

Cosminexus アプリケーションサーバ V8

概説

概説書

3020-3-U01-40

対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) ¹ , Windows Server 2003 R2 (x64) ¹ , Windows Server 2008 x86 , Windows Server 2008 x64 ¹ , Windows Server 2008 R2 ¹

P-2443-7B84 uCosminexus Application Server Standard-R 08-70

P-2443-7D84 uCosminexus Application Server Standard 08-70

P-2443-7K84 uCosminexus Application Server Enterprise 08-70

P-2443-7M84 uCosminexus Web Redirector 08-70

P-2443-7S84 uCosminexus Service Platform 08-70 ²

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Vista , Windows XP , Windows 7 (32bit) , Windows 7 (x64) ¹

P-2443-7E84 uCosminexus Developer Standard 08-70

P-2443-7F84 uCosminexus Developer Professional 08-70

P-2443-7T84 uCosminexus Service Architect 08-70 ²

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) ¹ , Windows Server 2003 R2 (x64) ¹ , Windows Server 2008 x86 , Windows Server 2008 x64 ¹ , Windows Server 2008 R2 ¹ , Windows Vista , Windows XP , Windows 7 (32bit) , Windows 7 (x64) ¹

P-2443-7H84 uCosminexus Client 08-70

適用 OS : Windows Server 2003 (x64) , Windows Server 2003 R2 (x64) , Windows Server 2008 x64 , Windows Server 2008 R2

P-2943-7B84 uCosminexus Application Server Standard-R 08-70

P-2943-7D84 uCosminexus Application Server Standard 08-70

P-2943-7K84 uCosminexus Application Server Enterprise 08-70

P-2943-7S84 uCosminexus Service Platform 08-70 ²

適用 OS : AIX 5L V5.3 , AIX V6.1 , AIX V7.1

P-1M43-7D81 uCosminexus Application Server Standard 08-70 ²

P-1M43-7K81 uCosminexus Application Server Enterprise 08-70 ²

P-1M43-7S81 uCosminexus Service Platform 08-70 ²

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

P-1J43-7D81 uCosminexus Application Server Standard 08-70

P-1J43-7K81 uCosminexus Application Server Enterprise 08-70

P-1J43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform (x86) , Red Hat Enterprise Linux 5 (x86) , Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) , Red Hat Enterprise Linux 5 (AMD/Intel 64) , Red Hat Enterprise Linux Server 6 (32-bit x86) , Red Hat Enterprise Linux Server 6 (64-bit x86_64)

P-9S43-7B81 uCosminexus Application Server Standard-R 08-70 ²

P-9S43-7D81 uCosminexus Application Server Standard 08-70 ²

P-9S43-7K81 uCosminexus Application Server Enterprise 08-70 ²

P-9S43-7M81 uCosminexus Web Redirector 08-70 ²

P-9S43-7S81 uCosminexus Service Platform 08-70 ²

注 1 WOW64 (Windows On Windows 64) 環境だけで使用できます。

注 2 この製品については、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」をご確認ください。

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、米国およびその他の国における International Business Machines Corporation の商標です。

AIX 5L は、米国およびその他の国における International Business Machines Corporation の商標です。

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。

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。

HP-UX は、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。

Internet Explorer は、米国 Microsoft Corporation の米国およ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その他の国における Intel Corporation の商標です。

J2EE は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Java は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JDBC は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JDK は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JSP は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

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

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft Internet Information Services は、米国 Microsoft Corporation 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、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Oracle 及び Oracle 10g は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Oracle 及び Oracle9i は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Oracle 及び Oracle Database 10g は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Oracle 及び Oracle Database 11g は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

SOAP (Simple Object Access Protocol) は、分散ネットワーク環境において XML ベースの情報を交換するための通信プロトコルの名称です。

Solaris は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社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。SPARC 商標がついた製品は、米国 Sun Microsystems, Inc. が開発したアーキテクチャに基づくものです。

SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Sun は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Sun Microsystems は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

VMware は、VMware, Inc. の米国および各国での登録商標または商標です。

VMware および ESX は、VMware, Inc. の米国および各国での登録商標または商標です。

VMware および VMware vSphere は、VMware, Inc. の米国および各国での登録商標または商標です。

W3C は、World Wide Web Consortium の商標 (多数の国において登録された) です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

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/>).

HITACHI
Inspire the Next

◎ 株式会社 日立製作所



マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

製品名	表記			
Microsoft(R) Cluster Service	Microsoft Cluster Service			
Microsoft(R) Internet Information Services 6.0	Microsoft IIS 6.0	Microsoft IIS		
Microsoft(R) Internet Information Services 7.0	Microsoft IIS 7.0			
Microsoft(R) Internet Information Services 7.5	Microsoft IIS 7.5			
Microsoft(R) SQL Server 2000	SQL Server 2000	SQL Server		
Microsoft(R) SQL Server 2005	SQL Server 2005			
Microsoft(R) SQL Server 2008	SQL Server 2008			
Microsoft(R) SQL Server 2000 Driver for JDBC	SQL Server 2000 Driver for JDBC	SQL Server の JDBC ドライバ		
Microsoft(R) SQL Server 2005 JDBC Driver	SQL Server 2005 JDBC Driver			
Microsoft(R) SQL Server JDBC Driver 2.0	SQL Server JDBC Driver			
Microsoft(R) SQL Server JDBC Driver 3.0				
Microsoft(R) Windows(R) 7 Ultimate (32bit)	Windows 7 (32bit)	Windows 7	Windows	
Microsoft(R) Windows(R) 7 Enterprise (32bit)				
Microsoft(R) Windows(R) 7 Professional (32bit)				
Microsoft(R) Windows(R) 7 Ultimate (x64)	Windows 7 (x64)			
Microsoft(R) Windows(R) 7 Enterprise (x64)				
Microsoft(R) Windows(R) 7 Professional (x64)				
Microsoft(R) Windows Server(R) 2003 , Enterprise Edition 日本語版 ¹	Windows Server 2003 Enterprise Edition	Windows Server 2003		
Microsoft(R) Windows Server(R) 2003 , Standard Edition 日本語版	Windows Server 2003 Standard Edition			
Microsoft(R) Windows Server(R) 2003 R2 , Enterprise Edition 日本語版 ¹	Windows Server 2003 R2 Enterprise Edition	Windows Server 2003 R2		
Microsoft(R) Windows Server(R) 2003 R2 , Standard Edition 日本語版	Windows Server 2003 R2 Standard Edition			
Microsoft(R) Windows Server(R) 2003 , Enterprise x64 Edition 日本語版 ¹	Windows Server 2003 Enterprise x64 Edition	Windows Server 2003 (x64)		

製品名	表記	
Microsoft(R) Windows Server(R) 2003 , Standard x64 Edition 日本語版	Windows Server 2003 Standard x64 Edition	
Microsoft(R) Windows Server(R) 2003 R2 , Enterprise x64 Edition 日本語版 ¹	Windows Server 2003 R2 Enterprise x64 Edition	Windows Server 2003 R2 (x64)
Microsoft(R) Windows Server(R) 2003 R2 , Standard x64 Edition 日本語版	Windows Server 2003 R2 Standard x64 Edition	
Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版 ²	Windows Server 2008 x86	Windows Server 2008
Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版		
Microsoft(R) Windows Server(R) 2008 Enterprise 日本語版 ²	Windows Server 2008 x64	
Microsoft(R) Windows Server(R) 2008 Standard 日本語版		
Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版 ²		
Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版	Windows Server 2008 R2	
Microsoft(R) Windows Vista(R) Business	Windows Vista Business	
Microsoft(R) Windows Vista(R) Enterprise	Windows Vista Enterprise	
Microsoft(R) Windows Vista(R) Ultimate	Windows Vista Ultimate	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP	
Windows(R) Internet Explorer(R)	Internet Explorer	
Windows Server(R) Failover Cluster	Windows Server Failover Cluster	

注 1 これらの製品を総称して , Windows Server 2003 Enterprise Edition と表記することがあります。

注 2 これらの製品を総称して , Windows Server 2008 Enterprise と表記することがあります。

発行

2011 年 7 月 3020-3-U01-40

著作権

All Rights Reserved. Copyright (C) 2008, 2011, Hitachi, Ltd.

変更内容

変更内容 (3020-3-U01-40) uCosminexus Application Server Enterprise 08-70 , uCosminexus Application Server Standard 08-70 , uCosminexus Application Server Standard-R 08-70 , uCosminexus Client 08-70 , uCosminexus Developer Professional 08-70 , uCosminexus Developer Standard 08-70 , uCosminexus Service Architect 08-70 , uCosminexus Service Platform 08-70 , uCosminexus Web Redirector 08-70

追加・変更内容	変更箇所
仮想化したサーバ上での J2EE アプリケーション実行環境に関する説明を変更した。また、08-50 モードの仮想サーバマネージャに関する説明を参考情報に移動した。	1.3
アプリケーション開発で使用する Eclipse のプラグインに WTP を追加した。	2.2.3 , 2.3.2 , 3.4 , 6.2 , 6.3.2
次の製品の適用 OS に AIX , HP-UX (IPF) を追加した。 <ul style="list-style-type: none"> • uCosminexus Application Server Enterprise • uCosminexus Application Server Standard • uCosminexus Service Platform 	2.4
次の製品の適用 OS に Red Hat Enterprise Linux Server 6 (32-bit x86) , Red Hat Enterprise Linux Server 6 (64-bit x86_64) を追加した。 <ul style="list-style-type: none"> • uCosminexus Application Server Enterprise • uCosminexus Application Server Standard • uCosminexus Application Server Standard-R • uCosminexus Service Platform • uCosminexus Web Redirector 	2.4
対応する XML , Web サービス関連の標準仕様を変更した。	付録 A

uCosminexus Application Server Enterprise 08-53 , uCosminexus Application Server Standard 08-53 , uCosminexus Application Server Standard-R 08-53 , uCosminexus Client 08-53 , uCosminexus Developer Professional 08-53 , uCosminexus Developer Standard 08-53 , uCosminexus Service Architect 08-53 , uCosminexus Service Platform 08-53 , uCosminexus Web Redirector 08-53

追加・変更内容	変更箇所
次の製品の適用 OS から AIX , HP-UX , Linux (IPF) を削除した。 <ul style="list-style-type: none"> • uCosminexus Application Server Enterprise • uCosminexus Application Server Standard • uCosminexus Service Platform 	2.4
Microsoft IIS 7.0 および Microsoft IIS 7.5 に対応した。	2.4
HiRDB Version 9 に対応した。	7.1
SQL Server 2008 に対応した。これに伴い、使用できる JDBC ドライバに SQL Server JDBC Driver 2.0 ,および SQL Server JDBC Driver 3.0 を追加した。	7.1

追加・変更内容	変更箇所
対象製品として uCosminexus Application Server Standard-R を追加した。	-

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 (3020-3-U01-20) uCosminexus Application Server Enterprise 08-50 , uCosminexus Application Server Standard 08-50 , uCosminexus Client 08-50 , uCosminexus Developer Professional 08-50 , uCosminexus Developer Standard 08-50 , uCosminexus Service Architect 08-50 , uCosminexus Service Platform 08-50 , uCosminexus Web Redirector 08-50

追加・変更内容
Cosminexus JMS プロバイダの機能を追加した。
仮想サーバ上で J2EE アプリケーションの実行環境を構築・運用できるようにした。
OpenTP1 からアプリケーションサーバを呼び出す機能 (TP1 インバウンド連携機能) を追加した。
JDK 6 に対応した。
<p>次の製品の適用 OS に Windows Server 2008 R2 を追加した。</p> <ul style="list-style-type: none"> • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise • uCosminexus Web Redirector • uCosminexus Client
<p>次の製品の適用 OS から Solaris を削除した。</p> <ul style="list-style-type: none"> • uCosminexus Application Server Standard • uCosminexus Application Server Enterprise
<p>次の製品の適用 OS に Windows 7 を追加した。</p> <ul style="list-style-type: none"> • uCosminexus Developer Standard • uCosminexus Developer Professional • uCosminexus Client • uCosminexus Operator
ホスト単位管理モデルを対象とした系切り替え構成を追加した。
スレッドの非同期並行処理の機能を追加した。
Server Plug-in の記述を削除し、代わりに運用管理ポータルまたは cosminexus.xml を使用方法に変更した。
Windows Server Failover Cluster に対応した。
メッセージのプリフィクス「KDAN」「KDJR」を追加した。
uCosminexus Service Platform の適用 OS に Windows Server 2008 R2 を追加した。
uCosminexus Service Architect の適用 OS に Windows 7 を追加した。

はじめに

このマニュアルは、Cosminexus（コズミネクサス）のアプリケーションサーバの概要について説明したものです。アプリケーションサーバの製品構成と、これらの製品で実現できることについて説明しています。また、アプリケーションサーバのマニュアル体系についても説明しています。

アプリケーションサーバでは、次に示すプログラムプロダクトを使用してシステムの構築・運用、またはアプリケーションの開発をします。

- uCosminexus Application Server Enterprise
- uCosminexus Application Server Standard
- uCosminexus Application Server Standard-R
- uCosminexus Client
- uCosminexus Developer Professional
- uCosminexus Developer Standard
- uCosminexus Service Architect
- uCosminexus Service Platform
- uCosminexus Web Redirector

対象読者

このマニュアルは、アプリケーションサーバを導入してシステムを設計・構築または運用する方、およびアプリケーションサーバで動作するアプリケーションを開発する方を対象としています。

次の内容を理解されていることを前提としています。

- Windows またはご使用の UNIX に関する知識
- Java EE に関する知識

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。

はじめに

●サーバホスト



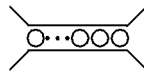
●クライアントホスト



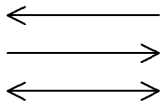
●データベース



●キュー



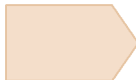
●制御の流れ



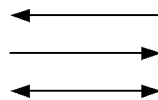
●データの流れ



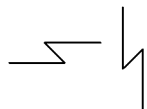
●工程、作業項目の流れ



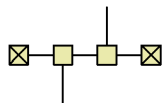
●その他の流れ



●通信回線



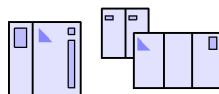
●バス型のLAN



●ネットワーク



●アプリケーションサーバ以外のシステム



●アプリケーション、プログラム



目次

1	アプリケーションサーバの概要	1
1.1	Cosminexus のアプリケーションサーバとは	2
1.2	アプリケーション実行環境とアプリケーション開発環境	3
1.2.1	J2EE アプリケーション実行環境の概要と特長	3
1.2.2	バッチアプリケーション実行環境の概要と特長	7
1.2.3	アプリケーション開発環境の概要と特長	8
1.2.4	アプリケーションの実行環境と開発環境の関係	9
1.3	仮想化したサーバ上での J2EE アプリケーション実行環境	11
1.4	Cosminexus ビジネスプロセス管理 / エンタープライズサービスバスとの関係	15
2	アプリケーションサーバの製品構成	19
2.1	製品の分類と特長	20
2.1.1	実行環境を構築する製品	20
2.1.2	開発環境を構築する製品	20
2.1.3	オプション製品	21
2.2	製品の概要	22
2.2.1	Application Server Standard	22
2.2.2	Application Server Enterprise	24
2.2.3	Developer Standard	28
2.2.4	Developer Professional	29
2.2.5	uCosminexus Client	29
2.2.6	uCosminexus Web Redirector	30
2.2.7	uCosminexus Operator	31
2.3	構成ソフトウェア	33
2.3.1	製品と構成ソフトウェアの対応	33
2.3.2	構成ソフトウェアの機能概要	35
2.3.3	実行環境の構成ソフトウェアの位置づけ (Application Server)	39
2.3.4	開発環境の構成ソフトウェアの位置づけ (Developer)	44
2.4	動作環境	46
3	システムのライフサイクル	51
3.1	実行環境および開発環境のライフサイクル	52

3.2 システムの仕様（機能）検討	54
3.3 システム設計と運用設計	60
3.4 アプリケーションの開発	61
3.5 システムの構築	62
3.6 システムの運用と保守	63

4

J2EE アプリケーションを実行する	65
4.1 J2EE アプリケーションの実行環境の特長	66
4.1.1 標準仕様への対応	66
4.1.2 システムの安定稼働の実現	66
4.1.3 可用性と耐障害性の向上	69
4.1.4 システム導入および拡張の容易化	72
4.1.5 システム監査によるシステムのセキュリティ確保	73
4.1.6 業務効率を向上させる運用管理の実現	76
4.1.7 Web サービスへの対応	78
4.1.8 信頼性の高い非同期通信の実現	79
4.2 J2EE アプリケーションの実行環境の種類	82
4.3 J2EE アプリケーションの実行環境を構築・運用する	83
4.3.1 システム構成の種類	83
4.3.2 スケールアウト構成のシステムの構築・運用	85
4.3.3 フロントエンド/バックエンド構成のシステムの構築・運用	86
4.3.4 構築した環境のチューニング	87
4.4 Web サービスに対応したアプリケーションの実行環境を構築・運用する	89
4.4.1 Web サービスの実行環境	89
4.4.2 Web サービスセキュリティの技術を利用した SOAP アプリケーションの実行環境	91
4.5 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する	93
4.5.1 Cosminexus JMS プロバイダを利用した実行環境	94
4.5.2 Cosminexus RM を利用した実行環境	95
4.6 導入から運用までの流れ	98

5

バッチアプリケーションを実行する	105
5.1 バッチアプリケーションの実行環境の特長	106
5.1.1 オープン環境でのバッチジョブの実行	106

5.1.2	コネクションプール/ステートメントプールを使用したデータベースアクセスの高速化	107
5.1.3	リソース排他状態でのフルガーベージコレクション実行を抑止	107
5.1.4	バッチアプリケーションのジョブスケジューリング	108
5.1.5	そのほかの特長	109
5.2	バッチアプリケーションの実行環境を構築・運用する	110
5.2.1	システム構成の種類	110
5.2.2	バッチアプリケーションの実行環境の構築・運用	112
5.3	導入から運用までの流れ	114

6

	アプリケーションを開発する	117
6.1	アプリケーション開発の特長	118
6.2	開発するアプリケーションの種類	120
6.3	J2EE アプリケーションを開発する	121
6.3.1	Developer が提供する MyEclipse を使用する	121
6.3.2	そのほかにも使用できる機能	123
6.4	Web サービスに対応したアプリケーションを開発する	124
6.4.1	JAX-WS 仕様に従った Web サービスを開発する機能を使用する	124
6.4.2	Web サービスセキュリティ技術を使用する	126
6.5	メッセージを使用した非同期通信のアプリケーションを開発する	127
6.6	バッチアプリケーションを開発する	128
6.7	アプリケーション開発の流れ	129
6.7.1	J2EE アプリケーション開発の流れ	129
6.7.2	バッチアプリケーション開発の流れ	134

7

	ほかの製品との連携	137
7.1	データベースとの連携	138
7.2	JP1 との連携	140
7.3	クラスタソフトウェアとの連携	141

8

	アプリケーションサーバのマニュアル体系	143
8.1	マニュアル体系	144
8.2	初期導入時のマニュアルの読書手順	150
8.3	物理サーバ上で実行環境を構築・運用する場合の読書手順	151

8.3.1	J2EE アプリケーションの実行環境を構築・運用する場合に参照するマニュアル	153
8.3.2	バッチアプリケーションの実行環境を構築・運用する場合に参照する マニュアル	155
8.3.3	Web サービスの実行環境を構築・運用する場合に参照するマニュアル	156
8.3.4	メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する場 合に参照するマニュアル	157
8.4	仮想サーバ上で実行環境を構築・運用する場合の読書手順	158
8.4.1	仮想サーバ上で実行環境を構築・運用する場合に参照するマニュアル	158
8.5	アプリケーションを開発する場合の読書手順	160
8.5.1	J2EE アプリケーションまたはバッチアプリケーションを開発する場合に参照する マニュアル	161
8.5.2	Web サービスを開発する場合に参照するマニュアル	163
8.5.3	メッセージを利用した非同期通信アプリケーションを開発する場合に参照する マニュアル	164

付録 165

付録 A	アプリケーションサーバが対応する標準仕様	166
付録 B	このマニュアルの参考情報	167
付録 B.1	関連マニュアル	167
付録 B.2	このマニュアルでの表記	170
付録 B.3	英略語	175
付録 B.4	KB (キロバイト) などの単位表記について	176
付録 C	用語解説	177

索引 183

1

アプリケーションサーバの概要

この章では、Cosminexus のアプリケーションサーバの概要、および環境ごとの特長について説明します。

アプリケーションサーバは、業務システムの中核に位置し、アプリケーションを実行する基盤となる製品です。標準技術である Java EE 5 に準拠した実行環境を構築・運用できます。

さらに、実行環境上で実行するアプリケーションを効率良く開発する環境も構築できます。

1.1 Cosminexus のアプリケーションサーバとは

1.2 アプリケーション実行環境とアプリケーション開発環境

1.3 仮想化したサーバ上での J2EE アプリケーション実行環境

1.4 Cosminexus ビジネスプロセス管理 / エンタープライズサービスバスとの関係

1.1 Cosminexus のアプリケーションサーバとは

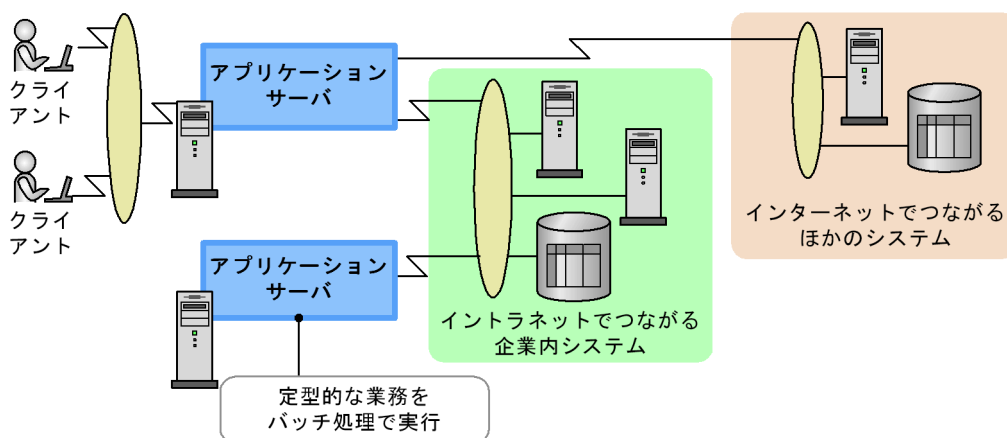
情報社会が急速に変化している現在、ビジネスを成長させるためには市場ニーズに柔軟に対応できるインフラが必要です。このようなインフラを実現するためには、拡張性に優れた情報システムの構築が必須であり、情報システムの優劣はビジネスの成功を左右する程の影響力を持ちます。さらに、社会変化のスピードに合わせて業務のライフサイクルが短くなっていくのに伴って、業務の変化に柔軟に対応でき、かつ既存資産をむだなく再利用できることも、情報システムの必須要件になってきています。

このような要件にこたえるシステムを実現するのが、Cosminexus のアプリケーションサーバです。

アプリケーションサーバは、ビジネス環境やビジネス戦略の急激な変化に柔軟に対応できる、信頼性と拡張性の高い情報システムの基盤となる製品です。標準技術である Java EE に準拠したアプリケーションの実行基盤を構築できるので、Java EE の特長である柔軟性、信頼性の高いシステムを効率的に開発・運用できるようになります。

アプリケーションサーバの概要を次の図に示します。

図 1-1 アプリケーションサーバの概要



アプリケーションサーバは、クライアントからオンラインで送信されたリクエストを基に業務を実行する基盤、および定期的な業務をバッチ処理で実行するための基盤になります。

また、アプリケーションサーバは、Java のほか、CORBA などの業界標準に準拠しています。このため、ほかのシステムとの連携や、ほかのシステムからの移植も円滑に実現できます。

1.2 アプリケーション実行環境とアプリケーション開発環境

この節では、アプリケーションサーバの特長について説明します。

アプリケーションサーバは、サービスを実行するための基盤（サービスデプロイメント）となる製品です。性能と信頼性を確保しながらユーザにサービスを提供する、アプリケーション実行環境を構築します。また、サービスとなるアプリケーションを開発する環境も構築できます。

アプリケーションの実行環境とアプリケーションの開発環境の特長を次に示します。

アプリケーション実行環境

アプリケーションを実行することでユーザにサービスを提供する基盤となる環境です。

アプリケーション実行環境では、次の2種類の業務を実行できます。

- オンライン業務（オンライン処理）
インターネットやイントラネット上のユーザから送信された要求を随時処理する形式の業務です。オンライン業務では、Java EE の技術を使用して開発されたアプリケーションを実行します。この環境を、J2EE アプリケーション実行環境といいます。
- バッチ業務（バッチ処理）
定型的な業務を決まった時間にまとめて処理する形式の業務です。従来メインフレーム上などで実行されていたバッチジョブを、オープン環境の技術である Java を使用して実現できます。Java で開発したバッチジョブを実行するためのアプリケーションをバッチアプリケーションといいます。バッチアプリケーションを実行するための環境をバッチアプリケーション実行環境といいます。

アプリケーション開発環境

アプリケーション実行環境上で動作するアプリケーションを開発する環境です。アプリケーションサーバでは、実行環境上で動作するアプリケーションの設計・開発からデバッグまでを統括的に支援する開発環境を構築・運用できます。

1.2.1 J2EE アプリケーション実行環境の概要と特長

ここでは、J2EE アプリケーション実行環境の概要と特長について説明します。詳細については、「4.1 J2EE アプリケーションの実行環境の特長」を参照してください。

(1) アプリケーション実行環境の概要

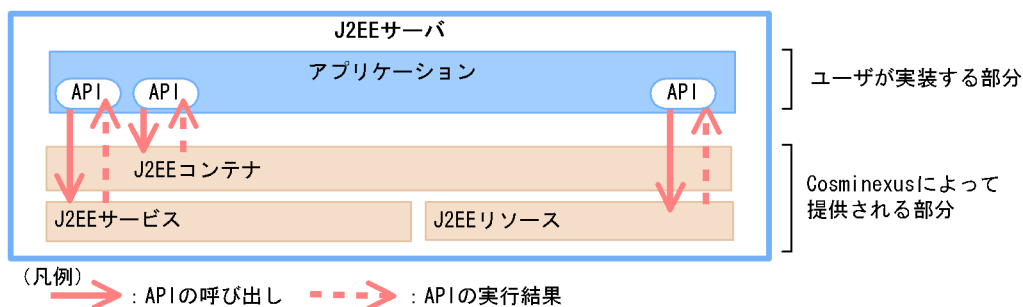
アプリケーションサーバは、標準仕様である Java EE に準拠したアプリケーションの実行環境を構築します。Java EE に準拠したアプリケーションを実行する機能を持つサーバプロセスを、J2EE サーバといいます。

1. アプリケーションサーバの概要

J2EE サーバは、J2EE コンテナ、J2EE サービス、J2EE リソースなどの Java EE で規定された仕様に従って、ユーザが開発した J2EE アプリケーションを実行するために必要な機能を提供します。例えば、J2EE サーバでは、トランザクション管理やセキュリティ管理など、複数の業務に共通する処理を実行する機能を、J2EE サービスや J2EE リソースによって提供しています。アプリケーション開発時には、アプリケーション内で J2EE コンテナ、J2EE サービス、J2EE リソースなどで提供されている API を呼び出すことで、煩雑なコーディングをしないで、複数の業務で共通の処理を実現できます。

アプリケーションと J2EE サーバの関係を次の図に示します。

図 1-2 アプリケーションと J2EE サーバの関係



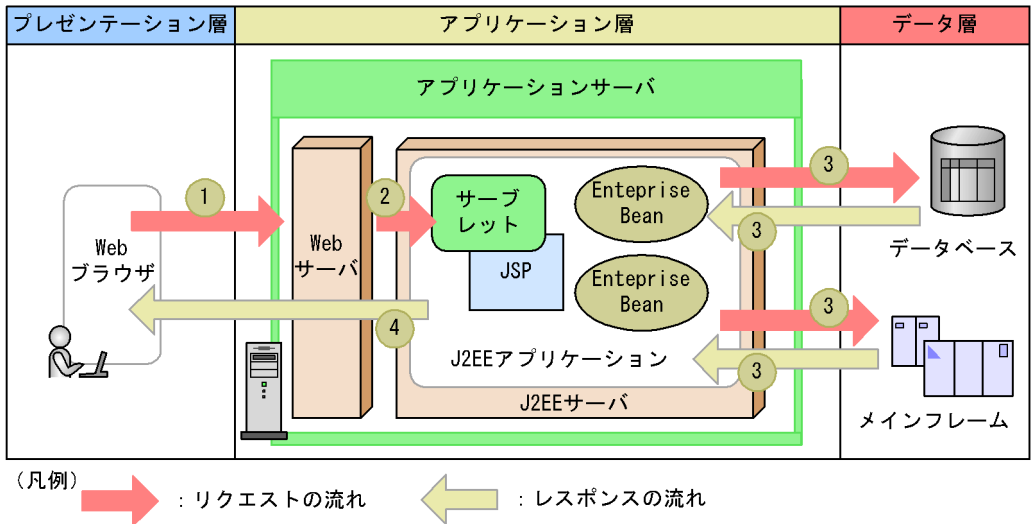
アプリケーションサーバは、J2EE サーバを中心とした、アプリケーションの実行環境になるサーバ基盤です。情報システムの間際に位置し、ユーザの要求とデータベースなど、業務システム間の処理の受け渡しをします。

アプリケーションサーバ上では、ユーザの要求に応じたサービスを提供するための J2EE アプリケーションが実行されます。J2EE アプリケーションとは、J2EE サーバ上で動作する、実現したい業務内容に応じて開発されたアプリケーションです。

J2EE アプリケーションは、ユーザからの要求を受け付け、処理を実行し、結果をユーザに返します。また、処理内容によっては、データベースやメインフレームなどのほかのシステムとデータをやり取りして、必要な情報を取得して処理を実行します。

アプリケーションサーバを中心としたシステムでのリクエスト処理の流れを次の図に示します。

図 1-3 アプリケーションサーバを中心としたシステムでのリクエスト処理の流れ



図で示した流れについて説明します。

1. ユーザが Web ブラウザ上で処理を実行すると、Web ブラウザからアプリケーションサーバに対してリクエストが送信されます。Web ブラウザからのリクエストは、アプリケーションサーバの一部である Web サーバが受け付けます。
2. アプリケーションサーバ内で、ユーザのリクエストに対応する J2EE アプリケーションが実行されます。J2EE アプリケーションは、Web サーバから転送されたリクエストを受け付けるためのプログラム（サーブレット、JSP）や、業務処理を実行するプログラム（Enterprise Bean など）で構成されています。
3. J2EE アプリケーションは、必要に応じて、ユーザのリクエストを処理するためにデータベースやほかのシステムにアクセスします。
4. 業務処理が完了したら、Web サーバ経由で Web ブラウザにレスポンスが送信されます。Web ブラウザ上で表示するための画面は、J2EE アプリケーション内のサーブレットや JSP で生成されます。生成された内容が、処理結果としてユーザが操作している Web ブラウザ上に表示されます。

アプリケーションサーバでは、安定稼働性や耐障害性が高く、優れたパフォーマンスを実現するシステムを構築できます。また、作業を円滑に実行するための機能によって、効率の良いシステム構築・システム運用を実現できます。

(2) J2EE アプリケーション実行環境の特長

J2EE アプリケーション実行環境の特長について説明します。

標準仕様への対応

Java EE に準拠した J2EE サーバなど、標準仕様に準拠したアプリケーションサーバの構築・運用を実現できます。

1. アプリケーションサーバの概要

システムの安定稼働の実現

Web アプリケーションに対する流量制御や、Enterprise Bean に対する流量制御、優先制御および負荷分散を実現できます。これによって、リクエスト数の急激な変動や一部のアプリケーションの高負荷状態にも柔軟に対応し、システムの安定稼働を実現できます。

可用性と耐障害性の向上

独自のメモリ空間を使用する明示管理ヒープ機能によって、フルガーベージコレクションの発生を抑制し、システムが停止する回数および時間を低減します。また、システムの稼働状態を監視することで障害の予兆を検知して障害を未然に防いだり、障害が発生した場合に速やかに自動回復したりする機能も備えています。これらの機能によって、可用性と耐障害性の高いシステムを実現できます。

システム導入および拡張の容易化

システムで設定が必要なパラメータを一括設定できます。また、アプリケーションの実行時情報を必要な部分だけ Cosminexus アプリケーション属性ファイル

(cosminexus.xml) に記述できます。この XML ファイルをアプリケーションに含めることで、アプリケーションのインポート後、すぐに業務を開始できます。これらの機能によって、システムのスケールアウト、スケールインにもスムーズに対応できます。

また、本番運用をする環境とは別のテスト環境でテストやチューニングを実施した場合に、チューニング済みの定義をまとめてファイルに出力できます。これを利用して本番環境を構築することで、テスト環境から本番環境への、構築済みのシステムのスムーズな移行を実現できます。

さらに、UNIX 上でアプリケーションサーバを使用している場合、稼働中の実行環境の情報を退避して、同じマシンに別の実行環境を構築できます。この機能は、本番稼働中の実行環境をバージョンアップしたり、本番稼働中の実行環境にパッチを適用したりする場合に有効です。バージョンアップやパッチ適用後に問題が発生した場合などに、速やかに元の環境を復元できます。

システム監査によるシステムのセキュリティ確保

組織の内部統制強化のニーズなどにこたえるため、「だれが」「いつ」「何を実施したか」という情報をログとして残しておけます。このログを使用すると、システムが正しい運用手順で管理・運用されていることを監査・証明できます。

また、データベースの機能と連携して、データベースアクセスの監査に有効な情報も出力できます。出力された情報を基に、アプリケーションサーバを経由して実行されたデータベースへのアクセスが、どのクライアントが送信したリクエストで実行されたのかを判別できます。これによって、データベースアクセスを伴う操作が、だれによって実施されたのかを監査・証明できます。

業務効率を向上させる運用管理の実現

複数のマシンで構成されるシステムを一つのマシンから一括運用できます。また、ほかの運用管理プログラムと連携することで、アプリケーションサーバ以外のシステムも含めた自動運用を実現できます。

さらに、サーバ上でのリクエスト処理の流れをトレースする性能解析トレース機能や、

クライアントの観点で性能を分析するクライアント性能モニタ機能によって、効率良くシステムのボトルネックを洗い出すことができます。

Web サービスへの対応

Web サービスに対応した実行環境を構築できます。また、Web サービスを開発する機能も備えています。

信頼性の高い非同期通信の実現

メッセージを使用した非同期通信を実現するための実行環境を構築し、送信側と受信側での処理の非同期実行を実現できます。メッセージの実装には、アプリケーションサーバが提供する JMS インタフェースを利用できます。

1.2.2 バッチアプリケーション実行環境の概要と特長

ここでは、バッチアプリケーション実行環境の概要と特長について説明します。詳細については、「5.1 バッチアプリケーションの実行環境の特長」を参照してください。

(1) バッチアプリケーション実行環境の概要

バッチアプリケーションの実行環境では、Java プログラムとして開発されたバッチアプリケーションを実行します。従来メインフレーム上で動作していたバッチジョブのオープン環境への移行を図れます。

バッチアプリケーションはサーバプロセス上で動作します。バッチアプリケーションを実行する機能を持つサーバプロセスを、バッチサーバといいます。バッチサーバを使用することによって、アプリケーションの実行ごとに JavaVM を起動するコストを抑えられます。これによって、効率良くバッチジョブを実行できます。

(2) バッチアプリケーション実行環境の特長

バッチアプリケーション実行環境の特長について説明します。

オープン環境でのバッチジョブの実行

Java の特長である柔軟性や運用容易性を兼ね備えたバッチジョブの実行環境を構築・運用できます。また、JP1/AJS2 や uCosminexus Batch Job Execution Server などの関連プログラムと連携することで、ジョブの自動実行や、メインフレームで実現していた内容に類似した処理の制御を実現できます。

コネクションプール/ステートメントプールを使用したデータベースアクセスの高速化

バッチアプリケーション実行環境では、コネクションプールやステートメントプールなどの機能を使用できます。これによって、バッチアプリケーションからデータベースへのアクセスを高速にできます。なお、コネクションプールおよびステートメントプールの機能は、リソースアダプタを使用して実現します。

リソース排他状態でのフルガーベージコレクション実行を抑制

バッチアプリケーション実行環境では、バッチサーバ上でフルガーベージコレクショ

1. アプリケーションサーバの概要

ンを実行するタイミングを制御できます。この制御では、リソースを排他していないタイミングを見計らってフルガーベージコレクションを実行して、不要なオブジェクトを解放します。これによって、意図しないタイミングでフルガーベージコレクションが実行されることを抑止できます。

この機能は、バッチアプリケーションとオンラインアプリケーションが同じリソースにアクセスする場合に有効です。

バッチアプリケーションのジョブスケジューリング

バッチアプリケーションは、バッチサーバごとに一つ実行できます。バッチアプリケーションのジョブスケジューリング機能を使用すると、バッチサーバの数以上にバッチアプリケーション実行要求があった場合に、実行要求を待たせることができます。これによって、エラーを返却することを防ぎます。

1.2.3 アプリケーション開発環境の概要と特長

ここでは、アプリケーションサーバのアプリケーション開発環境の概要と特長について説明します。詳細については、「6.1 アプリケーション開発の特長」を参照してください。

(1) アプリケーション開発環境の概要

アプリケーションサーバでは、アプリケーションサーバ上で実行する J2EE アプリケーションを開発するために、J2EE アプリケーションの開発からデバッグまでを一連の流れで実行できる、統合開発環境を提供しています。アプリケーションサーバの統合開発環境では、業務指向スタイルでの J2EE アプリケーションの開発を支援しています。業務指向スタイルとは、業務の流れに着目し、その業務を遂行するために必要な操作や手順を実装していく開発スタイルです。これによって、ユーザインタフェースの早期確定や、MVC アーキテクチャに従った効率の良い J2EE アプリケーション開発ができます。アプリケーションサーバの開発環境では、プログラミング・デバッグからアプリケーションサーバ独自の開発支援機能を使用したアプリケーションの実装までを、一貫したインタフェースで実行できます。

アプリケーションサーバでは、Web サービスの開発も支援しています。アプリケーションサーバでは、JAX-WS 仕様に従った Web サービスの開発を支援しています。また、アプリケーションサーバが 07-60 以前から提供していた SOAP アプリケーション開発支援機能を使用して Web サービス (SOAP アプリケーション) を開発することもできます。SOAP アプリケーションで送受信する SOAP メッセージは、セキュリティ機能によって機密保護ができます。

さらに、アプリケーションサーバでは、開発環境の中にデバッグ環境を構築できます。このため、アプリケーションの設計・プログラミングからテスト・デバッグまで、アプリケーション開発に必要な一連の作業を、すべて開発環境で実行できます。

(2) アプリケーション開発環境の特長

アプリケーションサーバで構築するアプリケーションの開発環境の特長について説明し

ます。

アプリケーション開発の一連の流れを開発環境上で実現

IDE である Eclipse および Eclipse のプラグインを使用して、アプリケーション開発時のプログラミング、テスト、デバッグの一連の作業を実施できます。

また、アプリケーションサーバの開発環境では、Eclipse のプラグインをまとめた統合開発環境である、MyEclipse を提供しています。これによって、アプリケーション開発、デバッグ環境上でのテスト、デバッグなどの操作を、すべて Eclipse 上で実行できます。

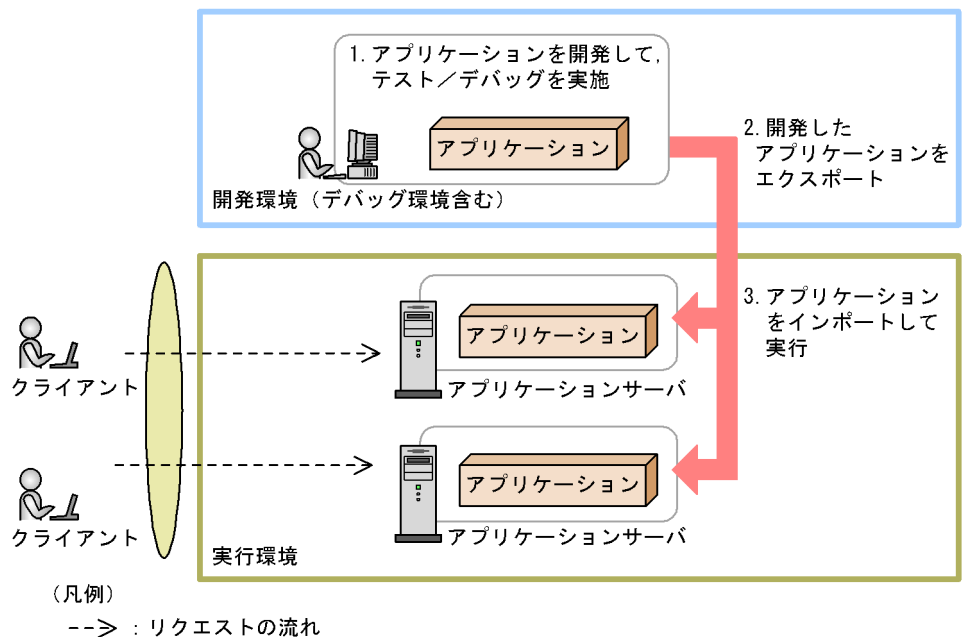
デバッグ環境と MyEclipse のセットアップを GUI で実行

ウィザードプログラム (GUI) によって、デバッグを実行するとき使用するサーバやデータベースのセットアップと、MyEclipse の Eclipse への組み込みとセットアップを実行できます。

1.2.4 アプリケーションの実行環境と開発環境の関係

アプリケーションの実行環境と開発環境の関係を、次の図に示します。

図 1-4 アプリケーションの実行環境と開発環境の関係



図の流れについて説明します。

1. 開発環境で、アプリケーションを開発して、テスト/デバッグを実施します。
2. 1. で開発したアプリケーションをエクスポートします。
3. 2. でエクスポートしたアプリケーションを実行環境にインポートします。

1. アプリケーションサーバの概要

なお、この図では、可用性を高めるために、一つのアプリケーションを複数の実行環境にインポートしています。

1.3 仮想化したサーバ上での J2EE アプリケーション実行環境

この節では、仮想化したサーバ上で構築・運用する J2EE アプリケーション実行環境について説明します。

近年、システムの高度化・複雑化が進み、システムで使用するサーバ数は増加しています。しかし、サーバの台数を増やすことで維持および管理コストが増大するおそれがあります。この問題に対する解決方法として、サーバの仮想化技術の適用が考えられます。仮想化技術の適用によって、物理サーバの台数はそのままサーバを増やすことができます。ただし、仮想化したサーバは増加傾向にあるため、仮想化したサーバ上に J2EE アプリケーションの実行環境を構築、運用するための負荷は減りません。

Cosminexus のアプリケーションサーバを使用すると、サーバの仮想化技術を使用したシステムの構築、運用が容易になります。これによって、次のことが実現できます。

複数の仮想サーバ上に同じ J2EE アプリケーションを実行する環境を一括で構築および運用できます。

Cosminexus のアプリケーションサーバで仮想サーバを構築するには、まず、仮想サーバ上に配置する次の情報をマスターとして登録します。

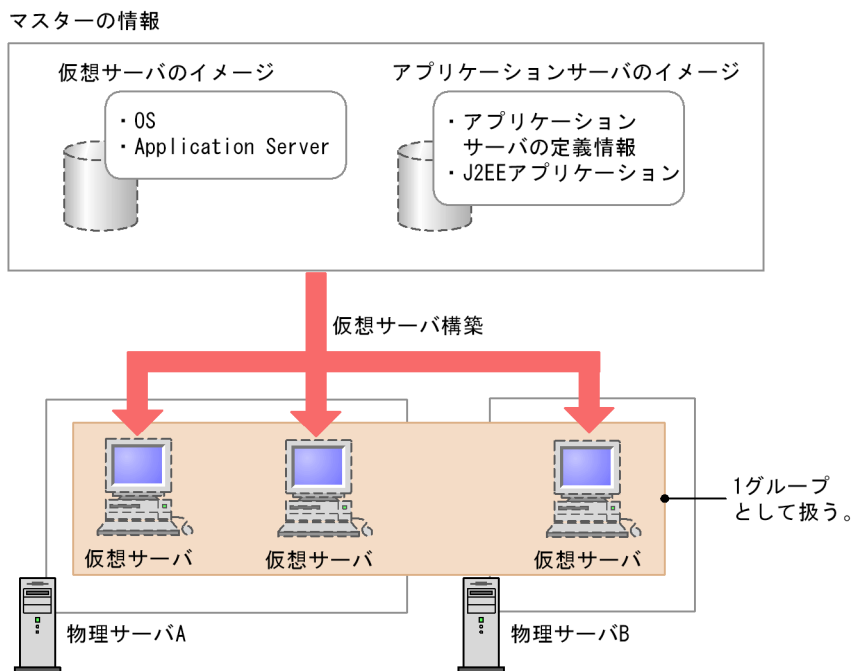
- 仮想サーバのイメージ（OS および仮想サーバにインストールする製品）
- アプリケーションサーバのイメージ（アプリケーションサーバの定義情報および J2EE アプリケーション）

登録したマスターの情報を基に仮想サーバを構築することで、同じ環境の仮想サーバを一括で複数構築できます。また、同じ J2EE アプリケーションが動作する同じ実行環境を持つ仮想サーバを一つのグループとして扱えるため、同じ環境の複数の仮想サーバを一括して起動または停止できます。

マスターの情報から仮想サーバを一括構築および運用する例を次の図に示します。

1. アプリケーションサーバの概要

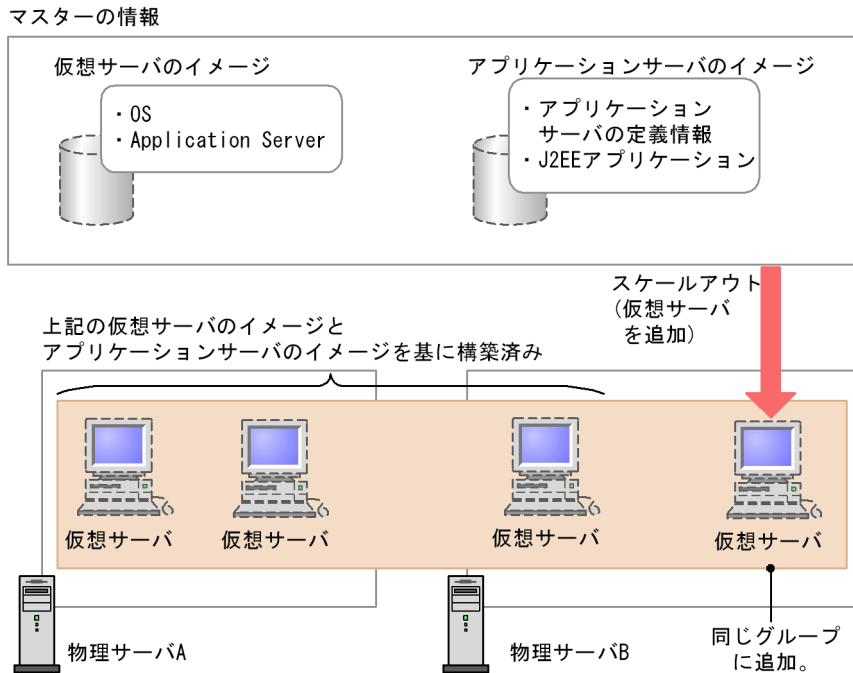
図 1-5 仮想サーバを一括構築および運用する例



この図の場合、三つの仮想サーバを一つのグループとして構築しています。このグループ内の仮想サーバをまとめて起動したり、停止したりできます。

コマンドの実行で、仮想サーバをスケールインまたはスケールアウトできます。特定の J2EE アプリケーションにアクセスが集中しているとき、コマンドを実行することで仮想サーバの数を容易に増やせます（スケールアウト）。このときも、マスターの情報が使われます。スケールアウトの例として、仮想サーバを一つ追加する例を次の図に示します。

図 1-6 仮想サーバを追加する例



スケールアウトで追加された仮想サーバは、構築済みの仮想サーバと同じグループに追加されるため、一括起動や一括停止の対象になります。

なお、仮想サーバのアクセス数が減り、仮想サーバの負荷が減ったときには、仮想サーバの数を減らすこともできます（スケールイン）。

仮想サーバマネージャに対する操作権限が管理できます。

仮想サーバマネージャを実行できるユーザのアカウントを、OSのアカウントとは別に管理できます。仮想サーバマネージャに対する操作権限として、機能全般に対するものと、仮想サーバの構築や運用など機能の一部に対するものが設定できます。ユーザごとに仮想サーバマネージャに対する操作権限を設定することで、セキュリティを強化できます。

1. アプリケーションサーバの概要

参考

08-50 モードの仮想サーバマネージャを利用する場合、Cosminexus のアプリケーションサーバで構築できる仮想化システムの構成パターンは決まっています。それぞれのシステム構成パターンについて説明します。

- パターン 1：テスト環境のオンデマンド提供
Web アプリケーションの開発者向けに提供するテスト環境を仮想化し、開発者の要望に応じてオンデマンドに提供する用途を想定しています。
- パターン 2：実行環境の持続的運用
Web システムを仮想化環境に構築して性能と信頼性を確保しつつ、各サーバの実行環境を持続的に運用することで安定稼働を維持する用途を想定しています。
- パターン 3：実行環境のスケールアウト
Web システムを仮想化環境に構築してシステム構成の柔軟性を生かします。業務量の増大に応じてスケールアウト（拡大）し、各サーバの環境を常に最新に保つ運用を想定しています。
- パターン 4：小規模部門サーバの集約
散在する部門サーバを仮想化環境に集約し、TCO を削減するシンプルな構成と運用を想定しています。

08-53 以降の場合に、08-50 モードの仮想サーバマネージャを利用するときは設定が必要です。詳細は、マニュアル「Cosminexus アプリケーションサーバ 仮想化システム構築・運用ガイド」の「付録 D 08-50 モードの仮想サーバマネージャを利用する場合の設定」を参照してください。

また、08-50 モードの仮想サーバマネージャでは、次のことが実現できます。

- 複数の仮想サーバ上に同じ J2EE アプリケーションを実行する環境を一括で構築および運用できます。
 - コマンドの実行で、仮想サーバをスケールインまたはスケールアウトできます。
 - 業務を止めることなく仮想サーバの内容を変更できます。
仮想サーバ上の OS の更新プログラムを適用したい場合や、アプリケーションサーバの定義情報を変更したい場合などは、イメージを更新すると、その更新内容を複数の仮想サーバに反映できます。
コマンド一つで複数の仮想サーバに対してイメージを反映できることが特長です。仮想サーバが稼働している場合でも、停止している場合でも変更できます。1 台ずつ反映していくことで、業務を完全に止めることなく変更内容を反映できます。
-

仮想サーバ上で動作するアプリケーションサーバを構築および運用する場合の詳細については、マニュアル「Cosminexus アプリケーションサーバ 仮想化システム構築・運用ガイド」を参照してください。

1.4 Cosminexus ビジネスプロセス管理 / エンタープライズサービスバスとの関係

この節では、Cosminexus のアプリケーションサーバと Cosminexus ビジネスプロセス管理 / エンタープライズサービスバスとの関係について説明します。Cosminexus では、ビジネスプロセス管理 (BPM) / エンタープライズサービスバス (ESB) を実現する製品として、サービスプラットフォーム (Service Platform および Service Architect) を提供しています。

サービスプラットフォームは、BPEL 準拠のビジネスプロセス管理を実現する機能と、サービスの統合を実現するエンタープライズサービスバスの機能を持つ製品です。

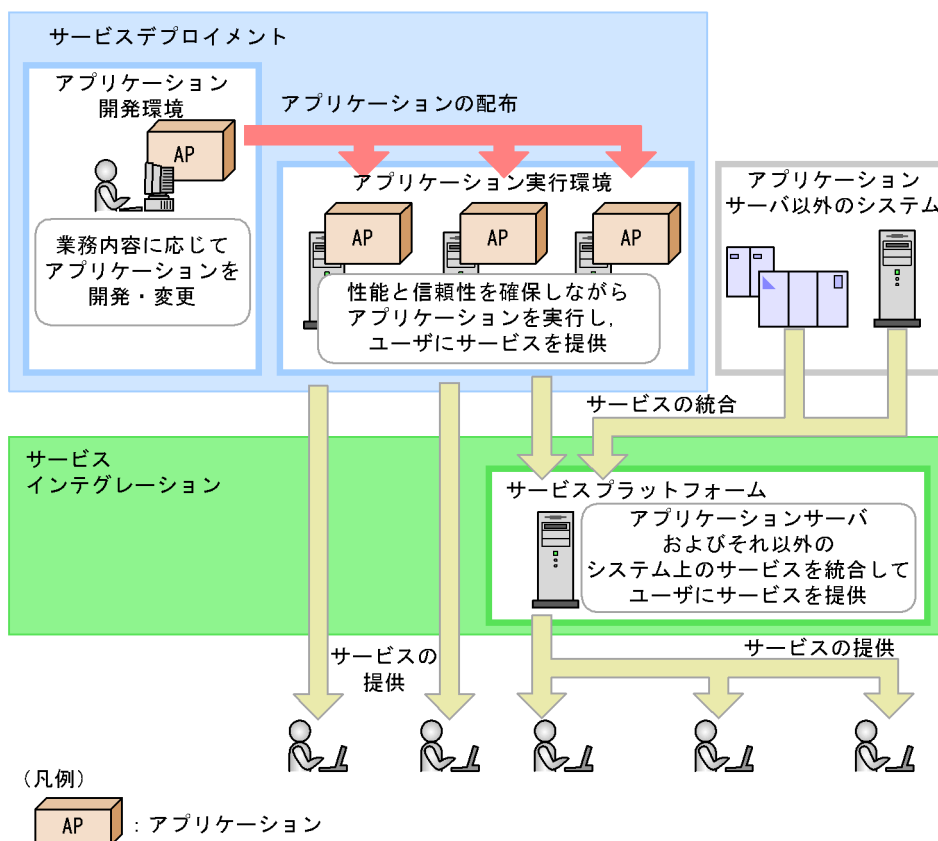
ESB では、業務に必要なアプリケーションをサービスとして利用します。既存のサービスや新規に作成したサービスを組み合わせて、新しいサービスの迅速な提供を実現します。複数のサービスを組み合わせる場合の実行順序は、ビジネスプロセスで定義します。

これらの機能によって、新規アプリケーションと既存アプリケーションを組み合わせた、ワンストップサービスを実現できます。

アプリケーションサーバとサービスプラットフォームを組み合わせた例を次の図に示します。

1. アプリケーションサーバの概要

図 1-7 アプリケーションサーバとサービスプラットフォームを組み合わせた例



アプリケーション実行環境では、性能と信頼性を確保しながら、ユーザからのリクエストを迅速に処理して、ユーザにサービスを提供します。また、バッチジョブを効率良く実行します。可用性や耐障害性の高い実行基盤として、24時間連続稼働が必要なサービス、停止することが許されない重要な業務、バッチジョブとして実行される基幹業務など、さまざまな要件のアプリケーションを実行するための環境になります。また、セキュリティに配慮したシステムや、内部統制に対応したシステムを実現します。

業務内容の変更や組織の組み替えなどが発生した場合に、新しいアプリケーションを開発したり、既存のアプリケーションを変更したりするためには、アプリケーション開発環境を使用できます。アプリケーション開発環境では、画面設計から、バックエンドシステムと連携する業務処理の実装まで、シームレスに効率良く実行できます。

業務内容の変更や組織の組み替えなどに対して、既存のアプリケーションや他システムが提供するサービスの再利用で対応する場合は、サービスプラットフォームを使用します。サービスプラットフォームでは、アプリケーション実行環境が提供するサービスやアプリケーションサーバ以外が提供するサービスを統合して、新しいサービスとしてユーザに提供できます。また、サービスプラットフォームは、複数のアプリケーションやシステムから提供される既存のサービスを、統一されたインターフェースで提供するた

めの基盤としても使用できます。

サービスプラットフォームの概要と特長については、マニュアル「Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス サービスプラットフォーム 概説」を参照してください。

参考

このマニュアルで記載しているアプリケーションサーバの機能は、サービスプラットフォームでも使用できます。

対応は次のとおりです。

- Application Server Standard または Application Server Enterprise の機能は、Service Platform でも使用できます。
 - Developer Professional の機能は、Service Architect でも使用できます。
-

2

アプリケーションサーバの製品構成

この章では、アプリケーションサーバの製品構成について説明します。アプリケーションサーバには、実行環境を構築する製品、開発環境を構築する製品、およびそれらの環境で必要に応じて使用するオプション製品があります。また、これらの製品は、構成ソフトウェアと呼ばれる複数のソフトウェアで構成されています。

2.1 製品の分類と特長

2.2 製品の概要

2.3 構成ソフトウェア

2.4 動作環境

2.1 製品の分類と特長

アプリケーションサーバの製品は、用途ごとに、3種類に分類できます。製品の分類を次の表に示します。

表 2-1 アプリケーションサーバの製品の分類

分類	製品
実行環境を構築する製品	<ul style="list-style-type: none">• Application Server Standard• Application Server Enterprise
開発環境を構築する製品	<ul style="list-style-type: none">• Developer Standard• Developer Professional
オプション製品	<ul style="list-style-type: none">• uCosminexus Client• uCosminexus Web Redirector• uCosminexus Operator

この節では、分類ごとの特長について説明します。

2.1.1 実行環境を構築する製品

Java EE に準拠した J2EE アプリケーションを、サーバサイドで実行するための実行環境を構築するための製品です。Web サービスに対応したアプリケーションや、メッセージによる非同期通信に対応したアプリケーションも実行できます。クライアントには、Web ブラウザ、または EJB クライアントアプリケーションを使用できます。

実行環境を構築する製品を次に示します。

Application Server Standard

Application Server Enterprise

Application Server Enterprise は、Application Server Standard に比べて、バックエンドシステムとしての機能に優れています。また、Application Server Enterprise では、バッチアプリケーションのジョブスケジューリングを実行する機能も使用できます。

なお、実行環境では、システム構成に応じてオプション製品も利用できます。

2.1.2 開発環境を構築する製品

アプリケーションサーバの実行環境で実行する J2EE アプリケーションの開発環境を構築するための製品です。IDE などを使用して開発した J2EE アプリケーションに対して、アセンブル、デバッグ、デプロイ、テストなどができます。

また、J2EE アプリケーションの開発に Eclipse を使用する場合、MyEclipse を使用できます。MyEclipse は、Eclipse 上でのアプリケーション開発で使用するプラグインをまと

めた統合開発環境です。MyEclipse を使用すると、UML の設計からコーディング、テスト、デバッグまで、一連の操作をまとめて実行できます。

開発環境を構築する製品を次に示します。

Developer Standard

Developer Professional

Developer Standard は、一つの J2EE サーバ内で動作するアプリケーションを開発するための製品です。Developer Professional は、別プロセスや別マシン上のアプリケーションを呼び出すアプリケーションを開発するための製品です。

なお、Developer Standard および Developer Professional は、Windows 環境だけで使用できます。

2.1.3 オプション製品

アプリケーションサーバで実行環境を構築する場合に、システム構成に応じて使用できる製品です。

業務システム内に、用途を限定したマシンを配置する場合に使用できます。

オプション製品を次に示します。

uCosminexus Client

実行環境上の Enterprise Bean をクライアントマシンのアプリケーションから直接呼び出す構成の場合に使用できます。また、Web サービス実行環境上の Web サービスを、クライアントマシンのアプリケーションから RPC を使用して呼び出す構成の場合にも使用できます。

uCosminexus Web Redirector

Web サーバとアプリケーションサーバを別のマシンに分けて配置する場合に使用できます。

uCosminexus Operator

実行環境と運用環境を別のマシンに分けて配置する場合に使用できます。UNIX 実行環境に対して Eclipse のプラグインを使用した運用操作をしたい場合は、必ず使用します。

これらのオプション製品を使用できる OS については、「2.4 (1) 前提 OS」を参照してください。

2.2 製品の概要

この節では、次の製品の概要について説明します。

実行環境を構築する製品

- Application Server Standard
- Application Server Enterprise

開発環境を構築する製品

- Developer Standard
- Developer Professional

オプション製品

- uCosminexus Client
- uCosminexus Web Redirector
- uCosminexus Operator

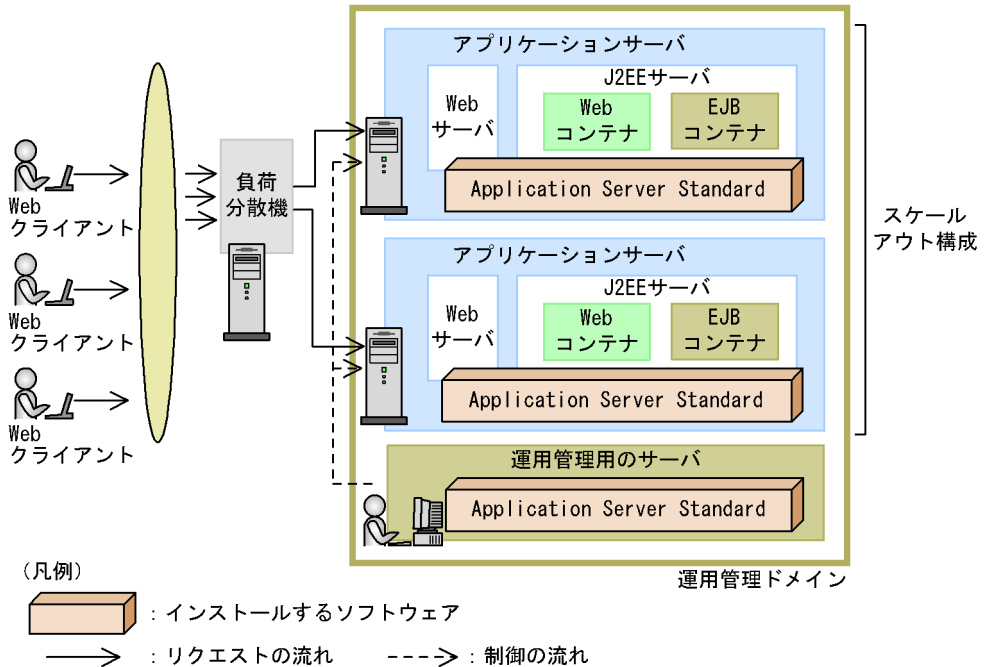
2.2.1 Application Server Standard

Application Server Standard は、実行環境を構築するための製品です。J2EE アプリケーションの実行環境と Java で開発したバッチアプリケーションの実行環境を構築できます。ここでは、それぞれの環境について説明します。

(1) Application Server Standard で構築できる J2EE アプリケーションの実行環境

Application Server Standard は、スケールアウト構成のシステムを構築・運用するのに適しています。スケールアウト構成とは、同じ処理をするサーバマシンを増やしてリクエスト処理を実行することで、システムの処理性能を高め、かつ可用性や信頼性を確保できる構成です。Application Server Standard で構築できる J2EE アプリケーションの実行環境（スケールアウトシステム）の構成例を次の図に示します。

図 2-1 Application Server Standard で構築できる J2EE アプリケーションの実行環境
(スケールアウトシステム)の構成例



スケールアウト構成の場合、Web クライアントからのリクエストを負荷分散機が受け付け、アプリケーションサーバに振り分けます。

Application Server Standard では、スケールアウト構成にしたシステムの一括構築や一括運用を実現できます。また、アプリケーションサーバを運用管理ドメインという範囲で一括管理して、ドメイン内の個別のサーバマシンに対するリモート操作での運用管理も実現できます。

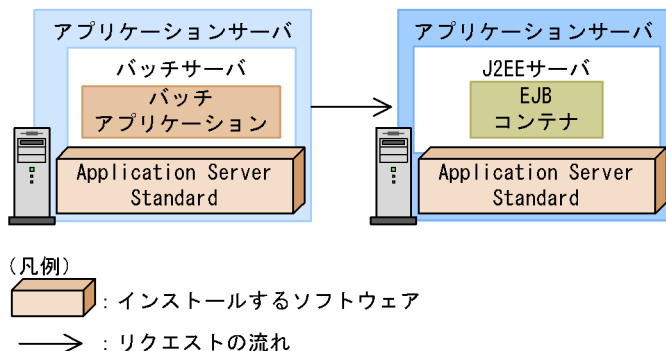
Application Server Standard は、アプリケーションサーバにするマシンにインストールします。業務システムを一括構築・運用する場合には、運用管理用のサーバを別のマシンとして構築するときには、運用管理用のサーバにするマシンにも Application Server Standard をインストールします。

(2) Application Server Standard を使用して構築できるバッチアプリケーションの実行環境

Application Server Standard で構築できるバッチアプリケーションの実行環境の構成例を次の図に示します。

2. アプリケーションサーバの製品構成

図 2-2 Application Server Standard で構築できるバッチアプリケーションを実行するシステムの構成例



バッチアプリケーションは、Application Server Standard をインストールして構築した環境で実行します。バッチアプリケーションから Enterprise Bean を呼び出す場合、Enterprise Bean は Application Server Standard または Application Server Enterprise をインストールして構築した環境の EJB コンテナで実行します。なお、バッチアプリケーションの実行環境では、運用管理機能はアプリケーションサーバと同じサーバで実行します。

2.2.2 Application Server Enterprise

Application Server Enterprise は、実行環境を構築するための製品です。オンライン処理を実行するアプリケーションや、ジョブスケジューリングの対象になるバッチアプリケーションの実行環境を構築できます。

Application Server Standard に比べて、次の機能が強化されています。

Enterprise Bean のスケジューリング機能（流量制御 / 優先制御 / 閉塞制御）と負荷分散機能

バッチアプリケーションのジョブスケジューリング機能

仮想サーバ上でのアプリケーションサーバの構築・運用支援機能

OpenTP1 の SUP から J2EE アプリケーションを呼び出す機能（TP1 インバウンド連携機能）

ここでは、Application Server Enterprise を使用して構築できる実行環境について説明します。

(1) Application Server Enterprise を使用して構築できる J2EE アプリケーションの実行環境

J2EE アプリケーションの実行環境を構築する場合、Application Server Enterprise は、フロントエンド / バックエンドで構成される業務システムを構築・運用する場合に適し

ています。フロントエンド/バックエンドシステムとは、業務システム内のサーバを、用途ごとに、次のようにフロントエンドとバックエンドに分けて運用するシステム構成です。

フロントエンド

Web ブラウザから送信されるリクエストを受け付ける部分です。受け付けたリクエストをバックエンドシステムの業務サービスに振り分けたり、バックエンドシステムからのレスポンスを Web ブラウザに送信したりします。

アプリケーションとして、サーブレット、JSP、Enterprise Bean などが動作します。

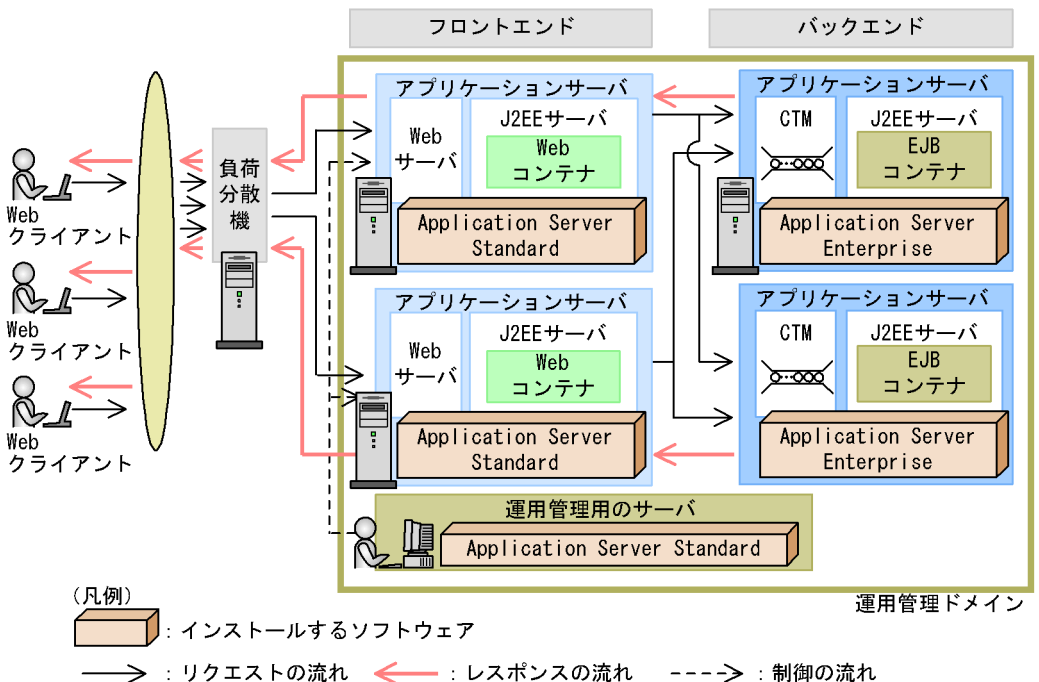
バックエンド

フロントエンドの背後で動作する、複数の業務に共通な業務サービス処理を実行するシステムです。Enterprise Bean などの業務処理プログラムが動作します。データベースアクセスなども実行します。

バックエンドシステムを Application Server Enterprise を使用して構築することによって、Enterprise Bean に対するスケジューリングや負荷分散などの機能を生かした、安定稼働性および可用性の高いシステムを構築できます。

Application Server Enterprise で構築できるフロントエンド/バックエンドシステムの構成例を次の図に示します。

図 2-3 Application Server Enterprise で構築できるフロントエンド/バックエンドシステムの構成例



2. アプリケーションサーバの製品構成

フロントエンド/バックエンドシステム構成の場合、Webクライアントからのリクエストを負荷分散機が受け付け、フロントエンドのアプリケーションサーバに振り分けます。フロントエンドのアプリケーションサーバは、バックエンドのアプリケーションサーバにリクエストを振り分けます。バックエンドのアプリケーションサーバでは、EJBスケジューラとしての機能を持つCTMによってリクエストの負荷分散が実行され、適切なバックエンドのアプリケーションサーバ上で業務処理プログラムが実行されます。

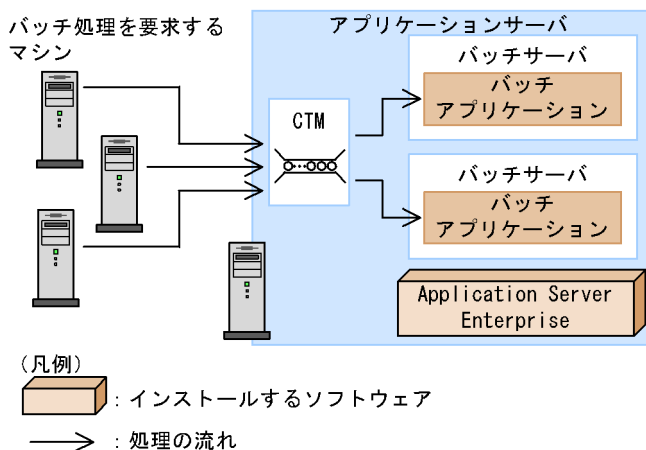
Application Server Enterprise で構築したシステムも、運用管理ドメインの範囲での一括管理ができます。

Application Server Enterprise は、バックエンドに配置するアプリケーションサーバのマシンにインストールします。また、フロントエンドのアプリケーションサーバ、または運用管理用のサーバにするマシンには、Application Server Standard または Application Server Enterprise をインストールします。

(2) Application Server Enterprise を使用して構築できるバッチアプリケーションの実行環境

Application Server Enterprise で構築できるバッチアプリケーションの実行環境の構成例（ジョブスケジューリング機能を使用する場合）を次の図に示します。

図 2-4 Application Server Enterprise で構築できるバッチアプリケーションを実行するシステムの構成例（ジョブスケジューリング機能を使用する場合）

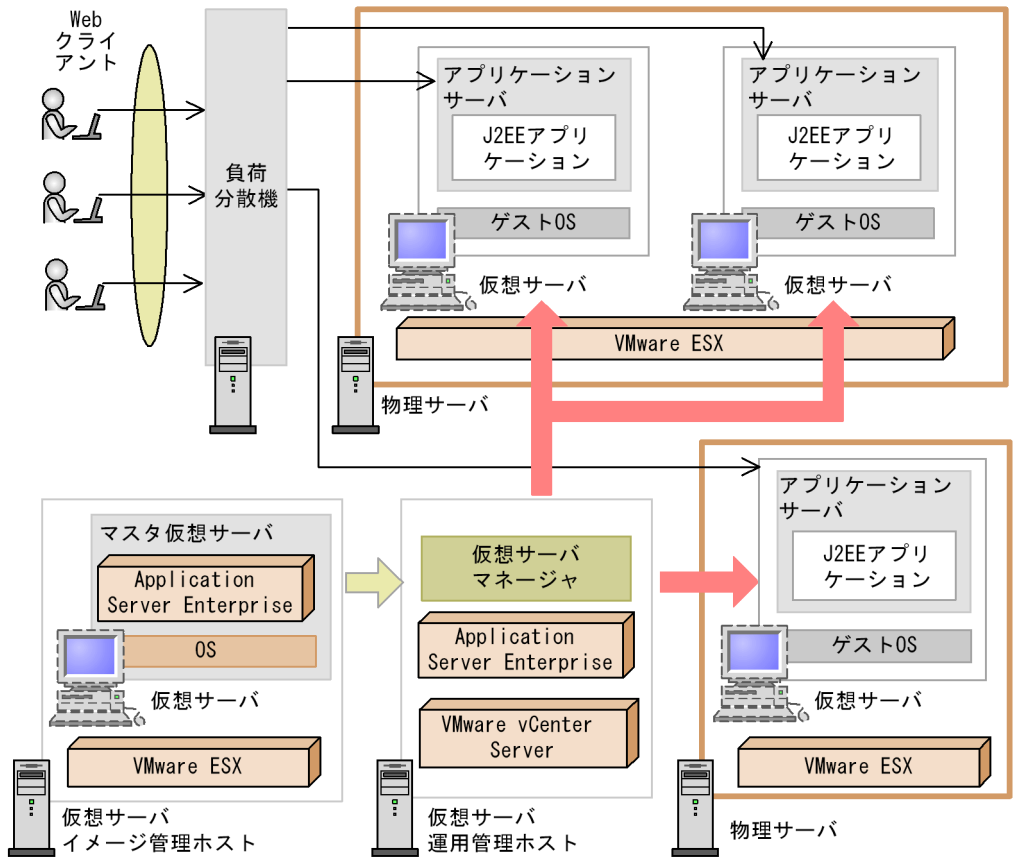


Application Server Enterprise をインストールして構築した環境のバッチサーバに対するバッチ処理要求は、CTM が受け付けます。一度に複数のバッチ処理が要求された場合は、CTM によって処理要求のスケジューリングと、複数のバッチサーバへの処理要求の振り分けが実施されます。実行できるバッチサーバがないときには、CTM のスケジューラキューで処理要求を滞留させることができます。





(3) Application Server Enterprise を使用して構築できる仮想サーバ上の J2EE アプリケーションの実行環境

Application Server Enterprise を使用して構築できる、仮想サーバ上の J2EE アプリケーションの実行環境の構成例を次の図に示します。

図 2-5 Application Server Enterprise で構築できる仮想サーバ上の J2EE アプリケーションの実行環境の構成例



(凡例)

-  : インストールするソフトウェア
-  : リクエストの流れ
-  : マスタ仮想サーバの情報の取り込み
-  : 構築・運用の流れ

Application Server Enterprise を使用すると、物理サーバ上に、J2EE アプリケーションを実行するアプリケーションサーバを配置した仮想サーバを構築できます。仮想サーバを構築するには、仮想サーバイメージ管理ホストと仮想サーバ運用管理ホストが必要になります。仮想サーバイメージ管理ホストでは、マスタ仮想サーバを作成します。作成したマスタ仮想サーバを仮想サーバ運用管理ホストの仮想サーバマネージャというプロセスに取り込み、仮想サーバ運用管理ホストから各物理サーバに対して、仮想サーバ

2. アプリケーションサーバの製品構成

の一括構築および一括運用を実施します。

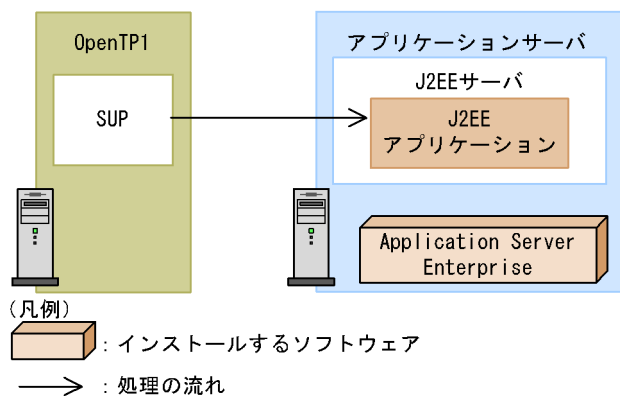
Web クライアントからのリクエストは、構築した複数の仮想サーバで処理されます。仮想サーバへのリクエストの振り分けには、負荷分散機を使用します。

仮想サーバ上に J2EE アプリケーションの実行環境を構築および運用する方法については、マニュアル「Cosminexus アプリケーションサーバ 仮想化システム構築・運用ガイド」を参照してください。

(4) Application Server Enterprise を使用して構築できる OpenTP1 の SUP からの呼び出しに対応する実行環境

OpenTP1 の SUP からの J2EE アプリケーションの呼び出しに対応する実行環境の構成例を次の図に示します。

図 2-6 Application Server Enterprise で構築できる OpenTP1 の SUP からの J2EE アプリケーションの呼び出しに対応する実行環境の構成例



Application Server Enterprise をインストールしたマシンで動作する J2EE アプリケーションを OpenTP1 の SUP から呼び出すことができます。

2.2.3 Developer Standard

Developer Standard は、開発環境を構築するための製品です。J2EE アプリケーションの作成・テスト・デバッグといったアプリケーション開発の一連の流れを、スタンドアロン PC 内で実行できます。

ポイント

Developer Standard では、一つのマシン内に構築した一つの J2EE サーバ内で動作する J2EE アプリケーションを開発できます。また、構築できるデバッグ環境も、スタンドアロン PC での運用を前提としています。このため、開発した J2EE アプリケーションを複数のマシンで構成される実行環境上で実行することを想定していたり、J2EE アプリケーション内でプロセス間呼び出しを実行する J2EE アプリケーションを開発したりする場合は、Developer Professional を使用してください。

Developer Standard で構築した開発環境でのアプリケーション開発の特長は次のとおりです。

アプリケーションの開発に、Eclipse および Eclipse のプラグイン（MyEclipse、WTP など）を使用できます。

なお、デバッグ環境はウィザードプログラムで一括構築できます。

2.2.4 Developer Professional

Developer Professional は、開発環境を構築するための製品です。Developer Standard の機能に加えて、次の機能を使用できます。

プロセス間でのリモート呼び出しをする J2EE アプリケーションの開発

リモート呼び出しを想定したデバッグ環境の構築

Developer Professional のデバッグ環境では、Application Server Enterprise で構築した実行環境と同じ機能を使用して J2EE アプリケーションをテストできます。

2.2.5 uCosminexus Client

uCosminexus Client は、次に示すクライアントの実行環境を構築するためのオプション製品です。Application Server Standard の機能のうち、クライアントを実行するために必要な機能だけで構成されています。

EJB クライアントアプリケーションの実行環境

アプリケーションサーバ上で動作している J2EE アプリケーション内の Enterprise Bean を、Web サーバ経由ではなく、クライアントマシン上のプログラムから RMI-IIOP 通信で直接呼び出すシステムを構築する場合に使用します。複数の PC で EJB クライアントアプリケーションを実行するシステム構成の場合に、各 PC に uCosminexus Client をインストールします。

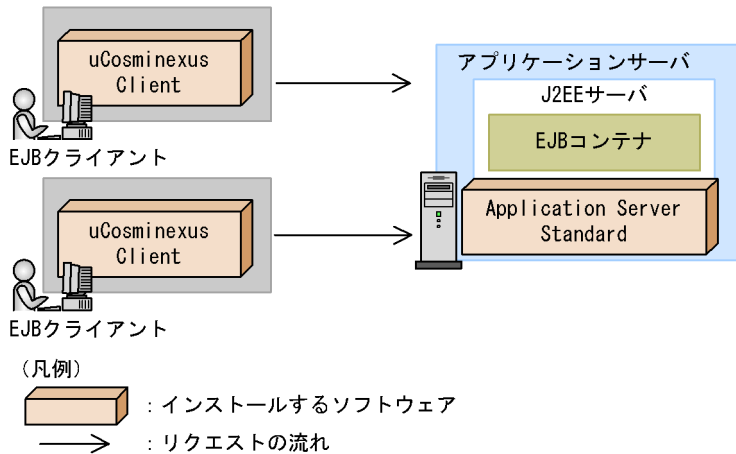
Web サービスのクライアントになる Java アプリケーションの実行環境

アプリケーションサーバ上で動作している Web サービスを、クライアントマシン上の Java アプリケーションから、RPC を使用して呼び出す場合に使用します。

2. アプリケーションサーバの製品構成

uCosminexus Client を利用したシステム構成の例を次の図に示します。この図は、EJB クライアントアプリケーションの実行環境の例です。

図 2-7 uCosminexus Client を利用したシステム構成の例



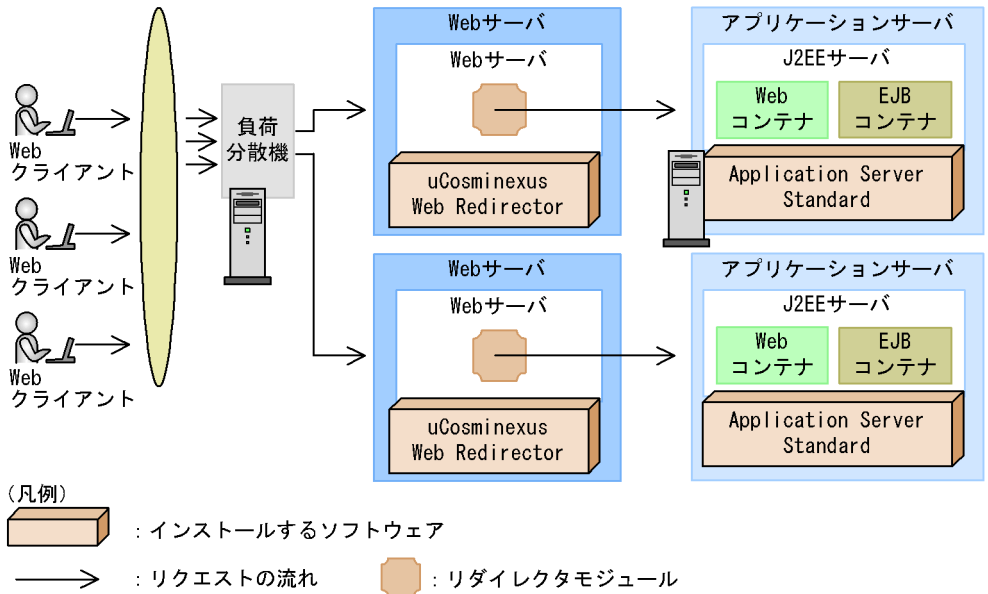
uCosminexus Client をインストールした EJB クライアントマシンからは、RMI-IIOP 通信によって、直接 J2EE サーバ上の Enterprise Bean にアクセスします。アプリケーションサーバに、構成ソフトウェアとして Cosminexus Component Transaction Monitor (CTM) を含む製品がインストールされている場合は、EJB スケジューラである CTM 経由でもアクセスできます。製品と構成ソフトウェアの対応については、「2.3.1 製品と構成ソフトウェアの対応」を参照してください。

2.2.6 uCosminexus Web Redirector

uCosminexus Web Redirector は、実行環境を構築する場合に、Web サーバとアプリケーションサーバを異なるマシンに配置するとき使用するオプション製品です。Web サーバを配置したマシンに uCosminexus Web Redirector をインストールして、Web サーバにアプリケーションサーバ上の Web コンテナと連携するためのモジュール（リダイレクタモジュール）を組み込んで動作させます。この機能を、Web Redirector 機能といいます。

uCosminexus Web Redirector を利用したシステム構成の例を次の図に示します。

図 2-8 uCosminexus Web Redirector を利用したシステム構成の例



uCosminexus Web Redirector は、Web サーバと同じマシンにインストールします。Web サーバが受け付けたリクエストは、リダイレクタモジュールを経由して、アプリケーションサーバの Web コンテナに送信されます。

なお、Web Redirector 機能は、Application Server でも提供されています。

2.2.7 uCosminexus Operator

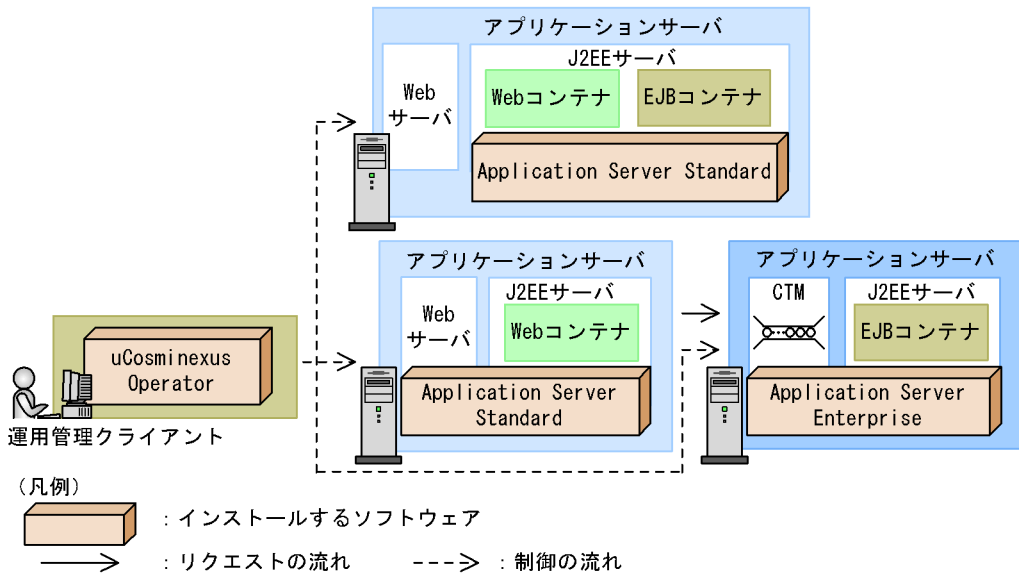
uCosminexus Operator は、Application Server Standard、または Application Server Enterprise をインストールした実行環境のマシンとは別の Windows マシンから、GUI (Eclipse のプラグイン) を使用して実行環境を操作するためのオプション製品です。

uCosminexus Operator は、UNIX マシン上に構築したアプリケーションサーバを Windows マシン上の Eclipse の GUI を使用して操作する場合に使用します。また、Application Server Standard、Application Server Enterprise または Service Platform をインストールした実行環境のマシンとは別の Windows マシンから、Eclipse の GUI を使用して実行環境を操作する場合にも必要です。

uCosminexus Operator を利用したシステム構成の例を次の図に示します。

2. アプリケーションサーバの製品構成

図 2-9 uCosminexus Operator を利用したシステム構成の例



uCosminexus Operator をインストールしたマシンから、アプリケーションサーバをインストールしたマシンの運用操作を実行します。

なお、uCosminexus Operator は、Application Server に同梱されています。

2.3 構成ソフトウェア

この節では、アプリケーションサーバの構成ソフトウェアについて説明します。構成ソフトウェアは、単体で動作させるのではなく、ほかの構成ソフトウェアの機能と組み合わせて動作させることによって、アプリケーションサーバとしての機能を実現します。

構成ソフトウェアは、アプリケーションサーバの機能に対応します。製品に含まれている構成ソフトウェアの種類によって、実現できる機能が異なります。

2.3.1 製品と構成ソフトウェアの対応

ここでは、製品と、その製品に含まれる構成ソフトウェアの対応を示します。

製品に対応する構成ソフトウェアを、次の表に示します。

表 2-2 製品と構成ソフトウェアの対応

構成ソフトウェア名	実行環境を構築する製品		開発環境を構築する製品	
	Application Server Standard	Application Server Enterprise	Developer Standard	Developer Professional
Cosminexus Application Development Plug-in	-	-		
Cosminexus Component Container				
Cosminexus Component Transaction Monitor	-		-	
Cosminexus Developer's Kit for Java				
Cosminexus Performance Tracer				
Cosminexus Reliable Messaging			-	
Cosminexus Server Plug-in				
Cosminexus TPBroker			-	

2. アプリケーションサーバの製品構成

構成ソフトウェア名	実行環境を構築する製品		開発環境を構築する製品	
	Application Server Standard	Application Server Enterprise	Developer Standard	Developer Professional
Cosminexus Web Services - Security			-	
Cosminexus XML Processor				
HiRDB Embedded Server Version 8	-	-		
Hitachi Web Server				
MyEclipse for Cosminexus	-	-		

(凡例)

:含まれます。 - :含まれません。

注

UNIX で動作する製品には含まれません。

オプション製品に対応する構成ソフトウェアを、次の表に示します。

表 2-3 製品と構成ソフトウェアの対応 (オプション製品)

構成ソフトウェア名	オプション製品		
	uCosminexus Client	uCosminexus Web Redirector	uCosminexus Operator
Cosminexus Component Container - Client		-	-
Cosminexus Component Container - Redirector	-		-
Cosminexus Developer's Kit for Java		-	
Cosminexus Performance Tracer			-
Cosminexus Server Plug-in	-	-	
Cosminexus TPBroker		-	-

構成ソフトウェア名	オプション製品		
	uCosminexus Client	uCosminexus Web Redirector	uCosminexus Operator
Cosminexus XML Processor		-	

(凡例)

:含まれます。 - :含まれません。

注

uCosminexus Web Redirector は、個別に提供されているほか、uCosminexus Application Server Standard または uCosminexus Application Server Enterprise にも同梱されています。
uCosminexus Operator は、uCosminexus Application Server Standard または uCosminexus Application Server Enterprise にも同梱されています。

2.3.2 構成ソフトウェアの機能概要

それぞれの構成ソフトウェアの機能概要について説明します。

(1) Cosminexus Application Development Plug-in

開発環境で使用する次の機能を提供する構成ソフトウェアです。

Cosminexus コネクタ (Eclipse プラグイン)

開発環境の MyEclipse から、デバッグ環境上の J2EE サーバを操作するための機能です。Eclipse プラグインとして提供します。

開発環境のインスタントセットアップ機能

ウィザードプログラム (GUI) によってアプリケーションの開発環境およびデバッグ環境をセットアップできる機能です。

MyEclipse セットアップ機能

ウィザードプログラム (GUI) によって MyEclipse をセットアップできる機能です。

WTP コネクタ (Eclipse プラグイン)

開発環境の WTP から、デバッグ環境上の J2EE サーバを操作するための機能です。Eclipse プラグインとして提供します。

(2) Cosminexus Component Container

アプリケーションの実行基盤の中核として、次のような機能を提供する構成ソフトウェアです。

J2EE アプリケーションの実行環境 (J2EE サーバ) としての機能

バッチアプリケーションの実行環境 (バッチサーバ) としての機能

Web サービスの実行・開発環境としての機能

2. アプリケーションサーバの製品構成

アプリケーションサーバを運用管理する機能

それぞれの機能の概要を示します。

(a) J2EE アプリケーションの実行環境 (J2EE サーバ) としての機能

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フレームワークである, J2EE サーバを実現するための機能です。Web コンテナ, EJB コンテナなどの機能を含みます。Java Platform, Enterprise Edition (Java EE) に含まれる仕様に準拠しています。Java EE 5 に含まれる仕様の詳細については, マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」の「2.3.9 サーバの動作モード」を参照してください。

さらに, 構成ソフトウェアに Cosminexus Component Transaction Monitor (CTM) を含む製品の場合は, CTM 機能と連携して, Enterprise Bean に対する動的負荷分散, 流量制御, 優先制御およびサービス閉塞に対応するための基盤機能も提供します。

(b) バッチアプリケーションの実行環境 (バッチサーバ) としての機能

バッチアプリケーションをサーバで実行するための機能です。バッチ処理の処理内容を Java で実装した Java アプリケーションを実行できます。サーバ上でアプリケーションを動作させることによって, JavaVM の起動コストを抑えられます。コネクションプールやステートメントプールを使用した効率の良いデータベースアクセスや, フルガーベージコレクションの制御なども実現できます。

(c) Web サービスの実行・開発環境としての機能

Web サービスの実行環境および開発環境としての機能です。

JAX-WS 仕様に準拠した Web サービスのサーバおよびクライアントの実行に必要な JAX-WS エンジンを提供します。

また, WSDL や SEI を起点とした Web サービスおよび Web サービスクライアントの開発を支援します。

参考

既存機能である SOAP アプリケーション開発支援機能も使用できます。

(d) アプリケーションサーバを運用管理する機能

アプリケーションサーバを運用管理するための機能です。次のような運用管理を実現できます。

アプリケーションサーバの一括構築・一括運用

J2EE サーバ内のアプリケーションやリソースの設定

複数の J2EE アプリケーションのユーザ管理機能をシームレスに連携した統合ユーザ

管理

アプリケーションサーバの各機能が出力するログの収集

また、JP1 などのほかの運用管理プログラム製品と連携して、アプリケーションサーバの運用管理をするためのコマンドも提供しています。

(3) Cosminexus Component Container - Client

EJB クライアントアプリケーションの実行環境を構築するための構成ソフトウェアです。Cosminexus Component Container のサブセットです。

(4) Cosminexus Component Container - Redirector

Web コンテナを Web サーバと連携させるための構成ソフトウェアです。この構成ソフトウェアで提供されるリダイレクタモジュールを Web サーバに登録することで、Web サーバあての HTTP リクエストのうち、特定のリクエストを指定した Web コンテナに処理させたり、複数の Web コンテナにリクエストを振り分けて処理させたりできます。

(5) Cosminexus Component Transaction Monitor

Enterprise Bean に対するクライアントからのリクエストをスケジューリングして、負荷分散や流量制御を実現する構成ソフトウェアです。アプリケーションごとにキューを管理して各サーバの負荷状況に応じて処理を分散させたり、一度に処理するリクエストの数を制御したり、業務処理プログラムを入れ替える時に特定の J2EE アプリケーションだけを閉塞させたりできます。これによって、システムが安定した状態で運転し続けることができるので、業務システムの可用性と信頼性が向上します。

また、バッチアプリケーションの実行もスケジューリングできます。

(6) Cosminexus Developer's Kit for Java

Java 2 Platform Standard Edition 5.0 または Java Platform, Standard Edition 6 に準拠した構成ソフトウェアです。対応する Sun Microsystems 社製の JDK のバージョンは JDK 5.0 または JDK 6 です。JDK 5.0 または JDK 6 で使用できる機能、コマンドおよび API については、Sun Microsystems 社が提供している JDK 5.0 または JDK 6 のドキュメントを参照してください。

! 注意事項

アプリケーションサーバの製品によって、同梱されている JDK のバージョンが異なります。

(7) Cosminexus Performance Tracer

処理性能のボトルネックを解析するためのトレース情報を出力する構成ソフトウェアです。アプリケーションサーバで構築したシステムでは、リクエストが処理される時に、

2. アプリケーションサーバの製品構成

決められたポイントごとに性能解析用のトレース情報を出力します。この情報を収集して分析することで、システムのボトルネックが調査できます。また、障害が発生した場合には、障害の発生個所を特定することもできます。

(8) Cosminexus Reliable Messaging

業務コンポーネント間、社内システム間、および社内システムと社外システム間で、非同期に高信頼なメッセージ送受信を実現する高信頼メッセージング基盤としての機能を提供する構成ソフトウェアです。WS-Reliability の仕様に準拠しています。

(9) Cosminexus Server Plug-in

J2EE サーバ上の J2EE アプリケーションまたは J2EE リソースを操作するための Eclipse プラグインです。

(10) Cosminexus TPBroker

サーブレットまたは JSP と Enterprise Bean の間などの通信で使用される、Java EE での RMI-IIOP 通信基盤、および分散トランザクション基盤になる構成ソフトウェアです。また、Java EE 環境での CORBA クライアントアプリケーションの実行環境および開発環境を提供します。また、EJB クライアントアプリケーションの実行環境で使用する、RMI-IIOP 通信基盤、および CORBA クライアントアプリケーションの実行環境としての機能も提供します。

(11) Cosminexus Web Services - Security

Web サービスのセキュリティの標準規格 WS-Security に準拠した機能を提供する構成ソフトウェアです。SOAP メッセージに対して、XML 署名を付れたり、SOAP メッセージを暗号化したりします。Cosminexus Web Services - Security の利用によって、セキュアな状態で SOAP メッセージの送受信ができます。また、SOAP メッセージに付いている署名を検証したり、暗号化された SOAP メッセージを復号化したりできます。

XML 署名の生成・検証、または XML 暗号によるデータの暗号化・復号化を行うアプリケーションの開発を支援する機能も提供しています。

(12) Cosminexus XML Processor

業界標準の JAXP1.3 をサポートした XML ドキュメントの読み取り、操作および生成を実行する構成ソフトウェアです。

(13) HiRDB Embedded Server Version 8

アプリケーション開発時、テストおよびデバッグに使用できるデータベースです。

(14) Hitachi Web Server

Apache2.0 をベースに Secure Sockets Layer (SSL) をサポートしたミッションクリ

ティカル分野向けの Web サーバです。

(15) MyEclipse for Cosminexus

Eclipse プラグインをまとめた統合開発環境 MyEclipse の日本語版環境です。

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。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。

2.3.3 実行環境の構成ソフトウェアの位置づけ (Application Server)

ここでは, 実行環境を構築する製品である Application Server (Application Server Standard または Application Server Enterprise) の各構成ソフトウェアの位置づけについて, 実行するアプリケーションの種類ごとに説明します。

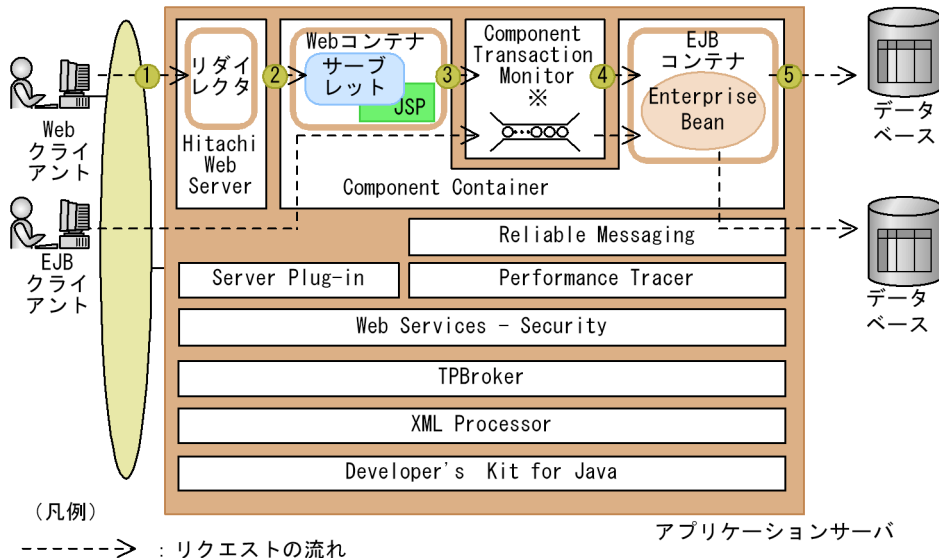
(1) J2EE アプリケーション実行時の構成ソフトウェアの位置づけ

アプリケーションサーバ上で J2EE アプリケーションを実行する場合に使用される Application Server の構成ソフトウェアの位置づけについて説明します。

Web クライアントまたは EJB クライアントからのリクエストを処理する場合に使用される Application Server の構成ソフトウェアの位置づけを次の図に示します。

2. アプリケーションサーバの製品構成

図 2-10 Application Server の構成ソフトウェアの位置づけ (Web クライアントまたは EJB クライアントからのリクエストを処理する場合)



注1 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

注2 図中の番号は、「(a)Webクライアントの場合」の説明に対応しています。

注※ Application Server Enterpriseの場合だけに含まれる構成ソフトウェアです。

参考

このほか、Application Server Enterprise では、Cosminexus Component Transaction Monitor の CTM のゲートウェイ機能を利用して TPBroker / OTM クライアントからのリクエストを受け付けることもできます。

構成ソフトウェアの位置づけとクライアントからのリクエストの流れについて、Web クライアントと EJB クライアントの場合に分けて説明します。

(a) Web クライアントの場合

リクエストの流れについて説明します。なお、項番は、図 2-10 の図中の番号に対応しています。

1. Web クライアントの Web ブラウザから送信したリクエストは、Web サーバ (Hitachi Web Server) が受け付けます。
2. Web サーバに登録されているリダイレクタというライブラリによって、リクエストは Web コンテナに振り分けられます。Web コンテナは、Cosminexus Component Container の構成要素の一つです。Web コンテナでは、サーブレットと JSP で構成される Web アプリケーションが動作しています。Web アプリケーションからは、業務処理プログラムである Enterprise

Bean を呼び出すことができます。

3. Application Server Enterprise の場合は、Web アプリケーションから業務処理プログラム (Stateless Session Bean) を呼び出すとき、Cosminexus Component Transaction Monitor (CTM) を使用してスケジューリングをするかどうかで、処理の流れが異なります。CTM によってリクエストのスケジューリングをする場合、Web アプリケーションから送信されたリクエストは、CTM のスケジュールキューに格納されます。CTM では、特定のサーバに負荷が集中しないように、キューを使用してリクエストの負荷分散や流量制御を実行します。
4. 適切にスケジューリングされたリクエストは、業務処理プログラムを実行する EJB コンテナに送信されます。なお、EJB コンテナは、Cosminexus Component Container の構成要素の一つです。また、CTM を使用していない場合は、Web アプリケーションから直接 EJB コンテナにリクエストが送信されます。EJB コンテナでは、リクエストに従って業務処理プログラムが実行されます。
5. EJB コンテナでは、処理内容に応じて、データベースへのアクセスが発生します。アプリケーションサーバでは、Cosminexus Component Container が提供する JCA 機能を使用して、データベースなどのエンタープライズ情報システム (EIS) に接続します。

処理が完了したリクエストの処理結果は、EJB コンテナから、CTM、Web コンテナ、Web サーバを経て、Web クライアントに返却されます。

(b) EJB クライアントの場合

EJB クライアントから送信したリクエストは、直接、CTM (Application Server Enterprise の場合) または EJB コンテナが受け付けます。そのあとの処理の流れは、Web クライアントからの処理の流れと同じです。

(2) 運用管理操作実行時の構成ソフトウェアの位置づけ

運用管理者から送信された Application Server に対する運用操作に関連するリクエストは、運用管理機能 (Cosminexus Manager 機能) が受け付けて、処理を実行します。Application Server の運用管理機能は、Cosminexus Component Container の機能として提供されています。なお、運用管理者からのリクエストは、CUI、Eclipse プラグイン (Cosminexus Server Plug-in) または Web ブラウザから送信されます。

また、「(1) J2EE アプリケーション実行時の構成ソフトウェアの位置づけ」で説明したリクエスト処理の過程では、主要なポイントでトレース情報が出力されています。このトレース情報は、Cosminexus Performance Tracer によって収集され、ファイルに出力されています。運用管理者は、このトレース情報を収集、監視して、アプリケーションサーバ全体の処理性能とボトルネックの確認などができます。

(3) Web サービス実行時の構成ソフトウェアの位置づけ

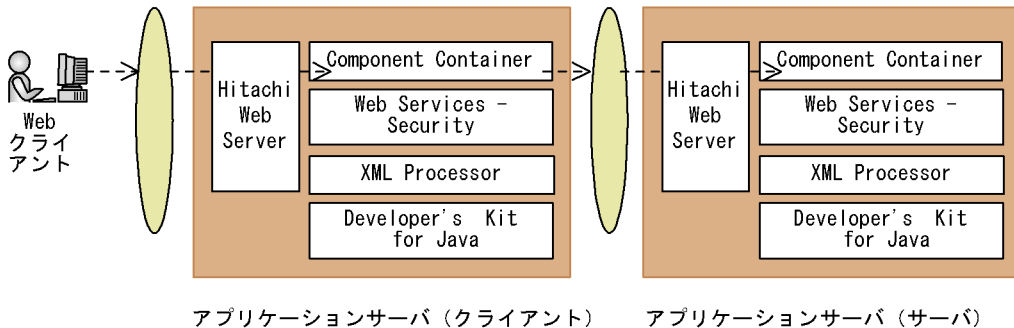
Web サービスの場合、Web クライアントから Web サーバが受け付けたリクエストは、

2. アプリケーションサーバの製品構成

SOAP 通信によって Web サービスのサーバマシンに送信されます。

Web サービスを実行する場合に使用される Application Server の構成ソフトウェアの位置づけについて、次の図に示します。

図 2-11 Application Server の構成ソフトウェアの位置づけ (SOAP アプリケーションを実行する場合)



(凡例)

-----> : リクエストの流れ

注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。
また、Webサービスの実行 (SOAP通信) に主に使用する構成ソフトウェアだけを記載しています。

Web サービスを実行する場合、クライアントとサーバ間の SOAP 通信は、JAX-WS エンジンを通じて実行されます。アプリケーションサーバでは、Cosminexus Component Container が JAX-WS エンジンとして機能します。

また、Cosminexus Component Container では、アプリケーションサーバが 05-00 以前から提供していた SOAP アプリケーション開発支援機能を使用して、Web サービスを実行することもできます。この機能を使用して実行する Web サービスを SOAP アプリケーションといいます。SOAP アプリケーションを使用した SOAP 通信でやり取りする XML データは、Web サービスセキュリティの技術を利用して盗聴または改ざんされることを防ぐことができます。Web サービスセキュリティを利用するための機能は、Cosminexus Web Services - Security が提供しています。

(4) メッセージによる処理の非同期実行時の構成ソフトウェアの位置づけ

メッセージによる処理の非同期実行は、使用する構成ソフトウェアによって、処理の流れが異なります。

参考

ここでは、アプリケーションサーバで提供する機能を使用する方法について説明します。このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用してメッセージによる処理の非同期実行を実現できます。任意のリソースアダプタを使用する場合の構成ソフトウェアおよびリソースアダプタの位置づけについては、使用するリソースアダプタの仕様をご確認ください。

(a) Cosminexus Component Container の Cosminexus JMS プロバイダの機能を使用する方法

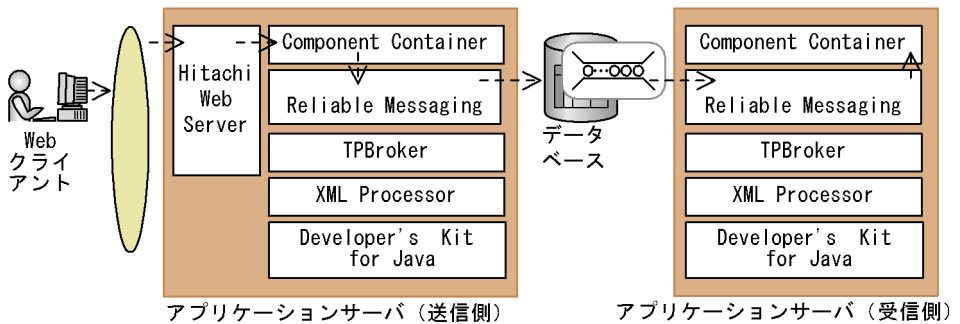
アプリケーションサーバ間でメッセージの非同期通信を実行する場合、送信側のアプリケーションサーバから送信したメッセージは、キューまたはトピックに蓄積されます。受信側のアプリケーションサーバがキューまたはトピックからメッセージを受け取ったときに、メッセージの処理が実行されます。

(b) Cosminexus Reliable Messaging を使用する方法

アプリケーションサーバ間でメッセージの非同期通信を実行する場合、送信側のアプリケーションサーバが受け付けたメッセージは、キューに蓄積されます。受信側のアプリケーションサーバがキューからメッセージを取り出したときに、メッセージの処理が実行されます。

メッセージによる非同期実行をする場合に使用される Application Server の構成ソフトウェアの位置づけを次の図に示します。

図 2-12 Application Server の構成ソフトウェアの位置づけ (Cosminexus Reliable Messaging を使用してメッセージによる処理の非同期実行をする場合)



(凡例)

-----> : リクエストの流れ

注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。また、メッセージによる処理の非同期実行で主に使用する構成ソフトウェアだけを記載しています。

アプリケーションの処理を非同期実行する場合、リクエストはメッセージとしてデータベース上のキューに格納されます。送信側でのメッセージのキューへの登録、および受信側でのキューからのメッセージの取り出しは、Cosminexus Reliable Messaging の機能で実行できます。キューを作成したデータベースへのアクセスには、Cosminexus

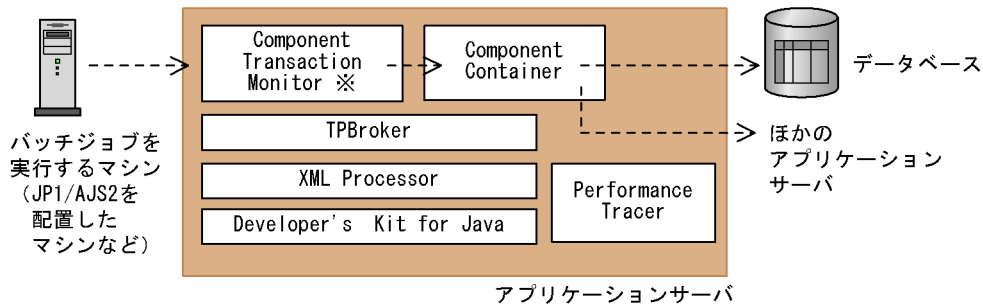
2. アプリケーションサーバの製品構成

Component Container の機能を使用します。

(5) バッチアプリケーション実行時の構成ソフトウェアの位置づけ

アプリケーションサーバ上でバッチアプリケーションを実行する場合に、使用される Application Server の構成ソフトウェアの位置づけを次の図に示します。

図 2-13 Application Server の構成ソフトウェアの位置づけ (バッチサーバでバッチ処理を実行する場合)



(凡例)

----> : バッチジョブの実行の流れ

注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

注※ Application Server Enterpriseの場合だけに含まれる構成ソフトウェアです。

バッチアプリケーションは Cosminexus Component Container の機能を使用して実行されます。また、ほかのアプリケーションサーバ上で動作している Enterprise Bean を呼び出す場合は、Cosminexus TPBroker の機能を使用してリモート呼び出しを実現します。

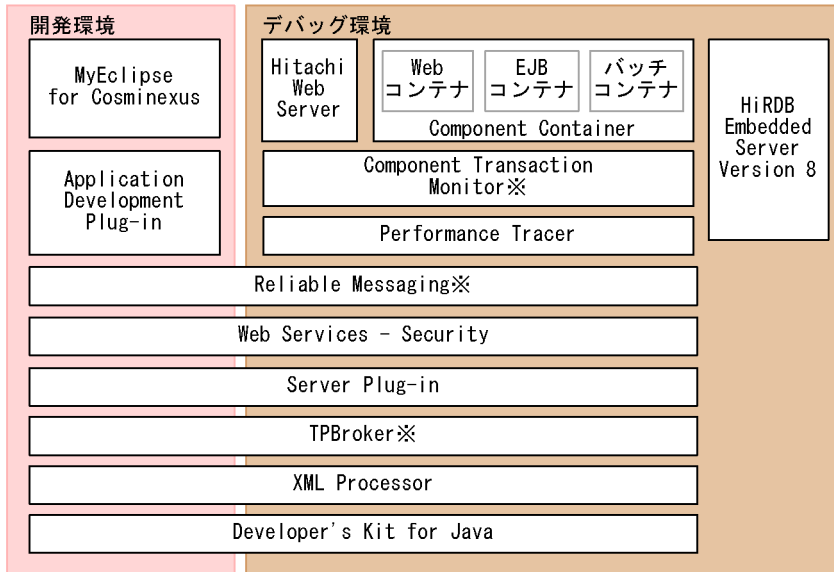
Application Server Enterprise の場合は、バッチアプリケーションの実行を Cosminexus Component Transaction Monitor (CTM) で制御できます。

2.3.4 開発環境の構成ソフトウェアの位置づけ (Developer)

ここでは、開発環境を構築する製品である Developer (Developer Standard または Developer Professional) の各構成ソフトウェアの位置づけについて説明します。

Developer の構成ソフトウェアの位置づけを次の図に示します。

図 2-14 Developer の構成ソフトウェアの位置づけ



注 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

注※ Developer Professionalの場合だけに含まれる構成ソフトウェアです。

「開発環境」に含まれる構成ソフトウェアは、主にアプリケーション開発時に使用する構成ソフトウェアです。「デバッグ環境」に含まれる構成ソフトウェアは、主にアプリケーションのテスト時に使用する構成ソフトウェアです。

(1) アプリケーション開発時に使用する構成ソフトウェアの位置づけ

Developer が提供する開発環境の構成ソフトウェアを使用して、J2EE アプリケーションを開発します。

MyEclipse for Cosminexus は、アプリケーションの統合開発環境です。Cosminexus Application Development Plug-in は、MyEclipse for Cosminexus からデバッグ環境上の J2EE サーバを操作するための機能（プラグイン）を提供します。

アプリケーション開発時には、各構成ソフトウェアが提供する API を使用して J2EE アプリケーションを開発できます。

(2) アプリケーションのテスト時に使用する構成ソフトウェアの位置づけ

Developer が提供するデバッグ環境の構成ソフトウェアを使用して、開発した J2EE アプリケーションを実行、テストし、デバッグをします。Developer 提供のプラグインを使用すると IDE からアプリケーションをデプロイできます。データベースを使用するアプリケーションも、Developer で提供している HiRDB Embedded Server Version 8 を使用してテストできます。

2.4 動作環境

アプリケーションサーバの動作環境について説明します。

(1) 前提 OS

対象製品ごとに、前提 OS を示します。

(a) Application Server Standard および Application Server Enterprise の前提 OS

Application Server Standard および Application Server Enterprise の前提 OS について、次の表に示します。

表 2-4 Application Server Standard および Application Server Enterprise の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 R2 Windows Server 2003 (x64) Windows Server 2003 R2 (x64) Windows Server 2008 R2 Windows Server 2008 x86 Windows Server 2008 x64
AIX	AIX 5L V5.3 AIX V6.1 AIX V7.1
HP-UX	HP-UX 11i V2 (IPF) HP-UX 11i V3 (IPF)
Linux	Red Hat Enterprise Linux AS 4 (x86) Red Hat Enterprise Linux ES 4 (x86) Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux 5 Advanced Platform (x86) Red Hat Enterprise Linux 5 (x86) Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) Red Hat Enterprise Linux 5 (AMD/Intel 64) Red Hat Enterprise Linux Server 6 (32-bit x86) Red Hat Enterprise Linux Server 6 (64-bit x86_64)

(b) uCosminexus Web Redirector の前提 OS

uCosminexus Web Redirector の前提 OS について、次の表に示します。

表 2-5 uCosminexus Web Redirector の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 R2 Windows Server 2003 (x64) Windows Server 2003 R2 (x64) Windows Server 2008 R2 Windows Server 2008 x86 Windows Server 2008 x64
Linux	Red Hat Enterprise Linux AS 4 (x86) Red Hat Enterprise Linux ES 4 (x86) Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) Red Hat Enterprise Linux 5 Advanced Platform (x86) Red Hat Enterprise Linux 5 (x86) Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64) Red Hat Enterprise Linux 5 (AMD/Intel 64) Red Hat Enterprise Linux Server 6 (32-bit x86) Red Hat Enterprise Linux Server 6 (64-bit x86_64)

(c) Developer Standard , Developer Professional および uCosminexus Operator の前提 OS

Developer Standard , Developer Professional および uCosminexus Operator の前提 OS について , 次の表に示します。なお , これらの製品は , UNIX には対応していません。

表 2-6 Developer Standard , Developer Professional および uCosminexus Operator の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows 7 Windows Server 2003 Windows Server 2003 R2 Windows Vista Windows XP

(d) uCosminexus Client の前提 OS

uCosminexus Client の前提 OS について , 次の表に示します。なお , uCosminexus Client は , UNIX には対応していません。

2. アプリケーションサーバの製品構成

表 2-7 uCosminexus Client の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows 7 Windows Server 2003 Windows Server 2003 R2 Windows Server 2003 (x64) Windows Server 2003 R2 (x64) Windows Server 2008 R2 Windows Server 2008 x86 Windows Server 2008 x64 Windows Vista Windows XP

(2) Web 環境

前提となる Web 環境について示します。なお、インプロセス HTTP サーバを使用する場合、Web サーバは不要です。

(a) Web サーバ

前提となる Web サーバについて次の表に示します。

表 2-8 アプリケーションサーバの前提となる Web サーバ

前提 OS	前提製品
Windows	Microsoft IIS Hitachi Web Server
UNIX	Hitachi Web Server

注 アプリケーションサーバの構成ソフトウェアです。

(b) Web ブラウザ

前提となる Web ブラウザについて、次の表に示します。

表 2-9 アプリケーションサーバの前提となる Web ブラウザ

Web 環境	前提製品
Web ブラウザ	HTTP1.0 または HTTP1.1 に対応したブラウザ (Internet Explorer 6.0 以降、または Mozilla FireFox2 以降)

(3) 言語

前提となる言語は、Java です。

(4) データベース

前提となるデータベースについては、「7.1 データベースとの連携」を参照してください。

3

システムのライフサイクル

この章では、アプリケーションサーバで実現するシステムのライフサイクル、およびライフサイクルを構成する段階について説明します。また、それぞれの段階での作業の詳細について知りたい場合に参照するマニュアルを紹介します。

3.1 実行環境および開発環境のライフサイクル

3.2 システムの仕様（機能）検討

3.3 システム設計と運用設計

3.4 アプリケーションの開発

3.5 システムの構築

3.6 システムの運用と保守

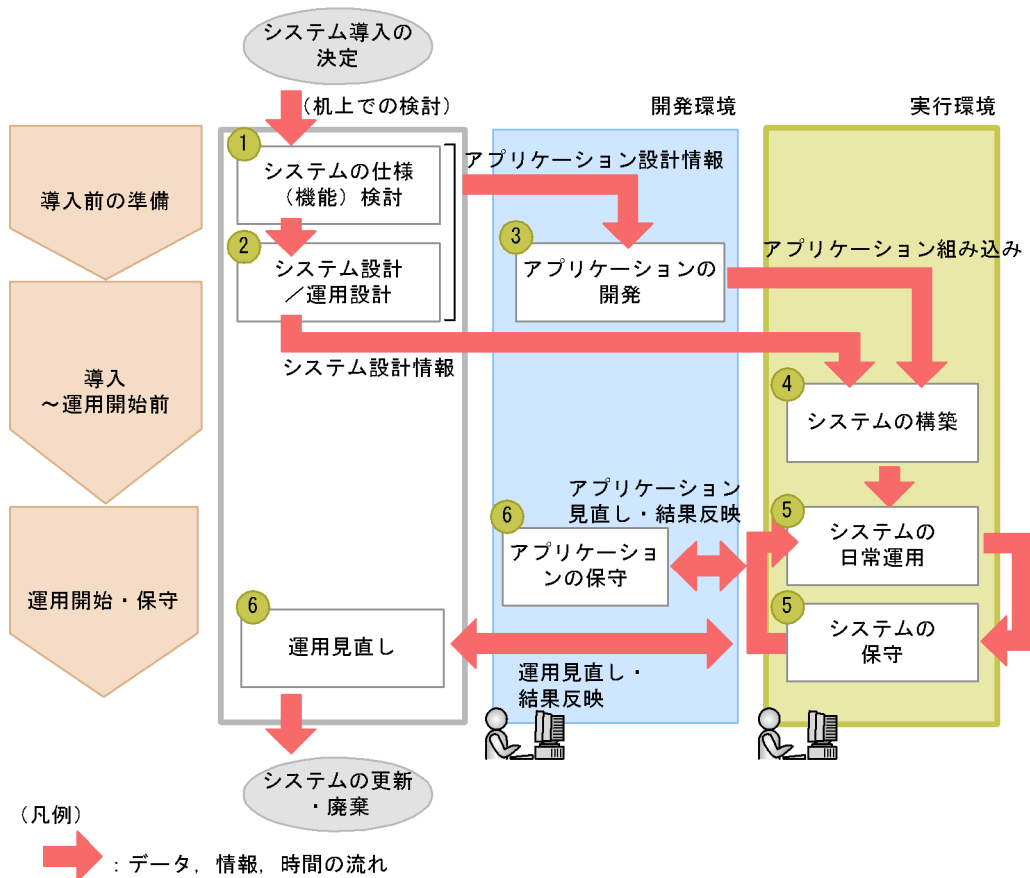
3.1 実行環境および開発環境のライフサイクル

実行環境および開発環境のシステムのライフサイクルには、次の段階があります。

1. システムの仕様（機能）検討
2. システム設計 / 運用設計
3. アプリケーションの開発
4. システムの構築
5. システムの日常運用と保守
6. 運用見直しとアプリケーションの保守

アプリケーションの開発環境および実行環境でのシステムのライフサイクルを図に表すと、次のようになります。

図 3-1 システムのライフサイクル



それぞれの段階について説明します。

1. システムの仕様（機能）検討
システムの目的や業務の内容に応じて、アプリケーションサーバのどの機能を使用するかを検討・決定します。具体的な作業内容については、「3.2 システムの仕様（機能）検討」で説明します。
2. システム設計 / 運用設計
システムの目的や実行する業務、およびシステムの使用方法に応じてシステムの構成と運用方法を設計します。具体的な作業内容については、「3.3 システム設計と運用設計」で説明します。
3. アプリケーションの開発
 1. および 2. で検討したアプリケーション設計情報に基づいて、業務に合ったアプリケーションを開発します。具体的な作業内容については、「3.4 アプリケーションの開発」で説明します。
4. システムの構築
 1. および 2. で検討したシステム設計情報に基づいて、システムを構築します。また、
 3. で開発したアプリケーションをシステムに組み込みます。具体的な作業内容については、「3.5 システムの構築」で説明します。
5. システムの日常運用と保守
アプリケーションを実行して、システムの運用を開始します。システムの運用では、日常運用と定期的なシステム保守を繰り返します。具体的な作業内容については、「3.6 システムの運用と保守」で説明します。
6. 運用見直し / アプリケーション保守
業務内容の変更やシステム規模の変更などに合わせて、運用方法を随時見直します。また、必要に応じてアプリケーションも保守します。具体的な作業内容については、「3.6 システムの運用と保守」で説明します。

ライフサイクルは、システムの更新または廃棄を決定するまで続きます。

3.2 システムの仕様（機能）検討

システムの目的や業務の内容に応じて、実行環境でどの機能を使用するかを検討・決定します。

使用する機能の検討では、機能について説明しているマニュアルを必要に応じて参照してください。使用できる機能は、J2EE アプリケーションを実行するシステムを構築するか、バッチアプリケーションを実行するシステムを構築するかによって異なります。

J2EE アプリケーションを実行するシステムの場合に使用できる主な機能および参照先のマニュアルについて次の表に示します。

表 3-1 アプリケーションサーバの主な機能および参照先のマニュアル（J2EE アプリケーションを実行するシステムの場合）

機能分類	概要	参照先マニュアル	参照箇所
Java 言語	J2SE 5.0 または Java SE 6 に対応しています。 ¹	- ²	- ²
Web サーバ	Apache 2.0 に対応した Web サーバ（Hitachi Web Server）を使用できます。	Hitachi Web Server	
Web コンテナ	Servlet 2.5 および JSP 2.1 に対応した Web コンテナを使用できます。また、インプロセス HTTP サーバ機能など、独自の機能を提供しています。	Cosminexus アプリケーションサーバ機能解説 基本・開発編（Web コンテナ）	全体
EJB コンテナ	次の Enterprise Bean を実行できる EJB コンテナを提供しています。 <ul style="list-style-type: none"> • Session Bean • Entity Bean • Message-driven Bean また、Timer Service の機能も使用できます。	Cosminexus アプリケーションサーバ機能解説 基本・開発編（EJB コンテナ）	全体
リソース接続とトランザクション管理	次の機能に対応したリソース接続とトランザクション管理ができます。 <ul style="list-style-type: none"> • コネクションプーリング • コネクションシェアリング • JNDI によるルックアップ • ローカルトランザクション • 2 フェーズコミットメント • 分散トランザクションの実現 また、OpenTP1 との接続や JMS 仕様に準拠した送信先（キューまたはトピック）との接続も実現できます。	Cosminexus アプリケーションサーバ機能解説 基本・開発編（コンテナ共通機能）	2.3, 3.4.2, 3.14.1, 3.14.3, 7.
スレッドの非同期並行処理	次の機能を使用できます。 <ul style="list-style-type: none"> • TimerManager を使用した非同期タイマ処理 • WorkManager を使用した非同期スレッド処理 	Cosminexus アプリケーションサーバ機能解説 拡張編	13.3, 13.4

機能分類	概要	参照先マニュアル	参照箇所
XML プロセッサ	次の機能を使用できます。 <ul style="list-style-type: none"> • DOM API • SAX API • XSLT トランスフォーマ (XPath を含む) 	Cosminexus XML Processor ユーザーズガイド	全体
Web サービス	JAX-WS 仕様に対応した Web サービスの実行および開発ができます。	Cosminexus アプリケーションサーバ Web サービス開発の手引	全体
Web サービスおよび SOAP 間通信 (既存機能)	次の機能を使用できます。 <ul style="list-style-type: none"> • RPC を利用した SOAP アプリケーションの開発 • メッセージングを利用した SOAP アプリケーションの開発 • EJB を利用した SOAP アプリケーションの開発 • UDDI クライアントライブラリが提供する JAXR API 	Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引	全体
OLTP 技術の適用	次の機能を使用できます。 <ul style="list-style-type: none"> • Web アプリケーションおよび URL グループ (業務ロジック) 単位の同時実行スレッド数制御 • Enterprise Bean の同時実行スレッド数制御 (CTM の利用) ³ 	Cosminexus アプリケーションサーバ機能解説 基本・開発編 (Web コンテナ)	2.15
		Cosminexus アプリケーションサーバ機能解説 拡張編	1.2.2
可用性向上	次の機能を使用できます。 <ul style="list-style-type: none"> • J2EE サーバ間のセッション情報の引き継ぎ • リソース枯渇監視 • 稼働情報監視 • 性能解析トレース / 障害解析トレースの出力 	Cosminexus アプリケーションサーバ機能解説 拡張編	5. , 7.15.1
		Cosminexus アプリケーションサーバ機能解説 運用 / 監視 / 連携編	4.2 , 8.4
セキュリティ管理	次の機能を使用できます。 <ul style="list-style-type: none"> • 統合ユーザ管理 • 監査ログ出力 • データベース監査証跡連携 	Cosminexus アプリケーションサーバ機能解説 拡張編	10.
		Cosminexus アプリケーションサーバ機能解説 運用 / 監視 / 連携編	6. , 7.
セキュリティ管理 (Web サービス) ³	次の機能を使用できます。 <ul style="list-style-type: none"> • SOAP メッセージの完全性および秘匿性の保証 • SOAP メッセージの認証 • XML 署名データの生成および検証 • XML 署名データの暗号化および復号化 	Cosminexus アプリケーションサーバ Web サービスセキュリティ使用の手引	1.2
		Cosminexus XML Security - Core ユーザーズガイド	全体

3. システムのライフサイクル

機能分類	概要	参照先マニュアル	参照箇所
フレームワークとライブラリ	次の機能を使用できます。 <ul style="list-style-type: none"> • JSF および JSTL • アプリケーションサーバ提供のはん用部品 	- 2	- 2
アプリケーション開発	次の機能を使用できます。 <ul style="list-style-type: none"> • MyEclipse を使用したアプリケーションの開発 • ローカルマシンでのデバッグおよびリモートマシンでのデバッグ 	Cosminexus アプリケーションサーバ アプリケーション開発ガイド	全体
システム構築	次の機能を使用できます。 <ul style="list-style-type: none"> • システム構成の一括定義と簡易構築 (Smart Composer 機能またはセットアップウィザード) • リモート環境からの各種サーバの設定 • アーカイブ形式または展開ディレクトリ形式でのアプリケーションのデプロイ • リソースアダプタのデプロイ • アプリケーションの設定 • リソースの設定 • 複数実行環境の切り替え (UNIX の場合) • 実行環境の移行性向上 (構築済みのシステムの移行) 	Cosminexus アプリケーションサーバ システム構築・運用ガイド	2., 8.
		Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	9., 10.
		Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)	12.
		Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	全体
		Cosminexus アプリケーションサーバ 機能解説 拡張編	11.
		Cosminexus アプリケーションサーバ システム構築・運用ガイド	9.9
		Cosminexus アプリケーションサーバ システム構築・運用ガイド	9.3
システム運用	次の機能を使用できます。 <ul style="list-style-type: none"> • サーバプロセスの一括起動, 個別起動, 自動起動, 監視, 再起動 • JSP 事前コンパイル • J2EE アプリケーションのリロード • 稼働情報監視によるイベント発行と自動アクション制御 	Cosminexus アプリケーションサーバ システム構築・運用ガイド	9.3
		Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)	2.5
		Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)	12.8
		Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	9.

機能分類	概要	参照先マニュアル	参照箇所
	<ul style="list-style-type: none"> ログ/トレース収集 ドメイン一括管理 	Cosminexus アプリケーションサーバ 機能解説 保守/移行/互換編 Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	1.2.1 11.3
クラスタソフトウェアとの連携	次のクラスタシステムを運用できます。 <ul style="list-style-type: none"> アプリケーションサーバまたは運用管理サーバを対象にした、コールドスタンバイでの 1:1 の系切り替え 相互スタンバイ構成 1 台のリカバリ専用サーバを用意した N:1 リカバリシステム構成 ホスト単位管理モデルを対象にしたコールドスタンバイでの系切り替え 	Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編	18.2
JP1 との連携	JP1 の各製品と連携して次の機能を使用できます。 <ul style="list-style-type: none"> 障害の集中監視 稼働性能の監視 システム構成定義と管理 ジョブによる運用の自動化 シナリオによる運用の自動化 SNMP での稼働情報の取得 監査ログの収集と一元管理 	Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編	12.

注 1 製品の型名によって使用できるバージョンが異なります。

注 2 対応するマニュアルはありません。

注 3 Application Server Standard では使用できません。

バッチアプリケーションを実行するシステムの場合に使用できる主な機能および参照先のマニュアルについて次の表に示します。

表 3-2 アプリケーションサーバの主な機能および参照先のマニュアル（バッチアプリケーションを実行するシステムの場合）

機能分類	概要	参照先マニュアル	参照箇所
Java 言語	J2SE 5.0 または Java SE 6 に対応しています。 ¹	2	2
バッチサーバ	次の機能を実現できるバッチサーバを提供しています。 <ul style="list-style-type: none"> バッチ処理を実装した Java アプリケーションの実行 フルガーベージコレクション実行制御 	Cosminexus アプリケーションサーバ 機能解説 拡張編	2.2, 8.

3. システムのライフサイクル

機能分類	概要	参照先マニュアル	参照箇所
リソース接続 とトランザク ション管理	次の機能に対応したリソース接続とトランザクション管理ができます。 <ul style="list-style-type: none"> コネクションプーリング コネクションシェアリング JNDI によるルックアップ <ul style="list-style-type: none"> ローカルトランザクション 	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)	2.3, 3.4.2, 3.14
		Cosminexus アプリケーションサーバ 機能解説 拡張編	2.
XML プロセッサ	次の機能を使用できます。 <ul style="list-style-type: none"> DOM API SAX API XSLT トランスフォーマ (XPath を含む) 	Cosminexus XML Processor ユーザーズガイド	全体
可用性向上	次の機能を使用できます。 <ul style="list-style-type: none"> リソース枯渇監視 稼働情報監視 性能解析トレース / 障害解析トレースの出力 	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	3., 4.
セキュリティ管理	次の機能を使用できます。 <ul style="list-style-type: none"> 監査ログ出力 データベース監査証跡連携 	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	6., 7.
アプリケーション開発	次の機能を使用できます。 <ul style="list-style-type: none"> MyEclipse を使用したアプリケーションの開発 ローカルマシンでのデバッグおよびリモートマシンでのデバッグ 	Cosminexus アプリケーションサーバ アプリケーション開発ガイド	13.
システム構築	次の機能を使用できます。 <ul style="list-style-type: none"> システム構成の一括定義と簡易構築 (Smart Composer 機能) リモート環境からの各種サーバの設定 リソースアダプタのデプロイ リソースの設定 複数実行環境の切り替え (UNIX の場合) 実行環境の移行性向上 (構築済みのシステムの移行) 	Cosminexus アプリケーションサーバ システム構築・運用ガイド	10.
		Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	9., 10.
		Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	全体
		Cosminexus アプリケーションサーバ 機能解説 拡張編	11.
		Cosminexus アプリケーションサーバ システム構築・運用ガイド	11.6
システム運用	次の機能を使用できます。 <ul style="list-style-type: none"> サーバプロセスの一括起動, 個別起動, 自動起動, 監視, 再起動 	Cosminexus アプリケーションサーバ システム構築・運用ガイド	11.2

機能分類	概要	参照先マニュアル	参照箇所
	<ul style="list-style-type: none"> 稼働情報監視によるイベント発行と自動アクション制御 	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	9.
	<ul style="list-style-type: none"> ログ / トレース収集 	Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	1.2.1
	<ul style="list-style-type: none"> ドメイン一括管理 	Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	11.3
クラスタソフトウェアとの連携	<p>次のクラスタシステムを運用できます。</p> <ul style="list-style-type: none"> アプリケーションサーバを対象にした、コールスタンバイでの 1:1 の系切り替え 相互スタンバイ構成 ホスト単位管理モデルを対象にしたコールスタンバイでの系切り替え 	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	19. , 20.4 , 20.5
JP1 との連携	<p>JP1 の各製品と連携して次の機能を使用できます。</p> <ul style="list-style-type: none"> 障害の集中監視 稼働性能の監視 システム構成定義と管理 ジョブによる運用の自動化 シナリオによる運用の自動化 SNMP での稼働情報の取得 監査ログの収集と一元管理 	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	12. , 13. , 14. , 15.

注 1 製品の型名によって使用できるバージョンが異なります。

注 2 対応するマニュアルはありません。

3.3 システム設計と運用設計

アプリケーションサーバのシステム設計と運用設計では、次の点を検討します。

システム構成をどうするか / システムのスケーラビリティをどのように確保するか
まず、J2EE アプリケーションによってオンライン処理を実行するシステムか、バッチアプリケーションによってバッチ処理を実行するシステムかを明確にします。
次に、システムの規模に応じて、Web サーバ、J2EE サーバ、バッチサーバおよび各プロセスを物理的にどのように配置するかを決めます。オンライン処理を実行するシステムの場合は、複数のサーバを使用して負荷を分散する必要があるかなどを検討します。スケーラビリティについてもよく検討しておく必要があります。実際の日常運用が開始されたあとで、システム規模を変更したり、トラブル発生時の影響をできるだけ局所的に抑えたりするためには、システムのスケーラビリティが確保されていることが重要です。

どのような運用のしかたをするか / 可用性と信頼性をどのように確保するか
業務システムを運用する場合には、システムの可用性と信頼性を確保しながら、できるだけ効率良く運用していく方法を検討します。アプリケーションサーバには、システム内の複数のサーバマシンを一括して管理、運用するための機能として、Management Server という運用管理機能があります。
また、システム全体を効率良く運用する方法として、JP1 やクラスタソフトウェアと連携するかどうか、あわせて検討してください。

システムのセキュリティをどのように確保するか
ミッション・クリティカルな業務システムの場合、信頼性の高いセキュリティを確保することは不可欠です。システム設計の段階で、セキュアなシステムを構築するための観点を明確にして、導入・運用時の手順を決めておく必要があります。また、外部ネットワークと接続するシステムを構築する場合は、ファイアウォール、侵入検知システムおよび SSL アクセラレータを適切に配置、設定してどのように信頼性の高いシステムを構築するかを、あわせて検討してください。

性能を向上させるためにどのようなチューニングが必要か
システム要件によっては、厳密な性能設計が必要です。プールやキャッシュを効果的に利用したり、タイムアウトを適切に設定したりすることで、システム全体の性能向上が図れます。

JavaVM をどのようにチューニングするか
アプリケーションサーバで動作する J2EE サーバなどのプロセスは、JavaVM 上で実行されます。JavaVM で使用するメモリ空間を適切に管理することによって、フルガベージコレクションの頻発を防ぎ、パフォーマンスの低下を防げます。

システム設計および運用設計の方法の詳細については、マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」を参照してください。

3.4 アプリケーションの開発

目的とする業務内容に合わせて、アプリケーションを開発します。

アプリケーションサーバでは、Java 言語を使用して開発した J2EE アプリケーションやバッチアプリケーションを実行できます。

一般的な J2EE アプリケーションは、MVC アーキテクチャに基づいて、コンポーネント化されたプログラム群によって構成されます。個々の業務処理プログラムは、EJB の仕様に従った Enterprise Bean として作成します。Enterprise Bean の組み合わせによって複雑なアプリケーションを開発したり、業務の変化に応じて Enterprise Bean を入れ替えたりすることで、再利用性の高い J2EE アプリケーションが開発できます。

アプリケーションサーバで構築したシステムで動作するアプリケーションは、次のような方法で開発できます。

Developer の機能を使用して開発する

Developer を使用する場合、Eclipse のプラグインである MyEclipse または WTP を使用してアプリケーションを開発できます。

MyEclipse では、UML を使用したアプリケーションの設計から、コーディング、ビルド、デバッグまでの一連の開発作業を実行できます。

なお、構築したテスト環境上の J2EE サーバの起動・停止や、アプリケーションの開始・停止などの操作も、MyEclipse から実行できます。

また、Developer では、WTP を使用したアプリケーションの開発もサポートしています。GUI を使用して `cosminexus.xml` を作成したり、構築したテスト環境上の J2EE サーバを操作したりできます。

Developer および IDE を使用しないで開発する

Developer や IDE を使用しなくても、アプリケーションは開発できます。この場合は、テキストエディタを使用してプログラムのソースを作成し、`javac` コマンドでコンパイルします。また、`jar` コマンドでアーカイブを作成して、アプリケーションを開発します。

アプリケーションの開発方法の詳細については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」を参照してください。

3.5 システムの構築

システム設計の結果を基に、システムを構築します。

システム構築とは、アプリケーションサーバの実行環境を J2EE アプリケーションまたはバッチアプリケーションが実行できる状態にすることです。J2EE アプリケーションを実行するシステムを構築する場合は、Application Server のインストール、J2EE サーバの設定、Web サーバとの連携、J2EE リソースと J2EE アプリケーションの設定、運用環境の構築などが含まれます。バッチアプリケーションを実行するシステムを構築する場合は、Application Server のインストール、バッチサーバの設定、リソースの設定などが含まれます。

システムは、Smart Composer 機能が提供しているコマンドを使用して構築します。Smart Composer 機能のコマンドを使用すると、J2EE サーバ、バッチサーバ、Web サーバなどのサーバプロセスを、一括で構築できます。また、J2EE アプリケーションを実行するシステムを構築する場合は、ウィザードプログラムによってシステムを構築する機能である、セットアップウィザードを使用することもできます。

Smart Composer 機能やセットアップウィザードを使用して構築したシステムは、Management Server というサーバプロセスの機能を使用して一括管理できます。

システム構築の概要と構築手順については、マニュアル「Cosminexus アプリケーションサーバシステム構築・運用ガイド」の「8. J2EE アプリケーションを実行するシステムの構築」を参照してください。また、機能ごとの詳細な設定方法については、次に示すマニュアルの中から、使用する機能に応じたマニュアルを参照してください。

- マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)」
- マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)」
- マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)」
- マニュアル「Cosminexus アプリケーションサーバ 機能解説 拡張編」
- マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編」
- マニュアル「Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編」

3.6 システムの運用と保守

アプリケーションの開発とシステムの構築ができれば、アプリケーションを実行して、運用を開始します。システムの運用では、日常の運用とシステム保守を繰り返します。

日常運用では、日常的なサーバの起動/停止のほか、システムを安定稼働させるために各種プロセスの監視、ログの収集、ユーザの管理などをします。

システムの保守では、スケールイン、スケールアウト、スケールアップ、スケールダウンなどのシステム規模の変更をしたり、発生したトラブルに対処（トラブルシューティング）したりします。

Smart Composer 機能を使用して構築したシステムでは、複数のホスト、プロセスおよびアプリケーションを、一括操作および一括監視できます。一括操作および一括監視には、Management Server の運用管理コマンド、または Smart Composer 機能のコマンドが使用できます。

システムの運用方法のうち、システムの起動、停止の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編」の「2. システムの起動と停止」を参照してください。ステータス監視などの作業の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編」の「2.5.2 ステータスの監視」を参照してください。

なお、運用方法について見直す場合は、必要に応じてシステム構築からやり直してください。

また、さらに広範囲なシステム全体をまとめて統合運用するために、JP1 を使用することもできます。JP1 と連携して実現できる運用については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編」の「12.2 JP1 との連携」を参照してください。それぞれの JP1 製品の詳細な機能および設定手順については、使用する JP1 製品のマニュアルを参照してください。

4

J2EE アプリケーションを実行する

この章では、J2EE アプリケーションを実行するための環境の概要について説明します。J2EE アプリケーションの実行環境は、Application Server Standard、Application Server Enterprise または Service Platform を使用して構築・運用します。

なお、開発環境用の製品である Developer Standard、Developer Professional、または Service Architect で構築したデバッグ環境でも、この章で説明する機能を使用できます。

4.1 J2EE アプリケーションの実行環境の特長

4.2 J2EE アプリケーションの実行環境の種類

4.3 J2EE アプリケーションの実行環境を構築・運用する

4.4 Web サービスに対応したアプリケーションの実行環境を構築・運用する

4.5 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する

4.6 導入から運用までの流れ

4.1 J2EE アプリケーションの実行環境の特長

この節では、J2EE アプリケーションの実行環境の特長について説明します。

4.1.1 標準仕様への対応

アプリケーションサーバは、Java EE 5 の標準仕様に準拠しています。

Java EE に準拠した実行環境では、複数の業務で共通に使用される機能を、Java EE の API を使用して実現できます。共通に使用される機能とは、例えば、データベースやメインフレームに接続する機能や、セッション管理機能、トランザクション管理機能などです。Java オブジェクトとリレーショナルデータベースとのマッピング (O/R マッピング) に関する仕様である、JPA の API も使用できます。

これらの機能は、アプリケーションサーバに含まれるモジュールである、コンテナ、サービスなどの形式で、アプリケーションに提供されます。Java EE の API を使用することで、個々の J2EE アプリケーション内での煩雑なコーディングを減らすことができます。さらに、Java EE 5 では、DD を省略して、アプリケーションの動作に必要な属性をアノテーションで指定することもできます。

このほか、アプリケーションサーバで構築したシステムは、XML、Web サービス、SSL、分散オブジェクトなどに関連した標準仕様にも対応しています。アプリケーションサーバが対応する標準仕様については、「付録 A アプリケーションサーバが対応する標準仕様」を参照してください。

4.1.2 システムの安定稼働の実現

業務システムには、安定して稼働し続けることが求められます。業務の内容によっては、一定時間にアクセスが急増したり、特定の処理に要求が集中したりする場合があります。急激な状況の変化に即応して、システムを安定稼働させるためには、次の制御が有効です。

流量制御

クライアントからのアクセス要求や処理要求などのリクエストに対して、処理の同時実行数を制御することで、リクエスト増加時のパフォーマンス低下を防いでシステムを安定稼働させる方法です。

優先制御

クライアントからのリクエスト増加時に、特定の処理要求を優先して処理することで、処理の緊急度や優先度に応じたリクエスト処理を実現する方法です。

負荷分散

リクエストを処理するサーバを 1 か所に集中させないで、複数の実行環境で分散させることで、パフォーマンスの低下を防いでシステムを安定稼働させる方法です。

アプリケーションサーバでは、J2EE アプリケーションに対して、きめ細やかな流量制御、優先制御を実現できます。これによって、システムの安定稼働に加えて、システムリソースの有効活用も実現できます。

ここでは、Web アプリケーションの流量制御、および OLTP 技術を適用した Enterprise Bean の流量制御と負荷分散について説明します。なお、それぞれの機能の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)」の「2.15 同時実行スレッド数の制御の概要」およびマニュアル「Cosminexus アプリケーションサーバ 機能解説 拡張編」の「3. CTM によるリクエストのスケジューリングと負荷分散」を参照してください。

(1) Web アプリケーションの流量制御

アプリケーションサーバでは、サーブレットや JSP によって構成される Web アプリケーションを、J2EE サーバの Web アプリケーション実行基盤である Web コンテナで実行します。

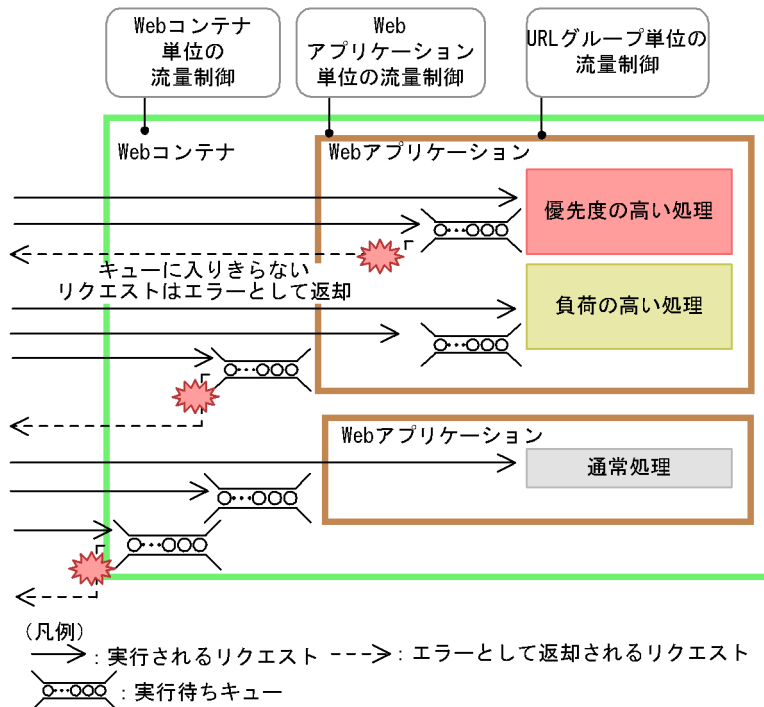
アプリケーションサーバの Web コンテナでは、Web コンテナ単位、Web アプリケーション単位および Web アプリケーション内の URL グループ (業務ロジック) 単位に同時に処理できるリクエスト数を設定したり、リクエストをキュー (待ち行列) の概念で管理したりできます。これによって、Web アプリケーションで同時に実行する処理数を、処理内容に応じて細かく制御できます。送信されたリクエスト数が突発的に急増した場合でも、処理するリクエスト数を一定の数に制御できるので、システムを安定した状態で稼働させることができます。

また、リクエストの重要度に応じたリクエストの実行が可能になります。例えば、緊急度の高い重要な処理は確実に処理できるようにすることで、負荷の高い業務処理がほかの業務に影響を与えることを防げます。

Web アプリケーションの流量制御の概要を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-1 Web アプリケーションの流量制御の概要



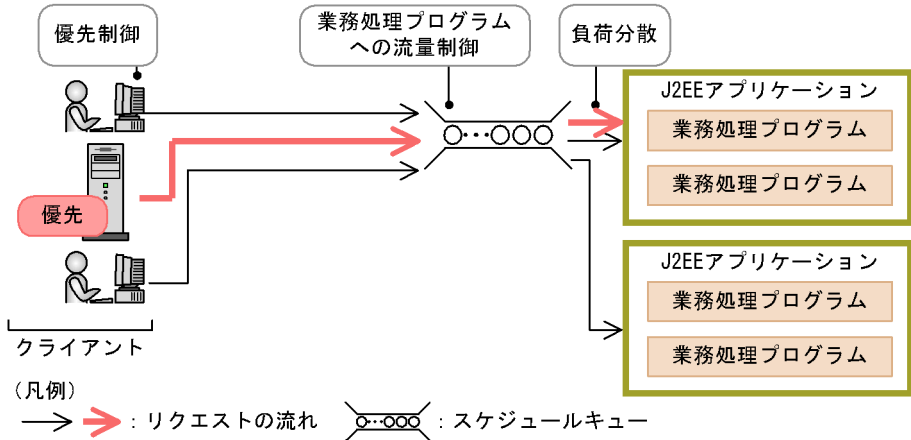
Web コンテナ単位、Web アプリケーション単位および URL グループ（業務ロジック）単位で流量制御を実施することによって、システムの安定稼働を実現しながら、負荷の高い処理があっても優先度の高い処理を確実に実行できるシステムを構築できます。

(2) OLTP 技術の Enterprise Bean への適用

OLTP 技術は、大規模な業務システムには不可欠な技術です。アプリケーションサーバでは、J2EE アプリケーション内の業務処理プログラム（Enterprise Bean）に対して、高度な OLTP 技術を適用できます。これによって、次のような処理を実現できます。なお、対象となる Enterprise Bean は、Stateless Session Bean です。

リクエストのスケジューリングを利用した流量制御と負荷分散
業務処理プログラムに対するリクエストをスケジューリングすることで、特定の J2EE アプリケーションに大量のリクエストが集中した場合に、リクエストを複数のサーバに振り分けて処理させたり、一度に処理させるリクエストの数（流量）を制御したりできます。また、リクエストの送信元であるクライアントに優先順位を設定して、そのクライアントから送信されたリクエストを優先的に処理するようにできます。さらに、特定の業務処理プログラムに対する処理を適切に分散させて負荷の集中を防ぐことで、システム全体としての処理性能の向上と、システムの安定稼働を図れます。Enterprise Bean の優先制御、流量制御および負荷分散の概要を次の図に示します。

図 4-2 Enterprise Bean の優先制御，流量制御および負荷分散の概要



J2EE アプリケーション内の業務処理プログラムへのリクエストを，スケジュールキューを経由させて実行することで，一度に大量のリクエストが送信された場合も同時に実行する数を制御したり，負荷分散したりできます。また，優先制御によって，重要なリクエストを速やかに確実に実行できます。

サービス閉塞の実現

J2EE アプリケーションの稼働中に特定の業務処理プログラムを入れ替えたい場合に，対象の業務処理プログラムに対応するリクエストを制御して，関連するサービスだけを安全に閉塞できます。特定の業務処理プログラムに障害が発生した場合などに，該当箇所を局所化し，縮退運転と回復によって，システム全体を止めないで業務処理プログラムの入れ替えができます。

4.1.3 可用性と耐障害性の向上

ミッションクリティカルな業務システムには，システムが提供するサービスをできるだけ停止することなく，安定して提供し続けられる仕組みが求められます。

例えば，提供するサービスの内容によっては，24 時間連続稼働が求められるサービスや，障害が発生して業務システムが止まることで大きな損失が発生するサービスもあります。

このため，業務システムの基盤であるアプリケーションサーバには，まず，障害を未然に防ぐこと，そして，もし障害が発生しても，障害の影響範囲を局所的に抑え，業務システムを止めることなく運用し続けられることが求められます。また，障害が発生した個所は迅速に回復できることが必要です。

ここでは，アプリケーションサーバで実現できる，可用性と耐障害性の高いシステムの特長について説明します。

(1) フルガーベージコレクションの発生抑止

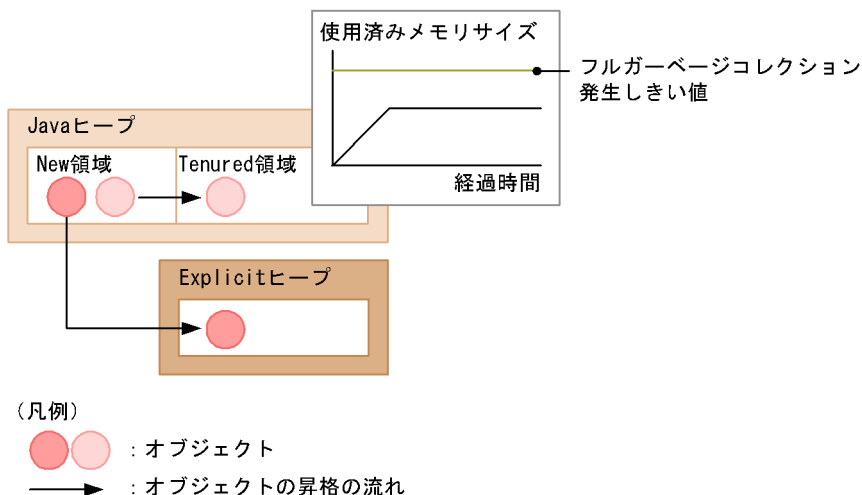
フルガーベージコレクションの発生を抑止することによって、システムが停止する回数を低減します。

フルガーベージコレクションは、Java ヒープの Tenured 領域のオブジェクトが増加することで発生します。Tenured 領域には、長寿命なオブジェクト（利用期間が長いオブジェクト）が配置されます。アプリケーションサーバでは、フルガーベージコレクション発生の要因になる長寿命なオブジェクトのうち、利用期間が明確なオブジェクトを独自のメモリ空間に配置します。このメモリ空間を Explicit ヒープといいます。Explicit ヒープはフルガーベージコレクションの対象にはならないため、これらのオブジェクトによるフルガーベージコレクション発生を抑止できます。

アプリケーションサーバでは、HTTP セッションに関するオブジェクトと、リダイレクタとの通信用オブジェクトを Explicit ヒープに配置します。

HTTP セッションに関するオブジェクトの場合の Explicit ヒープを使用したフルガーベージコレクションの発生抑止の概要を次の図に示します。

図 4-3 Explicit ヒープを使用したフルガーベージコレクションの発生抑止の概要



New 領域から Tenured 領域に昇格するオブジェクトの一部を Explicit ヒープに移動します。これによって、Tenured 領域のメモリサイズ増加を抑え、フルガーベージコレクション発生を抑止します。

このほか、ユーザアプリケーションの修正によって、アプリケーション内でフルガーベージコレクションの要因になっているオブジェクトを直接 Explicit ヒープに生成する実装もできます。

(2) 障害発生時の未然防止

アプリケーションやリソースの稼働状態を監視して、障害の予兆を検知し、障害発生を予防できます。また、しきい値を設定して監視することで、しきい値を超えた場合に自動的に対処するためのアクションも定義できます。

例えば、リクエストの集中によって JavaVM でフルガーベージコレクションが多発する場合に、一定時間内に発生するフルガーベージコレクションの回数にしきい値を設定して監視できます。しきい値を超えた場合にリクエストの同時実行数を動的に減らすなどの処理を自動実行することで、システムのスローダウンを予防できます。

(3) 障害発生時の可用性向上

障害発生時の可用性を高めるためには、次の機能を使用できます。

きめ細やかなタイムアウトの設定

クライアントと Web サーバ間、Web サーバと Web コンテナ間、EJB クライアントと EJB コンテナ間、EJB クライアントとネーミングサービス間などできめ細かくタイムアウトが設定できます。これによって、通信先のマシンに障害が発生している場合でも、数秒で障害を検知でき、無応答などによってサービスを停滞させません。また、アプリケーション内のメソッドレベルの処理にもタイムアウトを設定できます。これによって、無限ループなどの問題がアプリケーション内で発生した場合に速やかにその処理をキャンセルして、業務を継続できます。

クラスタソフトウェアとの連携

クラスタソフトウェアと連携してアプリケーションサーバや運用管理用のサーバをクラスタ構成にすることで、障害発生時に速やかに系切り替えを実行し、サービスの停止を防げます（1:1 クラスタ構成、相互切り替え構成、またはホスト単位管理モデルを対象とした切り替え構成）。

また、リソース使用中のアプリケーションサーバで障害が発生した場合に備え、1 台の待機サーバを用意しておくことで、障害発生時にすぐにリソースを解放しトランザクション処理を決着できます（N:1 クラスタ構成）。

セッション情報の引き継ぎ

Web システムで、すでにクライアントからのセッションが確立されているサーバに障害が発生したときに、セッション情報をほかのサーバに引き継いで、サービスを継続できます。セッション情報はデータベースで管理できます。Web サーバや J2EE サーバに障害が発生した場合も、データベースからセッション情報を回復し、サービスを継続します。

バックエンドシステムとの再接続

データベースなどのバックエンドシステムとの接続が異常終了した場合、自動再接続できます。

詳細なログの出力

アプリケーションサーバでは、日立固有の JavaVM を提供しています。日立固有の JavaVM は、Java オブジェクトの状態に関する詳細なログを出力する機能を備えてい

4. J2EE アプリケーションを実行する

ます。このログは、障害発生の要因分析や、性能向上のためのチューニングに活用できます。

4.1.4 システム導入および拡張の容易化

業務システムは、Web サーバ、J2EE サーバ、負荷分散機、運用管理用のサーバなど、複数の要素で構成されています。また、J2EE サーバ内にも、複数の機能があります。アプリケーションサーバを中心とした業務システムを、特長を生かして円滑に運用するためには、システムの用途に応じたシステム構成を検討し、それぞれの要素の多様なパラメタを、適切な関係で設定する必要があります。

ここでは、システムを導入および拡張するときの、アプリケーションサーバの特長について説明します。

(1) システムの一括構築の実現とスケールアウトの容易化

J2EE アプリケーションを実行するシステムは、次のどちらかの方法で一括して構築できます。

定義ファイルにパラメタを定義してコマンドを実行する方法

Smart Composer 機能を使用する方法です。複数のマシンに配置するサーバを一括構築できます。

対話形式の CUI プログラムの画面に設定値を入力する方法

セットアップウィザードを使用する方法です。対話形式で処理を進めることで、基本的な構成のシステムを構築できます。

構築するシステムの構成によって、これらの方法を使い分けることができます。

また、同じ業務サービスを提供するシステムは、ユニットという概念で管理できます。ユニット単位でシステム構成の変更やスケールアウトができるため、サービス提供規模などの変更に対応して、スムーズにシステムを拡張できます。

(2) テスト環境から本番環境へのスムーズな移行

システムを導入する場合、本番稼働用の実行環境（本番環境）でシステムを構築する前に、デバッグ環境（テスト環境）を構築して、テストおよびチューニングを実施することがあります。アプリケーションサーバでは、稼働中の J2EE サーバ上でテスト用の J2EE アプリケーションを動作させられます。これによって、本番環境と同じ環境でのテストができます。

さらに、テスト環境でチューニング済みの定義をまとめてファイルに出力できます。本番環境を構築するときには、このファイルを本番環境用の定義として使用できます。これによって、テスト環境でチューニングした内容を再度本番環境で設定する必要がなくなり、構築済みのシステムのスムーズな移行を実現できます。

(3) 開発環境でのアプリケーションの属性設定

アプリケーションサーバ独自の定義情報を開発環境で XML ファイル (cosminexus.xml) に定義できます。このファイルを含むアプリケーションを実行環境の J2EE サーバにインポートすることで、実行環境での属性設定が不要になります。

例えば、リソースとのリンク解決などを開発環境で定義できるため、実行環境ではインポートしてすぐにアプリケーションを開始できます。また、DD を変更してアプリケーションを入れ替えるときなどに、アプリケーションサーバ独自の定義をし直す必要がありません。

なお、実行環境でアプリケーションの属性を設定したい場合は、CUI (サーバ管理コマンド) を使用します。

(4) 複数の構築済み実行環境の切り替えが可能

同一製品のバージョンアップやパッチの違い、設定内容の違いなどによるシステムの動作を、1 台のマシンでの検証が必要な場合があります。Cosminexus のアプリケーションサーバでは、バージョンアップまたはパッチ適用の前の実行環境の情報を、一括して退避できます。退避後には、バージョンアップまたはパッチ適用後の別の実行環境を同じマシン内に構築できます。

これによって、複数の構築済み実行環境の切り替えが、効率良く、かつ確実に実行できます。

なお、この機能は、UNIX の場合に使用できる機能です。

4.1.5 システム監査によるシステムのセキュリティ確保

会計不祥事などの問題が多く発生する中で、組織には、内部統制の強化が強く求められています。

内部統制の目的は、いつ、だれが、どんな業務を実行したかを把握して、業務が各種法規制に準拠して遂行されていることを検証することです。このため、内部統制に対応するには、次のことが求められます。

- 正しい権利を持つ担当者が、正しい操作によって適切に業務を実施したことを検証できること。
- 検証結果に問題がないことを、監査者や評価者に対して証明できること。

これらに対応するためには、業務システムで、「だれが」「いつ」「何を実施したか」を監査し、監査結果を記録として管理しておく必要があります。

アプリケーションサーバでは、次の機能を提供しています。

- 監査ログの出力
- データベースと連携した監査証跡情報の出力

ここでは、これらの機能の概要を説明します。監査ログの出力の詳細については、マ

4. J2EE アプリケーションを実行する

マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編」の「6.4 監査ログの出力」を参照してください。データベースと連携した監査証跡情報の出力の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編」の「7.6.2 データベースの監査証跡情報の取得」を参照してください。

(1) 監査ログの出力

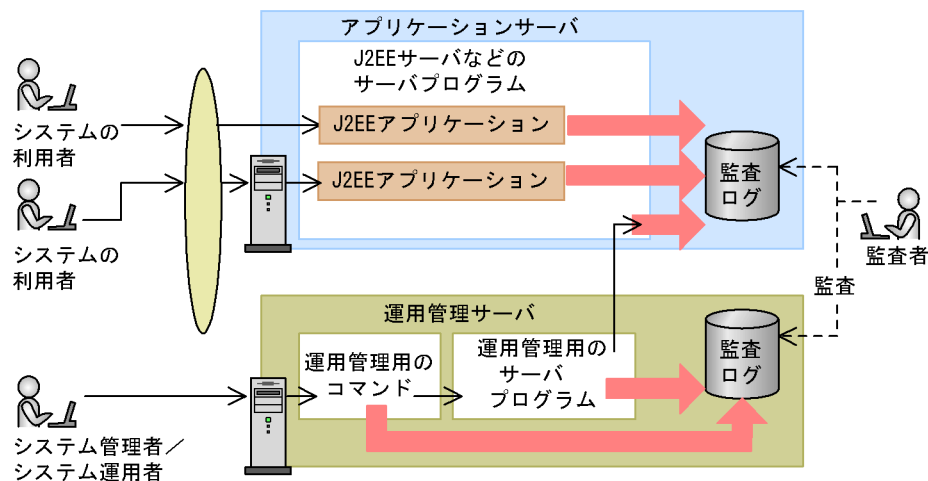
アプリケーションサーバで構築した業務システムでは、「だれが」「いつ」「何を実施したか」についての情報を、監査ログとして出力できます。

監査ログに出力されるのは、次の情報です。

- システム管理者やシステム運用者が実施した操作の履歴と、それに伴うプログラムの動作の履歴
- J2EE アプリケーションを通してシステムの利用者が実行した操作の履歴と、それに伴うプログラムの動作の履歴

アプリケーションサーバでの監査ログ出力の概要を次の図に示します。

図 4-4 アプリケーションサーバでの監査ログ出力の概要



(凡例)

- : アプリケーション実行またはコマンド実行による制御の流れ
- > : 監査の流れ
- ➡ : 監査ログ出力の流れ

なお、アプリケーションサーバが出力した監査ログは、JP1 と連携して、アプリケーションサーバ以外の日立のミドルウェアが出力する監査ログとまとめて管理できます。

JP1 との連携については、「7. ほかの製品との連携」を参照してください。

(2) データベースと連携した監査証跡情報の出力

アプリケーションサーバで構築した業務システムのバックエンドでは、多くの場合、データベースが動作しています。データベースには、漏洩や改ざんが許されない重要なデータが多く格納されています。これらの情報は、適正なセキュリティ管理が行われ、厳重に管理される必要があります。

データベースには、「だれが」「いつ」「どのようなデータベースアクセスを実行したか」を示す情報を出力できるものがあります。この情報を、監査証跡情報といいます。

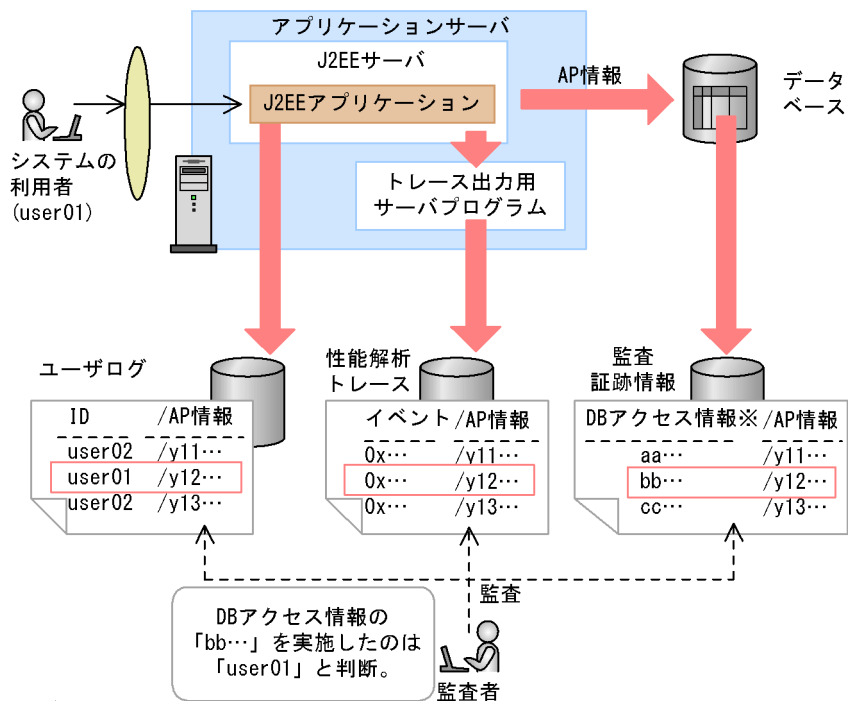
アプリケーションサーバでは、データベースが出力する監査証跡情報に、アプリケーションサーバのどのリクエストでデータベースアクセスが実行されたのかを示す情報を出力できます。この情報と J2EE アプリケーションで出力するログ情報などを組み合わせると、データベースアクセスが、アプリケーションサーバのどのユーザの操作の延長として実行されたのかを追跡できます。

なお、アプリケーションサーバが監査証跡情報を出力するために連携できるデータベースは、HiRDB です。

データベースと連携した監査証跡情報の出力の概要を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-5 データベースと連携した監査証跡情報の出力の概要



注 AP情報は、ルートアプリケーション情報（リクエストを特定するための情報）を示します。

注※ データベースが管理しているデータベースの操作履歴が出力されます。

この例では、システムの利用者（user01）がJ2EEアプリケーションを経由してデータベースにアクセスするとき、次の3種類の情報が出力されています。

- J2EEアプリケーションで出力するユーザログ
- トレース出力用サーバプログラム（パフォーマンストレーサ）が出力する性能解析トレース
- データベースで出力する監査証跡情報

これらの情報には、すべてリクエストを特定するための情報（ルートアプリケーション情報）が出力されています。監査者は、この情報を利用して、データベースアクセスがどのリクエストの延長として実行されたのか、そのリクエストを実行したのはどのユーザなのか、などを検証します。また、性能解析トレースを使用すると、そのリクエストがどのような流れで処理されたのかを検証することもできます。

4.1.6 業務効率を向上させる運用管理の実現

業務システムの運用管理は、規模が大きくなるほど複雑になります。できるだけ効率良く、運用コストを抑え、かつシステムの性能を最大限に生かせる運用方法が求められま

す。そのためには、操作性に優れたインタフェースや、自動運用の仕組みが必要です。アプリケーションサーバでは、これらの要件を満たすために、次の運用管理機能を提供しています。

業務システムの一括運用

Smart Composer 機能またはセットアップウィザードで構築した環境は、運用管理ドメインという概念で管理します。運用管理ドメインは、Management Server というプロセスで管理される範囲です。複数の J2EE サーバや Web サーバなどの各種サーバに対する、起動停止処理などを一括管理できます。

また、JP1 と連携すると、アプリケーションサーバ以外で構築されたシステムも含めた業務システム全体に対して、監視、運用の自動化、稼働状況の分析などの一括運用ができるようになります。

システムの稼働情報やリソースの使用状況の出力

システムの稼働状態やリソースの使用状況を出力できます。出力内容を監視することで、トラブル発生を未然に防止したり、発生したトラブルに対処したりできます。また、出力された情報は、システムのチューニングにも利用できます。これらの情報は、ファイルに出力することもできます。

運用作業の自動化

障害発生時のサーバ再起動や、運用監視機能と連携したイベント発行・アクション実行などの機能を利用することで、障害発生時の回復作業の自動化や、障害予防作業の自動化を図れます。

また、JP1 と連携することで、アプリケーションサーバ以外のシステムも含めた業務システム全体の運用作業を自動化できます。

ログの運用

アプリケーションサーバのシステムが出力したログを一括して収集できます。また、ユーザが開発した J2EE アプリケーションのログをアプリケーションサーバのシステムが出力したログと同じように扱うことができます。システム全体のログを一括管理することで、信頼性の高いログ運用を実現できます。

また、アプリケーションサーバで使用する日立の JavaVM では、障害発生時の要因分析やシステムの状態確認に利用できるよう、ログの出力内容が拡張されています。このログを利用して適切なチューニングを実施することで、システムの可用性向上を図れます。

トレース情報による性能解析

リクエストの実行時に、Web サーバからデータベース接続までの各ポイントでトレース情報を出力します。この情報を収集して分析することで、処理のボトルネックを明確にしたり、障害発生の原因になった箇所を特定したりできます。

クライアント性能の測定と分析

Web アプリケーションの処理時間をクライアントの視点で測定、分析できます。サーバプレットフィルタによってクライアントへのレスポンスにスクリプトを埋め込むことで、Web アプリケーションとの通信に掛かる時間、ブラウザでの描画時間、ユーザの

4. J2EE アプリケーションを実行する

操作時間などを測定できます。これによって、システム性能に問題がないか、性能上の問題がサーバ側またはクライアント側のどちらにあるのか、などを分析できます。

統合ユーザ管理

アプリケーションサーバで構築したシステムにログインするユーザを統合管理できます。それぞれの J2EE アプリケーションが管理しているユーザの情報が関連づけられて管理されるため、一度のログイン処理でさまざまな J2EE アプリケーションにログインできるようになります。なお、ユーザ情報を管理するリポジトリとしては、LDAP ディレクトリサービスやデータベースが利用できます。

メンテナンス時のアプリケーション入れ替えの効率化

アプリケーションをアーカイブ形式と展開ディレクトリ形式の 2 種類で管理できます。展開ディレクトリ形式のアプリケーションの場合、アプリケーションを入れ替えるときにアーカイブし直す必要がなく、変更したクラスファイルをリロードするだけで入れ替えができます。また、アーカイブ形式のアプリケーションの場合も、局所的な変更の場合にはアプリケーションの停止や再設定が不要な入れ替え（リデプロイ）ができます。これらによって、メンテナンス時の入れ替え効率を高め、システム運用コストを抑えられます。

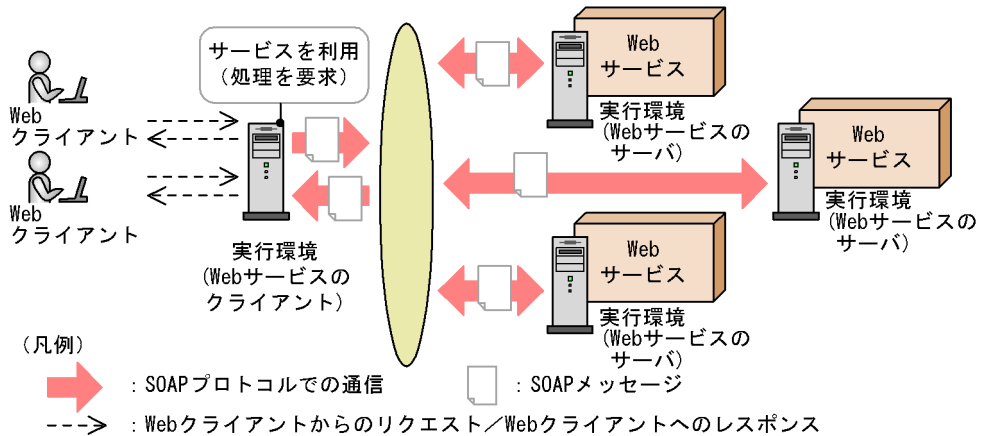
4.1.7 Web サービスへの対応

Web サービスとは、システム間でプログラムを使用して情報をやり取りし、ほかのシステムが持つアプリケーションを Web 上で使用できるようにした仕組みです。また、Web サービスの仕組みによってネットワーク経由で公開、実行できるアプリケーションのことも、Web サービスといえます。アプリケーションサーバでは、Web サービスの実行環境としての機能を提供するほか、Web サービスに対応したアプリケーションの開発を支援しています。

Web サービスの多くは、SOAP と呼ばれるプロトコルを使用して、ネットワーク上でのシステム間連携を実現します。SOAP では他システムのアプリケーションへのアクセスを実現するために、下位のトランスポート層に Web 標準の HTTP プロトコルを利用します。アプリケーションサーバでは、JAX-WS 2.1 仕様に従って、Web サービスクライアントと Web サービス間の SOAP メッセージのバインディングを実現します。

Web サービス利用環境の概要を次の図に示します。

図 4-6 Web サービス利用環境の概要



Web サービスの利用環境には、Web サービスのクライアントになる実行環境と、Web サービスのサーバになる実行環境が必要です。アプリケーションサーバでは、これら両方の実行環境を構築できます。

Web クライアントのユーザからのリクエストを受け付けた実行環境（Web サービスのクライアント）は、Web サービスを提供する実行環境（Web サービスのサーバ）に SOAP メッセージを送ります。Web サービスのサーバでは、処理を実行して、結果をレスポンスとして Web サービスのクライアントに送信します。Web サービスのクライアントと Web サービスのサーバ間の通信には SOAP プロトコルを使用します。

機能の詳細については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照してください。

参考

アプリケーションサーバ 07-60 までで提供していた Web サービス（SOAP アプリケーション）の実行基盤は、JAX-WS 2.1 仕様には準拠していません。JAX-WS 2.1 仕様準拠しない形式の Web サービスを使用する場合は、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

4.1.8 信頼性の高い非同期通信の実現

アプリケーション間で、メッセージを使用した非同期通信を実現できます。

非同期通信では、送信側のアプリケーションがメッセージを送信する際、キューまたはトピックにメッセージを登録します。

キューを使用する場合、受信側のアプリケーションは、メッセージを受信して、キューからメッセージを取り出します。これによって、送信側、受信側のアプリケーションが

4. J2EE アプリケーションを実行する

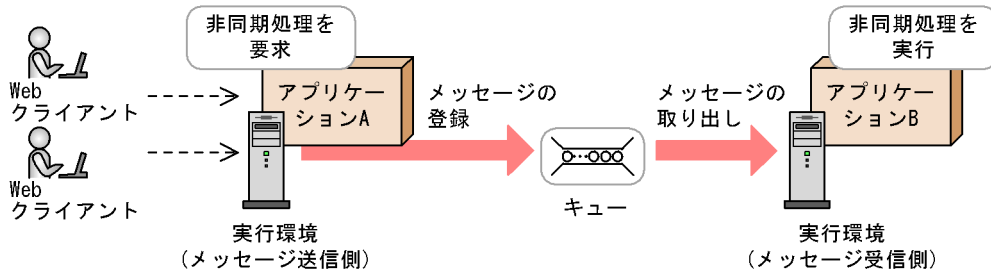
同時に動作していない場合も、メッセージのやり取りが可能になります。

トピックを使用する場合、あらかじめ購読を申し込んでいた受信側のアプリケーションに対して、メッセージが送信されます。購読は、複数のアプリケーションから申し込みます。

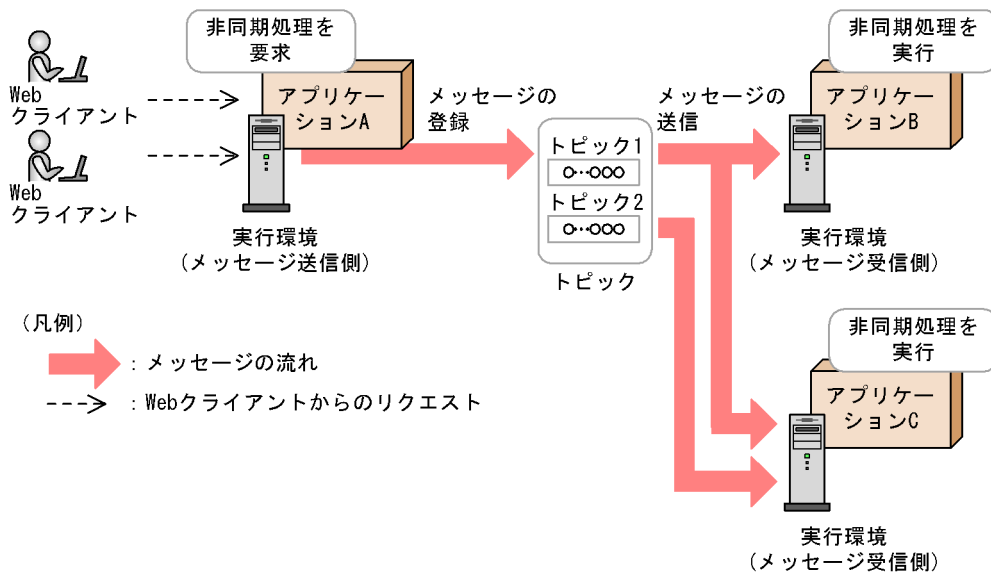
アプリケーションサーバで実現できる非同期通信の概要を次の図に示します。

図 4-7 アプリケーションサーバで実現できる非同期通信の概要

●キューを使用する場合



●トピックを使用する場合



なお、トピックは、Cosminexus JMS プロバイダの機能を使用した場合だけ使用できません。

また、Cosminexus RM を使用する場合、メッセージを管理するキューをデータベースで管理するため、高信頼なメッセージ管理ができます。また、確実に1回の配達保証、または順序保証の QoS (通信品質) などによってメッセージ通信の信頼性も確保できます。

さらに、JMS インタフェースを採用し、高信頼な標準プロトコルである WS-Reliability へも対応しているため、ほかのシステムとの連携やほかのシステムからのアプリケーションの移植などにも柔軟に対応できます。Cosminexus RM の機能の詳細については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

参考

この項の説明は、アプリケーションサーバが提供する機能を使用した場合の非同期通信の説明です。

このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用して、外部のリソースからのメッセージ送信を受け付けてアプリケーションサーバ上の処理を非同期で実行することもできます。詳細は、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「3.16.3 メッセージインフロー」を参照してください。

4.2 J2EE アプリケーションの実行環境の種類

アプリケーションサーバでは、Java EE に準拠した J2EE アプリケーションの実行環境を構築できます。また、クライアントとサーバ間の通信に SOAP を利用する SOAP アプリケーションの実行環境や、サーバ間の通信にメッセージを利用して非同期通信を実現するアプリケーションの実行環境も構築できます。

実行するアプリケーションの種類によって、実行環境の構成および使用する機能が異なります。例えば、J2EE アプリケーションでサーブレットや JSP を使用するか、または Enterprise Bean を使用するかによって、推奨されるシステム構成が異なります。さらに、システム構成によって、構築方法も異なります。

以降では、実行環境を次の 3 種類に分けて説明します。

1. J2EE アプリケーションの実行環境（2. および 3. に該当しない実行環境）
「4.3 J2EE アプリケーションの実行環境を構築・運用する」で説明します。
2. Web サービスを利用するためのアプリケーション（SOAP アプリケーション）の実行環境
「4.4 Web サービスに対応したアプリケーションの実行環境を構築・運用する」で説明します。
3. サーバ間でメッセージを利用した非同期通信を実行するアプリケーションの実行環境
「4.5 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する」で説明します。

なお、2. および 3. の環境は、1. の環境を前提としています。このため、これらの実行環境で利用する機能の検討やシステム構築、運用をする場合は、1. の実行環境で実施する内容に加えて 2. と 3. の環境で必要な内容を実施するようにしてください。

4.3 J2EE アプリケーションの実行環境を構築・運用する

この節では、J2EE アプリケーションの実行環境の構築・運用について説明します。

4.3.1 システム構成の種類

ここでは、実行環境のシステム構成として、次の 2 種類のシステム構成について説明します。

スケールアウトシステム

フロントエンド/バックエンドシステム

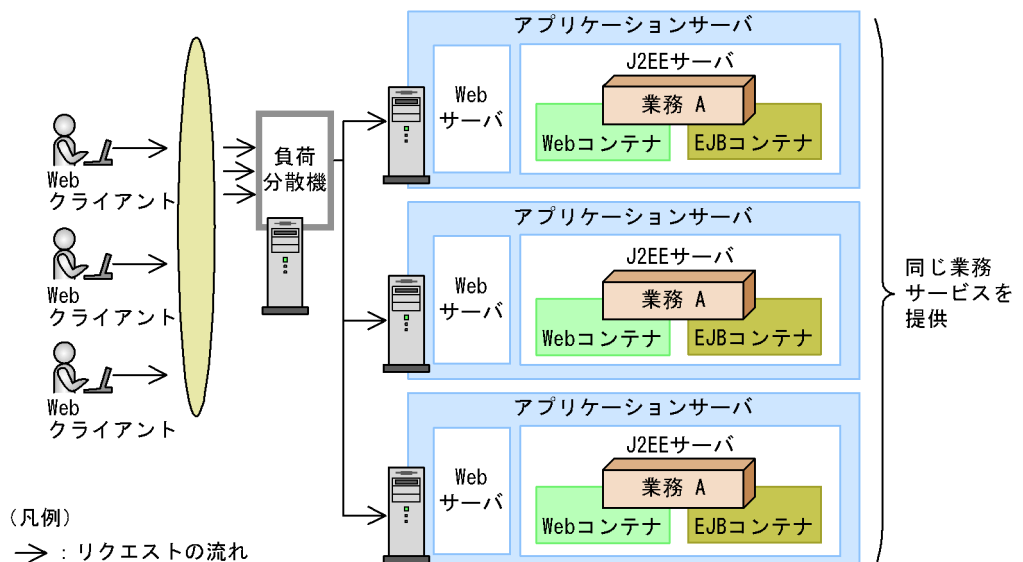
(1) スケールアウトシステム

スケールアウトとは、システム全体の処理性能を向上させることを目的として、サーバの台数を増やすことです。一方、サーバの台数を減らすことでシステム規模を縮小することをスケールインといいます。

スケールアウトシステムはシステム規模の拡大や縮小に柔軟に対応できる、スケーラビリティの高いシステムのことで、業務システムの運用規模や要件の変化に応じてスケールアウトまたはスケールインを実施できます。

スケールアウトシステムの構成例を次に示します。

図 4-8 スケールアウトシステムの構成例



この図の構成の場合、3 台のアプリケーションサーバは同じ業務を実行するサーバです。

4. J2EE アプリケーションを実行する

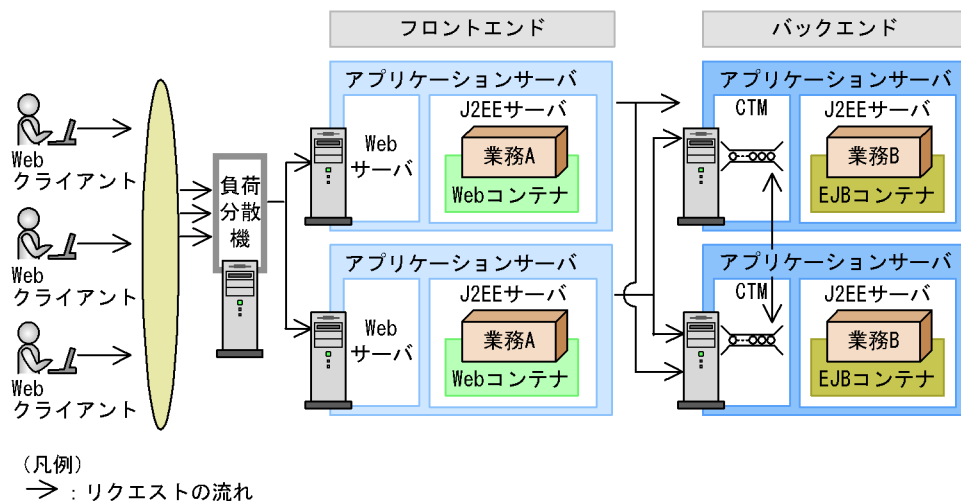
リクエスト数の増加などによって処理性能を上げる場合は、さらにアプリケーションサーバマシンの台数を増やすことで対応できます。

(2) フロントエンド/バックエンドシステム

フロントエンド/バックエンドシステムは、システムのスケーラビリティを確保しながら可用性を高めたシステムです。フロントエンドとバックエンドから構成されます。フロントエンドでは、Web ブラウザなどからのリクエストを受け付けて適切な業務処理プログラムに処理を振り分ける処理を実行します。バックエンドでは、フロントエンドから振り分けられた業務処理を実行し、必要に応じて EIS にアクセスします。

アプリケーションサーバの場合、フロントエンドでサーブレットや JSP を実行して、バックエンドで Enterprise Bean を実行する構成などが該当します。フロントエンド/バックエンドシステムの構成例を次に示します。

図 4-9 フロントエンド/バックエンドシステムの構成例



この図の構成の場合、フロントエンドの2台のアプリケーションサーバでは同じ処理を実行します。また、バックエンドの2台のアプリケーションサーバでも同じ処理を実行します。クライアントからのリクエストは、まず、負分散機によってフロントエンドのアプリケーションサーバに振り分けられます。また、Enterprise Bean のスケジューリング機能を持つ CTM を使用することによって、バックエンドでの負分散も実現できます。

フロントエンド、バックエンドはそれぞれスケールアウト構成になっているため、リクエスト数の増加などによって処理性能を上げる場合は、各層のアプリケーションサーバの台数を増やすことで対応できます。また、障害発生時の縮退運用や再起動時の影響の局所化などが図れるため、システム全体の可用性を向上させることができます。

 ポイント

バックエンドで CTM を利用する場合、バックエンドのマシンに次のどちらかの製品が必要です。

- Application Server Enterprise
- Service Platform

なお、アプリケーションの開発環境でのテスト時に CTM を利用する場合は、次のどちらかの製品を使用できます。

- Developer Professional
 - Service Architect
-

4.3.2 スケールアウト構成のシステムの構築・運用

アプリケーションサーバでは、システムを一括構築・運用するための機能として、次の機能を提供しています。

Smart Composer 機能

複数台のマシンで構成される J2EE アプリケーションの実行環境を一括して構築・運用できます。構築には、定義ファイルやコマンドを使用します。

セットアップウィザード

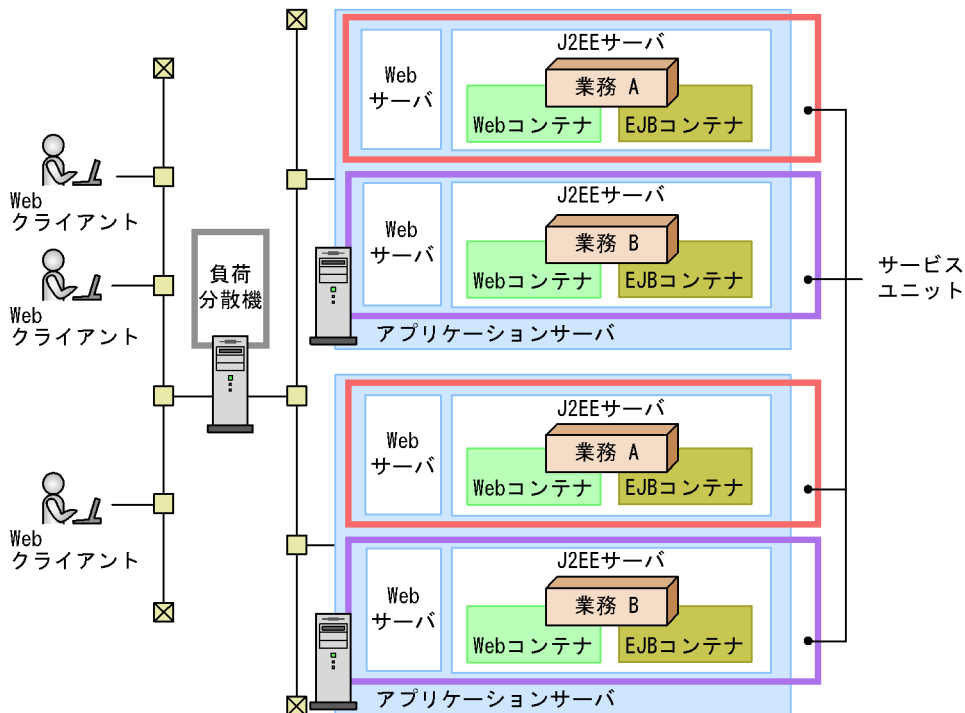
1 台のマシンで構成される J2EE アプリケーションの実行環境を構築・運用できます。構築には、対話形式の CUI プログラムを使用します。

アプリケーションサーバの推奨構成は、Web サーバと J2EE サーバを 1 : 1 の組み合わせで配置するスケールアウト構成です。なお、一つの業務処理を実行する Web サーバと J2EE サーバの組み合わせは、サービスユニットとして管理できます。サービスユニットは、一つの業務を実行するためのサーバの集合です。サービスユニット単位に、スケールアウトや、閉塞処理を実行できます。セッション情報を管理するためのサーバや、CTM を使用したリクエストのスケジューリングに使用するサーバの集合も、サービスユニットとして扱えます。

次に、スケールアウト構成の例を示します。この例では、一つのマシン内に二つのサービスユニットを配置しています。サービスユニットを二つ配置することで、例えば、業務 1 としてオンライン系の業務を実行し、業務 2 としてオフライン系の業務を実行するなどの運用も実現できます。

4. J2EE アプリケーションを実行する

図 4-10 Smart Composer 機能によって構築するスケールアウト構成の例



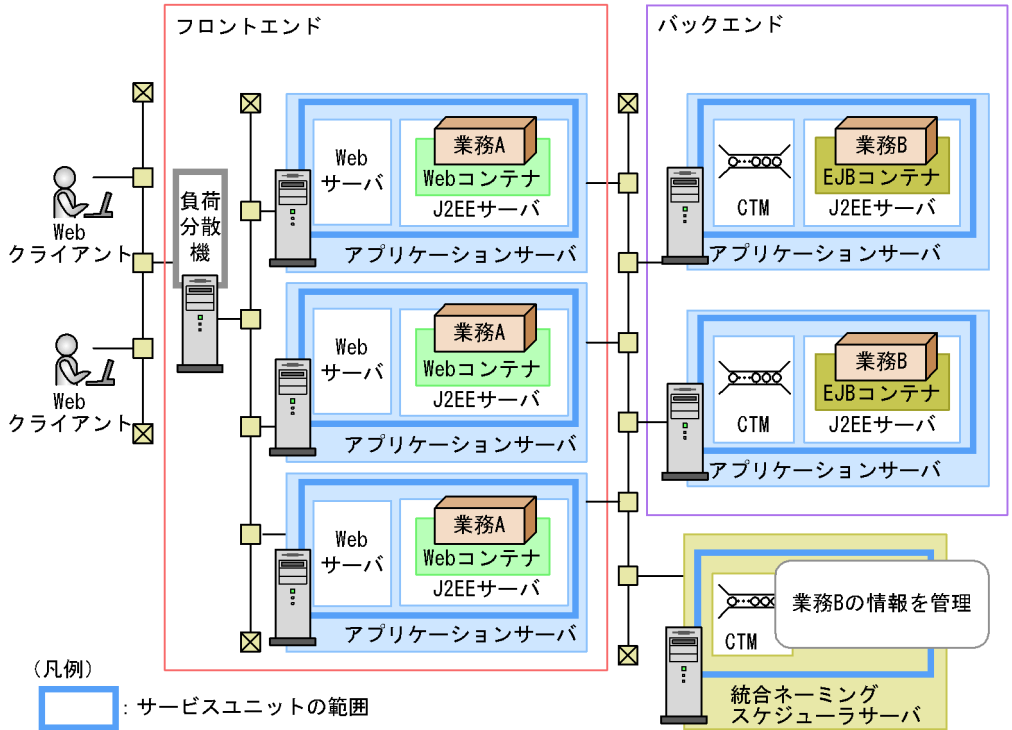
4.3.3 フロントエンド / バックエンド構成のシステムの構築・運用

フロントエンド / バックエンドシステムを構築する場合も、Smart Composer 機能またはセットアップウィザードを使用して構築できます。この場合は、フロントエンドのサービスユニットとバックエンドのサービスユニットを、それぞれスケールアウト構成で構築します。ただし、セットアップウィザードは、1 台のマシンで構成されるシステムを構築する場合だけに使用できます。

なお、Smart Composer 機能で CTM を使用するバックエンドシステムを構築する場合、CTM のスケジューリングの対象になる業務処理プログラム (Enterprise Bean) の情報を登録したサーバ (統合ネーミングスケジューラサーバ) は、J2EE アプリケーションとは別のサービスユニットとして扱います。

CTM を使用したフロントエンド / バックエンド構成として構築した例を次の図に示します。

図 4-11 Smart Composer 機能によって構築するフロントエンド/バックエンド構成の例 (CTM を使用する場合)



このほか、次のようなパターンの構成も、Smart Composer 機能を使用して構築できます。

負分散機の冗長化構成 (アクティブ - スタンバイ構成)

セッションの引き継ぎ (セッションフェイルオーバー機能) を実現する構成

Web サーバと J2EE サーバを分けた構成

インプロセス HTTP サーバ機能を使用する構成

アプリケーションサーバで実現できるシステム構成については、マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」の「3. システム構成の検討 (J2EE アプリケーション実行基盤)」を参照してください。また、Smart Composer 機能またはセットアップウィザードを使用した構築・運用方法については、マニュアル「Cosminexus アプリケーションサーバシステム構築・運用ガイド」を参照してください。

4.3.4 構築した環境のチューニング

Smart Composer 機能またはセットアップウィザードで構築したシステムは、次のどち

4. J2EE アプリケーションを実行する

らかの方法でチューニングできます。

- Smart Composer 機能
- 運用管理ポータル

運用管理ポータルは、Management Server を操作するための GUI プログラムです。Web ブラウザから Management Server が管理する各サーバの設定を変更できます。

また、Smart Composer 機能または運用管理ポータルでチューニングしたシステムは、セットアップウィザードのコマンドを使用して複製できます。

Smart Composer 機能を使用したチューニングについては、マニュアル「Cosminexus アプリケーションサーバシステム構築・運用ガイド」の「8.10 チューニング」を参照してください。運用管理ポータルについては、マニュアル「Cosminexus アプリケーションサーバ運用管理ポータル操作ガイド」を参照してください。

4.4 Web サービスに対応したアプリケーションの実行環境を構築・運用する

この節では、Web サービスに対応したアプリケーションの実行環境について説明します。

4.4.1 Web サービスの実行環境

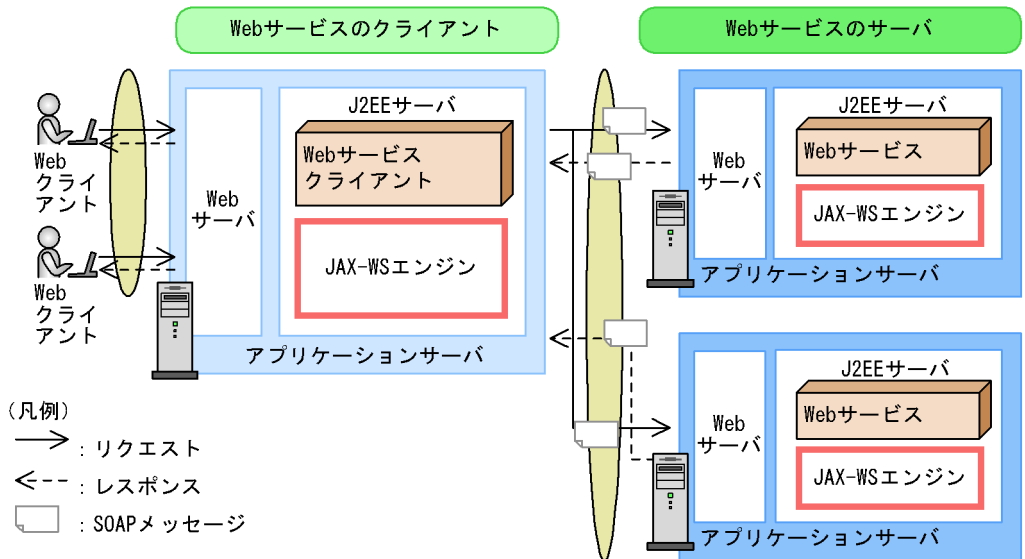
Web サービスは、Web の標準技術を利用してネットワーク上でサービスを公開、実行できるアプリケーションです。

Web サービスでは、クライアントとサーバ間のメッセージ交換用のプロトコルに SOAP を利用しています。クライアントからサーバにメッセージを送信して、サーバ側で提供されるサービスを利用します。SOAP アプリケーションのクライアントとサーバ間でやり取りするメッセージを SOAP メッセージといいます。

Web サービスの実行環境は、Web サービスを呼び出すクライアントと、Web サービスを提供するサーバで構成されます。クライアントとサーバには、SOAP メッセージのマーシャル処理とアンマーシャル処理を実行するための JAX-WS エンジンを配置します。

Web サービスの実行環境の例を次の図に示します。

図 4-12 Web サービスの実行環境の例



この図の場合は、Web サービスのクライアントをアプリケーションサーバとして構築しています。Web クライアントからのリクエストを受け付けたアプリケーションサーバが Web サービスのクライアントになり、サーバに当たるアプリケーションサーバ上の Web

4. J2EE アプリケーションを実行する

サービスを呼び出します。Web サービスのクライアントでは、JAX-WS エンジンによって、Java オブジェクトから SOAP メッセージ (XML データ) を生成します (マーシャル処理)。メッセージを受け取ったサーバでは、JAX-WS エンジンによって SOAP メッセージから Java オブジェクトを生成して (アンマーシャル処理)、Web サービスを実行します。

ポイント

Web サービスのクライアントとサーバに使用できる製品を次の表に示します。

表 4-1 Web サービスのクライアントとサーバに使用できる製品

構築する実行環境	使用できる製品
Web サービスのクライアント	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect uCosminexus Client
Web サービスのサーバ	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect

注 開発環境でのデバッグ環境として使用できます。

Web サービスで実現できることの詳細については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照してください。

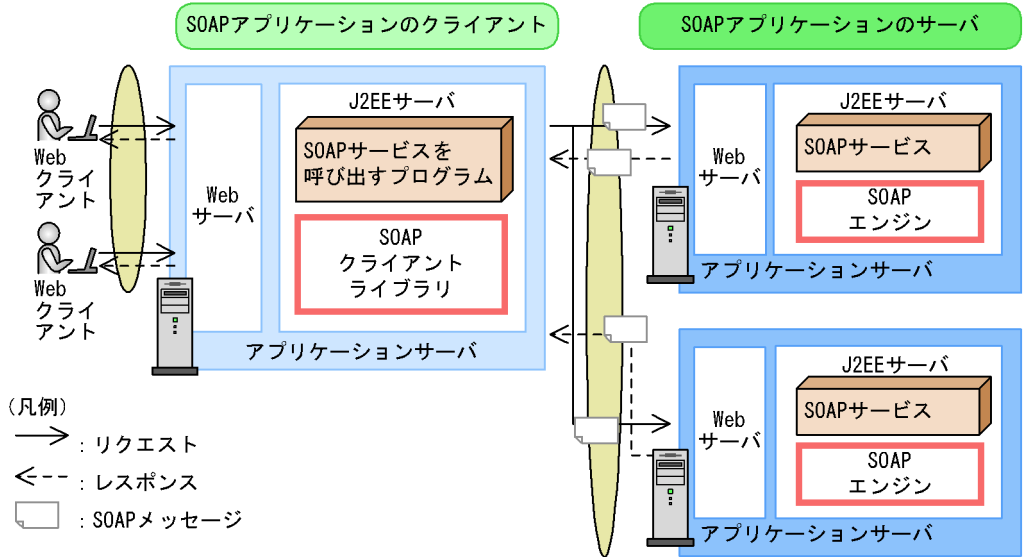
参考

07-60 以前からアプリケーションサーバが提供していた SOAP アプリケーション開発支援機能を使用した実行環境も構築できます。

従来の SOAP アプリケーション開発支援機能を使用する場合、クライアントには、SOAP アプリケーションを呼び出すための SOAP クライアントライブラリが必要です。サーバには、クライアントと SOAP サービスの送受信の仲介をするエンジンである、SOAP エンジンが必要です。SOAP アプリケーションのサーバは、アプリケーションサーバとして構築します。クライアントは、アプリケーションサーバ、またはコマンドラインで動作する Java アプリケーションの実行環境として構築します。

既存の SOAP アプリケーション開発支援機能を使用して構築した実行環境の例を次の図に示します。

図 4-13 SOAP アプリケーションの実行環境の例



この図の場合は、クライアントをアプリケーションサーバとして構築しています。Webクライアントからのリクエストを受け付けたアプリケーションサーバがSOAPアプリケーションのクライアントになり、サーバに当たるアプリケーションサーバ上のSOAPサービス呼び出します。SOAPアプリケーションのクライアントでは、SOAPクライアントライブラリを使用して、サーバにメッセージを送信します。メッセージを受け取ったサーバでは、SOAPエンジンを介してSOAPサービスを実行し、レスポンスをメッセージとしてクライアントに返します。

SOAPアプリケーション開発支援機能については、マニュアル「CosminexusアプリケーションサーバSOAPアプリケーション開発の手引」を参照してください。

4.4.2 Web サービスセキュリティの技術を利用した SOAP アプリケーションの実行環境

SOAPアプリケーション開発支援機能を使用している場合、データのやり取りにWebサービスセキュリティの技術を利用できます。

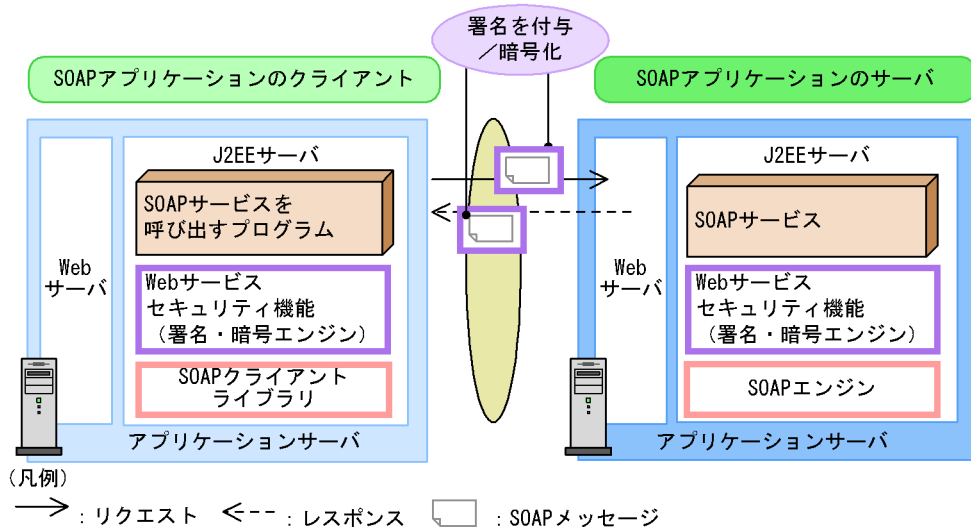
SOAPアプリケーションのクライアントとサーバ間の通信で送受信するSOAPメッセージは、XML形式のデータです。このデータの内容を改ざんされたり第三者による盗聴を防いだりするための技術として、Webサービスセキュリティという技術を使用できます。

Webサービスセキュリティの技術によって、SOAPメッセージの完全性と秘匿性が保証されます。また、SOAPメッセージの認証も実現できます。さらに、XML署名データを生成または検証したり、データを暗号化または復号化したりすることもできます。

Webサービスセキュリティの技術を利用したシステムの例を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-14 Web サービスセキュリティの技術を利用したシステムの例



ポイント

Web サービスセキュリティの技術を使用する場合に SOAP アプリケーションのクライアントとサーバに使用できる製品は、Web サービスセキュリティの技術を使用しない場合と同じです。「4.4.1 Web サービスの実行環境」を参照してください。

Web サービスセキュリティ技術の詳細については、マニュアル「Cosminexus アプリケーションサーバ Web サービスセキュリティ 使用の手引」を参照してください。XML 署名および XML 暗号の詳細については、マニュアル「Cosminexus XML Security - Core ユーザーズガイド」を参照してください。

4.5 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する

アプリケーション間での非同期通信には、メッセージを使用します。非同期通信で使用するメッセージとは、Java EE の仕様として規定された JMS インタフェースによって実装されたデータです。送信元のアプリケーションからのリクエストがメッセージの形式で送信されると、メッセージは一度キューまたはトピックに登録されます。

キューを使用する場合、受信側のアプリケーションは、メッセージを受信して、キューからメッセージを取り出します。これによって、送信側、受信側のアプリケーションが同時に動作していない場合も、メッセージのやり取りが可能になります。

トピックを使用する場合、あらかじめ購読を申し込んでいた受信側のアプリケーションに対して、メッセージが送信されます。購読は、複数のアプリケーションから申し込みます。また、永続化サブスクリバを使用することで、トピックへのメッセージ登録時に受信側のアプリケーションが動作していなくても、メッセージを配信できます。

アプリケーションサーバでは、メッセージを利用した非同期通信を実現するための機能として、次の 2 種類の機能を使用できます。

- Cosminexus JMS プロバイダ
- Cosminexus RM

それぞれの機能の特徴を次の表に示します。用途に応じた機能を使用してください。

表 4-2 メッセージを利用した非同期通信を実現するための機能の特徴

機能の種類	概要	使用できる送信先の種類	メッセージの保存先
Cosminexus JMS プロバイダ	JMS1.1 インタフェースに準拠したアプリケーションの実行環境を構築できます。	キューまたはトピック	ファイル
Cosminexus RM	JMS1.0 の PointToPoint インタフェースに従ったアプリケーションの実行環境を構築できます。さらに、DB Connector for Cosminexus RM との接続共有など、Cosminexus のアプリケーションサーバ独自の機能が利用できます。	キュー	データベース

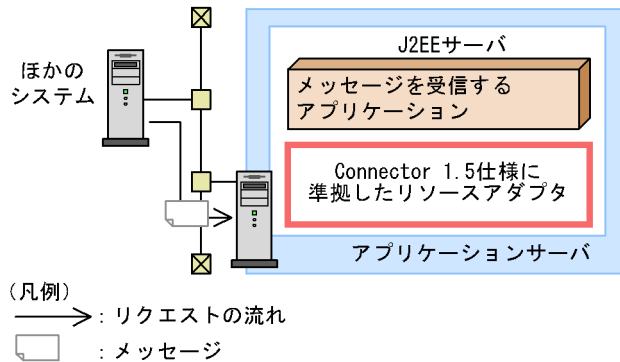
参考

この節の説明は、アプリケーションサーバの機能を使用した場合の非同期通信の説明です。このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用して、外部のリソースからのメッセージ送信を受け付け、アプリケーションサーバ上の処理を非同期で実行することもできます。

概要を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-15 Connector 1.5 仕様に準拠したリソースアダプタを使用する場合の非同期通信



Connector 1.5 仕様に準拠したリソースアダプタを使用したメッセージの非同期通信については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「3.16.3 メッセージインフロー」を参照してください。

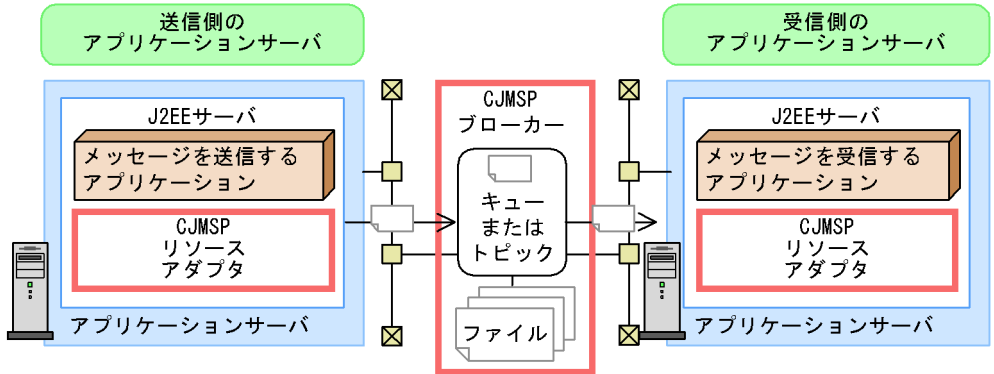
4.5.1 Cosminexus JMS プロバイダを利用した実行環境

Cosminexus JMS プロバイダを利用した実行環境では、JMS1.1 仕様に準拠したメッセージの送受信を実現できます。送信先にはキューまたはトピックを使用できます。永続化したメッセージをファイルに保存することで、受信側が受信できなかった場合に、正しく受信されるまでメッセージを保持できます。

Cosminexus JMS プロバイダを利用した実行環境では、アプリケーションサーバが提供する CJMSP リソースアダプタというリソースアダプタと、CJMSP ブローカーというプロセスを使用して、メッセージの送受信や送信先の管理などを実現します。

実行環境の例を次の図に示します。

図 4-16 メッセージを利用した非同期通信アプリケーションの実行環境の例
(Cosminexus JMS プロバイダを利用する場合)



(凡例)

→ : リクエストの流れ

□ : メッセージ

送信側のアプリケーションサーバ上のメッセージを送信するアプリケーションから、CJMSP リソースアダプタを経由してメッセージが送信され、キューまたはトピックに登録されます。キューに登録されたメッセージは、受信側のアプリケーションサーバ上の CJMSP リソースアダプタによって取り出され、メッセージを受信するアプリケーションに渡されます。トピックに登録されたメッセージは、あらかじめ購読を申請していた受信側のアプリケーションサーバ上のアプリケーションに、CJMSP リソースアダプタ経由で配信されます。

受信側のアプリケーションサーバでは、メッセージの到着を契機として、アプリケーションが実行されます。なお、同じマシン内のアプリケーション間で非同期通信を実現したい場合は、送信側のアプリケーションサーバと受信側のアプリケーションサーバを同じマシンにすることもできます。

4.5.2 Cosminexus RM を利用した実行環境

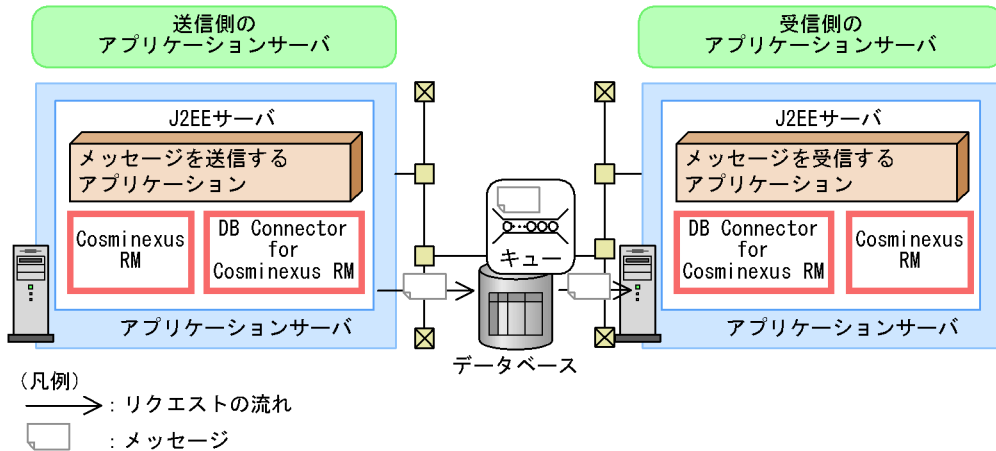
Cosminexus RM を利用した実行環境の場合、アプリケーションサーバはキューをデータベース上に構築します。データベースのデータとして登録することによって、メッセージを送信したときに受信側が受信できなかった場合に、正しく受信されるまでキューにメッセージを保持できます。これによって、確実なメッセージ配信を実現できます。

Cosminexus RM を利用した実行環境では、アプリケーションサーバが提供する Cosminexus RM および DB Connector for Cosminexus RM という二つのリソースアダプタを利用して、データベース上のキューの実現、およびアプリケーションからデータベース上のキューへのアクセスなどを実現します。

実行環境の例を次の図に示します。

4. J2EE アプリケーションを実行する

図 4-17 メッセージを利用した非同期通信アプリケーションの実行環境の例
(Cosminexus RM を利用する場合)

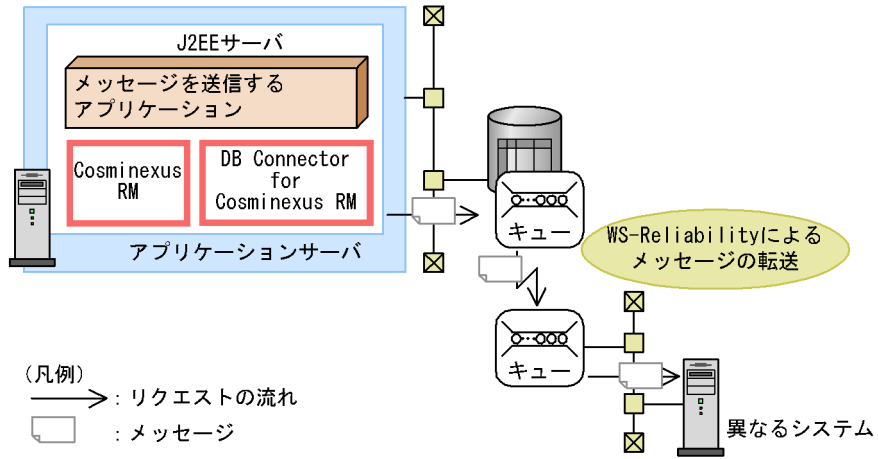


送信側のアプリケーションサーバ上で動作するメッセージを送信するアプリケーションから、Cosminexus RM および DB Connector for Cosminexus RM を経由してメッセージが送信され、キューに登録されます。メッセージがキューに登録されると、受信側のアプリケーションサーバ上の Cosminexus RM および DB Connector for Cosminexus RM によってメッセージが取り出され、メッセージを受信するアプリケーションに渡されます。受信側のアプリケーションサーバでは、メッセージの到着を契機として、アプリケーションが実行されます。なお、同じマシン内のアプリケーション間で非同期通信を実現したい場合は、送信側のアプリケーションサーバと受信側のアプリケーションサーバを同じマシンにすることもできます。

また、キューに登録されたメッセージを異なるシステムのキューに転送する場合は、WS-Reliability に基づいた転送が実行されます。WS-Reliability は、SOAP プロトコルをベースに、標準化団体 OASIS で標準化された、インターネット向けのオープンな非同期通信プロトコルです。異なるシステム間のキューでメッセージが転送される場合、メッセージは SOAP メッセージの形式で転送されます。

メッセージの転送の概要を、次の図に示します。

図 4-18 メッセージの転送の概要



ポイント

メッセージを利用した非同期通信アプリケーションの実行環境の送信側と受信側に使用できる製品を次の表に示します。

表 4-3 メッセージを利用した非同期通信アプリケーションの送信側と受信側に使用できる製品

構築する実行環境	使用できる製品
送信側	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect
受信側	Application Server Standard Application Server Enterprise Service Platform Developer Professional Service Architect (WS-Reliability に対応した異なるシステム)

注 開発環境でのデバッグ環境として使用できます。

メッセージを利用した非同期通信アプリケーションの実行環境については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

4.6 導入から運用までの流れ

J2EE アプリケーションを実行する環境の導入から構築・運用までの流れについて説明します。

3章で説明した流れに従って、次の工程を実施します。

システムの仕様（機能）検討

システム設計 / 運用設計

システムの構築

システムの日常運用 / 保守

また、必要に応じて運用の見直しを実施します。

それぞれの作業の概要を説明します。なお、構築・運用する環境によって固有の作業がある場合は、そのつど説明します。

(1) システムの仕様（機能）検討

システムの仕様（機能）検討では、実現したいシステムの要件に応じて J2EE アプリケーションの実行環境で実現できる機能を確認して、どの機能を使用するかを選択・検討します。

ここでは、主に、アプリケーションの実行基盤としての機能を選択してください。これらの機能は、システム構成やアプリケーションの実装に影響があります。ただし、アプリケーションの実行基盤を運用・保守するための機能にも、アプリケーションの実装が必要な機能があります。

選択する機能の例を示します。これらの機能については、「マニュアル」列に示したマニュアルで詳しく説明しています。

表 4-4 システムの仕様（機能）検討で選択する機能の例

分類	機能	参照先マニュアル	参照箇所
基本・開発機能	Web コンテナ	Cosminexus アプリケーションサーバ機能解説 基本・開発編 (Web コンテナ)	全体
	Web サーバ連携	Cosminexus アプリケーションサーバ機能解説 基本・開発編 (Web コンテナ)	3.
	インプロセス HTTP サーバ		4.
	サーブレットおよび JSP の実装	5.	

分類	機能	参照先マニュアル	参照箇所
	EJB コンテナ	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)	全体
	EJB クライアント	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)	3.
	Enterprise Bean 実装時の注意事項		4.
	ネーミング管理	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)	2.
	リソース接続とトランザクション管理		3.
	OpenTP1 からのアプリケーションサーバの呼び出し (TP1 インバウンド連携機能)		4.
	アプリケーションサーバでの JPA の利用		5.
	Cosminexus JPA プロバイダ		6.
	Cosminexus JMS プロバイダ		7.
	JavaMail の利用		8.
	セキュリティ管理		9.
	アプリケーションの属性管理		10.
	アノテーションの使用		11.
	J2EE アプリケーションの形式とデプロイ		12.
	コンテナ拡張ライブラリ		13.
拡張機能	バッチサーバによるアプリケーションの実行		Cosminexus アプリケーションサーバ 機能解説 拡張編
	CTM によるリクエストのスケジューリングと負荷分散	3.	
	バッチアプリケーションのスケジューリング	4.	
	J2EE サーバ間のセッション情報の引き継ぎ (セッションフェイルオーバー機能)	5.	
	明示管理ヒープ機能を使用したフルガーベージコレクションの抑止	8.	
	クライアント性能の測定と分析	9.	
	統合ユーザ管理	10.	
	アプリケーションのユーザログ出力	12.	
	スレッドの非同期並行処理	13.	

4. J2EE アプリケーションを実行する

分類	機能	参照先マニュアル	参照箇所
監視機能	稼働情報の監視（稼働情報収集機能）	Cosminexus アプリケーションサーバ機能解説 運用 / 監視 / 連携編	3.
	リソースの枯渇監視		4.
	監査ログ出力機能		6.
	データベース監査証跡連携機能		7.
	Management イベントの通知と Management アクションによる処理の自動実行		9.
保守機能	日立固有の JavaVM の機能	Cosminexus アプリケーションサーバ機能解説 保守 / 移行 / 互換編	8.

「マニュアル」列に示したマニュアルでは、システムの目的ごとに、どの機能を使用するとよいかを示しています。アプリケーションサーバで構築するシステムの目的として想定しているのは、次の項目です。

- 信頼性
- 性能
- 運用・保守性
- 拡張性
- そのほか

これらの項目と各機能の対応を確認して、必要な機能を選択してください。

また、JP1 やクラスタソフトウェアなどのほかの製品と連携する場合は、マニュアル「Cosminexus アプリケーションサーバ機能解説 運用 / 監視 / 連携編」を参照してください。

さらに、構築する実行環境に応じて、次のマニュアルも参照してください。

Web サービスの機能を使用する場合

マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照して、アプリケーションサーバで動作する Web サービス開発の概要と開発手順について確認し、実現する機能やシステム構成を検討してください。なお、従来の SOAP アプリケーション開発支援機能を使用したシステムを構築・運用する場合は、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

メッセージを利用した非同期通信アプリケーションを使用する場合

使用する機能に応じて、次のどちらかのマニュアルを参照して、メッセージを利用した非同期通信アプリケーションの実行環境で実現する機能やシステム構成を検討してください。

- Cosminexus JMS プロバイダを使用するとき

マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「7. Cosminexus JMS プロバイダ」を参照してください。

- Cosminexus RM を使用するとき

マニュアル「Cosminexus Reliable Messaging」を参照してください。

（2）システム設計 / 運用設計

システム設計は、次の考え方で進めます。

1. 使用する機能に応じてシステム構成を検討する。
2. システムの目的に応じたセキュリティの確保の方法を検討する。
3. システムの目的に応じたチューニング項目を検討する。

システム設計の考え方の詳細については、マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」を参照してください。

また、構築する実行環境に応じて、次のマニュアルも参照してください。

Web サービスの機能を使用する場合

マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照して、Web サービスをどのような形態で開発するかを検討し、それに従って、システム設計 / 運用設計を実施してください。クライアントとサーバのそれぞれについて、運用方法と構築方法を検討してください。なお、従来の SOAP アプリケーション開発支援機能を使用したシステムを構築・運用する場合は、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

メッセージを利用した非同期通信アプリケーションを使用する場合

マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「3.16.3 メッセージインフロー」またはマニュアル「Cosminexus Reliable Messaging」を参照して、実行したい業務処理に応じた送信側、受信側のアプリケーションの形態を検討します。それに従って、システム設計 / 運用設計を実施してください。クライアントとサーバのそれぞれについて、運用方法と構築方法を検討します。なお、システム構成の検討については、マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」の「3.8 サーバ間で非同期通信をする場合の構成を検討する」を参照してください。

（3）システムの構築

システムの構築は、Smart Composer 機能またはセットアップウィザードを使用して実施します。

システムを構築するときに参照するマニュアルを、次の表に示します。

4. J2EE アプリケーションを実行する

表 4-5 システムを構築するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus アプリケーションサーバシステム構築・運用ガイド	Smart Composer 機能またはセットアップウィザードによってシステムを構築する手順について知りたい場合に参照してください。インストール、システム構成に応じたシステム構築方法、J2EE アプリケーションの設定および動作確認といった、一とおりの作業手順について説明しています。 また、テスト環境でチューニングした定義を基に本番環境を構築する場合も、このマニュアルを参照してください。	
Hitachi Web Server	Hitachi Web Server のコマンドや定義ファイルを使用して Web サーバを構築したい場合に参照してください。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するポート番号について知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

また、構築する実行環境に応じて、次のマニュアルも参照してください。

メッセージを利用した非同期通信アプリケーションを使用する場合

マニュアル「Cosminexus アプリケーションサーバ機能解説 基本・開発編（コンテナ共通機能）」の「3.16.3 メッセージインフロー」またはマニュアル

「Cosminexus Reliable Messaging」を参照して、メッセージを利用した非同期通信を実現するために固有な環境構築を実施してください。

(4) システムの日常運用

システムの日常運用のうち、システムの起動や停止、および起動状態の確認といった、基本的な運用業務については、Smart Composer 機能を使用して実施します。また、稼働情報・トレースファイルを使用したシステム監視などのシステムの日常的な運用業務は、Smart Composer 機能以外の運用管理コマンドや、出力されたファイルなどを使用して実施します。

システムを日常運用するときに参照するマニュアルを次の表に示します。

表 4-6 システムを日常運用するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus アプリケーションサーバシステム構築・運用ガイド	Smart Composer 機能によってシステムを起動および停止する手順について知りたい場合に参照してください。また、システム内のサービスユニットの稼働状態について確認する場合も、このマニュアルを参照してください。	

マニュアル	目的	分類
Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	稼働情報の調査方法、J2EE アプリケーションの保守的な運用方法などについて知りたい場合に参照してください。また、ほかの製品と連携した運用について知りたい場合も、このマニュアルを参照してください。	
Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	性能解析トレースの調査方法について知りたい場合に参照してください。	
Hitachi Web Server	Hitachi Web Server のコマンドを使用して Web サーバを運用する場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

(5) システムの保守

J2EE アプリケーションの入れ替えやトラブルシュートなどの保守的な業務は、Smart Composer 機能以外の運用管理コマンドや、出力されたファイルなどを使用して実施します。また、システムの目的に応じたチューニングを実施する場合は、Smart Composer 機能または運用管理ポータルを使用します。

システムを保守するときに参照するマニュアルを次の表に示します。

表 4-7 システムを保守するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus アプリケーションサーバ システム構築・運用ガイド	Smart Composer 機能によってシステムを起動および停止する手順について知りたい場合に参照してください。また、システム内のサービスユニットの稼働状態について確認する場合も、このマニュアルを参照してください。	
Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	トラブルが発生した場合に出力されるログの詳細などについて知りたい場合は、このマニュアルを参照して対処してください。また、日立固有の JavaVM の機能について知りたい場合もこのマニュアルを参照してください。	
Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	J2EE サーバや Web サーバなどの設定を Web ブラウザの画面から変更してチューニングしたい場合に参照してください。	
Hitachi Web Server	Hitachi Web Server のログの詳細を確認する場合に参照してください。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するログについて知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じ

4. J2EE アプリケーションを実行する

で参照先のマニュアルも確認してください。

また、運用する実行環境に応じて、次のマニュアルも参照してください。

メッセージを利用した非同期通信アプリケーションを使用する場合

マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「3.16.3 メッセージインフロー」またはマニュアル

「Cosminexus Reliable Messaging」を参照して、メッセージを利用した非同期通信を実現するために固有な運用を実施してください。

5

バッチアプリケーションを実行する

この章では、バッチアプリケーションを実行するための環境の概要について説明します。

バッチアプリケーションの実行環境でバッチアプリケーションのジョブスケジューリング機能を使用する場合は、Application Server Enterprise または Service Platform を使用して構築・運用します。なお、開発環境用の製品である Developer Professional または Service Architect で構築したデバッグ環境でも、バッチアプリケーションのジョブスケジューリング機能を使用できます。

5.1 バッチアプリケーションの実行環境の特長

5.2 バッチアプリケーションの実行環境を構築・運用する

5.3 導入から運用までの流れ

5.1 バッチアプリケーションの実行環境の特長

この節では、バッチアプリケーションの実行環境の特長について説明します。

5.1.1 オープン環境でのバッチジョブの実行

バッチジョブをオープン環境である Java の実行環境で実行します。

バッチアプリケーションの実行環境は、Java プログラムとして開発されたバッチアプリケーションを実行する環境です。この環境によって、バッチジョブのオープン環境への移行を図れます。

現在、基幹業務の処理を実行するバッチジョブの多くは、メインフレーム上で実行されています。アプリケーションサーバでは、バッチジョブの実行環境を、オープン環境である Java の実行環境として構築します。これによって、Java の特長である柔軟性や運用容易性を兼ね備えたバッチジョブの実行環境を構築・運用できます。

バッチアプリケーションは、常駐型の JavaVM プロセスであるバッチサーバで実行します。これによって、バッチジョブを実行するたびに JavaVM を起動するコストを抑えられます。

また、次に示す関連プログラムと連携することで、メインフレームで実現していた内容に類似した処理の制御や自動実行が実現できます。

JP1/AJS2

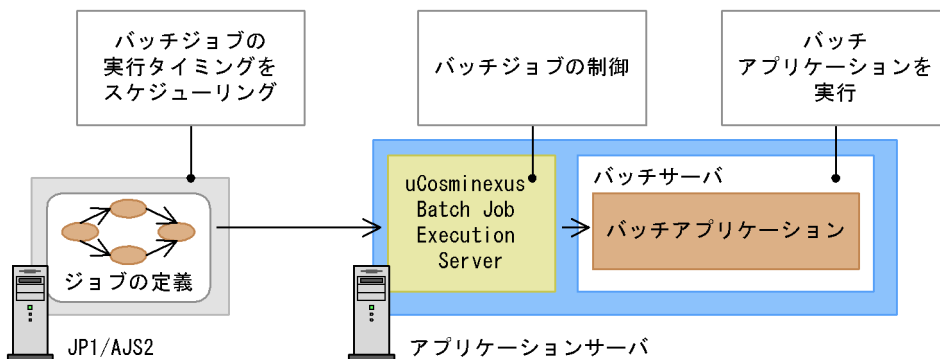
バッチアプリケーションを実行するタイミングをスケジューリングできます。これによって、バッチアプリケーションを使用する業務を自動化できます。

uCosminexus Batch Job Execution Server

メインフレームで実現していたジョブ制御言語によるジョブ管理のイメージで、バッチアプリケーションを実行できます。

バッチアプリケーションの実行環境でのバッチジョブ実行の概要を次の図に示します。

図 5-1 バッチアプリケーションの実行環境でのバッチジョブ実行の概要



この構成の場合は、バッチジョブを実行するタイミングを JP1/AJS2 で制御できます。また、uCosminexus Batch Job Execution Server を使用して定義した内容を基に、バッチジョブを実行できます。

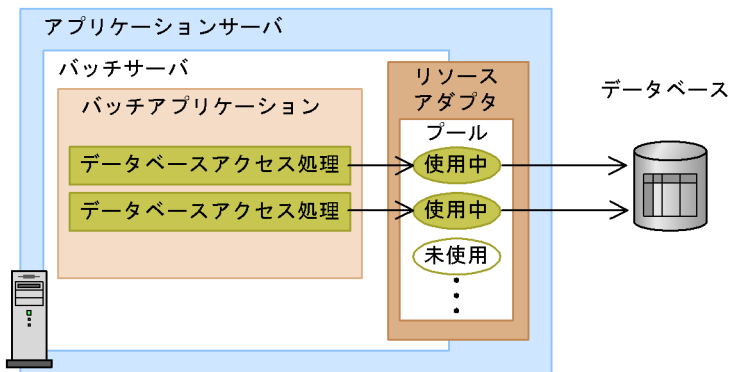
5.1.2 コネクションプール/ステートメントプールを使用したデータベースアクセスの高速化

データベースへの接続で使用するコネクションやステートメントの生成は、時間の掛かる処理です。コネクションプールおよびステートメントプールは、生成したコネクションやステートメントをプールしておくことで、処理性能の向上を図る機能です。

バッチアプリケーションの実行環境では、コネクションプールやステートメントプールなどの機能を使用できます。これによって、バッチアプリケーションからデータベースに高速にアクセスできます。なお、コネクションプールおよびステートメントプールの機能は、リソースアダプタを使用して実現します。

コネクションプールまたはステートメントプールを使用したデータベースアクセスの概要を次の図に示します。

図 5-2 コネクションプールまたはステートメントプールを使用したデータベースアクセスの概要



(凡例)

● および ○ : プールしていたコネクションまたはステートメント

バッチアプリケーションからデータベースにアクセスするときには、プールしていたコネクションまたはステートメントを使用します。

5.1.3 リソース排他状態でのフルガーベージコレクション実行を抑止

JavaVM では、空きメモリが一定のサイズ以下になると、フルガーベージコレクションが実行されます。フルガーベージコレクションが実行されると、その JavaVM 上での処理はすべて中断されます。バッチサーバ上のアプリケーションがリソースを排他状態で

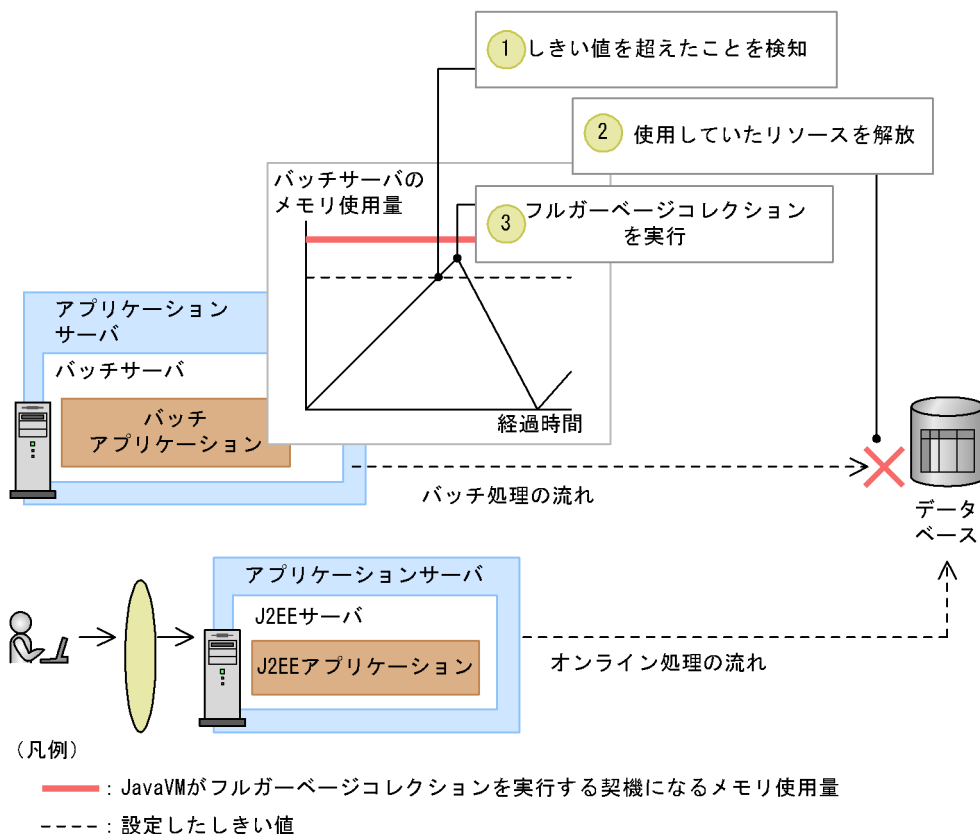
5. バッチアプリケーションを実行する

使用している時にフルガーページコレクションが実行されると、排他したままで処理が中断されます。この場合、同じリソースを使用したいオンライン処理の実行も中断されてしまいます。

バッチアプリケーションの実行環境では、バッチサーバ上でフルガーページコレクションを実行するタイミングを制御できます。この制御では、リソースが排他されていないタイミングを見計らってフルガーページコレクションを実行して、不要なオブジェクトを解放します。これによって、意図しないタイミングでフルガーページコレクションが実行されることを抑止できます。

バッチサーバでのフルガーページコレクション制御の概要を次の図に示します。

図 5-3 バッチサーバでのフルガーページコレクション制御の概要



1. でフルガーページコレクション制御によって、メモリ使用量がしきい値を超えたことが検知されます。バッチサーバでは、使用していたリソースを解放したことを 2. で確認してから、3. のフルガーページコレクションを実行します。

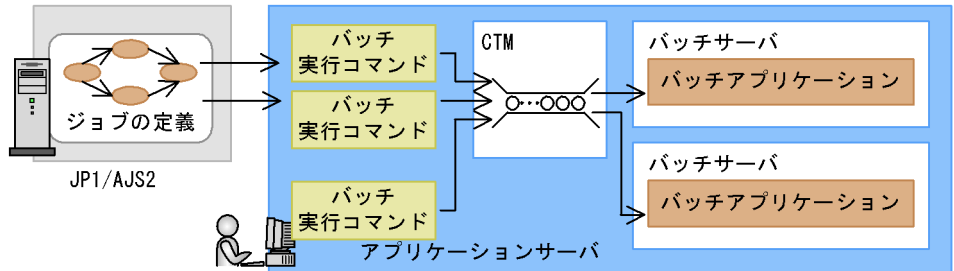
5.1.4 バッチアプリケーションのジョブスケジューリング

バッチアプリケーションは、バッチサーバごとに一つ実行できます。バッチサーバの数

を超えてバッチアプリケーションを実行しようとする時、エラーが返却されてしまいます。バッチアプリケーションのジョブスケジューリング機能を使用すると、バッチサーバの数を越えたジョブの実行要求があった場合に、実行要求を待たせることができます。これによって、エラーを返却することを防ぎます。

バッチアプリケーションのジョブスケジューリングには、CTM の機能を使用します。CTM を使用したジョブスケジューリングを次の図に示します。

図 5-4 CTM を使用したジョブスケジューリング



JP1/AJS2 経由で実行したバッチ実行コマンドや、直接実行したバッチ実行コマンドによって、バッチアプリケーションの実行要求が送信されます。CTM では、キューの概念によって、実行要求を管理します。実行可能なバッチサーバがある場合は処理を振り分け、ない場合はスケジュールキューに実行要求を滞留します。

5.1.5 そのほかの特長

5.1.4 までに示した以外に、バッチアプリケーションの実行環境には次の特長があります。これらは、J2EE アプリケーションの実行環境と同じです。

可用性と耐障害性の向上

「4.1.3 可用性と耐障害性の向上」を参照してください。ただし、セッション情報の引き継ぎは該当しません。

システム導入および拡張の容易化

「4.1.4 システム導入および拡張の容易化」を参照してください。

システム監査によるシステムのセキュリティ確保

「4.1.5 システム監査によるシステムのセキュリティ確保」を参照してください。

業務効率を向上させる運用管理の実現

「4.1.6 業務効率を向上させる運用管理の実現」を参照してください。ただし、統合ユーザ管理、メンテナンス時のアプリケーション入れ替えの効率化は該当しません。

5.2 バッチアプリケーションの実行環境を構築・運用する

この節では、バッチアプリケーションの実行環境の構築・運用について説明します。

5.2.1 システム構成の種類

ここでは、バッチアプリケーションの実行環境のシステム構成について説明します。

バッチアプリケーションの実行環境は、バッチサーバを配置したアプリケーションサーバを中心に構築します。また、業務の用途に合わせて、バッチジョブを自動実行するための関連プログラムを配置します。

ここでは、バッチアプリケーションの実行環境として、次の2種類のシステム構成について説明します。

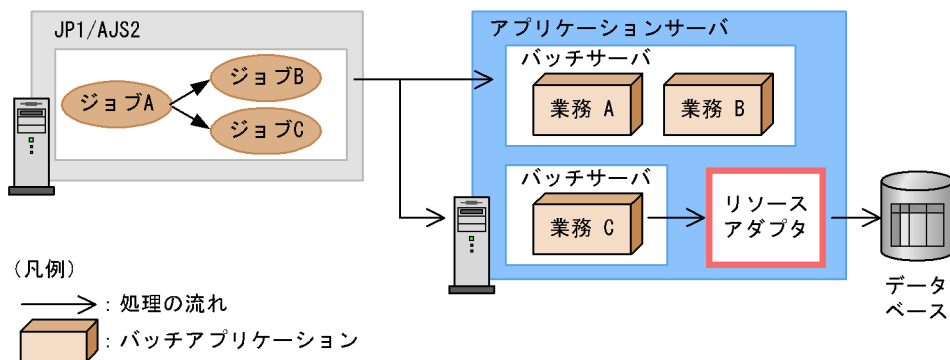
バッチ処理だけを実行するシステム

バッチ処理とオンライン処理を同時に実行するシステム

(1) バッチ処理だけを実行するシステム

バッチ処理だけを実行するシステムの構成例を次の図に示します。

図 5-5 バッチ処理だけを実行するシステムの構成例



この構成例では、1台のアプリケーションサーバ内に二つのバッチサーバを構築しています。それぞれのバッチサーバに対して、それぞれ異なるバッチアプリケーションを配置します。なお、一つのバッチサーバの中で複数のバッチアプリケーションを同時に実行することはできません。この図の場合、業務 A と業務 B は同時に実行されることはありません。業務 C は、リソースに接続するバッチアプリケーションです。

また、バッチサーバから、ほかの J2EE サーバ上で動作する J2EE アプリケーション (Enterprise Bean) を呼び出すこともできます。その場合、Enterprise Bean を実行するアプリケーションサーバは、フロントエンド/バックエンドシステムのバックエンド

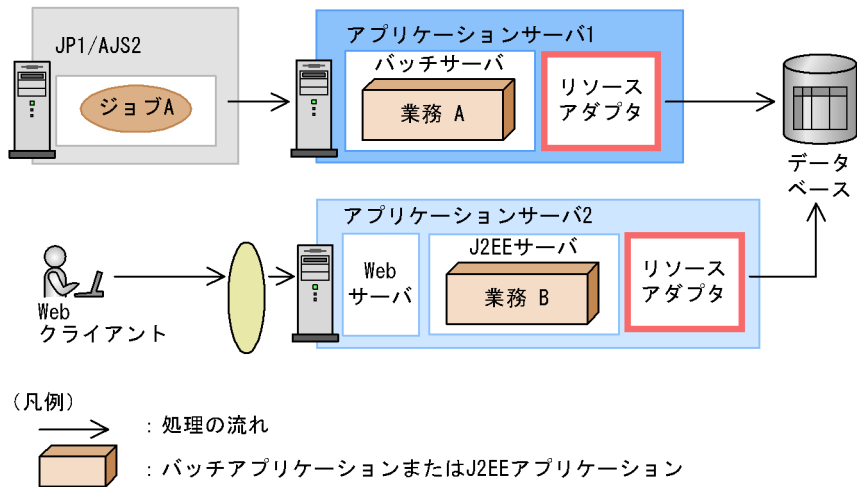
に該当します。バックエンドシステムについては、「4.3.1(2) フロントエンド/バックエンドシステム」を参照してください。

JP1 ジョブ運用管理サーバは、バッチジョブを自動実行する場合に必要な関連プログラムをインストールしたマシンです。必要に応じて配置します。

(2) バッチ処理とオンライン処理を同時に実行するシステム

一つのシステム内でバッチ処理とオンライン処理を同時に実行するシステムの構成例を次の図に示します。

図 5-6 バッチ処理とオンライン処理を同時に実行するシステムの構成例



この構成では、バッチ処理を実行するアプリケーションサーバ1と、オンライン処理を実行するアプリケーションサーバ2を構築します。また、バッチサーバ上のバッチアプリケーションと、J2EEサーバ上のJ2EEアプリケーションで同じリソースにアクセスします。

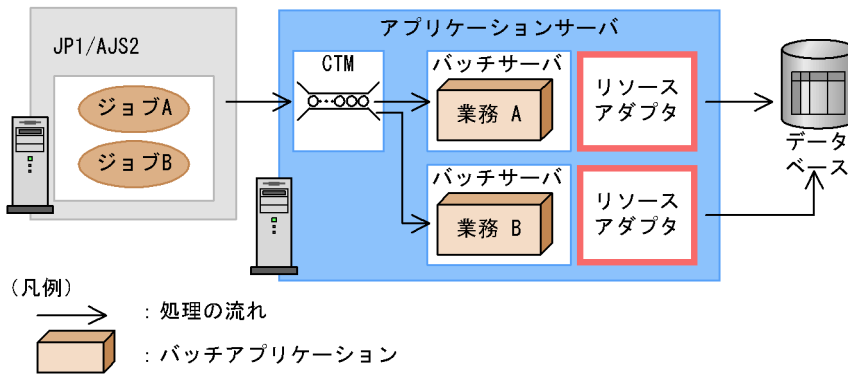
JP1 ジョブ運用管理サーバは、バッチジョブを自動実行する場合に必要な関連プログラムをインストールしたマシンです。必要に応じて配置します。

(3) バッチアプリケーションのジョブをスケジューリングするシステム

バッチアプリケーションのジョブをスケジューリングする場合のシステムの構成例を次の図に示します。

5. バッチアプリケーションを実行する

図 5-7 バッチアプリケーションのジョブをスケジューリングするシステムの構成例



この構成では、アプリケーションサーバにバッチサーバと CTM を構築します。どのバッチサーバでどのバッチアプリケーションを実行するかは、CTM が振り分けます。バッチアプリケーションの実行を要求するとき、実行するバッチサーバを意識する必要はありません。

5.2.2 バッチアプリケーションの実行環境の構築・運用

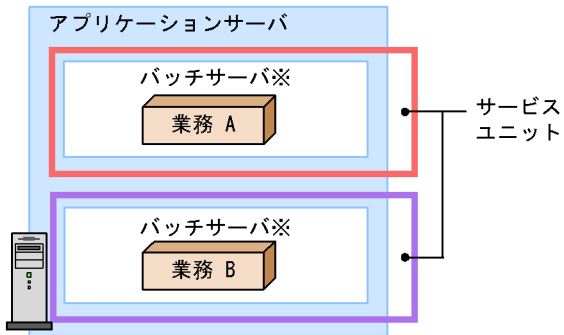
バッチアプリケーションの実行環境の構築・運用には、Smart Composer 機能を使用できます。Smart Composer 機能を使用すると、バッチアプリケーションの実行環境を、少ない手順で一括して構築・運用できます。

Smart Composer 機能では、一つの業務処理を実行するサーバの集合を、サービスユニットとして扱います。また、バッチサーバを J2EE サーバとして扱います。

Smart Composer 機能でバッチアプリケーションの実行環境を構築する場合は、サービスユニット内に J2EE サーバだけを構築するシステム構成パターンを選択して構築します。また、一部のパラメタに、バッチサーバ特有の値を設定します。

Smart Composer 機能によって構築するバッチアプリケーションの実行環境の例を、次の図に示します。

図 5-8 Smart Composer 機能によって構築するバッチアプリケーションの実行環境の例



注※ Smart Composer機能ではJ2EEサーバとして扱います。

バッチアプリケーションのジョブをスケジューリングする実行環境の構築・運用にも、Smart Composer 機能を使用します。この場合も、バッチサーバを J2EE サーバとして扱います。Smart Composer 機能の構成定義パターンのうち、CTM を使用するパターンのシステムを構築します。

なお、バッチアプリケーションのジョブをスケジューリングするシステムの場合、次のどちらかの製品が必要です。

- Application Server Enterprise
- Service Platform

5.3 導入から運用までの流れ

バッチアプリケーションを実行する環境の導入から構築・運用までの流れについて説明します。

3章で説明した流れに従って、次の工程を実施します。

システムの仕様（機能）検討

システム設計 / 運用設計

システムの構築

システムの日常運用 / 保守

また、必要に応じて運用の見直しを実施します。

それぞれの作業の概要を説明します。なお、構築・運用する環境によって固有の作業がある場合は、そのつど説明します。

(1) システムの仕様（機能）検討

システムの仕様（機能）検討では、実現したいシステムの要件に応じて、バッチアプリケーションの実行環境で実現できる機能を確認して、どの機能を使用するかを選択・検討します。

バッチアプリケーションの実行環境で使用できる機能の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 拡張編」の「2.2.3 バッチアプリケーションの実行環境の構築と運用」を参照してください。

(2) システム設計 / 運用設計

システム設計は、次の考え方で進めます。

1. 使用する機能に応じてシステム構成を検討する。
2. システムの目的に応じたチューニング項目を検討する。

システム設計の考え方の詳細については、マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」を参照してください。

(3) システムの構築

システムの構築は、Smart Composer 機能を使用して実施します。

システムを構築するときに参照するマニュアルを、次の表に示します。

表 5-1 システムを構築するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus アプリケーションサーバシステム構築・運用ガイド	Smart Composer 機能によってシステムを構築する手順について知りたい場合に参照してください。インストール、システム構成に応じたシステム構築方法および動作確認といった、一とりの作業手順について説明しています。 また、テスト環境でチューニングした定義を基に本番環境を構築する場合も、このマニュアルを参照してください。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するポート番号について知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

(4) システムの日常運用

システムの日常運用のうち、システムの起動や停止、および起動状態の確認といった、基本的な運用業務については、Smart Composer 機能を使用して実施します。また、稼働情報・トレースファイルを使用したシステム監視などのシステムの日常的な運用業務は、Smart Composer 機能以外の運用管理コマンドや、出力されたファイルなどを使用して実施します。

システムを日常運用するときに参照するマニュアルを、次の表に示します。

表 5-2 システムを日常運用するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus アプリケーションサーバシステム構築・運用ガイド	Smart Composer 機能によってシステムを起動および停止する手順について知りたい場合に参照してください。また、システム内のサービスユニットの稼働状態について確認する場合も、このマニュアルを参照してください。	
Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	稼働情報の調査方法、J2EE アプリケーションの保守的な運用方法などについて知りたい場合に参照してください。また、ほかの製品と連携した運用について知りたい場合も、このマニュアルを参照してください。	

5. バッチアプリケーションを実行する

マニュアル	目的	分類
Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	性能解析トレースの調査方法について知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

(5) システムの保守

トラブルシュートなどの保守的な業務は、Smart Composer 機能以外の運用管理コマンドや、出力されたファイルなどを使用して実施します。また、システムの目的に応じたチューニングを実施する場合は、Smart Composer 機能または運用管理ポータルを使用します。

システムを保守するときに参照するマニュアルを次の表に示します。

表 5-3 システムを保守するときに参照するマニュアル

マニュアル	目的	分類
Cosminexus アプリケーションサーバ システム構築・運用ガイド	Smart Composer 機能によってシステムを起動および停止する手順について知りたい場合に参照してください。また、システム内のサービスユニットの稼働状態について確認する場合も、このマニュアルを参照してください。	
Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	トラブルが発生した場合に出力されるログの詳細などについて知りたい場合は、このマニュアルを参照して対処してください。 また、日立固有の JavaVM の機能について知りたい場合もこのマニュアルを参照してください。	
Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	バッチサーバなどの設定を Web ブラウザの画面から変更してチューニングしたい場合に参照してください。	
TPBroker ユーザーズガイド	Cosminexus TPBroker が使用するログについて知りたい場合に参照してください。	

(凡例) : 必ず参照するマニュアル : 必要に応じて参照するマニュアル

なお、これらのマニュアルの中でほかのマニュアルを参照している場合は、必要に応じて参照先のマニュアルも確認してください。

6

アプリケーションを開発する

アプリケーションサーバでは、実行環境で動作するアプリケーションの開発を支援するための機能を提供しています。アプリケーションの開発環境は、Developer Standard、または Developer Professional を使用して構築します。なお、Service Architect を使用した環境でも、同様の機能を使用できます。この章では、開発環境が提供する開発支援機能を使用したアプリケーションの開発について説明します。

6.1 アプリケーション開発の特長

6.2 開発するアプリケーションの種類

6.3 J2EE アプリケーションを開発する

6.4 Web サービスに対応したアプリケーションを開発する

6.5 メッセージを使用した非同期通信のアプリケーションを開発する

6.6 バッチアプリケーションを開発する

6.7 アプリケーション開発の流れ

6.1 アプリケーション開発の特長

アプリケーションサーバの開発環境は、Developer をインストールして構築します。Developer では、実行環境上で動作するアプリケーションを、わかりやすい操作で効率良く開発できるようにするための開発環境を構築できます。

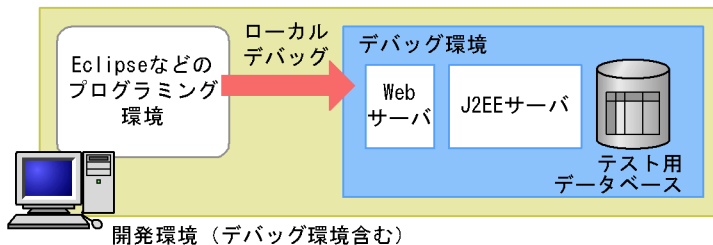
IDE として Eclipse を使用してアプリケーションを開発する場合、Developer で提供するプラグインを組み込んで使用することで、アプリケーション開発時の一連の作業をすべて Eclipse、または Eclipse のプラグインで実現できるようになります。

また、Developer で構築した開発環境では、開発したアプリケーションのデバッグ環境として、実行環境と同じ機能を持つ環境を構築できます。このため、アプリケーションの設計、プログラミングからテスト、デバッグまでの一連の流れを、すべて Developer で構築した開発環境上で実施できます。

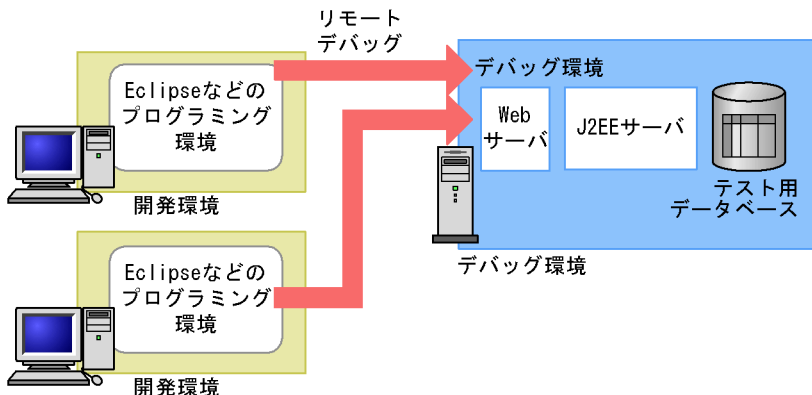
開発環境は、アプリケーションの開発方法やデバッグの実行方法に応じて、次の図に示す構成で構築できます。

図 6-1 開発環境の構成

- 開発環境とデバッグ環境を一つのマシンに構築する構成



- 開発環境とデバッグ環境を別のマシンに構築する構成



開発環境とデバッグ環境を一つのマシンに構築する構成

アプリケーションを一人の開発者が 1 台のマシンで開発する場合などに適しています。ローカルデバッグを実行できます。

開発環境とデバッグ環境を別のマシンに構築する構成

アプリケーションを複数の開発者のチームで開発する場合などに適しています。リモートデバッグを実行できます。なお、実行環境が Windows 以外の OS でテストを実行するときには、この構成にします。

それぞれの場合にインストールする製品を次の表に示します。

表 6-1 開発環境の構成ごとにインストールする製品

構成	開発環境	デバッグ環境
開発環境とデバッグ環境を一つのマシンに構築する構成	<ul style="list-style-type: none"> • Developer Standard • Developer Professional • Service Architect 	
開発環境とデバッグ環境を別のマシンに構築する構成	<ul style="list-style-type: none"> • Developer Professional • Service Architect 	<ul style="list-style-type: none"> • Application Server Standard • Application Server Enterprise • Service Platform

ポイント

Developer Standard は、一つの J2EE サーバ内で動作するアプリケーションを開発するための製品です。提供されるデバッグ環境も、1 台のアプリケーションサーバ内での動作を前提としています。このため、デバッグ環境をほかのマシンに構築する構成には対応していません。

このほか、Developer Standard と Developer Professional では、使用できる MyEclipse の機能も一部異なります。

Developer Standard と Developer Professional の機能差については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「付録 D.2 Developer Professional との機能差」を参照してください。

6.2 開発するアプリケーションの種類

アプリケーションサーバでは、J2EE アプリケーション、Web サービスに対応したアプリケーション、およびメッセージを使用した非同期通信用のアプリケーションを開発できます。それぞれのアプリケーション開発の概要を説明します。

J2EE アプリケーションの開発

JSP、サーブレット、Enterprise Bean を作成し、J2EE アプリケーションを開発できます。アプリケーションサーバでは、J2EE アプリケーションの開発に MyEclipse または WTP を利用できます。MyEclipse または WTP を使用すると、JSP、サーブレット、Enterprise Bean の作成からテスト・デバッグまでシームレスなアプリケーション開発が実現します。

J2EE アプリケーションの開発については、「6.3 J2EE アプリケーションを開発する」で説明します。

Web サービスに対応したアプリケーションの開発

Web サービスに対応したアプリケーション（Web サービス）を開発できます。アプリケーションサーバでは、Web サービスの開発および実行に必要なファイルやソースコード（スタブやスケルトンなど）を生成できます。

Web サービスの開発については、「6.4 Web サービスに対応したアプリケーションを開発する」で説明します。

メッセージを使用した非同期通信用のアプリケーションの開発

メッセージを使用した非同期通信用のアプリケーションを開発できます。アプリケーションサーバが提供する JMS インタフェースを利用してアプリケーションを開発できます。

非同期通信用のアプリケーションの開発については、「6.5 メッセージを使用した非同期通信用のアプリケーションを開発する」で説明します。

バッチアプリケーションの開発

バッチサーバ上で動作するバッチアプリケーションを開発できます。バッチアプリケーションの開発には、MyEclipse を利用できます。バッチアプリケーションの開発については、「6.6 バッチアプリケーションを開発する」を参照してください。

6.3 J2EE アプリケーションを開発する

この節では、アプリケーションサーバの機能を使用した J2EE アプリケーションの開発の概要について説明します。

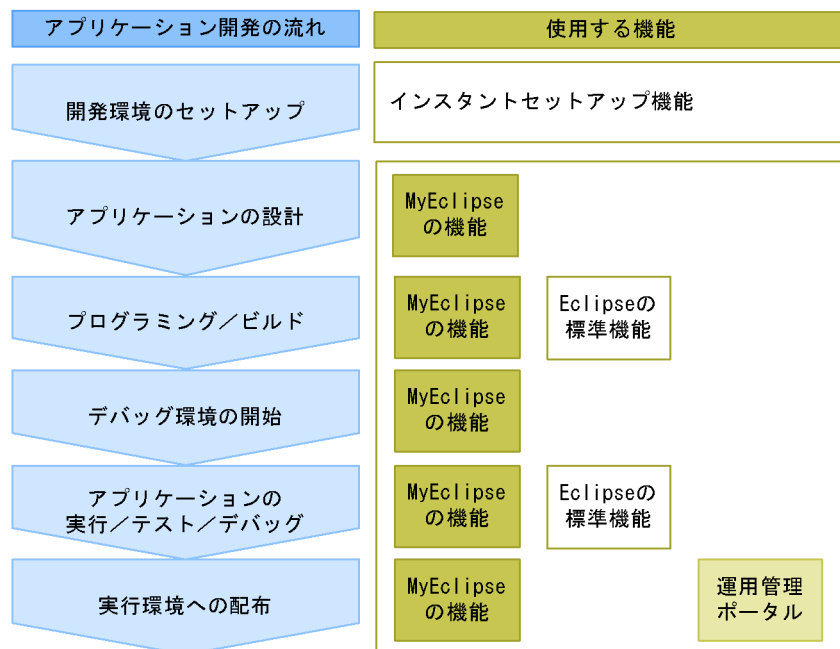
6.3.1 Developer が提供する MyEclipse を使用する

アプリケーションサーバの開発環境では、Developer が提供する MyEclipse を使用して、アプリケーションを開発できます。

MyEclipse 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た、Java アプリケーションの開発環境です。Java アプリケーションの上流設計から、アプリケーションの開発、テスト、デプロイまで、シームレスに実行できます。MyEclipse は Eclipse に組み込んで使用します。これによって、統合された環境で、効率良くアプリケーションを開発できます。なお、MyEclipse は、Developer の機能でセットアップできます。

アプリケーション開発の流れと Developer で使用する機能の関係について、次の図に示します。

図 6-2 アプリケーション開発の流れと使用する機能の関係



アプリケーション開発で実施する作業について説明します。

開発環境のセットアップ

6. アプリケーションを開発する

開発環境に必要な製品をインストールして、開発環境をセットアップします。開発環境のセットアップには、Developer が提供するインスタントセットアップ機能を使用できます。

インスタントセットアップ機能では、次の 2 種類のセットアップができます。

- デバッグ環境のセットアップ（開発環境のインスタントセットアップ機能）
アプリケーションのデバッグを実行するためのデバッグ環境をセットアップできます。デバッグ環境に必要なサーバプロセスとデータベースの環境構築をウィザードプログラムのダイアログから実行できます。
- MyEclipse のセットアップ（MyEclipse セットアップ機能）
Eclipse に MyEclipse を組み込んで、アプリケーション開発ができる状態にします。

アプリケーションの設計

MyEclipse では UML を使用した設計をサポートしています。UML を使用した設計を実施する場合は、MyEclipse の UML パースペクティブを利用できます。

プログラミング/ビルド

MyEclipse を使用してプロジェクトを作成し、JSP、サーブレット、および Enterprise Bean を作成します。なお、データベースなどのリソース接続に必要なリソースアダプタも、MyEclipse のプロジェクトに含めることができます。

MyEclipse の場合、アプリケーションのビルドは、アプリケーションのデプロイ時にあわせて実行されます。

また、Eclipse または MyEclipse の機能を使用して、標準の DD およびアプリケーションサーバ独自の属性を記述した XML ファイルを編集できます。アプリケーションサーバ独自の属性を記述したファイルは、`cosminexus.xml` といいます。アプリケーションのテストでデータベースなどに接続する場合は、`cosminexus.xml` に J2EE リソースのプロパティを設定してください。

デバッグ環境の開始

MyEclipse を使用して、デバッグ環境の J2EE サーバを開始します。また、テスト用のデータベースを開始します。なお、開発環境のインスタントセットアップ機能を使用して構築したシステムの場合は、スタートメニューからデータベースを開始または停止できます。それ以外の場合は、Developer で提供するコマンドを使用して、データベースを開始します。

アプリケーションの実行/テスト/デバッグ

MyEclipse を使用して、アプリケーションを J2EE サーバ上にデプロイします。アプリケーションのデバッグには、Eclipse や MyEclipse のデバッグ機能を使用します。JSP で Web アプリケーションを作成している場合は、JSP デバッグ機能を使用できます。JSP デバッグ機能を使用すると、JSP ファイルに対して、MyEclipse などのデバッグツールの機能を使用できます。JSP ファイルから変換した Java ソースファイルをデバッグする場合に比べて、修正個所の特定などが容易になります。

実行環境への配布

MyEclipse を使用して、アプリケーションを EAR 形式でエクスポートします。エク

サポートした EAR ファイルは、運用管理ポータルを使用して、実行環境にインポートします。

なお、実行環境にインポートしたアプリケーションの実行時属性の設定については、すでに `cosminexus.xml` で属性を設定しているため不要です。設定内容を変更したい場合や、ほかのリソースアダプタを使用したい場合には、必要に応じて `cosminexus.xml` を編集してください。

参考

開発環境のインスタントセットアップ機能を使用して構築したシステムの場合は、データベースへの接続に必要な設定をしたリソースアダプタがデプロイされています。

6.3.2 そのほかに使用できる機能

アプリケーションサーバでは、次のアプリケーションを開発するための機能も提供しています。

WTP を使用したアプリケーションの開発

JAXP で規定された API を利用したアプリケーションの開発

Java EE 環境で動作する CORBA クライアントアプリケーションの開発

これらの機能の詳細については、次のマニュアルを参照してください。

WTP を使用したアプリケーションの開発

マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「11. WTP を使用したアプリケーションの開発」

JAXP で規定された API を利用したアプリケーションの開発、および Java EE 環境で動作する CORBA クライアントアプリケーションの開発

マニュアル「Cosminexus XML Processor ユーザーズガイド」

マニュアル「Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド」

マニュアル「Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス」

注 このマニュアルに記載されている機能のうち、アプリケーションサーバで使用できる機能については、リリースノートを参照してください。

6.4 Web サービスに対応したアプリケーションを開発する

この節では、Web サービス開発時に使用できる機能の概要について説明します。

6.4.1 JAX-WS 仕様に従った Web サービスを開発する機能を使用する

JAX-WS 仕様に従った Web サービスの開発方法には、次の 2 種類があります。

WSDL を起点として開発する方法

最初に Web サービスのメタ情報として WSDL ファイルを作成する方法です。WSDL を基に、コマンドによって、SEI や Web サービスの実装クラスのスケルトンなどに該当する Java ソースを生成します。生成されたスケルトンを基に、Web サービスを実装します。

SEI (サービスエンドポイントインタフェース) を起点として開発する方法

最初に Web サービス実装クラスを作成する方法です。作成した Web サービス実装クラスを基に、コマンドによって Web サービスの実装に必要な追加の Java ソースを生成します。

また、Web サービスのクライアントは、WSDL ファイルから作成します。WSDL ファイルを基に、コマンドによって Web サービスのクライアントの実装に必要な Java ソースを生成します。生成した Java ソースに対して、必要な処理を実装します。

JAX-WS 仕様に従った Web サービスの開発方法については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照してください。

参考

SOAP アプリケーションの開発支援機能を使用する場合

07-60 までのアプリケーションサーバで提供していた SOAP アプリケーション開発支援機能も使用できます。この機能で開発するアプリケーションを SOAP アプリケーションといえます。

SOAP アプリケーションは、Web の標準技術を利用してネットワーク上でサービスを公開、実行できるアプリケーションです。アプリケーションサーバで開発できる SOAP アプリケーションの種類は、大きく分けて、次の 3 種類があります。

• RPC を利用した SOAP アプリケーション

RPC によって SOAP メッセージをやり取りするアプリケーションです。RPC を利用した SOAP アプリケーションでは WSDL が必要になります。アプリケーションサーバでは、WSDL 生成を支援する機能を使用できます。また、生成した WSDL から、実行環境での SOAP メッセージの送受信に必要なスタブおよびスケルトンも自動生成できます。

• メッセージングを利用した SOAP アプリケーション

スタブおよびスケルトンを使用しないで、XML データである SOAP メッセージを直接やり取りするアプリケーションです。XML データを直接やり取りするため、実行時には、アプリケーションサーバが提供する API を利用して、メッセージ生成処理をします。

• EJB を利用した SOAP アプリケーション

既存または新規の EJB を Web サービスとして利用するアプリケーションです。EJB を利用した SOAP アプリケーションでは WSDL が必要になります。

アプリケーションサーバでは、WSDL 生成を支援する機能を使用できます。また、生成した WSDL から、実行環境での SOAP メッセージの送受信に必要なスタブも自動生成できます。

SOAP アプリケーションを利用してネットワーク上のサービスを利用する場合、クライアント側とサーバ側に、それぞれ次の実装が必要です。

クライアント側

SOAP サービスを呼び出す処理、または SOAP サービスを呼び出すためのメッセージを生成する処理の実装。

サーバ側

SOAP サービスの実装。

アプリケーションサーバの開発環境では、SOAP アプリケーションの開発および実行に必要なファイルやソースコードを自動生成できます。新規に SOAP アプリケーションを開発する場合には、必要な定義ファイルおよびサーバとクライアントの実装の基になるスタブとスケルトンを自動生成できます。また、既存のアプリケーションを SOAP サービスにする場合は、必要な定義ファイルを自動生成できます。これによって、アプリケーション開発者は業務ロジックの実装に注力できます。

アプリケーションサーバで開発できる SOAP アプリケーションについては、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

また、SOAP アプリケーションでは、UDDI レジストリにアクセスするための UDDI クライアントライブラリも使用できます。

UDDI レジストリの利用について

ネットワーク上の Web サービスが増えてきた場合、Web サービスを登録・検索するた

6. アプリケーションを開発する

めの情報の格納先として、UDDI レジストリを利用できます。アプリケーションサーバで開発した SOAP アプリケーションを Web サービスとして UDDI レジストリに登録することもできます。

UDDI レジストリへの登録・検索方法については、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」の「8. UDDI クライアントの開発と実行」を参照してください。

6.4.2 Web サービスセキュリティ技術を使用する

ここでは、SOAP アプリケーション開発支援機能を使用する時に使用できる Web サービスセキュリティ技術について説明します。

Web サービスセキュリティ技術は、SOAP アプリケーションで送受信する XML 形式のデータを、改ざんされたり第三者から盗聴されたりすることを防ぐための技術です。また、W3C が規定している XML 署名および XML 暗号の仕様である、XML セキュリティの技術と組み合わせたセキュリティ確保も実現できます。

アプリケーションサーバが提供する API を使用して SOAP アプリケーションを実装することで、Web サービスセキュリティ技術を使用したアプリケーションを開発できます。

また、アプリケーションサーバでは、次の SOAP アプリケーションも開発できます。

XML 署名データを利用するアプリケーション

XML 暗号によってデータの暗号化 / 復号化を実施するアプリケーション

XML 署名データを利用することによって、電子データの完全性を保証したり、成り済ましたり否認を防止したりできます。XML 暗号を利用することによって、データの秘匿性を保証したり、鍵合意によって共通鍵を安全に作成したりできます。

Web サービスセキュリティの詳細については、マニュアル「Cosminexus アプリケーションサーバ Web サービスセキュリティ 使用の手引」を参照してください。また、XML 署名および XML 暗号については、マニュアル「Cosminexus XML Security - Core ユーザーズガイド」を参照してください。

6.5 メッセージを使用した非同期通信用のアプリケーションを開発する

メッセージを使用した非同期通信を実行するには、送信側のアプリケーションと受信側のアプリケーションが必要です。これらのアプリケーションは、アプリケーションサーバが提供する JMS インタフェースを使用して開発できます。また、扱うメッセージは、JMS インタフェースを使用した JMS メッセージです。JMS メッセージは Message インタフェースをルートインタフェースとするオブジェクトです。

送信側および受信側のアプリケーションは、実現する業務内容に合わせて、サーブレット、JSP または Enterprise Bean として作成します。受信側のアプリケーションを Message-driven Bean として作成することで、受信用のキューにメッセージが登録されたことを契機にメッセージを受け取り、必要な業務処理を実行するアプリケーションを開発できます。

非同期通信用アプリケーションの開発については、使用する機能に応じて、次のどちらかのマニュアルを参照してください。

- Cosminexus JMS プロバイダを使用する場合
マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「7. Cosminexus JMS プロバイダ」を参照してください。
- Cosminexus RM を使用する場合
マニュアル「Cosminexus Reliable Messaging」を参照してください。

参考

この節の説明は、アプリケーションサーバの機能を使用した場合の非同期通信の説明です。このほか、Connector 1.5 仕様に準拠した任意のリソースアダプタを使用して、外部のリソースからのメッセージ送信を受け付けて非同期で実行することもできます。この場合のアプリケーションの開発方法については、使用するリソースアダプタの仕様に従ってください。

6.6 バッチアプリケーションを開発する

この節では、バッチアプリケーションの開発について説明します。

バッチアプリケーションは、バッチ処理を実装した Java アプリケーションとして開発します。バッチアプリケーションの作成には、MyEclipse を使用できます。なお、リソース接続などの Java EE の機能をバッチアプリケーションに実装する場合は、必要なライブラリをビルドパスとして設定する必要があります。

バッチアプリケーションのデバッグに使用する環境は、開発環境インスタントセットアップ機能では作成できません。Java EE の機能を使用するバッチアプリケーションのデバッグを実行する場合は、コマンドによって手動でバッチサーバを構築する必要があります。

バッチアプリケーションの開発については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「13.1 バッチアプリケーションの開発の流れ」を参照してください。バッチアプリケーションの実行については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「13.7 バッチアプリケーションの実行」を参照してください。バッチサーバの環境構築については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「13.2 バッチサーバの環境構築」を参照してください。

6.7 アプリケーション開発の流れ

この節では、アプリケーション開発の流れについて、J2EE アプリケーション開発の流れとバッチアプリケーション開発の流れに分けて説明します。

6.7.1 J2EE アプリケーション開発の流れ

アプリケーション開発環境での、開発環境の準備と J2EE アプリケーション開発の流れについて説明します。次の図に、アプリケーションの開発の流れと、参照するマニュアルの対応を示します。

図 6-3 J2EE アプリケーション開発の流れと参照マニュアルの対応



注 マニュアル名称の「Cosminexus アプリケーションサーバ」または「Cosminexus」は省略しています。

注 1 次の6冊が該当します。

- Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)
- Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)
- Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)
- Cosminexus アプリケーションサーバ 機能解説 拡張編
- Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編

6. アプリケーションを開発する

- Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編

注 2 アプリケーションの形式によっては不要です。

注 3 開発環境で cosminexus.xml に定義済みの場合は不要です。

図中の (1) ~ (7) の作業の概要を説明します。なお、作成するアプリケーションによって固有の作業がある場合は、そのつど説明します。

(1) 開発環境のセットアップ

開発環境に必要な製品 (Developer Standard , Developer Professional または Service Architect) をインストールして、開発環境およびデバッグ環境をセットアップします。開発環境およびデバッグ環境のセットアップには、インスタントセットアップ機能を使用できます。この機能によって、次の環境をセットアップできます。

開発に必要な統合開発環境 (Eclipse および MyEclipse)

デバッグ環境 (J2EE サーバを含むアプリケーションサーバおよび組み込みデータベース)

ただし、インスタントセットアップ機能で構築できる環境は、アプリケーションを開発・デバッグするために必要な最小限の環境です。Web サーバや CTM など、開発環境のインスタントセットアップ機能の対象に含まれない機能を使用したテストを実行したい場合には、Smart Composer 機能を使用してシステムを構築してください。

インスタントセットアップ機能については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「2.4 MyEclipse セットアップ機能を使用したセットアップ」を参照してください。Smart Composer 機能については、マニュアル「Cosminexus アプリケーションサーバ システム構築・運用ガイド」を参照してください。

(2) アプリケーションの設計

実現したい業務の内容に応じて適切なアプリケーションの形態を検討、設計します。また、フレームワークを使用する場合は、どのフレームワークを使用するかも検討します。Web アプリケーションを開発する場合は、画面のレイアウトや画面遷移についても設計しておきます。

なお、UML を使用してアプリケーションを設計する場合は、MyEclipse の機能を利用することもできます。

(3) プログラミング / コンパイル / デバッグ

設計した内容に応じて、サーブレット、JSP、Enterprise Bean など処理を実装して、コンパイルします。実装やコンパイルには、Eclipse などを使用します。

また、アプリケーションの内容に応じた DD も定義します。コンパイルで問題が発生し

た場合はデバッグを実施します。

アプリケーションサーバ独自の機能を実装する場合は、機能解説のマニュアルを参照します。実行環境でのシステムの仕様（機能）検討で選択した機能に応じて、必要な機能を実装してください。参照するマニュアルについては、「4.6 導入から運用までの流れ」を参照してください。

また、アプリケーション開発で使用する機能やアプリケーションの種類に応じて、次のマニュアルも参照してください。

J2EE アプリケーションを開発する場合

JSP、サーブレット、および Enterprise Bean を作成します。

- JSP およびサーブレットの作成

MyEclipse を使用すると、ひな型を利用して JSP およびサーブレットを作成できます。また、MyEclipse のビジュアル JSP デザイナーを使用することで、グラフィカルに JSP を編集できます。

- Enterprise Bean の作成

Enterprise Bean の実装には、Eclipse の標準機能を使用できます。

J2EE アプリケーションを開発する場合は、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「1.5 J2EE アプリケーション開発の流れ」を参照してください。

また、J2EE アプリケーション開発とあわせて、アプリケーションやリソースの実行時属性を `cosminexus.xml` に定義します。`cosminexus.xml` の定義方法については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「10. アプリケーションの属性管理」を参照してください。`cosminexus.xml` の詳細については、マニュアル「Cosminexus アプリケーションサーバ リファレンス 定義編（アプリケーション/リソース定義）」の「2. Cosminexus アプリケーション属性ファイル（`cosminexus.xml`）」を参照してください。

Web サービスを開発する場合

サーバ側とクライアント側のアプリケーションの実装が必要です。開発方法には、WSDL を起点にする方法と、SEI を起点にする方法があります。WSDL を起点にする方法の詳細については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」の「2.1.1 WSDL を起点とした開発の流れ」を参照してください。SEI を起点にする方法の詳細については、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」の「2.1.2 SEI を起点とした開発の流れ」を参照してください。

なお、既存機能である SOAP アプリケーション開発支援機能を使用する場合は、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

また、Web サービスセキュリティの技術を使用する場合は、マニュアル

6. アプリケーションを開発する

「Cosminexus アプリケーションサーバ Web サービスセキュリティ 使用の手引」を参照してください。XML 署名および XML 暗号を使用する場合は、マニュアル「Cosminexus XML Security - Core ユーザーズガイド」を参照してください。

メッセージを使用した非同期通信用のアプリケーションを開発する場合

JMS インタフェースを使用してメッセージを操作するアプリケーションを開発します。

Cosminexus JMS プロバイダを使用する場合のアプリケーションの開発方法や注意事項などについては、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「7. Cosminexus JMS プロバイダ」を参照してください。

Cosminexus RM を使用する場合のアプリケーションの開発方法、および Cosminexus RM が提供するインタフェースについては、マニュアル「Cosminexus Reliable Messaging」を参照してください。なお、Cosminexus RM を使用する場合は、転送データ相互接続用インタフェースも使用できます。

(4) アーカイブファイルの作成

実装したサーブレット、JSP、Enterprise Bean を、WAR または EJB-JAR の形式にアーカイブします。さらに、WAR および EJB-JAR を EAR の形式にアーカイブします。アーカイブを作成するには、Eclipse などの IDE や、jar コマンドを使用します。

一度、J2EE サーバにデプロイしたアプリケーションは、サーバ管理コマンドという CUI を使用すると、EAR 形式でアーカイブして、J2EE サーバから取り出すことができます。Eclipse の操作方法については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」を参照してください。サーバ管理コマンドの操作方法については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション設定 操作ガイド」を参照してください。

また、アプリケーションサーバでは、アーカイブファイルを作成しないで、ディレクトリに展開されたファイル群をアプリケーション（展開ディレクトリ形式のアプリケーション）として J2EE サーバにインポートすることもできます。この形式を利用する場合は、アーカイブファイルの作成は不要です。

ポイント

- MyEclipse でアプリケーションを開発する場合、アプリケーション開発中にアプリケーションの形式（アーカイブ形式または展開ディレクトリ形式）を意識する必要はありません。
 - MyEclipse でアプリケーションをデプロイするときに形式を指定すると、アプリケーションは、指定した形式で J2EE サーバにデプロイされます。あらかじめ、アーカイブファイルを作成しておく必要はありません。
-

(5) デバッグ環境の開始

アプリケーションのテストを実行するための環境を準備します。J2EE サーバなど、アプリケーションサーバに必要なプロセスを起動します。また、必要に応じて、データベースなど連携するシステムも起動します。

デバッグ環境に必要なサーバプロセスの起動は、MyEclipse で実行できます。MyEclipse を使用したアプリケーション開発の延長でテストを実施する場合は、MyEclipse から起動すると便利です。

アプリケーションのデプロイ、実行、テストは、MyEclipse で実行できます。また、開発環境のインスタントセットアップ機能で構築したシステムの場合、テストに使用する組み込みデータベースの開始・停止などの操作は、スタートメニューから実行できます。それ以外の場合は、Developer で提供するコマンドを使用して、データベースを開始できます。

MyEclipse については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「2.4 MyEclipse セットアップ機能を使用したセットアップ」を参照してください。なお、デバッグ環境で JSP デバッグ機能を使用する場合は、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)」の「2.4 JSP デバッグ機能」を参照してください。

(6) 実行 / テスト / デバッグ

デバッグ環境にアプリケーションをデプロイ、実行して、テストを実施します。テストで問題があった場合はデバッグを実施します。

デバッグ環境でのアプリケーションの実行、テストについては、MyEclipse で実行します。MyEclipse でのアプリケーションの実行、テストについては、マニュアル「Cosminexus アプリケーションサーバ アプリケーション開発ガイド」の「2.4 MyEclipse セットアップ機能を使用したセットアップ」を参照してください。

ポイント

- 展開ディレクトリ形式のアプリケーションの場合、一度デプロイしたアプリケーションをデバッグで修正したときには、リロードによってアプリケーションを再読み込みできます。リロードをする場合、デバッグ時のアプリケーションの停止、削除、インポート後の再開などの作業が不要になります。アプリケーションのリロードの詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)」の「12.8 J2EE アプリケーションの更新検知とリロード」を参照してください。
 - デバッグ環境のチューニングが必要な場合、運用管理ポータルを使用します。運用管理ポータルは、Web ブラウザから J2EE サーバや Web サーバなどの設定を変更する機能です。運用管理ポータルの使用方法については、マニュアル「Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド」を参照してください。
-

6. アプリケーションを開発する

(7) アプリケーション / リソースの実行時属性の設定

アプリケーションやリソースの実行時属性の設定は、アプリケーション開発時に `cosminexus.xml` に定義します。このため、開発環境で `cosminexus.xml` に属性を設定済みの場合、この手順は不要です。ただし、J2EE アプリケーションを実行環境の J2EE サーバにインポートしたあとで実行時属性を変更する場合には、`cosminexus.xml` の定義を変更するほかに、サーバ管理コマンドを使用して変更することもできます。J2EE アプリケーションを EAR の形式にアーカイブしている場合、`cosminexus.xml` を編集するには、EAR を解凍する必要があるため手間が掛かります。サーバ管理コマンドを使用すると EAR を解凍することなく、アプリケーションやリソースの実行時属性を変更できません。なお、展開ディレクトリ形式のアプリケーションの場合は、`cosminexus.xml` を直接編集できます。サーバ管理コマンドの操作方法については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド」のサーバ管理コマンドによる操作の説明を参照してください。

6.7.2 バッチアプリケーション開発の流れ

アプリケーション開発環境での、開発環境の準備とバッチアプリケーション開発の流れについて説明します。次の図に、アプリケーションの開発の流れと、参照するマニュアルの対応を示します。

図 6-4 バッチアプリケーション開発の流れと参照マニュアルの対応



注 マニュアル名称の「Cosminexus アプリケーションサーバ」または「Cosminexus」は省略しています。

図中の (1) ~ (6) の作業の概要を説明します。なお、作成するアプリケーションによって固有の作業がある場合は、そのつど説明します。

(1) 開発環境のセットアップ

開発環境に必要な製品 (Developer Standard , Developer Professional または Service Architect) をインストールして、開発環境およびデバッグ環境をセットアップします。

MyEclipse の組み込みは、MyEclipse セットアップ機能を使用して実行できます。ただし、デバッグ環境の構築は、インスタントセットアップ機能では実行できません。デバッグ環境のバッチサーバは、コマンドを実行して構築してください。デバッグ用のバッチサーバの環境構築については、マニュアル「Cosminexus アプリケーションサーバアプリケーション開発ガイド」の「2. インストールとセットアップ」を参照してください。

(2) アプリケーションの設計

バッチジョブを実行するアプリケーションの設計をしてください。

(3) プログラミング / コンパイル / デバッグ

設計した内容に応じて、バッチアプリケーションを実装します。バッチアプリケーションは、バッチ処理を実行する Java アプリケーションです。実装やコンパイルには、Eclipse などを使用します。なお、バッチサーバで実行するバッチアプリケーションでは、リソース接続など、一部の Java EE の機能を使用できます。

バッチアプリケーションで使用できる機能については、マニュアル「Cosminexus アプリケーションサーバ機能解説 拡張編」の「2.3 バッチアプリケーション実行機能」を参照してください。

コンパイルで問題が発生した場合はデバッグを実施します。

(4) デバッグ環境の開始

アプリケーションのテストを実行するための環境を準備します。バッチサーバなど、アプリケーションサーバに必要なプロセスを起動します。また、必要に応じて、データベースなど連携するシステムも起動します。

デバッグ環境に必要なサーバプロセスの起動は、コマンドなどを実行して、個別に起動してください。

(5) 実行 / テスト / デバッグ

デバッグ環境にアプリケーションをデプロイ、実行して、テストを実施します。テストで問題があった場合はデバッグを実施します。

(6) リソースの実行時属性の設定

必要に応じて、バッチアプリケーションで利用するリソースをインポートして実行時属

6. アプリケーションを開発する

性を設定します。

リソースの実行時属性の設定は、サーバ管理コマンドで実行できます。サーバ管理コマンドの操作方法については、マニュアル「Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド」のサーバ管理コマンドによる操作の説明を参照してください。

7

ほかの製品との連携

この章では、アプリケーションサーバとほかの製品との連携について説明します。

アプリケーションサーバは、データベース、JP1 およびクラスタソフトウェアと連携して、システムを構築および運用できます。

7.1 データベースとの連携

7.2 JP1 との連携

7.3 クラスタソフトウェアとの連携

7.1 データベースとの連携

アプリケーションサーバは、次のデータベースと接続できます。

- HiRDB Version 9
- HiRDB Version 8
- XDM/RD E2
- Oracle 11g R1
- Oracle 11g R2
- Oracle 10g R2
- Oracle 10g
- Oracle9i R2
- Oracle9i
- SQL Server 2008
- SQL Server 2005
- SQL Server 2000

データベースごとに、使用できる機能が異なります。なお、SQL Server 2008、SQL Server 2005 および SQL Server 2000 は、Windows の場合にだけ使用できます。

ここでは、J2EE サーバから接続できるデータベースおよび接続に使用する JDBC ドライバについて説明します。データベースごとに接続に使用する JDBC ドライバが異なります。なお、データベースとの接続には、JDBC インタフェースを使用して接続する方法と、JDBC インタフェースに加えて JMS インタフェースを使用して接続する方法があります。

J2EE サーバから接続できるデータベース、JDBC ドライバおよび接続に使用するインタフェースの対応を、次の表に示します。

表 7-1 J2EE サーバから接続できるデータベース

データベース	JDBC ドライバ	JDBC インタフェース		JMS インタフェースおよび JDBC インタフェース	
		ローカルランザクション	グローバルランザクション	ローカルランザクション	グローバルランザクション
HiRDB Version 9	HiRDB Type 4 JDBC Driver				
HiRDB Version 8	HiRDB Type 4 JDBC Driver				
XDM/RD E2	HiRDB Type 4 JDBC Driver		-	-	-
Oracle 11g R1 11.1.0	Oracle JDBC Thin Driver				

データベース	JDBC ドライバ	JDBC インタフェース		JMS インタフェースおよび JDBC インタフェース	
		ローカルランザクション	グローバルランザクション	ローカルランザクション	グローバルランザクション
Oracle 11g R2 11.2.0	Oracle JDBC Thin Driver				
Oracle 10g R2 10.2.0	Oracle JDBC Thin Driver				
Oracle 10g 10.1.0	Oracle JDBC Thin Driver				
Oracle9i R2 9.2.0	Oracle JDBC Thin Driver				
SQL Server 2008	SQL Server JDBC Driver		-	-	-
SQL Server 2005	SQL Server 2005 JDBC Driver		-	-	-
SQL Server 2000	SQL Server 2000 Driver for JDBC		-	-	-

(凡例) : 使用できます。 - : 使用できません。

アプリケーションサーバでは、これらのデータベースと接続するためのリソースアダプタを提供しています。接続できるデータベースおよび使用できる機能の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「3.3 リソース接続」を参照してください。

7.2 JP1 との連携

JP1 は、複合的な業務システムの統合運用管理を実現する、日立の統合運用ソフトウェアです。アプリケーションサーバで構築したシステムは、JP1 と連携することによって、高い運用性を持つシステムとして構築・運用できます。

JP1 では、ジョブ管理、アベイラビリティ管理、ネットワーク管理など、多様なシステムの統合管理を実現できます。アプリケーションサーバで構築したシステムとほかの業務システムとを統合して、障害監視や稼働性能監視をしたり、サーバやアプリケーションの起動/停止を自動化したりできるようになります。

JP1 との連携で実現できる機能と使用する製品について、次の表に示します。

表 7-2 JP1 との連携で実現できる機能と使用する製品

機能	概要	使用する製品
障害の集中監視	システム全体を対象に、障害の集中監視ができます。	JP1/IM
稼働性能の監視	システム全体を対象に、稼働性能の監視ができます。	JP1/PFM
システム構成定義と管理	システム構築時に使用するシステムの構成情報を定義したり、定義したシステムの構成情報を収集したりできます。	JP1/IM・CM
ジョブによる運用の自動化	アプリケーションサーバで管理しているサーバやプロセスの起動/停止を JP1/AJS のジョブを使用して自動化できます。	JP1/AJS
シナリオによる運用の自動化	シナリオを使用してシステムの自動運転ができます。	JP1/AJS2・SO
SNMP での稼働情報の取得	SNMP で稼働情報を取得します。	JP1/Cm2/ESA
監査ログの収集と一元管理	アプリケーションサーバなど、システム内の製品が出力した監査ログを収集して一元管理します。	JP1/NETM/Audit

(凡例) - : 該当しません。

注 Smart Composer 機能を使用してシステムを構築している場合に実現できます。

JP1 との連携については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編」の「12. JP1 と連携したシステムの運用」を参照してください。また、SNMP での稼働情報の取得については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用/監視/連携編」の「8. 運用管理コマンドによる稼働情報の出力」を参照してください。

使用する製品の詳細については、それぞれの製品のマニュアルを参照してください。

7.3 クラスタソフトウェアとの連携

アプリケーションサーバで構築したシステムは、クラスタソフトウェアと連携させることによって、可用性を高めた運用を実現できます。

クラスタソフトウェアと連携してアプリケーションサーバで構築したシステムを運用することで、アプリケーションサーバに障害が発生したときに、待機させておいたアプリケーションサーバに切り替えたり、障害が発生したアプリケーションサーバのリカバリ処理を待機しているリカバリサーバで実施したりできます。また、運用管理用のサーバに障害が発生したときにも、待機させておいたサーバに切り替えることができます。これによって、障害によるサーバの不稼働時間を短縮でき、業務処理の中断を最小限に抑えることができます。

アプリケーションサーバが連携できるクラスタソフトウェアを次の表に示します。

表 7-3 連携できるクラスタソフトウェア

アプリケーションサーバが動作する OS	Microsoft Cluster Service	Windows Server Failover Cluster	HA モニタ
Windows	1	2	-
AIX	-	-	
HP-UX	-	-	
Linux	-	-	
Solaris	-	-	-

(凡例) : 使用できます。 - : 使用できません。

注 1 Windows Server 2003 Enterprise Edition が該当します。

注 2 Windows Server 2008 Enterprise が該当します。

クラスタソフトウェアと連携して実現できる機能の詳細については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編」の「18. クラスタソフトウェアとの連携」を参照してください。

8

アプリケーションサーバの マニュアル体系

この章では、アプリケーションサーバのマニュアルの体系について説明します。

8.1 マニュアル体系

8.2 初期導入時のマニュアルの読書手順

8.3 物理サーバ上で実行環境を構築・運用する場合の読書手順

8.4 仮想サーバ上で実行環境を構築・運用する場合の読書手順

8.5 アプリケーションを開発する場合の読書手順

8.1 マニュアル体系

アプリケーションサーバの実行環境および開発環境に対応するマニュアルの体系について説明します。

アプリケーションサーバのマニュアルは、次の4種類に分類できます。

アプリケーションサーバでできることの概要を知りたいときに参照するマニュアル
(このマニュアル)

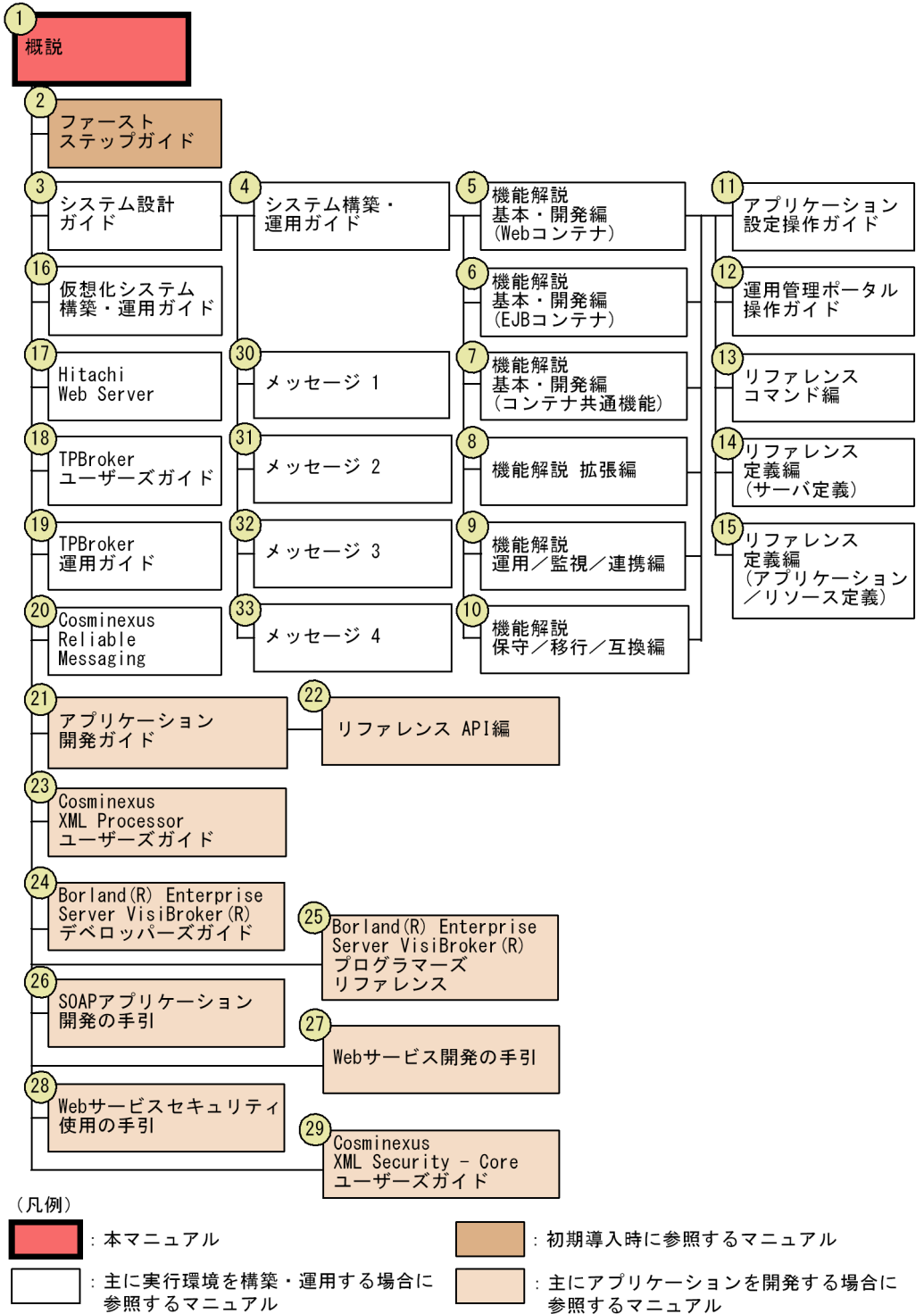
初期導入時に参照するマニュアル

主に実行環境を構築・運用する場合に参照するマニュアル
物理サーバ上で実行環境を構築・運用する場合と、仮想サーバ上で実行環境を構築・
運用する場合に分けて説明しています。

主にアプリケーションを開発する場合に参照するマニュアル

アプリケーションサーバのマニュアル体系を、次の図に示します。図中のマニュアル名の「Cosminexus アプリケーションサーバ」は省略しています。

図 8-1 アプリケーションサーバのマニュアル体系



8. アプリケーションサーバのマニュアル体系

それぞれのマニュアルで説明している内容の概要を次の表に示します。なお、表の項番は図 8-1 の項番と対応しています。

表 8-1 アプリケーションサーバのマニュアル概要

項番	マニュアル名	内容
1	Cosminexus アプリケーションサーバ概説	アプリケーションサーバの製品概要について説明しています。このマニュアルです。
2	Cosminexus アプリケーションサーバファーストステップガイド	サンプルプログラムを動作させるための開発環境または実行環境の構築手順について説明しています。
3	Cosminexus アプリケーションサーバ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。
4	Cosminexus アプリケーションサーバシステム構築・運用ガイド	セットアップウィザードまたは Smart Composer 機能を使用してシステムを構築・運用する方法について説明しています。
5	Cosminexus アプリケーションサーバ機能解説 基本・開発編 (Web コンテナ)	アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、J2EE サーバの機能のうち、Web アプリケーションの実行基盤である Web コンテナの機能について説明しています。また、Web サーバと連携して実現する機能についても説明しています。
6	Cosminexus アプリケーションサーバ機能解説 基本・開発編 (EJB コンテナ)	アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、J2EE サーバの機能のうち、Enterprise Bean の実行基盤である EJB コンテナの機能について説明しています。
7	Cosminexus アプリケーションサーバ機能解説 基本・開発編 (コンテナ共通機能)	アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、J2EE サーバの機能のうち、Web コンテナ、EJB コンテナの両方で使用できる機能について説明しています。
8	Cosminexus アプリケーションサーバ機能解説 拡張編	アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、アプリケーションサーバ独自の拡張機能について説明しています。
9	Cosminexus アプリケーションサーバ機能解説 運用 / 監視 / 連携編	アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。このマニュアルでは、システム運用時に使用する機能、システムを監視するための機能、およびほかの製品と連携するための機能について説明しています。

項番	マニュアル名	内容
10	Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	アプリケーションサーバの機能の詳細について、アプリケーションの実装方法や実行環境に必要な設定などを含めて解説しています。 このマニュアルでは、トラブルが発生した場合のシステム保守に必要な機能、および製品のバージョンアップに伴うシステムの移行について説明しています。また、旧バージョンのアプリケーションサーバで提供していた互換用の機能についても説明しています。
11	Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	サーバ管理コマンドおよび Server Plug-in を使用した J2EE アプリケーションおよびリソースの操作について説明しています。
12	Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	運用管理ポータルの画面および操作について説明しています。
13	Cosminexus アプリケーションサーバ リファレンス コマンド編	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使用するコマンドについて説明しています。
14	Cosminexus アプリケーションサーバ リファレンス 定義編 (サーバ定義)	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使用するファイルの定義方法について説明しています。 このマニュアルでは、J2EE サーバや Management Server などのサーバの定義に使用するファイルについて説明しています。
15	Cosminexus アプリケーションサーバ リファレンス 定義編 (アプリケーション / リソース定義)	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使用するファイルの定義方法について説明しています。 このマニュアルでは、アプリケーションやリソースの属性設定に使用するファイルについて説明しています。
16	Cosminexus アプリケーションサーバ 仮想化システム構築・運用ガイド	アプリケーションサーバを仮想化したサーバ上に構築する場合の設計、構築、運用の手順について説明しています。
17	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。
18	TPBroker ユーザーズガイド	Cosminexus TPBroker の概要、機能、および運用方法について説明しています。
19	TPBroker 運用ガイド	Cosminexus TPBroker の機能のうち、ORB 機能のトラブルシュート、ORB 機能の運用に必要な拡張機能、Cosminexus TPBroker とほかの製品との連携方法、およびバージョンアップ時の移行について説明しています。
20	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。
21	Cosminexus アプリケーションサーバ アプリケーション開発ガイド	アプリケーションの開発方法について説明しています。また、開発環境のセットアップ方法についても説明しています。

8. アプリケーションサーバのマニュアル体系

項番	マニュアル名	内容
22	Cosminexus アプリケーションサーバ 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。
23	Cosminexus XML Processor ユーザーズガイド	Cosminexus XML Processor が提供する XML パーサ・XSLT トランスフォーマの機能, 作成方法, および使用方法について説明しています。
24	Borland(R) Enterprise Server VisiBroker(R) デベロッパーズ ガイド	Borland Enterprise Server VisiBroker の基本的な使用方法および高度な機能の取り扱い方法について説明しています。
25	Borland(R) Enterprise Server VisiBroker(R) プログラマーズ リファレンス	Borland Enterprise Server VisiBroker が提供しているクラスとインタフェースの情報, プログラマツール, およびコマンドラインオプションについて説明しています。
26	Cosminexus アプリケーション サーバ SOAP アプリケーション開発の 手引	SOAP アプリケーション開発支援機能を使用して SOAP アプリケーションを開発, 実行する方法について説明しています。
27	Cosminexus アプリケーション サーバ Web サービス開発の手引	JAX-WS 仕様に従った Web サービスを開発・実行する方法について説明しています。
28	Cosminexus アプリケーション サーバ Web サービスセキュリティ 使用 の手引	Web サービスセキュリティ機能について説明しています。
29	Cosminexus XML Security - Core ユーザーズガイド	XML 署名データの生成・検証機能, および XML 暗号化・復号化機能について説明しています。
30	Cosminexus アプリケーション サーバ メッセージ 1	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出されるメッセージのうち, 次のメッセージについて説明します。 <ul style="list-style-type: none"> • KDAL で始まるメッセージ • KDAN で始まるメッセージ • KDCC で始まるメッセージ • KDCG で始まるメッセージ • Hitachi Web Server が出力するメッセージ
31	Cosminexus アプリケーション サーバ メッセージ 2	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出されるメッセージのうち, 次のメッセージについて説明します。 <ul style="list-style-type: none"> • KDJE で始まるメッセージ • KDJR で始まるメッセージ • KDJW で始まるメッセージ

項番	マニュアル名	内容
32	Cosminexus アプリケーションサーバ メッセージ 3	<p>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出力されるメッセージのうち、次のメッセージについて説明します。</p> <ul style="list-style-type: none"> • KECX で始まるメッセージ • KEDT で始まるメッセージ • KEOS で始まるメッセージ • KEUC で始まるメッセージ • KEXS で始まるメッセージ • KFCB で始まるメッセージ • KFCT で始まるメッセージ • KFDB で始まるメッセージ • KFDJ で始まるメッセージ • KFRM で始まるメッセージ <p>このほか、Cosminexus XML Processor または Cosminexus TPBroker が出力するプリフィックスがないメッセージについても説明します。</p>
33	Cosminexus アプリケーションサーバ メッセージ 4	<p>システムを監査するとき使用する次のメッセージについて説明します。</p> <ul style="list-style-type: none"> • KAWS で始まるメッセージ • KDJE で始まるメッセージ • KEOS で始まるメッセージ • KFCT で始まるメッセージ

8.2 初期導入時のマニュアルの読書手順

初期導入時とは、アプリケーションサーバの導入直後が該当します。初期導入時にマニュアルを参照する目的としては、次の目的が考えられます。

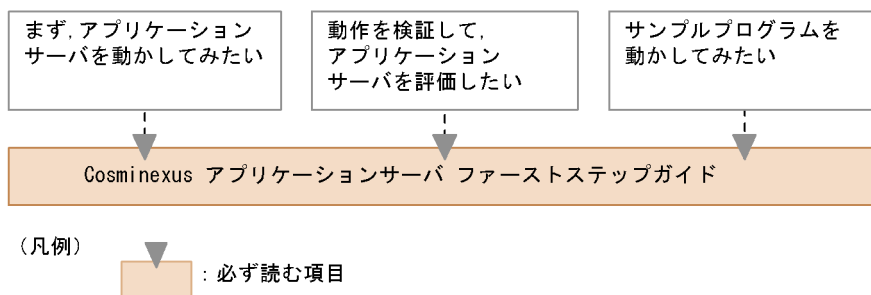
まずはアプリケーションサーバを動かしてみたい。

動作を検証して、アプリケーションサーバを評価したい。

サンプルプログラムを動かしてみたい。

初期導入時に参照するマニュアルを次の図に示します。

図 8-2 初期導入時のマニュアルの読書手順



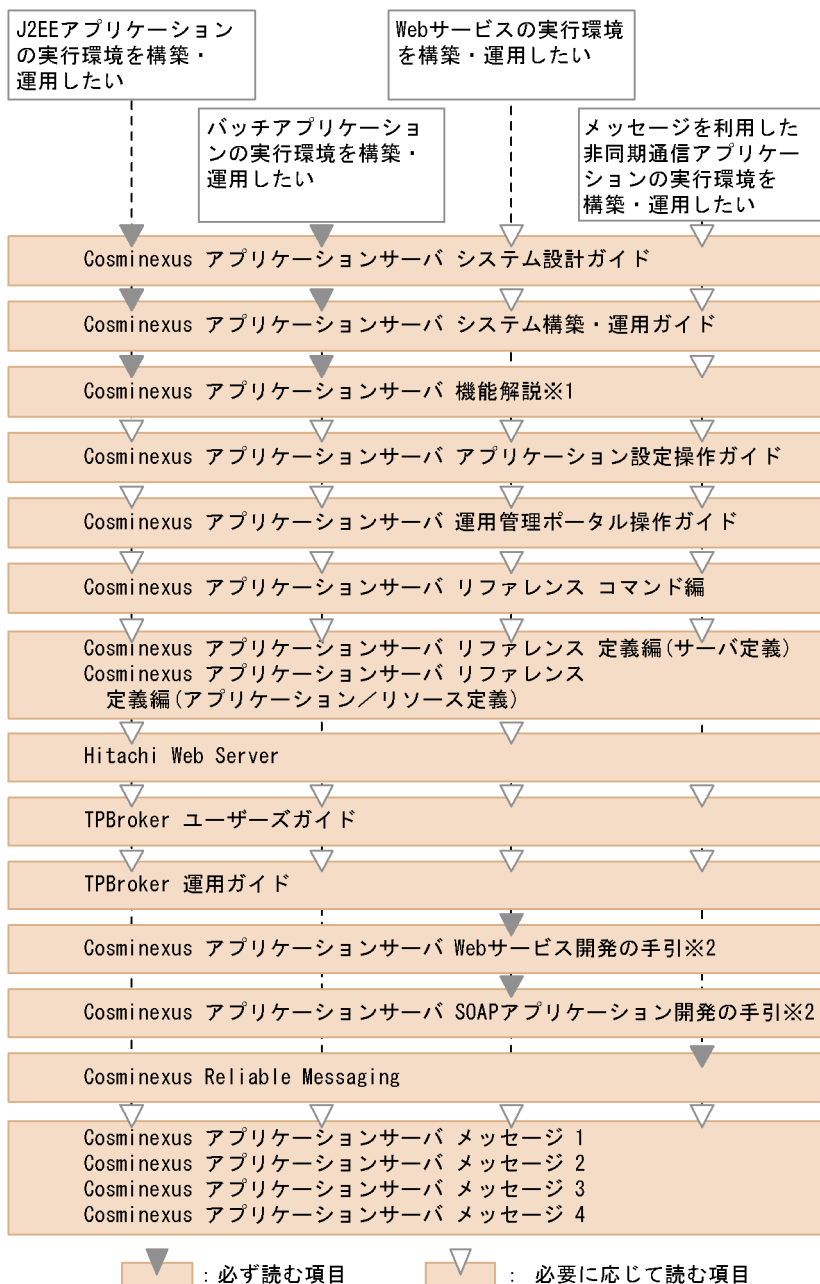
マニュアル「Cosminexus アプリケーションサーバ ファーストステップガイド」の手順に従って作業を進めることで、開発環境および実行環境をすばやく構築して、サンプルプログラムを動かすことができます。

8.3 物理サーバ上で実行環境を構築・運用する場合の読書手順

物理サーバ上で実行環境を構築・運用する場合に参照するマニュアルの読書手順を次の図に示します。

8. アプリケーションサーバのマニュアル体系

図 8-3 実行環境を構築・運用する場合に参照するマニュアルの読書手順



注※1 表8-1の項番5～項番10が該当します。

注※2 準拠する仕様によって、どちらかを選択してお読みください。

8.3.1 J2EE アプリケーションの実行環境を構築・運用する場合に参照するマニュアル

J2EE アプリケーションの実行環境を構築・運用する場合に必ず参照するマニュアルと参照する目的を次に示します。

表 8-2 J2EE アプリケーションの実行環境を構築・運用する場合に必ず参照するマニュアル

マニュアル名	参照する目的
Cosminexus アプリケーションサーバシステム設計ガイド	実行環境のシステム設計をするために参照します。システム構成、メモリの見積もり、チューニング方法などについて知りたい場合に参照してください。
Cosminexus アプリケーションサーバシステム構築・運用ガイド	設計したシステムを実際に構築・運用するために参照します。セットアップウィザードまたは Smart Composer 機能によってシステムを構築・運用する手順について知りたい場合に参照してください。
Cosminexus アプリケーションサーバ機能解説 (6 冊)	実行環境で使用する機能を理解し、機能を使用するためのアプリケーションの実装や実行環境の設定手順について確認するために参照します。

マニュアル「Cosminexus アプリケーションサーバ 機能解説 (6 冊)」は、目的に応じて選択してください。それぞれのマニュアルを参照する目的について次の表に示します。

表 8-3 「Cosminexus アプリケーションサーバ 機能解説 (6 冊)」を参照する目的

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ機能解説 基本・開発編 (Web コンテナ)	Web コンテナ上で動作する Web アプリケーションを実行する場合に参照します。使用できる機能の詳細、実装、設定などが確認できます。 また、Hitachi Web Server などの Web サーバとの連携方法について知りたい場合も、このマニュアルを参照してください。
Cosminexus アプリケーションサーバ機能解説 基本・開発編 (EJB コンテナ)	EJB コンテナ上で動作する EJB アプリケーションを実行する場合に参照します。使用できる機能の詳細、実装、設定などが確認できます。 また、Enterprise Bean を呼び出すクライアントアプリケーション (EJB クライアントアプリケーション) の機能について知りたい場合も、このマニュアルを参照してください。
Cosminexus アプリケーションサーバ機能解説 基本・開発編 (コンテナ共通機能)	次の機能を使用するアプリケーションを実行する場合に参照します。 <ul style="list-style-type: none"> • ネーミング管理 • リソース接続とトランザクション管理 • JPA (Cosminexus JPA プロバイダの機能を含む) • セキュリティ管理 • 属性管理とアノテーション • J2EE アプリケーションの形式とデプロイ機能 • コンテナ拡張ライブラリ 機能の詳細、実装、設定などが確認できます。

8. アプリケーションサーバのマニュアル体系

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ 機能解説 拡張編	<p>次の機能を使用するアプリケーションを実行する場合に参照します。</p> <ul style="list-style-type: none"> • CTM によるリクエストのスケジュールと負荷分散 • J2EE サーバ間のセッション情報の引き継ぎ機能（セッションフェイルオーバー機能） • クライアント性能モニタ機能 • 統合ユーザ管理機能 • 複数の構築済み実行環境の切り替え機能 • アプリケーションのユーザログ出力機能 <p>機能の詳細、実装、設定などが確認できます。</p>
Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	<p>システム運用時に使用する次の機能について知りたい場合に参照します。</p> <ul style="list-style-type: none"> • Smart Composer 機能以外の機能を使用したシステムの起動と停止 • 稼働情報の監視 • リソース枯渇情報の監視 • J2EE アプリケーションに関する運用機能 • システム監査機能（監査ログ） <p>また、JP1 製品など、ほかの製品と連携するための機能について知りたい場合もこのマニュアルを参照してください。</p>
Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	<p>トラブル発生時や、システム移行時に使用できる機能を知りたい場合に参照します。また、旧バージョンのアプリケーションサーバで提供していた互換用の機能について知りたい場合もこのマニュアルを参照してください。</p>

次に示すマニュアルは、J2EE アプリケーションの実行環境を構築・運用する場合に必要なに応じて参照するマニュアルです。それぞれのマニュアルの記載内容と参照する目的について次の表に示します。

表 8-4 J2EE アプリケーションの実行環境を構築・運用する場合に必要なに応じて参照するマニュアル

マニュアル名称	参照する目的
Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	サーバ管理コマンドまたは Server Plug-in を使用して J2EE サーバにアプリケーションまたはリソースをインポートしたり、インポート済みのアプリケーションまたはリソースを操作したりする場合に参照します。
Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド	運用管理ポータルを使用してシステムをチューニングする場合などに参照します。
Cosminexus アプリケーションサーバ リファレンス コマンド編	システムの構築または運用で Smart Composer 機能のコマンドやサーバ管理コマンドなどのコマンドを実行する場合に参照します。
Cosminexus アプリケーションサーバ リファレンス 定義編（サーバ定義）	システムの構築または運用でファイルを定義する場合に参照します。また、アプリケーションサーバで使用できる JavaVM のオプションについて知りたい場合も参照してください。
Cosminexus アプリケーションサーバ リファレンス 定義編（アプリケーション/ リソース定義）	

マニュアル名称	参照する目的
Hitachi Web Server	Web サーバとして Hitachi Web Server を使用する場合に、Web サーバの構築・運用をするときに参照します。
TPBroker ユーザーズガイド	Cosminexus TPBroker のログについて参照したい場合などに、参照します。
TPBroker 運用ガイド	Cosminexus TPBroker の機能のうち、ORB 機能の運用について知りたい場合に参照します。
Cosminexus アプリケーションサーバ メッセージ 1	システムの構築または運用時にメッセージが出力された場合に参照します。参照するマニュアルはプリフィックスから判断してください。
Cosminexus アプリケーションサーバ メッセージ 2	
Cosminexus アプリケーションサーバ メッセージ 3	
Cosminexus アプリケーションサーバ メッセージ 4	監査ログメッセージの出力内容について確認したい場合に参照します。

8.3.2 バッチアプリケーションの実行環境を構築・運用する場合に参照するマニュアル

バッチアプリケーションの実行環境を構築・運用する場合に必ず参照するマニュアルと参照する目的を次に示します。

表 8-5 バッチアプリケーションの実行環境を構築・運用する場合に必ず参照するマニュアル

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ システム設計ガイド	実行環境のシステム設計をするために参照します。 システム構成、メモリの見積もり、チューニング方法などについて知りたい場合に参照してください。
Cosminexus アプリケーションサーバ システム構築・運用ガイド	設計したシステムを実際に構築・運用するために参照します。 Smart Composer 機能によってシステムを構築・運用する手順について参照します。
Cosminexus アプリケーションサーバ 機能解説 (5 冊)	実行環境で使用する機能を理解し、機能を使用するためのアプリケーションの実装や実行環境の設定手順について確認するために参照します。

マニュアル「Cosminexus アプリケーションサーバ 機能解説 (5 冊)」は、目的に応じて選択してください。それぞれのマニュアルを参照する目的について次の表に示します。

8. アプリケーションサーバのマニュアル体系

表 8-6 「Cosminexus アプリケーションサーバ 機能解説（5冊）」を参照する目的

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ 機能解説 拡張編	バッチアプリケーションの実行基盤で使用できる機能について知りたい場合に参照してください。 また、次の機能について知りたい場合も、このマニュアルを参照してください。 <ul style="list-style-type: none">• CTM を使用したバッチアプリケーションのジョブスケジューリング機能• 複数の構築済み実行環境の切り替え機能• アプリケーションのユーザログ出力機能
Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)	これらのマニュアルに記載されている機能の一部は、バッチアプリケーション実行基盤で使用できます。どの機能を使用できるかについては、マニュアル「Cosminexus アプリケーションサーバ 機能解説 拡張編」を参照して確認してください。
Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)	
Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編	システム運用時に使用する次の機能について知りたい場合に参照します。 <ul style="list-style-type: none">• Smart Composer 機能以外の機能を使用したシステムの起動と停止• 稼働情報の監視• リソース枯渇情報の監視• システム監査機能（監査ログ） また、JP1 製品など、ほかの製品と連携するための機能について知りたい場合もこのマニュアルを参照してください。
Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編	トラブル発生時や、システム移行時に使用できる機能を知りたい場合に参照します。また、旧バージョンのアプリケーションサーバで提供していた互換用の機能について知りたい場合もこのマニュアルを参照してください。

バッチアプリケーションの実行環境を構築・運用する場合に必要なに応じて読むマニュアルは、J2EE アプリケーションの実行環境の場合と同じです。「8.3.1 J2EE アプリケーションの実行環境を構築・運用する場合に参照するマニュアル」を参照してください。

8.3.3 Web サービスの実行環境を構築・運用する場合に参照するマニュアル

Web サービスの実行環境を構築する場合、実行する Web サービスが準拠する使用に応じて参照するマニュアルを選択してください。

JAX-WS 仕様に準拠した Web サービスを実行する場合は、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照してください。

SOAP アプリケーション開発支援機能を使用して開発した SOAP アプリケーションを実行する場合は、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

必要に応じて読むマニュアルについては、「8.3.1 J2EE アプリケーションの実行環境を

構築・運用する場合に参照するマニュアル」を参照してください。

8.3.4 メッセージを利用した非同期通信アプリケーションの実行環境を構築・運用する場合に参照するマニュアル

Cosminexus JMS プロバイダを使用して非同期通信アプリケーションを開発する場合は、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「7. Cosminexus JMS プロバイダ」を参照してください。メッセージの送信先を管理するプロセスである CJMSP ブローカーと、CJMSP ブローカーとの接続用のリソースアダプタである CJMSP リソースアダプタを使用するための実行環境を構築・運用する方法については、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」の「7.3 CJMSP リソースアダプタと CJMSP ブローカーの配置」を参照してください。

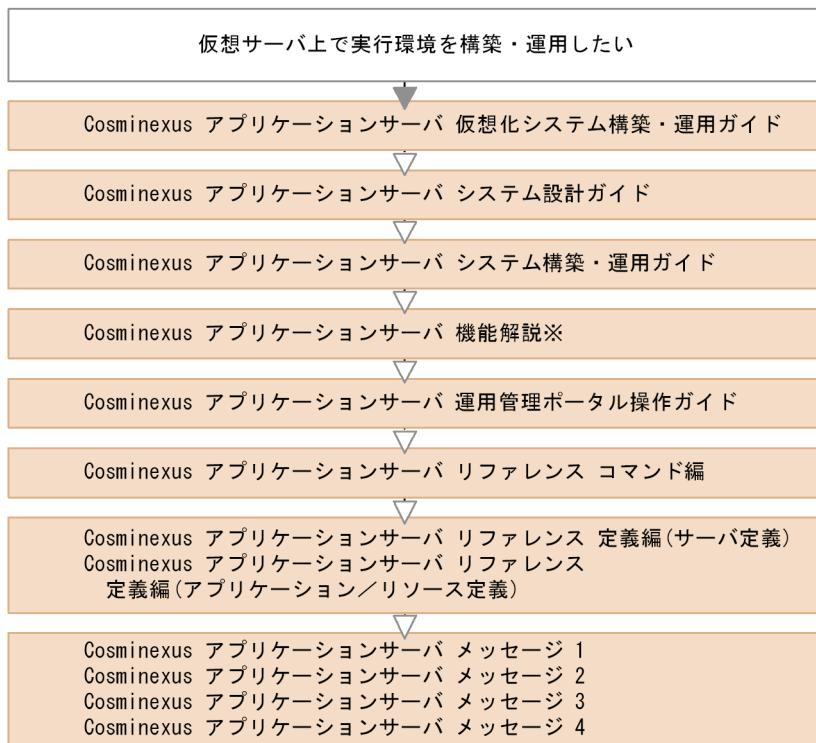
Cosminexus RM を使用して非同期通信アプリケーションの実行環境を構築・運用したい場合は、マニュアル「Cosminexus Reliable Messaging」を参照してください。マニュアル「Cosminexus Reliable Messaging」では、リソースアダプタである Cosminexus RM を使用して実行環境を構築・運用する方法について説明しています。



必要に応じて参照するマニュアルについては、「8.3.1 J2EE アプリケーションの実行環境を構築・運用する場合に参照するマニュアル」を参照してください。

8.4 仮想サーバ上で実行環境を構築・運用する場合の読書手順

仮想サーバ上で実行環境を構築・運用する場合に参照するマニュアルの読書手順を次の図に示します。

図 8-4 仮想サーバ上で実行環境を構築・運用する場合に参照するマニュアルの読書手順



(凡例)  : 必ず読む項目  : 必要に応じて読む項目
注※ 表8-1の項番5～項番10が該当します。

8.4.1 仮想サーバ上で実行環境を構築・運用する場合に参照するマニュアル

仮想サーバ上で実行環境を構築・運用する場合に必ず参照するマニュアルと参照する目的を次に示します。

表 8-7 仮想サーバ上で実行環境を構築・運用する場合に必ず参照するマニュアル

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ 仮想化システム構築・運用ガイド	システムを構築・運用する手順について知りたい場合に参照してください。

なお、仮想サーバ上で動作する J2EE アプリケーションの実行環境を構築・運用するには、次に示すマニュアルを必要に応じて参照してください。それぞれのマニュアルの記載内容と参照する目的について次の表に示します。

表 8-8 仮想サーバ上で実行環境を構築・運用する場合に必要なに応じて参照するマニュアル

マニュアル名称	参照する目的
Cosminexus アプリケーションサーバシステム設計ガイド	システムの設計をするために参照します。メモリの見積もりについて知りたい場合に参照してください。
Cosminexus アプリケーションサーバシステム構築・運用ガイド	設計したアプリケーションサーバの構成定義を作成するために参照します。
Cosminexus アプリケーションサーバ機能解説（6冊）	実行環境で使用する機能を理解し、機能を使用するためのアプリケーションの実装や実行環境の設定手順について確認するために参照します。
Cosminexus アプリケーションサーバ運用管理ポータル操作ガイド	運用管理ポータルを使用して、特定の仮想サーバ上のアプリケーションサーバの設定を一時的に変更したい場合に参照します。
Cosminexus アプリケーションサーバリファレンス コマンド編	システムの構築または運用でコマンドを実行する場合に参照します。
Cosminexus アプリケーションサーバリファレンス 定義編（サーバ定義）	システムの構築または運用でファイルを定義する場合に参照します。また、アプリケーションサーバで使用できる JavaVM のオプションについて知りたい場合も参照してください。
Cosminexus アプリケーションサーバリファレンス 定義編（アプリケーション/リソース定義）	
Cosminexus アプリケーションサーバメッセージ 1	システムの構築または運用時にメッセージが出力された場合に参照します。参照するマニュアルはプリフィックスから判断してください。
Cosminexus アプリケーションサーバメッセージ 2	
Cosminexus アプリケーションサーバメッセージ 3	
Cosminexus アプリケーションサーバメッセージ 4	監査ログメッセージの出力内容について確認したい場合に参照します。

8.5 アプリケーションを開発する場合の読書手順

アプリケーションを開発する場合に参照するマニュアルの読書手順を次の図に示します。

図 8-5 アプリケーションを開発する場合に参照するマニュアルの読書手順



注※1 表8-1の項番5～項番10が該当します。

注※2 準拠する仕様によって、どちらかを選択してお読みください。

8.5.1 J2EE アプリケーションまたはバッチアプリケーションを開発する場合に参照するマニュアル

J2EE アプリケーションまたはバッチアプリケーションを開発する場合に参照するマニユ

8. アプリケーションサーバのマニュアル体系

アルと参照する目的を次に示します。

表 8-9 J2EE アプリケーションまたはバッチアプリケーションを開発する場合に参照するマニュアル

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ アプリケーション開発ガイド	Developer の機能を使用したアプリケーションの開発手順について確認するために参照してください。また、開発環境を構築する手順について知りたい場合も、このマニュアルを参照してください。
Cosminexus アプリケーションサーバ 機能解説 (6 冊)	アプリケーションで使用できる機能を確認する場合に参照してください。また、機能を使用するために必要な実装について知りたい場合に参照してください。
Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	開発したアプリケーションをテストする場合に必要な応じて参照してください。
Cosminexus アプリケーションサーバ リファレンス API 編	アプリケーションサーバが提供する API を使用してアプリケーションを実装したい場合に必要に応じて参照してください。
Cosminexus XML Processor ユーザーズガイド	JAXP を使用してアプリケーションを実装したい場合に必要に応じて参照してください。
Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド	Java EE 環境で動作する CORBA クライアントアプリケーションを開発する場合に必要な応じて参照してください。
Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス	
Cosminexus アプリケーションサーバ メッセージ 1	アプリケーション開発時にメッセージが出力された場合に参照します。参照するマニュアルはプリフィックスから判断してください。
Cosminexus アプリケーションサーバ メッセージ 2	
Cosminexus アプリケーションサーバ メッセージ 3	

マニュアル「Cosminexus アプリケーションサーバ 機能解説 (6 冊)」は、目的に応じて選択してください。それぞれのマニュアルを参照する目的について次の表に示します。

表 8-10 「Cosminexus アプリケーションサーバ 機能解説 (6 冊)」を参照する目的

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)	Web アプリケーションを開発する場合の実装や DD の定義について知りたい場合に参照します。Web アプリケーション実装時の注意事項について知りたい場合も、このマニュアルを参照してください。

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ機能解説 基本・開発編 (EJB コンテナ)	Enterprise Bean を含むアプリケーションを開発する場合の実装や DD の定義について知りたい場合に参照します。 Enterprise Bean 実装時の注意事項について知りたい場合も、このマニュアルを参照してください。
Cosminexus アプリケーションサーバ機能解説 基本・開発編 (コンテナ共通機能)	次の機能の実装方法や注意事項、または DD の定義について知りたい場合に参照します。 <ul style="list-style-type: none"> • ネーミング管理 • リソース接続とトランザクション管理 • JPA (Cosminexus JPA プロバイダの機能を含む) • セキュリティ管理 • 属性管理とアノテーション • J2EE アプリケーションの形式とデプロイ機能 • コンテナ拡張ライブラリ
Cosminexus アプリケーションサーバ機能解説 拡張編	次の機能の実装方法や注意事項、または DD の定義について知りたい場合に参照します。 <ul style="list-style-type: none"> • セッションフェイルオーバー機能 • クライアント性能モニタ機能 • 統合ユーザ管理機能 • ユーザログ出力機能 <p>バッチアプリケーションの開発について知りたい場合もこのマニュアルを参照してください。</p>
Cosminexus アプリケーションサーバ機能解説 運用 / 監視 / 連携編	監査ログを出力するアプリケーションの開発方法について知りたい場合に参照します。
Cosminexus アプリケーションサーバ機能解説 保守 / 移行 / 互換編	日立固有の JavaVM の機能を実装したアプリケーションを開発する方法について知りたい場合に参照します。

8.5.2 Web サービスを開発する場合に参照するマニュアル

Web サービスを開発する場合、実行する Web サービスが準拠する使用に応じて参照するマニュアルを選択してください。

JAX-WS 仕様に準拠した Web サービスを開発する場合は、マニュアル「Cosminexus アプリケーションサーバ Web サービス開発の手引」を参照してください。

SOAP アプリケーション開発支援機能を使用して SOAP アプリケーションを開発する場合は、マニュアル「Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引」を参照してください。

このほか、次の表に示すマニュアルを必要に応じて参照してください。

表 8-11 Web サービスを開発する場合に必要なに応じて参照するマニュアル

マニュアル名	参照する目的
Cosminexus アプリケーションサーバアプリケーション開発ガイド	MyEclipse を使用した Web サービスの開発方法について知りたい場合に参照してください。

8. アプリケーションサーバのマニュアル体系

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	開発したアプリケーションをテストする場合に参照してください。
Cosminexus XML Processor ユーザーズガイド	JAXP を使用してアプリケーションを実装したい場合に参照してください。
Cosminexus アプリケーションサーバ Web サービスセキュリティ 使用の手引	Web サービスセキュリティ機能を使用して実装したい場合に参照してください。
Cosminexus XML Security - Core ユーザーズガイド	XML 署名データの生成・検証機能、および XML 暗号化・復号化機能を使用して実装したい場合に参照してください。
Cosminexus アプリケーションサーバ メッセージ 1	アプリケーション開発時にメッセージが出力された場合に参照します。参照するマニュアルはプリフィックスから判断してください。
Cosminexus アプリケーションサーバ メッセージ 2	
Cosminexus アプリケーションサーバ メッセージ 3	

8.5.3 メッセージを利用した非同期通信アプリケーションを開発する場合に参照するマニュアル

Cosminexus JMS プロバイダを使用して非同期通信アプリケーションを開発する場合は、マニュアル「Cosminexus アプリケーションサーバ 機能解説 基本・開発編（コンテナ共通機能）」を参照してください。

Cosminexus RM を使用して非同期通信アプリケーションを開発する場合は、マニュアル「Cosminexus Reliable Messaging」を参照してください。

このほか、次の表に示すマニュアルを必要に応じて参照してください。

表 8-12 メッセージを利用した非同期通信アプリケーションを開発する場合に必要な
じて参照するマニュアル

マニュアル名	参照する目的
Cosminexus アプリケーションサーバ アプリケーション開発ガイド	Developer の機能を使用したアプリケーションの開発手順について知りたい場合に参照してください。
Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド	開発したアプリケーションをテストする場合に参照してください。
Cosminexus アプリケーションサーバ メッセージ 1	アプリケーション開発時にメッセージが出力された場合に参照します。参照するマニュアルはプリフィックスから判断してください。
Cosminexus アプリケーションサーバ メッセージ 2	
Cosminexus アプリケーションサーバ メッセージ 3	

付録

付録 A アプリケーションサーバが対応する標準仕様

付録 B このマニュアルの参考情報

付録 C 用語解説

付録 A アプリケーションサーバが対応する標準仕様

ここでは、アプリケーションサーバが対応している標準仕様について説明します。

Java EE の標準仕様

アプリケーションサーバは、Java EE 5 に対応しています。サポートしている API のバージョンの詳細については、マニュアル「Cosminexus アプリケーションサーバシステム設計ガイド」の「2.3.9 サーバの動作モード」を参照してください。

XML, Web サービス関連の標準仕様

次の仕様に対応しています。

JAX-WS 2.1

JAXB 2.1

JAXP 1.3 と JAXP 1.4

JAXR 1.0

SAAJ 1.2 と SAAJ 1.3

SOAP 1.1 と SOAP 1.2

WSDL 1.1

WS-RM 1.1 と WS-RM 1.2

UDDI 2.0 と UDDI 3.0

WS-I Basic Profile 1.1

WS-Security 1.1

WS-Reliability 1.1

XML-Signature Syntax and Processing

XML Encryption Syntax and Processing

SSL 関連の標準仕様

- SSL バージョン 2, SSL バージョン 3
- TLS バージョン 1

OMG 分散オブジェクト関連の標準仕様

- CORBA 2.5
- CORBA Object Transaction Service 1.3

付録 B このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 B.1 関連マニュアル

このマニュアルの関連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。これらのマニュアルの概要、およびこれらのマニュアルを参照する目的については、「8. アプリケーションサーバのマニュアル体系」を参照してください。

- Cosminexus アプリケーションサーバ V8 ファーストステップガイド (3020-3-U02)
- Cosminexus アプリケーションサーバ V8 システム設計ガイド (3020-3-U03)
- Cosminexus アプリケーションサーバ V8 システム構築・運用ガイド (3020-3-U04)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (Web コンテナ) (3020-3-U05)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (EJB コンテナ) (3020-3-U06)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (コンテナ共通機能) (3020-3-U07)
- Cosminexus アプリケーションサーバ V8 機能解説 拡張編 (3020-3-U08)
- Cosminexus アプリケーションサーバ V8 機能解説 運用 / 監視 / 連携編 (3020-3-U09)
- Cosminexus アプリケーションサーバ V8 機能解説 保守 / 移行 / 互換編 (3020-3-U10)
- Cosminexus アプリケーションサーバ V8 アプリケーション設定操作ガイド (3020-3-U12)
- Cosminexus アプリケーションサーバ V8 運用管理ポータル操作ガイド (3020-3-U13)
- Cosminexus アプリケーションサーバ V8 リファレンス コマンド編 (3020-3-U14)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (サーバ定義) (3020-3-U15)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (アプリケーション / リソース定義) (3020-3-U16)
- Hitachi Web Server (3020-3-U17)
- Cosminexus アプリケーションサーバ V8 仮想化システム構築・運用ガイド (3020-3-U18)
- TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズガイド (3020-3-U19)
- TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker 運用ガイド (3020-3-U20)
- Cosminexus アプリケーションサーバ V8 Cosminexus Reliable Messaging (3020-3-U21)
- Cosminexus アプリケーションサーバ V8 アプリケーション開発ガイド (3020-3-U25)

- Cosminexus アプリケーションサーバ V8 リファレンス API 編 (3020-3-U26)
- Cosminexus アプリケーションサーバ V8 Cosminexus XML Processor ユーザーズガイド (3020-3-U27)
- Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガイド (3020-3-U28)
- Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス (3020-3-U29)
- Cosminexus アプリケーションサーバ V8 SOAP アプリケーション開発の手引 (3020-3-U30)
- Cosminexus アプリケーションサーバ V8 Web サービス開発の手引 (3020-3-U31)
- Cosminexus アプリケーションサーバ V8 Web サービスセキュリティ 使用の手引 (3020-3-U32)
- Cosminexus アプリケーションサーバ V8 Cosminexus XML Security - Core ユーザーズガイド (3020-3-U33)
- Cosminexus アプリケーションサーバ V8 メッセージ 1 KDAL-KDCG および Hitachi Web Server 編 (3020-3-U41)
- Cosminexus アプリケーションサーバ V8 メッセージ 2 KDJE-KDJW 編 (3020-3-U42)
- Cosminexus アプリケーションサーバ V8 メッセージ 3 KECX-KEDT / KEOS02000-29999 / KEUC-KFRM 編 (3020-3-U43)
- Cosminexus アプリケーションサーバ V8 メッセージ 4 監査ログ編 (3020-3-U44)
- Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス V8 サービスプラットフォーム 概説 (3020-3-U51)

なお、このマニュアルでは、次のマニュアルについて、対象 OS および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
Cosminexus アプリケーションサーバ V8 ファーストステップガイド	Cosminexus アプリケーションサーバ ファーストステップガイド
Cosminexus アプリケーションサーバ V8 システム設計ガイド	Cosminexus アプリケーションサーバ システム設計ガイド
Cosminexus アプリケーションサーバ V8 システム構築・運用ガイド	Cosminexus アプリケーションサーバ システム構築・運用ガイド
Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (Web コンテナ)	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (Web コンテナ)
Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (EJB コンテナ)	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (EJB コンテナ)
Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (コンテナ共通機能)	Cosminexus アプリケーションサーバ 機能解説 基本・開発編 (コンテナ共通機能)

正式名称	このマニュアルでの表記
Cosminexus アプリケーションサーバ V8 機能解説 拡張編	Cosminexus アプリケーションサーバ 機能解説 拡張編
Cosminexus アプリケーションサーバ V8 機能解説 運用 / 監視 / 連携編	Cosminexus アプリケーションサーバ 機能解説 運用 / 監視 / 連携編
Cosminexus アプリケーションサーバ V8 機能解説 保守 / 移行 / 互換編	Cosminexus アプリケーションサーバ 機能解説 保守 / 移行 / 互換編
Cosminexus アプリケーションサーバ V8 アプリケーション設定操作ガイド	Cosminexus アプリケーションサーバ アプリケーション設定操作ガイド
Cosminexus アプリケーションサーバ V8 運用管理ポータル操作ガイド	Cosminexus アプリケーションサーバ 運用管理ポータル操作ガイド
Cosminexus アプリケーションサーバ V8 リファレンス コマンド編	Cosminexus アプリケーションサーバ リファレンス コマンド編
Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (サーバ定義)	Cosminexus アプリケーションサーバ リファレンス 定義編 (サーバ定義)
Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (アプリケーション / リソース定義)	Cosminexus アプリケーションサーバ リファレンス 定義編 (アプリケーション / リソース定義)
Cosminexus アプリケーションサーバ V8 仮想化システム構築・運用ガイド	Cosminexus アプリケーションサーバ 仮想化システム構築・運用ガイド
TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズガイド	TPBroker ユーザーズガイド
TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker 運用ガイド	TPBroker 運用ガイド
Cosminexus アプリケーションサーバ V8 Cosminexus Reliable Messaging	Cosminexus Reliable Messaging
Cosminexus アプリケーションサーバ V8 アプリケーション開発ガイド	Cosminexus アプリケーションサーバ アプリケーション開発ガイド
Cosminexus アプリケーションサーバ V8 リファレンス API 編	Cosminexus アプリケーションサーバ リファレンス API 編
Cosminexus アプリケーションサーバ V8 Cosminexus XML Processor ユーザーズガイド	Cosminexus XML Processor ユーザーズガイド
Cosminexus アプリケーションサーバ V8 SOAP アプリケーション開発の手引	Cosminexus アプリケーションサーバ SOAP アプリケーション開発の手引
Cosminexus アプリケーションサーバ V8 Web サービス開発の手引	Cosminexus アプリケーションサーバ Web サービス開発の手引
Cosminexus アプリケーションサーバ V8 Web サービスセキュリティ 使用の手引	Cosminexus アプリケーションサーバ Web サービスセキュリティ 使用の手引
Cosminexus アプリケーションサーバ V8 Cosminexus XML Security - Core ユーザーズガイド	Cosminexus XML Security - Core ユーザーズガイド

正式名称	このマニュアルでの表記
Cosminexus アプリケーションサーバ V8 メッセージ 1 KDAL-KDCG および Hitachi Web Server 編	Cosminexus アプリケーションサーバ メッセージ 1
Cosminexus アプリケーションサーバ V8 メッセージ 2 KDJE-KDJW 編	Cosminexus アプリケーションサーバ メッセージ 2
Cosminexus アプリケーションサーバ V8 メッセージ 3 KECK-KEDT / KEOS02000-29999 / KEUC-KFRM 編	Cosminexus アプリケーションサーバ メッセージ 3
Cosminexus アプリケーションサーバ V8 メッセージ 4 監査ログ編	Cosminexus アプリケーションサーバ メッセージ 4
Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス V8 サービスプラットフォーム 概説	Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス サービスプラットフォーム 概説

付録 B.2 このマニュアルでの表記

このマニュアルでは、製品名を次のように表記しています。

表記		製品名
Application Server	Application Server Enterprise	uCosminexus Application Server Enterprise
	Application Server Standard	uCosminexus Application Server Standard uCosminexus Application Server Standard-R
Developer	Developer Professional	uCosminexus Developer Professional
	Developer Standard	uCosminexus Developer Standard
Eclipse		Eclipse 3.6.1
HiRDB	HiRDB Server	HiRDB Server Version 9
	HiRDB/Parallel Server	HiRDB/Parallel Server Version 8
	HiRDB/Single Server	HiRDB/Single Server Version 8
HiRDB Run Time または HiRDB クライアント		HiRDB/Run Time Version 8 HiRDB/Run Time Version 9
IPF		Itanium(R) Processor Family
JP1/AJS	JP1/AJS - Agent	JP1/Automatic Job Management System 2 - Agent
		JP1/Automatic Job Management System 3 - Agent
	JP1/AJS - Manager	JP1/Automatic Job Management System 2 - Manager
		JP1/Automatic Job Management System 3 - Manager

表記		製品名
	JP1/AJS - View	JP1/Automatic Job Management System 2 - View
		JP1/Automatic Job Management System 3 - View
JP1/AJS2 - SO	JP1/AJS2 - SO Manager	JP1/Automatic Job Management System 2 - Scenario Operation Manager
	JP1/AJS2 - SO View	JP1/Automatic Job Management System 2 - Scenario Operation View
JP1/Cm2	JP1/Cm2/ESA	JP1/Cm2/Extensible SNMP Agent
JP1/ESP		JP1/Extensible Service Probe
JP1/IM	JP1/IM - CM	JP1/Integrated Management - Central Information Master
	JP1/IM - Manager	JP1/Integrated Management - Manager
	JP1/IM - View	JP1/Integrated Management - View
JP1/NETM/Audit		JP1/NETM/Audit - Manager
JP1/PFM		JP1/Performance Management
JP1/SC/DPM		JP1/ServerConductor/Deployment Manager Standard Edition
Oracle	Oracle 10g	Oracle 10g
		Oracle Database 10g
	Oracle 10g R2	Oracle 10gR2
		Oracle Database 10g R2
	Oracle 11g	Oracle Database 11g
		Oracle Database 11g R2
	Oracle9i	Oracle9i
	Oracle9i R2	Oracle9i R2
Service Architect		uCosminexus Service Architect
Service Platform		uCosminexus Service Platform
Struts		Jakarta Struts 1.1
UNIX	AIX	AIX 5L V5.3
		AIX V6.1
		AIX V7.1
	HP-UX	HP-UX 11i V2 (IPF)
	HP-UX (IPF)	HP-UX 11i V3 (IPF)
	Linux	Red Hat Enterprise Linux(R) AS 4 (IPF)
	Linux (IPF)	Red Hat Enterprise Linux(R) 5 Advanced Platform (Intel Itanium)
		Red Hat Enterprise Linux(R) 5 (Intel Itanium)

表記		製品名
	Linux (x86/ AMD64 & Intel EM64T)	Red Hat Enterprise Linux AS 4 (x86)
		Red Hat Enterprise Linux ES 4 (x86)
		Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T)
		Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T)
		Red Hat Enterprise Linux 5 Advanced Platform (x86)
		Red Hat Enterprise Linux 5 (x86)
		Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)
		Red Hat Enterprise Linux 5 (AMD/Intel 64)
		Red Hat Enterprise Linux Server 6 (32-bit x86)
		Red Hat Enterprise Linux Server 6 (64-bit x86_64)
	Solaris	Solaris 10 (SPARC)
		Solaris 10 (x64)
		Solaris 9 (SPARC)
VMware	VMware ESX	VMware(R) ESX
	VMware Tools	VMware(R) Tools
	VMware vCenter Server	VMware(R) vCenter Server
	VMware vSphere Client	VMware(R) vSphere Client
XDM/RD E2		VOS3 XDM/RD E2

注 総称して、JP1/AJS2 と表記することがあります。

また、Linux に関しては、バージョンごとに次のように表記することがあります。

表記	OS 名
Red Hat Enterprise Linux 4	Red Hat Enterprise Linux(R) AS 4 (IPF)
	Red Hat Enterprise Linux(R) AS 4 (x86)
	Red Hat Enterprise Linux(R) ES 4 (x86)
	Red Hat Enterprise Linux(R) AS 4(AMD64 & Intel EM64T)
	Red Hat Enterprise Linux(R) ES 4(AMD64 & Intel EM64T)

表記	OS 名
Red Hat Enterprise Linux 5	Red Hat Enterprise Linux(R) 5 Advanced Platform (Intel Itanium)
	Red Hat Enterprise Linux(R) 5 (Intel Itanium)
	Red Hat Enterprise Linux(R) 5 Advanced Platform (x86)
	Red Hat Enterprise Linux(R) 5 (x86)
	Red Hat Enterprise Linux(R) 5 Advanced Platform (AMD/ Intel 64)
	Red Hat Enterprise Linux(R) 5 (AMD/Intel 64)
Red Hat Enterprise Server Linux 6	Red Hat Enterprise Linux Server 6 (32-bit x86)
	Red Hat Enterprise Linux Server 6 (64-bit x86_64)

このマニュアルで使用している表記と、対応するアプリケーションサーバの機能名を次に示します。

表記	アプリケーションサーバの機能名
CJMSP ブローカー	Cosminexus JMS プロバイダのブローカー機能
CJMSP リソースアダプタ	Cosminexus JMS プロバイダのリソースアダプタ
Cosminexus Developer's Kit for Java	Cosminexus Developer's Kit for Java TM
Cosminexus RM	Cosminexus Reliable Messaging
CTM	Cosminexus Component Transaction Monitor
DB Connector for Cosminexus RM	DB Connector for Cosminexus Reliable Messaging
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
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
Connector 1.0	J2EE TM Connector Architecture 1.0
Connector 1.5	J2EE TM Connector Architecture 1.5

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	J2EE TM 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
Java SE	Java TM Platform, Standard Edition
JavaVM	Java TM Virtual Machine
JAX-WS	Java TM API for XML Web Services
JAXP	Java TM API for XML Processing
JCA	J2EE TM Connector Architecture
JCE	Java TM Cryptography Extension
JDBC	Java TM Database Connectivity JDBC TM
JDK	Java TM Development Kit JDK TM
JMS	Java TM Message Service
JNDI	Java TM Naming and Directory Interface TM
JNI	Java TM Native Interface
JSF	JavaServer TM Faces Reference Implementation (RI) Version: 1.1_01 FCS
JSP	JavaServer Pages TM

表記	Java 関連用語
	JSP™
JSTL	JavaServer Pages™ Standard Tag Library
JTA	Java™ Transaction API
JTS	Java™ Transaction Service
SAAJ	SOAP with Attachments API for Java™
Servlet またはサーブレット	Java™ Servlet
WAR	Web ARchive

付録 B.3 英略語

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	Application Programming Interface
BPPEL	Business Process Execution Language
BPM	Business Process Management
CORBA	Common Object Request Broker Architecture
CUI	Character User Interface
DB	Database
DBMS	Database Management System
DD	Deployment Descriptor
EIS	Enterprise Information System
ESB	Enterprise Service Bus
GUI	Graphical User Interface
HA	High Availability
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IIOP	Internet Inter-Orb Protocol
LDAP	Lightweight Directory Access Protocol
MDA	Model Driven Architecture
OLTP	On-Line Transaction Processing
OMG	Object Management Group
ORB	Object Request Broker
OS	Operating System

英略語	英字での表記
OTM	Object Transaction Monitor
RDB	Relational Database
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SEI	Service Endpoint Interface
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SUP	Service Using Program
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URL	Uniform Resource Locator
VM	Virtual Machine
WSDL	Web Service Description Language
WTP	Eclipse Web Tools Platform
XML	Extensible Markup Language

付録 B.4 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。

付録 C 用語解説

アプリケーションサーバで使用する用語について説明します。

(英字)

Application Server

アプリケーションサーバ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。

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。

Cosminexus Component Transaction Monitor

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構成ソフトウェアです。

Cosminexus Manager

アプリケーションサーバのシステムの運用管理機能です。

Cosminexus Performance Tracer

リクエストが処理されるときに、決められたポイントごとに各機能が出力する性能解析情報をファイルに出力するための構成ソフトウェアです。

Cosminexus Reliable Messaging

アプリケーションサーバで構築したシステム上の 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構成ソフトウェアです。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 アプリケーションサーバ

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 J2EE 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。

CTM

Cosminexus Component Transaction Monitor のことです。リクエストのスケジューリングをするための機能を提供しています。

DB Connector

データベースに接続するためのリソースアダプタです。

Developer

アプリケーションサーバの開発環境を提供する製品です。アプリケーションの開発からデバッグまでを支援します。Developer Standard および Developer Professional の総称です。

Eclipse

Eclipse プロジェクト (eclipse.org) 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。

Eclipse プラグイン

Developer が提供する Eclipse 用のプラグインです。このプラグインを Eclipse に組み込むことで、Eclipse からアプリケーション開発ツールを起動したり、開発したアプリケーションを実行したりできます。

EIS

データベースやトランザクションサーバなど、企業内に構築されているバックエンドシステムです。

EJB クライアント

J2EE サーバ上で開始されている Enterprise Bean を呼び出すクライアントプログラムです。次の 3 種類があります。

- EJB クライアントアプリケーション
- サブレットまたは JSP などの Web アプリケーション
- ほかの Enterprise Bean

EJB クライアントアプリケーション

Enterprise Bean を呼び出す Java アプリケーションです。

EJB コンテナ

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 の実体は、EJB コンテナの中で実行されます。

J2EE アプリケーション

JSP、サブレット、Enterprise Bean などで構成されるアプリケーションです。アプリケーションサーバで扱う 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 サーバ

J2EE コンテナを生成、実行する環境です。

JavaVM

Java Virtual Machine の略です。Java 言語で作られたプログラムを実行するためのソフトウェアです。

アプリケーションサーバが使用する JavaVM は、構成ソフトウェアである Cosminexus Developer's Kit for Java によって提供される、独自の JavaVM です。

JP1

日立の統合システム運用管理ソフトウェアです。

JSP

JavaServer Pages の略です。Web ページを作成する HTML の中に JSP 固有のタグを埋め込んで Enterprise Bean を呼び出すことや、サブレットから値を受け取ることができます。また、簡単な

Java 言語を直接記述することもできます。

Management Server

アプリケーションサーバの運用管理を実現するサーバプログラムです。複数のマシンに配置されたアプリケーションサーバの運用管理を一括で実行できます。

MVC

Model と View ,そして Controller の頭文字を持ったものです。Model は、システムの中でロジックを担当します。View は、表示、入出力といった部分を担当します。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。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。

Developer では、構成ソフトウェアとして、MyEclipse の日本語版環境である MyEclipse for Cosminexus を提供しています。

Smart Composer 機能

アプリケーションサーバ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、一括で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、および J2EE アプリケーションや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、迅速にシステム構築ができます。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運用機能についてもサポートしています。

SOAP アプリケーション

Web の標準技術を利用してネットワーク上でサービスを公開、実行できるアプリケーションです。

uCosminexus Client

EJB クライアント環境を構築するための製品です。

uCosminexus Operator

Application Server Standard または Application Server Enterprise をインストールした実行環境のマシンとは別の Windows マシンから、GUI (Eclipse のプラグイン) を使用して実行環境を操作するための製品です。Application Server Standard または Application Server Enterprise に同梱されています。

uCosminexus Web Redirector

実行環境を構築する場合に、Web サーバとアプリケーションサーバを異なるマシンに配置するとき使用する製品です。Web Redirector 機能を提供します。

uCosminexus Web Redirector を使用できる OS については、「2.4 動作環境」を参照してください。

Web Redirector 機能

実行環境を構築する場合に、Web サーバとアプリケーションサーバを異なるマシンに配置するとき

に使用する機能です。uCosminexus Web Redirector または Application Server で提供される機能です。

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、HTML/XML ドキュメントなどの集合体です。

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。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する実行環境です。Web アプリケーションは、Web コンテナ上で動作します。

Java Servlet2.5 仕様、および JavaServer Pages Specification v2.1 仕様に準拠した Web アプリケーションを実行できます。

Web サーバ

Web ブラウザからのリクエスト受信および Web ブラウザへのデータ送信に関連する処理を実行するプログラムです。アプリケーションサーバでは、Hitachi Web Server、Microsoft IIS、またはインプロセス HTTP サーバを使用できます。インプロセス HTTP サーバは、J2EE サーバプロセス内で動作する Web サーバです。

Web サービス

インターネット上に分散するアプリケーションが提供するサービスを、クライアントに API として提供するサービスです。

(ア行)

アプリケーションサーバ

情報システムの中間に位置し、ユーザの要求（プレゼンテーション層）とデータベースなどの業務システム（データ層）の処理を橋渡すためのアプリケーション層を構築するためのミドルウェアです。業務の開発から運用まで一貫した環境を提供します。

(カ行)

業務アプリケーション

業務単位の J2EE アプリケーションや Web アプリケーション、パッチアプリケーションなどのことです。

(サ行)

サービスユニット

J2EE サーバや Web サーバで構成される業務サービスを提供する、閉じた部分系のことです。Smart Composer 機能では、サービスユニット単位でシステムを構築したり、一括起動や一括停止をしたりできます。

Smart Composer 機能を利用してシステムを構築するときの概念です。

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 サーバに対して、単に HTML 文書や画像ファイルを送るだけではなく、Web サーバと連携して、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。

スケールアウト

システム全体の処理性能を向上させることを目的として、サーバの台数を増やすことをいいます。

スケールイン

システムの規模を縮小する場合などに、サーバの台数を減らすことをいいます。

(タ行)

デプロイ

J2EE アプリケーションの場合、J2EE サーバ内にインポートした J2EE アプリケーションを、クライアントから実行可能な状態にすることです。

J2EE リソースアダプタの場合、J2EE サーバ内にインポートした J2EE リソースアダプタを、その J2EE サーバ上で動作するすべての J2EE アプリケーションから使用可能な状態にすることです。

(ハ行)

バッチアプリケーション

バッチ処理を実装した Java アプリケーションです。バッチ実行コマンドを使用して、バッチサーバ上で実行します。また、バッチ実行コマンドを JP1 のジョブとして定義しておくことで、JP1/AJS2 からバッチアプリケーションを実行できます。

バッチサーバ

バッチアプリケーションを実行するためのサーバプロセスです。バッチアプリケーションを実行できます。また、バッチアプリケーションからデータベースに接続したり、EJB にアクセスしたりするための機能も提供しています。

負荷分散機

Web ブラウザなどからのリクエストを一元的に受け付けて、同等の機能を持つ複数のサーバに転送して各サーバの負荷を分散させるための装置です。

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。アプリケーションフレームワークを使用することで、J2EE アプリケーションを容易に作成できます。

(ラ行)

リソースアダプタ

J2EE Connector Architecture によって、J2EE サーバと EIS を接続するための接続機能です。

リソースマネージャ

リソースを管理する機能です。DBMS などが該当します。

リダイレクタ

Web サーバに登録し Web コンテナへの接続を可能にするプラグインコンポーネント (ライブラリ) です。Web サーバに登録したリダイレクタによって、リクエストを URL パターンまたはラウンドロビン方式で複数の Web コンテナに振り分けて処理できます。

索引

数字

2 フェーズコミットメント 54

A

Apache2.0 38

Application Server 177

Application Server Enterprise 24

Application Server Standard 22

Application Server の構成ソフトウェアの位置づけ (SOAP アプリケーションを実行する場合) 42

Application Server の構成ソフトウェアの位置づけ (Web クライアントまたは EJB クライアントからのリクエストを処理する場合) 40

B

BPM 15

C

CJMSP ブローカー 94

CJMSP リソースアダプタ 94

CORBA クライアントアプリケーション 123
cosminexus.xml 73

Cosminexus Application Development Plug-in 35

Cosminexus Component Container 35, 40, 42, 177

Cosminexus Component Container - Client 37

Cosminexus Component Container - Redirector 37

Cosminexus Component Transaction Monitor 37, 41, 177

Cosminexus Developer's Kit for Java 37

Cosminexus JMS プロバイダを利用した実行環境 94

Cosminexus Manager 41, 177

Cosminexus Performance Tracer 37, 41, 177

Cosminexus Reliable Messaging 38, 177

Cosminexus RM 95

Cosminexus Server Plug-in 38, 41

Cosminexus TPBroker 38

Cosminexus Web Services - Security 38, 42

Cosminexus XML Processor 38

Cosminexus アプリケーションサーバ 177

Cosminexus コネクタ 35

Cosminexus ビジネスプロセス管理 / エンタープライズサービスバス 15

CTM 41, 177

D

DB Connector 177

DB Connector for Cosminexus RM 95

Developer 177

Developer Professional 29

Developer Standard 28

Developer の構成ソフトウェアの位置づけ 45

DOM API 55, 58

E

Eclipse 177

Eclipse プラグイン 178

EIS 41, 178

EJB クライアント 41, 178

EJB クライアントアプリケーション 178

EJB コンテナ 41, 54, 178

EJB を利用した SOAP アプリケーション 125

EJB を利用した SOAP アプリケーションの開発 55

Enterprise Bean の優先制御, 流量制御および負荷分散 69

ESB 15

H

HiRDB Embedded Server Version 8 38
Hitachi Web Server 38, 40

J

J2EE アプリケーション 4, 178
J2EE アプリケーション開発の流れ 129
J2EE アプリケーション実行環境 3
J2EE アプリケーションの実行環境の種類 82
J2EE アプリケーションのリロード 56
J2EE サーバ 3, 178
J2EE サーバ間のセッション情報の引き継ぎ 55
Java 2 Platform Standard Edition 5.0 37
Java EE 36
Java EE 5 66
Java Platform, Standard Edition 6 37
JavaServer Pages 178
JavaVM 178
Java 言語 54, 57
JAX-WS エンジン 89
JAXP 123
JMS インタフェース 127
JMS メッセージ 127
JNDI によるルックアップ 54, 58
JP1 140, 178
JP1 との連携 57, 59, 140
JSP 178
JSP 事前コンパイル 56
JSP デバッグ機能 122

M

Management Server 62, 179
MVC 179
MyEclipse 121, 179
MyEclipse for Cosminexus 39
MyEclipse セットアップ機能 35
MyEclipse を使用したアプリケーションの開発 56, 58

N

N:1 リカバリシステム構成 57

O

OLTP 技術 68
OLTP 技術の適用 55

R

RPC を利用した SOAP アプリケーション 125
RPC を利用した SOAP アプリケーションの開発 55

S

SAX API 55, 58
Smart Composer 機能 62, 85, 179
SOAP 89
SOAP アプリケーション 179
SOAP アプリケーションの実行環境の例 91
SOAP エンジン 90
SOAP メッセージ 89
SOAP メッセージの完全性および秘匿性の保証 55
SOAP メッセージの認証 55

U

uCosminexus Client 29, 179
uCosminexus Operator 31, 179
uCosminexus Web Redirector 30, 179
UDDI クライアントライブラリが提供する JAXR API 55
UDDI レジストリ 125
URL グループ 67

W

Web Redirector 機能 179
Web アプリケーション 180
Web アプリケーションの流量制御 67
Web 環境 48
Web クライアント 40

Web コンテナ 40, 54, 180
 Web サーバ 40, 48, 54, 180
 Web サービス 55, 78, 180
 Web サービスおよび SOAP 間通信 55
 Web サービスセキュリティ 91
 Web サービスセキュリティの技術を利用した
 システムの例 92
 Web サービスに対応したアプリケーション
 89
 Web サービス利用環境 79
 Web ブラウザ 48
 WS-Reliability 96
 WS-Security 38
 WTP 123
 WTP コネクタ 35

X

XML 暗号 126
 XML 署名データ 126
 XML 署名データの暗号化および復号化 55
 XML 署名データの生成および検証 55
 XML プロセッサ 55, 58
 XSLT トランスフォーマ 55, 58

あ

アーカイブファイルの作成 132
 アプリケーション / リソースの実行時属性の
 設定 134
 アプリケーション入れ替え 78
 アプリケーション開発 56, 58
 アプリケーション開発環境 3
 アプリケーションサーバ 4, 180
 アプリケーションサーバが対応する標準仕様
 166
 アプリケーションサーバの主な機能 57
 アプリケーションサーバの製品構成 19
 アプリケーション実行環境 3
 アプリケーションの開発 61
 アプリケーションの設計 130, 135
 アプリケーションの設定 56
 アプリケーションのデプロイ 56

い

イベント発行 56, 59
 インスタントセットアップ機能 122

う

運用管理機能 41
 運用作業の自動化 77
 運用設計 60

お

オプション製品 21

か

開発環境 8
 開発環境のインスタントセットアップ機能
 35
 開発環境の構成 118
 開発環境の構成ごとにインストールする製品
 119
 開発環境のセットアップ 130, 135
 開発環境を構築する製品 20
 仮想化したサーバ上での J2EE アプリケー
 ション実行環境 11
 稼働情報監視 55, 58
 可用性 69
 可用性向上 55, 58
 簡易構築 56, 58
 監査証跡情報 75
 監査ログ 74
 監査ログ出力 55, 58

き

業務アプリケーション 180
 業務効率を向上させる運用管理の実現 76
 業務指向スタイル 8
 業務システムの一括運用 77

く

クライアント性能の測定と分析 77
 クラスタソフトウェアとの連携 57, 59, 141

け

言語 48

こ

構成ソフトウェア 33
コールドスタンバイでの 1:1 の系切り替え
57, 59
コネクションシェアリング 54, 58
コネクションプーリング 54, 58
コンパイル 130, 135

さ

サービスデプロイメント 3
サービスプラットフォーム 15
サービス閉塞 69
サービスユニット 85, 180
サブレット 181

し

システム運用 56, 58
システム構成の一括定義 56, 58
システム構成の種類 83
システム構築 56, 58
システム設計 60
システム導入および拡張の容易化 72
システムの安定稼働 66
システムの運用と保守 63
システムの稼働情報やリソースの使用状況の
出力 77
システムの構築 62
システムの仕様（機能）検討 54
システムのライフサイクル 51, 52
実行 133, 135
実行環境 3, 66
実行環境と開発環境の関係 9
実行環境の移行性向上 56, 58
実行環境を構築する製品 20
自動アクション制御 56, 59
障害発生時の可用性向上 71
障害発生の未然防止 71
詳細なログの出力 71

信頼性の高い非同期通信 79

す

スケールアウト 83, 181
スケールアウト構成 22
スケールアウト構成の例 86
スケールアウトシステム 83
スケールアウトシステムの構成例 83
スケールイン 83, 181
スレッドの非同期並行処理 54

せ

性能解析トレース / 障害解析トレースの出力
55, 58
製品構成 19
製品と構成ソフトウェアの対応 33
セキュリティ管理 55, 58
セッション情報の引き継ぎ 71
セットアップウィザード 62, 85
前提 OS 46

そ

相互スタンバイ構成 57, 59

た

耐障害性 69
タイムアウトの設定 71

て

データベース 49
データベース監査証跡連携 55, 58
データベースとの連携 138
テスト 133, 135
デバッグ 130, 133, 135
デバッグ環境の開始 133, 135
デプロイ 181
展開ディレクトリ形式のアプリケーション
132

と

統合開発環境 8
 統合ユーザ管理 55, 78
 同時実行スレッド数制御 55
 ドメイン一括管理 57, 59
 トラブルシューティング 63
 トレース情報による性能解析 77

は

バックエンド 25, 84
 バッチアプリケーション 181
 バッチアプリケーション開発の流れ 134
 バッチアプリケーション実行環境 3
 バッチサーバ 7, 57, 106, 181

ひ

非同期通信アプリケーション 93

ふ

負荷分散 66
 負荷分散機 181
 複数実行環境の切り替え (UNIX の場合)
 56, 58
 フルガーベージコレクションの発生抑止 70
 フレームワーク 181
 フレームワークとライブラリ 56
 プログラミング 130, 135
 フロントエンド 25, 84
 フロントエンド/バックエンド構成の例
 (CTM を使用する場合) 87
 フロントエンド/バックエンドシステム
 25, 84
 フロントエンド/バックエンドシステムの構
 成例 84
 分散トランザクションの実現 54

ほ

ほかの製品との連携 137
 ホスト単位管理モデルを対象にしたコールド
 スタンバイでの系切り替え 57, 59

め

メッセージ 93
 メッセージの転送 97
 メッセージを利用した非同期通信アプリケー
 ションの実行環境の例 (Cosminexus JMS
 プロバイダを利用する場合) 95
 メッセージを利用した非同期通信アプリケー
 ションの実行環境の例 (Cosminexus RM
 を利用する場合) 96
 メッセージを利用した非同期通信アプリケー
 ションの送信側と受信側に使用できる製品
 97
 メッセージングを利用した SOAP アプリ
 ケーション 125
 メッセージングを利用した SOAP アプリ
 ケーションの開発 55

ゆ

優先制御 66

り

リソースアダプタ 182
 リソースアダプタのデプロイ 56, 58
 リソース枯渇監視 55, 58
 リソース接続とトランザクション管理
 54, 58
 リソースの実行時属性の設定 135
 リソースの設定 56, 58
 リソースマネージャ 182
 リダイレクタ 40, 182
 リモート環境からの各種サーバの設定
 56, 58
 流量制御 66

ろ

ローカルトランザクション 54, 58
 ローカルマシンでのデバッグおよびリモート
 マシンでのデバッグ 56, 58
 ログ/トレース収集 57, 59
 ログの運用 77