

Cosminexus V11 アプリケーションサーバ システム構築・運用ガイド

手引・文法・操作書

3021-3-J02-70

前書き

■ 対象製品

マニュアル「アプリケーションサーバ & BPM/ESB 基盤 概説」の前書きの対象製品の説明を参照してください。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、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、世界の多くの国で登録された International Business Machines Corporation の商標です。

Eclipse は、Eclipse Foundation, Inc.の商標です。

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

Microsoft, Internet Explorer, Microsoft Edge, SQL Server, Windows, Windows Server は、マイクロソフト企業グループの商標です。

Oracle(R), Java, MySQL 及び NetSuite は、Oracle, その子会社及び関連会社の米国及びその他の国における登録商標です。

Red Hat, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries. Linux(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat, および Red Hat Enterprise Linux は、米国およびその他の国における Red Hat, Inc.の登録商標です。Linux(R)は、米国およびその他の国における Linus Torvalds 氏の登録商標です。

UNIX は、The Open Group の登録商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc.により構築された開発ツール統合のためのオープンプラットフォームです。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

■ マイクロソフト製品のスクリーンショットの使用について

マイクロソフトの許可を得て使用しています。

■ 発行

2025年4月 3021-3-J02-70

■ 著作権

All Rights Reserved. Copyright (C) 2020, 2025, Hitachi, Ltd.

变更内容

変更内容(3021-3-J02-70) uCosminexus Application Server 11-60, uCosminexus Client 11-60, uCosminexus Developer 11-60, uCosminexus Service Architect 11-60, uCosminexus Service Platform 11-60

追加・変更内容	変更個所
インストール時の禁止文字について説明を追加した。	2.2.2
マニュアル訂正の内容を反映した。	-

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルをお読みになる際の前提情報については、マニュアル「アプリケーションサーバ & BPM/ESB 基盤 概説」のはじめにの説明を参照してください。

目次

前書き 2

変更内容 4

はじめに 5

1 システム構築・運用の概要 12

1.1 運用管理機能のツールとは 13

1.1.1 セットアップウィザードとは 13

1.1.2 運用管理ポータルとは 15

1.1.3 Smart Composer 機能とは 15

1.2 システム構築の概要 17

1.2.1 単一の Web フロントシステムを構築するには 17

1.2.2 同一構成のホストを複数配置して可用性を確保するには 18

1.2.3 J2EE サーバを複数配置してマルチテナント対応するには 20

1.2.4 リソースアダプタを複数配置してマルチテナント対応するには 21

1.3 システム運用の概要 23

1.4 運用管理ポータルまたは Smart Composer 機能で実現する運用形態の概要 24

1.5 このマニュアルの例題および画面操作で使用している記号 25

1.6 Windows 使用時の注意事項 26

1.6.1 管理者特権で実行する必要がある操作 26

1.6.2 JIS X0213:2004 に含まれる Unicode の補助文字を使用する場合の注意事項 27

2 システムの構築 28

2.1 データベースサーバの構築 29

2.1.1 データベースを準備する 29

2.2 アプリケーションサーバの構築 30

2.2.1 Application Server のインストールについて 30

2.2.2 Application Server を新規インストールする (Windows の場合) 34

2.2.3 Application Server を新規インストールする (AIX の場合) 37

2.2.4 Application Server を新規インストールする (Linux の場合) 39

2.2.5 データベースに接続するための製品をインストールする 44

2.2.6 システム環境をセットアップする (単一の Web フロントシステムを構築する) 44

2.2.7 システム環境をセットアップするときの注意事項 46

2.2.8 システム環境定義の設定とは 49

2.2.9 タイムアウトを設定する 49

2.2.10 業務アプリケーションを設定して開始する (GUI 利用時) 53

2.3	可用性を確保したシステムへの変更 56
2.3.1	同一構成のホストを複数配置して可用性を確保する 56
2.3.2	同一構成のホストを複数配置するときの注意事項 57
2.4	マルチテナントに対応したシステムへの変更 59
2.4.1	J2EE サーバを複数配置してマルチテナントに対応する 59
2.4.2	リソースアダプタを複数配置してマルチテナントに対応する 64

3 システムの運用 70

3.1	アプリケーションサーバの通常運用 71
3.1.1	システムを起動して業務を開始する (GUI 利用時) 71
3.1.2	システムの稼働状況を確認する (GUI 利用時) 72
3.1.3	システムを停止する (GUI 利用時) 73
3.2	アプリケーションサーバの保守運用 75
3.2.1	業務アプリケーションを入れ替える 75
3.2.2	パフォーマンスチューニングとは 78
3.2.3	JavaVM のメモリチューニングを反映する 78
3.2.4	同時実行数を最適化するためのチューニングを反映する 81
3.2.5	IP アドレスを変更する 88
3.2.6	データベース接続環境を変更する 91
3.2.7	システムの構成・定義情報とは 93
3.2.8	システムの構成・定義情報を取得する 94
3.2.9	不要になった J2EE サーバを削除する 96
3.2.10	運用管理機能の管理ユーザアカウントの設定を変更する 98
3.3	システム環境のアンセットアップと Application Server のアンインストール 99
3.3.1	システム環境をアンセットアップする 99
3.3.2	Application Server をアンインストールする (Windows の場合) 99
3.3.3	Application Server をアンインストールする (AIX の場合) 102
3.3.4	Application Server をアンインストールする (Linux の場合) 103
3.4	システムの運用時の注意事項 105
3.4.1	J2EE サーバを運用管理するときの注意事項 105
3.4.2	コマンドで指定する名称に関する注意事項 105
3.4.3	ネットワークドライブへアクセスするときの制限事項 (Windows の場合) 107
3.4.4	一時ファイルに関する注意事項 107
3.4.5	システム時刻を補正するときの注意事項 108

4 運用管理ポータルまたは Smart Composer 機能で実現する運用形態 109

4.1	Web サーバを別ホストに配置してマシン性能を向上するシステムの構築 110
4.1.1	Web サーバを別ホストに配置するには 110
4.1.2	データベースの環境を設定する (Oracle の設定) 114

4.1.3	Component Container 管理者を設定する (UNIX の場合)	115
4.1.4	Component Container 管理者を設定するときの注意事項 (UNIX の場合)	116
4.1.5	アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	120
4.1.6	データベース接続環境を設定する (HiRDB の設定)	121
4.1.7	データベース接続環境を設定する (Oracle の設定)	123
4.1.8	データベース接続環境を設定する (SQL Server の設定)	124
4.1.9	データベース接続環境を設定する (MySQL の設定)	126
4.1.10	データベース接続環境を設定する (PostgreSQL の設定)	127
4.1.11	データベースを設定するときの注意事項	128
4.1.12	環境変数とは	128
4.1.13	システムの環境変数を設定する	129
4.1.14	論理サーバの環境変数を設定する	129
4.1.15	論理サーバの環境変数を設定するときの確認事項	130
4.1.16	運用管理機能を構築する	134
4.1.17	Management Server を使用するための設定情報	135
4.1.18	運用管理エージェントを使用するための設定情報	138
4.1.19	運用管理エージェントを起動する	141
4.1.20	Management Server を起動する	141
4.1.21	簡易構築定義ファイルとは (J2EE アプリケーション使用時)	141
4.1.22	簡易構築定義ファイルを設定する	145
4.1.23	簡易構築定義ファイルを設定するときの注意事項 (J2EE アプリケーション使用時)	146
4.1.24	簡易構築定義ファイルの作成例	149
4.1.25	システム環境を構築する	151
4.1.26	システムを起動する (CUI 利用時)	151
4.1.27	リソースアダプタの設定とは	152
4.1.28	DB Connector を設定する (CUI 利用時)	152
4.1.29	DB Connector 以外のリソースアダプタを設定する (CUI 利用時)	154
4.1.30	リソースアダプタを開始する (CUI 利用時)	155
4.1.31	業務アプリケーションを設定して開始する (CUI 利用時)	156
4.1.32	リソースアダプタまたは業務アプリケーションを操作するときの確認事項 (CUI 利用時)	157
4.1.33	システム環境定義 (簡易構築定義ファイルのパラメタ) を変更する (CUI 利用時)	158
4.1.34	業務アプリケーションの設定を変更する (CUI 利用時)	160
4.1.35	リソースアダプタの設定 (データベースの接続環境) を変更する (CUI 利用時)	161
4.1.36	システムを停止する (CUI 利用時)	163
4.1.37	Management Server を停止する	163
4.1.38	運用管理エージェントを停止する	163
4.1.39	システム環境を削除する	164
4.2	ホストを増やしてシステムの規模を拡大するシステムの運用 (スケールアウト)	167
4.2.1	ホストを増やしてシステムの規模を拡大 (スケールアウト) するには	167

4.2.2	システム環境をスケールアウトする (CUI 利用時)	169
4.3	ホストを減らしてシステムの規模を縮小するシステムの運用 (スケールイン)	173
4.3.1	ホストを減らしてシステムの規模を縮小 (スケールイン) するには	173
4.3.2	システム環境をスケールインする (CUI 利用時)	174
4.4	構成・定義情報を出力してほかのホストにシステム環境を移行するシステムの運用	176
4.4.1	ほかのホストにシステム環境を移行するには	176
4.4.2	構成・定義情報を出力してシステム環境を移行する	177
4.4.3	移行時に使用する構成・定義情報の内容	179
4.4.4	構成・定義情報を出力してシステムを移行するときの注意事項	182
4.5	論理ユーザサーバを配置するシステムの構築	184
4.5.1	論理ユーザサーバを配置するには	184
4.5.2	論理ユーザサーバを簡易構築定義ファイルで設定する (CUI 利用時)	188
4.6	バッチアプリケーションを実行するシステムの構築	190
4.6.1	バッチアプリケーションを実行するには	190
4.6.2	簡易構築定義ファイルを設定するときの注意事項 (バッチアプリケーション使用時)	194
4.6.3	バッチアプリケーションを実行するシステムを運用するときの注意事項	199
4.7	運用管理機能から負荷分散機を制御するシステムの構築	200
4.7.1	運用管理機能から負荷分散機を制御するには	200
4.7.2	負荷分散機とは	202
4.7.3	負荷分散機を準備する	204
4.7.4	接続方法に API を使用するときの注意事項	205
4.7.5	負荷分散機へ接続する環境を設定する	205
4.7.6	負荷分散機を制御するときの注意事項	206
4.8	その他の機能を使用したシステムの構築	208
4.8.1	その他の機能を使用するには	208
4.8.2	簡易構築定義ファイルを設定するときの注意事項 (その他の機能使用時)	212
4.8.3	WebSocket を使用する場合の設定	217
4.8.4	HWSGracefulStopTimeout ディレクティブと Manager の停止監視時間の設定に関する注意点	218
4.9	運用上の注意事項	220

5 システムの構築・運用時に使用する画面、コマンドおよびファイル 221

5.1	システムの構築・運用時に使用する画面	222
5.2	システムの構築・運用時に使用するコマンド	225
5.3	システムの構築・運用時に使用するファイル	229

付録 232

付録 A	セットアップウィザードで設定されるパラメタ	233
付録 A.1	セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ (論理 Web サーバ)	233

付録 A.2	セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ（論理 J2EE サーバ）	234
付録 A.3	セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ（論理パフォーマンストレーサ）	236
付録 B	インストール後のディレクトリ構成	237
付録 B.1	Application Server のディレクトリ構成	237
付録 B.2	Component Container のディレクトリ構成	240
付録 C	作業ディレクトリ	247
付録 C.1	J2EE サーバの作業ディレクトリ	247
付録 C.2	バッチサーバの作業ディレクトリ	254
付録 C.3	作業ディレクトリに関する注意事項	256
付録 D	運用管理ポータルの操作で失敗したときの対処方法	258
付録 E	Management Server が自動で設定する内容	260
付録 E.1	Management Server が自動で生成するユーザ定義ファイル	260
付録 E.2	Management Server が CTM ドメインマネージャに自動で設定する内容	262
付録 E.3	Management Server が CTM に自動で設定する内容	263
付録 E.4	Management Server が J2EE サーバに自動で設定する内容	263
付録 E.5	Management Server がバッチサーバに自動で設定する内容	264
付録 E.6	Management Server が Web サーバに自動で設定する内容	265
付録 F	Application Server の更新インストール	267
付録 F.1	Application Server を更新インストールする（Windows の場合）	267
付録 F.2	Application Server を更新インストールする（UNIX の場合）	268
付録 G	JP1/SC/DPM を使ったディスク複製インストール	271
付録 G.1	Application Server をディスク複製インストールする（Windows の場合）	271
付録 G.2	Application Server をディスク複製インストールする（Linux の場合）	273
付録 H	リモートインストール	275
付録 H.1	Application Server をリモートインストールする（Windows の場合）	275
付録 H.2	Application Server をリモートインストールする（UNIX の場合）	275
付録 I	インストールおよびアンインストールするときの注意事項	276
付録 I.1	Application Server をインストールおよびアンインストールするときの注意事項	276
付録 I.2	構成ソフトウェアをインストールおよびアンインストールするときの注意事項	281
付録 I.3	ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項（Windows の場合）	288
付録 I.4	ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項（Linux の場合）	299
付録 I.5	リモートインストールをするときの注意事項	310
付録 I.6	Application Server のインストールおよびアンインストール時のエラーコード一覧（Windows の場合）	311
付録 I.7	Component Container およびリダイレクタ機能のインストールおよびアンインストール時のログ（UNIX の場合）	322
付録 J	Smart Composer 機能が実行する負荷分散機の CLI コマンドまたは API	328

付録 J.1	Smart Composer 機能が実行する負荷分散機の API	328
付録 J.2	Smart Composer 機能が実行する負荷分散機の CLI コマンド	328
付録 K	XDM/RD E2 の設定	335
付録 K.1	環境変数の設定	335
付録 K.2	環境変数のグループ登録	335
付録 K.3	タイムアウトの設定	336
付録 L	アプリケーションサーバが提供するサンプルプログラム	338
付録 L.1	サーバ管理コマンドによるサンプル実行環境の構築、運用手順	339
付録 L.2	Enterprise Bean (EJB2.0) のサンプルプログラム	341
付録 L.3	Enterprise Bean (アノテーション) のサンプルプログラム	360
付録 L.4	Enterprise Bean (Timer Service) のサンプルプログラム	365
付録 L.5	JavaBeans リソースのサンプルプログラム	367
付録 L.6	ユーザログ機能のサンプルプログラム	370
付録 L.7	Web アプリケーションのサンプルプログラム	378
付録 L.8	TP1 インバウンド連携機能のサンプルプログラム	382
付録 L.9	JSF のサンプルプログラム	386
付録 L.10	Bean Validation および CDI 連携アプリケーションのサンプル	388
付録 M	用語解説	392

索引 393

1

システム構築・運用の概要

アプリケーションサーバでは、オンライン業務およびバッチ業務のアプリケーションが動作するシステムを構築できます。システムを構築する際には、アプリケーションサーバの運用管理機能で提供するツールが使用できます。

この章では、運用管理機能のツールの概要、システムの構築・運用の概要について説明します。ツールを使用しない場合、サーバ管理コマンドや手動に置き換えて参照してください。

1.1 運用管理機能のツールとは

アプリケーションサーバの運用管理機能では、次のツールを提供しています。これらのツールは、使用したい手段（画面、コマンドなど）や、構築したいシステム構成によって選択できます。また、これらのツールは同じ運用管理機能で動作するので、作業のタイミング（構築、運用）や、設定する内容によって使用するツールを使い分けることができます。

- **セットアップウィザード**

構築時に使用するツールです。対話型ウィザードで表示される画面に従って、項目を選択または設定するとシステムが構築できます。

- **運用管理ポータル**

構築時、運用時に使用するツールです。GUI画面の画面項目を選択、またはテキストで値を入力するとシステムが構築できます。

- **Smart Composer 機能**

構築時、運用時に使用するツールです。ファイルでパラメタを設定し、コマンドでファイルの内容を反映するとシステムが構築できます。

オンライン業務を実行する、WebサーバとJ2EEサーバを配置するシステムを構築する場合は、システムの構築時にはセットアップウィザード、運用時には運用管理ポータルを使用します。セットアップウィザードを使用すると、製品をインストールしたあとは、対話形式で作業を進めるだけで、アプリケーションサーバがデフォルトで提供する機能を設定したシステムをスムーズに構築できます。運用管理ポータルを使用すると、GUI画面の表示項目やボタンをクリック操作するだけで、システム全体を一括で起動、停止したり、システムの稼働状況を確認したりする運用ができます。詳細設計が済んだシステムを構築する場合は、Smart Composer機能を使用します。

ただし、次の要件を満たす場合は、セットアップウィザードでは構築できません。この場合、運用管理ポータルまたはSmart Composer機能を使用してシステムを構築してください。

- バッチ業務のアプリケーション（バッチアプリケーション）を実行するシステムを構築したい

1.1.1 セットアップウィザードとは

セットアップウィザードは、ウィザードに従って対話形式で操作を進めることで、J2EEアプリケーションの実行環境を構築できるツールです。必要な設定を画面で選択しながら作業を進められるので、事前に複雑なパラメタの検討や定義ファイルの編集をしないで済みます。J2EEアプリケーションが動作するWebフロントシステム（WebサーバとJ2EEサーバを配置するシステム）を構築するのに適しています。

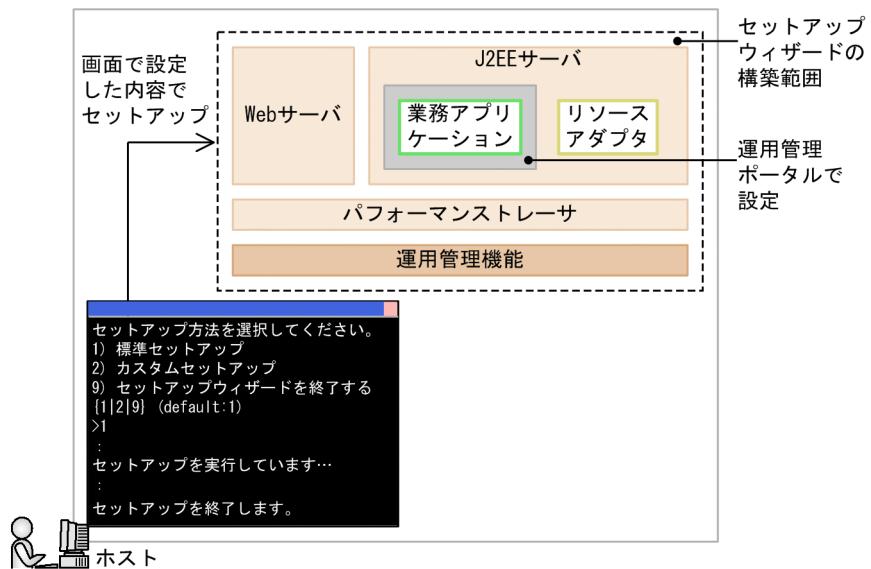
アプリケーションサーバのバージョンが11-00以降の場合、セットアップウィザードではJ2EEサーバを構築する際、互換モードを使用することはできません。

また、J2EEアプリケーションを動作させる環境に最小限必要な項目だけを最小限の手順で設定して構築できるので、製品を導入してからシステムを起動するまでに掛かる時間を短縮できます。構築したシステム

はチューニングもできます。まずは最小限の設定でシステムの運用を開始して、稼働状況に応じてシステム設定を最適化したり、システム構成を拡大したりするといった運用ができます。

セットアップウィザードによるシステム構築のイメージを次の図に示します。

図 1-1 セットアップウィザードによるシステム構築のイメージ



セットアップウィザードの画面に従って操作すると、次の作業が一括で実行できます。

1. システムの運用に必要なプロセス（運用管理機能）の構築
2. システムの実行に必要なプロセス（Web サーバ、J2EE サーバおよびパフォーマンストレーサ）の構築
3. データベースとの接続に必要なリソースアダプタの設定
4. 運用管理機能、ならびに Web サーバ、J2EE サーバおよびパフォーマンストレーサの起動

なお、業務アプリケーションの設定および開始は、セットアップウィザード実行後に運用管理ポータルを使用して実行します。

セットアップウィザードで提供するセットアップ方法には、次のものがあります。

- **標準セットアップ**

すべての設定項目をデフォルト値で構築するセットアップ方法です。データベースに接続するシステムも、使用するデータベースの設定以外はすべてデフォルト値で構築できます。Web サーバと J2EE サーバで構成されるシステムをセットアップできます。

- **カスタムセットアップ**

運用管理機能を使用するユーザ、各サーバで使用するポート番号などを任意の値に変更できるセットアップ方法です。設定値を変更しない項目は、デフォルト値のままセットアップできます。Web サーバと J2EE サーバを配置するシステムと、J2EE サーバだけを配置するシステムをセットアップできます。

セットアップウィザードのデフォルト値については、「[付録 A セットアップウィザードで設定されるパラメタ](#)」を参照してください。

1.1.2 運用管理ポータルとは

運用管理ポータルは、Web ブラウザ上に表示される GUI 画面を操作するだけで、J2EE アプリケーションまたはバッチアプリケーションの実行環境を構築・運用できるツールです。セットアップウィザードで構築したシステムのアプリケーションを登録して開始したり、システムを起動、停止したりといった運用ができます。システムの起動や停止のほかに、設定済みのサーバの情報を読み込んでほかのサーバに設定したり、ホストやサーバの単位でシステムのチューニングや構成変更をしたり、ホストやサーバの稼働状況を監視したりするといった運用ができます。

また、システム構築に必要な設定項目は画面に表示されるので、どのような設定があるのかを、すべて把握していくなくても使えます。セットアップウィザードでは構築できない、ホストごとにプロセスを柔軟に組み合わせるシステムを構築するのに適しています。

運用管理ポータルによるシステム構築のイメージを次の図に示します。

図 1-2 運用管理ポータルによるシステム構築のイメージ



運用管理ポータルの GUI 画面では、ホストやサーバの構成を定義し、各サーバの環境を設定して設定情報をホストに配布すると、システムが構築できます。データベースとの接続に必要なリソースアダプタや業務アプリケーションも設定して起動できます。なお、運用管理ポータルを使用する前には、運用管理機能の構築が必要です。

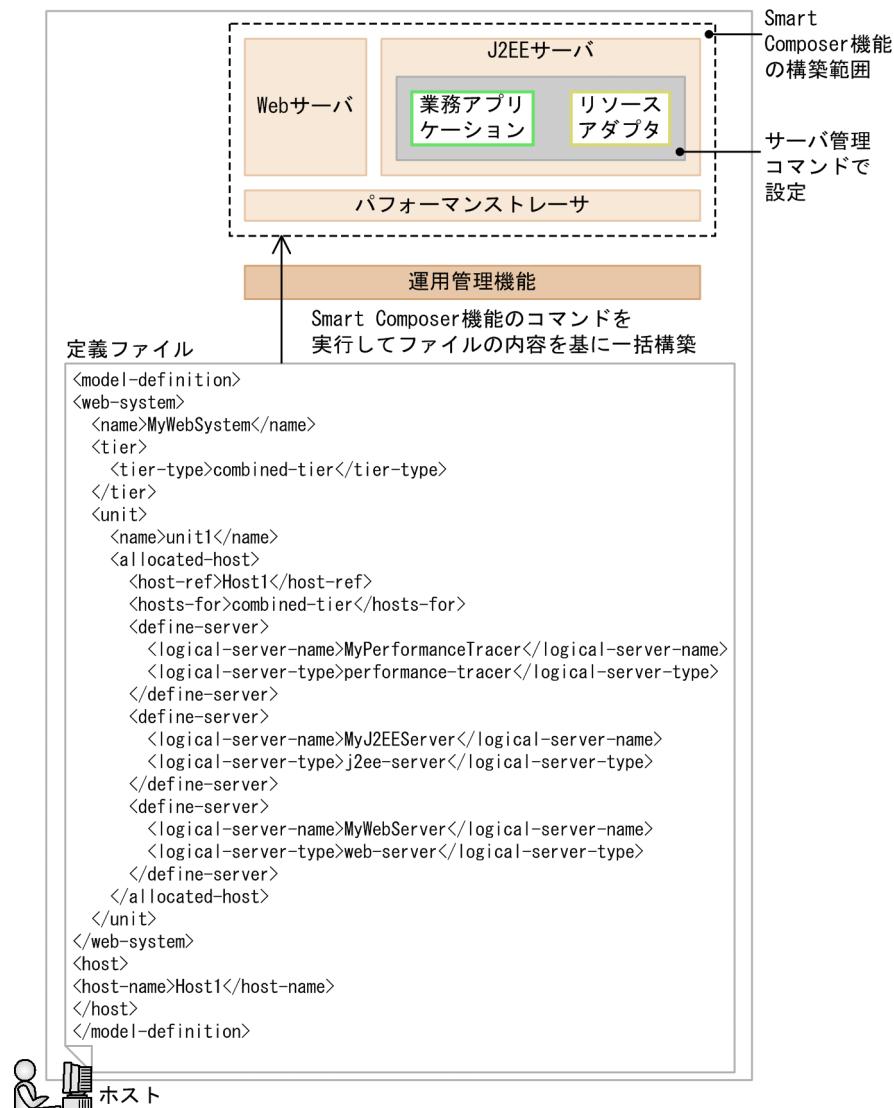
1.1.3 Smart Composer 機能とは

Smart Composer 機能は、複数のホストやサーバの構成、環境設定をファイルに定義してコマンドを実行するだけで、J2EE アプリケーションまたはバッチアプリケーションの実行環境を一括構築・運用できるツールです。Smart Composer 機能で使用する XML 形式のファイルは、あらかじめテンプレートが提供されています。さらに詳細設計で決めた設定を追加してシステムを構築できます。同じプロセスや設定を持つパターン化された構成のシステムを構築するのに適しています。

また、ファイルを使用して構築するので、ホストやサーバの数だけ同じ設定操作を繰り返す必要がありません。構築済みのシステムの設定情報をエビデンスとして残すこともできます。システムの起動や停止、一括設定変更、拡大（スケールアウト）、縮小（スケールイン）などを各サーバをまとめた単位で一括操作できるので、一部の業務だけを停止したり、業務を実施しながらシステムを拡大（スケールアウト）したりするといった運用ができます。

Smart Composer 機能によるシステム構築のイメージを次の図に示します。

図 1-3 Smart Composer 機能によるシステム構築のイメージ



Smart Composer 機能の定義ファイルで、ホストやサーバの構成、各サーバの環境設定を定義します。その定義ファイルを入力元として Smart Composer 機能のコマンドを実行すると、システムが構築できます。データベースとの接続に必要なリソースアダプタや業務アプリケーションは、Smart Composer 機能でシステムを構築したあとに、ほかのコマンド（サーバ管理コマンド）で設定します。なお、Smart Composer 機能を使用する前には、運用管理機能の構築が必要です。

1.2 システム構築の概要

ここでは、セットアップウィザードを使用して構築するシステムの概要について説明します。

システム構築では、まずは単一の Web フロントシステム（同一ホストに Web サーバと J2EE サーバを配置するシステム）を構築します。次に、そのシステムに対して設定変更や構成変更を必要に応じて実施して、システム要件を満たすシステムを完成させます。

システムの構築手順を次に示します。なお、手順 1.を実施するだけでも運用を開始できます。手順 2 および手順 3.は、システム要件に合わせて実施してください。

1. 単一の Web フロントシステムを構築します。 (1.2.1)

必要に応じて、システムの性能や信頼性などを満たすために、システム環境定義を設定または変更します。

2. 可用性を確保した Web フロントシステムに構成を変更します。 (1.2.2)

同一構成のホストを複数配置して可用性を確保します。

3. マルチテナントに対応した Web フロントシステムに構成を変更します。 (1.2.3 または 1.2.4)

J2EE サーバまたはリソースアダプタを複数配置してマルチテナントに対応します。

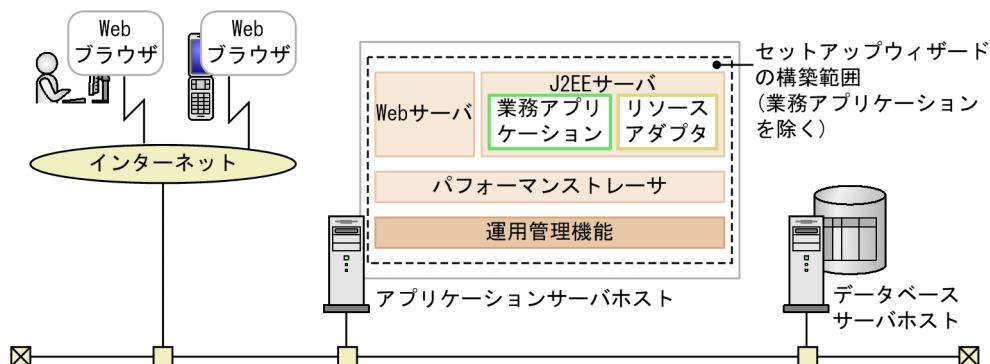
1.2.1 単一の Web フロントシステムを構築するには

単一の Web フロントシステムは、同一ホストに Web サーバと J2EE サーバを配置するシステムです。このシステムは、セットアップウィザードに従って操作すると構築できます。

- システム構成例

単一の Web フロントシステムの構成例を次の図に示します。

図 1-4 単一の Web フロントシステムの構成例



- 実施する作業と作業の要否

単一の Web フロントシステムの構築時に実施する作業と作業の要否を次の表に示します。

表 1-1 単一の Web フロントシステムの構築時に実施する作業と作業の要否

単一の Web フロントシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
データベースの準備	○※	データベースに接続する場合	2.1.1
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	○※	データベースに接続する場合	2.2.5
システム環境のセットアップ	○	必ず実施	2.2.6
タイムアウトの設定	△	システム要件に合わせたシステム環境定義に変更する場合	2.2.9
業務アプリケーションの設定と開始	○	必ず実施	2.2.10

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注※ このマニュアルで説明するシステム構成例では、データベースに接続することを前提としています。このため、この作業は必ず実施します。

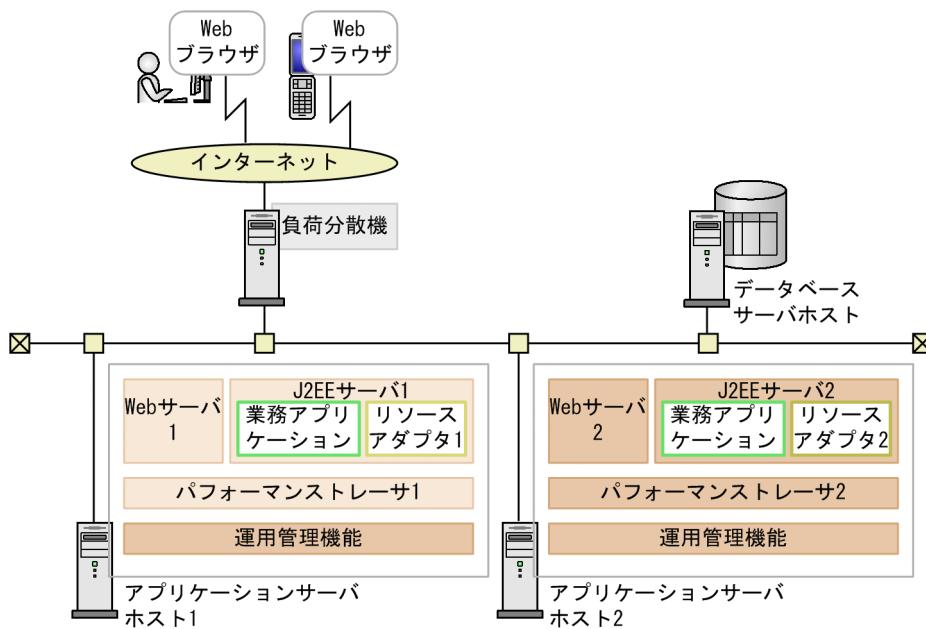
1.2.2 同一構成のホストを複数配置して可用性を確保するには

同一構成のホストを複数配置すると、負荷分散機を利用することで、障害が発生したホストだけを停止して縮退運用できるので、システム全体の可用性を確保できます。同一構成のホストを複数配置して可用性を確保した Web フロントシステムは、単一の Web フロントシステムをベースに、負荷分散機を配置し、同じ構成のホストを複数して追加すると構築できます。

- ・ システム構成例

同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構成例を次の図に示します。

図 1-5 同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構成例



- 実施する作業と作業の要否

同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構築時に実施する作業と作業の要否を次の表に示します。

表 1-2 同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構築時に実施する作業と作業の要否

同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照箇所
データベースの準備	○※	データベースに接続する場合	2.1.1
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	○※	データベースに接続する場合	2.2.5
システム環境のセットアップ	○	必ず実施	2.2.6
タイムアウトの設定	△	システム要件に合わせたシステム環境定義に変更する場合	2.2.9
業務アプリケーションの設定と開始	○	必ず実施	2.2.10
可用性を確保するための設定	○	必ず実施	2.3.1

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注※ このマニュアルで説明するシステム構成例では、データベースに接続することを前提としています。このため、この作業は必ず実施します。

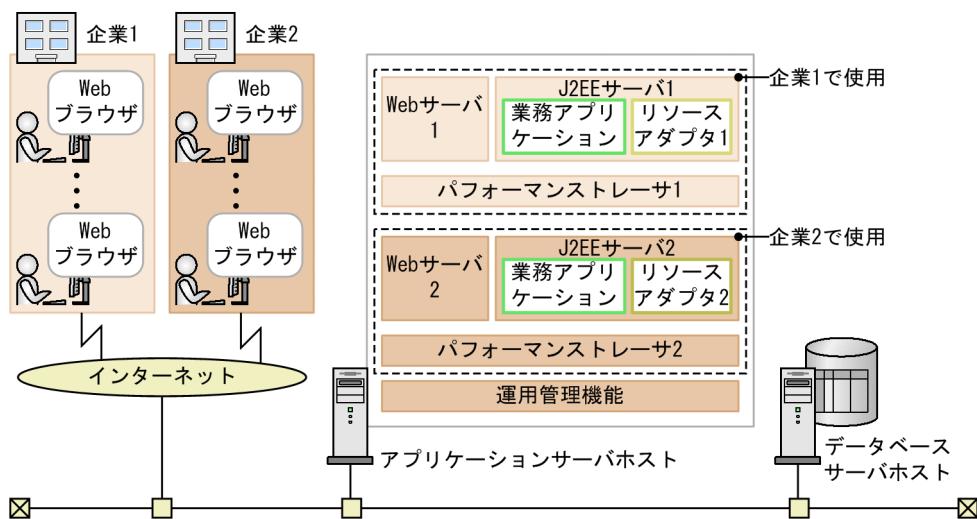
1.2.3 J2EE サーバを複数配置してマルチテナント対応するには

J2EE サーバを複数配置すると、ホストのリソースを有効活用して、かつテナントごとに業務処理（アプリケーション）を分離させたマルチテナントの構成に対応できます。J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムは、単一の Web フロントシステムをベースに、J2EE サーバを含む論理サーバのまとまりをホストに追加すると構築できます。

- システム構成例

J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構成例を次の図に示します。

図 1-6 J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構成例



- 実施する作業と作業の要否

J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否を次の表に示します。

表 1-3 J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否

J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照箇所
データベースの準備	<input type="radio"/> ※	データベースに接続する場合	2.1.1
Application Server のインストール	<input type="radio"/>	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	<input type="radio"/> ※	データベースに接続する場合	2.2.5
システム環境のセットアップ	<input type="radio"/>	必ず実施	2.2.6

J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
タイムアウトの設定	△	システム要件に合わせたシステム環境定義に変更する場合	2.2.9
業務アプリケーションの設定と開始	○	必ず実施	2.2.10
J2EE サーバの複数配置	○	必ず実施	2.4.1

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注※ このマニュアルで説明するシステム構成例では、データベースに接続することを前提としています。このため、この作業は必ず実施します。

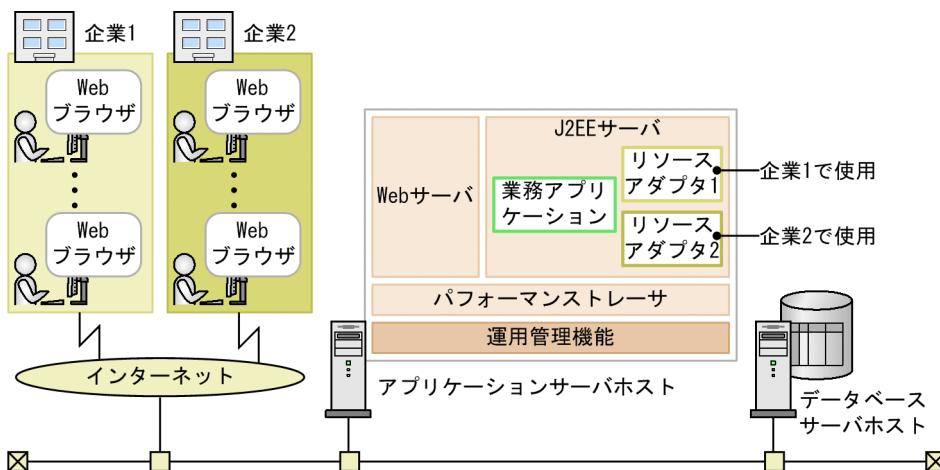
1.2.4 リソースアダプタを複数配置してマルチテナント対応するには

リソースアダプタを複数配置すると、データベースを共有（データベース数の増加を抑止）して、かつテナントごとにデータを分離させたいマルチテナントの構成に対応できます。リソースアダプタを複数配置してマルチテナント対応する Web フロントシステムは、リソースアダプタを J2EE サーバに追加し、業務アプリケーションからリソースアダプタを参照するようにすると構築できます。

- システム構成例

リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構成例を次の図に示します。

図 1-7 リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構成例



- 実施する作業と作業の要否

リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否を次の表に示します。

表 1-4 リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否

リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
データベースの準備	○※	データベースに接続する場合	2.1.1
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	○※	データベースに接続する場合	2.2.5
システム環境のセットアップ	○	必ず実施	2.2.6
タイムアウトの設定	△	システム要件に合わせたシステム環境定義に変更する場合	2.2.9
業務アプリケーションの設定と開始	○	必ず実施	2.2.10
リソースアダプタの複数配置	○	必ず実施	2.4.2

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注※ このマニュアルで説明するシステム構成例では、データベースに接続することを前提としています。このため、この作業は必ず実施します。

1.3 システム運用の概要

運用を開始したシステムでは、業務アプリケーションの入れ替え、稼働統計情報を利用したパフォーマンスチューニングなど、システムの保守を目的とした運用が発生します。運用作業は、実施するタイミングや内容によって、通常運用、保守運用、障害運用に分類できます。これらの作業では、運用管理ポータル(GUI) または Smart Composer 機能 (CUI) のどちらかのツールを使用します。

●通常運用とは

アプリケーションサーバで実施する通常運用では、運用開始後に日々実施する、次の作業が該当します。システムの起動・停止に加えて、稼働状況の確認は、システムを構成する論理サーバのステータスや稼働状況を適宜確認して、安定した稼働状態を保っていくために必要な作業になります。

- ・システムの起動（業務の開始）
- ・システムの稼働状況の確認
- ・システムの停止（業務の終了）

通常運用の手順については、「[3.1 アプリケーションサーバの通常運用](#)」を参照してください。

●保守運用とは

アプリケーションサーバで実施する保守運用では、運用状況に合わせて実施する、次の作業が該当します。サービスの提供方法やシステムのメンテナンスに応じて業務アプリケーションの設定を変更して入れ替えたり、稼働状況の資料を分析して動作環境を最適化したり、運用規模を調整したりするなど、運用状況によって必要な作業になります。

- ・業務アプリケーションの入れ替え
- ・パフォーマンスチューニング
- ・IP アドレスの変更
- ・データベース接続環境の変更
- ・システムの構成・定義情報の取得
- ・不要な J2EE サーバの削除
- ・運用管理機能の管理ユーザアカウントの設定変更

保守運用の手順については、「[3.2 アプリケーションサーバの保守運用](#)」を参照してください。

●障害運用とは

アプリケーションサーバでの障害発生時には、障害情報を取得して、業務を再開するためにシステムを回復する作業が必要になります。これを障害運用といいます。運用を再開したあとは、障害時に取得した障害情報を基に、障害の原因を解析する作業も実施します。アプリケーションサーバの機能では、プロセスがダウンしたり、ハングアップ（無応答）したりすると、プロセスを自動再起動して回復する運用も提供しています。障害運用の事前設定や障害情報の取得などについては、マニュアル「[アプリケーションサーバ 機能解説 保守／移行編](#)」を参照してください。

1.4 運用管理ポータルまたは Smart Composer 機能で実現する運用形態の概要

ここでは、運用管理ポータルまたは Smart Composer 機能で構築することをお勧めするシステムの構成について説明します。

運用管理ポータルまたは Smart Composer 機能を使用すると、次の表に示す運用形態のシステムを構築できます。

表 1-5 運用管理ポータルまたは Smart Composer 機能で実現するシステムの運用形態

項番	システムの運用形態	参照先
1	Web サーバを別ホストに配置してマシン性能を向上する運用形態	4.1
2	ホストを増やしてシステムの規模を拡大する運用形態（スケールアウト）	4.2
3	ホストを減らしてシステムの規模を縮小する運用形態（スケールイン）	4.3
4	構成・定義情報を出力してほかのホストにシステム環境を移行する運用形態	4.4
5	論理ユーザサーバを利用する運用形態	4.5
6	バッチアプリケーションを実行する環境を利用する運用形態	4.6
7	運用管理機能から負荷分散機を制御する運用形態	4.7

このほかにも、アプリケーションサーバの機能を利用した、次のようなシステムが構築できます。

- 運用管理サーバモデル（運用管理機能を別ホストに配置する）
- CTM によるスケジューリング機能
- MQ モデル（TP1/Message Queue 連携）
- リバースプロキシ

その他の機能を利用する場合は、使用する機能に合わせたシステム構成の検討、パラメタ設計などの詳細設計をしてから、システムを構築します。この場合も運用管理ポータルまたは Smart Composer 機能で構築します。その他の機能を利用したシステムについては、「[4.8 その他の機能を使用したシステムの構築](#)」を参照してください。なお、その他の機能については、次に示す機能解説のマニュアルを参照してください。

- マニュアル「[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「[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「[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「[アプリケーションサーバ 機能解説 拡張編](#)」
- マニュアル「[アプリケーションサーバ 機能解説 セキュリティ管理機能編](#)」
- マニュアル「[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「[アプリケーションサーバ 機能解説 保守／移行編](#)」

1.5 このマニュアルの例題および画面操作で使用している記号

このマニュアルの例題および画面操作で使用している記号について説明します。

- 定義例およびコマンドの実行例で使用している記号

このマニュアルの定義例およびコマンドの実行例で使用している記号について次の表に示します。

表 1-6 定義例およびコマンドの実行例で使用している記号

記号	意味
< >	この記号で囲まれている項目は、該当する要素やファイルなどを指定したり、該当する要素が表示されたりすることを示します。 (例 1) <プロパティ> プロパティを記述します。またはプロパティが表示されます。 (例 2) <ファイル名> ファイル名を指定します。

- 画面操作の説明で使用している記号

このマニュアルの画面操作 (CUI および GUI) の説明で使用している記号について次に示します。

表 1-7 画面操作の説明で使用している記号

記号	意味
[]	この記号で囲まれている項目は画面名または画面に表示されている項目であることを表します。
< >	< >内の名称がユーザの環境によって異なることを表します。

1.6 Windows 使用時の注意事項

ここでは、Windows 環境で、アプリケーションサーバを使用したシステムを構築、運用する場合の注意事項について説明します。

1.6.1 管理者特権で実行する必要がある操作

ここでは、Windows 環境で、管理者特権で実行する必要がある操作について説明します。なお、システムドライブを C ドライブとして説明します。

- プログラムのインストール

アプリケーションサーバの製品や構成ソフトウェア、Eclipse などのプログラムを、C:¥Program Files 以下のディレクトリにインストールする場合、管理者特権で実行する必要があります。インストール時に、[UAC] ダイアログが表示される場合は、[UAC] ダイアログで管理者アカウントのパスワードを入力してください。

- アプリケーションサーバが提供するコマンドの使用

アプリケーションサーバが提供するコマンドは、管理者特権で実行する必要があります。アプリケーションサーバが提供するコマンドは、「管理者：コマンドプロンプト」で実行してください。

「管理者：コマンドプロンプト」は、Windows で提供されている機能を使用して起動してください。

- アプリケーションサーバが提供する定義ファイルの更新

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

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

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

- Eclipse での操作

C:¥Program Files 以下のディレクトリまたは C:¥windows 以下のディレクトリに対する操作には制限があります。これらの制限されたディレクトリに対して、Eclipse のワークスペースを保存すると、リダイレクション機能によって、操作対象のファイルはリダイレクトされ、次の場所に格納されます。

```
C:¥Users¥<ユーザ名>¥AppData¥Local¥VirtualStore¥Program Files
```

保存したファイルの実体は上記の場所に格納されていますが、Eclipse 上からは C:¥Program Files 以下に実ファイルがあるかのように、ファイルを読み込んだり、書き込んだりできます。

- サンプルプログラムの使用

アプリケーションサーバが提供するサンプルプログラムを実行する場合は、書き込み権限のある別のディレクトリへサンプルプログラムをコピーして使用することをお勧めします。

アプリケーションサーバの製品をデフォルトでインストールした場合、サンプルプログラムは、C:\Program Files 以下のディレクトリに格納されます。管理者特権のないユーザがサンプルプログラムを実行すると、サンプルプログラムで出力されるログファイルなどが、次に示すディレクトリ以下に保存されます。

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

このため、アプリケーションサーバの製品をデフォルトでインストールしたディレクトリのまま、サンプルプログラムを使用する場合は、管理者特権で実行してください。サンプルプログラムを管理者特権で使用する場合は、「管理者：コマンドプロンプト」で実行する必要があります。エクスプローラからバッチファイルを右クリックして、[管理者として実行] をクリックした場合、正しく動作しません。なお、アプリケーションサーバが提供するサンプルプログラムについては、「[付録 L アプリケーションサーバが提供するサンプルプログラム](#)」を参照してください。

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

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

Unicode の補助文字を使用する場合の注意事項を次に示します。

- リクエストで使用する場合の注意事項

アプリケーションサーバに対して、Microsoft Edge などのクライアントから Unicode の補助文字を含むリクエストを送信した場合、Unicode の補助文字は、ログや PRF トレースに正しく出力されません。ただし、その場合も、Unicode の補助文字以外の文字は、ログや PRF トレースに正しく出力されます。

また、リクエストに Unicode の補助文字が含まれる場合も、アプリケーションは正しく動作します。リクエストでの Unicode の補助文字の使用を制限したい場合には、アプリケーションでの対応などを検討してください。

- 環境構築／運用時の注意事項

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

Unicode の補助文字を使用できない定義の例を示します。

- EAR, WAR, JAR, EJB-JAR, サーブレット, JSP, クラス, メソッド, 引数、または変数の名称
- DD 内の各種定義
- システムのインストール先の指定値
- そのほか、各種定義ファイルの設定値

2

システムの構築

この章では、単一の Web フロントシステムの構築手順について説明します。さらに、単一の Web フロントシステムから、可用性を確保したシステムやマルチテナントに対応したシステムへ変更する手順についても説明します。

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

2.1 データベースサーバの構築

J2EE サーバでリソース（データベース）を使用するために、データベースサーバを構築します。ここでは、データベースサーバで実施する作業について説明します。

2.1.1 データベースを準備する

データベースサーバに接続して、データベース製品をインストールします。

セットアップウィザードでセットアップするシステムから接続できるデータベースは、次のとおりです。使用する環境に合わせて、どちらかの製品をインストールしてください。各製品のインストール方法については、それぞれの製品のマニュアルを参照してください。

- HiRDB (HiRDB/Parallel Server Version 9 以降および HiRDB/Single Server Version 9 以降)
インストール方法およびデータベースサーバの初期設定については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
- Oracle (Oracle 11g R2 以降)
インストール方法およびデータベースサーバの初期設定については、Oracle のマニュアルを参照してください。

アプリケーションサーバの構築時に、データベース接続環境を設定する際には、次の情報を指定します。データベースサーバの構築時にこれらの情報を控えておいてください。

HiRDB の場合

- データベースのホスト名
- データベースのポート番号
- データベースのアクセスユーザ
- アクセスユーザのパスワード

Oracle の場合

- データベース名 (SID)
- データベースのホスト名
- データベースのポート番号
- データベースのアクセスユーザ
- アクセスユーザのパスワード

2.2 アプリケーションサーバの構築

ここでは、「[1.2.1 単一の Web フロントシステムを構築するには](#)」で説明した、単一の Web フロントシステム（Web サーバおよび J2EE サーバを同じホストに配置するシステム）のアプリケーションサーバを構築する手順について説明します。なお、例題で使用する J2EE アプリケーションは、ファイル名を「MyApp1.ear」、表示名を「MyApp」とします。

2.2.1 Application Server のインストールについて

ここでは、Application Server のインストール方法の種類について説明します。また、アプリケーションサーバ製品に含まれる構成ソフトウェアの一つである、Component Container をインストールしたときのユーザ環境の扱いや、製品に対応する構成ソフトウェアの依存関係についても説明します。

(1) インストール方法の種類

Application Server をインストールする方法には、次の種類があります。

- 新規インストール

Application Server がインストールされていない環境に、Application Server をインストールすることです。インストールする構成ソフトウェアは、「標準」、「カスタム」から選択できます。「標準」を選択すると、Application Server で推奨する構成ソフトウェアをすべてインストールします。「カスタム」を選択すると、Application Server に対応する構成ソフトウェアのうち、選択した構成ソフトウェアをインストールします。詳細は、「[2.2.2 Application Server を新規インストールする \(Windows の場合\)](#)」、「[2.2.3 Application Server を新規インストールする \(AIX の場合\)](#)」、または「[2.2.4 Application Server を新規インストールする \(Linux の場合\)](#)」を参照してください。

- 更新インストール

Application Server がインストール済みの環境に、Application Server（同一製品）をインストールすることです。インストールする構成ソフトウェアは、「標準」、「カスタム」から選択できます。「標準」を選択すると、Application Server で推奨する構成ソフトウェアをすべて再インストールします。「カスタム」を選択すると、Application Server に対応する構成ソフトウェアのうち、選択した構成ソフトウェアを再インストールします。詳細は、「[付録 F Application Server の更新インストール](#)」を参照してください。

更新インストールを行う場合、インストール時の Java バージョンの環境の選択では、すでにインストールされている Cosminexus Developer's Kit for Java (TM) と同じ Java バージョンを選択してください。すでにインストールされている Cosminexus Developer's Kit for Java (TM) と異なる Java バージョンを選択した場合は、更新インストールを行うことができません。

そのほかのインストール方法として、次のものがあります。

- JP1/SC/DPM を使ったディスク複製インストール

JP1/SC/DPM を使用して作成したディスクイメージの複製をインストールすることです。詳細は、「[付録 G JP1/SC/DPM を使ったディスク複製インストール](#)」を参照してください。

- **リモートインストール**

JP1/NETM/DM や Groupmax Remote Installation を使用し、ネットワークを介してソフトウェアをクライアントへ配布するインストールのことです。詳細は、「[付録 H リモートインストール](#)」を参照してください。

なお、例題で説明するアプリケーションサーバでのインストール方法は、新規インストールとします。

注意事項

Application Server のインストール時に、使用する JDK のバージョンを選択する必要があります。インストールする JDK によって、アプリケーションサーバで使用できる機能が異なります。目的に合わせて使用する JDK を決定してください。また、インストール時にも次の違いがあります。

- JDK11 の場合：旧バージョンからの更新インストールおよび新規インストールが可能
- JDK17 の場合：新規インストールだけが可能

機能の違いについては、マニュアル「[アプリケーションサーバ 機能解説 保守／移行編](#)」の「10.3.3 アプリケーションサーバ 11-00, 11-10, 11-20, 11-30, 11-40, または 11-50 から, 11-60 までの仕様変更」を参照してください。

(2) Component Container インストール時のユーザ環境の扱い

アプリケーションサーバ製品に含まれる構成ソフトウェアの一つである、Component Container をインストールするとき、インストールの種類によってユーザ環境の扱いが異なります。インストールの種類ごとのユーザ環境の扱いを次の表に示します。

表 2-1 インストールの種類ごとのユーザ環境の扱い

インストールの種類	J2EE サーバまたはバッチサーバのユーザ定義	サーバ管理コマンドのユーザ定義	サーバの作業ディレクトリ*
新規インストール	インストーラが、デフォルトの J2EE サーバまたはバッチサーバをセットアップし、ユーザ定義を作成します。	インストーラが、デフォルトのユーザ定義を作成します。	J2EE サーバまたはバッチサーバが、初回起動時に作成します。
更新インストール 同一バージョン同士 (上書き) で更新	前の設定を引き継ぎます。 デフォルトのサーバ (ホスト名称のサーバ) の定義ディレクトリ (<Application Server のインストールディレクトリ> >¥CC¥server¥usrconf¥ejb¥ <ホスト名称>, または /opt/)	前の設定を引き継ぎます。	前の設定を引き継ぎます。

インストールの種類		J2EE サーバまたはバッチサーバのユーザ定義	サーバ管理コマンドのユーザ定義	サーバの作業ディレクトリ*
異なるバージョン同士(バージョンアップ)で更新		Cosminexus/CC/server/usrconf/ejb/<ホスト名称>がない場合には、定義ファイルが自動的にセットアップされます。		
		前の設定を引き継ぎます。	前の設定を引き継ぎます。	インストール後、cjenvupdate コマンド (J2EE サーバまたはバッチサーバの移行コマンド) を実行して移行してください。移行情報は、既存のディレクトリに格納され、そのまま利用されます。 詳細は、マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「10.2.2 更新インストールの場合」を参照してください。

注※ サーバの作業ディレクトリは、J2EE サーバとバッチサーバの場合で異なります。J2EE サーバの作業ディレクトリについては、「[付録 C.1 J2EE サーバの作業ディレクトリ](#)」を参照してください。バッチサーバの作業ディレクトリについては、「[付録 C.2 バッチサーバの作業ディレクトリ](#)」を参照してください。

(3) 製品に対応する構成ソフトウェアについて

各製品に含まれている構成ソフトウェアは、アプリケーションサーバおよび BPM/ESB 基盤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。製品と構成ソフトウェアの対応を次の表に示します。なお、各構成ソフトウェアの詳細は、マニュアル「アプリケーションサーバ & BPM/ESB 基盤 概説」の「2.2.2 構成ソフトウェアの機能概要」を参照してください。

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	アプリケーションサーバ		BPM/ESB 基盤		オプション製品
	Application Server	Developer	Service Platform	Service Architect	
Application Development Plug-in	—	○	—	○	—
Component Container	○	○	○	○	—
Component Container - Client	—	—	—	—	○
Component Container - Redirector	○	—	○	—	—
Component Transaction Monitor	○	○	○	○	—

構成ソフトウェア名	アプリケーションサーバ		BPM/ESB 基盤		オプション製品
	Application Server	Developer	Service Platform	Service Architect	Client
Developer's Kit for Java	○	○	○	○	○
HTTP Server	○	○	○	○	—
Performance Tracer	○	○	○	○	○
Reliable Messaging	○	○	○	○	—
Service Coordinator	—	—	○	○	—
Service Development Plug-in	—	—	—	○	—
TPBroker	○	○	○	○	○
Web Services - Security	○	○	○	○	—
XML Processor	○	○	○	○	○
HiRDB/Single Server Version 10	—	○	—	○	—

(凡例)

○：含まれます。 —：含まれません。

また、構成ソフトウェア間では、次の表に示す依存関係があります。インストールおよびアンインストール時には注意してください。

表 2-3 構成ソフトウェア間の依存関係

操作対象の構成ソフトウェア	前提となる構成ソフトウェア
Application Development Plug-in	<ul style="list-style-type: none"> Component Container Developer's Kit for Java XML Processor
Component Container	<ul style="list-style-type: none"> Developer's Kit for Java Performance Tracer TPBroker XML Processor
Component Container - Client	<ul style="list-style-type: none"> Developer's Kit for Java Performance Tracer TPBroker
Component Container - Redirector	<ul style="list-style-type: none"> Performance Tracer
Component Transaction Monitor	<ul style="list-style-type: none"> Performance Tracer

操作対象の構成ソフトウェア	前提となる構成ソフトウェア
Developer's Kit for Java	—
HTTP Server	—
Performance Tracer	—
Reliable Messaging	<ul style="list-style-type: none"> Component Container
Service Coordinator	<ul style="list-style-type: none"> Component Container Component Transaction Monitor Developer's Kit for Java HTTP Server Performance Tracer Reliable Messaging TPBroker XML Processor
Service Development Plug-in	<ul style="list-style-type: none"> Component Container Developer's Kit for Java Service Coordinator XML Processor
TPBroker	<ul style="list-style-type: none"> Developer's Kit for Java
Web Services - Security	<ul style="list-style-type: none"> Component Container Developer's Kit for Java XML Processor
XML Processor	<ul style="list-style-type: none"> Component Container* Developer's Kit for Java
HiRDB/Single Server Version 10	—
HiRDB SQL Executer Version 9	—

(凡例) －：前提となる構成ソフトウェアがないことを示します。

注※ Client 以外の場合に、スキーマキャッシュ機能を使用するときには前提となります。Client の場合は前提となりません。

(4) 製品の提供媒体について

製品の提供媒体が CD-ROM の場合の手順を説明しています。製品の提供媒体が DVD-ROM の場合は、CD-ROM を DVD-ROM に読み替えてください。

2.2.2 Application Server を新規インストールする (Windows の場合)

アプリケーションサーバに接続して、Application Server を新規インストールします。新規インストールでは、製品の提供媒体であるインストーラを使用します。インストーラの使用方法については、製品の提供媒体のドキュメントを参照してください。

新規インストールの手順を次に示します。インストール作業には Administrator 権限が必要です。インストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を、インストール時のエラーコードについては、「[付録 I.6 Application Server のインストールおよびアンインストール時のエラーコード一覧 \(Windows の場合\)](#)」を参照してください。

手順

1. 製品の提供媒体を CD-ROM ドライブにセットします。

[日立総合インストーラ] ダイアログに、「選択されたソフトウェアをインストールします。」と表示されます。

[日立総合インストーラ] ダイアログが表示されない場合、エクスプローラを使用して、CD-ROM ディレクトリの「HCD_INST.EXE」をダブルクリックしてください。

2. Application Server を選択した状態で、[インストール実行] ボタンをクリックします。

[インストール処理開始の確認 - 日立総合インストーラ] ダイアログに、「インストールを開始します。よろしいですか？」と表示されます。

3. [OK] ボタンをクリックします。

[uCosminexus Application Server セットアッププログラムへようこそ] ダイアログが表示されます。

4. [次へ] ボタンをクリックします。

[インストール先の選択] ダイアログが表示されます。

5. 必要に応じて [インストール先のフォルダ] を選択して、[次へ] ボタンをクリックします。

[機能の選択] ダイアログが表示されます。

注意事項

[インストール先のフォルダ] には、次のパスを指定しないでください。

- ネットワークドライブ
- 2バイトコードを含むパス
- ¥ / : , ; * ? " < > | \$ % ^ ' ! () = + { } @ [] &を含むパス※
- 半角 50 文字を超えるパス
- 末尾が¥のパス (C:¥など)

注※ プログラムフォルダ名にも指定できません。「;」と「/」はすべての入力情報にも指定できません。

6. [標準 - このセットアップを推奨します。] の左にあるボタンをクリックします。

[使用する Java バージョンの選択] ページが表示されます。

注意事項

使用する Java バージョンの環境を選択してください。選択した Java バージョンの Cosminexus Developer's Kit for Java (TM) がインストールされます。Java バージョンの違いにより使用できる機能範囲が異なりますので、マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「10.3.3 アプリケーションサーバ 11-00, 11-10, 11-20, 11-30, 11-40, または 11-50 から, 11-60 までの仕様変更」を参照してください。

7. 「JDK17」または「JDK11」のどちらかを選択して、[次へ] ボタンをクリックします。

[ユーザ情報] ダイアログが表示されます。

参考

製品の構成ソフトウェアを選択して新規インストールすることもできます。この場合は、[カスタム - 詳しいユーザやシステム管理者のためのセットアップ] の左にあるボタンをクリックし、表示される [プログラムの選択] ダイアログで、インストールする構成ソフトウェア (プログラム) を選択して、[次へ] ボタンをクリックします。

製品に対応している構成ソフトウェアについては、「2.2.1(3) 製品に対応する構成ソフトウェアについて」を参照してください。

8. [ユーザ名] および [会社名] を入力して [次へ] ボタンをクリックします。

注意事項

[ユーザ名] および [会社名] にスラッシュ (/), セミコロン (;), ダブルクオーテーション ("") を含めることはできません。

[プログラム フォルダの選択] ダイアログが表示されます。

9. 必要に応じて [プログラムフォルダ] を変更して、[次へ] ボタンをクリックします。

[インストールの開始] ダイアログが表示されます。

10. 設定した内容を確認して、問題がなければ [次へ] ボタンをクリックします。

インストールが開始されます。インストールが完了すると、[セットアップの完了] ダイアログが表示されます。

11. [完了] ボタンをクリックします。

OS を再起動するかどうかを確認する画面が表示されます。

12. [はい] ボタンをクリックします。

OS が再起動し、Application Server の新規インストールが完了します。

13. 修正パッチを適用します。

修正パッチとは、製品リリース後に提供される改良版のことです。

Application Server のインストールが完了したら、ソフトウェアサポートサービスの Web サイトから最新の修正パッチを入手して、適用してください。

ソフトウェアサポートサービスの Web サイトにアクセスできない環境の場合は、製品に同梱されている修正パッチ CD を利用してください。

2.2.3 Application Server を新規インストールする (AIX の場合)

アプリケーションサーバに接続して、Application Server を新規インストールします。

新規インストールでは、製品の提供媒体である PP インストーラを使用します。

新規インストールの手順を次に示します。インストール作業には root 権限が必要です。新規インストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を参照してください。

インストール前に環境変数 PATH に /usr/bin, /bin を設定してください。この設定はインストール後も有効となるようにしてください。

構成ソフトウェアの Component Container に関する操作権限をスーパーユーザ以外の一般ユーザ (Component Container 管理者) に与えることができます。この場合、インストールの前後で作業が必要になります。詳細は、「[4.1.3 Component Container 管理者を設定する \(UNIX の場合\)](#)」を参照してください。

手順

1. アプリケーションサーバに root 権限 (スーパーユーザ) でログインします。
2. PP インストーラ実行時の言語種別と、実行するターミナルの言語が一致しているかどうかを確認し、一致していない場合は一致させます。
3. 製品の提供媒体を CD-ROM ドライブにセットします。
4. 製品の提供媒体が CD-ROM の場合は、CD-ROM ファイルシステムをマウントします。

コマンドの実行例を次に示します。下線部には、デバイススペシャルファイル名、および CD-ROM ファイルシステムのマウントディレクトリ名を指定します。なお、これらの名称は、OS、ハードウェア、および環境によって異なります。

(実行例)

```
mount -r -v cdrfs /dev/cd0 /cdrom
```

5. セットアッププログラムを起動します。

コマンドの実行例を次に示します。下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。

(実行例)

```
/cdrom/aix/setup /cdrom
```

CD-ROM セットアッププログラムによって、PP インストーラと常駐プロセス自動起動プログラムがハードディスク上にインストールされ、PP インストーラが自動的に起動されます。

注意事項

CD-ROM のディレクトリ名やファイル名は、マシン環境によって記述した内容と見え方が異なることがあります。ls コマンドで確認し、表示されたファイル名をそのまま入力してください。

6. PP インストーラのメインメニューで、[I] キーを押します。

PP インストール画面が表示されます。

7. プログラムにカーソルを移動させ、[スペース] キーを押します。

すべてのプログラム（Application Server に対応している構成ソフトウェア、および製品の共通モジュール）を選択します。選択したプログラムの左側には<@>が表示されます。なお、プログラムを選択してインストールすることもできます。

注意事項

製品に対応している構成ソフトウェアについては、「2.2.1(3) 製品に対応する構成ソフトウェアについて」を参照してください。製品の共通モジュールは、製品によって表示されるプログラム名が異なります。

8. すべてのプログラムの左側に<@>が表示されていることを確認して、[I] キーを押します。

画面の最下行に「Install PP? (y: install, n: cancel)==>」メッセージが表示されます。

9. [y] キーまたは [Y] キーを押します。

インストールが開始されます。

[n] キーまたは [N] キーを押すと、インストールが中止されて PP インストール画面に戻ります。

10. インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

PP インストーラのメインメニューに戻ります。

11. 必要に応じて、PP インストーラのメインメニューで [L] キーを押して、インストール済みのプログラムを確認します。

PP 一覧表示画面が表示されます。[P] キーを押すと、インストール済みのプログラム一覧が「/tmp/hitachi_PPLIST」に出力されます。[Q] キーを押すと、PP インストーラのメインメニューに戻ります。

12. PP インストーラのメインメニューで、[Q] キーを押します。

Application Server の新規インストールが完了します。

13. 修正パッチを適用します。

修正パッチとは、製品リリース後に提供される改良版のことです。

Application Server のインストールが完了したら、ソフトウェアサポートサービスの Web サイトから最新の修正パッチを入手して、適用してください。

ソフトウェアサポートサービスの Web サイトにアクセスできない環境の場合は、製品に同梱されている修正パッチ CD を利用してください。

2.2.4 Application Server を新規インストールする (Linux の場合)

アプリケーションサーバに接続して、Application Server を新規インストールします。

新規インストールでは、製品の提供媒体である PP インストーラを使用します。Linux の場合、Reliable Messaging および Web Services - Security をインストールする PP インストーラと、それ以外の構成ソフトウェアと製品の共通モジュールをインストールする PP インストーラが異なります。

新規インストールの手順を次に示します。インストール作業には root 権限が必要です。インストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を参照してください。

インストール前に環境変数 PATH に /usr/bin, /bin を設定してください。この設定はインストール後も有効となるようにしてください。

構成ソフトウェアの Component Container に関する操作権限をスーパーユーザ以外の一般ユーザ (Component Container 管理者) に与えることができます。この場合、インストールの前後で作業が必要になります。詳細は、「[4.1.3 Component Container 管理者を設定する \(UNIX の場合\)](#)」を参照してください。

手順

1. アプリケーションサーバに root 権限 (スーパーユーザ) でログインします。

2. 次のパッケージがインストールされていることを確認します。

パッケージの中には、ご使用の Red Hat Enterprise Linux Server のバージョンによってはデフォルトでインストールされないものがあります。

- als-audio (x86_64)
- bash (x86_64)
- bzip2-libs (x86_64)
- chkconfig (x86_64)
- coreutils (x86_64)
- cpio (x86_64)
- cups-libs (x86_64)
- findutils (x86_64)

- fontconfig (x86_64)
- freetype (x86_64)
- gawk (x86_64)
- GConf2 (x86_64)^{※1}
- gdb (x86_64)
- glib2 (x86_64)
- glibc (i686)
- glibc (x86_64)
- graphite2 (x86_64)^{※2}
- grep (x86_64)
- gtk2 (x86_64)
- gtk3 (x86_64)
- gzip (x86_64)
- harfbuzz(x86_64)^{※2}
- krb5-libs (x86_64)
- ksh (x86_64)
- libbrotli(x86_64)^{※2}
- libgcc (i686)
- libgcc (x86_64)
- libpng (x86_64)
- libstdc++ (i686)
- libstdc++ (x86_64)
- libX11 (x86_64)
- libXau (x86_64)
- libxcb (x86_64)
- libxcrypt (x86_64)^{※1}
- libxcrypt-compat (x86_64)^{※2}
- libXext (x86_64)
- libXi (x86_64)
- libXrender (x86_64)
- libXtst (x86_64)

- lksctp-tools (x86_64)
- ncompress (x86_64)^{※1}
- ncurses (x86_64)
- net-tools (x86_64)
- nss (x86_64)
- pcre(x86_64)^{※2}
- pcsc-lite-libs (x86_64)
- procps-ng (x86_64)
- rpm (x86_64)
- sed (x86_64)
- sysstat (x86_64)
- tar (x86_64)
- tcsh (x86_64)
- which (x86_64)
- zlib (x86_64)

注※1

Red Hat Enterprise Linux 8 (AMD/Intel 64) をご使用の場合だけ必要です。

注※2

Red Hat Enterprise Linux 9 (AMD/Intel 64) をご使用の場合だけ必要です。

パッケージがインストールされているかどうかは、 rpm コマンドを実行して確認します。コマンドの実行例と実行結果の例を次に示します。

(実行例)

```
# rpm -q --qf '%{NAME}-%{ARCH}¥n' ncurses
```

(実行結果の例)

```
ncurses-x86_64
```

この例では、64 ビット版の ncurses パッケージがインストールされていることを示します。

「package <パッケージ名> is not installed」メッセージが表示された場合は、パッケージがインストールされていないので、<パッケージ名>に示すパッケージをインストールしてください。なお、インストール時には、関連するパッケージも必要に応じてインストールしてください。

3. PP インストーラ実行時の言語種別と、実行するターミナルの言語が一致しているかどうかを確認し、一致していない場合は一致させます。

4. 製品の提供媒体を CD-ROM ドライブにセットします。

5. 製品の提供媒体が CD-ROM の場合は、CD-ROM ファイルシステムをマウントします。

コマンドの実行例を次に示します。下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。なお、マウントディレクトリ名は、OS、ハードウェア、および環境によって異なります。

(実行例)

```
mount -r -o mode=0544 /dev/cdrom /mnt/cdrom
```

6. Reliable Messaging および Web Services - Security をインストールするセットアッププログラムを起動します。

コマンドの実行例を次に示します。下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。

(実行例)

```
/mnt/cdrom/linux/setup /mnt/cdrom
```

CD-ROM セットアッププログラムによって、PP インストーラと常駐プロセス自動起動プログラムがハードディスク上にインストールされ、PP インストーラが自動的に起動されます。

注意事項

CD-ROM のディレクトリ名やファイル名は、マシン環境によって記述した内容と見え方が異なることがあります。ls コマンドで確認し、表示されたファイル名をそのまま入力してください。

7. PP インストーラのメインメニューで、[I] キーを押します。

PP インストール画面が表示されます。

8. プログラムにカーソルを移動させ、[スペース] キーを押します。

すべてのプログラム (Reliable Messaging および Web Services - Security) を選択します。選択したプログラムの左側には<@>が表示されます。なお、プログラムを選択してインストールすることもできます。

9. すべてのプログラムの左側に<@>が表示されていることを確認して、[I] キーを押します。

画面の最下行に「Install PP? (y: install, n: cancel)==>」メッセージが表示されます。

10. [y] キーまたは [Y] キーを押します。

インストールが開始されます。

[n] キーまたは [N] キーを押すと、インストールが中止されて PP インストール画面に戻ります。

11. インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

PP インストーラのメインメニューに戻ります。

12. PP インストーラのメインメニューで、[Q] キーを押します。

PP インストーラが終了します。

13. Reliable Messaging および Web Services - Security 以外と共通モジュールをインストールするセットアッププログラムを起動します。

コマンドの実行例を次に示します。下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。

(実行例)

```
/mnt/cdrom/x64lin/setup /mnt/cdrom
```

CD-ROM セットアッププログラムによって、PP インストーラと常駐プロセス自動起動プログラムがハードディスク上にインストールされ、PP インストーラが自動的に起動されます。

注意事項

CD-ROM のディレクトリ名やファイル名は、マシン環境によって記述した内容と見え方が異なることがあります。ls コマンドで確認し、表示されたファイル名をそのまま入力してください。

14. PP インストーラのメインメニューで、[I] キーを押します。

PP インストール画面が表示されます。

15. プログラムにカーソルを移動させ、[スペース] キーを押します。

すべてのプログラム (Application Server に対応している Reliable Messaging および Web Services - Security 以外の構成ソフトウェア、および製品の共通モジュール) を選択します。選択したプログラムの左側には<@>が表示されます。なお、プログラムを選択してインストールすることもできます。

注意事項

製品に対応している構成ソフトウェアについては、「2.2.1(3) 製品に対応する構成ソフトウェアについて」を参照してください。製品の共通モジュールは、製品によって表示されるプログラム名が異なります。

16. すべてのプログラムの左側に<@>が表示されていることを確認して、[I] キーを押します。

画面の最下行に「Install PP? (y: install, n: cancel)==>」メッセージが表示されます。

17. [y] キーまたは [Y] キーを押します。

インストールが開始されます。

[n] キーまたは [N] キーを押すと、インストールが中止されて PP インストール画面に戻ります。

18. インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

PP インストーラのメインメニューに戻ります。

19. 必要に応じて、PP インストーラのメインメニューで [L] キーを押して、インストール済みのプログラムを確認します。

PP 一覧表示画面が表示されます。[P] キーを押すと、インストール済みのプログラム一覧が「/tmp/hitachi_PPLIST」に出力されます。[Q] キーを押すと、PP インストーラのメインメニューに戻ります。

20. PP インストーラのメインメニューで、[Q] キーを押します。

Application Server の新規インストールが完了します。

21. 修正パッチを適用します。

修正パッチとは、製品リリース後に提供される改良版のことです。

Application Server のインストールが完了したら、ソフトウェアサポートサービスの Web サイトから最新の修正パッチを入手して、適用してください。

ソフトウェアサポートサービスの Web サイトにアクセスできない環境の場合は、製品に同梱されている修正パッチ CD を利用してください。

2.2.5 データベースに接続するための製品をインストールする

アプリケーションサーバに接続して、データベースに接続するための製品（HiRDB Run Time または Oracle JDBC Thin Driver）をインストールします。

インストール方法についてはマニュアル「HiRDB UAP 開発ガイド」または Oracle のマニュアルを参照してください。

2.2.6 システム環境をセットアップする（単一の Web フロントシステムを構築する）

単一の Web フロントシステムのシステム環境をセットアップします。

参考

説明で使用するコマンドの参照先、コマンドの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

システム環境のセットアップには、セットアップウィザードを使用します。システム環境のセットアップ時の注意事項については、「2.2.7 システム環境をセットアップするときの注意事項」を参照してください。

注意事項

セットアップウィザードでセットアップしたシステム環境では、次に示す項目は自動設定されます。これらの項目にほかの内容を設定したい場合は、運用管理ポータルまたは Smart Composer 機能を使用してシステム環境を構築してください。

- ホスト名にはコンピュータ名が設定されます。設定された名称は変更できません。
- 論理サーバ名には次の名称が設定されます。設定された名称は変更できません。

パフォーマンストレーサ：MyPerformanceTracer

J2EE サーバ：MyJ2EEServer

Web サーバ：MyWebServer

(1) セットアップウィザードの起動とシステム環境のセットアップ

セットアップウィザードを起動して、システム環境をセットアップします。

1. コマンドプロンプトで `setupwizard` コマンドを実行し、セットアップウィザードを起動します。

コマンドの実行例を次に示します。

```
setupwizard
```

2. [ようこそ] 画面で「1」を入力して、Enter キーを押します。

3. [セットアップ方法の選択] 画面で「1」を入力して、Enter キーを押します。

4. [データベース接続環境の設定] 画面で「y」を入力して、Enter キーを押します。

5. 使用するデータベース (HiRDB の場合は「1」, Oracle の場合は「2」) を入力して、Enter キーを押します。

6. 画面の指示に従って、データベースに対する接続情報を入力して、Enter キーを押します。

データベースに対する接続情報として、データベースのホスト名、ポート番号、アクセスユーザ、アクセスユーザのパスワードなどを画面の指示に従って入力します。

入力項目が終了すると、「現在の設定」を確認する画面が表示されます。

7. 「現在の設定」の内容を確認したら「1」を入力して、Enter キーを押します。

[セットアップ情報] 画面が表示されます。

参考

「現在の設定」の内容に誤りがある場合は「2」を入力して、手順 4. から設定し直してください。

8. [セットアップ情報] 画面で「1」を入力して、Enter キーを押します。

セットアップを開始し、処理ごとにセットアップの実行状態を示すメッセージが表示されます。

すべての処理でセットアップが完了すると、セットアップ情報ファイルおよび簡易構築定義ファイルの保存と、セットアップウィザードの終了を示すメッセージが表示されます。

(2) OS の起動と同時に運用管理機能を自動起動するための設定

OS を起動すると同時に、運用管理機能を自動的に起動するように設定します。

1. コマンドプロンプトで `mngautorun` コマンドを実行し、ホストの起動と同時に運用管理機能 (Management Server および運用管理エージェント) を起動するように設定します。

コマンドの実行例を次に示します。

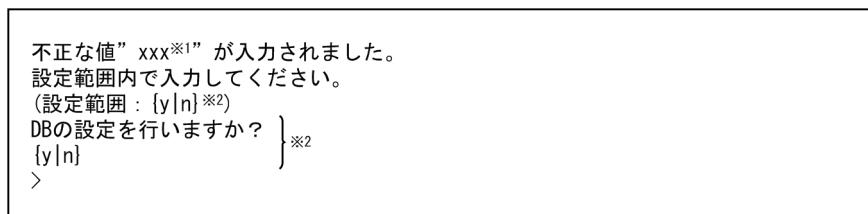
```
mngautorun both -sync
```

2.2.7 システム環境をセットアップするときの注意事項

ここでは、システム環境をセットアップするときの注意事項について説明します。

- ・セットアップウィザードを実行できるユーザは次のとおりです。次に示すユーザ以外がセットアップウィザードを実行した場合、エラーとなります。
 - ・Windows の場合
管理者特権でコマンドを実行してください。
 - ・UNIX の場合
root 権限のあるユーザでコマンドを実行してください。
- ・コマンドを実行している間は、Ctrl+C キーなどで外部から停止しないでください。
- ・セットアップ処理の途中でエラーが発生した場合は、セットアップ処理は停止され、セットアップ失敗の画面が表示されます。この場合、セットアップ中のシステムを削除してから再度セットアップを実行してください。セットアップ中のシステムを削除するには、セットアップウィザードでアンセットアップを実行します。アンセットアップについては「3.3.1 システム環境をアンセットアップする」を参照してください。
- ・セットアップウィザード操作時に、設定値の入力を求められている場合に何も入力しないでEnter キーを押すと、設定項目にデフォルト値が適用されます。なお、デフォルト値がない設定項目もあります。デフォルト値がない設定項目でEnter キーを押すとエラーとなり、再度設定値の入力が求められます。
- ・設定値の入力を求められている場合に、設定できない値を入力してEnter キーを押すとエラーとなり、次の画面が表示されます。

図 2-1 設定できない値を入力した場合の画面



注※1 入力した値が表示されます。

注※2 設定項目によって表示が異なります。

- ・設定できる値が「0 文字以上」の設定項目について設定値の入力を求められている場合に何も入力しないでEnter キーを押すと、空文字が適用されます。
なお、一度値を設定した項目について値を再入力する場合は次の画面が表示されます。

図 2-2 設定できる値が「0 文字以上」の設定項目について値を再入力する場合の画面

データベースのアクセスユーザを入力してください。※
(前回の入力値から変更しない場合は、同じ値を入力してください)
>

注※ 設定項目によって表示が異なります。

この画面が表示された場合に何も入力しないで Enter キーを押すと、設定済みの値は適用されないで、空文字が適用されます。

- ・セットアップウィザードでは、セットアップウィザードを実行するホスト上にだけシステム環境をセットアップできます。ほかのホスト上にはシステム環境をセットアップできません。
- ・標準セットアップでセットアップウィザードを実行すると、次の表に示すポート番号を使用したシステム環境がセットアップされます。

表 2-4 標準セットアップで使用するポート番号

ポート番号	用途
80	Web サーバのポート番号
900	インプロセスのネーミングサービスのポート番号
8008	http のポート番号
28008	管理用サーバのポート番号
20295	運用管理エージェントのポート番号
23152	RMI レジストリのポート番号
28005	Management Server 終了要求受信ポート番号
28009	Management Server 内部通信用ポート番号
28080	Management Server 接続 HTTP ポート番号

これらのポート番号をすでに使用している場合は、標準セットアップでは構築できません。[セットアップ方法の選択] 画面で「2」(カスタムセットアップ) を入力して、次の画面で重複しないポート番号を設定してください。

- ・[運用管理機能の設定] 画面
- ・[システム環境の設定] 画面

なお、ポート番号以外を標準セットアップと同じ設定にする場合は、[運用管理ユーザの設定] 画面、[J2EE サーバの起動ユーザの設定] 画面 (UNIX の場合) および [システム構成の選択] 画面で「1」(デフォルト) を入力してセットアップしてください。

- ・セットアップに失敗した場合に [セットアップ失敗] 画面で y を入力すると、障害に対処するための資料として、snapshot ログが自動的に収集されます。

snapshot ログの格納先と収集内容を次に示します。

snapshot ログの格納先

- ・<セットアップウィザードのログの出力先>/snapshot.zip

snapshot ログの収集内容

- ・<Application Server のインストールディレクトリ>/manager/config/snapshotlog.conf で示す範囲
- ・<セットアップウィザードのログの出力先>/setupwizard<ファイル面数>.log
- ・<セットアップウィザードのログの出力先>/maintenance/setupwizard<ファイル面数>.log
- ・<Application Server のインストールディレクトリ>/manager/setup/config/
- ・<Application Server のインストールディレクトリ>/manager/setup/tmp/

<セットアップウィザードのログの出力先>は、 setup.cfg の setup.log.dir で指定した出力先です。 setup.cfg については、 マニュアル「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)」の「8.2.18 setup.cfg (セットアップウィザード用設定ファイル)」を参照してください。

- ・セットアップウィザードで使用するファイルの一覧を次の表に示します。

表 2-5 セットアップウィザードで使用するファイルの一覧

分類	ファイル名称	概要	出力先
セットアップ情報リスト	setupwizardlist.txt	セットアップ処理完了（成功）後出力されます。 セットアップウィザードで構築した環境を確認するためのファイルです。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/setup/config
セットアップ情報ファイル	setupwizard.dat	[セットアップ情報] 画面で「セットアップ情報を保存する」を選択した場合に出力されます。 一括セットアップで使用できます。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/setup/config UNIX の場合 /opt/Cosminexus/manager/setup/config
	setupwizard_success.dat	セットアップ処理完了（成功）後に自動的に出力されます。 一括セットアップまたはシステムの複製で使用できます。	
	setupwizard_failed.dat	セットアップ処理完了（失敗）後に自動的に出力されます。 一括セットアップで使用できます。 ただし、このファイルを使用して一括セットアップをする場合は、セットアップ処理失敗の原因を取り除く必要があります。	
ログファイル	setupwizard<ファイル面数>.log	セットアップウィザードのログが出力されます。 最大ファイルサイズ、最大ファイル面数などのログファイルの設定は、 setup.cfg (セットアップウィザード用設定ファイル) で指定できます。	任意 (setup.cfg の setup.log.dir で指定した出力先)
簡易構築定義ファイル	cmxsetupwz.xml	セットアップ処理完了（成功）後出力されます。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/setup/config

分類	ファイル名称	概要	出力先
		セットアップウィザードで構築した環境の定義内容を確認するためのファイルです。	<Application Server のインストールディレクトリ >manager/setup/config
DB Connector の Connector 属性ファイル	DBConnector_HiRDB_Type4_CP_cfg.xml または DBConnector_Oracle_C_P_cfg.xml	セットアップ処理完了（成功）後出力されます。 セットアップウィザードで構築した環境で使用する DB Connector の定義内容を確認するためのファイルです。	• UNIX の場合 /opt/Cosminexus/manager/setup/config
セットアップウィザード用設定ファイル	setup.cfg	セットアップウィザードのログファイルの出力先, 最大ファイルサイズ, 最大ファイル面数などを指定するファイルです。	

2.2.8 システム環境定義の設定とは

システム環境のセットアップが完了した時点では, 各種定義ファイルのパラメタには, デフォルト値が設定されています。システム要件を満たしたシステム環境にするためには, 各種定義ファイルのパラメタの設定値を必要に応じて変更します。この作業をシステム環境定義の設定といいます。

システム要件で設定する代表的なパラメタとして, タイムアウトを設定する作業について説明します。

- タイムアウトの設定

タイムアウトの設定は, システムのトラブル発生を検知して, リクエストの応答が返らなくなることを防ぎ, 適宜リソースを解放することを目的とします。システムのタイムアウト, トランザクションのタイムアウト, データベースのタイムアウトなどが設定できます。タイムアウトに設定する値の考え方については, マニュアル「アプリケーションサーバシステム設計ガイド」の「8.6 タイムアウトを設定する」を参照してください。

2.2.9 タイムアウトを設定する

次に示すタイムアウトを設定する手順について説明します。

- J2EE アプリケーションのメソッドタイムアウト
- レスポンスデータ待ちの通信タイムアウト
- トランザクションのトランザクションタイムアウト時間のデフォルト
- データベースのタイムアウト (HiRDB の場合)

参考

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) J2EE アプリケーションのメソッドタイムアウトの設定

Web コンテナ上でのリクエスト処理待ち時間を J2EE アプリケーションの実行時間監視機能を利用して設定します。

1. cosminexus.xml を編集して、<任意のディレクトリ>下の META-INF ディレクトリに格納します。

cosminexus.xml では、<war>タグ内で次の指定をします。

- <servlet>－<method-observation-timeout>タグに、J2EE アプリケーションのメソッドタイムアウトの値を指定します。

<war>タグ内に複数の<servlet>タグがある場合は、それぞれに指定します。

2. コマンドプロンプトで、jar コマンドを実行し、アーカイブ形式の J2EE アプリケーションに手順 1.で編集した cosminexus.xml を設定します。

コマンドの実行例を次に示します。

```
jar uvf <MyApp1.earのファイルパス> -C <META-INFディレクトリを置いたディレクトリ> <META-INFディレクトリ下のcosminexus.xmlのファイルパス>
```

<MyApp1.ear のファイルパス>は、J2EE アプリケーション (MyApp1.ear) を格納したディレクトリのパスとファイル名 (MyApp1.ear) を指定します。<META-INF ディレクトリを置いたディレクトリ>は、cosminexus.xml の格納先が次のような構造の場合、<任意のディレクトリ>となります。

```
<任意のディレクトリ>
  └META-INF
    └cosminexus.xml
```

<META-INF ディレクトリ下の cosminexus.xml のファイルパス>は、手順 1.で cosminexus.xml を格納した META-INF ディレクトリのパスとファイル名 (cosminexus.xml) を指定します。

(2) レスポンスデータ待ちの通信タイムアウトの設定

Web サーバに登録したリバースプロキシでの Web コンテナからのデータ受信待ち時間を設定します。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。
3. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。

4. [サーバビュー] タブで [DefaultDomain] – [論理 Web サーバ] – [Web サーバ] – [MyWebServer] をクリックし, [リバースプロキシ] タブをクリックします。
5. [リバースプロキシの設定] 画面で [通信タイムアウト] に値を設定して [適用] ボタンをクリックします。
環境設定情報の保存を示すメッセージが表示されます。
6. [設定情報の配布] アンカーをクリックします。
7. [設定情報の配布] 画面で [全て配布] ボタンをクリックします。
[ステータス] が「空欄」から「配布中」, 「配布済」へと遷移します。

参考

[ステータス] が「配布済」の論理サーバがある場合は, [ステータス] が「空欄」の論理サーバだけを選択して配布できます。この場合, [配布] をチェックしてから, [選択して配布] ボタンをクリックしてください。選択した論理サーバの [ステータス] が「配布済」となります。

8. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
9. [サーバビュー] タブで [DefaultDomain] をクリックし, [一括再起動] タブをクリックします。
10. [一括再起動] 画面で [実行] ボタンをクリックします。
一括再起動要求が受け付けられ, [論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」から「停止中」, 「停止」, 「起動中」, 「稼働中」へと遷移します。

(3) トランザクションのトランザクションタイムアウト時間のデフォルトの設定

EJB コンテナとデータベースのトランザクションのタイムアウトを設定します。

1. Web ブラウザを起動し, 「<http://localhost:28080/mngsvr/index.jsp>」に接続して, 運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して, [ログイン] ボタンをクリックします。
3. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。
4. [サーバビュー] タブで [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] をクリックし, [サービス] – [トランザクション] タブをクリックします。
5. [トランザクションの設定] 画面で [タイムアウト時間] に値を設定して [適用] ボタンをクリックします。
環境設定情報の保存を示すメッセージが表示されます。

6. [設定情報の配布] アンカーをクリックします。
 7. [設定情報の配布] 画面で [全て配布] ボタンをクリックします。
[ステータス] が「空欄」から「配布中」、「配布済」へと遷移します。
- 参考**

[ステータス] が「配布済」の論理サーバがある場合は、[ステータス] が「空欄」の論理サーバだけを選択して配布できます。この場合、[配布] をチェックしてから、[選択して配布] ボタンをクリックしてください。選択した論理サーバの [ステータス] が「配布済」となります。
8. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
 9. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括再起動] タブをクリックします。
 10. [一括再起動] 画面で [実行] ボタンをクリックします。
一括再起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」から「停止中」、「停止」、「起動中」、「稼働中」へと遷移します。

(4) レスポンスタイムアウトおよびリクエスト間隔タイムアウトの設定 (HiRDB の場合)

データベースシステムのサーバ側の障害を検知するためのタイムアウト（レスポンスタイムアウト）、およびクライアント側の障害を検知するためのタイムアウト（リクエスト間隔タイムアウト）を設定します。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。
3. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
4. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[開始/停止] タブをクリックします。
5. [リソースアダプタの開始/停止] 画面で [ステータス] が「稼働」の場合は [操作] の [停止] アンカーをクリックします。
6. [リソースアダプタの停止] 画面で [はい] ボタンをクリックします。
7. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから停止し直してください。

8. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし, [プロパティ設定] タブをクリックします。
9. [リソースアダプタのプロパティ設定] 画面で [操作] の [基本設定] アンカーをクリックします。
10. [リソースアダプタの基本設定] 画面で [HiRDB クライアント環境変数 (environmentVariables)] に PDCWAITTIME と PDSWAITTIME の値を設定して, [設定] ボタンをクリックします。
レスポンスタイムアウト (PDCWAITTIME) と, リクエスト間隔タイムアウト (PDSWAITTIME) が設定できます。
11. [リソースアダプタの基本設定] 画面で表示されたメッセージを確認して [戻る] アンカーをクリックします。
失敗を示すメッセージが表示された場合は, 障害要因を取り除いてからプロパティを設定し直してください。
12. [戻る] アンカーをクリックします。
13. [開始/停止] タブをクリックし, [リソースアダプタの開始/停止] 画面で [操作] の [開始] アンカーをクリックします。
14. [リソースアダプタの開始] 画面で [はい] ボタンをクリックします。
15. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は, [ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し, 障害要因を取り除いてから開始し直してください。

2.2.10 業務アプリケーションを設定して開始する (GUI 利用時)

業務を開始するためには, J2EE アプリケーションを Management Server にアップロードしたあと, インポート, 開始する必要があります。J2EE アプリケーションのアップロード, インポートおよび開始の手順について説明します。これらの作業には, 運用管理ポータルを使用します。なお, 説明で使用する業務アプリケーションは, cosminexus.xml を含めたアーカイブ形式の J2EE アプリケーション (EAR 形式または ZIP 形式のアプリケーション) です。

参考

説明で使用する画面については, 「5. システムの構築・運用時に使用する画面, コマンドおよびファイル」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(2) 論理サーバの一括起動」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) 論理サーバの一括起動

1. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] をクリックします。
3. [論理サーバの稼働状況] 画面で論理サーバの稼働状況を確認します。
4. 「停止」の場合は [一括起動] タブをクリックします。「稼働中」の場合は「(3) 業務アプリケーションの設定 (GUI 利用時)」に進みます。
5. [一括起動] 画面で [実行] ボタンをクリックします。

一括起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「停止」から「起動中」、「稼働中」へと遷移します。

(3) 業務アプリケーションの設定 (GUI 利用時)

1. アプリケーションサーバの<任意のディレクトリ>に、入手した J2EE アプリケーション (MyApp1.ear) を格納します。
2. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
3. [DefaultDomain] をクリックします。
4. [J2EE アプリケーションのアップロード] 画面で [J2EE アプリケーションファイル] に手順 1.で<任意のディレクトリ>に格納した J2EE アプリケーション (MyApp1.ear) を入力して、[アップロード] ボタンをクリックします。
5. 表示されたメッセージを確認して [戻る] アンカーをクリックします。
失敗を示すメッセージが表示された場合は、障害要因を取り除いてからアップロードし直してください。
[アップロードディレクトリファイル一覧] に手順 4.でアップロードした J2EE アプリケーションのファイル名 (MyApp1.ear) が表示されます。
6. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[インポート] タブをクリックします。
7. [J2EE アプリケーションのインポート] 画面で [J2EE アプリケーションファイル] から手順 4.でアップロードしたファイル (MyApp1.ear) を選択し、[実行] ボタンをクリックします。
8. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてからインポートし直してください。

(4) 業務アプリケーションの開始 (GUI 利用時)

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[開始/停止] タブをクリックします。
3. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、「(3) 業務アプリケーションの設定 (GUI 利用時)」でインポートした J2EE アプリケーションの表示名 (MyApp) の [操作] の [開始] アンカーをクリックします。
4. [J2EE アプリケーションの開始] 画面で [はい] ボタンをクリックします。
5. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「稼働」になります。

2.3 可用性を確保したシステムへの変更

ここでは、単一の Web フロントシステムから可用性を確保したシステム環境へ変更する手順について説明します。

2.3.1 同一構成のホストを複数配置して可用性を確保する

ここでは、単一の Web フロントシステムで可用性を確保するために、同一構成のホストを複数配置して負荷分散機で負荷分散するシステム環境に変更する手順について説明します。システム構成については、「1.2.2 同一構成のホストを複数配置して可用性を確保するには」を参照してください。

同一構成のホストは、セットアップウィザードを使用して、ホストを複製して作成します。可用性を確保するために負荷分散機を利用します。同一構成のホストの複数配置時の注意事項については、「2.3.2 同一構成のホストを複数配置するときの注意事項」を参照してください。

参考

説明で使用するコマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 複製情報の出力

複製元のホストで、システムを複製するための情報（簡易構築定義ファイル）を出力します。

1. 複製元のホストのコマンドプロンプトで、setupwizard コマンドに-export オプションを指定して実行し、簡易構築定義ファイルを<任意のディレクトリ>にエクスポートします。

コマンドの実行例を次に示します。

```
setupwizard -export -omd <簡易構築定義ファイルのエクスポート先のファイルパス>
```

<簡易構築定義ファイルのエクスポート先のファイルパス>には、簡易構築定義ファイルを格納するディレクトリのパスとファイル名を指定します。簡易構築定義ファイルは XML 形式のファイルです。

(2) 複製情報を使ったセットアップ

システムを複製するための情報（簡易構築定義ファイル）と、セットアップ情報ファイルを使用して、セットアップを実行します。

1. 複製先として新たなホストを用意し、次の製品をインストールします。

- Application Server
- HiRDB Run Time または Oracle JDBC Thin Driver

2. 複製先のホストの<任意のディレクトリ>に、次のファイルをコピーします。

- ・「(1) 複製情報の出力」で出力した簡易構築定義ファイル
 - ・複製元のホストのセットアップ情報ファイル (setupwizard_success.dat)
- セットアップ情報ファイルの格納先を次に示します。

OS 種別	格納先
Windows	<Application Server のインストールディレクトリ>¥manager¥setup¥config¥
UNIX	/opt/Cosminexus/manager/setup/config/

3. 複製先のホストのコマンドプロンプトで, setupwizard コマンドに手順 2.でコピーした簡易構築定義ファイルとセットアップ情報ファイルを指定して実行し, システム環境を複製します。

コマンドの実行例を次に示します。

```
setupwizard -f <セットアップ情報ファイルのファイルパス> -md <簡易構築定義ファイルのファイルパス>
```

<セットアップ情報ファイルのファイルパス>, および<簡易構築定義ファイルのファイルパス>には, 手順 2.でコピーした格納先を指定してください。

4. [セットアップ情報] 画面で「1」を入力して, Enter キーを押します。

5. 複製先の J2EE サーバに業務アプリケーションを設定して開始します。

2.3.2 同一構成のホストを複数配置するときの注意事項

同一構成のホストを複数配置するときの注意事項を次に示します。

- ・複製元のホストで, 複製できないチューニングが実施されている場合, 複製元または複製先のホストで次の動作が発生して, 複製できないときがあります。
 - ・複製元のホストで, 簡易構築定義ファイルのエクスポートでエラーが発生します。
 - ・複製先のホストが, 複製できないチューニング情報を無視して複製されます。この場合, 複製先のホストには, 複製元のホストのセットアップ時の設定が適用されます。

複製できないチューニング項目とホストの動作を次の表に示します。

表 2-6 複製できないチューニング項目とホストの動作

項目番	チューニング項目	ホストの動作
1	Management Server の設定	<p>Management Server の設定情報のうち, 次の項目に変更があると, エクスポートでエラーが発生します。この場合は, 複製できません。</p> <ul style="list-style-type: none"> ・管理ユーザーアカウントの設定の「管理ユーザ ID」および「パスワード」 ・ネットワークの設定の「Management Server 接続 HTTP ポート番号」 ・ホストの定義の「運用管理エージェントのポート番号」

項目番	チューニング項目	ホストの動作
		なお、上記以外の Management Server の設定情報を変更しても、エクスポートではエラーになりません。ただし、複製先のホストには、複製元のホストのセットアップ時の設定が適用されます。
2	Component Container 管理者の設定	Component Container 管理者の設定を変更しても、エクスポートではエラーになりません。ただし、複製先のシステムには、複製元のホストのセットアップ時の設定が適用されます。
3	Web システムの設定	<p>Web システムの設定情報のうち、次の項目以外に変更があると、エクスポートでエラーが発生します。この場合は、複製できません。</p> <ul style="list-style-type: none"> Smart Composer 機能の簡易構築定義ファイルを使用した、論理サーバのパラメタの設定 運用管理ポータルを使用した、「論理サーバの環境設定」および「論理サーバの起動／停止」の設定
4	データベースの接続設定	データベースの接続設定のうち、JDBC ドライバの設定については、複製できます。JDBC ドライバ以外の設定項目を変更しても、エクスポートではエラーなりません。ただし、複製先のホストには、複製元のホストのセットアップ時の設定が適用されます。

- 複製先のホストのプラットフォームは、複製元のホストと同じにしてください。複製先が、複製元と異なるプラットフォームの場合、複製できないときがあります。
- 複製先のホストでは、setupwizard コマンドを使用してエクスポートした簡易構築定義ファイルを使用してください。setupwizard コマンド以外の方法でエクスポートした簡易構築定義ファイルを使用した場合、複製できないときがあります。

2.4 マルチテナントに対応したシステムへの変更

ここでは、単一の Web フロントシステムからマルチテナントに対応したシステムへ変更する手順について説明します。単一の Web フロントシステムをマルチテナントに対応させる方法には次のものがあります。

- J2EE サーバを複数配置する方法
ホストのリソースを有効活用して、かつテナントごとに業務処理（アプリケーション）を分離できます。
- リソースアダプタを複数配置する方法
データベースを共有（データベース数の増加を抑止）して、かつテナントごとにデータを分離できます。

2.4.1 J2EE サーバを複数配置してマルチテナントに対応する

ここでは、単一の Web フロントシステムからホストのリソースを有効活用してマルチテナントへ対応するシステムに変更する手順について説明します。システム構成については、「[1.2.3 J2EE サーバを複数配置してマルチテナント対応するには](#)」を参照してください。

参考

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「[5. システムの構築・運用時に使用する画面、コマンドおよびファイル](#)」を参照してください。

J2EE サーバを複数配置するホストは、Smart Composer 機能を使用して Web システムを複製したあとに論理サーバ（Web サーバ、J2EE サーバおよびパフォーマンストレーサ）を追加して作成します。ここでは、追加する J2EE サーバ上で動作する業務アプリケーションには、同一のアプリケーションを使用するとします。例題で使用する設定内容を次の表に示します。

表 2-7 例題で使用する設定内容（J2EE サーバを複数配置してマルチテナントに対応する Web フロントシステム）

項目	設定例
Web システムの名称	MyWebSystem
追加するサービスユニットの名称	unit2
追加する論理 Web サーバの名称	MyWebServer2
追加する論理 J2EE サーバの名称	MyJ2EEServer2
追加する論理パフォーマンストレーサの名称	MyPerformanceTracer2
構成変更定義ファイルのファイル名称	MyWebSystemAddUnit.xml

(1) Web システムの複製

1. コマンドプロンプトで cmx_export_model コマンドを実行し、簡易構築定義ファイルを<任意のディレクトリ>にエクスポートします。

コマンドの実行例を次に示します。

```
cmx_export_model -m localhost:28080 -u admin -o <簡易構築定義ファイルのエクスポート先のファイルパス>
```

<簡易構築定義ファイルのエクスポート先のファイルパス>には、簡易構築定義ファイルを格納するディレクトリのパスとファイル名を指定します。簡易構築定義ファイルは XML 形式のファイルです。

2. コマンドプロンプトで cmx_delete_system コマンドを実行し、システムを削除します。

システムを複製するための準備として、ホスト上に構築済みのシステムを削除します。

コマンドの実行例を次に示します。

```
cmx_delete_system -m localhost:28080 -u admin -s MyWebSystem
```

3. コマンドプロンプトで cmx_build_system コマンドを実行し、手順 1.でエクスポートした簡易構築定義ファイルから Web システムを複製します。

コマンドの実行例を次に示します。

```
cmx_build_system -m localhost:28080 -u admin -f <簡易構築定義ファイルのファイルパス>
```

<簡易構築定義ファイルのファイルパス>には、手順 1.の簡易構築定義ファイルを格納されているディレクトリのパスとファイル名を指定します。

(2) 論理サーバ (Web サーバ, J2EE サーバおよびパフォーマンストレーサ) の追加

1. 構成変更定義ファイルのテンプレートファイル (cmxaddcombinedmodel.xml) を任意の場所にコピーして、任意の名称を変更します。

サービスユニットやホストを追加する構成変更定義ファイルのテンプレートファイルは、次の場所に提供されています。ここではファイル名を「MyWebSystemAddUnit.xml」に変更します。

Windows の場合

```
<Application Server のインストールディレクトリ  
>¥manager¥config¥templates¥cmxaddcombinedmodel.xml
```

UNIX の場合

```
/opt/Cosminexus/manager/config/templates/cmxaddcombinedmodel.xml
```

2. テキストエディタなどで、手順 1.でコピーした構成変更定義ファイルを編集します。

構成変更定義ファイルには、追加するサービスユニットの定義および論理サーバの定義を指定します。また、追加する論理サーバ名、および次の表に示す論理サーバのコンフィグレーション定義の指定値が

重複しないようにしてください。論理サーバごとに重複しないように設定するコンフィグレーション定義を次の表に示します。

表 2-8 論理サーバごとに重複しないように設定するコンフィグレーション定義

論理サーバの種類	項目	param-name の指定値
パフォーマンストレーサ	PRF 識別子	PRFID
J2EE サーバ	JTA リカバリの固定ポート番号	ejbserver.distributedtx.recovery.port
	管理用サーバのポート番号	ejbserver.http.port
	インプロセスのネーミングサービス用のポート番号	ejbserver.naming.port
	RMI レジストリのポート番号	ejbserver.rmi.naming.port
	NIO HTTP サーバのポート番号	webserver.connector.nio_http.port
	稼働情報取得時のリクエスト受付ポート番号	ejbserver.rmi.remote.listener.port
Web サーバ	http ポート番号	Listen

構成変更定義ファイルの編集例を次に示します。背景色付きの太字の部分が編集する箇所です。

```

<?xml version="1.0" encoding="UTF-8"?>
<unit-addition xmlns="http://www.cosminexus.com/mngsvr/schema/UnitAddition-2.5">
  <web-system>
    <name>MyWebSystem</name>

    <!--Specify the settings for the Service Unit to add.-->
    <unit>
      <name>unit2</name>
      <allocated-host>
        <host-ref>@myhost</host-ref>
        <hosts-for>combined-tier</hosts-for>
        <define-server>
          <logical-server-name>MyPerformanceTracer2</logical-server-name>
          <logical-server-type>performance-tracer</logical-server-type>
        </define-server>
        <configuration>
          <param>
            <param-name>PRFID</param-name>
            <param-value>PRF_ID2</param-value>
          </param>
        </configuration>
        </define-server>
        <define-server>
          <logical-server-name>MyJ2EEServer2</logical-server-name>
          <logical-server-type>j2ee-server</logical-server-type>
        </define-server>
        <configuration>
          <param>
            <param-name>ejbserver.distributedtx.recovery.port</param-name>
            <param-value>20303</param-value>
          </param>
          <param>
            <param-name>ejbserver.http.port</param-name>
            <param-value>8081</param-value>
          </param>
        </configuration>
      </unit>
    </web-system>
  </unit-addition>

```

```

</param>
<param>
  <param-name>ejbserver.naming.port</param-name>
  <param-value>901</param-value>
</param>
<param>
  <param-name>ejbserver.rmi.naming.port</param-name>
  <param-value>23153</param-value>
</param>
<param>
  <param-name>webserver.connector.nio_http.port</param-name>
  <param-value>8008</param-value>
</param>
<param>
  <param-name>ejbserver.rmi.remote.listener.port</param-name>
  <param-value>23550</param-value>
</param>
</configuration>
</define-server>
<define-server>
  <logical-server-name>MyWebServer2</logical-server-name>
  <logical-server-type>web-server</logical-server-type>
<configuration>
  <param>
    <param-name>Listen</param-name>
    <param-value>81</param-value>
  </param>
</configuration>
</define-server>
</allocated-host>
</unit>
</web-system>
</unit-addition>

```

注 ホストの定義（<host>タグ）は削除してください。

3. コマンドプロンプトで cmx_change_model コマンドと cmx_build_system コマンドを実行し、手順 2.で編集した構成変更定義ファイルから論理サーバ（Web サーバ、J2EE サーバおよびパフォーマンストレーサ）を追加します。

コマンドの実行例を次に示します。

```

cmx_change_model -m localhost:28080 -u admin -f <構成変更定義ファイルのファイルパス>
cmx_build_system -m localhost:28080 -u admin -s MyWebSystem

```

<構成変更定義ファイルのファイルパス>には、手順 1.の構成変更定義ファイル（MyWebSystemAddUnit.xml）が格納されているディレクトリのパスとファイル名を指定します。

（3）運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(4) 追加した論理サーバの一括起動」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。

2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(4) 追加した論理サーバの一括起動

1. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括起動] タブをクリックします。
3. [一括起動] 画面で [実行] ボタンをクリックします。
一括起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」となります。

(5) 追加した J2EE サーバ (MyJ2EEServer2) へのリソースアダプタの設定と開始

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer2] – [リソース] – [リソースアダプタ] をクリックし、[インポート] タブをクリックします。
3. [リソースアダプタのインポート] 画面で MyJ2EEServer2 にインポートするリソースアダプタの情報を設定して、[実行] ボタンをクリックします。
4. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてからインポートし直してください。
5. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[プロパティ設定] タブをクリックします。
6. [リソースアダプタのプロパティ設定] 画面で手順 3.でインポートしたリソースアダプタの [操作] の [基本設定] アンカーをクリックします。
7. [リソースアダプタの基本設定] 画面で基本情報やプロパティを設定し、[設定] ボタンをクリックします。
8. [リソースアダプタの基本設定] 画面で表示されたメッセージを確認して [戻る] アンカーをクリックします。
失敗を示すメッセージが表示された場合は、障害要因を取り除いてからプロパティを設定し直してください。
9. [リソースアダプタの開始/停止] 画面で [操作] の [開始] アンカーをクリックします。
10. [リソースアダプタの開始] 画面で [はい] ボタンをクリックします。

11. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

(6) 追加した J2EE サーバへ (MyJ2EEServer2) の業務アプリケーションの設定と開始

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer2] – [アプリケーション] をクリックし、[インポート] タブをクリックします。
3. [J2EE アプリケーションのインポート] 画面で [J2EE アプリケーションファイル] から同一の J2EE アプリケーション (MyApp) を選択し、[実行] ボタンをクリックします。

参考

ここでは、追加した J2EE サーバ上でも同じ業務アプリケーションを動作させるため、アップロード済みの J2EE アプリケーションをインポートします。異なる業務アプリケーションを動作させる場合は、J2EE アプリケーションをアップロードしてからインポートしてください。

4. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてからインポートし直してください。

5. [開始/停止] タグをクリックします。

6. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、手順 3.でインポートしたアプリケーションの [操作] の [開始] アンカーをクリックします。

7. [J2EE アプリケーションの開始] 画面で [はい] ボタンをクリックします。

8. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「稼働」になります。

2.4.2 リソースアダプタを複数配置してマルチテナントに対応する

ここでは、単一の Web フロントシステムからデータベースを共有してマルチテナントに対応するシステムへ変更する手順について説明します。システム構成については、「[1.2.4 リソースアダプタを複数配置してマルチテナント対応するには](#)」を参照してください。

参考

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

リソースアダプタを複数配置するホストは、運用管理ポータルを使用してリソースアダプタを追加して作成します。また、リソースアダプタを利用する業務アプリケーションと関連づけます。例題で使用する設定内容を次の表に示します。

表 2-9 例題で使用する設定内容（リソースアダプタを複数配置してマルチテナントに対応する Web フロントシステム）

項目	設定例
追加するリソースアダプタの名称	DB_Connector_for_HiRDB_Type4_2
インポートする RAR ファイル	DBConnector_HiRDB_Type4_CP.rar

（1）運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「（2）リソースアダプタの追加」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

（2）リソースアダプタの追加

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] - [論理 J2EE サーバ] - [J2EE サーバ] - [MyJ2EEServer] - [リソース] - [リソースアダプタ] をクリックし、[インポート] タブをクリックします。
3. [リソースアダプタのインポート] 画面で MyJ2EEServer にインポートするリソースアダプタの情報を設定して、[実行] ボタンをクリックします。
インポートするリソースアダプタの情報として、DB Connector を選択して、プルダウンメニューからインポートする RAR ファイル (DBConnector_HiRDB_Type4_CP.rar) を選択し、リソースアダプタ名称を選択して、追加するリソースアダプタの名称 (DB_Connector_for_HiRDB_Type4_2) を指定します。
4. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてからインポートし直してください。

5. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし, [プロパティ設定] タブをクリックします。
6. [リソースアダプタのプロパティ設定] 画面で手順 3.でインポートしたリソースアダプタの [操作] の [基本設定] アンカーをクリックします。
7. [リソースアダプタの基本設定] 画面で基本情報やプロパティを設定し, [設定] ボタンをクリックします。
8. [リソースアダプタの基本設定] 画面で表示されたメッセージを確認して [戻る] アンカーをクリックします。
失敗を示すメッセージが表示された場合は, 障害要因を取り除いてからプロパティを設定し直してください。

(3) リソースアダプタの開始

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし, [開始/停止] タブをクリックします。
3. [リソースアダプタの開始/停止] 画面で [操作] の [開始] アンカーをクリックします。
4. [リソースアダプタの開始] 画面で [はい] ボタンをクリックします。
5. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は, [ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し, 障害要因を取り除いてから開始し直してください。

(4) 業務アプリケーションとの関連づけの設定

(a) java:comp/env でのルックアップ利用時

J2EE アプリケーションで lookup メソッドの引数に 「java:comp/env/<<res-ref-name>タグ指定値>」 と指定して実装している場合は, 運用管理ポータルを使用して, 属性を編集した J2EE アプリケーションを入れ替えてから開始します。

1. cosminexus.xml を編集して, <任意のディレクトリ>下の META-INF ディレクトリに格納します。
アプリケーション開発者にリソース参照の名前解決をするための情報を確認して, cosminexus.xml で <ejb-jar> タグ, <war> タグ, <entity> タグまたは <message> タグ内に次の指定をします。
EJB-JAR の属性 (<ejb-jar> タグ内) に指定する場合
 - <session> – <resource-ref> タグに, リソースの参照 (<res-ref-name> タグ) を指定します。
 - <session> – <resource-ref> – <linked-to> タグに, 関連づける DB Connector の表示名を指定します。

WARの属性（<war>タグ内）に指定する場合

- ・<resource-ref>タグに、リソースの参照（<res-ref-name>タグ）を指定します。
- ・<resource-ref>－<linked-to>タグに、関連づけるDB Connectorの表示名を指定します。

Entity Bean 属性の属性（<entity>タグ内）に指定する場合

- ・<resource-ref>タグに、リソースの参照（<res-ref-name>タグ）を指定します。
- ・<resource-ref>－<linked-to>タグに、関連づけるDB Connectorの表示名を指定します。

Message-driven Bean 属性の属性（<message>タグ内）に指定する場合

- ・<resource-ref>タグに、リソースの参照（<res-ref-name>タグ）を指定します。
- ・<resource-ref>－<linked-to>タグに、関連づけるDB Connectorの表示名を指定します。

2. コマンドプロンプトで、jar コマンドを実行し、アーカイブ形式の J2EE アプリケーションに手順 1.で編集した cosminexus.xml を設定します。

コマンドの実行例を次に示します。

```
jar uvf <MyApp1.earのファイルパス> -C <META-INFディレクトリを置いたディレクトリ> <META-INFディレクトリ下のcosminexus.xmlのファイルパス>
```

<MyApp1.ear のファイルパス>は、J2EE アプリケーション（MyApp1.ear）を格納したディレクトリのパスとファイル名（MyApp1.ear）を指定します。<META-INF ディレクトリを置いたディレクトリ>は、cosminexus.xml の格納先が次のような構造の場合、<任意のディレクトリ>となります。

```
<任意のディレクトリ>
  └META-INF
    └cosminexus.xml
```

<META-INF ディレクトリ下の cosminexus.xml のファイルパス>は、手順 1.で cosminexus.xml を格納した META-INF ディレクトリのパスとファイル名（cosminexus.xml）を指定します。

3. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
4. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[J2EE アプリケーションの開始/停止] タブをクリックします。
5. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、アプリケーションの表示名（MyApp）の [操作] の [停止] アンカーをクリックします。
6. [J2EE アプリケーションの停止] 画面で [はい] ボタンをクリックします。
7. [実行結果] を確認して [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから停止し直してください。
[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「停止」になります。

8. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし, [削除] タブをクリックします。
9. [J2EE アプリケーションの削除] 画面で [インポート J2EE アプリケーション一覧] にある, アプリケーションの表示名 (MyApp) の [操作] の [削除] アンカーをクリックします。
10. [J2EE アプリケーションの削除] 画面で [はい] ボタンをクリックします。
11. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は, [ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し, 障害要因を取り除いてから削除し直してください。
[実行結果] に「成功」が表示された場合は, [J2EE アプリケーションの削除] 画面の [インポート J2EE アプリケーション一覧] からアプリケーションが削除されます。
12. [DefaultDomain] をクリックします。
13. [J2EE アプリケーションのアップロード] 画面で [J2EE アプリケーションファイル] に手順 2.で再作成した J2EE アプリケーション (MyApp1.ear) を入力して, [アップロード] ボタンをクリックします。
14. 表示されたメッセージを確認して [戻る] アンカーをクリックします。

失敗を示すメッセージが表示された場合は, 障害要因を取り除いてからアップロードし直してください。
[アップロードディレクトリファイル一覧] に手順 13.でアップロードした J2EE アプリケーションが表示されます。
15. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし, [インポート] タブをクリックします。
16. [J2EE アプリケーションのインポート] 画面で [J2EE アプリケーションファイル] から手順 13.でアップロードしたファイルを選択し, [実行] ボタンをクリックします。
17. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は, [ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し, 障害要因を取り除いてからインポートし直してください。
18. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし, [開始/停止] タブをクリックします。
19. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある, 手順 16.でインポートしたアプリケーションの [操作] の [開始] アンカーをクリックします。
20. [J2EE アプリケーションの開始] 画面で [はい] ボタンをクリックします。
21. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は, [ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し, 障害要因を取り除いてから開始し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「稼働」になります。

(b) ユーザ指定名前空間機能を利用して付与した名称でのルックアップ利用時

J2EE アプリケーションで lookup メソッドの引数に任意の名称を指定して実装している場合は、運用管理ポータルを使用して、リソースアダプタの Connector 属性ファイルでリソースの別名を設定します。ここでは、任意の名称として「jdbc/DataSource」を指定している場合の例について説明します。

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[プロパティ設定] タブをクリックします。
3. [リソースアダプタのプロパティ設定] 画面で「(2) リソースアダプタの追加」でインポートしたりソースアダプタの [操作] の [Connector 属性ファイル編集] アンカーをクリックします。
4. [リソースアダプタの Connector 属性ファイル編集] 画面で<resource-external-property> – <optional-name>タグにリソースの別名を設定し、[設定] ボタンをクリックします。設定例を次に示します。ここでは、リソースの別名を「jdbc/DataSource」とします。

```
<connector-runtime>
  <property>
    :
  </property>
  <resource-external-property>
    <optional-name>jdbc/DataSource</optional-name>
  </resource-external-property>
</connector-runtime>
```

5. [リソースアダプタの基本設定] 画面で表示されたメッセージを確認して [戻る] アンカーをクリックします。

失敗を示すメッセージが表示された場合は、障害要因を取り除いてからプロパティを設定し直してください。

3

システムの運用

この章では、「2. システムの構築」で構築した Web フロントシステムを使用した起動や停止などの通常運用のほかに、アプリケーションの入れ替え、パフォーマンスチューニングなどの保守運用の手順について説明します。

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

3.1 アプリケーションサーバの通常運用

ここでは、次に示すアプリケーションサーバで実施する通常運用について説明します。

- ・システムの起動（業務の開始）
- ・システムの稼働状況の確認
- ・システムの停止（業務の終了）

3.1.1 システムを起動して業務を開始する（GUI 利用時）

停止中のシステムを起動して、業務（アプリケーション）を開始する手順について説明します。システムの起動とアプリケーションの開始には、運用管理ポータルを使用します。なお、運用管理機能（Management Server および運用管理エージェント）はホストの起動と同時に起動されます。ここでは、ホストは起動済みであるとします。

参考

説明で使用する画面については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

（1）運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は、「（2）論理サーバの起動」に進みます。

1. Web ブラウザを起動し、[http://localhost:28080/mngsvr/index.jsp] に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

（2）論理サーバの起動

1. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括起動] タブをクリックします。
3. [一括起動] 画面で [実行] ボタンをクリックします。

一括起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「停止」から「起動中」、「稼働中」へと遷移します。

注意事項

論理 J2EE サーバを停止する前に、リソースアダプタや業務アプリケーションを停止している場合は、論理サーバを起動したあとにリソースアダプタと業務アプリケーションを開始してください。

3.1.2 システムの稼働状況を確認する (GUI 利用時)

システムの運用を開始したあと、J2EE サーバなどの論理サーバが問題なく動作しているかを稼働状況で確認します。システムの稼働状況の確認には、運用管理ポータルを使用します。運用管理ポータルでは、運用管理ドメイン単位、ホスト単位および論理サーバ単位に稼働状況を確認できます。論理サーバ単位の場合は、論理サーバの種類の単位でも稼働状況を確認できます。

参考

説明で使用する画面については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は、「(2) 運用管理ドメイン内の論理サーバの稼働状況の確認」、「(3) ホスト内の論理サーバの稼働状況の確認」または「(4) 論理サーバごとの稼働状況の確認」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) 運用管理ドメイン内の論理サーバの稼働状況の確認

1. [運用管理ポータル] 画面で [論理サーバの運用監視] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] をクリックします。
[論理サーバのステータス監視] 画面に、運用管理ドメイン内のすべての論理サーバの稼働状況が表示されます。
3. [論理サーバのステータス監視] 画面で [<論理サーバ名>] アンカーをクリックします。
[<サーバの種類>の稼働情報監視] 画面に、選択した論理サーバの種類の稼働状況が表示されます。例えば、<論理サーバの種類>に「論理 J2EE サーバ」、<サーバの種類>に「J2EE サーバ」を選択すると、[J2EE サーバのステータス監視] 画面が表示されます。

(3) ホスト内の論理サーバの稼働状況の確認

1. [運用管理ポータル] 画面で [論理サーバの運用監視] をクリックします。
2. [ホストビュー] タブで [ホスト] – [<ホスト名>] をクリックします。
[論理サーバのステータス監視] 画面に、ホスト内のすべての論理サーバの稼働状況が表示されます。

(4) 論理サーバごとの稼働状況の確認

1. [運用管理ポータル] 画面で [論理サーバの運用監視] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] – [<論理サーバの種類>] – [<サーバの種類>] をクリックします。
[<サーバの種類>のステータス監視] 画面に、選択した論理サーバの種類の稼働状況が表示されます。例えば、<論理サーバの種類>に [論理 J2EE サーバ]、<サーバの種類>に [J2EE サーバ] を選択すると、[J2EE サーバのステータス監視] 画面が表示されます。
3. [<サーバの種類>のステータス監視] 画面で [<論理サーバ名>] アンカーをクリックします。
[<サーバの種類>の稼働情報監視] 画面に、選択した論理サーバの稼働状況が表示されます。例えば、<論理サーバの種類>に [論理 J2EE サーバ]、<サーバの種類>に [J2EE サーバ]、 [<論理サーバ名>] に [MyJ2EEServer] を選択すると、[J2EE サーバの稼働情報監視] 画面が表示されます。

参考

[<サーバの種類>の稼働情報監視] 画面は、[サーバビュー] タブで [DefaultDomain] – [<論理サーバの種類>] – [<サーバの種類>] – [<論理サーバ名>] をクリックしても表示できます。

3.1.3 システムを停止する (GUI 利用時)

稼働中の論理サーバおよび運用管理機能を停止して、アプリケーションサーバを停止する手順について説明します。システムの停止には、運用管理ポータルを使用します。なお、リソースアダプタやアプリケーションでは停止操作をしません。論理サーバの停止と同時に停止しておくと、論理サーバの起動と同時に開始できます。

参考

説明で使用する画面、コマンドの参照先、コマンドの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は、「(2) 論理サーバの停止」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) 論理サーバの停止

1. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括停止] タブをクリックします。
3. [一括停止] 画面で [実行] ボタンをクリックします。

一括停止要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」から「停止中」、「停止」へと遷移します。

(3) 運用管理機能の停止

1. 運用管理ポータルにログインしている場合は、[ログアウト] アンカーをクリックし、表示されたメッセージを確認して [ログイン画面] をクリックして、Web ブラウザを閉じます。
2. コマンドプロンプトで mngsvrctl コマンドのサブコマンド「stop」を実行し、Management Server を停止します。

コマンドの実行例を次に示します。

```
mngsvrctl stop -sync
```

3. コマンドプロンプトで adminagentctl コマンドのサブコマンド「stop」を実行し、運用管理エージェントを停止します。

コマンドの実行例を次に示します。

```
adminagentctl stop -sync
```

3.2 アプリケーションサーバの保守運用

ここでは、運用を開始したあとに、アプリケーションサーバで実施する次の保守運用について説明します。

- ・業務アプリケーションの入れ替え
- ・パフォーマンスチューニング
- ・IP アドレスの変更
- ・データベース接続環境の変更
- ・システムの構成・定義情報の取得
- ・不要な J2EE サーバの削除
- ・運用管理機能の管理ユーザアカウントの設定変更

3.2.1 業務アプリケーションを入れ替える

システムの運用を開始したあとで、業務アプリケーションのバージョンアップやメンテナンスを実施するために、「2.2.10 業務アプリケーションを設定して開始する (GUI 利用時)」で設定したアーカイブ形式の J2EE アプリケーションを入れ替える手順について説明します。

参考

説明で使用する画面については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

通常、J2EE アプリケーションを入れ替える場合には、J2EE サーバ上で動作している J2EE アプリケーションを停止したあと削除し、新しい J2EE アプリケーションをインポートして開始する必要があります。これらの作業には、運用管理ポータルを使用します。なお、説明で使用する業務アプリケーションは、cosminexus.xml を含めたアーカイブ形式の J2EE アプリケーションです。例題で使用する設定内容を次の表に示します。

表 3-1 例題で使用する設定内容 (業務アプリケーションの入れ替え)

項目	設定例
入れ替え前の J2EE アプリケーションのファイル名	MyApp1.ear
入れ替え後の J2EE アプリケーションのファイル名	MyApp2.ear
J2EE アプリケーションの表示名	MyApp

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(2) 入れ替え前の J2EE アプリケーションの停止」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) 入れ替え前の J2EE アプリケーションの停止

J2EE アプリケーションが停止済みの場合は「(3) 入れ替え前の J2EE アプリケーションの削除」に進みます。

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] - [論理 J2EE サーバ] - [J2EE サーバ] - [MyJ2EEServer] - [アプリケーション] をクリックし、[J2EE アプリケーションの開始/停止] タブをクリックします。
3. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、入れ替え前のアプリケーションの表示名 (MyApp) の [操作] の [停止] アンカーをクリックします。
4. [J2EE アプリケーションの停止] 画面で [はい] ボタンをクリックします。
5. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから停止し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「停止」になります。

(3) 入れ替え前の J2EE アプリケーションの削除

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] - [論理 J2EE サーバ] - [J2EE サーバ] - [MyJ2EEServer] - [アプリケーション] をクリックし、[削除] タブをクリックします。
3. [J2EE アプリケーションの削除] 画面で [インポート J2EE アプリケーション一覧] にある、入れ替え前のアプリケーションの表示名 (MyApp) の [操作] の [削除] アンカーをクリックします。
4. [J2EE アプリケーションの削除] 画面で [はい] ボタンをクリックします。
5. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから削除し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの削除] 画面の [インポート J2EE アプリケーション一覧] からアプリケーションが削除されます。

(4) 入れ替え後の J2EE アプリケーションの設定

1. アプリケーションサーバの<任意のディレクトリ>に、入手した入れ替え後の J2EE アプリケーション (MyApp2.ear) を格納します。
2. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
3. [DefaultDomain] をクリックします。
4. [J2EE アプリケーションのアップロード] 画面で [J2EE アプリケーションファイル] に手順 1.で<任意のディレクトリ>に格納した J2EE アプリケーション (MyApp2.ear) を入力して、[アップロード] ボタンをクリックします。
5. 表示されたメッセージを確認して [戻る] アンカーをクリックします。
失敗を示すメッセージが表示された場合は、障害要因を取り除いてからアップロードし直してください。
[アップロードディレクトリ一覧] に手順 4.でアップロードした J2EE アプリケーションのファイル名 (MyApp2.ear) が表示されます。
6. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[インポート] タブをクリックします。
7. [J2EE アプリケーションのインポート] 画面で [J2EE アプリケーションファイル] から手順 4.でアップロードしたファイル (MyApp2.ear) を選択し、[実行] ボタンをクリックします。
8. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてからインポートし直してください。

(5) 入れ替え後の J2EE アプリケーションの開始

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[開始/停止] タブをクリックします。
3. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、「(4) 入れ替え後の J2EE アプリケーションの設定」でインポートした J2EE アプリケーションの表示名 (MyApp) の [操作] の [開始] アンカーをクリックします。
4. [J2EE アプリケーションの開始] 画面で [はい] ボタンをクリックします。
5. [実行結果] を確認して [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「稼働」になります。

(6) 運用管理ポータルからのログアウト

1. [ログアウト] アンカーをクリックします。
2. 表示されたメッセージを確認して [ログイン画面] をクリックし、Web ブラウザを閉じます。

3.2.2 パフォーマンスチューニングとは

パフォーマンスチューニングとは、システムの処理性能や信頼性を高めるために、システムの動作環境を最適化することです。パフォーマンスチューニングを実施することで、システムの性能を最大限に生かし、安定稼働させられるようになります。

ここでは、パフォーマンスチューニングの代表的な次の項目について、パラメタの設定値が妥当であるかを検証し、適切なチューニングを実施して最適化する作業について説明します。

- JavaVM のメモリチューニング (Java ヒープおよび Explicit ヒープのチューニング)
GC の発生間隔や GC に掛かる時間は、Java ヒープのメモリサイズによって変動するため、この値を適切に見積もることでシステムの処理性能が向上します。また、適切な Explicit ヒープのメモリサイズを確保することで、FullGC の発生を防ぐことができます。JavaVM のメモリチューニングの考え方については、マニュアル「アプリケーションサーバ システム設計ガイド」の「7. JavaVM のメモリチューニング」を参照してください。
- 同時実行数の最適化
クライアントからのリクエスト数が増加して、システムの処理能力を超えると、安定したサービスを提供できなくなります。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ことができます。同時実行数の最適化の考え方については、マニュアル「アプリケーションサーバ システム設計ガイド」の「8.3 同時実行数を最適化する」を参照してください。

3.2.3 JavaVM のメモリチューニングを反映する

次に示す代表的な項目について、JavaVM のメモリサイズを設定する手順について説明します。

- Java ヒープ全体のメモリサイズ
- Explicit ヒープのメモリサイズ
- Metaspace 領域のメモリサイズ

参考

説明で使用する画面については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(2) Java ヒープ全体のメモリサイズの設定」、「(3) Explicit ヒープのメモリサイズの設定」または「(4) Metaspace 領域のメモリサイズの設定」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) Java ヒープ全体のメモリサイズの設定

1. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] をクリックし、[JVM] – [起動パラメタ] タブをクリックします。
3. [起動パラメタの設定] 画面で [メモリ使用量] に初期サイズと最大サイズを指定します。
4. [適用] ボタンをクリックします。

環境設定情報の保存を示すメッセージが表示され、Java ヒープ全体のメモリサイズが設定されます。

5. [設定情報の配布] アンカーをクリックします。

参考

ほかの項目を続けて設定する場合は、[戻る] アンカーをクリックすると、[起動パラメタの設定] 画面へ戻れます。

6. [設定情報の配布] 画面で [ステータス] が「空欄」の論理サーバの [配布] をチェックしてから、[選択して配布] ボタンをクリックします。
[ステータス] が「空欄」から「配布中」、「配布済」へと遷移します。

参考

[全て配布] ボタンをクリックすると、[ステータス] が「配布済」の論理サーバにも設定情報が再配布されます。

(3) Explicit ヒープのメモリサイズの設定

1. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] をクリックし, [JVM] – [起動パラメタ] タブをクリックします。
3. [起動パラメタの設定] 画面で [明示管理ヒープ機能] の Explicit ヒープ領域の最大サイズを指定します。
4. [適用] ボタンをクリックします。
環境設定情報の保存を示すメッセージが表示され, Explicit ヒープのメモリサイズが設定されます。
5. [設定情報の配布] アンカーをクリックします。

参考

ほかの項目を続けて設定する場合は, [戻る] アンカーをクリックすると, [起動パラメタの設定] 画面へ戻れます。

6. [設定情報の配布] 画面で [ステータス] が「空欄」の論理サーバの [配布] をチェックしてから, [選択して配布] ボタンをクリックします。
[ステータス] が「空欄」から「配布中」, 「配布済」へと遷移します。

参考

[全て配布] ボタンをクリックすると, [ステータス] が「配布済」の論理サーバにも設定情報が再配布されます。

(4) Metaspace 領域のメモリサイズの設定

1. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] をクリックし, [JVM] – [起動パラメタ] タブをクリックします。
3. [起動パラメタの設定] 画面で [拡張起動パラメタ] に次の拡張起動パラメタを設定して [有効] をチェックします。
 - -XX:MetaspaceSize=<値>
 - -XX:MaxMetaspaceSize=<値>
4. [適用] ボタンをクリックします。
環境設定情報の保存を示すメッセージが表示され, Metaspace 領域のメモリサイズが設定されます。
5. [設定情報の配布] アンカーをクリックします。

参考

ほかの項目を続けて設定する場合は、[戻る] アンカーをクリックすると、[起動パラメタの設定] 画面へ戻れます。

6. [設定情報の配布] 画面で [ステータス] が「空欄」の論理サーバの [配布] をチェックしてから、[選択して配布] ボタンをクリックします。

[ステータス] が「空欄」から「配布中」、「配布済」へと遷移します。

参考

[全て配布] ボタンをクリックすると、[ステータス] が「配布済」の論理サーバにも設定情報が再配布されます。

(5) 論理サーバの再起動

1. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。

2. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括再起動] タブをクリックします。

3. [一括再起動] 画面で [実行] ボタンをクリックします。

一括再起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」から「停止中」、「停止」、「起動中」、「稼働中」へと遷移します。

(6) 運用管理ポータルからのログアウト

1. [ログアウト] アンカーをクリックします。

2. 表示されたメッセージを確認して [ログイン画面] をクリックし、Web ブラウザを閉じます。

3.2.4 同時実行数を最適化するためのチューニングを反映する

次に示す代表的な項目について、同時実行数を設定する手順について説明します。

- Web コンテナ単位、Web アプリケーション単位の同時実行数
 - 同時に接続できるクライアントの最大数
 - Web サーバ起動時のサーバプロセス数
 - Web コンテナがリクエストを処理する同時実行数
 - デフォルトの実行待ちキューサイズ
 - Web コンテナ単位での最大同時実行スレッド数

- デフォルトの実行待ちキューサイズ
- Stateless Session Bean の同時実行数（プールで管理するインスタンスの最小値）
- プールの最小値
- プールの最大値
- URL グループ単位の同時実行数
 - Web アプリケーション単位での最大同時実行スレッド数
 - Web アプリケーションの占有スレッド数
 - Web アプリケーション単位の実行待ちキューサイズ
 - Web アプリケーション単位のリクエスト実行待ちキュー格納数の監視に関するプロパティ
 - URL グループ単位の同時実行スレッド数制御の定義名
 - URL グループ単位での最大同時実行スレッド数
 - URL グループ単位の占有スレッド数
 - URL グループ単位の実行待ちキューサイズ
 - URL グループ単位の制御対象となる URL パターン
 - URL グループ単位の実行待ちリクエスト数の監視に関するプロパティ

参考

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) Web コンテナまたは Web アプリケーション単位の同時実行数の設定

Web コンテナまたは Web アプリケーション単位の同時実行数を設定しない場合は「(2) URL グループ単位の同時実行数の設定」に進みます。

1. cosminexus.xml を編集して、<任意のディレクトリ>下の META-INF ディレクトリに格納します。
cosminexus.xml では、<ejb-jar>タグ内で次の指定をします。
 - <session>－<stateless>－<pooled-instance>－<minimum>タグ (Stateless Session Bean の同時実行数 (プールで管理するインスタンスの最小値)) を指定します。
 複数の<session>タグがある場合は、それぞれに指定します。
2. コマンドプロンプトで、jar コマンドを実行し、アーカイブ形式の J2EE アプリケーションに手順 1.で編集した cosminexus.xml を上書きします。
コマンドの実行例を次に示します。

3. システムの運用

```
jar uvf <MyApp1.earのファイルパス> -C <META-INFディレクトリを置いたディレクトリ> <META-INFディレクトリ下のcosminexus.xmlのファイルパス>
```

<MyApp1.ear のファイルパス>は、J2EE アプリケーション (MyApp1.ear) を格納したディレクトリのパスとファイル名 (MyApp1.ear) を指定します。<META-INF ディレクトリを置いたディレクトリ>は、cosminexus.xml の格納先が次のような構造の場合、<任意のディレクトリ>となります。

```
<任意のディレクトリ>
  └META-INF
    └cosminexus.xml
```

<META-INF ディレクトリ下の cosminexus.xml のファイルパス>は、手順 1.で cosminexus.xml を格納した META-INF ディレクトリのパスとファイル名 (cosminexus.xml) を指定します。

3. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
4. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。
5. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。
6. [サーバビュー] タブで [DefaultDomain] – [論理 Web サーバ] – [Web サーバ] – [MyWebServer] をクリックし、[設定] タブをクリックします。
7. [Web サーバの設定] 画面で次の値を設定して [適用] ボタンをクリックします。
 - [サーバスレッド数] (Windows の場合)
 - [同時接続クライアント数] (UNIX の場合)
 - [サーバ起動時のサーバプロセス数] (UNIX の場合)環境設定情報の保存を示すメッセージが表示され、値が設定されます。
8. [戻る] アンカーをクリックします。
9. [サーバビュー] タブで [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] をクリックし、[HTTP サーバ] – [通信・スレッド制御] タブをクリックします。
10. [通信・スレッド制御に関する設定] 画面で次の値を設定して [適用] ボタンをクリックします。
 - [最大接続数]
 - [最大スレッド数]環境設定情報の保存を示すメッセージが表示され、値が設定されます。

KEOS19237-W メッセージが出力されても無視し、次の手順に進んでください。このメッセージは、「HttpSession のセッション ID に付加するサーバ ID」にデフォルト値 (J2EE サーバが自動的に付加する値) が使用されることを示します。

参考

[通信・スレッド制御に関する設定] 画面で次の値を設定して [適用] ボタンをクリックすると、KEOS19237-W メッセージは出力されなくなります。

- ・「 HttpSession のセッション ID に付加するサーバ ID 」

11. [設定情報の配布] アンカーをクリックします。

参考

ほかの項目を続けて設定する場合は、[戻る] アンカーをクリックすると、[通信・スレッド制御に関する設定] 画面へ戻れます。

12. [設定情報の配布] 画面で [ステータス] が「空欄」の論理サーバの [配布] をチェックしてから、[選択して配布] ボタンをクリックします。

[ステータス] が「空欄」から「配布中」、「配布済」へと遷移します。

参考

[全て配布] ボタンをクリックすると、[ステータス] が「配布済」の論理サーバにも設定情報が再配布されます。

13. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。

14. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括再起動] タブをクリックします。

15. [一括再起動] 画面で [実行] ボタンをクリックします。

一括再起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」から「停止中」、「停止」、「起動中」、「稼働中」へと遷移します。

16. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。

17. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[開始/停止] タブをクリックします。

18. [リソースアダプタの開始/停止] 画面で [ステータス] が「稼働」の場合は [操作] の [停止] アンカーをクリックします。

19. [リソースアダプタの停止] 画面で [はい] ボタンをクリックします。

20. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから停止し直してください。

21. [プロパティ設定] タブをクリックし、[操作] の [基本設定] アンカーをクリックします。

22. [リソースアダプタの基本設定] 画面で次の値を設定して、[設定] ボタンをクリックします。
- プールの最小値 (MinPoolSize)
 - プールの最大値 (MaxPoolSize)
23. [リソースアダプタの基本設定] 画面で表示されたメッセージを確認して [戻る] アンカーをクリックします。
- 失敗を示すメッセージが表示された場合は、障害要因を取り除いてからプロパティを設定し直してください。
24. [開始/停止] タブをクリックし、[リソースアダプタの開始/停止] 画面で [操作] の [開始] アンカーをクリックします。
25. [リソースアダプタの開始] 画面で [はい] ボタンをクリックします。
26. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
- [実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

(2) URL グループ単位の同時実行数の設定

URL グループ単位の同時実行数を設定しない場合は「(3) 運用管理ポータルの起動とログイン」に進みます。

1. cosminexus.xml を編集して、<任意のディレクトリ>下の META-INF ディレクトリに格納します。
- cosminexus.xml では、<war> タグ内で次の指定をします。
- <thread-control> – <thread-control-max-threads> タグ (Web アプリケーション単位での最大同時実行スレッド数) を指定します。
 - <thread-control> – <thread-control-exclusive-threads> タグ (Web アプリケーションの占有スレッド数) を指定します。
 - <thread-control> – <thread-control-queue-size> タグ (Web アプリケーション単位の実行待ちキューサイズ) を指定します。
 - <thread-control> – <resource-watcher> タグ以下 (Web アプリケーション単位のリクエスト実行待ちキュー格納数の監視に関するプロパティ) を指定します。
 - <thread-control> – <urlgroup-thread-control> – <urlgroup-thread-control-name> タグ (URL グループ単位の同時実行スレッド数制御の定義名) を指定します。
 - <thread-control> – <urlgroup-thread-control> – <urlgroup-thread-control-max-threads> タグ (URL グループ単位での最大同時実行スレッド数) を指定します。
 - <thread-control> – <urlgroup-thread-control> – <urlgroup-thread-control-exclusive-threads> タグ (URL グループ単位の占有スレッド数) を指定します。

- ・<thread-control>－<urlgroup-thread-control>－<urlgroup-thread-control-queue-size>タグ (URL グループ単位の実行待ちキューサイズ) を指定します。
- ・<thread-control>－<urlgroup-thread-control>－<urlgroup-thread-control-mapping>タグ (URL グループ単位の制御対象となる URL パターン) を指定します。
- ・<thread-control>－<urlgroup-thread-control>－<stats-monitor>－<waiting-request-count>タグ以下 (URL グループ単位の実行待ちリクエスト数の監視に関するプロパティ) を指定します。

2. コマンドプロンプトで、jar コマンドを実行し、アーカイブ形式の J2EE アプリケーションに手順 1.で編集した cosminexus.xml を設定します。

コマンドの実行例を次に示します。

```
jar uvf <MyApp1.earのファイルパス> -C <META-INFディレクトリを置いたディレクトリ> <META-INFディレクトリ下のcosminexus.xmlのファイルパス>
```

<MyApp1.ear のファイルパス>は、J2EE アプリケーション (MyApp1.ear) を格納したディレクトリのパスとファイル名 (MyApp1.ear) を指定します。<META-INF ディレクトリを置いたディレクトリ>は、cosminexus.xml の格納先が次のような構造の場合、<任意のディレクトリ>となります。

```
<任意のディレクトリ>
  ↘META-INF
    ↘cosminexus.xml
```

<META-INF ディレクトリ下の cosminexus.xml のファイルパス>は、手順 1.で cosminexus.xml を格納した META-INF ディレクトリのパスとファイル名 (cosminexus.xml) を指定します。

(3) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(4) J2EE アプリケーションの設定変更」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(4) J2EE アプリケーションの設定変更

cosminexus.xml を編集して再作成した J2EE アプリケーション (MyApp1.ear) をアップロードしてインポートし直します。

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[J2EE アプリケーションの開始/停止] タブをクリックします。

3. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、入れ替え前のアプリケーションの表示名 (MyApp) の [操作] の [停止] アンカーをクリックします。
4. [J2EE アプリケーションの停止] 画面で [はい] ボタンをクリックします。
5. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから停止し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「停止」になります。
6. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[削除] タブをクリックします。
7. [J2EE アプリケーションの削除] 画面で [インポート J2EE アプリケーション一覧] にある、入れ替え前のアプリケーションの表示名 (MyApp) の [操作] の [削除] アンカーをクリックします。
8. [J2EE アプリケーションの削除] 画面で [はい] ボタンをクリックします。
9. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから削除し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの削除] 画面の [インポート J2EE アプリケーション一覧] からアプリケーションが削除されます。
10. [DefaultDomain] をクリックします。
11. [J2EE アプリケーションのアップロード] 画面で [J2EE アプリケーションファイル] に、「(1) Web コンテナまたは Web アプリケーション単位の同時実行数の設定」または「(2) URL グループ単位の同時実行数の設定」で再作成した J2EE アプリケーション (MyApp1.ear) を入力して、[アップロード] ボタンをクリックします。
12. 表示されたメッセージを確認して [戻る] アンカーをクリックします。

失敗を示すメッセージが表示された場合は、障害要因を取り除いてからアップロードし直してください。

[アップロードディレクトリファイル一覧] に手順 11.でアップロードした J2EE アプリケーションのファイル名 (MyApp1.ear) が表示されます。
13. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[インポート] タブをクリックします。
14. [J2EE アプリケーションのインポート] 画面で [J2EE アプリケーションファイル] から手順 11.でアップロードしたファイル (MyApp1.ear) を選択し、[実行] ボタンをクリックします。
15. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてからインポートし直してください。

16. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [アプリケーション] をクリックし、[開始/停止] タブをクリックします。
17. [J2EE アプリケーションの開始/停止] 画面で [インポート J2EE アプリケーション一覧] にある、手順 14.でインポートした J2EE アプリケーションの表示名 (MyApp) の [操作] の [開始] アンカーをクリックします。
18. [J2EE アプリケーションの開始] 画面で [はい] ボタンをクリックします。

19. [実行結果] を確認して [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

[実行結果] に「成功」が表示された場合は、[J2EE アプリケーションの開始/停止] 画面の [ステータス] が「稼働」になります。

(5) 運用管理ポータルからのログアウト

1. [ログアウト] アンカーをクリックします。
2. 表示されたメッセージを確認して [ログイン画面] をクリックし、Web ブラウザを閉じます。

3.2.5 IP アドレスを変更する

システムが稼働しているホストの IP アドレスを変更する手順について説明します。なお、システム稼働中に DHCP などを用いて IP アドレスを自動変更する運用はサポートしていません。

参考

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「[5. システムの構築・運用時に使用する画面、コマンドおよびファイル](#)」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(2) 論理サーバの停止」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。

2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) 論理サーバの停止

1. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
2. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括停止] タブをクリックします。
3. [一括停止] 画面で [実行] ボタンをクリックします。

一括停止要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「稼働中」から「停止中」、「停止」へと遷移します。

(3) 運用管理機能の停止

1. 運用管理ポータルにログインしている場合は、[ログアウト] アンカーをクリックし、表示されたメッセージを確認して [ログイン画面] をクリックして、Web ブラウザを閉じます。
2. コマンドプロンプトで mngautorun コマンドを実行し、Management Server および運用管理エージェントがホストの起動と同時に起動する設定を解除します。

コマンドの実行例を次に示します。

```
mngautorun -d
```

3. コマンドプロンプトで mngsvrctl コマンドのサブコマンド「stop」を実行し、Management Server を停止します。

コマンドの実行例を次に示します。

```
mngsvrctl stop -sync
```

4. コマンドプロンプトで adminagentctl コマンドのサブコマンド「stop」を実行し、運用管理エージェントを停止します。

コマンドの実行例を次に示します。

```
adminagentctl stop -sync
```

(4) IP アドレスの設定変更

1. OS の手順に従って、IP アドレスを変更します。再起動が必要な場合は、システムを再起動します。
2. Manager で使用するファイルで IP アドレスを設定している個所を、変更後の IP アドレスに変更します。
Manager で使用するファイルについては、マニュアル「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)」の「8. Manager で使用するファイル」を参照してください。
3. 次に示すディレクトリ下にあるファイルをすべて削除します。

Windows の場合

- <Application Server のインストールディレクトリ>/manager/tmp
- <Application Server のインストールディレクトリ>/manager/spool/lsinfo

UNIX の場合

- /opt/Cosminexus/manager/tmp
- /opt/Cosminexus/manager/spool/lsinfo

(5) システムの起動と設定情報の配布

1. コマンドプロンプトで adminagentctl コマンドのサブコマンド「start」を実行し、運用管理エージェントを起動します。

コマンドの実行例を次に示します。

```
adminagentctl start -sync
```

2. コマンドプロンプトで mngsvrctl コマンドのサブコマンド「start」を実行し、Management Server を起動します。

コマンドの実行例を次に示します。

```
mngsvrctl start -sync
```

3. コマンドプロンプトで mngautorun コマンドを実行し、ホストの起動と同時に Management Server および運用管理エージェントを起動するように設定します。

コマンドの実行例を次に示します。

```
mngautorun both -sync
```

4. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。

5. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

6. [運用管理ポータル] 画面で [論理サーバの環境設定] をクリックします。

7. [サーバビュー] タブで [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] をクリックし、[設定情報の配布] アンカーをクリックします。

8. [設定情報の配布] 画面で [ステータス] が「空欄」の論理サーバの [配布] をチェックしてから、[選択して配布] ボタンをクリックします。

[ステータス] が「空欄」から「配布中」、「配布済」へと遷移します。

参考

[全て配布] ボタンをクリックすると、[ステータス] が「配布済」の論理サーバにも設定情報が再配布されます。

9. [運用管理ポータル] 画面で [論理サーバの起動／停止] をクリックします。
10. [サーバビュー] タブで [DefaultDomain] をクリックし、[一括起動] タブをクリックします。
11. [一括起動] 画面で [実行] ボタンをクリックします。
一括起動要求が受け付けられ、[論理サーバの稼働状況] 画面の論理サーバの稼働状況が「停止」から「起動中」、「稼働中」へと遷移します。

(6) 運用管理ポータルからのログアウト

1. [ログアウト] アンカーをクリックします。
2. 表示されたメッセージを確認して [ログイン画面] をクリックし、Web ブラウザを閉じます。

3.2.6 データベース接続環境を変更する

システムの運用を開始したあとで、データベースのアクセス方法を最適化するためにデータベース接続環境をチューニングし、設定を変更する手順について説明します。データベース接続環境のチューニングの考え方については、マニュアル「アプリケーションサーバシステム設計ガイド」の「8.5 データベースへのアクセス方法を最適化する」を参照してください。

ここでは、次に示す代表的な項目について、コネクションプーリングのチューニングパラメタを設定する手順について説明します。

- コネクションプールにプールするコネクションの最小値
- コネクションプールにプールするコネクションの最大値

参考

説明で使用する画面については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(2) リソースアダプタの停止」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。

2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) リソースアダプタの停止

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[開始/停止] タブをクリックします。
3. [リソースアダプタの開始/停止] 画面で [ステータス] が「稼働」の場合は [操作] の [停止] アンカーをクリックします。
4. [リソースアダプタの停止] 画面で [はい] ボタンをクリックします。
5. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。
[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから停止し直してください。

(3) リソースアダプタのプロパティの設定変更

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[プロパティ設定] タブをクリックします。
3. [リソースアダプタのプロパティ設定] で [操作] の [基本設定] アンカーをクリックします。
4. [リソースアダプタの基本設定] 画面で次の値を設定して、[設定] ボタンをクリックします。
 - プールの最小値 (MinPoolSize)
 - プールの最大値 (MaxPoolSize)
5. [リソースアダプタの基本設定] 画面で表示されたメッセージを確認して [戻る] アンカーをクリックします。
失敗を示すメッセージが表示された場合は、障害要因を取り除いてからプロパティを設定し直してください。
6. [リソースアダプタの基本設定] 画面で [戻る] アンカーをクリックします。
[リソースアダプタのプロパティ設定] 画面に戻ります。

参考

[リソースアダプタの基本設定] 画面の画面項目にないチューニングパラメタを指定する場合は、[Connector 属性ファイル編集] アンカーをクリックし、[リソースアダプタの Connector 属性ファイル編集] 画面でファイルに直接パラメタを設定してください。

(4) リソースアダプタの開始

1. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。
2. [DefaultDomain] – [論理 J2EE サーバ] – [J2EE サーバ] – [MyJ2EEServer] – [リソース] – [リソースアダプタ] をクリックし、[開始/停止] タブをクリックします。
3. [リソースアダプタの開始/停止] 画面で [操作] の [開始] アンカーをクリックします。
4. [リソースアダプタの開始] 画面で [はい] ボタンをクリックします。
5. [実行結果] に「成功」が表示されたら [戻る] アンカーをクリックします。

[実行結果] に「失敗」が表示された場合は、[ログの表示] タブをクリックして [ログの表示] 画面のメッセージを確認し、障害要因を取り除いてから開始し直してください。

(5) 運用管理ポータルからのログアウト

1. [ログアウト] アンカーをクリックします。
2. 表示されたメッセージを確認して [ログイン画面] をクリックし、Web ブラウザを閉じます。

3.2.7 システムの構成・定義情報とは

システムの構成・定義情報とは、構築したシステムに配置している論理サーバの種類やリソースアダプタなどの構成を記述した定義ファイル、および運用管理機能の動作環境などを設定した定義ファイルのことです。代表的な定義ファイルを次に示します。

システムの構成に関するファイル

- 簡易構築定義ファイル
Smart Composer 機能のコマンドで構築する Web システムを定義するファイルです。また、リバースプロキシ、Web コンテナ間のリクエスト送受信時や、レスポンス送受信時の通信タイムアウトも設定します。
- Connector 属性ファイル
リソースアダプタの属性を定義するファイルです。
- cosminexus.xml (アプリケーション属性ファイル)

EJB-JAR, Session Bean, Entity Bean, Message-driven Bean, WAR, フィルタ, サーブレット/JSP およびリソースアダプタに関する、アプリケーションサーバ独自の定義情報を定義するファイルです。

運用管理機能の動作環境に関する定義ファイル

- manager.cfg (Manager 設定ファイル)

Management Server, 運用管理エージェント、およびそれらの上で動作するコンポーネントやコマンドに共通なプロパティを指定するファイルです。ログの出力先や面数などの共通な設定で使用します。

- adminagent.properties (運用管理エージェントプロパティファイル)

運用管理エージェントのプロパティを指定するファイルです。論理サーバの動作確認レベルやsnapshot ログの収集先の設定などで使用します。

- mserver.properties (Management Server 環境設定ファイル)

Management Server のプロパティを指定するファイルです。Management Server の環境設定や障害検知時コマンドの動作の設定などで使用します。

これらのファイルを、システムの構築時や設定変更時などのタイミングでアプリケーションサーバと別のホストに取得しておくと、障害発生時などに、以前の構成や定義情報の設定内容を確認したい場合などに使用できます。

3.2.8 システムの構成・定義情報を取得する

システムの構成・定義情報を取得する手順について説明します。

参考

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) snapshot ログによるシステムの定義情報の取得

1. コマンドプロンプトで mngsvrutil コマンドを実行し、システムの定義情報を取得します。

運用管理機能の動作環境に関する定義ファイルなどが取得できます。

コマンドの実行例を次に示します。

```
mngsvrutil -m <Management Serverのホスト名> -u admin -t <ホスト名> -k host collect snapshot
```

<Management Server のホスト名>には Management Server が稼働しているホスト名を、<ホスト名>にはアプリケーションサーバのホスト名を指定します。

取得される snapshot ログのファイル名は、表示されるメッセージで確認してください。snapshot ログのデフォルトの格納先を次に示します。

OS 種別	格納先
Windows	<Manager のログ出力ディレクトリ>※1¥snapshot¥snapshot-<ホスト名>-log-<日付※2>.zip
	<Manager のログ出力ディレクトリ>※1¥snapshot¥snapshot-<ホスト名>-log-<日付※2>.2.zip
UNIX	<Manager のログ出力ディレクトリ>※1/snapshot/snapshot-<ホスト名>-log-<日付※2>.zip
	<Manager のログ出力ディレクトリ>※1/snapshot/snapshot-<ホスト名>-log-<日付※2>.2.zip

注※1 <Manager のログ出力ディレクトリ>は、manager.cfg (Manager ログ設定ファイル) で指定されたディレクトリを指します。デフォルト値は、<Application Server のインストールディレクトリ>¥manager¥log です。

注※2 日付の形式は、次のどれかになります。

- yyyyMMddHHmmss
- yyyyMMddHHmmss+hhmm
- yyyyMMddHHmmss-hhmm

(2) 簡易構築定義ファイルの取得

1. コマンドプロンプトで cmx_export_model コマンドを実行し、簡易構築定義ファイルをエクスポートします。

コマンドの実行例を次に示します。

```
cmx_export_model -m localhost:28080 -u admin -o <簡易構築定義ファイルのファイルパス>
```

<簡易構築定義ファイルのファイルパス>には、簡易構築定義ファイルを格納するディレクトリのパスとファイル名を指定します。簡易構築定義ファイルは XML 形式のファイルです。

(3) Connector 属性ファイルの取得

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。

2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

3. [運用管理ポータル] 画面で [論理サーバのアプリケーション管理] をクリックします。

4. [DefaultDomain] - [論理 J2EE サーバ] - [J2EE サーバ] - [MyJ2EEServer] - [リソース] - [リソースアダプタ] をクリックし、[プロパティ設定] タブをクリックします。

5. [リソースアダプタのプロパティ設定] で [操作] の [Connector 属性ファイル編集] アンカーをクリックします。

6. [リソースアダプタの Connector 属性ファイル編集] 画面に表示されるファイルの内容をコピーして、<任意のディレクトリ>に<任意のファイル名>で保存します。
Connector 属性ファイルは XML 形式のファイルです。
7. [ログアウト] アンカーをクリックします。
8. 表示されたメッセージを確認して [ログイン画面] をクリックし、Web ブラウザを閉じます。

(4) アプリケーション属性ファイル (cosminexus.xml) の取得

1. <任意のディレクトリ>下の META-INF ディレクトリに格納してある cosminexus.xml を取得し、<任意のディレクトリ>に格納します。

3.2.9 不要になった J2EE サーバを削除する

システムの運用中に、業務規模を縮小したい場合やテナントを減らしたい場合など、不要になった J2EE サーバを削除する手順について説明します。

J2EE サーバは、Smart Composer 機能を使用して削除します。

参考

説明で使用するコマンドの参照先、コマンドの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 削除対象のシステム環境の内容確認

1. コマンドプロンプトで cmx_list_model コマンドを実行し、削除対象の名称を確認します。
コマンドの実行例を次に示します。

```
cmx_list_model -m localhost:28080 -u admin -modeltype SERVEREX -all
```

アプリケーションサーバに構築されている Web システム、論理サーバの一覧が 출력されます。
次に示す名称を確認してください。

- 削除対象の J2EE サーバが配置されている Web システムの名称
- 削除対象の J2EE サーバが配置されているサービスユニットの名称
- 削除対象の Web システム内の Web サーバの論理 Web サーバ名
- 削除対象の J2EE サーバの論理 J2EE サーバ名

(コマンドの実行結果の例)

```
MyPerformanceTracer, PRF, <ホスト名>, combined-tier, unit1, MyWebSystem
MyJ2EEServer, J2EEServer, <ホスト名>, combined-tier, unit1, MyWebSystem
```

```
MyWebServer, Webserver, <ホスト名>, combined-tier, unit1, MyWebSystem
MyPerformanceTracer2, PRF, <ホスト名>, combined-tier, unit2, MyWebSystem
MyJ2EEServer2, J2EEServer, <ホスト名>, combined-tier, unit2, MyWebSystem
MyWebServer2, Webserver, <ホスト名>, combined-tier, unit2, MyWebSystem
```

削除対象の J2EE サーバの論理 J2EE サーバ名を MyJ2EEServer2 とした場合、それぞれの名称は次のとおりです。

項目	名称
削除対象の J2EE サーバが配置されている Web システムの名称	MyWebSystem
削除対象の J2EE サーバが配置されているサービスユニットの名称	unit2
削除対象の Web システム内の Web サーバの論理 Web サーバ名	MyWebServer2

(2) 削除対象の Web サーバ、J2EE サーバおよびパフォーマンストレーサの停止

1. コマンドプロンプトで cmx_stop_target コマンドを実行し、削除対象の J2EE サーバが配置されているサービスユニット (unit2) を停止します。

コマンドの実行例を次に示します。

```
cmx_stop_target -m localhost:28080 -u admin -s MyWebSystem -unit unit2 -mode ALL
```

「(1) 削除対象のシステム環境の内容確認」で確認した削除対象の J2EE サーバが配置されている Web システムの名称とサービスユニットの名称を指定します。

(3) Web サーバ、J2EE サーバおよびパフォーマンストレーサの削除

1. コマンドプロンプトで cmx_delete_system コマンドを実行し、削除対象の J2EE サーバが配置されているサービスユニット (unit2) を削除します。

コマンドの実行例を次に示します。

```
cmx_delete_system -m localhost:28080 -u admin -s MyWebSystem -unit unit2
```

「(1) 削除対象のシステム環境の内容確認」で確認した削除対象の J2EE サーバが配置されている Web システムの名称とサービスユニットの名称を指定します。

2. コマンドプロンプトで hwsserveredit コマンドを実行し、Web サーバの環境を削除します。

コマンドの実行例を次に示します。

```
hwsserveredit -delete HWS_MyWebServer2
```

「(1) 削除対象のシステム環境の内容確認」で確認した削除対象の Web システム内の Web サーバの論理 Web サーバ名に「HWS_」を付けた形式で指定します。

3. コマンドプロンプトで cjsetup コマンドを実行し、J2EE サーバの環境を削除します。

コマンドの実行例を次に示します。

```
cjsetup -d MyJ2EEServer2
```

「(1) 削除対象のシステム環境の内容確認」で確認した削除対象の J2EE サーバの論理 J2EE サーバ名を指定します。

3.2.10 運用管理機能の管理ユーザーアカウントの設定を変更する

運用管理ポータルにログインする際に入力する管理ユーザアカウントの設定を変更する手順について説明します。Web フロントシステムでは、管理ユーザ ID のパスワードは設定されていません。ここでは、管理ユーザ ID のパスワード (adminpw) を設定する例を示します。

参考

説明で使用する画面については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

(1) 運用管理ポータルの起動とログイン

運用管理ポータルにログイン済みの場合は「(2) 管理ユーザアカウントの設定変更」に進みます。

1. Web ブラウザを起動し、「<http://localhost:28080/mngsvr/index.jsp>」に接続して、運用管理ポータルを起動します。
 2. [Management Server ログイン] 画面で管理ユーザ ID に「admin」を入力して、[ログイン] ボタンをクリックします。

(2) 管理ユーザーアカウントの設定変更

1. [運用管理ポータル] 画面で [Cosminexus Management Server の設定] をクリックします。
 2. [Cosminexus Management Server の設定] – [管理ユーザーアカウントの設定] をクリックします。
 3. [管理ユーザーアカウントの設定] 画面で [パスワード] と [パスワードの確認入力] に「adminpw」を指定し, [適用] ボタンをクリックします。
失敗を示すメッセージが表示された場合は, 障害要因を取り除いてから管理ユーザーアカウントを設定し直してください。
 4. [ログイン画面へ戻る] アンカーをクリックします。

次回から [Management Server ログイン] 画面では、管理ユーザ ID に「admin」、パスワードに「adminpw」を入力して、[ログイン] ボタンをクリックします。

3.3 システム環境のアンセットアップと Application Server のアンインストール

ここでは、運用中の障害などによってシステム環境を再構築するために、既存のシステム環境をアンセットアップしたり、Application Server をアンインストールしたりする作業について説明します。

3.3.1 システム環境をアンセットアップする

不要になったシステム環境をアンセットアップする手順について説明します。システム環境のアンセットアップには、セットアップウィザードを使用します。

参考

説明で使用するコマンドの参照先、コマンドの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

アンセットアップを実施すると、次の処理が実行されます。

- 運用管理機能の停止と削除、および設定情報の初期化
- システム環境（論理サーバ）の停止と削除
- 運用管理ユーザ（Management Server の管理ユーザアカウント）の設定解除
- J2EE サーバの起動ユーザ（Component Container 管理者）の設定解除（UNIX の場合）

1. コマンドプロンプトで setupwizard コマンドを実行し、不要なシステム環境を削除します。

コマンドの実行例を次に示します。

```
setupwizard -clean
```

アンセットアップを実行するかどうかを確認するメッセージが表示されます。

2. 「y」を入力して、Enter キーをクリックします。

アンセットアップ処理の実行中を示すメッセージが表示されたあと、完了を示すメッセージが表示されます。

3.3.2 Application Server をアンインストールする（Windows の場合）

アプリケーションサーバに接続して、Application Server をアンインストールします。

注意事項

Application Server 以外に不要な製品があれば、その製品もアンインストールします。

アンインストールの方法には次の 2 種類があります。

- ・製品のアンインストーラを利用する場合
- ・プログラムのアンインストールまたは変更を利用する場合

製品の構成ソフトウェアすべて、または構成ソフトウェアを選択してアンインストールできます。アンインストール作業には Administrator 権限が必要です。アンインストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を、アンインストール時のエラーコードについては、「[付録 I.6 Application Server のインストールおよびアンインストール時のエラーコード一覧 \(Windows の場合\)](#)」を参照してください。

注意事項

Windows のメニュー名の表記は、使用する Windows のバージョンによって異なります。ご使用の OS に合わせて読み替えて操作してください。

(1) 製品のアンインストーラを利用する場合

製品のアンインストーラを利用する場合の手順を次に示します。ここでは、製品の構成ソフトウェアすべてをアンインストールする手順を説明します。

手順

1. 構成ソフトウェアのすべてのプロセスが実行中でないことを確認します。実行中のプロセスがあれば停止します。
2. スタートメニューから、[Cosminexus] – [uCosminexus Application Server アンインストール] を選択します。
アンインストールする構成ソフトウェアを確認する画面（「uCosminexus Application Server をすべてアンインストールしますか？ [いいえ] を選択するとアンインストールするプログラムを選択できます。」）が表示されます。
3. [はい] ボタンをクリックします。
[アンインストールの開始] ダイアログが表示されます。

参考

製品の構成ソフトウェアを選択してアンインストールすることもできます。この場合は、[いいえ] ボタンをクリックし、表示される [プログラムの選択] ダイアログで、アンインストールする構成ソフトウェア（プログラム）を選択して、[次へ] ボタンをクリックします。

製品に対応している構成ソフトウェアについては、「2.2.1(3) 製品に対応する構成ソフトウェアについて」を参照してください。

4. 設定内容を確認して、問題がなければ [次へ] ボタンをクリックします。

アンインストールが開始されます。アンインストールが完了すると、[セットアップの完了] ダイアログが表示されます。

5. [完了] ボタンをクリックします。

Application Server のアンインストールが完了します。

(2) プログラムのアンインストールまたは変更を利用する場合

プログラムのアンインストールまたは変更を利用する場合の手順を次に示します。ここでは、製品の構成ソフトウェアすべてをアンインストールする手順を説明します。

手順

1. 構成ソフトウェアのすべてのプロセスが実行中でないことを確認します。実行中のプロセスがあれば停止します。

2. スタートメニューから、[コントロールパネル] – [プログラムと機能] を選択します。

[プログラムのアンインストールまたは変更] ダイアログが表示されます。

3. [uCosminexus Application Server] を選択して、右クリックで表示される [アンインストールと変更] ボタンをクリックします。

アンインストールする構成ソフトウェアを確認する画面（「uCosminexus Application Server をすべてアンインストールしますか？ [いいえ] を選択するとアンインストールするプログラムを選択できます。」）が表示されます。

4. [はい] ボタンをクリックします。

[アンインストールの開始] ダイアログが表示されます。

参考

製品の構成ソフトウェアを選択してアンインストールすることもできます。この場合は、[いいえ] ボタンをクリックし、表示される [プログラムの選択] ダイアログで、アンインストールする構成ソフトウェア（プログラム）を選択して、[次へ] ボタンをクリックします。

製品に対応している構成ソフトウェアについては、「2.2.1(3) 製品に対応する構成ソフトウェアについて」を参照してください。

5. 設定内容を確認して、問題がなければ [次へ] ボタンをクリックします。

アンインストールが開始されます。アンインストールが完了すると、[セットアップの完了] ダイアログが表示されます。

6. [完了] ボタンをクリックします。

Application Server のアンインストールが完了します。

3.3.3 Application Server をアンインストールする (AIX の場合)

アプリケーションサーバに接続して、Application Server をアンインストールします。

注意事項

Application Server 以外に不要な製品があれば、その製品もアンインストールします。

アンインストールでは、製品の提供媒体である PP インストーラを使用します。

アンインストールの手順を次に示します。アンインストール作業には root 権限が必要です。アンインストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を参照してください。

手順

- 構成ソフトウェアのすべてのプロセスが実行中でないことを確認します。実行中のプロセスがあれば停止します。
- アプリケーションサーバに root 権限（スーパーユーザ）でログインします。
- 次のコマンドを入力して、PP インストーラを起動します。

```
/etc/hitachi_setup
```

- PP インストーラのメインメニューで、[D] キーを押します。

PP 削除画面が表示されます。

- プログラムにカーソルを移動させ、[スペース] キーを押します。

すべてのプログラム (Application Server に対応している構成ソフトウェア、および製品の共通モジュール) を選択します。選択したプログラムの左側には<@>が表示されます。なお、プログラムを選択してアンインストールすることもできます。

- すべてのプログラムの左側に<@>が表示されていることを確認して、[D] キーを押します。

画面の最下行に「Delete PP? (y: delete, n: cancel)==>」メッセージが表示されます。

- [y] キーまたは [Y] キーを押します。

アンインストールが開始されます。

[n] キーまたは [N] キーを押すと、アンインストールが中止されて PP 削除画面に戻ります。

- インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

PP インストーラのメインメニューに戻ります。

9. PP インストーラのメインメニューで、[Q] キーを押します。

Application Server のアンインストールが完了します。

3.3.4 Application Server をアンインストールする (Linux の場合)

アプリケーションサーバに接続して、Application Server をアンインストールします。

注意事項

Application Server 以外に不要な製品があれば、その製品もアンインストールします。

アンインストールでは、製品の提供媒体である PP インストーラを使用します。

アンインストールの手順を次に示します。アンインストール作業には root 権限が必要です。アンインストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を参照してください。

手順

1. 構成ソフトウェアのすべてのプロセスが実行中でないことを確認します。実行中のプロセスがあれば停止します。
2. アプリケーションサーバに root 権限（スーパーユーザ）でログインします。
3. 次のコマンドを入力して、PP インストーラを起動します。

```
/etc/hitachi_x64setup
```

4. PP インストーラのメインメニューで、[D] キーを押します。

PP 削除画面が表示されます。

5. プログラムにカーソルを移動させ、[スペース] キーを押します。

すべてのプログラム（Application Server に対応している構成ソフトウェア、および製品の共通モジュール）を選択します。選択したプログラムの左側には<@>が表示されます。なお、プログラムを選択してアンインストールすることもできます。

6. すべてのプログラムの左側に<@>が表示されていることを確認して、[D] キーを押します。

画面の最下行に「Delete PP? (y: delete, n: cancel)==>」メッセージが表示されます。

7. [y] キーまたは [Y] キーを押します。

アンインストールが開始されます。

[n] キーまたは [N] キーを押すと、アンインストールが中止されて PP 削除画面に戻ります。

8. インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

PP インストーラのメインメニューに戻ります。

9. PP インストーラのメインメニューで、[Q] キーを押します。

Application Server のアンインストールが完了します。

3.4 システムの運用時の注意事項

ここでは、システムの運用時に注意することについて説明します。

3.4.1 J2EE サーバを運用管理するときの注意事項

(1) RMI レジストリに関する注意事項

Management Server は、J2EE サーバの運用監視に、J2EE サーバ上で動作する RMI レジストリを使用します。そのため、Management Server が管理している RMI レジストリの情報と実際に動作している J2EE サーバの RMI レジストリの情報が一致していないと、運用監視に失敗する場合があります。Management Server の管理している RMI レジストリの情報は、次のタイミングで更新されます。J2EE サーバ上で動作する RMI レジストリと異ならないように注意してください。

- ・運用管理ポータルの「Management Server の設定」で構成情報の回復を実行した場合
- ・運用管理ポータルの「論理サーバの環境設定」で【適用】ボタンを押した場合
- ・運用管理ポータルの「論理サーバの環境設定」からサーバの設定の読み込みを実行した場合
- ・cmx_build_model.exe コマンドを実行した場合
- ・cmx_change_model.exe コマンドを実行した場合

運用管理ポータルの操作については、マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」の次の個所を参照してください。

- ・8. Management Server の設定
- ・10. 論理サーバの環境設定

各コマンドの詳細については、マニュアル「アプリケーションサーバ リファレンス コマンド編」の次の個所を参照してください。

- ・cmx_build_model (Web システムの情報モデルの生成)
- ・cmx_change_model (Web システムの情報モデルの変更)

3.4.2 コマンドで指定する名称に関する注意事項

運用作業でコマンドを実行する際、コマンドによって次の四つの異なる名称を指定します。各名称の対応を確認するには、コマンドや簡易構築定義ファイルを使用する必要があります。

- ・Web システム名
Smart Composer 機能のコマンドに指定します。

例：cmx_start_target -s <Web システム名>

- サービスユニット名

Smart Composer 機能のコマンドに指定します。

例：cmx_start_target -s <Web システム名> -unit <サービスユニット名>

- 論理サーバ名

運用管理コマンド（mngsvrutil）に指定します。

例：mngsvrutil start -t <論理サーバ名>

- 実サーバ名

サーバ管理コマンドに指定します。

例：cjstartsv <実サーバ名>

各名称の対応は、次の方法で確認してください。

1. cmx_list_model コマンドを実行することで、次の名称の対応を確認します。

- ホスト名
- Web システム名
- サービスユニット名
- 論理サーバ名

実行例

```
cmx_list_model -m localhost:28080 -u admin -p admin -modeltype SERVEREX
cmx_sample-system_unit1_PRF_01,PerformanceTracer,host1,combined-tier,unit1,sample-system
cmx_sample-system_unit1_J2EE_01,J2EEServer,host1,combined-tier,unit1,sample-system
cmx_sample-system_unit1_HTTP_01,WebServer,host1,combined-tier,unit1,sample-system
cmx_sample-system_unit2_PRF_01,PerformanceTracer,host2,combined-tier,unit2,sample-system
cmx_sample-system_unit2_J2EE_01,J2EEServer,host2,combined-tier,unit2,sample-system
cmx_sample-system_unit2_HTTP_01,WebServer,host2,combined-tier,unit2,sample-system
```

2. 簡易構築定義ファイルで、次の名称の対応を確認します。

- 論理サーバ名（logical-server-name）

論理サーバ名が定義されていない場合は、自動生成されます。

- 実サーバ名（realservername）

実サーバ名が定義されていない場合は、論理サーバ名と同じになります。

3. システムの運用

3.4.3 ネットワークドライブへアクセスするときの制限事項 (Windows の場合)

Windows の場合、次の表に示す構成ソフトウェアは、ネットワークドライブへアクセスするときに制限が発生します。構成ソフトウェアごとの制限事項を次の表に示します。

また、Windows の仕様によって、サービスから起動するプログラムは、ネットワークドライブへのアクセスに制限がありますので、注意してください。

表 3-2 ネットワークドライブへアクセスするときの制限事項

構成ソフトウェア	制限事項
Component Transaction Monitor	Windows のサービスを利用して Component Transaction Monitor を起動する場合は、CTMSPOOL 環境変数にネットワークドライブ上のパスを指定しないでください。
HTTP Server	HTTP Server からネットワークドライブにアクセスできません。
Performance Tracer	Windows のサービスを利用して Performance Tracer を起動する場合は、PRFSPOOL 環境変数にネットワークドライブ上のパスを指定しないでください。
Reliable Messaging	次に示す Reliable Messaging のコンフィグレーションプロパティには、ネットワークドライブ上のファイルやディレクトリを指定しないでください。 <ul style="list-style-type: none">• QueueConfigFileName• QueueMakeFileName• RMTRTransferControlDir
TPBroker	ORB 機能で使用する環境変数やプロパティに、ネットワークドライブまたはネットワークドライブ上のファイルを指定しないでください。
HiRDB/Single Server Version 10	電源入力時は、ネットワークドライブは接続されていないため、Windows のサービスを利用して起動できません。このため、ネットワークドライブへ HiRDB/Single Server Version 10 をインストールしたり、データベースの環境を構築したりしないでください。

3.4.4 一時ファイルに関する注意事項

アプリケーションサーバでは、プログラム動作中に、システム共通の一時ディレクトリ※に一時ファイルを作成します。

そのため、プログラムを正常に動作させるためには、一時ファイルの作成先に書き込み権限が必要となります。また、プログラム動作中は、一時ファイルの作成先および一時ファイルを削除しないでください。

注※

Windows :

TMP もしくは TEMP 環境変数で指定されたディレクトリか、または OS がユーザに割り当てた一時ディレクトリ

UNIX :

/tmp ディレクトリ

3.4.5 システム時刻を補正するときの注意事項

NTP プログラム（Windows の場合は Windows Time サービス、UNIX の場合は ntpd コマンド、ntpdate コマンドなど）を使用してシステム時刻を補正する場合は、次の点に注意して運用してください。なお、システム時刻補正については、使用している OS のマニュアルを参照してください。

- ・時刻飛びが発生しないように時刻のずれを徐々に補正する、slew モードの範囲で製品を使用してください。
- ・一度に数十秒以上の大幅な時刻補正が必要な場合は、製品のプロセスを停止してから時刻補正を実施するようにしてください。
- ・時刻補正によってシステム時刻が戻った場合または進んだ場合、次のような事象が発生することがあります。これらの事象を考慮して、システム設計やタイムアウト設計をしてください。
 - ・トランザクションや通信、システム監視などで、タイムアウトが想定より遅くなったり早くなったりする
 - ・ログや履歴の時系列が崩れる
- ・一部の構成ソフトウェアでは、機能の性質上、時刻が戻ると動作が不正になることがあります。

4

運用管理ポータルまたは Smart Composer 機能で実現する運用形態

この章では、運用管理ポータルまたは Smart Composer 機能で実現するシステムを構築および運用する手順について説明します。

説明で使用する画面、コマンドおよびファイルの参照先、コマンドおよびファイルの格納先については、「5. システムの構築・運用時に使用する画面、コマンドおよびファイル」を参照してください。

4.1 Web サーバを別ホストに配置してマシン性能を向上するシステムの構築

ここでは、Web サーバを配置するホストと、J2EE サーバを配置するホストを分けたシステムを構築する手順について説明します。

4.1.1 Web サーバを別ホストに配置するには

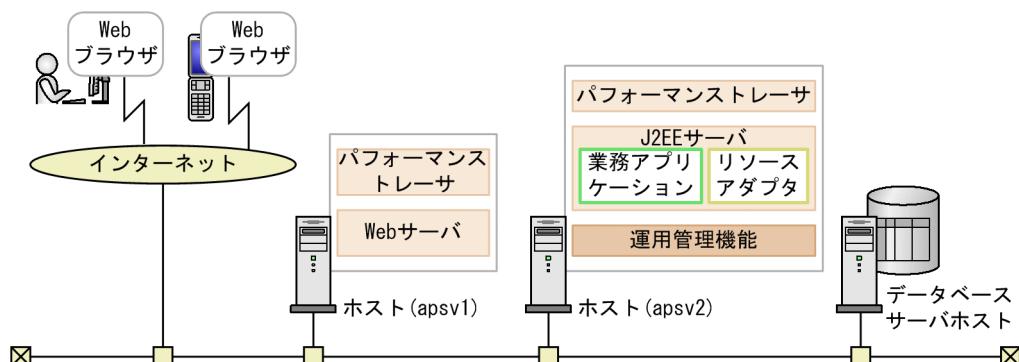
Web サーバを別ホストに配置するシステムでは、HTML ファイルなどの静的なコンテンツを配置するホストと、業務アプリケーションを実行するホストを分けることで、パフォーマンスの向上が図れます。

Smart Composer 機能の場合は、Web サーバを配置するホスト (http-tier) および J2EE サーバを配置するホスト (j2ee-tier) や論理サーバの環境設定を定義した簡易構築定義ファイルから、コマンドを使用して構築します。また、運用管理ポータルの場合は、[運用管理ポータル] 画面の各画面で、それぞれのホストに論理サーバ (Web サーバや J2EE サーバなど) を配置して、各論理サーバの設定情報をホストに配布して構築します。

- システム構成例

Web サーバを別ホストに配置するシステムのシステム構成例を次の図に示します。なお、このシステムでは、運用管理機能である Management Server は、J2EE サーバを配置するホストに配置します。また、この構成例では、ネットワークは単一構成 (業務 LAN と管理 LAN を分けない構成) とし、各ホストは物理 NIC を一つ使用するものとします。

図 4-1 Web サーバを別ホストに配置するシステムのシステム構成例



例題で使用する設定内容を次の表に示します。

表 4-1 例題で使用する設定内容 (Web サーバを別ホストに配置するシステムの構築)

分類	項目	設定例
Web サーバを配置するホスト	ホストの名称	apsv1
	IP アドレス (管理 IP アドレス) ※1	192.168.100.11
	論理 Web サーバの名称	MyWebSystem_HTTP01
	論理パフォーマンストレーサの名称	MyWebSystem_PRF01

分類	項目	設定例
J2EE サーバを配置するホスト ^{※2}	ホストの名称	apsv2
	IP アドレス (管理 IP アドレス) ^{※1}	192.168.100.12
	論理 J2EE サーバの名称	MyWebSystem_J2EE01
	論理パフォーマンストレーサの名称	MyWebSystem_PRF02
	J2EE アプリケーションのファイル名	MyApp.ear
	J2EE アプリケーションの表示名	MyApp
	運用管理機能の管理ユーザ ID	admin
	運用管理機能の管理ユーザパスワード	admin

注※1 ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

注※2 運用管理機能 (Management Server) は、J2EE サーバを配置するホストに配置します。

・実施する作業と作業の要否

Web サーバを別ホストに配置するシステムの構築時に実施する作業と作業の要否を、利用する構築ツールごとに次の表に示します。

表 4-2 Web サーバを別ホストに配置するシステムの構築時に実施する作業と作業の要否 (運用管理ポータル利用時)

Web サーバを別ホストに配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
データベースの準備	△	データベースに接続する場合	このマニュアル	2.1.1
データベースの環境の設定 (Oracle の設定)	△	データベース (Oracle) に接続する場合		4.1.2
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき		4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	このマニュアル	2.2.2
		必ず実施 (OS が AIX の場合)		2.2.3
		必ず実施 (OS が Linux の場合)		2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	このマニュアル	2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合		4.1.5

Web サーバを別ホストに配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合		4.1.6
		データベース (Oracle) に接続する場合		4.1.7
		データベース (SQL Server) に接続する場合		4.1.8
		データベース (MySQL) に接続する場合		4.1.9
		データベース (PostgreSQL) に接続する場合		4.1.10
システムの環境変数の設定	○	必ず実施		4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合		4.1.14
運用管理機能の構築	○	必ず実施	運用管理ポータル	2.2
運用管理エージェントの起動	○	必ず実施		4.1.2
Management Server の起動	○	必ず実施		4.1.2
システム環境の設定	○	必ず実施		3.2
システム環境のセットアップ	○	必ず実施		
システムの起動 (論理サーバの起動)	○	必ず実施		4.1.2
リソースアダプタの設定	△	リソースアダプタ (DB Connector) によるリソース接続をする場合		3.3
		リソースアダプタ (DB Connector 以外) によるリソース接続をする場合		
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合		4.1.2
業務アプリケーションの設定と開始	○	必ず実施		3.4, 4.1.2
システム環境定義の変更	△	運用で必要な場合		4.2
業務アプリケーションの設定変更	△	運用で必要な場合		4.2, 4.3
リソースアダプタの設定変更	△	運用で必要な場合		4.2
システムの停止 (論理サーバの停止)	△	運用で必要な場合		4.1.4
Management Server の停止	△	運用で必要な場合		4.1.4

Web サーバを別ホストに配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
運用管理エージェントの停止	△	運用で必要な場合		4.1.4
システム環境の削除	△	運用で必要な場合		3.6

(凡例)

運用管理ポータル：マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」を示します。

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

表 4-3 Web サーバを別ホストに配置するシステムの構築時に実施する作業と作業の要否 (Smart Composer 機能利用時)

Web サーバを別ホストに配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
データベースの準備	△	データベースに接続する場合	2.1.1
データベースの環境の設定 (Oracle の設定)	△	データベース (Oracle) に接続する場合	4.1.2
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき	4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合	4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合	4.1.6
		データベース (Oracle) に接続する場合	4.1.7
		データベース (SQL Server) に接続する場合	4.1.8
		データベース (MySQL) に接続する場合	4.1.9
		データベース (PostgreSQL) に接続する場合	4.1.10
システムの環境変数の設定	○	必ず実施	4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合	4.1.14
運用管理機能の構築	○	必ず実施	4.1.16

Web サーバを別ホストに配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
運用管理エージェントの起動	○	必ず実施	4.1.19
Management Server の起動	○	必ず実施	4.1.20
システム環境の設定（簡易構築定義ファイルの設定）	○	必ず実施	4.1.22
システム環境の構築（セットアップ）	○	必ず実施	4.1.25
システムの起動	○	必ず実施	4.1.26
リソースアダプタの設定	△	リソースアダプタ（DB Connector）によるリソース接続をする場合	4.1.28
		リソースアダプタ（DB Connector 以外）によるリソース接続をする場合	4.1.29
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合	4.1.30
業務アプリケーションの設定と開始	○	必ず実施	4.1.31
システム環境定義の変更	△	運用で必要な場合	4.1.33
業務アプリケーションの設定変更	△	運用で必要な場合	4.1.34
リソースアダプタの設定変更	△	運用で必要な場合	4.1.35
システムの停止	△	運用で必要な場合	4.1.36
Management Server の停止	△	運用で必要な場合	4.1.37
運用管理エージェントの停止	△	運用で必要な場合	4.1.38
システム環境の削除	△	運用で必要な場合	4.1.39

（凡例）

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

4.1.2 データベースの環境を設定する（Oracle の設定）

ここでは、Oracle JDBC Thin Driver を使用する場合に、データベース側で必要な設定について説明します。詳細については、Oracle のマニュアルを参照してください。

- トランザクションリカバリを行うために付与する権限

グローバルトランザクションを使用する場合は、トランザクションリカバリを行うために、Oracle の接続ユーザに次の権限を付与する必要があります。

- SYS.DBA_PENDING_TRANSACTIONS の SELECT 権限

- FORCE ANY TRANSACTION 権限
- SYS.DBMS_SYSTEM を EXECUTE する権限

権限を付与する方法については、Oracle のマニュアルを参照してください。

• 動的パフォーマンスビューの参照権の設定

Oracle に接続するユーザは、動的パフォーマンスビューの参照権を持っている必要があります。次のどちらかの方法で Oracle に接続するユーザに動的パフォーマンスビューの参照権を設定してください。

- 「GRANT SELECT_CATALOG_ROLE TO <ユーザ名>;」を実行する。
 - 「GRANT SELECT ON V_\$INSTANCE TO <ユーザ名>;」, 「GRANT SELECT ON V_\$PROCESS TO <ユーザ名>;」, および「GRANT SELECT ON V_\$SESSION TO <ユーザ名>;」を実行する。
- #### • タイムアウトの設定

Oracle のサーバ定義の DISTRIBUTED_LOCK_TIMEOUT パラメタに設定します。

なお、このほか XAOpenString の SesTm パラメタの設定がタイムアウトに影響しますが、このパラメタはチューニングできません。

詳細については、次の個所またはマニュアルを参照してください。

- マニュアル「アプリケーションサーバ システム設計ガイド」の「8.6.6 データベースでのタイムアウトを設定する」
- マニュアル「アプリケーションサーバ システム設計ガイド」の「8.6.8 タイムアウトを設定するチューニングパラメタ」
- Oracle のマニュアル

4.1.3 Component Container 管理者を設定する (UNIX の場合)

UNIX の場合、サーバのセットアップ、起動と停止などはスーパーユーザで実施しますが、スーパーユーザ以外の一般ユーザでも権限を与えることで実施できるようになります。このユーザを Component Container 管理者といいます。これによって、スーパーユーザ以外のユーザでもシステムを構築、運用できるようになります。この設定は、各ホスト (J2EE サーバを配置するホスト) で実施します。

ここでは、Component Container 管理者を設定する手順について説明します。Component Container 管理者の設定する場合は新規インストールまたは更新インストールの前後で作業が必要になります。

Component Container 管理者を設定するときの注意事項については、「[4.1.4 Component Container 管理者を設定するときの注意事項 \(UNIX の場合\)](#)」を参照してください。

1. アプリケーションサーバに root 権限 (スーパーユーザ) でログインします。
 2. Component Container 管理者を登録するグループ (Component Container 管理者グループ) を登録します。
- グループの登録方法については、OS のドキュメントを参照してください。

3. 手順 2.で登録したグループに、Component Container 管理者を登録します。

管理者の登録方法については、OS のドキュメントを参照してください。

4. 新規インストールまたは更新インストールを実施します。

新規インストールの場合

AIX のときは「[2.2.3 Application Server を新規インストールする \(AIX の場合\)](#)」を、Linux のときは「[2.2.4 Application Server を新規インストールする \(Linux の場合\)](#)」を参照してください。

更新インストールの場合

「[付録 F.2 Application Server を更新インストールする \(UNIX の場合\)](#)」を参照してください。

5. Component Container のインストールディレクトリの所有者とグループを、手順 2.および手順 3.で登録した Component Container 管理者と Component Container 管理者グループに変更します。

6. cjenvsetup コマンドを実行して、Component Container 管理者をセットアップします。

Component Container 管理者の設定が完了します。設定完了後は、Component Container 管理者が実施できる操作は、一部の操作を除いてスーパーユーザで操作できなくなります。J2EE サーバやバッチサーバのセットアップ、起動と停止などの操作は Component Container 管理者が実施してください。

4.1.4 Component Container 管理者を設定するときの注意事項 (UNIX の場合)

Component Container 管理者の設定後、Component Container 管理者が実施できる操作は、一部の操作を除き、スーパーユーザでは操作できなくなるので注意してください。

Component Container 管理者を設定した場合に、スーパーユーザおよび Component Container 管理者が実施できる操作を次の表に示します。

表 4-4 Component Container 管理者の設定がある場合に実施できる操作

操作	スーパーユーザ	Component Container 管理者
Component Container のインストール	○	×
Component Container 管理者のセットアップ (cjenvsetup コマンド)	○	×
作業ディレクトリおよびユーザ定義ファイルの移行 (cjenvupdate コマンド、cjrarupdate コマンド)	×	○
J2EE サーバまたはバッチサーバのセットアップ／アンセットアップ (cjsetup コマンド)	×	○
J2EE サーバまたはバッチサーバの起動／停止 (cjstartsv コマンド、cjstopsv コマンド)	×	○
サーバ管理コマンドの実行	○	○

操作	スーパーユーザ	Component Container 管理者
スレッドダンプの取得 (cjdumpserv コマンド)	○	○
OS 状態情報の取得 (cjgetsysinfo コマンド)	○	○
バッチアプリケーションの実行 (cjexecjob コマンド) *	○	○
バッチアプリケーションの強制停止 (cjkilljob コマンド) *	○	○
バッチアプリケーション情報の一覧表示 (cjlistjob コマンド) *	○	○

(凡例) ○: 実行できます。 ×: 実行できません。

注※ バッチアプリケーションの場合に使用できるコマンドです。バッチアプリケーションは、バッチサーバ上で動作するアプリケーションです。

なお、Component Container 管理者を設定しない場合は、スーパーユーザが運用管理者となるため、Component Container 管理者の操作もスーパーユーザが実施します。

参考

Component Container 管理者を設定していても、運用管理コマンドの実行には影響ありません。

Management Server のセットアップ時に設定した管理ユーザ ID で、システム運用時の操作およびコマンドを実行できます。

Component Container 管理者の設定、および Component Container 管理者による運用に関する注意事項を次に示します。

- Component Container 管理者のアカウント

Component Container 管理者のアカウントは、Component Container をインストールするホストに一つだけしか登録できないため、複数のアカウントを Component Container 管理者として登録できません。なお、Component Container 管理者として使用できるアカウントは、OS にログインできるアカウントに限定されます。

- Component Container 管理者の管理対象のサーバ

cjsetup コマンドでセットアップする J2EE サーバやバッチサーバだけでなく、インストール時にデフォルトでセットアップされるホスト名のサーバについても、Component Container 管理者が所有者となります。サーバごとに異なる Component Container 管理者を設定することはできません。

- Component Container 管理者で運用する場合のポート番号の変更

Component Container 管理者で CORBA ネーミングサービスを起動する場合（インプロセスで起動する場合も含みます）には、簡易構築定義ファイルでポート番号の設定を変更します。論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に、ejbserver.naming.port パラメタで、ポート番号に 1,024 番以上を設定してください。CORBA ネーミングサービスをデフォルトのポート番号 (900 番)、または 0~1,023 番のポート番号で起動する場合には、スーパーユーザで手動起動してください。

なお、root 権限を持たない Component Container 管理者では、0～1,023 番のポート番号を開けません。このため、Component Container 管理者を設定して運用する場合には、CORBA ネーミングサービス以外のポート番号についても、1,024 番以上に変更してください。

- **Component Container のインストールディレクトリ属性の変更**

Component Container のインストールディレクトリの属性（所有者／グループ）を変更した場合、Component Container 管理者セットアップコマンドの実行が完了するまでサーバを操作（セットアップ／アンセットアップ、および起動・停止）しないでください。

- **Component Container 管理者の変更**

Component Container 管理者を設定していた場合は、更新インストール後も同じユーザが Component Container 管理者となります。Component Container 管理者を変更したい場合には、Component Container のインストールディレクトリ（/opt/Cosminexus/CC）の所有者およびグループを変更して、Component Container 管理者をセットアップしてください。

Component Container 管理者をスーパーユーザに戻す場合も、同様の方法で変更できます。

- **ディレクトリおよびファイルの所有者の変更**

簡易構築定義ファイルの論理 J2EE サーバ（j2ee-server）の<configuration>タグ内では、次に示すディレクトリを変更して使用できます。

変更できるディレクトリ	指定方法
製品の JavaVM（以降、JavaVM と呼びます）ログ出力先ディレクトリ	JavaVM 起動パラメタを指定します。 JavaVM 起動パラメタの指定内容を次に示します。 <param-name>タグ add.jvm.arg <param-value>タグ -XX:HitachiJavaLog:<ディレクトリ名>
J2EE サーバまたはバッチサーバのログ出力ディレクトリ	ejb.server.log.directory パラメタで指定します。
JSP 用一時ディレクトリ	webserver.work.directory パラメタで指定します。
TPBroker のトレース出力先ディレクトリ	vbroker.orb.htc.tracePath パラメタで指定します。
稼働情報ファイルの出力先ディレクトリ	ejbserver.management.stats_file.dir パラメタで指定します。

これらのディレクトリを変更して使用している場合には、Component Container 管理者のセットアップ後に、次のどちらかを実行してください。

- 各ディレクトリおよびファイルの所有者を Component Container 管理者に変更する。
- 各ディレクトリおよびファイルに、Component Container 管理者の書き込み権限を設定する。

- **Component Container 管理者を設定している場合のディレクトリの属性**

Component Container 管理者を設定した場合、次に示すディレクトリ下のサブディレクトリおよびファイルの属性（所有者／グループ）が「Component Container 管理者/Component Container 管理者グループ」に変更されます。

- /opt/Cosminexus/CC

- ・<作業ディレクトリ>/ejb/
- ・<作業ディレクトリ>/web/

ただし、/opt/Cosminexus/CC 下の一部のファイル (/opt/Cosminexus/CC/server/bin/cjenvsetup, /opt/Cosminexus/CC/server/bin/cjgetsysinfo や、インストールログなど) の属性は変更されません。

- **EJB クライアントアプリケーションのログ出力先ディレクトリのアクセス権の設定**

Component Container 管理者を設定している場合、Component Container のインストールディレクトリ (/opt/Cosminexus/CC) 以下のファイルおよびディレクトリの属性（所有者／グループ）が「Component Container 管理者／Component Container 管理者グループ」に変更されます。このため、EJB クライアントアプリケーションのログ出力先にデフォルトの /opt/Cosminexus/CC/client/logs を使用していて、かつ Component Container 管理者と EJB クライアントアプリケーションの実行ユーザのアカウントが異なる場合、EJB クライアントアプリケーションのログ出力時にエラーとなることがあります。この場合、ログ出力先ディレクトリに出力されたログファイルのモードを、グループやほかのすべてのユーザに対して書き込みを許可するように設定してください。アクセス権の設定については、マニュアル「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)」の「3.8.5 ログ出力先ディレクトリのアクセス権の設定」を参照してください。

- **運用監視エージェントのログファイルのアクセス権の設定**

運用監視エージェントは、「/opt/Cosminexus/manager/log」ディレクトリに、次に示す名称のログファイルを出力します。

- mngagent-<ドメイン名>-<Agent 名>.[n].log

注 ファイル名の[n]の部分には、1～指定したログの面数の通し番号が付きます。

Component Container 管理者を設定したあとに、次のどちらかを実行してください。

- ログファイルの所有者を Component Container 管理者に変更する。
- ログファイルに、Component Container 管理者の書き込み権限を設定する。

- **Component Container 管理者を設定している場合のメッセージ**

Component Container 管理者を設定している場合、アプリケーションおよび各種リソースをインポートするときに、インポート対象ファイルの読み取り権限がないために次のメッセージが表示されることがあります。このような場合は、インポート対象ファイルのアクセス権を確認してください。

アクセス権の設定が不正な場合、メッセージに、KDJE37324-E が表示されます。

- **Component Container 管理者を設定している場合の J2EE サーバまたはバッチサーバのセットアップ**

Component Container 管理者を設定している場合、Management Server のサーバ名と同じ名前で J2EE サーバやバッチサーバをセットアップしないでください。なお、Component Container は、インストール時にホスト名で J2EE サーバやバッチサーバをセットアップしているため、Management Server のサーバ名をホスト名でセットアップしないでください。

Component Container 管理者のセットアップ (cjenvsetup コマンド) の実行情報はメッセージログおよびテキストファイルに出力されます。実行情報の出力内容と出力先を次に示します。

表 4-5 Component Container 管理者のセットアップ (cjenvsetup コマンド) の実行情報の出力内容と出力先一覧

コマンド実行情報	出力内容	出力先ファイル名※1
コマンド実行メッセージ	次のメッセージが出力されます。 <ul style="list-style-type: none"> • KDJE40074-I • KDJE40040-I (正常の場合) • KDJE40041-E (異常の場合) 	cjenvsetupmessage*.log
変更前ファイル (ディレクトリ) 情報	コマンド実施前の所有者変更対象のファイル (ディレクトリ) ※2 情報の一覧※3。	before_cjenvsetup_files*.txt
変更後ファイル (ディレクトリ) 情報	コマンド実施後の所有者変更対象のファイル (ディレクトリ) 情報※2 の一覧※3。 なお、エラーが発生した場合は、エラー発生時点 (コマンド終了時点) の途中結果が出力されます。	after_cjenvsetup_files*.txt

注※1

出力ディレクトリは、<Component Container のインストールディレクトリ>/logs/です。

注※2

所有者変更対象となるファイル (ディレクトリ) は、<Component Container のインストールディレクトリ>下、<作業ディレクトリ>/ejb 下、および<作業ディレクトリ>/web 下の各ファイルです。なお、シンボリックリンクファイルのリンク先がディレクトリの場合は、シンボリックリンク先のディレクトリ下のファイルも対象となります。

注※3

1行目にファイル (ディレクトリ) 一覧出力日時、2行目以降にファイル (ディレクトリ) の一覧が表示されます。出力例を次に示します。

```
Mon Dec 10 12:00:00 JST 2012 ← ファイル(ディレクトリ)一覧出力日時
/opt/Cosminexus/CC:           ← ファイル(ディレクトリ)一覧
  total 16
  drwxr-xr-x  2 0      0          256 Dec 10 11:00 adapters
  drwxr-xr-x  2 0      0          256 Dec 10 11:00 admin
  .
  .
```

なお、ファイル (ディレクトリ) の一覧情報の書式は OS のバージョン、設定によって異なります。

4.1.5 アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)

UNIX の場合、次のディレクトリのシンボリックリンクをサポートしています。

- /opt/Cosminexus/CC

- /opt/Cosminexus/CCR
- /opt/Cosminexus/manager
- /opt/Cosminexus/c4web

ただし、これらのディレクトリより下位のディレクトリのシンボリックリンク、およびこれら以外のディレクトリのシンボリックリンクは未サポートとなります。

展開ディレクトリ形式の J2EE アプリケーションのアプリケーションディレクトリ、およびその下位ディレクトリのシンボリックリンクは未サポートとなります。

4.1.6 データベース接続環境を設定する (HiRDB の設定)

J2EE サーバ上での J2EE アプリケーションからデータベースへ接続するためには、リソースアダプタとして DB Connector を使用します。また、DB Connector を使用してデータベースに接続するためには、データベースに対応する JDBC ドライバが必要です。

ここでは、DB Connector を使用して HiRDB に接続する際に必要なデータベース接続環境の設定について説明します。JDBC ドライバは HiRDB Type4 JDBC Driver を使用します。この設定は、J2EE サーバを配置するホストで実施します。

なお、データベースを設定するときの注意事項については、「[4.1.11 データベースを設定するときの注意事項](#)」を参照してください。

(1) HiRDB Type4 JDBC Driver の準備

ここでは、HiRDB Type4 JDBC Driver の準備について説明します。

1. HiRDB Type4 JDBC Driver の JAR ファイルを用意します。

JAR ファイルについては、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

2. 簡易構築定義ファイルに、手順 1. で用意した HiRDB Type4 JDBC Driver の JAR ファイルのパスを設定します。

簡易構築定義ファイルを作成するときに、物理ティアの定義（<tier> タグの定義）の J2EE サーバの定義で、<configuration> タグに次の設定を追加してください。

<param-name> タグの指定値	<param-value> タグの指定値
add.class.path	<JAR ファイルのクラスパス>

簡易構築定義ファイルの作成については、「[4.1.22 簡易構築定義ファイルを設定する](#)」を参照してください。設定内容の詳細については、マニュアル「[アプリケーションサーバリファレンス 定義編\(サーバ定義\)](#)」の「[4.3 簡易構築定義ファイル](#)」を参照してください。

(2) HiRDB クライアント環境変数グループ設定ファイルの作成

HiRDB クライアント環境変数グループを使用するには、HiRDB のクライアント環境変数グループの設定ファイルを作成します。この場合、DB Connector のプロパティ定義内の HiRDB クライアント環境変数グループ名を指定する部分には、ここで作成した設定ファイルのパスを指定してください。設定ファイルの作成方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

なお、グローバルトランザクションを使用する場合には、必ず HiRDB クライアント環境変数グループを設定する必要があります。グローバルトランザクションを使用しない場合には、HiRDB クライアント環境変数グループの使用は任意です。

(3) システムプロパティの設定

HiRDB の設定は、システムプロパティとしても設定できます。プロパティ名は、"HiRDB_for_Java_"で始まります。J2EE サーバでのシステムプロパティは、J2EE サーバ用ユーザプロパティファイル (usrconf.properties) に設定します。

(例) HiRDB_for_Java_UPDATECOUNT_BEHAVIOR=TRUE

HiRDB のシステムプロパティの詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

なお、Connector 属性ファイルや HiRDB クライアント環境変数グループ設定ファイルで設定できるプロパティについては、システムプロパティでは設定しないでください。

(4) タイムアウトの設定

HiRDB のタイムアウトは、次に示す HiRDB のクライアント環境変数を、DB Connector の environmentVariables プロパティまたは HiRDB のクライアント環境変数グループの設定ファイルに追加することで設定します。

- PDSWAITTIME
- PDCWAITTIME
- PDNBLOCKWAITTIME

なお、コネクション確立時のタイムアウトは、DB Connector の loginTimeout プロパティに設定します。ただし、HiRDB Type4 JDBC Driver の setLoginTimeout メソッドに値を設定した場合でも、ログイン時のタイムアウトができないことがあります。

DB Connector のプロパティは、システム構築前、またはシステム構築後のリソースアダプタのデプロイ時に設定します。リソースアダプタのデプロイについては、「[4.1.28 DB Connector を設定する \(CUI 利用時\)](#)」または「[4.1.29 DB Connector 以外のリソースアダプタを設定する \(CUI 利用時\)](#)」を参照してください。

また、タイムアウトの設定の詳細については、マニュアル「アプリケーションサーバ システム設計ガイド」の「8.6.6 データベースでのタイムアウトを設定する」を参照してください。DB Connector のプロパティの設定については、マニュアル「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「4.2.2 DB Connector のプロパティ定義」を参照してください。

4.1.7 データベース接続環境を設定する (Oracle の設定)

J2EE アプリケーションからデータベースへ接続するためには、リソースアダプタとして DB Connector を使用します。また、DB Connector を使用してデータベースに接続するためには、データベースに対応する JDBC ドライバが必要です。

ここでは、DB Connector を使用して Oracle に接続する際に必要なデータベース接続環境の設定について説明します。JDBC ドライバは Oracle JDBC Thin Driver を使用します。この設定は、J2EE サーバを配置するホストで実施します。

なお、データベースを設定するときの注意事項については、「[4.1.11 データベースを設定するときの注意事項](#)」を参照してください。

(1) Oracle JDBC Thin Driver の準備

ここでは、Oracle JDBC Thin Driver を使用する場合に、J2EE サーバ側で必要な設定について説明します。なお、Oracle JDBC Thin Driver を使用する場合、J2EE サーバに追加する環境変数はありません。

アプリケーションサーバがサポートするバージョンの、Oracle JDBC Thin Driver の JAR ファイルを入手してください。Oracle JDBC Thin Driver の入手方法の具体例としては、J2EE サーバを動作させるマシンに Oracle クライアントをインストールする方法があります。Oracle クライアントのインストール手順の概略を次に示します。

1. J2EE サーバを動作させるマシンに、アプリケーションサーバがサポートするバージョン番号の Oracle クライアントをインストールします。

このとき、Oracle JDBC Thin Driver もインストールされます。

2. アプリケーションサーバがサポートするパッチレベル（パッチ番号）のパッチを適用します。

3. 簡易構築定義ファイルに、Oracle JDBC Thin Driver の JAR ファイルのパスを指定します。

簡易構築定義ファイルを作成するときに、物理ティアの定義（<tier> タグの定義）の J2EE サーバの定義で、<configuration> タグに Oracle JDBC Thin Dirver の JAR ファイルのパスを追加してください。簡易構築定義ファイルの作成については、「[4.1.22 簡易構築定義ファイルを設定する](#)」を参照してください。設定内容の詳細については、マニュアル「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）」の「[4.3 簡易構築定義ファイル](#)」を参照してください。

4. J2EE サーバに、Oracle JDBC Thin Driver 用の DB Connector をインポートします。

DB Connector のインポートについては、「[4.1.28 DB Connector を設定する \(CUI 利用時\)](#)」を参照してください。

Oracle JDBC Thin Driver 用の DB Connector は、2 種類あります。トランザクションの管理方法によって、次のどちらかのファイルを選択してください。

DBConnector_Oracle_CP.rar

ローカルトランザクション、またはトランザクション管理なしで使用する場合（トランザクションのサポートレベルに LocalTransaction または NoTransaction を指定する場合）に選択します。Oracle JDBC Thin Driver の ConnectionPoolDataSource を使用してデータベースに接続します。

DBConnector_Oracle_XA.rar

グローバルトランザクションを使用する場合（トランザクションのサポートレベルに XATransaction を指定する場合）に選択します。Oracle JDBC Thin Driver の XADataSource を使用してデータベースに接続します。

トランザクションの管理方法の詳細については、マニュアル「[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「[3.4 トランザクション管理](#)」を参照してください。

4.1.8 データベース接続環境を設定する (SQL Server の設定)

J2EE アプリケーションからデータベースへ接続するためには、リソースアダプタとして DB Connector を使用します。また、DB Connector を使用してデータベースに接続するためには、データベースに対応する JDBC ドライバが必要です。

ここでは、DB Connector を使用して SQL Server に接続する際に必要なデータベース接続環境の設定について説明します。JDBC ドライバは SQL Server の JDBC ドライバを使用します。この設定は、J2EE サーバを配置するホストで実施します。

なお、データベースを設定するときの注意事項については、「[4.1.11 データベースを設定するときの注意事項](#)」を参照してください。

(1) データベース側での設定

DB Connector を使用して SQL Server に接続する場合は、データベースのマシン上に SQL Server をインストールして初期設定を済ませておいてください。詳細については、SQL Server のマニュアルを参照してください。また、SQL Server に接続するためには、認証モードの設定が必要です。認証モードはインストール時に指定しますが、インストール後の変更もできます。

アプリケーションサーバで設定できる SQL Server の認証モードは、設定できる認証方法によって、混合モードと Windows 認証モードの 2 種類に分かれています。認証モード、および認証方法についてそれぞれ説明します。

認証モード

- 混合モード

Windows 認証と SQL Server 認証の両方の認証方法を使用できる認証モードです。

- **Windows 認証モード**

Windows 認証を使用できる認証モードです。

認証方法

- **Windows 認証**

SQL Server のインスタンスへの接続の試行を検証するメカニズムの一つです。ユーザが、接続時に Windows のユーザまたはグループによって識別されます。

- **SQL Server 認証**

SQL Server のインスタンスへの接続の試行を検証するメカニズムの一つです。接続するには、SQL Server ログイン ID およびパスワードを指定する必要があります。SQL Server インスタンスでは、ログイン ID とパスワードの組み合わせが有効であることを確認してから接続します。

(2) J2EE サーバ側での設定 (SQL Server JDBC Driver の場合)

アプリケーションサーバがサポートするバージョンの SQL Server JDBC Driver の JAR ファイルを入手して、J2EE サーバのクラスパスに指定してください。JAR ファイルの入手方法とクラスパスの指定方法について説明します。

1. Microsoft 社のホームページからアプリケーションサーバがサポートするバージョン番号の SQL Server JDBC Driver をダウンロードします。

2. J2EE サーバを動作させるマシンに、ダウンロードした SQL Server JDBC Driver をインストールします。

3. J2EE サーバのユーザクラスパスに JAR ファイルを指定します。

指定する JAR ファイルは、<SQL Server JDBC Driver のインストールディレクトリ>¥sqljdbc_<バージョン>¥<language>ディレクトリ下にある JDBC ドライバの JAR ファイルです。

簡易構築定義ファイルでは、物理ティアの定義（<tier> タグの定義）の J2EE サーバの定義で、<configuration> タグに JAR ファイルの設定を追加します。なお、指定内容については、マニュアル「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)」の「4.3 簡易構築定義ファイル」を参照してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<model-definition xmlns="http://www.cosminexus.com/mngsvr/schema/ModelDefinition-2.5">
  <!-- Webシステムの属性定義 -->
  <web-system>
    <name>MyWebSystem</name>
    :
    <!-- 物理ティア(combined-tier)の定義 -->
    <tier>
      <tier-type>combined-tier</tier-type>
      <!-- SQLサーバを使用するための設定 -->
      <configuration>
        <logical-server-type>j2ee-server</logical-server-type>
        <param>
          <param-name>add.class.path</param-name>
        
```

```

<param-value><SQL Server JDBC DriverのJARファイルのパスを指定></param-value>
<!-- Windows認証を使用するための設定 -->
<param-name>add.library.path</param-name>
<param-value><SQL Server JDBC Driverのdllファイルを格納したディレクトリのパス
を指定></param-value>
</param>
</configuration>
</tier>
:
</web-system>
<host>
:
</model-definition>

```

注 背景色付きの太字の部分が編集部分です。

Windows 認証を使用する場合は、JAR ファイルの設定に加えて、次の設定も必要となります。

Windows 認証を使用する場合に必要となる設定について説明します。

4. J2EE サーバのコンテナ拡張ライブラリの JNI 用ライブラリの検索パスに dll ファイルが格納されているディレクトリを指定します。

指定するディレクトリは、<SQL Server JDBC Driver のインストールディレクトリ>¥sqljdbc_<バージョン>¥<language>¥auth¥<x64>ディレクトリです。

5. J2EE サーバに、SQL Server 用の DB Connector (DBConnector_SQLServer_CP.rar) をインポートします。

DB Connector の詳細については、マニュアル「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「3.6 データベースへの接続」を参照してください。

cjimportres コマンドを使用して、DBConnector_SQLServer_CP.rar をインポートしてください。

(3) タイムアウトの設定

必要に応じて、SQL Server のタイムアウトを設定してください。タイムアウトの設定については、マニュアル「アプリケーションサーバ システム設計ガイド」の「8.6.6 データベースでのタイムアウトを設定する」を参照してください。

4.1.9 データベース接続環境を設定する (MySQL の設定)

J2EE サーバ上の J2EE アプリケーションからデータベースへ接続するためには、リソースアダプタとして DB Connector を使用します。また、DB Connector を使用してデータベースに接続するためには、データベースに対応する JDBC ドライバが必要です。

ここでは、DB Connector を使用して MySQL に接続する際に必要なデータベース接続環境の設定について説明します。JDBC ドライバは MySQL Connector/J を使用します。この設定は、J2EE サーバを配置するホストで実施します。

なお、データベースを設定するときの注意事項については、「4.1.11 データベースを設定するときの注意事項」を参照してください。

(1) MySQL Connector/J の準備

アプリケーションサーバがサポートするバージョンの MySQL Connector/J の JAR ファイルを入手して、J2EE サーバのクラスパスに指定してください。JAR ファイルの入手方法とクラスパスの指定方法について説明します。

1. アプリケーションサーバがサポートするバージョン番号の MySQL Connector/J をダウンロードします。
2. J2EE サーバを動作させるマシンに、ダウンロードした MySQL Connector/J を配置します。
3. 簡易構築定義ファイルに、MySQL Connector/J の JAR ファイルのパスを指定します。

簡易構築定義ファイルを作成するときに、物理ティアの定義（<tier>タグの定義）の J2EE サーバの定義で、<configuration>タグに次の設定を追加してください。

<param-name>タグの指定値	<param-value>タグの指定値
add.class.path	<JAR ファイルのクラスパス>

簡易構築定義ファイルの作成については、「4.1.22 簡易構築定義ファイルを設定する」を参照してください。

設定内容の詳細については、マニュアル「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)」の「4.3 簡易構築定義ファイル」を参照してください。

4. J2EE サーバに、MySQL Connector/J 用の DB Connector をインポートします。

DB Connector のインポートについては、「4.1.28 DB Connector を設定する (CUI 利用時)」を参照してください。

4.1.10 データベース接続環境を設定する (PostgreSQL の設定)

J2EE サーバ上の J2EE アプリケーションからデータベースへ接続するためには、リソースアダプタとして DB Connector を使用します。また、DB Connector を使用してデータベースに接続するためには、データベースに対応する JDBC ドライバが必要です。

ここでは、DB Connector を使用して PostgreSQL に接続する際に必要なデータベース接続環境の設定について説明します。JDBC ドライバは PostgreSQL JDBC Driver を使用します。この設定は、J2EE サーバを配置するホストで実施します。

なお、データベースを設定するときの注意事項については、「4.1.11 データベースを設定するときの注意事項」を参照してください。

(1) PostgreSQL JDBC Driver の準備

アプリケーションサーバがサポートするバージョンの PostgreSQL JDBC Driver の JAR ファイルを入手して、J2EE サーバのクラスパスに指定してください。JAR ファイルの入手方法とクラスパスの指定方法について説明します。

1. アプリケーションサーバがサポートするバージョン番号の PostgreSQL JDBC Driver をダウンロードします。
2. J2EE サーバを動作させるマシンに、ダウンロードした PostgreSQL JDBC Driver を配置します。
3. 簡易構築定義ファイルに、PostgreSQL JDBC Driver の JAR ファイルのパスを指定します。

簡易構築定義ファイルを作成するときに、物理ティアの定義（<tier>タグの定義）の J2EE サーバの定義で、<configuration>タグに次の設定を追加してください。

<param-name>タグの指定値	<param-value>タグの指定値
add.class.path	<JAR ファイルのクラスパス>

簡易構築定義ファイルの作成については、「[4.1.22 簡易構築定義ファイルを設定する](#)」を参照してください。

設定内容の詳細については、マニュアル「[アプリケーションサーバ リファレンス 定義編\(サーバ定義\)](#)」の「[4.3 簡易構築定義ファイル](#)」を参照してください。

4. J2EE サーバに、PostgreSQL JDBC Driver 用の DB Connector をインポートします。

DB Connector のインポートについては、「[4.1.28 DB Connector を設定する \(CUI 利用時\)](#)」を参照してください。

4.1.11 データベースを設定するときの注意事項

DB Connector を使用する場合、DB Connector で使用する JDBC ドライバの JAR ファイルは、コンテナ拡張ライブラリとして設定します。アプリケーション内に含めている場合は、アプリケーション内から削除してください。

4.1.12 環境変数とは

アプリケーションサーバで設定する環境変数には、Application Server が参照する環境変数と、Management Server を使用して構築した論理サーバが参照する環境変数があります。Application Server が参照する環境変数をシステムの環境変数といい、論理サーバが参照する環境変数を論理サーバの環境変数といいます。

4.1.13 システムの環境変数を設定する

Management Server を使用してシステムを構築する場合には、次に示すシステムの環境変数を設定します。システムの環境変数の設定方法については、OS のドキュメントを参照してください。この設定は、各ホストで実施します。

- CJMEMDUMP_PATH (Windows の場合)

論理 J2EE サーバが強制停止された場合に、ユーザダンプを取得するときは、ユーザダンプの出力先ディレクトリを設定します。

ユーザダンプファイル (cjmemdump.dmp) は、この環境変数に設定したディレクトリ下に作成されます。

(例) 環境変数 CJMEMDUMP_PATH に「C:\temp」を設定した場合

```
C:\temp\cjmemdump.dmp
```

この環境変数を設定する場合は、次の点に注意してください。

- adminagent.properties (運用管理エージェントプロパティファイル) の adminagent.forcestop.threaddump キーには、「true」を指定してください。「true」を指定していない場合、環境変数 CJMEMDUMP_PATH を設定してもユーザダンプは出力されません。
- 環境変数 CJMEMDUMP_PATH には、存在するディレクトリを設定してください。ディレクトリが存在しない場合、ユーザダンプは出力されません。
- cjmemdump.dmp が、環境変数 CJMEMDUMP_PATH に設定したディレクトリにある場合、そのファイルは上書きされます。

- TZ (OS 共通)

タイムゾーンを指定します。通常は、「JST-9」を指定してください。ほかの値も指定できます。

注意事項

(Windows の場合)

環境変数 ComSpec は、Windows のインストール時に、システム環境変数として設定されます。環境変数 ComSpec がない場合、PRF の起動に失敗するなどの問題が発生します。

環境変数 ComSpec が設定されていることを確認して、存在しない場合には、コマンドラインインターフェース (cmd.exe など) のフルパスを設定してください。

4.1.14 論理サーバの環境変数を設定する

Management Server を使用する場合には、次の定義個所で論理サーバに環境変数が設定できます。この設定は、各ホストで実施します。論理サーバの環境変数の設定時の確認事項については、「[4.1.15 論理サーバの環境変数を設定するときの確認事項](#)」を参照してください。

1. 簡易構築定義ファイルもしくは構成変更定義ファイルの user.env.variable パラメタ、または運用管理ポータルの【環境変数の設定】画面
2. adminagent.xml (運用管理エージェント設定ファイル)
3. OS の環境変数 (Windows の場合はシステム環境変数、UNIX の場合は運用管理エージェントを起動するシェルでの環境変数)

同一の環境変数をそれぞれの定義個所で設定している場合は、1., 2., 3.の順序で設定値が優先され、論理サーバに設定されます。

2. (adminagent.xml) には、インストール時点で環境変数が設定されています。3. (OS の環境変数) で、2. (adminagent.xml) に設定されている環境変数を設定しても、設定は有効になりません。このため、論理サーバの環境変数は、1. (簡易構築定義ファイル、構成変更定義ファイル、運用管理ポータル)、または 2. (adminagent.xml) で設定することをお勧めします。

注意事項

Windows の場合、論理 Web サーバに設定した環境変数は HTTP Server のサービスプログラム (制御プロセス、サーバプロセス) には引き継がれません。そのため、CGI プログラムなどから参照できません。

4.1.15 論理サーバの環境変数を設定するときの確認事項

インストール時点で adminagent.xml (運用管理エージェント設定ファイル) には、Management Server が自動で設定する論理サーバの環境変数が設定されます。Management Server が自動で設定する論理サーバの環境変数を OS ごとに次の表に示します。

表 4-6 Management Server が自動で設定する論理サーバの環境変数 (Windows の場合)

論理サーバ	環境変数	設定値
論理パフォーマンストレーサ	Path	\${cosminexus.home}¥PRF¥bin;%{Path}
論理スマートエージェント	Path	\${cosminexus.home}¥TPB¥bin;%{Path}
	TPDIR	\${cosminexus.home}¥TPB
	VBROKER_ADMIN	\${cosminexus.home}¥TPB¥adm
論理ネーミングサービス	CLASSPATH	設定されません。
	Path	\${cosminexus.home}¥jdk¥bin;\${cosminexus.home}¥TPB¥bin;%{Path}
	TPDIR	\${cosminexus.home}¥TPB
	VBROKER_ADMIN	\${cosminexus.home}¥TPB¥adm

論理サーバ	環境変数	設定値
論理 CTM ドメインマネージャ	CTMDIR	`\${cosminexus.home}/CTM
	Path	`\${cosminexus.home}/CTM/bin;\${cosminexus.home}/TPB/bin;%{Path}
	TPDIR	`\${cosminexus.home}/TPB
	VBROKER_ADM	`\${cosminexus.home}/TPB/adm
論理 CTM	CTMDIR	`\${cosminexus.home}/CTM
	Path	`\${cosminexus.home}/CTM/bin;\${cosminexus.home}/TPB/bin;%{Path}
	TPDIR	`\${cosminexus.home}/TPB
	VBROKER_ADM	`\${cosminexus.home}/TPB/adm
論理 J2EE サーバ	Path	`\${cosminexus.home}/CTM/bin;\${cosminexus.home}/jdk/bin;\${cosminexus.home}/TPB/bin;\${cosminexus.home}/PRF/bin;%{Path}
	TPDIR	`\${cosminexus.home}/TPB
	VBROKER_ADM	`\${cosminexus.home}/TPB/adm

表 4-7 Management Server が自動で設定する論理サーバの環境変数 (UNIX の場合)

論理サーバ	環境変数	設定値
論理パフォーマンストレーサ	PATH	`\${cosminexus.home}/PRF/bin:\${PATH}
	ライブラリパス ^{※1}	`\${cosminexus.home}/PRF/lib:%{ライブラリパス ^{※1} }
	AIXTHREAD_SCOPE ^{※2}	S
	EXTSHM ^{※2}	ON
	LDR_CNTRL ^{※2}	MAXDATA=0x40000000
	NODISCLAIM ^{※2}	true
	PSALLOC ^{※2}	early
論理スマートエージェント	PATH	`\${cosminexus.home}/TPB/bin:/bin:/usr/bin:\${PATH}
	TPDIR	`\${cosminexus.home}/TPB
	VBROKER_ADM	`\${cosminexus.home}/TPB/adm
	ライブラリパス ^{※1}	`\${cosminexus.home}/TPB/lib:/opt/hitachi/common/lib:%{ライブラリパス ^{※1} }
	AIXTHREAD_COND_DEBUG ^{※2}	設定されません。
	AIXTHREAD_MUTEX_DEBUG ^{※2}	設定されません。

論理サーバ	環境変数	設定値
論理ネーミングサービス	AIXTHREAD_RWLOCK_DEBUG※2	設定されません。
	AIXTHREAD_SCOPE※2	S
	NODISCLAIM※2	true
	PSALLOC※2	early
論理ネーミングサービス	CLASSPATH	設定されません。
	PATH	<ul style="list-style-type: none"> AIX の場合 \${cosminexus.home}/jdk/bin:\${cosminexus.home}/TPB/bin:/usr/bin:\${PATH} Linux の場合 \${cosminexus.home}/jdk/bin:\${cosminexus.home}/TPB/bin:/bin:/usr/bin:\${PATH}
	TPDIR	\${cosminexus.home}/TPB
	VBROKER_ADMIN	\${cosminexus.home}/TPB/admin
	ライブラリパス※1	\${cosminexus.home}/TPB/lib:\${ライブラリパス※1}
	AIXTHREAD_SCOPE※2	S
	NODISCLAIM※2	true
論理 CTM ドメインマネージャ	CTMDIR	\${cosminexus.home}/CTM
	PATH	\${cosminexus.home}/CTM/bin:\${cosminexus.home}/TPB/bin:\${PATH}
	TPDIR	\${cosminexus.home}/TPB
	VBROKER_ADMIN	\${cosminexus.home}/TPB/admin
	ライブラリパス※1	\${cosminexus.home}/CTM/lib:\${cosminexus.home}/TPB/lib:\${ライブラリパス※1}
	AIXTHREAD_SCOPE※2	S
	EXTSHM※2	ON
	LDR_CNTRL※2	MAXDATA=0x40000000
	PSALLOC※2	early
	NODISCLAIM※2	true
論理 CTM	CTMDIR	\${cosminexus.home}/CTM
	PATH	\${cosminexus.home}/CTM/bin:\${cosminexus.home}/TPB/bin:\${PATH}

論理サーバ	環境変数	設定値
	TPDIR	\${cosminexus.home}/TPB
	VBROKER_ADM	\${cosminexus.home}/TPB/adm
	ライブラリパス※1	\${cosminexus.home}/CTM/lib:\${cosminexus.home}/TPB/lib:\${ライブラリパス※1}
	AIXTHREAD_SCOPE※2	S
	EXTSHM※2	ON
	LDR_CNTRL※2	MAXDATA=0x40000000
	PSALLOC※2	early
	NODISCLAIM※2	true
論理 J2EE サーバ	CSCCFJ_SERVER_HOME	\${cosminexus.home}/CC
	PATH	\${cosminexus.home}/jdk/bin:\${cosminexus.home}/TPB/bin:\${cosminexus.home}/PRF/bin:/bin:/usr/bin:\${PATH}
	TPDIR	\${cosminexus.home}/TPB
	VBROKER_ADM	\${cosminexus.home}/TPB/adm
	ライブラリパス※1	\${cosminexus.home}/TPB/lib: \${cosminexus.home}/PRF/lib:\${cosminexus.home}/CTM/lib:/opt/hitachi/common/lib:\${ライブラリパス※1}
	AIXTHREAD_COND_DEBUG※2	OFF
	AIXTHREAD_MUTEX_DEBUG※2	OFF
	AIXTHREAD_RWLOCK_DEBUG※2	OFF
	AIXTHREAD_SCOPE※2	S
	EXTSHM※2	ON
	NODISCLAIM※2	true
	PSALLOC※2	early
論理 Web サーバ	PATH	\${cosminexus.home}/PRF/bin:\${PATH}
	ライブラリパス※1	\${cosminexus.home}/PRF/lib:\${ライブラリパス※1}
	NODISCLAIM※2	true
	PSALLOC※2	early

注※1 ライブラリパスに指定する環境変数名は、OS ごとに異なります。

- AIX の場合 : LIBPATH

- Linux の場合 : LD_LIBRARY_PATH

注※2 AIX 固有の環境変数です。AIX 以外の OS には該当しません。

4.1.16 運用管理機能を構築する

Management Server をセットアップして、運用管理機能を構築する手順について説明します。運用管理機能の構築時に使用する情報については、「[4.1.17 Management Server を使用するための設定する情報](#)」および「[4.1.18 運用管理エージェントを使用するための設定する情報](#)」を参照してください。この設定は、Management Server を配置するホストで実施します。ここでは、Management Server は、J2EE サーバを配置するホストに配置しています。また、ホストの起動と同時に Management Server および運用管理エージェントを起動するように設定しています。

1. J2EE サーバを配置するホストでコマンドプロンプトを起動します。
2. mngsvrctl コマンドのサブコマンド「setup」を実行し、Management Server をセットアップします。コマンドの実行例を次に示します。ここでは、管理ユーザ ID に「admin」、管理ユーザパスワードに「admin」を設定します。

```
mngsvrctl setup -u admin -p admin
```

3. mngautorun コマンドを実行し、ホストの起動と同時に Management Server および運用管理エージェントを起動するように設定します。

コマンドの実行例を次に示します。

```
mngautorun both -sync
```

参考

Management Server および運用管理エージェントを障害発生時に自動再起動させる設定もできます。また、UNIX の場合は、ホストの停止と同時に Management Server および運用管理エージェントを停止するように設定できます。Management Server および運用管理エージェントの自動再起動および自動停止の設定については、マニュアル「[アプリケーションサーバ 機能解説 運用／監視／連携編](#)」の次の箇所を参照してください。

- 2.6.4 自動再起動の設定
- 2.6.5 自動停止の設定

4. Management Server の環境を設定します。

Management Server を使用するための設定する情報には、次のようなものがあります。「[4.1.17 Management Server を使用するための設定する情報](#)」を参照して、設定します。

- Management Server が使用するポート番号
- Management Server へのアクセスを許可するホスト

- Management Server を使用するユーザーアカウント
- Management Server が出力するログに関する情報

5. 運用管理エージェントの環境を設定します。

運用管理エージェントを使用するために設定する情報には、次のようなものがあります。「[4.1.18 運用管理エージェントを使用するために設定する情報](#)」を参照して、設定します。

- 運用管理エージェントが使用するポート番号
- 運用管理エージェントによる論理サーバの監視に関する情報（動作確認レベルや、動作確認間隔など）
- HTTP Server の動作確認に関する情報

4.1.17 Management Server を使用するために設定する情報

Management Server を使用するために設定する情報について説明します。なお、Management Server を使用する際には、管理ユーザーアカウントを設定しない運用もできます。

(1) Management Server のプロパティの設定

Management Server のプロパティファイル (mserver.properties) で、Management Server が使用するポート番号や、Management Server へのアクセスを許可するホストなどを指定できます。

mserver.properties に指定するキーを次に示します。キーの詳細、およびここで説明していないキーについては、マニュアル「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)」の「[8.2.6 mserver.properties \(Management Server 環境設定ファイル\)](#)」を参照してください。

- adminagent.connector.comm.state.cache_max_time
運用管理エージェントとの通信状態をキャッシュする最大時間を指定します。
- webserver.connector.ajp13.port
Management Server 内部通信用ポート番号を指定します。
- webserver.connector.http.port
Management Server 接続 HTTP ポート番号を指定します。
- webserver.shutdown.port
Management Server 終了要求受信ポート番号を指定します。
- webserver.connector.http.permitted.hosts
Management Server へのアクセスを許可するホストおよび運用管理エージェント稼働ホストの、IP アドレスまたはホスト名を指定します。
- mngsvr.myhost.name
運用管理機能 (Management Server) を配置するホストに、論理サーバを配置するホスト (アプリケーションサーバ) からアクセス可能な IP アドレス、またはその IP アドレスを示すホスト名を指定します。

ます。運用管理機能（Management Server）を配置するホストが複数個のIPアドレスを持っていて、ホスト名が意図するIPアドレスに変換されないおそれがある場合に、このキーを指定します。

これらのキーのほかに、mserver.propertiesでは次の設定ができます。

- snapshot ログ収集や障害検知時コマンドによる資料取得のためのキーを設定できます。snapshot ログ収集の設定、および障害検知時コマンドによる資料取得のための設定については、マニュアル「アプリケーションサーバ機能解説 保守／移行編」の次の個所を参照してください。
 - 3.3.1 障害検知時コマンドによる資料取得の設定（J2EE アプリケーションを実行するシステム）
 - 3.3.2 障害検知時コマンドによる資料取得の設定（バッチアプリケーションを実行するシステム）
 - 3.3.3 snapshot ログ収集の設定（J2EE アプリケーションを実行するシステム）
 - 3.3.4 snapshot ログ収集の設定（バッチアプリケーションを実行するシステム）
- 管理ユーザアカウントを設定しないで Management Server にログインすることもできます。この場合、管理ユーザアカウントの省略機能を有効にします。機能の設定については、「(3) 管理ユーザアカウントを設定しない場合の設定」を参照してください。

また、必要に応じて、cmxserver.properties（サーバ設定プロパティファイル）で、Smart Composer機能で使用するコマンドの実行環境を設定します。

cmxserver.propertiesに指定するキーを次に示します。

- cmx.max_wait_time
共通オプション-waitを指定して実行したコマンドに有効で、すでに実行しているコマンドの完了を待つときの最大待ち時間を指定します。デフォルトで、180秒が設定されています。

(2) Management Server の動作環境の設定

mngsvrctl コマンドや Management Server のプロパティで、Management Server を利用するユーザのアカウント情報、Management Server が outputするログ情報など、Management Server の動作環境を設定してください。

Management Server の動作環境の設定内容を次の表に示します。

表 4-8 Management Server の動作環境の設定内容

設定項目	設定内容	設定の要否
管理ユーザアカウントの設定	Management Server のセットアップ時に管理ユーザアカウントを設定していない場合は、mngsvrctl コマンドに引数「setup」を指定して実行するか、または運用管理ポータルの「[管理ユーザアカウントの設定]」画面で設定します。	○※
ログの設定	mserver.properties または運用管理ポータルの「[ログの設定]」画面で、Management Server が outputするログのログレベル、ファイル数、サイズなどを設定します。 次のキーを指定します。	△

設定項目	設定内容	設定の要否
	<ul style="list-style-type: none"> com.cosminexus.mngsvr.log.level キーに, Management Server のログの出力レベルを指定します。 com.cosminexus.mngsvr.log.rotate キーに, Management Server のログのファイル面数を指定します。 com.cosminexus.mngsvr.log.size キーに, Management Server のログのファイルサイズを指定します。 	
開始時の設定	mserver.properties または運用管理ポータルの [開始時の設定] 画面で, Management Server を起動するときに, 論理サーバを一括起動するかどうかを設定します。一括起動する場合には, この設定が必要です。	△
性能解析トレース収集の設定	mserver.properties または運用管理ポータルの [性能解析トレース収集の設定] 画面で, com.cosminexus.mngsvr.trace キーに, Management Server で保持する性能解析トレースファイルの面数を設定します。	△
JP1 連携の設定	mserver.properties または運用管理ポータルの [JP1 連携の設定] 画面で, com.cosminexus.mngsvr.jp1event.enabled キーに, Management Server の障害情報（検知した障害や論理サーバの起動の通知など）を JP1 イベントとして発行するかどうかを指定します。このほかに, com.cosminexus.mngsvr.jp1event から始まるキーで, Management Server の障害情報を JP1 へ通知するための設定をします。JP1 と連携する機能を利用する場合には, この設定が必要です。JP1 との連携については, マニュアル「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。	△

(凡例)

○：設定します。

△：デフォルトの設定を変更する場合, 特定の機能を使用する場合に設定します。

注※

Management Server のセットアップ時に, 管理ユーザアカウントを設定している場合は, 設定不要です。ただし, セットアップ時に設定した管理ユーザアカウントを変更したい場合は, mngsvrctl コマンドを使用して再設定してください。

(3) 管理ユーザアカウントを設定しない場合の設定

管理ユーザアカウントとは, 運用管理ポータルにログインしたり, Management Server に対してコマンド（例：mngsvrutil コマンド）を実行したりするときに使用するユーザ権限です。管理ユーザアカウントの設定は Management Server のセットアップ時などに実施しますが, 管理ユーザアカウントを設定しないで Management Server を使用することもできます。管理ユーザアカウントを設定しない場合, 運用管理ポータルの操作および Management Server に対してコマンドを実行するときには次のようになります。

- 運用管理ポータルの [ログイン] 画面が表示されません。URL を指定するとログインなしで運用管理ポータルを使用できます。
- Management Server に対するコマンドの実行時に, -u (管理ユーザ ID) および-p (管理ユーザパスワード) の指定が不要になります。これらのオプションを指定しなくても, コマンドを実行できます。

管理ユーザアカウントを設定しない場合は, mserver.properties の com.cosminexus.mngsvr.management_user_account.enabled キーに false を設定します。ただし, 管

理ユーザアカウントを設定していないと、だれでも運用管理ポータルやコマンドの実行を通して Management Server を操作できます。複数人で Management Server を操作するような運用環境の場合は、管理ユーザアカウントを設定することをお勧めします。

4.1.18 運用管理エージェントを使用するために設定する情報

運用管理エージェントを使用するために設定する情報について説明します。

(1) 運用管理エージェントのプロパティの設定

運用管理エージェントのプロパティファイル (adminagent.properties) で、運用管理エージェントが使用するポート番号などを指定できます。

- adminagent.adapter.port

運用管理エージェントが Management Server との通信に使用するポート番号を指定します。

また、このほかに snapshot ログの収集先も設定できます。snapshot ログの収集先の設定については、マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の次の個所を参照してください。

- 3.3.3 snapshot ログ収集の設定 (J2EE アプリケーションを実行するシステム)
- 3.3.4 snapshot ログ収集の設定 (バッチアプリケーションを実行するシステム)

(2) 運用管理エージェントによる論理サーバの監視に関する設定

システムを運用する場合には、Management Server の運用管理コマンド (mngsvrutil) を使用して論理サーバのステータスを確認することで、論理サーバの起動／停止状態がわかります。Management Server で論理サーバを管理する場合には、運用管理エージェントが定期的に論理サーバのプロセス監視および動作確認を実施し、論理サーバのステータスを Management Server に通知します。運用管理エージェントが論理サーバの動作確認をするときのレベルや間隔の設定は変更できます。必要に応じて、論理サーバの動作確認に関する項目を adminagent.properties で設定してください。なお、障害発生時の論理サーバの自動再起動については、マニュアル「アプリケーションサーバ 機能解説 運用／監視／連携編」の「2.4.3 論理サーバの自動再起動」を参照してください。

参考

Management Server で Web サーバの動作確認をする場合、動作確認用のログを通常のログ（エラーログ、アクセスログ、リクエストログなど）と別に出力できます。設定方法については、マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「3.3.8 Web サーバのログ取得の設定」を参照してください。

- 論理サーバの動作確認レベルの設定

運用管理エージェントでは、論理サーバが正常に動作しているかどうかを、プロセスの存在を確認したり、論理サーバにアクセスしたりすることで判断します。論理サーバの動作確認レベルは、adminagent.<論理サーバの種類>.watch.level キーに指定します。必要に応じて、論理サーバの動作確認レベルを変更してください。指定できる値は「1」または「2」です。「1」はプロセスの存在を確認するだけで論理サーバの動作を確認します。また、「2」はプロセスの存在を確認し、実際に論理サーバにアクセスして動作を確認します。指定できる値は論理サーバの種類によって異なります。なお、Web サーバで SSL 通信だけを有効にしている場合、論理 Web サーバの動作確認レベルはプロセスの存在を確認するだけに変更 (adminagent.hws.watch.level=1 を設定) してください。

- **論理サーバの動作確認間隔の設定**

論理サーバの動作確認の間隔（秒）は、adminagent.<論理サーバの種類>.watch.interval キーに指定します。デフォルトは 10 秒です。必要に応じて、論理サーバの動作確認間隔を変更してください。

論理サーバの動作確認間隔を変更する場合は、論理サーバの動作確認間隔（秒）が停止監視時間（分）または強制停止監視時間（分）よりも短くなるようにしてください。動作確認の間隔が論理サーバの監視時間以上の場合は、論理サーバの停止または強制停止に失敗することがあります。

- **論理サーバの動作確認タイムアウトの設定**

論理サーバの動作確認のタイムアウト時間（秒）は、adminagent.<論理サーバの種類>.watch.timeout キーに指定します。デフォルトは 60 秒です。必要に応じて、論理サーバの動作を確認するときのタイムアウト時間を変更してください。

- **論理サーバの動作確認リトライの設定**

論理サーバの動作確認のリトライ回数は、adminagent.<論理サーバの種類>.watch.retry_count キーに指定します。デフォルトは 1 回です。

また、adminagent.watch.retry_timeout.enabled キーでは、タイムアウトの発生をリトライの対象にするかどうかを指定できます。デフォルトは false（リトライの対象にしない）です。なお、新規インストール時は true（リトライの対象にする）が指定されています。

必要に応じて、論理サーバの動作を確認するときのリトライの設定を変更してください。

- **論理サーバの動作確認の開始時間の設定**

論理サーバを起動してから動作確認（動作確認レベル 2）を開始するまでの時間（秒）は、adminagent.<論理サーバの種類>.watch.start_time キーに指定します。デフォルト値は論理サーバの種類によって異なります。必要に応じて、論理サーバの動作確認の開始時間を変更してください。

なお、論理サーバの動作確認の開始時間を変更する場合は、「論理サーバの動作確認の開始時間（秒）<論理サーバの起動監視時間（分）」となるように注意してください。論理サーバの起動監視時間は、運用管理ポータルの「論理サーバの起動／停止」の【起動／停止の設定】画面で設定できます。起動監視時間が動作確認の開始時間よりも小さい場合は、論理サーバの起動に失敗します。

(3) HTTP Server の動作確認の設定

Management Server では、Web サーバとして HTTP Server を管理できます。HTTP Server の動作確認は、Web サーバ定義ファイル (httpsd.conf) の最後の Listen ディレクティブのポート番号に対して、http でリクエストを送信することで実施します。そのため、最後の Listen ディレクティブで SSL が有効になっている場合は動作確認できません。最後の Listen ディレクティブは SSL を無効にしてください。

また、Web サーバ定義ファイル (httpsd.conf) 内のすべての Listen ディレクティブで SSL が有効になっている場合は、HTTP Server へのアクセスで HTTP Server の動作確認を実施しないで、プロセスの存在確認だけを実施します。Web サーバが SSL 有効・無効を判定できなかった場合、Web サーバの起動監視はしないで、タイムアウトによって停止処理をします。タイムアウトするまでの間、Web サーバは稼働状態の可能性があります。

ここでは、動作確認用 HTTP メソッドの設定、および HTTP Server の動作確認用 URL の設定について説明します。ただし、HTTP Server 動作確認用 URL は、動作確認用 HTTP メソッドの設定で「HEAD」を指定している場合にだけ設定できます。

• 動作確認用 HTTP メソッドの設定

論理サーバの動作確認レベル (adminagent.<論理サーバの種類>.watch.level キー) で、「2」を指定 (プロセスの存在を確認し、実際に論理サーバにアクセスして動作を確認する) した場合の HTTP Server の動作確認用 HTTP メソッドを指定します。指定できる値は「HEAD」または「OPTIONS」です。「HEAD」を指定した場合は、HTTP Server 動作確認用 URL (adminagent.hws.watch.url キー) で指定された URL にアクセスして応答が返ることを確認します。また、「OPTIONS」を指定した場合は、Web サーバが HTTP メソッドを受け付け可能な状態であるかを確認します。

なお、「OPTIONS」を指定した場合、HTTP Server の設定で、ルートディレクトリに対してすべてのホストからのアクセスを禁止していると、運用管理エージェントからの動作確認アクセスも拒否されてしまうため、正しく稼働判定できなくなり、HTTP Server の起動に失敗します。その場合、ループバックアドレス (ホストの固定をしている場合は固定した IP アドレス) からのアクセスを許可する必要があります。Basic 認証の設定をしている場合など、アクセスを制限している場合も正しく稼働判定できなくなるので注意してください。

• HTTP Server 動作確認用 URL の設定

運用管理エージェントは、HTTP Server が正常に動作しているかどうかを、設定した URL に対して定期的にダミーアクセスをすることで判断します。ダミーアクセスのレスポンスが返らないような状態 (無応答の状態) になると、HTTP Server が異常と判断し、障害を検出します。このため、Web サーバを使用する場合には、動作確認用 URL として実際にアクセスできる URL を設定してください。

HTTP Server 動作確認用 URL は、adminagent.hws.watch.url キーに指定します。

URL には、http で始まる絶対パス、またはルートコンテキストからの相対パスを指定してください。デフォルト値は、ルートコンテキスト (http://localhost:<ポート番号>/) です。同一のホストに複数の Web サーバを構築する場合は相対パスで指定してください。

(例)

- 動作確認用 URL に「http://HostA:80/index.html」を指定する場合の例を次に示します。

```
adminagent.hws.watch.url=http://HostA:80/index.html
```

- 動作確認用 URL に「http://localhost:<ポート番号>/index.html」を指定する場合（複数 Web サーバ対応）の例を次に示します。

```
adminagent.hws.watch.url=index.html
```

4.1.19 運用管理エージェントを起動する

アプリケーションサーバごとにコマンドプロンプトを起動し、adminagentctl コマンドを実行して、運用管理エージェントを起動します。Windows の場合はサービスからも起動できます。

コマンドの実行例を次に示します。

```
adminagentctl start -sync
```

4.1.20 Management Server を起動する

Management Server をセットアップしたホスト（J2EE サーバを配置するホスト）でコマンドプロンプトを起動し、mngsvrctl コマンドを実行して、Management Server を起動します。Windows の場合はサービスからも起動できます。

コマンドの実行例を次に示します。

```
mngsvrctl start -sync
```

4.1.21 簡易構築定義ファイルとは（J2EE アプリケーション使用時）

簡易構築定義ファイルとは、Smart Composer 機能のコマンドで構築する Web システムを定義する XML 形式のファイルです。

（1）簡易構築定義ファイルの構成

簡易構築定義ファイルでは、Web システムの次の構成について設定します。

- Web システムの属性定義（<web-system>タグの定義）
- 負荷分散機の定義（<load-balancer>タグの定義）※1
 - cookie スイッチングの定義（<load-balancer>タグの<cookie-switching>タグの定義）※2
 - 仮想サーバの定義（<load-balancer>タグの<virtual-server>タグの定義）
- 物理ティアの定義（<tier>タグの定義）
 - 物理ティアに属する論理サーバの種類ごとの定義（<tier>タグの<configuration>タグの定義）
 - データベースを使用するための定義（<tier>タグの<configuration>タグの定義）※3
- サービスユニットの定義（<unit>タグの定義）
 - サービスユニットを構成するホストの定義（<unit>タグの<allocation-host>タグの定義）

- ホストが属する物理ティアの論理サーバごとの定義（<unit>タグの<allocation-host>タグの<define-server>タグの定義）
- ホストの定義（<host>タグの定義）

注※1

Smart Composer 機能で負荷分散機の設定、および実サーバポートの開始・閉塞をしない場合は、負荷分散機の定義（<load-balancer>タグの定義）を記述する必要はありません。

注※2

負荷分散機のCookieスイッチング機能を使用する場合に必要です。この設定によって、一連のHTTPリクエストが一つのWebサーバまたはJ2EEサーバで処理されます。また、負荷分散機のCookieスイッチング機能を使用する場合は、物理ティアの定義（<tier>タグ）で、J2EEサーバのWebコンテナの設定に「webserver.container.server_id.enabled=true」を指定してください。

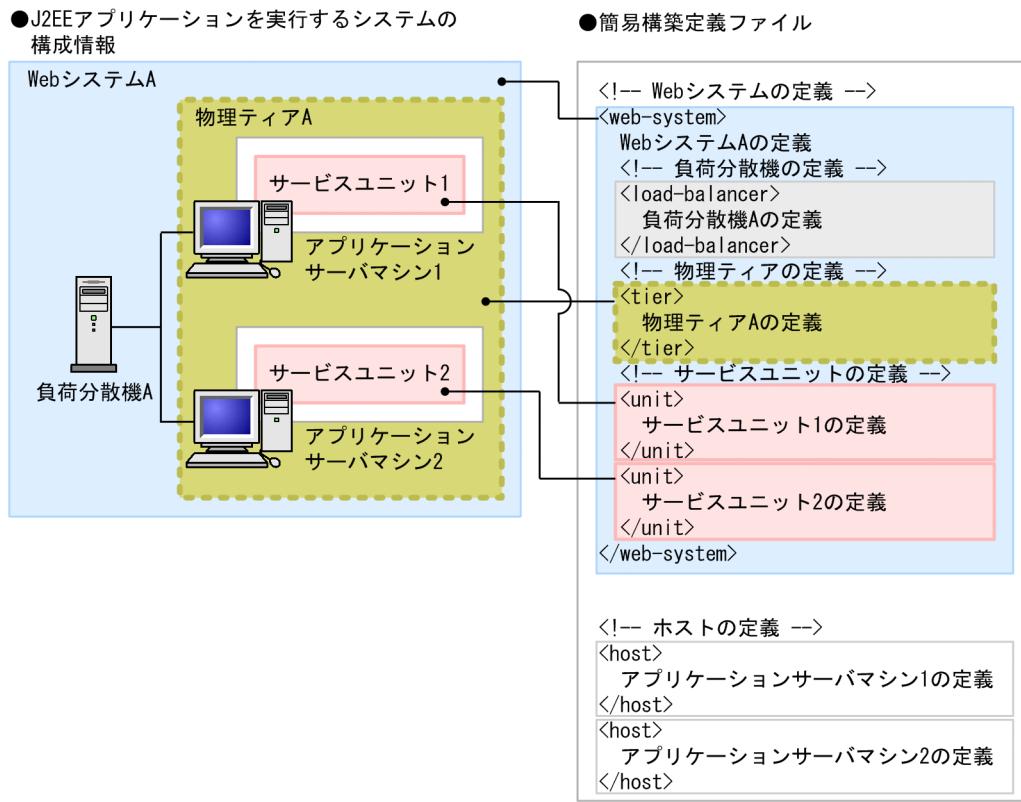
注※3

データベースを使用するための定義については、次のどれかを参照してください。

- [「4.1.6 データベース接続環境を設定する（HiRDBの設定）」](#)
- [「4.1.7 データベース接続環境を設定する（Oracleの設定）」](#)
- [「4.1.8 データベース接続環境を設定する（SQL Serverの設定）」](#)
- [「4.1.9 データベース接続環境を設定する（MySQLの設定）」](#)
- [「4.1.10 データベース接続環境を設定する（PostgreSQLの設定）」](#)

J2EEアプリケーションを実行するシステムの構成情報と簡易構築定義ファイルの定義との対応を次の図に示します。

図 4-2 J2EE アプリケーションを実行するシステムの構成情報と簡易構築定義ファイルの定義との対応



この図に示すように、簡易構築定義ファイルでは、論理サーバの構成や環境設定だけでなく、負荷分散機やホストも定義できます。なお、構築済みの Web システムなどで使用しているホストを使用する場合には、ホストの定義を省略することもできます。

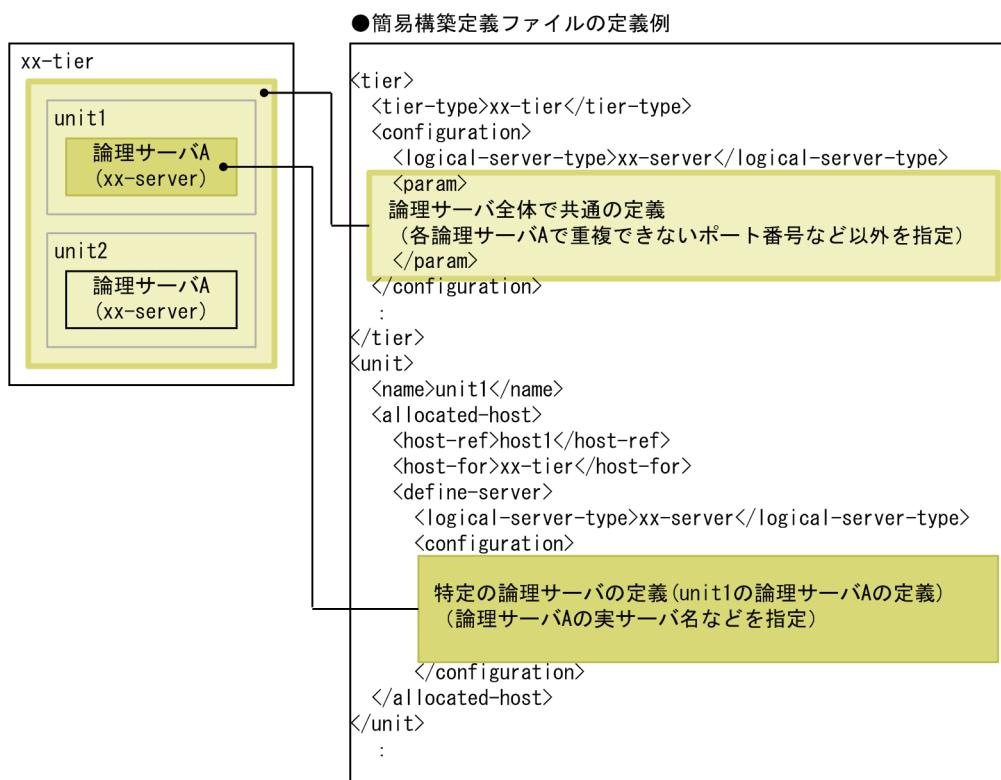
簡易構築定義ファイルのタグおよびパラメタについては、マニュアル「アプリケーションサーバリファレンス 定義編(サーバ定義)」の「4.3 簡易構築定義ファイル」を参照してください。

(2) コンフィグレーション定義で使用するタグと適用範囲

論理サーバの動作設定や、機能を使用するための設定などは、論理サーバのコンフィグレーションで定義します。論理サーバのコンフィグレーションは、論理サーバ全体、および論理サーバごとに定義できます。論理サーバ全体で共通の定義を設定する場合は物理ティアで、論理サーバごとの定義を設定する場合はサービスユニットで定義します。物理ティアで定義すると、その物理ティアに属するホストに配置する論理サーバ全体で共通のコンフィグレーションが設定されます。物理ティアおよびサービスユニットのコンフィグレーション定義で、同じパラメタを指定した場合は、サービスユニットの定義、物理ティアの定義の順に優先して適用されます。なお、Management Server が自動設定する内容については、「付録 E Management Server が自動で設定する内容」を参照してください。

論理サーバのコンフィグレーション定義の適用範囲を次の図に示します。

図 4-3 論理サーバのコンフィグレーション定義の適用範囲



コンフィグレーション定義時に使用する簡易構築定義ファイルの主なタグを次の表に示します。

表 4-9 論理サーバのコンフィグレーション定義時に使用する簡易構築定義ファイルの主なタグ

タグ名称	説明
<configuration>	論理サーバの動作環境を設定します。動作環境を設定する論理サーバの種類は、<logical-server-type>タグで指定します。 <configuration>タグは、物理ティアの定義（<tier>タグ内）、およびサービスユニットの定義（<unit>タグ内）に設定できます。<tier>タグ内または<unit>タグ内のどちらの<configuration>タグに設定するかによって、設定内容を適用する論理サーバの範囲が異なります。論理サーバ全体で共通の設定を変更したい場合は、<tier>タグ内の<configuration>タグを編集し、特定の論理サーバの設定を変更したい場合は、<unit>タグ内の<configuration>タグを編集します。
<logical-server-type>	<configuration>タグで動作環境を設定する論理サーバの種類を定義します。論理サーバの種類を次に示します。 <ul style="list-style-type: none"> 論理 Web サーバ : web-server 論理 J2EE サーバ : j2ee-server 論理 パフォーマンストレーサ : performance-tracer 論理 CTM ドメインマネジャー : ctm-domain-manager 論理 CTM : component-transaction-monitor 論理 スマートエージェント : smart-agent 論理 ユーザサーバ : user-server (<tier>タグ内には定義できません)
<param>	論理サーバの動作環境を設定するパラメタ（パラメタ名と値の組み合わせ）を定義します。1種類のパラメタごとに<param>タグで囲んで定義します。

タグ名称	説明
<param-name>	論理サーバの動作環境を設定するパラメタの名称を定義します。
<param-value>	<param-name>タグで指定したパラメタの設定値を定義します。

注 各論理サーバの<configuration>タグ内で、パラメタ名(<param-name>タグ)と値(<param-value>タグ)の設定を省略すると、論理サーバの動作環境にはデフォルトの値が設定されます。デフォルトの設定のまま利用する場合は、定義不要です。デフォルトで設定されている動作を変更したい場合や、アプリケーションサーバの機能を有効にしたい場合は、パラメタと値の設定を追加または変更してください。

4.1.22 簡易構築定義ファイルを設定する

簡易構築定義ファイルを設定する手順について説明します。簡易構築定義ファイル設定時の注意事項については、「4.1.23 簡易構築定義ファイルを設定するときの注意事項 (J2EE アプリケーション使用時)」を参照してください。

1. 簡易構築定義ファイルのテンプレートファイル (cmxdefcombinedmodel.xml) を任意の場所にコピーします。

簡易構築定義ファイルのテンプレートファイルは、次の場所に提供されています。

Windows の場合

```
<Application Serverのインストールディレクトリ>¥manager¥config¥templates¥cmxdefcombinedmodel.xml
```

UNIX の場合

```
/opt/Cosminexus/manager/config/templates/cmxdefcombinedmodel.xml
```

注意事項

互換モードを使用する場合は、「cmxdefcombinedmodel.xml」を「cmxdefcombinedmodel_V9.xml」に読み替えてください。

参考

テンプレートファイルの文字エンコーディングは、UTF-8 です。テンプレートファイルの内容は、セットアップウィザードで「標準セットアップ」を選択して、データベース接続環境を設定しない状態でセットアップした場合と同じシステム構成です。

2. 手順 1.のテンプレートファイルを任意の名称に変更します。

テキストエディタなどを利用して、簡易構築定義ファイルに手順 3.~手順 7.の情報を定義します。

3. Management Server で運用する Web システム名、Web システムの表示名などを定義します。

1 Web システムは、<web-system>タグから</web-system>タグまでの範囲に定義します。

4. 負荷分散機を使用する場合は、<load-balancer>タグを追加して、使用する負荷分散機の種類や、管理用のIPアドレスなどを定義します。

負荷分散機を使用しないシステムの場合、負荷分散機の定義は不要です。

5. システム構成やJ2EEアプリケーションで使用する機能に応じて、物理ティアを定義します。必要に応じて、論理サーバ全体で共通のコンフィグレーションを定義します。

システム構成やJ2EEアプリケーションで使用する機能に応じて、その機能を実現するために必要な論理サーバが決定されます。Smart Composer機能では、同一のデプロイメント（論理サーバの配置、定義設定など）を適用するホスト（サーバマシン）の集合を物理ティアという単位で定義します。物理ティアは、<tier>タグから</tier>タグまでの範囲に、物理ティアの種類や、論理サーバ全体で共通のコンフィグレーションなどを定義します。

6. 論理サーバ（サービスユニット）を配置するホストを定義します。必要に応じて、論理サーバごとのコンフィグレーションを定義します。

Smart Composer機能では、論理サーバの集合をサービスユニットという単位で定義します。サービスユニットは、<unit>タグから</unit>タグまでの範囲に、サービスユニットを配置するホストや、そのホストが属する物理ティア、論理サーバごとのコンフィグレーションなどを定義します。

7. ホストを定義します。

物理ティアに属するホストのホスト名や、運用管理エージェントのホスト名などを定義します。ホストは、Webシステムに依存しません。Management Serverで定義済みのホストを利用する場合は、ホストの定義を省略できます。ホストは、<host>タグから</host>タグまでの範囲に定義します。

8. 手順3.～手順7.で定義した情報を保存します。

4.1.23 簡易構築定義ファイルを設定するときの注意事項（J2EEアプリケーション使用時）

簡易構築定義ファイルを設定するときの注意事項について説明します。

（1）アプリケーションサーバの機能を有効にするための設定

簡易構築定義ファイルのコンフィグレーション定義では、デフォルトが設定されています。使用するアプリケーションサーバの機能によっては、デフォルトの設定では無効なものもあります。この場合は、簡易構築定義ファイルのパラメタと値を追加または変更して、機能を有効にする設定をします。なお、機能を使用するために設定する項目のうち、一部の設定項目は、簡易構築定義ファイルで定義できません。簡易構築定義ファイルで設定できない項目は、ほかのユーザ定義ファイルを使用して定義します。

例えば、J2EEサーバが使用するネーミングサービス、トランザクション、コネクション、コンテナ、ログ、セキュリティの設定を変更したい場合、J2EEサーバのJavaVMの起動オプションなどを変更したい場合などには、簡易構築定義ファイルで設定を変更します。簡易構築定義ファイルに、テキストエディタなどを使用して、変更対象のパラメタ名と変更後の値を設定します。ただし、J2EEサーバのセキュリティ

ポリシーと Web アプリケーションのプロパティを変更したい場合には、Management Server では設定できません。簡易構築定義ファイル以外のユーザ定義ファイル (server.policy, hitachi_web.properties) をテキストエディタなどで編集して、動作設定を変更してください。

注意事項

server.policy または hitachi_web.properties を編集する場合は、編集対象の J2EE サーバのセットアップが完了していることが前提になります。また、UNIX の場合、動作設定を変更するときには、root 権限 (Component Container 管理者を設定していない場合)、または Component Container 管理者の権限 (Component Container 管理者を設定している場合) が必要です。

(2) AllText パラメタでの httpsd.conf の直接設定

簡易構築定義ファイル作成時に、論理 Web サーバ (web-server) の<configuration>タグ内の SetBy パラメタで「text」を指定した場合に、AllText パラメタで httpsd.conf を直接設定するときの注意事項について説明します。

- Management Server は、httpsd.conf の最後に記述された Listen ディレクティブの値で、Web サーバの動作確認をします。AllText パラメタ内に Listen ディレクティブを複数記述するときには、動作確認で使用する値を最後に記述してください。
- HWSGracefulStopTimeout ディレクティブの設定が必要になる場合があります。「[4.8.4 HWSGracefulStopTimeout ディレクティブと Manager の停止監視時間の設定に関する注意点](#)」を参照してください。
- Web サーバの snapshot ログの収集対象を変更するときは、AllText パラメタではなく、snapshot ログ収集対象定義ファイルで編集してください。なお、デフォルトの収集対象は、<HTTP Server のインストールディレクトリ>/servers/HWS_<論理 Web サーバの実サーバ名>/logs です。
- AllText パラメタ内に次の Include ディレクティブの指定があると、Web システムの構築時に警告メッセージが出力されます。SetBy パラメタで「text」を指定した場合には、次の Include ディレクティブを指定しないでください。

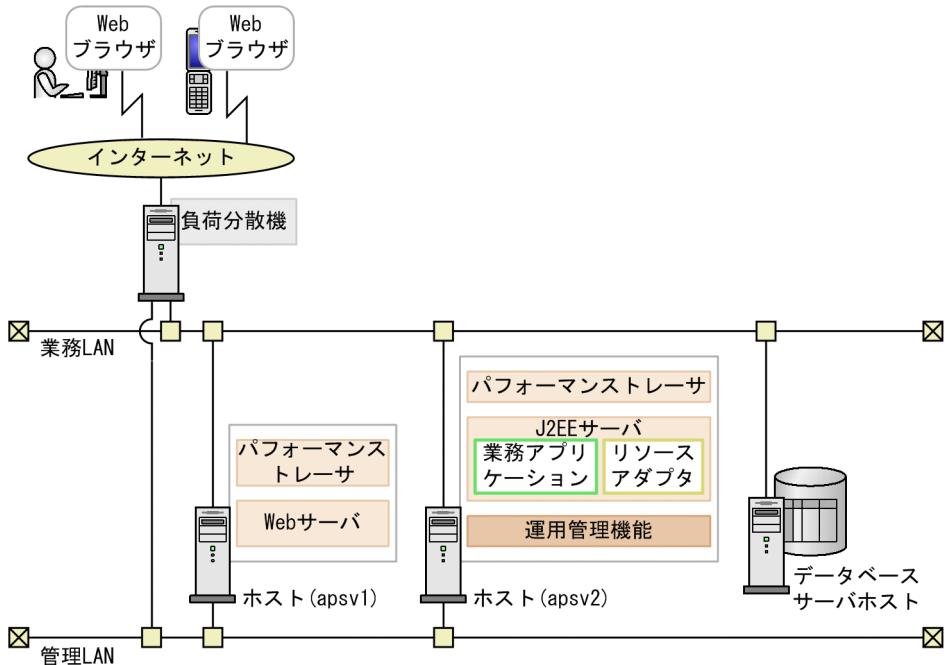
```
Include <HTTP Serverのインストールディレクトリ>/servers/HWS_<論理Webサーバの実サーバ名>/conf/httpsd_manager.conf
```

(3) 一つのホストを複数のネットワークセグメントに接続する場合の環境設定での注意

ここでは、一つのホストを複数のネットワークセグメントに接続する場合に、環境設定で注意することについて説明します。

次の図に示すように、ホストが複数のネットワークセグメントに接続されていて、複数個の IP アドレスを持つ場合は、IP アドレスおよびホスト名の設定に注意が必要です。

図 4-4 一つのホストを複数のネットワークセグメントに接続している例



この図のようなシステム構成の場合、簡易構築定義ファイルのホストの定義では、次のように設定してください。

- <host-name>タグには、次のどちらかを設定してください。
 - サーバ間、または負荷分散機とサーバ間の通信に使用するネットワークの IP アドレス
 - サーバ間、または負荷分散機とサーバ間の通信に使用するネットワークのホスト名
- <agent-host>タグには、次のどちらかを設定してください。
 - Management Server とサーバ間の通信に使用するネットワークの IP アドレス
 - Management Server とサーバ間の通信に使用するネットワークのホスト名
- 次の表に示すパラメタでホスト名または IP アドレスを指定して、使用するネットワークセグメントを特定してください。各論理サーバで固定にするパラメタを次の表に示します。

表 4-10 固定にするパラメタ

対象になる論理サーバ	パラメタ	対象になるホスト*
Web サーバ	Listen	運用ホスト
J2EE サーバ	mngagent.connector.host	管理ホスト
	vbroker.se.iiop_tp.host	運用ホスト
	webserver.connector.http.bind_host	運用ホスト
	webserver.connector.nio_http.bind_host	運用ホスト
ネーミングサービス	ns.bind.host	運用ホストまたは管理ホスト (ユーザが選択)

対象になる論理サーバ	パラメタ	対象になるホスト※
CTM	ctm.NsBindHost	運用ホスト

注※ 運用ホストとは、簡易構築定義ファイルの<host-name>タグに指定したホストです。管理ホストとは、簡易構築定義ファイルの<agent-host>タグに指定したホストです。

4.1.24 簡易構築定義ファイルの作成例

Web サーバを別ホストに配置するシステムを構築する場合の簡易構築定義ファイルの作成例について説明します。例題で使用する設定内容を次の表に示します。

表 4-11 例題で使用する設定内容（簡易構築定義ファイルの作成例）

分類	項目	設定例
システム全体	簡易構築定義ファイルの名称	MyWebSystemDef.xml
	Web システムの名称	MyWebSystem
	サービスユニットの名称	unit1
Web サーバを配置するホスト	ホストの名称	apsv1
	IP アドレス（管理 IP アドレス）※	192.168.100.11
	論理 Web サーバの名称	MyWebSystem_HTTP01
	論理パフォーマンストレーサの名称	MyWebSystem_PRF01
J2EE サーバを配置するホスト	ホストの名称	apsv2
	IP アドレス（管理 IP アドレス）※	192.168.100.12
	論理 J2EE サーバの名称	MyWebSystem_J2EE01
	論理パフォーマンストレーサの名称	MyWebSystem_PRF02

注※ ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

MyWebSystemDef.xml の作成例

```

<?xml version="1.0" encoding="UTF-8"?>
<model-definition xmlns="http://www.cosminexus.com/mngsvr/schema/ModelDefinition-2.5">
  <web-system>
    <name>MyWebSystem</name>
    <display-name>MyWebSystem</display-name>
    <description>This is a sample.</description>

    <!--Specify the configurations for the Tier.-->
    <tier>
      <tier-type>http-tier</tier-type>
    </tier>
    <tier>
      <tier-type>j2ee-tier</tier-type>
    </tier>
  </web-system>
</model-definition>

```

```

</tier>

<!--Specify the hosts for each Service Unit.-->
<unit>
  <name>unit1</name>
  <allocated-host>
    <host-ref>apsv1</host-ref>
    <hosts-for>http-tier</hosts-for>
    <define-server>
      <logical-server-name>MyWebSystem_HTTP01</logical-server-name>
      <logical-server-type>web-server</logical-server-type>
    </define-server>
    <define-server>
      <logical-server-name>MyWebSystem_PRF01</logical-server-name>
      <logical-server-type>performance-tracer</logical-server-type>
    </define-server>
  </allocated-host>
  <allocated-host>
    <host-ref>apsv2</host-ref>
    <hosts-for>j2ee-tier</hosts-for>
    <define-server>
      <logical-server-name>MyWebSystem_J2EE01</logical-server-name>
      <logical-server-type>j2ee-server</logical-server-type>
    </define-server>
    <define-server>
      <logical-server-name>MyWebSystem_PRF02</logical-server-name>
      <logical-server-type>performance-tracer</logical-server-type>
    </define-server>
  </allocated-host>
</unit>
</web-system>

<!--Specify the settings for the host.-->
<host>
  <host-name>apsv1</host-name>※
  <agent-host>apsv1</agent-host>※
</host>
<host>
  <host-name>apsv2</host-name>※
  <agent-host>apsv2</agent-host>※
</host>
</model-definition>

```

注※ IP アドレスの変更による影響を軽減できるように、<host-name> タグおよび<agent-host> タグは、IP アドレスではなくホスト名で指定することをお勧めします。ただし、ホスト名の変更が発生する場合は、<host-name> タグおよび<agent-host> タグには IP アドレスを指定してください。Smart Composer 機能を使用して、ホスト名を指定して構築したシステムでホスト名を変更する場合は、システムを再構築する必要があります。

4.1.25 システム環境を構築する

システムの一括構築では、情報モデルの生成、その情報モデルを基にした Web サーバおよび J2EE サーバのセットアップ、および各論理サーバの環境設定を一つのコマンドを使用して一括で実施します。

運用管理機能 (Management Server) を配置するホスト (apsv2 (192.168.100.12)) でコマンドプロンプトを起動し、cmx_build_system コマンドを実行して、システムを一括構築します。cmx_build_system コマンドの-f オプションには、簡易構築定義ファイルを指定します。

コマンドの実行例を次に示します。ここでは、簡易構築定義ファイルの名称を MyWebSystemDef.xml とします。

```
cmx_build_system -m 192.168.100.12 -u admin -p admin -f <MyWebSystemDef.xmlのファイルパス>
```

<MyWebSystemDef.xml のファイルパス>には、MyWebSystemDef.xml を格納したディレクトリのパスとファイル名 (MyWebSystemDef.xml) を指定します。

4.1.26 システムを起動する (CUI 利用時)

システムが稼働状態になると、サービスユニット内の論理サーバが起動されて、リクエストの受け付けや処理が実施されるようになります。

参考

論理サーバの起動順序は、デフォルトの設定を使用することをお勧めします。なお、論理サーバの起動順序は、簡易構築定義ファイルで各論理サーバの<configuration>タグ内に mstartup.no パラメタで設定できます。

運用管理機能 (Management Server) を配置するホスト (apsv2 (192.168.100.12)) でコマンドプロンプトを起動し、cmx_start_target コマンドを実行して、Web システム内のすべてのサービスユニットを稼働状態で起動します。稼働状態にするには、-mode オプションに「ALL」を指定します。

コマンドの実行例を次に示します。ここでは、Web システムの名称を MyWebSystem とします。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL -strict
```

cmx_start_target コマンドの-s オプションには、簡易構築定義ファイル (MyWebSystemDef.xml) の<web-system>タグの<name>タグで指定した Web システム名を指定します。

4.1.27 リソースアダプタの設定とは

J2EE サーバ上の J2EE アプリケーションからデータベースに接続するためには、リソースアダプタを使用します。J2EE サーバでは、データベースのほかに、OpenTP1、SMTP サーバ、JavaBeans リソースなどのリソースが利用できます。リソースの種類によっては、接続にリソースアダプタを使用するものと、リソースアダプタを使用しないものがあります。リソースの種類と接続方法については、マニュアル「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「3.3.1 リソースへの接続方法」を参照してください。

また、リソースの接続に DB Connector を使用する場合と、ほかのリソースアダプタを使用する場合で、Connector 属性ファイルの取得方法が異なります。DB Connector を使用する場合は、システムを起動する前に Connector 属性ファイルのテンプレートを利用して編集できます。DB Connector 以外のリソースアダプタを使用する場合は、システムを起動したあと、Connector 属性ファイルを取得し、編集します。また、リソースアダプタのインポートと開始には、サーバ管理コマンドを使用します。サーバ管理コマンドでのリソースアダプタの操作については、マニュアル「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「4. リソースアダプタの設定」を参照してください。

4.1.28 DB Connector を設定する (CUI 利用時)

リソースアダプタとして DB Connector を使用する場合の設定手順について説明します。この設定は、J2EE サーバを配置するホストで実施します。例題で使用する設定内容を次の表に示します。

表 4-12 例題で使用する設定内容 (DB Connector の設定)

項目	設定例
リソースアダプタのファイル名	DBConnector_HiRDB_Type4_CP.rar*
リソースアダプタの表示名	DB_Connector_for_HiRDB_Type4
Connector 属性ファイルのファイル名	MyDBConnectorAttr.xml
リソースアダプタをインポートする J2EE サーバの名称	MyWebSystem_J2EE01

注※ HiRDB Type4 JDBC Driver を使用して、ローカルトランザクションまたはトランザクション管理なしで HiRDB に接続する場合に使用する RAR ファイルです。

なお、提供される RAR ファイルの種類や使い分けについては、マニュアル「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「3.6.4 DB Connector (RAR ファイル) の種類」を参照してください。

(1) Connector 属性ファイルの取得と編集

1. Connector 属性ファイルを取得します。

アプリケーションサーバでは、DB Connector の Connector 属性ファイルのテンプレートファイルを提供しています。テンプレートファイルの格納先を次に示します。

Windows の場合

<Application Server のインストールディレクトリ>\CC\admin\templates

UNIX の場合

/opt/Cosminexus/CC/admin/templates

DBConnector_HiRDB_Type4_CP.rar を使用する場合、

DBConnector_HiRDB_Type4_CP_cfg.xml を取得します。

Connector 属性ファイルのテンプレート、およびテンプレートファイルに関する注意事項については、マニュアル「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「4.1 Connector 属性ファイル」を参照してください。

2. Connector 属性ファイルを編集します。

取得した Connector 属性ファイルを任意の場所に保存します。ここでは、Connector 属性ファイルの名称を MyDBConnectorAttr.xml として保存します。

保存した Connector 属性ファイルにデータベースへの接続の設定などの必要な項目を定義してください。

(2) リソースアダプタの設定

1. J2EE サーバを配置するホストでコマンドプロンプトを起動します。

2. cjimportres コマンドを実行し、リソースアダプタを J2EE サーバへインポートします。

コマンドの実行例を次に示します。

```
cjimportres MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -type rar -f <DB  
Connector_HiRDB_Type4_CP.rar のファイルパス>
```

<DBConnector_HiRDB_Type4_CP.rar のファイルパス>は、Windows の場合は「<Application Server のインストールディレクトリ>\CC\DBConnector\DBConnector_HiRDB_Type4_CP.rar」、UNIX の場合は「/opt/Cosminexus/CC/DBConnector/DBConnector_HiRDB_Type4_CP.rar」です。

3. cjdeployrar コマンドを実行し、J2EE サーバにリソースアダプタをデプロイします。

コマンドの実行例を次に示します。

```
cjdeployrar MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Con  
nector_for_HiRDB_Type4
```

4. cjsetrarprop コマンドを実行し、「(1) Connector 属性ファイルの取得と編集」で編集した Connector 属性ファイルをリソースアダプタに反映します。

コマンドの実行例を次に示します。

```
cjsetrarprop MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Con  
nector_for_HiRDB_Type4 -c <MyDBConnectorAttr.xml のファイルパス>
```

<MyDBConnectorAttr.xml のファイルパス>には、「(1) Connector 属性ファイルの取得と編集」で MyDBConnectorAttr.xml を格納したディレクトリのパスとファイル名 (MyDBConnectorAttr.xml) を指定します。

(3) リソースアダプタの接続テスト

1. cjtestres コマンドを実行し、リソースアダプタの接続を確認します。

コマンドの実行例を次に示します。

```
cjtestres MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Connector_for_HiRDB_Type4
```

4.1.29 DB Connector 以外のリソースアダプタを設定する (CUI 利用時)

リソースアダプタとして DB Connector 以外を使用する場合の設定手順について説明します。この設定は、J2EE サーバを配置するホストで実施します。例題で使用する設定内容を次の表に示します。

表 4-13 例題で使用する設定内容 (DB Connector 以外のリソースアダプタの設定)

項目	設定例
リソースアダプタのファイル名	MyResourceAdapter.rar
リソースアダプタの表示名	MyResourceAdapter
Connector 属性ファイルのファイル名	MyResourceAdapter.xml
リソースアダプタをインポートする J2EE サーバの名称	MyWebSystem_J2EE01

なお、提供される RAR ファイルの種類や使い分けについては、マニュアル「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「3.3.2 リソースアダプタの種類」を参照してください。

(1) リソースアダプタの設定

1. J2EE サーバを配置するホストでコマンドプロンプトを起動します。

2. cjimportres コマンドを実行し、リソースアダプタを J2EE サーバへインポートします。

コマンドの実行例を次に示します。

```
cjimportres MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -type rar -f <MyResourceAdapter.rarのファイルパス>
```

<MyResourceAdapter.rar のファイルパス>には、MyResourceAdapter.rar を格納したディレクトリのパスとファイル名 (MyResourceAdapter.rar) を指定します。

3. cjdeployrar コマンドを実行し、J2EE サーバにリソースアダプタをデプロイします。

コマンドの実行例を次に示します。

```
ejdeployrar MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname MyResourceAdapter
```

(2) Connector 属性ファイルの取得と編集

1. J2EE サーバを配置するホストでコマンドプロンプトを起動します。

2. cjgetrarprop コマンドを実行し、Connector 属性ファイルを取得します。

コマンドの実行例を次に示します。

```
cjgetrarprop MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname MyResourceAdapter -c <Connector属性ファイルの取得先のファイルパス>
```

<Connector 属性ファイルの取得先のファイルパス>には、Connector 属性ファイルを格納するディレクトリのパスとファイル名（MyResourceAdapter.xml）を指定します。

3. Connector 属性ファイルを編集します。

取得した Connector 属性ファイルで、データベースへの接続の設定など必要な項目を編集してください。

4. cjsetrarprop コマンドを実行し、編集した Connector 属性ファイルをリソースアダプタに反映します。

コマンドの実行例を次に示します。

```
cjsetrarprop MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname MyResourceAdapter -c <Connector属性ファイルの取得先のファイルパス>
```

<Connector 属性ファイルの取得先のファイルパス>には、手順 3.で編集した Connector 属性ファイルが格納されているディレクトリのパスとファイル名（MyResourceAdapter.xml）を指定します。

(3) リソースアダプタの接続テスト

1. cjtestres コマンドを実行し、リソースアダプタの接続を確認します。

コマンドの実行例を次に示します。

```
cjtestres MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname MyResourceAdapter
```

4.1.30 リソースアダプタを開始する (CUI 利用時)

設定したリソースアダプタを開始する手順について説明します。リソースアダプタを設定した、J2EE サーバが稼働しているアプリケーションサーバで開始します。

アプリケーションサーバのコマンドプロンプトで、cjstarrar コマンドを実行し、リソースアダプタを開始します。ここでは、リソースアダプタとして DB Connector (DB_Connector_for_HiRDB_Type4) を開始するとします。

コマンドの実行例を次に示します。

```
ejstartrar MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Connecto  
r_for_HiRDB_Type4
```

4.1.31 業務アプリケーションを設定して開始する (CUI 利用時)

業務を開始するためには、J2EE アプリケーションをインポートして開始する必要があります。J2EE アプリケーションのインポートと開始には、サーバ管理コマンドを使用します。J2EE アプリケーションを設定する、J2EE サーバが稼働しているアプリケーションサーバで設定します。サーバ管理コマンドでの J2EE アプリケーションの操作の詳細については、マニュアル「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」の次の個所を参照してください。

- 8.1 J2EE アプリケーションのインポート
- 10.2.1 J2EE アプリケーションの開始

ここで説明する J2EE アプリケーションは、cosminexus.xml を含むアプリケーションとして作成済みであることを前提としています。例題で使用する設定内容を次の表に示します。

表 4-14 例題で使用する設定内容 (J2EE アプリケーションの設定と開始)

項目	設定内容
J2EE アプリケーションのファイル名	MyApp.ear
J2EE アプリケーションの表示名	MyApp
J2EE アプリケーションをインポートする J2EE サーバの名称	MyWebSystem_J2EE01

1. J2EE サーバを配置するホストでコマンドプロンプトを起動します。

2. cjimportapp コマンドを実行し、J2EE アプリケーションを J2EE サーバへインポートします。

コマンドの実行例を次に示します。

```
cjimportapp MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -f MyApp.ear
```

3. cjstartapp コマンドを実行し、J2EE アプリケーションを開始します。

コマンドの実行例を次に示します。

```
cjstartapp MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -name MyApp
```

4.1.32 リソースアダプタまたは業務アプリケーションを操作するときの確認事項 (CUI 利用時)

(1) サーバ管理コマンド実行時に設定する情報

リソースアダプタまたは業務アプリケーションを設定したり開始したりする場合、サーバ管理コマンドを使用します。サーバ管理コマンドで引数にサーバ名を指定する場合は、実サーバ名を指定します。簡易構築定義ファイル作成時に実サーバ名を設定していない場合、実サーバ名は論理サーバ名と同じです。なお、論理サーバ名の設定を省略している場合、論理サーバ名は「cmx_<Web システム名>_<サービスユニット名>_<種別>_<通番>」の形式で定義されます。論理サーバの種類と<種別>との対応を次の表に示します。

表 4-15 論理サーバの種類と<種別>との対応

論理サーバの種類	<種別>
論理 Web サーバ	HTTP
論理 J2EE サーバ	J2EE
論理パフォーマンストレーサ	PRF
論理 CTM ドメインマネジャ	CTMDM
論理 CTM	CTM
論理スマートエージェント	SA
論理ユーザサーバ	USER

なお、<通番>は、基本的に「01」が設定されます。ただし、CTM を使用するシステムのように論理サーバを複数配置する場合は、重複しない番号が割り振られます。<通番>は、cmx_list_model コマンドで確認してください。

(2) サーバ管理コマンドを使用する操作

サーバ管理コマンドを使用する操作は、サービスユニットごとに実行します。

このマニュアルで説明しているシステムは、サービスユニットが一つの例です。複数のサービスユニットを設定している場合は、コマンド実行例中の J2EE サーバ名および-nameserver 指定値の値を、それぞれのサービスユニットごとに変更して、サーバ管理コマンドを実行してください。

(3) J2EE アプリケーションのプロパティについて

このマニュアルで説明している J2EE アプリケーションでは、プロパティの設定に cosminexus.xml を使用します。このため、あらかじめ作成した cosminexus.xml を、EJB-JAR、WAR などと一緒に J2EE アプリケーションに含めて使用してください。

次に示す場合、J2EE アプリケーションのプロパティは、属性ファイル（アプリケーション統合属性ファイル）で設定できます。

- ・インポート済みの J2EE アプリケーションのプロパティを編集する場合
- ・cosminexus.xml を含まないアプリケーションの場合

設定方法については、マニュアル「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「9.2 アプリケーション統合属性ファイルによるプロパティ設定」を参照してください。

4.1.33 システム環境定義（簡易構築定義ファイルのパラメタ）を変更する（CUI 利用時）

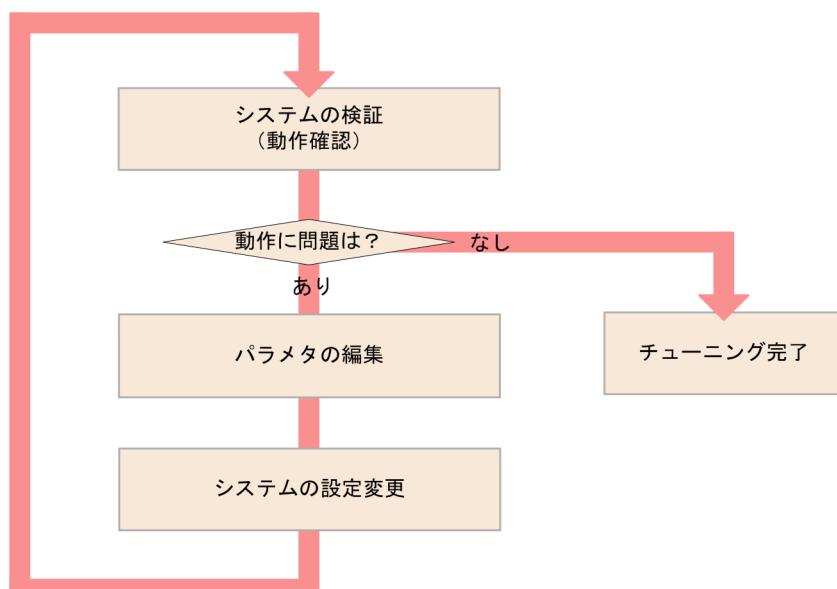
簡易構築定義ファイルのパラメタだけを編集する場合のチューニングの手順について説明します。例題で使用する設定内容を次の表に示します。

表 4-16 例題で使用する設定内容（システム環境定義の変更）

項目	設定例
Web システムの名称	MyWebSystem
簡易構築定義ファイルの名称	MyWebSystemDef.xml
運用管理機能（Management Server）を配置するホスト	apsv2 (192.168.100.12)

簡易構築定義ファイルのパラメタだけを編集する場合のチューニングの流れを次の図に示します。

図 4-5 簡易構築定義ファイルのパラメタだけを編集する場合のチューニングの流れ



(1) パラメタの編集

簡易構築定義ファイル (MyWebSystemDef.xml) のパラメタ (<param>タグで指定する論理サーバのパラメタ) の値を編集して、チューニングを実施します。チューニングできる項目、およびパフォーマンスチューニングの詳細については、マニュアル「アプリケーションサーバ システム設計ガイド」の次の個所を参照してください。

- 8.1.3 アプリケーションの種類ごとにチューニングできる項目
- 8.2 チューニングの方法

(2) システムの設定変更

運用管理機能 (Management Server) を配置するホストで cmx_build_system コマンドを実行し、Management Server 上のシステムの情報モデルを変更します。変更した情報モデルを実システムに適用します。

1. 運用管理機能 (Management Server) を配置するホストでコマンドプロンプトを起動します。
2. システムが稼働している場合は、cmx_stop_target コマンドを実行し、システムを停止します。
システム内のすべてのサービスユニットを停止状態にします。停止状態にするには、-mode オプションに「ALL」を指定します。
コマンドの実行例を次に示します。

```
cmx_stop_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL
```

3. cmx_build_system コマンドを実行し、システムの情報モデルを変更して実システムに展開します。
Management Server 上のシステムの情報モデルを変更し、変更した情報モデルからシステムを構築します。-f オプションには、パフォーマンスチューニングを実施した簡易構築定義ファイルを指定します。-f オプションで指定したファイルの内容をシステムに反映するには、-change オプションを指定します。
コマンドの実行例を次に示します。

```
cmx_build_system -f <MyWebSystemDef.xmlのファイルパス> -change -m 192.168.100.12 -u admin -p admin -s MyWebSystem
```

<MyWebSystemDef.xml のファイルパス>には、「(1) パラメタの編集」で編集した簡易構築定義ファイルの格納先ディレクトリのパスとファイル名 (MyWebSystemDef.xml) を指定します。

4. cmx_start_target コマンドを実行し、構成を変更したシステムを起動します。
システム内のすべてのサービスユニットを稼働状態にします。稼働状態にするには、-mode オプションに「ALL」を指定します。
コマンドの実行例を次に示します。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL -strict
```

4.1.34 業務アプリケーションの設定を変更する (CUI 利用時)

J2EE アプリケーションの設定を変更する場合のチューニングの手順について説明します。例題で使用する設定内容を次の表に示します。

表 4-17 例題で使用する設定内容 (業務アプリケーションの設定変更)

項目	設定例
Web システムの名称	MyWebSystem
J2EE アプリケーションのファイル名	MyApp.ear
J2EE アプリケーションの表示名	MyApp

J2EE アプリケーションの設定を変更する、J2EE サーバが稼働しているアプリケーションサーバで実施します。

1. cosminexus.xml で J2EE アプリケーションのプロパティを編集して、<任意のディレクトリ>下の META-INF ディレクトリに格納します。
2. コマンドプロンプトで、jar コマンドを実行し、アーカイブ形式の J2EE アプリケーションに手順 1.で編集した cosminexus.xml を上書きします。

コマンドの実行例を次に示します。

```
jar uvf <MyApp.earのファイルパス> -C <META-INFディレクトリを置いたディレクトリ> <META-INF  
ディレクトリ下のcosminexus.xmlのファイルパス>
```

<MyApp.ear のファイルパス> は、J2EE アプリケーション (MyApp.ear) を格納したディレクトリのパスとファイル名 (MyApp.ear) を指定します。<META-INF ディレクトリを置いたディレクトリ> は、cosminexus.xml の格納先が次のような構造の場合、<任意のディレクトリ> となります。

```
<任意のディレクトリ>  
└META-INF  
  └cosminexus.xml
```

<META-INF ディレクトリ下の cosminexus.xml のファイルパス> は、手順 1.で cosminexus.xml を格納した META-INF ディレクトリのパスとファイル名 (cosminexus.xml) を指定します。

3. システムが停止している場合は、運用管理機能 (Management Server) を配置するホスト (192.168.100.12) でコマンドプロンプトを起動して、cmx_start_target コマンドを実行し、システムを起動します。

システム内のすべてのサービスユニットを準備状態にします。準備状態にするには、-mode オプションに「READY」を指定します。

コマンドの実行例を次に示します。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode READY
```

4. J2EE サーバを配置するホストでコマンドプロンプトを起動します。

5. cjreplaceapp コマンドを実行し、再作成した J2EE アプリケーションをリデプロイします。

コマンドの実行例を次に示します。

```
cjreplaceapp MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -name MyApp -f  
<再作成したJ2EEアプリケーションのファイルパス>
```

<再作成した J2EE アプリケーションのファイルパス>には、再作成した MyApp.ear を格納したディレクトリのパスとファイル名 (MyApp.ear) を指定します。

6. cjstartapp コマンドを実行し、J2EE アプリケーションを開始します。

コマンドの実行例を次に示します。

```
cjstartapp MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -name MyApp
```

参考

インポート済みの J2EE アプリケーションのプロパティは、属性ファイル（アプリケーション統合属性ファイル）でも変更できます。変更方法については、マニュアル「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「9.2 アプリケーション統合属性ファイルによるプロパティ設定」を参照してください。

7. 運用管理機能 (Management Server) を配置するホスト (192.168.100.12) でコマンドプロンプトを起動して、cmx_start_target コマンドを実行し、システムを起動します。

システム内のすべてのサービスユニットを稼働状態にします。稼働状態にするには、-mode オプションに「ALL」を指定します。

コマンドの実行例を次に示します。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL -strict
```

4.1.35 リソースアダプタの設定（データベースの接続環境）を変更する（CUI 利用時）

リソースアダプタの設定を変更する場合のチューニングの手順について説明します。例題で使用する設定内容を次の表に示します。

表 4-18 例題で使用する設定内容（リソースアダプタの設定変更）

項目	設定例
Web システムの名称	MyWebSystem
リソースアダプタの表示名	DB_Connector_for_HiRDB_Type4
Connector 属性ファイルのファイル名	MyDBConnectorAttr.xml
リソースアダプタをインポートする J2EE サーバの名称	MyWebSystem_J2EE01

リソースアダプタの設定を変更する、J2EE サーバが稼働しているアプリケーションサーバで実施します。

1. システムが停止している場合は、運用管理機能 (Management Server) を配置するホスト (192.168.100.12) でコマンドプロンプトを起動して、cmx_start_target コマンドを実行し、システムを起動します。

システム内のすべてのサービスユニットを準備状態にします。準備状態にするには、-mode オプションに「READY」を指定します。

コマンドの実行例を次に示します。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode READY
```

2. J2EE サーバを配置するホストでコマンドプロンプトを起動します。

3. cjgetrarprop コマンドを実行し、Connector 属性ファイルを取得します。

コマンドの実行例を次に示します。

```
cjgetrarprop MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Connector_for_HiRDB_Type4 -c <Connector属性ファイルの取得先のファイルパス>
```

<Connector 属性ファイルの取得先のファイルパス>には、Connector 属性ファイルを格納するディレクトリのパスとファイル名 (MyDBConnectorAttr.xml) を指定します。

4. Connector 属性ファイルを編集します。

取得した Connector 属性ファイルで、データベースへの接続の設定など必要な項目を編集してください。

5. cjsetrarprop コマンドを実行し、編集した Connector 属性ファイルをリソースアダプタに反映します。

コマンドの実行例を次に示します。

```
cjsetrarprop MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Connector_for_HiRDB_Type4 -c <Connector属性ファイルの取得先のファイルパス>
```

<Connector 属性ファイルの取得先のファイルパス>には、手順 4.で編集した Connector 属性ファイルが格納されているディレクトリのパスとファイル名 (MyDBConnectorAttr.xml) を指定します。

6. cjstartrar コマンドを実行し、リソースアダプタを開始します。

コマンドの実行例を次に示します。

```
cjstartrar MyWebSystem_J2EE01 -nameserver corbaname::192.168.100.12:900 -resname DB_Connector_for_HiRDB_Type4
```

7. 運用管理機能 (Management Server) を配置するホスト (192.168.100.12) でコマンドプロンプトを起動して、cmx_start_target コマンドを実行し、システムを起動します。

システム内のすべてのサービスユニットを稼働状態にします。稼働状態にするには、-mode オプションに「ALL」を指定します。

コマンドの実行例を次に示します。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL -strict
```

4.1.36 システムを停止する (CUI 利用時)

業務が実行中の場合は、すべてのリクエストの処理が完了したあと、サービスユニット内の論理サーバを停止します。

運用管理機能 (Management Server) を配置するホスト (apsv2 (192.168.100.12)) でコマンドプロンプトを起動し、cmx_stop_target コマンドを実行して、Web システム内のすべてのサービスユニットを停止状態で停止します。停止状態にするには-mode オプションに「ALL」を指定します。

1. 運用管理機能 (Management Server) を配置するホストでコマンドプロンプトを起動します。
2. 実行中のリクエストの処理が完了したら、cmx_stop_target コマンドを実行し、システム内のすべてのサービスユニットを一括停止します。

システム内のすべてのサービスユニットを停止状態にします。停止状態にするには、-mode オプションに「ALL」を指定します。

コマンドの実行例を次に示します。

```
cmx_stop_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL
```

サービスユニット内のすべての論理サーバが停止されます。

4.1.37 Management Server を停止する

Management Server をセットアップしたホスト（運用管理機能を配置するホスト）でコマンドプロンプトを起動し、mngsvrctl コマンドを実行して、Management Server を停止します。なお、Windows の場合はサービスからも停止できます。

コマンドの実行例を次に示します。

```
mngsvrctl stop -sync
```

4.1.38 運用管理エージェントを停止する

アプリケーションサーバごとにコマンドプロンプトを起動し、adminagentctl コマンドを実行して、運用管理エージェントを停止します。なお、Windows の場合はサービスからも停止できます。

コマンドの実行例を次に示します。

```
adminagentctl stop -sync
```

4.1.39 システム環境を削除する

不要になったシステムを削除する手順について説明します。

1. J2EE アプリケーションやリソースアダプタを停止します。

J2EE アプリケーションやリソースアダプタは、サーバ管理コマンドを使用して停止します。

2. コマンドプロンプトを起動します。

手順 2.～手順 6.は、Web サーバを配置するホストと、J2EE サーバを配置するホストの両方で実施します。なお、ホストの実施順序は順不同です。

3. 次のプロセスが起動している場合、これらのプロセスを停止します。

- Management Server を使用しないで起動された CTM および OTM (TPBroker Object Transaction Monitor) のプロセス

4. UNIX の場合、運用管理エージェントおよび Management Server で自動再起動を設定しているときは mngautorun コマンドを実行し、自動再起動の設定を解除してから OS を再起動します。

コマンドの実行例を次に示します。

```
mngautorun -d
```

5. mngunsetup コマンドを実行し、表示されたメッセージを確認して「y」を入力します。

アンセットアップを実行するかどうかを確認するメッセージが表示されます。

「y」を入力すると、次の処理が実行されます。

- 運用管理エージェント停止、および Management Server の停止と削除
- 運用管理エージェントおよび Management Server の自動起動および自動再起動の解除
- 論理サーバの停止

論理パフォーマンストレーサ、論理スマートエージェント、論理ネーミングサービス、論理 CTM ドメインマネージャ、論理 CTM (同時に起動されるネーミングサービスも含む)、論理 J2EE サーバ、論理 Web サーバ、および論理ユーザサーバ※を停止します。停止監視時間を経過しても停止しない論理サーバは、Management Server が強制停止します。なお、各論理サーバの停止監視時間 (簡易構築定義ファイルの mstartup.watchtime パラメタ) は、論理サーバを起動したときに設定されている値が有効になります。

注※ 論理ユーザサーバは監視対象プロセスを停止します。停止用コマンドが指定されている場合は、停止用コマンドでプロセスを停止し、指定されていない場合は、起動種別が直接起動のときは起動用コマンドのプロセス、間接起動のときは論理ユーザサーバに定義されたプロセス ID 取得用コマンドで取得されるプロセスを停止します。

• 論理サーバの強制停止

論理パフォーマンストレーサ、論理スマートエージェント、論理ネーミングサービス、論理 CTM ドメインマネージャ、論理 CTM (同時に起動されるネーミングサービスも含む)、論理 J2EE サーバ、論理 Web サーバ、および論理ユーザサーバ※を強制停止します。強制停止監視時間を経過しても

停止しない論理サーバは、Management Server がそのプロセスを強制停止します。なお、各論理サーバの強制停止監視時間（簡易構築定義ファイルの mstartup.force.watchtime パラメタ）は、論理サーバを起動したときに設定されている値が有効になります。

注※ 論理ユーザサーバは監視対象プロセスを強制停止します。強制停止用コマンドが指定されている場合は、強制停止用コマンドでプロセスを停止し、指定されていない場合は、起動種別が直接起動のときは起動用コマンドのプロセス、間接起動のときは論理ユーザサーバに定義されたプロセス ID 取得用コマンドで取得されるプロセスを停止します。

- 論理 J2EE サーバ、および論理 Web サーバの環境（実サーバ）削除、論理 CTM ドメインマネジャの情報削除
- Management Server が管理している情報（論理サーバの構成情報や設定情報、および管理ユーザアカウントの情報）の初期化
- Component Container 管理者の設定解除（UNIX の場合）

設定が解除され、スーパーユーザのユーザ（オーナー）/グループが設定されます。設定値を OS ごとに示します。

AIX の場合 : root/system

Linux の場合 : root/root

6. UNIX の場合、必要に応じて、mngenvsetup コマンドを実行し、Management Server 管理グループの設定を解除します。

コマンドの実行例を次に示します。

```
mngenvsetup -mng -d
```

注意事項

アンセットアップに成功しても、システム環境によっては Application Server で起動されたプロセスが残ることがあります。この場合、Application Server のアンインストールに失敗することがあります。アンセットアップ後にアンインストールを実施して失敗したときは、「[付録 I.6 Application Server のインストールおよびアンインストール時のエラーコード一覧（Windows の場合）](#)」、または「[付録 I.7 Component Container およびリダイレクタ機能のインストールおよびアンインストール時のログ（UNIX の場合）](#)」を参照してエラーの対処を実施したあと、再度アンインストールを実施してください。

参考

mngunsetup コマンドを実行すると、Management Server および運用管理エージェントもアンセットアップされます。構築したシステム環境（論理サーバやホスト）だけを削除したい場合は、J2EE アプリケーションやリソースを停止したあと、次の手順を実施してください。ここでは、Web システムの名称を MyWebSystem、運用管理機能（Management Server）を配置するホストを apsv2（192.168.100.12）とします。

1. 運用管理機能 (Management Server) を配置するホストでコマンドプロンプトを起動します。
2. cmx_stop_target コマンドを実行し、システムを停止します。
削除対象となるシステムを停止します。
コマンドの実行例を次に示します。

```
cmx_stop_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -mode ALL
```

3. cmx_delete_system コマンドを使用して Management Server 上からシステムを削除します。
Management Server で管理されているシステムの情報モデルを削除します。
コマンドの実行例を次に示します。

```
cmx_delete_system -m 192.168.100.12 -u admin -p admin -s MyWebSystem
```

4. Web サーバや J2EE サーバの環境をアンセットアップします。
cmx_delete_system コマンドでは、サーバの環境はアンセットアップされません。ホスト上の Web サーバや J2EE サーバの環境は、それぞれ次の方法でアンセットアップします。

- Web サーバの環境のアンセットアップ

HTTP Server の hwsserveredit コマンドを使用して、Web サーバをアンセットアップします。HTTP Server の hwsserveredit コマンドには、「HWS_」を付加した Web サーバの論理サーバ名※を指定します。

- J2EE サーバの環境のアンセットアップ

cjsetup コマンドを使用して、J2EE サーバをアンセットアップします。cjsetup コマンドには、J2EE サーバの論理サーバ名※を指定します。

注※

論理サーバ名は cmx_list_model コマンドを使用して確認できます。cmx_delete_system コマンドを実行する前に論理サーバ名を確認してください。

ただし、システム構築時に、Web サーバや J2EE サーバの実サーバ名を指定している場合は、アンセットアップ時には実サーバ名を指定してください。実サーバ名の指定は、簡易構築定義ファイルの物理ティアの realservername パラメタに指定します。

アンセットアップ時に指定するサーバ名の詳細は、マニュアル「アプリケーションサーバリファレンス コマンド編」の「cjsetup (J2EE サーバのセットアップとアンセットアップ)」を参照してください。

実サーバ名を指定していない J2EE サーバをアンセットアップする場合の、コマンドの実行例を次に示します。

```
cjsetup -d MyWebSystem_J2EE01
```

4.2 ホストを増やしてシステムの規模を拡大するシステムの運用（スケールアウト）

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、ホストの台数を増やすことをいいます。システムをスケールアウトすると、業務全体を停止することなく、システムの規模を拡張して構成を変更できます。

4.2.1 ホストを増やしてシステムの規模を拡大（スケールアウト）するには

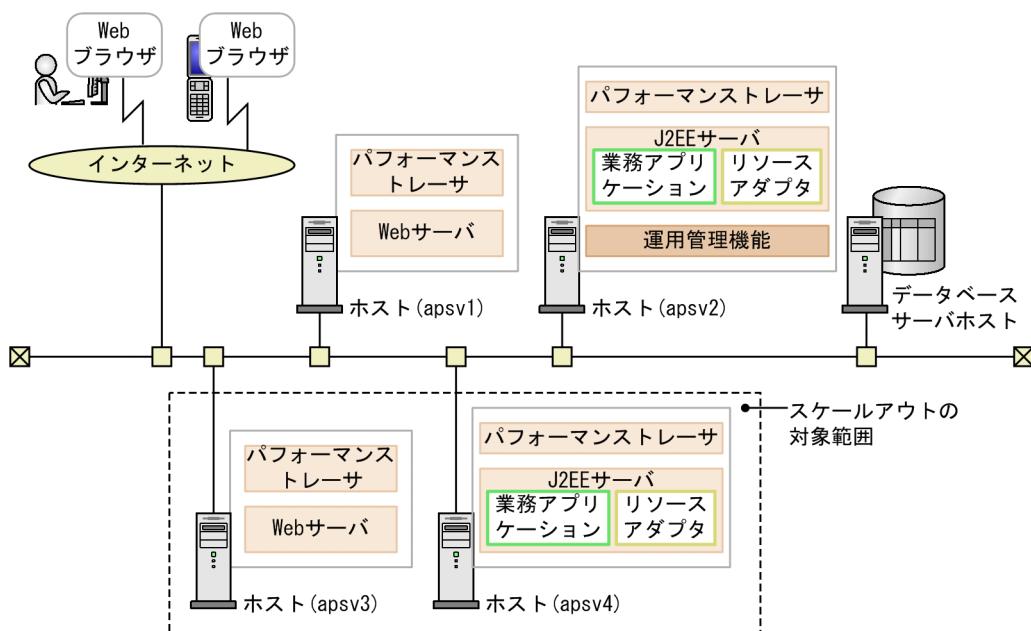
スケールアウトでは、ホストの台数を増やし、システムの規模を拡大することで、システム全体の処理性能の向上が図れます。

Smart Composer 機能の場合は、サービスユニット単位にホストおよび論理サーバを一括で増やせます。また、運用管理ポータルの場合は、[運用管理ポータル] 画面の各画面で個々にホストや論理サーバを増やせます。

- システム運用例

ホストを増やしてシステムの規模を拡大するシステム運用例を次の図に示します。ここでは、「4.1 Web サーバを別ホストに配置してマシン性能を向上するシステムの構築」で構築したシステムを使用します。

図 4-6 ホストを増やしてシステムの規模を拡大するシステム運用例



例題で使用する設定内容を次の表に示します。

表 4-19 例題で使用する設定内容（システムのスケールアウト）

分類	項目	設定例
システム全体	構成変更定義ファイルの名称	MyWebSystemAdd.xml
	Web システムの名称	MyWebSystem
	追加するサービスユニットの名称	unit2
追加する Web サーバ を配置するホスト	ホストの名称	apsv3
	IP アドレス（管理 IP アドレス）※	192.168.100.13
	論理 Web サーバの名称	MyWebSystem_HTTP02
	論理パフォーマンストレーサの名称	MyWebSystem_PRF03
追加する J2EE サーバ を配置するホスト	ホストの名称	apsv4
	IP アドレス（管理 IP アドレス）※	192.168.100.14
	論理 J2EE サーバの名称	MyWebSystem_J2EE02
	論理パフォーマンストレーサの名称	MyWebSystem_PRF04
	リソースアダプタのファイル名	DBConnector_HiRDB_Type4_CP.rar
	リソースアダプタの表示名	DB_Connector_for_HiRDB_Type4
	Connector 属性ファイルのファイル名	MyDBConnectorAttr.xml
	J2EE アプリケーションのファイル名	MyApp.ear
	J2EE アプリケーションの表示名	MyApp

注※ ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

• 実施する作業と作業の要否

システムをスケールアウトする時に実施する作業と作業の要否を次の表に示します。システムのスケールアウトは、Smart Composer 機能を使用して実施します。

表 4-20 システムをスケールアウトする時に実施する作業と作業の要否（Smart Composer 機能利用時）

システムをスケールアウトする時に実施する作業	作業の要否	作業の実施条件	参照箇所
システムの構築	○	必ず実施	4.1
追加するホストの準備	○	必ず実施	4.2.2(1)
構成変更定義ファイルの作成	○	必ず実施	4.2.2(2)
システムの情報モデルの変更	○	必ず実施	4.2.2(3)
サービスユニットの追加	○	必ず実施	4.2.2(4)

（凡例）

○：必ず実施する作業です。

4.2.2 システム環境をスケールアウトする (CUI 利用時)

ここでは、Smart Composer 機能を使用して、サービスユニット (unit2) を追加してスケールアウトする手順について説明します。

(1) 追加するホストの準備

システムをスケールアウトする前に、あらかじめ追加するアプリケーションサーバの OS やアプリケーションをセットアップしてください。追加するホストで実施する作業と作業の要否を次の表に示します。

表 4-21 追加するホストで実施する作業と作業の要否

追加するホストで実施する作業	作業の要否	作業の実施条件	参照箇所
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき	4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合	4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合	4.1.6
		データベース (Oracle) に接続する場合	4.1.7
		データベース (SQL Server) に接続する場合	4.1.8
		データベース (MySQL) に接続する場合	4.1.9
		データベース (PostgreSQL) に接続する場合	4.1.10
システムの環境変数の設定	○	必ず実施	4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合	4.1.14
運用管理エージェントの起動	○	必ず実施	4.1.19

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

また、Web システムの構成を変更すると Web システムが動作するために必要なリソースも変更する必要があります。Web システムが動作するために必要なリソースの見積もりについては、マニュアル「アプリケーションサーバ システム設計ガイド」の「5. 使用するリソースの見積もり (J2EE アプリケーション実行基盤)」を参照してください。

(2) 構成変更定義ファイルの作成

サービスユニット・ホスト追加用の構成変更定義ファイルを任意の場所にコピーして、ファイルの内容を編集します。

- 構成変更定義ファイルのテンプレートファイル (cmxaddcombinedmodel.xml) を任意の場所にコピーします。

サービスユニットやホストを追加する構成変更定義ファイルのテンプレートファイルは、次の場所に提供されています。

Windows の場合

```
<Application Server のインストールディレクトリ>¥manager¥config¥templates¥cmxaddcombinedmodel.xml
```

UNIX の場合

```
/opt/Cosminexus/manager/config/templates/cmxaddcombinedmodel.xml
```

- 手順 1 のテンプレートファイルを任意の名称を変更します。

ここではファイル名を「MyWebSystemAdd.xml」に変更します。

- テキストエディタなどを利用して、構成変更定義ファイルに追加するサービスユニットおよびホストを定義します。

MyWebSystemAdd.xml の作成例

```
<?xml version="1.0" encoding="UTF-8"?>
<unit-addition xmlns="http://www.cosminexus.com/mngsvr/schema/UnitAddition-2.5">
  <web-system>
    <name>MyWebSystem</name>

    <!--Specify the settings for the Service Unit to add.-->
    <unit>
      <name>unit2</name>
      <allocated-host>
        <host-ref>apsv3</host-ref>
        <hosts-for>http-tier</hosts-for>
      </allocated-host>
      <allocated-host>
        <host-ref>apsv4</host-ref>
        <hosts-for>j2ee-tier</hosts-for>
      </allocated-host>
    </unit>
  </web-system>

  <!--Specify the settings for the host to add.-->
  <host>
```

```
<host-name>apsv3</host-name>
<agent-host>apsv3</agent-host>
</host>
<host>
  <host-name>apsv4</host-name>
  <agent-host>apsv4</agent-host>
</host>
</unit-addition>
```

注 背景色付きの太字の部分が編集部分です。

(3) システムの情報モデルの変更

作成した構成変更定義ファイルの内容を、Management Server に登録されているシステムの情報モデルに反映します。

1. 運用管理機能 (Management Server) を配置するホストでコマンドプロンプトを起動します。
2. cmx_change_model コマンドを実行し、Management Server に登録されているシステムの情報モデルを変更します。

コマンドの実行例を次に示します。

```
cmx_change_model -m 192.168.100.12 -u admin -p admin -f <MyWebSystemAdd.xmlのファイルパス>
```

< MyWebSystemAdd.xml のファイルパス >には、MyWebSystemAdd.xml を格納したディレクトリのパスとファイル名 (MyWebSystemAdd.xml) を指定します。

(4) サービスユニットの追加

変更したシステムの情報モデルを実システムに適用して、サービスユニットを追加します。

1. 運用管理機能 (Management Server) を配置するホストでコマンドプロンプトを起動します。
2. cmx_build_system コマンドを実行し、変更したシステムの情報モデルを実システムに展開します。

コマンドの実行例を次に示します。

```
cmx_build_system -m 192.168.100.12 -u admin -p admin -s MyWebSystem
```

3. cmx_start_target コマンドを実行し、追加したサービスユニットを起動します。

追加したサービスユニット「unit2」を稼働状態にします。稼働状態にするには、-mode オプションに「ALL」を指定します。

```
cmx_start_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -unit unit2 -mode ALL -strict
```

4. 追加したサービスユニットにある J2EE サーバを配置するホストでコマンドプロンプトを起動します。
5. リソースアダプタを設定して開始します。

コマンドの実行例を次に示します。

```
cjimportres MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -type rar -f <DB  
Connector_HiRDB_Type4_CP.rarのファイルパス>  
  
cjdeployrar MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -resname DB_Conn  
ector_for_HiRDB_Type4  
  
cjsetrarprop MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -resname DB_Conn  
ector_for_HiRDB_Type4 -c <MyDBConnectorAttr.xml※のファイルパス>  
  
cjtestres MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -resname DB_Conn  
ector_for_HiRDB_Type4  
  
cjstartrar MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -resname DB_Conne  
ctor_for_HiRDB_Type4
```

注※ スケールアウトの対象とするサービスユニットのリソースアダプタに反映したConnector属性ファイルを使用します。Connector属性ファイルを保管していない場合は、スケールアウトの対象とするサービスユニットにデプロイしたリソースアダプタからサーバ管理コマンド(cjgetrarprop)を使用して取得します。取得したConnector属性ファイルは必要に応じて編集します。

6. J2EE アプリケーションを設定して開始します。

コマンドの実行例を次に示します。

```
cjimportapp MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -f MyApp.ear  
  
cjstartapp MyWebSystem_J2EE02 -nameserver corbaname::192.168.100.14:900 -name MyApp
```

4.3 ホストを減らしてシステムの規模を縮小するシステムの運用（スケールイン）

スケールインとは、システムの規模を縮小する場合などに、ホストの台数を減らすことをいいます。システムをスケールインすると、業務全体を停止することなく、システムの規模を縮小して構成を変更できます。

4.3.1 ホストを減らしてシステムの規模を縮小（スケールイン）するには

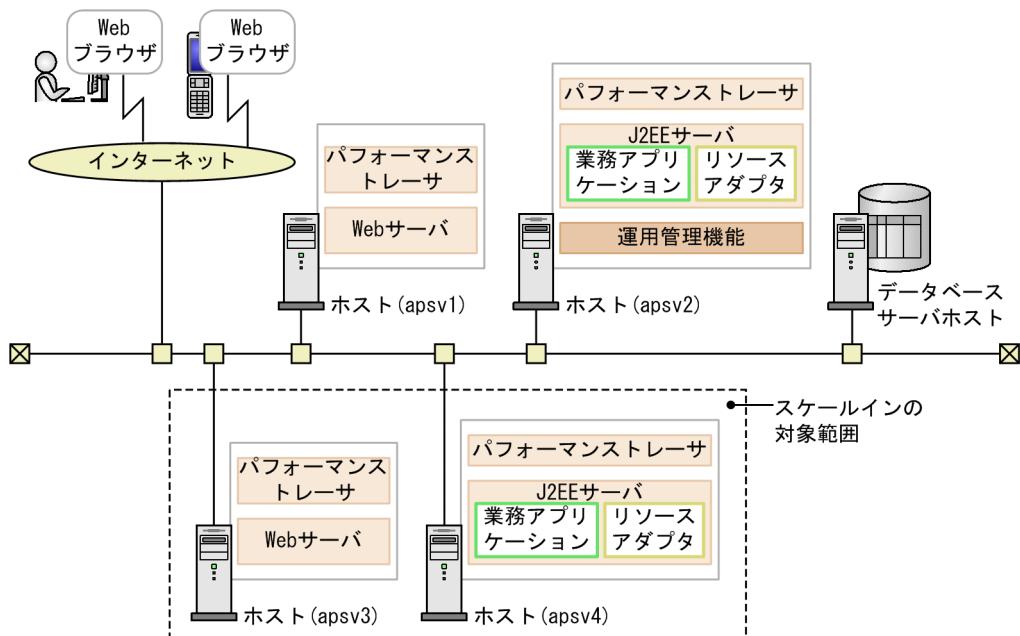
スケールインでは、ホストの台数を減らし、システムの規模を縮小することで、不要なホストを減らし、メンテナンスなどに掛かる作業コストの削減が図れます。

Smart Composer 機能の場合は、サービスユニット単位にホストおよび論理サーバを一括で減らせます。また、運用管理ポータルの場合は、[運用管理ポータル] 画面の各画面で個々にホストや論理サーバを減らせます。

- システム運用例

ホストを減らしてシステムの規模を縮小するシステム運用例を次の図に示します。ここでは、「4.2 ホストを増やしてシステムの規模を拡大するシステムの運用（スケールアウト）」で構築したシステムを使用します。

図 4-7 ホストを減らしてシステムの規模を縮小するシステム運用例



例題で使用する設定内容を次の表に示します。

表 4-22 例題で使用する設定内容（システムのスケールイン）

分類	項目	設定例
システム全体	Web システムの名称	MyWebSystem

分類	項目	設定例
	削除するサービスユニットの名称	unit2
削除する Web サーバ を配置するホスト	ホストの名称	apsv3
	IP アドレス (管理 IP アドレス) *	192.168.100.13
	論理 Web サーバの名称	MyWebSystem_HTTP02
	論理パフォーマンストレーサの名称	MyWebSystem_PRF03
削除する J2EE サーバ を配置するホスト	ホストの名称	apsv4
	IP アドレス (管理 IP アドレス) *	192.168.100.14
	論理 J2EE サーバの名称	MyWebSystem_J2EE02
	論理パフォーマンストレーサの名称	MyWebSystem_PRF04

注※ ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

・実施する作業と作業の要否

システムをスケールインする時に実施する作業と作業の要否を次の表に示します。システムのスケールインは、Smart Composer 機能を使用して実施します。

表 4-23 システムをスケールインする時に実施する作業と作業の要否 (Smart Composer 機能利用時)

システムをスケールインする時に実施する作業	作業の要否	作業の実施条件	参照個所
サービスユニットの停止	○	必ず実施	4.3.2(1)
サービスユニットの削除と論理サーバの環境のアンセット アップ	○	必ず実施	4.3.2(2)

(凡例)

○：必ず実施する作業です。

4.3.2 システム環境をスケールインする (CUI 利用時)

ここでは、Smart Composer 機能を使用して、サービスユニット (unit2) を削除してスケールインする手順について説明します。

システムをスケールインする前に、必要に応じて、Web システムが動作するために必要なリソースも変更します。Web システムが動作するために必要なリソースの見積もりについては、マニュアル「アプリケーションサーバ システム設計ガイド」の「5. 使用するリソースの見積もり (J2EE アプリケーション実行基盤)」を参照してください。

(1) サービスユニットの停止

- 運用管理機能 (Management Server) を配置するホストでコマンドプロンプトを起動します。
- cmx_stop_target コマンドを実行し、削除するサービスユニット「unit2」を停止します。
サービスユニットを停止状態にします。停止状態にするには、-mode オプションに「ALL」を指定します。
コマンドの実行例を次に示します。

```
cmx_stop_target -m 192.168.100.12 -u admin -p admin -s MyWebSystem -unit unit2 -mode ALL
```

(2) サービスユニットの削除と論理サーバの環境のアンセットアップ

- cmx_delete_system コマンドを実行し、サービスユニットを削除します。
削除するサービスユニットを停止して、Management Server に登録されているシステムの情報モデルから削除します。

```
cmx_delete_system -m 192.168.100.12 -u admin -p admin -s MyWebSystem -unit unit2
```
- Web サーバや J2EE サーバの環境をアンセットアップします。
cmx_delete_system コマンドでは、サーバの環境はアンセットアップされません。Web サーバや J2EE サーバの環境は、それぞれ次の方法でアンセットアップします。
 - Web サーバの環境のアンセットアップ
HTTP Server の hwsserveredit コマンドを使用して、Web サーバをアンセットアップします。
HTTP Server の hwsserveredit コマンドには、「HWS_」を付加した Web サーバの論理サーバ名※を指定します。
 - J2EE サーバの環境のアンセットアップ
cjsetup コマンドを使用して、J2EE サーバをアンセットアップします。cjsetup コマンドには、J2EE サーバの論理サーバ名※を指定します。

注※

システム構築時に、Web サーバや J2EE サーバの実サーバ名を指定している場合は、アンセットアップ時には実サーバ名を指定してください。

4.4 構成・定義情報を出力してほかのホストにシステム環境を移行するシステムの運用

システム環境の移行とは、テスト環境で構築したシステムを本番環境に移行する場合などに、構築済みのシステムの設定内容をほかの環境に移行することをいいます。

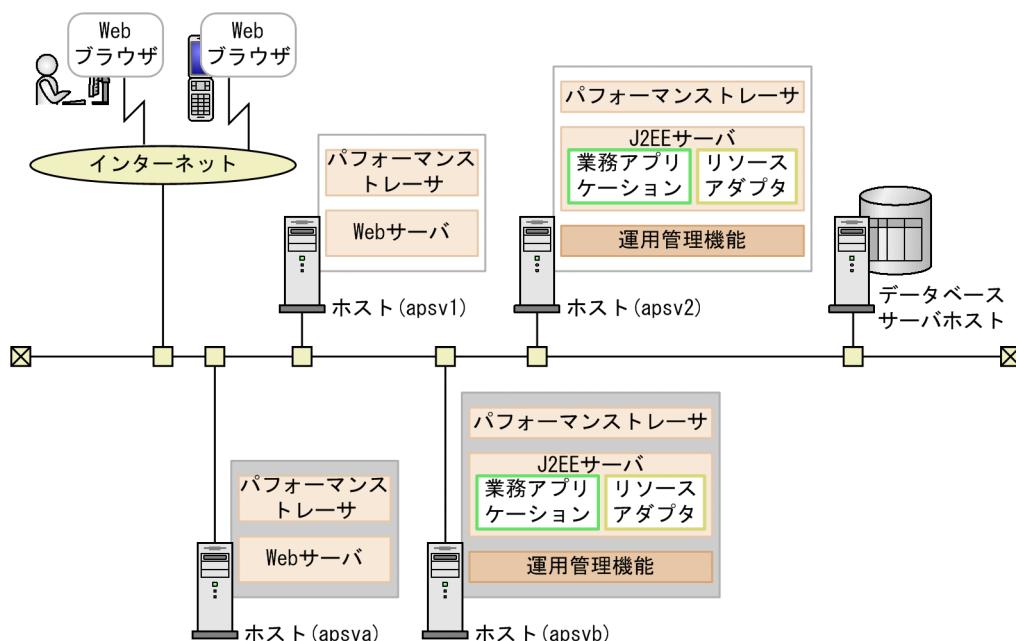
4.4.1 ほかのホストにシステム環境を移行するには

システム環境の移行を利用すると、ほかの環境でシステムを構築して稼働するまでに掛かる時間を短縮できます。

- ・システム運用例

構成・定義情報を出力してほかのホストにシステム環境を移行するシステム運用例を次の図に示します。ここでは、移行元の環境として、「4.1 Web サーバを別ホストに配置してマシン性能を向上するシステムの構築」で構築したシステムを使用します。

図 4-8 構成・定義情報を出力してほかのホストにシステム環境を移行するシステム運用例



(凡例)

■: 移行先の環境を示します。

例題で使用する設定内容を次の表に示します。

表 4-24 例題で使用する設定内容（システム環境の移行）

分類	項目	設定例
システム全体	移行情報を出力する簡易構築定義ファイルの名称	MyWebSystemexport.xml

分類	項目	設定例	
移行元の環境	Web サーバを配置するホスト	ホストの名称	apsv1
		IP アドレス (管理 IP アドレス) ^{※1}	192.168.100.11
移行先の環境	Web サーバを配置するホスト	ホストの名称	apsv2
		IP アドレス (管理 IP アドレス) ^{※1}	192.168.100.12
	J2EE サーバを配置するホスト ^{※2}	ホストの名称	apsva
		IP アドレス (管理 IP アドレス) ^{※1}	192.168.100.21
	J2EE サーバを配置するホスト ^{※2}	ホストの名称	apsvb
		IP アドレス (管理 IP アドレス) ^{※1}	192.168.100.22

注※1 ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

注※2 運用管理機能 (Management Server) は、J2EE サーバを配置するホストに配置します。

・実施する作業と作業の要否

システム環境を移行する時に実施する作業と作業の要否を次の表に示します。システム環境の移行は、Smart Composer 機能を使用して実施します。

表 4-25 システム環境を移行する時に実施する作業と作業の要否 (Smart Composer 機能利用時)

システム環境を移行する時に実施する作業	作業の要否	作業の実施条件	参照箇所
移行元のシステム環境の構築	○	必ず実施	4.1
移行先の環境でのホストの準備	○	必ず実施	4.4.2(1)
構成・定義情報の出力	○	必ず実施	4.4.2(2)
移行先のシステム環境の構築	○	必ず実施	4.4.2(3)

(凡例)

○：必ず実施する作業です。

4.4.2 構成・定義情報を出力してシステム環境を移行する

ここでは、Smart Composer 機能を使用して、構築済みのシステムをほかの環境に移行する手順について説明します。システム環境の移行時に使用する構成・定義情報の内容については、「[4.4.3 移行時に使用する構成・定義情報の内容](#)」を参照してください。また、システム環境の移行時の注意事項については、「[4.4.4 構成・定義情報を出力してシステムを移行するときの注意事項](#)」を参照してください。

(1) 移行先の環境でのホストの準備

1. 移行先の環境でホストを用意して、製品のインストールと初期設定、運用管理機能の構築をします。

移行先のシステムで Management Server が動作している必要があります。インストールと初期設定を済ませておいてください。

移行先のホストで実施する作業と作業の要否を次の表に示します。

表 4-26 移行先のホストで実施する作業と作業の要否

移行先のホストで実施する作業	作業の要否	作業の実施条件	参照箇所
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき	4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合	4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合	4.1.6
		データベース (Oracle) に接続する場合	4.1.7
		データベース (SQL Server) に接続する場合	4.1.8
		データベース (MySQL) に接続する場合	4.1.9
		データベース (PostgreSQL) に接続する場合	4.1.10
システムの環境変数の設定	○	必ず実施	4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合	4.1.14
運用管理機能の構築	○	必ず実施	4.1.16
運用管理エージェントの起動	○	必ず実施	4.1.19
Management Server の起動	○	必ず実施	4.1.20

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

(2) 構成・定義情報の出力

1. 移行元の環境の運用管理機能 (Management Server) を配置するホスト (apsv2 (192.168.100.12)) でコマンドプロンプトを起動します。
2. cmx_export_model コマンドを実行し、構築済みのシステムの設定内容を簡易構築定義ファイルに出力します。

コマンドの実行例を次に示します。ここでは、出力する簡易構築定義ファイルの名称を MyWebSystemexport.xml とします。

```
cmx_export_model -m 192.168.100.12 -u admin -p admin -o <MyWebSystemexport.xmlのファイルパス>
```

<MyWebSystemexport.xml のファイルパス>には、MyWebSystemexport.xml を格納するディレクトリのパスとファイル名 (MyWebSystemexport.xml) を指定します。

(3) 移行先のシステム環境の構築

1. 移行先の環境の運用管理機能 (Management Server) を配置するホスト (apsvb (192.168.100.22)) でコマンドプロンプトを起動します。
2. cmx_build_system コマンドを実行し、移行先の環境に構築済みのシステムと同じシステムを構築します。

コマンドの実行例を次に示します。cmx_build_system コマンドの-f オプションには、「(2) 構成・定義情報の出力」で出力した簡易構築定義ファイルを指定します。

```
cmx_build_system -m 192.168.100.22 -u admin -p admin -f <MyWebSystemexport.xmlのファイルパス>
```

4.4.3 移行時に使用する構成・定義情報の内容

ここでは、cmx_export_model コマンドを使用して構築済みのシステムの設定内容を出力した場合の出力内容について説明します。

(1) XML スキーマのバージョン

XML スキーマのバージョンは、最新のバージョンで出力されます。旧バージョンの簡易構築定義ファイルを使用して構築したシステムに対して設定内容を出力した場合、最新の XML スキーマのバージョンに従って、タグ名称が最新の名称に変換されます。

(2) 物理ティアの構成

Smart Composer 機能を使用して構築している場合は、簡易構築定義ファイルで定義した物理ティア (combined-tier, http-tier, j2ee-tier, または ctm-tier) で出力されます。Smart Composer 機能を使用

しないで構築したシステムの場合は、free-tier で出力されます。例えば、運用管理ポータルを使用して構築したシステムは free-tier で出力されます。free-tier では、システムの構成に応じて、次に示す論理サーバが出力されます。

- 論理 Web サーバ
- 論理 J2EE サーバ
- 論理パフォーマンストレーサ
- 論理 CTM ドメインマネジャ
- 論理 CTM
- 論理スマートエージェント
- 論理ユーザサーバ
- 論理ネーミングサービス

出力される論理サーバのパラメタについては、マニュアル「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)」の「4.8.1 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」を参照してください。

(3) 論理サーバのコンフィグレーションに設定しているパラメタ

出力規則

- 非推奨のパラメタは、推奨のパラメタに変換されます。
- 値を指定しているパラメタは出力されます。値を省略しているパラメタは出力されません。
- 簡易構築定義ファイルでパラメタにデフォルト値を設定している場合、または運用管理ポータルでキーにデフォルト値を設定している場合、そのパラメタは出力されます。

出力順序

論理サーバの種類ごとに、パラメタの種類によって出力順序があります。また、同じパラメタの種類の中でのパラメタの出力順序は、パラメタ名称の昇順です。パラメタの出力順序を、次の表に示します。

表 4-27 パラメタの出力順序

パラメタを設定するサーバ	出力順序	パラメタの種類
J2EE サーバ	1	各論理サーバ共通で指定できるパラメタ
	2	J2EE サーバ用オプション定義のパラメタ
	3	J2EE サーバ用ユーザプロパティのパラメタ
Web サーバ	1	各論理サーバ共通で指定できるパラメタ
	2	HTTP Server 定義のパラメタ
そのほかの論理サーバ	1	各論理サーバ共通で指定できるパラメタ
	2	該当する論理サーバのパラメタ

設定先

構築済みのシステムの論理サーバによってパラメタの設定先が異なります。パラメタの設定先を、次の表に示します。なお、free-tier では、物理ティアの定義にパラメタを設定できないため、サービスユニットの定義にすべてのパラメタが設定されます。

表 4-28 パラメタの設定先

同じ種類の論理サーバの数	パラメタに指定している値	パラメタの設定先
1	—	物理ティアの定義
2~n	すべての論理サーバで同じ	物理ティアの定義
	論理サーバごとに異なる	サービスユニットの定義

(凡例) —: なし

- システムの中に同じ種類の論理サーバが一つしかない場合、すべてのパラメタが物理ティアの定義に設定されます。
- システムの中に同じ種類の論理サーバが複数ある場合に、同じ種類の論理サーバで、同じパラメタに同じ値をそれぞれ指定していると、パラメタの値は、物理ティアの定義にそれぞれ設定されます。
- システムの中に同じ種類の論理サーバが複数ある場合に、同じ種類の論理サーバで、同じパラメタに異なる値をそれぞれ指定していると、パラメタはサービスユニットの定義にそれぞれ設定されます。

(4) 出力例

簡易構築定義ファイルの出力例を次に示します。

ここでは、簡易構築定義ファイルを使用して構築した combined-tier 構成のシステムに、運用管理ポータルでホスト (host1)、論理 J2EE サーバ (j2ee01)、および論理パフォーマンストレーサ (prf01) を追加したシステムの設定内容の出力例を示します。

簡易構築定義ファイルの出力例

```
<?xml version="1.0" encoding="UTF-8" ?>
<model-definition xmlns="http://www.cosminexus.com/mngsvr/schema/ModelDefinition-2.5">
  <web-system>
    <name>sample-system</name>
    <tier>
      <tier-type>combined-tier</tier-type>
      <configuration>
        <logical-server-type>j2ee-server</logical-server-type>
        <param>
          <param-name>ejbserver.http.port</param-name>
          <param-value>8085</param-value>
        </param>
      </configuration>
    </tier>
    <unit>
      <name>unit1</name>
      <allocated-host>
        <host-ref>apsv1</host-ref>
        <hosts-for>combined-tier</hosts-for>
      </allocated-host>
    </unit>
  </web-system>
</model-definition>
```

```

</web-system>
<web-system>※1
  <tier>
    <tier-type>free-tier※2</tier-type>
  </tier>
  <unit>※1
    <allocated-host>
      <host-ref>host1</host-ref>
      <hosts-for>free-tier</hosts-for>
      <define-server>
        <logical-server-name>prf01</logical-server-name>
        <logical-server-type>performance-tracer</logical-server-type>
      </define-server>
      <define-server>
        <logical-server-name>j2ee01</logical-server-name>
        <logical-server-type>j2ee-server</logical-server-type>
      <configuration>
        <param>※3
          <param-name>ejbserver.server.prf.PRFID</param-name>
          <param-value>PRF_ID</param-value>
        </param>
      </configuration>
    </define-server>
    </allocated-host>
  </unit>
</web-system>
<host>
  <host-name>apsv1</host-name>
</host>
<host>
  <host-name>host1</host-name>
</host>
</model-definition>

```

注※1

free-tier には、Web システム名、およびサービスユニット名がありません。

注※2

ほかのどの物理ティアの構成にも当てはまらない構成が含まれていると、その構成が free-tier として出力されます。

注※3

free-tier では、ティアの定義にパラメタを設定できないため、サービスユニットの定義にパラメタが設定されます。

4.4.4 構成・定義情報を出力してシステムを移行するときの注意事項

構成・定義情報を出力して、構築済みのシステムを移行するときの注意事項を次に示します。

- ・ 移行後のシステム環境に、free-tier が含まれている場合、Smart Composer 機能のコマンドで実施できる運用作業は次のとおりです。

- ・システムの起動・停止
- ・システム中の論理サーバのステータス監視
- ・システム中の論理サーバの cmx_build_system の実行状態の表示

free-tier を含むシステムについてほかの運用作業をする場合は、運用管理ポータル、またはサーバ管理コマンドを使用してください。

- ・Smart Composer 機能のコマンドで簡易構築定義ファイルに出力できるのは、システム環境の内容のうち、簡易構築定義ファイルで定義できる情報だけです。簡易構築定義ファイル以外のファイルを直接編集している場合、その編集内容は反映されません。
- ・Web システムに関連づいていない論理サーバを含むシステムを構築する場合は、次に示す手順で、移行先の環境に、構築済みのシステムと同じシステムを構築してください。

1. 移行元の環境で、cmx_export_model コマンドを実行し、構築済みのシステムの設定内容を簡易構築定義ファイルに出力します。

コマンドの実行例を次に示します。ここでは、出力する簡易構築定義ファイルの名称を MyWebSystemexport.xml とします。

```
cmx_export_model -m 192.168.100.12 -u admin -p admin -o <MyWebSystemexport.xmlのファイルパス>
```

2. 移行先の環境で、cmx_build_model コマンドを実行し、手順 1.で出力した簡易構築定義ファイルからシステムの情報モデルを構築します。

コマンドの実行例を次に示します。

```
cmx_build_model -m 192.168.100.22 -u admin -p admin -f <MyWebSystemexport.xmlのファイルパス>
```

3. 移行先の環境で、cmx_build_system コマンドを実行し、システムの情報モデルを基に、Web システムに関連づいていない論理サーバを含むシステムを構築します。

コマンドの実行例を次に示します。なお、構築するシステムに Web システムに関連づいている論理サーバが含まれる場合、-s オプションに Web システム名を指定して、再度 cmx_build_system コマンドを実行します。ここでは、Web システムを MyWebSystem とします。

- ・構築するシステムに Web システムに関連づいていない論理サーバだけ含まれる場合

```
cmx_build_system -m 192.168.100.22 -u admin -p admin -ns
```

- ・構築するシステムに Web システムに関連づいていない論理サーバおよび Web システムに関連づいている論理サーバが含まれる場合

```
cmx_build_system -m 192.168.100.22 -u admin -p admin -s MyWebSystem
```

```
cmx_build_system -m 192.168.100.22 -u admin -p admin -ns
```

4. 運用管理ポータルまたは Smart Composer 機能で実現する運用形態

4.5 論理ユーザサーバを配置するシステムの構築

論理ユーザサーバは、ユーザが定義する任意のプロセス（ユーザサーバ）を論理サーバとして定義するものです。特定のサービスやプロセスを論理ユーザサーバとして定義しておくことで、そのサービスやプロセスが Management Server の管理対象となり、ほかの論理サーバと同様に一括管理できるようになります。

ユーザサーバは、アプリケーションサーバ上で動作します。アプリケーションサーバ以外にも、アプリケーションサーバをインストールしているホストであれば、ユーザサーバを配置できます。その場合、これらのホストには、運用管理エージェントも配置する必要があります。

4.5.1 論理ユーザサーバを配置するには

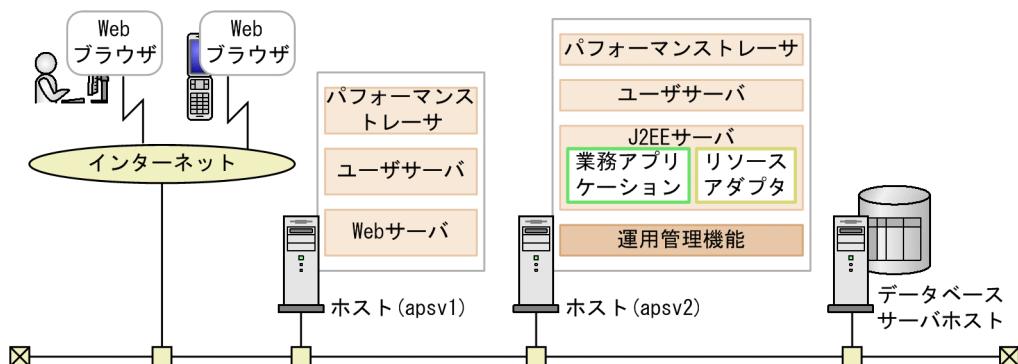
論理ユーザサーバを使用すると、ユーザが定義する任意のプロセスを Management Server で運用管理できるようになります。

Smart Composer 機能の場合は、論理ユーザサーバを簡易構築定義ファイルで定義します。また、運用管理ポータルでは論理ユーザサーバを定義できませんが、定義された論理ユーザサーバに対して起動、停止および運用監視の操作ができます。運用管理ポータルで論理ユーザサーバを使用する場合は、論理ユーザサーバをファイルで定義したあと、運用管理コマンドでそのファイルを基に論理ユーザサーバを設定します。

- システム構成例

論理ユーザサーバを配置するシステムの構成例を次の図に示します。ここでは、「4.1 Web サーバを別ホストに配置してマシン性能を向上するシステムの構築」で構築したシステムに論理ユーザサーバを配置します。

図 4-9 論理ユーザサーバを配置するシステムの構成例



例題で使用する設定内容を次の表に示します。

表 4-29 例題で使用する設定内容（論理ユーザサーバの配置）

分類	項目	設定例
システム全体	簡易構築定義ファイルの名称	MyWebSystemDef.xml
	Web システムの名称	MyWebSystem

分類	項目	設定例
	サービスユニットの名称	unit1
Web サーバを配置するホスト	ホストの名称	apsv1
	IP アドレス (管理 IP アドレス) ※	192.168.100.11
	論理ユーザサーバの名称	UserServer1
J2EE サーバを配置するホスト	ホストの名称	apsv2
	IP アドレス (管理 IP アドレス) ※	192.168.100.12
	論理ユーザサーバの名称	UserServer2

注※ ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

・実施する作業と作業の要否

論理ユーザサーバに配置するシステムの構築時に実施する作業と作業の要否を、利用する構築ツールごとに次の表に示します。

表 4-30 論理ユーザサーバを配置するシステムの構築時に実施する作業と作業の要否（運用管理ポータル利用時）

論理ユーザサーバを配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
データベースの準備	△	データベースに接続する場合	このマニュアル	2.1.1
データベースの環境の設定（Oracle の設定）	△	データベース（Oracle）に接続する場合		4.1.2
Component Container 管理者の設定（UNIX の場合）	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき		4.1.3
Application Server のインストール	○	必ず実施（OS が Windows の場合）		2.2.2
		必ず実施（OS が AIX の場合）		2.2.3
		必ず実施（OS が Linux の場合）		2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合		2.2.5
アプリケーションサーバでサポートしているシンボリックリンク（UNIX の場合）	△	OS が UNIX の場合		4.1.5
データベース接続環境の設定	△	データベース（HiRDB）に接続する場合		4.1.6
		データベース（Oracle）に接続する場合		4.1.7

論理ユーザサーバを配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
		データベース (SQL Server) に接続する場合		4.1.8
		データベース (MySQL) に接続する場合		4.1.9
		データベース (PostgreSQL) に接続する場合		4.1.10
システムの環境変数の設定	○	必ず実施		4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合		4.1.14
運用管理機能の構築	○	必ず実施	運用管理ポータル	2.2
運用管理エージェントの起動	○	必ず実施		4.1.2
Management Server の起動	○	必ず実施		4.1.2
システム環境の設定	○	必ず実施		3.2
システム環境のセットアップ	○	必ず実施		
論理ユーザサーバの設定	○	必ず実施		3.2.4
システムの起動 (論理サーバの起動)	○	必ず実施		4.1.2
リソースアダプタの設定	△	リソースアダプタ (DB Connector) によるリソース接続をする場合		3.3
		リソースアダプタ (DB Connector 以外) によるリソース接続をする場合		
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合		4.1.2
業務アプリケーションの設定と開始	○	必ず実施		3.4, 4.1.2

(凡例)

運用管理ポータル：マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」を示します。

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

表 4-31 論理ユーザサーバを配置するシステムの構築時に実施する作業と作業の要否 (Smart Composer 機能利用時)

論理ユーザサーバを配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
データベースの準備	△	データベースに接続する場合	2.1.1

論理ユーザサーバを配置するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
データベースの環境の設定 (Oracle の設定)	△	データベースが Oracle の場合	4.1.2
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき	4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合	4.1.5
データベース接続環境の設定	△	データベースが HiRDB の場合	4.1.6
		データベースが Oracle の場合	4.1.7
		データベース (SQL Server) に接続する場合	4.1.8
		データベース (MySQL) に接続する場合	4.1.9
		データベース (PostgreSQL) に接続する場合	4.1.10
システムの環境変数の設定	○	必ず実施	4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合	4.1.14
運用管理機能の構築	○	必ず実施	4.1.16
運用管理エージェントの起動	○	必ず実施	4.1.19
Management Server の起動	○	必ず実施	4.1.20
システム環境の設定 (簡易構築定義ファイルの設定)	○	必ず実施	4.1.22, 4.5.2
システム環境の構築 (セットアップ)	○	必ず実施	4.1.25
システムの起動	○	必ず実施	4.1.26
リソースアダプタの設定	△	DB Connector による接続の場合	4.1.28
		DB Connector 以外による接続の場合	4.1.29
リソースアダプタの開始	△	リソースと接続する場合	4.1.30
業務アプリケーションの設定と開始	○	必ず実施	4.1.31

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

4.5.2 論理ユーザサーバを簡易構築定義ファイルで設定する (CUI 利用時)

ここでは、Smart Composer 機能を使用して、次に示す論理ユーザサーバを追加する場合の簡易構築定義ファイルの作成例について説明します。

論理ユーザサーバは、特定のサービスユニットに関連づけると、サービスユニットの起動／停止と同時に、ユーザサーバも起動したり、停止したりできます。複数のユーザサーバを、一つのサービスユニットに関連づけることができますが、一つのユーザサーバを、複数のサービスユニットに関連づけることはできません。

MyWebSystemDef.xml の作成例

```
<?xml version="1.0" encoding="UTF-8"?>
<model-definition xmlns="http://www.cosminexus.com/mngsvr/schema/ModelDefinition-2.5">
  <web-system>
    <name>MyWebSystem</name>
    <display-name>MyWebSystem</display-name>
    <description>This is a sample.</description>

    <!--Specify the configurations for the Tier.-->
    <tier>
      <tier-type>http-tier</tier-type>
    </tier>
    <tier>
      <tier-type>j2ee-tier</tier-type>
    </tier>

    <!--Specify the hosts for each Service Unit.-->
    <unit>
      <name>unit1</name>
      <allocated-host>
        <host-ref>apsv1</host-ref>
        <hosts-for>http-tier</hosts-for>
        <define-server>
          <logical-server-name>UserServer1</logical-server-name>
          <logical-server-type>user-server</logical-server-type>
          <configuration>
            <param>
              <!-- サーバ起動用コマンド -->
              <param-name>start.arg</param-name>
              <param-value>c:/test/start.exe</param-value>
            </param>
          </configuration>
        </define-server>
      </allocated-host>
      <allocated-host>
        <host-ref>apsv2</host-ref>
        <hosts-for>j2ee-tier</hosts-for>
        <define-server>
          <logical-server-name>UserServer2</logical-server-name>
          <logical-server-type>user-server</logical-server-type>
        </define-server>
      </allocated-host>
    </unit>
  </web-system>
</model-definition>
```

```

<configuration>
<param>
<!-- サーバ起動用コマンド -->
<param-name>start.arg</param-name>
<param-value>c:/test/start.exe</param-value>
</param>
</configuration>
</define-server>
</allocated-host>
</unit>
</web-system>

<!--Specify the settings for the host.-->
<host>
<host-name>apsv1</host-name>
<agent-host>apsv1</agent-host>
</host>
<host>
<host-name>apsv2</host-name>
<agent-host>apsv2</agent-host>
</host>
</model-definition>

```

注 論理ユーザサーバのパラメタは、物理ティアの定義には設定できません。論理ユーザサーバのパラメタを設定する場合は、サービスユニットの定義に設定してください。この定義例では、サービスユニットの起動と同時にユーザサーバを起動するように、サーバ起動用コマンドを指定しています。

4.6 バッチアプリケーションを実行するシステムの構築

バッチアプリケーションを実行するシステムは、定型的または定期的な作業をまとめて処理するバッチ業務を実行できるシステムです。バッチアプリケーションは、バッチ業務を実行するためのアプリケーションで、バッチサーバ上で動作します。このシステムは、セットアップウィザードでは構築できません。Smart Composer 機能または運用管理ポータルで構築してください。

4.6.1 バッチアプリケーションを実行するには

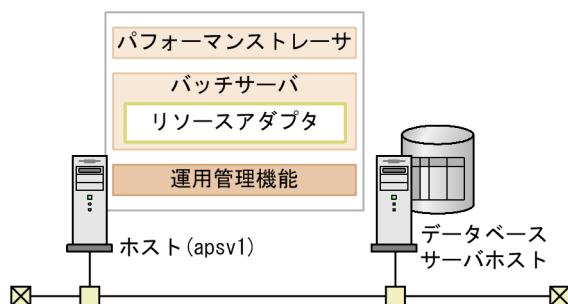
バッチアプリケーションを実行するシステムでは、バッチサーバを配置したアプリケーションサーバを中心構築します。また、業務の用途に合わせて、バッチジョブを自動実行するための関連プログラムを配置したり、バッチアプリケーションの実行をスケジュールするためのアプリケーションサーバの機能を設定したりします。

Smart Composer 機能の場合は、バッチサーバを配置するホスト (j2ee-tier) や論理サーバの環境設定を定義した簡易構築定義ファイルから、コマンドを使用して構築します。また、運用管理ポータルの場合は、[運用管理ポータル] 画面の各画面で、それぞれのホストに論理サーバ（バッチサーバなど）を配置して、各論理サーバの設定情報をホストに配布して構築します。

- システム構成例

バッチアプリケーションを実行するシステムの構成例を次の図に示します。バッチアプリケーションを実行するシステムの場合、運用管理機能 (Management Server) はバッチサーバを配置するホストに設定します。また、この構成例では、ネットワークは単一構成（業務 LAN と管理 LAN を分けない構成）とし、各ホストは物理 NIC を一つ使用するものとします。

図 4-10 バッチアプリケーションを実行するシステムの構成例



- 実施する作業と作業の要否

バッチアプリケーションを実行するシステムの構築時に実施する作業と作業の要否を、利用する構築ツールごとに次の表に示します。なお、参照先の記述にある「J2EE サーバ」を「バッチサーバ」に置き換えてお読みください。

表 4-32 バッチアプリケーションを実行するシステムの構築時に実施する作業と作業の要否
(運用管理ポータル利用時)

バッチアプリケーションを実行するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
データベースの準備	△	データベースに接続する場合	このマニュアル	2.1.1
データベースの環境の設定 (Oracle の設定)	△	データベース (Oracle) に接続する場合		4.1.2
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき		4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)		2.2.2
		必ず実施 (OS が AIX の場合)		2.2.3
		必ず実施 (OS が Linux の場合)		2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合		2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合		4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合		4.1.6
		データベース (Oracle) に接続する場合		4.1.7
		データベース (SQL Server) に接続する場合		4.1.8
		データベース (MySQL) に接続する場合		4.1.9
		データベース (PostgreSQL) に接続する場合		4.1.10
システムの環境変数の設定	○	必ず実施		4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合		4.1.14
運用管理機能の構築	○	必ず実施	運用管理ポータル	2.2
運用管理エージェントの起動	○	必ず実施		4.1.2
Management Server の起動	○	必ず実施		4.1.2

バッチアプリケーションを実行するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
システム環境の設定	○	必ず実施		5.2
システム環境のセットアップ	○	必ず実施		
システムの起動（論理サーバの起動）	○	必ず実施		6.1.2
リソースアダプタの設定	△	リソースアダプタ（DB Connector）によるリソース接続をする場合		5.3
		リソースアダプタ（DB Connector 以外）によるリソース接続をする場合		
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合		6.1.2
システム環境定義の変更	△	運用で必要な場合		6.2
リソースアダプタの設定変更	△	運用で必要な場合		6.2
システムの停止（論理サーバの停止）	△	運用で必要な場合		6.1.4
Management Server の停止	△	運用で必要な場合		4.1.4
運用管理エージェントの停止	△	運用で必要な場合		4.1.4
システム環境の削除	△	運用で必要な場合		5.4

（凡例）

運用管理ポータル：マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

**表 4-33 バッチアプリケーションを実行するシステムの構築時に実施する作業と作業の要否
(Smart Composer 機能利用時)**

バッチアプリケーションを実行するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
データベースの準備	△	データベースに接続する場合	2.1.1
データベースの環境の設定（Oracle の設定）	△	データベースが Oracle の場合	4.1.2
Component Container 管理者の設定（UNIX の場合）	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき	4.1.3
Application Server のインストール	○	必ず実施（OS が Windows の場合）	2.2.2
		必ず実施（OS が AIX の場合）	2.2.3
		必ず実施（OS が Linux の場合）	2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	2.2.5

バッチアプリケーションを実行するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
Application Server のインストール	○	必ず実施	2.2.1
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合	4.1.5
データベース接続環境の設定	△	データベースが HiRDB の場合	4.1.6
		データベースが Oracle の場合	4.1.7
		データベース (SQL Server) に接続する場合	4.1.8
		データベース (MySQL) に接続する場合	4.1.9
		データベース (PostgreSQL) に接続する場合	4.1.10
システムの環境変数の設定	○	必ず実施	4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合	4.1.14
運用管理機能の構築	○	必ず実施	4.1.16
運用管理エージェントの起動	○	必ず実施	4.1.19
Management Server の起動	○	必ず実施	4.1.20
システム環境の設定 (簡易構築定義ファイルの設定)	○	必ず実施	4.1.22, 4.6.2
システム環境の構築 (セットアップ)	○	必ず実施	4.1.25
システムの起動	○	必ず実施	4.1.26
リソースアダプタの設定	△	DB Connector による接続の場合	4.1.28
		DB Connector 以外による接続の場合	4.1.29
リソースアダプタの開始	△	リソースと接続する場合	4.1.30
システム環境定義の変更	△	運用で必要な場合	4.1.33
リソースアダプタの設定変更	△	運用で必要な場合	4.1.35
システムの停止	△	運用で必要な場合	4.1.36
Management Server の停止	△	運用で必要な場合	4.1.37
運用管理エージェントの停止	△	運用で必要な場合	4.1.38
システム環境の削除	△	運用で必要な場合	4.1.39

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注 バッチサーバを実行するシステムの運用時の注意事項については、「4.6.3 バッチアプリケーションを実行するシステムを運用するときの注意事項」を参照してください。

4.6.2 簡易構築定義ファイルを設定するときの注意事項（バッチアプリケーション使用時）

ここでは、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に、簡易構築定義ファイルを設定するときの注意事項について説明します。

・バッチサーバとして動作するための設定

サーバプロセスをバッチサーバとして構築し、動作させるためには設定が必要です。バッチサーバは論理 J2EE サーバとして定義します。バッチサーバの設定は、論理 J2EE サーバ (j2ee-server) の <configuration> タグ内に指定します。

バッチサーバに関する定義を次の表に示します。

表 4-34 バッチサーバに関する定義

項目	パラメタ	設定内容	必須または任意
バッチサーバとしてサーバを構築するための設定	batch.service.enabled	バッチサーバとして構築するかどうかを指定します。パラメタの値は必ず true を指定してください。	必須
SecurityManagerを使用しない設定	use.security	バッチサーバの起動時に SecurityManager を使用するかどうかを指定します。パラメタの値は必ず false を指定してください。	必須
ライトトランザクション機能を有効にするための設定	ejbserver.distributedtx.XA Transaction.enabled	グローバルトランザクションを使用するかどうかを指定します。バッチサーバの場合、グローバルトランザクションは使用できません。ローカルトランザクションを使用します。このため、ローカルトランザクションに最適化された環境を提供する、ライトトランザクション機能を使用します。パラメタの値は必ず false を指定してください。なお、このパラメタはデフォルトの設定が false のため、変更しないでください。	任意
明示管理ヒープ機能を無効にする設定	add.jvm.arg	このパラメタは JDK11 を使用する場合だけ指定できます。明示管理ヒープ機能を使用するかどうかを指定します。バッチサーバの場合、明示管理ヒープ機能を使用しないときは、明示管理ヒープ機能を無効にすることをお勧めします。明示管理ヒープ機能を無効にするには、パラメタの値に -XX:-HitachiUseExplicitMemory を指定してください。デフォルトの設定の場合、明示管理ヒープ機能は有効 (-XX:+HitachiUseExplicitMemory) です。	任意

項目	パラメタ	設定内容	必須または任意
実サーバ名の設定	realservername	バッチサーバの実サーバ名を指定します。実サーバ名は、同一ホスト内でユニークになるように指定してください。省略した場合は、論理サーバ名が設定されます。	任意
JavaVM 終了メソッド呼び出し時の JavaVM の動作設定	ejbserver.batch.application.exit.enabled	<p>次の JavaVM 終了メソッドをバッチアプリケーションで呼び出した時に、JavaVM を終了するかどうかを指定します。</p> <ul style="list-style-type: none"> • <code>java.lang.System.exit(int)</code> • <code>java.lang.Runtime.exit(int)</code> • <code>java.lang.Runtime.halt(int)</code> <p>デフォルトは「true」（JavaVM を終了しないでバッチアプリケーションのスレッドを終了する）です。「false」を指定すると、JavaVM 終了メソッドの呼び出し時に、バッチサーバごと JavaVM が終了されます。詳細については、マニュアル「アプリケーションサーバ 機能解説 拡張編」の「2.3.7 バッチアプリケーションの実装（バッチアプリケーションの作成規則）」を参照してください。</p>	任意

（凡例）必須：必ず指定してください。 任意：必要に応じて設定してください。

・バッチアプリケーションのスケジューリング機能を使用するための設定

バッチアプリケーションのスケジューリング機能を使用する場合、物理ティアの種類に ctm-tier を定義します。この場合、バッチサーバとして動作するための設定に加えて、バッチアプリケーションのスケジューリング機能の設定が必要です。これらの設定は、各論理サーバの<configuration>タグ内に指定します。

バッチサーバとバッチアプリケーションのスケジューリング機能に関する定義を次の表に示します。

表 4-35 バッチサーバとバッチアプリケーションのスケジューリング機能に関する定義

論理サーバの種類	項目	パラメタ	設定内容	必須または任意
論理 CTM (component-transaction-monitor)	ORB ゲートウェイ機能を使用する設定	ctm.Agent	CTM レギュレータの ORB ゲートウェイ機能を使用するかどうかを指定します。スケジューリング機能を使用する場合、CTM レギュレータの ORB ゲートウェイ機能を使用します。パラメタの値は必ず 1 を指定してください。	必須
	CTM を識別するための設定	ctmid	CTM 識別子を指定します。運用管理ドメイン内でユニークな値を設定する必要があります。CTM 識別子の設定は省略できます。	任意
論理 J2EE サーバ (j2ee-server)	バッチサーバとしてサーバを構築するための設定	batch.service.enabled	バッチサーバとして構築するかどうかを指定します。パラメタの値は必ず true を指定してください。	必須

論理サーバの種類	項目	パラメタ	設定内容	必須または任意
	SecurityManager を使用しない設定	use.security	バッチサーバの起動時に SecurityManager を使用するかどうかを指定します。パラメタの値は必ず false を指定してください。	必須
	ライトトランザクション機能を有効にするための設定	ejbserver.distributedtx.XATransaction.enabled	グローバルトランザクションを使用するかどうかを指定します。バッチサーバの場合、グローバルトランザクションは使用できません。ローカルトランザクションを使用します。このため、ローカルトランザクションに最適化された環境を提供する、ライトトランザクション機能を使用します。パラメタの値は必ず false を指定してください。なお、このプロパティはデフォルトの設定が false のため、変更しないでください。	必須
	スマートエージェントを使用する設定	vbroker.agent.enabledLocator	スマートエージェントを使用するかどうかを指定します。 スケジューリング機能を使用する場合、スマートエージェントを使用します。パラメタの値は必ず true を指定してください。	必須
	明示管理ヒープ機能を無効にする設定	add.jvm.arg	このパラメタは JDK11 を使用する場合だけ指定できます。 明示管理ヒープ機能を使用するかどうかを指定します。 バッチサーバの場合、明示管理ヒープ機能を使用しないときは、明示管理ヒープ機能を無効にすることをお勧めします。明示管理ヒープ機能を無効にすることには、パラメタの値に-XX:-HitachiUseExplicitMemory を指定してください。デフォルトの設定の場合、明示管理ヒープ機能は有効 (-XX:+HitachiUseExplicitMemory) です。	任意
	実サーバ名の設定	realservername	バッチサーバの実サーバ名を指定します。実サーバ名は、同一ホスト内でユニークになるように指定してください。省略した場合は、論理サーバ名が設定されます。	任意
	JavaVM 終了メソッド呼び出し時の JavaVM の動作設定	ejbserver.batch.application.exit.enabled	次の JavaVM 終了メソッドをバッチアプリケーションで呼び出した時に、JavaVM を終了するかどうかを指定します。 <ul style="list-style-type: none"> java.lang.System.exit(int) java.lang.Runtime.exit(int) 	任意

論理サーバの種類	項目	パラメタ	設定内容	必須または任意
			<ul style="list-style-type: none"> java.lang.Runtime.halt(int) <p>デフォルトは「true」(JavaVM を終了しないでバッチアプリケーションのスレッドを終了する) です。「false」を指定すると、JavaVM 終了メソッドの呼び出し時に、バッチサーバごと JavaVM が終了されます。詳細については、マニュアル「アプリケーションサーバ 機能解説 拡張編」の「2.3.7 バッチアプリケーションの実装 (バッチアプリケーションの作成規則)」を参照してください。</p>	
	スケジュールグループ名の設定	ejbserver.batch.schedule.group.name	<p>CTM によって管理されるバッチサーバ群のスケジュールグループ名を指定します。デフォルトでは JOBGROUP が設定されます。</p> <p>CTM は、スケジュールグループごとにバッチアプリケーションの実行をスケジューリングします。</p> <p>複数のスケジュールグループを使用してスケジュールキューを分ける場合には、バッチサーバごとにスケジュールグループ名を設定してください。</p>	任意
	スケジュールキューの長さの設定	ejbserver.batch.queue.length	CTM で作成されるスケジュールキューの長さを指定します。デフォルトでは 50 が設定されます。	任意

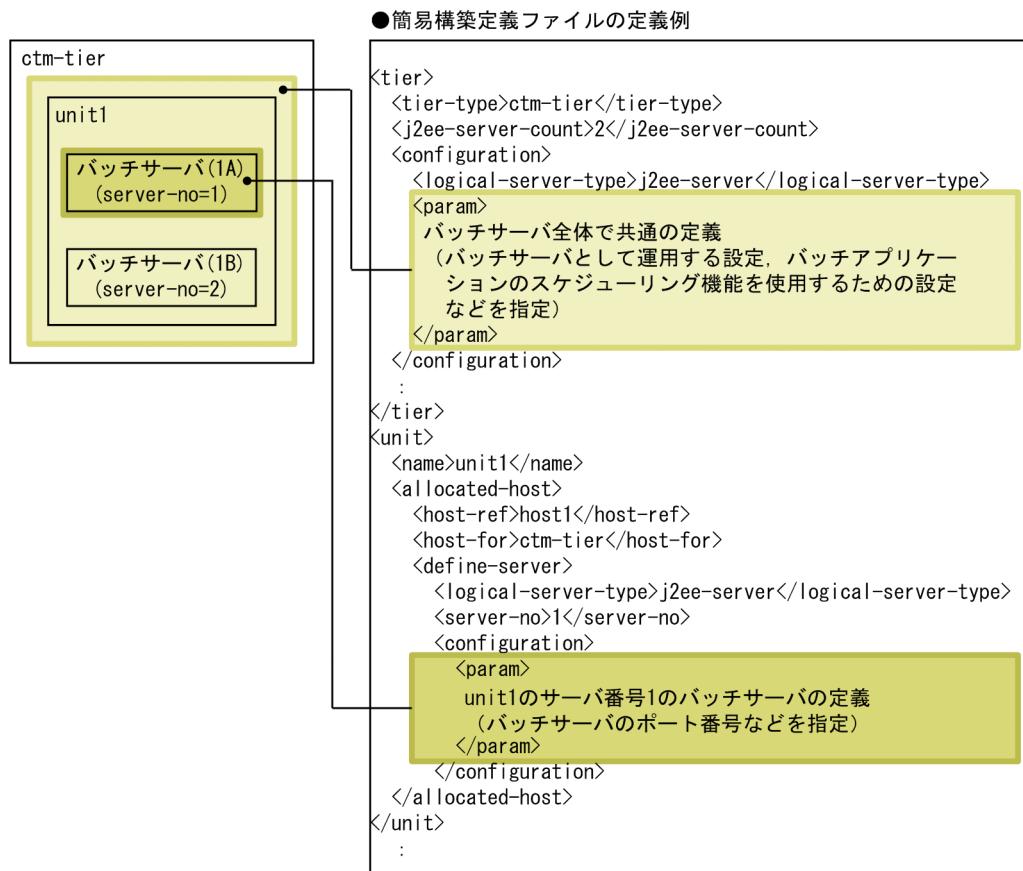
(凡例) 必須: 必ず指定してください。 任意: 必要に応じて設定してください。

• バッチサーバを複数配置するときの注意事項

バッチアプリケーションのスケジューリング機能を使用する場合に、複数のバッチサーバを配置するときは、物理ティアの定義で、1 ホストに配置するバッチサーバの数を設定します。

複数のバッチサーバを配置する場合も、物理ティアおよびサービスユニットの定義で、コンフィグレーションを定義できます。バッチサーバのコンフィグレーション定義の適用範囲を次の図に示します。

図 4-11 バッヂサーバのコンフィグレーション定義の適用範囲



この図の場合に、各サービスユニットのバッヂサーバごとに適用されるコンフィグレーション定義を次の表に示します。

表 4-36 各サービスユニットのバッヂサーバごとに適用されるコンフィグレーション定義

サービスユニット	バッヂサーバ	適用されるコンフィグレーション定義
unit1	バッヂサーバ (1A)	<ul style="list-style-type: none"> サービスユニットの定義 (特定のバッヂサーバの定義) 物理ティアの定義 (バッヂサーバ全体で共通の定義)
	バッヂサーバ (1B)	物理ティアの定義 (バッヂサーバ全体で共通の定義)

この表のように、バッヂサーバごとに適用されるコンフィグレーション定義が複数ある場合、各定義で同じパラメタを指定したときは、次の 1., 2.の順に優先して適用されます。

1. サービスユニットの定義 (特定のバッヂサーバの定義)
 2. 物理ティアの定義 (バッヂサーバ全体で共通の定義)
- バッヂサーバで使用しない定義
- バッヂサーバでは負荷分散機 (<load-balancer>タグ) を使用しません。

4.6.3 バッチアプリケーションを実行するシステムを運用するときの注意事項

バッチアプリケーションを実行するシステムでは、J2EE アプリケーションを実行するシステムと同様の運用ができますが、次の点に注意してください。参照先の記述にある「J2EE サーバ」を「バッチサーバ」に置き換えてお読みください。

- 1 ホストに配置するバッチサーバの数は、Web システムの構成変更時に変更できません。
- バッチアプリケーションのスケジューリング機能を使用するシステム (ctm-tier) では、スケールアウトおよびスケールインを実施できません。
- バッチアプリケーションを実行するシステムの設定を変更する場合、次のパラメタの値は変更しないでください。
 - batch.service.enabled (バッチサーバの拡張パラメタ)
 - use.security (SecurityManager の使用を設定するパラメタ)
- グローバルトランザクションは使用できないため、DB Connector の RAR ファイルには、グローバルトランザクション用の RAR ファイルは使用できません。
- リソースに接続するバッチアプリケーションの場合、ユーザ指定名前空間機能を使用してリソースアダプタに別名を設定します。バッチアプリケーションからリソースアダプタをロックアップするには、ユーザ指定名前空間機能で設定した別名を使用してください。ユーザ指定名前空間機能での任意のルックアップ名称は、Connector 属性ファイルで設定します。

4.7 運用管理機能から負荷分散機を制御するシステムの構築

負荷分散機は、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。負荷分散機を使用すると、トラフィックを効率的に分散し、処理性能を向上できます。アプリケーションサーバでは、運用管理機能から負荷分散機を制御できます。運用管理機能から負荷分散機を制御するシステムは、Smart Composer 機能の場合にだけ構築できます。運用管理ポータルの場合、このシステムは構築できません。

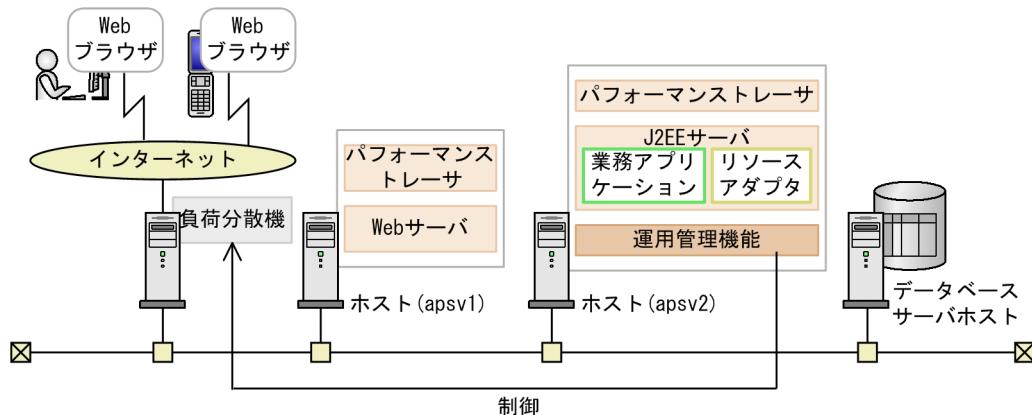
4.7.1 運用管理機能から負荷分散機を制御するには

Smart Composer 機能を使用したシステムで運用管理機能から負荷分散機を制御する場合、負荷分散機を準備する以外に、運用管理機能を配置するホストで負荷分散機へ接続する環境を設定する必要があります。

- システム構成例

運用管理機能から負荷分散機を制御するシステムの構成例を次の図に示します。ここでは、「4.1 Web サーバを別ホストに配置してマシン性能を向上するシステムの構築」で構築したシステムに負荷分散機を配置して運用管理機能から制御するように設定します。

図 4-12 運用管理機能から負荷分散機を制御するシステムの構成例



例題で使用する設定内容を次の表に示します。

表 4-37 例題で使用する設定内容（運用管理機能からの負荷分散機の制御）

分類	項目	設定例
システム全体	簡易構築定義ファイルの名称	MyWebSystemDef.xml
	Web システムの名称	MyWebSystem
	サービスユニットの名称	unit1
負荷分散機	負荷分散機の名称	MyAX2000
	負荷分散機の種類	ACOS
	負荷分散機の管理 IP アドレス	192.168.100.10

分類	項目	設定例
	仮想サーバの名称	www.sample.com
	仮想サーバの IP アドレス	192.168.1.10
	仮想サーバのポート番号	80
Web サーバを配置するホスト	ホストの名称	apsv1
	IP アドレス (管理 IP アドレス) ※	192.168.100.11
J2EE サーバを配置するホスト	ホストの名称	apsv2
	IP アドレス (管理 IP アドレス) ※	192.168.100.12

注※ ネットワークが単一構成の場合、IP アドレスは、業務 IP アドレスと管理 IP アドレスの両方を兼ねます。

• 実施する作業と作業の要否

運用管理機能から負荷分散機を制御するシステムの構築時に実施する作業と作業の要否を次の表に示します。運用管理機能からの負荷分散機の制御は、Smart Composer 機能を使用して実施します。

表 4-38 運用管理機能から負荷分散機を制御するシステムの構築時に実施する作業と作業の要否 (Smart Composer 機能利用時)

運用管理機能から負荷分散機を制御するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照箇所
データベースの準備	△	データベースに接続する場合	2.1.1
データベースの環境の設定 (Oracle の設定)	△	データベース (Oracle) に接続する場合	4.1.2
負荷分散機の準備	○	必ず実施する	4.7.3
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき	4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)	2.2.2
		必ず実施 (OS が AIX の場合)	2.2.3
		必ず実施 (OS が Linux の場合)	2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合	2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合	4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合	4.1.6
		データベース (Oracle) に接続する場合	4.1.7
		データベース (SQL Server) に接続する場合	4.1.8

運用管理機能から負荷分散機を制御するシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照個所
		データベース (MySQL) に接続する場合	4.1.9
		データベース (PostgreSQL) に接続する場合	4.1.10
システムの環境変数の設定	○	必ず実施	4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合	4.1.14
運用管理機能の構築	○	必ず実施	4.1.16
運用管理エージェントの起動	○	必ず実施	4.1.19
Management Server の起動	○	必ず実施	4.1.20
負荷分散機への接続の設定	○	必ず実施	4.7.5(1)
システム環境の設定 (簡易構築定義ファイルの設定)	○	必ず実施	4.1.22 , 4.7.5(2)
システム環境の構築 (セットアップ)	○	必ず実施	4.1.25
システムの起動	○	必ず実施	4.1.26
リソースアダプタの設定	△	リソースアダプタ (DB Connector) によるリソース接続をする場合	4.1.28
		リソースアダプタ (DB Connector 以外) によるリソース接続をする場合	4.1.29
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合	4.1.30
業務アプリケーションの設定と開始	○	必ず実施	4.1.31

(凡例)

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注

運用管理機能からの負荷分散機制御時の注意事項については、「[4.7.6 負荷分散機を制御するときの注意事項](#)」を参照してください。

4.7.2 負荷分散機とは

負荷分散機を使用すると、一つの仮想 IP アドレスで複数のサーバを管理するロードバランシング機能によって、トラフィックを効率的に分散し、処理性能を向上できます。負荷分散機として BIG-IP (BIG-IP

v9, BIG-IP v10.1, BIG-IP v10.2, BIG-IP v11) または ACOS (AX2000, AX2500, BS320) が使用できます。

Smart Composer 機能で構築したシステムは、負荷分散機の Cookie スイッチング機能を使用して運用することもできます。Cookie スイッチング機能とは、Cookie を利用してセッションを維持する機能です。プロキシ経由などで要求元 IP アドレスからクライアントが特定できない場合でも、同じクライアントからの一連の HTTP リクエストを同じサービスユニットに振り分けることができます。

参考

スイッチングモードについて

- 08-53 以降では、BIG-IP の Cookie スイッチング機能のうち、Smart Composer 機能で対応しているスイッチングモードはインサートモードだけです。
- ACOS の場合、サーバからのレスポンスに対して ACOS が cookie を設定します。設定する cookie の形式は次のとおりです。

Set-Cookie:<cookie 名>-<仮想ポート>=<実サーバ IP アドレス>_<実ポート>

また、Smart Composer 機能で構築したシステムでは、異なる Web システム間や、異なる運用管理ドメインの同じ Web システム間で負荷分散機を共有したり、一つの Web システム内で負荷分散機を冗長化構成で配置したりできます。

注意事項

Web サーバ連携を使用しないパターン (j2ee-tier), および free-tier の場合、負荷分散機を配置できません。

(1) 負荷分散機の制御方法

アプリケーションサーバから負荷分散機を制御するには、CLI コマンドまたは API を使用します。この場合、lb.properties (負荷分散機定義プロパティファイル) で、負荷分散機への接続情報を設定します。設定方法については、「[4.7.5 負荷分散機へ接続する環境を設定する](#)」を参照してください。

(2) Web システム間での負荷分散機の共有

Smart Composer 機能で構築する Web システムでは、異なる Web システム間で、同じ (1 台の) 負荷分散機を共有できます。なお、1 ホスト内の異なる Web システムで、1 台の負荷分散機を共有する場合は、それぞれの Web システムで使用する負荷分散機の仮想サーバを分ける必要があります。

負荷分散機を共有する場合の注意事項については、「[4.7.6 負荷分散機を制御するときの注意事項](#)」を参照してください。

(3) 負荷分散機の冗長化

Smart Composer 機能で構築する Web システムでは、一つの Web システム内に、負荷分散機を冗長化構成（アクティブ・スタンバイ構成）で配置できます。配置できる負荷分散機の台数は、2 台です。負荷分散機を冗長化構成で配置すると、実行系（アクティブ）の負荷分散機で障害が発生した場合に、リクエストの振り分け処理が自動的に待機系（スタンバイ）の負荷分散機に切り替わって、業務を続行できます。負荷分散機を冗長化する場合の注意事項については、「[4.7.6 負荷分散機を制御するときの注意事項](#)」を参照してください。

参考

既存の Web システムの負荷分散機を冗長化構成に変更したい場合は、既存の Web システムを削除してから、負荷分散機を冗長化した Web システムを再構築してください。

4.7.3 負荷分散機を準備する

Smart Composer 機能では、負荷分散機として BIG-IP または ACOS が使用できます。BIG-IP の場合は BIG-IP v9, BIG-IP v10.1, BIG-IP v10.2、または BIG-IP v11、ACOS の場合は AX2000, AX2500、または BS320 が使用できます。

アプリケーションサーバは、使用する負荷分散機によって接続方法が異なります。

- BIG-IP (BIG-IP v9, BIG-IP v10.1, BIG-IP v10.2、または BIG-IP v11) の場合
接続方法は、API (SOAP アーキテクチャ) を使用した直接接続です。API を使用する場合の注意事項については、「[4.7.4 接続方法に API を使用するときの注意事項](#)」を参照してください。

参考

BIG-IP v9, BIG-IP v10.1、および BIG-IP v10.2 の場合、ssh プロトコルを使用した直接接続も使用できます。接続方式に ssh プロトコルを使用した直接接続を選択した場合は、Management Server を配置するホストで事前設定が必要です。設定方法については、「[付録 J.2\(2\) BIG-IP v9, BIG-IP v10.1、または BIG-IP v10.2 の場合](#)」を参照してください。また、Red Hat Enterprise Linux 9.1 (AMD/Intel 64) 以降を使用する場合は ssh プロトコルを使用した直接接続を使用できません。API を使用してください。

- ACOS (AX2000 または BS320) の場合
接続方法は、telnet プロトコルを使用した直接接続です。
- ACOS (AX2500) の場合
接続方法は、API (REST アーキテクチャ) を使用した直接接続です。API を使用する場合の注意事項については、「[4.7.4 接続方法に API を使用するときの注意事項](#)」を参照してください。

4.7.4 接続方法に API を使用するときの注意事項

負荷分散機の接続方法に API を使用する場合は、Management Server を配置するホストで事前に負荷分散機の接続環境の設定が必要です。設定方法については、マニュアル「アプリケーションサーバ 機能解説 セキュリティ管理機能編」の「8.4 負荷分散機の接続環境の設定」を参照してください。

4.7.5 負荷分散機へ接続する環境を設定する

Smart Composer 機能から負荷分散機を制御する場合は、負荷分散機への接続の設定をする必要があります。また、簡易構築定義ファイルで負荷分散機の定義（<load-balancer>タグの定義）を設定する必要があります。

(1) 負荷分散機への接続の設定

使用する負荷分散機の種類、接続方式など、負荷分散機へのアクセスに必要な接続情報を Smart Composer 機能で定義します。運用管理機能を配置するホストで、lb.properties（負荷分散機定義プロパティファイル）に負荷分散機への接続情報を設定します。設定する情報には、負荷分散機独自のものと、共通のものがあります。

telnet を使用した直接接続を使用する負荷分散機（AX2000、BS320）の lb.properties の設定例を次に示します。API を使用した直接接続を使用する負荷分散機の設定例については、マニュアル「アプリケーションサーバ 機能解説 セキュリティ管理機能編」の「8.5 Management Server (Smart Composer 機能) での負荷分散機の接続情報の設定」を参照してください。

```
lb.list=192.168.100.10
lb.enable_passwd.192.168.100.10=adminpw

lb.connect_type.192.168.100.10=telnet
#lb.telnet.port.192.168.100.10=23
lb.telnet.user.192.168.100.10=user01
lb.telnet.passwd.192.168.100.10=user01pw
lb.telnet.user_prompt.192.168.100.10="AX login: "
lb.telnet.passwd_prompt.192.168.100.10="Password: "
lb.telnet.login_prompt.192.168.100.10=>
#lb.telnet.login_timeout.192.168.100.10=10
#lb.telnet.command_timeout.1192.168.100.10=10
```

lb.properties には、負荷分散機の接続ユーザ名、および接続ユーザのパスワードを記述するため、これらの情報が漏洩しないように適切にファイルのアクセス権を設定してください。

(2) 簡易構築定義ファイルの設定

簡易構築定義ファイルの負荷分散機の定義（<load-balancer>タグの定義）で、次のタグを定義します。

- cookie スイッチングの定義（<load-balancer>タグの<cookie-switching>タグの定義）※

- 仮想サーバの定義（<load-balancer>タグの<virtual-server>タグの定義）

注※

負荷分散機のCookieスイッチング機能を使用する場合に必要です。この設定によって、一連のHTTPリクエストが一つのWebサーバまたはJ2EEサーバで処理されます。また、負荷分散機のCookieスイッチング機能を使用する場合は、物理ティアの定義（<tier>タグ）で、J2EEサーバのWebコンテナの設定に「webserver.container.server_id.enabled=true」を指定してください。

簡易構築定義ファイルの作成例を次に示します。ここでは、負荷分散機を定義する<load-balancer>タグ内の定義について説明します。

簡易構築定義ファイルの<load-balancer>タグの定義例

```
<!--Specify the settings of load balancer.-->
<load-balancer>
  <name>MyAX2000</name>
  <load-balancer-type>ACOS</load-balancer-type>
  <management-host>192.168.100.10</management-host>
  <virtual-server>
    <virtual-server-name>www.sample.com</virtual-server-name>
    <ip-address>192.168.1.10</ip-address>
    <http-port>80</http-port>
  </virtual-server>
</load-balancer>
```

4.7.6 負荷分散機を制御するときの注意事項

- システムを一括構築するときの注意事項（負荷分散機制御をする場合）

Smart Composer機能では、負荷分散機の基本的な情報（仮想サーバや実サーバの定義、仮想サーバポートと実サーバポートとのバインドなど）だけ設定します。この設定を基にシステム構築時に負荷分散機に仮想サーバや実サーバを作成します。特定のネットワーク構成をサポートするためなどの設定はシステムの一括構築後、手動で設定してください。

Smart Composer機能が実行する負荷分散機のCLIコマンドおよびAPIについては、「[付録J Smart Composer機能が実行する負荷分散機のCLIコマンドまたはAPI](#)」を参照してください。

- 負荷分散機を共有するときの注意事項

異なるWebシステム間で1台の負荷分散機を共有する場合は、それぞれのWebシステムで使用する負荷分散機の仮想サーバを分ける必要があります。

簡易構築定義ファイルでは、次の定義で注意が必要です。

- 負荷分散機の定義

各Webシステムの負荷分散機のIPアドレスに、同じ値を設定してください。

- 仮想サーバの定義（負荷分散機の定義）

各Webシステムの仮想サーバ名、仮想サーバのIPアドレスおよびポート番号が重複しないように設定してください。

また、ACOS の場合は、次の 1. または 2. の条件を満たす構成のとき、自動的にポート変換する設定（オートトランスレーション機能の使用）が必要です。

1. 仮想サーバまたは実サーバに設定したポート番号が 1023 より大きいとき（well-known でないとき）
2. 実サーバのポート番号とそれに対応する仮想サーバのポート番号が異なるとき
(例) 仮想サーバのポート番号は 80 だが、振り分け先のポート番号は 8081 にしたいとき

オートトランスレーション機能の設定は、あらかじめ負荷分散機に対して実施しておいてください。設定の詳細については、ACOS のドキュメントを参照してください。

• 負荷分散機を冗長化するときの注意事項

負荷分散機を冗長化する場合は、それぞれの負荷分散機への接続の設定をする必要があります。

lb.properties には、2 台分の負荷分散機へのアクセスに必要な接続情報を設定します。この場合、それぞれの負荷分散機の接続方式 (lb.connect_type) には、同じ値を設定してください。

telnet を使用した直接接続を使用する負荷分散機 (AX2000, BS320) を冗長化する場合の lb.properties の設定例を次に示します。ここでは、負荷分散機 LB1 と LB2 を配置した場合で、LB1 の管理 IP アドレスを「192.168.1.100」、LB2 の管理 IP アドレスを「192.168.1.101」として説明します。

```
lb.list=192.168.1.100,192.168.1.101

#冗長構成の負荷分散機LB1の設定
lb.enable_passwd.192.168.1.100=adminpw
lb.connect_type.192.168.1.100=telnet
lb.telnet.user.192.168.1.100=user01
lb.telnet.passwd.192.168.1.100=user01pw
lb.telnet.user_prompt.192.168.1.100="AX login: "
lb.telnet.passwd_prompt.192.168.1.100="Password: "
lb.telnet.login_prompt.192.168.1.100=>

#冗長構成の負荷分散機LB2の設定
lb.enable_passwd.192.168.1.101=adminpw
lb.connect_type.192.168.1.101=telnet
lb.telnet.user.192.168.1.101=user01
lb.telnet.passwd.192.168.1.101=user01pw
lb.telnet.user_prompt.192.168.1.101="AX login: "
lb.telnet.passwd_prompt.192.168.1.101="Password: "
lb.telnet.login_prompt.192.168.1.101=>
```

注 背景色付きの太字の部分は同じ値を設定してください。

参考

冗長化した負荷分散機を複数の Web システムで共有する場合は、各 Web システムの負荷分散機の定義で、<management-host> タグおよび<redundant-management-host> タグに同じ IP アドレスを設定してください。

4.8 その他の機能を使用したシステムの構築

その他の機能とは、前節までで説明していないアプリケーションサーバの機能のことです。ここでは、他の機能を使用したシステムを構築する手順について説明します。

4.8.1 その他の機能を使用するには

他の機能を使用したシステムを構築する場合、まず、使用する機能に合わせて必要な論理サーバを検討してシステム構成を決定します。使用する機能を決定する際には、次に示す機能解説のマニュアルを参照してください。

- ・マニュアル「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「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「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「アプリケーションサーバ 機能解説 拡張編」
- ・マニュアル「アプリケーションサーバ 機能解説 セキュリティ管理機能編」
- ・マニュアル「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「アプリケーションサーバ 機能解説 保守／移行編」

次に、使用する機能に合わせて各論理サーバの環境を設定してシステムを構築します。デフォルトの設定で有効な機能と無効な機能がありますので、必要に応じて論理サーバの環境設定を変更してください。

他の機能を使用したシステムの構築時に実施する作業と作業の要否を、利用する構築ツールごとに次の表に示します。

表 4-39 他の機能を使用したシステムの構築時に実施する作業と作業の要否（運用管理ポータル利用時）

他の機能を使用したシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照箇所
システム設計の準備	○	必ず実施	設計	2 章
システム構成の検討	○	必ず実施		3 章
使用するリソースの見積もり	○	必ず実施		5 章
データベースの準備	△	データベースに接続する場合	このマニュアル	2.1.1
データベースの環境の設定（Oracle の設定）	△	データベース（Oracle）に接続する場合		4.1.2

他の機能を使用したシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき		4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)		2.2.2
		必ず実施 (OS が AIX の場合)		2.2.3
		必ず実施 (OS が Linux の場合)		2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合		2.2.5
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合		4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合		4.1.6
		データベース (Oracle) に接続する場合		4.1.7
		データベース (SQL Server) に接続する場合		4.1.8
		データベース (MySQL) に接続する場合		4.1.9
		データベース (PostgreSQL) に接続する場合		4.1.10
システムの環境変数の設定	○	必ず実施		4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合		4.1.14
運用管理機能の構築	○	必ず実施	運用管理ポータル	2.2
運用管理エージェントの起動	○	必ず実施		4.1.2
Management Server の起動	○	必ず実施		4.1.2
システム環境の設定	○	必ず実施		3.2
システム環境のセットアップ	○	必ず実施		
システムの起動 (論理サーバの起動)	○	必ず実施		4.1.2
リソースアダプタの設定	△	リソースアダプタ (DB Connector) によるリソース接続をする場合		3.3

その他の機能を使用したシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
		リソースアダプタ (DB Connector 以外) によるリソース接続をする場合		
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合		4.1.2
業務アプリケーションの設定と開始	△	オンライン業務 (J2EE アプリケーション) を実行する場合		3.4, 4.1.2

(凡例)

設計：マニュアル「アプリケーションサーバ システム設計ガイド」を示します。

運用管理ポータル：マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」を示します。

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

表 4-40 その他の機能を使用したシステムの構築時に実施する作業と作業の要否 (Smart Composer 機能利用時)

その他の機能を使用したシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
システム設計の準備	○	必ず実施	設計	2 章
システム構成の検討	○	必ず実施		3 章
使用するリソースの見積もり	○	必ず実施		5 章
データベースの準備	△	データベースに接続する場合	このマニュアル	2.1.1
データベースの環境の設定 (Oracle の設定)	△	データベース (Oracle) に接続する場合		4.1.2
Component Container 管理者の設定 (UNIX の場合)	△	OS が UNIX の場合で一般ユーザに操作権限を与えるとき		4.1.3
Application Server のインストール	○	必ず実施 (OS が Windows の場合)		2.2.2
		必ず実施 (OS が AIX の場合)		2.2.3
		必ず実施 (OS が Linux の場合)		2.2.4
データベースに接続するための製品のインストール	△	データベースに接続する場合		2.2.5

他の機能を使用したシステムの構築時に実施する作業	作業の要否	作業の実施条件	参照先マニュアル	参照個所
アプリケーションサーバでサポートしているシンボリックリンク (UNIX の場合)	△	OS が UNIX の場合		4.1.5
データベース接続環境の設定	△	データベース (HiRDB) に接続する場合		4.1.6
		データベース (Oracle) に接続する場合		4.1.7
		データベース (SQL Server) に接続する場合		4.1.8
		データベース (MySQL) に接続する場合		4.1.9
		データベース (PostgreSQL) に接続する場合		4.1.10
システムの環境変数の設定	○	必ず実施		4.1.13
論理サーバの環境変数の設定	△	インストール時の環境変数を変更する場合		4.1.14
運用管理機能の構築	○	必ず実施		4.1.16
運用管理エージェントの起動	○	必ず実施		4.1.19
Management Server の起動	○	必ず実施		4.1.20
システム環境の設定 (簡易構築定義ファイルの設定)	○	必ず実施		4.1.22, 4.8.2 ※
システム環境の構築 (セットアップ)	○	必ず実施		4.1.25
システムの起動	○	必ず実施		4.1.26
リソースアダプタの設定	△	リソースアダプタ (DB Connector) によるリソース接続をする場合		4.1.28
		リソースアダプタ (DB Connector 以外) によるリソース接続をする場合		4.1.29
リソースアダプタの開始	△	リソースアダプタによるリソース接続をする場合		4.1.30
業務アプリケーションの設定と開始	△	オンライン業務 (J2EE アプリケーション) を実行する場合		4.1.31

(凡例)

設計：マニュアル「アプリケーションサーバ システム設計ガイド」を示します。

○：必ず実施する作業です。

△：作業の実施条件を満たす場合に実施する作業です。

注※ 次の機能を使用するシステムを構築する場合に参照してください。

- CTM を使用したスケジューリング機能
- NIO HTTP サーバ機能 (Web サーバ連携を行わない場合)

4.8.2 簡易構築定義ファイルを設定するときの注意事項（その他の機能使用時）

ここでは、次のアプリケーションサーバの機能を使用するシステムを、Smart Composer 機能を使用して、簡易構築定義ファイルで設定するときの注意事項について説明します。

- CTM を使用したスケジューリング機能
- NIO HTTP サーバ機能 (Web サーバ連携を行わない場合)

(1) J2EE アプリケーションを実行するシステムで CTM を使用したスケジューリング機能を使用する場合

CTM を使用するシステムでは、CTM にリクエストを送信するクライアントの定義が必要です。クライアントとして、Smart Composer 機能や運用管理ポータルを使用して構築した Web フロントシステム以外にも、EJB クライアントやユーザが独自に定義する Web フロントシステムが利用できます。

統合ネーミングスケジューラサーバ（グローバル CORBA ネーミングサービスを配置したホスト）を使用した、統合ネーミングスケジューラサーバモデルで構築する場合、簡易構築定義ファイルでは、統合ネーミングスケジューラサーバと、アプリケーションサーバを定義する必要があります。統合ネーミングスケジューラサーバとアプリケーションサーバは、それぞれ別の Web システムで定義します。また、CTM にリクエストを送信するシステムとして、Smart Composer 機能で構築する Web フロントシステムを使用する場合は、スケジューリング機能で使用する Web システムとは別に Web システムを定義してください。

Web システムごとに、簡易構築定義ファイルで必要な定義について説明します。

(a) アプリケーションサーバ (CTM) 用の Web システム

この Web システムでは、アプリケーションサーバを定義します。

- Web システムの属性定義
アプリケーションサーバを構築する Web システムを定義します。
- 負荷分散機の定義
負荷分散機は定義できません。負荷分散機を定義した場合は、エラーとなります。
- 物理ティアの定義
物理ティアの種類には ctm-tier を定義します。

アプリケーションサーバでは、複数の J2EE サーバが配置できます。複数の J2EE サーバを配置する場合は、物理ティアの定義で、1 ホストに配置する J2EE サーバの数を設定します。なお、1 ホストに配置する J2EE サーバの数は、Web システムの構成変更時に変更できません。

ctm-tier では、CTM ドメインマネージャ、CTM、スマートエージェントおよびパフォーマンストレーサのコンフィグレーションと、J2EE サーバ全体で共通のコンフィグレーションを設定できます。複数の J2EE サーバを配置する場合は、同じ番号（<server-no> タグ）を持つ J2EE サーバで共通のコンフィグレーションも設定できます。

J2EE サーバごとに適用されるコンフィグレーション定義が複数ある場合、各定義で同じパラメタを指定したときは、次の 1., 2., 3. の順に優先して適用されます。

1. サービスユニットの定義（特定の J2EE サーバの定義）
2. 物理ティアの定義（同じ番号を持つ J2EE サーバで共通の定義）
3. 物理ティアの定義（J2EE サーバ全体で共通の定義）

統合ネーミングスケジューラサーバとの関係を定義するため、次に示すパラメタは、統合ネーミングスケジューラサーバと同じ値を設定してください。

- CTM ドメインマネージャが属する CTM ドメイン名（論理 CTM ドメインマネージャの ctmdomname）
- CTM ドメインマネージャで使用するポート番号（論理 CTM ドメインマネージャの cdm.port）
- スマートエージェントで使用するポート番号（論理スマートエージェントの smartagent.port）

- サービスユニットの定義

CTM 関連のプロセス、J2EE サーバを配置するサービスユニットを定義します。

サービスユニットごとに、CTM ドメインマネージャ、CTM、スマートエージェントおよびパフォーマンストレーサのコンフィグレーションと、特定の J2EE サーバのコンフィグレーションを設定できます。複数の J2EE サーバを配置する場合、サービスユニット内の J2EE サーバは、番号（<server-no> タグ）で特定します。

CTM を識別する CTM 識別子（ctmid）は、運用管理ドメイン内でユニークな値を設定する必要があります。CTM 識別子（ctmid）の設定は省略できます。

- ホストの定義

CTM 関連のプロセス、J2EE サーバを配置するアプリケーションサーバを定義します。

(b) 統合ネーミングスケジューラサーバ用の Web システム

統合ネーミングスケジューラサーバモデルで構築する場合に、この Web システムが必要になります。この Web システムでは、統合ネーミングスケジューラサーバ（グローバル CORBA ネーミングサービスを配置したマシン）を定義します。

- Web システムの属性定義

統合ネーミングスケジューラサーバを構築する Web システムを定義します。

- 負荷分散機の定義

負荷分散機は定義できません。負荷分散機を定義した場合は、エラーとなります。

- 物理ティアの定義

物理ティアの種類には ctm-tier を定義します。

統合ネーミングスケジューラサーバでは J2EE サーバを配置しないため、 ctm-tier で、 J2EE サーバの数に 0 を設定します。

ctm-tier では、 CTM ドメインマネージャ、 CTM、スマートエージェントおよびパフォーマンストレーサのコンフィグレーションを設定できます。

- サービスユニットの定義

CTM 関連のプロセスを配置するサービスユニットを定義します。

- ホストの定義

統合ネーミングスケジューラサーバを定義します。

(2) NIO HTTP サーバ機能 (Web サーバ連携を行わない場合) を使用する場合

簡易構築定義ファイルでは、次の定義が必要になります。なお、 j2ee-tier では負荷分散機が使用できないため、負荷分散機の定義は削除します。負荷分散機を使用する場合は、運用管理ポータルを使用して構築してください。

- 物理ティアの定義

物理ティアの種類には j2ee-tier を定義します。また、 j2ee-tier で、 NIO HTTP サーバ機能を使用するための設定を定義します。

- サービスユニットの定義

J2EE サーバを配置するサービスユニットを定義します。 NIO HTTP サーバが使用するポート番号を定義します。

(3) 各機能で共通の注意事項

各機能を使用する場合に、 1 ホストに複数のサービスユニットを配置できます。ただし、同じ CTM ドメインのサービスユニットは、 1 ホスト内に複数配置できません。 1 ホスト内に複数のサービスユニットを配置するときの注意事項を次に示します。

(a) 論理サーバの設定に関する注意事項

1 ホストに複数のサービスユニットをすると、同じホスト内に同じ種類の論理サーバが複数存在することになります。このため、簡易構築定義ファイルを作成する際には、各論理サーバのコンフィグレーションで、次に示す項目の指定値を重複しないようにしてください。

表 4-41 論理サーバごとに重複しないように設定する項目

論理サーバの種類	項目	param-name の指定値
Web サーバ	http ポート番号	Listen

論理サーバの種類	項目	param-name の指定値
J2EE サーバ	NIO HTTP サーバのポート番号	webserver.connector.nio_http.port ^{※1}
	ajp13 のポート番号	webserver.connector.ajp13.port ^{※2}
	インプロセスのネーミングサービス用のポート番号	ejbserver.naming.port
	インプロセスの HTTP サーバのポート番号	webserver.connector.inprocess_http.port ^{※2}
	JTA リカバリの固定ポート番号	ejbserver.distributedtx.recovery.port
	RMI レジストリのポート番号	ejbserver.rmi.naming.port
	管理用サーバのポート番号	ejbserver.http.port
	稼働情報取得時のリクエスト受付ポート番号	ejbserver.rmi.remote.listener.port
パフォーマンストレーサ	PRF 識別子	PRFID
スマートエージェント	スマートエージェントが監視するポート番号	smartagent.port
CTM ドメインマネジャ	CTM ドメインマネジャが属する CTM ドメイン名	ctmdomname
	CTM 実行環境ディレクトリ	ctmspool
	CTM ドメインマネジャが CTM ドメイン構成情報を交換するために使用するポート番号	cdm.port
CTM	CTM の識別子	ctmid ^{※3}
	CTM が使用するポート番号	ctm.port
	CTM と対となるネーミングサービスのポート番号	ctm.ns.port

注※1 推奨モードで使用されます。推奨モードを使用しない場合は Management Server で無視されます。

注※2 V9 互換モードで使用されます。V9 互換モードを使用しない場合は Management Server で無視されます。

注※3 ctmid の設定は省略できます。省略時は、ホスト間で一意になるように Management Server で自動的に設定されます。

参考

1 ホスト内にある複数のサービスユニットを同時に起動しない運用をする場合は、各サービスユニットの同じ種類の論理サーバに同じ値を設定できます。

ホストに配置するサービスユニットが属する Web システムによって、これらの設定項目をどの定義のコンフィグレーションで指定するかが異なります。

- 同じ Web システムのサービスユニットを同じホストに配置する場合
各サービスユニットの論理サーバ定義（<define-server> タグ）内の<configuration> タグで指定します。
- 異なる Web システムのサービスユニットを同じホストに配置する場合

各 Web システムの物理ティア定義（<tier> タグ）内の<configuration>タグで指定します。

(b) Web サーバのポート番号に関する注意事項

サーバマシン内に複数のサービスユニットを配置する場合に、Web サーバのポート番号を変更するときは、次に示す制限事項があります。なお、複数のサービスユニットが、同一の Web システム内にある場合も、異なる Web システムにある場合も、この制限事項の対象となります。

- 複数のサービスユニットの Web サーバで同じポート番号を設定している場合

1 台のサーバマシン内で、複数のサービスユニットを同時に起動しない運用をする場合は、各サービスユニットの Web サーバに同じポート番号を設定できます。

この場合は、次のどちらかの手順で変更してください。

- すべてのサービスユニットの Web サーバのポート番号を同時に変更します。
- 一部のサービスユニットの Web サーバのポート番号を変更します。そのあと、ほかのサービスユニットの Web サーバのポート番号を変更します。

注意事項

一部のサービスユニットの Web サーバのポート番号を変更した場合、ほかのサービスユニットの Web サーバのポート番号は削除され、ポート番号を削除された Web サーバは使用できなくなります。このため、必ず、あとからほかのサービスユニットの Web サーバのポート番号も変更してください。

- 複数のサービスユニットの Web サーバで異なるポート番号を設定している場合

ここでは、サービスユニット 1 (unit1) の Web サーバでポート番号に 80、サービスユニット 2 (unit2) の Web サーバでポート番号に 81 を設定している場合を例に説明します。ポート番号の設定変更の方法と変更可否を次の表に示します。

表 4-42 ポート番号の変更方法と変更可否

ポート番号の変更方法	論理サーバのパラメタ変更用の構成変更定義 ファイルの設定方法	変更後の Web サーバの ポート番号		変更可否
		unit1	unit2	
各サービスユニットの Web サーバ のポート番号を同じ番号へ変更	サービスユニット 1 の Web サーバにポート番号 81 を設定します。	81	81	○
	サービスユニット 2 の Web サーバにポート番号 80 を設定します。	80	80	○
各サービスユニットの Web サーバ のポート番号の入れ替え	サービスユニット 1 の Web サーバにポート番号 81 を、サービスユニット 2 の Web サーバにポート番号 80 を設定します。	81	80	×
一方のサービスユニットの Web サーバを他方で設定済みのポート 番号に変更して、他方のサービス	サービスユニット 1 の Web サーバにポート番号 81 を、サービスユニット 2 の Web サーバにポート番号 82 を設定します。	81	82	×

ポート番号の変更方法	論理サーバのパラメタ変更用の構成変更定義 ファイルの設定方法	変更後の Web サーバの ポート番号		変更可否
		unit1	unit2	
ユニットの Web サーバを未使用の ポート番号 (82 とする) へ変更	サービスユニット 1 の Web サーバにポート番 号 82 を、サービスユニット 2 の Web サーバ にポート番号 80 を設定します。	82	80	×
一方のサービスユニットの Web サーバを未使用のポート番号 (82 とする) へ変更	サービスユニット 1 の Web サーバにポート番 号 82 を設定します。	82	81	○
	サービスユニット 2 の Web サーバにポート番 号 82 を設定します。	80	82	○

(凡例) ○: 変更できます。 ×: 変更できません。

参考

設定済みのポート番号に変更する場合に、1回の操作で変更できない変更方法は、いったん未
使用のポート番号を設定してから入れ替えたり、複数回に分けて変更したりするなどの、設定
変更操作を実施すれば変更できます。

4.8.3 WebSocket を使用する場合の設定

(1) WebSocket のログを出力する場合の設定

論理 Web サーバの J2EE サーバとの連携方法がリバースプロキシの環境で、WebSocket を使用したアプ
リケーションを実行する場合、論理 Web サーバに HWSWebSocketLog ディレクティブを設定してくだ
さい。HWSWebSocketLog ディレクティブの設定例を次の表に示します。

表 4-43 HWSWebSocketLog ディレクティブの設定例

対象 OS	設定例
Windows	HWSWebSocketLog " *"<Application Server のインストールディレクトリ>/httpsd/sbin/ rotatelogs.exe*" "<Application Server のインストールディレクトリ>/httpsd/servers/HWS_< 論理 Web サーバの実サーバ名>/logs/hws_websocket_log" <分割時間間隔> -fnum <ファイル数 > -diff <分割基準時間差分>"
UNIX	HWSWebSocketLog " /opt/hitachi/httpsd/sbin/rotatelogs /opt/hitachi/httpsd/servers/ HWS_<論理 Web サーバの実サーバ名>/logs/hws_websocket_log <分割時間間隔> -fnum <ファイル数 > -diff <分割基準時間差分>"

注※ ファイルおよびディレクトリのパスは、絶対パスで設定してください。

HWSWebSocketLog ディレクティブの詳細は、マニュアル「HTTP Server」を参照してください。

運用管理ポータルで設定する場合は、追加ディレクティブで設定してください。追加ディレクティブについては、マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」の「10.9.1 Web サーバの設定」を参照してください。

Smart Composer で設定する場合は、AppendDirectives パラメタ（追加ディレクティブ）で設定してください。AppendDirectives パラメタについては、マニュアル「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)」の「4.10.2 Web サーバ用定義を設定するパラメタ」を参照してください。

4.8.4 HWSGracefulStopTimeout ディレクティブと Manager の停止監視時間の設定に関する注意点

次の表に示す条件に応じて、論理 Web サーバに HWSGracefulStopTimeout ディレクティブを設定してください。

表 4-44 HWSGracefulStopTimeout ディレクティブの設定に関する注意点

Cosminexus Component Container のバージョン	WebSocket 使用有無	注意点
11-00-01 より前	—	設定不要※1
11-00-01 以降	使用しない	停止監視時間を、HWSGracefulStopTimeout ディレクティブのデフォルト値以上の値に変更する場合、HWSGracefulStopTimeout ディレクティブには、0、または停止監視時間より大きな値を設定してください。※2※3※5
	使用する	HWSGracefulStopTimeout ディレクティブには、1 以上、かつ停止監視時間より小さな値を設定してください。※2※4※5

(凡例) —：該当しません。

注※1 Manager が HWSGracefulStopTimeout ディレクティブに自動で 0 を設定します。

注※2 停止監視時間と HWSGracefulStopTimeout ディレクティブの両方に 0 を設定すると、Cosminexus HTTP Server が正常停止できなかった場合に、無限に待ち続けてしまうため、停止監視時間の 0（監視しない）と、HWSGracefulStopTimeout ディレクティブの 0 を、同時に設定しないでください。

注※3 Manager から停止した場合に停止監視時間より先に Cosminexus HTTP Server がタイムアウトするのを防ぐためです。

注※4 Manager から停止した場合に停止監視時間より先に Cosminexus HTTP Server がタイムアウトしないことによって、Manager が Cosminexus HTTP Server を強制停止するのを防ぐためです。

注※5 KeepAliveTimeout ディレクティブには、停止監視時間より小さい値を設定してください。

HWSGracefulStopTimeout ディレクティブを運用管理ポータルで設定する場合は、追加ディレクティブで設定してください。追加ディレクティブについては、マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」の「10.9.1 Web サーバの設定」を参照してください。Smart Composer で設定する場合は、AppendDirectives パラメタ（追加ディレクティブ）で設定してください。

AppendDirectives パラメタについては、マニュアル「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)」の「4.10.2 Web サーバ用定義を設定するパラメタ」を参照してください。

KeepAliveTimeout ディレクティブを運用管理ポータルで設定する場合は、KeepAlive 接続時の要求待ち時間で設定してください。KeepAlive 接続時の要求待ち時間については、マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」の「10.9.1 Web サーバの設定」を参照してください。Smart Composer で設定する場合は、KeepAliveTimeout パラメタで設定してください。KeepAliveTimeout パラメタについては、マニュアル「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)」の「4.10.2 Web サーバ用定義を設定するパラメタ」を参照してください。

4.9 運用上の注意事項

- 次の Manager のコマンドや操作を、同種・異種の組み合わせに関係なく、100 以上同時に実行しないでください。実行した場合、コマンドは接続エラーとなり、運用管理ポータルの操作では 503 エラーが表示されます。

<コマンドの実行>

- Smart Composer 機能で使用するコマンド
- mngsvrutil コマンド

<運用管理ポータルの操作>

- ブラウザ画面で運用管理ポータルを開き、機能操作を行う
- Manager を日本語ロケール以外のロケールで使用する場合に、次の条件がすべて重なるときは、ログファイルに出力されるメッセージが文字化けします。
 - 運用管理ポータルの設定項目の値に日本語が指定され、その値がログに出力される
 - 簡易構築定義ファイルおよび構成変更定義ファイルで論理サーバのコンフィグレーションのパラメータに日本語の値が指定され、その値がログに出力される
- UNIX の場合、Management Server、運用管理エージェント、および次に示すコマンドのロケールを統一してください。
 - mngsvrutil コマンド
 - snapshotlog コマンド
 - Smart Composer 機能で使用するコマンド (cmx_ から始まるコマンド)
- 運用管理エージェントまたは Management Server の起動に続けて、運用管理コマンド (mngsvrutil) または Smart Composer 機能などで Management Server にアクセスする場合、事前に次のどちらかの対策をしてください。
 - 運用管理エージェントまたは Management Server の起動時に同期実行を有効にし、起動に続けてコマンドを実行する。
 - mngsvrutil check または adminagentcheck コマンドで、運用管理エージェントまたは Management Server が起動済みであることを確認する。

5

システムの構築・運用時に使用する画面、コマンド およびファイル

この章では、システムの構築・運用で使用する運用管理ポータルの画面、コマンドおよびファイルについて説明します。

5.1 システムの構築・運用時に使用する画面

ここでは、システムの構築・運用時に使用する運用管理ポータルの画面について説明します。

表 5-1 システムの構築・運用時に使用する運用管理ポータルの画面

画面名	画面の説明	「運用管理ポータル操作ガイド」の参照箇所※
【Cosminexus Management Server の設定】画面	Management Server に関する設定のルートです。	8.1
【J2EE アプリケーションのアップロード】画面	J2EE アプリケーションを Management Server 稼働ホストにアップロードします。	12.2.1
【J2EE アプリケーションのインポート】画面	論理 J2EE サーバにアーカイブ形式の J2EE アプリケーションをインポートします。	12.3.3
【J2EE アプリケーションの開始／停止】画面	論理 J2EE サーバにインポートした J2EE アプリケーションを開始します。	12.3.1
【J2EE アプリケーションの開始】画面		
【J2EE アプリケーションの削除】画面	インポート済みの停止している J2EE アプリケーションを削除します。	12.3.5
【JP1 連携の設定】画面	Application Server の障害情報を JP1 イベントとして発行できます。	8.2.7
【Management Server ログイン】画面	運用管理ポータルにログインするかどうかを設定します。	7.1.2
【Web コンテナの設定】画面	Web コンテナが起動するときのオプションを設定します。	10.8.4
【Web サーバの設定】画面	Web サーバを稼働させるために必要な情報を設定します。	10.9.1
【一括起動】画面	運用管理ドメイン内のすべての論理サーバを一括起動できます。	11.3.2
【一括再起動】画面	運用管理ドメイン内のすべての論理サーバを一括で再起動できます。	11.3.4
【一括停止】画面	運用管理ドメイン内のすべての論理サーバを一括停止できます。	11.3.3
【運用管理ポータル】画面	運用管理ポータルのトップページです。	7.2
【開始時の設定】画面	Management Server を起動するときに、論理サーバを一括起動するかどうかを設定します。	8.2.5

画面名	画面の説明	「運用管理ポータル操作ガイド」の参照箇所※
【環境変数の設定】画面	サーバ起動時の環境変数の情報を設定します。	10.8.27
【管理ユーザアカウントの設定】画面	Management Server ログインするための管理ユーザアカウントを設定します。	8.2.1
【<サーバの種類>の稼働情報監視】画面	J2EE サーバ内の、J2EE コンテナや JavaVM の稼働情報を監視できます。	13.3
【<サーバの種類>のステータス監視】画面	選択した論理サーバのステータスを監視できます。	13.2.3
【性能解析トレース収集の設定】画面	性能解析トレースファイルを圧縮する ZIP 形式のファイルの面数を指定します。	8.2.6
【設定情報の配布】画面	設定した情報を各サーバに配布します。	10.10.1
【トランザクションの設定】画面	トランザクションの情報を設定します。	10.8.6
【リソースアダプタの Connector 属性ファイル編集】画面	Connector 属性ファイルの内容を表示または設定します。	12.4.5
【リソースアダプタのインポート】画面	該当論理 J2EE サーバにリソースアダプタをインポートします。	12.4.4
【リソースアダプタの開始】画面	論理 J2EE サーバにインポートしたりソースアダプタを開始します。	12.4.1
【リソースアダプタの開始／停止】画面		
【リソースアダプタの基本設定】画面	リソースアダプタのプロパティの基本的な設定を表示または設定します。	12.4.5
【リソースアダプタの停止】画面	稼働中のリソースアダプタを停止します。	12.4.2
【リソースアダプタのプロパティ設定】画面	論理 J2EE サーバにインポートしたりソースアダプタから、データベースに接続するためのプロパティを表示または設定します。	12.4.5
【ログの設定】画面	Management Server が output するログ情報について設定します。	8.2.3
【ログの表示】画面	運用管理 ドメイン内の論理サーバ、J2EE アプリケーションおよびリソースアダプタにかかる実行結果を、ログとして一覧で参照できます。	10.10.2, 11.3.6, 12.3.6, 12.4.7
【論理サーバの稼働状況】画面	運用管理 ドメイン内のすべての論理サーバの稼働状況を、一覧で確認できます。	11.3.1

画面名	画面の説明	「運用管理ポータル操作ガイド」の参照箇所※
[論理サーバのステータス監視] 画面	運用管理ドメイン内にあるすべての論理サーバのステータスを監視できます。	13.2.2

注※ マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」の参照先を示します。

5.2 システムの構築・運用時に使用するコマンド

ここでは、システムの構築・運用時に使用するコマンドの種類、OSごとのコマンドの格納先について説明します。

表 5-2 システムの構築・運用時に使用するコマンド

コマンド名	コマンドの説明	コマンドの格納先	「コマンド編」の参照箇所 ^{※1}
adminagentctl	運用管理エージェントの起動と停止	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	7.2
cjdeployrar	リソースアダプタのデプロイ	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.4
cjgetrarprop	RAR ファイルの属性の取得	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	
cjimportapp	J2EE アプリケーションのインポート	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.3
cjimportres	リソースのインポート	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.4
cjreplaceapp	アプリケーションの入れ替え	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.3
cjsetrarprop	RAR 属性設定	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.4

コマンド名	コマンドの説明	コマンドの格納先	「コマンド編」の参照箇所 ^{※1}
		/opt/Cosminexus/CC/admin/bin/	
cjsetup	J2EE サーバのセットアップとアンセットアップ	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥server¥bin¥ UNIX の場合 /opt/Cosminexus/CC/server/bin/ 	2.2
cjstartapp	J2EE アプリケーションの開始	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.3
cjstarrar	リソースアダプタの開始	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	2.4
cjtestres	リソースの接続テスト	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥bin¥ UNIX の場合 /opt/Cosminexus/CC/admin/bin/ 	
cmx_build_model	Web システムの情報モデルの生成	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	8.3
cmx_build_system	Web システムの構築	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	
cmx_change_model	Web システムの情報モデルの変更	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	
cmx_delete_system	Web システムの削除	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	

コマンド名	コマンドの説明	コマンドの格納先	「コマンド編」の参照箇所 ^{※1}
		/opt/Cosminexus/manager/bin/	
cmx_export_model	簡易構築定義ファイルの出力	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	
cmx_list_model	Web システムの情報モデルの取得	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	
cmx_start_target	Web システムまたはサービスユニットの開始	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	
cmx_stop_target	Web システムまたはサービスユニットの停止	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	
hwsserveredit	複数の Web サーバ環境の生成	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥httpsd¥admin¥bin¥ UNIX の場合 /opt/hitachi/httpsd/admin/bin/ 	—※2
jar	アプリケーションへのアプリケーション属性ファイルの設定	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥jdk¥bin¥ UNIX の場合 /opt/Cosminexus/jdk/bin/ 	—
mngautorun	自動起動および自動再起動の設定 ／設定解除	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	7.2
mngenvsetup	管理グループの設定	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストール ディレクトリ>¥manager¥bin¥ UNIX の場合 /opt/Cosminexus/manager/bin/ 	

コマンド名	コマンドの説明	コマンドの格納先	「コマンド編」の参照箇所 ^{※1}
		/opt/Cosminexus/manager/bin/	
mngsvrctl	Management Server の起動／停止／セットアップ	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/bin UNIX の場合 /opt/Cosminexus/manager/bin/ 	
mngsvrutil	Management Server の運用管理コマンド	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/bin UNIX の場合 /opt/Cosminexus/manager/bin/ 	
setupwizard	システムのセットアップ, アンセットアップと複製	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/setup/bin UNIX の場合 /opt/Cosminexus/manager/setup/bin/ 	

(凡例)

– : 該当しません。

注※1 マニュアル「アプリケーションサーバ リファレンス コマンド編」の参照先を示します。

注※2 hwsserveredit コマンドについては、マニュアル「HTTP Server」の「4.12.1 複数の Web サーバ環境の生成 (hwsserveredit コマンド)」を参照してください。

5.3 システムの構築・運用時に使用するファイル

ここでは、システムの構築・運用時に使用するファイルの種類、テンプレートファイルや、OS ごとの格納場所について説明します。

表 5-3 システムの運用環境の構築・運用時に使用するファイル

ファイル名	ファイルの説明	ファイルの格納先	「定義編(サーバ定義)」の参照個所 ^{※1}
adminagent.properties	運用管理エージェントプロパティファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/config UNIX の場合 /opt/Cosminexus/manager/config/ 	8.2.1
adminagent.xml	運用管理エージェント設定ファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/config UNIX の場合 /opt/Cosminexus/manager/config/ 	8.2.4
Connector 属性ファイル	リソースの設定で使用する属性ファイル	—	— ^{※2}
cosminexus.xml	アプリケーション属性ファイル	<EAR のルート>/META-INF/cosminexus.xml	— ^{※3}
lb.properties	負荷分散機定義プロパティファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/config UNIX の場合 /opt/Cosminexus/manager/config/ 	4.2.4
manager.cfg	Manager 設定ファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/config UNIX の場合 /opt/Cosminexus/manager/config/ 	8.2.9
mngagent.<実サーバ名>.properties	運用監視エージェントプロパティファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/config UNIX の場合 /opt/Cosminexus/manager/config/ 	8.2.5
mserver.properties	Management Server 環境設定ファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/manager/config 	8.2.6

ファイル名	ファイルの説明	ファイルの格納先	「定義編(サーバ定義)」の参照個所※1
		<ul style="list-style-type: none"> UNIX の場合 /opt/Cosminexus/manager/config/ 	
mevent.<論理サーバ名>.properties	Management イベント発行用プロパティファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥config¥ UNIX の場合 /opt/Cosminexus/manager/config/ 	8.2.11
.mngsvrutilrc	mngsvrutil コマンドのクライアント側定義ファイル	各 OS ユーザのホームディレクトリ	8.2.14
mngsvrutilcl.properties	mngsvrutil コマンドのクライアント側共通定義ファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥config¥ UNIX の場合 /opt/Cosminexus/manager/config/ 	8.2.16
setupwizard_success.dat	セットアップ情報ファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ >¥manager¥setup¥config¥ UNIX の場合 /opt/Cosminexus/manager/setup/config/ 	—※4
簡易構築定義ファイル	Smart Composer 機能で使用するファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ >¥manager¥config¥templates¥cmxde fcombinedmodel.xml※5 UNIX の場合 /opt/Cosminexus/manager/config/templates/cmxdedefcombinedmodel.xml※5 	4.3
構成変更定義ファイル	Smart Composer 機能で使用するファイル	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ >¥manager¥config¥templates¥cmxm odcombinedmodel.xml※5 <Application Server のインストールディレクトリ >¥manager¥config¥templates¥cmxad dcombinedmodel.xml※5 UNIX の場合 	4.4

ファイル名	ファイルの説明	ファイルの格納先	「定義編(サーバ定義)」の参照個所※1
		/opt/Cosminexus/manager/config/templates/ cmxmodcombinedmodel.xml※5 /opt/Cosminexus/manager/config/templates/ cmxaddcombinedmodel.xml※5	

(凡例)

– : 該当しません。

注※1 マニュアル「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)」の参照先を示します。

注※2 Connector 属性ファイルについては、マニュアル「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「4.1 Connector 属性ファイル」を参照してください。

注※3 cosminexus.xml については、マニュアル「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. 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。

注※4 setupwizard_success.dat については、「[2.2.7 システム環境をセットアップするときの注意事項](#)」を参照してください。

注※5 テンプレートファイルです。

付録

付録 A セットアップウィザードで設定されるパラメタ

セットアップウィザードでシステムをセットアップした場合、システムの各パラメタには、次の値が使用されます。

1. 画面で値を設定したパラメタ

設定した値が使用されます。

2. 画面で値を設定していないパラメタ

簡易構築定義ファイルに指定できるパラメタの省略値が使用されます。

簡易構築定義ファイルに指定できるパラメタ、およびパラメタの省略値については、マニュアル「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)」の「4.3 簡易構築定義ファイル」を参照してください。

ただし、2.のパラメタについては省略値と異なる値が設定されるものがあります。

ここでは、2.のパラメタのうち、省略値と異なる値が設定されるパラメタについて、設定される論理サーバごとに説明します。

付録 A.1 セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ (論理 Web サーバ)

セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタのうち、論理 Web サーバに設定されるパラメタについて次の表に示します。パラメタの詳細については、マニュアル「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)」の「4.10 論理 Web サーバで指定できるパラメタ」を参照してください。

表 A-1 パラメタの省略値と異なる値が設定されるパラメタ (論理 Web サーバ)

パラメタ名	パラメタの省略値	セットアップウィザードで設定される値	説明
HWSKeepStartServers ^{※1}	Off	On	サーバプロセス稼働数が維持されます。
KeepAliveTimeout	15	3	KeepAlive 接続時の要求待ち時間として 3 (単位:秒) を指定します。
Listen	—	80	リクエストを受け付けるポート番号として 80 を指定します。 カスタムセットアップの [システム環境の設定] 画面で Web サーバのポート番号を設定した場合、設定した値が有効になります。
MaxClients.worker ^{※1}	400	20	同時に接続できるクライアントの最大数として 20 を指定します。

パラメタ名	パラメタの省略値	セットアップウィザードで設定される値	説明
StartServers.worker ^{※1}	2	1	Web サーバ起動時のサーバプロセス数として 1 を指定します。
ThreadsPerChild ^{※2}	50	20	サーバとして起動するスレッド数として 20 を指定します。なお、指定したスレッド数はサーバの最大同時接続数を示します。
HttpsErrorMethod	Div	Wrap	HTTP Server のエラーログをラップアラウンド方式で出力します。
ThreadsPerChild.worker ^{※1}	40	20	一つのサーバプロセスに生成するサーバスレッド数として 20 を指定します。

(凡例) - : なし

注※1 使用する OS が UNIX の場合にだけ指定値が有効になります。

注※2 使用する OS が Windows の場合にだけ指定値が有効になります。

付録 A.2 セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ (論理 J2EE サーバ)

セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタのうち、論理 J2EE サーバに設定されるパラメタについて次の表に示します。パラメタの詳細については、マニュアル「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)」の「4.11 論理 J2EE サーバで指定できるパラメタ」を参照してください。

表 A-2 パラメタの省略値と異なる値が設定されるパラメタ (論理 J2EE サーバ)

パラメタ名	パラメタの省略値	セットアップウィザードで設定される値	説明
ejbserver.connecti onpool.applicatio nAuthentication.d isabled	false	true	コンテナ管理のサインオンの最適化機能を有効にします。
ejbserver.ext.met hod_observation.i nterval	0	10	J2EE アプリケーション実行時間監視機能を使用します。また、タイムアウトとメソッドキャンセルを異なるスレッドで実行する時間間隔として 10 (単位:秒) を指定します。
ejbserver.jta.Tran sactionManager.d efaultTimeOut	180	60	J2EE サーバ上で開始されるトランザクションのトランザクショントライアウトの値として 60 (単位:秒) を指定します。
ejbserver.logger.c hannels.define.W ebServletLogFile.f ilesize	4194304	16777216	Web サーブレットログのファイルサイズとして 16777216 (単位:バイト) を指定します。

パラメタ名	パラメタの省略値	セットアップウィザードで設定される値	説明
add.jvm.arg	–	-Xms512m	Java ヒープメモリ領域の初期サイズとして 512 (単位: メガバイト) を指定します。
	–	-Xmx512m	Java ヒープメモリ領域の最大サイズとして 512 (単位: メガバイト) を指定します。
	–	-XX:MetaspaceSize=128m	Metaspace 領域の FullGC のしきい値を変更する計算に使用する値であり、しきい値の最低値となる値を指定します。初期サイズとして 128 (単位: メガバイト) を指定します。
	–	-XX:MaxMetaspaceSize=128m	Metaspace 領域の最大サイズとして 128 (単位: メガバイト) を指定します。
	–	-XX:+HitachiVerboseGC	拡張 verbosegc 情報を出力します。
	–	-XX:HitachiVerboseGCIntervalTime=0	拡張 verbosegc 情報を出力する時間の間隔として 0 (単位: 秒) を指定します。
	–	-XX:+HitachiVerboseGCPrintCause	GC の要因内容を、拡張 verbosegc 情報の行末に出力します。
	–	-XX:+HitachiOutputMilliTime	JavaVM ログファイルに出力する日時に、ミリ秒まで出力します。
	–	-XX:+HitachiCommVerboseGC	拡張 verbosegc 情報の出力を、CSV ファイルで取得できるようにコンマ形式で出力します。
	–	-XX:HitachiJavaLogFile=4096k	JavaVM の作成するログファイルの最大サイズとして 4096 (単位: キロバイト) を指定します。
	–	-XX:HitachiJavaLogNumberOfFile=4	JavaVM の作成するログファイルの最大数として 4 を指定します。
webserver.connector.nio_http.port	8080	80	システム構成の選択時、J2EE サーバだけを配置するシステム (j2ee-tier) を選択した場合に設定されます。NIO HTTP サーバのポート番号を標準のポート番号に指定します。
cpp.library.version	Linux (64 ビット版) の場合 6 それ以外の場合 5	6	C++ライブラリのバージョンを指定します。このパラメタは Linux (64 ビット版を除く) だけで設定されます。
use.security	true	false	SecurityManager を使用しません。

(凡例) – : なし

注※ Shift_JIS 互換オプションを指定すると、エンコーディング名「Shift_JIS」が「MS932」の代替名として使用されます。

付録 A.3 セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ（論理パフォーマンストレーサ）

セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタのうち、論理パフォーマンストレーサに設定されるパラメタについて次の表に示します。パラメタの詳細については、マニュアル「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)」の「4.12 論理パフォーマンストレーサで指定できるパラメタ」を参照してください。

表 A-3 パラメタの省略値と異なる値が設定されるパラメタ（論理パフォーマンストレーサ）

パラメタ名	パラメタの省略値	セットアップウィザードで設定される値	説明
mstartup.retrywait	0	60	自動再起動リトライ間隔として 60（単位：秒）を指定します。
PrfTraceLevel	STANDARD	00140000	パフォーマンストレーサのトレース取得レベルを次のように指定します。 <ul style="list-style-type: none">• JCA/DB Connector 詳細• 他機能レイヤ 標準

付録 B インストール後のディレクトリ構成

ここでは、Application Server のインストール後のディレクトリ構成について説明します。また、Application Server の構成ソフトウェアのうち、Component Container のディレクトリ構成について説明します。

インストールディレクトリを次に示します。

- Windows のインストールディレクトリ（デフォルト）
<OS インストールドライブ>:\Program Files\Hitachi\Cosminexus
- UNIX のインストールディレクトリ
/opt/Cosminexus

付録 B.1 Application Server のディレクトリ構成

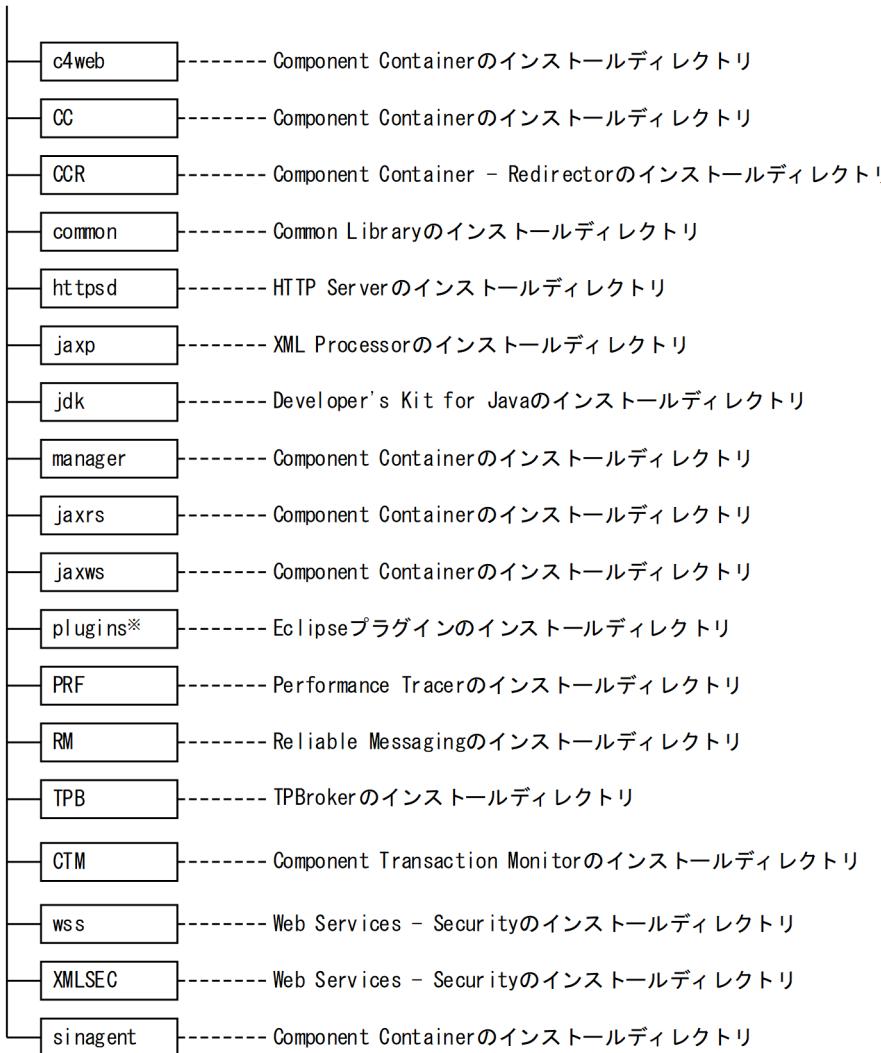
Application Server のディレクトリ構成を OS ごとに説明します。

- Windows の場合

Windows の場合の Application Server のディレクトリ構成を次の図に示します。

図 B-1 Application Server のディレクトリ構成 (Windows の場合)

〈Application Serverのインストールディレクトリ〉



(凡例)

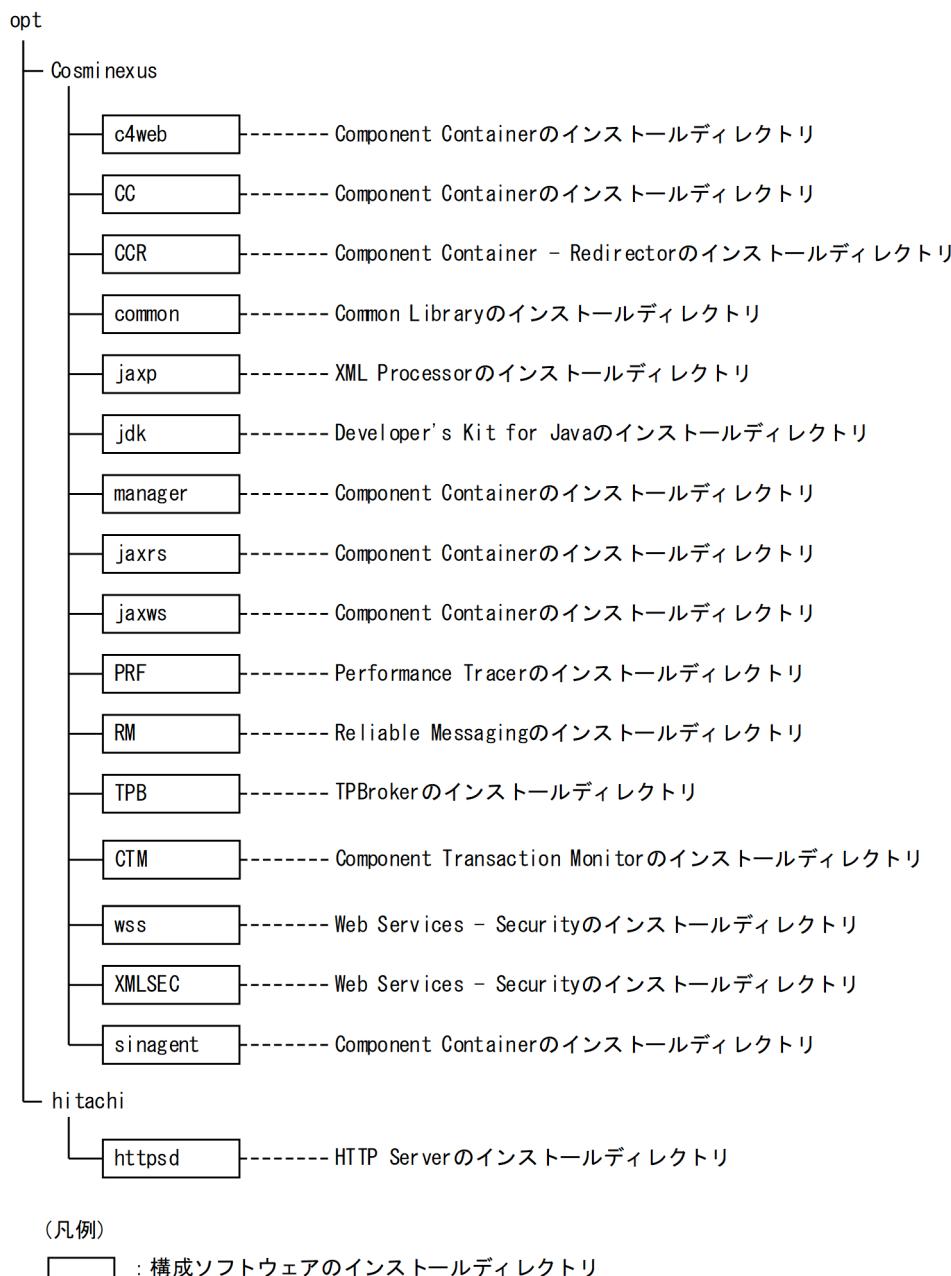
: 構成ソフトウェアのインストールディレクトリ

注※ Windowsの場合だけインストールされるディレクトリです。

• UNIX の場合

UNIX の場合の Application Server のディレクトリ構成を次の図に示します。

図 B-2 Application Server のディレクトリ構成 (UNIX の場合)



注意事項

インストールディレクトリについて

Application Server をインストールすると、/opt または/opt/Cosminexus のディレクトリ属性が変更されることがあります。

付録 B.2 Component Container のディレクトリ構成

付録 B.1 の図中にある、Component Container のディレクトリの下位のディレクトリ構成をディレクトリごとに表に示します。

• CC ディレクトリ

Component Container の提供する機能のうち、コンテナなどの基盤部分の機能を提供するディレクトリです。<Application Server のインストールディレクトリ>/CC (Windows の場合)、または/opt/Cosminexus/CC (UNIX の場合) です。

表 B-1 Component Container のディレクトリ構成 (CC ディレクトリ)

ディレクトリ		説明
adapters	—	アダプタを格納するディレクトリです。
	OpenTP1	TP1 インバウンドアダプタ格納用ディレクトリです。
admin	—	J2EE サーバ管理用ディレクトリです。
	bin	サーバ管理コマンド格納用ディレクトリです。
	dtds	CUI 用の DTD ファイル格納用ディレクトリです。
	logs	サーバ管理コマンドのログ格納用ディレクトリです。
	sysconf	サーバ管理コマンドのシステム定義ファイル格納用ディレクトリです。ただし、修正禁止です。
	templates	属性ファイルのテンプレート格納用ディレクトリです。
	usrconf	サーバ管理コマンドのユーザ定義ファイル格納用ディレクトリです (カスタマイズ用)。
batch	—	バッチアプリケーション用のディレクトリです。
	bin	バッチアプリケーションの実行ファイル格納用ディレクトリです。
	logs	バッチアプリケーションのログ出力用ディレクトリです。
	sysconf	バッチアプリケーションのシステム定義ファイル格納用ディレクトリです。ただし、修正禁止です。
	templates	バッチアプリケーションのユーザ定義ファイルのひな型格納用ディレクトリです。
cjmsp	—	CJMS プロバイダ用のディレクトリです。
	bin	CJMS プロバイダのコマンド格納用ディレクトリです。

ディレクトリ			説明
lib	props	—	CJMS プロバイダのライブラリファイルおよび RAR ファイル格納用ディレクトリです。
		—	CJMS プロバイダのプロパティファイル格納用ディレクトリです。
		broker	CJMS プロバイダのプローカプロパティファイル格納用ディレクトリです。
		templates	CJMSP リソースアダプタの Connector 属性ファイルのテンプレート格納用ディレクトリです。
	var	—	CJMS プロバイダのデフォルトディレクトリです。
		admin	CJMS プロバイダの管理コマンド用ディレクトリです。
		config	CJMS プロバイダの管理コマンドのプロパティ格納用ディレクトリです。
	instances	—	CJMS プロバイダのインスタンス格納用ディレクトリです。
client	—	—	EJB クライアントアプリケーション用のディレクトリです。
	bin	—	EJB クライアントアプリケーションの実行ファイル格納用ディレクトリです。
	lib	—	EJB クライアントアプリケーションの JAR ファイル格納用ディレクトリです。
	logs	—	EJB クライアントアプリケーションのログ出力用ディレクトリです。
	sysconf	—	EJB クライアントアプリケーションのシステム定義ファイル格納用ディレクトリです。ただし、修正禁止です。
	templates	—	EJB クライアントアプリケーションのユーザ定義ファイルのひな型格納用ディレクトリです。
DBConnector	—	—	DB Connector の RAR ファイル格納用ディレクトリです。
	ReliableMessaging	—	DB Connector for Reliable Messaging の RAR ファイル格納用ディレクトリです。
	ClusterPool	—	クラスタコネクションプール機能（互換機能）を使用する場合の RAR ファイル格納用ディレクトリです。
examples	—	—	サンプルプログラム格納用ディレクトリです。
lib	—	—	Component Container 用クラスライブラリファイル格納用ディレクトリです。

ディレクトリ	説明	
logs	インストール時のログ格納用ディレクトリです。	
server	—	—
	bin	J2EE サーバ用コマンドツール格納用ディレクトリです。
	lib	J2EE サーバ用ライブラリファイル格納用ディレクトリです。
	dtds	Component Container で使用する DTD ファイル格納用ディレクトリです。ただし、修正禁止です。
	schemas	DD 用スキーマ定義ファイル格納用ディレクトリです。
	public	J2EE サーバのデフォルトの作業用ディレクトリです。
	repository	一時作業用ディレクトリです。
	sysconf	システム定義用ファイル格納用ディレクトリです。ただし、修正禁止です。
	templates	コンフィグレーションファイルやプロパティファイルなどのひな型格納用ディレクトリです。
usrconf	—	—
	ejb	—
		<ローカルホスト名>
		J2EE サーバのユーザ定義ファイルの格納用ディレクトリです（カスタマイズ用）。
spool	—	一時ディレクトリです。
web	—	—
	redirector	リダイレクタ（互換機能）本体／定義ファイル格納用ディレクトリです。
	logs	リダイレクタ（互換機能）のログ出力用ディレクトリです。

(凡例) — : 該当しません。

• manager ディレクトリ

Component Container の提供する機能のうち、アプリケーションサーバの運用管理機能を提供するディレクトリです。

表 B-2 Component Container のディレクトリ構成 (manager ディレクトリ)

ディレクトリ			説明
bin			運用管理機能で使用するコマンドなどの格納用ディレクトリです。
config	—		運用管理機能で使用するコマンドなどの格納用ディレクトリです。
	templates		運用管理機能で使用する定義ファイルなどの格納用ディレクトリです。
containers	—		—
m	m	—	
		usrconf	
		webapps	
		mngsvr	
		WEB-INF	
		vmxsvr	—
		image	仮想サーバマネージャのヘッダタイトルのロゴ格納用ディレクトリです。
		WEB-INF	URL とサーブレットクラスの対応付けを行うファイルなどの格納用ディレクトリです。
examples	—		
lib	—		
log	—		
modules	—		
spool	—		ログ収集先ディレクトリです。
	mibs		MIB 格納用ディレクトリです。
tmp	—		
apps	—		
externals	—		

ディレクトリ			説明
jp1	jp1	—	JP1 連携用ファイル格納用ディレクトリです。 なお、このディレクトリは Windows 用です。
	ajs3_custom_jobs		カスタムジョブ格納用ディレクトリです。 なお、このディレクトリは Windows 用です。
	mngsvrmonitor		モニタ起動コマンド格納用ディレクトリです。 なお、このディレクトリは Windows 用です。
	ScenarioTemplate		シナリオテンプレート格納用ディレクトリです。
compat3	compat3	—	旧バージョンの Management Server の動作ファイル格納用ディレクトリです。 なお、このディレクトリは Windows, AIX 用です。
	bin		運用管理機能で使用するコマンドなどの格納用ディレクトリです。
	config	—	運用管理機能で使用する定義ファイルなどの格納用ディレクトリです。
		Administrator	ユーザパスワードやユーザ登録情報などの格納用ディレクトリです。
	fwj		業務ごとのプロパティファイル格納用ディレクトリです。
	log		トレースファイル格納用ディレクトリです。
	tmp		一時的なワークディレクトリです。
setup	setup	—	セットアップウィザードディレクトリです。
	bin		—
	config		セットアップウィザード用設定ファイルやセットアップ情報を保持するファイルなどを格納するディレクトリです。
	lib		—
	log		セットアップウィザードのトレースファイル用ディレクトリです。
	tmp		セットアップウィザードの一時的なワークディレクトリです。

ディレクトリ		説明
vmx*	—	08-50 モードの仮想サーバマネージャで使用するファイル格納用ディレクトリです。 また、直下には、仮想サーバマネージャ GUIへのログイン URL のファイルも格納されています。
bin		08-50 モードの仮想サーバマネージャで使用するコマンドなどの格納用ディレクトリです。
config		08-50 モードの仮想サーバマネージャで使用する定義ファイルなどの格納用ディレクトリです。
log		08-50 モードの仮想サーバマネージャのトレースファイル格納用ディレクトリです。
spool		08-50 モードの仮想サーバマネージャの処理データ格納用ディレクトリです。
templates		08-50 モードの仮想サーバマネージャで使用する定義ファイルのテンプレート格納用ディレクトリです。
vmi	—	仮想サーバマネージャで使用するファイル格納用ディレクトリです。
bin		仮想サーバマネージャで使用するコマンドなどの格納用ディレクトリです。
config	—	仮想サーバマネージャで使用する定義ファイルなどの格納用ディレクトリです。
	lb	仮想サーバマネージャで使用する負荷分散機接続設定プロパティファイルの格納用ディレクトリです。
spool		仮想サーバマネージャの処理データ格納用ディレクトリです。
templates		仮想サーバマネージャで使用する定義ファイルのテンプレート格納用ディレクトリです。

(凡例) — : 該当しません。

注※ このディレクトリ配下のファイルは、このバージョンでは使用できません。

- **c4web ディレクトリ**

Component Container の提供する機能のうち、SOAP アプリケーション実行基盤の機能を提供するディレクトリです。

- **jaxrs ディレクトリ**

Component Container の提供する機能のうち, JAX-RS 1.1 仕様に対応した機能を提供するディレクトリです。

- **jaxws ディレクトリ**

Component Container の提供する機能のうち, JAX-WS エンジンの機能を提供するディレクトリです。

- **sinagent ディレクトリ**

Component Container の提供する機能のうち, サーバ通信エージェントの機能を提供するディレクトリです。

付録 C 作業ディレクトリ

作業ディレクトリとは、usrconf.cfg (J2EE サーバ用オプション定義ファイル)、またはusrconf.cfg (バッチサーバ用オプション定義ファイル) のejb.public.directory キーに指定したディレクトリです。

付録 C.1 J2EE サーバの作業ディレクトリ

J2EE サーバは作業ディレクトリ中にサーバ名やアプリケーションの表示名、EAR ファイル中のファイルなどを用いてディレクトリやファイルを生成します。ディレクトリ名やファイル名、パス全体の長さには上限があり、作業ディレクトリ全体がパス全体の上限に収まらないと J2EE サーバは正常に動作できません。ここでは、作業ディレクトリの構成と作業ディレクトリのパス長の見積もり式について説明します。

(1) J2EE サーバの作業ディレクトリの構成

J2EE サーバの作業ディレクトリは、J2EE サーバが動作する上で必要な作業領域です。この領域には、ユーザのパスワード情報、インポートされたリソースアダプタ、EAR ファイル、EJB-JAR ファイル、WAR ファイル、作成された J2EE アプリケーションなどの情報が格納されます。

- J2EE サーバは、作業ディレクトリ以下に J2EE サーバごとの専用ディレクトリ構造を持ちます。このため、複数の J2EE サーバで共通のディレクトリを指定できます。
- 作業ディレクトリを変更して J2EE サーバを開始すると、既存の J2EE アプリケーションなどは利用できません。
- 作業ディレクトリを別のマシンに移動した場合、別のマシン上の J2EE サーバでは認識できません。
- 作業ディレクトリでは、2 バイトコードを含むパス名称を利用できません。
- 作業ディレクトリのデフォルトを次に示します。

- Windows の場合

<Application Server のインストールディレクトリ>¥CC¥server¥public

- UNIX の場合

/opt/Cosminexus/CC/server/public

作業ディレクトリは、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に、ejb.public.directory パラメタで変更できます。

J2EE サーバには、次の作業ディレクトリがあります。

- J2EE アプリケーション (コンフィグレーション関連)
- J2EE アプリケーション (デプロイ関連)
- J2EE リソース
- J2EE リソースアダプタ (コンフィグレーション関連)

- J2EE リソースアダプタ（デプロイ関連）
- Web アプリケーション（JSP 関連）
- Web アプリケーション（その他）

上記の作業ディレクトリの構成についてそれぞれ説明します。また、移行コマンド実行後のディレクトリ構成についても説明します。

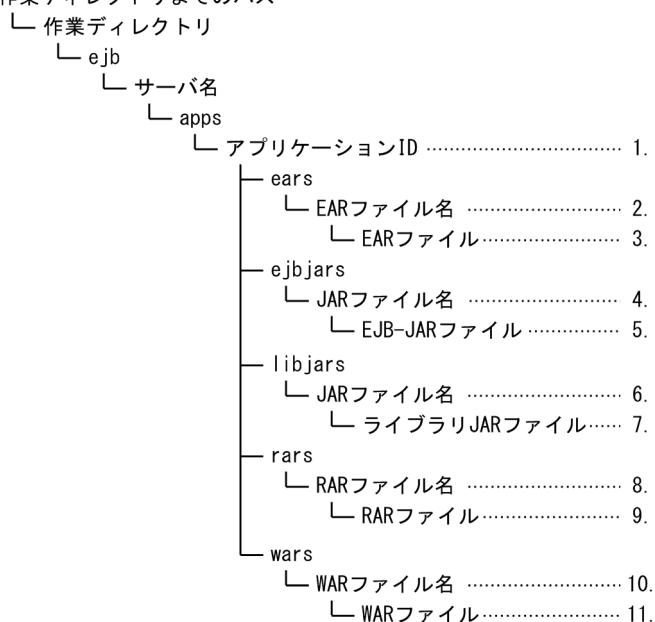
ディレクトリ構成を図で示したあと、各ディレクトリを説明します。なお、図中の番号と説明の番号は対応しています。また、図中では、同一ディレクトリに複数の種類のファイルが入る場合は、いちばん長いファイル名だけを記載しています。

(a) J2EE アプリケーション（コンフィグレーション関連）

J2EE アプリケーション（コンフィグレーション関連）のディレクトリ構成を次の図に示します。

図 C-1 J2EE アプリケーション（コンフィグレーション関連）のディレクトリ構成

作業ディレクトリまでのパス



1. J2EE アプリケーションごとに作られるディレクトリです。

2. EAR ファイルをインポートした場合に作られるディレクトリです。ディレクトリ名は EAR ファイル名を基に付けられます。

3. EAR ファイルは、17 けたの数字.ear の名前でここに格納されます。

4. EJB-JAR ファイルごとに作られるディレクトリです。ディレクトリ名は EJB-JAR ファイル名を基に付けられます。

5. EJB-JAR ファイルは、17 けたの数字.jar の名前でここに格納されます。

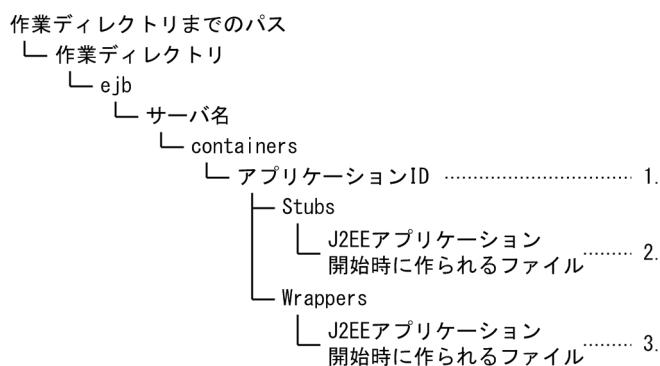
6. ライブラリ JAR ファイルごとに作られるディレクトリです。ディレクトリ名はライブラリ JAR ファイル名を基に付けられます。

- 7.ライブラリ JAR ファイルは、17 けたの数字.jar の名前でここに格納されます。
- 8.RAR ファイルごとに作られるディレクトリです。ディレクトリ名は RAR ファイル名を基に付けられます。
- 9.RAR ファイルは、17 けたの数字.rar の名前でここに格納されます。
- 10.WAR ファイルごとに作られるディレクトリです。ディレクトリ名は WAR ファイル名を基に付けられます。
- 11.WAR ファイルは、17 けたの数字.jar の名前でここに格納されます。

(b) J2EE アプリケーション (デプロイ関連)

J2EE アプリケーション (デプロイ関連) のディレクトリ構成を次の図に示します。

図 C-2 J2EE アプリケーション (デプロイ関連) のディレクトリ構成



- 1.J2EE アプリケーションごとに作られるディレクトリです。
- 2.J2EE アプリケーションの開始時に作られるファイルです。J2EE アプリケーション中にホーム、リモートインターフェース、java.rmi.Remote を実装したインターフェースがあると、それらのパッケージ名やクラス名を基にファイルが作られます。
- 3.J2EE アプリケーションの開始時に作られるファイルです。J2EE アプリケーション中にホーム、リモート、ローカル、ローカルホームインターフェースがあると、それらのパッケージ名やクラス名を基にファイルが作られます。

(c) J2EE リソース

J2EE リソースのディレクトリ構成を次の図に示します。

図 C-3 J2EE リソースのディレクトリ構成

作業ディレクトリまでのパス

```
└─ 作業ディレクトリ
  └─ ejb
    └─ サーバ名
      └─ import
        └─ ejbjars
          └─ JARファイル名 ..... 1.
            └─ EJB-JARファイル ..... 2.

        └─ jdbcjars
          └─ JARファイル名 ..... 3.
            └─ JDBC ドライバファイル ..... 4.

        └─ mails
          └─ JARファイル名 ..... 5.
            └─ hitachi-mail.xml ..... 6.

        └─ rars
          └─ RARファイル名 ..... 7.
            └─ RARファイル ..... 8.

      └─ wars
        └─ WARファイル名 ..... 9.
          └─ WARファイル ..... 10.
```

1. EJB-JAR ファイルごとに作られるディレクトリです。ディレクトリ名は EJB-JAR ファイル名を基に付けられます。
2. EJB-JAR ファイルは、17 けたの数字.jar の名前でここに格納されます。
3. データソースごとに作られるディレクトリです。ディレクトリ名は JDBC ドライバのファイル名を基に付けられます。
4. JDBC ドライバのファイルは、17 けたの数字.jar の名前でここに格納されます。
5. JavaMail の設定ごとに作られるディレクトリです。ディレクトリ名は ejbmail を基に付けられます。
6. JavaMail の設定を保存したファイルがここに保存されます。
7. RAR ファイルごとに作られるディレクトリです。ディレクトリ名は RAR ファイル名を基に付けられます。
8. RAR ファイルは、17 けたの数字.rar の名前でここに格納されます。
9. WAR ファイルごとに作られるディレクトリです。ディレクトリ名は WAR ファイル名を基に付けられます。
10. WAR ファイルは、17 けたの数字.jar の名前でここに格納されます。

(d) J2EE リソースアダプタ (コンフィグレーション関連)

J2EE リソースアダプタ (コンフィグレーション関連) のディレクトリ構成を次の図に示します。

図 C-4 J2EE リソースアダプタ（コンフィグレーション関連）のディレクトリ構成

作業ディレクトリまでのパス

```
└─ 作業ディレクトリ
  └─ ejb
    └─ サーバ名
      └─ rars
        └─ RARファイル名 ..... 1.
          └─ RARファイル ..... 2.
```

1. RAR ファイルごとに作られるディレクトリです。ディレクトリ名は RAR ファイル名を基に付けられます。

2. RAR ファイルは、17 けたの数字.rar の名前でここに格納されます。

(e) J2EE リソースアダプタ（デプロイ関連）のディレクトリ構成

J2EE リソースアダプタ（デプロイ関連）のディレクトリ構成を次の図に示します。なお、図中では、同一ディレクトリに複数の種類のファイルが入る場合は、いちばん長いファイル名だけを記載しています。

図 C-5 J2EE リソースアダプタ（デプロイ関連）のディレクトリ構成

作業ディレクトリまでのパス

```
└─ 作業ディレクトリ
  └─ ejb
    └─ サーバ名
      └─ rarjars
        └─ JARファイル ..... 1.
      └─ nativelibs
        └─ ネイティブラリファイル ..... 2.
```

1. RAR ファイル内の JAR ファイルは、17 けたの数字.jar の名前でここに格納されます。

2. RAR ファイル内のネイティブラリファイルは、ここに格納されます。

(f) Web アプリケーション（JSP 関連）のディレクトリ構成

Web アプリケーション（JSP 関連）のディレクトリ構成を次の図に示します。

図 C-6 Web アプリケーション（JSP 関連）のディレクトリ構成

JSP用の一時ディレクトリ

```
└─ コンテキストルート ..... 1.
  └─ JSPのコンパイル結果 ..... 2.
```

1. コンテキストルートに指定した名前でディレクトリが作られます。/が含まれている場合は 3 文字の文字列に置き換えられます。

2. JSP をコンパイルしてできたクラスファイルが格納されます。JSP のファイル名に英数字以外の文字が含まれている場合は、それぞれ 6 文字の文字列に置き換えられます。また、ファイル名が数字で始まる場合は 1 文字付加されます。

(g) Web アプリケーション（その他）のディレクトリ構成

Web アプリケーション（その他）のディレクトリ構成を次の図に示します。

図 C-7 Web アプリケーション（その他）のディレクトリ構成

作業ディレクトリまでのパス

```
└─ 作業ディレクトリ
    └─ web
        └─ サーバ名
            └─ コンテキストルート ..... 1.
                └─ WAR内のファイル ..... 2.
```

1. コンテキストルートに指定した名前でディレクトリが作られます。/が含まれている場合は3文字の文字列に置き換えられます。

2. WAR ファイルに含まれているファイルがここに解凍されます。

(h) J2EE サーバ、または Web コンテナサーバ（互換機能）の移行コマンド実行後のディレクトリ構成

J2EE サーバ、または Web コンテナサーバの移行コマンド実行時、作業ディレクトリの下に作業ディレクトリ_old の名前でディレクトリを作成し、そこに旧バージョンの作業ディレクトリ下のファイルを退避します。将来、アップグレードインストールをするためには、パス長に作業ディレクトリ_old 分の余裕が必要です。J2EE サーバ、または Web コンテナサーバの移行コマンド実行後のディレクトリ構成を次の図に示します。

図 C-8 J2EE サーバ、または Web コンテナサーバの移行コマンド実行後のディレクトリ構成

作業ディレクトリまでのパス

```
└─ 作業ディレクトリ
    └─ ejb
        └─ サーバ名
    └─ 作業ディレクトリ_old
        └─ ejb
            └─ サーバ名
```

退避

(i) リソースアダプタの移行コマンド実行後のディレクトリ構成

リソースアダプタの移行コマンド実行時、rarupdate_bk の名前でディレクトリを作成し、各 J2EE サーバのリソースアダプタのファイルを退避します。作業ディレクトリが「・・・¥<ディレクトリ名>、または・・・/<ディレクトリ名>」の場合、バックアップ RAR ディレクトリは、「・・・¥<ディレクトリ名>¥rarupdate_bk、または・・・/<ディレクトリ名>/rarupdate_bk」になります。リソースアダプタの移行コマンド実行後のディレクトリ構成を次の図に示します。

図 C-9 リソースアダプタの移行コマンド実行後のディレクトリ構成

バックアップ前

```
作業ディレクトリまでのパス
└─ 作業ディレクトリ
    └─ ejb
        └─ サーバ名
        :
    └─ rarupdate_bk
        └─ ejb
            └─ サーバ名
            :
    └─ rarupdate_bk
        └─ ejb
            └─ サーバ名
            :
```

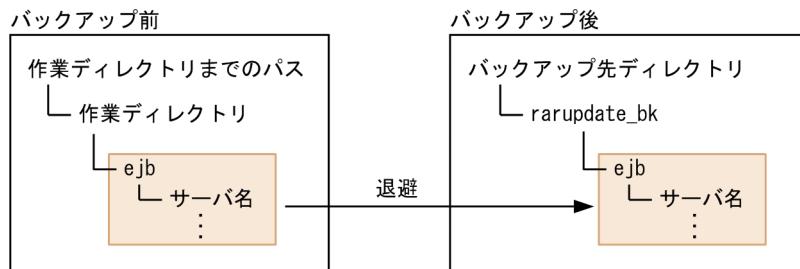
退避

バックアップ後

```
作業ディレクトリまでのパス
└─ 作業ディレクトリ
    └─ rarupdate_bk
        └─ ejb
            └─ サーバ名
            :
    └─ rarupdate_bk
        └─ ejb
            └─ サーバ名
            :
```

作業ディレクトリの下に rarupdate_bk のパス長を追加する余裕がない場合は、移行コマンド実行時に -backupto オプションでバックアップ先ディレクトリを指定し、パス長に余裕がある領域にファイルを退避できます。-backupto オプションを使用する場合のリソースアダプタの移行コマンド実行後のディレクトリ構成を次の図に示します。

図 C-10 リソースアダプタの移行コマンド実行後のディレクトリ構成（-backupto オプションでバックアップ先ディレクトリを指定する場合）



(2) 作業ディレクトリのパス長の見積もり式

作業ディレクトリのパス長の見積もり式は次のようにになります（単位：バイト）。これらの式がすべて満たされるように、各種パラメタを調整してください。なお、見積もり式中の「最大パス長」とは、OS で規定されているパス長の制限のことです。OS のパス長の制限については、ご使用の OS のドキュメントなどで確認してください。なお、Windows の場合、最大パス長は 259 バイトです。

- $A + 2 \times B + C + D + E + 58 \leq \text{最大パス長}$
- $A + 2 \times B + C + E' + 61 \leq \text{最大パス長}$
- $A + 2 \times B + C + E'' + 50 \leq \text{最大パス長}$
- $A + 2 \times B + C + D + (F + 37 \text{ または } M + 68 \text{ のどちらか大きい方の値}) + 23 \leq \text{最大パス長}$ ($F \leq 35$ の場合は 35 で換算)
- $A + 2 \times B + C + G + 22 \leq \text{最大パス長}$ ($G \leq 18$ の場合は 18 で換算)
- $A + B + C + H + I + 8 \leq \text{最大パス長}$
- $A + 2 \times B + C + J + 34 \leq \text{最大パス長}$
- $J + K + 18 \leq \text{最大パス長}$
- $A + 2 \times B + C + L + 43 \leq \text{最大パス長}$ (Reliable Messaging 使用時)
- $K + L + 27 \leq \text{最大パス長}$ (Reliable Messaging 使用時)
- $A + 2 \times B + C + H + L + I + N + 22 \leq \text{最大パス長}$
- $A + 2 \times B + C + H + L + O + N + 39 \leq \text{最大パス長}$
- $P + H + I + N + 3 \leq \text{最大パス長}$
- $O + H + M + N + 20 \leq \text{最大パス長}$

(凡例)

- A : 作業ディレクトリまでのパスの長さ
- B : 作業ディレクトリ名の長さ
- C : 最も長いサーバ名の長さ
- D : 最も長い J2EE アプリケーション名の長さ
- E : インポートしたときの EJB-JAR ファイル, WAR ファイル, RAR ファイル, ライブラリ JAR, EAR のファイル名のうち, 最も長いファイル名 (EAR ファイルに含まれているファイルも含む) の長さ
- E' : インポートしたときの EJB-JAR ファイル, WAR ファイル, RAR ファイル, JDBC ドライバ, メールのファイル名のうち, 最も長いファイル名 (EAR ファイルに含まれているファイルも含む) の長さ (メールは 7 で換算)
- E'' : インポートしたときの RAR ファイル名のうち, 最も長いファイル名の長さ
- F : J2EE アプリケーションに含まれる EJB のインターフェースクラス, java.rmi.Remote を実装したインターフェースやクラス, EJB のコンポーネントクラスのうち, 最も長いファイル名 (パッケージ名も含む) の長さ
- G : ネイティブライブラリファイルのうち, 最も長いファイル名の長さ
- H : 最も長いコンテキストルートの長さ (/を含む場合は/ $\times 3$ で換算)
- I : WAR ファイル内で最も長いパスの長さ
- J : 最も長いリソースアダプタ表示名の長さ
- K : ログファイル出力先ディレクトリのパスの長さ (簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に, ejb.server.log.directory パラメタで指定する値)
- L : 最も長い Reliable Messaging の表示名の長さ
- M : J2EE アプリケーションに含まれる EJB の Bean クラスのうち, 最も長いクラス名 (パッケージ名を含まない) の長さ
- N : 最も長い JSP/タグファイルコンパイル結果のクラス名の長さ
- O : 最も長いタグファイルのパッケージ名の長さ
- P : JSP 用テンポラリディレクトリまでのパスの長さ

付録 C.2 バッチサーバの作業ディレクトリ

バッチサーバは作業ディレクトリ中にサーバ名などを用いてディレクトリやファイルを生成します。ディレクトリ名やファイル名, パス全体の長さには上限があり, 作業ディレクトリ全体がパス全体の上限に収まらないとバッチサーバは正常に動作できません。ここでは, 作業ディレクトリの構成と作業ディレクトリのパス長の見積もり式について説明します。

(1) バッチサーバの作業ディレクトリの構成

バッチサーバの作業ディレクトリは, バッチサーバが動作する上で必要な作業領域です。この領域には, ユーザのパスワード情報, インポートされたリソースアダプタなどの情報が格納されます。

- ・バッチサーバは、作業ディレクトリ以下にバッチサーバごとの専用ディレクトリ構造を持ちます。このため、複数のバッチサーバで共通のディレクトリを指定できます。
- ・作業ディレクトリを別のマシンに移動した場合、別のマシン上のバッチサーバでは認識できません。
- ・作業ディレクトリでは、2バイトコードを含むパス名称を利用できません。
- ・作業ディレクトリのデフォルトを次に示します。

- Windows の場合

<Application Server のインストールディレクトリ>¥CC¥server¥public

- UNIX の場合

/opt/Cosminexus/CC/server/public

作業ディレクトリは、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に、ejb.public.directory パラメタで変更できます。

バッチサーバには、次の作業ディレクトリがあります。

- 1.J2EE リソース
- 2.J2EE リソースアダプタ (コンフィグレーション関連)
- 3.J2EE リソースアダプタ (デプロイ関連)

なお、2.および3.はJ2EE サーバの場合と同じです。詳細は、「[付録 C.1\(1\)\(d\) J2EE リソースアダプタ \(コンフィグレーション関連\)](#)」および「[付録 C.1\(1\)\(e\) J2EE リソースアダプタ \(デプロイ関連\)](#)」を参照してください。

ここでは、J2EE リソースの作業ディレクトリの構成について説明します。また、移行コマンド実行後のディレクトリ構成についても説明します。

(a) J2EE リソース

J2EE リソースのディレクトリ構成を図で示したあと、各ディレクトリを説明します。なお、図中の番号と説明の番号は対応しています。また、図中では、同一ディレクトリに複数の種類のファイルが入る場合は、いちばん長いファイル名だけを記載しています。

図 C-11 J2EE リソースのディレクトリ構成

作業ディレクトリまでのパス

```

└─ 作業ディレクトリ
  └─ ejb
    └─ サーバ名
      └─ import
        └─ rars
          └─ RARファイル名 ..... 1.
            └─ RARファイル..... 2.
  
```

1.RAR ファイルごとに作られるディレクトリです。ディレクトリ名は RAR ファイル名を基に付けられます。

2.RAR ファイルは、17 けたの数字.rar の名前でここに格納されます。

(b) バッチサーバの移行コマンド実行後のディレクトリ構成

バッチサーバの移行コマンド実行時、作業ディレクトリの下に作業ディレクトリ_old の名前でディレクトリを作成し、そこに旧バージョンの作業ディレクトリ下のファイルを退避します。バッチサーバの移行コマンド実行後のディレクトリ構成は、J2EE サーバの場合と同じです。詳細は、「[付録 C.1\(l\)\(h\) J2EE サーバ、または Web コンテナサーバ（互換機能）の移行コマンド実行後のディレクトリ構成](#)」を参照してください。この際、「J2EE サーバ」を「バッチサーバ」に置き換えてお読みください。

(c) リソースアダプタの移行コマンド実行後のディレクトリ構成

リソースアダプタの移行コマンド実行時、rarupdate_bk の名前でディレクトリを作成し、各バッチサーバのリソースアダプタのファイルを退避します。リソースアダプタの移行コマンド実行後のディレクトリ構成については、「[付録 C.1\(l\)\(i\) リソースアダプタの移行コマンド実行後のディレクトリ構成](#)」を参照してください。

(2) 作業ディレクトリのパス長の見積もり式

作業ディレクトリのパス長の見積もり式は次のようにになります（単位：バイト）。これらの式がすべて満たされるように、各種パラメタを調整してください。なお、見積もり式中の「最大パス長」とは、OS で規定されているパス長の制限のことです。OS のパス長の制限については、ご使用の OS のドキュメントなどで確認してください。なお、Windows の場合、最大パス長は 259 バイトです。

- $A + 2 \times B + C + D + 61 \leq \text{最大パス長}$
- $A + 2 \times B + C + E + 34 \leq \text{最大パス長}$
- $E + F + 18 \leq \text{最大パス長}$

(凡例)

A : 作業ディレクトリまでのパスの長さ

B : 作業ディレクトリ名の長さ

C : 最も長いサーバ名の長さ

D : インポートしたときの RAR ファイル名のうち、最も長いファイル名の長さ

E : 最も長いリソースアダプタ表示名の長さ

F : ログファイル出力先ディレクトリのパスの長さ（簡易構築定義ファイルの論理 J2EE サーバ（j2ee-server）の<configuration>タグ内に、ejb.server.log.directory パラメタで指定する値）

付録 C.3 作業ディレクトリに関する注意事項

作業ディレクトリの運用では、次の点に注意してください。

- J2EE アプリケーションやクライアントアプリケーションで、作業ディレクトリ以下の製品が使用するディレクトリおよびファイルを使用しないでください。また、作業ディレクトリ以下のディレクトリおよびファイル構成を変更しないでください。

J2EE サーバまたはバッチサーバを起動した場合に、例外が発生することがあります。

- J2EE サーバまたはバッチサーバのカレントディレクトリ※には、J2EE アプリケーションで独自ファイルの入出力はできます。ただし、製品で使用するディレクトリ、ファイルと名称が同じにならないようしてください。

注※ J2EE サーバのカレントディレクトリは「<作業ディレクトリ>/ejb/<J2EE サーバ名>」、バッチサーバのカレントディレクトリは「<作業ディレクトリ>/ejb/<バッチサーバ名>」です。

付録 D 運用管理ポータルの操作で失敗したときの対処方法

運用管理ポータルの操作で失敗を示すメッセージやログが画面に表示された場合は、メッセージを確認して障害要因を取り除いてから、操作を再実行する必要があります。運用管理ポータルの操作で失敗したときの対処手順を次に示します。

1. [ログの表示] タブをクリックして、[ログの表示] 画面に出力されるエラーメッセージの時刻を確認します。

2. 次のログファイルを参照して、手順 1.で確認した時刻に対応するエラーメッセージの内容を確認して障害要因を特定します。

確認対象のログファイルを次の表に示します。なお、ログファイル内のメッセージは、時刻の古い順に出力されます。メッセージについては、マニュアル「アプリケーションサーバメッセージ(構築／運用／開発用)」を参照してください。

表 D-1 確認対象のログファイル

分類	確認対象	ファイル名
サーバ管理コマンド	稼働ログ	<ul style="list-style-type: none">Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥cjmessage[n].logUNIX の場合 /opt/Cosminexus/CC/admin/logs/cjmessage[n].log
	障害発生時の例外情報	<ul style="list-style-type: none">Windows の場合 <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥cjexception[n].logUNIX の場合 /opt/Cosminexus/CC/admin/logs/cjexception[n].log
J2EE サーバ	稼働ログ	<ul style="list-style-type: none">Windows の場合 <ejb.server.log.directory>※1¥cjmessage[n].logUNIX の場合 <ejb.server.log.directory>※1/cjmessage[n].log
	障害発生時の例外情報	<ul style="list-style-type: none">Windows の場合 <ejb.server.log.directory>※1¥cjexception[n].logUNIX の場合 <ejb.server.log.directory>※1/cjexception[n].log
	Web サーブレットログ	<ul style="list-style-type: none">Windows の場合 <ejb.server.log.directory>※1¥web_servlet[n].logUNIX の場合 <ejb.server.log.directory>※1/web_servlet[n].log
	ユーザ出力ログ	<ul style="list-style-type: none">Windows の場合 <ejb.server.log.directory>※1¥user_out[n].log

分類	確認対象	ファイル名
		<ul style="list-style-type: none"> UNIX の場合 <ejb.server.log.directory>※1/user_out[n].log
	ユーザエラーログ	<ul style="list-style-type: none"> Windows の場合 <ejb.server.log.directory>※1\\$user_err[n].log UNIX の場合 <ejb.server.log.directory>※1/user_err[n].log
	JavaVM の保守情報および GC のログ	<ul style="list-style-type: none"> Windows の場合 <ejb.server.log.directory>※1\\$javalog[nn].log UNIX の場合 <ejb.server.log.directory>※1/javalog[nn].log
	エラーリポートファイル (Windows の場合) ※2	<作業ディレクトリ>\ejb\\$<サーバ名>\hs_err_pid<サーバプロセスのプロセス ID>.log

注※1 <ejb.server.log.directory>は、簡易構築定義ファイルの論理J2EE サーバ (j2ee-server) の<configuration>タグ内に、ejb.server.log.directory パラメタで指定したディレクトリを指します。デフォルト値は、<Application Server のインストールディレクトリ>\CC\\$server\\$public\\$ejb\\$<サーバ名称>\logs です。

注※2 保守員に送付する資料です。

3. 手順 2.で特定した障害要因を取り除いて、障害を回復します。

付録 E Management Server が自動で設定する内容

Management Server の機能を使用して、論理サーバのコンフィグレーション定義または環境設定で設定した内容は、自動的に CTM デーモンの起動コマンドのオプション、またはユーザ定義ファイルに反映されます。

付録 E.1 Management Server が自動で生成するユーザ定義ファイル

Management Server の機能を使用して、論理サーバのコンフィグレーション定義または環境設定で設定した内容から生成されるユーザ定義ファイルについて、J2EE アプリケーションを実行するシステムの場合とバッチアプリケーションを実行するシステムの場合に分けて説明します。なお、ユーザ定義ファイルは、システムを構築する対象のホストに生成されます。

注意事項

システムの一貫性を保つため、細かい設定を変更する場合でも、構築時に使用した Management Server の機能 (Smart Composer 機能、運用管理ポータルなど) から実行するようにしてください。テキストエディタなどで個別にファイルを直接編集する場合には、スペルミスなどに十分注意してください。

• J2EE アプリケーションを実行するシステムの場合

論理サーバのコンフィグレーション定義または環境設定で設定した内容から生成されるユーザ定義ファイルを次の表に示します。

表 E-1 論理サーバのコンフィグレーション定義または環境設定で設定した内容から生成されるユーザ定義ファイル (J2EE アプリケーションを実行するシステムの場合)

論理サーバ	生成されるユーザ定義ファイル名	ファイルの概要	ファイルの格納先
J2EE サーバ	auto_explicit_memory.cfg	明示管理ヒープ機能の自動配置機能で使用する自動配置設定ファイル。	<ul style="list-style-type: none">Windows の場合 <Application Server のインストールディレクトリ>¥CC¥server¥usrconf¥ejb¥<サーバ名>/UNIX の場合 /opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名>/
	userprf.cfg	ユーザ拡張性能解析トレースで使用するユーザ拡張性能解析トレース設定ファイル。	
	usrconf.cfg	J2EE サーバの起動オプションが定義されたファイル。 このファイルで JavaVM の起動オプションも定義します。	<ul style="list-style-type: none">Windows の場合 <Application Server のインストールディレクトリ>¥CC¥server¥usrconf¥ejb¥<サーバ名>¥

論理 サーバ	生成されるユーザ定義ファ イル名	ファイルの概要	ファイルの格納先
	usrconf.properties	J2EE サーバを実行するために必要な JavaVM のシステムプロパティを指定するファイル。	<ul style="list-style-type: none"> UNIX の場合 /opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名>/
	mngagent.<実サーバ名>.properties	J2EE サーバで運用監視するために必要な運用監視エージェントを設定するファイル。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥config¥
	mevent.<実サーバ名>.properties	J2EE サーバで Management イベントを発行するために必要なプロパティを指定するファイル。	<ul style="list-style-type: none"> UNIX の場合 /opt/Cosminexus/manager/config/
	manager.<実サーバ名>.properties	J2EE サーバで JP1 イベントを発行するために必要なプロパティを指定するファイル。	
Web サー バ	httpsd.conf	Web サーバを稼働させるために必要な情報が定義されているファイル。	<ul style="list-style-type: none"> Windows の場合 <HTTP Server のインストールディレクトリ>¥servers¥HWS_<サーバ名>¥conf¥または<HTTP Server のインストールディレクトリ>¥conf¥ UNIX の場合 /opt/hitachi/httpsd/servers/HWS_<サーバ名>/conf/または/opt/hitachi/httpsd/conf/
	mod_jk.conf	V9 互換モードの J2EE サーバと連携する場合に、HTTP Server 用のリダイレクタ定義（ログレベルなど）や URL マッピングを定義するファイル。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥web¥redirector¥servers¥<サーバ名>¥または<Application Server のインストールディレクトリ>¥CC¥web¥redirector¥
	workers.properties	V9 互換モードの J2EE サーバと連携する場合に、ワーカの定義およびワーカごとの各パラメタを定義するファイル。	<ul style="list-style-type: none"> UNIX の場合 /opt/Cosminexus/CC/web/redirector/servers/<サーバ名>/または/opt/Cosminexus/CC/web/redirector/

- バッチアプリケーションを実行するシステムの場合

論理サーバのコンフィグレーション定義または環境設定で設定した内容から生成されるユーザ定義ファイルを次の表に示します。

表 E-2 論理サーバのコンフィグレーション定義または環境設定で設定した内容から生成されるユーザ定義ファイル（バッチアプリケーションを実行するシステムの場合）

論理サーバ	生成されるユーザ定義ファイル名	ファイルの概要	ファイルの格納先
バッチサーバ	auto_explicit_memory.cfg	明示管理ヒープ機能の自動配置機能で使用する自動配置設定ファイル。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>/CC/server/usrconf/ejb/<サーバ名>/ UNIX の場合 /opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名>/
	userprf.cfg	ユーザ拡張性能解析トレースで使用するユーザ拡張性能解析トレース設定ファイル。	
	usrconf.cfg	バッチサーバの起動オプションが定義されたファイル。 このファイルで JavaVM の起動オプションも定義します。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥CC¥server¥usrconf¥ejb¥<サーバ名>¥ UNIX の場合 /opt/Cosminexus/CC/server/usrconf/ejb/<サーバ名>/
	usrconf.properties	バッチサーバを実行するために必要な JavaVM のシステムプロパティを指定するファイル。	
	mngagent.<実サーバ名>.properties	バッチサーバで運用監視するために必要な運用監視エージェントを設定するファイル。	<ul style="list-style-type: none"> Windows の場合 <Application Server のインストールディレクトリ>¥manager¥config¥ UNIX の場合 /opt/Cosminexus/manager/config/
	mevent.<実サーバ名>.properties	バッチサーバで Management イベントを発行するために必要なプロパティを指定するファイル。	
	manager.<実サーバ名>.properties	バッチサーバで JP1 イベントを発行するために必要なプロパティを指定するファイル。	

付録 E.2 Management Server が CTM ドメインマネージャに自動で設定する内容

CTM デーモンの起動コマンド (ctmdmstart) の、CTM ドメインマネージャの設定に関するオプションに、Management Server が自動的に設定する内容を次の表に示します。

表 E-3 Management Server が自動的に設定する内容 (CTM ドメインマネージャの場合)

起動コマンド	オプション
ctmdmstart	-CTMMYHost <「論理サーバの構成定義」で CTM ドメインマネージャを定義したホストの IP アドレス>

起動コマンド	オプション
	-CTMAutoForce
	-Dvbroker.orb.isNTService=true [※]

注※

Windows の場合に設定されます。

付録 E.3 Management Server が CTM に自動で設定する内容

CTM デーモンの起動コマンド (ctmstart) の、 CTM の設定に関するオプションに、 Management Server が自動的に設定する内容を次の表に示します。

表 E-4 Management Server が自動的に設定する内容 (CTM の場合)

起動コマンド	オプション
ctmstart	-CTMMyHost <「論理サーバの構成定義」で CTM を定義したホストの IP アドレス>
	-CTMClientConnectCount<CTM レギュレータプロセス数 + OTM ゲートウェイプロセス数> ^{※1}
	-Dvbroker.orb.isNTService= true ^{※2}

注※1

CTM レギュレータプロセス数と OTM ゲートウェイプロセス数の合計が 256 未満の場合には、 -CTMClientConnectCount オプションの設定値は 256 になります。

注※2

Windows の場合に設定されます。

付録 E.4 Management Server が J2EE サーバに自動で設定する内容

J2EE サーバに関連するユーザ定義ファイルに、 Management Server が自動的に設定する内容を次の表に示します。

表 E-5 Management Server が自動的に設定する内容 (J2EE サーバの場合)

ユーザ定義ファイル	設定される内容
usrconf.properties	ejbserver.manager.agent.Agent.conf=<環境設定で作成される mngagent.<実サーバ名>.properties ファイルの位置>
	ejbserver.manager.agent.Agent.enabled=true
	ejbserver.ctm.CTMMYHost=<「運用管理ドメインの構成定義」で定義したホスト名>
	manager.j2ee.compat=<J2EE サーバの互換モードの使用有無>

ユーザ定義ファイル	設定される内容
usrconf.cfg	add.jvm.arg=-XX:HitachiAutoExplicitMemoryFile:<Application Server のインストールディレクトリ>/CC/server/usrconf/ejb/<実サーバ名>/auto_explicit_memory.cfg ^{*1}
	add.jvm.arg=-Djvm.userprf.File=<Application Server のインストールディレクトリ>/CC/server/usrconf/ejb/<実サーバ名>/userprf.cfg ^{*2}
mngagent.<実サーバ名>.properties	mngagent.domain_name=<運用管理ドメイン名（内部名）> mngagent.agent_name=<該当ノードの論理サーバ名>
mevent.<実サーバ名>.properties	manager.mevent.send.host=<Management Server の IP アドレス> manager.mevent.send.port=<Management Server の HTTP ポート番号> manager.mevent.logical_server_name=<論理サーバ名>
manager.<実サーバ名>.properties	manager.jp1event.logical_server_name=<論理サーバ名> manager.jp1event.domain_name=<運用管理ドメイン名>

注※1 運用管理ポータルまたは Smart Composer 機能で、明示管理ヒープ機能の自動配置機能で使用する自動配置設定ファイルの内容を指定している場合に設定します。

注※2 運用管理ポータルまたは Smart Composer 機能で、ユーザ拡張性能解析トレースのユーザ拡張性能解析トレース設定ファイルの内容を指定している場合に設定します。

付録 E.5 Management Server がバッチサーバに自動で設定する内容

バッチサーバに関連するユーザ定義ファイルに、Management Server が自動的に設定する内容を次の表に示します。

表 E-6 Management Server が自動的に設定する内容（バッチサーバの場合）

ユーザ定義ファイル	設定される内容
usrconf.properties	ejbserver.manager.agent.Agent.conf=<環境設定で作成される mngagent.<実サーバ名>.properties ファイルの位置>
	ejbserver.manager.agent.Agent.enabled=true
	ejbserver.ctm.CTMMyHost=<「運用管理ドメインの構成定義」で定義したホスト名>
mngagent.<実サーバ名>.properties	mngagent.domain_name=<運用管理ドメイン名（内部名）>
	mngagent.agent_name=<該当ノードの論理サーバ名>
mevent.<実サーバ名>.properties	manager.mevent.send.host=<Management Server の IP アドレス>
	manager.mevent.send.port=<Management Server の HTTP ポート番号>
	manager.mevent.logical_server_name=<論理サーバ名>

ユーザ定義ファイル	設定される内容
manager.<実サーバ名>.properties	manager.jplevent.logical_server_name=<論理サーバ名>
	manager.jplevent.domain_name=<運用管理ドメイン名>

付録 E.6 Management Server が Web サーバに自動で設定する内容

Web サーバに関連するユーザ定義ファイルに、Management Server が自動的に設定する内容を次の表に示します。

表 E-7 Management Server が自動的に設定する内容 (Web サーバの場合)

ユーザ定義ファイル	設定される内容	
httpsd.conf ^{※1}	Windows の場合	Include <Application Server のインストールディレクトリ>/CC/web/redirector/servers/<論理サーバ名>/mod_jk.conf ^{※4}
		Include <HTTP Server のインストールディレクトリ>/servers/HWS_<論理 Web サーバの実サーバ名>/conf/httpsd_manager.conf ^{※2}
		ThreadLimit <ThreadsPerChild と同じ値>
		PidFile <HTTP Server のインストールディレクトリ>/servers/HWS_<論理 Web サーバの実サーバ名>/logs/httpd.pid
	UNIX の場合	Include <Application Server のインストールディレクトリ>/CC/web/redirector/servers/<論理サーバ名>/mod_jk.conf ^{※4}
		Include <HTTP Server のインストールディレクトリ>/servers/HWS_<論理 Web サーバの実サーバ名>/conf/httpsd_manager.conf ^{※2}
		ThreadLimit <ThreadsPerChild と同じ値> ^{※5}
		ServerLimit <MaxClients の値>÷<ThreadsPerChild の値> ^{※5}
mod_jk.conf ^{※4}	Windows の場合	MinSpareServers 5 ^{※6}
		PidFile <HTTP Server のインストールディレクトリ>/servers/HWS_<論理 Web サーバの実サーバ名>/logs/httpd.pid
	UNIX の場合	LoadModule jk_module <Application Server のインストールディレクトリ>/CC/web/redirector/<mod_jk ファイル名> ^{※3}
		JkWorkersFile <Application Server のインストールディレクトリ>/CC/web/redirector/servers/<論理サーバ名>/workers.properties
	UNIX の場合	LoadModule jk_module /opt/Cosminexus/CC/web/redirector/<mod_jk ファイル名> ^{※3}
		JkWorkersFile /opt/Cosminexus/CC/web/redirector/servers/<論理サーバ名>/workers.properties

注※1

Web サーバの運用時には、httpsd.conf ファイルの内容を運用環境に合わせて適切に設定する必要があります。設定する内容や方法については、マニュアル「HTTP Server」を参照してください。

注※2

簡易構築定義ファイルで論理 Web サーバ (web-server) の AppendDirectives パラメタ、または運用管理ポータルの [Web サーバの設定] 画面で「項目ごとに設定します。」の「追加ディレクティブ」に値を指定した場合に設定されます。

注※3

<mod_jk ファイル名>は、OS によって異なります。

注※4

V9 互換モードの J2EE サーバと連携する場合、または V11 以降でリダイレクタを使用する場合に設定されます。

注※5

J2EE サーバとの連携方法がリバースプロキシの場合だけ設定されます。値が小数になる場合は、小数点以下が切り捨てされ、整数値が出力されます。なお、小数点以下を切り捨てたときの値が 0 のときには 1 が設定されます。

注※6

J2EE サーバとの連携方法がリダイレクタの場合、かつ httpsd.conf に MinSpareServers が設定されていない場合だけ設定されます。

付録 F Application Server の更新インストール

Application Server を更新インストールする手順について説明します。更新インストールとは、Application Server がインストール済みの環境に、Application Server（同一製品）をインストールすることです。

更新インストールでは、インストールする製品とインストール済みの製品のバージョンによって、更新処理のしかたが異なります。製品が同一バージョン同士であれば上書きされ、製品が異なるバージョン同士（例えば、09-00 をインストール済みの環境に 11-00 をインストールする場合など）であればバージョンアップされます。

また、Component Container がインストールされている場合、Component Container の更新インストール時に、インストーラによってユーザ環境の移行が実行されます。更新インストールでのユーザ環境の扱いについては、「[2.2.1\(2\) Component Container インストール時のユーザ環境の扱い](#)」を参照してください。

付録 F.1 Application Server を更新インストールする（Windows の場合）

更新インストールでは、製品の提供媒体であるインストーラを使用します。

更新インストールの手順を次に示します。インストール作業には Administrator 権限が必要です。更新インストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を、更新インストール時のエラーコードについては、「[付録 I.6 Application Server のインストールおよびアンインストール時のエラーコード一覧（Windows の場合）](#)」を参照してください。なお、インストーラの使用方法については、製品の提供媒体のドキュメントを参照してください。

手順

1. 構成ソフトウェアのすべてのプロセスが実行中でないことを確認します。実行中のプロセスがあれば停止します。

2. 製品の提供媒体を CD-ROM ドライブにセットします。

[日立総合インストーラ] ダイアログに、「選択されたソフトウェアをインストールします。」と表示されます。

[日立総合インストーラ] ダイアログが表示されない場合、エクスプローラを使用して、CD-ROM ディレクトリの「HCD_INST.EXE」をダブルクリックしてください。

3. Application Server を選択した状態で、[インストール実行] ボタンをクリックします。

[機能の選択] ダイアログが表示されます。

4. [標準 - このセットアップを推奨します。すべてのプログラム機能を再インストールします。] の左にあるボタンをクリックします。

[インストールの開始] ダイアログが表示されます。

5. 設定した内容を確認して、問題がなければ [次へ] ボタンをクリックします。

インストールが開始されます。インストールが完了すると、[セットアップの完了] ダイアログが表示されます。

6. [完了] ボタンをクリックします。

OS を再起動するかどうかを確認する画面が表示されます。

7. [はい] ボタンをクリックします。

OS が再起動し、Application Server の更新インストールが完了します。

更新インストール後の作業については、マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「10.2.2 更新インストールの場合」を参照してください。

付録 F.2 Application Server を更新インストールする (UNIX の場合)

更新インストールでは、製品の提供媒体である PP インストーラを使用します。

更新インストールの手順を次に示します。インストール作業には root 権限が必要です。更新インストールするときの注意事項については、「[付録 I.1 Application Server をインストールおよびアンインストールするときの注意事項](#)」を参照してください。

なお、更新インストールの操作は、Component Container 管理者の設定の有無によって異なります。詳細は、「[4.1.4 Component Container 管理者を設定するときの注意事項 \(UNIX の場合\)](#)」を参照してください。

手順

1. アプリケーションサーバに root 権限（スーパーユーザ）でログインします。

2. 構成ソフトウェアのすべてのプロセスが実行中でないことを確認します。実行中のプロセスがあれば停止します。

3. 製品の提供媒体を CD-ROM ドライブにセットします。

4. 製品の提供媒体が CD-ROM の場合は、CD-ROM ファイルシステムをマウントします。

コマンドの実行例を次に示します。下線部には、デバイススペシャルファイル名、および CD-ROM ファイルシステムのマウントディレクトリ名を指定します。なお、これらの名称は、OS、ハードウェア、および環境によって異なります。

(AIX の実行例)

```
mount -r -v cdrfs /dev/cd0 /cdrom
```

(Linux の実行例)

```
mount -r -o mode=0544 /dev/cdrom /mnt/cdrom
```

5. セットアッププログラムを起動します。

コマンドの実行例を次に示します。下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。Linux の場合、Reliable Messaging および Web Services - Security をインストールする PP インストーラと、それ以外の構成ソフトウェアと製品の共通モジュールをインストールする PP インストーラが異なります。

(AIX の実行例)

```
/cdrom/aix/setup /cdrom
```

(Linux の実行例)

```
/mnt/cdrom/x64lin/setup /mnt/cdrom
```

CD-ROM セットアッププログラムによって、PP インストーラと常駐プロセス自動起動プログラムがハードディスク上にインストールされ、PP インストーラが自動的に起動されます。

注意事項

CD-ROM のディレクトリ名やファイル名は、マシン環境によって記述した内容と見え方が異なることがあります。ls コマンドで確認し、表示されたファイル名をそのまま入力してください。

6. PP インストーラのメインメニューで、[I] キーを押します。

PP インストール画面が表示されます。

7. プログラムにカーソルを移動させ、[スペース] キーを押します。

すべてのプログラム（Application Server に対応している構成ソフトウェア、および製品の共通モジュール）を選択します。選択したプログラムの左側には<@>が表示されます。なお、プログラムを選択してインストールすることもできます。

注意事項

製品に対応している構成ソフトウェアについては、「[2.2.1\(3\) 製品に対応する構成ソフトウェアについて](#)」を参照してください。製品の共通モジュールは、製品によって表示されるプログラム名が異なります。

また、JDK11 から JDK17 以降への更新インストールはサポートしていません。

詳細は「[2.2.1 Application Server のインストールについて](#)」を参照してください。

8. すべてのプログラムの左側に<@>が表示されていることを確認して、[I] キーを押します。

画面の最下行に「Install PP? (y: install, n: cancel)==>」メッセージが表示されます。

9. [y] キーまたは [Y] キーを押します。

インストールが開始されます。

[n] キーまたは [N] キーを押すと、インストールが中止されて PP インストール画面に戻ります。

10. インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

PP インストーラのメインメニューに戻ります。

11. PP インストーラのメインメニューで、[Q] キーを押します。

Application Server の更新インストールが完了します。

更新インストール後の作業については、マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「10.2.2 更新インストールの場合」を参照してください。

付録 G JP1/SC/DPM を使ったディスク複製インストール

JP1/SC/DPM を使ったディスク複製インストールを利用する場合の手順について説明します。

JP1/SC/DPM を使ったディスク複製インストールとは、JP1/SC/DPM を使用して作成したディスクイメージの複製をインストールすることです。JP1/SC/DPM を使ったディスク複製インストールを利用するかどうかは、アプリケーションサーバおよびBPM/ESB 基盤の製品によって異なります。

注意事項

AIX の場合、Application Server および Service Platform では、JP1/SC/DPM を使ったディスク複製インストールに対応していません。ディスク複製インストールを利用する手順を次に示します。

手順

1. インストール済みの製品 (Application Server または Service Platform) をアンインストールします。

製品のアンインストール手順については、「[3.3.3 Application Server をアンインストールする \(AIX の場合\)](#)」を参照してください。

2. ディスク複製インストールを実施します。

ディスク複製インストールの操作・運用方法については、JP1/ServerConductor/Deployment Manager のマニュアルを参照してください。

3. 製品 (Application Server または Service Platform) をインストールします。

製品のインストール手順については、「[2.2.3 Application Server を新規インストールする \(AIX の場合\)](#)」を参照してください。

付録 G.1 Application Server をディスク複製インストールする (Windows の場合)

アプリケーションサーバおよびBPM/ESB 基盤の製品ごとに、ディスク複製インストールを利用する場合の手順について説明します。

(1) Application Server および Service Platform の場合

Application Server および Service Platform は、JP1/SC/DPM を使ったディスク複製インストールに対応しています。JP1/SC/DPM のディスク複製インストール機能、および仮想化環境が提供するイメージファイル化による複製機能が使用できます。

なお、次の構成ソフトウェアには、JP1/SC/DPM を使ったディスク複製インストールの利用に前提条件があるので、利用前に確認してください。

- Component Container (J2EE アプリケーションを実行するシステム (J2EE サーバ) 使用時)
EJB クライアントアプリケーションが稼働するマシンは、ディスク複製インストールに対応していません。
- TPBroker
JP1/SC/DPM との連携方法については、マニュアル「TPBroker 運用ガイド」を参照してください。

ディスク複製インストールを利用する手順を次に示します。

手順

1. ディスク複製インストール元のマシンでアプリケーションサーバを構築します。

製品をインストールしたディスク複製インストール元のマシンに、システム環境をセットアップしたあと、各種設定をカスタマイズしたり、業務アプリケーションを設定したりします。

2. Application Server および Service Platform の構成ソフトウェアごとに、ディスク複製インストールを実施する前に必要な作業を実施します。

実施する作業については、「[付録 I.3 ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項 \(Windows の場合\)](#)」の構成ソフトウェアごとの「複製を実施する前にすること」を参照してください。

3. JP1/SC/DPM を使用したディスク複製インストールを実施します。

JP1/SC/DPM を使ったディスク複製インストールの操作・運用方法については、JP1/ServerConductor/Deployment Manager のマニュアルを参照してください。仮想化環境が提供するイメージファイル化による複製機能については、仮想化ソフトウェアのドキュメントを参照してください。

4. Application Server および Service Platform の構成ソフトウェアごとに、ディスク複製インストールを実施したあとに必要な作業を実施します。

実施する作業については、「[付録 I.3 ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項 \(Windows の場合\)](#)」の構成ソフトウェアごとの「複製を実施したあとにすること」を参照してください。

5. ディスク複製インストール先のマシンに接続するマシンでシステムの要件に合わせて各種設定をカスタマイズします。

各種設定には、例えば、次のようなものがあります。システムの要件に合わせて変更してください。

- システム構成上でフロントエンドに配置される負荷分散機の設定
- DBMS に関する設定
- システム構成上でバックエンドに配置される J2EE サーバ、または Web コンテナサーバに関する設定

(2) Developer, Service Architect, および Client の場合

Developer, Service Architect および Client は, JP1/SC/DPM を使ったディスク複製インストールに対応していません。ディスク複製インストールを利用する手順を次に示します。

手順

1. インストール済みの製品 (Developer, Service Architect または Client) をアンインストールします。

製品のアンインストール手順については, 「[3.3.2 Application Server をアンインストールする \(Windows の場合\)](#)」を参照してください。

2. ディスク複製インストールを実施します。

ディスク複製インストールの操作・運用方法については, JP1/ServerConductor/Deployment Manager のマニュアルを参照してください。

3. 製品 (Developer, Service Architect または Client) をインストールします。

製品のインストール手順については, 「[2.2.2 Application Server を新規インストールする \(Windows の場合\)](#)」を参照してください。

付録 G.2 Application Server をディスク複製インストールする (Linux の場合)

Application Server および Service Platform は, JP1/SC/DPM を使ったディスク複製インストールに対応しています。JP1/SC/DPM のディスク複製インストール機能, および仮想化環境が提供するイメージファイル化による複製機能が使用できます。

なお, 次の構成ソフトウェアには, JP1/SC/DPM を使ったディスク複製インストールの利用に前提条件があるので, 利用前に確認してください。

- Component Container (J2EE アプリケーションを実行するシステム (J2EE サーバ) 使用時)
EJB クライアントアプリケーションが稼働するマシンは, ディスク複製インストールに対応していません。
- TPBroker
JP1/SC/DPM との連携方法については, マニュアル「TPBroker 運用ガイド」を参照してください。

ディスク複製インストールを利用する手順を次に示します。

手順

1. ディスク複製インストール元のマシンでアプリケーションサーバを構築します。

製品をインストールしたディスク複製インストール元のマシンに, システム環境をセットアップしたあと, 各種設定をカスタマイズしたり, 業務アプリケーションを設定したりします。

2. Application Server および Service Platform の構成ソフトウェアごとに、ディスク複製インストールを実施する前に必要な作業を実施します。

実施する作業については、「[付録 I.4 ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項 \(Linux の場合\)](#)」の構成ソフトウェアごとの「複製を実施する前にすること」を参照してください。

3. JP1/SC/DPM を使用したディスク複製インストールを実施します。

JP1/SC/DPM を使ったディスク複製インストールの操作・運用方法については、JP1/ServerConductor/Deployment Manager のマニュアルを参照してください。仮想化環境が提供するイメージファイル化による複製機能については、仮想化ソフトウェアのドキュメントを参照してください。

4. Application Server および Service Platform の構成ソフトウェアごとに、ディスク複製インストールを実施したあとに必要な作業を実施します。

実施する作業については、「[付録 I.4 ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項 \(Linux の場合\)](#)」の構成ソフトウェアごとの「複製を実施したあとにすること」を参照してください。

5. ディスク複製インストール先のマシンに接続するマシンでシステムの要件に合わせて各種設定をカスタマイズします。

各種設定には、例えば、次のようなものがあります。システムの要件に合わせて変更してください。

- システム構成上でフロントエンドに配置される負荷分散機の設定
- DBMS に関する設定
- システム構成上でバックエンドに配置される J2EE サーバ、または Web コンテナサーバに関する設定

付録 H リモートインストール

リモートインストールを利用する場合の手順について説明します。リモートインストールとは、JP1/NETM/DM や Groupmax Remote Installation を使用し、ネットワークを介してソフトウェアをクライアントへ配布するインストールのことです。

付録 H.1 Application Server をリモートインストールする (Windows の場合)

アプリケーションサーバおよび BPM/ESB 基盤のすべての製品は、JP1/NETM/DM および Groupmax Remote Installation を使ったリモートインストール（ソフトウェア配布）に対応しています。各製品を運用するサーバで JP1/NETM/DM や Groupmax Remote Installation を運用することによって、製品の新規インストールおよび更新インストールができます。なお、JP1/NETM/DM を使った実際のリモートインストール方法については、JP1/NETM/DM のマニュアルを参照してください。また、Groupmax Remote Installation を使ったリモートインストール方法については、Groupmax Remote Installation のマニュアルを参照してください。

リモートインストールをするときの注意事項については、「[付録 I.5 リモートインストールをするときの注意事項](#)」を参照してください。

付録 H.2 Application Server をリモートインストールする (UNIX の場合)

Application Server および Service Platform は、JP1/NETM/DM を使ったリモートインストール（ソフトウェア配布）に対応しています。各製品を運用するサーバで JP1/NETM/DM を運用することによって、製品の新規インストールおよび更新インストールができます。なお、JP1/NETM/DM を使った実際のリモートインストール方法については、JP1/NETM/DM のマニュアルを参照してください。

付録I インストールおよびアンインストールするときの注意事項

ここでは、Application Server および構成ソフトウェアを、インストールおよびアンインストールするときの注意事項について説明します。

付録I.1 Application Server をインストールおよびアンインストールするときの注意事項

Application Server をインストールおよびアンインストールするときの注意事項を作業のタイミングに分けて説明します。

(1) 作業前に確認する注意事項

分類	注意事項	新規インストール	更新インストール	アンインストール
OS 共通	<p>ユーザ定義ファイルなど、カスタマイズ用のファイルが格納されたディレクトリ以外は、すべて削除されます。このため、製品のディレクトリ下に必要なファイルを格納しないでください。</p> <p>削除対象となっているディレクトリは無条件に削除されます。このため、これらのディレクトリにユーザ固有のファイルなどを格納しないでください。</p>	—	—	○
	<p>このバージョンの製品または評価版をインストールしてアンインストールしたあとに古いバージョンの製品をインストールする場合は、インストール先にこのバージョンで作成された環境（<Application Server のインストールディレクトリ>¥CC、または/opt/Cosminexus/CC）が残っていないことを確認してください。このバージョンで作成された環境が残っていると、古いバージョンへ移行されないため、製品の動作やその後の更新インストールなどが保証されません。このバージョンで作成された環境が残っている場合は、インストール先を変更するか、このバージョンで作成された環境を削除または移動してからインストールしてください。</p>	○	—	—
	<p>この製品のインストーラでは、旧バージョンのアプリケーションからの移行（更新インストール）および上書きインストールを実施すると、この製品が提供しているファイルは初期化されます。そのため、<Application Server のインストールディレクトリ>¥jdk 下の必要なファイルは、インストーラ実行前に退避しておくことを推奨します。</p> <p>ただし、<Application Server のインストールディレクトリ>¥jdk 下の、この製品が提供していないファイルは削除されずに残ります。これらのファイルについては、ファイル削除やファイル移動など、必要に応じた操作をしてください。</p> <p>旧バージョンとこのバージョンのインストーラの変更点について、または、他の更新インストールおよび上書きインストールの詳細については、マニュアル「Cosminexus V11 アプリケーションサーバ 機能解説 保守／移行編」の「10. 旧バージョンのアプリケーションサーバからの移行 (J2EE サーバモードの場合)」を参照してください。</p>	—	○	—

分類	注意事項	新規インストール	更新インストール	アンインストール
	製品を更新インストールする場合、構築済みの環境にインストールされている Java バージョンと同じ Cosminexus Developer's Kit for Java (TM) を選択してください。	—	○	—
Windows 共通	この製品は、後継製品とする旧製品のアプリケーションサーバがインストール済みの環境にはインストールできません。この場合は、インストール済みの旧製品をアンインストールしたあとに、インストールしてください。	○	—	—
	製品をインストールする前に、インストール先ディスクの空き容量を確認し、インストールに必要な空き容量が十分確保されていることを確認してください。	○	—	—
	製品を更新インストールする場合は、必ずインストール前に、ユーザ作成ファイルなどの必要なファイルをバックアップしてください。 また、異なるバージョンへの更新インストール（バージョンアップ）をする場合、製品で提供しているファイルをほかのディレクトリにコピーして運用しているときは、更新インストール後に新しくインストールされたファイルをコピーし直してから運用してください。	—	○	—
	製品をインストールおよびアンインストールする前に、起動中のアプリケーションは終了させてください。常駐プログラムも終了させてください。	○	○	○
	この製品が提供するプログラムやサービスを実行中の状態で、インストールまたはアンインストールすると障害が発生することがあります。製品が提供するプログラムやサービス、および不要なプログラムやサービスは停止した状態にしてから、インストールまたはアンインストールしてください。	○	○	○
	この製品が提供するファイルを使用中の状態で、インストールまたはアンインストールすると障害が発生することがあります。製品が提供するファイルは使用しない状態にしてから、インストールまたはアンインストールしてください。	○	○	○
	「Hitachi Web Server」サービスが登録されている環境では、同一バージョンの更新インストール（上書き）をする前に、「Hitachi Web Server」サービスを停止してください。	—	○	—
	製品および構成ソフトウェアのインストールおよびアンインストール開始時に、セットアップの準備を示すダイアログが表示されます。このダイアログ内のキャンセルボタンは押さないでください。	○	○	○
	バージョンダウンとなるような更新インストールは実行しないでください。	—	○	—
	Management Server および運用管理エージェントのサービスのログオンアカウントには、デフォルトであるローカルシステムアカウント以外は指定しないでください。ただし、J2EE アプリケーションからネットワークリソースへアクセスする場合には、運用管理エージェントのサービスのログオンアカウントをローカルホストの Administrators グループに所属するように設定してください。	○	○	—
	単体版の Web Server がインストール済みの環境に、HTTP Server はインストールできません。単体版の Web Server をアンインストールしてから、インストールしてください。	—	○	—

分類	注意事項	新規インストール	更新インストール	アンインストール
	単体版の Web Server をアンインストールする前に、必要なファイルをバックアップしてください。	—	—	○
	単体版の Web Server がインストール済みの環境に製品をインストールする場合は、単体版の Web Server で使用していたファイル (httpsd.conf などのコンフィグファイルやユーザが作成したコンテンツ) を移行し、新しいインストールディレクトリに対応するように、必要な修正をしてください。	○	—	—
	インストール先に<システムドライブ>:¥Windows¥System32 ディレクトリ下は指定しないでください。	○	—	—
UNIX 共通	<p>次の製品を使用する場合には、製品の共通モジュールを必ずインストールしてください。</p> <ul style="list-style-type: none"> Application Server Service Platform また、アプリケーションサーバ Version 6 以前の Common Library がすでにインストールされている場合は、Common Library をアンインストールしたあとに、製品の共通モジュールをインストールしてください。 	○	—	—
	製品のインストールおよびアンインストールでは、PP インストーラを使用します。インストールおよびアンインストールには root 権限が必要です。	○	○	○
	<p>トレース共通ライブラリは、製品の構成ソフトウェアをアンインストールしたときに、必要に応じてアンインストールされます。ただし、トレース共通ライブラリのアンインストールに失敗してメッセージが表示された場合には、コマンドを実行してトレース共通ライブラリ単体でアンインストールすることもできます。</p> <p>アンインストールするには、次のコマンドを実行します。</p> <pre>/opt/hitachi/HNTRLib2/etc/hntr2setup</pre> <p>メニューが表示されるので、通常アンインストールは「9」を、強制アンインストールは「0」を選択してください。</p>	—	—	○
	OS インストール後のデフォルトの設定では、/etc/hosts ファイルに 127.0.0.1 の IP アドレスに対して自ホスト名が設定されます。TPBroker を使って通信する場合には、ネットワークインターフェースに割り当てられた IP アドレスに対して自ホスト名が設定されている必要があります。そのため、/etc/hosts ファイルには、自ホスト名に対してネットワークインターフェースに割り当てられた適切な IP アドレスを設定してください。	○	—	—
	OS インストール後のデフォルトの設定では、/etc/hosts ファイルに 127.0.0.1 の IP アドレスに対して自ホスト名が設定されます。Performance Tracer を使う場合には、ネットワークインターフェースに割り当てられた IP アドレスに対して自ホスト名が設定されている必要があります。そのため、/etc/hosts ファイルには、自ホスト名に対してネットワークインターフェースに割り当てられた適切な IP アドレスを設定してください。	○	—	—
	Application Server および Service Platform の場合、インストール先に、ディスク占有量分の空きがあることを確認してください。ディスク占有量については、「リリースノート」でご確認ください。	○	—	—

分類	注意事項	新規インストール	更新インストール	アンインストール
	Application Server および Service Platform の場合、一時ディレクトリ (/tmp) に、ディスク占有量分の空きがあることを確認してください。ディスク占有量については、「リリースノート」でご確認ください。/tmp に空きがない場合は、容量に空きがあるディレクトリを環境変数 TMPDIR に設定してからインストールしてください。	○	—	—

(凡例) ○: 該当します。 —: 該当しません。

(2) 作業中に確認する注意事項

分類	注意事項	新規インストール	更新インストール	アンインストール
Windows 共通	「Hitachi Web Server」サービスが登録されている環境では、同一バージョンの更新インストール（上書き）時にエラー（レジストリアクセスエラーなど）が発生した場合は、アンインストールしたあと、再度インストールしてください。	—	○	—
	インストーラを使用してインストールしているときは、絶対に、製品の提供媒体を CD-ROM ドライブから外したり、インストーラを強制終了させたりしないでください。これらの操作をすると、インストール情報に不整合が発生し、以降、インストールおよびアンインストールが実行できなくなるおそれがあります。	○	○	—
	インストールの処理を完了するためにインストール終了後、必ず OS を再起動してください。なお、インストール後、アンインストールする場合もインストールを完了させるために OS を再起動してから、アンインストールしてください。	○	○	○
	製品をアンインストールした場合、または製品に含まれる構成ソフトウェアをアンインストールした場合は、アンインストールを完了させるために必ず OS を再起動させてください。	—	—	○
	製品および製品に含まれる構成ソフトウェアでは、インストール終了処理に時間が掛かることがあります。インストールが完了し、次の画面に移るまで待ってください。	○	○	—
	インストール中にキャンセルした場合は、キャンセルした構成ソフトウェアからあとのインストール処理がキャンセルされます。キャンセルした構成ソフトウェアより前に処理が完了した構成ソフトウェアはインストールされた状態になります。 また、アンインストール中にキャンセルした場合も、キャンセルした構成ソフトウェアからあとのアンインストール処理がキャンセルされます。キャンセルした構成ソフトウェアより前に処理が完了した構成ソフトウェアはアンインストールされた状態になります。	○	○	○
	途中で一度もアンインストールしないで、同一バージョンの更新インストールを複数回実行したあとにアンインストールをすると、エラーが発生して失敗することがあります。この場合は、再度アンインストールを実行してください。一部のディレクトリおよびファイルが削除されないことがありますが、次回のインストールの実行には影響ありません。	—	—	○

分類	注意事項	新規インストール	更新インストール	アンインストール
	インストールの途中でインストーラから呼び出している conime.exe プロセスが終了しないまま、インストール処理がハングアップすることがあります。conime.exe プロセスが存在している状態でインストール処理がハングアップした場合は、タスクマネージャ (Taskmgr.exe) などから起動中の conime.exe を終了させてください。	○	○	—

(凡例) ○: 該当します。 −: 該当しません。

(3) 作業後に確認する注意事項

分類	注意事項	新規インストール	更新インストール	アンインストール
OS 共通	インストールに失敗した場合は、次の点を確認してください。 <ul style="list-style-type: none"> 起動しているサービスやアクセス中のファイル、ディレクトリがないか ファイル、ディレクトリのアクセス権が拒否されていないか 	○	○	—
	異なるバージョンへの更新インストール（バージョンアップ）をして環境を移行した場合、作業ディレクトリは旧バージョンで使用していたディレクトリを引き続き使用します。旧バージョンの作業ディレクトリを誤って削除しないようにしてください。	—	○	—
	運用管理エージェントの設定ファイルである adminagent.xml は、インストール時に上書きされ、既存のファイルは adminagent.xml.bak として退避されます。adminagent.xml を編集して運用している場合は、新たにインストールされたファイルに、以前に編集した項目を再編集してください。	—	○	—
	snapshot ログ収集先対象リストファイルである snapshotlog.conf、および snapshotlog.2.conf は、インストール時に上書きされ、既存のファイルは snapshotlog.conf.bak、および snapshotlog.2.conf.bak として退避されます。snapshot ログの収集対象を編集して運用している場合は、新たにインストールされたファイルに、以前に編集した項目を再編集してください。	—	○	—
Windows 共通	「Hitachi Web Server」サービスが登録されている環境では、同一バージョンの更新インストール（上書き）をしたあと、「Hitachi Web Server」サービスがそのまま残ります。「Cosminexus HTTP Server」サービスは登録されません。	—	○	—
	「Hitachi Web Server」サービスが登録されている環境では、Web Server から異なるバージョンへの更新インストール（バージョンアップ）をする場合、HTML マニュアル、GUI サーバ管理機能に関するファイル、ディレクトリ、ユーティリティなどが削除されないで残ることがあります。これらのファイルは更新後のバージョンでは使用できません。	—	○	—
UNIX 共通	インストール先のディレクトリおよびファイルは、root 権限のあるユーザだけが更新できます。このため、Component Container 管理者を設定している場合は、次に示すディレクトリおよびファイルに、Component Container 管理者の書き込み権限を設定してください。 <ul style="list-style-type: none"> /opt/Cosminexus/c4web/attachments ディレクトリ /opt/Cosminexus/c4web/conf ディレクトリ以下のファイル 	○	○	—

分類	注意事項	新規インストール	更新インストール	アンインストール
	<ul style="list-style-type: none"> • /opt/Cosminexus/c4web/logs ディレクトリ <p>また、添付ファイルの退避先ディレクトリや、トレースファイルの出力先ディレクトリを指定している場合は、そのディレクトリの属性も設定してください。</p> <p>なお、同一バージョンの更新インストール（上書き）をした場合、各ディレクトリおよびファイルの属性はデフォルトの状態に戻ります。この場合、運用に合わせて、各ディレクトリおよびファイルの属性を変更してください。</p>			

(凡例) ○：該当します。 –：該当しません。

付録 I.2 構成ソフトウェアをインストールおよびアンインストールするときの注意事項

構成ソフトウェアをインストールおよびアンインストールするときの注意事項を作業のタイミングに分けて説明します。

(1) 作業前に確認する注意事項

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
OS 共通	Component Container	<p>Component Container を新規インストールする場合にインストールディレクトリが作成されているときは、インストールディレクトリにファイルがないことを確認してください。</p> <p>製品がインストールするファイルまたはディレクトリと同名のまたはディレクトリがあると、ファイルまたはディレクトリが上書きされます。必要に応じて、ディレクトリやファイルの移動、バックアップの作成、またはインストール先の変更をしてください。</p> <p>Windows の場合（デフォルト）</p> <ul style="list-style-type: none"> • <OS インストールドライブ>:\$Program Files\$Hitachi\$Cosminexus\$CC • <OS インストールドライブ>:\$Program Files\$Hitachi\$Cosminexus\$manager • <OS インストールドライブ>:\$Program Files\$Hitachi\$Cosminexus\$c4web • <OS インストールドライブ>:\$Program Files\$Hitachi\$Cosminexus\$plugins <p>UNIX の場合（固定）</p> <ul style="list-style-type: none"> • /opt/Cosminexus/CC <p>Component Container で異なるバージョンへの更新インストール（バージョンアップ）をする場合、次のディレクトリが削除されます。必要なファイルがある場合は、更新インストール前に退避してください。</p>	○	–	–

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
		<p>Windows の場合</p> <ul style="list-style-type: none"> • <Application Server のインストールディレクトリ>¥CC¥admin¥help • <Application Server のインストールディレクトリ>¥CC¥examples • <Application Server のインストールディレクトリ>¥CC¥gui <p>UNIX の場合</p> <ul style="list-style-type: none"> • /opt/Cosminexus/CC/admin/help • /opt/Cosminexus/CC/examples • /opt/Cosminexus/CC/gui 			
		Component Container で異なるバージョンへの更新インストール(バージョンアップ)をする場合、作業ディレクトリのディスク使用量の 2 倍以上の空き容量を確保してください。	—	○	—
		Component Container をインストールまたはアンインストールする場合、HTTP Server を停止してください。停止しないでインストールまたはアンインストールすると、リダイレクタおよび Performance Tracer を正常に更新またはアンインストールできません。	○	○	○
		Component Container をインストールまたはアンインストールする場合、J2EE サーバ、Web コンテナサーバ、および CORBA ネーミングサービスを停止してください。	○	○	○
		<p>Component Container で異なるバージョンへの更新インストール(バージョンアップ)をする場合、次の構成ソフトウェアがインストールされていたときはアンインストールしてください。ただし、環境を引き継ぐため、アンインストールした構成ソフトウェアのディレクトリは削除しないでください。</p> <ul style="list-style-type: none"> • Component Library • Manager • Web Services - Base 	—	○	—
	Component Container - Redirector	リダイレクタ機能をインストールまたはアンインストールする場合、HTTP Server を停止してください。停止しないでインストールまたはアンインストールすると、リダイレクタおよび Performance Tracer を正常に更新またはアンインストールできません。	○	○	○
Windows 共通	Component Container	Component Container をインストールする場合、Application Server のインストール先ディレクトリ名に"#"を含むパスを指定しないでください。	○	○	—
	Component Transaction Monitor	Component Transaction Monitor をインストールするためには、ディスク占有量に記載した容量が必要です。この容量には、インストール作業に必要な一時的な領域を含んでいます。この容量が確保されていないディスクへ Component Transaction Monitor をインストールしようとした場合、ディスクの空き容量が 5%以下になったときにインストール処理が中断します。この際、ディスクの空き領域を	○	—	—

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
		<p>確保するか、またはインストール作業を中止することができます。ディスクの空き領域を確保した場合は、インストールは正常に終了します。インストール作業を中止した場合は、部分的にインストールされたすべてのファイルを削除してください。インストール作業の進行状況によっては、Component Transaction Monitor をアンインストールするか、Component Transaction Monitor のインストールディレクトリ以下をすべて手作業で削除する必要があります。</p>			
	Developer's Kit for Java	<p>Developer's Kit for Java を更新インストールすると、<Application Server のインストールディレクトリ>/jdk/lib/security/cacerts にあるファイルは初期化されます。必要なファイルは退避しておいてください。</p> <p>ただし、次に示すファイルは、更新インストール前のファイルが、同じディレクトリに別名（ファイル名_バージョン番号_日付時刻）で保存されるため、退避不要です。</p> <ul style="list-style-type: none"> • <Application Server のインストールディレクトリ>/jdk/lib/security/cacerts • <Application Server のインストールディレクトリ>/jdk/conf/logging.properties • <Application Server のインストールディレクトリ>/jdk/conf/net.properties • <Application Server のインストールディレクトリ>/jdk/conf/sound.properties • <Application Server のインストールディレクトリ>/jdk/conf/management/jmxremote.access • <Application Server のインストールディレクトリ>/jdk/conf/management/management.properties • <Application Server のインストールディレクトリ>/jdk/conf/security/java.policy • <Application Server のインストールディレクトリ>/jdk/conf/security/java.security <p>更新インストール後に、ファイルの内容を戻したい場合は、別名で保存されたファイルを元のファイル名に戻すか、別名で保存されたファイルの内容を初期化されたファイルに取り込んでください。</p>	—	○	—
	HTTP Server	<p>HTTP Server の次に示す作業の前に、「Cosminexus HTTP Server」サービスを停止してください。</p> <ul style="list-style-type: none"> • 同一バージョンの更新インストール（上書き）前 • アンインストール前 	—	○	○
		HTTP Server のアンインストール前に、httpsd コマンドで登録したサービスを削除してください。	—	—	○
		HTTP Server のアンインストールでは、ファイルに読み取り専用属性が付与されている場合も削除されます。	—	—	○
	Performance Tracer	Performance Tracer をインストールするためには、ディスク占有量に記載した容量が必要です。この容量には、インストール作業に必要	○	—	—

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
		な一時的な領域を含んでいます。この容量が確保されていないディスクへ Performance Tracer をインストールしようとした場合、ディスクの空き容量が 5%以下になったときにインストール処理が中断します。この際、ディスクの空き領域を確保するか、またはインストール作業を中止することができます。ディスクの空き領域を確保した場合は、インストールは正常に終了します。インストール作業を中止した場合は、部分的にインストールされたすべてのファイルを削除してください。インストール作業の進行状況によっては、Performance Tracer をアンインストールするか、Performance Tracer のインストールディレクトリ以下をすべて手作業で削除する必要があります。			
UNIX 共通	Component Container	Component Container のインストールディレクトリは、「/opt/Cosminexus/CC」固定です。	○	—	—
		Component Container をインストールする場合、umask の影響を受けるファイルがあります。適切な umask を設定してください。	○	—	—
		Component Container を更新インストールする場合に、インストールディレクトリが変更されたときは、新規インストールになります。	—	○	—
	Component Container - Redirector	リダイレクタ機能のインストールディレクトリは、「/opt/Cosminexus/CCR」固定です。	○	—	—
		リダイレクタ機能をインストールする場合、umask の影響を受けるファイルがあります。適切な umask を設定してください。	○	—	—
	Developer's Kit for Java	Developer's Kit for Java を更新インストールすると、opt/Cosminexus/jdk にあるファイルはすべて削除されます。必要なファイルは退避しておいてください。 ただし、次に示すファイルは、更新インストール前のファイルが、同じディレクトリに別名（ファイル名_バージョン番号_日付時刻）で保存されるため、退避不要です。 <ul style="list-style-type: none">• /opt/Cosminexus/jdk/lib/security/cacerts• /opt/Cosminexus/jdk/conf/logging.properties• /opt/Cosminexus/jdk/conf/net.properties• /opt/Cosminexus/jdk/conf/sound.properties• /opt/Cosminexus/jdk/conf/management/jmxremote.access• /opt/Cosminexus/jdk/conf/management/management.properties• /opt/Cosminexus/jdk/conf/security/java.policy• /opt/Cosminexus/jdk/conf/security/java.security 更新インストール後に、ファイルの内容を戻したい場合は、別名で保存されたファイルを元のファイル名に戻すか、別名で保存されたファイルの内容を初期化されたファイルに取り込んでください。	—	○	—
	HTTP Server	すでに HTTP Server がインストールされているシステムに再インストールする場合、同一バーションの更新インストール（上書き）ができます。このとき、HTTP Server をインストールする前に、HTTP	—	○	—

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
		Server の全プログラムおよび関連ファイルへの操作をすべて終了してください。			
		HTTP Server をインストールする場合、コンフィグファイルがすでにあるときは、バージョン番号を付与した別名称でインストールされます。	○	○	－
		HTTP Server をインストールする場合、/opt/hitachi/httpsd/htdocs/index.html がすでにあるときは上書きされません。	○	○	－
		HTTP Server で同一バージョンの更新インストール（上書き）をする場合、再インストール後も使用するファイル（httpsd.conf, sslc.cnfなどのコンフィグファイルやユーザが作成したコンテンツ）は、あらかじめ退避しておくことをお勧めします。	－	○	－
		PP インストーラを使った HTTP Server のインストールに失敗した場合は、HTTP Server の全プログラムおよび関連ファイルへの操作がすべて終了していることを確認してから、再度インストールしてください。また、HTTP Server で提供しているディレクトリと同一名称のファイルがあった場合、インストールに失敗することがあります。/opt/hitachi/httpsd 以下に同名のファイルがないことを確認してから、再度インストールしてください。	○	－	－
		Web Server から HTTP Server へ異なるバージョンへの更新インストール（バージョンアップ）をする場合、以前のバージョンの HTML マニュアル、GUI サーバ管理機能に関するファイル、ディレクトリ、一部のユーティリティは削除されないで残ります。これらのファイルの使用はサポートしていません。	－	○	－
		HTTP Server のアンインストール時、/opt/hitachi/httpsd 以下のファイルはすべて削除されます。必要なファイルはあらかじめ退避してください。	－	－	○
		Web Services - Security をインストールする環境に、バージョン 6 以前の XML Security がすでにインストールされている場合は、XML Security をアンインストールしたあとに、Web Services - Security をインストールしてください。	○	－	－
	XML Processor	XML Processor で同一バージョンの更新インストール（上書き）をする場合、/opt/Cosminexus/jaxp にあるすべてのファイルをいったん削除します。そのため、上書きインストール後に必要なファイルは、あらかじめ退避しておいてください。	－	○	－

(凡例) ○：該当します。 －：該当しません。

(2) 作業中に確認する注意事項

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
Windows 共通	HTTP Server	HTTP Server の同一バージョンの更新インストール（上書き）で、エラー（レジストリアクセスエラーなど）が発生した場合は、アンインストールしたあとに、再度インストールしてください。	－	○	－

(凡例) ○：該当します。 －：該当しません。

(3) 作業後に確認する注意事項

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
OS 共通	Component Container	Component Container で異なるバージョンへの更新インストール（バージョンアップ）をして環境を移行した場合、作業ディレクトリは旧バージョンで使用していたディレクトリを引き続き使用します。旧バージョンの作業ディレクトリを誤って削除しないようにしてください。	－	○	－
		Management Server 上で論理 Web サーバを削除しないで、HTTP Server をアンインストールして再インストールした場合、Management Server で管理されている論理 Web サーバは削除されませんが、実サーバ（Web サーバ（HTTP Server））は削除されます。このため、再インストール後、Management Server で該当する論理 Web サーバを再セットアップする必要があります。この場合は、運用管理ポータルで再度論理 Web サーバのセットアップと配布を実行してください。 また、httpsd.conf ファイルは、運用管理ポータルで論理 Web サーバの設定情報を配布した時に新規作成または上書きされます。このため、手動でhttpsd.conf ファイルを修正していた場合は、運用管理ポータルで論理 Web サーバの設定情報を配布したあとに、必要に応じて、再度httpsd.conf ファイルを修正してください。なお、運用管理ポータルの「Web サーバの設定」画面で、httpsd.conf ファイルの内容を編集して配布している場合は、修正不要です。	○	－	－
		Component Container で同一バージョンの更新インストール（上書き）をしても、アプリケーションサーバにインポート、デプロイ済みの DB Connector はアップデートされません。インポート、デプロイ済みの DB Connector をアップデートするためには、cjraruupdate コマンドを実行してください。cjraruupdate コマンドの実行形式を次に示します。 <code>/opt/Cosminexus/CC/server/bin/cjraruupdate -type dbconnector</code> 詳細は、マニュアル「アプリケーションサーバリファレンス コマンド編」の「cjraruupdate（リソースアダプタのバージョンアップ）」を参照してください。	－	○	－

分類	構成ソフトウェア	注意事項	新規インストール	更新インストール	アンインストール
		cjrarupdate コマンド実行時、DB Connector に変更がない場合は KDJ42249-W メッセージが出力されます。メッセージが出力されても問題はありません。	－	○	－
		Component Container をインストールした場合、古いバージョンで取得した J2EE アプリケーションの RMI-IIOP スタブおよびインターフェースは使用できません。cjstartapp コマンド実行後、cjgetstubsjar コマンドを実行して J2EE アプリケーションの RMI-IIOP スタブおよびインターフェースを取得し直してください。 詳細は、マニュアル「アプリケーションサーバリファレンス コマンド編」の「cjgetstubsjar (アプリケーションの RMI-IIOP スタブおよびインターフェースの取得)」を参照してください。	○	○	－
	Reliable Messaging	Reliable Messaging を更新インストールすると、Reliable Messaging が提供するコマンドのリクエストタイムアウト値はデフォルト値 (0) に戻ります。<Application Server のインストールディレクトリ>/RM/bin ディレクトリ下にある、バッチファイルまたはシェルスクリプトを編集して、リクエストタイムアウト値を変更している場合は、再度、値を設定してください。	－	○	－
		Reliable Messaging のコマンドを実行すると、<Application Server のインストールディレクトリ>/RM/logs にコマンド実行時のログが出力されます。アンインストール時、このディレクトリ下にあるログディレクトリやファイルが残ることがあります。	－	－	○
Windows 共通	Developer's Kit for Java	Developer's Kit for Java をアンインストールしても一部のファイルやディレクトリが削除されないで、<Application Server のインストールディレクトリ>/jdk 以下に残ることがあります。削除されなかったファイルやディレクトリは手動で削除してください。	－	－	○
		Developer's Kit for Java の更新インストール時に、インストーラが別名で保存したファイルは、アンインストールしても削除されません。また、更新インストールを繰り返し実施すると、実施した回数分、別名で保存したファイルが作成されます。必要に応じて不要なファイルやディレクトリは削除してください。	－	○	○
	HTTP Server	HTTP Server の新規インストールでは、「Cosminexus HTTP Server」サービスのスタートアップの種類は「手動」で登録されます。	○	－	－
		HTTP Server のアンインストールでは、インストール後に作成されるファイル、コンフィグファイルは削除されません。また、そのほかのファイルやディレクトリが削除されないで残ることがあります。	－	－	○
UNIX 共通	Component Container	Component Container をアンインストールしても/opt/Cosminexus/CC/spool ディレクトリは削除されません。アンインストール後に、手動で削除してください。	－	－	○

(凡例) ○：該当します。 －：該当しません。

付録 I.3 ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項 (Windows の場合)

構成ソフトウェアごとに、複製をするときの注意事項を説明します。

(1) Component Container での注意事項

Component Container での注意事項を次に示します。Component Container では、構築するシステム構成によって実施する作業などが異なります。

(a) J2EE アプリケーションを実行するシステム (J2EE サーバ) 使用時

J2EE アプリケーションを実行するシステム (J2EE サーバ) 使用時、Component Container での注意事項を次に示します。

- 複製を実施する前にすること

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1. および手順 2. を実施してください。Management Server を使用しているときは手順 3. も実施してください。

手順

1. 複製元のマシンで、次に示すプログラムを終了します。

- J2EE サーバ (グローバルトランザクションを使用している場合は、開始済みのトランザクションを決着させてから終了してください)
 - Web コンテナサーバ
 - 運用管理エージェント
 - Management Server
 - HTTP Server、または Microsoft IIS

2. 複製元のマシンで、次のディレクトリ下にファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥maintenance¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥maintenance¥mmap¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥rmi¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥rmi¥mmap¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥mmap¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥TPB¥logj¥comtrc¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥TPB¥logj¥mdltrc¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥TPB¥logj¥*
- <Application Server のインストールディレクトリ>¥CC¥client¥logs¥*

- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥mmap¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥system¥ejbcl¥mmap¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥system¥ejbcl¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥system¥TPB¥logj¥comtrc¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥system¥TPB¥logj¥mdltrc¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥system¥TPB¥logj¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名>¥logs¥TPB¥logj¥comtrc¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名>¥logs¥TPB¥logj¥mdltrc¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名>¥logs¥TPB¥logj¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名>¥logs¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥redirector¥servers¥<サーバ名>¥logs¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥redirector¥servers¥<サーバ名>¥logs¥mmap¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥redirector¥logs¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥redirector¥logs¥mmap¥*
- ・<Application Server のインストールディレクトリ>¥manager¥spool¥default¥*
- ・<Application Server のインストールディレクトリ>¥manager¥spool¥lsinfo¥*
- ・<Application Server のインストールディレクトリ>¥manager¥tmp¥*
- ・<Application Server のインストールディレクトリ>¥manager¥setup¥log¥*
- ・<Application Server のインストールディレクトリ>¥manager¥setup¥log¥maintenance¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥watch¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥CC¥maintenance¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥CC¥rmi¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥connectors¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥connectors¥<アプリケーション表示名>¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥connectors¥TEST#<アプリケーション表示名>¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥TPB¥logj¥mdltrc¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥TPB¥logj¥comtrc¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥TPB¥logj¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥http¥comm¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥http¥thr¥*

- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥http¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥*
- ・<Manager ログ出力ディレクトリ>¥maintenance¥*
- ・<Manager ログ出力ディレクトリ>¥maintenance¥mmap¥*
- ・<Manager ログ出力ディレクトリ>¥message¥*
- ・<Manager ログ出力ディレクトリ>¥message¥mmap¥*
- ・<Manager ログ出力ディレクトリ>¥trace¥*
- ・<Manager ログ出力ディレクトリ>¥trace¥mmap¥*
- ・<Manager ログ出力ディレクトリ>¥*
- ・<Manager ログ出力ディレクトリ>¥<snapshot ログの出力先ディレクトリ>¥*
- ・<監査ログ出力ディレクトリ>¥*
- ・<監査ログのメッセージログ出力ディレクトリ>¥*
- ・<監査ログのメッセージログ出力ディレクトリ>¥mmap¥*
- ・<監査ログの例外情報出力ディレクトリ>¥*
- ・<監査ログの例外情報出力ディレクトリ>¥mmap¥*

参考

各ディレクトリのデフォルトを次に示します。

- ・<作業ディレクトリ> : <Application Server のインストールディレクトリ>¥CC¥server¥public
- ・<Manager ログ出力ディレクトリ> : <Application Server のインストールディレクトリ>¥manager¥log
- ・<snapshot ログの出力先ディレクトリ> : <Manager ログ出力ディレクトリ>¥snapshot
- ・<監査ログ出力ディレクトリ> : <Application Server のインストールディレクトリ>¥auditlog
- ・<監査ログのメッセージログ出力ディレクトリ> : <Application Server のインストールディレクトリ>¥auditlog
- ・<監査ログの例外情報出力ディレクトリ> : <Application Server のインストールディレクトリ>¥auditlog

3. Management Server を使用している場合は、複製元のマシンでプロパティファイルの設定を変更します。

複製元のマシンで Management Server が動作しているかどうかで作業が異なります。

複製元のマシンで Management Server が動作していないとき

複製先のマシンがマルチホームドホストのときは、あらかじめ、複製元のマシンの運用管理エージェントプロパティファイル (adminagent.properties) に、「adminagent.adapter.bind_host=<複製先の管理 IP アドレスと一致する正規表現>」を設定してください。

複製元のマシンで Management Server が動作しているとき

次の表に示すプロパティを設定してください。

ファイル名	プロパティ	値
.cmxrc および cmxclient.properties	cmx.lb	keepvs
adminagent.properties	adminagent.adapter.allowedHosts	mserver.properties の mngrsvr.myhost.name プロパティが設定されているときは、同じ値を指定してください。 mserver.properties の mngrsvr.myhost.name プロパティが設定されていないときは、ホスト用 管理 IP アドレスと一致する正規表現を設定して ください。
mserver.properties	mngsvr.myhost.name	複製先のマシンがマルチホームドホストのとき は、<複製先の管理 IP アドレスと一致する正規 表現>を設定してください。
	webserver.connector.http.bind_host	
adminagent.properties	adminagent.adapter.bind_host	
.mngrsvrutilrc および mngrsvrutilcl.properties	mngrsvrutil.connect.host	ループバックアドレス以外を設定するときは、< 複製先の管理 IP アドレスと一致する正規表現> を設定してください。
	mngrsvrutil.target_name	
.cmxrc および cmxclient.properties	cmx.connect.host	

・複製を実施したあとにすること

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.～手順 3.を実施してください。Management Server を使用しているときは手順 4.も実施してください。

手順

1. 次に示すプロパティを J2EE サーバ用ユーザプロパティファイル (usrconf.properties)，およびサーバ管理コマンド用ユーザプロパティファイル (usrconf.properties) で設定している場合は、複製先のマシンで値を変更します。

各プロパティで適切な値に変更してください。

- vbroker.se.iiop_tp.host
- ejbserver.ctm.CTMMyHost
- vbroker.se.iiop_ts.proxyHost
- vbroker.se.iiop_tp.proxyHost
- ejbserver.naming.host

2. 次に示すプロパティを J2EE サーバ用ユーザプロパティファイル (usrconf.properties)，および Web コンテナサーバ用ユーザプロパティファイル (usrconf.properties) で設定している場合は、複製先のマシンで値を変更します。

各プロパティの値を複製先の IP アドレスに変更してください。

- webserver.connector.http.bind_host
- webserver.connector.nio_http.bind_host

各プロパティの値を J2EE サーバ、または Web コンテナサーバごとにユニークな値に設定し直してください。

- webserver.container.server_id.value
- webserver.session.server_id.value

3. Management Server を使用している場合は、複製先のマシンで次の操作を実施します。

1. 複製で追加したホストの定義
2. 複製先ホスト上に定義されている複製元のホスト定義の削除（Management Server が動作しているマシンを複製したときだけ）
3. 複製で追加した論理サーバの定義
4. 3.で追加した論理サーバのセットアップ（実サーバ名指定）
5. 3.で追加した論理サーバの設定ファイルの読み込み
6. 3.で追加した論理サーバの設定ファイルの配布
7. サーバ管理コマンドで EAR ファイル、および RAR ファイルを削除した後の再配布（J2EE サーバのアプリケーション管理機能を使用しているときだけ）

(b) バッチアプリケーションを実行するシステム（バッチサーバ）使用時

バッチアプリケーションを実行するシステム（バッチサーバ）使用時、Component Container での注意事項を次に示します。

• 複製を実施する前にすること

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1. および手順 2. を実施してください。Management Server を使用しているときは手順 3. も実施してください。

手順

1. 複製元のマシンで、次に示すプログラムを終了します。

- バッチサーバ（リソースアダプタを開始したまま終了しても構いません）
- 運用管理エージェント
- Management Server

2. 複製元のマシンで、次のディレクトリ下にファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥maintenance¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥maintenance¥mmap¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥rmi¥*
- <Application Server のインストールディレクトリ>¥CC¥admin¥logs¥CC¥rmi¥mmap¥*

- ・<Application Server のインストールディレクトリ>¥CC¥admin¥logs¥mmap¥*
- ・<Application Server のインストールディレクトリ>¥CC¥admin¥logs¥TPB¥logj¥comtrc¥*
- ・<Application Server のインストールディレクトリ>¥CC¥admin¥logs¥TPB¥logj¥mdltrc¥*
- ・<Application Server のインストールディレクトリ>¥CC¥admin¥logs¥TPB¥logj¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥*
- ・<Application Server のインストールディレクトリ>¥CC¥client¥logs¥mmap¥*
- ・<Application Server のインストールディレクトリ>¥CC¥web¥containers¥<サーバ名>¥logs¥*
- ・<Application Server のインストールディレクトリ>¥manager¥spool¥default¥*
- ・<Application Server のインストールディレクトリ>¥manager¥spool¥lsinfo¥*
- ・<Application Server のインストールディレクトリ>¥manager¥tmp¥*
- ・<Application Server のインストールディレクトリ>¥manager¥setup¥log¥*
- ・<Application Server のインストールディレクトリ>¥manager¥setup¥log¥maintenance¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥watch¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥CC¥maintenance¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥CC¥rmi¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥connectors¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥TPB¥logj¥mdltrc¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥TPB¥logj¥comtrc¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥TPB¥logj¥*
- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥*
- ・<Manager ログ出力ディレクトリ>¥maintenance¥*
- ・<Manager ログ出力ディレクトリ>¥maintenance¥mmap¥*
- ・<Manager ログ出力ディレクトリ>¥message¥*
- ・<Manager ログ出力ディレクトリ>¥message¥mmap¥*
- ・<Manager ログ出力ディレクトリ>¥trace¥*
- ・<Manager ログ出力ディレクトリ>¥trace¥mmap¥*
- ・<Manager ログ出力ディレクトリ>¥*
- ・<snapshot ログの出力先ディレクトリ>¥*
- ・<監査ログ出力ディレクトリ>¥*
- ・<監査ログのメッセージログ出力ディレクトリ>¥*
- ・<監査ログのメッセージログ出力ディレクトリ>¥mmap¥*
- ・<監査ログの例外情報出力ディレクトリ>¥*
- ・<監査ログの例外情報出力ディレクトリ>¥mmap¥*

参考

各ディレクトリのデフォルトを次に示します。

- ・<作業ディレクトリ> : <Application Server のインストールディレクトリ>¥CC¥server¥public
- ・<Manager ログ出力ディレクトリ> : <Application Server のインストールディレクトリ>¥manager¥log
- ・<snapshot ログの出力先ディレクトリ> : <Manager ログ出力ディレクトリ>¥snapshot
- ・<監査ログ出力ディレクトリ> : <Application Server のインストールディレクトリ>¥auditlog
- ・<監査ログのメッセージログ出力ディレクトリ> : <Application Server のインストールディレクトリ>¥auditlog
- ・<監査ログの例外情報出力ディレクトリ> : <Application Server のインストールディレクトリ>¥auditlog

3. Management Server を使用している場合は、複製元のマシンでプロパティファイルの設定を変更します。

複製元のマシンで Management Server が動作しているかどうかで作業が異なります。

複製元のマシンで Management Server が動作していないとき

複製先のマシンがマルチホームドホストのときは、あらかじめ、複製元のマシンの運用管理エージェントプロパティファイル (adminagent.properties) に、「adminagent.adapter.bind_host=<複製先の管理 IP アドレスと一致する正規表現>」を設定してください。

複製元のマシンで Management Server が動作しているとき

次の表に示すプロパティを設定してください。

ファイル名	プロパティ	値
adminagent.properties	adminagent.adapter.allowedHosts	mserver.properties の mngrsvr.myhost.name プロパティが設定されているときは、同じ値を指定してください。 mserver.properties の mngrsvr.myhost.name プロパティが設定されていないときは、ホスト用管理 IP アドレスと一致する正規表現を設定してください。
mserver.properties	mngrsvr.myhost.name	複製先のマシンがマルチホームドホストのときは、<複製先の管理 IP アドレスと一致する正規表現>を設定してください。
	webserver.connector.http.bind_host	
adminagent.properties	adminagent.adapter.bind_host	
.mngrsvrutilrc および mngrsvrutilcl.properties	mngrsvrutil.connect.host	ループバックアドレス以外を設定するときは、<複製先の管理 IP アドレスと一致する正規表現>を設定してください。
	mngrsvrutil.target_name	
.cmxrc および cmxclient.properties	cmx.connect.host	

- ・複製を実施したあとにすること

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.を実施してください。Management Server を使用しているときは手順 2.も実施してください。

手順

1. 次に示すプロパティをバッチサーバ用ユーザプロパティファイル (usrconf.properties)、およびサーバ管理コマンド用ユーザプロパティファイル (usrconf.properties) で設定している場合は、複製先のマシンで値を変更します。

各プロパティで適切な値に変更してください。

- vbroker.se.iiop_tp.host
- ejbserver.naming.host

2. Management Server を使用している場合は、複製先のマシンで次の操作を実施します。

1. 複製で追加したホストの定義

2. 複製先ホスト上に定義されてい複製元のホスト定義の削除 (Management Server が動作しているマシンを複製したときだけ)

3. 複製で追加した論理サーバの定義

4. 3.で追加した論理サーバのセットアップ (実サーバ名指定)

5. 3.で追加した論理サーバの設定ファイルの読み込み

6. 3.で追加した論理サーバの設定ファイルの配布

7. サーバ管理コマンドで RAR ファイルを削除したとの再配布 (J2EE サーバのアプリケーション管理機能を使用しているときだけ)

(2) Component Container - Redirector での注意事項

Component Container - Redirector での注意事項を次に示します。

• 複製を実施する前にすること

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.および手順 2.を実施してください。

手順

1. 複製元のマシンで、次に示すプログラムを終了します。

- HTTP Server、または Microsoft IIS

2. 複製元のマシンで、次のディレクトリ下にログファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- <Application Server のインストールディレクトリ>¥CCR¥web¥redirector¥servers¥<サーバ名>¥logs¥*

- <Application Server のインストールディレクトリ>¥CCR¥web¥redirector¥servers¥<サーバ名>¥logs¥mmap¥*

- <Application Server のインストールディレクトリ>¥CCR¥web¥redirector¥logs¥*

- ・<Application Server のインストールディレクトリ>¥CCR¥web¥redirector¥logs¥mmap¥*

(3) Reliable Messaging での注意事項

Reliable Messaging での注意事項を次に示します。

- **複製を実施する前にすること**

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.および手順 2.を実施してください。

手順

1. J2EE サーバに Reliable Messaging をデプロイし、Reliable Messaging が開始している場合は、J2EE サーバ上で Reliable Messaging を停止します。

2. 複製元のマシンで、次のディレクトリ下にログファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- ・<作業ディレクトリ>¥ejb¥<サーバ名>¥logs¥RM¥maintenance

<作業ディレクトリ>のデフォルトは、<Application Server のインストールディレクトリ>¥CC¥server¥public です。

- **複製を実施したあとにすること**

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.～手順 5.を実施してください。

手順

1. 複製後のシステムでホスト名または IP アドレスを変更した場合は、次に示す Reliable Messaging の設定を更新します。

- ・環境変数 HRM_CMD_HOST の設定を、複製後のシステムのホスト名または IP アドレスに更新してください。

- ・キュー間転送機能を使用し、転送先システムのホスト名または IP アドレスを変更したときは、次に示す手順で転送元システムに存在するあて先と、そのあて先を使用する転送キューを再作成してください。

1. 転送元システムのあて先を使用しているすべての転送キューにメッセージがないことを確認します。

2. 転送元システムのあて先を使用しているすべての転送キューを削除します。

3. 転送元システムのあて先を削除します。

4. 変更後のホスト名または IP アドレスを指定して、転送元システムのあて先を新規作成します。

5. 転送元システムのあて先を使用する転送キューを新規作成します。

2. Reliable Messaging のシステム名が一意となるように、コンフィグレーションプロパティの RMSystemName と、環境変数 HRM_SYSTEM_NAME を変更します。

3. 手順 2.で変更したシステム名に対応する管理情報テーブルを作成します。

管理情報テーブルの作成方法については、マニュアル「Reliable Messaging」を参照してください。

4. DBMS に HiRDB を使用し、XA トランザクションを使用する構成の場合は、Reliable Messaging の連携先 DB Connector の XAOpenString が、同一 HiRDB に接続するリソースアダプタ間で一意になるように、連携先 DB Connector のコンフィグレーションプロパティの description と XAOpenString を変更します。

5. Reliable Messaging を開始して、必要なキューを作成します。

(4) Component Transaction Monitor での注意事項

Component Transaction Monitor での注意事項を次に示します。

- **複製を実施したあとにすること**

複製を実施したあとに、複製先のマシンで環境変数 CTMSPOOL に設定したディレクトリ下を削除してください。

(5) Performance Tracer での注意事項

複製を実施したあとに、複製先のマシンで環境変数 PRFSPOOL に設定したディレクトリ下を削除してください。

(6) HTTP Server での注意事項

HTTP Server での注意事項を次に示します。

- **複製を実施する前にすること**

複製を実施する前に、複製元のマシンで HTTP Server のすべてのプログラム、および関連ファイルへのすべての操作を終了してください。

- **複製を実施したあとにすること**

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.～手順 3.を実施してください。

手順

1. 必要に応じて、複製後のディレクティブの設定を変更します。

- ネットワーク関連のディレクティブ

ServerName, <VirtualHost>, BindAddress, Listen, NameVirtualHost

- ファイルシステム関連のディレクティブ

<Directory>, <DirectoryMatch>, Alias, AliasMatch, AuthGroupFile, AuthUserFile, CustomLog, DocumentRoot, ErrorLog, HWSRequestLog, HWSTraceIdFile, HWSTraceLogFile, Include, LoadFile, LoadModule, PidFile, QOSResponse, ScriptAlias, ScriptAliasMatch, ScriptLog, ServerRoot, SSLCACertificateFile, SSLCertificateFile, SSLCertificateKeyFile, SSLCertificateKeyPassword, SSLCRLDERPath, SSLCRLPEMPath, TransferLog, TypesConfig, UserDir

2. PidFile ディレクティブに指定されているファイルがコピーされている場合は、HTTP Server を起動する前に削除します。
3. ログ関連のファイルがコピーされている場合は、必要に応じて、不要なファイルを削除します。

(7) TPBroker での注意事項

TPBroker での注意事項を次に示します。

- **複製を実施したあとにすること**

複製を実施したあとに、実施する作業について説明します。手順 1.～手順 2.を実施してください。

手順

1. 仮想化プラットフォームが提供するイメージファイル化による複製機能を使用する場合に、次の設定があるときは、指定したホスト名や IP アドレスをシステムに合わせて変更します。

- agentaddr ファイル（環境変数 OSAGENT_ADDR_FILE またはプロパティ vbroker.agent.addrFile でも指定できます）
- localaddr ファイル（環境変数 OSAGENT_LOCAL_FILE またはプロパティ vbroker.agent.localFile でも指定できます）
- htc.clienthandleraddr ファイル（環境変数 HVI_OSGENT_CLIENTHANDLERADDR_FILE でも指定できます）
- プロパティ vbroker.agent.addr
- プロパティ vbroker.se.<サーバエンジン>.host
- プロパティ vbroker.se.<サーバエンジン>.proxyHost
- プロパティ vbroker.orb.initRef
- プロパティ vbroker.orb.defaultInitRef (C++だけ)
- コマンドラインオプション ORBInitRef
- コマンドラインオプション ORBDefaultInitRef (C++だけ)
- コマンドラインオプション ORBAgentAddr
- osagent のコマンドラインオプション -a
- 環境変数 OSAGENT_ADDR

また、ホスト名や IP アドレスを変更する前に作成した IOR ファイルは、変更後に再作成し、再配布してください。

2. 仮想化環境での複製機能を使用し、かつグローバルトランザクションを使用する場合、仮想化環境の複製元にインプロセストランザクションサービスのステータスファイルを含まないように、インプロセストランザクションサービスのステータスファイルがあるときは削除します。

(8) Service Coordinator での注意事項

Service Coordinator での注意事項を次に示します。

- **複製を実施する前にすること**

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.～手順 2.を実施してください。

手順

1. マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「実行環境に必要なソフトウェアを設定する」までの作業を実施します。
2. 次に示す構成ソフトウェアの「複製を実施する前にすること」の作業を実施します。
 - ・ Component Container
 - ・ Reliable Messaging
 - ・ HTTP Server
 - ・ TPBroker
- ・ 複製を実施したあとにすること

複製を実施したあとに、実施する作業について説明します。手順 1.～手順 2.を実施してください。

手順

1. 複製先のマシンで、次に示す構成ソフトウェアの「複製を実施したあとにすること」の作業を実施します。
 - ・ Component Container
 - ・ Reliable Messaging
 - ・ Component Transaction Monitor
 - ・ Performance Tracer
 - ・ HTTP Server
 - ・ TPBroker
2. 複製元のマシン、および複製先のマシンで、マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「HCSC サーバをセットアップする」以降の作業を実施します。

付録 I.4 ディスク複製インストールおよび仮想化プラットフォームが提供する複製機能を使用するときの注意事項 (Linux の場合)

構成ソフトウェアごとに、複製をするときの注意事項を説明します。

(1) Component Container での注意事項

Component Container での注意事項を次に示します。Component Container では、構築するシステム構成によって実施する作業などが異なります。

(a) J2EE アプリケーションを実行するシステム (J2EE サーバ) 使用時

J2EE アプリケーションを実行するシステム (J2EE サーバ) 使用時、Component Container での注意事項を次に示します。

- **複製を実施する前にすること**

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.および手順 2.を実施してください。Management Server を使用しているときは手順 3.も実施してください。

手順

1. 複製元のマシンで、次に示すプログラムを終了します。

- J2EE サーバ（グローバルトランザクションを使用している場合は、開始済みのトランザクションを決着させてから終了してください）
- Web コンテナサーバ
- 運用管理エージェント
- Management Server
- HTTP Server

2. 複製元のマシンで、次のディレクトリ下にファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- /opt/Cosminexus/CC/admin/logs/*
- /opt/Cosminexus/CC/admin/logs/CC/maintenance/*
- /opt/Cosminexus/CC/admin/logs/CC/maintenance/mmap/*
- /opt/Cosminexus/CC/admin/logs/CC/rmi/*
- /opt/Cosminexus/CC/admin/logs/CC/rmi/mmap/*
- /opt/Cosminexus/CC/admin/logs/mmap/*
- /opt/Cosminexus/CC/admin/logs/TPB/logj/comtrc/*
- /opt/Cosminexus/CC/admin/logs/TPB/logj/mdltrc/*
- /opt/Cosminexus/CC/admin/logs/TPB/logj/*
- /opt/Cosminexus/CC/client/logs/*
- /opt/Cosminexus/CC/client/logs/mmap/*
- /opt/Cosminexus/CC/client/logs/system/ejbcl/mmap/*
- /opt/Cosminexus/CC/client/logs/system/ejbcl/*
- /opt/Cosminexus/CC/client/logs/system/TPB/logj/comtrc/*
- /opt/Cosminexus/CC/client/logs/system/TPB/logj/mdltrc/*
- /opt/Cosminexus/CC/client/logs/system/TPB/logj/*
- /opt/Cosminexus/CC/web/containers/<サーバ名>/logs/TPB/logj/comtrc/*
- /opt/Cosminexus/CC/web/containers/<サーバ名>/logs/TPB/logj/mdltrc/*
- /opt/Cosminexus/CC/web/containers/<サーバ名>/logs/TPB/logj/*
- /opt/Cosminexus/CC/web/containers/<サーバ名>/logs/*
- /opt/Cosminexus/CC/web/redirector/servers/<サーバ名>/logs/*
- /opt/Cosminexus/CC/web/redirector/servers/<サーバ名>/logs/mmap/*
- /opt/Cosminexus/CC/web/redirector/logs/*

- /opt/Cosminexus/CC/web/redirector/logs/mmap/*
- /opt/Cosminexus/manager/spool/default/*
- /opt/Cosminexus/manager/spool/lsinfo/*
- /opt/Cosminexus/manager/tmp/*
- /opt/Cosminexus/manager/setup/log/*
- /opt/Cosminexus/manager/setup/log/maintenance/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/watch/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/CC/maintenance/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/CC/rmi/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/connectors/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/connectors/<アプリケーション表示名>/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/connectors/TEST#<アプリケーション表示名>/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/TPB/logj/mdltrc/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/TPB/logj/comtrc/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/TPB/logj/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/http/comm/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/http/thr/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/http/*
- <作業ディレクトリ>/ejb/<サーバ名>/logs/*
- <Manager ログ出力ディレクトリ>/maintenance/*
- <Manager ログ出力ディレクトリ>/maintenance/mmap/*
- <Manager ログ出力ディレクトリ>/message/*
- <Manager ログ出力ディレクトリ>/message/mmap/*
- <Manager ログ出力ディレクトリ>/trace/*
- <Manager ログ出力ディレクトリ>/trace/mmap/*
- <Manager ログ出力ディレクトリ>/*
- <snapshot ログの出力先ディレクトリ>/*
- <監査ログ出力ディレクトリ>¥*
- <監査ログのメッセージログ出力ディレクトリ>¥*
- <監査ログのメッセージログ出力ディレクトリ>¥mmap¥*
- <監査ログの例外情報出力ディレクトリ>¥*
- <監査ログの例外情報出力ディレクトリ>¥mmap¥*

参考

各ディレクトリのデフォルトを次に示します。

- <作業ディレクトリ> : /opt/Cosminexus/CC/server/public

- ・<Manager ログ出力ディレクトリ> : /opt/Cosminexus/manager/log
- ・<snapshot ログの出力先ディレクトリ> : <Manager ログ出力ディレクトリ>/snapshot
- ・<監査ログ出力ディレクトリ> : /opt/Cosminexus/auditlog
- ・<監査ログのメッセージログ出力ディレクトリ> : /opt/Cosminexus/auditlog
- ・<監査ログの例外情報出力ディレクトリ> : /opt/Cosminexus/auditlog

3. Management Server を使用している場合は、複製元のマシンでプロパティファイルの設定を変更します。

複製元のマシンで Management Server が動作しているかどうかで作業が異なります。

複製元のマシンで Management Server が動作していないとき

複製先のマシンがマルチホームドホストのときは、あらかじめ、複製元のマシンの運用管理エージェントプロパティファイル (adminagent.properties) に、「adminagent.adapter.bind_host=<複製先の管理 IP アドレスと一致する正規表現>」を設定してください。

複製元のマシンで Management Server が動作しているとき

次の表に示すプロパティを設定してください。

ファイル名	プロパティ	値
.cmxrc および cmxclient.properties	cmx.lb	keepvs
adminagent.properties	adminagent.adapter.allowedHosts	mserver.properties の mngrsvr.myhost.name プロパティが設定されているときは、同じ値を指定してください。 mserver.properties の mngrsvr.myhost.name プロパティが設定されていないときは、ホスト用 管理 IP アドレスと一致する正規表現を設定して ください。
mserver.properties	mngrsvr.myhost.name	複製先のマシンがマルチホームドホストのとき は、<複製先の管理 IP アドレスと一致する正規 表現>を設定してください。
	webserver.connector.http.bind_host	
adminagent.properties	adminagent.adapter.bind_host	
.mngrsvrutilrc および mngrsvrutilcl.properties	mngrsvrutil.connect.host	ループバックアドレス以外を設定するときは、< 複製先の管理 IP アドレスと一致する正規表現> を設定してください。
	mngrsvrutil.target_name	
.cmxrc および cmxclient.properties	cmx.connect.host	

・複製を実施したあとにすること

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.～手順 3.を実施してください。Management Server を使用しているときは手順 4.も実施してください。

手順

1. 次に示すプロパティを J2EE サーバ用ユーザプロパティファイル (usrconf.properties), およびサーバ管理コマンド用ユーザプロパティファイル (usrconf.properties) で設定している場合は、複製先のマシンで値を変更します。

各プロパティで適切な値に変更してください。

- vbroker.se.iiop_tp.host
- ejbserver.ctm.CTMMyHost
- ejbserver.naming.host
- ejbserver.rmi.naming.host

2. 次に示すプロパティを J2EE サーバ用ユーザプロパティファイル (usrconf.properties), および Web コンテナサーバ用ユーザプロパティファイル (usrconf.properties) で設定している場合は、複製先のマシンで値を変更します。

各プロパティの値を複製先の IP アドレスに変更してください。

- webserver.connector.http.bind_host
- webserver.connector.nio_http.bind_host

各プロパティの値を J2EE サーバ、または Web コンテナサーバごとにユニークな値に設定し直してください。

- webserver.container.server_id.value
- webserver.session.server_id.value

3. Management Server を使用している場合は、複製先のマシンで次の操作を実施します。

1. 複製で追加したホストの定義

2. 複製先ホスト上に定義されている複製元のホスト定義の削除 (Management Server が動作しているマシンを複製したときだけ)

3. 複製で追加した論理サーバの定義

4. 3.で追加した論理サーバのセットアップ (実サーバ名指定)

5. 3.で追加した論理サーバの設定ファイルの読み込み

6. 3.で追加した論理サーバの設定ファイルの配布

7. サーバ管理コマンドで EAR ファイル、および RAR ファイルを削除した後の再配布 (J2EE サーバのアプリケーション管理機能を使用しているときだけ)

(b) バッチアプリケーションを実行するシステム（バッチサーバ）使用時

バッチアプリケーションを実行するシステム（バッチサーバ）使用時、Component Container での注意事項を次に示します。

• 複製を実施する前にすること

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.および手順 2.を実施してください。Management Server を使用しているときは手順 3.も実施してください。

手順

1. 複製元のマシンで、次に示すプログラムを終了します。

- ・バッチサーバ（リソースアダプタを開始したまま終了しても構いません）
- ・運用管理エージェント
- ・Management Server

2. 複製元のマシンで、次のディレクトリ下にファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- ・/opt/Cosminexus/CC/admin/logs/*
- ・/opt/Cosminexus/CC/admin/logs/CC/maintenance/*
- ・/opt/Cosminexus/CC/admin/logs/CC/maintenance/mmap/*
- ・/opt/Cosminexus/CC/admin/logs/CC/rmi/*
- ・/opt/Cosminexus/CC/admin/logs/CC/rmi/mmap/*
- ・/opt/Cosminexus/CC/admin/logs/mmap/*
- ・/opt/Cosminexus/CC/admin/logs/TPB/logj/comtrc/*
- ・/opt/Cosminexus/CC/admin/logs/TPB/logj/mdltrc/*
- ・/opt/Cosminexus/CC/admin/logs/TPB/logj/*
- ・/opt/Cosminexus/CC/client/logs/*
- ・/opt/Cosminexus/CC/client/logs/mmap/*
- ・/opt/Cosminexus/CC/web/containers/<サーバ名>/logs/*
- ・/opt/Cosminexus/manager/spool/default/*
- ・/opt/Cosminexus/manager/spool/lsinfo/*
- ・/opt/Cosminexus/manager/tmp/*
- ・/opt/Cosminexus/manager/setup/log/*
- ・/opt/Cosminexus/manager/setup/log/maintenance/*"
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/watch/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/CC/maintenance/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/CC/rmi/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/connectors/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/TPB/logj/mdltrc/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/TPB/logj/comtrc/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/TPB/logj/*
- ・<作業ディレクトリ>/ejb/<サーバ名>/logs/*
- ・<Manager ログ出力ディレクトリ>/maintenance/*
- ・<Manager ログ出力ディレクトリ>/maintenance/mmap/*
- ・<Manager ログ出力ディレクトリ>/message/*
- ・<Manager ログ出力ディレクトリ>/message/mmap/*
- ・<Manager ログ出力ディレクトリ>/trace/*

- ・<Manager ログ出力ディレクトリ>/trace/mmap/*
- ・<Manager ログ出力ディレクトリ>/*
- ・<snapshot ログの出力先ディレクトリ>/*
- ・<監査ログ出力ディレクトリ>¥*
- ・<監査ログのメッセージログ出力ディレクトリ>¥*
- ・<監査ログのメッセージログ出力ディレクトリ>¥mmap¥*
- ・<監査ログの例外情報出力ディレクトリ>¥*
- ・<監査ログの例外情報出力ディレクトリ>¥mmap¥*

参考

各ディレクトリのデフォルトを次に示します。

- ・<作業ディレクトリ> : /opt/Cosminexus/CC/server/public
- ・<Manager ログ出力ディレクトリ> : /opt/Cosminexus/manager/log
- ・<snapshot ログの出力先ディレクトリ> : <Manager ログ出力ディレクトリ>/snapshot
- ・<監査ログ出力ディレクトリ> : /opt/Cosminexus/auditlog
- ・<監査ログのメッセージログ出力ディレクトリ> : /opt/Cosminexus/auditlog
- ・<監査ログの例外情報出力ディレクトリ> : /opt/Cosminexus/auditlog

3. Management Server を使用している場合は、複製元のマシンでプロパティファイルの設定を変更します。

複製元のマシンで Management Server が動作しているかどうかで作業が異なります。

複製元のマシンで Management Server が動作していないとき

複製先のマシンがマルチホームドホストのときは、あらかじめ、複製元のマシンの運用管理エージェントプロパティファイル (adminagent.properties) に、「adminagent.adapter.bind_host=<複製先の管理 IP アドレスと一致する正規表現>」を設定してください。

複製元のマシンで Management Server が動作しているとき

次の表に示すプロパティを設定してください。

ファイル名	プロパティ	値
adminagent.properties	adminagent.adapter.allowedHosts	mserver.properties の mngrsvr.myhost.name プロパティが設定されているときは、同じ値を指定してください。 mserver.properties の mngrsvr.myhost.name プロパティが設定されていないときは、ホスト用管理 IP アドレスと一致する正規表現を設定してください。
mserver.properties	mngrsvr.myhost.name	複製先のマシンがマルチホームドホストのときは、<複製先の管理 IP アドレスと一致する正規表現>を設定してください。
	webserver.connector.http.bind_host	

ファイル名	プロパティ	値
adminagent.properties	adminagent.adapter.bind_host	
.mngsvrutilrc および mngsvrutilcl.properties	mngsvrutil.connect.host	ループバックアドレス以外を設定するときは、<複製先の管理 IP アドレスと一致する正規表現>を設定してください。
	mngsvrutil.target_name	
.cmxrc および cmxclient.properties	cmx.connect.host	

- **複製を実施したあとにすること**

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.を実施してください。Management Server を使用しているときは手順 2.も実施してください。

手順

1. 次に示すプロパティをバッチサーバ用ユーザプロパティファイル (usrconf.properties)、およびサーバ管理コマンド用ユーザプロパティファイル (usrconf.properties) で設定している場合は、複製先のマシンで値を変更します。

各プロパティで適切な値に変更してください。

- vbroker.se.iiop_tp.host
- ejbserver.naming.host
- ejbserver.rmi.naming.host

2. Management Server を使用している場合は、複製先のマシンで次の操作を実施します。

1. 複製で追加したホストの定義

2. 複製先ホスト上に定義されている複製元のホスト定義の削除 (Management Server が動作しているマシンを複製したときだけ)

3. 複製で追加した論理サーバの定義

4. 3.で追加した論理サーバのセットアップ (実サーバ名指定)

5. 3.で追加した論理サーバの設定ファイルの読み込み

6. 3.で追加した論理サーバの設定ファイルの配布

7. サーバ管理コマンドで RAR ファイルを削除したとの再配布 (J2EE サーバのアプリケーション管理機能を使用しているときだけ)

(2) Component Container - Redirector での注意事項

Component Container - Redirector での注意事項を次に示します。

- **複製を実施する前にすること**

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.および手順 2.を実施してください。

手順

1. 複製元のマシンで、次に示すプログラムを終了します。

- HTTP Server

2. 複製元のマシンで、次のディレクトリ下にファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- /opt/Cosminexus/CCR/web/redirector/servers/<サーバ名>/logs/*
- /opt/Cosminexus/CCR/web/redirector/servers/<サーバ名>/logs/mmap/*
- /opt/Cosminexus/CCR/web/redirector/logs/*
- /opt/Cosminexus/CCR/web/redirector/logs/mmap/*

(3) Reliable Messaging での注意事項

Reliable Messaging での注意事項を次に示します。

• 複製を実施する前にすること

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1. および手順 2. を実施してください。

手順

1. J2EE サーバに Reliable Messaging をデプロイし、Reliable Messaging が開始している場合は、J2EE サーバ上で Reliable Messaging を停止します。

2. 複製元のマシンで、次のディレクトリ下にログファイルがある場合は削除します。ただし、ディレクトリは削除しないでください。

- <作業ディレクトリ>/ejb/<サーバ名>/logs/RM/maintenance
<作業ディレクトリ>のデフォルトは、/opt/Cosminexus/CC/server/public です。

• 複製を実施したあとにすること

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.～手順 5. を実施してください。

手順

1. 複製後のシステムでホスト名または IP アドレスを変更した場合は、次に示す Reliable Messaging の設定を更新します。

- 環境変数 HRM_CMD_HOST の設定を、複製後のシステムのホスト名または IP アドレスに更新してください。

• キュー間転送機能を使用し、転送先システムのホスト名または IP アドレスを変更したときは、次に示す手順で転送元システムに存在するあて先と、そのあて先を使用する転送キューを再作成してください。

1. 転送元システムのあて先を使用しているすべての転送キューにメッセージがないことを確認します。

2. 転送元システムのあて先を使用しているすべての転送キューを削除します。

3. 転送元システムのあて先を削除します。

4. 変更後のホスト名または IP アドレスを指定して、転送元システムのあて先を新規作成します。
 5. 転送元システムのあて先を使用する転送キューを新規作成します。
2. Reliable Messaging のシステム名が一意となるように、コンフィグレーションプロパティの RMSystemName と、環境変数 HRM_SYSTEM_NAME を変更します。
3. 手順 2.で変更したシステム名に対応する管理情報テーブルを作成します。
管理情報テーブルの作成方法については、マニュアル「Reliable Messaging」を参照してください。
4. DBMS に HiRDB を使用し、XA トランザクションを使用する構成の場合は、Reliable Messaging の連携先 DB Connector の XAOpenString が、同一 HiRDB に接続するリソースアダプタ間で一意になるように、連携先 DB Connector のコンフィグレーションプロパティの description と XAOpenString を変更します。
5. Reliable Messaging を開始して、必要なキューを作成します。

(4) Component Transaction Monitor での注意事項

Component Transaction Monitor での注意事項を次に示します。

- 複製を実施したあとにすること

複製を実施したあとに、複製先のマシンで環境変数 CTMSPOOL に設定したディレクトリ下を削除してください。

(5) Performance Tracer での注意事項

複製を実施したあとに、複製先のマシンで環境変数 PRFSPOOL に設定したディレクトリ下を削除してください。

(6) HTTP Server での注意事項

HTTP Server での注意事項を次に示します。

- 複製を実施する前にすること

複製を実施する前に、複製元のマシンで HTTP Server のすべてのプログラム、および関連ファイルへのすべての操作を終了してください。

- 複製を実施したあとにすること

複製を実施したあとに、複製先のマシンで実施する作業について説明します。

複製先のマシンで、手順 1.~手順 3.を実施してください。

手順

1. 必要に応じて、複製後のディレクティブの設定を変更します。

- ネットワーク関連のディレクティブ

ServerName, <VirtualHost>, BindAddress, Listen, NameVirtualHost

- ファイルシステム関連のディレクティブ

<Directory>, <DirectoryMatch>, AccessFileName, Alias, AliasMatch, AuthGroupFile, AuthUserFile, CoreDumpDirectory, CustomLog, DocumentRoot, ErrorLog, HWSRequestLog, HWSTraceIdFile, HWSTraceLogFile, Include, LoadFile, LoadModule, PidFile, QOSResponse, ScriptAlias, ScriptAliasMatch, ScriptLog, ServerRoot, SSLCACertificateFile, SSLCACertificatePath, SSLCacheServerPath, SSLCacheServerPort, SSLCacheServerRunDir, SSLCertificateFile, SSLCertificateKeyFile, SSLCertificateKeyPassword, SSLCRLDERPath, SSLCRLPEMPath, TransferLog, TypesConfig, UserDir

2. PidFile ディレクティブに指定されているファイルがコピーされている場合は、HTTP Server を起動する前に削除します。
3. ログ関連のファイルがコピーされている場合は、必要に応じて、不要なファイルを削除します。

(7) TPBroker での注意事項

TPBroker での注意事項を次に示します。

- **複製を実施したあとにすること**

複製を実施したあとに、実施する作業について説明します。手順 1.～手順 2.を実施してください。

手順

1. 仮想化プラットフォームが提供するイメージファイル化による複製機能を使用する場合に、次の設定があるときは、指定したホスト名や IP アドレスをシステムに合わせて変更します。

- agentaddr ファイル（環境変数 OSAGENT_ADDR_FILE またはプロパティ vbroker.agent.addrFile でも指定できます）
- localaddr ファイル（環境変数 OSAGENT_LOCAL_FILE またはプロパティ vbroker.agent.localFile でも指定できます）
- htc.clienthandleraddr ファイル（環境変数 HVI_OSGENT_CLIENTHANDLERADDR_FILE でも指定できます）
- プロパティ vbroker.agent.addr
- プロパティ vbroker.se.<サーバエンジン>.host
- プロパティ vbroker.se.<サーバエンジン>.proxyHost
- プロパティ vbroker.orb.initRef
- プロパティ vbroker.orb.defaultInitRef (C++だけ)
- コマンドラインオプション ORBInitRef
- コマンドラインオプション ORBDefaultInitRef (C++だけ)
- コマンドラインオプション ORBagentAddr
- osagent のコマンドラインオプション -a
- 環境変数 OSAGENT_ADDR

また、ホスト名や IP アドレスを変更する前に作成した IOR ファイルは、変更後に再作成し、再配布してください。

2. 仮想化環境での複製機能を使用し、かつグローバルトランザクションを使用する場合、仮想化環境の複製元にインプロセストランザクションサービスのステータスファイルを含まないように、インプロセストランザクションサービスのステータスファイルがあるときは削除します。

(8) Service Coordinator での注意事項

Service Coordinator での注意事項を次に示します。

・複製を実施する前にすること

複製を実施する前に、複製元のマシンで実施する作業について説明します。

複製元のマシンで、手順 1.～手順 2.を実施してください。

手順

1. マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「実行環境に必要なソフトウェアを設定する」までの作業を実施します。

2. 次に示す構成ソフトウェアの「複製を実施する前にすること」の作業を実施します。

- ・ Component Container
- ・ Reliable Messaging
- ・ HTTP Server
- ・ TPBroker

・複製を実施したあとにすること

複製を実施したあとに、実施する作業について説明します。手順 1.～手順 2.を実施してください。

手順

1. 複製先のマシンで、次に示す構成ソフトウェアの「複製を実施したあとにすること」の作業を実施します。

- ・ Component Container
- ・ Reliable Messaging
- ・ Component Transaction Monitor
- ・ Performance Tracer
- ・ HTTP Server
- ・ TPBroker

2. 複製元のマシン、および複製先のマシンで、マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「HCSC サーバをセットアップする」以降の作業を実施します。

付録 I.5 リモートインストールをするときの注意事項

リモートインストールをするときの注意事項を次の表に示します。

表 I-1 リモートインストールをするときの注意事項

分類	注意事項	新規インストール	更新インストール
Windows	各製品の構成ソフトウェアをリモートインストールする場合は、構成ソフトウェアのリモートインストールの前に、必ず製品のリモートインストールを実施してください。構成ソフトウェアのリモートインストールは、製品のインストール先の指定に従って実施されます。	○	○
	新規インストールするインストール先は、指定によって次のようにになります。 ドライブとパスの両方とも省略（デフォルト値を使用）している場合 <システムがインストールされているドライブ名>¥Program Files¥Hitachi¥Cosminexus ディレクトリを作成して、インストールします。 ドライブを指定した場合 指定したドライブに¥Program Files¥Hitachi¥Cosminexus ディレクトリを作成して、インストールします。 パスを指定した場合 指定したパスにインストールします。	○	—
	インストール済みの製品と同じディレクトリに更新インストールします。	—	○

(凡例) ○：該当します。—：該当しません。

付録 I.6 Application Server のインストールおよびアンインストール時のエラーコード一覧 (Windows の場合)

ここでは、インストールおよびアンインストール時の障害を通知するコードについて説明します。

・製品固有のリターンコードおよび構成ソフトウェア固有のエラーコード

製品および構成ソフトウェアの固有のコードの意味と対処を次の表に示します。次の表にコードがない場合は、「[表 I-3 製品および構成ソフトウェア共通のコードの意味と対処](#)」に示す製品共通のリターンコードおよび構成ソフトウェア共通のエラーコードを参照してください。なお、リターンコードおよびエラーコードが取得できない場合は、「**」が出力されます。

表 I-2 製品および構成ソフトウェアの固有のコードの意味と対処

製品または構成ソフトウェアの名称	コード	意味	対処
Application Server	04	Application Server のインストールは正常終了しましたが、製品の修正パッチ適用時に障害が発生しました。	<Application Server のインストールディレクトリ>¥install.log の "PATCH END" のあとに出力されている修正パッチ適用時のエラーコードを確認し、「 表 I-4 修正パッチ適用時のエラーコードの意味と対処 」に示す対処を実施してください。
	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。

製品または構成ソフトウェアの名称	コード	意味	対処
	91	未サポートのプラットフォームでインストールを実行しました。	Application Server がサポートしているプラットフォームへインストールしてください。
	92	Application Server 以外の製品がインストール済みです。	インストール済みの Application Server 以外の製品をアンインストールしたあとに, Application Server をインストールしてください。
	93	新しいバージョンの Application Server がインストール済みです。	インストールしようとしている Application Server のバージョンを確認してください。
	94	共用モジュールのファイル転送で障害が発生しました。	保守員に連絡してください。
	95	構成ソフトウェアのインストーラの起動で障害が発生しました。	保守員に連絡してください。
	96	構成ソフトウェアのインストーラの処理で障害が発生しました。	構成ソフトウェアのエラーコードを参照して原因を取り除いたあとに再度インストールを実行してください。
	97	その他の障害が発生しました。	HiRDB/Single Server Version 10 のインストールおよびアンインストールで障害が発生したときは, 「97,XX,YY」の形式でリターンコードを出力することがあります。YY は, HiRDB/Single Server Version 10 のエラーコードを参照してください。 上記以外のときは, 保守員に連絡してください。
	98	インストール先ディレクトリ名の文字数が制限値を超えるました。	インストール先ディレクトリ名は半角 50 文字以下にしてください。
	99	Application Server のアンインストール処理でタイムアウトが発生しました。	保守員に連絡してください。
Client	04	Client のインストールは正常終了しましたが, 製品の修正パッチ適用時に障害が発生しました。	<Client のインストールディレクトリ>install.log の"PATCH END"のあとに出力されている修正パッチ適用時のエラーコードを確認し, 「表 I-3 製品および構成ソフトウェア共通のコードの意味と対処」に示す対処を実施してください。
	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
	91	未サポートのプラットフォームでインストールを実行しました。	Client がサポートしているプラットフォームへインストールしてください。
	92	Client 以外の製品がインストール済みです。	インストール済みの Client 以外の製品をアンインストールしたあとに, Client をインストールしてください。

製品または構成ソフトウェアの名称	コード	意味	対処
Client	93	新しいバージョンの Client がインストール済みです。	インストールしようとしている Client のバージョンを確認してください。
	94	共用モジュールのファイル転送で障害が発生しました。	保守員に連絡してください。
	95	構成ソフトウェアのインストーラの起動で障害が発生しました。	保守員に連絡してください。
	96	構成ソフトウェアのインストーラの処理で障害が発生しました。	構成ソフトウェアのエラーコードを参照して原因を取り除いたあとに再度インストールを実行してください。
	97	その他の障害が発生しました。	HiRDB/Single Server Version 10 のインストールおよびアンインストールで障害が発生したときは、「97,XX,YY」の形式でリターンコードを出力することがあります。YY は、HiRDB/Single Server Version 10 のエラーコードを参照してください。 上記以外のときは、保守員に連絡してください。
	98	インストール先ディレクトリ名の文字数が制限値を超えるました。	インストール先ディレクトリ名は半角 50 文字以下にしてください。
	99	Client のアンインストール処理でタイムアウトが発生しました。	保守員に連絡してください。
	04	Developer のインストールは正常終了しましたが、製品の修正パッチ適用時に障害が発生しました。	<Developer のインストールディレクトリ>install.log の"PATCH END"のあとに出力されている修正パッチ適用時のエラーコードを確認し、「表 I-3 製品および構成ソフトウェア共通のコードの意味と対処」に示す対処を実施してください。
	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
Developer	91	未サポートのプラットフォームでインストールを実行しました。	Developer がサポートしているプラットフォームへインストールしてください。
	92	Developer 以外の製品がインストール済みです。	インストール済みの Developer 以外の製品をアンインストールしたあとに、Developer をインストールしてください。
	93	新しいバージョンの Developer がインストール済みです。	インストールしようとしている Developer のバージョンを確認してください。
	94	共用モジュールのファイル転送で障害が発生しました。	保守員に連絡してください。
	95	構成ソフトウェアのインストーラの起動で障害が発生しました。	保守員に連絡してください。

製品または構成ソフトウェアの名称	コード	意味	対処
	96	構成ソフトウェアのインストーラの処理で障害が発生しました。	構成ソフトウェアのエラーコードを参照して原因を取り除いたあとに再度インストールを実行してください。
	97	その他の障害が発生しました。	HiRDB/Single Server Version 10 のインストールおよびアンインストールで障害が発生したときは、「97,XX,YY」の形式でリターンコードを出力することがあります。YY は、HiRDB/Single Server Version 10 のエラーコードを参照してください。 上記以外のときは、保守員に連絡してください。
	98	インストール先ディレクトリ名の文字数が制限値を超えるました。	インストール先ディレクトリ名は半角 50 文字以下にしてください。
	99	Developer のアンインストール処理でタイムアウトが発生しました。	保守員に連絡してください。
Service Architect	04	Service Architect のインストールは正常終了しましたが、製品の修正パッチ適用時に障害が発生しました。	<Service Architect のインストールディレクトリ>¥install.log の "PATCH END" のあとに出力されている修正パッチ適用時のエラーコードを確認し、「表 I-3 製品および構成ソフトウェア共通のコードの意味と対処」に示す対処を実施してください。
	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
	91	未サポートのプラットフォームでインストールを実行しました。	Service Architect がサポートしているプラットフォームへインストールしてください。
	92	Service Architect 以外の製品がインストール済みです。	インストール済みの Service Architect 以外の製品アンインストールしたあとに、Service Architect をインストールしてください。
	93	新しいバージョンの Service Architect がインストール済みです。	インストールしようとしている Service Architect のバージョンを確認してください。
	94	共用モジュールのファイル転送で障害が発生しました。	保守員に連絡してください。
	95	構成ソフトウェアのインストーラの起動で障害が発生しました。	保守員に連絡してください。
	96	構成ソフトウェアのインストーラの処理で障害が発生しました。	構成ソフトウェアのエラーコードを参照して原因を取り除いたあとに再度インストールを実行してください。
	97	その他の障害が発生しました。	HiRDB/Single Server Version 10 のインストールおよびアンインストールで障害が発生したときは、「97,XX,YY」の形式でリターンコードを出力することがあります。YY は、

製品または構成ソフトウェアの名称	コード	意味	対処
			HiRDB/Single Server Version 10 のエラーコードを参照してください。 上記以外のときは、保守員に連絡してください。
	98	インストール先ディレクトリ名の文字数が制限値を超えるました。	インストール先ディレクトリ名は半角 50 文字以下にしてください。
	99	Service Architect のアンインストール処理でタイムアウトが発生しました。	保守員に連絡してください。
Service Platform	04	Service Platform のインストールは正常終了しましたが、製品の修正パッチ適用時に障害が発生しました。	<Service Platform のインストールディレクトリ>¥install.log の"PATCH END"のあとに出力されている修正パッチ適用時のエラーコードを確認し、「表 I-3 製品および構成ソフトウェア共通のコードの意味と対処」に示す対処を実施してください。
	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
	91	未サポートのプラットフォームでインストールを実行しました。	Service Platform がサポートしているプラットフォームへインストールしてください。
	92	Service Platform 以外の製品がインストール済みです。	インストール済みの Service Platform 以外の製品をアンインストールしたあとに、Service Platform をインストールしてください。
	93	新しいバージョンの Service Platform がインストール済みです。	インストールしようとしている Service Platform のバージョンを確認してください。
	94	共用モジュールのファイル転送で障害が発生しました。	保守員に連絡してください。
	95	構成ソフトウェアのインストーラの起動で障害が発生しました。	保守員に連絡してください。
	96	構成ソフトウェアのインストーラの処理で障害が発生しました。	構成ソフトウェアのエラーコードを参照して原因を取り除いたあとに再度インストールを実行してください。
	97	その他の障害が発生しました。	HiRDB/Single Server Version 10 のインストールおよびアンインストールで障害が発生したときは、「97,XX,YY」の形式でリターンコードを出力することがあります。YY は、HiRDB/Single Server Version 10 のエラーコードを参照してください。 上記以外のときは、保守員に連絡してください。
	98	インストール先ディレクトリ名の文字数が制限値を超えるました。	インストール先ディレクトリ名は半角 50 文字以下にしてください。

製品または構成ソフトウェアの名称	コード	意味	対処
	99	Service Platform のアンインストール処理でタイムアウトが発生しました。	保守員に連絡してください。
Application Development Plug-in	90	新しいバージョンの製品がインストール済みです。	インストールしようとしている製品のバージョンを確認してください。
Component Container	8C	<p>次のどちらかの理由でインストールが拒否されました。</p> <ol style="list-style-type: none"> インストーラを使用しないで、個別の製品のインストーラを直接起動した場合 管理者権限のないユーザでセットアップを実行した場合 	<p>1.の場合 インストーラのセットアッププログラム (HCD_INST.EXE) を使用してください。</p> <p>2.の場合 管理者権限のあるユーザでログインして、再度セットアップを実行してください。</p>
	90	Cosminexus Server Component Container for Java 01-00 または 01-01 がインストール済みです。	Cosminexus Server Component Container for Java 01-00 または 01-01 をアンインストールしたあとに、インストールを実行してください。
	91	Component Server for Java 01-00 がインストール済みです。	Component Server for Java 01-00 をアンインストールしたあとに、インストールを実行してください。
	92	<p>バージョンダウンの可能性があります。</p> <ul style="list-style-type: none"> 新しいバージョンの製品がインストール済みの場合 評価版がインストールされている状態で、正規版のインストールを実行した場合 	すでにインストールされている製品のバージョンと、インストールする製品のバージョンを確認してください。
	93	コマンドの実行に失敗しました。	保守員に連絡してください。
	95	サービスの起動を検出しました。	起動している製品のサービスをすべて停止させ、再度セットアップを実行してください。
	97	実行中のプログラムまたはファイルを検出しました。	起動している Component Container のプログラムと、Web サーバなどの関連プログラムをすべて停止させて、再度セットアップを実行してください。
	98	OS の再起動後に更新予約されているが、現在未更新のプログラムまたはファイルを検出しました。	OS を再起動し、プログラムまたはファイルが更新されてから、再度セットアップを実行してください。
	99	SI Navigation System - Agent がインストールされています。	SI Navigation System - Agent をアンインストールしたあとに、再度インストールを実行してください。
Component Container - Client	8C	次のどちらかの理由でインストールが拒否されました。	<p>1.の場合 インストーラのセットアッププログラム (HCD_INST.EXE) を使用してください。</p>

製品または構成ソフトウェアの名称	コード	意味	対処
		1. インストーラを使用しないで、個別の製品のインストーラを直接起動した場合 2. 管理者権限のないユーザでセットアップを実行した場合	2.の場合 管理者権限のあるユーザでログインして、再度セットアップを実行してください。
	90	Cosminexus Server Component Container for Java 01-00 または 01-01 がインストール済みです。	Cosminexus Server Component Container for Java 01-00 または 01-01 をアンインストールしたあとに、インストールを実行してください。
	91	Component Server for Java 01-00 がインストール済みです。	Component Server for Java 01-00 をアンインストールしたあとに、インストールを実行してください。
	92	バージョンダウンの可能性があります。 • 新しいバージョンの製品がインストール済みの場合 • 評価版がインストールされている状態で、正規版のインストールを実行した場合	すでにインストールされている製品のバージョンと、インストールする製品のバージョンを確認してください。
	97	実行中のプログラムまたはファイルを検出しました。	起動している Component Container のプログラムと、Web サーバなどの関連プログラムをすべて停止させて、再度セットアップを実行してください。
	98	OS の再起動後に更新予約されているが、現在未更新のプログラムまたはファイルを検出しました。	OS を再起動し、プログラムまたはファイルが更新されてから、再度セットアップを実行してください。
	99	SI Navigation System - Agent がインストールされています。	SI Navigation System - Agent をアンインストールしたあとに、再度インストールしてください。
Component Container - Redirector	8C	次のどちらかの理由でインストールが拒否されました。 1. インストーラを使用しないで、個別の製品のインストーラを直接起動した場合 2. 管理者権限のないユーザでセットアップを実行した場合	1.の場合 インストーラのセットアッププログラム (HCD_INST.EXE) を使用してください。 2.の場合 管理者権限のあるユーザでログインして、再度セットアップを実行してください。
	90	Cosminexus Server Component Container for Java 01-00 または 01-01 がインストール済みです。	Cosminexus Server Component Container for Java 01-00 または 01-01 をアンインストールしたあとに、インストールを実行してください。
	91	Component Server for Java 01-00 がインストール済みです。	Component Server for Java 01-00 をアンインストールしたあとに、インストールを実行してください。

製品または構成ソフトウェアの名称	コード	意味	対処
	92	バージョンダウングレードの可能性があります。 <ul style="list-style-type: none"> 新しいバージョンの製品がインストール済みの場合 評価版がインストールされている状態で、正規版のインストールを実行した場合 	すでにインストールされている製品のバージョンと、インストールする製品のバージョンを確認してください。
	97	実行中のプログラムまたはファイルを検出しました。	起動している Component Container - Redirecter のプログラムと、Web サーバなどの関連プログラムをすべて停止させて、再度セットアップを実行してください。
	98	OS の再起動後に更新予約されているが、現在未更新のプログラムまたはファイルを検出しました。	OS を再起動し、プログラムまたはファイルが更新されてから、再度セットアップを実行してください。
	99	SI Navigation System - Agent がインストールされています。	SI Navigation System - Agent をアンインストールしたあとに、再度インストールしてください。
Component Transaction Monitor	—	固有なエラーコードはありません。	—
Developer's Kit for Java	90	新しいバージョンの Developer's Kit for Java がインストール済みです。	Developer's Kit for Java をアンインストールしたあとに、インストールを実行してください。
	91	ファイルのコピーに失敗しました。	インストール先のディスク容量が不足していないか、アクセス権限が正しく設定されているかを確認してください。
	97	実行中のプログラムまたはファイルを検出しました。	起動している Developer's Kit for Java のプログラムと、関連するファイルにアクセスしているすべてのプログラムを終了させて、再度インストールを実行してください。
HTTP Server	83	HTTP Server のサービスが動作中です。または、レジストリ操作でエラーが発生しました。	HTTP Server が動作中かどうかを確認してください。 動作中の場合は、停止して再操作してください。 動作していない場合は、保守員に連絡してください。
	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
	91	未サポートのプラットフォームでインストールを実行しました。	製品がサポートしているプラットフォームへインストールしてください。

製品または構成ソフトウェアの名称	コード	意味	対処
	94	単体版の Web Server がインストール済みの環境へ、製品をインストールしました。	単体版の Web Server をアンインストールしてから、製品をインストールしてください。
Performance Tracer	—	固有なエラーコードはありません。	—
Reliable Messaging	90	バージョンチェックでエラーが発生しました。	インストール済みの Reliable Messaging をアンインストールしたあと、再度インストールを実行してください。
	91	ファイル転送でエラーが発生しました。	再度インストールを実行してください。
Service Coordinator	90	ダウングレードインストールです。	インストールしようとしている製品のバージョンを確認してください。
Service Development Plug-in	90	新しいバージョンの製品がインストール済みです。	インストールしようとしている製品のバージョンを確認してください。
TPBroker	90	新しいバージョンの製品がインストール済みです。	インストールしようとしている製品のバージョンを確認してください。 再度インストールが必要な場合は、インストールされている製品と同じバージョン、または新しいバージョンの製品をインストールしてください。
	91	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
	93	内部エラーが発生しました。	すべてのプログラムを終了したあと、またはOSの再起動後、再度インストールを実行してください。
	94	旧バージョンで提供していたファイルの削除に失敗しました。	すべてのプログラムを終了後、またはOSの再起動後、再度インストールを実行してください。
	95	レジストリ操作で異常が発生しました。	すべてのプログラムを終了後、またはOSの再起動後、再度インストールを実行してください。
	96	システム環境の調査で異常が発生しました。	すべてのプログラムを終了後、またはOSの再起動後、再度インストールを実行してください。
	97	ファイルのコピーに失敗しました。	ディスク容量が不足していないか、アクセス権限が正しく設定されているかを確認してください。 問題がない場合、すべてのプログラムを終了したあと、またはOSの再起動後、再度インストールを実行してください。

製品または構成ソフトウェアの名称	コード	意味	対処
	98	レジストリ情報の設定に失敗しました。	すべてのプログラムを終了したあと、またはOSの再起動後、再度インストールを実行してください。
	99	媒体が不正です。	すべてのプログラムを終了したあと、またはOSの再起動後、再度インストールを実行してください。
Web Services - Security	90	管理者権限のないユーザがインストールまたはアンインストールを実行しました。	管理者権限のあるユーザがインストールしてください。
	93	新しいバージョンの製品がインストール済みです。	インストールしようとしている製品のバージョンを確認してください。
XML Processor	90	新しいバージョンの製品がインストール済みです。	インストールしようとしている製品のバージョンを確認してください。
HiRDB/Single Server Version 10	92	適用 OS ではありません。	製品がサポートしているプラットフォームへインストールしてください。
	94	同じセットアップ識別子 (_CS0) の組み込み HiRDB のサービスが停止状態ではありません。	起動済みのサービスを停止してからインストールを実行してください。
	95	同じセットアップ識別子 (_CS0) の組み込み HiRDB がインストールされています。	保守員に連絡してください。
	96	同じセットアップ識別子 (_CS0) の組み込み HiRDB がすでにインストールされています。	保守員に連絡してください。
	9C	実行中のプロセスが残っています。	インストール時は、OSを再起動して再実行してください。 アンインストール時は、アンインストール後、直ちにOSを再起動してください。

(凡例) - : 該当しません。

注 製品と、製品に含まれる構成ソフトウェアについては、「2.2.1(3) 製品に対応する構成ソフトウェアについて」を参照してください。

・ 製品共通のリターンコードおよび構成ソフトウェア共通のエラーコード

製品および構成ソフトウェアの共通のコードの意味と対処を次の表に示します。

表 I-3 製品および構成ソフトウェア共通のコードの意味と対処

コード	意味	対処
00	正常終了しました。	対処はありません。
01	正常終了しました。次の処理が該当します。 • 異なるバージョンへの更新インストール (バージョンアップ)	対処はありません。

コード	意味	対処
	<ul style="list-style-type: none"> アンインストール済みの環境に対するアンインストール 	
82	<p>次のインストール後処理でエラーが発生しました。</p> <ul style="list-style-type: none"> アイコン追加処理 グループ追加処理 	保守員に連絡してください。
84	インストール時にメモリ不足が発生しました。	ほかのプログラムを終了させるなど、メモリを確保してから、再度実行してください。
85	ディスクのアクセスエラーが発生しました。	保守員に連絡してください。
88	<p>起動パラメタが誤っています。</p> <p>更新インストールが指示されたが、該当するプログラムがインストールされていないなどの要因が考えられます。</p>	<p>更新インストールで該当するプログラムがインストールされていない場合は、新規インストールで再度実行してください。</p> <p>上記以外の場合は、保守員に連絡してください。</p>
8A	構成ソフトウェアの新規インストールができません。	保守員に連絡してください。
8C	<p>インストールが拒否されました。</p> <p>前提プログラムがないなどの要因が考えられます。</p>	<p>前提プログラムがない場合は、前提となるプログラムをインストールしたあと、再度実行してください。</p> <p>上記以外の場合は、保守員に連絡してください。</p>
80, 81, 87, 89, 8B, 8D, 8E, 8F	インストールが異常終了しました。	保守員に連絡してください。
9A	インストールがキャンセルされました。	対処はありません。
9B	対応する製品がインストールされていません。	製品をインストールしたあと、再度リモートインストールを実行してください。
9C	製品情報の取得に失敗しました。	製品が正しくインストールされているかを確認したあと、再度インストールを実行してください。
9D	インストール履歴ファイル、またはインストール状態ファイルの出力に失敗しました。	ディスク容量が不足していないか、アクセス権限が正しく設定されているかを確認してください。これらの問題がない場合、すべてのプログラムを終了したあと、またはOSの再起動後、再度インストールを実行してください。

・修正パッチ適用時のエラーコード

アプリケーションサーバおよびBPM/ESB基盤の製品では、製品をインストールすると、インストールが正常終了したあとにその製品の修正パッチが適用されます。修正パッチ適用時にエラーが発生した場合は、<製品のインストールディレクトリ>¥install.logの"PATCH END"のあとに、エラーコードが出力されます。修正パッチ適用時のエラーコードの意味と対処を次の表に示します。

表I-4 修正パッチ適用時のエラーコードの意味と対処

コード	意味	対処
0 または 1	適用する修正パッチはありません。または、修正パッチは正常に適用されました。	対処は不要です。

コード	意味	対処
129	修正パッチ適用時にメモリ不足が発生しました。	ほかのプログラムを終了させるなど、メモリを確保してから、再度実行してください。
134	修正パッチ適用時にディスク容量不足が発生しました。	ハードディスク容量を確保してから、再度実行してください。
144	プログラムまたはサービスが実行中のため、修正パッチ適用に失敗しました。	製品に関連するプログラムまたはサービスを停止してから、再度実行してください。 停止後に実行しても同じエラーになる場合は、製品の提供媒体が不正な可能性があるので、保守員へ連絡してください。
上記以外	製品の提供媒体が不正な可能性があります。	保守員へ連絡してください。

付録 I.7 Component Container およびリダイレクタ機能のインストールおよびアンインストール時のログ (UNIX の場合)

Component Container およびリダイレクタ機能のインストールおよびアンインストールの実行状況は、次に示すインストールログに出力されます。

- Component Container の場合 : /opt/Cosminexus/CC/logs/install.log
- リダイレクタ機能の場合 : /opt/Cosminexus/CCR/logs/install.log

インストールログは、最大で 4 面 install1.log~install4.log のバックアップを作成し、最新のインストールログは常に install.log になります。

インストールおよびアンインストール時にエラーが発生した場合は、インストールログの内容を基に対処してください。エラーの原因が突き止められない場合は、次の表に示すファイルを保守員に送付して、お問い合わせください。

表 I-5 インストールおよびアンインストールでのエラー発生時に送付するファイル

出力対象	エラーの発生タイミング	
	インストール時	アンインストール時
Component Container	<ul style="list-style-type: none"> /opt/Cosminexus/CC/logs/install.log (バックアップファイル含む) /etc/.hitachi/.install.log (バックアップファイル含む) 	<ul style="list-style-type: none"> /opt/Cosminexus/CC/logs/install.log (バックアップファイル含む) /etc/.hitachi/.uninstall.log (バックアップファイル含む)
リダイレクタ機能	<ul style="list-style-type: none"> /opt/Cosminexus/CCR/logs/install.log (バックアップファイル含む) /etc/.hitachi/.install.log (バックアップファイル含む) 	<ul style="list-style-type: none"> /opt/Cosminexus/CCR/logs/install.log (バックアップファイル含む) /etc/.hitachi/.uninstall.log (バックアップファイル含む)

- 出力形式

インストールログのフォーマットを次に示します。

```
MM-DD-YYYY/hh:mm:ss/rc=y/msg=zz...zz
```

• 出力項目

インストールログのフォーマットで示した各項目について説明します。

表 I-6 出力項目 (Component Container およびリダイレクタ機能のインストールログ)

出力項目	意味
MM	月 (01~12)
DD	日 (01~31)
YYYY	西暦 (2003 以降)
hh	時間 (00~23)
mm	分 (00~59)
ss	秒 (00~59)
rc	リターンコードを示すラベル
y	リターンコードの内容 (10 進数)
msg	ログメッセージの内容を示すラベル
zz...zz	トレース情報およびエラー情報の内容 <ul style="list-style-type: none">接頭辞が[I:]の場合：トレース情報接頭辞が[E:]の場合：エラー情報

• トレース情報およびエラー情報の内容

トレース情報およびエラー情報の内容について説明します。トレース情報およびエラー情報の説明で使用する可変値を次の表に示します。

表 I-7 トレース情報およびエラー情報の説明で使用する可変値

可変値	内容
aa...aa	<ul style="list-style-type: none">started : 開始ended : 終了
bb...bb	バンドル名称 <ul style="list-style-type: none">Cosminexus Component ContainerCosminexus Component Container - Redirector
cc...cc	コマンドの名称
dd...dd	ファイルの名称
ee...ee	ディレクトリの名称
ff...ff	ファイルまたはディレクトリの名称
gg...gg	追加する定義の内容

可変値	内容
hh...hh	旧バージョン（システムにセットアップ済みのバージョン）
ii...ii	旧 SI バージョン（システムにセットアップ済みの SI バージョン）
jj...jj	新バージョン（システムにセットアップする製品のバージョン）
kk...kk	新 SI バージョン（システムにセットアップする製品の SI バージョン）
ll...ll	情報 1（引数などの付加情報）
mm...mm	情報 2（引数などの付加情報）
nn...nn	情報 3（引数などの付加情報）
oo...oo	コマンドの戻り値
rr...rr	サービス名（Component Container の場合だけ出力）

出力項目 zz...zz のトレース情報の内容と意味を次の表に示します。

表 I-8 トレース情報の内容と意味 (Component Container およびリダイレクタ機能のインストールログ)

項目番	γ の内容	zz...zz の内容	意味
1	0	I:Setup of HNTRLib2 has aa...aa.	HNTRLib2 のセットアップ処理を aa...aa しました。
2	0	I:Extraction processing for HNTRLib2 has aa...aa.	HNTRLib2 の展開処理を aa...aa しました。
3	0	I:Installation of HNTRLib2 has aa...aa.	HNTRLib2 のインストール処理を aa...aa しました。
4	0	I:Registration processing for the HNTRLib2 daemon has aa...aa.	HNTRLib2 へのデーモン登録処理を aa...aa しました。
5	0	I:Startup processing of the HNTRLib2 daemon has aa...aa.	HNTRLib2 のデーモン起動処理を aa...aa しました。
6	0	I:Registration of the bundled program product bb...bb into HNTRLib2 has aa...aa.	HNTRLib2 へバンドル PP 名称[bb...bb]の登録を aa...aa しました。
7	0	I:The command has aa...aa. (Command = cc...cc)	コマンド cc...cc を aa...aa しました。
8	0	I:The file to be copied already exists in the copy destination. (file = dd...dd)	コピー指定されたファイルと同名のファイルがすでにコピー先に存在しました。
9	0	I:The directory to be copied already exists in the copy destination. (directory = ee...ee)	コピー指定されたディレクトリと同名のディレクトリがすでにコピー先に存在しました。
10	0	I:The directory to be created already exists. (directory = ee...ee)	作成指示されたディレクトリは、すでに存在しました。
11	0	I:The file or directory to be deleted does not exist. (file or directory = ff...ff)	削除指示されたファイルまたはディレクトリはすでに存在していません。

項目番	γ の内容	zz...zz の内容	意味
12	0	I:The file or directory to be copied does not exist. (file or directory = ff...ff)	コピーしようとしたファイルまたはディレクトリは存在していません。
13	0	I:The specified directory does not exist. (directory = ee...ee)	指定されたディレクトリは存在していません。
14	0	I:The specified definition already exists. [File=dd...dd,Define=gg...gg]	追加指定された定義 gg...gg は、すでにファイル dd...dd に存在します。
15	0	I:The specified bundle name is already registered. (bundle name = bb...bb)	指定されたバンドル名称 bb...bb はすでに登録されています。
16	0	I:The specified name has already been de-registered. (bundle name = bb...bb)	指定されたバンドル名称 bb...bb はすでに登録解除されています。
17	0	I:HNTRLib2 was not uninstalled because other applications are registered in it.	HNTRLib2 に、ほかの PP が登録されているため、HNTRLib2 のアンインストールは実行されませんでした。
18	0	I:Cancellation processing of the HNTRLib2 daemon has aa...aa.	HNTRLib2 からのデーモン登録解除処理を aa...aa しました。
19	0	I:Uninstallation of HNTRLib2 has aa...aa.	HNTRLib2 のアンインストールを aa...aa しました。
20	0	I:An overwrite installation will be performed. (version = hh...hh, SI version = ii...ii)	バージョン hh...hh・SI バージョン ii...ii の上書きインストールを実行します。
21	0	I:An upgrade installation will be performed. (earlier version = hh...hh new version = jj...jj, new SI version = kk...kk)	旧バージョン hh...hh に対して、新バージョン jj...jj・SI バージョン kk...kk の上書きインストールを実行します。
22	0	I:An upgrade installation will be performed. (earlier version = hh...hh, earlier SI version = ii...ii, new version = jj...jj, new SI version = kk...kk)	旧バージョン hh...hh・SI バージョン ii...ii に対して、新バージョン jj...jj・SI バージョン kk...kk の上書きインストールを実行します。
23	0	I:Version information was acquired from the file dd...dd. (version = hh...hh)	ファイル dd...dd からバージョン情報 hh...hh を取得しました。
24	0	I:Version information was acquired from the file dd...dd. (version = hh...hh, SI version = ii...ii)	ファイル dd...dd からバージョン情報 hh...hh・SI バージョン ii...ii を取得しました。
25	0	I:The shift processing to a version jj...jj from a version hh...hh is aa...aa.	旧バージョン hh...hh から新バージョン jj...jj へ移行処理を aa...aa しました。
26	0	I:The upgrade from the earlier version to the new version was not performed because the earlier version had already been uninstalled. (earlier version = jj...jj, new version = hh...hh)	旧バージョンのアンインストールが実行済みだったため、旧バージョン jj...jj から新バージョン hh...hh へのアップグレードは実行されません。
27	oo...oo	I:The command has executed. (command = cc...cc)	コマンド cc...cc を実行しました。

出力項目 zz...zz のエラー情報の内容と意味および対処を次の表に示します。

表 I-9 エラー情報の内容と意味および対処 (Component Container およびリダイレクタ機能のインストールログ)

項目番	γ の 内容	zz...zz の内容	意味	対処
1	1	E:The specified file does not exist. (file = dd...dd)	編集などの対象として指定されたファイルは存在しませんでした。	適切な権限で実行しているか確認し、再度セットアップを実行してください。
2	1	E:The specified directory does not exist. (directory = ee...ee)	指定されたディレクトリは存在しないディレクトリでした。	適切な権限で実行しているか確認し、再度セットアップを実行してください。
3	92	E:A downgrade installation cannot be performed. (installed version = hh...hh, installed SI version = ii...ii, version to be installed = jj...jj, SI version to be installed = kk...kk)	バージョンダウンインストールはできません。すでにインストールされているバージョン = hh...hh, SI バージョン ii...ii, セットアップしようとしている新バージョン = jj...jj, SI バージョン = kk...kk です。	セットアップしようとしている製品より、新しいバージョンの製品がすでにインストールされています。セットアップする製品のバージョンを確認してください。古いバージョンの製品をインストールするには、現在インストールされている新しいバージョンの製品をアンインストールしてからインストールしてください。
4	92	E:A downgrade installation cannot be performed. (installed version = hh...hh, new version to be installed = jj...jj, new SI version to be installed = kk...kk)	バージョンダウンインストールはできません。すでにインストールされているバージョン = hh...hh, セットアップしようとしている新バージョン = jj...jj, SI バージョン = kk...kk です。	セットアップしようとしている製品より、新しいバージョンの製品がすでにインストールされています。セットアップする製品のバージョンを確認してください。古いバージョンの製品をインストールするには、現在インストールされている新しいバージョンの製品をアンインストールしてからインストールしてください。
5	92	E:Upgrade of this version is not supported. (installed version = hh...hh, new version to be installed = jj...jj, new SI version to be installed = kk...kk)	すでにインストールされているバージョン = hh...hh からセットアップしようとしている新バージョン = jj...jj, SI バージョン = kk...kk へのアップグレードはサポートされていません。	アップグレードをサポートしていない旧製品が存在します。この製品をセットアップする前に旧製品をアンインストールして、再度セットアップを実行してください。
6	92	E:Upgrading this version is not supported. (installed version = hh...hh, installed SI version = ii...ii, version to be installed = jj...jj, SI version to be installed = kk...kk)	すでにインストールされているバージョン = hh...hh, SI バージョン ii...ii からセットアップしようとしている新バージョン = jj...jj, SI バージョン = kk...kk へのアップグレードはサポートされていません。	アップグレードをサポートしていない旧製品が存在します。この製品をセットアップする前に旧製品をアンインストールして、再度セットアップを実行してください。
7	94	E:An attempt to install the Cosminexus Component Container has failed.*	Component Container のインストールに失敗しました。	該当する製品をアンインストールしたあと、再度インストールしてください。

項目番号	γの内容	zz...zz の内容	意味	対処
8	95	E:An attempt to perform setup failed because the rr...rr service is running.※	rr...rr サービスが起動中のため、セットアップに失敗しました。	サービスを停止したあと、再度セットアップを実行してください。
9	97	E:An executing or locked file was detected. (file = dd...dd)	ビジー状態（実行中）またはロックされているファイル dd...dd を検出しました。	Component Container のプログラム、および関連する Web サーバなどを停止させて、再度セットアップを実行してください。
10	oo...o o	E:The command returned an error. (command = cc...cc)	コマンド cc...cc がエラーを返しました。	コマンドの名称とエラー情報から対処を実施し、再度セットアップを実行してください。
11	oo...o o	E:The command returned an error. (command = cc...cc, information 1 = ll...ll)	コマンド cc...cc がエラーを返しました。情報 1 =ll...ll	コマンドの名称とエラー情報から対処を実施し、再度セットアップを実行してください。
12	oo...o o	E:The command returned an error. (command = cc...cc, information 1 = ll...ll, information 2 = mm...mm)	コマンド cc...cc がエラーを返しました。情報 1 =ll...ll, 情報 2 =mm...mm	コマンドの名称とエラー情報から対処を実施し、再度セットアップを実行してください。
13	oo...o o	E:The command returned an error. (command = cc...cc, information 1 = ll...ll, information 2 = mm...mm, information 3 = nn...nn)	コマンド cc...cc がエラーを返しました。情報 1 =ll...ll, 情報 2 =mm...mm, 情報 3 =nn...nn	コマンドの名称とエラー情報から対処を実施し、再度セットアップを実行してください。

注※ Component Container の場合に出力されるインストールログです。リダイレクタ機能では出力されません。

付録 J Smart Composer 機能が実行する負荷分散機の CLI コマンドまたは API

Smart Composer 機能で提供するコマンドを実行すると、Management Server と接続して、負荷分散機の設定をします。負荷分散機の設定は、業務リクエストを Web サーバに振り分けるための設定で、仮想サーバや実サーバの作成、仮想サーバと実サーバの接続、および実サーバへのリクエスト振り分けの開始・停止などを設定します。ここでは、Smart Composer 機能が実行する負荷分散機の CLI コマンドと API について説明します。それぞれ使用する製品を次に示します。

- ・負荷分散機の API を使用する製品
BIG-IP v9※、BIG-IP v10.1※、BIG-IP v10.2※、BIG-IP v11、ACOS (AX2500)
- ・負荷分散機の CLI コマンドを使用する製品
ACOS (AX2000, BS320)

注※ 負荷分散機の CLI コマンドも使用できます。

付録 J.1 Smart Composer 機能が実行する負荷分散機の API

Smart Composer 機能が実行する負荷分散機の API については、マニュアル「アプリケーションサーバ機能解説 セキュリティ管理機能編」の「8.3 運用管理機能が実行する負荷分散機の API」を参照してください。

付録 J.2 Smart Composer 機能が実行する負荷分散機の CLI コマンド

Smart Composer 機能が実行する負荷分散機の CLI コマンドを製品ごとに説明します。負荷分散機の CLI コマンドの詳細は、使用する製品のドキュメントを参照してください。

(1) AX2000 または BS320 の場合

AX2000 または BS320 の場合、負荷分散機の接続方式として telnet プロトコルを使用した直接接続が使用できます。

負荷分散機として ACOS (AX2000 または BS320) を使用する場合に、Smart Composer 機能が実行する負荷分散機の CLI コマンドを次の表に示します。ACOS の CLI コマンドの詳細は、使用する製品のドキュメントを参照してください。

表 J-1 Smart Composer 機能が実行する負荷分散機の CLI コマンド (AX2000 または BS320 の場合)

Smart Composer 機能のコマンド	実行する負荷分散機の CLI コマンド
共通	enable config show version exit
cmx_build_system (Web システムを構築する場合)	cookie スイッチングの使用なし※1
	slb server <実サーバ名> <実サーバ IP アドレス> port <実ポート> tcp
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable
	slb virtual-server <仮想サーバ名> <仮想サーバ IP アドレス> port <仮想ポート> http service-group <仮想サーバ名>
	write memory
	cookie スイッチングの使用あり※1
	slb server <実サーバ名> <実サーバ IP アドレス> port <実ポート> tcp
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable
cmx_build_system (スケールアウトする場合)	slb template persist cookie <仮想サーバ名> name <cookie 名> expire 0
	slb virtual-server <仮想サーバ名> <仮想サーバ IP アドレス> port <仮想ポート> http service-group <仮想サーバ名> template persist cookie <仮想サーバ名>
	write memory
cmx_build_system (実ポートを変更する場合)	slb server <実サーバ名> <実サーバ IP アドレス> port <変更後の実ポート> tcp exit no port <実ポート> tcp
	slb service-group <仮想サーバ名> tcp

Smart Composer 機能のコマンド	実行する負荷分散機の CLI コマンド
	member <実サーバ名>:<変更後の実ポート> disable write memory
cmx_delete_system (サービスユニットを削除する場合) ^{※2}	slb server <実サーバ名> <実サーバ IP アドレス> no port <実ポート> tcp write memory
cmx_delete_system (ホスト単位モデルのスケールインを行う場合) ^{※3}	no slb server <実サーバ名> write memory
cmx_delete_system (Web システムを削除する場合)	no slb virtual-server <仮想サーバ名> no slb service-group <仮想サーバ名> no slb server <実サーバ名> write memory
	no slb virtual-server <仮想サーバ名> no slb service-group <仮想サーバ名> no slb server <実サーバ名> no slb template persist cookie <仮想サーバ名>
	write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> enable write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable write memory
	slb service-group <仮想サーバ名> tcp member <実サーバ名>:<実ポート> disable write memory

注※1 仮想サーバの初期化を実施しない場合 (-lb keepvs を指定した場合) も実行する CLI コマンドは同じです。

注※2 実サーバを共有するそのほかのサービスユニットがある場合は、サーバごとの削除はしません。

注※3 実サーバを共有するそのほかのサービスユニットがない場合は、サーバごと削除します。

参考

ACOS の場合、仮想サーバ名をサービスグループ名およびテンプレート名として使用します。

(2) BIG-IP v9, BIG-IP v10.1, または BIG-IP v10.2 の場合

BIG-IP v9, BIG-IP v10.1, または BIG-IP v10.2 の場合、負荷分散機の接続方式として ssh プロトコルを使用した直接接続が使用できます。なお、09-00 以降では、負荷分散機の接続方式として API を使用した直接接続が使用できます。

ssh プロトコルを使用した直接接続する場合は、Management Server を配置するホストに ssh のクライアント環境を設定する必要があります。ssh のクライアント環境の設定方法を OS ごとに次に示します。

- Windows の場合

Management Server を配置するホストに ssh クライアントプログラムが必要です。使用できる ssh クライアントプログラムは、PuTTY の CLI コマンドです。事前に PuTTY を入手し、インストールします。また、PuTTY のインストールフォルダ以下にある plink コマンドが格納されたフォルダを、システム環境変数 PATH に追加します。PuTTY は、該当ページ (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>) からダウンロードできます。

- UNIX の場合

ssh コマンドは、初めて使用する際に接続確認の問い合わせ要求を実行します。

root 権限のあるユーザで ssh コマンドを実行して負荷分散機と接続し、接続確認の問い合わせ要求 (Are you sure you want to continue connecting (yes/no)?) に対して「yes」を応答してください。「yes」を応答したあと、ssh コマンドは接続確認の問い合わせ要求を実行しなくなります。

■ 注意事項

AIX の場合、負荷分散機との接続形態に ssh の直接使用を選択すると、ssh コマンドが必要になります。ssh コマンドを使用するために必要なソフトウェアは事前に入手してください。

Smart Composer 機能が実行する BIG-IP の CLI コマンドを製品ごとに説明します。BIG-IP の CLI コマンドの詳細については、使用する製品のドキュメントを参照してください。

(a) BIG-IP v9 または BIG-IP v10.1 の場合

負荷分散機として BIG-IP v9 または BIG-IP v10.1 を使用する場合に、Smart Composer 機能が実行する負荷分散機の CLI コマンドを次の表に示します。

表 J-2 Smart Composer 機能が実行する負荷分散機の CLI コマンド (BIG-IP v9 または BIG-IP v10.1 の場合)

Smart Composer 機能のコマンド	実行する負荷分散機の CLI コマンド
共通	b save b version
cmx_build_system (Web システムを構築する場合)	cookie スイッチングの使用なし
	b pool <仮想サーバ名> {member <実サーバ IP アドレス>:<実ポート>} b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session disable b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート>} pool <仮想サーバ名> profile http enable
cookie スイッチングの使用あり	b pool <仮想サーバ名> {member <実サーバ IP アドレス>:<実ポート>} b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add

Smart Composer 機能のコマンド		実行する負荷分散機の CLI コマンド
		<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session disable</pre>
		<pre>b profile persist <仮想サーバ名> {mode cookie cookie mode insert cookie name <cookie 名>}</pre>
		<pre>b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http persist <仮想サーバ名> enable}</pre>
cmx_build_system (スケールアウトする場合)		<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add</pre>
cmx_build_system (実ポートを変更する場合)		<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session disable</pre>
cmx_build_system (負荷分散機の仮想サーバを初期化しないで Web システムを構築する場合)	cookie スイッチングの使用なし	<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add</pre>
		<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session disable</pre>
		<pre>b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http enable}</pre>
cmx_delete_system (サービスユニットを削除、およびホスト単位管理モデルをスケールインする場合)	cookie スイッチングの使用あり	<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add</pre>
		<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session disable</pre>
		<pre>b profile persist <仮想サーバ名> {mode cookie cookie mode insert cookie name <cookie 名>}</pre>
		<pre>b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http persist <仮想サーバ名> enable}</pre>
cmx_delete_system (Web システムを削除する場合)	cookie スイッチングの使用なし	<pre>b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> delete</pre>
		<pre>b node <実サーバ IP アドレス> delete</pre>
		<pre>b virtual <仮想サーバ名> delete</pre>
		<pre>b pool <仮想サーバ名> delete</pre>
		<pre>b node <実サーバ IP アドレス> delete</pre>
	cookie スイッチングの使用あり	<pre>b virtual <仮想サーバ名> delete</pre>
		<pre>b pool <仮想サーバ名> delete</pre>
		<pre>b node <実サーバ IP アドレス> delete</pre>

Smart Composer 機能のコマンド	実行する負荷分散機の CLI コマンド
	b profile persist <仮想サーバ名> delete
cmx_start_target	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session enable
cmx_stop_target	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session disable

参考

BIG-IP v9, BIG-IP v10.1 の場合、仮想サーバ名をプール名として使用します。

(b) BIG-IP v10.2 の場合

負荷分散機として BIG-IP v10.2 を使用する場合に、Smart Composer 機能が実行する負荷分散機の CLI コマンドを次の表に示します。

表 J-3 Smart Composer 機能が実行する負荷分散機の CLI コマンド (BIG-IP v10.2 の場合)

Smart Composer 機能のコマンド	実行する負荷分散機の CLI コマンド	
共通	b save b version	
cmx_build_system (Web システムを構築する場合)	cookie スイッチングの使用なし	b pool <仮想サーバ名> {member <実サーバ IP アドレス>:<実ポート>} b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user disabled b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http enable}
		b pool <仮想サーバ名> {member <実サーバ IP アドレス>:<実ポート>} b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user disabled b profile persist <仮想サーバ名> {mode cookie cookie mode insert cookie name <cookie 名>}
		b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http persist <仮想サーバ名> enable}
		b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user disabled

Smart Composer 機能のコマンド	実行する負荷分散機の CLI コマンド
cmx_build_system (実ポートを変更する場合)	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<変更後の実ポート> add
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<変更後の実ポート> session user disabled
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> delete
cmx_build_system (負荷分散機の仮想サーバを初期化しないで Web システムを構築する場合)	cookie スイッチングの使用なし
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user disabled
	b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http enable}
cmx_delete_system (サービスユニットを削除, およびホスト単位管理モデルをスケールインする場合)	cookie スイッチングの使用あり
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> add
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user disabled
	b profile persist <仮想サーバ名> {mode cookie cookie mode insert cookie name <cookie 名>}
cmx_delete_system (Web システムを削除する場合)	b virtual <仮想サーバ名> {destination <仮想サーバ IP アドレス>:<仮想ポート> pool <仮想サーバ名> profile http persist <仮想サーバ名> enable}
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> delete
cmx_delete_system (Web システムを削除する場合)	cookie スイッチングの使用なし
	b virtual <仮想サーバ名> delete
	b pool <仮想サーバ名> delete
	b node <実サーバ IP アドレス> delete
	cookie スイッチングの使用あり
	b virtual <仮想サーバ名> delete
cmx_start_target	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user enabled
	b pool <仮想サーバ名> member <実サーバ IP アドレス>:<実ポート> session user disabled

参考

BIG-IP v10.2 の場合, 仮想サーバ名をプール名として使用します。

付録 K XDM/RD E2 の設定

HiRDB Type4 JDBC Driver を使用して XDM/RD E2 に接続する場合の設定について説明します。なお、データベースとして XDM/RD E2 を使用する場合は、XDM/RD E2 をインストールして初期設定を済ませておいてください。詳細については、マニュアル「HiRDB XDM/RD E2 接続機能」を参照してください。なお、HiRDB Type4 JDBC Driver の設定については、「[4.1.6 データベース接続環境を設定する \(HiRDB の設定\)](#)」を参照してください。

ここでは、XDM/RD E2 で J2EE リソースを扱う場合に必要な次の設定について説明します。

- 環境変数の設定
- 環境変数のグループ登録
- タイムアウトの設定

付録 K.1 環境変数の設定

次の環境変数を設定する必要があります。詳細については、マニュアル「HiRDB XDM/RD E2 接続機能」、およびマニュアル「HiRDB UAP 開発ガイド」を参照してください。

PDHOST

データベースのホスト名または IP アドレスを指定します。

PDUSER

データベースのユーザ名およびデータベースのパスワードを指定します。

PDNAMEPORT

DB コネクションサーバのサーバスケジュール番号を指定します。

PDSRVTYPE

サーバの種別として、「VOS3」（固定値）を指定します。

PDCLTRDNODE

接続する XDM/RD E2 のデータベース識別子として、RD ノード名称（例：XRAA）を指定します。

PDXAMODE

トランザクションの移行機能は使用しないので、「0」（固定値）を指定します。

PDCWAITTIME

レスポンスタイムアウトとして、トランザクションタイムアウトの値よりも大きな値を指定します。

付録 K.2 環境変数のグループ登録

環境変数のグループは、次の方法で登録します。

- **Windows の場合**

HiRDB のクライアント環境変数登録ツールで「システムグループ」を選択して、環境変数のグループを登録します。この場合、DB Connector のプロパティ定義では、ここで登録したグループ名を指定してください。

- **UNIX の場合**

HiRDB のクライアント環境変数グループの設定ファイルに登録します。この場合、DB Connector のプロパティ定義では、ここで登録した環境変数グループの設定ファイルのパスを指定してください。

なお、環境変数のグループ登録の詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

付録 K.3 タイムアウトの設定

XDM/RD E2 のタイムアウトは、HiRDB のクライアント環境変数、または DB コネクションサーバのコントロール空間起動制御文／サーバ空間起動制御文に設定します。

XDM/RD E2 のタイムアウトを設定するチューニングパラメタを次の表に示します。

表 K-1 XDM/RD E2 のタイムアウトを設定するチューニングパラメタ

タイムアウトの種類	設定個所	設定方法 (パラメタ名)	設定内容
レスポンスタイムアウト	HiRDB クライアント環境変数	PDCWAITTIME	任意の値を指定します。※1
SQL 実行 CPU 時間タイムアウト	DB コネクションサーバのコントロール空間起動制御文／サーバ空間起動制御文	SQLCTIME	任意の値を指定します。※2
SQL 実行経過時間タイムアウト	DB コネクションサーバのコントロール空間起動制御文／サーバ空間起動制御文	SQLETIME	任意の値を指定します。※2
トランザクション経過時間タイムアウト	DB コネクションサーバのコントロール空間起動制御文／サーバ空間起動制御文	SVETIME	任意の値を指定します。※2
サーバ処理要求の間隔時間	DB コネクションサーバのコントロール空間起動制御文／サーバ空間起動制御文	SVINTERVAL	コネクションプーリング機能を使用する場合は、「0」を指定してください。 ※2

注※1

詳細については、マニュアル「HiRDB XDM/RD E2 接続機能」を参照してください。

注※2

詳細については、マニュアル「VOS3 Database Connection Server」を参照してください。

タイムアウトの設定については、マニュアル「アプリケーションサーバ システム設計ガイド」の「8.6.6 データベースでのタイムアウトを設定する」を参照してください。

付録 L アプリケーションサーバが提供するサンプルプログラム

アプリケーションサーバでは、次に示すサンプルプログラムを提供しています。

- Enterprise Bean (EJB2.0) のサンプルプログラム
- Enterprise Bean (アノテーション) のサンプルプログラム
- Enterprise Bean (Timer Service) のサンプルプログラム
- Java Beans リソースのサンプルプログラム
- ユーザログ機能のサンプルプログラム
- Web アプリケーションのサンプルプログラム
- TP1 インバウンド連携機能のサンプルプログラム
- JSF のサンプルプログラム
- Bean Validation および CDI 連携アプリケーションのサンプル

注意事項

提供しているサンプルプログラムは、わかりやすさを優先するため、紹介している多くのコードで、エラー処理やセキュリティ脆弱性問題を回避するための処理などを省略しています。そのため、サンプルプログラムを悪意のある第三者から攻撃された場合は、重大な問題が発生するおそれがあります。本番環境ではサンプルプログラムを使用しないでください。サンプルプログラムを本番環境で使用して、サンプルプログラムを原因とする問題が発生しても保証しません。

ここでは、各サンプルの概要、サンプルの実行手順、およびサンプルの実行例を説明します。なお、サンプルプログラムは、サーバ管理コマンドを使用した実行手順と実行例について説明しています。

注意事項

UNIX 環境で使用する場合は、本文を次のとおりに読み替えてください。

- Windows 用のディレクトリ (<Application Server のインストールディレクトリ>¥) を、UNIX 用のディレクトリ (/opt/Cosminexus/) に読み替えてください。
- Windows 環境で使用するバッチファイルは、UNIX 環境ではシェルスクリプトになります。拡張子を除いて読み替えてください。
(例) Windows 環境 : testClient.bat, UNIX 環境 : testClient

なお、アプリケーションサーバが提供するサンプルプログラムを実行する場合は、書き込み権限のある別のディレクトリへサンプルプログラムをコピーして使用することをお勧めします。

付録 L.1 サーバ管理コマンドによるサンプル実行環境の構築、運用手順

ここでは、サーバ管理コマンドを使用してサンプルプログラムを実行する場合の J2EE サーバの構築、運用手順を説明します。この手順は、付録 I.2 以降の各サンプルプログラムを実行するときに必要になります。

(1) J2EE サーバのセットアップ

コマンドプロンプトで `cjsetup` コマンドを入力し、J2EE サーバをセットアップします。セットアップの実行例を次に示します。実行例では、「MyServer」という名称の J2EE サーバをセットアップしています。

```
C:\> cd Program Files\Hitachi\Cosminexus\CC\server\bin
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>cjsetup MyServer
KDJE40001-I The setup for the J2EE server has finished successfully. Server name = MyServer
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>
```

(2) J2EE サーバのカスタマイズ

アプリケーションにアクセスするための設定をします。アプリケーションにアクセスするには、次のどちらかの方法を使用します。

- Web サーバと連携する。
- J2EE サーバの NIO HTTP サーバに直接アクセスする。

ここでは、設定や運用が容易な J2EE サーバの NIO HTTP サーバに直接アクセスする方法を説明します。

注意事項

NIO HTTP サーバでは、8008 番ポートをデフォルトで使用します。ポート番号を変更するには、ユーザ定義ファイル（`usrconf.properties`）に、`webserver.connector.nio_http.port` プロパティキーを定義します。

NIO HTTP サーバで使用するポート番号に 8081 番を定義する例を示します。

```
webserver.connector.nio_http.port=8081
```

(3) J2EE サーバの起動

`cjstartsv` コマンドでセットアップ済みの J2EE サーバを起動します。

```
C:\> cd Program Files\Hitachi\Cosminexus\CC\server\bin
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>cjstartsv MyServer
<起動ログが出力される>
:
KDJE30028-I The J2EE server has started. Server name = MyServer
```

(4) リソースアダプタの設定

次の手順でリソースアダプタの設定をします。

1. `cjimportres` コマンドでリソースアダプタをインポートします。
2. `cjdeployrar` コマンドを使用して、リソースアダプタをデプロイします。
3. `cjgetrarprop` コマンドで Connector 属性ファイルを取得し、ファイル編集後に、`cjsetrarprop` コマンドで編集内容を反映させます。
4. `cjtestres` コマンドを使用して、リソースアダプタのテストをします。
5. `cjstarrar` コマンドを使用して、リソースアダプタを開始します。

(5) アプリケーションのインポート、設定、および開始

サンプルで提供するバッチファイル（deployApp.bat）を使用して、アプリケーションを操作します。このバッチファイルでは、アプリケーションの J2EE サーバへのインポートから、設定、開始が実行されます。

このバッチファイルで実行される処理の内容を示します。

1. `cjimportapp` コマンドが実行され、アプリケーションが J2EE サーバにインポートされます。
2. `cjstartapp` コマンドが実行され、アプリケーションが開始されます。
3. `cjlistapp` コマンドが実行され、J2EE サーバ上のアプリケーションの状態が一覧表示されます。この一覧を確認することで、アプリケーションが稼働状態（running）になっていることを確認できます。

(6) アプリケーションの停止、削除

サンプルで提供するバッチファイル（unDeployApp.bat）を使用して、アプリケーションを操作します。このバッチファイルでは、アプリケーションの停止と J2EE サーバからの削除が実行されます。

このバッチファイルで実行される処理の内容を示します。

1. `cjstopapp` コマンドが実行され、アプリケーションが停止します。
2. `cjdeleteapp` コマンドが実行され、アプリケーションが J2EE サーバから削除されます。
3. `cjlistapp` コマンドが実行され、J2EE サーバ上のアプリケーションの状態が一覧表示されます。この一覧を確認することで、アプリケーションが削除されたことを確認できます。

(7) J2EE サーバの終了

`cjstopsv` コマンドで J2EE サーバを終了します。

- `cjstopsv` コマンドのコンソール

```
> cd Program Files\Hitachi\Cosminexus\CC\server\bin  
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>cjstopsv MyServer  
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>
```

- J2EE サーバのコンソール

```
C:\> cd Program Files\Hitachi\Cosminexus\CC\server\bin  
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>cjstartsv MyServer  
<起動ログが出力される>  
:  
KDJE30028-I The J2EE server has started. Server name = MyServer  
:  
<終了ログが出力される>  
C:\Program Files\Hitachi\Cosminexus\CC\server\bin>
```

付録 L.2 Enterprise Bean (EJB2.0) のサンプルプログラム

Enterprise Bean (EJB2.0) を使用したサンプルプログラムは、次に示すディレクトリに格納されています。

<Application Server のインストールディレクトリ>\CC\examples\ejb\ejb20

ここでは、サンプルプログラムの一覧、実行例、および各サンプルプログラムの概要について説明します。

(1) Enterprise Bean (EJB2.0) のサンプルプログラム一覧

サンプルプログラムの一覧およびサンプルプログラム実行時の注意事項を示します。サンプルプログラムの一覧は、Enterprise Bean の種類と動作概要をまとめた一覧と、利用する技術、およびリソースをまとめた一覧に分けて示します。

注意事項

Enterprise Bean のサンプルプログラムは、現在使用している製品のバージョン・リビジョンのものを使用してください。旧バージョン・リビジョンのサンプルプログラムの動作は保証されません。

(a) サンプルプログラム一覧 (Enterprise Bean の種類と動作内容)

表 L-1 サンプルプログラムの一覧 (Enterprise Bean の種類と動作内容)

サンプル名	Enterprise Bean 種別	パーシスタンス	トランザクション	動作概要
account	Entity Bean	BMP	—	預金や払い戻しに応じて残高を計算し、データベースを更新します。
adder	Stateful Session Bean	—	CMT	入力数値の合計を計算して、ブラウザ上に表示します。
bank	Stateful Session Bean	—	CMT	当座預金口座から普通預金口座に資金を移動し、データベースを更新します。

サンプル名	Enterprise Bean 種別	パーシスタンス	トランザクション	動作概要
cart	Stateful Session Bean	—	CMT	購入する本をショッピングカートに入れます。カートにない本をカートから出そうとした場合は、例外を発生させます。
checker	Stateful Session Bean	—	CMT	割引対象となる金額を超えた場合、割引率を基に割引後の金額を計算します。割引対象となる金額と割引率は、Enterprise Bean の環境エントリで指定します。
cmp20product	Entity Bean	CMP2.0	—	商品情報をデータベースに登録し、登録された商品情報を検索します。このサンプルは、CMP 2.0 形式の Enterprise Bean を利用します。
confirmier	Stateless Session Bean	—	CMT	商品の注文に対する確認メールを送信します。
converter	Stateless Session Bean	—	CMT	通貨の種類をドルから円、円からユーロに変換します。
enroller	Stateless Session Bean	—	CMT	学生と科目の追加、および受講登録をして、データベースを更新します。
	Entity Bean	BMP	—	
jsptag [※]	Stateless Session Bean	—	CMT	通貨の種類をドルから円、円からユーロに変換します。このサンプルは、JSP を利用してブラウザ上に表示します。
jsptobean	Entity Bean	BMP	—	預金や払い戻しに応じて残高を計算し、データベースを更新します。残高は、JSP を利用してブラウザ上に表示します。
localadder	Stateful Session Bean	—	CMT	入力数値の合計を計算して、ブラウザ上に表示します。このサンプルは、サーブレットからローカルインターフェースを利用して Enterprise Bean を呼び出します。
message	Stateless Session Bean	—	CMT	JMS のキューを利用して、メッセージを送受信します。
	Message -driven Bean	—	CMT	
order	Entity Bean	BMP	—	注文番号と対応する商品情報を検索して表示します。注文番号と商品情報はデータベースで管理します。
product	Entity Bean	CMP	—	商品情報をデータベースに登録し、登録された商品情報を検索します。
salesrep	Entity Bean	BMP	—	顧客の追加と、担当する販売員の変更を行い、データベースを更新します。
shipper	Stateful Session Bean	—	CMT	注文に対する商品在庫を計算し、データベースを更新します。在庫が一定の数量を下回った場合、引き当て処理を呼び出します。
	Entity Bean	BMP	—	

サンプル名	Enterprise Bean 種別	パーシスタンス	トランザクション	動作概要
storagebin	Entity Bean	BMP	—	データベースの部品情報テーブルと部品在庫情報テーブルから、対応する部品の情報と在庫を検索します。
teller	Stateful Session Bean	—	BMT	銀行用端末機から当座預金のお金引き出し、残高を計算します。
warehouse	Stateless Session Bean	—	CMT	注文を受けた商品の状態、および商品の在庫を更新します。

(凡例)

—：該当しないことを示します。

注※ jsptag サンプルをコンパイルおよびアーカイブした EAR ファイル (converter.ear) も提供しています。

(b) サンプルプログラムの一覧 (接続するリソース、クライアントの種類)

サンプルプログラムの一覧と、接続するリソース、クライアントの種類を次の表に示します。

表 L-2 サンプルプログラム一覧 (接続するリソース、クライアントの種類)

サンプル	接続するリソース※1			クライアントの種類※2	
	データベース	メール	JMS	EJB クライアントア プリケーション	Web ブラウザ
account	○	—	—	○	—
adder	—	—	—	—	○
bank	○	—	—	○	—
cart	—	—	—	○	—
checker	—	—	—	○	—
confirmier	—	○	—	○	—
converter	—	—	—	○	—
enroller	○	—	—	○	—
jsptag	—	—	—	—	○
jsptobean	○	—	—	—	○
order	○	—	—	○	—
product	○	—	—	○	—
salesrep	○	—	—	○	—
shipper	○	—	—	○	—
storagebin	○	—	—	○	—
teller	○	—	—	○	—

サンプル	接続するリソース※1			クライアントの種類※2	
	データベース	メール	JMS	EJB クライアントア プリケーション	Web ブラウザ
warehouse	○	—	—	○	—
message	—	—	○	○	—
cmp20product	○	—	—	○	—
localadder	—	—	—	—	○

(凡例)

○：該当することを示します。 —：該当しないことを示します。

注※1 接続するリソースごとに、必要な設定があります。接続するリソースに応じて、「付録 L.2(1)(c) サンプルプログラムで使用するデータベースの設定」、「付録 L.2(1)(d) サンプルプログラムで使用するメールの設定」または「付録 L.2(1)(e) サンプルプログラムで使用するJMSの設定」を参照してください。

注※2 クライアントの種類に応じて、実行方法などが異なります。詳細は、「付録 L.2(1)(f) サンプルプログラムのクライアントの種類」を参照してください。

(c) サンプルプログラムで使用するデータベースの設定

ここでは、サンプルプログラムで使用するデータベースの設定について説明します。

「表 L-2 サンプルプログラム一覧（接続するリソース、クライアントの種類）」の「データベース」列に「○」が付いているサンプルプログラムでは、リソースとしてデータベースを利用します。サンプルプログラムを実行する前に、データベースおよびデータベース接続関連の設定を適切にしてください。

なお、データベースとして利用できるのは、Oracle または HiRDB です。

必要な設定を次に示します。

- HiRDB を利用する場合、外部キーの指定（参照制約）を有効に設定してください。
設定方法については、マニュアル「HiRDB システム導入・設計ガイド」を参照してください。
- 利用するデータベース種別に従って、サンプルが提供する「createTable_Oracle.sql」（Oracle の場合）または「createTable_HiRDB.sql」（HiRDB の場合）を実行して、データベースのテーブルを初期化してください。なお、HiRDB を利用する場合、外部キーの指定（参照制約）が有効になっていないと、「createTable_HiRDB.sql」の実行に失敗します。
- サーバ管理コマンドで DB Connector の設定をします。
DB Connector は、「DBConnector_Oracle_CP.rar」または「DBConnector_HiRDB_Type4_CP.rar」を利用することをお勧めします。
「DBConnector_Oracle_XA.rar」または「DBConnector_HiRDB_Type4_XA.rar」を利用した場合は、ライトトランザクション機能が無効になるように設定してから J2EE サーバを起動して、DB Connector を設定します。ライトトランザクション機能は、usrconf.properties の ejbserver.distributedtx.XATransaction.enabled キーに true を指定すると無効になります。
- リソースのリンクを解決します。

リンクの解決には、サンプルで提供するアプリケーション属性ファイル「cosminexus.xml」を利用します。「cosminexus.xml」の<resource-ref>タグにリソースの表示名を指定して、サーバ管理コマンドを利用して設定してください。リソースのリンク解決をしてから、アプリケーションのデプロイ操作をしてください。

- サンプルプログラムのうち、「product」と「cmp20product」では、「cosminexus.xml」の<schema-name>タグに「user1」が設定されています。利用するデータベース環境に応じて、適切な値に変更してからアプリケーションのデプロイ操作を実行してください。
- サンプルプログラムのうち、「teller」の実行にHiRDBを使用する場合、TellerEJB.javaファイル中の日付指定関数を指定している個所を、HiRDB用に書き換えてコンパイルしてください。

(d) サンプルプログラムで使用するメールの設定

ここでは、サンプルプログラムで使用するメールの設定について説明します。

「表 L-2 サンプルプログラム一覧（接続するリソース、クライアントの種類）」の「メール」列に「○」が付いているサンプルプログラムでは、リソースとしてメールを利用します。サンプルプログラムを実行する前に、メールサーバへの接続関連の設定を適切にしてください。

必要な設定を次に示します。

- EJB クライアントアプリケーションのソースファイル（ConfirmClient.java）に、受信メールアドレスがコーディングされています。この個所を適切な受信メールアドレスに書き換えてからコンパイルしてください。
- サーバ管理コマンドでメールコンフィグレーションを設定する必要があります。サーバ管理コマンドの利用方法については、マニュアル「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「6.3 メールコンフィグレーションの設定」を参照してください。
- リソースのリンクを解決します。ただし、メールコンフィグレーションの表示名を「Cosminexus_SMTP_Server」と定義した場合は、リンクの解決は不要です。これ以外の表示名を定義した場合は、リンクを解決してください。

リンクの解決には、サンプルで提供するアプリケーション属性ファイル「cosminexus.xml」を利用します。「cosminexus.xml」の<resource-ref>タグにリソースの表示名を指定して、サーバ管理コマンドで設定してください。リソースのリンク解決をしてから、アプリケーションのデプロイ操作をしてください。

(e) サンプルプログラムで使用する JMS の設定

ここでは、サンプルプログラムで使用する JMS の設定について説明します。

「表 L-2 サンプルプログラム一覧（接続するリソース、クライアントの種類）」の「JMS」列に「○」が付いているサンプルプログラムでは、リソースとして JMS を利用します。サンプルプログラムを実行する前に、JMS への接続関連の設定を適切にしてください。

必要な設定を次に示します。

- JMS として TP1/Message Queue - Access を使用するため、あらかじめ TP1/Message Queue - Access をインストールしておきます。また、OpenTP1 および TP1/Message Queue のサーバ実行環境もあらかじめ準備しておきます。
- サーバ管理コマンドによって、TP1/Message Queue - Access の設定をします。
ライトトランザクション機能を無効にして J2EE サーバを起動してから、TP1/Message Queue - Access の設定をします。なお、ライトトランザクション機能は、usrconf.properties の ejbserver.distributedtx.XATransaction.enabled キーに true を指定すると無効になります。
また、TP1/Message Queue - Access のトランザクションサポートレベルを、「XATransaction」にしてください。
- リソースのリンクを解決します。ただし、TP1/Message Queue - Access のデフォルトの表示名を変更しないで、かつ、キュー定義ファイルに表示名が「QUEUE1」となるキューを定義している場合、リンクの解決は不要です。TP1/Message Queue - Access のデフォルトの表示名を変更した場合や、キュー定義ファイルに表示名が「QUEUE1」となるキューを定義していない場合は、リンクの解決が必要です。
リンクの解決には、サンプルで提供するアプリケーション属性ファイル「cosminexus.xml」を利用します。「cosminexus.xml」の<resource-ref>タグ、<resource-env-ref>タグ、および<message-ref>タグにリソースの表示名を指定して、サーバ管理コマンドで設定してください。リソースのリンク解決をしてから、アプリケーションのデプロイ操作をしてください。

(f) サンプルプログラムのクライアントの種類

サンプルプログラムのクライアントには、EJB クライアントアプリケーションと、Web ブラウザで表示するアプリケーションの 2 種類があります。

- 「表 L-2 サンプルプログラム一覧（接続するリソース、クライアントの種類）」の「EJB クライアントアプリケーション」列に「○」が付いているサンプルプログラムは、EJB クライアントアプリケーションです。サンプルとして提供されている「testClient.bat」を使用して、クライアントアプリケーションを開始してください。
- 「表 L-2 サンプルプログラム一覧（接続するリソース、クライアントの種類）」の「Web ブラウザ」列に「○」が付いているサンプルプログラムは、Web ブラウザから操作する Web アプリケーションです。クライアントから操作する場合は、Web ブラウザを起動して、次の URL を指定してください。

`http://<マシン名>:<ポート番号>/<ContextRoot>/<ターゲットファイル>`

例えば、ローカルホストで動作するサンプル「adder」の場合は、次のように指定します。

`http://localhost/adder/adder_war/adder.html`

(2) Enterprise Bean (EJB2.0) のサンプル実行手順

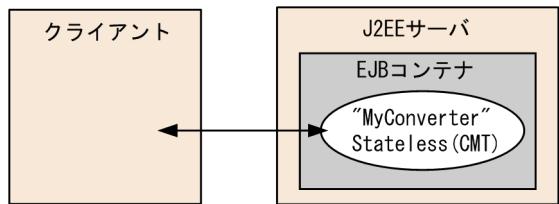
converter を例に、サンプルの実行例を示します。converter は、クライアントの Java アプリケーションから Enterprise Bean を呼び出すプログラムです。データベースなどのリソースは利用しません。リソースを利用する場合は、ここで示す手順に加え、リソースにアクセスできる環境を整える必要があります。

リソースの設定方法については、マニュアル「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。

(a) converter の概要

通貨の種類をドルから円、円からユーロに変換します。converter の構成を次に示します。

図 L-1 converter の構成



(b) コンパイル

サンプルで提供するバッチファイル (compileBean.bat および compileClient.bat) を使用して、Enterprise Bean とクライアントをコンパイルします。このバッチファイルで、アプリケーション (.ear ファイル) が作成されます。

```
C:>cd C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\converter
C:\Program Files\Hitachi\Cosminexus\CC\...\converter>compileBean
C:\Program Files\Hitachi\Cosminexus\CC\...\converter>
C:\Program Files\Hitachi\Cosminexus\CC\...\converter>compileClient
C:\Program Files\Hitachi\Cosminexus\CC\...\converter>
```

(c) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(d) J2EE サーバのカスタマイズ

「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(e) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(f) アプリケーションのデプロイ

サンプルで提供するバッチファイル (deployApp.bat) を使用して、アプリケーションをデプロイします。このバッチファイルでは、アプリケーションの J2EE サーバへのインポートから、アプリケーションの設定、開始、EJB のスタブおよびインターフェースファイルの取得までが実行されます。

このバッチファイルで実行される処理の内容を示します。

- アプリケーションのインポート、設定、開始、一覧表示

実行内容については、「付録 L.1(5) アプリケーションのインポート、設定、および開始」を参照してください。

- Enterprise Bean のスタブ・インターフェースファイルの取得
cjgetsubsjar コマンドで、Enterprise Bean のスタブとインターフェースを取得します。

(g) アプリケーションの実行

サンプルで提供するバッチファイル (testClient.bat) を使用して、アプリケーションを実行します。実行後、EJB クライアントアプリケーションの起動コンソール画面に次のように表示されれば成功です。

```
C:¥>cd C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ ejb¥ejb20¥converter  
C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ ejb¥ejb20¥converter>testClient  
KDJE40053-I The cjclstartap command will now start. (directory for the  
user definition file = C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥  
ejb¥ejb20¥converter, PID = 3684)  
12160.0  
0.77  
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)  
C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ ejb¥ejb20¥converter>
```

(h) アプリケーションのアンデプロイ

「付録 L.1(6) アプリケーションの停止、削除」を参照してください。

(i) J2EE サーバの終了

「付録 L.1(7) J2EE サーバの終了」を参照してください。

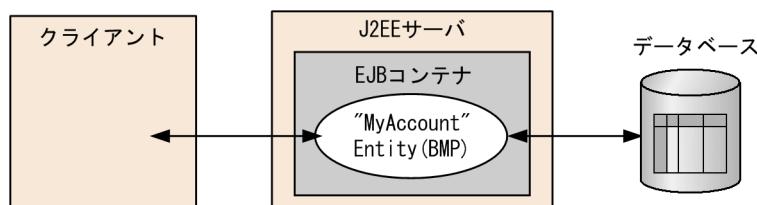
(3) Enterprise Bean (EJB2.0) のサンプルの概要

Enterprise Bean (2.0) のサンプルの概要、構成、実行例を示します。なお、必要な設定については、「付録 L.2(1)(b) サンプルプログラムの一覧 (接続するリソース、クライアントの種類)」を参照してください。

(a) account の概要

預金や払い戻しに応じて残高を計算し、銀行口座テーブルを更新します。account の構成を次に示します。

図 L-2 account の構成



account の実行例を示します。

```
C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ejb¥ejb20¥account>testClient  
KDJE40053-I The cjclstartap command will now start. (directory for the
```

```

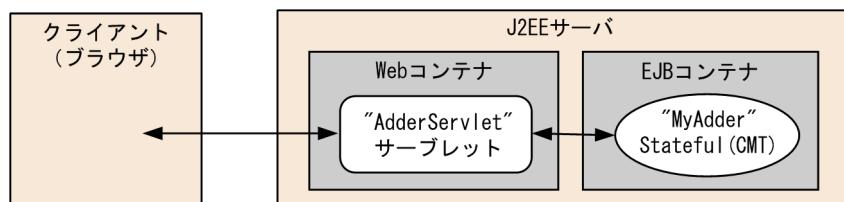
user definition file = C:\Program Files\Hitachi\Cosminexus\CC\examples\
ejb\ejb20\account, PID = 3684)
balance = 68.25
balance = 32.55
456: 44.77
730: 19.54
268: 100.07
456: 44.77
836: 32.55
KDJE40054-I The cjlstartap command was stopped. (PID = 3684, exit status = 0)
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\account>

```

(b) adder の概要

ブラウザから入力された数値の合計を計算します。adder の構成を次に示します。

図 L-3 adder の構成



実行時には、ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/adder.html
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは次の URL を指定します。

```
http://localhost:8008/adder/adder_war/adder.html
```

adder の実行例を示します。

図 L-4 adder の実行例

The running total is: 2

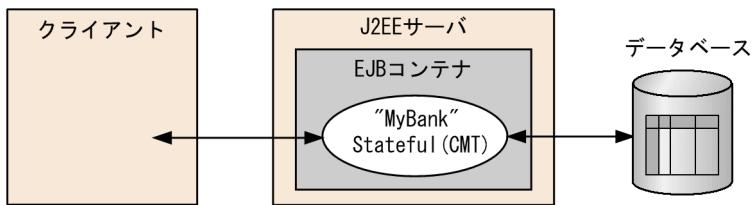
Please enter an integer:

adder の実行例画面。表示部には「The running total is: 2」があります。入力部には「Please enter an integer:」と入力欄があります。ボタン部には「クリア送信」があります。

(c) bank の概要

当座預金口座から普通預金口座に資金を移動します。資金はデータベースで管理し、資金移動時に、対応する当座預金口座テーブルと普通預金口座テーブルを更新します。bank の構成を次に示します。

図 L-5 bank の構成



bank の実行例を示します。

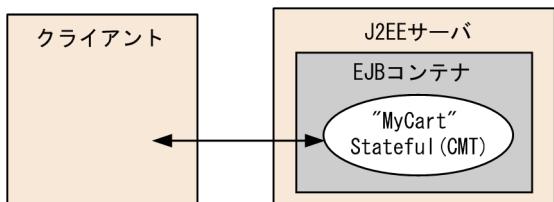
```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\bank>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\Hitachi\Cosminexus\CC\examples\
ejb\ejb20\bank, PID = 3684)
checking: 60.0
saving: 540.0
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\bank>
```

(d) cart の概要

本を購入するショッピングカートです。3冊の本をカートに入れ、カートにない本1冊をカートから出そうとして「BookException」を発生させます。cart の構成を次に示します。

図 L-6 cart の構成



cart の実行例を示します。

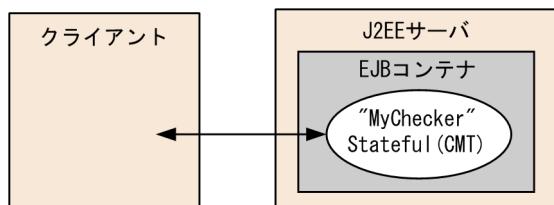
```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\cart>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\Hitachi\Cosminexus\CC\examples\
ejb\ejb20\cart, PID = 3684)
The Martian Chronicles
2001 A Space Odyssey
The Left Hand of Darkness
Caught a BookException: Alice in Wonderland not in cart.
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\cart>
```

(e) checker の概要

割引対象となる金額を超えた場合、割引率を基に割引後の金額を計算します。割引対象となる金額と割引率は、Enterprise Bean の環境エントリで指定します。checker の構成を次に示します。

図 L-7 checker の構成



checker の実行例を示します。

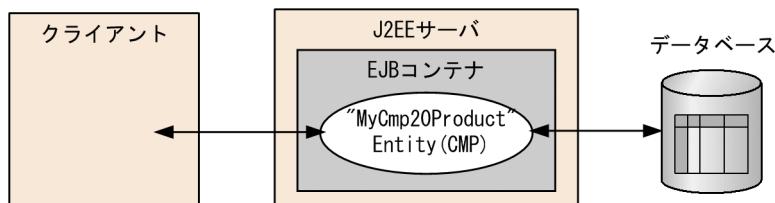
```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\checker>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\Hitachi\Cosminexus\CC\examples\
ejb\ejb20\checker, PID = 3684)
discount = 4200.0
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\checker>
```

(f) cmp20product の概要

商品情報をデータベースに登録し、登録された商品情報を検索します。このサンプルは、CMP 2.0 形式の Enterprise Bean を利用します。cmp20product の構成を次に示します。

図 L-8 cmp20product の構成



cmp20product の実行例を示します。

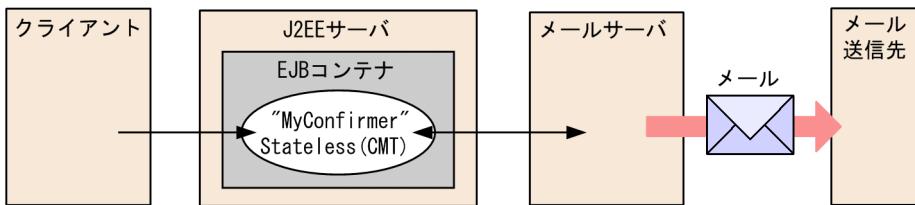
```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\cmp20product>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\cmp20product, PID = 3684)
Ceramic Dog: 10.0
Ceramic Dog: 14.0
Chrome Fish: 22.0
999: Ivory Cat 19.0
789: Ivory Cat 33.0
123: 14.0
456: 13.0
999: 19.0
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\cmp20product>
```

(g) confirmmer の概要

商品の注文に対する確認メールを送信します。confirmmer の構成を次に示します。

図 L-9 confirmmer の構成



confirmmer のクライアント（メール送信側）の実行例を示します。

```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\confirmmer>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\Hitachi\Cosminexus\CC\examples\
ejb\ejb20\confirmmer, PID = 3684)
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\confirmmer>
```

クライアント実行後、ConfirmmerClient.java で指定したアドレスにメールが送信されます。受信されるメールの例を示します。

件名:"Test Message from ConfirmmerEJB"
文面:
Thank you for your order.
We received your order on 2002/03/18 12:35.

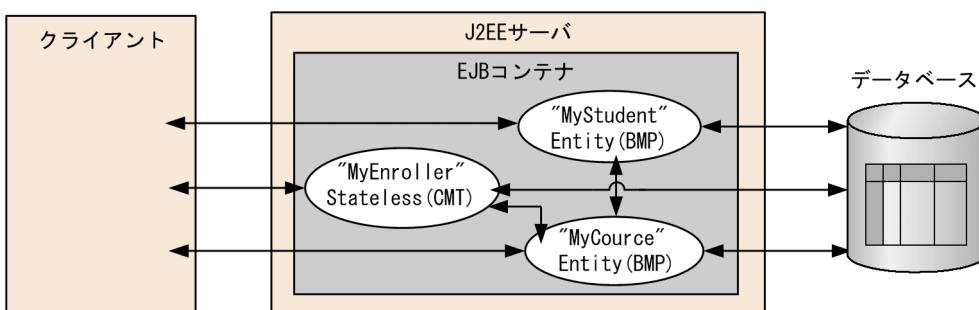
(h) converter の概要

converter については、「[付録 L.2\(2\) Enterprise Bean \(EJB2.0\) のサンプル実行手順](#)」を参照してください。

(i) enroler の概要

学生と科目的追加、および受講登録をして、対応する学生情報テーブル、科目テーブル、受講登録情報テーブルを更新します。enroler の構成を次に示します。

図 L-10 enroler の構成



enroller の実行例を示します。

```
C:\Program Files\Hitachi\Cosminexus\examples\ejb\ejb20\enroller>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\examples\
ejb\ejb20\enroller, PID = 3684)
Denise Smith:
220 Power J2EE Programming
333 XML Made Easy
777 An Introduction to Java Programming

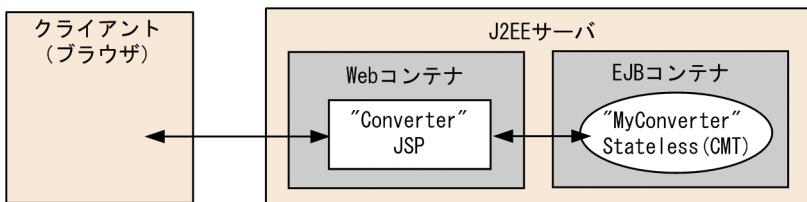
An Introduction to Java Programming:
823 Denise Smith
456 Joe Smith
388 Elizabeth Willis
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\examples\ejb\ejb20\enroller>
```

(j) jsptag の概要

通貨の種類をドルから円、円からユーロに変換します。変換した結果は、JSP を利用して、ブラウザに表示します。jsptag の構成を次に示します。

図 L-11 jsptag の構成



実行時には、ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/Converter.jsp
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは次の URL を指定します。

```
http://localhost:8008/jsptag/converter_war/Converter.jsp
```

jsptag の実行例を示します。

図 L-12 jsptag の実行例

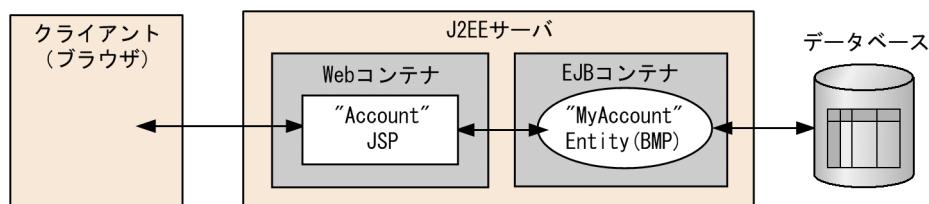
Converter JSP Example

```
dollarToYen: 12160.0
yenToEuro: 0.77
```

(k) jsptobean の概要

預金や払い戻しに応じて残高を計算し、データベースを更新します。残高は、JSP を利用してブラウザ上に表示します。jsptobean の構成を次に示します。

図 L-13 jsptobean の構成



実行時には、ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/Account.jsp
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは次の URL を指定します。

```
http://localhost:8008/jsptobean/account_war/Account.jsp
```

jsptobean の実行例を示します。

図 L-14 jsptobean の実行例

Account JSP Example

Account ID Balance
First Name Last Name

Action :

Create Find Debit Credit

Amount

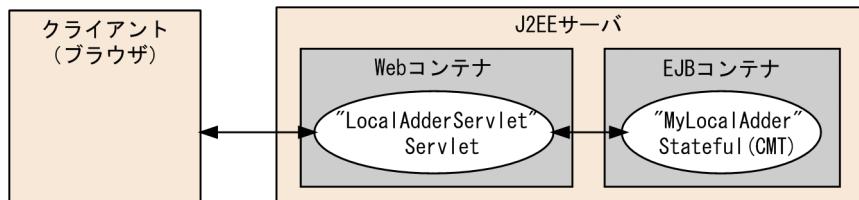
Status :

Created account '1'

(l) localadder の概要

入力数値の合計を計算して、ブラウザ上に表示します。このサンプルは、サーブレットからローカルインターフェースを利用して Enterprise Bean を呼び出します。localadder の構成を次に示します。

図 L-15 localadder の構成



実行時には、ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/adder.html
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは次の URL を指定します。

```
http://localhost:8008/localadder/localadder_war/adder.html
```

localadder の実行例を示します。

図 L-16 localadder の実行例

The screenshot shows a web page with the following content:

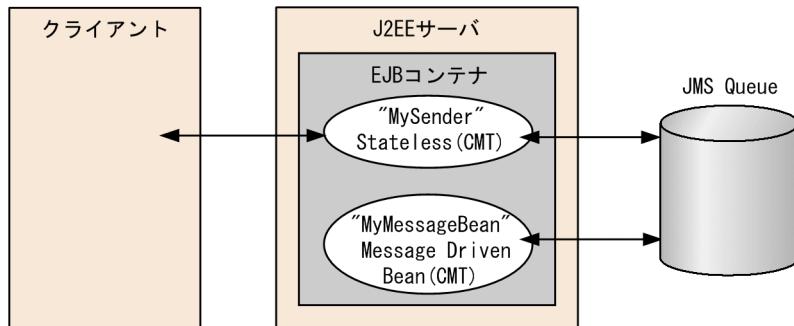
The running total is: 2

Please enter an integer:

(m) message の概要

Session Bean から JMS の Queue へ送信したメッセージを、Message-driven Bean で受信します。message の構成を次に示します。

図 L-17 message の構成



message のクライアントの実行例を示します。

```
C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ejb¥ejb20¥message>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:¥Program Files¥HITACHI¥Cosminexus¥CC¥examples¥
ejb¥ejb20¥message, PID = 3684)
sending message=MY MESSAGE
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ejb¥ejb20¥message>
```

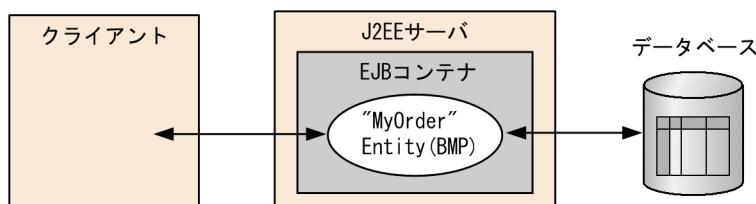
クライアント実行後、メッセージの送受信が正常に完了した場合は、J2EE サーバの起動コンソール上に次のように表示されます。

```
...
MessageBean: onMessage ()
MessageBean: received message=MY MESSAGE
...
```

(n) order の概要

注文番号と対応する商品情報を検索して表示します。注文番号と商品情報は、それぞれ注文情報テーブルと商品情報テーブルで管理します。order の構成を次に示します。

図 L-18 order の構成



order の実行例を示します。

```
C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ejb¥ejb20¥order>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥
ejb¥ejb20¥order, PID = 3684)
about to create
123 1 p23 12.0
123 2 p67 89.0
123 3 p11 41.0

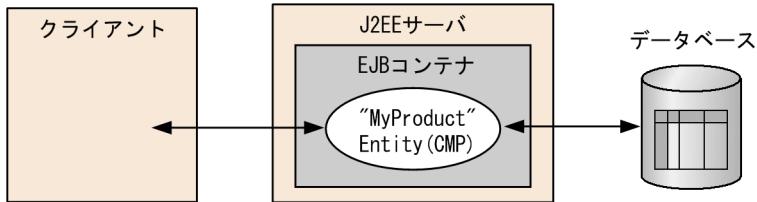
123
456
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:¥Program Files¥Hitachi¥Cosminexus¥CC¥examples¥ejb¥ejb20¥order>
```

(o) product の概要

商品名や価格などの商品情報をデータベースに登録し、登録された商品情報を検索します。商品情報は、商品情報テーブルで管理します。product の構成を次に示します。

図 L-19 product の構成



product の実行例を示します。

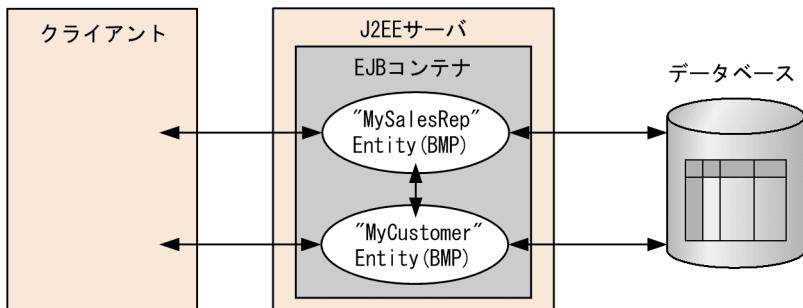
```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\product>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\product, PID = 3684)
Ceramic Dog: 10.0
Ceramic Dog: 14.0
Chrome Fish: 22.0
999: Ivory Cat 19.0
789: Ivory Cat 33.0
123: 14.0
456: 13.0
999: 19.0
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\product>
```

(p) salesrep の概要

顧客の追加と、担当する販売員の変更を行い、データベースを更新します。二つの Enterprise Bean を連携し、顧客テーブルと販売員テーブルへの検索と更新を行います。salesrep の構成を次に示します。

図 L-20 salesrep の構成



salesrep の実行例を示します。

```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\salesrep>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\salesrep, PID = 3684)
customerId = 221
customerId = 388
customerId = 456
customerId = 844
```

```

987: Mary Jackson
221: Alice Smith
388: Bill Williamson
456: Joe Smith
844: Buzz Murphy
KDJE40054-I The cjc1startap command was stopped. (PID = 3684, exit status = 0)

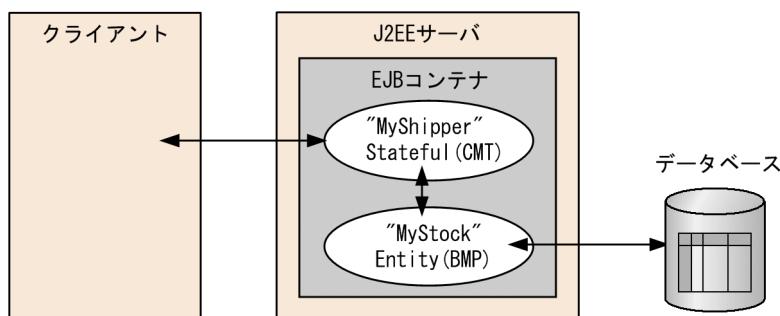
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\salesrep>

```

(q) shipper の概要

注文に対する商品在庫を計算し、データベースを更新します。在庫が一定の数量を下回った場合、引き当て処理を呼び出します。二つの Enterprise Bean を連携し、商品在庫情報テーブルへの検索と更新を行います。shipper の構成を次に示します。

図 L-21 shipper の構成



shipper の実行例を示します。

```

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\shipper>testClient
KDJE40053-I The cjc1startap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\shipper, PID = 3684)
KDJE40054-I The cjc1startap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\shipper>

```

shipper の実行後、J2EE サーバの起動コンソール上に次のように表示されます。

```

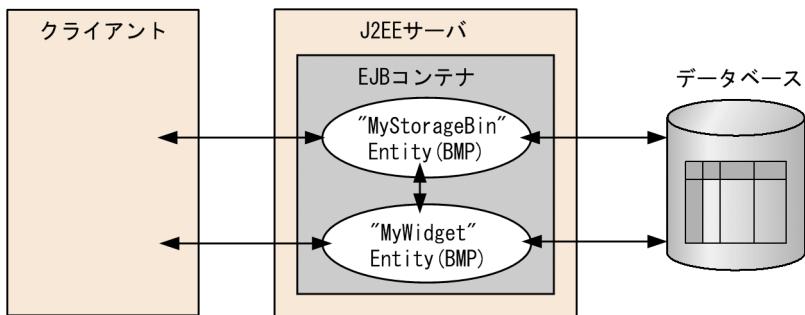
...
ShipperEJB : ejbCreate()
ShipperEJB : shipItem()
StockEJB : updateOnHand()
StockEJB : reorderNeeded()
...

```

(r) storagebin の概要

データベースから部品の情報と在庫を検索します。二つの Enterprise Bean を連携し、部品情報テーブルと部品在庫情報テーブルへの検索を行います。storagebin の構成を次に示します。

図 L-22 storagebin の構成



storagebin の実行例を示します。

```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\storagebin>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\storagebin, PID = 3684)
777 388 500 1.0 Duct Tape
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

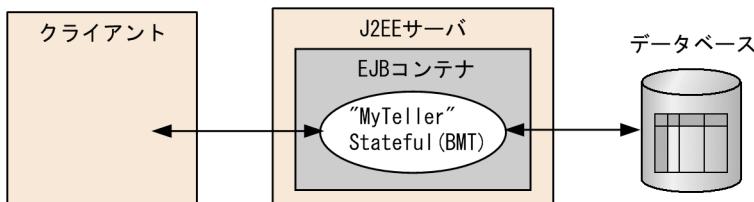
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\storagebin>
```

(s) teller の概要

銀行用端末機から当座預金を引き出し、残高を計算します。データベースの（銀行用端末機の）入金情報テーブルと取引履歴情報テーブルに対し、取引内容を追加します。

teller の構成を次に示します。

図 L-23 teller の構成



teller の実行例を示します。

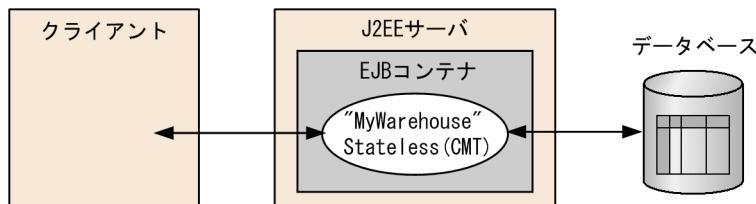
```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\teller>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\teller, PID = 3684)
checking = 500.0
checking = 440.0
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\teller>
```

(t) warehouse の概要

注文を受けた商品の状態、および商品の在庫を更新します。データベースの商品情報テーブルと在庫情報テーブルのそれぞれを更新します。warehouse の構成を次に示します。

図 L-24 warehouse の構成



warehouse の実行例を示します。

```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\warehouse>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\HITACHI\Cosminexus\CC\examples\
ejb\ejb20\warehouse, PID = 3684)
status = shipped
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)

C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\ejb20\warehouse>
```

付録 L.3 Enterprise Bean (アノテーション) のサンプルプログラム

アノテーションに対応した Enterprise Bean のサンプルプログラムは、次の 3 種類を提供します。

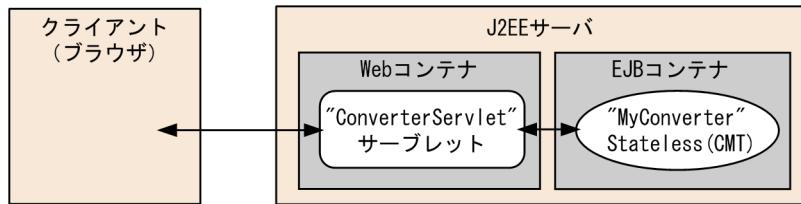
- アノテーション対応の Enterprise Bean (データベース接続なし)
格納場所は<Application Server のインストールディレクトリ
>\examples\ejb\annotation\stateless です。
- アノテーション対応の Enterprise Bean (データベース接続あり)
格納場所は<Application Server のインストールディレクトリ
>\examples\ejb\annotation\stateful です。
- Enterprise Bean (EJB2.0) から接続するアノテーション対応の Enterprise Bean
格納場所は<Application Server のインストールディレクトリ
>\examples\ejb\annotation\home です。

サンプルプログラムの概要と、実行手順を示します。

(1) アノテーション対応の Enterprise Bean (データベース接続なし) の概要

データベースに接続しない場合のアノテーション対応の Enterprise Bean サンプルプログラム「annotation_stateless」の構成を次に示します。

図 L-25 annotation_stateless の構成（アノテーション対応・データベース接続なし）



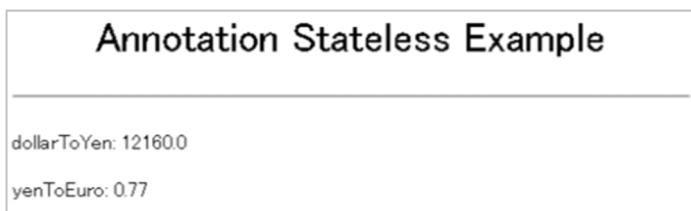
Web ブラウザ経由でサーブレットにアクセスし、実行結果をブラウザ上に表示します。このサンプルプログラムでは、サーブレットからビジネスインターフェースを使用した Enterprise Bean を呼び出します。

アノテーション対応の Enterprise Bean の参照を取得するため、サーブレット側で@EJB アノテーションを使用します。

```
@EJB(beanName="MyConverter")
Converter converter;
```

実行例を次に示します。

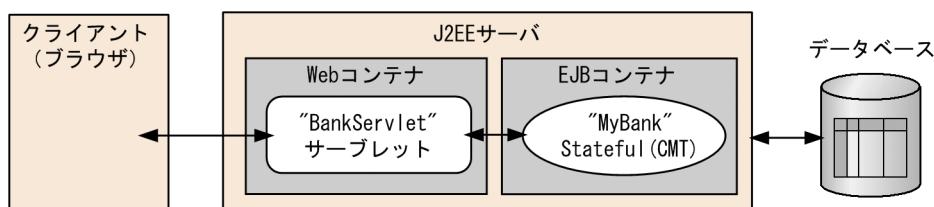
図 L-26 アノテーション対応 Enterprise Bean のサンプルプログラム（データベース接続なし）の実行例



(2) アノテーション対応の Enterprise Bean（データベース接続あり）の概要

データベースに接続する場合のアノテーション対応の Enterprise Bean サンプルプログラム「annotation_stateful」の構成を次に示します。

図 L-27 annotation_stateful の構成（アノテーション対応・データベース接続あり）



このサンプルプログラムでは、アノテーションを使用してデータベースに接続します。アノテーション対応の Enterprise Bean で、@Resource アノテーションを使用してリソースを取得します。

```
@Resource(mappedName="DB_Connector_for_Cosminexus_Driver")
private DataSource ds;
```

サーブレット側からビジネスインターフェースを使用した Stateful Session Bean を lookup するには、JNDI を使用します。

```
Context initial = new InitialContext();
duke = (Bank)initial.lookup("java:comp/env/ejb/Bank");
```

実行例を次に示します。

図 L-28 アノテーション対応 Enterprise Bean のサンプルプログラム（データベース接続あり）の実行例

Annotation Stateful Example

Invoke EJB with DBConnection

UserID:

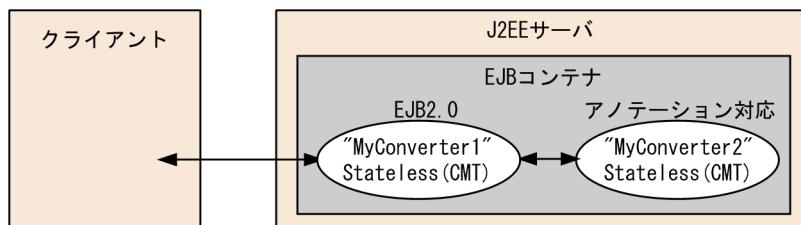
Transacted Amount:

ユーザ ID、金額を入力して処理を実行します。指定できるユーザ ID は 001 だけです。初期残高は 10,000 円です。

(3) Enterprise Bean (EJB2.0) から接続するアノテーション対応の Enterprise Bean の概要

アノテーション対応の Enterprise Bean にホームインターフェースの定義を追加することで、EJB2.0 からアノテーション対応の Enterprise Bean を呼び出せます。この処理を実行するサンプルプログラム「annotation_home」の構成を次に示します。

図 L-29 annotation_home の構成（アノテーション対応・EJB2.0 からの接続）



このサンプルプログラムでは、EJB2.0 の Enterprise Bean からローカルホームインターフェースが使用されているアノテーション対応の Enterprise Bean を呼び出します。EJB2.0 の Enterprise Bean のコードでは、ローカルホームインターフェースを lookup します。

```
...
Context ctx = new InitialContext();
Converter2LocalHome home = (Converter2LocalHome)ctx.lookup("java:comp/env/ejb/Converter2");
...
```

アノテーション対応の Enterprise Bean のローカルホームインターフェースは、Enterprise Bean に @LocalHome アノテーションを指定します。

```
@Stateless(name="MyConverter2")
@LocalHome(value=Converter2LocalHome.class)
public class Converter2EJB {
    ...
}
```

(4) Enterprise Bean (アノテーション) のサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) サンプルプログラムのコンパイル

サンプルで提供するバッチファイルを使用して、サンプルプログラムをコンパイルします。

アノテーション対応の Enterprise Bean (データベース接続あり／なし) のサンプルプログラムは、「compile.bat」を実行します。

EJB2.0 の Enterprise Bean からアノテーション対応の Enterprise Bean へ接続するサンプルは、「compileBean.bat」および「compileClient.bat」を実行します。

(b) SQL の実行 (アノテーション対応の Enterprise Bean (データベース接続ありの場合))

データベースに接続するサンプルプログラムの場合は、あらかじめ SQL を実行してください。SQL 文はサンプルで提供しています (createTable_Oracle.sql, createTable_HiRDB.sql)。

(c) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(d) J2EE サーバのカスタマイズ

「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(e) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(f) リソースアダプタの設定 (アノテーション対応の Enterprise Bean (データベース接続ありの場合))

データベースに接続するサンプルプログラムの場合は、リソースアダプタの設定が必要です。「[付録 L.1\(4\) リソースアダプタの設定](#)」を参照してください。

(g) アプリケーションのデプロイ

サンプルで提供するバッチファイル (deployApp.bat) を使用して、アプリケーションをデプロイします。このバッチファイルは、アプリケーションのインポート、開始、一覧表示、および Enterprise Bean のスタブ・インターフェースの取得をします。

このバッチファイルで実行される処理の内容を示します。

- アプリケーションのインポート、開始、一覧表示

実行内容については、「[付録 L.1\(5\) アプリケーションのインポート、設定、および開始](#)」を参照してください。

- Enterprise Bean のスタブ・インターフェースファイルの取得

cjgetsubsjar コマンドで、Enterprise Bean のスタブとインターフェースを取得します。

(h) アプリケーションの実行

アプリケーションごとに実行方法を示します。

- アノテーション対応の Enterprise Bean (データベース接続なし) の場合

ブラウザを起動し、URL に次の文字列を指定します。

http://<マシン名>:<ポート番号>/<ContextRoot>/<ターゲットファイル>

- アノテーション対応の Enterprise Bean (データベース接続あり) の場合

ブラウザを起動し、URL に次の文字列を指定します。

http://<マシン名>:<ポート番号>/<ContextRoot>/<ターゲットファイル>

- Enterprise Bean (EJB2.0) から接続するアノテーション対応の Enterprise Bean の場合

サンプルで提供するバッチファイル (testClient.bat) を使用して EJB クライアントアプリケーションを実行します。実行例を示します。

```
C:\Program Files\Hitachi\Cosminexus\CC\examples\ejb\annotation\home>testClient
KDJE40053-I The cjclstartap command will now start. (directory for the
user definition file = C:\Program Files\Hitachi\Cosminexus\CC\examples\
ejb\annotation\home, PID = 3684)
93.632
KDJE40054-I The cjclstartap command was stopped. (PID = 3684, exit status = 0)
```

(i) アプリケーションのアンデプロイ

「[付録 L.1\(6\) アプリケーションの停止、削除](#)」を参照してください。

(j) J2EE サーバの終了

「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。

付録 L.4 Enterprise Bean (Timer Service) のサンプルプログラム

Timer Service を実装するサンプルプログラムは、次のディレクトリに格納されます。

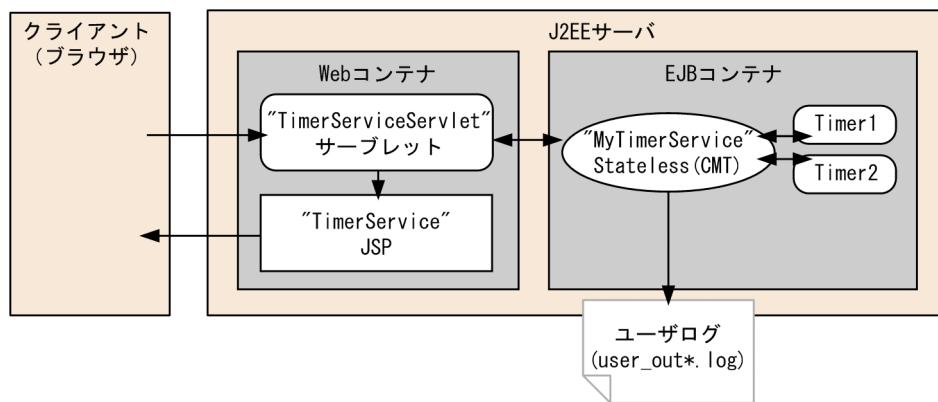
<Application Server のインストールディレクトリ>¥CC¥examples¥ejb¥timerservice

サンプルの概要と実行手順を示します。

(1) Enterprise Bean (Timer Service) のサンプルプログラムの概要

Timer Service のサンプルプログラム「TimerService」の構成を次に示します。

図 L-30 TimerService の構成



このサンプルプログラムでは、Web ブラウザ経由で EJB タイマの生成、一覧表示、キャンセルができます。EJB タイマがタイムアウトすると、ユーザログにメッセージを出力します。

実行例を次に示します。

図 L-31 TimerService のサンプルプログラムの実行例

Timer Service Example

Creation

Type	First Timeout	Interval	Information	Action
single-event	60 seconds later	---	MyTimer01	Create
interval	60 seconds later	300 seconds	MyTimer02	Create

List

Information	Remaining	Action
MyTimer01	21 seconds	<input type="button" value="Cancel"/>
MyTimer02	240 seconds	<input type="button" value="Cancel"/>
Default	160 seconds	<input type="button" value="Cancel"/>

サンプルでは、次の機能を提供します。

- EJB タイマの生成

アプリケーションの開始時に、EJB タイマが一つ生成されます。サーブレットに対して DD に<load-on-startup>タグを指定し、サーブレットの init メソッドで EJB タイマを生成する Stateless Session Bean のメソッドを呼び出します。

また、Web ページ EJB タイマ種別とタイムアウト時間と文字列情報を指定して [Create] ボタンをクリックすると、EJB タイマが生成されます。

- EJB タイマー一覧の表示

Web ページに EJB タイマの一覧が表示されます。[Refresh] ボタンをクリックすると、最新の EJB タイマー一覧が表示されます。

- EJB タイマのキャンセル

Web ページの [Cancel] ボタンをクリックすると、該当する EJB タイマがキャンセルされます。

- ユーザログへの出力

EJB タイマがタイムアウト日時になると、タイムアウトメソッドが呼び出されユーザログにメッセージが出力されます。出力例を下記に示します。

```
The timeout method was invoked. (Information = MyTimer02)
```

(2) Enterprise Bean (Timer Service) のサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) サンプルプログラムのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して、サンプルプログラムをコンパイルします。このバッチファイルで、EJB-JAR ファイル (timerservice.jar) と WAR ファイル (timerservice.war) と EAR ファイル (timerservice.ear) が作成されます。

(b) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(c) J2EE サーバのカスタマイズ

「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(d) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(e) アプリケーションのデプロイ

「[付録 L.1\(5\) アプリケーションのインポート、設定、および開始](#)」を参照してください。

(f) アプリケーションの実行

ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/TimerServiceServlet
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは、次の URL を指定します。

```
http://localhost:8008/timerservice/TimerServiceServlet
```

(g) アプリケーションのアンデプロイ

「[付録 L.1\(6\) アプリケーションの停止、削除](#)」を参照してください。

(h) J2EE サーバの終了

「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。

付録 L.5 JavaBeans リソースのサンプルプログラム

JavaBeans リソースを使用するサンプルプログラムは、次のディレクトリに格納されます。

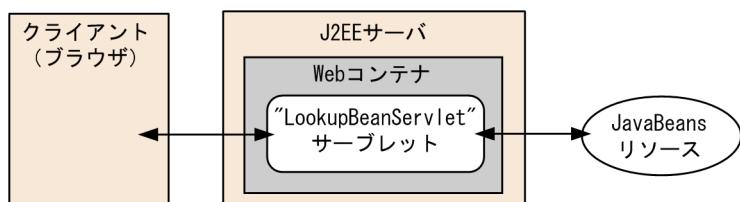
<Application Server のインストールディレクトリ>¥CC¥examples¥javabeans

サンプルの概要と実行手順を説明します。

(1) JavaBeans リソースのサンプルプログラムの概要

JavaBeans リソースのサンプルプログラムの構成を次に示します。

図 L-32 JavaBeans リソースのサンプルプログラムの構成



このサンプルプログラムは、JavaBeans リソース属性ファイルのプロパティに指定された各設定情報に基づき、ブラウザにその値を表示します。

実行例を次に示します。

図 L-33 JavaBeans リソースのサンプルプログラムの実行例

Property Value of JavaBeans Resource	
String:	ABC
int:	32768
char:	z
long:	2147483648
boolean:	true
float:	12.5
double:	1234.5
short:	32767
byte:	127

(2) JavaBeans リソースのサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) サンプルプログラムのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して、サンプルプログラムをコンパイルします。このバッチファイルで、JavaBeans リソース (MyJavaBean.jar) と EAR ファイル (LookupBean.ear) が作成されます。

(b) J2EE サーバのセットアップ

「付録 L.1(1) J2EE サーバのセットアップ」を参照してください。

(c) J2EE サーバのカスタマイズ

「付録 L.1(2) J2EE サーバのカスタマイズ」を参照してください。

(d) J2EE サーバの起動

「付録 L.1(3) J2EE サーバの起動」を参照してください。

(e) JavaBeans リソースのインポート、開始

サンプルで提供するバッチファイル (deployJb.bat) を使用して、JavaBeans リソースをデプロイします。このバッチファイルでは、JavaBeans リソースのインポート、開始が実行されます。

このバッチファイルで実行される処理の内容を示します。

- cjimportjb コマンドが実行され、JavaBeans リソースが J2EE サーバにインポートされます。使用される JavaBeans リソース属性ファイルは 「jb_resource.xml」 です。
- cjstartjb コマンドが実行され、JavaBeans リソースが開始されます。使用される JavaBeans リソース表示名は 「JavaBean_resource」 です。

- `cjlistjb` コマンドが実行され、J2EE サーバ上の JavaBeans リソースの状態が一覧表示されます。この一覧を確認することで、JavaBeans リソースが稼働状態 (running) になっていることを確認できます。

(f) アプリケーションのデプロイ

「[付録 L.1\(5\) アプリケーションのインポート、設定、および開始](#)」を参照してください。

(g) アプリケーションの実行

ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/servlet/LookupBeanServlet
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは、次の URL を指定します。

```
http://localhost:8008/web_jb/servlet/LookupBeanServlet
```

(h) アプリケーションのアンデプロイ

「[付録 L.1\(6\) アプリケーションの停止、削除](#)」を参照してください。

(i) JavaBeans リソースの停止

`cjstopjb` コマンドで JavaBeans リソースを停止します。

```
C:>cd Program Files\Hitachi\Cosminexus\CC\admin\bin  
C:\Program Files\Hitachi\Cosminexus\CC\admin\bin>cjstopjb MyServer -resname JavaBean_resource
```

(j) J2EE サーバの再起動

`cjstopsv` コマンドで J2EE サーバを終了し、`cjstartsv` コマンドで J2EE サーバを起動します。J2EE サーバの終了方法については、「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。J2EE サーバの起動方法については、「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(k) JavaBeans リソースの削除

`cjdeletejb` コマンドで JavaBeans リソースを削除します。

```
C:>cd Program Files\Hitachi\Cosminexus\CC\admin\bin  
C:\Program Files\Hitachi\Cosminexus\CC\admin\bin>cjdeletejb MyServer -resname JavaBean_resource
```

(l) J2EE サーバの終了

「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。

付録 L.6 ユーザログ機能のサンプルプログラム

ユーザログ機能の動作確認用のサンプルプログラムとして、userlog が提供されています。また、このサンプルプログラムを実行するためのソースファイルと定義ファイルのサンプルとして次の表に示すファイルが提供されています。

表 L-3 ユーザログ機能の動作確認用のサンプルプログラム

ファイル名	用途
UserLogServlet.java	ユーザログのサーブレットプログラムです。
UserLogHome.java	ホームインターフェースです。
UserLog.java	リモートインターフェースです。
AdderEJB.java	整数の合計を計算する Enterprise Bean の実装クラスです。
sample.properties	プロパティにセットするロガーのサンプルプロパティです。
MyFormatter.java	フォーマッタとして提供される独自クラスです。
MyFilter.java	フィルタとして提供される独自クラスです。

これらのファイルは、次の場所にあります。

<Application Server のインストールディレクトリ>¥CC¥examples¥userlog

ユーザログ機能のサンプルプログラムの概要と実行手順を説明します。

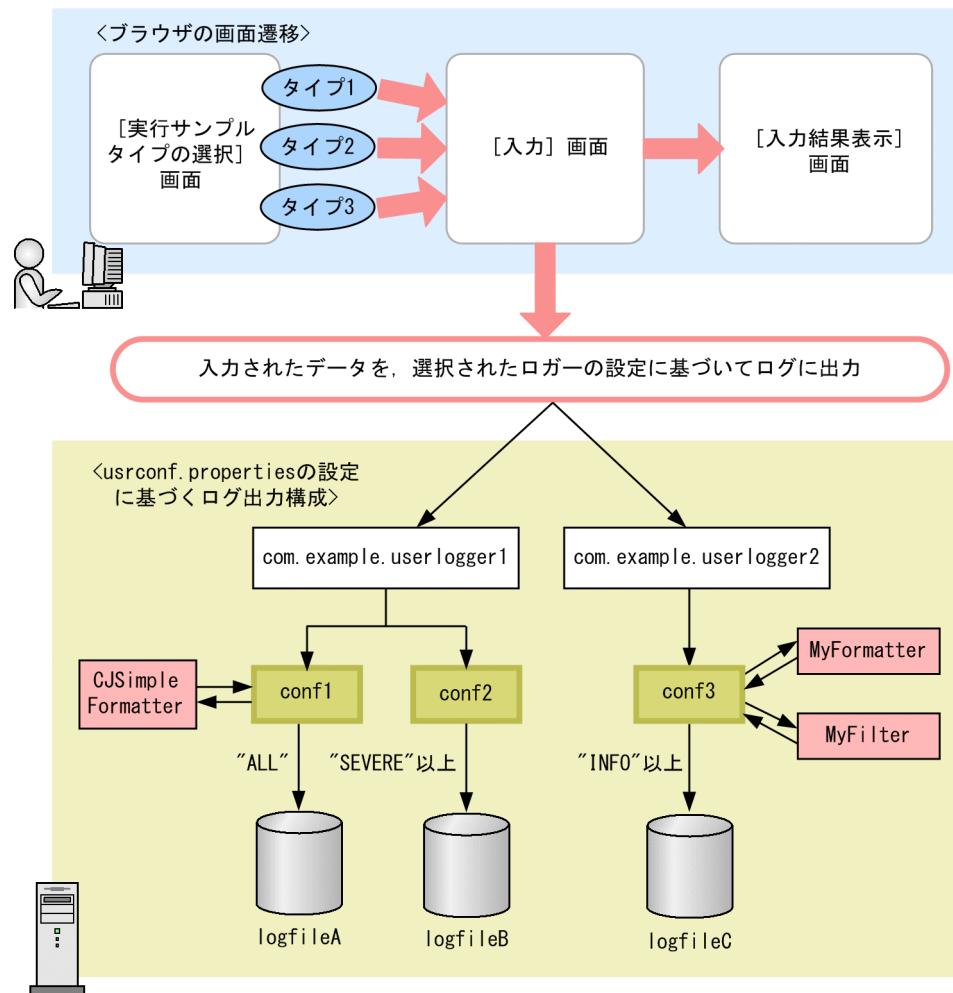
(1) ユーザログ機能のサンプルプログラムの概要

このサンプルプログラムでは、三つのタイプのログ出力を実行できます。

- タイプ 1 : Sample for outputting messages from servlet
入力されたメッセージがサーブレット側から出力されます。
- タイプ 2 : Sample for outputting the entered numbers and total value from AdderEJB
入力された整数の合計を計算する「AdderEJB (Enterprise Bean)」から、入力した値とその値を足した合計値がログファイルに出力されます。
- タイプ 3 : Sample for outputting exception logs from the servlet
Exception が発生し、その内容がサーブレットからログファイルに出力されます。

サンプルプログラムでは、次の図に示す流れで処理が行われます。

図 L-34 ユーザログのサンプルプログラムの構造 (サーブレット, Enterprise Bean)



(凡例)

□ : ロガー名
■ : フォーマッタ, フィルタ
■ : ハンドラ
→ : ユーザログのデータの流れ

[入力] 画面で選択するロガー名およびログレベルによって、出力先のログファイルが異なります。

- ロガー名「com.example.userlogger1」を選択した場合は、どのレベルを指定しても logfileA に出力されます。
- ロガー名「com.example.userlogger1」を選択して、ログレベルで「SEVERE」以上を指定した場合は、logfileA と logfileB に出力されます。
- ロガー名「com.example.userlogger2」を選択し、ログレベルで「INFO」以上を指定した場合、かつ「MyFilter」によってフィルタリングされなかった場合に、logfileC に出力されます。出力フォーマッタは「MyFormatter (独自クラス)」が使用されます。

MyFormatter の仕様

次の形式で出力します。

message : <入力されたメッセージ>

LEVEL : <指定されたレベル>

MyFilter の仕様

メッセージの頭に「debug:」が付いているものは、ログに出力しません。

(2) ユーザログ機能のサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(b) サンプルファイルのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して、ユーザログサンプルをコンパイルします。

コンパイルすると「myloglib.jar」というファイルが作成されます。これを任意のディレクトリに格納し、JAR ファイルにパスを通します。パスを通す方法については、マニュアル「[アプリケーションサーバ 機能解説 拡張編](#)」の「8.7.2 コンテナ拡張ライブラリを利用する方法」を参照してください。

(c) プロパティ情報の設定

セットアップしたサーバにプロパティ情報を設定します。セットアップしたサーバの usrconf.properties の内容の一部を、サンプルで提供する設定ファイル (sample.properties) に書き換えます。Management Server を利用してシステムを構築している場合は、マニュアル「[アプリケーションサーバ 機能解説 拡張編](#)」の「8.8 J2EE アプリケーションのユーザログ出力の設定」を参照して設定してください。

セットアップしたサーバを「MyServer」としたときの usrconf.properties の格納場所と、usrconf.properties ファイル内の書き換え位置を次に示します。

- ファイルの場所

<Application Server のインストールディレクトリ
>¥CC¥server¥usrconf¥ejb¥MyServer¥usrconf.properties

- ファイルの書き換え位置

次に示す箇所を、sample.property の内容に書き換えます。

```
:  
# user-log handler settings  
#ejbserver.application.userlog.CJLogHandler.<Specify handler-name>.path=user_log  
#ejbserver.application.userlog.CJLogHandler.<Specify handler-name>.count=2  
:  
# user-log logger settings  
#ejbserver.application.userlog.Logger.<Specify logger-name>.handlers=<Specify handler-class>  
#ejbserver.application.userlog.Logger.<Specify logger-name>.useParentHandlers=true  
:
```

サンプルで提供する sample.properties の内容を示します。

```

# user-log handler settings
#conf1 settings
ejbserver.application.userlog.CJLogHandler.conf1.path=userlog/logFileA
ejbserver.application.userlog.CJLogHandler.conf1.level=ALL
ejbserver.application.userlog.CJLogHandler.conf1.appname=userlog_app
ejbserver.application.userlog.CJLogHandler.conf1.formatter=com.hitachi.software.ejb.application.userlog.CJSimpleFormatter

#conf2 settings
ejbserver.application.userlog.CJLogHandler.conf2.path=userlog/logFileB
ejbserver.application.userlog.CJLogHandler.conf2.level=SEVERE
ejbserver.application.userlog.CJLogHandler.conf2.appname=userlog_app

#conf3 settings
ejbserver.application.userlog.CJLogHandler.conf3.path=userlog/logFileC
ejbserver.application.userlog.CJLogHandler.conf3.level=INFO
ejbserver.application.userlog.CJLogHandler.conf3.appname=userlog_app

# user-log logger settings
#logger1 settings
ejbserver.application.userlog.Logger.com.example.userlogger1.handlers=com.hitachi.software.ejb.application.userlog.CJMessageFileHandler;conf1,com.hitachi.software.ejb.application.userlog.CJMessageFileHandler;conf2
ejbserver.application.userlog.Logger.com.example.userlogger1.level=ALL

#logger2 settings
ejbserver.application.userlog.Logger.com.example.userlogger2.handlers=com.hitachi.software.ejb.application.userlog.CJMessageFileHandler;conf3
ejbserver.application.userlog.Logger.com.example.userlogger2.level=INFO

ejbserver.application.userlog.loggers=com.example.userlogger1, com.example.userlogger2

```

NIO HTTP サーバの設定については「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(d) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(e) アプリケーションのデプロイ

「[付録 L.1\(5\) アプリケーションのインポート, 設定, および開始](#)」を参照してください。

(f) アプリケーションの実行

ブラウザを起動し, URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/<ContextRoot>/userlog.html
```

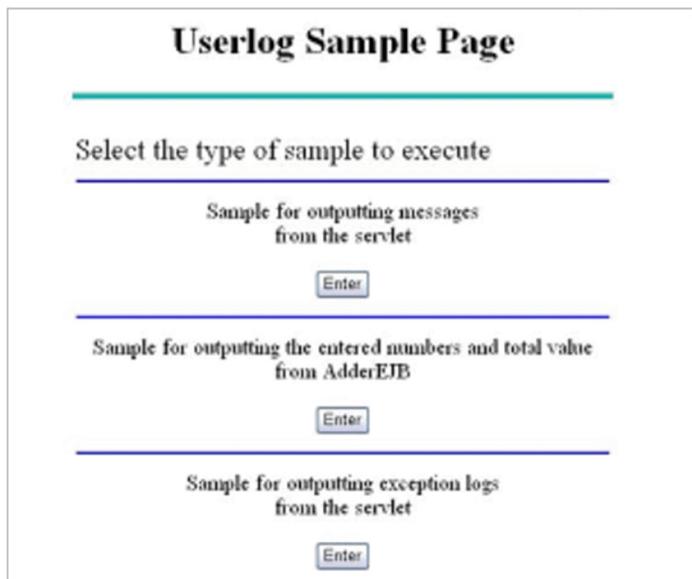
J2EE サーバの NIO HTTP サーバに直接アクセスする場合, デフォルトでは次の URL を指定します。

```
http://localhost:8008/userlog/userlog_war/userlog.html
```

(g) userlog の実行例

userlog の実行例を示します。アプリケーションを実行して、URL を指定すると、次に示す [実行サンプルタイプの選択] 画面が表示されます。

図 L-35 [実行サンプルタイプの選択] 画面 (ユーザログのサンプルプログラム)



実行するサンプルのタイプを選択し、各データ入力フォームへ移ります。この画面以降の操作とログ出力例については、選択したタイプごとに説明します。

出力結果のログファイルは、セットアップしたサーバ名を「MyServer」とした場合、usrconf.properties に設定したファイル名で、次に示すディレクトリに出力されます。

<Application Serverのインストールディレクトリ>¥CC¥server¥public¥MyServer¥logs¥user¥user.log

- タイプ1 : outputting message の実行例

[実行サンプルタイプの選択] 画面で「Sample for outputting message from the servlet」タイプを選んだ場合の画面遷移とログファイルの出力例を示します。

タイプ選択後、表示された [入力] 画面で次の値を入力します。

- ロガー名 : com.example.userlogger1
- メッセージ : AAA
- メッセージレベル : SEVERE (致命的)

次に、タイプ1を選択した場合の [入力] 画面を示します。

図 L-36 [入力] 画面 (タイプ1: ユーザログのサンプルプログラム)

Message Input Page

Select a logger_name to output

Enter a message to output to the log

Select a level for the message to be output

入力後、[send] ボタンをクリックすると、[入力結果表示] 画面が表示されます。

図 L-37 [入力結果表示] 画面 (タイプ1: ユーザログのサンプルプログラム)

Input Data

```
logger : com.example.userlogger1
handler : 1
filter : null
formatter : com.hitachi.software.ejb.application.user.log.CJSimpleFormatter

handler : 2
filter : null
formatter : null

logmsg : AAA
level : SEVERE
```

この例では、logfileA と logfileB に出力されます。

logfileA の出力結果を示します。

```
0310 2005/01/21 20:37:53.296      userlog_app      00922804 01000BCF Msg_100
2005/01/21 20:37:53|UserLogServlet|doPost|致命的|AAA
```

logfileB の出力結果を示します。

```
0888 2005/01/21 19:41:03.609      userlog_app      00922804 009B6220 Msg_100
AAA
```

- ・ タイプ2: outputting the entered numbers and total value の実行例

[実行サンプルタイプの選択] 画面で「Sample for outputting the entered numbers and total value from AdderEJB」タイプを選んだ場合の画面遷移とログファイルの出力例を示します。

タイプ選択後、表示された [入力] 画面で次の値を入力します。

- ・ ロガー名: com.example.userlogger1
- ・ 入力数: 10

次に、タイプ2を選択した場合の [入力] 画面を示します。

図 L-38 [入力] 画面 (タイプ2: ユーザログのサンプルプログラム)

Adder

Select a logger_name to output

com.example.userlogger1

Enter an integer :

10

send

入力後、[send] ボタンをクリックすると、[入力結果表示] 画面が表示されます。

図 L-39 [入力結果表示] 画面 (タイプ2: ユーザログのサンプルプログラム)

Input Data

logger : com.example.userlogger1

handler : 1
filter : null
formatter : com.hitachi.software.ejb.application.userlog.CJSimpleFormatter

handler : 2
filter : null
formatter : null

int : 10

total : 10

return

Top

この例では、logfileA だけに出力されます。

logfileA の出力結果を示します。

```
0324 2005/01/21 20:38:59.500      userlog_app      00922804 015C998A Adder_200
2005/01/21 20:38:59|AdderEJB|add|情報|inputNumber = 10
```

```
0327 2005/01/21 20:38:59.500 userlog_app 00922804 015C998A Adder_201
2005/01/21 20:38:59|AdderEJB|getTotal|情報|total = 10
```

- ・ タイプ3: outputting exception logs の実行例

[実行サンプルタイプの選択] 画面で「Sample for outputting exception logs from the servlet」タイプを選んだ場合の画面遷移とログファイルの出力例を示します。

タイプ選択後、表示された [入力] 画面で次の値を入力します。

- ・ ロガー名: com.example.userlogger2

次に、タイプ3を選択した場合の [入力] 画面を示します。

図 L-40 [入力] 画面 (タイプ3: ユーザログのサンプルプログラム)

Exception

Select a logger_name to output

com.example.userlogger2

Cause an exception

send

入力後、[send] ボタンをクリックすると、[入力結果表示] 画面が表示されます。

図 L-41 [入力結果表示] 画面 (タイプ3: ユーザログのサンプルプログラム)

Input Data

```
logger : com.example.userlogger2
handler : 1
filter : MyFilter
formatter : MyFormatter
```

return

Top

この例では、logfileCだけに出力されます。

logfileCの出力結果を示します。

```
1077 2005/01/21 19:55:19.328 userlog_app 00922804 006458A6 Exp_300
message:Exception LEVEL:SEVERE
```

(h) アプリケーションのアンデプロイ

「付録 L.1(6) アプリケーションの停止, 削除」を参照してください。

(i) J2EE サーバの終了

「付録 L.1(7) J2EE サーバの終了」を参照してください。

付録 L.7 Web アプリケーションのサンプルプログラム

Web アプリケーションのサンプルプログラムは、次のディレクトリに格納されます。

<Application Server のインストールディレクトリ>¥CC¥examples¥webapps

サンプルプログラムの概要と実行手順を示します。

(1) Web アプリケーションのサンプルプログラムの概要

Web アプリケーションのサンプルとして、Servlet 2.5 仕様に準じたサーブレットおよび JSP のサンプルプログラムを利用できます。ここでは、前提事項、サンプルプログラムの種類、およびサンプルプログラムの構成について説明します。

(a) 前提事項

Web アプリケーションのサンプルプログラムの前提事項を次に示します。

- Web アプリケーションのバージョンは、Servlet 2.5 であること。
- J2EE サーバモードで実行すること。

また、Web アプリケーションのサンプルプログラムでは、JSTL 1.2 を使用します。JSTL 1.2 を使用するためには、usrconf.cfg にクラスパスを設定する必要があります。

(b) サンプルプログラムの種類

提供するサンプルプログラムの種類を次に示します。

サーブレット

- 「Hello World」の表示
- HTTP リクエスト情報の表示
- HTTP ヘッダの表示
- HTTP リクエストのパラメタ表示
- Cookie 操作
- HTTP セッションの操作

JSP

- Expression Language^{※1}
- シンプルタグハンドラ, JSP フラグメント
- タグファイル^{※1}
- XML シンタックスの JSP^{※1※2}
- その他 JSP2.0 仕様の機能
- JSP1.2 仕様の機能

注※1

JSTL を使用するサンプルプログラムです。

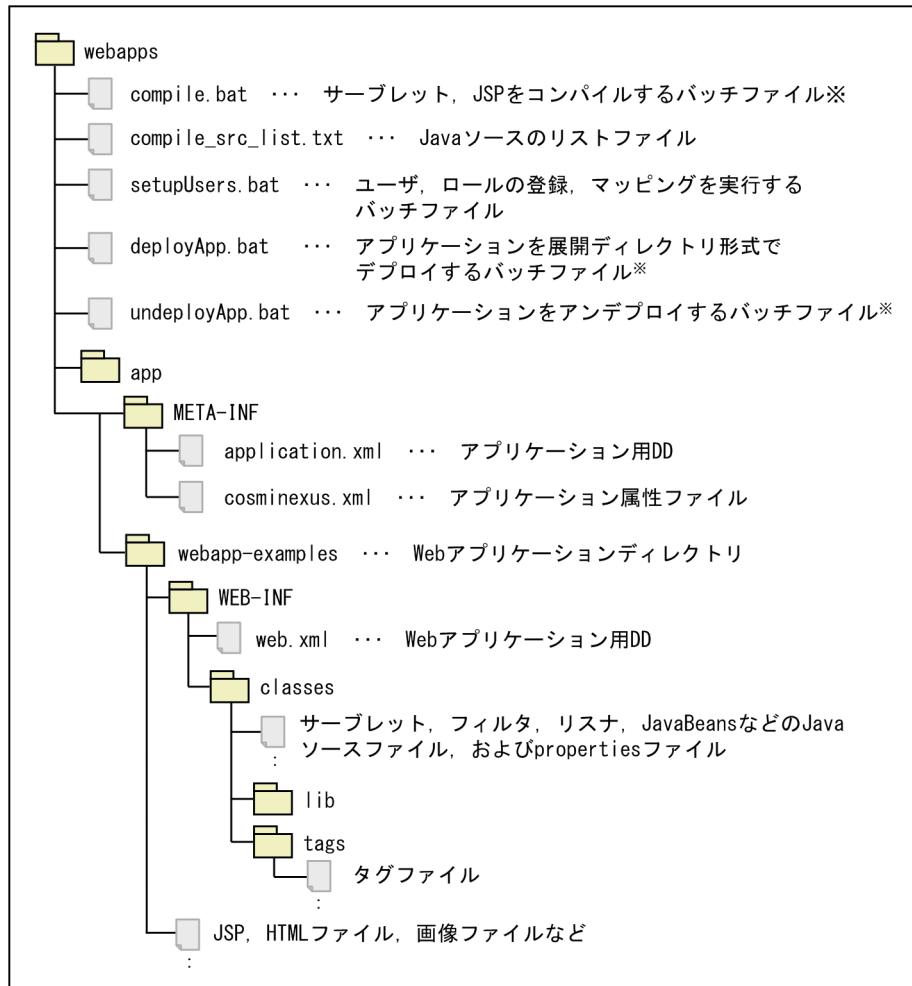
注※2

このサンプルでは, 実行結果として SVG 形式のデータを返します。SVG 形式のデータをブラウザで表示するには, SVG に対応したブラウザが必要です。

(c) サンプルプログラムの構成

Web アプリケーションのサンプルプログラムは, 展開ディレクトリ形式のアプリケーションとしてそのまま使用できます。<Application Server のインストールディレクトリ>¥CC¥examples¥webapps 以下の構成を次に示します。

図 L-42 webapps ディレクトリ以下の構成



注※ UNIX版ではシェルスクリプトです。拡張子はありません。

以降に説明する Web アプリケーションのサンプルプログラムの実行手順では, <Application Server のインストールディレクトリ>¥CC¥examples の下にある webapp ディレクトリを, <webapps ディレクトリ>と表記します。

(2) Web アプリケーションのサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。この手順では, 次の機能を使用します。

- 展開ディレクトリ形式のデプロイ
- JSP の事前コンパイル

(a) サンプルプログラムのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して, Web アプリケーションのサーブレット, フィルタ, リスナ, JavaBeans の Java ソースファイル, および JSP ファイルをコンパイルします。

なお, サーブレットやフィルタなど Java ソースファイルは javac コマンドが使用され, JSP ファイルのコンパイルには, cjjspc コマンドが使用されます。

(b) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(c) J2EE サーバのカスタマイズ

「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(d) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(e) ユーザ、ロールの登録

サンプルで提供するバッチファイル (setupUsers.bat) を使用して、ユーザおよびロールを登録します。このバッチファイルでは、J2EE サーバ名「MyServer」の J2EE サーバに対し、ユーザの登録とロールの登録の処理が実行されます。

このバッチファイルで実行される処理の内容を示します。

1. cjaddsec コマンドが実行され、次のユーザが登録されます。

ユーザ 1：ユーザ名「user1」、パスワード「passwd1」

ユーザ 2：ユーザ名「user2」、パスワード「passwd2」

2. cjaddsec コマンドが実行され、ロール「role1」とロール「role2」が登録されます。

3. cjmapsec コマンドが実行され、次のマッピングが登録されます。

マッピング 1：ロール「role1」とユーザ「user1」

マッピング 2：ロール「role2」とユーザ「user2」

ここで登録したユーザ名とパスワードは、JSP のサンプルページの「Security」で使用します。

(f) アプリケーションのデプロイ

「[付録 L.1\(5\) アプリケーションのインポート、設定、および開始](#)」を参照してください。

(g) アプリケーションの実行

ブラウザを起動し、URL に次の文字列を指定して Web アプリケーションのトップページにアクセスします。

http://<マシン名>/examples

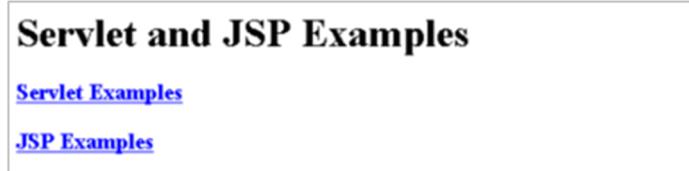
ポート番号をカスタマイズしている場合は、<マシン名>のあとに、「:<ポート番号>」を指定してください。

J2EE サーバを起動したホストからアクセスする場合は、次のように指定します。

http://localhost/examples

アクセスに成功すると、次のようなトップ画面が表示されます。

図 L-43 Web アプリケーションのサンプルプログラムの実行例



The screenshot shows a simple web interface with a title bar 'Servlet and JSP Examples'. Below the title, there are two menu items: 'Servlet Examples' and 'JSP Examples', both in blue text.

(h) アプリケーションのアンデプロイ

「付録 L.1(6) アプリケーションの停止、削除」を参照してください。

(i) J2EE サーバの終了

「付録 L.1(7) J2EE サーバの終了」を参照してください。

付録 L.8 TP1 インバウンド連携機能のサンプルプログラム

TP1 インバウンド連携機能を使用するサンプルプログラムは、次のディレクトリに格納されます。

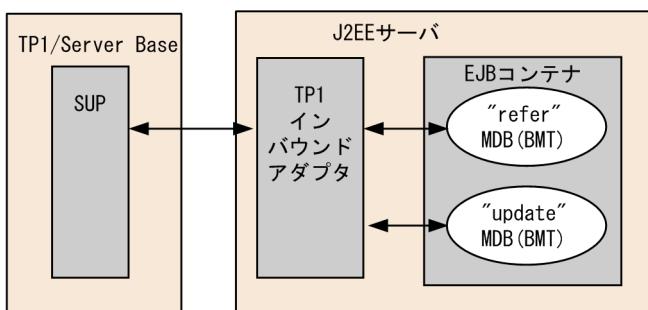
<Application Server のインストールディレクトリ>¥CC¥examples¥tpl inbound

サンプルプログラムの概要と実行手順を説明します。

(1) TP1 インバウンド連携機能のサンプルプログラムの概要

TP1 インバウンド連携機能のサンプルプログラムの構成を次に示します。

図 L-44 TP1 インバウンド連携機能のサンプルプログラムの構成



注 MDBはMessage-driven Beanを示します。

このサンプルプログラムは、TP1/Server Base が提供する Base サンプルの C 言語の SUP から RPC 要求電文を受け付け、Base サンプルの C 言語の SPP と同等の処理を実行する Message-driven Bean のプログラムです。Base サンプルの C 言語の SPP と同等の処理とは、RPC 要求電文の受け付け時に Message-driven Bean が実行され、実行結果が RPC 応答電文として C 言語の SUP に返される処理を指します。

Message-driven Bean の処理は、Base サンプルの C 言語の SPP と同様に顧客情報の取得や更新を実施します。

サンプルプログラムでは、メモリ上のテーブルに顧客情報を格納します。サンプルプログラムで使用するテーブル構成を「[表 L-4 サンプルプログラムで使用するテーブルの構成](#)」に、テーブルの内容を「[表 L-5 サンプルプログラムで使用するテーブルの内容](#)」に示します。

表 L-4 サンプルプログラムで使用するテーブルの構成

カラム	説明
name	名前
sex	性別
age	年齢
sale	販売額

表 L-5 サンプルプログラムで使用するテーブルの内容

name	sex	age	sale
Suzuki	M	30	1000000
Okada	F	23	1500000
Yoshida	F	26	800000
Saitoh	M	24	1000000
Itoh	M	35	1800000
Nishikawa	M	20	300000
Katoh	F	28	1000000
Satoh	F	27	2100000
Hasegawa	M	25	600000
Watanabe	M	31	1600000

(2) TP1 インバウンド連携機能のサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) TP1/Server Base の Base サンプルの設定

TP1/Server Base の Base サンプルに対して、次の設定を実施してください。

- ユーザサービスネットワーク定義によるスケジューラダイレクトの設定

TP1 インバウンドアダプタをユーザサービスネットワーク定義によるスケジューラダイレクト機能を使って呼び出すために、OpenTP1 のユーザサービス定義、ユーザサービスネットワーク定義の定義を設定します。設定する内容について次に示します。

- ユーザサービスネットワーク定義に、TP1 インバウンドアダプタのサービスグループ名、ホスト名、およびポート番号を指定します。設定項目について次の表に示します。

表 L-6 ユーザサービスネットワーク定義の設定項目

設定項目	設定する内容
サービスグループ名	Base サンプルで指定しているサービスグループ名「base_svg」を指定します。
ホスト名	アプリケーションサーバのホスト名を指定します。※
ポート番号	TP1 インバウンドアダプタの Connector 属性ファイルの scd_port プロパティで定義したポート番号を指定します。

注※ OpenTP1 では、RPC の送信元と送信先の IP アドレスを比較し、同一マシンである場合は UNIX ドメイン通信を、異なるマシンである場合は INET ドメイン通信をします。TP1 インバウンドアダプタでは、INET ドメイン通信だけをサポートするため、OpenTP1 と TP1 インバウンドアダプタを同一のマシンで利用する場合は、IP alias 機能によって割り振った送信元とは異なる IP アドレスをホスト名に指定する必要があります。

ユーザサービスネットワーク定義の例

```
dcsvgdef -g base_svg -h ホスト名[:ポート番号]
```

- ユーザサービス定義の rpc_destination_mode に definition を設定します。これによって、dc_rpc_call 関数を呼び出す際に、ユーザサービスネットワーク定義の指定値から宛先を検索するようになります。

ユーザサービス定義の例

```
set rpc_destination_mode = definition
```

なお、詳細については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

・性能解析トレースの設定

TP1 インバウンドアダプタで OpenTP1 の性能解析トレースの情報を引き継ぐためには、システム共通定義に性能解析トレースの設定を実施する必要があります。設定内容について次の表に示します。

表 L-7 性能解析トレースの設定内容

設定内容	設定の目的
set prf_trace=Y	性能検証用トレース情報を取得するために設定します。
set rpc_prf_trace_level=00010001	OpenTP1 の PRF 情報を Application Server に伝播するために設定します。

システム共通定義については、マニュアル「OpenTP1 システム定義」を参照してください。

(b) サンプルプログラムのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して、サンプルプログラムをコンパイルします。このバッチファイルで、EAR ファイルが作成されます。

(c) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(d) J2EE サーバのカスタマイズ

「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(e) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(f) TP1 インバウンドアダプタのインポート, 設定, および開始

TP1 インバウンドアダプタのインポートおよび設定方法については、「[付録 L.1\(4\) リソースアダプタの設定](#)」を参照してください。ただし、参照先の記述のうち、cjtestres コマンドを使用したリソースアダプタの接続テストは不要です。接続テストを実施するとメッセージ KDJ48606-E が発生します。TP1 インバウンドアダプタの設定では、Connector 属性ファイルの service_group プロパティには Base サンプルで指定しているサービスグループ名「base_svg」を指定します。

(g) アプリケーションのデプロイ

「[付録 L.1\(5\) アプリケーションのインポート, 設定, および開始](#)」を参照してください。

(h) SUP の実行

TP1/Server Base で提供している Base サンプルを参照してください。

(i) アプリケーションのアンデプロイ

「[付録 L.1\(6\) アプリケーションの停止, 削除](#)」を参照してください。

(j) J2EE サーバの終了

「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。

(3) サンプルプログラムの注意事項

ここでは、サンプルプログラムを使用するときの注意事項について説明します。

- TP1 インバウンドアダプタ接続関連の設定について

このサンプルプログラムは、リソースとして TP1 インバウンドアダプタを使用するため、TP1 インバウンドアダプタ接続に関する設定を適切に実施してください。

- リソースのリンク解決について

TP1 インバウンドアダプタのデフォルトの表示名を変更しなければ、リンク解決は不要です。TP1 インバウンドアダプタのデフォルトの表示名を変更した場合は、リンク解決が必要になります。リンク解決の方法を次に示します。

- サンプルで提供する cosminexus.xml の<message-ref>にリソースの表示名を設定します。設定方法については、マニュアル「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. 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。

- アプリケーションをデプロイします。

- サンプルプログラムを実行するプラットフォームについて

このサンプルプログラムは、エンディアンを意識したユーザデータの作成、取得は実施しません。このため、エンディアンが同一のプラットフォームで SUP を実行してください。エンディアンが異なるプラットフォームで SUP を実行すると、サンプルプログラムの「販売額」が不正な値となります。

付録 L.9 JSF のサンプルプログラム

JSF を使用するサンプルプログラムは、次のディレクトリに格納されています。

<Application Server のインストールディレクトリ>¥CC¥examples¥jsf

サンプルの概要と実行手順を説明します。

(1) JSF のサンプルプログラムの概要

このサンプルプログラムでは、次の動作をする JSF のアプリケーションを提供しています。

- 文字列を入力してボタンをクリックすると、画面に入力した文字列が表示されます。
- 入力した文字列が 15 文字を超えている場合にボタンをクリックすると、画面に「Length is greater than allowable maximum of 15」が表示されます。

JSF のサンプルプログラムでは、JSF を使用します。JSF を使用するためには、usrconf.cfg にクラスパスを設定する必要があります。

JSF のサンプルプログラムの実行例を次の図に示します。

図 L-45 JSF のサンプルプログラムの実行例

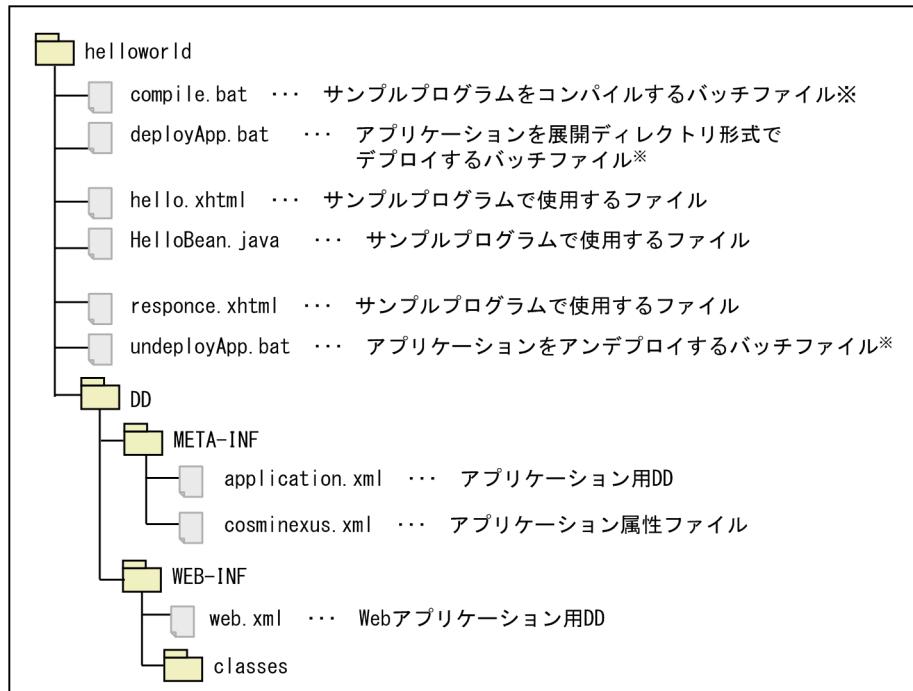


What is your name?

Submit

サンプルプログラムの構成を次の図に示します。

図 L-46 JSF のサンプルプログラムの構成



注※ UNIX版ではシェルスクリプトです。拡張子はありません。

(2) JSF のサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) J2EE サーバのセットアップ

「[付録 L.1\(1\) J2EE サーバのセットアップ](#)」を参照してください。

(b) J2EE サーバのカスタマイズ

「[付録 L.1\(2\) J2EE サーバのカスタマイズ](#)」を参照してください。

(c) J2EE サーバの起動

「[付録 L.1\(3\) J2EE サーバの起動](#)」を参照してください。

(d) サンプルプログラムのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して、サンプルプログラムをコンパイルします。このバッチファイルで、WAR ファイル (helloworld.war) と EAR ファイル (helloworld.ear) が作成されます。

(e) アプリケーションデプロイ

サンプルで提供するバッチファイル (deployApp.bat) を使用して、アプリケーションをデプロイします。「[付録 L.1\(5\) アプリケーションのインポート、設定、および開始](#)」を参照してください。

(f) アプリケーションの実行

ブラウザを起動し、URL に次の文字列を指定します。

```
http://<マシン名>:<ポート番号>/ helloworld/
```

J2EE サーバの NIO HTTP サーバに直接アクセスする場合、デフォルトでは次の URL を指定します。

```
http://localhost:8008/helloworld/
```

(g) アプリケーションのアンデプロイ

「[付録 L.1\(6\) アプリケーションの停止、削除](#)」を参照してください。

(h) J2EE サーバの終了

「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。

付録 L.10 Bean Validation および CDI 連携アプリケーションのサンプル

Bean Validation および CDI 連携アプリケーションを使用するサンプルプログラムは、次のディレクトリに格納されています。

<Application Server のインストールディレクトリ>/¥CC¥examples¥bv_cdi

サンプルの概要と実行手順を説明します。

(1) Bean Validation および CDI 連携アプリケーションのサンプルプログラムの概要

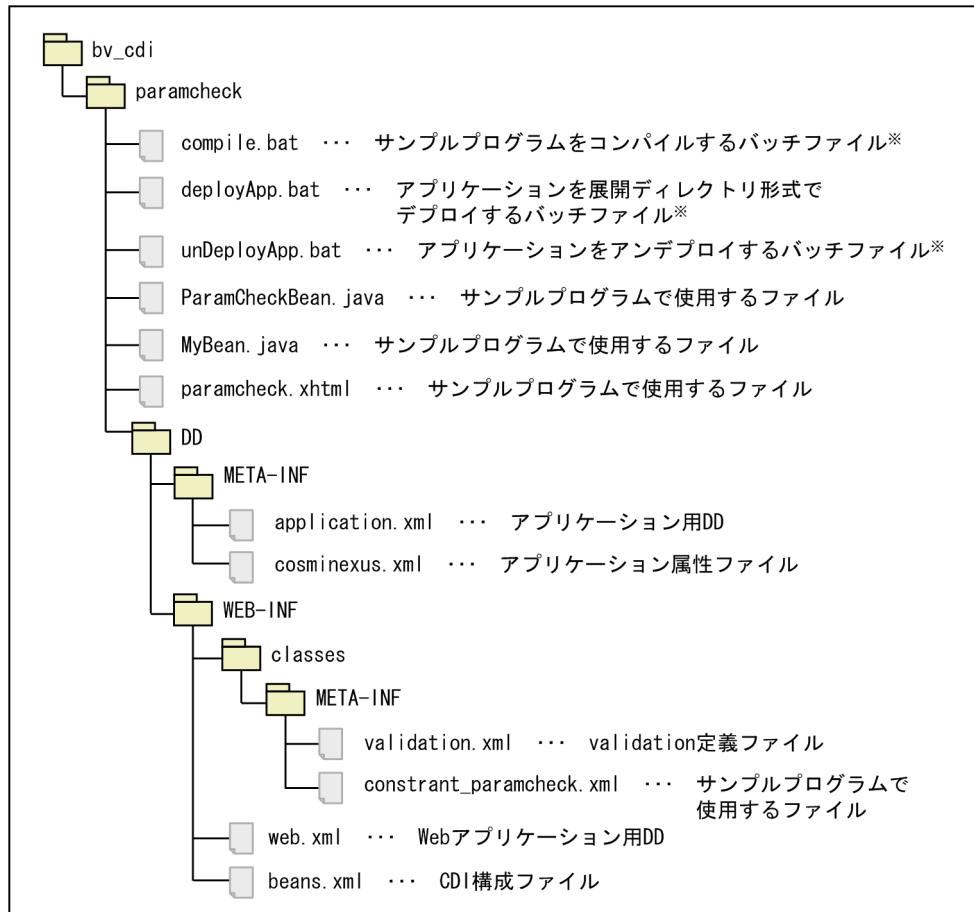
このサンプルプログラムでは、Bean Validation、CDI、および JSF を組み合わせたサンプルを提供しています。

Bean Validation および CDI アプリケーションのサンプルでは、実行前に次に示す設定を実施する必要があります。

- JSF を使用するための usrconf.cfg へのクラスパスの設定
 - CDI を使用するための server.policy へのアクセス権の設定
- server.policy については、マニュアル「[アプリケーションサーバリファレンス 定義編\(サーバ定義\)](#)」の「[2.2.4 server.policy \(J2EE サーバ用セキュリティポリシーファイル\)](#)」を参照してください。

<Application Server のインストールディレクトリ>/¥CC¥examples¥bv_cdi 以下の構成を次に示します。

図 L-47 bv_cdi ディレクトリ以下の構成



注※ UNIX版ではシェルスクリプトです。拡張子はありません。

(2) Bean Validation および CDI 連携アプリケーションのサンプルプログラムの実行手順

サンプルプログラムの実行手順を示します。

(a) サンプルプログラムのコンパイル

サンプルで提供するバッチファイル (compile.bat) を使用して、サンプルプログラムをコンパイルします。このバッチファイルで、WAR ファイル (paramcheck.war) と EAR ファイル (paramcheck.ear) が作成されます。

(b) J2EE サーバのセットアップ

「付録 L.1(1) J2EE サーバのセットアップ」を参照してください。

(c) J2EE サーバのカスタマイズ

「付録 L.1(2) J2EE サーバのカスタマイズ」を参照してください。

(d) J2EE サーバの起動

「付録 L.1(3) J2EE サーバの起動」を参照してください。

(e) アプリケーションのデプロイ

「付録 L.1(5) アプリケーションのインポート, 設定, および開始」を参照してください。

(f) アプリケーションの実行

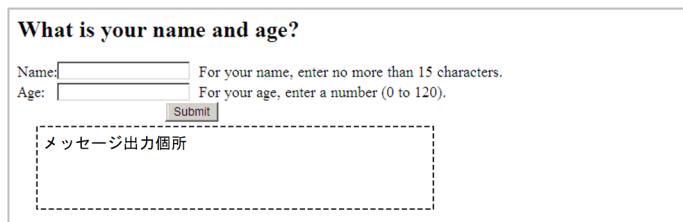
ブラウザを起動し, URL に次の文字列を指定して, アプリケーションを実行します。

http://<マシン名>:<ポート番号>/paramcheck/

ポート番号をカスタマイズしている場合は, カスタマイズした<ポート番号>を指定してください。

Bean Validation および CDI 連携アプリケーションのサンプルプログラムの実行例を次に示します。

図 L-48 CDI 連携アプリケーションのサンプルプログラムの実行例



What is your name and age?

Name: For your name, enter no more than 15 characters.

Age: For your age, enter a number (0 to 120).

Submit

メッセージ出力個所

[Submit] ボタンをクリックすると, メッセージ出力個所に検証結果が表示されます。

出力されるメッセージの条件および出力メッセージを次の表に示します。

表 L-8 出力されるメッセージの条件および出力メッセージ

項番	条件	出力メッセージ ^{※1}
1	検証に成功した場合	Validation Successful!
2	Name テキストボックスに 16 文字以上の文字列が入力されている場合 ^{※2}	size must be between 0 and 15 The name must be no more than 15 characters.
3	Age テキストボックスに 0 より小さい値が入力されている場合 ^{※2}	must be greater than or equal to 0 The age is less than the minimum value. Enter a number from 0 through 120.
4	Age テキストボックスに 120 より大きい値が入力されている場合 ^{※2}	must be less than or equal to 120 The age exceeds the maximum value. Enter a number from 0 through 120.
5	Age テキストボックスに数値に変換できない値が入力されている場合 ^{※2}	For your age, enter a number (0 to 120).
6	検証処理で例外が発生した場合	Validation Failed.

注※1 検証に失敗した原因が複数ある場合は、それぞれのメッセージが連続して出力されます。

注※2 これらの条件で出力されるメッセージは、項番 6 の「検証処理で例外が発生した場合」は出力されません。

(g) アプリケーションのアンデプロイ

「[付録 L.1\(6\) アプリケーションの停止、削除](#)」を参照してください。

(h) J2EE サーバの終了

「[付録 L.1\(7\) J2EE サーバの終了](#)」を参照してください。

付録 M 用語解説

マニュアルで使用する用語について

マニュアル「アプリケーションサーバ & BPM/ESB 基盤 用語解説」を参照してください。

索引

A

account の概要 348
adder の概要 349
AllText パラメタでの httpsd.conf の直接設定 147
Application Server のディレクトリ構成 237

B

bank の概要 349

C

cart の概要 350
checker の概要 350
CJMEMDUMP_PATH (Windows の場合) 129
CLI コマンド 328
cmp20product の概要 351
cmxdefcombinedmodel.xml 145
Component Container 管理者 115
Component Container 管理者を設定する (UNIX の場合) 115
Component Container 管理者を設定するときの注意事項 (UNIX の場合) 116
Component Container のディレクトリ構成 240
confirmer の概要 352
Connector 属性ファイルの取得と編集 [DB Connector 以外の場合] 155
Connector 属性ファイルの取得と編集 [DB Connector の場合] 152
Cookie スイッチング機能 203

D

DB Connector 以外のリソースアダプタを設定する (CUI 利用時) 154
DB Connector を設定する (CUI 利用時) 152

E

Eclipse での操作 26
enroller の概要 352
Enterprise Bean (EJB2.0) のサンプル実行手順 346

Enterprise Bean (EJB2.0) のサンプルの概要 348
Enterprise Bean (EJB2.0) のサンプルプログラム 341
Enterprise Bean (Timer Service) のサンプルプログラム 365
Enterprise Bean (アノテーション) のサンプルプログラム 360
Enterprise Bean (アノテーション) のサンプルプログラムの実行手順 363
Explicit ヒープのメモリサイズの設定 80

H

HTTP Server 動作確認用 URL の設定 140
HTTP Server の動作確認の設定 139

J

J2EE アプリケーションのプロパティについて 157
J2EE アプリケーションのメソッドタイムアウトの設定 50
J2EE アプリケーションを実行するシステムの構成情報と簡易構築定義ファイルの定義との対応 143
J2EE サーバの作業ディレクトリ 247
J2EE サーバの作業ディレクトリの構成 247
J2EE サーバを複数配置してマルチテナント対応するには 20
J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステム 20
J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構成例 20
J2EE サーバを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否 20
J2EE サーバを複数配置してマルチテナントに対応する 59
JavaBeans リソースのサンプルプログラム 367
JavaVM のメモリチューニング 78
JavaVM のメモリチューニングを反映する 78
Java ヒープおよび Explicit ヒープのチューニング 78
Java ヒープ全体のメモリサイズの設定 79

JIS X0213:2004 に含まれる Unicode の補助文字を使用する場合の注意事項 27

JP1/SC/DPM を使ったディスク複製インストール 271

jsptag の概要 353

jsptobean の概要 354

L

localladder の概要 355

M

Management Server が CTM ドメインマネージャに自動で設定する内容 262

Management Server が CTM に自動で設定する内容 263

Management Server が J2EE サーバに自動で設定する内容 263

Management Server が Web サーバに自動で設定する内容 265

Management Server が自動で生成するユーザ定義ファイル 260

Management Server が自動で設定する内容 260

Management Server が自動で設定する論理サーバの環境変数 (UNIX の場合) 131

Management Server が自動で設定する論理サーバの環境変数 (Windows の場合) 130

Management Server がバッチサーバに自動で設定する内容 264

Management Server の動作環境の設定 136

Management Server のプロパティの設定 135

Management Server を起動する 141

Management Server を使用するために設定する情報 135

Management Server を停止する 163

message の概要 355

Metaspace 領域のメモリサイズの設定 80

O

order の概要 356

OS の起動と同時に運用管理機能を自動起動するための設定 45

P

product の概要 356

S

salesrep の概要 357

shipper の概要 358

Smart Composer 機能 15

Smart Composer 機能で実現する運用形態 109

storagebin の概要 358

T

teller の概要 359

TP1 インバウンド連携機能のサンプルプログラム 382

TZ (OS 共通) 129

U

URL グループ単位の同時実行数の設定 85

W

warehouse の概要 360

Web アプリケーションのサンプルプログラム 378

Web コンテナまたは Web アプリケーション単位の同時実行数の設定 82

Web サーバを別ホストに配置してマシン性能を向上するシステムの構築 110

Web システム間での負荷分散機の共有 203

Web システムの複製 60

Windows [注意事項] 26

X

XDM/RD E2 の設定 335

あ

アプリケーションサーバが提供するコマンドの使用 26

アプリケーションサーバが提供する定義ファイルの更新 26

アプリケーションサーバでサポートしているシンボリックリンク 120

アプリケーションサーバの機能を有効にするための設定 146

アプリケーションサーバの構築 30

い

移行先の環境でのホストの準備 177

移行先のシステム環境の構築 179

移行時に使用する構成・定義情報の内容 179

入れ替え後の J2EE アプリケーションの開始 77

入れ替え後の J2EE アプリケーションの設定 77

入れ替え前の J2EE アプリケーションの削除 76

入れ替え前の J2EE アプリケーションの停止 76

インストール後のディレクトリ構成 237

う

運用 70

運用管理エージェントによる論理サーバの監視に関する設定 138

運用管理エージェントのプロパティの設定 138

運用管理エージェントを起動する 141

運用管理エージェントを使用するために設定する情報 138

運用管理エージェントを停止する 163

運用管理機能の管理ユーザアカウントの設定を変更する 98

運用管理機能のツール 13

運用管理機能の停止 74

運用管理機能を構築する 134

運用管理ドメイン内の論理サーバの稼働状況の確認 72

運用管理ポータル 15

運用管理ポータルで実現する運用形態 109

運用管理ポータルの操作で失敗したときの対処方法 258

か

カスタムセットアップ 14

可用性を確保した Web フロントシステム 18

可用性を確保した Web フロントシステムの構成例 19

可用性を確保した Web フロントシステムの構築時に実施する作業と作業の要否 19

簡易構築定義ファイル 141

簡易構築定義ファイルの構成 141

簡易構築定義ファイルの作成例 149

簡易構築定義ファイルを設定する 145

簡易構築定義ファイルを設定するときの注意事項 (J2EE アプリケーション使用時) 146

簡易構築定義ファイルを設定するときの注意事項 (バッチアプリケーション使用時) 194

環境構築／運用時の注意事項 27

環境変数 128

管理者特権で実行する必要がある操作 26

管理ユーザアカウントの設定変更 98

管理ユーザアカウントを設定しない場合の設定 137

き

業務アプリケーションとの関連づけの設定 66

業務アプリケーションの開始 (GUI 利用時) 55

業務アプリケーションの設定 (GUI 利用時) 54

業務アプリケーションの設定を変更する (CUI 利用時) 160

業務アプリケーションを入れ替える 75

業務アプリケーションを設定して開始する (CUI 利用時) 156

業務アプリケーションを設定して開始する (GUI 利用時) 53

こ

更新インストール 267

構成変更定義ファイルの作成 170

構成・定義情報の出力 179

構成・定義情報を出力してシステムを移行するときの注意事項 182

構築 28

コンフィグレーション定義で使用するタグと適用範囲 143

さ

サーバ管理コマンド実行時に設定する情報 157

サーバ管理コマンドによるサンプル実行環境の構築、運用手順 339

サーバ管理コマンドを使用する操作 157

サービスユニットの削除 175

サービスユニットの追加 171

サービスユニットの停止 175
作業ディレクトリに関する注意事項 256
作業ディレクトリのパス長の見積もり式 [J2EE サーバ] 253
作業ディレクトリのパス長の見積もり式 [バッチサーバ] 256
削除対象のシステム環境の内容確認 96
サンプルプログラム一覧 (Enterprise Bean の種類と動作内容) 341
サンプルプログラムで使用する JMS の設定 345
サンプルプログラムで使用するデータベースの設定 344
サンプルプログラムで使用するメールの設定 345
サンプルプログラムのクライアントの種類 346
サンプルプログラムの使用 26

し

システム環境定義 (簡易構築定義ファイルのパラメタ) を変更する (CUI 利用時) 158
システム環境定義の設定 49
システム環境のセットアップ 45
システム環境をアンセットアップする 99
システム環境を構築する 151
システム環境を削除する 164
システム環境をスケールアウトする (CUI 利用時) 169
システム環境をスケールインする (CUI 利用時) 174
システム環境をセットアップする 44
システム環境をセットアップするときの注意事項 46
システム構築の概要 17
システムの運用 70
システムの稼働状況を確認する (GUI 利用時) 72
システムの環境変数を設定する 129
システムの構成・定義情報 93
システムの構成・定義情報を取得する 94
システムの構築 28
システムの構築・運用時に使用する画面 222
システムの構築・運用時に使用するコマンド 225
システムの構築・運用時に使用するファイル 229
システムの情報モデルの変更 171

システムの設定変更 159
システムを一括構築するときの注意事項 (負荷分散機制御をする場合) 206
システムを起動して業務を開始する (GUI 利用時) 71
システムを起動する (CUI 利用時) 151
システムを停止する (CUI 利用時) 163
システムを停止する (GUI 利用時) 73
障害運用 23
新規インストール 30

す

スケールアウト 167
スケールイン 173

せ

接続方法に API を使用するときの注意事項 205
セットアップウィザード 13
セットアップウィザードで使用するファイルの一覧 48
セットアップウィザードで設定されるパラメタ 233
セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ (論理 J2EE サーバ) 234
セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ (論理 Web サーバ) 233
セットアップウィザードでパラメタの省略値と異なる値が設定されるパラメタ (論理パフォーマンストレーサ) 236
セットアップウィザードの起動 45

た

タイムアウトの設定 49
タイムアウトを設定する 49
単一の Web フロントシステム 17
単一の Web フロントシステムの構成例 17
単一の Web フロントシステムの構築時に実施する作業と作業の要否 18
単一の Web フロントシステムを構築する 44
単一の Web フロントシステムを構築するには 17

つ

通常運用 23

て

- データベースサーバの構築 29
- データベース接続環境を設定する (HiRDB の設定) 121
- データベース接続環境を設定する (MySQL の設定) 126
- データベース接続環境を設定する (Oracle の設定) 123
- データベース接続環境を設定する (PostgreSQL の設定) 127
- データベース接続環境を設定する (SQL Server の設定) 124
- データベース接続環境を変更する 91
- データベースの環境を設定する (Oracle の設定) 114
- データベースを準備する 29

と

- 同一構成のホストを複数配置して可用性を確保した Web フロントシステム 18
- 同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構成例 19
- 同一構成のホストを複数配置して可用性を確保した Web フロントシステムの構築時に実施する作業と作業の要否 19
- 同一構成のホストを複数配置して可用性を確保する 56
- 同一構成のホストを複数配置して可用性を確保するには 18
- 同一構成のホストを複数配置するときの注意事項 57
- 動作確認用 HTTP メソッドの設定 140
- 同時実行数の最適化 78
- 同時実行数を最適化するためのチューニングを反映する 81
- トランザクションのトランザクションタイムアウト時間のデフォルトの設定 51

は

- バッチアプリケーションのスケジューリング機能を使用するための設定 195
- バッチアプリケーションを実行するシステムを運用するときの注意事項 199
- バッチサーバで使用しない定義 198

バッチサーバとして動作するための設定 194

バッチサーバの作業ディレクトリ 254

バッチサーバの作業ディレクトリの構成 254

バッチサーバを複数配置するときの注意事項 197

パフォーマンスチューニング 78

ひ

標準セットアップ 14

ふ

- 負荷分散機 200, 202
- 負荷分散機の API 328
- 負荷分散機の CLI コマンド 328
- 負荷分散機の冗長化 204
- 負荷分散機の制御方法 203
- 負荷分散機へ接続する環境を設定する 205
- 負荷分散機への接続の設定 205
- 負荷分散機を共有するときの注意事項 206
- 負荷分散機を準備する 204
- 負荷分散機を冗長化するときの注意事項 207
- 負荷分散機を制御するときの注意事項 206
- 複製情報の出力 56
- 複製情報を使ったセットアップ 56
- プログラムのインストール 26

ほ

保守運用 23

ホスト内の論理サーバの稼働状況の確認 73

ま

- マルチテナント対応する Web フロントシステム [リソースアダプタの複数配置] 21
- マルチテナントに対応した Web フロントシステムの構成例 [J2EE サーバの複数配置] 20
- マルチテナントに対応した Web フロントシステムの構成例 [リソースアダプタの複数配置] 21
- マルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否 [J2EE サーバの複数配置] 20

マルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否 [リソースアダプタの複数配置] 22

マルチテナントに対応した Web フロントシステム [J2EE サーバの複数配置] 20

イ

ユーザログ機能のサンプルプログラム 370

リ

リクエストで使用する場合の注意事項 27

リソースアダプタの接続テスト [DB Connector 以外の場合] 155

リソースアダプタの接続テスト [DB Connector の場合] 154

リソースアダプタの設定 (データベースの接続環境) を変更する (CUI 利用時) 161

リソースアダプタの設定 [DB Connector 以外の場合] 154

リソースアダプタの設定 [DB Connector の場合] 153

リソースアダプタの追加 65

リソースアダプタのプロパティの設定変更 92

リソースアダプタまたは業務アプリケーションを操作するときの確認事項 (CUI 利用時) 157

リソースアダプタを開始する (CUI 利用時) 155

リソースアダプタを複数配置してマルチテナント対応する Web フロントシステム 21

リソースアダプタを複数配置してマルチテナント対応するには 21

リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構成例 21

リソースアダプタを複数配置してマルチテナントに対応した Web フロントシステムの構築時に実施する作業と作業の要否 22

リソースアダプタを複数配置してマルチテナントに対応する 64

リモートインストール 275

れ

レスポンスタイムアウトおよびリクエスト間隔タイムアウトの設定 (HiRDB の場合) 52

レスポンスデータ待ちの通信タイムアウトの設定 50

ろ

論理サーバ (Web サーバ, J2EE サーバおよびパフォーマンストレーサ) の追加 60

論理サーバごとの稼働状況の確認 73

論理サーバの環境のアンセットアップ 175

論理サーバの環境変数を設定する 129

論理サーバの環境変数を設定するときの確認事項 130

論理サーバの起動 71

論理サーバのコンフィグレーション定義時に使用する簡易構築定義ファイルの主なタグ 144

論理サーバのコンフィグレーション定義の適用範囲 (J2EE アプリケーションを実行するシステム) 144

論理サーバの停止 74

論理サーバの動作確認間隔の設定 139

論理サーバの動作確認タイムアウトの設定 139

論理サーバの動作確認の開始時間の設定 139

論理サーバの動作確認リトライの設定 139

論理サーバの動作確認レベルの設定 138