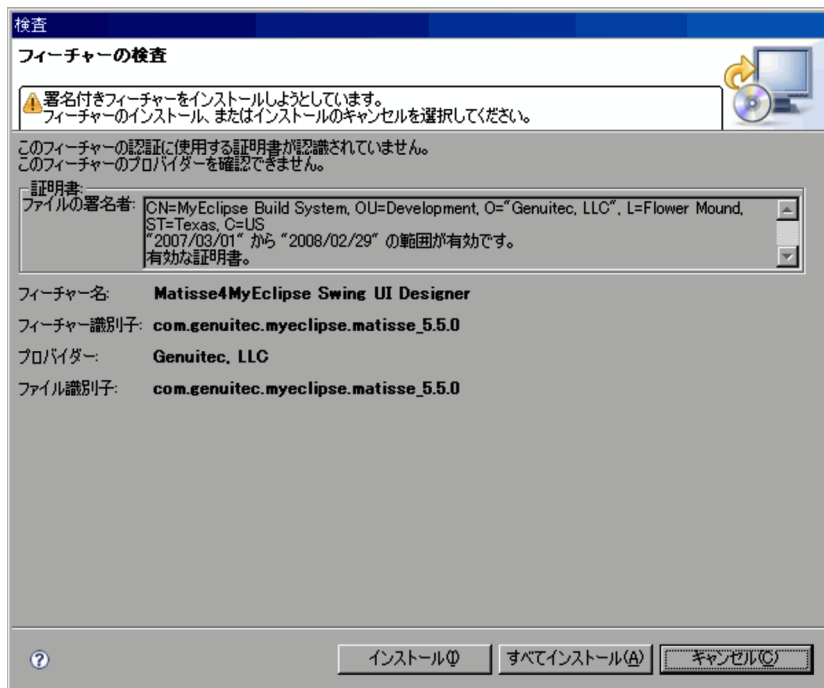
4. [検索結果] ページで , [インストールするフィーチャーの選択] リストにある [Matisse4MyEclipse] をチェックして , [次へ] ボタンをクリックします。
ダイアログ名が [インストール] に変わり , [フィーチャー ・ ライセンス] ページが表示されます。



5. [フィーチャー・ライセンス] ページで, [使用条件の条項に同意します] を選択して, [次へ] ボタンをクリックします。
[インストール] ページが表示されます。
[ロケーションの変更] ボタンをクリックして, インストール先を MyEclipse インストール済み環境または別の任意の場所に変更してください。



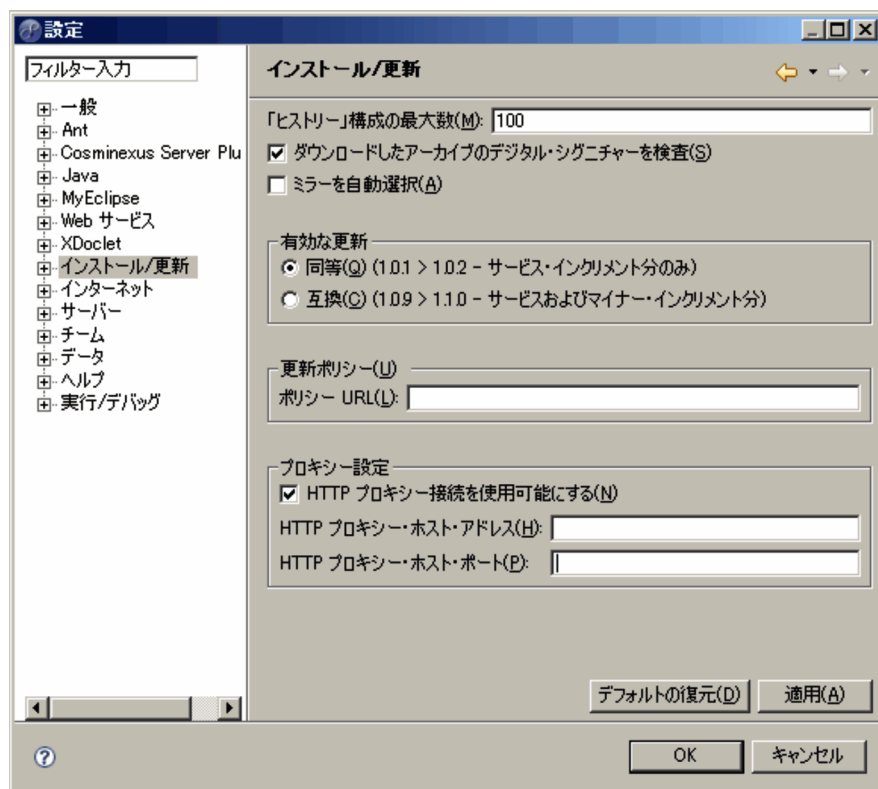
6. [インストール] ページの [終了] ボタンをクリックします。
ダイアログ名が [更新] に変わり , [フィーチャーの検査] ページが表示されます。



7. [フィーチャーの検査] ページで , [すべてインストール] ボタンをクリックします。
ダイアログ名が [インストール / 更新] に変わります。
8. [インストール / 更新] ダイアログで [はい] ボタンをクリックします。
Matisse4MyEclipse フィーチャがインストールされ , Eclipse が再起動されます。

! 注意事項

プロキシ環境下で Matisse4MyEclipse フィーチャをインストールする場合は , [設定] ダイアログの [インストール / 更新] ページでプロキシサーバを設定してください。



付録 D.2 Matisse4MyEclipse フィーチャのインストール確認

MyEclipse に Matisse4MyEclipse がインストールされていることを確認します。
Matisse4MyEclipse がインストールされているかどうかは、[MyEclipse Enterprise Workbench について] ダイアログで確認します。次の手順で確認してください。

1. Eclipse のメニューから [ヘルプ] - [MyEclipse Enterprise Workbench について] を選択します。
[MyEclipse Enterprise Workbench について] ダイアログが表示されます。
2. [MyEclipse Enterprise Workbench について] ダイアログで、MyEclipse アイコンをクリックします。



[MyEclipse Enterprise Workbench フィーチャについて] ダイアログが表示されます。

[MyEclipse Enterprise Workbench フィーチャについて] ダイアログのリストの中に「MyEclipse Visual Swing Designer Feature」が表示されていることを確認してください。「MyEclipse Visual Swing Designer Feature」が表示されていれば、Matisse4MyEclipse フィーチャはインストールされています。

付録 E テスト用実行環境のシステムのチューニング

構築したテスト用のシステムのパラメタをチューニングする場合は、運用管理ポータルという Management Server を操作するための GUI を使用できます。運用管理ポータルを使用するためには、Management Server のセットアップ、および Management Server および運用管理エージェントの起動が必要です。

ただし、これらの設定は Management Server 設定時に実施済みのため、ここで特に設定する必要はありません。

運用管理ポータルへのログイン

運用管理ポータルへは、Web ブラウザからログインします。Management Server を起動しているホスト上、または Management Server を起動しているホストとネットワークで接続されているホスト上で Web ブラウザを起動して、URL に「http://<ホスト名>:<ポート番号>/mnngsvr/」を指定します。インストール初期状態での URL は「http://localhost:28080/mnngsvr/」です。

運用管理ポータルを起動すると、ログイン画面が表示されます。ログイン画面で入力する管理ユーザ ID とパスワードには、次の値を入力してください。

- 管理ユーザ ID

インスタントセットアップ機能で環境構築をした場合は、「admim」を入力してください。インスタントセットアップ機能で環境構築をしていない場合は、Management Server 設定時に指定した「管理ユーザ ID」を入力します。

- パスワード

インスタントセットアップ機能で環境構築をした場合は、「admim」を入力してください。Management Server を設定するときに指定したインスタントセットアップ機能で環境構築をしていない場合は、Management Server 設定時に指定した「管理ユーザパスワード」を入力します。

[ログイン] ボタンをクリックすると、運用管理ポータル画面が表示されます。

運用管理ポータルでの設定変更

運用管理ポータルにログインすると、[運用管理ポータル] 画面が表示されます。システムのプロパティは、[運用管理ポータル] 画面の [論理サーバの環境設定] アンカーをクリックして表示される「論理サーバの環境設定」の各画面で変更します。論理サーバごとに設定する画面が異なります。

なお、運用管理ポータルの画面と各画面で設定できる内容については、マニュアル「Cosminexus 運用管理操作ガイド」を参照してください。

付録 F Cosminexus Version 6 から Cosminexus Version 7 への移行

Cosminexus Version 6 から Cosminexus Version 7 にバージョンアップする場合、アプリケーションサーバの移行およびソースプログラムの移行が必要になります。ここでは、アプリケーションサーバの移行およびソースプログラムの移行について説明します。

付録 F.1 アプリケーションサーバの移行

Cosminexus Version 6 で設定した開発環境は、Cosminexus Version 7 へ移行できます。Cosminexus Version 7 では、Smart Composer 機能を使用して開発環境を構築します。ここでは、Cosminexus Version 6 から Cosminexus Version 7 へのアップグレードインストールではなく、新規インストールで Cosminexus Version 7 に移行する手順を説明します。

(1) Cosminexus Version 6 と Cosminexus Version 7 の違い

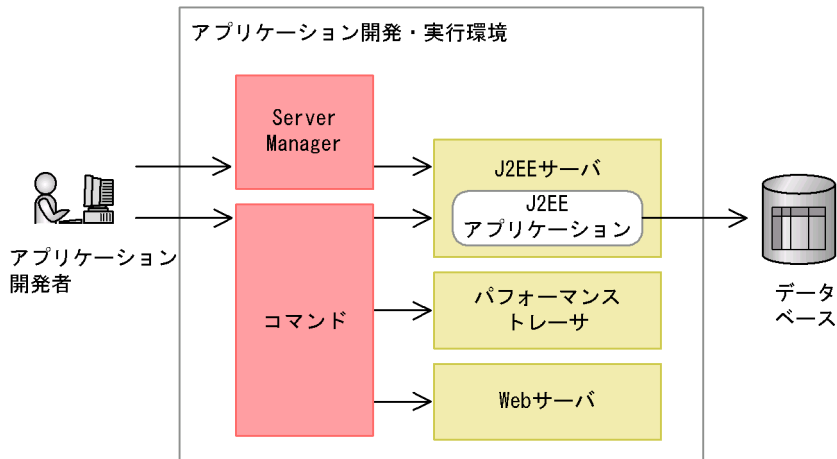
Cosminexus Version 7 では、Cosminexus Version 6 で実行していたコマンドの操作や定義ファイルの設定を Eclipse のプラグインおよび MyEclipse で実行できます。ここでは、Cosminexus Version 6 と Cosminexus Version 7 のアプリケーション開発環境の相違点を説明します。

(a) 開発・実行環境の違い

Cosminexus Version 6 の環境では、Server Manager とコマンドが必要なものに対して、Cosminexus Version 7 の環境では、Eclipse のプラグインおよび MyEclipse の機能を利用して操作できます。

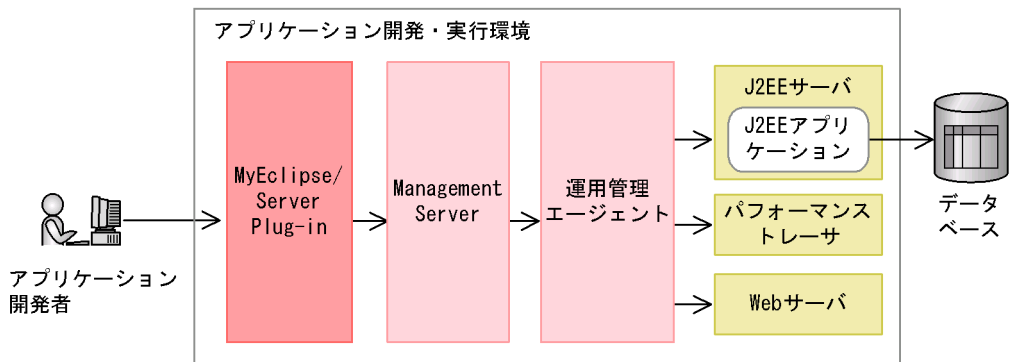
Cosminexus Version 6 のアプリケーション開発環境および実行環境を、次の図に示します。

図 F-1 Cosminexus Version 6 のアプリケーション開発・実行環境



Cosminexus Version 7 のアプリケーション開発環境および実行環境を、次の図に示します。

図 F-2 Cosminexus Version 7 のアプリケーション開発・実行環境



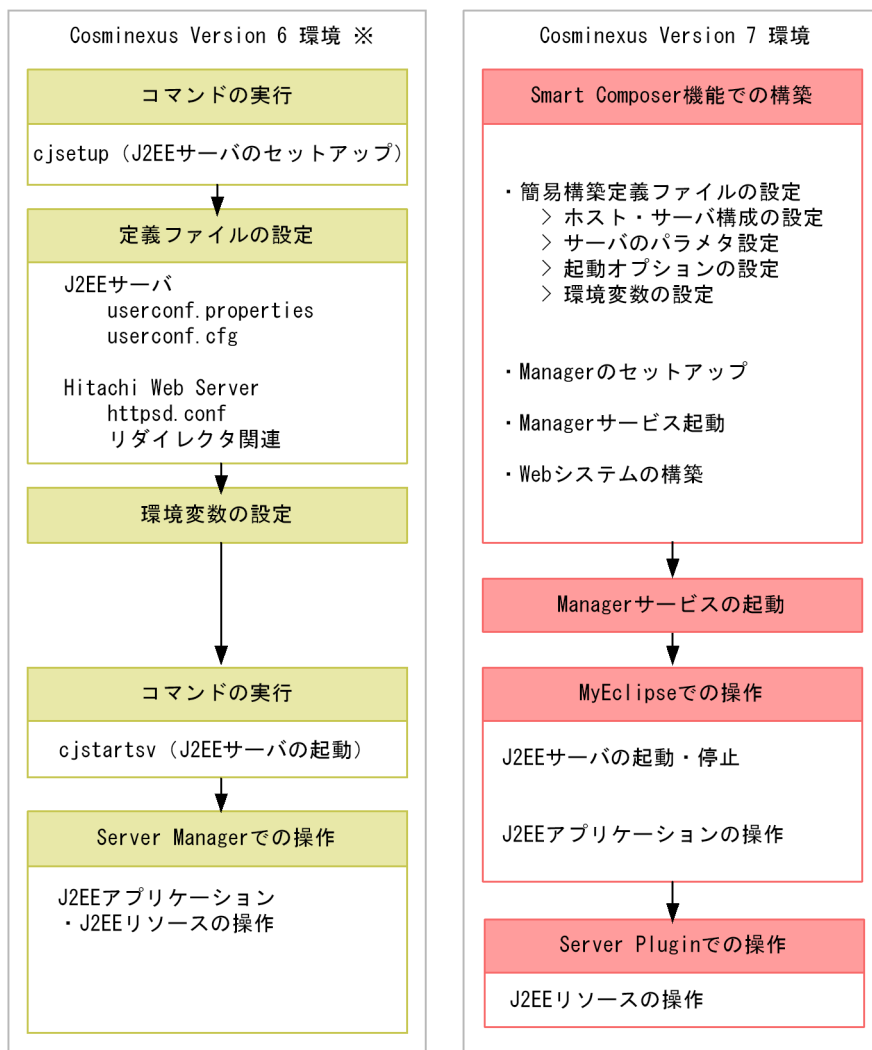
Cosminexus Version 6 では、Server Manager とコマンドを使用して、パフォーマンストレーサ、J2EE サーバ、Web サーバを操作します。このとき、コマンドが Web サーバ内にある論理サーバと J2EE アプリケーションを直接操作します。

一方、Cosminexus Version 7 では Management Server を使用して、MyEclipse からは J2EE サーバを、Server Plug-in からはパフォーマンストレーサ、J2EE サーバ、Web サーバを操作できます。

(b) 開発環境の構築・運用手順の違い

次の図は、Cosminexus Version 6 と Cosminexus Version 7 の開発環境の構築手順と運用手順について示します。

図 F-3 Cosminexus Version 6 と Cosminexus Version 7 の開発環境の構築・運用手順



注※ Management Serverは使用しないものとしています。

Cosminexus Version 6 のアプリケーション開発環境は、cjsetup コマンドを実行したり、定義ファイルを直接編集したりして J2EE サーバを構築します。また、J2EE アプリケーション開発環境の J2EE サーバを開始したり、J2EE アプリケーションや J2EE リソースを操作したりするには、cjstartsv コマンドおよび Server Manager を使用します。

Cosminexus Version 7 の J2EE アプリケーション開発環境を構築する場合は、Smart Composer 機能を使用します。また、J2EE アプリケーション開発環境の J2EE サーバを開始したり、J2EE アプリケーションや J2EE リソースを操作したりするには、MyEclipse や Server Plug-in を使用します。なお、Smart Composer 機能を使用して開

発環境を構築したり、MyEclipse や Server Plug-in を使用した J2EE アプリケーション開発をしたりする場合は、Management Server が必要になります。

構築手順で設定する定義ファイルやコマンドについては、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

(2) 開発環境の移行手順

Cosminexus Version 6 環境のアプリケーション開発環境を Cosminexus Version 7 へ移行する手順を説明します。

(a) Cosminexus Version 6 の開発環境の退避

Cosminexus Version 7 をインストールする前に、Cosminexus Version 6 の環境を退避させる必要があります。次に示す属性ファイルおよび定義ファイルを退避させてください。

J2EE アプリケーションとリソースアダプタの属性の退避

J2EE サーバにデプロイ済みの J2EE アプリケーションやリソースアダプタは、それぞれの属性ファイルを退避させる必要があります。次のコマンドを使用して、属性ファイルを取得してください。なお、コマンドの使用方法については、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

- J2EE アプリケーションの場合

```
cjgetappprop
```

- リソースアダプタの場合

```
cjgetrarprop
```

また、デプロイ済みの J2EE アプリケーションを、属性が含まれた状態の EAR ファイルでエクスポートする場合は、Server Manager または cjexportapp コマンドを使用します。Server Manager で EAR ファイルをエクスポートする方法については、マニュアル「Cosminexus Version 6 アプリケーション設定操作ガイド」を参照してください。

定義ファイルの追加パラメタ情報の退避

Cosminexus Version 6 の環境で、J2EE サーバおよび Web サーバの定義ファイルにパラメタの追加および変更をしている場合は、その定義ファイルまたは変更内容を退避させる必要があります。次の表に示す定義ファイルに、パラメタの追加および変更がある場合は、その定義ファイルを退避させてください。なお、デフォルト値を使用しているパラメタについては、退避させる必要はありません。

表 F-1 退避対象となる定義ファイルの一覧

対象論理サーバ	ファイル名	説明	Cosminexus Version 6 でのファイルの格納先
J2EE サーバ	usrconf.cfg	J2EE サーバ用オプション定義ファイル	<Cosminexus のインストールディレクトリ>¥CC¥server¥usrconf¥ejb¥<J2EE サーバ名>¥
	usrconf.properties	J2EE サーバ用ユーザプロパティファイル	<Cosminexus のインストールディレクトリ>¥CC¥server¥usrconf¥ejb¥<J2EE サーバ名>¥
Web サーバ	httpsd.conf	Hitachi Web Server 定義ファイル	<Cosminexus のインストールディレクトリ>¥httpsd¥conf¥
	mod_jk.conf	Hitachi Web Server 用リダイレクタ動作定義ファイル	<Cosminexus のインストールディレクトリ>¥CC¥web¥redirector¥
	workers.properties	ワーカー定義ファイル	<Cosminexus のインストールディレクトリ>¥CC¥web¥redirector¥

注

httpsd.conf でファイルパスを指定している場合は、その指定先のファイルになります。

(b) Cosminexus Version 7 のインストール

新しい環境に Cosminexus Version 7 をインストールします。なお、インストールはアップグレードインストールではなく、新規にインストールしてください。インストール方法については、「2. インストールとセットアップ」を参照してください。

簡易構築定義ファイルの準備

Smart Composer 機能を使用して環境構築するために、簡易構築定義ファイルを準備します。ここでは、簡易構築定義ファイルのデフォルトの構成と、カスタマイズについて説明します。

簡易構築定義ファイルのデフォルトの構成

簡易構築定義ファイルのデフォルトの構成は、次のようになっています。ここでは、簡易構築定義ファイル名を「MyWebSystem.xml」、Web システム名を「MyWebSystem」とします。

```

<?xml version="1.0" encoding="UTF-8" ?>
<model-definition xmlns="http://www.cosminexus.com/mngsvr/schema/
ModelDefinition-2.5">
  <!-- Webシステムの属性定義 -->
  <web-system>
    <name>MyWebSystem 1</name>
    <!-- 物理ティアの定義 -->
    <tier>
      <tier-type>combined-tier</tier-type>
    </tier>
  <!-- サービスユニットの定義 -->
  <unit>
    <name>unit1</name>
    <allocated-host>
      <host-ref>MyHost 2</host-ref>
      <hosts-for>combined-tier</hosts-for>
      <define-server>
        <logical-server-name>MyJ2EE 3</logical-server-name>
        <logical-server-type>j2ee-server</logical-server-type>
        <configuration>
          <param>
            <param-name>additional.startcmd 4</param-name>
            <param-value>-nosecurity 4</param-value>
          </param>
        </configuration>
      </define-server>
      <define-server>
        <logical-server-name>MyPRF 5</logical-server-name>
        <logical-server-type>performance-tracer</logical-server-type>
      </define-server>
      <define-server>
        <logical-server-name>MyHWS 6</logical-server-name>
        <logical-server-type>web-server</logical-server-type>
      </define-server>
    </allocated-host>
  </unit>
</web-system>
  <!-- ホストの定義 -->
  <host>
    <host-name>MyHost 2</host-name>
    <agent-host>192.168.1.10 7</agent-host>
  </host>
</model-definition>

```

注 1

J2EE アプリケーションの実行環境を識別する Web システム名を任意で指定します。
1 ホストで複数のアプリケーション実行環境を作成する場合は、それぞれ異なる Web システム名を指定してください。

注 2

J2EE アプリケーションの実行環境ホストのホスト名を任意で指定します。
サービスユニットの定義、ホストの定義で同じホスト名を指定してください。

注 3

J2EE アプリケーションを実行する J2EE サーバ名を任意で指定します。

J2EE サーバ名は、Cosminexus Version 6 でセットアップしたときのサーバ名を使用してください。

注 4

J2EE サーバを起動するときのコマンドオプションを指定します。
このサンプルファイルでは、セキュリティマネージャを使用しないで起動する場合のオプションを指定しています。

注 5

パフォーマンストレーサのサーバ名を任意で指定します。

注 6

Web サーバ名を任意で指定します。

注 7

J2EE アプリケーションの実行環境のホストの IP アドレスを指定します。
ホスト名は指定できません。

簡易構築定義ファイルのカスタマイズ

Cosminexus Version 6 の開発環境の設定を Cosminexus Version 7 の開発環境に移行する場合は、簡易構築定義ファイルにある各サーバの <configuration> タグの定義を使って、次の設定を追加パラメタとして設定します。

- J2EE サーバおよび Web サーバの定義ファイルに追加および変更したパラメタ
- J2EE サーバおよび Web サーバに設定した起動オプションや環境変数

なお、パラメタの指定方法および簡易構築定義ファイルに指定できるパラメタについては、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

定義ファイルとパラメタの設定例を次に示します。

J2EE サーバの設定ファイルに追加したパラメタを指定する場合

ここでは、J2EE サーバの設定ファイル「usrconf.properties」に追加したパラメタ「ejbserver.compiler.jvm.minHeapSize=64」を指定しています。

```

:
<define-server>
  <logical-server-name>MyJ2EE 1</logical-server-name>
  <logical-server-type>j2ee-server</logical-server-type>
  <configuration 2>
    <param>
      <param-name>ejbserver.compiler.jvm.minHeapSize 3</param-name>
      <param-value>64 4</param-value>
    </param>
    <param>
      :
    </param>
  </configuration 2>
</define-server>
<define-server>
  :

```

注 1

指定対象の J2EE サーバ名であることを確認してください。

注 2

<configuration> タグ間にパラメタを指定してください。また、複数のパラメタを指定できます。

注 3

パラメタ名を指定します。

注 4

パラメタ値を指定します。

Web サーバの設定ファイルに追加したパラメタを指定する場合

ここでは、Web サーバの設定ファイル「httpsd.conf」に追加したパラメタ「Port 81」を指定しています。

```

:
<define-server>
  <logical-server-name>MyHWS 1</logical-server-name>
  <logical-server-type>web-server</logical-server-type>
  <configuration 2>
    <param>
      <param-name>Port 3</param-name>
      <param-value>81 4</param-value>
    </param>
    <param>
      :
    </param>
  </configuration 2>
</define-server>
<define-server>
  :

```

注 1

指定対象の Web サーバ名であることを確認してください。

注 2

<configuration> タグ間にパラメタを指定してください。また、複数のパラメタを指定できます。

注 3

パラメタ名を指定します。

注 4

パラメタ値を指定します。

起動オプションを指定する場合

ここでは、J2EE サーバに起動オプション「-nosecurity」を指定しています。

```
      :
    <define-server>
      <logical-server-name>MyJ2EE 1</logical-server-name>
      <logical-server-type>j2ee-server</logical-server-type>
      <configuration 2>
        <param>
          <param-name>additional.startcmd 3</param-name>
          <param-value>-nosecurity 4</param-value>
        </param>
        <param>
          :
```

注 1

指定対象の J2EE サーバ名であることを確認してください。

注 2

<configuration> タグ間にパラメタを指定してください。また、複数のパラメタを指定できます。

注 3

パラメタ名に「additional.startcmd」を指定します。

注 4

パラメタ値に、起動オプション「-nosecurity」を指定します。

環境変数を設定する場合（サーバ共通）

ここでは、J2EE サーバに環境変数 AAA=aaa を指定し、BBB=bbb を無効にしています。

```

:
<define-server>
<logical-server-name>MyJ2EE 1</logical-server-name>
<logical-server-type>web-server</logical-server-type>
<configuration 2>
  <param>
    <param-name>user.env.variable.m 3</param-name>
    <param-value>.AAA=aaa 4</param-value>
    <param-value>#BBB=bbb 4</param-value>
  </param>
  <param>
    :

```

注 1

指定対象の J2EE サーバ名であることを確認してください。

注 2

<configuration> タグ間にパラメタを指定してください。また、複数のパラメタを指定できます。

注 3

パラメタ名に「user.env.variable.m」を指定します。

注 4

<param-value> タグに、「変数名 = 値」の形式で環境変数を指定します。複数指定する場合は、<param-value> タグを追加してください。

なお、設定を有効にする場合には変数名の先頭に「.」を指定し、設定を無効にする場合には変数名の先頭に「#」を指定してください。

(c) Management Server の設定

MyEclipse および Server Plug-in を使用するために、Management Server の設定ファイルにプロパティを追加します。ファイルの格納先と追加するプロパティは次のとおりです。

表 F-2 MyEclipse および Server Plug-in を使用するために設定するファイル

ファイル名	ファイル格納先	追加するプロパティ
mserver.properties	<Cosminexus のインストールディレクトリ>¥manager¥config¥	com.cosminexus.mnagsvr.management.enabled=true
		com.cosminexus.mnagsvr.management.connector.enabled=true
		com.cosminexus.mnagsvr.management.port=28099
adminagent.properties	<Cosminexus のインストールディレクトリ>¥manager¥config¥	adminagent.j2ee.process.console_event.enabled=true

注

このキーは J2EE サーバのコンソール情報を出力するキーです。ユーザサーバのコンソール情報を出力する場合は、adminagent.userserver.process.console_event.enabled キーに true を設定してください。

なお、Management Server 実行中にプロパティを設定しても反映されません。設定を反映するには、Management Server および運用管理エージェントを再起動する必要があります。

(d) Management Server のセットアップ

Management Server をセットアップします。次のコマンドを実行して、セットアップを実行してください。このとき、Management Server の管理ユーザ ID および管理ユーザパスワードもあわせて設定します。

```
prompt > cd <Cosminexusのインストールディレクトリ>%manager%bin
prompt > mngsvrctl setup -u <管理ユーザID> -p <管理ユーザパスワード>
```

(e) Management Server のサービスの起動

次のサービスを起動します。スタートメニューの [設定] - [コントロールパネル] - [管理ツール] - [サービス] を選択し、[サービス] ダイアログを表示させます。[サービス] ダイアログで次のサービスを起動してください。

- Cosminexus Management Server - Administration Agent
- Cosminexus Management Server

(f) Web システムの構築

簡易構築定義ファイルで設定した Web システムの環境を構築します。次のコマンドを実行して、構築してください。

```
prompt > cd <Cosminexusのインストールディレクトリ>%manager%bin
prompt > cmx_build system -m localhost:28080 -u <管理ユーザID> -p <管理ユーザパスワード> -f <簡易構築定義ファイル>
```

(3) 移行した開発環境の操作

Cosminexus Version 7 に移行した開発環境の操作について説明します。

(a) Management Server と運用管理エージェントの自動起動、自動停止

構築した開発環境で運用する場合は、Management Server および運用管理エージェントを常に起動しておく必要があります。そこで、Management Server および運用管理エージェントを OS の起動時に自動起動し、OS の停止時に自動停止するように設定します。次のコマンドを実行して、Management Server および運用管理エージェントの自動起動、自動停止を設定してください。


```
prompt > cd <Cosminexusのインストールディレクトリ>%manager%bin
prompt > mngautorun both
```

(b) 論理サーバの起動, 停止

Web システム上にある論理サーバの起動および停止は, MyEclipse および Server Plug-in で実施します。Server Plug-in を使用した論理サーバの起動および停止については, マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

(c) J2EE アプリケーションとリソースアダプタの操作

J2EE アプリケーションおよびリソースアダプタのデプロイ, アンデプロイ, 起動, および停止は, MyEclipse および Server Plug-in で実施します。Server Plug-in を使用した J2EE アプリケーションおよびリソースアダプタの操作については, マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

なお, J2EE アプリケーションおよびリソースアダプタをデプロイした場合は, デプロイ後に Cosminexus Version 6 の開発環境から退避させた属性ファイルをインポートしてください。

(d) Web システムの削除

Web システムを削除する場合は, 次のコマンドを実行します。ただし, コマンドを実行する場合は, Management Server および運用管理エージェントが起動中で, Web システム上のサーバが停止していることを確認してください。

```
prompt > cd <Cosminexusインストールディレクトリ>%manager%bin
prompt > cmx_delete_system -m localhost:28080 -u <管理ユーザID> -p <管理ユーザパスワード> -s MyWebSystem
```

(e) Web システムの再構築する場合の注意事項

Web システムを再構築する場合は, 次の点に注意してください。

- いったん Web システムを削除すること。
- 管理ユーザ ID とパスワードを設定しないこと。

付録 G JavaMail の使用例

JavaMail は、Web アプリケーション、または EJB アプリケーションで使用できます。ここでは、JavaMail の送信方法を説明します。JavaMail は、次の手順で送信します。

1. Session オブジェクトの取得
2. メッセージの作成
3. メッセージの送信

なお、以降の説明では、次のクラスのパッケージ名を省略します。

- javax.mail.NoSuchProviderException
- javax.mail.Session
- javax.mail.Transport
- javax.mail.MessagingException
- javax.mail.internet.MimeMessage
- javax.mail.internet.InternetAddress

付録 G.1 Session オブジェクトの取得

Session オブジェクトの取得方法および設定方法には、メールコンフィグレーションを使用する場合と、メールコンフィグレーションを使用しない場合があります。ここでは、それぞれの Session オブジェクトの取得方法と設定方法を説明します。

(1) メールコンフィグレーションを使用する場合

ここでは、メールコンフィグレーションを使用する場合の Session オブジェクトの取得方法と設定方法を説明します。

(a) メールコンフィグレーションの設定

メールコンフィグレーションを取得する前に、メールコンフィグレーションの設定とリソースのリンク解決をします。メールコンフィグレーションの設定およびリソースのリンク解決の手順については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

メールコンフィグレーションの設定例を示します。属性ファイルを次のように作成します。

```
<?xml version="1.0" encoding="MS932"?>
<!DOCTYPE hitachi-mail-property PUBLIC
'-//Hitachi, Ltd.//DTD Mail Property 7.0//EN'
'http://localhost/hitachi-mail-property_7_0.dtd'>
<hitachi-mail-property>
  <description>メール属性ファイルの例</description>
  <display-name>TheMailSession</display-name>
  <from>from_address@example.com</from>
  <server>MyMailServer</server>
  <runtime/>
</hitachi-mail-property>
```

次に、リソースの属性設定コマンドを使って、属性ファイルをリソースとして登録します。コマンドの指定例を示します。

```
cjsetresprop -type mail -resname TheMailSession -c mail.xml
```

(b) メールコンフィグレーションの取得

メールコンフィグレーションは、アノテーションまたは JNDI を使用して取得します。それぞれの取得方法を説明します。

アノテーションを使用したメールコンフィグレーションの取得

アノテーションを使用する場合は、次のような記述をしてメールコンフィグレーションを取得します。

```
// フィールド変数
// アノテーションによるリソースの設定
@Resource(mappedName="TheMailSession")
Session session;
```

JNDI を使用したメールコンフィグレーションの取得

アノテーションを使用しない場合は、リソースのリンク解決をして、JNDI を使用してメールコンフィグレーションを取得します。

```
Session session = null;
try {
    javax.naming.InitialContext initial =
        new javax.naming.InitialContext();
    session = (Session) initial.lookup("java:comp/env/TheMailSession");
} catch (javax.naming.NamingException e) {
    // リソースが見つからなかった等
}
```

(2) メールコンフィグレーションを使用しない場合

メールコンフィグレーションを使用しない場合は、Session クラスのファクトリメソッドを使用して、Session オブジェクトを取得できます。この場合、メールコンフィグレーションで設定する項目は取得できないので、プログラム内で設定する必要があります。

```
java.util.Properties prop = new java.util.Properties();  
// mail.hostプロパティを設定。  
// 他、mail.transport.protocol、mail.user、mail.from等必要なプロパティを設定。  
prop.setProperty("mail.host", "smtp.example.com");  
Session session = Session.getInstance(prop);
```

付録 G.2 メッセージの作成

送信するメッセージを作成します。

メッセージの内容に、次の項目を設定してください。

- From ヘッダフィールド
引数を設定しない場合は、InternetAddress クラスの getLocalAddress メソッドの戻り値が From ヘッダフィールドに設定されます。メールコンフィグレーションを使用した場合、メールコンフィグレーションで設定した From ヘッダフィールドの値がメッセージに設定されます。
- To ヘッダフィールド
MimeMessage.RecipientType.TO の代わりに MimeMessage.RecipientType.CC、MimeMessage.RecipientType.BCC を指定することで、CC、BCC ヘッダフィールドを設定できます。
- Subject ヘッダフィールド
charset パラメタを指定しない場合は、mail.mime.charset プロパティの値がエンコードに使用されます。
- ユーザ定義フィールド
- メール本文
メールの内容にコンテンツタイプ text/plain として本文を設定します。また、charset パラメタを指定しない場合は、mail.mime.charset プロパティの値がエンコードに使用されます。
- Date ヘッダフィールド

メッセージ内容の設定例を示します。

```

MimeMessage msg = new MimeMessage(session);
try{
    // Fromヘッダフィールドの設定
    msg.setFrom();

    // Toヘッダフィールドの設定
    msg.setRecipients(MimeMessage.RecipientType.TO,
        "to_address@example.com");

    // Subjectヘッダフィールドの設定
    msg.setSubject("メッセージ主題", "ISO-2022-JP");

    // ユーザ定義フィールドの設定
    msg.setHeader("MyHeader", "MySendMailClient");

    // メール本文の設定
    msg.setText("メールメッセージ本文", "ISO-2022-JP");

    // Dateヘッダフィールドの設定
    msg.setSentDate(new java.util.Date());

} catch (MessagingException e) {
    // メールアドレスの解析失敗等
}

```

付録 G.3 メッセージの送信

Transport クラスの send メソッドを使用して、メッセージを送信します。

```

try {
    Transport.send(msg);
} catch (MessagingException e) {
    // メッセージの送信に失敗した場合
}

```

また、複数のメッセージを送信する場合は、Transport オブジェクトを作成することで、1 接続で複数のメッセージを送信できます。

```

// Transportクラスのオブジェクトを得る
Transport transport = null;
try {
    transport = session.getTransport("smtp");
} catch (NoSuchProviderException e) {
    // プロバイダが見つからない場合
    // デフォルトの設定ではSMTPプロバイダが設定されているため発生しない
}
try {
    // SMTPサーバへの接続
    transport.connect();

    // 送信前に、ヘッダの更新を行う
    msg.saveChanges();

    // メッセージの送信
    transport.sendMessage(msg, msg.getAllRecipients());
}

```

```
// 複数のメッセージを送信する場合sendMessage()メソッドを繰り返し呼び出す。  
// transport.sendMessage(msg2, msg2.getAllRecipients());  
// ...  
} catch (MessagingException e) {  
    // SMTPサーバへの接続、メッセージの送信失敗時  
} finally {  
    try{  
        // SMTPサーバからの切断  
        transport.close();  
    }catch(MessagingException e){  
        // サーバ切断失敗時  
    }  
}
```

付録 G.4 電子メールアドレス指定時の注意事項

次のクラスのメソッドには、電子メールアドレスは RFC822 に準拠した電子メールアドレスを指定してください。

- javax.mail.internet.MimeMessage
- javax.mail.internet.InternetAddress
- javax.mail.Transport
- 上記三つのクラスを継承したクラス

ただし、これらのメソッドに対して電子メールアドレスの検証をしない場合や、検証をする API が RFC822 に違反する電子メールアドレスをエラーとしない場合があるので注意してください。

付録 H web.xml 編集時の注意事項（互換用の動作モードの場合）

旧バージョンとの互換用のシステムの動作モードとして、ベーシックモードおよびサーブレットエンジンモードがあります。ここでは、ベーシックモードまたはサーブレットエンジンモードの場合の、Web アプリケーション作成時の注意事項について説明します。

（１）ベーシックモードの場合

1.4 モードとの差異はありません。1.4 モードの場合の web.xml 編集時の注意事項については、「4.6.5 web.xml 編集時の注意事項」を参照してください。

（２）サーブレットエンジンモードの場合

サーブレットエンジンモードの場合で、web.xml を編集するときには次の内容に注意してください。

なお、これ以外の注意事項については、1.4 モードとの差異はありません。このほかの注意事項については、「4.6.5 web.xml 編集時の注意事項」を参照してください。

セキュリティロール使用時の設定について

web.xml の <security-role> タグを使用する場合、実行環境の web-users.xml 上で <security-role> の <role-name> と同じ名称でロールを定義してください。

web.xml の記述内容とサーバの動作

Servlet 2.3 に対応したアプリケーションの場合で、サーブレットエンジンモードで起動したときの、web.xml の記述内容とサーバの動作を次の表に示します。

表 H-1 web.xml の記述内容とサーブレットエンジンモードの場合のサーバの動作

web.xml の記述内容	サーバの動作
プロパティに webserver.xml.validate=false を設定している場合で、DTD の仕様で必須となっているタグを記述しないとき	無視して実行されるか、リクエスト実行時にエラーとなります。 必須のタグは、必ず記述してください。
<filter> タグに含まれる <filter-name> タグ要素を空要素にし、<filter-mapping> タグに含まれる <filter-name> タグ要素も空要素にした場合	指定されたフィルタが実行されます。
<listener> タグに含まれる <listener-class> タグ要素を空要素にした場合	サーバ起動時に該当する Web アプリケーションが無効となります。
プロパティに webserver.xml.validate=false を設定している場合で、DTD の仕様で一つだけ指定が許されているタグを複数記述したとき	後ろに記述された値を使用して正常に動作することがありますが、一つだけ指定が許されているタグを複数記述しないでください。
要素が同じ <filter-mapping> タグを複数記述した場合	記述された回数分の doFilter メソッドが呼び出されます。

web.xml の記述内容	サーバの動作
親タグと、その下に要素としてキーとなるタグとそのほかの情報を持つタグを複数記述した場合で、キーとなるタグの要素が同じでそのほかの情報が異なるとき	あとに記述された親タグを有効にします。
プロパティに webserver.xml.validate=false を設定している場合で、DTD の仕様で規定されていない順序でタグを記述した場合	正常に動作することがありますが、規定されている順序でタグを記述するようにしてください。
<jsp-file> タグに「/」で始まらない文字列を記述した場合	サーバ起動時にエラーになります。
次に示す <url-pattern> タグに、「*.」以外の「/」で始まらない文字列を記述した場合 <ul style="list-style-type: none"> <servlet-mapping><url-pattern> <security-constraint><web-resource-collection><url-pattern> 	サーバ起動時にエラーになります。
<mime-mapping> タグに含まれる <extension> タグ要素に空文字を記述した場合	設定を無視して起動されます。
<mime-mapping> タグに含まれる <mime-type> タグ要素に空文字を記述した場合	正常に動作します。
<error-page> タグに含まれる <error-code> タグ要素に空文字を記述した場合	設定を無視して起動されます。
<transport-guarantee> タグに空文字を記述した場合	「CONFIDENTIAL」として動作します。
<form-login-page> タグまたは <form-error-page> タグに空文字を記述した場合	サーバ起動時にエラーになります
<env-entry-type> タグ、<ejb-ref-type> タグ、または <ejb-ref-type> タグに空文字を設定した場合	正常に起動します。
<security-constraint> を設定して <auth-method> を省略した場合	サーバ起動時にエラーになります。

サーブレットエンジンモードで使用できないタグ

web.xml に記述する次のタグは、JNDI 名前空間に登録するため、サーブレットエンジンモードでは使用できません。J2EE サーバモードで有効になります。

- env-entry
- ejb-ref
- ejb-local-ref
- resource-ref
- resource-env-ref

Servlet 2.4 仕様でサポートされない web.xml の要素

Cosminexus では、次に示す Servlet 2.4 仕様の web.xml の要素は定義できません。
定義した場合、デプロイ時にエラーになります。

- message-destination
- message-destination-ref
- service-ref

また、サーブレットエンジンモードの場合、次のプロパティの定義で web.xml および
タブライブラリ・ディスクリプタ（TLD ファイル）のチェックを無効化すると、
Servlet 2.3 仕様の web.xml に Servlet 2.4 の要素を記述してもデプロイ時にエラーは
通知されません。無効な要素として無視されます。

webserver.xml.validate=false

なお、Servlet 2.4 仕様で追加、変更された仕様を、Cosminexus 上で使用する場合は、
上記以外の注意事項については、「4.6.6 Servlet 2.4 仕様で追加、変更された仕様
についての注意事項（web.xml）」を参照してください。

付録Ⅰ インスタントセットアップ機能実行時のエラーメッセージ一覧

ここでは、インスタントセットアップ機能を実行したときにコンソールに出力されるエラーメッセージについて示します。なお、ここで説明するエラーメッセージは、メッセージプリフィックスが KEDT のメッセージだけです。それ以外のエラーメッセージについては、次の表に示すどれかのマニュアルを参照してください。

表 I-1 エラーメッセージの参照先（プリフィックス KEDT 以外）

プリフィックス	参照先
KDAL	マニュアル「Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編」
KDJE	
KEOS	マニュアル「Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編」
KEUC	
KFCB	
KFCT	マニュアル「Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編」
KFDB	
KFDJ	

エラーメッセージは次の形式で出力されます。

KEDTnnnnn-E

- KEDT
メッセージプリフィックス ID を表します。この内容は固定です。
- nnnnn
メッセージ番号です。
- E
メッセージのレベルを示します。E (Error) は、エラーレベルのトラブルが発生したことを通知するメッセージです。このメッセージが出力されたときは、処理を中断します。

KEDT10013-E

The error occurred by the copying file. (details=< ファイル（ディレクトリ）の絶対パス >)

説明

ファイル、またはディレクトリのコピーでエラーが発生しました。

対処

次の内容を確認してください。

- < ファイル（ディレクトリ）の絶対パス > のファイル、またはディレクトリが存在すること。

- ・ <ファイル（ディレクトリ）の絶対パス> のファイル，またはディレクトリの読み込み権限があること。
- ・ コピー先ディレクトリに書き込み権限があること。

コピー対象がファイルの場合は，次の内容も確認してください。

- ・ コピー先ディレクトリに同じ名前のフォルダがないこと。
- ・ コピー先ディレクトリに同じ名前のファイルがある場合，読み込み権限と書き込み権限があること。

コピー対象がディレクトリの場合は，次の内容も確認してください。

- ・ コピー先ディレクトリに同じ名前のファイルがないこと。
- ・ コピー先ディレクトリに同じ名前のフォルダがある場合，書き込み権限があること。

KEDT10014-E

The error occurred by the editing file.(details=< ファイルの絶対パス >)

説明

ファイルの編集でエラーが発生しました。

対処

次の内容を確認してください。

- ・ <ファイルの絶対パス> のファイルが存在すること。
- ・ <ファイルの絶対パス> のファイルの読み込み権限および書き込み権限があること。

KEDT10015-E

The error occurred by the executing file.(details=< ファイルの絶対パス >)

説明

ファイルの実行でエラーが発生しました。

対処

このメッセージの前後に出力されているエラーメッセージを参照して対処してください。

上記の対応で対処方法がわからない場合は，次のログファイルを参照してください。

- ・ 呼び出し先のコマンド名（ファイル名）が「cmx」から始まる場合
 <Cosminexus のインストールディレクトリ>•manager•log•message に出力されるログファイル
- ・ 呼び出し先のコマンド名（ファイル名）が「cj」から始まる場合
 <Cosminexus のインストールディレクトリ>•CC•admin•logs に出力されるログファイル

KEDT10016-E

The error occurred by the making folder.(details=< フォルダの絶対パス >)

説明

フォルダの作成でエラーが発生しました。

対処

次の内容を確認してください。

- <フォルダの絶対パス>のフォルダ、およびファイルが存在しないこと。
- <フォルダの絶対パス>の親ディレクトリに書き込み権限があること。

KEDT10019-E

The error occurred by the extracting archive files.(details=< ファイルの絶対パス >)

説明

アーカイブファイルの展開でエラーが発生しました。

対処

次の内容を確認してください。

- <ファイルの絶対パス>のファイルが存在すること。
- <ファイルの絶対パス>のファイルの読み込み権限があること。
- アーカイブの展開先ディレクトリに書き込み権限があること。
- アーカイブの展開先ディレクトリと同じパスのファイルがないこと。
- アーカイブの展開先のドライブに十分な空き容量があること。

上記の条件を満たしていてもエラーが発生している場合は、<ファイルの絶対パス>のファイルが壊れているおそれがあります。この場合は、ファイルのダウンロードをやり直してください。

KEDT10025-E

The error occurred by the adding extension location.(details=< フォルダの絶対パス >)

説明

拡張ロケーションの追加でエラーが発生しました。

対処

次のどちらかの作業をしてください。

次の内容を確認してください。

- <フォルダの絶対パス>のフォルダが存在すること。
- <フォルダの絶対パス>のフォルダの読み込み権限があること。

上記の条件を満たしていてもエラーが発生する場合は、<フォルダの絶対パス>のフォルダに展開したアーカイブファイルが壊れているおそれがあります。この場合は、アーカイブファイルのダウンロードをやり直してください。

KEDT10031-E

The error occurred by the deleting folder. (details=< フォルダの絶対パス >)

説明

フォルダの削除でエラーが発生しました。

対処

次の内容を確認してください。

- <フォルダの絶対パス>のフォルダが存在すること。
- <フォルダの絶対パス>のフォルダの書き込み権限があること。
- Eclipse のインストール先フォルダの場合は、Eclipse が起動していないこと。

付録 J データベース構築・削除時のエラーメッセージ一覧

ここでは、データベースの構築・削除時にエラーファイルに出力されるメッセージについて示します。ここに示した以外のデータベースのエラーメッセージについてはマニュアル「HiRDB メッセージ」を参照してください。

メッセージは次の形式で出力されます。

KFPXnnnnnn-E

- KFPX
メッセージプリフィックス ID を表します。この内容は固定です。
- nnnnn
メッセージ番号です。
- E
メッセージインジケータ（エラーレベル）です。
E：エラーメッセージ（Error）

出力されるメッセージのエラーファイルの格納先を、次に示します。

データベースの構築時

<Cosminexus のインストールディレクトリ >¥DB¥PDISTUP¥spool¥pdi_err.txt

データベースの削除時

<Cosminexus のインストールディレクトリ >¥DB¥PDISTUP¥spool¥pdirst_err.txt

KFPX29601-E

Invalid value in aa....aa option

説明

aa....aa オプションの指定値が不正です。

対処

オプションの指定値を見直してください。

KFPX29602-E

Invalid option specified, option name=aa....aa

説明

指定できないオプション aa....aa を指定しています。

対処

オプションを見直してください。

KFPX29603-E

Specified directory aa....aa not HiRDB directory

説明

指定されたディレクトリ aa....aa は、HiRDB 運用ディレクトリではありません。
ディレクトリ名が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイト
に続いて二つの半角ピリオド (..) を付けて表示します。

対処

ディレクトリを見直してください。

KFPX29604-E

Command argument " hirdb_directory " must be specified

説明

HiRDB 運用ディレクトリが指定されていません。

対処

HiRDB 運用ディレクトリを指定してください。

KFPX29605-E

Specified port_no already used, port_no=aa....aa

説明

ポート番号 aa....aa が重複しています。

対処

ポート番号を見直してください。

KFPX29606-E

Specified Error found in system definition input file, file=aa....aa, line=bb....bb

説明

システム定義入力ファイル aa....aa のエラー行 bb....bb に誤りがあります。
ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに
続いて二つの半角ピリオド (..) を付けて表示します。
エラー行が 30 バイトを超える場合、超える分は切り捨てて先頭 30 バイトに続いて
二つの半角ピリオド (..) を付けて表示します。

対処

システム定義入力ファイルのエラー行を見直してください。

KFPX29607-E

File not found, file_name=aa....aa

説明

次のどれかのファイル aa....aa が存在しません。

- システム定義入力ファイル
- 初期値ファイル
- HiRDB サーバ用 OS 環境変数設定ファイル

ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

ファイルが存在するか見直してください。

HiRDB サーバ用 OS 環境変数設定ファイルの場合、OS 登録コマンドを実行する必要があります。実行しているか確認してください。

KFPX29608-E

File open failed, file_name=aa....aa

説明

次のどれかのファイル aa....aa のオープンに失敗しました。

- システム定義入力ファイル
- 初期値ファイル
- HiRDB サーバ用 OS 環境変数設定ファイル
- テーブル定義ファイル
- 運用管理スクリプトファイル

ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

ファイルがオープンできる状態にあるか見直してください。

KFPX29609-E

File create failed, file_name=aa....aa

説明

次のどれかのファイル aa....aa の生成に失敗しました。

- システム定義ファイル
- HiRDB 環境構築用シェル (バッチ) / 制御文ファイル
- セットアップ処理取り消し用シェル (バッチ) ファイル

ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

ファイルが作成できる状態にあるか見直してください。

KFPX29610-E

File read failed, file_name=aa....aa

説明

システム定義入力ファイル aa....aa の読み込みに失敗しました。

ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

ファイルが読み込める状態にあるか見直してください。

KFPX29611-E

Directory create failed, directory_name=aa....aa

説明

次のどれかのディレクトリ aa....aa の作成に失敗しました。

- HiRDB ファイルシステム領域作成用ディレクトリ
- 運用管理スクリプト作成先ディレクトリ (\$PDDIR/conf/emb)
- アンロードログファイル出力先ディレクトリ

ディレクトリ名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

ディレクトリが作成できる状態であるか見直してください。

KFPX29612-E

Insufficient disk capacity for rdarea

説明

RD エリア用 HiRDB ファイルシステム領域用のディスクの空き領域が不足しています。

対処

RD エリアの容量、またはディスクとの空き領域を見直してください。

KFPX29613-E

Insufficient disk capacity for system_file_A

説明

システムファイル A 用 HiRDB ファイルシステム領域用のディスクの空き領域が不足しています。

対処

システムファイルの容量、またはディスクの空き領域を見直してください。

KFPX29614-E

Insufficient disk capacity for system_file_B

説明

システムファイル B 用 HiRDB ファイルシステム領域用のディスクの空き領域が不足しています。

対処

システムファイルの容量，またはディスクの空き領域を見直してください。

KFPX29615-E

Insufficient disk capacity for drive: aa....aa

説明

r オプションで指定した HiRDB ファイルシステム領域用 aa....aa の空き領域が不足しています。

aa....aa：Windows 版の場合：ドライブ名

UNIX 版の場合：スペシャルファイル名

ドライブ名またはスペシャルファイル名が 126 バイトを超える場合，超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

HiRDB ファイルシステム領域の容量，またはディスクの空き領域を見直してください。

KFPX29616-E

Errors occurred in creation of HiRDB file system area: aa....aa

説明

HiRDB ファイルシステム領域の作成でエラーが発生しました。

aa....aa：エラーの発生した運用コマンドまたはユーティリティのメッセージ

メッセージが 126 バイトを超える場合，超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

HiRDB のエラーメッセージを調査してください。

KFPX29617-E

Errors occurred in initialization of system file: aa....aa

説明

システムファイルの初期設定でエラーが発生しました。

aa....aa：エラーの発生した運用コマンドまたはユーティリティのメッセージ

メッセージが 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

HiRDB のエラーメッセージを調査してください。

KFPX29618-E

Errors occurred in initialization of database: aa....aa

説明

HiRDB の起動またはデータベースの初期設定でエラーが発生しました。

aa....aa : エラーの発生した運用コマンドまたはユーティリティのメッセージ
メッセージが 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

HiRDB のエラーメッセージを調査してください。

KFPX29619-E

Errors occurred in creation of database schema: aa....aa

説明

ユーザデータベースの作成でエラーが発生しました。

aa....aa : エラーの発生した運用コマンドまたはユーティリティのメッセージ
メッセージが 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

HiRDB のエラーメッセージを調査してください。

KFPX29620-E

Errors occurred in creation of temporary file , file=aa....aa

説明

一時ファイル aa....aa の作成に失敗しました。

ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

\$PDDIR/pdistup/tmp 下に一時ファイル aa....aa が作成できる状態になっているかを調査してください。

KFPX29621-E

Invalid file format, file=aa....aa

説明

次のどちらかのファイル aa....aa の書式が不正です。

- setup.ini (Windows の場合)
- HiRDB サーバ用 OS 環境変数設定ファイル

ファイル名称が 126 バイトを超える場合、超える分は切り捨てて先頭 126 バイトに続いて二つの半角ピリオド (..) を付けて表示します。

対処

- setup.ini の場合
HiRDB 環境が壊れているおそれがあるため、インストールから実行し直してください。
- HiRDB サーバ用 OS 環境変数設定ファイルの場合
HiRDB サーバ用 OS 環境変数設定ファイルが壊れているおそれがあるため、OS 登録コマンドから実行し直してください。

KFPX29622-E

Invalid specification of HiRDB file system area

説明

システム B のディレクトリを指定したとき、初期値入力ファイルで、一つのファイルシステム領域にシステムファイル A およびシステムファイル B の両方を指定しています。

対処

引数の指定値、または初期値入力ファイルの指定を見直してください。

KFPX29623-E

Invalid specification of user name in user table definition

説明

テーブル定義ファイルに指定されているユーザ名の指定が不正です。

対処

ユーザ名の指定を見直してください。

KFPX29624-E

HiRDB setup environment already exists

説明

HiRDB のセットアップ環境がすでに存在します。

対処

HiRDB 運用ディレクトリの指定値を見直すか、アンセットアップ (pdirst) でアンセットアップ後、再実行してください。

KFPX29625-E

Insufficient memory on PROCESS, code=aa....aa

説明

メモリが不足しています。

メモリが不足している処理の詳細コード aa....aa は次のとおりです。

01：引数解析処理

02：初期設定処理

03：システム定義入力ファイル読み込み処理

04：システム定義作成処理

05：データベース初期設定ファイル作成処理

06：取り消し用ファイル作成処理

対処

次のどれかの方法で、使用できるメモリを増やしてください。

- 同時実行しているプロセス数を減らす。
- スワップ領域を増やす。
- 実メモリを増設する。

KFPX29626-E

Internal error: creation of database information

説明

内部エラーです。データベース情報の作成処理でエラーが発生しました。

対処

保守員に連絡してください。

KFPX29627-E

Internal error: creation of temporary object

説明

内部エラーです。内部オブジェクトの作成処理でエラーが発生しました。

対処

保守員に連絡してください。

KFPX29628-E

Not specified "user" value for pd_ha operand in system definition

説明

システム共通定義の pd_ha オペランドに "use" が指定されていません。

対処

システム定義入力ファイル（システム共通定義 pdsys）の次の項目を見直してください。

- pd_ha オペランドが指定されているか。
- pd_ha オペランドに " use " が指定されているか。

KFPX29629-E

Number of aa....aa exceeds maximum

説明

次のどれかの指定値が指定できる最大数を超えています。

最大数を超えた指定値 aa....aa は次のとおりです。

- values for -r option : -r オプション指定数
- HiRDB file system areas : HiRDB ファイルシステム領域数
- RDAREAs : RD エリア数
- system log files : システムログファイル数
- synchronization point dump files : シンクポイントダンプファイル数
- status files : ステータスファイル数

対処

次の項目を見直してください。

aa....aa : values for -r option の場合

-r オプションに 129 個以上のエイリアス名が定義されていないか。

aa....aa : values for -r option 以外の場合

aa....aa で示された領域, またはファイルが初期値ファイル (pdi_s.ini) の定義で制限値を超えていないか。

KFPX29630-E

Invalid value in initial value definition file, section= aa....aa, entry= bb....bb

説明

初期値ファイルの指定値が不正であるか, 必要なエントリが設定されていません。

指定値に不正があるセクションまたはエントリは次のとおりです。

aa....aa : 指定値に不正があるセクション名

bb....bb : 指定値に不正があるエントリ名

対処

不正があるセクションおよびエントリの指定値を見直し, 再実行してください。

付録 K 用語解説

(数字)

1.4 モード

サーバの動作モードです。J2EE 1.4 の機能を使用できます。データベースを含む複数のリソースのトランザクション管理ができます。

(英字)

Application Server

Cosminexus の実行環境を構築する基盤製品です。Application Server Standard と、Application Server Enterprise の総称です。

BMP Entity Bean

BMP Entity Bean の BMP とは、Bean-Managed Persistence の略で、永続化に関する処理を、Bean 自身が行うという意味です。リソースへのアクセスコードは、すべてアプリケーション開発者が記述する必要があります。実装は CMP よりも手間が掛かりますが、CMP より適用範囲が広く自由度も高くなります。

CMP Entity Bean

CMP Entity Bean の CMP とは、Container-Managed Persistence の略で、永続化に関する処理を、EJB コンテナが行うという意味です。CMP の場合、開発者はリソース（データベースなど）へのアクセスのためのコードを書く必要がなく、リソースとのやり取りはすべて EJB コンテナが行います。

CORBA ネーミングサービス

CORBA の仕様に準拠した、リモートオブジェクトの格納場所を管理するためのネーミングサービスです。構成ソフトウェアである Cosminexus TPBroker によって提供される機能です。

Cosminexus

アプリケーションサーバを中核とした、性能および信頼性の高いアプリケーションを実行および開発するためのシステム構築基盤製品です。

Cosminexus Application Development Plug-in

アプリケーションの開発、テスト時に使用するプラグインです。

Cosminexus cFramework

MVC (Model View Controller) アーキテクチャに従い、J2EE に準拠する Web アプリケーションのコンポーネント・フレームワークを提供する製品です。Application Server Standard、Application Server Enterprise、および Developer Professional にバンドルされます。なお、Developer Professional にバンドルされる Cosminexus cFramework には、実行用のライセンスは含まれません。

Cosminexus Component Container

サーバサイドの業務処理プログラム（ビジネスロジック）をコンポーネントとして実行するための構成ソフトウェアです。また、Cosminexus システムの運用管理をするための Management Server、統合ユーザ管理、snapshot ログ収集などの機能も提供しています。

Cosminexus Component Transaction Monitor (CTM)

クライアントからのリクエストのスケジューリングを実現するための構成ソフトウェアです。

Cosminexus Reliable Messaging

Cosminexus システム上の J2EE アプリケーションがメッセージを使用して非同期に通信するためのミドルウェアです。JMS インタフェースでのメッセージ通信機能を J2EE アプリケーションに提供します。

Cosminexus コネクタ

Eclipse のプラグインです。MyEclipse から J2EE サーバを起動したり、J2EE アプリケーションを J2EE サーバにデプロイしたりするときに使用します。

DD (Deployment Descriptor)

アプリケーションを運用環境に配置するときの定義情報を記述したものです。EJB 用、Web アプリケーション用、Enterprise アプリケーション用などが Sun からの仕様に規定されています。

Dependency Injection

EJB やインターセブタクラスのフィールドや setter メソッドに、@EJB や @Resource アノテーションを指定することで、EJB やリソースへの参照を EJB コンテナが自動的にセットする機能です。Dependency Injection を使用すると、EJB やリソースへの参照を JNDI を使用してルックアップする必要がなくなります。

Developer

Cosminexus の開発環境を構築する基盤製品です。Developer Standard および Developer Professional の総称です。

EAR (Enterprise ARchive) ファイル

J2EE アプリケーションを構成する複数の EJB-JAR ファイル、WAR ファイル、および DD を EAR ファイル形式でパッケージ化したものです。

Eclipse

Eclipse Foundation が提供するオープンソースの統合開発環境です。ソースコードの編集支援機能やデバッグ機能など、アプリケーションの開発効率を向上させる各種機能を備えています。

EJB (Enterprise JavaBeans)

分散ビジネスアプリケーションを開発・デプロイするためのコンポーネントアーキテクチャです。

EJB-JAR ファイル

Enterprise Bean、EJB-JAR DDなどをJAR形式に圧縮したものです。

EJB クライアントアプリケーション

J2EE サーバ上で開始している J2EE アプリケーション中の Enterprise Bean を呼び出すプログラ

ムのことです。

EJB コンテナ

Enterprise Bean の実体は、EJB コンテナの中で実行されます。EJB コンテナは、Enterprise Bean を制御すると同時に、通信、トランザクション管理などのシステムレベルのサービスを提供する実行環境です。

Enterprise Bean

業務処理プログラムを EJB アーキテクチャに従って作成したものです。

Entity Bean

永続化されたデータそのものを表す Bean です。Entity Bean は、さらに CMP Entity Bean と BMP Entity Bean に分類されます。

HTTP レスポンス圧縮機能

サーブレット、JSP、および静的コンテンツへの HTTP リクエストに対する HTTP レスポンスを、サーブレットフィルタによって gzip 形式に圧縮する機能です。この機能を使用することで、Web コンテナと Web クライアント（Web ブラウザなど）間の通信に掛かる時間を削減し、システムの応答時間を短縮できます。

IDE (Integrated Development Environment)

GUI を使用して Java プログラムを実装したり、WAR ファイルなどを作成したりできる統合開発環境です。Cosminexus では、Eclipse に対応しています。

J2EE アプリケーション

JSP、サーブレット、Enterprise Bean などで構成されるアプリケーションです。Cosminexus で扱う J2EE アプリケーションの形式には、EAR ファイル形式でパッケージ化されたアーカイブ形式の J2EE アプリケーションと、アーカイブ化しない展開ディレクトリ形式の J2EE アプリケーションがあります。EAR ファイル形式でパッケージ化された J2EE アプリケーションの場合、複数の EJB-JAR ファイル、複数の WAR ファイル、および一つの DD から構成されます。

J2EE アプリケーション実行時間の監視機能

EJB および Web アプリケーションのリクエスト実行時間を監視する機能です。実行時間監視中に障害を検知すると、メソッドタイムアウト機能およびメソッドキャンセル機能が動作します。

J2EE コンテナ

J2EE コンポーネントへ各種 API を提供する、Web コンテナと EJB コンテナから構成されます。

J2EE コンポーネント

サーブレット、JSP、Enterprise Bean などの J2EE アプリケーションを構成するプログラムのことです。

J2EE サーバ

J2EE コンテナを生成、実行する環境です。

J2EE サービス

J2EE コンテナの部品機能として利用されます。J2EE コンポーネントに JNDI , JDBC , JTA , RMI-IIOP , JavaMail の API を提供します。

JAAS (Java Authentication and Authorization Service)

Java 標準のユーザ認証の仕様です。Cosminexus では、JAAS に準拠したユーザ管理機能を、Cosminexus Manager の統合ユーザ管理フレームワークで提供しています。

jaas.conf (JAAS のコンフィグレーションファイル)

ユーザ認証ライブラリおよびシングルスサインオンライブラリの機能を使用するために必要な JAAS 認証用の設定ファイルです。

JAR (Java ARchive) ファイル

Java プログラムの実行に必要なクラスファイルやデータファイルを一つにまとめるためのアーカイブです。

JavaBeans

Java 言語を使用して開発され、部品化されたプログラムを組み合わせるアプリケーションを構築するための手法です。

JDBC (Java Database Connectivity)

Java のプログラムでデータベースを利用するための API です。Java プログラムから、JDBC API を経由して「JDBC ドライバ」を呼び出し、ドライバがデータベース、サーバへの接続機能を提供します。

JNDI (Java Naming and Directory Interface)

Java プラットフォーム用の標準拡張機能で、Java テクノロジーに対応したアプリケーションに、企業内の複数のネーミングおよびディレクトリサービスへの統一したインタフェースを提供します。JNDI は、Java Enterprise API セットの一部として、異機種システムの混在した社内のネーミングおよびディレクトリサービスに、スムーズに接続できるようにします。開発者は、JNDI を使用することで、移植性の高いアプリケーションを開発できます。

JSF

Web アプリケーションのユーザインタフェースを開発するためのフレームワークです。MVC (Model View Controller) アーキテクチャの Controller であるサーブレットで Web アプリケーションを管理する機能のほか、ユーザインタフェースの構成要素、Swing が持つようなイベント処理とコンポーネントのレンダリング要素を部品として持っており、それらの部品をフォームに配置することでユーザインタフェースを構築できます。

JSP (JavaServer Pages)

Web サーバ側で動的に Web 画面を作成するための仕組みです。Web 画面を作成する HTML の中に JSP 固有のタグを埋め込んで JavaBeans から値を受け取ったりできます。

JSP 事前コンパイル

Web アプリケーションに含まれる JSP ファイルおよびタグファイルをデプロイ前にコンパイルする機能です。デプロイ前にコンパイルしておくことで、Web アプリケーションに対する初回リクエスト

ト時のレスポンスタイムを短縮できます。

JSP 事前コンパイルの実行時に Java ソースファイルを生成したり、コンパイル対象外ファイルを指定したりすることもできます。

JTA (Java Transaction API)

J2EE アーキテクチャにトランザクション処理サービスを提供する API です。JTA には、トランザクションマネージャがトランザクション処理を行うときに、トランザクションに関与する各種コンポーネントとの間で必要となる API が定義されています。

Message-driven Bean

メッセージ受信を契機に、特定のメソッドを実行できる JMS 対応の Enterprise Bean です。

Session Bean、Entity Bean とは異なり、ホームインタフェースとリモートインタフェースを持たないため、クライアントから直接呼び出されることはありません。

MVC (Model / View / Controller)

MVC とは、Model と View、そして Controller の頭文字をとったものです。Model は、システムの中でロジックを担当します。View は、表示、入出力などの部分を担当します。Controller は、View と Model を制御します。

MyEclipse

Java アプリケーションの開発環境です。MyEclipse には、Eclipse でアプリケーションを開発するときに使用するプラグインがまとめられています。MyEclipse を使用するとアプリケーションの作成、J2EE サーバへのアプリケーションのデプロイ、J2EE サーバの起動・停止、データベースの操作などができます。また、Struts や JSF などのフレームワークを使用したアプリケーションの開発や、UML によるアプリケーション設計もできます。

RMI-IIOP (Remote Method Invocation over Internet Inter-ORB Protocol)

JavaRMI と JavaIDL を統合した API です。EJB コンポーネントとそれぞれのコンテナは、

Remote Method Invocation (RMI) テクノロジーを使って、分散オブジェクト間のメソッド呼び出しを実装します。

Server Plug-in

アプリケーションの実行環境および運用環境で使用する Eclipse 用のプラグインです。Server Plug-in を組み込むことで、アプリケーションを実行するときに必要なプロパティの設定や、論理サーバの起動、停止などの一連の操作が Eclipse 上で実行できるようになります。

Session Bean

Session Bean は、主に業務処理を実行するための Bean です。Session Bean は、クライアントを一つだけ持つことができます。クライアントが終了すると、対応する Session Bean も終了します。クライアントとのセッションの間だけ動作するため、一時的であり、非永続的な Bean です。Session Bean は、Stateful Session Bean と、Stateless Session Bean に分類されます。

SFO サーバ

システム内の J2EE サーバの J2EE アプリケーション内で生成されたグローバルセッション情報を管理する J2EE サーバです。セッションフェイルオーバー機能を使用するときにシステム構成に含めます。

Smart Composer 機能

Cosminexus を使用してシステムを構築する場合に、一般的な 3 階層モデルのアプリケーションサーバのシステムを、簡単に構築および運用できるよう支援する機能です。システム全体に対して、システムの設定や、アプリケーションおよびリソースアダプタのデプロイを一括で実行できるので、簡単、迅速にシステムを構築できます。

構築したシステムを一括で起動したり、一括でシステムの設定を変更したりするなどの運用機能もサポートしています。

Stateful Session Bean

Stateful Session Bean は、内部的な状態を保持できる Bean です。Stateless Session Bean とは異なり、ホームインタフェースからの生成によってインスタンス化されたあとは、メソッドの呼び出しごとに生成、消滅することはありません。セッション単位で連続した処理を行う場合に適した Bean です。

Stateless Session Bean

Stateless Session Bean は、内部的な状態を保持しない Bean です。Bean のインスタンスの生成・消滅は、すべて EJB コンテナが独自に管理しています。同じ Bean のメソッドを続けて呼び出した場合にも、それらが同じインスタンスである保証はありません。そのため、Stateless Session Bean では、内部状態を保持して、複数メソッドで連続した処理を行うことはできません。1 メソッドで完結するような処理を記述する場合に適しています。

Struts

Web アプリケーションのユーザインタフェースを開発するためのフレームワークです。MVC (Model View Controller) アーキテクチャの Controller であるサーブレットで Web アプリケーションを管理する機能のほか、ユーザインタフェースの構成要素を部品として持っており、それらの部品をフォームに配置することでユーザインタフェースを構築できます。

Timer Service

EJB コンテナで提供される機能です。指定した時刻、経過時間、または間隔で、EJB コンテナがタイムアウトメソッドをコールバックします。

UML (Unified Modeling Language)

OMG が制定したモデリング言語です。オブジェクト指向をベースとするアプリケーションの構造を、ソースコードよりも抽象化した形で構造的かつ形式的な形で表現できる表記方法を提供します。UML が提供する表記方法は、分析する内容や目的に応じて、ユースケース図、クラス図、シーケンス図などに分けられます。

WAR (Web ARchive) ファイル

Web アプリケーションの構成要素を WAR 形式に圧縮したファイルです。Web アプリケーションの実行に必要な HTML ファイル、JSP ページ、Java クラスファイル、JAR ファイル、および Web アプリケーション配置記述子が含まれます。

Web アプリケーション

Web ブラウザを備えたクライアントを対象に作成されたアプリケーションです。具体的には、サーブレットプログラム、JSP ページ、および HTML/XML ドキュメントなどの集合体です。

Web 画面

このマニュアルでは、Web アプリケーションの画面遷移を構成する、個々の画面を指します。

Web コンテナ

J2EE アーキテクチャの Web コンポーネント規約を実装するコンテナです。実行環境としてセキュリティ、トランザクションなどの各種サービスを提供します。Web アプリケーションは、Web コンテナ上で動作します。

(ア行)

アーカイブ形式

J2EE アプリケーションなどを構成するプログラムや DD を、アセンブルして WAR ファイル、EAR ファイル、または ZIP ファイルにまとめた J2EE アプリケーションです。

アクション

Web 画面の遷移上での動作を示す用語です。

アセンブル

単体では動作しない EJB-JAR をアプリケーションの 1 構成要素として位置づけるための組み立て作業のことです。

アノテーション

ソースコードにクラスやメソッドの付加情報などを埋め込むための記述方式です。アノテーションを指定することで、Enterprise Bean が依存しているリソースやほかの Enterprise Bean への参照を取得できます。

インプロセス HTTP サーバ

J2EE サーバのインプロセスで動作する Web サーバ機能です。Web コンテナの機能の一部として提供されます。

ウィザード

パッケージ・ソフト製品に組み込まれるヘルプ機能の一種です。設定作業を支援する対話型のナビゲータ機能を一般にウィザードと呼びます。表や文書の書式設定やクロス集計表の作成など、細かなノウハウが求められる複雑なパラメタ設定作業を対話形式で誘導（ナビゲーション）します。

(カ行)

開発環境インスタントセットアップ

Developer が提供する機能です。Eclipse と MyEclipse のインストール、J2EE アプリケーションの開発に必要なサーバ環境の構築を支援します。

サーバ環境については、Management Server、ホスト、パフォーマンスストレサ、J2EE サーバ、および組み込みデータベースを構築できます。ただし、Eclipse および MyEclipse のインストールに必要なアーカイブファイルをあらかじめダウンロードしておく必要があります。

カスタムログインモジュール

Cosminexus の標準ログインモジュール以外の認証モジュールを使用して、アプリケーションのユーザ認証をする場合に作成する実装クラスです。

組み込みデータベース

Developer に標準提供されているデータベースのことです。ローカルマシンでのアプリケーション開発、テストなどで利用できます。

グローバルセッション

セッションフェイルオーバー機能では、障害発生前の HTTP セッションと、障害発生後に引き継がれた HTTP セッションを一つのセッションとして扱います。これをグローバルセッションといいます。

グローバルセッション ID

グローバルセッションを管理する ID です。システム内でグローバルセッション情報を一意に識別するために、グローバルセッションに付けられます。

グローバルセッション情報

グローバルセッションが持つ情報をグローバルセッション情報といいます。グローバルセッション情報は SFO サーバで管理されます。

コンテキストルート

コンテキストのルートパスです。コンテキスト内の Web アプリケーションにアクセスするときに URL 上に指定します。コンテキストパスとも呼びます。

コンテナ拡張ライブラリ

Enterprise Bean、サーブレットおよび JSP から共通して利用できる、ユーザ作成のライブラリです。コンテナ拡張ライブラリは、J2EE の仕様外です。

(実行)

サーバ管理コマンド

J2EE サーバで管理しているアプリケーションおよびリソースの設定するためのコマンド群です。

サーバ起動・停止フック機能

J2EE サーバまたは Web コンテナサーバの起動・停止時に、コンテナ拡張ライブラリを使用するための初期化処理および停止処理を自動的に呼び出す機能です。

サーブレット

サーバ側で Java を実行させる方法の一つです。サーブレットは、Web サーバと連携して、Web サーバに単に HTML 文書や画像ファイルを送るだけではなく、アプリケーションを実行し、その結果を HTML 文書として送り返す機能を提供します。

サーブレットフィルタ

サーブレット/JSP へのリクエストや、サーブレット/JSP からのレスポンスを、フィルタリングする機能です。サーブレットフィルタによって、サーブレット/JSP の実行前のリクエスト、または実

行後のレスポンスをラップして、データの変更、リソースに対するトレースの取得などができます。Cosminexus では、セッションフェイルオーバー用のフィルタ、および HTTP レスポンス圧縮フィルタを提供しています。

条件分岐

Web 画面から入力された情報や、業務処理プログラムの処理結果によって、遷移先を分岐させる処理をいいます。

スマートエージェント

1.4 モードでグローバルトランザクションを使用する場合、または CTM を使用する場合に、動的な分散ディレクトリサービスを提供するプロセスです。

Management Server を利用する場合、スマートエージェントは論理サーバとして扱えます。OTS、TCS、CTM はスマートエージェントによって管理されます。

静的コンテンツ

HTML ファイルや画像ファイルなど、クライアントからの要求に対する応答に使用するファイルのうち、リクエスト内容に影響されないで常に同じ内容になるコンテンツのことです。

静的コンテンツキャッシュ機能

静的コンテンツをメモリにキャッシュする機能です。一度アクセスした静的コンテンツの内容をメモリにキャッシュすることで、二度目以降のアクセス時にファイルシステムへのアクセス回数を減らし、応答速度を向上できます。

セッション

Web アプリケーションに対する一連の作業を示す単位です。セッションは通常、Web クライアントから Web サーバへの複数のリクエストの集合から構成されます。

セッションフェイルオーバー機能

アプリケーション実行中のセッション情報を管理し、J2EE サーバで障害が発生した場合には、管理しているセッション情報をほかの J2EE サーバに引き渡す機能です。J2EE サーバで障害が発生し、ほかの J2EE サーバにリクエストが転送された場合でも、障害発生前の状態で業務を続行できます。

(タ行)

テストモード

アプリケーションの動作モードです。テストモードは、テスト用のアプリケーションを本番環境と同じように動作させるためのモードです。

デプロイ

アプリケーションや Web サービスなどを運用環境に配置し、利用できる状態にすることです。

展開ディレクトリ形式

J2EE アプリケーションを構成するプログラムやファイルを、アーカイブしないでディレクトリの状態のままデプロイする形式です。展開ディレクトリ形式の J2EE アプリケーションの場合、J2EE アプリケーションの入れ替えを容易にするリロード機能を使用できます。

統合ユーザ管理

Cosminexus システムにログインするユーザを統合管理するための仕組みです。ユーザ認証を実現する統合ユーザ管理フレームワークと、リポジトリ管理およびリソース監視を実現する統合ユーザ管理 GUI が使用できます。

統合ユーザ管理フレームワーク

統合ユーザ管理でのユーザ認証を実現するフレームワークです。Java 標準仕様 (JAAS) に従った API が提供されています。

(八行)

フレームワーク

ある機能やサービスを規定するための枠組みです。フレームワークを使用することで、業務で使用するアプリケーションを容易に作成できます。

ベーシックモード

旧バージョンとの互換性を確保するためのサーバの動作モードの一つです。
単一リソースとのトランザクション管理ができます。

(マ行)

メソッドキャンセル機能

J2EE アプリケーション実行時間の監視機能実行時間の監視機能の機能の一つです。メソッドタイムアウト機能でタイムアウトが通知されたあと、メソッドのキャンセルを実施する機能です。キャンセル時にデータベースにアクセスしている場合は、トランザクションの強制決着もします。

メソッドタイムアウト機能

J2EE アプリケーション実行時間の監視機能実行時間の監視機能の機能の一つです。監視基盤にあるリクエストのうち、一定時間内に終了しなかったメソッド処理を、タイムアウトとしてユーザに通知します。

(ヤ行)

ユーザスレッド

ユーザがサーブレットおよび JSP の中で明示して生成するスレッドのことです。

(ラ行)

ライブラリ JAR

JAR ファイル形式でパッケージされたライブラリです。J2EE アプリケーション内の J2EE コンポーネントから利用できます。

リダイレクタ

Web サーバに登録し、Web コンテナへ接続できるようにするプラグインコンポーネントです。

リデプロイ機能

アプリケーションを入れ替えるときに使用する機能です。リデプロイ機能による入れ替えは、アプリケーションのテスト時などに、修正したアプリケーションと動作中のアプリケーションを入れ替えたいときに使用します。

リロード機能

J2EE サーバ上のアプリケーションを停止させないで、アプリケーションを構成するプログラムやファイルを入れ替える機能です。リロード機能は、展開ディレクトリ形式でデプロイされたアプリケーションに対して実行できます。

ロール

コンテキストに対し、アクセス制御するときに使用される単位です。ロールはグループごとに定義されます。また、アクセス制御するコンテキストについては、そのコンテキストにアクセスするのに必要なロールが定義されます。アクセスしたユーザの持つロールがコンテキストに定義されたロールと一致した場合、そのコンテキストへのアクセスは成功します。

ログインモジュール

Java 標準仕様（JAAS）に準拠した統合ユーザ管理のユーザ認証に使用する仕組みです。

論理サーバ

Management Server の運用管理の対象になる、サーバまたはクラスタです。サーバには、Web サーバ、J2EE サーバなどがあります。クラスタとは、ある共通の機能を提供するサーバの集合です。

索引

数字

- 1.4 モード 379
- 1 台のマシンで開発・テストする場合の構成 15

A

- application.xml 編集時の注意事項 146
- Application Server 379

B

- BMP Entity Bean 379

C

- cjjsp2java コマンド実行時に出力されるメッセージ 272
- cjjsp2java コマンドで生成したソースファイルのコンパイル 276
- cjjsp2java コマンドの使用例 273
- cjjspc コマンドを使用した JSP 事前コンパイルの注意事項 262
- cjjspc コマンドを使用した JSP の事前コンパイル 260
- CMP Entity Bean 379
- CMP フィールドおよび CMR フィールドの命名規則 147
- [Connector 属性] エディタ 318
- Connector 属性ファイルのエクスポート 234
- [CONNECT] ダイアログ 93
- <context-root> タグの設定 146
- CORBA ネーミングサービス 379
- Cosminexus 379
- [Cosminexus 7] ページ 69, 293
- Cosminexus Application Development Plug-in 379
- Cosminexus cFramework 379
- Cosminexus Component Container 380
- Cosminexus Component Transaction Monitor (CTM) 380

- [Cosminexus Management Server リモート管理機能へログイン] ダイアログ 65
- Cosminexus Reliable Messaging 380
- [Cosminexus Server Plug-in] ページ 62
- Cosminexus Version 6 から Cosminexus Version 7 への移行 346
- Cosminexus Version 6 と Cosminexus Version 7 の違い 346
- Cosminexus Version 6 の開発環境の退避 349
- Cosminexus が提供するコマンドの使用 23
- Cosminexus が提供する定義ファイルの更新 24
- Cosminexus コネクタ 8, 380
- Cosminexus コネクタの組み込み確認 293
- Cosminexus コネクタの設定 302
- Cosminexus でサポートする DD について 142
- Cosminexus を再インストールしたときの注意事項 32

D

- DB_PORT (ポート番号) 315
- DB_SIZE (データベースの領域サイズ) 314
- DD (Deployment Descriptor) 380
- DD および TLD ファイルの検証 262
- DD 作成時の注意事項 147
- DD の編集 142
- Dependency Injection 380
- Developer 380
- Developer Professional との機能差 324
- Developer Standard 以外の J2EE サーバから Developer Standard の J2EE サーバへインポートする手順 336
- Developer Standard が提供するテスト用実行環境 323
- Developer Standard 使用時の注意事項 323
- Developer Standard での J2EE アプリケーションのインポート手順 336

Developer Standard と Developer Professional を比較した場合の機能差 324
 Developer Standard の J2EE サーバから Developer Standard 以外の J2EE サーバへインポートする手順 336
 Developer で開発する J2EE アプリケーションの形式 10
 Developer で提供する機能 6
 Developer による J2EE アプリケーション開発の特長 2
 Developer のアンインストール 87
 Developer のインストール 30
 Developer のディレクトリ構成 31
 <display-name> タグの設定 146
 <display-name> タグ編集時の注意事項 148, 149
 DTD に従っていない ejb-jar.xml の取り扱い 147

E

[EAR のエクスポート] ページ 213
 EAR ファイル 380
 EAR ファイル / ZIP ファイルのインポート 252
 EAR ファイル作成時の jar コマンド指定例 251
 EAR ファイルの構成例 14
 EAR ファイルの作成 226
 EAR ファイルの作成 (jar コマンド) 250
 EAR ファイルのディレクトリ構成 251
 Eclipse 380
 eclipse.ini 290
 Eclipse インストールの手順 286
 [Eclipse と MyEclipse のセットアップ] ページ 41, 48
 Eclipse のインストール 286
 Eclipse のコンソールの設定 72
 Eclipse の設定 55
 Eclipse への組み込み 289
 Eclipse へのプラグインの組み込み 292
 Eclipse または MyEclipse での操作 24
 Eclipse を使用しない J2EE アプリケーションの開発 242

Eclipse を使用しない J2EE アプリケーションの開発の概要 242
 Eclipse を使用しない開発 22
 Eclipse を使用しない場合の開発環境 242
 EJB (Enterprise JavaBeans) 380
 <ejb-client-jar> タグの使用について 148
 ejb-jar.xml ファイルのひな型の作成 114
 ejb-jar.xml 編集時の注意事項 147
 EJB-JAR ファイル 14, 380
 EJB-JAR ファイル作成時の jar コマンド指定例 250
 EJB-JAR ファイルの作成 (jar コマンド) 249
 EJB-JAR ファイルのディレクトリ構成 250
 EJB QL での 2 バイトコードの使用について 147
 EJB クライアントアプリケーション 380
 EJB クライアントアプリケーションで RMI-IIOP インタフェースを使用するときの仕組み 195
 EJB コンテナ 381
 <ejb> タグの設定 146
 [EJB プロジェクト・モジュールの定義] ページ 130
 EJB プロジェクト作成時の注意事項 113
 EJB プロジェクトで RMI-IIOP インタフェースを使用するときの仕組み 193
 EJB プロジェクトの作成手順 111
 [EJB プロジェクトの作成] ページ 111
 EJB プロジェクトのビルド 228
 EJB プロジェクトのビルドファイルの例 228
 EJB プロジェクト名の注意事項 113
 Enterprise Bean 381
 Enterprise Bean クラスの作成 182
 Entity Bean 381

H

HiRDB SQL Executer のインストール 92
 HTTP レスポンス圧縮機能 381

I

IDE (Integrated Development Environment) 381
 INPUT_FILE_DIR (入力ファイルのディレクトリ) 315

J

J2EE Application Client のデプロイと実行について 114
 J2EE アプリケーション 381
 J2EE アプリケーション開発で使用する機能 6
 J2EE アプリケーション開発の流れ 19
 J2EE アプリケーション開発の流れ (Eclipse を使用しない場合) 242
 J2EE アプリケーション実行時間の監視機能 381
 J2EE アプリケーションとリソースアダプタの属性の退避 349
 J2EE アプリケーションのアンデプロイ 208
 J2EE アプリケーションのインポート 252
 J2EE アプリケーションのエクスポート 213
 J2EE アプリケーションの開始 253
 J2EE アプリケーションの開発環境 (Eclipse を使用しない場合) 242
 J2EE アプリケーションの開発サイクル 19
 J2EE アプリケーションの開発手順 20
 J2EE アプリケーションの削除 254
 J2EE アプリケーションの実行での機能差 326
 J2EE アプリケーションの実装および設定の注意事項 329
 J2EE アプリケーションの属性設定 252
 J2EE アプリケーションの停止 253
 J2EE アプリケーションのテスト 199
 J2EE アプリケーションのテストの流れ 200
 J2EE アプリケーションのデバッグ 212
 J2EE アプリケーションのデプロイ 205
 J2EE アプリケーションのデプロイおよびアンデプロイ 205
 J2EE アプリケーションのリロード 253
 J2EE アプリケーション配布の流れ 224
 J2EE コンテナ 381
 J2EE コンポーネント 381
 J2EE サーバ 381
 J2EE サーバ起動時の動作の設定 305
 J2EE サーバの運用での機能差 326
 J2EE サーバの開始 203
 J2EE サーバの開始および停止 203
 J2EE サーバの起動構成 (デバッグモード) 203
 J2EE サーバの再起動 204
 J2EE サーバの設定 283
 J2EE サーバの選択 60
 J2EE サーバの停止 204
 J2EE サーバのデバッグ起動 218
 J2EE サーバへのデプロイ・デバッグ 3
 J2EE サーバを開始したときの [サーバー] ビュー 203
 J2EE サービス 382
 JAAS 382
 jaas.conf 382
 JAAS のコンフィグレーションファイル 382
 JAR ファイル 382
 JavaBeans 382
 JavaMail の使用例 358
 JavaVM 起動オプションの変更 269
 JavaVM のセキュリティポリシーの設定 300
 javax.mail.internet.InternetAddress 362
 javax.mail.internet.MimeMessage 362
 javax.mail.Transport 362
 Java ソースファイルの保存 267
 Java ソースファイルを利用して JSP ファイルをデバッグする 270
 Java プログラムのコンパイル (javac コマンド) 247
 JDBC 382
 JDK の確認 56
 JIS X0213:2004 に含まれる Unicode の補助文字を使用する場合の注意事項 25
 JNDI 382
 JSF (JavaServer Faces) 382
 JSF を利用した開発 238
 JSP 382

JSP コンパイル時の Java 言語仕様のバージョン指定 267
 JSP 事前コンパイル 382
 JSP 事前コンパイル機能 258
 JSP 事前コンパイル機能の前提条件 258
 JSP 事前コンパイル結果のバージョンチェック 262
 JSP 事前コンパイルで出力されるファイル 259
 JSP 事前コンパイルとチェック 257
 JSP 事前コンパイルの対象ファイル 259
 JSP 事前コンパイルの対象ファイルと出力ファイル 259
 JSP ソースのチェック 272
 JSP ファイル単位の JSP 事前コンパイル 260
 JSP ファイルの作成 159
 JSP ファイルの事前コンパイル 258
 JSP ファイルの編集 161
 JSP ファイルの文字エンコーディングの指定 269
 JSP ファイルパスの指定方法 261
 JSP ファイルまたはタグファイルから生成された Java ソースのコンパイルオプション 262
 JSP ワークディレクトリ名の変更 268
 JTA 383

K

KEDT10013-E 366
 KEDT10014-E 367
 KEDT10015-E 367
 KEDT10016-E 367
 KEDT10019-E 368
 KEDT10025-E 368
 KEDT10031-E 368
 KFPX29601-E 370
 KFPX29602-E 370
 KFPX29603-E 371
 KFPX29604-E 371
 KFPX29605-E 371
 KFPX29606-E 371
 KFPX29607-E 371

KFPX29608-E 372
 KFPX29609-E 372
 KFPX29610-E 373
 KFPX29611-E 373
 KFPX29612-E 373
 KFPX29613-E 373
 KFPX29614-E 374
 KFPX29615-E 374
 KFPX29616-E 374
 KFPX29617-E 374
 KFPX29618-E 375
 KFPX29619-E 375
 KFPX29620-E 375
 KFPX29621-E 375
 KFPX29622-E 376
 KFPX29623-E 376
 KFPX29624-E 376
 KFPX29625-E 377
 KFPX29626-E 377
 KFPX29627-E 377
 KFPX29628-E 377
 KFPX29629-E 378
 KFPX29630-E 378

L

<load-on-startup> タグ指定時の注意事項 149

M

Management Server リモート管理機能 8
 Management Server リモート管理機能からのログアウト (Server Plug-in) 65
 Management Server リモート管理機能の設定 301
 Management Server リモート管理機能への接続ホストの追加 304
 Management Server リモート管理機能へのログイン (Server Plug-in) 65
 Management Server リモート管理機能へのログインおよびログアウト (MyEclipse) 58

Management Server リモート管理機能への
ログインおよびログアウト(Server Plug-in)
64

Matisse4MyEclipse フィーチャのインス
トール 338

Matisse4MyEclipse フィーチャのインス
トール確認 343

Matisse4MyEclipse フィーチャのインス
トール手順 338

Message-driven Bean 383

<module> タグの id 属性の設定 146

MVC (Model/View/Controller) 383

MyEclipse 2, 7, 383

[MyEclipse Enterprise Workbench について
] ダイアログ 291, 343

[MyEclipse XML エディター] の構成 143

[MyEclipse XML エディター] の [設計] タ
ブの基本操作 143

[MyEclipse XML エディター] の操作方法
142

[MyEclipse XML エディター] の表示方法
142

MyEclipse 以外のツールで同じ名称の J2EE
アプリケーションをインポートしている場合
215

MyEclipse および Server Plug-in で表示され
る J2EE サーバ名 216

MyEclipse が提供する機能の一覧 7

MyEclipse が提供するスニペット 166

MyEclipse 使用時の注意事項 215

MyEclipse で J2EE アプリケーションを開発
する場合 270

MyEclipse で作成するプロジェクトの構成
108

MyEclipse での UML を利用した設計 240

MyEclipse でのプロジェクトの作成 108

MyEclipse と Server Plug-in を併用する場合
215

MyEclipse との連携によるスムーズな J2EE
アプリケーション開発 2

MyEclipse のアンインストール 321

MyEclipse のインストール 288

MyEclipse のインストール確認 291

MyEclipse のインストール準備 288

MyEclipse の起動 283

MyEclipse の機能差 333

MyEclipse の組み込み 288

MyEclipse のスニペットカテゴリ 165

MyEclipse の動作確認用サンプルプログラム
278

MyEclipse のプロキシの設定 67

MyEclipse のワークスペースに Java ソース
ファイルを反映する 270

MyEclipse を利用するための設定 296

P

PSWD (パスワード) 314

R

RDAREA_DIR (RD エリアを作成するディ
レクトリ) 315

RD エリアの削除 99

RD エリアの削除の流れ 98

RD エリアの追加 98

RD エリアの追加と削除 97

RD エリアの追加の流れ 97

RD エリアの閉塞 99

RD エリア用ファイル領域の初期化 97

RD エリアを指定したテーブル作成 98

Relationship の設定の注意事項 148

RMI-IIOP 383

RMI-IIOP インタフェースを利用した J2EE
アプリケーションの呼び出し 188

RMI-IIOP スタブが自動的に作成, 取得され
る場合の J2EE アプリケーションのディレク
トリ構成 254

RMI-IIOP スタブの用意 254

RMI-IIOP スタブを手動で取得する場合の
J2EE アプリケーションのディレクトリ構成
255

<run-as> タグと Web コンテナの認証の関連
について 149

S

<security-constraint> タグの設定 154

<security-constraint> タグの複数定義 154
 Server Plug-in 9, 383
 Server Plug-in から起動した J2EE サーバでのデバッグ 218
 Server Plug-in で J2EE サーバを起動している場合 215
 Server Plug-in の組み込み確認 294
 Server Plug-in のコンソールの設定 74
 Server Plug-in の設定 61
 Servlet 2.4 仕様でサポートされない
 web.xml の要素 154
 Servlet 2.4 仕様で追加, 変更された仕様についての注意事項 (web.xml) 154
 Session Bean 383
 Session オブジェクトの取得 358
 SFO サーバ 383
 Smart Composer 機能 384
 Stateful Session Bean 384
 Stateless Session Bean 384
 Struts 384
 Struts を利用した開発 238

T

<taglib-location> タグに指定したパスの大文字, 小文字が異なる場合の動作 153
 Timer Service 384

U

UML (Unified Modeling Language) 384
 UML とは 240
 UML の利用による開発効率の向上 3
 UML を利用した開発 22, 240
 <url-pattern> の改行コード 155
 USRID (認可識別子) 314

W

WAR ファイル 14, 384
 WAR ファイル作成時の jar コマンド指定例 249
 WAR ファイルの作成 (jar コマンド) 248
 WAR ファイルのディレクトリ構成 248
 <web-uri> タグの設定 146

web.xml の DOCTYPE 宣言の注意事項 154
 web.xml の記述内容とサーバの動作 149, 363
 web.xml の記述内容とサーブレットエンジンモードの場合のサーバの動作 363
 web.xml 編集時の注意事項 148
 web.xml 編集時の注意事項 (互換用の動作モードの場合) 363
 Web アプリケーション 384
 Web アプリケーション単位の JSP 事前コンパイル 260
 Web 画面 385
 Web コンテナ 385
 Web コンテナ単位でのエラーページのカスタマイズ 155
 [Web プロジェクト・モジュールの定義] ページ 129
 Web プロジェクト作成時の注意事項 124
 Web プロジェクトで RMI-IIOP インタフェースを使用するときの仕組み 189
 Web プロジェクトの作成 121
 Web プロジェクトの作成手順 121
 [Web プロジェクトの作成] ページ 121
 Web プロジェクトのビルド 229
 Web プロジェクトのビルドファイルの例 229
 [Web プロジェクト] ページ 191
 Web プロジェクト名の注意事項 124
 Windows Vista 使用時の注意事項 23

X

[XML ウィザード] ページ 176
 [XML スキーマ・ファイルの選択] ページ 116
 [XML ファイルの作成] ページ 115
 [XML ファイル名] ページ 116

あ

アーカイブ形式 385
 アーカイブ形式の J2EE アプリケーション 13

アーカイブ形式の J2EE アプリケーションの概要 13
 アーカイブ形式の J2EE アプリケーションの構成 14
 アーカイブ形式の J2EE アプリケーションの作成 248
 アーカイブ形式の注意事項 210
 アクション 385
 アクセスする URL パターンの定義の注意事項 148
 アセンブル 385
 アノテーション 385
 アノテーションの記述方法 197
 アノテーション有効時の EJB-JAR ファイルの作成について 114
 アノテーション有効時の WAR ファイルの作成について 124
 アブストラクトスキーマ名の指定時の注意事項 147
 アプリケーションサーバの移行 346
 アプリケーションディレクトリ 11
 アプリケーションディレクトリのインポート 252
 アプリケーションディレクトリの構成例 12
 アプリケーションディレクトリの作成 244
 アプリケーション統合属性ファイルのエディタの設定変更 75
 [アプリケーション統合属性] ページ 75
 アンインストール 320
 [アンセットアップの確認] ページ 83
 [アンセットアップの完了] ページ 84

い

移行した開発環境の操作 356
 インスタントセットアップ機能 9
 インスタントセットアップ機能実行時のエラーメッセージ一覧 366
 インスタントセットアップ機能使用時の注意事項と機能差 335
 インスタントセットアップ機能で構築した J2EE サーバの設定変更 77
 インスタントセットアップ機能で構築した環境のアンセットアップ 82

インスタントセットアップ機能の実行 23
 インスタントセットアップ機能を使用したセットアップ 34
 インスタントセットアップ機能を使用しない環境の構築手順 285
 [インストール/更新] ダイアログ 338
 インストール後のディレクトリ構成 30
 [インストール済みの JRE] ページ 56
 インストールとセットアップの流れ 28
 [インストール] ページ 341
 インプロセス HTTP サーバ 385
 [インポート] ダイアログ 136

う

ウィザード 385
 運用管理ポータルでの設定変更 345
 運用管理ポータルへのログイン 345

え

[エクスポート] ダイアログ 189
 エラーページ設定時の注意事項 149
 [エンタープライズ Java プロジェクトの作成] ページ 127
 エンタープライズアプリケーションプロジェクト作成時の注意事項 133
 エンタープライズアプリケーションプロジェクトの作成 127
 エンタープライズアプリケーションプロジェクトの作成手順 127
 エンタープライズアプリケーションプロジェクトの注意事項 133
 エンタープライズアプリケーションプロジェクトのデブロイ 283
 エンタープライズアプリケーションプロジェクトのビルド 231
 エンタープライズアプリケーションプロジェクトのビルドファイルの例 232
 エンタープライズアプリケーションプロジェクトのモジュールの変更手順 134
 エンタープライズアプリケーションプロジェクト名についての注意事項 133

か

開発環境インスタントセットアップ 385
 [開発環境インスタントセットアップ] ダイアログ 40, 47, 77, 82
 開発環境およびテスト用実行環境の構築手順 (インスタントセットアップ機能を使用しない場合) 285
 開発環境の移行手順 349
 開発環境の簡易構築 5
 開発環境の構築 295
 開発環境の設定変更 (任意の作業) 67
 開発環境のマシン構成 15
 開発用データベースの標準提供 4
 [拡張インターフェースの選択] ダイアログ 185, 186
 [拡張] タブ 164
 カスタムセットアップ 47
 カスタムログインモジュール 386
 簡易構築定義ファイルのカスタマイズ 352
 簡易構築定義ファイルの作成 300
 簡易構築定義ファイルのデフォルトの構成 350
 環境構築 / 運用時の注意事項 26
 環境変数の設定 31
 管理者特権で実行する必要がある操作 23

き

[起動] ページ 305
 [基本] タブ 164
 旧バージョンで使用していた Eclipse を引き続き使用する場合 (Eclipse 3.2.1 以前) 287
 旧バージョンで使用していた Eclipse を引き続き使用する場合 (Eclipse 3.2.2 以上) 287
 業務処理プログラム作成の流れ 180
 業務処理プログラムの作成 179
 業務処理プログラムの実装 184

く

クエリメソッドのタグを記述するときの注意事項 147
 組み込み DB プロンプトの起動 97
 組み込みデータベース 9, 90, 386

組み込みデータベースの開始 282
 組み込みデータベースの開始および停止 202
 組み込みデータベースの構築手順 307
 組み込みデータベースの削除 316
 [組み込みデータベースのセットアップ] ページ 52
 組み込みデータベースの操作 93
 組み込みデータベースのテーブルの作成の流れ 90
 組み込みデータベースの動作制御の設定 100
 組み込みデータベースへの接続 93
 組み込みデータベース用の設定ファイルの設定項目 313
 クラスファイルの変更 253
 グローバルセッション 386
 グローバルセッション ID 386
 グローバルセッション情報 386

け

ゲートウェイ指定機能を使用する場合の注意事項 151
 [検索結果] ページ 339

こ

[更新アクセス先サイト] ページ 338
 [構成およびデバッグ] ダイアログ 218
 異なるマシンで開発・テストする場合の構成 16
 コマンド実行時のログの出力 269
 [コンソール] ページ 72, 74
 コンソールへの情報出力の設定 302
 コンテキストルート 386
 コンテキストルートの URL の指定について 124
 コンテキストルートの変更手順 124
 コンテナ拡張ライブラリ 386
 [コンパイラー] ページ 57
 コンパイル対象外ファイル 263
 コンパイル対象外ファイルに指定するための前提条件 263
 コンパイル対象外ファイルを直接指定する方法 264

コンパイル対象外リストファイル 263
 コンパイル対象外リストファイルの作成 263
 コンパイル対象外リストファイルの指定 264
 コンパイル対象外リストファイルを指定する方法 263
 コンパイル不要な JSP ファイルの指定 263

さ

[サーバー・エクスプローラー] ビュー 317
 [サーバー・デプロイメント] ダイアログ 205
 サーバ管理コマンド 386
 サーバ管理コマンドと J2EE サーバの通信に使用するポートの固定方法の機能差 334
 サーバ起動・停止フック機能 386
 [サーバのセットアップ] ページ 51
 サブレット 386
 サブレットエンジンモード 363
 サブレットの作成・編集 175
 サブレットフィルタ 386
 サンプルプログラムの概要 278
 サンプルプログラムの実行手順 282
 サンプルプログラムの前提環境 279
 サンプルプログラムの提供形態と格納先 280
 サンプルプログラムのディレクトリ構成 281
 サンプルプログラムの内容 278
 サンプルプログラムのプロジェクトの構成 280
 サンプルプロジェクト (Bank) のデプロイ 284

し

システム構築 299
 システム構築時の機能差 324
 実行環境への Connector 属性ファイルのインポート 236
 実行環境への J2EE アプリケーションのインポート 235
 実行環境への J2EE アプリケーションの配布 223
 実行環境への配布 251
 実行結果リストファイルの出力 265

実行中の J2EE サーバに関する注意事項 204
 実行モードでの開始 204
 [実装されたインターフェースの選択] ダイアログ 182
 指定したエラーページが表示されたレスポンスのステータスコード 155
 条件分岐 387
 [詳細設定] タブ 63
 [新規 Java インターフェース] ダイアログ 184, 185
 [新規 Java クラス] ダイアログ 182
 [新規 Java パッケージ] ダイアログ 181
 [新規 JSP ページ] ダイアログ 159
 [新規サブレット] ダイアログ 175
 [新規接続ホスト] ダイアログ 304
 [新規] ダイアログ 114
 [新規デプロイメント] ダイアログ 206
 [新規名前空間情報] ダイアログ 119
 [新規プロジェクト] ダイアログ 111, 121, 127

す

[スニペット] ビュー 165
 スニペットを使用した JSP ファイルの編集 164
 スマートエージェント 387

せ

静的コンテンツ 387
 静的コンテンツキャッシュ機能 151, 387
 静的コンテンツキャッシュ機能を使用する場合の DD の定義 151
 セキュリティロール使用時の設定について 149, 363
 セキュリティロールリファレンスの設定について 147
 [設計] タブ 143
 セッション 387
 セッションタイムアウトの設定時の注意事項 148
 セッションフェイルオーバー機能 387
 接続先のポート番号の変更 303

接続先ホストの設定 61
 接続先ホストの設定変更 303
 [設定変更の完了] ページ 79
 [設定変更] ページ 78
 セットアップする環境の設定内容 35
 [セットアップの確認] ページ 44, 52
 [セットアップの完了] ページ 44, 53
 [セットアップの種類の選択] ページ 40, 48
 セットアップの前提条件 34

そ

[操作の選択] ページ 40, 47, 77, 82
 [ソース] タブ 143
 ソケット操作のブロックのタイムアウト設定
 変更 70
 そのほかの開発 237
 そのほかの開発の特長 22

て

定義ファイルの追加パラメタ情報の退避 349
 データベース構築・削除時のエラーメッセージ一覧 370
 データベース構築後にユーザ定義およびテーブル定義を行う方法 308
 テーブルの作成 95
 テーブルの参照 96
 [デザイン] タブ 161
 テストモード 387
 テスト用実行環境で使用する機能 9
 テスト用実行環境のシステムのチューニング
 345
 デバッグの実行 220
 デプロイ 387
 デプロイ作業ディレクトリの設定変更 69
 デプロイしたプロジェクトに対する操作の注意 208
 デプロイまたはアンデプロイできない場合の
 対処方法 216
 展開ディレクトリ形式 3, 387
 展開ディレクトリ形式でのモジュールプロ
 ジェクトの入れ替え 211

展開ディレクトリ形式で反映されるファイル
 の範囲 211
 展開ディレクトリ形式の J2EE アプリケー
 ション 10
 展開ディレクトリ形式の J2EE アプリケー
 ションの概要 10
 展開ディレクトリ形式の J2EE アプリケー
 ションの構成 12
 展開ディレクトリ形式の注意事項 210
 展開ディレクトリ形式の利用による開発効率
 の向上 3
 電子メールアドレス指定時の注意事項 362

と

統合ユーザ管理 388
 統合ユーザ管理フレームワーク 388

は

バージョン 07-20 とバージョン 07-50 以降の
 機能差 210
 [パッケージ・エクスプローラー] ビュー
 136
 パッケージの作成 181
 [パレットのカスタマイズ] ダイアログ 173

ひ

標準セットアップ 39
 ビルドファイルの作成 226
 ビルドファイルの編集・実行 227

ふ

[ファイル・システム] ページ 138
 [フィーチャー・ライセンス] ページ 340
 [フィーチャーの検査] ページ 342
 [フィーチャーの更新] ページ 338
 フィルタ機能を使用する場合の定義 156
 プラグインのアンインストール 321
 プラグインの組み込み 292
 プラグインの組み込み確認 293
 プラグインまたは MyEclipse のアンインス
 トール 23

プラグインを使用するための設定 301
 フレームワーク 388
 フレームワークの利用による生産性の向上 3
 フレームワークを利用した開発 22, 238
 [プレビュー] タブ 163
 [プロキシー] ページ 67
 プログラムのインストール 23
 プロジェクト作成の流れ 109
 プロジェクトの入れ替え 209
 プロジェクトの入れ替え機能の違い 211
 プロジェクトの入れ替え時の注意事項 210
 プロジェクトの入れ替え手順 209
 プロジェクトの作成 107
 プロジェクトのデバッグの設定 218
 プロジェクトのデプロイ 205
 [プロパティ<:プロジェクト名>] ダイア
 ログ 125
 [プロパティ] ビュー 163
 プロパティのサポート範囲 331

へ

ベーシックモード 363, 388
 別の EJB-JAR ファイルに含まれるクラスを
 参照する場合の注意事項 113

ほ

ホームインタフェースの実装 185

め

メールコンフィグレーションの取得 359
 メールコンフィグレーションの設定 358
 メールコンフィグレーションを使用しない場
 合 359
 メソッドキャンセル機能 388
 メソッドタイムアウト機能 388
 メッセージの作成 360
 メッセージの送信 361

も

[モジュール・プロジェクト] ダイアログ
 134

ゆ

ユーザインタフェースの開発の流れ 158
 ユーザスレッド 388
 ユーザ定義およびテーブル定義後にデー
 タベースを構築する方法 311

ら

ライブラリ JAR 388
 [ライブラリー・セット] ページ 297
 ライブラリパスの確認手順 296
 ライブラリパスの変更手順 297

り

リクエストで使用する場合の注意事項 25
 リソースアダプタのインポート 136
 リソースアダプタのプロパティ設定 317
 リダイレクタ 389
 リダイレクトによるエラーページの生成 156
 リデプロイ機能 389
 リモートインタフェースの実装 184
 [リモート管理] ページ 71
 リロード機能 4, 11, 389

る

[ルート要素の選択] ページ 118

ろ

ローカル変数情報の出力の設定 57
 ロール 389
 [ログイン - リモート管理] ダイアログ 59
 ログインモジュール 389
 論理サーバ 389

ソフトウェアマニュアルのサービス ご案内

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

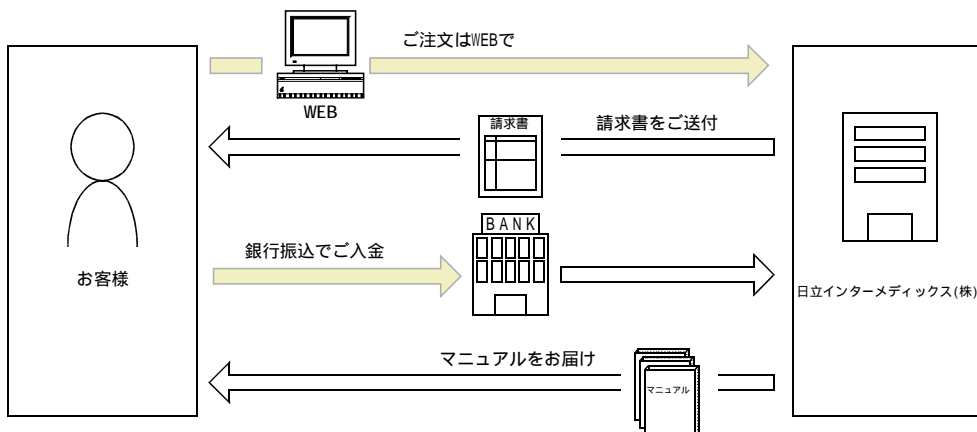
(1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえ WEB からご注文ください。ご注文先は日立インターメディアックス(株)となります。

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

入金確認後 7 日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。