



Adobe

LiveCycle™ ES2 のインストールの準備 (シングルサーバー)

2010年9月24日

Adobe® LiveCycle™ ES2

バージョン9

© 2010 Adobe Systems Incorporated and its licensors. All rights reserved.

Adobe® LiveCycle® ES2 のインストールの準備 (シングルサーバー)

2010 年 9 月 24 日

This guide is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the guide for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the guide; and (2) any reuse or distribution of the guide contains a notice that use of the guide is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Adobe, the Adobe logo, Adobe Reader, Acrobat, Flash, Flex, Flex Builder, FrameMaker, LiveCycle, PageMaker, Photoshop, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. AIX and IBM are trademarks of International Business Machines Corporation in the United States, other countries, or both. Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries. Oracle, Java, Solaris, and Sun are trademarks or registered trademarks of Oracle and/or its affiliates. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Oracle Corporation. Red Hat is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Microsoft, SharePoint, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Novell and SUSE are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. UNIX is a trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. All other trademarks are the property of their respective owners.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

目次

このドキュメントの内容.....	6
事前準備.....	6
このドキュメントの内容.....	6
このドキュメントの対象読者.....	6
1 必要システム構成.....	10
LiveCycle ES2 のパッチアップデート.....	10
サードパーティのインフラストラクチャのサポート.....	10
サードパーティのパッチのサポート.....	10
Microsoft Windows オペレーティングシステム.....	10
Linux オペレーティングシステム.....	13
IBM AIX.....	15
Sun Solaris.....	15
必要システム構成.....	16
最小ハードウェア要件.....	16
Intel x86 の互換性.....	18
推奨ハードウェア要件.....	18
AIX、Linux、および Solaris に関するその他の要件.....	18
LiveCycle Content Services ES2 に関するその他のハードウェア要件.....	20
クライアント側コンピュータの推奨ハードウェア要件.....	20
サポートされているソフトウェア.....	21
Windows でのインストールユーザーアカウント.....	25
64 ビット Windows インストール用の設定.....	25
データベースを使用するための最小限のユーザー権限.....	26
Web ブラウザのサポート.....	27
PDF Generator ES2 および PDF Generator 3D ES2 に関するその他の要件.....	29
Connector for IBM Content Manager に関するその他の要件.....	35
LiveCycle ES2 Connector for IBM FileNet に関するその他の要件.....	37
LDAP の設定.....	37
グローバルドキュメントストレージディレクトリ.....	38
グローバルドキュメントストレージディレクトリの計画と作成.....	38
グローバルドキュメントストレージディレクトリの場所.....	38
グローバルドキュメントストレージディレクトリのサイズ決定要因.....	39
グローバルドキュメントストレージディレクトリの保護.....	39
グローバルドキュメントストレージディレクトリのバックアップ.....	39
LiveCycle Business Activity Monitoring ES2 の要件.....	40
基本要件.....	40
Business Activity Monitoring ES2 データベースの作成.....	41
JBoss の要件 (BAM).....	41
WebLogic の要件 (BAM).....	42
WebSphere の要件 (BAM).....	42

インストールに関する考慮事項	43
ウイルススキャンの無効化 (Windows のみ)	43
ネットワークドライブからのインストール	43
LiveCycle ES2 と Luna HSM クラスタの併用	43
手動による Acrobat の使用制限	43
一時ディレクトリ	43
LiveCycle ES2 IPv6 のサポート	44
CIFS を有効にするためのサーバー設定	47
仮想インターフェ이스の作成 (AIX、Linux、および Solaris のみ)	48
CIFS 用の Windows Server の設定	48
ドキュメントフォーム変数および電子署名を使用するプロセス	49
2 LiveCycle ES2 の秘密鍵証明書と公開鍵証明書	52
Reader Extensions ES2 の使用権限秘密鍵証明書の取得	52
Digital Signatures ES2 で使用する電子証明書の取得	53
3 LiveCycle ES2 データベースの作成	54
データベース設定の要件	54
Oracle データベースの作成	54
ユーザーアカウントと権限	55
SQL Server データベースの作成	56
SQL Server データベースドライバのインストール	56
LiveCycle ES2 用の SQL Server の設定	56
SQL Server データベースのサイズ変更	58
LiveCycle ES2 データベースユーザー、スキーマおよびログインの作成	58
DB2 データベースの作成	60
DB2 ユーザーアカウント	60
DB2 に関する LiveCycle Content Services ES2 のその他の要件	65
MySQL データベースの作成	65
MySQL データベースの場合のプロセスでのドキュメントの使用に関する制限	66
MySQL ユーザーアカウント	66
MySQL 初期サーバー設定についての要件	66
4 JBoss Application Server の設定	68
アドビにより事前設定された JBoss	68
Sun Java JDK for JBoss のインストール	69
Linux および Solaris のログオンスクリプト	70
JBoss の起動と停止	70
JBoss アプリケーションサーバーの設定に関する注意事項	71
アドビの事前設定 JBoss 4.2.1 用 LiveCycle ES2 データベース接続の設定	71
アドビの事前設定 JBoss 4.2.1 用 MySQL の設定	72
アドビの事前設定 JBoss 4.2.1 用 Oracle の設定	74
アドビの事前設定 JBoss 4.2.1 用 SQL Server の設定	77
次の手順	80
5 WebLogic Server の設定	81
概要	81
WebLogic 用の LiveCycle ES2 の要件	81
WebLogic Server のインストール	82
Solaris (64 ビット Java) での WebLogic 10g R3 の設定	82
JRockit Java JDK のインストール	82
WebLogic Server ドメインの作成	84
ノードマネージャの起動	86

新しい WebLogic 管理対象サーバーの作成	86
WebLogic Server の設定に関する問題.....	88
Web サービス用の WebLogic の設定.....	88
WebLogic の停止と再起動.....	89
LiveCycle ES2 データベース接続の設定.....	90
次の手順.....	90
6 WebSphere Application Server の設定.....	91
WebSphere Application Server のインストール.....	91
WebSphere への Fix Pack のインストール.....	91
JAVA_HOME および PATH 環境変数の設定.....	91
WebSphere Base の起動と WebSphere Administrative Console へのアクセス.....	92
WebSphere ND の起動と WebSphere Administrative Console へのアクセス.....	93
ディレクトリ権限の設定.....	93
WebSphere Application Server の準備.....	94
必要なアプリケーションサーバーインスタンス.....	94
SOAP 要求タイムアウト値の延長.....	94
デプロイヤーのヒープサイズの拡大.....	95
LiveCycle ES2 データベース接続の設定.....	96
次の手順.....	96
7 事前設定環境のチェックリスト.....	97
A 付録 - JBoss の手動設定.....	99
JBoss 用 JDK のインストール.....	99
JAVA_HOME 環境変数の設定 (Windows).....	99
PATH 環境変数の設定 (Windows).....	100
JAVA_HOME 環境変数の設定 (Linux および Solaris).....	100
PATH 環境変数の設定 (Linux および Solaris).....	100
JAVA_Home 環境変数の設定の検証 (Windows、Linux、または Solaris).....	100
JBoss の手動インストール.....	100
JBoss のダウンロード.....	101
JBoss の起動と停止.....	101
JBoss の起動.....	101
JBoss の停止.....	101
JBoss 設定の変更.....	102
JMS およびクラスタ化の設定ファイルの削除.....	102
クラスタ設定の無効化.....	102
JBoss 設定の変更.....	103
hibernate オプションの変更 (Content Services ES2 のみ).....	108
LiveCycle ES2 データベース接続.....	108
手動でインストールした JBoss での MySQL の設定.....	108
手動でインストールした JBoss 用の Oracle の設定.....	111
SQL Server の設定.....	114
次の手順.....	117

このドキュメントの内容

このドキュメントでは、Microsoft® Windows®、Red Hat® Linux®、SUSE™ Linux、IBM® AIX® および Sun™ Solaris™ で Adobe LiveCycle ES2 (Adobe LiveCycle Enterprise Suite 2) を自動または手動でインストールおよびデプロイする前に、サーバー環境を準備するために必要な情報を提供します。

事前準備

最新バージョンの [LiveCycle ES2 マニュアル](#) が手元にあることを確認してください。

このドキュメントの内容

このドキュメントには次の種類の情報が含まれています。

- LiveCycle ES2 のインストールプロセスを正常に実行するために必要なハードウェアおよびソフトウェアのすべての要件と設定。
- LiveCycle ES2 をインストールおよびデプロイせずに実行できるが、インストールまたはデプロイに必要なすべてのタスク。

Adobe LiveCycle Data Services ES2 のインストールについて詳しくは、Data Services ES2 モジュールに付属するドキュメントを参照してください。

このドキュメントの作業を完了したら、該当するアプリケーションサーバーの『LiveCycle ES2 のインストールおよびデプロイ』ドキュメントに進んでください。使用できるすべてのドキュメントのリストについては、[「関連情報」\(9 ページ\)](#) を参照してください。

このドキュメントの対象読者

このドキュメントは、LiveCycle ES2 をインストール、設定、管理およびデプロイする前に、開発、ステージングおよび実稼働環境でのアプリケーションサーバーとデータベースサーバーの準備を担当する管理者または開発者を対象にしています。このドキュメントで扱う内容は、JBoss、WebSphere、WebLogic Application Server、Red® Hat® Linux®、SUSE™ Linux、Microsoft® Windows®、IBM AIX® または Sun™ Solaris™ の各オペレーティングシステム、MySQL、Oracle®、IBM DB2® または SQL Server の各データベースサーバーおよび Web 環境に関する十分な知識がある読者を想定しています。

このドキュメントで使用する表記

このドキュメントで使用する一般的なファイルパスの命名規則は、次のとおりです。

名前	説明	デフォルト値
[LiveCycleES2 root]	すべての LiveCycle ES2 モジュールで使用するインストールディレクトリ。インストールディレクトリには、Adobe LiveCycle Configuration Manager、LiveCycle ES2 SDK およびインストールされた各 LiveCycle ES2 モジュールが (製品マニュアルと共に) サブディレクトリに分けられて格納されています。このディレクトリには、サードパーティのテクノロジーに関連したディレクトリも含まれます。	Windows : C:¥Adobe¥Adobe LiveCycle ES2¥ AIX、Linux および Solaris : /opt/adobe/adobe_livecycle_es2/
[appserver root]	LiveCycle ES2 サービスを実行するアプリケーションサーバーのホームディレクトリ。	Windows 上の JBoss 4.2.0 または 4.2.1 : C:¥jboss Linux および Solaris 上の JBoss 4.2.0 または 4.2.1 : /opt/jboss Windows 上の JBoss Enterprise Application Platform 4.3 : C:¥jboss-eap-4.3¥jboss-as Linux および Solaris 上の JBoss Enterprise Application Platform 4.3 : /opt/jboss-eap-4.3/jboss-as WebSphere (Windows) : C:¥Program Files¥IBM¥WebSphere¥AppServer WebSphere (Linux および Solaris) : /opt/IBM/WebSphere/AppServer WebSphere (AIX) : /usr/IBM/WebSphere/AppServer または /opt/IBM/WebSphere/AppServer WebLogic 10g (Windows) : C:¥bea¥wlserver_10.3 WebLogic 11g (Windows) : C:¥Oracle¥Middleware¥wlserver_10.3 WebLogic 10g (Linux および Solaris) : /opt/bea/wlserver_10.3 WebLogic 11g (Linux および Solaris) : /opt/Oracle/Middleware/wlserver_10.3

名前	説明	デフォルト値
WL_HOME	WL_HOME 環境変数に指定されている、WebLogic のインストールディレクトリ。	WebLogic 10g (Windows) : C:¥bea¥ WebLogic 11g (Windows) : C:¥Oracle¥Middleware¥ WebLogic 10g (Linux および Solaris) : /opt/bea/ WebLogic 11g (Linux および Solaris) : /opt/Oracle/
[appserverdomain]	WebLogic で設定したドメイン。デフォルトのドメインは base_domain です。	WebLogic 10g (Windows) : C:¥bea¥user_projects¥domains¥base_domain WebLogic 11g (Windows) : C:¥Oracle¥Middleware¥user_projects¥domains¥base_domain WebLogic 11g (Linux および Solaris) : /opt/Oracle/user_projects/domains/ base_domain
[dbserver root]	LiveCycle ES2 データベースサーバーがインストールされている場所。	データベースタイプとインストール時の設定によって異なります。

このドキュメントに記述されているディレクトリの場所に関するほとんどの情報は、すべてのプラットフォームに当てはまります (AIX、Linux および Solaris では、すべてのファイル名とパスにおいて大文字と小文字が区別されます)。プラットフォーム固有の情報は、必要に応じて特記します。

関連情報

次の表に、LiveCycle ES2 の学習に役立つ情報を示します。

情報	参照先
LiveCycle ES2 およびモジュールに関する一般的な情報	LiveCycle ES2 の概要
この LiveCycle ES2 リリースの新機能	LiveCycle ES2 の新機能
LiveCycle ES2 の用語	LiveCycle ES2 用語集
LiveCycle ES2 に統合できる他のサービスや製品	http://www.adobe.com/jp/products/livecycle
他の LiveCycle ES2 モジュール	http://www.adobe.com/jp/devnet/
LiveCycle ES2 のインストール	LiveCycle ES2 のインストールおよびデプロイ (JBoss 版) LiveCycle ES2 のインストールおよびデプロイ (WebSphere 版) LiveCycle ES2 のインストールおよびデプロイ (WebLogic 版)
自動オプションを使用した LiveCycle ES2 のインストール	LiveCycle ES2 の自動インストールおよびデプロイ
LiveCycle ES2 のすべてのドキュメント	Adobe LiveCycle ES2 ドキュメンテーション
LiveCycle ES2 リリース情報および製品に加えられた最新の変更	LiveCycle ES2 リリースノート
現在のバージョンに関するパッチアップデート、テクニカルノートおよび関連情報	LiveCycle テクニカルサポート

1 必要システム構成

サーバー環境を準備するために次のタスクを実行します。

1. インストールメディアが損傷していないことを確認します。インストールメディアからインストールするのではなく、LiveCycle ES2 をインストールする予定のコンピュータのハードディスクにインストーラファイルをコピーすることをお勧めします。
また、インストールメディアの MD5 チェックサムが、[アドビの Web サイト](#)に公開されているチェックサムと一致していることを確認します。
2. すべてのシステムコンポーネントのバージョンが必要システム構成を満たしていることを確認します。
3. 必要に応じてハードウェアを準備します。
4. オペレーティングシステムをインストールし、必要なすべてのパッチとサービスパックを適用して更新します。
5. データベースサーバーをインストールおよび設定します。
6. アプリケーションサーバーをインストールおよび設定します。

1.1 LiveCycle ES2 のパッチアップデート

LiveCycle ES2 をインストールする前に、[LiveCycle テクニカルサポート](#)にある必要なパッチアップデートをダウンロードします。

1.2 サードパーティのインフラストラクチャのサポート

1.2.1 サードパーティのパッチのサポート

このドキュメントで説明されるサードパーティの参照プラットフォームは、サードパーティのインフラストラクチャの特定のパッチレベルを示していますが、それは Adobe LiveCycle ES2 の本バージョンの開発およびリリース時において最新のものです。

サードパーティのパッチのサポートおよびソフトウェア互換性に関するアドビのポリシーについては、ナレッジベースの記事「[Adobe LiveCycle Third-party Patch Support Policy](#)」を参照してください。

1.2.2 Microsoft Windows オペレーティングシステム

LiveCycle ES2 は、次の Microsoft Windows オペレーティングシステムをサポートしています。

- Windows Server 2008 Standard および Enterprise Edition R2 (64 ビット)
- Windows Server 2008 Standard、Enterprise Edition SP1 以降 (64 ビット) および VMWare ESX/GSX のアーキテクチャ
- Windows Server 2003 Standard Edition および Enterprise Edition SP2 以降 (32 ビットおよび 64 ビット)
- Windows Server 2003 R2 Standard SP2 以降 (32 ビットおよび 64 ビット)
- 32 ビット、64 ビットおよび VMWare ESX/GSX のアーキテクチャで稼働する Windows Server 2003 R2 Enterprise Edition SP2 以降

注意： LiveCycle ES2 サーバーは Microsoft Windows XP (SP2 または SP3)、Windows Vista (SP1、全エディション、32 ビットおよび 64 ビット)、Windows 7 (32 ビットおよび 64 ビット) の実稼働以外の環境でサポートされます。ただし、これらのオペレーティングシステムは LiveCycle のクライアントに対して、および LiveCycle の開発者用ツールのインストールでサポートされています。

注意： LiveCycle Business Activity Monitoring ES2 は 32 ビットシステムでサポートされていません。Business Activity Monitoring ES2 は 64 ビットシステムにデプロイしてください。

1.2.2.1 Windows Server 2008

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
Red Hat JBoss Application Server 4.2.1 および JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨) Microsoft SQL Server 2005 SP3、2008 Oracle 10g、11g
Oracle WebLogic 10g R3 (Standard Edition および Enterprise Edition)	Oracle JRockit® JDK 6	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨) IBM DB2 9.1 以降の 9.x バージョン Oracle 10g、11g Microsoft SQL Server 2005 SP3、2008
9.0.0.2 の新機能 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® JDK 6	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨) IBM DB2 9.1 以降の 9.x バージョン Microsoft SQL Server 2005 SP3、2008 Oracle 10g、11g
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> IBM DB2 9.1 以降の 9.x バージョン Oracle 10g、11g MS SQL Server 2005 SP3、2008

注意： 64 ビットアプリケーションサーバーで PDF Generator ES2 を使用する場合は、追加の 32 ビット JVM をインストールする必要があります。

1.2.2.2 Windows Server 2003

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
Red Hat JBoss Application Server 4.2.0	Sun JDK 5.0 Update 11 (または 5.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨) Microsoft SQL Server 2005 SP3、2008 Oracle 10g、11g
Red Hat JBoss Application Server 4.2.1 および JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	32 ビット OS および 32 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨)
Red Hat JBoss Application Server 4.2.1 および JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> Microsoft SQL Server 2005 SP3、2008 Oracle 10g、11g MySQL 5.0.18 および 5.1.30
Oracle WebLogic 10g R3 (Standard Edition および Enterprise Edition)	Oracle JRockit® JDK 6	32 ビット OS および 32 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨)
Oracle WebLogic 10g R3 (Standard Edition および Enterprise Edition)	Oracle JRockit® JDK 6	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨) IBM DB2 9.1 以降の 9.x バージョン Oracle 10g、11g Microsoft SQL Server 2005 SP3、2008
9.0.0.2 の新機能 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® JDK 6	32 ビット OS および 32 ビット JVM	<ul style="list-style-type: none"> MySQL 5.1.30
9.0.0.2 の新機能 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® JDK 6	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> MySQL 5.0.18 または 5.1.30 (推奨)

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
IBM WebSphere 6.1.0.21 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g ● Microsoft SQL Server 2005 SP3、2008
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g ● Microsoft SQL Server 2005 SP3、2008

注意： 64 ビットアプリケーションサーバーで PDF Generator ES2 を使用する場合は、追加の 32 ビット JVM をインストールする必要があります。サポートされるソフトウェアについて詳しくは、[「サポートされているソフトウェア」\(21 ページ\)](#) を参照してください。

1.2.3 Linux オペレーティングシステム

LiveCycle ES2 のサポート対象は次のとおりです。

- Red Hat Enterprise Linux AP または ES 5 (Intel/AMD 64 ビットアーキテクチャ)
- SUSE Linux Enterprise Server 10.0 および 11.0 (Intel/AMD 64 ビットアーキテクチャ)

注意： PDF Generator 3D ES2 は Windows 以外のプラットフォームでサポートされていません。

注意： 64 ビットアプリケーションサーバーで PDF Generator ES2 を使用する場合は、追加の 32 ビット JVM をインストールする必要があります。

注意： Linux オペレーティングシステムの場合は、X Window ライブラリがインストールされている必要があります。この設定は PDF Generator ES2 および Forms ES2 には必須です。詳しくは、オペレーティングシステムのドキュメントを参照してください。

1.2.3.1 Red Hat Enterprise Linux

警告： フォームをレンダリングするときの問題を回避するために、cURL、Xorg-x11-apps、Compat-libstdc++ の各パッケージおよびすべての依存関係を Red Hat Linux コンピュータにインストールしてください。

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
Red Hat JBoss Application Server 4.2.0	Sun JDK 5.0 Update 11 (または 5.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● MySQL 5.0.18 または 5.1.30 (推奨) ● Oracle 10g、11g
Red Hat JBoss Application Server 4.2.1 および JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● MySQL 5.0.18 または 5.1.30 (推奨) ● Oracle 10g、11g

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
Oracle WebLogic 10g R3 (Standard Edition および Enterprise Edition)	Oracle JRockit® JDK 6	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● Oracle 10g、11g
9.0.0.2 の新機能 Oracle WebLogic 11g R1 - 10.3.3	Oracle JRockit® JDK 6	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● Oracle 10g、11g
IBM WebSphere 6.1.0.21 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g

1.2.3.2 SUSE Linux

注意：オペレーティングシステムに X Window ライブラリがインストールされていることを確認してください。この設定は PDF Generator ES2 および Forms ES2 には必須です。詳しくは、オペレーティングシステムのドキュメントを参照してください。

警告：SUSE Linux Enterprise Server に付属の glibc-locale-32bit ライブラリをインストールしないと、LiveCycle ES2 で PDF ファイルが生成されません。デフォルトでこのライブラリファイルはインストールされないため、インストールするには YaST を使用する必要があります (SUSE Linux Enterprise Server 10.0 のドキュメントを参照)。

注意：LiveCycle ES2 を SUSE Linux 11 にインストールする予定になっている場合は、libstdc++-libc6.2-2.so.3 ライブラリもインストールする必要があります。SUSE Linux 11 には、これらのライブラリがデフォルトでは含まれていません。詳しくは、[こちら](#)から Novell の Web ページを参照してください。これらのライブラリは、Adobe Central Pro Output Server を実行するために必要です。

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
Red Hat JBoss Application Server 4.2.1 および JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● MySQL 5.0.18 または 5.1.30 (推奨) ● Oracle 10g、11g
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g

1.2.4 IBM AIX

LiveCycle ES2 は AIX 5L 5.3 および AIX 6.1 (いずれも 64 ビットアーキテクチャ) をサポートします。

注意： PDF Generator 3D ES2 は Windows 以外のプラットフォームでサポートされていません。

注意： オペレーティングシステムに X Window ライブラリがインストールされていることを確認してください。この設定は PDF Generator ES2 および Forms ES2 には必須です。詳しくは、オペレーティングシステムのドキュメントを参照してください。

注意： サポートされているプラットフォームの詳細な一覧については、[「サポートされているソフトウェア」\(21 ページ\)](#) を参照してください。

1.2.4.1 AIX 5.3

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
IBM WebSphere 6.1.0.21 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none">● IBM DB2 9.1 以降の 9.x バージョン● Oracle 11g、10g
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none">● IBM DB2 9.1 以降の 9.x バージョン● Oracle 10g、11g

1.2.4.2 AIX 6.1

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none">● IBM DB2 9.1 以降の 9.x バージョン● Oracle 10g、11g

注意： サポートされているデータベースエディションの詳細な一覧については、[「サポートされているソフトウェア」\(21 ページ\)](#) を参照してください。

1.2.5 Sun Solaris

LiveCycle ES2 は、Sun Solaris 10 (SPARC® アーキテクチャ) オペレーティングシステムをサポートします。[「AIX、Linux、および Solaris に関するその他の要件」\(18 ページ\)](#) も参照してください。

注意： PDF Generator 3D ES2 は Windows 以外のプラットフォームでサポートされていません。

注意： オペレーティングシステムに X Window ライブラリがインストールされていることを確認してください。この設定は PDF Generator ES2 および Forms ES2 には必須です。詳しくは、オペレーティングシステムのドキュメントを参照してください。

警告： ファイルを抽出するのに Solaris tar コマンドを使用しないでください。このコマンドを使用すると、ファイルが失われるなどのエラーが発生します。[GNU tar ツール](#) をダウンロードし、このツールを使用して、Solaris 環境ですべてのファイルを抽出します。

アプリケーションサーバー	JDK	OS/JVM アーキテクチャ	データベース
Red Hat JBoss Application Server 4.2.0	Sun JDK 5.0 Update 11 (または 5.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● Oracle 10g、11g
Red Hat JBoss Application Server 4.2.1 および JBoss Enterprise Application Platform 4.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● MySQL 5.0.18 または 5.1.30 (推奨) ● Oracle 10g、11g
Oracle WebLogic 10g R3 (Standard Edition および Enterprise Edition)	Sun JDK 6.0 Update 7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● Oracle 10g、11g
9.0.0.2 の新機能 Oracle WebLogic 11g R1 - 10.3.3	Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● Oracle 10g、11g
IBM WebSphere 6.1.0.21 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.5 SR8	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g
9.0.0.2 の新機能 IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition)	WebSphere Java SDK 1.6 SR7	64 ビット OS および 64 ビット JVM	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン ● Oracle 10g、11g

注意： 64 ビットアプリケーションサーバーで PDF Generator ES2 を使用する場合は、追加の 32 ビット JVM をインストールする必要があります。サポートされるソフトウェアについて詳しくは、[「サポートされているソフトウェア」\(21 ページ\)](#) を参照してください。

1.3 必要システム構成

1.3.1 最小ハードウェア要件

次の表に、LiveCycle ES2 でサポートされる最小ハードウェア要件を示します。

オペレーティングシステム	最小ハードウェア要件
Microsoft Windows Server® 2008 R1/R2	Intel® Pentium® 4、2.8 GHz または同等のプロセッサ VMWare ESX 3.0 以降 RAM : 4 GB ディスクの空き容量 : 5.4 GB の一時ディスク容量と LiveCycle ES2 用の 3.4 GB
Microsoft Windows Server® 2003 Enterprise Edition または Standard Edition の SP2 および R2 (32 ビットおよび 64 ビットアーキテクチャ)	Intel® Pentium® 4、2.8 GHz または同等のプロセッサ VMWare ESX 3.0 以降 RAM : 3 GB (32 ビット) または 4 GB (64 ビット) ディスクの空き容量 : 5.4 GB の一時ディスク容量と LiveCycle ES2 用の 3.4 GB Business Activity Monitoring ES2 をデプロイする場合の RAM 要件は、負荷が大きい開発では 8 GB、実稼働環境では 16 GB、評価では 2GB
Sun Solaris 10	UltraSPARC® IIIi 1.5 GHz プロセッサ Solaris コンテナ (ゾーン) パーティション RAM : 3 GB (64 ビット OS および 64 ビット JVM) ディスクの空き容量 : 5.4 GB の一時ディスク容量と LiveCycle ES2 用の 3.4 GB
IBM AIX 5L 5.3	P4 pSeries 615 (Model 6C3) 7029-6C3、1.2 GHz プロセッサ LPAR パーティション RAM : 3 GB (64 ビット OS および 64 ビット JVM) ディスクの空き容量 : 5.4 GB の一時ディスク容量と LiveCycle ES2 用の 3.4 GB
SUSE Linux Enterprise Server 10.0 および 11.0 (64 ビット版のみ)	デュアルコア 1 GHz プロセッサ VMWare ESX 3.0 以降 RAM : 3 GB (64 ビット OS および 64 ビット JVM) ディスクの空き容量 : 5.4 GB の一時ディスク容量と LiveCycle ES2 用の 3.4 GB
Red Hat Enterprise Linux AP または ES 5 (64 ビット版のみ)	デュアルコア 1 GHz プロセッサ VMWare ESX 3.0 以降 RAM : 3 GB (64 ビット OS および 64 ビット JVM) ディスクの空き容量 : 5.4 GB の一時ディスク容量と LiveCycle ES2 用の 3.4 GB

注意 : LiveCycle Business Activity Monitoring ES2 はシステムでサポートされていません。Business Activity Monitoring ES2 は 64 ビットシステムにデプロイしてください。LiveCycle ES2 インストーラで 32 ビットシステムに Business Activity Monitoring ES2 ファイルセットをインストールできても、設定時に失敗します。

1.3.2 Intel x86 の互換性

サポート対象の Windows および UNIX 系の環境において、LiveCycle ES2 では、サポート対象の 32 ビットまたは 64 ビットのいずれかのオペレーティングシステムを実行している Intel および AMD64 互換チップセットがサポートされています。

1.3.3 推奨ハードウェア要件

小規模な実稼働環境の場合：

Intel の環境： Pentium 4、2.8 GHz 以上。デュアルコアプロセッサを使用するとパフォーマンスがさらに向上します。

Sun SPARC の環境： UltraSPARC V 以降。

IBM AIX の環境： Power4 以降。

メモリ要件： 3.2 GB の RAM。

1.3.4 AIX、Linux、および Solaris に関するその他の要件

警告： AIX、Linux、および Solaris オペレーティングシステムでは、Adobe の Web サイトからインストーラをダウンロードするときにバイナリモードを使用します。

1.3.4.1 UTF-8 のインストールおよび設定

AIX、Linux、または Solaris オペレーティングシステムに LiveCycle ES2 をインストールする場合、US English バージョンの UTF-8 ロケールをインストールおよび設定する必要があります (まだインストールされていない場合)。このタスクを実行するには、オペレーティングシステムのインストールメディア (CD または DVD) が必要です。

注意： Linux プラットフォームの場合、このロケールはデフォルトで en_US.utf8 という名前でインストールされます。ロケールは `locale -a` コマンドを使用して確認できます。

▶ AIX に UTF-8 をインストールするには

1. US English UTF-8 ロケールがインストールされていないことを確認するには、コマンドプロンプトに "locale -a" と入力します。コマンドの出力結果に EN_US.UTF-8 というエントリが含まれないことを確認します。
2. コマンドプロンプトのルートで "smitty mle_add_lang" と入力し、AIX SMIT ユーティリティに (テキストモードで) アクセスします。
3. AIX 5.3 および AIX 6.1 では、表示される画面のドロップダウンリスト「CULTURAL CONVENTION」と「LANGUAGE TRANSLATION」の両方で、「UTF-8 US English [EN_US]」を選択します。

注意：「INPUT DEVICE/DIRECTORY」はデフォルトの「/dev/cd0」設定のままにします。

4. **Enter** キーを押して先に進みます。次のようなメッセージが表示されます。

```
installp: Device /dev/cd0 not ready for operation.  
Please insert media and press Enter to continue.
```

5. ディスクドライブに適切な AIX インストールディスクを挿入します。
6. コマンドが完了したら、SMIT ユーティリティを終了し、"locale -a" コマンドを入力してロケールに EN_US.UTF-8 が設定されたことを確認します。

▶ Solaris 10 に UTF-8 をインストールするには

1. US English UTF-8 ロケールがインストールされていないことを確認するには、コマンドプロンプトに "locale -a" と入力します。コマンドの出力結果に EN_US.UTF-8 というエントリが含まれないことを確認します。
2. ディスクドライブに Solaris のインストール CD #1 を挿入し、"/cdrom/sol_10_807_sparc/s0" などの適切な場所にマウントします。
3. root として "localeadm -a nam -d /cdrom/sol_10_807_sparc/s0" コマンドを入力します。

注意： このコマンドを実行すると、en_US.UTF-8 ロケールのみを指定した場合でも、North America (nam) 地域のすべてのロケールがインストールされます。

4. コマンドが完了したら、"locale -a" コマンドを入力してロケールに EN_US.UTF-8 が設定されたことを確認します。

注意： [Solaris ロケールについての FAQ](#) のリンクを参照してください。

1.3.4.2 Solaris および Linux でのファイル制限値の設定

Solaris または Linux の環境での StuckThread 問題を回避するには、/etc/system ファイルで rlim 値を追加するか、大きい値に変更します。

▶ rlim の値を変更する：

1. **Linux** の場合は、/etc/security/limits.conf ファイルを探して開きます。
Solaris の場合は、/etc/system ファイルを探して開きます。
2. **Linux** の場合は、/etc/security/limits.conf ファイルに次の行を追加します。

```
<app_group> soft nofile 8192
```

```
<app_group> hard nofile 8192
```

<app_group> を、アプリケーションサーバーを実行するユーザーグループに置き換えます。<app_group> をアスタリスク (*) に置き換えて、すべてのユーザーおよびユーザーグループを一致と見なすこともできます。

Solaris の場合は、/etc/system ファイル内の rlim 値を探して、次のように変更します。

```
set rlim_fd_cur:プロセスごとのファイル記述子の論理的最大数。この値を 8192 以上に設定します。
```

```
set rlim_fd_max:プロセスごとのファイル記述子の物理的 maximum。この値を 8192 以上に設定します。  
(この変更は、デフォルト値が 8192 未満の場合にのみ必要です)。この値を変更するには、スーパーユーザーの権限が必要です。
```

注意： rlim_fd_max 値は、rlim_fd_cur 値以上にする必要があります。

3. ファイルを保存して閉じます。
4. コンピュータを再起動します。

▶ 更新された設定の確認：

1. 新しいシェルを起動します。
2. ulimit -n と入力して **Enter** キーを押します。
3. 返される値が、設定した rlim の値に一致していることを確認します。

更新後の設定に一致しない値がある場合は、説明どおりに手順を実行したことを確認してコンピュータを再起動します。

1.3.5 LiveCycle Content Services ES2 に関するその他のハードウェア要件

LiveCycle Content Services ES2 をインストールして DB2 データベースと併用する場合は、データベースのコンピュータに少なくとも 2 GB の RAM が必要です。

1.3.6 クライアント側コンピュータの推奨ハードウェア要件

Workbench ES2

- インストール用のディスク容量：
 - 1.5 GB (Workbench ES2 のみの場合)
 - 1.7 GB (Workbench ES2、Designer ES2 およびサンプルアセンブリを 1 つのドライブにフルインストールした場合)
 - 一時インストールディレクトリ用に 400 MB (ユーザーの一時ディレクトリに 200 MB、Windows 一時ディレクトリに 200 MB)

注意：これらの場所がすべて 1 つのドライブ上にある場合は、インストール時に 1.5 GB の空き容量が必要です。一時ディレクトリにコピーされるファイルは、インストールが完了すると削除されます。

- Workbench ES2 を実行するためのメモリ：2 GB の RAM
- ハードウェア要件：Intel® Pentium® 4 または同等の AMD の 1 GHz プロセッサ
- 1024 X 768 ピクセル以上のモニタ解像度、16 ビットカラー以上
- LiveCycle ES2 サーバーに対する TCP/IPV4 または TCP/IPV6 ネットワーク接続

注意：Workbench ES2 を Windows にインストールするには、管理者権限が必要です。管理者以外のアカウントを使用してインストールする場合は、適切なアカウントの資格情報が求められます。

Designer ES2

- Adobe Reader 9.3 以上。3D 機能を含む Designer ES2 のすべての新機能を利用するには、Acrobat 9.3 Pro Extended (推奨) が必要です。
- Adobe Flash Player 9.0 以降。
- (オプション) Flex Builder™ 3.0 以降。
- (オプション) Flex SDK 3.4 (Designer ES2 に付属のフォームガイドコンポーネントをカスタマイズするために必要)。

注意：Designer ES2 インストーラに含まれている Flex SDK 3.4 のみを使用してください。アドビの Web サイトから入手したバージョンの Flex SDK は使用しないでください。

注意：詳しくは、『[Installing Your Development Environment](#)』を参照してください。

エンドユーザーのハードウェア要件：

- LiveCycle Workspace ES2：1 GB の RAM (Adobe Flash と Adobe Reader の要件を含む)
- Adobe Flash Player 9 以降：512 MB の RAM (1 GB 推奨)
- Adobe Reader 8.x 以降：128 MB の RAM (256 MB 推奨)

注意：Web ブラウザ要件については、『[エンドユーザー用ユーザーインターフェイス](#)」(27 ページ)を参照してください。

1.3.7 サポートされているソフトウェア

必要なソフトウェア	サポートされるバージョン
オペレーティングシステム	<ul style="list-style-type: none"> ● Microsoft Windows <ul style="list-style-type: none"> ● Windows Server 2008 Enterprise Edition または Standard Edition SP1 以降 (64 ビット) ● Windows Server 2008 R2 Standard Edition または Enterprise Edition ● Windows Server 2003 Enterprise Edition または Standard Edition SP2 以降 (32 ビットおよび 64 ビット) ● Windows Server 2003 R2 Standard Edition SP2 以降 (32 ビットおよび 64 ビット) ● 32 ビット、64 ビットおよび VMWare ESX/GSX のアーキテクチャで稼働する Windows Server 2003 R2 Enterprise Edition SP2 以降 ● (PDF Generator 3D ES2) Microsoft Windows <ul style="list-style-type: none"> ● Windows Server 2008 Enterprise Edition または Standard Edition SP1 以降 (64 ビット) ● Windows Server 2008 R2 Standard Edition または Enterprise Edition ● Windows Server 2003 Enterprise Edition または Standard Edition SP2 以降 (32 ビットおよび 64 ビット) ● Windows Server 2003 R2 Standard Edition SP2 以降 (32 ビットおよび 64 ビット) ● 32 ビット、64 ビットおよび VMWare ESX/GSX のアーキテクチャで稼働する Windows Server 2003 R2 Enterprise Edition SP2 以降 ● (Workbench ES2) Microsoft Windows <ul style="list-style-type: none"> ● Windows 7 32 ビットおよび 64 ビット (Home Basic、Home Premium、Professional および Enterprise Editions) ● Windows Vista 32 ビットおよび 64 ビット (Business、Home Basic、Home Premium および Ultimate エディション) ● Windows XP Professional SP2 以降 ● Windows Server 2003 Enterprise Edition または Standard Edition SP2 以降 (32 ビットおよび 64 ビット) ● Windows Server 2003 R2 Standard Edition SP2 以降 (32 ビットおよび 64 ビット) ● Windows Server 2008 Enterprise Edition または Standard Edition SP1 以降 ● Windows Server 2008 R2 Standard または Enterprise Edition ● Sun Solaris 10 (64 ビットエディションのみ) ● IBM AIX 5L 5.3 および 6.1 (64 ビットエディションのみ) ● SUSE Linux Enterprise Server 10.0 (64 ビット版のみ) ● SUSE Linux Enterprise Server 11.0 (64 ビット版のみ) ● Red Hat Enterprise Linux AP または ES 5 (64 ビット版のみ)

必要なソフトウェア	サポートされるバージョン
アプリケーションサーバー	<ul style="list-style-type: none"> ● JBoss Application Server 4.2.0 および 4.2.1 ● JBoss Enterprise Application Platform 4.3 ● IBM WebSphere 6.1.0.21 (Base Edition および Network Deployment Edition) ● IBM WebSphere 7.0.0.9 (Base Edition および Network Deployment Edition) ● Oracle WebLogic 10g R3 (Standard エディションおよび Enterprise エディション) ● Oracle WebLogic 11g R3 - 10.3.3
Web ブラウザ	<p>Web ブラウザの一覧については、「Web ブラウザのサポート」(27 ページ) を参照してください。</p>
JDK	<ul style="list-style-type: none"> ● すべてのプラットフォーム上の JBoss 4.2.1 および JBoss EAP 4.3 : Sun Developer Network で入手可能な Sun Java SE Development Kit (JDK) 6.0 Update 14 (または 6.0 の以降の更新)。 ● Windows および Linux 上の WebLogic : Oracle で入手可能な Oracle JRockit Java 6 Update 5 以降、または Oracle JRockit Mission Control 3.0.3。 ● WebLogic (Solaris) : Sun Developer Network から入手可能な Sun Java SE Development Kit (JDK) 6.0 Update 7 以降。北米では夏時間 (TZ) の更新を実行する必要があります (Sun Updater Tool より入手可能)。 ● すべてのプラットフォーム上の WebSphere 6.1 : WebSphere Java SDK 1.5 SR8。 ● すべてのプラットフォーム上の WebSphere 7.0 : WebSphere Java SDK 1.6 SR7。 <p>LiveCycle Rights Management ES2 で AES 256 暗号化を使用するには、Java SE Downloads から Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy ファイルを入手し、インストールする必要があります。</p> <p>注意： これらの要件はオプションであり、Advanced Encryption Standard (AES) 256 を使用する場合にのみ必要です。</p>
32 ビット JDK (64 ビット アプリケーションサーバーにのみ必要)	<ul style="list-style-type: none"> ● Sun Developer Network で入手可能な Sun JDK 6 Update 14 (または 6.0 の以降の更新)。
データベース	<ul style="list-style-type: none"> ● IBM DB2 9.1 以降の 9.x バージョン (Enterprise Edition) ● Microsoft SQL Server 2005 SP3 または 2008 (Standard Edition および Enterprise Edition) ● MySQL 5.0.18 または 5.1.30 (推奨) InnoDB エンジン ● Oracle 10g または 11g (Standard Edition および Enterprise Edition) <p>注意： LiveCycle ES2 はデータベースベンダーの互換性に関する記述に準拠しています (ベンダーの Web サイトを参照)。</p>

必要なソフトウェア	サポートされるバージョン
データベースドライバ	<ul style="list-style-type: none"> ● サポート対象の全バージョンの DB2 について IBM DB2 9.x ドライバが必要： db2jcc.jar (バージョン 3.50.152) ● SQL Server JDBC 1.2 : sqljdbc.jar (Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 の両方に対応) ● MySQL : Connector/J 5.1.6 : mysql-connector-java-5.1.6-bin.jar ● Oracle 10g および 11g : JDK 1.5 では ojdbc5.jar (リリース 11.1.0.6)、JDK 1.6 では ojdbc6.jar (リリース 11.1.0.6) <p>注意： LiveCycle インストールメディアの [DVD_root]/third_party/db/database ディレクトリに、これらのデータベースドライバがあります。</p> <p>注意： LiveCycle Business Activity Monitoring ES2 メタデータデータベースはすべて上記のドライバを使用します。</p>
LDAP サーバー	<ul style="list-style-type: none"> ● Sun ONE 5.1、5.2、6.x ● Microsoft Active Directory 2003 および 2008 ● Novell® eDirectory 8.7.3 ● IBM Tivoli Directory Server 6.0 ● IBM Domino Enterprise Server 8.0 および以降の 8.0 の更新
電子メールサーバー	<ul style="list-style-type: none"> ● Microsoft Exchange 2000、2003、2007 ● Lotus Notes/Domino 6/7 ● SendMail (Red Hat 5 に付属) ● Novell GroupWise 6/7
LiveCycle Data Services ES2	<ul style="list-style-type: none"> ● Adobe Flex® Builder™ 2.0.1 Hot Fix 2 以降 ● Adobe Flash® Builder 4 以降 <p>または</p> <ul style="list-style-type: none"> ● Flex SDK 2.0.1 Hot Fix 2 以降 ● Flex は次のタスクに必要です。 <ul style="list-style-type: none"> ● LiveCycle Data Services ES2 の使用方法 ● LiveCycle Designer ES2 でのフォームガイドのカスタマイズ ● LiveCycle Workspace ES2 のカスタマイズ ● LiveCycle Workspace ES2 用の Flex アプリケーションの作成 ● Flex を使用した LiveCycle ES2 API の呼び出し

必要なソフトウェア	サポートされるバージョン
LiveCycle ES2 コネクタ	<p>Connector for EMC® Documentum® :</p> <ul style="list-style-type: none"> ● EMC Documentum Content Server 6.0 (およびこれ以降のサービスパック) ● EMC Documentum Content Server 6.5 (およびこれ以降のサービスパック) <p>さらに、使用している LiveCycle ES2 サーバーに、使用しているバージョンの Content Server に対応するバージョンの EMC Documentum Foundation Classes (DFC) をインストールします (EMC Documentum から入手可能な『Documentum Foundation Classes Installation Guide』を参照)。</p> <p>Connector for IBM FileNet:</p> <ul style="list-style-type: none"> ● IBM FileNet P8 Content Engine 4.0.x および 4.5 ● IBM FileNet P8 Process Engine 4.0.x および 4.5 <p>注意: Process Engine のバージョンは、ご使用の Content Engine に対応している必要があります (例えば、Content Engine 4.0.x には Process Engine 4.0.x を使用します)。</p> <p>さらに、使用している LiveCycle ES2 サーバーに、使用しているバージョンの Content Engine に対応するバージョンの IBM FileNet P8 Content Java API (4.0.x Content Engine Java API、4.5 Content Engine Java API など) をインストールします (「LiveCycle ES2 Connector for IBM FileNet に関するその他の要件」 (37 ページ) を参照)。必要な JAR ファイルの一覧については、IBM FileNet から入手可能な『Content Java API Developer's Guide』を参照してください。</p> <p>Connector for IBM Content Manager:</p> <ul style="list-style-type: none"> ● IBM Content Manager 8.4 (およびこれ以降の Fix Pack) <p>さらに、使用している LiveCycle ES2 サーバーに、使用しているバージョンの IBM Content Manager に対応するバージョンの IBM ソフトウェアをインストールします。</p> <ul style="list-style-type: none"> ● DB2 Universal Database Client (IBM Content Manager が LiveCycle ES2 サーバーと同じサーバー上にある場合は不要) ● IBM から入手可能な Information Integrator for Content (II4C) <p>注意: Oracle データベースで実行されている IBM Content Manager はサポートされていません。</p> <p>Connector for Microsoft SharePoint :</p> <ul style="list-style-type: none"> ● Microsoft SharePoint Server 2007 <p>また、SharePoint Server を実行するサーバーには Microsoft .NET Framework 3.5 がインストールされている必要があります。</p>
PDF クライアント	<ul style="list-style-type: none"> ● Adobe Acrobat® Professional、Acrobat® Standard、および Acrobat Pro Extended、バージョン 8.0 ~ 9.3 ● Adobe Reader バージョン 8.0 ~ 9.3 <p>注意: ドキュメント保護のポリシーにより、Acrobat Professional、Acrobat Standard、または Acrobat Pro Extended のバージョン 8.0 から 9.3 が必要です。Adobe Reader には、ポリシーで保護されたドキュメントの使用権限がありません。</p> <ul style="list-style-type: none"> ● Apple® QuickTime 7 Player または Pro (埋め込みビデオを PDF マルチメディアに変換するために使用)

注意： LiveCycle Business Activity Monitoring ES2 は 32 ビットシステムではサポートされません。64 ビットシステムにデプロイする必要があります。LiveCycle ES2 インストーラで 32 ビットシステムに Business Activity Monitoring ES2 ファイルセットをインストールできても、設定時に失敗します。

1.3.8 Windows でのインストールユーザーアカウント

Windows にインストールする場合、管理者権限を持っている必要があります。管理者以外のアカウントでインストーラを実行する場合は、管理者権限を持つアカウントの資格情報を入力します。

1.3.9 64 ビット Windows インストール用の設定

64 ビット Windows Server 2008、Windows Vista または Windows 7 オペレーティングシステムでは、次のように管理者承認モードのセキュリティオプションを変更します。

1. スタート/コントロールパネル/管理ツール/ローカルセキュリティポリシー/ローカルポリシー/セキュリティオプションを選択します。
2. 「ユーザーアカウント制御：管理者承認モードでの管理者に対する昇格時のプロンプトの動作」を探し、「確認を要求しないで昇格する」に設定します。
3. コンピュータを再起動します。

警告： PDF Generator ES2 または PDF Generator 3D ES2 が正しく動作するには、Windows UAC を無効なままにする必要があります。インストールおよび設定のプロセスでは、「確認を要求しないで昇格する」オプションを有効にして UAC をオンにしても構いませんが、PDF Generator ES2 または PDF Generator 3D ES2 を実行するためには UAC を完全に無効にする必要があります。

評価用のシステムでインストールと設定を行う場合は、実稼働環境用のコンピュータにデプロイした後や、PDF Generator ES2 または PDF Generator 3D ES2 をアンインストールした後で、コンピュータ上で UAC を有効にできます。

▶ Vista 上での Windows UAC の無効化：

1. システム構成ユーティリティにアクセスするには、スタート/ファイル名を指定して実行を選択し、「名前」ボックスに `MSCONFIG` と入力します。
2. 「ツール」タブをクリックし、スクロールして「UAC の無効化」を選択します。
3. 「起動」をクリックして新しいウィンドウでコマンドを実行します。
4. 完了したら、コマンドウィンドウを閉じ、システム構成ウィンドウを閉じます。
5. コンピュータを再起動します。

UAC を再度有効にするには、上記の手順を繰り返し、「UAC の有効化」を選択してから「起動」をクリックします。

▶ Windows Server 2008 または Windows 7 での Windows UAC の無効化：

1. スタート/コントロールパネル/システムとセキュリティを選択し、「アクションセンター」にある「ユーザーアカウント制御設定の変更」を選択します。
2. 「コンピューターに対する変更の通知を受け取るタイミングの選択」画面で、スライダを「通知しない」まで動かします。
3. 「OK」をクリックします。

UAC を再度有効にするには、上記の手順を繰り返し、「ユーザーアカウント制御 (UAC) を使ってコンピュータの保護に役立たせる」オプションを選択してコンピュータを再起動します。

1.3.10 データベースを使用するための最小限のユーザー権限

データベース	初期化権限	ランタイム権限
Oracle	CREATE SESSION CREATE CLUSTER CREATE TABLE CREATE VIEW CREATE SEQUENCE UNLIMITED TABLE SPACE	CREATE SESSION UNLIMITED TABLE SPACE (ユーザーのクォータを設定しない場合にのみ必要) CREATE TABLE
MySQL	SELECT INSERT UPDATE DELETE CREATE DROP REFERENCES INDEX ALTER CREATE_TEMP_TABLE LOCK_TABLES	SELECT INSERT UPDATE DELETE
SQL Server - DB レベル	Create Table Create View Connect	Connect
SQL Server - スキーマレベル	Alter Insert References Select Update Delete	Insert Select Update Delete
DB2	詳しくは、 「DB2 ユーザーアカウント」(60 ページ) を参照してください。	詳しくは、 「DB2 ユーザーアカウント」(60 ページ) を参照してください。

1.3.11 Web ブラウザのサポート

1.3.11.1 エンドユーザー用ユーザーインターフェイス

エンドユーザーのコンポーネントには、次のモジュールが含まれます。

- LiveCycle Workspace ES2 (Flash Player が必要)
- **注意:** Workspace ES2 には、また Workspace ES2 のフォームガイドを使用するには、Adobe Flash Player 9.0.115.0 以降が必要です。
- LiveCycle Reader Extensions ES2 (Flash Player が必要)
- LiveCycle Rights Management ES2 (Flash Player が必要)
- LiveCycle PDF Generator ES2 および LiveCycle PDF Generator 3D ES2 (ブラウザのみ)
- LiveCycle Content Services ES2 (ブラウザのみ)

オペレーティングシステム	Flash Player	対応ブラウザ
Microsoft Windows 7	Flash Player 10	Microsoft Internet Explorer 8 Firefox 3.0 以降 ⁽¹⁾
Microsoft Windows Vista™	Flash Player 9 または 10	Microsoft Internet Explorer 7 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Windows 2000	Flash Player 9 または 10	Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Windows XP	Flash Player 9 または 10	Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Windows Server 2008	Flash Player 10	Internet Explorer 8 Firefox 3.0 以降 ⁽¹⁾
Windows Server 2003	Flash Player 9 または 10	Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Mac OS X v 10.4.x または 10.5.x (PowerPC)	Flash Player 9 または 10	Firefox 3.0 以降 (Workspace ES2 ではサポートされません) ^{(1) (2)} Safari 3.x または 4.x (Workspace ES2 および Content Services ES2 にはバージョン 3.0.3 以降が必要)
Mac OS X v 10.4.x、10.5.x または 10.6.x (Intel)	Flash Player 9 または 10	Firefox 3.0 以降 (Workspace ES2 ではサポートされません) ^{(1) (2)} Safari 3.x または 4.x (Workspace ES2 および Content Services ES2 にはバージョン 3.0.3 以降が必要)

⁽¹⁾ 「以降」のバージョンには、メジャーリビジョンが含まれます。例えば、Internet Explorer 6 以降には Internet Explorer 7 と 8 も含まれます。

⁽²⁾ Workspace ES2 は、Windows では Internet Explorer および Firefox がサポートされていますが、Mac では Safari 3.0.3 以降のみがサポートされます。

● LiveCycle Forms ES2

オペレーティングシステム	Flash Player	対応ブラウザ
Microsoft Windows 7	該当なし	Microsoft Internet Explorer 8 Firefox 3.0 以降 ⁽¹⁾ Netscape 8.x 以降
Microsoft Windows Vista™	該当なし	Microsoft Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾ Netscape 8.x 以降
Windows XP	該当なし	Microsoft Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾ Netscape 8.x 以降
Windows Server 2008	該当なし	Firefox 3.0 以降 ⁽¹⁾
Windows Server 2003	該当なし	Firefox 3.0 以降 ⁽¹⁾
Linux (Intel)	該当なし	Firefox 3.0 以降 ⁽¹⁾ Netscape 8.x 以降
Mac OS X v 10.4.x または 10.5.x (PowerPC)	該当なし	Safari 3.x または 4.x
Mac OS X v 10.4.x、10.5.x または 10.6.x (Intel)	該当なし	Safari 3.x または 4.x

(1) 「以降」のバージョンには、メジャーリビジョンが含まれます。例えば、Internet Explorer 6 以降には Internet Explorer 7 と 8 も含まれます。

● LiveCycle Business Activity Monitoring ES2

オペレーティングシステム	Flash Player	対応ブラウザ
Microsoft Windows Vista™	Flash Player 9 または 10	Microsoft Internet Explorer 7 以降 ⁽¹⁾
Windows XP	Flash Player 9 または 10	Microsoft Internet Explorer 6 以降 ⁽¹⁾
Windows Server 2000	Flash Player 9 または 10	Microsoft Internet Explorer 6 以降 ⁽¹⁾

(1) 「以降」のバージョンには、メジャーリビジョンが含まれます。例えば、Internet Explorer 6 以降には Internet Explorer 7 と 8 も含まれます。

1.3.11.2 管理者用ユーザーインターフェイス

オペレーティングシステム	Flash Player	対応ブラウザ
Microsoft Windows 7	該当なし	Internet Explorer 8 Firefox 3.0 以降 ⁽¹⁾
Microsoft Windows Vista	該当なし	Internet Explorer 7 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Windows 2000	該当なし	Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Windows XP	該当なし	Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Windows Server 2008	該当なし	Internet Explorer 8 Firefox 3.0 以降 ⁽¹⁾
Windows Server 2003	該当なし	Internet Explorer 6 以降 ⁽¹⁾ Firefox 3.0 以降 ⁽¹⁾
Linux (Intel)	該当なし	Firefox 3.0 以降 ⁽¹⁾

⁽¹⁾ 「以降」のバージョンには、メジャーリビジョンが含まれます。例えば、Microsoft Internet Explorer 6 以降には Microsoft Internet Explorer 7 および 8 も含まれます。

1.3.11.3 Firefox での Content Services ES2 の使用

Firefox が `file:///` で始まるローカルリンクを処理する方法についてはいくつかの制限があります。Firefox で Content Services ES2 を使用するには、<http://forge.alfresco.com/projects/firefox-ext/> からプラグインをダウンロードし、インストールして Firefox のインストールを更新する必要があります。

1.3.12 PDF Generator ES2 および PDF Generator 3D ES2 に関するその他の要件

注意： PDF Generator 3D ES2 は Windows 環境でのみサポートされています。

注意： オペレーティングシステムに X Window ライブラリがインストールされていることを確認してください。この設定は PDF Generator ES2 および Forms ES2 には必須です。詳しくは、オペレーティングシステムのドキュメントを参照してください。

注意： PDF Generator ES2 がデプロイされた Windows 2008 マシン上の SendToPrinter API で、共有プリンタのプロトコルを使用することはできません。CIFS または DirectIP などの代替プロトコルを使用してください。

1.3.12.1 Windows のユーザーアカウント

次のタスクには管理者権限があるユーザーアカウントを使用する必要があります。

- Microsoft Office のインストール
- PDF Generator ES2 または PDF Generator 3D ES2 のインストール
- PDF Generator ES2 または PDF Generator 3D ES2 対応の Acrobat 9.3 Professional Extended のインストール
- アプリケーションサーバープロセスの実行

1.3.12.2 PDF Generator での 64 ビットアプリケーションサーバーの使用

PDF Generator ES2 または PDF Generator 3D ES2 を Windows システムまたは UNIX 系システム上の 64 ビットアプリケーションサーバーで使用する場合は、アプリケーションサーバーが使用する 64 ビット Java 6 JDK の他に、32 ビット Java 6 JDK がインストールされている必要があります。環境変数 `JAVA_HOME_32` を設定します。この変数は、64 ビットアプリケーションサーバーが使用している Windows および UNIX 系システム上の 32 ビット JDK を示す必要があります。指定するパスは、指定したインストールディレクトリと、インストール先のオペレーティングシステムによって変わります。

注意： 32 ビット Sun JDK をインストールし、そのインストールディレクトリを指定するように `JAVA_HOME_32` を設定します。「[Sun Java 6 Release Notes > Supported System Configurations](#)」を参照し、使用しているオペレーティングシステム (AIX 以外) 用の 32 ビットバージョンをダウンロードしてください。

警告： `JAVA_HOME_32` は環境変数としてのみ設定し、`PATH` には含めないでください。`JAVA_HOME_32` を `PATH` に含めると、EAR のデプロイ時、またはサーバーの再起動時に Java core ダンプが表示される場合があります。

▶ Windows の `JAVA_HOME_32` 変数を設定するには

1. スタート/コントロールパネル/システムの順に選択します。
2. 「詳細設定」タブをクリックします。
3. 「環境変数」をクリックし、「システム環境変数」で「新規」をクリックします。
4. 環境変数 `JAVA_HOME_32` を入力します。このディレクトリは、JDK を含むディレクトリです。例えば、次のコードを入力します。

```
D:¥Program Files (x86)¥Java¥jdk1.6.0_14
```

▶ Linux または Solaris で `JAVA_HOME_32` 変数を設定するには

次の例に示すように、Borne シェルおよび Bash シェルについて、サポート対象の JDK の `JAVA_HOME_32` 変数を設定します。

```
JAVA_HOME_32=/opt/jdk1.6.0_14
export JAVA_HOME_32
```

1.3.12.3 ネイティブファイルを変換するためのソフトウェアのインストール

PDF Generator ES2 または PDF Generator 3D ES2 をインストールする前に、PDF 変換サポートを必要とするファイル形式に対応したソフトウェアをインストールし、アプリケーションサーバープロセスの実行に使用されているユーザーアカウントを使用して、ソフトウェアのライセンスを手動でアクティベートします。

PDF Generator ES2 または PDF Generator 3D ES2 をサポートする各ネイティブアプリケーションの LiveCycle ES2 サーバー 1 ライセンスをアクティベートする必要があります。LiveCycle ES2 デプロイメントで使用するネイティブアプリケーションごとにライセンス契約を参照し、LiveCycle ES2 デプロイメントで指定されたライセンス要件を満たしていることを確認してください。通常、ネイティブアプリケーションサポートを使用する各 LiveCycle ES2 ユーザーは、そのネイティブアプリケーションを使用するコンピュータでライセンスをアクティベートする必要があります。

PDF Generator ES2 または PDF Generator 3D ES2 の機能を拡張すると、次のアプリケーションを使用して PDF に変換できるファイルの種類を追加できます。

- Microsoft Office 2003、2007 (DOC、XLS、PPT、RTF、TXT、Microsoft Office open XML 形式)
- Microsoft Office Visio 2003、2007 (VSD)
- Microsoft Publisher 2003、2007 (PUB)
- Microsoft Project 2003、2007 (MPP)
- AutoCAD 2005、2006、2007、2008 (DWG、DXF、DWF)

注意： AutoCAD を使用した DWG、DXF および DWF ファイルのネイティブファイル変換は、32 ビット環境でのみサポートされます。Acrobat を使用した DWG、DXF および DWF ファイルの変換は、32 および 64 ビットプラットフォームの両方でサポートされます。

- Corel WordPerfect 12、X4 (WPD)
- Adobe Photoshop® CS2
- Adobe FrameMaker® 7.2、8.0 (FM)
- Adobe PageMaker® 7.0 (PMD、PM6、P65、PM)
- OpenOffice 2.4.2、3.1 (ODT、ODP、ODS、ODG、ODF、SXW、SXI、SXC、SXD、SXM)

注意： OpenOffice 3.1 で作成したドキュメントを変換するには、OpenOffice 3.1 以降をサーバーにインストールする必要があります。OpenOffice 2.4.2 では、それより新しいバージョンの OpenOffice で作成したドキュメントを変換できません。

次のネイティブファイルの形式を変換するために、ネイティブソフトウェアアプリケーションをインストールする必要はありません。

- Print ファイル (PS、PRN、EPS)
- Web ファイル (HTML)
- 画像ファイル (JPEG、GIF、BMP、TIFF、PNG)

1.3.12.4 Acrobat

PDF Generator ES2 または PDF Generator 3D ES2 を使用するには、Acrobat Pro Extended をインストールする必要があります。Acrobat は、LiveCycle ES2 インストーラを実行する前にインストールする必要があります。

LiveCycle ES2 インストーラを実行すると、Acrobat_PATH 環境変数が自動的に設定されます。手動で設定する必要がある場合、使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイガイド](#) の「環境変数の設定」を参照してください。アプリケーションサーバーを再起動する必要があります。

注意： 環境変数 Acrobat_PATH では、大文字と小文字が区別されます。

LiveCycle Encryption サービスで AES 256 暗号化を使用して PDF ドキュメントを暗号化するには、Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy ファイルを入手し、インストールする必要があります。例えば Sun JDK の場合は、[Java SE Downloads](#) から JCE ファイルをダウンロードします。

ポリシーファイルをダウンロードしたら、[JAVA_HOME]/jre/lib/security フォルダにある既存の local_policy.jar ファイルおよび US_export_policy.jar ファイルを、ダウンロードした JAR ファイルに置き換えます。

注意： LiveCycle 自動インストールを使用している 64 ビットの Windows サーバーでは、
[LiveCycleES2 root]¥Java¥jdk1.6.0_14¥jre¥lib¥security に、ダウンロードした JAR ファイルをコピーします。

注意： 使用する JDK に応じて、Sun または IBM の Web サイトから必要な JAR ファイルをダウンロードします。

1.3.12.5 QuickTime 7

PDF Generator ES2 では、PowerPoint プレゼンテーションや PDF マルチメディアファイルなどのファイルに埋め込まれているビデオを変換する場合は、QuickTime 7 (Player または Pro) がインストールされている必要があります。このアプリケーションは、[Apple Downloads](#) サイトから入手できます。

1.3.12.6 Windows 環境変数の設定

FrameMaker、Photoshop、PageMaker、WordPerfect などのアプリケーションから PDF ドキュメントを作成する場合は、Windows の環境変数を設定する必要があります。

これらの環境変数の名前を以下に示します。

- FrameMaker_PATH
- Notepad_PATH (この変数は空のままにしてください)
- OpenOffice_PATH
- PageMaker_PATH
- Photoshop_PATH
- WordPerfect_PATH
- Acrobat_PATH

これらの環境変数はオプションであり、PDF Generator ES2 または PDF Generator 3D ES2 で対応するアプリケーションを使用して PDF ファイルを変換する場合にのみ設定する必要があります。環境変数の値には、対応するアプリケーションを起動する際に使用する実行ファイルの絶対パスを含める必要があります。

例えば、FrameMaker_PATH 変数には C:¥Program Files¥Adobe¥FrameMaker7.2¥FrameMaker.exe を含めることができます。ただし、OpenOffice_PATH は他の環境変数とは設定方法が異なります。この変数は、(実行ファイルへのパスではなく、) OpenOffice のインストールフォルダに設定する必要があります。Windows での OpenOffice_PATH の一般的な値は、C:¥Program Files¥OpenOffice.org 3.0¥ です。

Word、PowerPoint、Excel、Visio および Project などの Microsoft Office アプリケーションのパスや AutoCAD のパスは不要です。これらのアプリケーションがサーバーにインストールされている場合は、Generate PDF サービスが自動的にこれらのアプリケーションを起動します。

注意： Acrobat_PATH 環境変数は大文字と小文字が区別されます。

▶ Windows 環境変数を設定するには

1. スタート/コントロールパネル/システムの順に選択します。
2. 「詳細設定」タブをクリックします。
3. 「環境変数」をクリックし、「システム環境変数」で「新規」をクリックします。

4. 設定が必要な環境変数の名前 (FrameMaker_PATH など) を入力します。このディレクトリは、実行ファイルを含むディレクトリです。例えば、次のコードを入力します。

```
C:¥Program Files¥Adobe¥FrameMaker7.2¥FrameMaker.exe
```

1.3.12.7 Network Printer Client のインストール

PDF Generator ES2 には、PDF Generator ES2 インターネットプリンタのインストールに使用する Network Printer Client インストーラが含まれます。インストールの完了後は、クライアントコンピュータの既存のプリンタのリストに PDF Generator ES2 プリンタが追加されます。その後、このプリンタを使用してドキュメントを送信し、PDF に変換することができます。Network Printer Client のインストールについて詳しくは、使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイ](#) ドキュメントを参照してください。

注意： PDF Generator ES2 Network Printer クライアントは、次の 32 ビットの Windows プラットフォームでのみサポートされます。Windows XP、Windows 2000、Windows Server 2003、Windows Vista。

1.3.12.8 Service Control Manager コマンドラインツール

Windows で PDF Generator ES2 の自動インストールを行う場合は、インストール前に sc.exe (Service Control Manager コマンドラインツール) が Windows 環境にインストールされていることを確認します。一部の Windows サーバーでは、sc.exe がプレインストールされていません。デフォルトでは、sc.exe ファイルは C:¥Windows¥system32 ディレクトリにインストールされます。ほとんどの OS のインストールでは、このツールがインストールされます。このツールがインストールされていない場合は、使用しているバージョンの Windows 用の Windows リソースキットでこのツールを手に入れます。サーバーにツールがインストールされていることを確認するには、コマンドプロンプトに sc.exe と入力します。ツールの使用状況が返されます。

1.3.12.9 ヘッドレスモードの設定

ヘッドレスモード環境 (モニター、キーボードまたはマウスのないサーバー) で PDF Generator ES2 を実行している場合、x11 ライブラリをインストールする必要があります。一部の Linux では、これらのライブラリがデフォルトでインストールされません。そのため、ライブラリを取得して手動でインストールする必要があります。詳しくは、ご使用のオペレーティングシステムのヘルプシステムを参照してください。

1.3.12.10 マルチスレッドファイル変換の有効化

デフォルトでは、PDF Generator ES2 は、一度に 1 つの OpenOffice、Microsoft Word または PowerPoint ドキュメントのみを変換できます。マルチスレッド変換を有効にすると、OpenOffice または PDFMaker の複数のインスタンスを起動して PDF Generator ES2 で同時に複数のドキュメントを変換できます (PDFMaker は、Word 文書と PowerPoint ドキュメントの変換に使用されます)。

注意： Microsoft Word 2003 および PowerPoint 2003 でのマルチスレッドファイル変換はサポートされていません。また、Microsoft Excel (2003 および 2007 の両バージョン) もサポートされていません。マルチスレッドファイル変換を有効にするには、Microsoft Word 2007 および PowerPoint 2007 にアップグレードしてください。

OpenOffice または PDFMaker の各インスタンスは、それぞれ別のユーザーアカウントを使用して起動されます。追加する各ユーザーアカウントは、LiveCycle ES2 サーバーコンピュータ上での管理者権限を持つ有効なユーザーである必要があります。

Windows プラットフォームでは、プロセスレベルトークンを置き換えるための権限を追加する必要があります ([「プロセスレベルトークンの置き換え」権限の付与 \(Windows のみ\)](#) (35 ページ) を参照)。

64 ビットの Windows では、Windows UAC を無効にする必要があります ([「64 ビット Windows インストール用の設定」\(25 ページ\)](#) を参照)。

すべてのプラットフォームで、ユーザー権限も設定する必要があります ([「PDF Generator ES2 のマルチユーザーサポート」\(34 ページ\)](#) を参照)。Windows 2003 または 2008 上で OpenOffice、Microsoft Word または Microsoft PowerPoint のユーザーを追加する場合や、Linux または Solaris 上で OpenOffice のユーザーを追加する場合は、すべてのユーザーに関して、最初に表示されるアクティベート用のダイアログを閉じます。

LiveCycle ES2 サーバーの設定が完了したら、LiveCycle 管理コンソールで LiveCycle ES2 ユーザーアカウントを追加して、AIX、Linux、または Solaris プラットフォームでパスワードの入力要求をオフにします。使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイガイド](#) の「マルチスレッドファイル変換のユーザーアカウントの設定」を参照してください。

Linux または Solaris での OpenOffice に必要な追加設定

1. /etc/sudoers ファイルで、追加のユーザー (LiveCycle ES2 サーバーを実行する管理者以外) のエントリを追加します。例えば、ユーザー lcadm、サーバー myhost として LiveCycle ES2 を実行している場合、user1 および user2 として動作させるには、/etc/sudoers に次のエントリを追加します。

```
lcadm myhost=(user1) NOPASSWD: ALL
```

```
lcadm myhost=(user2) NOPASSWD: ALL
```

この設定により、lcadm は、ホスト myhost において user1 または user2 として、パスワードの入力を求められることなくすべてのコマンドを実行できるようになります。

2. ユーザーアカウントの追加で追加したすべてのユーザーが LiveCycle ES2 サーバーに接続できるようにします。例えば、user1 というローカルユーザーに LiveCycle ES2 サーバーに接続する権限を許可するには、次のコマンドを使用します。

```
xhost +local:user1@
```

詳しくは、xhost コマンドのドキュメントを参照してください。

3. サーバーを再起動します。

1.3.12.11 PDF Generator ES2 のマルチユーザーサポート

Windows 環境でネイティブファイルおよび OpenOffice ファイルのマルチユーザーサポートを有効にするには、次の権限を持つユーザーを 3 人以上追加する必要があります。AIX、Linux、または Solaris プラットフォームでは、ユーザーを 1 人以上作成する必要があります。

プラットフォーム	ユーザー権限
Windows Server 2008	管理者権限を持ち、UAC を無効にしたユーザー
Windows Server 2003	管理者権限を持つユーザー
AIX、Linux または Solaris	sudo 権限を持つユーザー

PDF Generator ES2 ネイティブ変換用のユーザーを追加する場合は、アプリケーションサーバーを実行するユーザーに「プロセスレベルトークンの置き換え」権限を付与する必要があります。[「「プロセスレベルトークンの置き換え」権限の付与 \(Windows のみ\)」\(35 ページ\)](#) を参照してください。

1.3.12.12 「プロセスレベルトークンの置き換え」権限の付与 (Windows のみ)

Windows サービスとしてではなくコマンドプロンプトからアプリケーションサーバーを起動する際に使用するユーザーアカウントには、「プロセスレベルトークンの置き換え」権限が必要です。この設定は PDF Generator ES2 には必須です。

▶ 「プロセスレベルトークンの置き換え」権限を付与するには

1. スタート／ファイル名を指定して実行をクリックして、gpedit.msc と入力します。
2. グループポリシーダイアログボックスで、コンピュータの構成／Windows の設定／セキュリティの設定／ローカルポリシー／ユーザー権利の割り当てを選択して、「プロセスレベルトークンの置き換え」をダブルクリックします。
3. 「ユーザーまたはグループの追加」をクリックし、アプリケーションサーバーを起動するコマンドプロンプトを開くための Windows ユーザーアカウントを追加します。
4. Windows を再起動して、アプリケーションサーバーを起動します。

1.3.13 Connector for IBM Content Manager に関するその他の要件

LiveCycle ES2 Connector for IBM Content Manager では、次のソフトウェアがインストールされている必要があります (両方とも IBM の Web サイトから入手可能) :

- DB2 Universal Database Client
- IBM Information Integrator for Content (II4C)

使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイ](#) ドキュメントの「デプロイメント完了後の作業」の章を参照してください。

▶ 単一の IBM Content Manager データストアに対する接続を設定するには

1. DB2 Configuration Assistant を起動します。
2. Selected / Add Database Using Wizard をクリックします。
3. 「Manually Configure a Connection to a Database」を選択し、「Next」をクリックします。
4. 「TCP/IP」を選択して、「Next」をクリックします。
5. 以下の TCP/IP 通信オプションを指定して、「Next」をクリックします。
 - 「Host Name」ボックスに、DB2 Content Manager をホストするサーバーのホスト名を入力します。
 - 「Service Name」ボックスは空にしておきます。
 - 「Port Number」ボックスに、ポート番号を入力します。DB2 Content Manager のデフォルトのポート番号は 50000 です。
6. 「Database Name」ボックスに IBM Content Manager データストア名を入力し、「Database Alias」ボックスにデータストアのエイリアス名を入力して、「Next」をクリックします。
7. 「Next」をクリックして、デフォルトのデータソース設定を受け入れます。
8. 「Operating System」リストで、使用しているオペレーティングシステムを選択し、「Next」をクリックします。

9. 以下のシステムオプションを指定して、「Next」をクリックします。
 - 「System Name」ボックスに、DB2 をホストするサーバー名を入力します。「Discover」をクリックすると、DB2 Content Manager では指定したシステム名を検索し、システムが見つからない場合、すべての DB2 インスタンスを示します。
 - 「Host Name」ボックスにホスト名を入力するか、または「View Details」をクリックして、前の手順で指定したシステムのドメインと IP アドレスを表示します。
 - 「Operating System」リストで、DB2 Content Manager をデプロイしたオペレーティングシステム (Windows、Linux または AIX) を選択します。
10. (オプション) 「Security」オプションを指定するには、「Use Authentication Value in Server's DBM Configuration」を選択して、「Finish」をクリックします。
11. Test Connection ダイアログボックスで、必要に応じて接続をテストします。

▶ 複数の IBM Content Manager データストアに対する接続を設定するには

1. [「単一の IBM Content Manager データストアに対する接続を設定するには」 \(35 ページ\)](#) に記載されている手順に従って初期接続を設定します。
2. cmbicmsrvs.ini ファイル (データストア情報を格納するファイル) を以下のように変更して、データベース接続を追加します。
 - コマンドプロンプトウィンドウで、ディレクトリを [I14C home]/bin (例えば Windows では C:\Program Files\ibm\db2cmv8\、AIX、Linux、または Solaris では /opt/IBM/db2cmv8) に変更します。
 - cmbenv81.bat (Windows) または cmbenv81.sh (AIX、Linux、または Solaris) ファイルを実行して、I14C の Java ユーティリティ用の環境およびクラスパスを設定します。
 - ディレクトリを [I14C working directory]/cmgmt/connectors に変更します。ここで、[I14C working directory] は以下のいずれかです。
 - (Windows) C:\Program Files\ibm\db2cmv8
 - (Linux、AIX) /home/ibmcmadm
 - (Solaris) /export/home/ibmcmadm
 - `java com.ibm.mm.sdk.util.cmbicmsrvsicm -a add -s <library server database name> -sm <database schema name>` コマンドを実行します。<library server database name> は、上述の手順 6 で設定した Database Alias と同じです。

注意： 次の手順では、DB2 の権限を持たないユーザーが cmbicmenv.ini ファイルを使用して接続証明書を共有することができません。

▶ IBM Content Manager データストアにマルチユーザー接続を設定するには

1. コマンドプロンプトウィンドウで、ディレクトリを [I14C home]/bin (例えば Windows では C:\Program Files\ibm\db2cmv8\、AIX、Linux、または Solaris では /opt/IBM/db2cmv8) に変更します。
2. cmbenv81.bat (Windows) または cmbenv81.sh (AIX、Linux、または Solaris) ファイルを実行して、I14C の Java ユーティリティ用の環境およびクラスパスを設定します。
3. ディレクトリを [I14C working directory]/cmgmt/connectors に変更します。ここで、[I14C working directory] は以下のいずれかです。
 - (Windows) C:\Program Files\ibm\db2cmv8
 - (Linux、AIX) /home/ibmcmadm
 - (Solaris) /export/home/ibmcmadm

4. `java com.ibm.mm.sdk.util.cmbenvicm -a add -s <library server database name> -u <database user ID> -p <database password>` コマンドを実行します。<library server database name> は、上述の手順 6 で設定した Database alias と同じです。

1.3.14 LiveCycle ES2 Connector for IBM FileNet に関するその他の要件

これらの要件はオプションであり、Connector for IBM FileNet をインストールする場合のみ必要です。

IBM FileNet 4.0

LiveCycle ES2 を IBM FileNet 4.0 Content Engine に接続する場合は、Content Engine Java Client をインストールする必要があります。デフォルトで C:\Program Files\FileNet\Content Engine に配置される IBM FileNet 4.0 Content Server インストーラを使用します。コンポーネント選択画面では Java クライアントコンポーネントのみを選択します。

IBM FileNet 4.0 Process Engine を設定するには、pe.jar ファイルを、Process Engine ディレクトリから LiveCycle ES2 をホストするコンピュータにコピーします。C:\FileNetPE\files ディレクトリを作成して、ここに pe.jar ファイルをコピーします。Process Engine クライアントインストールディレクトリは、C:\FileNetPE に変わりました。

IBM FileNet 4.5

LiveCycle ES2 を IBM FileNet 4.5 Content Engine に接続する場合は、デフォルトで C:\Program Files\FileNet に配置される Content Engine クライアントをインストールする必要があります。インストール時に、コンポーネント選択画面で、Application Engine または Process Engine から 1 つ以上のコンポーネントを選択します。

IBM FileNet 4.5 Process Engine の場合は、デフォルトで C:\Program Files\FileNet\BPMClient に配置される IBM FileNet 4.5 Process Engine Client をインストールする必要があります。インストール時に、コンポーネント選択画面で「Other」オプションを選択します。

1.4 LDAP の設定

この設定はオプションであり、ユーザーを認証するために LDAP ディレクトリを使用する場合にのみ必要です。LiveCycle Policy Server 7.x または Rights Management ES をアップグレードすると、LDAP の環境設定は自動的に移行されます。

既存の LDAP サーバーおよびデータベースがない場合、ベンダーのマニュアルに従って LDAP サーバーおよびデータベースをインストールし、設定してください。サポートされている LDAP サーバーの一覧については、[「サポートされているソフトウェア」\(21 ページ\)](#) を参照してください。LiveCycle ES2 設定プロセス中に使用する LDAP の管理者名とパスワードを控えておいてください。LiveCycle ES2 サービスをインストールおよびデプロイした後で、LiveCycle ES2 を設定して LDAP データベースに接続します。この設定には User Manager サービスを使用します。使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイ](#) ドキュメントを参照してください。

1.5 グローバルドキュメントストレージディレクトリ

グローバルドキュメントストレージディレクトリ (GDS) は、プロセス内で使用される長期間有効なファイルや重要な LiveCycle ES2 製品コンポーネントを格納するために使用します。長期間有効なファイルの有効期間内には、LiveCycle ES2 システムが複数回再起動され、期間は数日間、数年間に渡る場合もあります。該当するファイルには、PDF ファイル、ポリシー、フォームテンプレートなどがあります。長期間有効なファイルは、多数の LiveCycle ES2 デプロイメントの全体的な状態の中で重要な部分です。長期間有効なドキュメントが一部でも失われたり破損したりすると、LiveCycle ES2 サーバーが不安定な状態になるおそれがあります。非同期ジョブの呼び出しの入力ドキュメントも GDS ディレクトリに保存されます。これらのドキュメントは、要求を処理するために使用可能である必要があります。

LiveCycle ES2 データベースへのドキュメントの保存を使用することもできます。詳しくは、[LiveCycle ES2 管理ヘルプ](#)を参照してください。

1.5.1 グローバルドキュメントストレージディレクトリの計画と作成

GDS ディレクトリの場所、サイズ、およびセキュリティを、事前に計画する必要があります。次の節を参照してください。

- [「グローバルドキュメントストレージディレクトリの場所」 \(38 ページ\)](#)
- [「グローバルドキュメントストレージディレクトリのサイズ決定要因」 \(39 ページ\)](#)
- [「グローバルドキュメントストレージディレクトリの保護」 \(39 ページ\)](#)
- [「グローバルドキュメントストレージディレクトリのバックアップ」 \(39 ページ\)](#)

GDS ディレクトリは、LiveCycle ES2 データベースを初期化する前に作成する必要があります。

1.5.2 グローバルドキュメントストレージディレクトリの場所

GDS ディレクトリの場所は、LiveCycle ES2 のインストール後に、LiveCycle Configuration Manager で設定できます。指定する GDS ディレクトリは可用性が高く、パフォーマンスを向上させるために短時間でアクセスできなければなりません。GDS ディレクトリが共有ネットワークドライブ上にある場合は、場所を `¥¥computer_name¥GDS` として指定することをお勧めします。

インストール時に場所を指定しないと、アプリケーションサーバーのインストールディレクトリの下にあるディレクトリがデフォルトの場所になります。

- (JBoss) [appserver root]/server/lc_<dbname>/svcnative/DocumentStorage
- (WebLogic) [appserverdomain]/adobe/<server>/DocumentStorage
- (WebSphere) [appserver root]/installedApps/adobe/<server>/DocumentStorage または [appserver root]/profiles/<profileName>/installedApps/adobe/<server>/DocumentStorage

インストールの完了後に GDS ディレクトリの場所を変更する場合 ([LiveCycle ES2 管理ヘルプ](#)を参照)、GDS ディレクトリ用の適切な場所を計画する必要があります。

警告： GDS ディレクトリがドライブのルートにある場合 (D:¥ など) は、Windows ではモジュールのデプロイメントが失敗します。GDS の場合、ディレクトリがドライブのルートではなく、サブディレクトリに配置されていることを確認する必要があります。例えば、ディレクトリは単に D:¥ではなく D:¥GDS にする必要があります。

1.5.3 グローバルドキュメントストレージディレクトリのサイズ決定要因

共有ディレクトリのサイズは、デプロイメントに予測される LiveCycle ES2 の使用状況によって決まります。GDS ディレクトリには少なくとも 10 GB のディスク容量を割り当てる必要があります。ただし、次の要因もこのサイズの決定に影響します。

- LiveCycle ES2 で処理されるドキュメントの標準的な分量。大量のドキュメントを処理するには、大容量の GDS ディレクトリが必要です。
- LiveCycle ES2 で処理されるドキュメントの標準的なサイズ。サイズの大きいドキュメントを処理するには、大容量の共有 GDS ディレクトリが必要です。
- LiveCycle ES2 で処理されるドキュメントの複雑性。複雑なドキュメント（複数の LiveCycle ES2 サービスで処理されるドキュメント）を処理するには、大容量の GDS ディレクトリが必要です。

1.5.4 グローバルドキュメントストレージディレクトリの保護

GDS へのアクセスは保護する必要があります。このディレクトリの長期間有効なドキュメントには、LiveCycle ES2 SDK またはユーザーインターフェイスを使用してアクセスするときに特殊な秘密鍵証明書が必要とする情報など、機密性の高いユーザー情報が含まれる場合があります。

使用しているオペレーティングシステムに適したセキュリティ方式を使用してください。アプリケーションサーバーを実行する際に使用するオペレーティングシステムアカウントだけが、このディレクトリに対する読み取りと書き込みの権限を持つようにすることをお勧めします。

注意： GDS ディレクトリのファイルまたはディレクトリを誤って削除すると、LiveCycle ES2 のインストールが動作不能になる場合があります。

1.5.5 グローバルドキュメントストレージディレクトリのバックアップ

グローバルドキュメントストレージディレクトリをバックアップして、障害が発生した場合に管理者が LiveCycle ES2 を復元できるようにする必要があります。

グローバルドキュメントストレージディレクトリが使用できなくなるかまたはエラーのため失われた場合、GDS ディレクトリおよびデータベースが一貫したバックアップによって復元されるか、LiveCycle ES2 が新規インストールにより再び初期化されるまで、LiveCycle ES2 は実行されません。

LiveCycle ES2 データベースをドキュメントストレージ用に使用する場合は、データベースのバックアップと同時に GDS のバックアップが行われます。詳しくは、[LiveCycle ES2 管理ヘルプ](#)を参照してください。

1.6 LiveCycle Business Activity Monitoring ES2 の要件

注意： Business Activity Monitoring ES2 (BAM) はシステムでサポートされません。システムに BAM ファイルセットと LiveCycle ES2 をインストールできる場合でも、BAM を 64 ビットシステムにデプロイする必要があります。

1.6.1 基本要件

1.6.1.1 専用の JVM

Business Activity Monitoring ES2 では、専用の JVM が必要です。LiveCycle ES2 および Business Activity Monitoring ES2 を同じコンピュータにデプロイする場合、Business Activity Monitoring ES2 が専用の JVM で実行される必要があることに注意してください。このため、以下のように Business Activity Monitoring ES2 は LiveCycle ES2 とは別にデプロイする必要があります。

- JBoss 上にデプロイする場合は、完全に個別な 2 つの JBoss 実装上に 2 つのアプリケーションをデプロイする必要があります。
- WebLogic または WebSphere 上にデプロイする場合は、完全に個別のサーバー定義で 2 つのアプリケーションをデプロイする必要があります。

1.6.1.2 クライアントのメモリ

Business Activity Monitoring ES2 にアクセスするクライアントコンピュータには少なくとも 512 MB の RAM が必要です。推奨は 1 GB です。

1.6.1.3 Web ブラウザ

Business Activity Monitoring ES2 では、Microsoft Internet Explorer 6.0 (パッチ 828750 を適用済み) 以降のブラウザを対象に、BAM Workbench および BAM Dashboard へのアクセスをテストしています。Firefox はサポートされません。

注意： また、Adobe Flash のバージョン 9.0.115.0 以降をインストールする必要があります。

注意： Business Activity Monitoring ES2 の実装がアジア言語で実行されている場合、適切な言語がサポートされるようにブラウザを設定する必要があります。Internet Explorer または Windows のドキュメントを参照してください。

1.6.1.4 メールサーバー

Business Activity Monitoring ES2 では、電子メール通知を配信するために Simple Mail Transfer Protocol (SMTP) 電子メールサーバーが必要になります。このサーバーは Business Activity Monitoring ES2 の外部にあり、電子メールシステム管理者が管理しています。この管理者に依頼して、Business Activity Monitoring ES2 専用のアカウントを設定します。メール送信用のアカウントとパスワード、送信元アドレスフィールドで使用するアドレス、および電子メールサーバーホストの名前が必要です。

1.6.2 Business Activity Monitoring ES2 データベースの作成

Business Activity Monitoring ES2 の機能を使用するために必要なデータベースとして、メタデータデータベースと地理的データベースの 2 つがあります。

BAM Server によって監視されるプロセス評価基準の定義を保存するには、Business Activity Monitoring ES2 メタデータデータベースを作成します。このデータベースには、アラートの詳細やディスクに保存されるオブジェクト実行時データが保存されます。

Business Activity Monitoring ES2 メタデータは大きくなる場合があるので、BAM Server メタデータデータベースには 50 MB 以上を割り当ててください。実稼働用デプロイメントでは、200 MB 以上を割り当ててください。

BAM Server では、BAM Server メタデータデータベース設定の一部で固有の設定が必要になる場合があります。この設定は、BAM Server をホストするアプリケーションサーバーと BAM Server メタデータを保存するデータベースサーバーの種類によって異なります。

また、ダッシュボードで地理的グラフを使用するために必要な地理的マップをサポートするには、Business Activity Monitoring ES2 地理的データベースを作成します。地理的データベースのサイズは、アクセス対象の地理的コードのエントリ数によって異なります。100 MB～150 MB が推奨されます。

BAM Server が BAM Server メタデータデータベースおよび地理的データベースに接続するために使用する、ユーザーアカウントを作成する必要があります。データベースの権限については、[「データベースを使用するための最小限のユーザー権限」\(26 ページ\)](#) を参照してください。その他すべてのデータベース設定については、[LiveCycle ES2 データベース用に説明されている設定を使用してください。](#)[「LiveCycle ES2 データベースの作成」\(54 ページ\)](#) で、使用しているデータベースタイプの項目を参照してください。

1.6.3 JBoss の要件 (BAM)

メタデータおよび地理的データベース：

データベース	JDBC ドライバ
Oracle	<ul style="list-style-type: none">● Oracle Thin ドライバ (ojdbc6.jar) リリース 11.1.0.6 <p>注意： どちらの設定でも Oracle 10g JDBC ドライバを使用します。</p>
MS SQL Server	<ul style="list-style-type: none">● sqljdbc.jar (バージョン 1.2) <p>注意： Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 のどちらのサーバーにも SQL Server JDBC ドライバ 1.2 を使用します。</p> <p>注意： バージョン 2000 のドライバがクラスパスに含まれていないことを確認します。</p>
MySQL	<ul style="list-style-type: none">● mysql-connector-java-5.1.6-bin.jar

注意： sqljdbc.jar ドライバには、Microsoft の既知の問題があります。この問題への対処方法については、Microsoft Knowledge Base (サポート技術情報) の[記事 917054](#) を参照してください。

1.6.4 WebLogic の要件 (BAM)

メタデータおよび地理的データベース：

データベース	JDBC ドライバ
IBM DB2	<ul style="list-style-type: none">db2jcc.jar (バージョン 3.50.152)
MS SQL Server	<ul style="list-style-type: none">sqljdbc.jar (バージョン 1.2) <p>注意： Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 のどちらのサーバーにも SQL Server JDBC ドライバ 1.2 を使用します。</p> <p>注意： バージョン 2000 のドライバがクラスパスに含まれていないことを確認します。</p>
MySQL	<ul style="list-style-type: none">mysql-connector-java-5.1.6-bin.jar
Oracle	<ul style="list-style-type: none">Oracle Thin ドライバ (ojdbc6.jar) リリース 11.1.0.6 <p>注意： どちらの設定でも Oracle 10g JDBC ドライバを使用します。</p>

1.6.5 WebSphere の要件 (BAM)

メタデータおよび地理的データベース：

データベース	JDBC ドライバ
IBM DB2	<ul style="list-style-type: none">db2jcc.jar (バージョン 3.50.152)
MS SQL Server	<ul style="list-style-type: none">sqljdbc.jar (バージョン 1.2) <p>注意： Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 のどちらのサーバーにも SQL Server JDBC ドライバ 1.2 を使用します。</p>
Oracle	<ul style="list-style-type: none">Oracle Thin ドライバ (JDK 1.6 では ojdbc6.jar、JDK 1.5 では ojdbc5.jar) リリース 11.1.0.6 <p>注意： どちらの設定でも Oracle 10g JDBC ドライバを使用します。</p>

1.7 インストールに関する考慮事項

1.7.1 ウイルススキャンの無効化 (Windows のみ)

インストールの速度を上げるには、インストール中にオンアクセスウイルススキャンソフトウェアを無効にします。

1.7.2 ネットワークドライブからのインストール

LiveCycle ES2 は、インストールメディアまたはローカルのディスクドライブからインストールすることのみお勧めします。ネットワーク経由でソフトウェアをインストールしようとする、起動およびインストールにおいて大幅な遅延が発生します。また、ネットワークドライブからインストールすると、ディレクトリパスが長くなり、LiveCycle ES2 のインストーラがインストールを続行できなくなる可能性があります。

1.7.3 LiveCycle ES2 と Luna HSM クラスタの併用

SafeNet Luna がイーサネット接続されたハードウェアセキュリティモジュール (HSM) クラスタを使用する場合は、デバイスで HAOnly モードが有効になっていることを確認します。

▶ Luna デバイスで HAOnly モードを有効にするには

1. Luna クライアントに付属の vtl ツールを使用して、HAOnly モードが有効になっているかどうかを確認します。次のように入力します。

```
vtl haAdmin -HAOnly -show
```

2. 1) HAOnly モードが有効でない場合は、次のように入力します。

```
vtl haAdmin -HAOnly -enable
```

1.7.4 手動による Acrobat の使用制限

ネイティブドキュメントの変換に使用する PDF Generator ES2 または PDF Generator 3D ES2 をインストールした場合は、収録されている Acrobat インストールの使用が Generate PDF および Generate 3D PDF のサービスに制限され、他の使用にはライセンスが供与されません。

1.7.5 一時ディレクトリ

LiveCycle ES2 では、ドキュメントオブジェクト用に設定された最大インラインサイズより容量の大きいドキュメントを格納するために一時ディレクトリが必要です。この設定に適切な値を指定することによって LiveCycle ES2 サーバーのパフォーマンスを向上させる方法については、この[ブログ](#)を参照してください。

(AIX、Linux、および Solaris のみ) ルートユーザー以外がアプリケーションサーバーを実行している場合、ユーザーは指定された一時ディレクトリに対して全権限を持つ必要があります。

注意：一時ディレクトリを作成しない場合は、システム設定のデフォルトの場所が使用されます。

1.7.6 LiveCycle ES2 IPv6 のサポート

LiveCycle ES2 では IPv6 がサポートされています。LiveCycle ES2 のインストールドキュメントに定義されているデフォルトの設定では、IPv4 をデフォルトの IP プロトコルとして設定しています。これは、このプロトコルがサードパーティのインフラストラクチャと最も互換性があるためです。

デプロイメントに必要な場合を除き、IPv6 は有効にしないでください。LiveCycle ES2 で IPv6 サポートを有効にすると、サポート対象のプラットフォーム設定が少なくなります。IPv6 を有効にする場合は、その前に、使用するすべてのサードパーティソフトウェア、ハードウェアおよびネットワークが IPv6 をサポートしていることを確認する必要があります。

注意： IPv6 環境で CIFS を有効にする場合は、LiveCycle Configuration Manager を使用して LiveCycle ES2 インストールを設定した後に、IPv6 設定を明示的に有効にする必要があります。使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイガイド](#) の「IPv6 モードでの CIFS の有効化」を参照してください。

1.7.6.1 サポートされている IPv6 の設定

IPv6 はすべてのインフラストラクチャコンポーネントでサポートされているわけではありません。例えば、MySQL データベースおよび Oracle データベースでは IPv6 はサポートされていません。ただし、アプリケーションサーバーとデータベースの間の接続を IPv4 で、残りの通信を IPv6 経由で設定することにより、これらのデータベースも使用できます。

- IPv6 をサポートするデータベース：Microsoft SQL Server 2005 および 2008、DB2 9.1 以降の 9.x バージョン
- IPv6 をサポートしないデータベース：Oracle 10g、11g、MySQL 5.1
- IPv6 をサポートするアプリケーションサーバー：JBoss 4.2.0、4.2.1、JBoss EAP 4.3、WebLogic 10g R3 および 11g R1、WebSphere 7.0.0.9

1.7.6.2 IPv6 実装のガイドライン

IPv6 実装を部分的または全体的に使用する場合は、次の点に注意してください。

- LiveCycle ES2 をインストール後は、LiveCycle ES2 インストーラから直接 LiveCycle Configuration Manager を起動するオプションは使用しないでください。代わりに、[LiveCycle ES2 root]¥configurationManager¥bin¥IPv6 ディレクトリに移動して、IPv6 固有のスクリプト (ConfigurationManager_IPv6.bat または ConfigurationManager_IPv6.sh) を実行して LiveCycle Configuration Manager を起動します。
- LiveCycle Configuration Manager を使用してアプリケーションサーバーの設定を検証することを選択している場合は、アプリケーションサーバーに対して IPv6 を有効にした後に検証が失敗します。プロセス中はこのエラーメッセージは無視して構いません。IPv6 モードでアプリケーションサーバーを再起動した後で、アプリケーションサーバーをデータベースに接続できます。
- (WebLogic のみ) 管理対象サーバーに関してのみ IPv6 を有効にする必要があります。管理サーバーは引き続き IPv4 上で実行でき、IPv4 アドレスを使用してアクセスできます。ただし、IPv6 環境で起動した管理対象サーバーには、IPv6 アドレスで、または DNS によって解決されたホスト名でのみアクセスできます。
- (WebLogic のみ) アプリケーションサーバーのホストコンピュータ上で LiveCycle Configuration Manager を実行している場合であっても、ブートストラップや LiveCycle ES2 モジュールのデプロイの際には管理対象サーバーのリスンアドレスを指定する必要があります。このリスンアドレスは、コンピュータの IPv6 アドレスに解決される DNS 名である必要があります。
- データベースサーバーと Pure IPv6 通信を行うには、数値の IPv6 アドレスに解決されるデータベースのホスト名を使用するように、EDC_DS および IDP_DS の両方の接続設定を変更します。

- (クラスタインストールのみ) LiveCycle ES2 をサーバークラスタにインストールする場合は、各クラスタノードの DNS または `hosts` ファイルで、各クラスタノードの数値の IPv6 アドレスをコンピュータのホスト名にマップする必要があります。hosts ファイルは次の場所にあります。
 - Solaris : /etc/inet/ipnodes
 - Windows : C:\Windows\system32\drivers\etc\hosts
 - Linux : /etc/hosts
- (JBoss 自動インストールおよび BAM 用 JBoss) 次のファイルを編集します。
 - [LiveCycle ES2 root]\%jboss%bin%run.bat : -Djava.net.preferIPv4Stack=true を -Djava.net.preferIPv6Stack=true に変更します。
 - [LiveCycle ES2 root]\%jboss%bin%service.bat : - b 0.0.0.0 を -b <hostname resolved to IPv6 address> に置き換えます。
- データベースドライバなど、多くのソフトウェアコンポーネントでは、数値の IPv6 アドレスが完全にはサポートされていません。そのため、数値の IPv6 アドレスの代わりに DNS 解決されたホスト名を使用することをお勧めします。
- IPv6 環境では、Microsoft SQL Server を使用している場合は、データベースサーバーの IP アドレスを次の形式で指定する必要があります。この文字列で、;serverName はキーワードなので、実際のサーバー名には置き換えないでください。

```
jdbc:sqlserver://;serverName=<IPv6 address>;  
portNumber=<port>;databaseName=<db_name>
```

ここで、数値の IPv6 アドレスの代わりに、SQL Server データベースのホスト名を指定することもできます。

1.7.6.3 JBoss 用の IPv6 の設定

1. バンドルされている JBoss を解凍します。
2. adobe-ds.xml ファイルおよびデータベースに固有のデータソース設定ファイルを、LiveCycle ES2 データベースに接続するように変更します。
3. login-config.xml ファイルを、LiveCycle ES2 データベースに接続するように変更します。
4. 次のファイルを変更して IPv6 を有効にします。
 - [LiveCycleES2 root]\%jboss%bin%run.bat : このファイルでは次の手順を実行します。
 - -Djava.net.preferIPv4Stack=true を -Djava.net.preferIPv6Stack=true に変更します。
 - -Djava.net.preferIPv6Addresses=true 引数を追加します。
 - [LiveCycleES2 root]\%jboss%bin%service.bat : - b 0.0.0.0 を -b <hostname mapped to IPv6 address> に置き換えます。
5. [LiveCycleES2 root]\%configurationManager%bin%IPv6%ConfigurationManager_IPv6.bat または ConfigurationManager_IPv6.sh スクリプトを呼び出して、LiveCycle Configuration Manager を起動します。
6. LiveCycle Configuration Manager で、EAR を設定し、ブートストラップして LiveCycle ES2 モジュールをデプロイする手順を選択します。
7. LiveCycle Configuration Manager のプロセスが完了したら、これらの EAR ファイルを [LiveCycleES2 root]\%jboss%server%all%deploy ディレクトリにコピーします。

8. コマンドラインから JBoss を起動します。

注意： Linux 上で JBoss を実行している場合は、IPv6 を使用するよう run.sh ファイルを編集する必要があります。

9. IPv6 アドレスにマップされるコンピュータの LiveCycle Configuration Manager ホスト名を指定してから、アプリケーションサーバーをブートストラップして LiveCycle ES2 モジュールをデプロイします。

1.7.6.4 WebLogic 用の IPv6 の設定

1. インストーラを使用して LiveCycle ES2 をインストールします。
2. インストーラが完了しても LiveCycle Configuration Manager を起動しないでください。
[LiveCycleES2 root]¥configurationManager¥bin¥IPv6¥ConfigurationManager_IPv6.bat または ConfigurationManager_IPv6.sh スクリプトを呼び出して、LiveCycle Configuration Manager を起動します。
3. LiveCycle EAR、WebLogic アプリケーションサーバーを設定するオプションを選択し、LiveCycle Configuration Manager を使用してアプリケーションサーバーの設定を検証します。

注意： LiveCycle Configuration Manager によるデータソースの検証が失敗したというエラーメッセージが表示されます。これは、アプリケーションサーバーがまだ IPv6 モードで起動されていないのに、データソースが IPv6 モードで設定されていることが原因です。この段階では、この警告は無視して構いません。

4. WebLogic Server Administration Console から、管理対象サーバーのアプリケーション「Server Start」引数を変更して IPv6 を有効にします。
 - -Djava.net.preferIPv4Stack=true を -Djava.net.preferIPv6Stack=true に変更します。
 - -Djava.net.preferIPv6Addresses=true 引数を追加します。
5. 管理対象サーバーのリスンアドレスを変更し、IPv6 アドレスを使用して有効にします。
 - WebLogic Server Administration Console で、Environment / Servers / 「[Managed Server Name] Configuration」タブを選択します。
 - 「Listen Address」フィールドに、コンピュータのホスト名を入力します。このホスト名が、このコンピュータの IPv6 アドレスに解決されることを確認してください。
6. 変更を保存して、管理対象サーバーを再起動します。
7. [LiveCycleES2 root]¥configurationManager¥bin¥IPv6¥ConfigurationManager_IPv6.bat または ConfigurationManager_IPv6.sh スクリプトを呼び出して、LiveCycle Configuration Manager を起動します。
8. LiveCycle Configuration Manager で、EAR をデプロイし、ブートストラップして LiveCycle ES2 モジュールをデプロイする手順を選択します。
9. 管理対象サーバーのリスンアドレスフィールドで指定したものと同一ホスト名を入力します。

注意： LiveCycle Configuration Manager を同じコンピュータで実行している場合であっても、ブートストラップや LiveCycle ES2 モジュールのデプロイの際には管理対象サーバーのリスンアドレスを指定する必要があります。

1.7.6.5 WebSphere 用の IPv6 の設定

1. インストーラスクリプトを使用して LiveCycle ES2 をインストールします。インストール完了後、指示があっても LiveCycle Configuration Manager を起動しないでください。
2. [LiveCycleES2 root]¥configurationManager¥bin¥IPv6 ディレクトリに移動して、IPv6 固有のスクリプト (ConfigurationManager_IPv6.bat または ConfigurationManager_IPv6.sh) を実行して LCM を起動します。
3. LiveCycle Configuration Manager のオプションを使用して、EAR およびアプリケーションサーバーを設定します。
4. LiveCycle Configuration Manager の手順に従ってアプリケーションサーバーを設定します。データベースを設定中に、IPv6 アドレスにマップするデータベースのホスト名を指定します。
5. LiveCycle Configuration Manager によってアプリケーションサーバーの設定を検証します。データソースの検証が失敗しても、警告を無視します。データソースは WebSphere Administrative Console から検証できます。
6. WebSphere Administrative Console で、JAVA_OPTIONS が指定されているページに移動して、次のタスクを実行します。
 - -Djava.net.preferIPv4Stack=true を -Djava.net.preferIPv6Stack=true に変更します。
 - -Djava.net.preferIPv6Addresses=true 引数を追加します。
7. WebSphere Administrative Console を使用して、EAR ファイルを手動で WebSphere Application Server にデプロイします。設定済みの EAR ファイルは、[LiveCycleES2 root]¥configurationManager¥export フォルダにあります。
8. WebSphere Application Server を再起動します。
9. [LiveCycleES2 root]¥configurationManager¥bin¥IPv6 ディレクトリに移動し、ConfigurationManager_IPv6.bat または ConfigurationManager_IPv6.sh を実行して LiveCycle Configuration Manager を起動します。
10. LiveCycle Configuration Manager で、ブートストラップして LiveCycle ES2 モジュールをデプロイするオプションを選択します。IPv6 アドレスにマップされているホスト名をアプリケーションサーバーに指定します。

注意： IPv6 環境でアプリケーションサーバーを起動後 (-Djava.net.preferIPv6Stack=true フラグを使用) は、IPv6 アドレスまたは IPv6 アドレスにマップされているホスト名でのみアクセスできます。

1.8 CIFS を有効にするためのサーバー設定

LiveCycle ES2 の Windows クライアント用ネットワークフォルダとして Content Services ES2 へのファイルアクセスを有効にするには、Content Services ES2 で Common Internet File System (CIFS) を有効にする必要があります。CIFS を有効にするには、あらかじめ次の手順を実行する必要があります。

- [「仮想インターフェイスの作成 \(AIX、Linux、および Solaris のみ\)」 \(48 ページ\)](#)
- [「CIFS 用の Windows Server の設定」 \(48 ページ\)](#)

注意： IPv6 環境で CIFS を有効にする場合は、LiveCycle Configuration Manager を使用して LiveCycle ES2 インストールを設定した後に、IPv6 設定を明示的に有効にする必要があります。使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイガイド](#) の「IPv6 モードでの CIFS の有効化」を参照してください。

1.8.1 仮想インターフェイスの作成 (AIX、Linux、および Solaris のみ)

AIX、Linux、または Solaris プラットフォームにデプロイされている LiveCycle ES2 で CIFS を有効にする場合は、仮想インターフェイスを作成して、LiveCycle ES2 を実行しているサーバー上の IP アドレスを割り当てる必要があります。これが必要なのは、CIFS プロトコルでは、AIX、Linux、または Solaris コンピュータ上の Samba サービスで使用されているものと同じポートを使用するからです。この仮想 IP アドレスは、LiveCycle Configuration Manager を使用して CIFS を有効にするときに、代替 IP アドレスとして指定します。

IPv6 上で LiveCycle ES2 を使用する場合、CIFS サーバー上で仮想 IPv6 インターフェイスを作成する必要があります。作成するインターフェイスは、CIFS クライアントと同じネットワーク内に存在する必要があります。詳しくは、オペレーティングシステムに関連するドキュメントを参照してください。

注意：再起動後も保持されるようにするには、作成した仮想インターフェイスを /etc/hosts ファイルに追加する必要があります。

1.8.2 CIFS 用の Windows Server の設定

LiveCycle ES2 をホストする Windows Server 2003 および 2008 コンピュータで CIFS を有効にするには、手動で設定する必要があります。CIFS が有効になると、ユーザーはネットワークフォルダとして Content Services ES2 リポジトリにアクセスし、ローカルファイルシステムで様々なファイル操作を実行できます。LiveCycle Content Services ES2 では、ディレクトリプロバイダとして ActiveDirectory を使用するエンタープライズドメインユーザーについて CIFS がサポートされます。

注意：Java ベースの CIFS 実装の場合は、CIFS サーバーに静的 IP アドレスがあることを確認してください。Windows ネイティブ実装の場合は、静的 IP アドレスは必要ありません。

Windows コンピュータで、次の手順を実行する必要があります。

- [「NetBIOS over TCP/IP の有効化」 \(48 ページ\)](#)
- [「IP アドレスの追加」 \(48 ページ\)](#)
- [「SMB over NetBIOS レジストリの無効化 \(Windows 2003 のみ\)」 \(49 ページ\)](#)
- [「ファイルとプリンタの共有の無効化 \(Windows 2008 のみ\)」 \(49 ページ\)](#)

1.8.2.1 NetBIOS over TCP/IP の有効化

NetBIOS over TCP/IP を有効にして、LiveCycle ES2 サーバーに接続するクライアントの要求をサーバーホスト名用に解決できるようにする必要があります。

1. ローカルエリアの接続プロパティダイアログボックスの「全般」タブで、「インターネットプロトコル」を選択して、「プロパティ」をクリックします。
2. インターネットプロトコル (TCP/IP) のプロパティダイアログボックスの「全般」タブで、サーバーに静的 IP アドレスがあることを確認します。「詳細設定」をクリックします。
3. TCP/IP 詳細設定ダイアログボックスで、「WINS」タブを選択して「NetBIOS over TCP/IP を有効にする」を選択します。

1.8.2.2 IP アドレスの追加

注意：この手順は、Java ベースの CIFS 実装のみに必要です。

1. ローカルエリアの接続プロパティダイアログボックスの「全般」タブで、「インターネットプロトコル」を選択して、「プロパティ」をクリックします。

2. インターネットプロトコル (TCP/IP) のプロパティダイアログボックスの「全般」タブで、サーバーに静的 IP アドレスがあることを確認します。「詳細設定」をクリックします。
3. TCP/IP 詳細設定ダイアログボックスで、「IP 設定」タブを選択して「追加」をクリックします。
4. 静的 IP アドレスを指定して「追加」をクリックします。

1.8.2.3 SMB over NetBIOS レジストリの無効化 (Windows 2003 のみ)

Windows レジストリを編集することによって、SMB over NetBIOS を無効にする必要があります。

1. Windows レジストリエディタで、HKEY_LOCAL_MACHINE / SYSTEM / CurrentControlSet / Services / NetBT / Parameters に移動します。
2. DWORD 「SMBDeviceEnabled」を 0 に設定します。存在しない場合は、SMBDeviceEnabled という名前で新しい DWORD 値を追加して、0 に設定します。

1.8.2.4 ファイルとプリンタの共有の無効化 (Windows 2008 のみ)

警告： この手順でファイルとプリンタの共有を無効にすると、Windows ファイル共有プロトコルを使用してサーバーに直接アクセスすることができなくなります。このコンピュータは、Windows ネットワーク表示に表示されません。

- ネットワーク設定に移動して、「Microsoft ネットワーク用ファイルとプリンタ共有」の選択を解除して、「適用」をクリックします。

1.9 ドキュメントフォーム変数および電子署名を使用するプロセス

LiveCycle を以前のバージョンからアップグレードして LiveCycle ES2 サーバーを変更する場合、ドキュメントフォーム変数または電子署名を使用するプロセスが中断される可能性があります。原因は、これらのフォームに送信 URL が設定されており、フォームは一度しかレンダリングされないためです。サーバーを変更すると証明書が破損します。

次の中から、自分の LiveCycle ES2 環境に最適な解決方法を選択してください。

解決方法 1： リモートサーバーをアップグレードするか、リモートサーバーに移行する前に、フォームドキュメント変数を使用するすべてのプロセスを完了します。アップグレード後に従来の LiveCycle サーバーを維持する場合は、この方法を選択します。また、このアプローチを選択すると、フォーム送信のリダイレクトを管理するために「煩雑な」作業を行う必要がなくなります。この方法は、多くの未処理のプロセスがある場合は実用的ではありません。

解決方法 2： アップグレード対象のサーバーの運用が停止されない場合は、リバースプロキシによるアプローチを推奨します。この方法では、移行されたすべてのプロセスが完了するまで古いシステムでリバースプロキシが保持されます。

解決方法 3： Apache mod_rewrite モジュールを使用すると、各フォームの埋め込み URL をクライアントに配信するときにこれらの URL を変更できます。

注意： IPv6 上で LiveCycle を使用する場合、PDF 作成に EJB の呼び出しを使用するクライアントから例外が報告されます。これは Sun JDK 6 固有の既知の問題です。詳しくは、http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6230761 を参照してください。

LiveCycle ES2 Central Migration Bridge

LiveCycle ES2 Central Migration Bridge サービスを使用すると、Adobe Central Pro Output Server または Web Output Pak の製品から既存のアプリケーションを移行して、LiveCycle ES2 Output サービスで動作させることができます。Central Migration Bridge サービスを使用した移行を行うと、LiveCycle ES2 環境で、現在の IFD/MDF テンプレート、データ変換スクリプトおよび DAT ファイルを使用できるようになります。

注意： Central Migration Bridge が有用なのは、移行対象の既存の Central Pro アプリケーションがある場合のみです。

Central Migration Bridge の使用許可

Central Migration Bridge サービスを使用するには、Adobe Central Pro Output Server 5.7 の有効なライセンスを所有しているか、または Adobe Central Pro Output Server 5.7 移行契約を締結している必要があります。Central Pro Output Server 5.7 をインストールするには、既存のメディアおよび既存の製品認証コード (PAC) を使用します。PAC は特定のオペレーティングシステムプラットフォーム用です。これが LiveCycle ES2 のインストール先のオペレーティングシステムプラットフォームと異なる場合は、そのオペレーティングシステムの PAC を取得する必要があります。移行、移行契約の実行、または Central Pro Output Server 5.7 メディアや PAC の取得方法について詳しくは、アドビの営業担当者にお問い合わせください。

インストールに関する考慮事項

Central Migration Bridge サービスは、Central Pro (バージョン 5.7) 実行可能ファイルと直接やり取りします。Central Pro は LiveCycle ES2 と同じサーバーにインストールする必要がありますが、LiveCycle ES2 をインストールするための前提条件ではありません (つまり、LiveCycle ES2 より前でも後でもインストールできます)。インストール手順については、Central Pro のドキュメントセットを参照してください。

警告： Central Pro を起動したり、自動的に実行するように Central Pro のプロパティを変更したりしないでください。

Windows では、Central Pro サービス *Adobe Central Output Server* は、手動のサービスとしてインストールされます。このサービスを実行したり、自動的に実行するようにこのサービスのプロパティを変更したりしないでください。

AIX、Linux、または Solaris では、Central Pro デーモン *jfdaemon* を起動しないでください。コンピュータの再起動時に *jfdaemon* を起動するようにコンピュータの起動スクリプトを編集している場合は、このデーモンが自動的に起動しないようにスクリプトを変更します (Central Pro のインストールドキュメントを参照)。Central は、コマンドラインから *jfserver* プロセスを起動することによって起動しないでください。

注意： Central Migration Bridge を呼び出す LiveCycle ES2 ユーザーには、Central Pro インストールディレクトリに対するアクセス権と、Central Pro 実行ファイルの実行権限が必要です。

LiveCycle ES2 自動インストール

高速モードを使用して、LiveCycle ES2 を自動環境でインストールおよび設定する場合は、Central Migration Bridge サービスはデフォルトでインストールおよび設定されます。何らかの入力を求められることはありません。

注意： Adobe Central Pro 製品がデフォルトのディレクトリにインストールされていることを確認してください。

LiveCycle ES2 カスタムインストール

カスタムモード (部分的な自動または手動) を使用して LiveCycle ES2 をインストールおよび設定する場合は、LiveCycle Configuration Manager で、Central Migration Bridge をデプロイに含めるよう求められます。

デフォルトでは、サービスは Central Pro のデフォルトのインストールパスを使用します。Central Pro が別の場所にインストールされている場合は、LiveCycle 管理コンソールに移動して、Central Migration Bridge Service 用の [Central Install Dir] の設定を更新してください。

LiveCycle ES2 のインストールが完了したら、Central Pro がデフォルトの場所にインストールされていない場合は、次の手順を実行して、LiveCycle ES2 が適切なディレクトリを参照するように指定します。

1. LiveCycle 管理コンソールにログインします。
2. サービス/アプリケーションおよびサービス/サービスの管理をクリックします。
3. 「Central Migration Bridge:1.0」 サービスをクリックします。
4. Central Pro インストールディレクトリの正しいパスを入力します。
5. 「保存」をクリックします。

注意： この設定は、Workbench ES2 でも可能です ([「Creating LiveCycle ES2 Processes」](#) または [「Creating LiveCycle ES2.5 Processes」](#) を適宜参照)。

2

LiveCycle ES2 の秘密鍵証明書と公開鍵証明書

この章では、次のタスクを実行する方法について説明します。

- Adobe LiveCycle Reader Extensions ES2 の使用権限秘密鍵証明書の取得
- Adobe LiveCycle Digital Signatures ES2 で使用する電子証明書の取得

2.1 Reader Extensions ES2 の使用権限秘密鍵証明書の取得

LiveCycle Reader Extensions ES2 の使用権限秘密鍵証明書は Reader Extensions ES2 に固有な電子証明書で、これにより、PDF ドキュメントで Adobe Reader の追加機能を使用できるようになります。秘密鍵証明書がインストールされていない場合、Reader Extensions ES2 ユーザーは使用権限をドキュメントに適用できません。この機能に標準の電子証明書を使用することはできません。専用の使用権限秘密鍵証明書を使用する必要があります。

使用権限秘密鍵証明書は、Reader Extensions ES2 で処理する各 PDF ファイルの追加機能を拡張します。ソフトウェアライセンスの最も重要な部分であり、安全性の高い環境で注意して保管する必要があります。

次のタイプの使用権限秘密鍵証明書を取得できます。

Customer Evaluation (カスタマー評価用) : Reader Extensions ES2 を評価するお客様に提供される、有効期限が短い秘密鍵証明書。この秘密鍵証明書を使用してドキュメントに適用されている使用権限は、証明書の有効期限が切れると失効します。この種類の秘密鍵証明書は、2～3 か月間のみ有効です。

Production (実稼働環境用) : 全製品を購入したお客様に提供される、有効期間が長い秘密鍵証明書。Production (実稼働環境用) 証明書は、ユーザーごとに固有ですが、複数のシステムにインストールできます。

使用権限秘密鍵証明書は、秘密鍵証明書にアクセスするための公開鍵、秘密鍵およびパスワードを含む電子証明書として提供されます。

Reader Extensions ES2 の評価版を注文した場合、評価用の使用権限秘密鍵証明書は、製品注文先の営業担当者または評価版をダウンロードした Web サイトから提供されます。

Reader Extensions ES2 製品版を購入した場合、実稼働環境用の使用権限秘密鍵証明書は、Electronic Software Download (ESD) によって配信されます。実稼働環境用の使用権限秘密鍵証明書は、会社に固有のものであり、必要とする特定の追加機能を有効にすることができます。

Reader Extensions ES2 を自社のソフトウェアに統合したパートナーまたはソフトウェアプロバイダを通じて Reader Extensions ES2 を入手した場合、使用権限秘密鍵証明書は、アドビシステムズ社からパートナーに提供され、その後パートナーからお客様に提供されます。

注意 : 使用権限秘密鍵証明書は、一般的なドキュメントの署名や識別情報のアサーションには使用できません。これらの場合は、自己署名証明書を使用するか、認証局 (CA) から識別情報証明書を取得してください。

2.2 Digital Signatures ES2 で使用する電子証明書の取得

電子証明書は、Digital Signatures ES2 で使用するために必要です。電子証明書は、LiveCycle ES2 をインストールおよび設定した後で設定および管理できますが、インストールの前に取得しておく、LiveCycle ES2 をデプロイした時点で使用できる状態になります。

公開鍵の電子証明書は認証局 (CA) から発行され、電子メールか Web を介して公開鍵証明書ファイルとして送信されます。この公開鍵証明書ファイルには、ドキュメントの暗号化と署名に使用する公開鍵 (公開鍵証明書とも呼ばれる) と秘密鍵 (秘密鍵証明書とも呼ばれる) への参照が含まれます。公開鍵証明書には実際の秘密鍵は含まれませんが、その代わりに、暗号化されたファイルまたは HSM に秘密鍵を安全に保管しているユーザーを識別するための参照が含まれます。

Internet Explorer (Windows) または OpenSSL (AIX、Linux および Solaris) を使用して、コンピュータで利用可能な任意の互換性のある公開鍵証明書ストアに保管されている公開鍵証明書の PFX ファイル、P12 ファイルおよび CER ファイルを書き出すことができます。PFX ファイルは、公開鍵証明書ストアまたは秘密鍵証明書自体によって許可された方式でのみ書き出すことができます。秘密鍵証明書に対応する公開鍵を保持する CER ファイルは、Internet Explorer または OpenSSL のいずれかを使用して PFX ファイルから書き出すこともできます。

注意： LiveCycle ES2 で使用する公開鍵証明書、秘密鍵証明書および証明書失効リスト (CRL) は、Web ベースの LiveCycle 管理コンソールからアクセス可能な Trust Store の管理を使用して設定および管理できます ([LiveCycle 管理ヘルプ](#)を参照)。

CRL 配布ポイントは、特定の CER ファイルまたは PFX ファイルに対応する CRL をダウンロードできる場所を示しています。

次の種類のファイルがサポートされます。

公開鍵証明書： DER でエンコードされた X509v3 および base64 でエンコードされた証明書 (.cer) ファイル。trust.xml ファイルを検証する公開鍵証明書には、DER でエンコードされた証明書または base64 でエンコードされた証明書のいずれかを使用します。

秘密鍵証明書： 標準の PKCS12 形式 (.pfx ファイルおよび .p12 ファイル) の、最大 4096 ビットの RSA 秘密鍵証明書および DSA 秘密鍵証明書。

CRL： base64 および DER でエンコードされた CRL ファイル。

秘密鍵 (秘密鍵証明書) のセキュリティを保護することは、機密情報の安定性を確保する上で欠かせません。通常、物理的なストレージデバイス (一般的にハードウェアセキュリティモジュール (HSM) と呼ばれる) を使用すると、秘密鍵のセキュリティを最大限に保護することができます。物理デバイスを使用しない場合は、機密性の高い秘密鍵や公開鍵証明書は安全な場所にある暗号化ファイルに格納することが重要です。

Digital Signatures ES2 では、業界標準である PKCS #11 インターフェイスを使用して HSM と通信することができます。HSM ストレージシステムのインストールと設定に必要なリソースおよびツールは、HSM ベンダーから取得できます。

3

LiveCycle ES2 データベースの作成

ここでは、LiveCycle ES2 で使用するデータベースを設定する方法について説明します。

データベースには以下のエレメントが含まれます。

- LiveCycle ES2 サービス
- LiveCycle ES2 実行時の設定
- LiveCycle ES2 のプロセスデータ
- 顧客のプロセスの定義とテンプレート
- アプリケーションサーバー管理データ

データベースを作成する前に、プリインストール要件を読み、必要なソフトウェアがインストールされていることを確認する必要があります ([「必要システム構成」\(10 ページ\)](#) を参照)。

LiveCycle ES2 を初めてインストールする場合は、空のデータベースを作成する必要があります。LiveCycle ES2 をサポートするのに必要なすべてのテーブルは、LiveCycle Configuration Manager を使用して LiveCycle ES2 データベースを初期化するときに作成されます (使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイ](#) ドキュメントを参照してください)。

Business Activity Monitoring をインストールする場合は、メタデータデータベースと地理的データベースを作成する必要があります。LiveCycle Process Management ES2 の LiveCycle ES2 Business Activity Monitoring ES2 (BAM) メタデータデータベースおよび地理的データベースの作成については、使用しているアプリケーションサーバー版の [LiveCycle ES2 のインストールおよびデプロイ](#) ドキュメントを参照してください。

3.1 データベース設定の要件

ここでは、LiveCycle ES2 データベースに必要な特殊なテーブル、インデックス、その他のプロパティのうち、LiveCycle Configuration Manager で設定されないものについて説明します。

3.2 Oracle データベースの作成

Oracle 10g または Oracle 11g のインストール時に作成されたデフォルトデータベースを使用しない場合は、Database Configuration Assistant ツールを使用して新しいデータベースを作成します。LiveCycle ES2 は、WebLogic、WebSphere および JBoss アプリケーションサーバー上の Oracle 10g および Oracle 11g をサポートしています。

Business Activity Monitoring ES2 をインストールする場合、まずメタデータデータベースと地理的データベースを作成する必要があります。 [「Business Activity Monitoring ES2 データベースの作成」\(41 ページ\)](#) を参照してください。

注意： LiveCycle ES2 用の Oracle データベースインスタンスの設定では、トランザクション処理用のテンプレートまたは一般処理用のテンプレートを使用することができます。データベースインスタンスの設定にカスタムデータベーステンプレートを使用する場合は、最低限のコンポーネントとして **Oracle JVM** および **Enterprise Manager Repository** を含める必要があります。

Oracle データベースを作成するときは次の手順を実行します。

- 初期データベースサイズは最小値 500 MB に設定します。LiveCycle Content Services ES2 をデプロイする場合はこの初期サイズを増やします。
- LiveCycle ES2 アプリケーションからの永続データに応じてデータベースのサイズを増やせるようにユーザーのクォータを作成します。
- UTF-8 エンコードのサポートを有効にします。
- Database Character Set を Unicode (AL32UTF8) に、National Character Set を AL16UTF16 (Unicode UTF-16 universal character set) に設定します。
- NLS_LENGTH_SEMANTICS を BYTE に設定します (必要な場合)。他の値を設定するとデータベースの初期化が失敗します。
- 「Transaction Processing」を使用して Oracle をインストールし、サーバーの接続モードを「Dedicated Processing」に設定する必要があります。
- Content Services ES2 をデプロイする場合、Oracle データベースのブロックサイズを 16K に設定します。Oracle のインストールディレクトリ内にある initSID.ora ファイルの db_block_size の値を変更します。

注意： この値は使用しているオペレーティングシステムによって異なります。Windows プラットフォームの場合、範囲は 2K ~ 16K です。

3.2.1 ユーザーアカウントと権限

データベース上に新規ユーザーアカウントを作成し、次のシステム権限を割り当てます。

- CREATE SEQUENCE
- CREATE VIEW
- UNLIMITED TABLESPACE
- CREATE TABLE
- CREATE CLUSTER
- CREATE SESSION

警告： (WebLogic のみ) WebLogic と Oracle データベースを使用する場合は、データベースユーザー名の先頭の文字として数字を使用したり、ユーザー名にハイフン (-) を使用したりできません。この制限を守らないと、データベースは正常にブートストラップされません。

ユーザー名は、AIX、Linux または Solaris 上にデプロイする場合は 8 文字以下に、Windows の場合は 12 文字以下にする必要があります。

アプリケーションサーバーでデータソースを設定する場合は、次の情報が必要です。

- SID (サービス ID)
- Oracle ユーザーアカウントのユーザー名とパスワード
- データベースサーバーのホスト名または IP アドレス
- Oracle リスナーポート番号 (デフォルトのポートは 1521)

Oracle 10g または Oracle 11g の使用方法について詳しくは、該当するユーザーマニュアルを参照してください。

3.3 SQL Server データベースの作成

LiveCycle ES2 が実行時データと設定データを格納するために使用する SQL Server データベースを作成することができます。SQL Server データベースの作成方法については、SQL Server のマニュアルを参照してください。LiveCycle ES2 は、JBoss、WebLogic および WebSphere 上の SQL Server 2005 SP3 および 2008 (英語版および日本語版) をサポートしています。

SQL Server データベースを作成し、アプリケーションサーバー上でデータソースを設定する際に使用するユーザーアカウントを作成し、DB_OWNER の権限を割り当てます。データベースおよびユーザーアカウントの作成について詳しくは、SQL Server 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ーでデータソースを設定する場合は、次の情報が必要です。

- データベース名
- SQL Server ユーザーアカウントのユーザー名とパスワード
- データベースサーバーのホスト名または IP アドレス
- SQL Server ポート番号

3.3.1 SQL Server データベースドライバのインストール

注意： まだ行っていない場合は、Microsoft の Web サイトの [Microsoft ダウンロードセンター](#) から SQL Server JDBC Driver 1.2 (全プラットフォーム用) をダウンロードしてください。LiveCycle ES2 のインストール先のサーバーに、SQL Server データベースドライバをインストールするようにしてください。

この Web サイトの指示に従って、ドライバをダウンロードし、インストールします。システム上でドライバをインストールしたディレクトリの場所を控えておいてください。

注意： Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 のどちらのサーバーにも SQL Server JDBC ドライバ 1.2 を使用します。

3.3.2 LiveCycle ES2 用の SQL Server の設定

LiveCycle ES2 データベースを作成する前に、次の設定を変更して SQL Server を最適化します。

3.3.2.1 メモリの拡張

SQL Server のデフォルトの設定では、積極的なメモリ割り当ては行われていません。この場合、SQL Server データベースのほとんどのデプロイメントでパフォーマンスが大きく低下します。

注意： この節の処理は実行することをお勧めしますが、省略することもできます。

▶ SQL Server 用にメモリを拡張するには

1. Microsoft SQL Server Management Studio を使用して、LiveCycle ES2 データベースをホストするデータベースサーバーに接続します。
2. データベースサーバー接続を右クリックして、「Properties」をクリックします。
3. メモリページを選択し、「Minimum Server Memory (in MB)」ボックスにサーバー上の空きメモリのサイズと等しいサイズを入力します。
4. SQL Server データベースを再起動します。

3.3.2.2 プロセッサの優先度の設定

専用のデータベースサーバー (LiveCycle ES2 の実稼働環境へのインストールに推奨) 上では、SQL Server プロセスはシステムの CPU リソースを過度に消費しないように設定されます。

注意： この節の処理は実行することをお勧めしますが、省略することもできます。

▶ プロセッサの優先度を設定するには

1. Microsoft SQL Server Management Studio を使用して、LiveCycle ES2 データベースをホストするデータベースサーバーに接続します。
2. データベースサーバー接続を右クリックして、「Properties」をクリックします。
3. プロセッサページを選択し、「Boost SQL Server Priority」を選択します。
4. SQL Server データベースを再起動します。

3.3.2.3 復旧間隔の拡張

この設定は、クラッシュの後、回復までデプロイメントを待機する時間を指定します。SQL Server のデフォルトの設定は 1 分です。この設定値を大きくすると、データベースログからデータベースファイルに変更を書き込む頻度が低くなるので、パフォーマンスが向上します。この設定では、トランザクションの動作が損なわれることはありませんが、起動時に再生されるログファイルのサイズは影響を受けます。

注意： この節の処理は実行することをお勧めしますが、省略することもできます。

▶ 復旧間隔を拡張するには

1. Microsoft SQL Server Management Studio を使用して、LiveCycle ES2 データベースをホストするデータベースサーバーに接続します。
2. データベース接続を右クリックして、「プロパティ」をクリックします。
3. データベースの設定ページを選択し、「Recovery Interval (Minutes)」ボックスに 5 と入力します。
4. SQL Server データベースを再起動します。

3.3.2.4 統合セキュリティ

注意： これはオプション設定です。

SQL Server 2005 統合セキュリティを使用する場合は、SQL Server 2005 データベースを WebSphere または WebLogic の混合モードに設定する必要があります。JBoss (Windows) では、SQL Server 2005 データベースを「混合モード」または「Windows 認証モード」のいずれかに設定できます。JBoss (Windows) 用に Windows 認証モードに設定する場合は、[「Windows 上での統合セキュリティの設定」\(79 ページ\)](#) の「Windows 上での統合セキュリティの設定」の手順に従う必要があります。

注意： SQL Server データベースは、Windows または SQL Server 認証タイプで設定できます。ただし、LiveCycle ES2 では、認証タイプを SQL Server 認証に設定する必要があります。これは、Windows 認証が SQL Server で使用されている場合、LCM がデータベース接続の検証に失敗するためです。

3.3.3 SQL Server データベースのサイズ変更

SQL Server で提供されるデフォルトのデータベースサイズは、LiveCycle ES2 には小さすぎます。データベースが自動拡張に設定されている場合でも、データベースが大きくなると、または大きくなってディスクがフラグメント化し始めると、パフォーマンスの低下などの予期しない結果が生じることがあります。デプロイ要件を反映するように、作成時に事前にデータベースサイズを割り当てるのが最適です。

中規模のデプロイメント: LDAP ディレクトリ内におよそ 100,000 のユーザーと 10,000 のグループが存在する環境。「Database Data Initial Size」を 1 GB に設定し、自動拡張を 250 MB に設定します。

大規模のデプロイメント: LDAP ディレクトリ内におよそ 350,000 のユーザーと 10,000 を超えるグループが存在する環境。「Database Data Initial Size」を 2 GB に設定し、自動拡張を 1 GB に設定します。

注意: データベースの拡張は、常に特定のサイズに制限されます。管理者は、LiveCycle ES2 データベースのリソース利用状況を監視して、限定された領域、すなわちデータベースが存在するディスク上で使用可能な領域が失われないことを確認します。

3.3.4 LiveCycle ES2 データベースユーザー、スキーマおよびログインの作成

これで、LiveCycle ES2 データベースユーザー、スキーマおよびログインを作成することができます。

警告: LiveCycle ES2 のデータベースインスタンスを作成するときには、必ず SQL_Latin1_General_CP1_CI_AS 照合 (データベースを日本語環境で実行する場合は Japanese_CI_AS 照合) を使用してください。それ以外の照合を使用すると、データベースの初期化に失敗する場合があります。SQL Server データベースの作成時に使用した照合と異なる照合を、LiveCycle ES2 データベースインスタンスに使用できません。

▶ LiveCycle ES2 データベースを作成するには

1. Microsoft SQL Server Management Studio を使用して、「Server」をクリックし、「Database」を右クリックして、「New Database」を選択します。
2. 選択したデータベースの名前を入力します。

注意: データベース名は非常に重要です。選択した名前は、この後の手順で一貫して使用する必要があります。手順では database_name として参照されます。
3. **(SQL Server 2005 のみ)** 「フルテキストインデックスを使用する」が選択解除されていることを確認します。
4. 「Database Data Initial Size MB」ボックスに適切な値を入力します。
 - 小規模な開発または小規模な実稼働システムでは、200 MB を指定します。
 - 大規模なシステムについては、[「SQL Server データベースのサイズ変更」\(58 ページ\)](#) を参照してください。
5. 「Database Data Autogrowth」ボックスに、50% と入力します。
6. 「Database Log Initial Size」ボックスに適切な値を入力します。
 - 小規模な開発または小規模な実稼働システムでは、20 MB を指定します。
 - 大規模なシステムについては、[「SQL Server データベースのサイズ変更」\(58 ページ\)](#) を参照してください。
7. 「Database Log Autogrowth」ボックスに、50% と入力します。
8. 「OK」をクリックすると、データベースが作成されます。

3.3.4.1 SQL Server での LiveCycle ES2 ユーザーの作成

以下の手順では、[database_name] はデータベースの作成時に指定した名前を表し、[database_username] は新規ユーザーに指定する名前を表します。

▶ LiveCycle ES2 ユーザーを作成するには

1. Microsoft SQL Server Management Studio を使用して、LiveCycle ES2 データベースを作成したデータベースサーバーに接続します。
2. Server / Security をクリックし、「Logins」を右クリックして、「New Login」を選択します。
3. ログイン名 [database_username] を入力し、次に、「SQL Server Authentication」を選択して、新しいパスワードを入力します。
4. 「Enforce Password Expiration」を選択解除します。また、「User must change password on next login」も選択解除します。
5. デフォルトのデータベースは「Master」のままにし、「OK」をクリックします。
6. Server / Databases / [database_name] / Security をクリックし、「Schemas」を右クリックして、「New Schema」を選択します。
7. 「Schema Name」ボックスに [database_username] を入力し、「OK」をクリックします。
8. Server / Databases / [database_name] / Security をクリックし、「Users」を右クリックして、「New User」を選択します。
9. 新しいユーザーダイアログボックスで、ログイン名とユーザー名 [database_username] を入力します。
10. デフォルトのスキーマを [database_username] に設定します。

注意： スキーマ名は [database_username] と同じである必要があります。

11. 「このユーザーが所有するスキーマ」領域で、[database_username] スキーマを選択します。
12. 「Database Role Membership」領域で、「db_owner」を選択します。

3.3.4.2 LiveCycle ES2 ユーザーとデータベースの関連付け

LiveCycle ES2 ユーザーを作成した後に、このユーザーを LiveCycle ES2 データベースに関連付けます。

▶ LiveCycle ES2 ユーザーをデータベースに関連付けるには

1. Security / Logins をクリックし、[database_username] を右クリックして、「Properties」をクリックします。
2. ログインのプロパティの全般ページで、ユーザーのデフォルトのデータベースを [database_name] に設定します。
3. 「User Mapping」ページを選択し、「Users Mapped To This Login」セクションで、[database_name] が選択されていること、「User」が [database_username] に設定されていること、「Default Schema」が [database_username] に設定されていることを確認します。
4. 「Users Mapped To This Login」テーブルで [database_name] が選択されていることを確認し、「Database Role Membership For [database_name]」テーブルで「db_owner」および「public」が選択されていることを確認して、「OK」をクリックします。

3.3.4.3 LiveCycle ES2 データベースの分離レベルの設定

SQL Server 2005 には、いくつかの新しい分離レベルが提供されています。LiveCycle ES2 では、長時間実行されているトランザクションと同時期に短時間の読み取りが何度も発生する場合のデッドロックを管理するために、固有の分離レベルが必要です。

▶ 分離レベルを設定するには

1. 「Databases」をクリックし、[database_name] を右クリックして、「New Query」を選択します。

注意： [database_name] は、データベースの作成時に指定した名前を表します。

2. クエリーパネルで、次のテキストを入力します。

```
ALTER DATABASE [database_name]
SET READ_COMMITTED_SNAPSHOT ON
GO
```

3. 「Execute」をクリックします。応答がメッセージパネルに表示されます。

3.4 DB2 データベースの作成

この節で説明しているスクリプトのいずれかを実行して DB2 データベースを作成します。このスクリプトは、1 GB のメモリをデータベースに使用するシステムに合わせたものです。システムのデータベース用に割り当てられているメモリ容量がこれより少ない場合、システム設定の調整について詳しくは、DB2 のマニュアルを参照してください。LiveCycle ES2 は、DB2 と WebSphere 6.1、WebSphere 7.0 および WebLogic の組み合わせをサポートしています。

アプリケーションサーバーでデータソースを設定する場合は、次の情報が必要です。

- データベース名
- DB2 ユーザーアカウントのユーザー名とパスワード
- データベースサーバーのホスト名または IP アドレス
- DB2 ポート番号

3.4.1 DB2 ユーザーアカウント

LiveCycle ES2 には、専用のシステムデータベースアカウントが必要です。アカウントがスキーマ所有者の場合は、それ以外の権限は必要ありません。スキーマ所有者は、スキーマ内の任意のオブジェクトを変更、作成、破棄できます。アカウントがスキーマ所有者ではない場合は、スキーマの権限 ALTERIN、CREATEIN、DROPIN が必要です。さらに、ユーザーは DB2USERS グループに属している必要があります。

DB2 ユーザーアカウントは、製品のインストール時とシステムの実行時の両方にテーブルを作成するために、スキーマ所有者の権限を持つ必要があります。アプリケーションサーバーによっては、データベースユーザーとスキーマに対して、データベースを内部で使用するための要件が追加される場合があります。

ユーザーアカウントの作成について詳しくは、DB2 のマニュアルを参照してください。

ユーザー名は、AIX、Linux または Solaris 上にデプロイする場合は 8 文字以下に、Windows の場合は 12 文字以下にする必要があります。

▶ **DB2 データベースを作成するには：**

1. **(Content Services ES2 は含まれません)** DB2 をホストするコンピュータで、使用しているデータベースバージョン用の次の DB2 スクリプトを含む新しいテキストファイルを作成します。

注意： 次のテキストには、改行が含まれています。このドキュメント以外の場所にこのテキストをコピーする場合は、改行を削除してください。

DB2 9.1

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root¥DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root¥DBNAME_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP8K;
commit work;
connect reset;
connect to dbname;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool bp8k immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

DB2 9.5

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL "BP8K" SIZE 50000 PAGESIZE 8192;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root¥DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root¥DBNAME_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16 BUFFERPOOL
BP8K;
commit work;
connect reset;
connect to dbname;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool bp8k immediate size 32000;
```

```
commit work;  
connect reset;  
update db cfg for dbname using dbheap 4000;  
update db cfg for dbname using logbufsz 2048;  
update db cfg for dbname using locklist 2000;  
update db cfg for dbname using chngpgs_thresh 40;  
update db cfg for dbname using logfilsiz 4000;  
deactivate database dbname;  
activate database dbname;
```

2. **(Content Services ES2 は含まれます)** Content Services ES2 を他の LiveCycle ES2 モジュールのいずれかと併用する場合、使用しているデータベースバージョン用の次の DB2 スクリプトを使用します。

注意： 次のテキストには、改行が含まれています。このドキュメント以外の場所にこのテキストをコピーする場合は、改行を削除してください。

DB2 9.1

```
create database dbname using codeset utf-8 territory default;  
connect to dbname;  
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;  
CREATE BUFFERPOOL BP32K SIZE 4000 PAGESIZE 32768 NOT EXTENDED STORAGE;  
connect reset;  
connect to dbname;  
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP  
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING  
( 'DB2_root¥DBNAME_TEMP' )  
EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;  
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP  
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING  
( FILE 'DB2_root¥DBNAME_DATA' 64000 ) EXTENTSIZE 16 PREFETCHSIZE 16  
BUFFERPOOL BP8K;  
commit work;  
connect reset;  
connect to dbname;  
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_32K IN DATABASE PARTITION GROUP  
IBMTEMPGROUP PAGESIZE 32768 MANAGED BY SYSTEM USING  
( 'DB2_root¥DBNAME32k_TEMP' ) EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL  
BP32K;  
CREATE REGULAR TABLESPACE DBNAME_DATA_32K IN DATABASE PARTITION GROUP  
IBMDEFAULTGROUP PAGESIZE 32768 MANAGED BY DATABASE USING  
( FILE 'DB2_root¥DBNAME32k_DATA' 9000 ) EXTENTSIZE 16  
PREFETCHSIZE 16 BUFFERPOOL BP32K;  
alter bufferpool ibmdefaultbp immediate size 96000;  
alter bufferpool bp8k immediate size 32000;  
commit work;  
connect reset;  
update db cfg for dbname using dbheap 4000;  
update db cfg for dbname using logbufsz 2048;  
update db cfg for dbname using locklist 2000;  
update db cfg for dbname using chngpgs_thresh 40;  
update db cfg for dbname using logfilsiz 4000;  
deactivate database dbname;  
activate database dbname;
```

DB2 9.5

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL BP8K SIZE 50000 PAGESIZE 8192;
CREATE BUFFERPOOL BP32K SIZE 500 PAGESIZE 32768;
CREATE TEMPORARY TABLESPACE dbname_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root¥dbname8_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE dbname_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root¥dbname8_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP8K;
CREATE TEMPORARY TABLESPACE dbname_TEMP_32K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 32768 MANAGED BY SYSTEM USING
('DB2_root¥dbname32_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL
BP32K;
CREATE REGULAR TABLESPACE dbname_DATA_32K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 32768 MANAGED BY DATABASE USING
(FILE'DB2_root¥dbname32_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16
BUFFERPOOL BP32K;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool BP8K immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

3. スクリプトを次のように変更します。

- *dbname* および *DBNAME* のインスタンスを、使用する LiveCycle ES2 データベースの名前に置き換えます。
- 同じホストに複数のデータベースインスタンスを作成する場合、合計の物理メモリの 10% を超えないバッファプールサイズを使用して、各データベースインスタンス用に一意の名前を付けたバッファプールを作成します (BP8K_1 など)。例えば、物理メモリが 1 GB のマシンでは、合計のバッファプールサイズが 100 MB (「SIZE 100000」) を超えないようにします。
- *DB2_root* を、DB2 がインストールされているルートディレクトリへのパスに置き換えます。
- コマンドに改行が含まれていないことと、各コマンドがセミコロン (;) で終了していることを確認します。
- 次の行の 9000 を、データベースのサイズに基づいて変更します。

```
(FILE'DB2_root¥DBNAME_DATA'9000)
```

この値は、データベースを初期化するのに必要な最小ページ数を指定します。データベースを初期化した後に、DB2 管理ツールを使用してこの値を変更することもできます。

4. テキストファイルを DB2 コマンドラインプロセッサがアクセスできる場所に保存します。

5. DB2 コマンドプロンプトを開き、次のコマンドを入力してスクリプトを実行します。

```
db2 -tf <path_to_script_file>/<script_file_name>
```

DB2 データベースを同時使用に対応するように設定する必要があります。

▶ **DB2 を同時使用対応に設定するには**

1. DB2 コントロールセンターを開きます。
 - (Windows) スタート / すべてのプログラム / IBM DB2 / General Administration Tools / Control Center を選択します。
 - (AIX、Linux および Solaris) コマンドプロンプトから、db2jcc コマンドを入力します。
2. DB2 コントロールセンターのオブジェクトツリーで、「All Databases」をクリックします。
3. LiveCycle ES2 製品用に作成したデータベースを右クリックし、「Configuration Advisor」をクリックします。
4. Configuration Advisor ウィザードの手順に従い、次のプロパティを設定します。

DB2 のプロパティ	必要な値
Workload type	Mixed
Average number of SQL transactions per unit of work	10 未満
Database Administration Priority	Faster transaction performance
Populated Database	データベースインスタンスの現在の状態に従って設定します。LiveCycle Configuration Manager によって初期化されていない場合、インスタンスは設定されません。
Isolation Level	Cursor Stability

5. 「完了」をクリックします。

▶ **新規データベースにユーザーを追加するには**

1. IBM コントロールセンターにログインします。
2. [database_name] / User and Group Object / DB Users をクリックします。
3. 「Add User」を選択してユーザーを選択します。
4. 「Authorities」で、「Connect to database」、「Create tables」および「Database administration authority」を選択し、「Apply」を選択します。
5. 「OK」をクリックします。

3.4.2 DB2 に関する LiveCycle Content Services ES2 のその他の要件

「[DB2 データベースを作成するには :](#)」(61 ページ) の手順 2 に示したスクリプトにより、DB2 の追加データベースページサイズが作成され、32 KB に設定されます。LiveCycle Content Services ES2 をデプロイしていない場合は、デフォルトの DB2 データベースページサイズ 8 KB を使用できます。

3.4.2.1 複数の DB2 スキーマの設定

1 つの DB2 データベースインスタンスで複数のスキーマを使用して実行するように Content Services ES2 を設定すると、Content Services ES2 のデプロイは、最初のノードでは成功しますが、後続のノードではすべて失敗します。DB2 では大文字と小文字が区別され、値は大文字で入力する必要があります。この問題を回避するには、次の JVM 引数をアプリケーションサーバーに追加する必要があります。

```
-Dhibernate.default_schema=<schema_name>
```

注意： <schema_name> は、大文字のスキーマ名に置き換える必要があります。

3.5 MySQL データベースの作成

注意： このドキュメントで説明する内容は、MySQL を手動でインストールするユーザーを対象にしています。自動インストールに関する説明はありません。『[LiveCycle ES2 の自動インストールおよびデプロイ \(JBoss 版\)](#)』を参照してください。

MySQL ツールを使用して、LiveCycle ES2 で使用する MySQL 5 データベースを作成し、データベースに接続する際にアプリケーションサーバーで使用する MySQL ユーザーアカウントを作成します。さらに、MySQL データベースサーバーの設定も変更する必要があります。データベースおよびユーザーアカウントの作成について詳しくは、MySQL のマニュアルを参照してください。LiveCycle ES2 は、MySQL 5 と、JBoss 4.2.0、JBoss 4.2.1、WebLogic 10g R3 および WebLogic 11g R1 の組み合わせをサポートします。

注意： MySQL では、ユーザー名やパスワードに特殊文字やスペースを使用することはサポートしません。ユーザー名とパスワードがこの制限に従っていることを確認してください。また、Scheduler のエラーを避けるために MySQL データベース名ではハイフン (-) を使用しないでください。

注意： AIX、Linux または Solaris では、lower_case_table_names システム変数に 1 を設定して、テーブル名で大文字と小文字が区別されるようにする必要があります。Windows システムでは、このパラメータを指定しても何も効果はありません。テーブル名での大文字と小文字の区別の設定について詳しくは、<http://dev.mysql.com/doc/refman/5.1/en/identifier-case-sensitivity.html> を参照してください。

注意： グラフィカルユーザーインターフェイス (GUI) を使用して、MySQL でデータベーススキーマおよび新規ユーザーを設定する場合は、MySQL Administrator ツールをインストールする必要があります (MySQL のユーザーマニュアルを参照)。

MySQL データベースをインストールするときは、MYSQL UI の変数セクションで UTF-8 文字エンコードを指定する必要があります。

JBoss でデータソースを設定する際は、次の情報が必要です。

- MySQL ユーザーアカウントのユーザー名とパスワード
- データベースサーバーのホスト名または IP アドレス
- MySQL ポート番号

3.5.1 MySQL データベースの場合のプロセスでのドキュメントの使用に関する制限

MySQL データベースでは、テーブルの行に格納できるデータの容量が制限されます。ドキュメントの複数インスタンスに関連するプロセスでは、MySQL の行サイズ制限を上回る記憶領域が必要となることがあります。LiveCycle ES2 では、1つのプロセスインスタンスのデータがテーブルの1つの行に格納されるので、大容量データに関連するプロセスの場合、行サイズに関する MySQL の制限を超えて、エラーが発生することがあります。

ドキュメントの多数のインスタンスに関連するプロセスを実行して、データベースでエラーが発生した場合は、ドキュメントを処理する操作が複数のサブプロセスで実行されるようにプロセスを再設計することができます。例えば、プロセスを再設計して操作を2つのプロセスで実行することができます。最初のプロセスが元のプロセスの前半の操作に対応し、2番目のプロセスが後半の操作に対応します。この場合、最初のプロセスに2番目のプロセスを呼び出す処理を追加して、2番目のプロセスがサブプロセスとして実行されるようにする必要があります。

3.5.2 MySQL ユーザーアカウント

作成する MySQL ユーザーアカウントには、LiveCycle ES2 データベース内のテーブルにアクセスするために次の権限が必要です。

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- REFERENCES
- INDEX
- ALTER
- CREATE_TMP_TABLE
- LOCK_TABLES

3.5.3 MySQL 初期サーバー設定についての要件

my.ini ファイル (Windows) または /etc/my.cnf ファイル (AIX、Linux または Solaris) の `mysqld` セクションでは次の設定が必要です。

- `max_allowed_packet=25M`

注意： この設定に失敗すると、「パケットが大きすぎる」というエラーが発生し、結果としてモジュールのデプロイが失敗します。

my.ini ファイル (Windows) または /etc/my.cnf ファイル (AIX、Linux または Solaris) の `mysqld` セクションでは次の設定をお勧めします。

- `key_buffer_size=64M`
- `sort_buffer_size=1M`
- `default-storage-engine=INNODB`
- `innodb_buffer_pool_size=100M`
- `innodb_additional_mem_pool_size=5M`

- innodb_log_file_size=170M
- innodb_log_buffer_size=8M
- innodb_flush_log_at_trx_commit=2
- query_cache_type=2
- default-character-set=utf8
- read_buffer_size=1M
- read_rnd_buffer_size=1M

4 JBoss Application Server の設定

JBoss Application Server を使用する場合は、3 種類のインストールオプションがあります。

- JBoss 4.2.1 の自動オプション (Windows のみ)
- アドビにより事前設定された JBoss と JBoss 4.2.1
- 手動による JBoss の設定と JBoss 4.2.0 または JBoss 4.2.1 ([「付録 - JBoss の手動設定」\(99 ページ\)](#) を参照)

ここでは、自動インストールオプションを使用しない場合のアドビの事前設定 JBoss Application Server の使用について説明します。以下のいずれかのシナリオに当てはまる場合は、自動インストール以外の方法を使用する必要があります。

- Linux または Solaris オペレーティングシステムを使用している。
- Oracle または SQL Server を使用しているか、自動セットアップによりデフォルトでインストールされる MySQL の使用を希望していない。
- JBoss 設定を詳細に制御する必要がある。

LiveCycle ES2 は、Windows Server 2003 SP2 および R2 SP2 (Enterprise Edition または Standard Edition)、Windows Server 2008、Red Hat Enterprise Linux 5.0、Red Hat Enterprise Linux Advanced Platform 5.0、Solaris 10、SUSE Enterprise Linux 10 の各プラットフォームで JBoss を実行します ([「サードパーティのインフラストラクチャのサポート」\(10 ページ\)](#) を参照してください)。

注意： JBoss 版での LiveCycle ES2 の自動インストールを使用する場合は、このドキュメントを読む必要はありません。自動インストールの環境を準備するために必要なすべての手順は、『[自動インストールおよびデプロイ \(JBoss 版\)](#)』に記載されています。

この章では、次の表記を使用します。

名前	デフォルト値
[appserver root]	(Windows) C:\jboss (Linux、Solaris) /opt/jboss

4.1 アドビにより事前設定された JBoss

アドビの事前設定 JBoss ソフトウェアは、LiveCycle ES2 メディアの一部として提供されます。これは、JBoss をインストールする最も単純な方法です。JBoss の詳しい知識は必要ありません。

JBoss のサポート対象バージョンをインターネットからダウンロードする場合は、[「付録 - JBoss の手動設定」\(99 ページ\)](#) の設定手順を実行する必要があります。

アドビは、標準の JBoss 4.2.1 アーカイブを JBoss Web サイトからダウンロードし、LiveCycle ES2 をスタンドアロン設定で実行するために必要なオプションを使用して JBoss を設定します。これらの設定については、次の節で確認できます。

注意： JBoss 4.2.1 のデプロイメントによって、[appserver root]/server/ フォルダ以下に、各データベースタイプのデータソースのディレクトリが個別に作成されます。各データベースの先頭には「lc_」が付きます。そのため、MySQL、Oracle および MS SQL データベースのデータソースファイルは、それぞれ lc_mysql、lc_oracle および lc_sqlserver の各ディレクトリに保存されます。以降の節では、これらの個別ディレクトリを示すのに lc_<db-name> を使用します。

Electronic Software Distribution (ESD) をアドビの Web サイトからダウンロードした場合は、JBoss_DVD.zip (Windows) または JBoss_DVD_unix.tar.gz (Linux または Solaris) のいずれかのファイルをファイルシステムに抽出します。メインのアーカイブファイルが抽出されたら、アーカイブされたディレクトリ ¥additional、¥livecycle_server および ¥third_party を抽出します。

注意： 元の ESD ファイルのディレクトリ階層は変更しないようにしてください。

事前設定 JBoss ソフトウェアは、インストールメディアまたは Electronic Software Distribution (ESD) ダウンロードの /third_party ディレクトリにあります。このフォルダには次の 2 つの JBoss インストーラが含まれています。

- jboss.zip ファイルには、アドビの事前設定 JBoss 4.2.1 が含まれています。この JBoss バージョンは、LiveCycle ES2 サーバーのインストールに使用します。
- jboss_4.2.1_bam.zip ファイルには、Business Activity Monitoring ES2 用のアドビの事前設定 JBoss 4.2.1 が含まれています。

インストール要件に応じて、JBoss のディレクトリおよびその内容を、JBoss のインストール先にコピーします。どちらの JBoss をインストールしてもすべての設定手順が完了しますが、[「アドビの事前設定 JBoss 4.2.1 用 LiveCycle ES2 データベース接続の設定」\(71 ページ\)](#) で説明している手順は除きます。MySQL を使用する場合は、この節の手順を完了する必要があります。MySQL データベースを使用しない場合は、データベースの適切な設定手順を完了する必要があります。設定手順の詳細は、この章で後述します。

注意： JBoss 4.2.0 を使用する場合は、JBoss 4.2.0 インストーラを JBoss ダウンロードサイトからダウンロードして、手動で設定する必要があります。アドビで提供しているのは、LiveCycle ES2 インストールメディアで事前設定されている JBoss 4.2.1 のみです。

4.2 Sun Java JDK for JBoss のインストール

JBoss 4.2.0 および JBoss 4.2.1 の場合、次の Sun JDK バージョンを www.java.sun.com からダウンロードし、インストールする必要があります。

- (JBoss 4.2.0) Sun JDK 5.0 Update 11 (または 5.0 の以降の更新)
- (全プラットフォームの JBoss 4.2.1) Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)

Sun Java JDK がインストールされている場所を示す JAVA_HOME 環境変数を作成または設定します。

▶ JAVA_HOME 環境変数を設定するには (Windows) :

1. スタート/コントロールパネル/システムの順に選択します。
2. 「詳細設定」タブをクリックして「環境変数」をクリックします。
3. 「新しいシステム変数」領域で、「新規」をクリックします。

- 新しいシステム変数ダイアログボックスで、変数名として `JAVA_HOME` を入力し、Java JDK をインストールしたディレクトリを入力します。このディレクトリには `/bin` サブディレクトリが含まれています。例えば、次のパスを入力します。

```
C:¥Program Files¥Java¥jdk1.6.0_14
```

▶ **PATH 環境変数を設定するには (Windows) :**

- スタート/コントロールパネル/システムの順に選択します。
- 「詳細設定」タブをクリックして「環境変数」をクリックします。
- 「システム環境変数」領域で、**Path** を選択して、「編集」をクリックします。
- 値の先頭に次のテキストを追加します。

```
%JAVA_HOME%¥bin;
```

▶ **JAVA_HOME 環境変数を設定するには (Linux または Solaris) :**

- 次の例に示すように、Borne および Bash シェルで `JAVA_HOME` 変数を設定することをお勧めします。

```
JAVA_HOME=/usr/java  
export JAVA_HOME
```

▶ **PATH 環境変数を設定するには (Linux または Solaris) :**

- 次の例に示すように、Borne および Bash で `PATH` 変数を設定することをお勧めします。

```
PATH=$JAVA_HOME/bin:$PATH  
export PATH
```

▶ **(Windows、Linux または Solaris) JAVA_HOME 環境変数の設定を検証するには :**

(オプション) コマンドプロンプトを開いて、次のコマンドを実行します。

```
java -version
```

インストールした Java バージョン (1.6.0_14 など) で始まる応答を受信します。

4.3 Linux および Solaris のログオンスクリプト

実稼働で使用するための環境を準備するために、ユーザーのログイン時に環境を自動的に設定することをお勧めします。それには、JBoss プロセスの実行に使用するユーザーのユーザーログオンスクリプトで `JAVA_HOME` および `PATH` 環境変数を設定する必要があります。

4.4 JBoss の起動と停止

この章で説明する一部の手順では、製品をデプロイする JBoss のインスタンスを停止して起動する必要があります。

注意 : これらの手順は、アドビの事前設定 JBoss および手動でインストールした JBoss ソフトウェアの両方に適用されます。

JBoss の起動設定はすべて `[appserver root]/server` ディレクトリにあります。アドビの事前設定 JBoss の場合、どの起動設定を呼び出すかは、インストール済みのデータベース (`lc_mysql`、`lc_oracle` および `lc_sqlserver`) によって異なります。Red Hat から直接入手した JBoss の場合は、JBoss で提供されている起動設定のいずれか (`all`、`default` または `minimal`)、またはカスタム設定を使用します。

▶ JBoss を起動するには：

1. コマンドプロンプトで [appserver root]/bin に移動します。
2. 次のコマンドを入力して、アプリケーションサーバーを起動します。
 - (Windows) `run.bat -c [config] -b [server_IP_Address]`
 - (Linux および Solaris) `./run.sh -c [config] -b [server_IP_Address]`

[config] はデータベースに必要な設定、[server_IP_Address] はサーバーの実際の IP アドレスに置き換えます。

注意： アドビの事前設定 JBoss にリモートからアクセスする場合は、特定の IP アドレス、または `-b 0.0.0.0` を使用するすべてのインターフェイスにバインドする必要があります。ホスト名または IP アドレスを使用する場合、localhost を含む URL を使用しても JBoss に接続できません。

例えば、Windows で実行している SQL Server データベース用に事前設定の JBoss インスタンスを起動するには、次のように入力します。

```
run.bat -c lc_sqlserver -b [server_IP_Address]
```

▶ JBoss を停止するには：

1. コマンドプロンプトで [appserver root]/bin に移動します。
2. 次のコマンドを入力して、アプリケーションサーバーを停止します。
 - (Windows) `shutdown.bat -s <server name>:<jndi-port>`
 - (Linux および Solaris) `./shutdown.sh -s <server name>:<jndi-port>`

4.5 JBoss アプリケーションサーバーの設定に関する注意事項

JBoss アプリケーションサーバーは様々な XML 設定ファイルを使用して設定されます。これらの設定ファイルのいずれかを編集するためには、事前に JBoss を停止する必要があります。JBoss の実行中にこれらのファイルを変更すると、JBoss で障害が発生する可能性があります。Linux または Solaris でも、JBoss は .property 設定ファイルをいくつか使用しますが、これらは UNIX テキストファイルでなければなりません。どのような場合でも Windows 環境でこれらのファイルを編集するときには注意してください。

4.6 アドビの事前設定 JBoss 4.2.1 用 LiveCycle ES2 データベース接続の設定

LiveCycle ES2 データベースへの接続を設定するには、次のタスクを実行する必要があります。

- LiveCycle ES2 データソースを設定します。
- データベースをデフォルトのデータソースとして使用するよう JBoss を設定します。

アプリケーションサーバーのインストールディレクトリにデータベースドライバをインストールする必要があります。ドライバは、LiveCycle Configuration Manager およびアプリケーションサーバーが LiveCycle ES2 データベースに接続できるようにするために必要です。使用するデータベースの種類に対応したドライバをインストールする必要があります。

データベースに接続するデータソースを設定する必要があります。JBoss の場合は、MySQL、SQL Server または Oracle データソースを設定できます。

注意： 以下の操作に進む前に、JBoss が実行されていないことを確認してください。

4.6.1 アドビの事前設定 JBoss 4.2.1 用 MySQL の設定

アドビの事前設定 JBoss から LiveCycle ES2 データを格納している MySQL データベースへの接続を有効にするには、データソースファイルを作成して、LiveCycle ES2 のデプロイ先 JBoss のインスタンスにデプロイする必要があります。

手動でインストールした JBoss を使用する場合は、[「手動でインストールした JBoss での MySQL の設定」\(108 ページ\)](#) を参照してください。

注意： <http://community.jboss.org/wiki/EncryptingDataSourcePasswords> で説明されているいずれかの方法を使用して、データソースファイル (adobe-ds.xml および mysql-ds.xml) および login-config.xml ファイルのパスワードを暗号化します。
http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clear.html の手順を使用することもできます。

4.6.1.1 adobe-ds.xml ファイルの編集

MySQL データソースを設定する前に、データベースを MySQL に作成しておく必要があります ([「MySQL データベースの作成」\(65 ページ\)](#) を参照)。

1. テキストエディタで [appserver root]/server/lc_mysql/deploy/adobe-ds.xml ファイルを開き、IDP_DS と EDC_DS の両方に関して次の行を検索します。

```
<connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>  
<driver-class>com.mysql.jdbc.Driver</driver-class>  
<user-name>adobe</user-name>  
<password>adobe</password>
```

2. 次のテキストをデータベースに固有の値に置き換えます。

- localhost : データベースをホストするコンピュータの名前、IP アドレス、または完全修飾パス。デフォルトは localhost です。
- 3306 : データベースへのアクセスに使用するポート。デフォルトのポートは 3306 です。
- adobe : LiveCycle ES2 データを格納しているデータベースの名前。デフォルト値 adobe をデータベースの名前に置き換えます。

3. <connection-url> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

4. データソース接続の最小値と最大値が次のように設定されていることを確認します。

- IDP_DS の場合：

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>30</max-pool-size>
```
- EDC_DS の場合：

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```

注意： LiveCycle ES2 サーバーで処理する負荷が大きい場合は、JDBC 接続の最大数を増やして、すべてのジョブが確実に処理されるようにします。そのような場合は、IDP_DS と EDC_DS の両方で <max-pool-size> を 50 以上に増やします。

5. ファイルを保存します。

4.6.1.2 mysql-ds.xml ファイルの編集

MySQL データベースと共に LiveCycle ES2 を実行する場合は、MySQL を JBoss のデフォルトのデータソースに設定する必要があります。この手順では、MySQL JDBC ドライバが [appserver root]/server/lc_mysql/lib ディレクトリにインストールされていることを前提としています。

1. [appserver root]/server/lc_mysql/deploy/mysql-ds.xml ファイルをテキストエディタで開き、`<local-tx-datasource>` エレメントを MySQL の接続設定の内容に変更します。

```
<jndi-name>DefaultDS</jndi-name>
<connection-url>jdbc:mysql://localhost:3306/adobe/</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```

 - **localhost** : この値を、データベースをホストするサーバー名に置き換えます。
 - **3306** : この値を、データベースサーバーのポート番号に置き換えます。
 - **adobe** : この値を、LiveCycle ES2 に接続するデータベースに置き換えます。
2. `<connection-url>` 設定に続く行で、`user-name` および `password` 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。
3. ファイルを保存して閉じます。

4.6.1.3 login-config.xml ファイルの編集

1. [appserver root]/server/lc_mysql/conf/login-config.xml ファイルをテキストエディタで開き、`<policy>` エレメント内の次のコードを変更します。

```
<application-policy name="MySqlDbRealm">
  <authentication>
    <login-module
      code="org.jboss.resource.security.ConfiguredIdentityLoginModule" flag
      = "required">
      <module-option name="principal">adobe</module-option>
      <module-option name="userName">adobe</module-option>
      <module-option name="password">adobe</module-option>
      <module-option
        name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
        name=Default DS </module-option>
    </login-module>
  </authentication>
</application-policy>
```
2. 太字のテキストをデータベースに固有の値に置き換えて、アプリケーションサーバーがデータベースにアクセスできるようにします。
3. ファイルを保存して閉じます。
4. JBoss を起動します。

4.6.2 アドビの事前設定 JBoss 4.2.1 用 Oracle の設定

JBoss から LiveCycle ES2 データを格納している Oracle データベースへの接続を有効にするには、データソースファイルを作成して、LiveCycle ES2 のデプロイ先 JBoss のインスタンスにデプロイする必要があります。

注意： <http://community.jboss.org/wiki/EncryptingDataSourcePasswords> で説明されているいずれかの方法を使用して、データソースファイル (adobe-ds.xml および oracle-ds.xml) および login-config.xml ファイルのパスワードを暗号化します。
http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clear.html の手順を使用することもできます。

4.6.2.1 adobe-ds.xml ファイルの編集

1. [appserver root]/server/lc_oracle/deploy/adobe-ds.xml ファイルをテキストエディタで開き、次の行を見つけてください。

```
<connection-url>jdbc:oracle:thin:@localhost:1521:adobe</connection-url>  
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>  
<user-name>adobe</user-name>  
<password>adobe</password>
```

2. 上述の行の以下のテキストをデータベースに固有の値に置き換えます。

- **localhost** : データベースをホストするコンピュータの名前、IP アドレス、または完全修飾パス。デフォルトは localhost です。
- **1521** : データベースへのアクセスに使用するポート。デフォルトのポートは 1521 です。
- **adobe** : LiveCycle ES2 データを格納しているデータベースの SID。デフォルト値 adobe をデータベースの SID に置き換えます。

3. <connection-url> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

4. (Oracle RAC でのみ) 手順 1 で示されている接続 URL を次の接続 URL に置き換えます。

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS=(  
PROTOCOL=TCP) (HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP)  
(HOST=yourhost2) (PORT=1521)) (LOAD_BALANCE=on) (FAILOVER=on))  
(CONNECT_DATA=(SERVER=dedicated) (SERVICE_NAME=service.yourcompany.com)  
(FAILOVER_MODE=(TYPE=session) (METHOD=basic) (RETRIES=10) (DELAY=3))))
```

注意： このエントリが adobe-ds.xml ファイル内に 1 行で表示されることを確認してください。

5. (Oracle RAC でのみ) 手順 4 で示されている接続 URL に含まれる次のテキストをデータベースに固有の値に置き換えます。

- **yourhost1** : データベースをホストするクラスタ内の最初のノードの名前、IP アドレス、または完全修飾ドメイン名。
- **yourhost2** : データベースをホストするクラスタ内の 2 番目のノードの名前、IP アドレスまたは完全修飾ドメイン名。

注意： データベースをホストするクラスタは任意の数のノードで構成することができます。yourhost1 および yourhost2 は、クラスタが 2 つのノードで構成される場合の例です。

- **service.yourcompany.com** : Oracle RAC データベースのサービス名。

6. データソース接続の最小値と最大値が次のように設定されていることを確認します。

- IDP_DS の場合：

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>30</max-pool-size>
```
- EDC_DS の場合：

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```

注意：また、`<no-tx-datasource>` タグを囲むコメント処理を削除し、BAM データベースのホスト名、ユーザー名およびパスワードの値を指定する必要があります。

注意：LiveCycle ES2 サーバーで処理する負荷が大きい場合は、JDBC 接続の最大数を増やして、すべてのジョブが確実に処理されるようにします。そのような場合は、IDP_DS と EDC_DS の両方で `<max-pool-size>` を 50 以上に増やします。

7. ファイルを保存します。

4.6.2.2 oracle-ds.xml ファイルの編集

Oracle データベースと共に LiveCycle ES2 を実行する場合は、Oracle を JBoss のデフォルトのデータソースに設定する必要があります この手順では、Oracle JDBC ドライバが `[appserver root]/server/lc_oracle/lib` ディレクトリにインストールされていることを前提としています。

1. `[appserver root]/server/lc_oracle/deploy/oracle-ds.xml` ファイルをテキストエディタで開き、

```
<local-tx-datasource> エlement を Oracle の接続設定の内容に変更します。  
<jndi-name>DefaultDS</jndi-name>  
<connection-url>jdbc:oracle:thin@localhost:1521:adobe</connection-url>  
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>  
<user-name>adobe</user-name>  
<password>adobe</password>
```

2. 上述の行の太字のテキストをデータベースに固有の次の値に置き換えます。

localhost : この値を Oracle サーバーのホスト名に置き換えます。

1521 : Oracle でデフォルトのポートを使用していない場合は、この値を適切なポート番号に変更します。

adobe : この値をデータベースの SID に置き換えます。

デフォルトのユーザー名とパスワードを、データベースへのアクセス時にアプリケーションサーバーが認証情報として使用するユーザー名とパスワードに置き換えます。

3. `<connection-url>` 設定に続く行で、`user-name` および `password` 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

4. (Oracle RAC でのみ) 手順 1 で示されている接続設定を次の接続 URL に置き換えます。

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS=(  
  (PROTOCOL=TCP) (HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP)  
  (HOST=yourhost2) (PORT=1521)) (LOAD_BALANCE=on) (FAILOVER=on))  
  (CONNECT_DATA=(SERVER=dedicated) (SERVICE_NAME=service.yourcompany.com)  
  (FAILOVER_MODE=(TYPE=session) (METHOD=basic) (RETRIES=10) (DELAY=3))))
```

注意：このエントリが `oracle-ds.xml` ファイル内に 1 行で表示されることを確認してください。

- (Oracle RAC でのみ) 手順 4 で示されている接続 URL に含まれる次のテキストをデータベースに固有の値に置き換えます。
 - yourhost1** : データベースをホストするクラスタ内の最初のノードの名前、IP アドレス、または完全修飾ドメイン名。
 - yourhost2** : データベースをホストするクラスタ内の 2 番目のノードの名前、IP アドレス、または完全修飾ドメイン名。
注意 : データベースをホストするクラスタは任意の数のノードで構成することができます。**yourhost1** および **yourhost2** は、クラスタが 2 つのノードで構成される場合の例です。
 - service.yourcompany.com** : Oracle RAC データベースのサービス名。
- ファイルを保存して閉じます。

4.6.2.3 login-config.xml ファイルの編集

- [appserver root]/server/lc_oracle/conf/login-config.xml ファイルをテキストエディタで開き、<policy> 要素内の次のコードを変更します。

```
<application-policy name="OracleDbRealm">
  <authentication>
    <login-module
      code="org.jboss.resource.security.ConfiguredIdentityLoginModule" flag
      = "required">
      <module-option name="principal">adobe</module-option>
      <module-option name="userName">adobe</module-option>
      <module-option name="password">adobe</module-option>
      <module-option
        name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
        name=DefaultDS</module-option>
    </login-module>
  </authentication>
</application-policy>
```
- 太字のテキストをデータベースに固有の値に置き換えて、アプリケーションサーバーがデータベースにアクセスできるようにします。
注意 : Oracle RAC の場合、**adobe** をデータベースに固有のサービス名に置き換えます。
- ファイルを保存して閉じます。
- JBoss を起動します。

4.6.3 アドビの事前設定 JBoss 4.2.1 用 SQL Server の設定

JBoss から LiveCycle ES2 データを格納している SQL Server データベースへの接続を有効にするには、SQL Server のデータソースファイルを作成して、LiveCycle ES2 のデプロイ先 JBoss のインスタンスにデプロイする必要があります (例えば、[appserver root]/server/lc_sqlserver/deploy)。

注意：以下の操作に進む前に、JBoss が実行されていないことを確認してください。

注意：<http://community.jboss.org/wiki/EncryptingDataSourcePasswords> で説明されているいずれかの方法を使用して、データソースファイル (adobe-ds.xml および mssql-ds.xml) および login-config.xml ファイルのパスワードを暗号化します。
http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clear.html の手順を使用することもできます。

4.6.3.1 adobe-ds.xml ファイルの編集

1. [appserver root]/server/lc_sqlserver/deploy/adobe-ds.xml ファイルをテキストエディタで開き、次の行を見つけてください。

```
<connection-url>jdbc:sqlserver://localhost:1433;DatabaseName=adobe</connection-url>  
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>  
<user-name>adobe</user-name>  
<password>adobe</password>
```

2. 上述の行の以下のテキストをデータベースに固有の値に置き換えます。

- **localhost** : データベースをホストするコンピュータの名前、IP アドレス、または完全修飾パス。デフォルトは localhost です。
- **1433** : データベースへのアクセスに使用するポート。デフォルトのポートは 1433 です。
- **adobe** : LiveCycle ES2 データを格納しているデータベースの名前。デフォルト値の adobe を、独自のデータベース名に更新する必要があります。
- **username** および **password** : アプリケーションサーバーがデータベースへのアクセスに使用するユーザー名およびパスワード。デフォルト値 adobe および adobe は、データベースの秘密鍵証明書で更新する必要があります。

3. <driver-class> が次のように設定されていることを確認します。

```
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
```

4. <connection-url> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

5. データソース接続の最小値と最大値が次のように設定されていることを確認します。

- IDP_DS の場合 :
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
- EDC_DS の場合 :
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>

注意： LiveCycle ES2 サーバーで処理する負荷が大きい場合は、JDBC 接続の最大数を増やして、すべてのジョブが確実に処理されるようにします。そのような場合は、IDP_DS と EDC_DS の両方で <max-pool-size> を 50 以上に増やします。

6. ファイルを adobe-ds.xml という名前で [appserver root]/server/lc_sqlserver/deploy ディレクトリに保存します。

4.6.3.2 mssql-ds.xml ファイルの編集

MS SQL Server データベースと共に LiveCycle ES2 を実行する場合は、SQL Server を JBoss のデフォルトのデータソースに設定する必要があります この手順では、SQL Server JDBC ドライバが [appserver root]/server/lc_sqlserver/lib ディレクトリにインストールされていることを前提としています。

1. [appserver root]/server/lc_sqlserver/deploy/mssql-ds.xml ファイルをテキストエディタで開き、<local-tx-datasource> エレメントを SQL Server の接続設定の内容に変更します。

```
<jndi-name>DefaultDS</jndi-name>
<connection-url>jdbc:sqlserver://localhost:1433;DatabaseName=adobe</connection-url>
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```

2. 上述の行の太字のテキストをデータベースに固有の次の値に置き換えます。

localhost : この値を SQL Server のホスト名に置き換えます。

1433 : SQL Server でデフォルトのポートを使用していない場合は、この値を適切なポート番号に変更します。

adobe : この値を SQL Server データベース名に置き換えます。

3. <connection-url> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

注意： SQL Server の JDBC ドライバでは数値の IPv6 アドレスは解析されません。IPv6 環境に SQL Server をデプロイする場合、次の形式でサーバー名を指定する必要があります。

```
<connection-url>jdbc:sqlserver://;serverName=<Numeric IPv6 address of the database host>;:1433;databaseName=<LiveCycleES2 database></connection-url>
```

注意： この形式でサーバーの IPv6 アドレスを指定する場合、キーワードである ;serverName の文字列を、実際のサーバー名に置き換える必要があります。

4. ファイルを保存して閉じます。

4.6.3.3 login-config.xml ファイルの編集

1. [appserver root]/server/lc_sqlserver/conf/login-config.xml ファイルをテキストエディタで開き、<policy> エレメント内の次のコードを変更します。

```
<application-policy name="MSSQLDbRealm">
  <authentication>
    <login-module
      code="org.jboss.resource.security.ConfiguredIdentityLoginModule" flag="required">
      <module-option name="principal">adobe</module-option>
      <module-option name="userName">adobe</module-option>
```

```
<module-option name="password">adobe</module-option>
<module-option
name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
name=DefaultDS</module-option>
</login-module>
</authentication>
</application-policy>
```

2. 太字のテキストをデータベースに固有の値に置き換えて、アプリケーションサーバーがデータベースにアクセスできるようにします。
3. ファイルを保存して閉じます。
4. JBoss を起動します。

4.6.3.4 Windows ログインの LiveCycle ES2 データベースユーザーへのマップ

1. Microsoft SQL Server Management Studio を使用して、LiveCycle ES2 データベースのホストであるデータベースサーバーに接続します。
2. 認証モードを「Windows 認証」に設定します。
3. セキュリティ／ログインで、Windows ドメインユーザーの新しいアカウントを作成して、「Windows 認証」を選択します。
4. ログイン - 新規作成画面で「ユーザーマッピング」をクリックして、新しいユーザーのデータベースおよびデフォルトスキーマを設定します。
5. データベースロールとして「db_owner」を選択し、「OK」をクリックします。

作成したユーザーを確認するには、ツリーで LiveCycle ES2 データベースを展開して、Security／Users を開きます。新しいユーザーがユーザーのリストに表示されます。

4.6.3.5 Windows 上での統合セキュリティの設定

注意： JBoss サービスは、[「Windows ログインの LiveCycle ES2 データベースユーザーへのマップ」\(79 ページ\)](#) で設定した Windows ユーザーとして実行する必要があります。JBoss がサービスとして実行されていない場合は、そのユーザーとして Windows にログインし、コマンドプロンプトから JBoss を起動する必要があります。

1. 次の例に示すように、[appserver root]¥server¥lc_sqlserver¥deploy にある adobe-ds.xml ファイルを変更し、integratedSecurity=true を接続 URL に追加します。

```
jdbc:sqlserver://<serverhost>:<port>;databaseName=<dbname>;integratedSecurity=true
```

2. 上述の例の太字の値を、データベースサーバーに適した値に置き換えます。
3. JBoss を実行しているコンピュータの Windows システムパス (C:¥Windows) に sqljdbc_auth.dll ファイルを追加します。sqljdbc_auth.dll ファイルは、Microsoft SQL JDBC 1.2 ドライバのインストール先 (32 ビットオペレーティングシステムの場合のデフォルトは <InstallDir>/sqljdbc_1.2/enu/auth/x86、64 ビットオペレーティングシステムの場合のデフォルトは <InstallDir>/sqljdbc_1.2/enu/auth/x64) と同じ場所にあります。

注意： Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 のどちらのサーバーにも SQL Server JDBC ドライバ 1.2 を使用します。

4. JBoss for Adobe LiveCycle サービスのプロパティを開き、「ログオン」タブをクリックします。
5. 「アカウント」を選択し、[「Windows ログインの LiveCycle ES2 データベースユーザーへのマップ」\(79 ページ\)](#) でマップしたユーザーアカウントの値を入力します。コマンドラインから JBoss を実行する場合は、このユーザーとしてログインする必要があります。

4.7 次の手順

『[LiveCycle ES2 のインストールおよびデプロイ \(JBoss 版\)](#)』に記載されている手順に従って、LiveCycle ES2 をインストールします。

5 WebLogic Server の設定

この章では、LiveCycle ES2 のインストールをホストする Oracle WebLogic Server のインストールと設定の方法について説明します。

ここでは、次のような規則を使用します。

名前	WebLogic 10g のデフォルト値	WebLogic 11g のデフォルト値
[appserver root]	(Windows) C:\%bea%\wlserver_10.3 (Linux および Solaris) /opt/boa/wlserver_10.3	(Windows) C:\%Oracle%\Middleware%\wlserver_10.3 (Linux および Solaris) /opt/Oracle/Middleware/wlserver_10.3
WL_HOME	(Windows) C:\%bea (Linux および Solaris) /opt/boa	(Windows) C:\%Oracle%\Middleware (Linux および Solaris) /opt/Oracle/Middleware
[appserverdomain] デフォルトのドメインは base_domain です。	(Windows) C:\%bea%\user_projects\domains\ base_domain (Linux および Solaris) /opt/boa/user_projects/domains/ base_domain	(Windows) C:\%Oracle%\Middleware%\user_projects\ domains\base_domain (Linux および Solaris) /opt/Oracle/Middleware/user_projects/ domains/base_domain

5.1 概要

次のタスクを実行し、Oracle WebLogic Server 10g 環境をインストールおよび設定します。

1. WebLogic Server を取得し、製造元から提供されるマニュアルに従ってインストールします ([「WebLogic Server のインストール」 \(82 ページ\)](#) を参照)。
2. WebLogic Server Administration Console にアクセスします。
3. 管理対象サーバーを作成します ([「新しい WebLogic 管理対象サーバーの作成」 \(86 ページ\)](#) を参照)。
4. アプリケーションサーバーを設定します ([「WebLogic Server の設定に関する問題」 \(88 ページ\)](#) を参照)。

5.2 WebLogic 用の LiveCycle ES2 の要件

WebLogic の起動方法には、主に次の 2 つがあります。

- ノードマネージャを使用して、管理サーバーから設定する (推奨)
- StartManagedWebLogic スクリプトの設定で、直接管理対象サーバーを起動する

LiveCycle Configuration Manager の自動設定機能を使用する場合 (推奨)、ノードマネージャを使用して管理サーバーにより設定する必要があります。

管理対象サーバーを直接起動する場合も、LiveCycle Configuration Manager の自動設定機能は使用できません。ただし、LiveCycle Configuration Manager の変更については StartManagedWebLogic スクリプトを手動で入力する必要があります。適切な手順については、『[LiveCycle ES2 のインストールおよびデプロイ \(WebLogic 版\)](#)』を参照してください。

5.3 WebLogic Server のインストール

使用しているオペレーティングシステム版の Oracle WebLogic Server を取得し、付属のマニュアルに従ってインストールします (<http://edocs.bea.com/wls/docs103/getstart/overview.html> を参照)。

注意： WebLogic Server のインストールと実行には、コンピュータの管理者の権限を持つユーザーアカウントを使用する必要があります。

5.3.1 Solaris (64 ビット Java) での WebLogic 10g R3 の設定

Solaris で 64 ビット Java を使用するように WebLogic をインストールする際には、追加の手順を実行する必要があります。

1. -d64 フラグを使用して WebLogic インストーラを実行します (詳しくは、WebLogic のドキュメントを参照してください)。
2. 新たに起動したコマンドプロンプトで [appserver_root]/server/bin ディレクトリに移動します。
3. `export JAVA_OPTIONS=-d64` と入力します。
4. スクリプトを使用してノードマネージャを起動します。 `startNodeManager.sh` と入力します。

注意： 64 ビット Java を使用して WebLogic Administration Server を起動する必要があります。
`startWeblogic.sh` スクリプトを使用して WebLogic Administration Server を起動する前に、環境内の `JAVA_OPTIONS=-d64` を設定してください。

5.4 JRockit Java JDK のインストール

ここでは、JRockit Java 6 JDK (WebLogic 10g R3 および 11g R1 用) をインストールする方法について説明します。

▶ (Windows または Linux のみ) JDK をインストールするには：

- 使用しているオペレーティングシステム用の Oracle JRockit™ Java JDK 6.0 update 7 (または 6.0 の以降の更新) を取得し、JRockit の指示に従ってインストールします。

`JAVA_HOME` および `PATH` 環境変数は、LiveCycle ES2 をデプロイするサーバーの JDK を指定する必要があります。

▶ `JAVA_HOME` 環境変数を設定するには (Windows)：

1. スタート/コントロールパネル/システムを選択します。
2. 「詳細設定」タブをクリックして、「環境変数」をクリックします。
3. 「システム環境変数」領域で、「新規」をクリックします。

4. 変数名として JAVA_HOME と入力し、Java SDK をインストールしたディレクトリを入力します。このディレクトリは、WebLogic によって Java SDK がインストールされた場所であり、/bin サブディレクトリが含まれています。例えば、次のテキストを入力します。

```
C:\Program Files\JRockit Real Time\jrrt-3.0.0-1.6.0
```

注意： OpenOffice.org ファイルの変換に PDF Generator ES2 を使用している場合、Sun JDK がインストールされているディレクトリを示すように JAVA_HOME_32 環境変数を設定する必要があります。
<http://www.java.sun.com> から Sun JDK をダウンロードします。

▶ **PATH 環境変数を設定するには (Windows)：**

1. スタート/コントロールパネル/システムを選択します。
2. 「詳細設定」タブをクリックして、「環境変数」をクリックします。
3. 「システム環境変数」領域で、PATH 変数を選択して、「編集」をクリックします。
4. 変数値の先頭に次のテキストを追加します。

```
%JAVA_HOME%\bin;
```

注意： PATH 環境変数の最後のエントリの末尾にスラッシュ (/) が付いていないことを確認します。スラッシュがあると、WebLogic サーバーインスタンスは起動時に失敗します。スラッシュが付いている場合は削除し、変更内容を保存します。

▶ **JAVA_HOME 環境変数を設定するには (Linux および Solaris)：**

- 次の例に示すように、Borne シェルおよび Bash シェルの JAVA_HOME 変数を設定します。

```
JAVA_HOME=/opt/jrockit 1.6.0.12  
export JAVA_HOME
```

注意： 指定するパスは、指定したインストールディレクトリと、インストール先のオペレーティングシステムによって変わります。

注意： JAVA_HOME_32 環境変数を設定する必要があるのは、LiveCycle PDF Generator ES2 を使用して WebLogic Server 10g 環境で Open Office ファイルを変換する場合です。

▶ **PATH 環境変数を設定するには (Linux および Solaris)：**

- 次の例に示すように、Borne シェルおよび Bash シェルの PATH 変数を設定します。

```
PATH=$JAVA_HOME/bin:$PATH  
export PATH
```

▶ **(Windows、Linux または Solaris) JAVA_HOME 環境変数の設定を検証するには：**

(オプション) コマンドプロンプトを開いて、次のコマンドを実行します。

```
java -version
```

「Java version 1.6.0_14」(または 1.6.0 以降のバージョン) で始まる応答が返されます。

5.5 WebLogic Server ドメインの作成

WebLogic Server にアプリケーションをデプロイするには、WebLogic Server ドメインが必要です。「ドメイン」は、WebLogic Server の基本的な管理単位です。WebLogic Server ドメインの基本的な 2 種類は次のとおりです。

管理対象サーバーを含むドメイン (推奨)：通常、実稼働環境は管理サーバーと 1 つ以上の管理対象サーバーで構成されます。管理サーバーでは管理操作が実行されます。アプリケーションとリソースは個々の管理対象サーバーにデプロイされます。

スタンドアロンサーバードメイン：この種類のドメインは、単一のサーバーインスタンスが管理サーバーと管理対象サーバーの両方の機能を果たす開発環境やテスト環境に使用できます。これによって管理が容易になり、開発環境で WebLogic のホットデプロイ機能を利用できるようになります (非推奨)。

注意： Oracle は、実稼働環境ではアプリケーションをドメイン内の管理対象サーバーのみにデプロイし、管理サーバーは管理タスク専用とすることを推奨しています。

▶ 管理対象サーバーが含まれる WebLogic 10g ドメインを作成するには

1. コマンドプロンプトで、[appserver root]/common/bin ディレクトリに移動し、次のコマンドを入力して WebLogic Configuration Wizard を起動します。
 - (Windows) `config.cmd`
 - (Linux および Solaris) `./config.sh`
2. 「Welcome」画面で、「Create a new WebLogic domain」を選択して「Next」をクリックします。
3. 「Select Domain Source」画面で、「Generate a domain configured automatically to support the following products」を選択し、「Next」をクリックします。

注意： デフォルトでは、「WebLogic Server (Required)」が選択されています。「Workshop for WebLogic Platform」が選択解除されていることを確認します。

4. 「Configure Administrator Username and Password」画面に、WebLogic ユーザー名とパスワードを入力し、パスワードを再入力して確認し、「Next」をクリックします。LiveCycle Configuration Manager で入力を求められるため、このユーザー名とパスワードを記録します。
5. WebLogic Domain Startup Mode パネルで、「Production Mode」を選択します。

警告： 「Development Mode」の使用はお勧めしません。

6. 右側のウィンドウで、使用しているオペレーティングシステムに応じた JDK を選択し、「Next」をクリックします。
 - (Windows、Linux) **JRockit SDK 1.6** 以降の場所。例えば、`C:\Program Files\JRockit Real Time\jrrt-3.0.0-1.6.0` (Windows) や `/opt/JRockit Real Time/jrrt-3.0.0-1.6.0` (Linux) です。
 - (Solaris) **Sun JDK 1.6** 以降の場所。例えば、`/opt/jdk1.6.0_14`。
7. 「Customize Environment and Services Settings」画面で、「No」が選択されていることを確認し、「Next」をクリックします。
8. 「Create WebLogic Domain」画面で、デフォルト値をそのまま使用するか、必要に応じてドメイン名と場所を入力し、「Create」をクリックします。
9. 「Creating Domain」画面で、設定の作成が完了したら次のタスクを実行します。
 - (Windows) 「Start Admin Server」を選択して「Done」をクリックします。
 - (Linux および Solaris) 「Done」をクリックし、コマンドプロンプトで [appserverdomain] に移動して、`./startWebLogic.sh` と入力して、サーバーを起動します。

10. (Windows、Linux、Solaris) プロンプトが表示されたら、手順 4 で入力した WebLogic のユーザー名とパスワードを入力します。

注意： コマンドプロンプトで管理サーバーが起動します。コマンドプロンプトにユーザー名とパスワードを入力します。

▶ **管理対象サーバーが含まれる WebLogic 11g ドメインを作成するには**

1. コマンドプロンプトで、[appserver root]/common/bin ディレクトリに移動し、次のコマンドを入力して WebLogic Configuration Wizard を起動します。
 - (Windows) `config.cmd`
 - (Linux および Solaris) `./config.sh`
2. 「Welcome」画面で、「Create a new WebLogic domain」を選択して「Next」をクリックします。
3. 「Select Domain Source」画面で、「Generate a domain configured automatically to support the following products」を選択し、「Next」をクリックします。
4. 「Create WebLogic Domain」画面で、デフォルト値をそのまま使用するか、必要に応じてドメイン名と場所を入力し、「Create」をクリックします。
5. 「Configure Administrator Username and Password」画面に、WebLogic ユーザー名とパスワードを入力し、パスワードを再入力して確認し、「Next」をクリックします。LiveCycle Configuration Manager で入力を求められるため、このユーザー名とパスワードを記録します。
6. WebLogic Domain Startup Mode パネルで、「Production Mode」を選択します。

警告： 「Development Mode」の使用はお勧めしません。

7. 右側のウィンドウで、使用しているオペレーティングシステムに応じた JDK を選択し、「Next」をクリックします。
 - (Windows、Linux) **JRockit SDK 1.6** 以降の場所。例えば、`C:\Program Files\JRockit Real Time\jrrt-3.0.0-1.6.0` (Windows) や `/opt/JRockit Real Time/jrrt-3.0.0-1.6.0` (Linux) です。
 - (Solaris) **Sun JDK 1.6** 以降の場所。例えば、`/opt/jdk1.6.0_14`。
8. 「Creating Domain」画面で、設定の作成が完了したら次のタスクを実行します。
 - (Windows) 「Start Admin Server」を選択して「Done」をクリックします。
 - (Linux および Solaris) 「Done」をクリックし、コマンドプロンプトで [appserverdomain] に移動して、`./startWebLogic.sh` と入力して、サーバーを起動します。
9. (Windows、Linux、Solaris) プロンプトが表示されたら、手順 4 で入力した WebLogic のユーザー名とパスワードを入力します。

注意： コマンドプロンプトで管理サーバーが起動します。コマンドプロンプトにユーザー名とパスワードを入力します。

5.6 ノードマネージャの起動

この章の残りの節に進む前に、ノードマネージャを起動する必要があります。

▶ ノードマネージャを起動するには：

- 新しいコマンドプロンプトで、[appserver root]¥server¥bin ディレクトリに移動して、適切なコマンドを入力します。

(Windows) startNodeManager.cmd

(UNIX) ./startNodeManager.sh

注意： Windows で、ノードマネージャを Windows サービスとしてインストールした場合は、ノードマネージャサービスを開始する必要があります。それ以外の場合は、コマンドラインオプションを使用してノードマネージャを起動します。

5.7 新しい WebLogic 管理対象サーバーの作成

LiveCycle ES2 アプリケーションは、ドメイン内の管理対象サーバーにデプロイする必要があります。管理サーバーは管理タスク専用にしてください。サーバードメイン、管理サーバーおよび管理対象サーバーの作成方法について詳しくは、WebLogic の製品マニュアルを参照してください。

注意： 64 ビット Java を使用して WebLogic Administration Server を起動する必要があります。

startWeblogic.sh スクリプトを使用して WebLogic Administration Server を起動する前に、環境内の JAVA_OPTIONS=-d64 を設定してください。

注意： 次の手順では、デフォルトプロパティに必要な変更について説明します。明記されていないプロパティについては、既存の設定を使用してください。画面について詳しくは、WebLogic Server Administration Console の WebLogic のヘルプを参照してください。

▶ 新しい WebLogic 管理対象サーバーを作成するには

1. WebLogic Administration Server がまだ実行されていない場合は、コマンドプロンプトで [WL_HOME]¥user_projects¥domains¥[domain name] ディレクトリに移動して、該当するコマンドを入力します。
 - (Windows) startWebLogic.cmd
 - (Linux および Solaris) ./startWebLogic.sh
2. WebLogic Server Administration Console にアクセスするには、Web ブラウザの URL 行に `http:// [host name] :7001/console` と入力します。
3. この WebLogic 設定の作成時に使用したユーザー名とパスワードを入力して、「Log In」をクリックします。
4. 「Change Center」で、「Lock & Edit」をクリックします。
5. 「Domain Structure」で、Environment / Servers をクリックし、右側のウィンドウで「New」をクリックします。
6. Create New Server ページの「Server Name」ボックスに、管理対象サーバーの名前 (server1 など) を入力します。
7. 「Server Listen Address」ボックスに、コンピュータ名または IP アドレスを入力します。

注意： アプリケーションサーバーを設定する場合と、LiveCycle Configuration Manager の実行時にデータベースを初期化する場合にも同じ値を使用する必要がありますので、この値 (デフォルトは localhost) を控えておいてください。

8. 「Server Listen Port」ボックスに、現在使用中でないポート番号 (8001 など) を入力します。ポート 7001 は既に管理サーバーに使用されています。
9. 「Finish」をクリックし、「Activate Changes」をクリックします。
10. 「Change Center」で、「Lock & Edit」をクリックします。
11. 「Domain Structure」で、Environment / Machines をクリックし、右側のウィンドウで「New」をクリックします。
12. Create New Machine ページの「Name」ボックスにマシン名を入力し、使用しているオペレーティングシステムを「Machine OS」リストから選択し、「OK」をクリックします。
13. Environment / Servers をクリックし、作成した管理対象サーバーの名前をクリックします。
14. 「Machine」リストで、作成したマシンを選択します。
15. リスポートが手順 8 で入力したポート番号と同じであることを確認します。
16. 「保存」をクリックします。

LiveCycle Configuration Manager では WebLogic のメモリの最小値および最大値を設定しないため、WebLogic Server Administration Console を使用して手動で LiveCycle ES2 を設定する必要があります。

17. 「Configuration」タブで、「Server Start」タブをクリックします。
18. 「Arguments」ボックスに、次に示す適切なテキストを入力して、管理対象サーバーのメモリサイズを設定します。
 - (32 ビット) -Xms256m -Xmx1024m -XX:MaxPermSize=256m
 - (64 ビット) -Xms256m -Xmx1792m -XX:MaxPermSize=512m

注意： 64 ビット JVM for Solaris の場合のみ、MaxPermSize パラメータを設定する必要があります。

19. 「Save」をクリックし、「Activate Changes」をクリックします。

▶ **WebLogic 10g 上の管理対象サーバーのメモリ設定を完了するには：**

1. 次のディレクトリに移動します。WL_HOME¥user_projects¥domains¥[appserverdomain]¥bin。
2. テキストエディタで次のファイルを編集します。
 - (Windows) setDomainEnv.cmd
 - (Linux および Solaris) setDomainEnv.sh
3. MEM_ARGS=-Xms256m -Xmx512m という行を探して、MEM_ARGS=-Xms256m -Xmx1024m に変更します。
4. -XX:MaxPermSize=128m という行を探して、32 ビットバージョンでは -XX:MaxPermSize=256m、64 ビットバージョンでは -XX:MaxPermSize=512m に変更します。

注意： 64 ビット JVM for Solaris の場合のみ、MaxPermSize パラメータを設定する必要があります。

5. 変更内容を保存してファイルを閉じます。

▶ WebLogic 11g 上の管理対象サーバーのメモリ設定を完了するには：

1. 次のディレクトリに移動します。WL_HOME¥user_projects¥domains¥[appserverdomain]¥bin。
2. テキストエディタで次のファイルを編集します。
 - (Windows) setDomainEnv.cmd
 - (Linux および Solaris) setDomainEnv.sh
3. WLS_MEM_ARGS_64BIT=-Xms256m -Xmx512m という行を探して、WLS_MEM_ARGS=-Xms256m -Xmx1024m に変更します。
4. -XX:MaxPermSize=128m という行を探して、32 ビットバージョンでは -XX:MaxPermSize=256m、64 ビットバージョンでは -XX:MaxPermSize=512m に変更します。

注意： 64 ビット JVM for Solaris の場合のみ、MaxPermSize パラメータを設定する必要があります。

5. 変更内容を保存してファイルを閉じます。

5.8 WebLogic Server の設定に関する問題

LiveCycle ES2 インストールのインストールでアプリケーションサーバーのパフォーマンスを最適化するには、次の設定変更を行う必要があります。

5.8.1 Web サービス用の WebLogic の設定

LiveCycle ES2 が Web サービスを使用して要求を受け入れるようにするには、サーブレットコンテナの認証を変更する次の手順を完了する必要があります。

▶ サーブレットコンテナの認証を変更するには：

1. WebLogic Administration Server を起動します。
2. コマンドプロンプトを開き、この例のように setWLSEnv を使用して環境を設定します。
 - (Windows) [WL_HOME]¥wlserver_10.3¥server¥bin¥ フォルダから、setWLSEnv.cmd と入力します。
 - (Linux および Solaris) [WL_HOME]/wlserver_10.3/server/bin/ ディレクトリから、setWLSEnv.sh と入力します。
3. 次のコマンドを入力して WebLogic Scripting Tool を起動します。

```
java weblogic.WLST
```

「Exception in thread "main" java.lang.NoClassDefFoundError: weblogic/WLST」
と表示された場合は、次のタスクを実行します。

- コマンドプロンプトで、<WL_Home>/wlserver_10.3/server/bin/setWLSEnv.cmd を使用して環境を設定します。
- コマンドプロンプトで、setWLSEnv.cmd または ./setWLSEnv.sh コマンドから返されたクラスパスをコピーして、次の例に示すように export CLASSPATH を追加します。

```
CLASSPATH=C:¥bea¥patch_wls1030¥profiles¥default¥sys_manifest_classpath¥  
weblogic_patch.jar;C:¥bea¥patch_cie660¥profiles¥default¥sys_manifest_  
classpath¥weblogic_patch.jar;C:¥bea¥JROCKI~1¥lib¥tools.jar;C:¥bea¥  
WLSERV~1.3¥server¥lib¥weblogic_sp.jar;C:¥bea¥WLSERV~1.3¥server¥lib¥  
weblogic.jar;C:¥bea¥modules¥features¥weblogic.server.modules_  
10.3.0.0.jar;C:¥bea¥WLSERV~1.3¥server¥lib¥webservices.jar;C:¥bea¥
```

```
modules¥ORGAPA~1.5/lib/ant-all.jar;C:¥bea¥modules¥NETSFA~1.0_1/lib/  
ant-contrib.jar export CLASSPATH
```

- コマンドプロンプトで、setWLSEnv.cmd または ./setWLSEnv.sh から返されたパスをコピーして、export PATH を追加します。次に例を示します。

```
PATH=C:¥bea¥patch_wls1030¥profiles¥default¥native;C:¥bea¥patch_cie660¥  
profiles¥default¥native;C:¥bea¥WLSERV~1.3¥server¥native¥win¥32;C:¥bea¥  
WLSERV~1.3¥server¥bin;C:¥bea¥modules¥ORGAPA~1.5¥bin;C:¥bea¥JROCKI~1¥jre¥  
bin;C:¥bea¥JROCKI~1¥bin;C:¥Program Files¥Common Files¥NetSarang;c:¥  
Program Files¥Java¥jdk1.5.0_17¥bin;C:¥WINDOWS¥system32;C:¥WINDOWS;C:¥  
WINDOWS¥System32¥Wbem;C:¥Program Files¥Perforce;C:¥bea¥WLSERV~1.3¥  
server¥native¥win¥32¥oci920_8 export PATH
```

4. WLST で以下のコマンドを入力し、サーバーコンテナの認証を更新します。

```
connect ('<WebLogic username>', '<WebLogic password>', '<WebLogic URL>')  
edit ()  
startEdit ()  
cd ('SecurityConfiguration')  
cd ('<domain name>')  
set ('EnforceValidBasicAuthCredentials', 'false')  
activate ()  
exit ()
```

注意： WebLogic URL は、t3://hostname:[port] の形式で指定します。[port] は管理サーバーポート (通常 7001) です。

5.9 WebLogic の停止と再起動

注意： すべての設定変更を完了した後、その変更を有効にするには WebLogic を再起動します。WebLogic 管理対象サーバー、ノードマネージャ、WebLogic 管理サーバーも再起動する必要があります。WebLogic Administration Server は、64 ビット Java を使用して起動する必要があります。startWeblogic.sh スクリプトを使用して WebLogic Administration Server を起動する前に、環境に JAVA_OPTIONS=-d64 が設定されていることを確認してください。

▶ WebLogic 管理対象サーバーの停止：

1. WebLogic Server Administration Console の「Domain Structure」で、ドメイン名をクリックします。
2. 「Control」タブをクリックし、停止するサーバーの横にあるチェックボックスを選択します。
3. 「Shutdown」をクリックして、次のいずれかのオプションを選択します。

When work completes： 選択されたサーバーが正常に停止され、管理対象サーバーは処理中のすべての要求を完了するようサブシステムに通知します。正常に停止すると、WebLogic Server のサブシステムでは、現在進行中の特定のアプリケーションの処理を完了できます。

Force Shutdown Now： 強制的な停止が開始され、管理対象サーバーはサブシステムに対して、実行中の要求を直ちに破棄するよう指示します。

4. WebLogic Server Administration Console のプロンプトで、「Yes」をクリックしてコマンドの実行を確認します。

管理対象サーバーが停止したことを確認するには、「Control」タブの一番下のテーブルを参照します。このテーブルには、すべてのサーバーのリストが表示され、それぞれの現在の状態が示されます。

▶ WebLogic 管理サーバーの停止

1. コマンドプロンプトで、WL_HOME¥user_projects¥domains¥[appserverdomain]¥bin に移動します。
2. 次のコマンドを入力します。
 - (Windows) stopWebLogic.cmd
 - (Linux、UNIX) ./stopWebLogic.sh
3. WebLogic のユーザー名とパスワードを入力します (WebLogic のインストール時にセキュリティを有効にした場合)。

▶ WebLogic 管理サーバーの再起動

1. コマンドプロンプトで、WL_HOME/user_projects/domains/[appserverdomain] に移動します。
2. 次のコマンドを入力します。
 - (Windows) startWebLogic.cmd
 - (Linux、UNIX) ./startWebLogic.sh
3. WebLogic のユーザー名とパスワードを入力します (WebLogic のインストール時にセキュリティを有効にした場合)。

▶ WebLogic Node Manager の再起動

- コマンドプロンプトで、[appserver root]/server/bin ディレクトリに移動して、該当するコマンドを入力します。
 - (Windows) startNodeManager.cmd
 - (Linux、UNIX) ./startNodeManager.sh

▶ WebLogic 管理対象サーバーの再起動

1. WebLogic 管理サーバーが起動したら、WebLogic Server Administration Console にログインします。
2. 「Domain Structure」で、Environment / Servers をクリックし、右側のウィンドウで管理対象サーバーをクリックします。
3. 次の画面で、「Control」タブをクリックし、起動する管理対象サーバーの横にあるチェックボックスを選択します。
4. 「Start」をクリックして、「Yes」をクリックします。

5.10 LiveCycle ES2 データベース接続の設定

データベース接続に必要な設定について詳しくは、『[LiveCycle ES2 のインストールおよびデプロイ \(WebLogic 版\)](#)』を参照してください。

5.11 次の手順

『[LiveCycle ES2 のインストールおよびデプロイ \(WebLogic 版\)](#)』の手順に従って、LiveCycle ES2 をインストールします。

6

WebSphere Application Server の設定

この章では、LiveCycle ES2 のインストールをホストする WebSphere Application Server のインストールと設定の方法について説明します。

ここでは、次のような規則を使用します。

名前	デフォルト値
[appserver root]	(Windows) C:\Program Files\IBM\WebSphere\AppServer (Linux および Solaris) /opt/IBM/WebSphere/AppServer (AIX) /usr/IBM/WebSphere/AppServer または /opt/IBM/WebSphere/AppServer

6.1 WebSphere Application Server のインストール

LiveCycle ES2 製品を実行するには、WebSphere Application Server をインストールする必要があります。LiveCycle ES2 で使用する WebSphere は、WebSphere Base として、またはベースプロファイルが 1 つの WebSphere ND としてインストールできます。どちらの方法が適しているか判断するには、WebSphere のドキュメントを参照してください。

6.1.1 WebSphere への Fix Pack のインストール

(WebSphere 6.1) WebSphere 6.1 のインストール後、LiveCycle ES2 をデプロイする前に WebSphere 6.1.0.21 および SR 8 に更新する必要があります。WebSphere の更新ファイルにアクセスするには、[WebSphere サポート Web サイト](#)を参照してください。

(WebSphere 7.0) WebSphere 7.0 のインストール後、LiveCycle ES2 をデプロイする前に WebSphere 7.0.0.9 および SR 7 に更新する必要があります。WebSphere の更新ファイルにアクセスするには、[WebSphere サポート Web サイト](#)を参照してください。

6.1.2 JAVA_HOME および PATH 環境変数の設定

WebSphere のインストール時に、Java SDK (JDK) がインストールされています。JAVA_HOME および PATH 環境変数には、LiveCycle ES2 をデプロイする JDK を指定できます。

▶ **JAVA_HOME 環境変数を設定するには (Windows) :**

1. スタート/コントロールパネル/システムを選択します。
2. 「詳細設定」タブをクリックして「環境変数」をクリックします。
3. 「システム環境変数」領域で、「新規」ボタンをクリックします。
4. JAVA_HOME を変数名として入力し、WebSphere にバンドルされた JDK のインストール先ディレクトリを値として指定します。[「サードパーティのインフラストラクチャのサポート」\(10 ページ\)](#)を参照してください。例えば、次のパスを入力します。

C:\Program Files\IBM\WebSphere\AppServer\java

▶ **PATH 環境変数を設定するには (Windows) :**

1. スタート/コントロールパネル/システムを選択します。
2. 「詳細設定」タブをクリックして、「環境変数」をクリックします。
3. 「システム環境変数」領域で、Path を選択して「編集」をクリックし、変数値の先頭に次のテキストを追加します。

```
%JAVA_HOME%\bin;
```

▶ **DISPLAY 設定を設定するには (AIX) :**

- Windows マシンを使用して AIX コンソールにアクセスしている場合、コマンドプロンプトで次のテキストを入力します。

```
export DISPLAY= <IP Address of the Windows machine telnet, or cygwin to  
Linux or Solaris machine>:0.0
```

注意 : DISPLAY 設定が正しくない場合、LiveCycle Configuration Manager の起動時に 「No JRE is found...」 というエラーが表示されます。

▶ **JAVA_HOME 環境変数を設定するには (AIX、Linux および Solaris) :**

- 次の例に示すように、Borne シェルおよび Bash シェルの JAVA_HOME 変数を設定します。

```
JAVA_HOME=/opt/IBM/WebSphere/AppServer/java  
export JAVA_HOME
```

注意 : 指定するパスは、指定したインストールディレクトリと、インストール先のオペレーティングシステムによって変わります。

▶ **PATH 環境変数を設定するには (AIX、Linux および Solaris) :**

- 次の例に示すように、Borne シェルおよび Bash シェルの PATH 変数を設定します。

```
PATH=$JAVA_HOME/bin:$PATH  
export PATH
```

▶ **(Windows、AIX、Linux または Solaris) JAVA_HOME 環境変数の設定を検証するには :**

(オプション) コマンドプロンプトを開いて、次のコマンドを実行します。

```
java -version
```

Java のインストールバージョンで始まる応答が返されます。

6.1.3 WebSphere Base の起動と WebSphere Administrative Console へのアクセス

この節のいくつかの手順では、WebSphere Application Server の管理コンソールにアクセスする必要があります。

▶ **WebSphere Base を起動するには**

1. サーバーがまだ実行されていない場合、WebSphere Administrative Console を起動し、[application server home]/bin で、次の適切なコマンドを入力します。server1 は、使用しているサーバーインスタンスの名前に置き換えます。

- (Windows) startServer.bat server1
- (AIX、Linux および Solaris) ./startServer.sh server1

▶ **WebSphere Administrative Console にアクセスするには：**

1. WebSphere Application Server を起動します。
2. Web ブラウザのアドレスバーに適切な URL を入力します。

`http:// [host name] : [port] /ibm/console`

現在使用しているサーバーを管理する場合は、`[host name]` を `localhost` に置き換えます。ポート番号はアプリケーションサーバーによって異なり、管理セキュリティが有効かどうかによっても異なります。WebSphere のデフォルトのポート番号は 9060 です。管理セキュリティが有効な場合、デフォルトの SSL ポート番号は 9043 です。

3. WebSphere 管理セキュリティが有効になっている場合は、表示されるボックスに WebSphere のユーザー ID とパスワードを入力します。
4. 「Log In」をクリックします。

6.1.4 WebSphere ND の起動と WebSphere Administrative Console へのアクセス

1. コマンドプロンプトを開き、`[WebSphere ND root]/profiles/Dmgr01/bin` に移動します。
2. (**Windows**) `startManager.bat` を実行します。
(**AIX、Linux および Solaris**) `startManager.sh` を実行します。
3. `[WebSphere ND root]/profiles/[profile name]/bin` に移動し、`startNode.bat` を実行します。

4. サービスが起動したら、Web ブラウザのアドレスバーに
`http:// [host name] : [port] /ibm/console` という URL を入力してサーバーに接続します。

現在使用しているサーバーを管理する場合は、`[host name]` を `localhost` に置き換えます。ポート番号はアプリケーションサーバーによって異なり、管理セキュリティが有効かどうかによっても異なります。WebSphere のデフォルトのポート番号は 9060 です。管理セキュリティが有効な場合、デフォルトの SSL ポート番号は 9043 です。

5. WebSphere 管理セキュリティが有効になっている場合は、表示されるボックスに WebSphere のユーザー ID とパスワードを入力して、「Log In」をクリックします。
6. WebSphere Administrative Console のナビゲーションツリーで、Servers / Server Types / WebSphere Application Servers をクリックします。
7. リストからサーバーを選択し、「Start」をクリックします。サーバーの起動時にサーバーの状態が変化することに注意してください。

6.1.5 ディレクトリ権限の設定

LiveCycle ES2 アプリケーションは、ファイルを `[appserver root]/installableApps` ディレクトリに抽出する必要があります。このため、そのディレクトリには書き込み権限を付与することが重要です。次の節では、書き込み権限を付与できない場合にファイルの抽出先を変更する方法について説明します。

▶ **ファイルの抽出先を変更するには**

1. WebSphere Administrative Console にログインします。
2. 次のいずれかを実行します。
 - (WebSphere 6.1) Servers / Application servers をクリックし、ご使用のサーバー名 (例えば、「server1」) をクリックします。
 - (WebSphere 7.0) Servers / Server Types / WebSphere Application Servers をクリックし、ご使用のサーバー名 (例えば、「server1」) をクリックします。
3. 「Server Infrastructure」で、Java and Process Management / Process Definition をクリックします。
4. 「Additional Properties」で「Java Virtual Machine」をクリックし、次に「Custom Properties」をクリックします。
5. 「New」をクリックし、adobeidp.RootDirectory という名前のカスタムプロパティを作成します。
6. adobeidp.RootDirectory の値を、アドビのネイティブファイルの抽出先となるパス (例えば、[appserver root]/profiles/[server]/installedApps) に設定します。
7. 「OK」または「Apply」をクリックします。
8. 「Messages」ボックスで「Save directly to master configuration」をクリックし、アプリケーションサーバーを再起動します。

6.2 WebSphere Application Server の準備

ここでは、LiveCycle ES2 のデプロイメント用のアプリケーションサーバーインスタンスを準備および設定する方法について説明します。

6.2.1 必要なアプリケーションサーバーインスタンス

LiveCycle ES2 では、1 つ以上のアプリケーションサーバーインスタンスが必要です。LiveCycle ES2 はアプリケーションサーバーインスタンスの 1 つにデプロイされます。また、LiveCycle ES2 Business Activity Monitoring を使用する場合は、LiveCycle ES2 は別のアプリケーションサーバーインスタンスにインストールする必要があります。

6.2.2 SOAP 要求タイムアウト値の延長

LiveCycle ES2 の SOAP 要求タイムアウト値を変更する必要があります。

▶ **SOAP 要求タイムアウト値を増やすには**

1. [appserver root] ディレクトリに移動し、soap.client.props という名前のすべてのファイルを検索します。この名前のファイルは複数存在する場合があります。例えば、AIX、Linux または Solaris サーバー上には、次のファイルが存在します。
 - [appserver root]/profileTemplates/default/documents/properties/soap.client.props
 - [appserver root]/profiles/<LiveCycle Server Profile Name>/properties/soap.client.props
 - [appserver root]/profiles/<LiveCycle Server Profile Name>/temp/soap.client.props
 - (WebSphere 6.1 のみ) [appserver root]/profileTemplates/cell/default/documents/properties/soap.client.props
 - (WebSphere 6.1 のみ) [appserver root]/cip/profileTemplates/minimal/documents/properties/soap.client.props

2. 各 soap.client.props ファイルをテキストエディタで開き、com.ibm.SOAP.requestTimeout プロパティを探して、値を 180 から 1800 に変更します。
3. 各 soap.client.props ファイルを保存します。
4. WebSphere Administrative Console のナビゲーションツリーで、次のいずれかを実行します。
 - (WebSphere 6.1) Servers / Application Servers をクリックし、右側のウィンドウでサーバー名をクリックします。
 - (WebSphere 7.0) Servers / Server Types / WebSphere Application servers をクリックし、右側のウィンドウでサーバー名をクリックします。
5. 「Server Infrastructure」で、Administration / Administration Services をクリックします。
6. 「Additional Properties」で「JMX Connectors」をクリックし、「SOAPConnector」をクリックします。
7. 次の画面で「Custom properties」をクリックし、「requestTimeout」をクリックします。
8. 必要に応じて、次の画面で「Value」ボックスの 600 を 1800 に変更します。「OK」または「Apply」をクリックします。
9. 「Messages」ボックスの「Save directly to master configuration」をクリックします。

注意： Content Services ES2 をデプロイする場合、WebSphere Application Server と IBM DB2 64 ビットのデータベースの組み合わせには、-Dhibernate.dialect=org.hibernate.dialect.DB2Dialect という特別の設定が必要です。この設定は、同じ DB2 インスタンス上に複数のスキーマがある場合にのみ必要です。

6.2.3 デプロイのヒープサイズの拡大

タイムアウトエラーを回避するために、ejbdeploy.bat/sh スクリプトでヒープサイズを拡大する必要があります。

▶ AIX、Linux または Solaris

1. [appserver root]/deploytool/itp/ ディレクトリに移動して、ejbdeploy.sh を編集用に開きます。
2. (Solaris のみ) SunOS セクションで、EJBDEPLOY_JVM_OPTIONS 属性を探して -XX:PermSize オプションの値を 256m に変更し、-Xverify オプションの値が none であることを確認します。
3. \$JAVA_CMD¥ セクションのヒープサイズを以下の値に変更します。
-Xms256m -Xmx512m
4. ファイルを保存して閉じます。

▶ Windows

1. [appserver root]¥deploytool¥itp に移動して、ejbdeploy.bat ファイルをテキストエディタで開きます。
2. %JAVA_HOME% で始まる行を探して、引数 -Xmx を探します。
3. この引数を -Xmx512M に変更します。
4. ファイルを保存して閉じます。

▶ **MaxPermSize の拡大 (Solaris 上の WebSphere) :**

1. WebSphere Administrative Console にログインします。
2. WebSphere Administrative Console のナビゲーションツリーで、次のいずれかを実行します。
 - (WebSphere 6.1) Servers / Application Servers をクリックし、右側のウィンドウでサーバー名をクリックします。
 - (WebSphere 7.0) Servers / Server Types / WebSphere Application servers をクリックし、右側のウィンドウでサーバー名をクリックします。
3. 「Server Infrastructure」で、Java and Process Management / Process Definition をクリックします。
4. 「Additional Properties」で、「Java Virtual Machine」をクリックします。
5. 「Generic JVM Arguments」に、MaxPermSize パラメータとして `-XX:MaxPermSize=512m` を入力します。
6. 「OK」または「Apply」をクリックします。
7. 「Messages」ボックスで「Save directly to master configuration」をクリックし、アプリケーションサーバーを再起動します。

6.3 LiveCycle ES2 データベース接続の設定

データベース接続に必要な設定について詳しくは、『[LiveCycle ES2 のインストールおよびデプロイ \(WebSphere 版\)](#)』を参照してください。

6.4 次の手順

『[LiveCycle ES2 のインストールおよびデプロイ \(WebSphere 版\)](#)』の手順に従って、LiveCycle ES2 をインストールします。

7

事前設定環境のチェックリスト

アプリケーションサーバーインストールガイドに進む前に、システム設定に関する次の情報を記録済みであることを確認してください。

サーバー設定

事前設定	データを記録	必須かどうか
必要システム構成が満たされているかどうか	<ここにテキストを入力>	必須
JDKのインストール場所（32ビットまたは64ビットOS）	<ここにテキストを入力>	必須
グローバルストレージディレクトリの作成	<ここにテキストを入力>	デフォルトの場所を使用しない場合のみ。

アプリケーションサーバーの設定

事前設定	データを記録	必須かどうか
アプリケーションサーバーのホスト名またはIPアドレス	<ここにテキストを入力>	必須
アプリケーションサーバーの資格情報	<ここにテキストを入力>	必須
アプリケーションサーバーのポート番号	<ここにテキストを入力>	必須
次の項目の説明どおりに事前設定が完了しているか。 「JBoss Application Server の設定」 (68 ページ)、 「WebLogic Server の設定」 (81 ページ) または 「WebSphere Application Server の設定」 (91 ページ) ?	<ここにテキストを入力>	必須

データベースの設定

事前設定	データを記録	必須かどうか
データベースのホスト名またはIPアドレス	<ここにテキストを入力>	必須
データベースアカウントの資格情報	<ここにテキストを入力>	必須
データベースのポート番号	<ここにテキストを入力>	必須
「LiveCycle ES2 データベースの作成」 (54 ページ) の説明どおりに事前設定が完了しているか	<ここにテキストを入力>	必須

LDAP サーバー

事前設定	データを記録	必須かどうか
LDAP サーバーのホスト名または IP アドレス	<ここにテキストを入力>	LDAP を使用してユーザーを認証する場合にのみ必要。
LDAP アカウントの資格情報	<ここにテキストを入力>	LDAP を使用してユーザーを認証する場合にのみ必要。

PDF Generator ES / PDF 3D Generator ES の設定 (Windows)

事前設定	データを記録	必須かどうか
Microsoft Office のインストール	<ここにテキストを入力>	必須。Acrobat および Microsoft Office をインストールするには、管理者権限を持つユーザーアカウントを使用する必要があります。
Acrobat のインストール	<ここにテキストを入力>	必須。Acrobat および Microsoft Office をインストールするには、管理者権限を持つユーザーアカウントを使用する必要があります。
その他のネイティブアプリケーションソフトウェアのインストール	<ここにテキストを入力>	必須。
Windows 環境変数の設定	<ここにテキストを入力>	必須。

A 付録 - JBoss の手動設定

この付録では、インターネットからダウンロードできるデフォルトの JBoss Application Server で必要な設定について説明します。このオプションは、詳細インストールの場合にのみ検討してください。通常、JBoss の高度な知識が必要です。

この章のトピックの内容は JBoss バージョン 4.2.0 および 4.2.1 が対象です。バージョン固有の手順にはその旨が示されています。

LiveCycle ES2 は、Windows Server 2003 および 2008 (Enterprise Edition または Standard Edition)、Red Hat Linux ES/AS 5.0、SUSE Linux ES 10 のプラットフォーム、および Solaris 10 の JBoss 上で動作します ([「サードパーティのインフラストラクチャのサポート」\(10 ページ\)](#) を参照)。

注意： JBoss 版での LiveCycle ES2 の自動インストールを使用する場合は、このドキュメントを読む必要はありません。自動インストールの環境を準備するために必要なすべての手順は、『[LiveCycle ES2 の自動インストールおよびデプロイ](#)』に記載されています。

この章では、次の表記を使用します。

名前	デフォルト値
[appserver root]	(JBoss 4.2.0 または 4.2.1 (Windows)) : C:\jboss (JBoss 4.2.0 または 4.2.1 (Linux および Solaris)) : /opt/jboss (JBoss EAP 4.3 (Windows)) : C:\jboss-eap-4.3\jboss-as (JBoss EAP 4.3 (Linux および Solaris)) : /opt/jboss-eap-4.3/jboss-as

A.1 JBoss 用 JDK のインストール

使用している JBoss のバージョンに従って、www.java.sun.com から次の Sun JDK バージョンのいずれかをダウンロードし、インストールします。

- (JBoss 4.2.0) Sun JDK 5.0 Update 11 (または 5.0 の以降の更新)
- (全プラットフォームの JBoss 4.2.1 または JBoss EAP 4.3) Sun JDK 6.0 Update 14 (または 6.0 の以降の更新)

JDK がインストールされている場所を示す `JAVA_HOME` 環境変数を作成または設定します。

A.1.1 JAVA_HOME 環境変数の設定 (Windows)

1. スタート/コントロールパネル/システムの順に選択します。
2. 「詳細設定」タブをクリックします。
3. 「環境変数」をクリックし、「システム環境変数」で、「新規」をクリックします。

4. 新しいシステム変数ダイアログボックスで、変数名として JAVA_HOME を入力し、JDK をインストールしたディレクトリを入力します。このディレクトリは、/bin サブディレクトリを含むディレクトリです。例えば、次のパスを入力します。

```
C:¥Program Files¥Java¥jdk1.6.0_14
```

A.1.2 PATH 環境変数の設定 (Windows)

5. スタート/コントロールパネル/システムを選択します。
6. 「詳細設定」タブをクリックして、「環境変数」をクリックします。
7. 「システム環境変数」領域で、PATH 変数を選択して、「編集」をクリックします。
8. 変数値の先頭に次のテキストを追加します。

```
%JAVA_HOME%¥bin;
```

注意： PATH 環境変数の最後のエントリの末尾にスラッシュ (/) が付いていないことを確認します。スラッシュがあると、WebLogic サーバーインスタンスは起動時に失敗します。スラッシュが付いている場合は削除し、変更内容を保存します。

A.1.3 JAVA_HOME 環境変数の設定 (Linux および Solaris)

- 次の例に示すように、Borne シェルおよび Bash シェルの JAVA_HOME 変数を設定します。

```
JAVA_HOME=/usr/java  
export JAVA_HOME
```

A.1.4 PATH 環境変数の設定 (Linux および Solaris)

- 次の例に示すように、Borne および Bash の PATH 変数を設定します。

```
PATH=$JAVA_HOME/bin:$PATH  
export PATH
```

A.1.5 JAVA_Home 環境変数の設定の検証 (Windows、Linux、または Solaris)

(オプション) コマンドプロンプトを開いて、次のコマンドを実行します。

```
java -version
```

「Java version 1.6.0_14」(または 1.6.0 以降のバージョン) で始まる応答が返されます。

A.2 JBoss の手動インストール

JBoss を手動で設定して LiveCycle ES2 を実行する場合は、JBoss Application Server をダウンロードして展開します。

A.2.1 JBoss のダウンロード

1. [JBoss Application Server Download](#) サイトを参照してください。
2. 4.2.0 または 4.2.1 のエントリを見つけ、その横にある「Download」リンクをクリックします。
3. ダウンロードページのファイルのリストで、オペレーティングシステムに対応するファイルをクリックします。
 - (Windows) jboss-4.2.0.GA.zip または jboss-4.2.1.GA.zip
 - (Linux) jboss-4.2.0.GA-src.tar.gz または jboss-4.2.1.GA-src.tar.gz
4. ダウンロードが完了したら、サーバー上のディレクトリを選択してファイルを抽出します。

A.3 JBoss の起動と停止

この付録で説明する一部の手順では、製品をデプロイする JBoss のインスタンスを停止して起動する必要があります。

注意：これらの手順は、アドビの事前設定 JBoss および手動でインストールした JBoss ソフトウェアの両方に適用されます。

A.3.1 JBoss の起動

1. コマンドプロンプトで [appserver root]/bin に移動します。
2. 次のコマンドを入力して、アプリケーションサーバーを起動します。
 - (Windows) `run.bat -c [config] -b [server_IP_Address]`
 - (Linux および Solaris) `./run.sh -c [config] -b [server_IP_Address]`

[config] はデータベースに必要な設定、[server_IP_Address] はサーバーの実際の IP アドレスに置き換えます。例えば、次のように指定します。

 - アドビの事前設定 JBoss を使用する場合：
 - (Windows) `run.bat -c lc <database name> -b [server_IP_Address]`
 - (Linux および Solaris) `./run.sh -c lc <database name> -b [server_IP_Address]`
 - 手動で設定した JBoss を使用する場合：
 - (Windows) `run.bat -c all -b [server_IP_Address]`
 - (Linux および Solaris) `./run.sh -c all -b [server_IP_Address]`

A.3.2 JBoss の停止

1. コマンドプロンプトで [appserver root]/bin に移動します。
2. 次のコマンドを入力して、アプリケーションサーバーを停止します。
 - (Windows) `shutdown.bat -s <server name>:<jndi-port>`
 - (Linux および Solaris) `./shutdown.sh -s <server name>:<jndi-port>`

A.4 JBoss 設定の変更

JBoss Application Server は様々な XML 設定ファイルを使用して設定されます。これらの設定ファイルのいずれかを編集するためには、事前に JBoss を停止する必要があります。JBoss の実行中にこれらのファイルを変更すると、JBoss で障害が発生する可能性があります。JBoss には、.property ファイルという形式の設定ファイルがいくつかあります。.property ファイルを Windows 環境で編集する場合は、.property ファイルが常に Linux または Solaris 上の UNIX テキストファイルとして保存されている必要があります。

A.4.1 JMS およびクラスタ化の設定ファイルの削除

LiveCycle ES2 では JMS 設定を使用しないので、ダウンロードした JBoss に含まれる以下のファイルとディレクトリを削除できます。以下に示すディレクトリとその内容の全体を削除してください。

注意： LiveCycle Foundation JMS サービスを有効にする必要がある場合や、JMS サービスに依存する他のサービスやアプリケーションをデプロイする必要がある場合は、この手順をスキップしてください。

- [appserver root]¥server¥all¥deploy-hasingleton
- (JBoss 4.2.0 および 4.2.1 のみ) [appserver root]¥server¥all¥deploy¥jms
- [appserver root]¥server¥all¥deploy¥deploy-hasingleton-service.xml

JBoss EAP 4.3 を使用する場合は、次の追加手順を実行してください。

- [appserver root]¥server¥all¥deploy¥jboss-messaging.sar フォルダを削除します。
- [appserver root]¥server¥all¥conf¥jbossjta-properties.xml を開いて、次のプロパティをコメントアウトします。

```
com.arjuna.ats.jta.recovery.XAResourceRecovery.JBMESSAGING1
```

シングルサーバーに JBoss をインストールする場合は、クラスタの設定に関する以下のファイルとディレクトリを削除します。

- [appserver root]¥server¥all¥deploy¥deploy.last (JBoss 4.2.0 および JBoss EAP 4.3 のみ)
- [appserver root]¥server¥all¥deploy¥jboss-web-cluster.sar
- [appserver root]¥server¥all¥farm
- [appserver root]¥server¥all¥deploy¥cluster-service.xml
- [appserver root]¥server¥all¥deploy¥ejb3-clustered-sfsbcache-service.xml
- [appserver root]¥server¥all¥deploy¥ejb3-entity-cache-service.xml

A.4.2 クラスタ設定の無効化

クラスタ化に固有の設定ファイルを削除することに加えて、jboss-service.xml を編集してクラスタ化を無効にする必要があります。

1. [appserver root]¥server¥all¥deploy¥jboss-web.deployer¥META-INF¥jboss-service.xml ファイルをエディタで開きます。

2. 以下の行を見つけてコメントアウトします。

```
<depends>jboss.cache:service=TomcatClusteringCache</depends>  
<depends>jboss.service=TransactionManager</depends>
```

3. ファイルを保存します。

A.4.3 JBoss 設定の変更

JBoss 設定を変更して LiveCycle ES2 用に JBoss をカスタマイズするには、次の手順を実行します。

- jacorb.properties ファイルを更新します。
- quartz.jar ライブラリおよび commons-collections.jar ライブラリを更新します。
- EAR および WAR ファイルのクラスローディングの分離を変更します。
- JavaServer Faces (JSF) ライブラリを無効にします (JBoss 4.2.0 のみ)。
- クラスタサービスを無効にします (シングルノードへのインストールの場合のみ)。
- run.bat ファイルを変更します (Windows の場合)。
- run.conf を変更します。
- run.sh ファイルを変更します (Linux および Solaris の場合、64 ビット版のみ)。
- log4j.xml を変更します。
- JMX コンソールおよび Web コンソールを削除します。
- 認証されたユーザーが不要になるように、jmx-invoker-service.xml ファイルを変更します。

▶ jacorb.properties ファイルの更新

1. [appserver root]/server/all/conf/jacorb.properties ファイルをテキストエディタで開きます。
2. jacorb.poa.thread_pool_max という設定を見つけて、値を 16 に変更します。

▶ quartz.jar ライブラリの更新 (JBoss 4.2.0 のみ)

1. [appserver root]/jboss_4.2.0/patches ディレクトリに移動します。
2. quartz.jar ファイルを、このディレクトリから [appserver root]/server/all/lib ディレクトリにコピーします。

▶ EAR ファイルのクラスローディングの分離の変更

1. [appserver root]/server/all/conf/jboss-service.xml ファイルをエディタで開きます。
2. <attribute name="CallByValue"> の値を true に変更します。
3. ファイルを保存して閉じます。
4. [appserver root]/server/all/deploy/ear-deployer.xml ファイルをエディタで開きます。
5. <attribute name="Isolated"> の値を true に変更します。
6. <attribute name="CallByValue"> の値を true に変更します。
7. ファイルを保存して閉じます。

▶ JSF ライブラリの無効化

1. [appserver root]/server/all/deploy/jboss-web.deployer ディレクトリに移動し、jsf-libs サブディレクトリとディレクトリ内のすべてのファイルを削除します。
2. [appserver root]/server/all/deploy/jboss-web.deployer/conf/web.xml から、以下のエントリがコメントアウトされていない場合は、コメントアウトまたは削除します。

```
<!-- Comment/Remove this -->
  <!-- Configures JSF for a web application if the
javax.faces.webapp.FacesServlet is declared -->
```

```
<!-- in web.xml.
-->
<!--
<listener>

<listener-class>org.jboss.web.jsf.integration.config.JBossJSFConfigureListen
er</listener-class>
</listener>
-->
<!-- Comment/Remove this -->
<!-- Listens to all web app lifecycle events so that @PreDestroy can be
called on -->
<!-- JSF managed beans that go out of scope. You can comment this out if you
-->
<!-- don't use JSF or you don't use annotations on your managed beans.
-->
<!--
<listener>

<listener-class>com.sun.faces.application.WebappLifecycleListener</listener-
class>
</listener>
-->

<!--
<init-param>
<description>JSF standard tlds</description>
<param-name>tagLibJar0</param-name>
<param-value>jsf-libs/jsf-impl.jar</param-value>
</init-param>
-->
```

▶ run.bat ファイルの変更 (Windows のみ)

1. [appserver root]/bin/run.bat ファイルをエディタで開きます。
2. rem Setup JBoss specific properties の直後の行を次のように変更し、UTF-8 へのエンコードを設定します。

```
set JAVA_OPTS=%JAVA_OPTS% -Dadobeidp.serverName=server1
-Dfile.encoding=utf8 -Dprogram.name=%PROGRAMNAME%
```

注意： この新しい行に改行が含まれないようにしてください。

3. rem Sun JVM memory allocation pool parameters の直後の行を次のように変更します。
 - (32 ビット Windows) set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=256m -Xms1024m -Xmx1024m -Djava.net.preferIPv4Stack=true
 - (64 ビット Windows) set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=512m -Xms1024m -Xmx1792m -Djava.net.preferIPv4Stack=true
4. (オプション) IPv6 モードで実行するように次のように JBoss Application Server を変更します。
 - -Djava.net.preferIPv4Stack=false のように変更します。
 - 文字列 -Djava.net.preferIPv6Stack=true を挿入します。

注意：アプリケーションサーバーログで起動時に次のエラーが記録される場合は、IPv6 スタックの値を削除して、IPV4 の値を true に戻します。

```
"13:37:44,488 WARN [HANamingService] Failed to start AutomaticDiscovery
java.net.SocketException: bad argument for IP_MULTICAST_IF: address not bound to any interface
at java.net.PlainDatagramSocketImpl.socketSetOption(Native Method)at
java.net.PlainDatagramSocketImpl.setOption(PlainDatagramSocketImpl.java:260)"
```

5. ファイルを保存して閉じます。

▶ run.sh ファイルの変更 (JBoss、Solaris 10、Red Hat 5.0、64 ビット専用)

Sun の Solaris および Linux JDK では、64 ビット機能を使用するために追加の引数が必要です。この設定を変更しないと、Sun JDK では既定で 32 ビットしかサポートされません。

注意：ルートユーザー以外で JBoss を実行している場合、`-Djava.io.tmpdir="location"` を使用して、一時ディレクトリの場所をアクセス権があるディレクトリに設定します。

1. [appserver root]/bin/run.sh ファイルをエディタで開きます。

2. 次の行を探します。

```
JAVA_OPTS=$JAVA_OPTS -XX:PermSize=256m -XX:MaxPermSize=256m -Xms1024m
-Xmx1024m
```

注意：JAVA_OPTS 引数は、run.sh ファイルではなく run.conf ファイルにある場合があります。両方のファイルでこの引数を探して、必要な変更を行います。

3. `-XX:MaxPermSize=256m` 引数および `-Xmx1024m` 引数を次のように変更します。

```
JAVA_OPTS=%JAVA_OPTS% -XX:PermSize=256m -XX:MaxPermSize=512m -Xms1024m
-Xmx1792m
```

4. ファイル内の JAVA_OPTS 割り当て文の直後に次の文を追加して、エンコードを UTF-8 に設定します。

```
JAVA_OPTS="$JAVA_OPTS -d64 -Dadobeidp.serverName=server1
-Dfile.encoding=utf8 -Dprogram.name=$PROGNAME
```

注意：このエントリが run.sh ファイル内に 1 行で表示されることを確認してください。

5. (オプション) IPv6 モードで実行するように次のように JBoss Application Server を変更します。

- `-Djava.net.preferIPv4Stack=false` のように変更します。
- `-Djava.net.preferIPv6Stack=true` を追加します。

注意：アプリケーションサーバーログで起動時に次のエラーが記録される場合は、IPv6 スタックの値を削除して、IPV4 の値を true に戻します。

```
"13:37:44,488 WARN [HANamingService] Failed to start AutomaticDiscovery
java.net.SocketException: bad argument for IP_MULTICAST_IF: address not bound to any interface
at java.net.PlainDatagramSocketImpl.socketSetOption(Native Method)"
```

6. ファイルを保存して閉じます。

▶ run.conf ファイルの変更

1. [appserver root]/bin/run.conf ファイルをエディタで開きます。
2. JAVA_OPTS="-server -Xms128m -Xmx128m" を見つけて、次のように変更します (32 ビットオペレーティングシステムの場合)。

```
JAVA_OPTS="-server -XX:MaxPermSize=256m -Xms1024m -Xmx1024m"  
-Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000
```

64 ビットオペレーティングシステムの場合は、行を次のように変更します。

```
JAVA_OPTS="-server -d64 -XX:MaxPermSize=512m -Xms1024m -Xmx1792m"  
-Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000
```

3. (Linux のみ) 「[HANamingService] Failed to start AutomaticDiscovery」という警告を削除するために "-Djava.net.preferIPv4Stack=true" フラグも JAVA_OPTS に追加する必要があります。
4. ファイルを保存して閉じます。

▶ ログレベルを DEBUG から INFO に引き上げるための log4j.xml ファイルの変更

1. [appserver root]/server/all/conf/jboss-log4j.xml ファイルをエディタで開きます。
2. FILE appender セクションで次のテキストを見つけて、太字で表示されている行を追加します。
<appender name="FILE" class="org.jboss.logging.appender.DailyRollingFileAppender">

```
<errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>  
<param name="File" value="{jboss.server.home.dir}/log/cluster.log"/>  
<param name="Threshold" value="INFO"/>  
<param name="Append" value="false"/>
```

3. CONSOLE appender セクションで次のテキストを見つけます。

```
<appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">  
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>  
  <param name="Target" value="System.out"/>  
  <param name="Threshold" value="INFO"/>
```

最後の行を次のように変更します。

```
<param name="Threshold" value="WARN"/>
```

4. Limit categories セクションを見つけて、以下を追加します。

```
<category name="com.adobe">  
  <priority value="INFO"/>  
</category>  
<category name="org.apache.xml.security.signature.Reference">  
  <priority value="WARN"/>  
</category>  
<category name="org.alfresco">  
  <priority value="WARN"/>  
</category>  
<category name="org.alfresco.repo.policy">  
  <priority value="WARN"/>  
</category>  
<category name="org.springframework">  
  <priority value="WARN"/>
```

```
</category>
<category name="org.hibernate">
  <priority value="WARN"/>
</category>
<category name="org.hibernate.cache.ReadWriteCache">
  <priority value="ERROR"/>
</category>
<category name="org.hibernate.cache.EhCacheProvider">
  <priority value="ERROR"/>
</category>
<category name="org.hibernate.engine.
StatefulPersistenceContext.ProxyWarnLog">
  <priority value="ERROR"/>
</category>
<category name="org.jbpm.jpdl.xml.JpdlXmlReader">
  <priority value="ERROR"/>
</category>
```

5. ファイルを保存して閉じます。

▶ JMX コンソールおよび Web コンソールの削除

JBoss から安全性の低いデフォルトの Web ページを削除するには、次のディレクトリを削除します。

```
[appserver root]/server/all/deploy/jmx-console.war
[appserver root]/server/all/deploy/management
```

▶ jmx-invoker-service.xml ファイルの変更

1. [appserver root]/server/all/deploy ディレクトリに移動して、jmx-invoker-service.xml ファイルをテキストエディタで開きます。

2. invoke セクションで次の行がコメントアウトされていることを確認します。

```
<interceptor code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
securityDomain="java:/jaas/jmx-console"/>
```

3. ファイルを保存して閉じます。

▶ jbossjta-properties.xml ファイルの変更

1. [appserver root]/server/all/conf ディレクトリにある jbossjta-properties.xml を探して、エディタで開き、トランザクション管理が期待どおり機能することを確認します。

2. プロパティエレメントを探して次のように変更します (太字で示してあります)。太字のプロパティが jbossjta-properties.xml ファイルにない場合は、このプロパティを追加します。

```
<properties depends="arjuna" name="jta">
  <property name="com.arjuna.ats.jta.allowMultipleLastResources"
value="true"/>
  <!-- ... other properties ... -->
</properties>
```

3. ファイルを保存して閉じます。

注意： 詳しくは、JBoss の記事 11443 (<http://www.jboss.org/community/docs/DOC-11443>) を参照してください。

A.4.4 hibernate オプションの変更 (Content Services ES2 のみ)

手動で設定した JBoss 環境に Content Services ES2 をインストールおよびデプロイする場合、JAVASSIST ではなく CGLIB バイトコードプロバイダを使用するように hibernate を変更する必要があります。変更しないと、Content Services ES2 のデプロイは失敗します。

▶ Hibernate バイトコードプロバイダの変更

1. persistence.properties ファイルを検索して開きます。
`[JBOSS_HOME]/server/all/deploy/ejb3.deployer/META-INF/`
2. hibernate.bytecode.provider を見つけて、値を cglib に変更します。
3. ファイルを保存して閉じます。

A.5 LiveCycle ES2 データベース接続

LiveCycle ES2 データベース接続を設定するには、次のタスクを実行する必要があります。

- LiveCycle ES2 データソースを設定します。
- データベースをデフォルトのデータソースとして使用するよう JBoss を設定します。

アプリケーションサーバーのインストールディレクトリにデータベースドライバをインストールする必要があります。ドライバは、LiveCycle Configuration Manager およびアプリケーションサーバーが LiveCycle ES2 データベースに接続できるようにするために必要です。データベースに使用するデータベースタイプに対応するドライバをインストールします。

データベースに接続するデータソースを設定する必要があります。JBoss の場合は、MySQL、Oracle、または SQL Server データソースを設定できます。

注意： 以下の操作に進む前に、JBoss が実行されていないことを確認してください。

A.5.1 手動でインストールした JBoss での MySQL の設定

JBoss から LiveCycle ES2 データを格納している MySQL データベースへの接続を有効にするには、次のタスクを実行する必要があります。

- MySQL JDBC ドライバを取得して、LiveCycle ES2 をデプロイする JBoss のインスタンスにコピーします。
- データソースファイルを作成し、LiveCycle ES2 をデプロイする JBoss のインスタンスにデプロイします。
- <http://community.jboss.org/wiki/EncryptingDataSourcePasswords> で説明されているいずれかの方法を使用して、データソースファイル (adobe-ds.xml および mysql-ds.xml) および login-config.xml ファイルのパスワードを暗号化します。
http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearart.html の手順を使用することもできます。

アドビの事前設定 JBoss を使用している場合は、「[アドビの事前設定 JBoss 4.2.1 用 MySQL の設定](#)」(72 ページ) を参照してください。

A.5.1.1 MySQL データソースの設定

MySQL データソースを設定する前に、データベースを MySQL に作成しておく必要があります (「[MySQL データベースの作成](#)」(65 ページ) を参照)。

A.5.1.2 MySQL データベースドライバのインストール

- インストール DVD の [DVD_root]/third_party/db/mysql ディレクトリから [appserver root]/server/all/lib ディレクトリに mysql-connector-java-5.1.6-bin.jar ドライバファイルをコピーします。

A.5.1.3 adobe-ds.xml ファイルの編集

1. インストール DVD の [DVD_root]/third_party/datasources/lc_mysql/deploy ディレクトリから [appserver root]/server/all/deploy ディレクトリに adobe-ds.xml ファイルをコピーします。

2. adobe-ds.xml ファイルをテキストエディタで開き、次の行を見つけます。

```
<connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>  
<driver-class>com.mysql.jdbc.Driver</driver-class>  
<user-name>adobe</user-name>
```

3. <password>adobe</password> 次の値をデータベースに固有の値に置き換えます。

- localhost: データベースをホストするコンピュータの名前、IP アドレス、または完全修飾パス。デフォルトは localhost です。
- 3306: データベースへのアクセスに使用するポート。デフォルトのポートは 3306 です。
- adobe: LiveCycle ES2 データを格納しているデータベースの名前。デフォルト値の adobe を、独自のデータベース名に更新する必要があります。

4. <connection-url> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

5. データソース接続の最小値と最大値を次のように変更します。

- IDP_DS :

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>30</max-pool-size>
```

- EDC_DS :

```
<min-pool-size>1</min-pool-size>  
<max-pool-size>20</max-pool-size>
```

注意: LiveCycle ES2 サーバーで処理する負荷が大きい場合は、JDBC 接続の最大数を増やして、すべてのジョブが確実に処理されるようにします。このような場合は、IDP_DS と EDC_DS の両方に関して、<max-pool-size> を 50 以上に増やします。

6. ファイルを保存します。

A.5.1.4 データソースとしての MySQL の設定

1. [appserver root]/server/all/deploy ディレクトリに移動し、hsqldb-ds.xml ファイルを削除します。
2. [appserver root]/docs/examples/jca ディレクトリに移動し、[appserver root]/server/all/deploy ディレクトリに mysql-ds.xml ファイルをコピーします。

3. [appserver root]/server/all/deploy/mysql-ds.xml ファイルをテキストエディタで開き、<local-tx-datasource> エレメントを MySQL の接続設定の内容に変更します。

```
<jndi-name>MySQLDS</jndi-name>  
<connection-url>jdbc:mysql://mysql-hostname:3306/jbossdb</connection-url>  
<driver-class>com.mysql.jdbc.Driver</driver-class>  
<user-name>x</user-name>  
<password>y</password>
```

4. 太字の値をデータベースに固有の次の値に置き換えます。
 - `MySQLDS` : `DefaultDS` に変更します。
 - `mysql-hostname`、`3306`、`jbossdb`、`x`、および `y` : アプリケーションサーバーがデータベースへのアクセスに使用するデータベース値。
5. ファイルを保存して閉じます。

A.5.1.5 standardjbosscomp-jdbc.xml ファイルの編集

1. `[appserver root]\%server%\all%\conf%\standardjbosscomp-jdbc.xml` ファイルを開き、次のエレメントを変更します (これらのエレメントは連続しているとは限りません)。

```
<!-- optional since 4.0 <datasource-mapping>Hypersonic SQL
</datasource-mapping> -->
  <fk-constraint>false</fk-constraint>
```

 - `<datasource-mapping>` タグから `<!-- optional since 4.0` テキストおよび末尾の `-->` コメントタグを削除して、コメント化を解除します。
 - `<datasource-mapping>` エレメントで `Hypersonic SQL` を `MySQL` に置き換えます。最終的に、`<datasource-mapping>` タグは次の行のようになります。

```
  <datasource-mapping>MySQL</datasource-mapping>
```
 - `<fk-constraint>` エレメントで `false` を `true` に置き換えます。
2. ファイルを保存して閉じます。

A.5.1.6 login-config.xml ファイルの編集

1. `[appserver root]\%server%\all%\conf%\login-config.xml` ファイルをテキストエディタで開き、`<policy>` エレメント内に次のテキストを追加します。

```
<application-policy name="MySQLDbRealm">
  <authentication>
    <login-module
      code="org.jboss.resource.security.ConfiguredIdentityLoginModule" flag
      = "required">
      <module-option name="principal">adobe</module-option>
      <module-option name="userName">adobe</module-option>
      <module-option name="password">adobe</module-option>
      <module-option
        name="managedConnectionFactoryName">jboss.jca:service=LocalTxCM,
        name=DefaultDS </module-option>
    </login-module>
  </authentication>
</application-policy>
```
2. 太字の値をデータベースに固有の次の値に置き換えます。
3. ファイルを保存して閉じます。
4. JBoss を起動します。

A.5.2 手動でインストールした JBoss 用の Oracle の設定

JBoss から LiveCycle ES2 データを格納している Oracle データベースへの接続を有効にするには、LiveCycle ES2 を手動でデプロイする場合、次のタスクを実行する必要があります。

- Oracle JDBC ドライバを取得して、LiveCycle ES2 をデプロイする JBoss のインスタンスにコピーします。
- データソースファイルを作成し、LiveCycle ES2 をデプロイする JBoss のインスタンスにデプロイします。
- <http://community.jboss.org/wiki/EncryptingDataSourcePasswords> で説明されているいずれかの方法を使用して、データソースファイル (adobe-ds.xml および oracle-ds.xml) および login-config.xml ファイルのパスワードを暗号化します。
http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clear.html の手順を使用することもできます。

A.5.2.1 Oracle 11g データベースドライバのインストール

[LiveCycleES2 root]/lib/db/oracle ディレクトリから [appserver root]/server/all/lib ディレクトリに JDK 1.5 用の ojdbc5.jar および JDK 1.6 用の ojdbc6.jar ドライバファイルをコピーします。また、Oracle 11g Release 1 (11.1.0.6 Thin) ドライバは、[JDBC Driver Downloads](#) サイトからダウンロードできます。

A.5.2.2 adobe-ds.xml ファイルの編集

1. [DVD_root]/third_party/datasources/lc_oracle/deploy ディレクトリから [appserver root]/server/all/deploy ディレクトリに adobe-ds.xml ファイルをコピーします。
2. adobe-ds.xml ファイルをテキストエディタで開き、次の行を見つけます。

```
<connection-url>jdbc:oracle:thin:@localhost:1521:adobe</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```
3. 次の値をデータベースに固有の値に置き換えます。
 - **localhost** : データベースをホストするコンピュータの名前、IP アドレス、または完全修飾パス。デフォルトは localhost です。
 - **1521** : データベースへのアクセスに使用するポート。デフォルトのポートは 1521 です。
 - **adobe** : デフォルト値 adobe をデータベース SID に変更します。
4. <connection-url> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。
5. (Oracle RAC でのみ) 手順 2 で示されている接続 URL を次の接続 URL に置き換えます。

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS=(
PROTOCOL=TCP) (HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP)
(HOST=yourhost2) (PORT=1521)) (LOAD_BALANCE=on) (FAILOVER=on))
(CONNECT_DATA=(SERVER=dedicated) (SERVICE_NAME=service.yourcompany.com)
(FAILOVER_MODE=(TYPE=session) (METHOD=basic) (RETRIES=10) (DELAY=3))))
```

注意 : このエントリが adobe-ds.xml ファイル内に 1 行で表示されることを確認してください。

6. (Oracle RAC でのみ) 手順 5 で示されている接続 URL に含まれる次のテキストをデータベースに固有の値に置き換えます。
 - **yourhost1** : データベースをホストするクラスタ内の最初のノードの名前、IP アドレス、または完全修飾ドメイン名。

- **yourhost2** : データベースをホストするクラスタ内の 2 番目のノードの名前、IP アドレス、または完全修飾ドメイン名。
 - 注意** : データベースをホストするクラスタは任意の数のノードで構成することができます。 **yourhost1** および **yourhost2** は、クラスタが 2 つのノードで構成される場合の例です。
 - **service.yourcompany.com** : Oracle RAC データベースのサービス名。
7. データソース接続の最小値と最大値を次のように変更します。
- IDP_DS :

```
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
```
 - EDC_DS :

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```
- 注意** : LiveCycle ES2 サーバーで処理する負荷が大きい場合は、JDBC 接続の最大数を増やして、すべてのジョブが確実に処理されるようにします。このような場合は、IDP_DS と EDC_DS の両方に関して、`<max-pool-size>` を 50 以上に増やします。
8. ファイルを保存します。

A.5.2.3 データソースとしての Oracle の設定

Oracle データベースと共に LiveCycle ES2 を実行する場合は、Oracle を JBoss のデフォルトのデータソースに設定する必要があります。この手順では、Oracle JDBC ドライバが [appserver root]/server/all/lib ディレクトリにインストールされていることを前提としています。

1. [appserver root]/server/all/deploy ディレクトリに移動し、hsqldb-ds.xml ファイルを削除します。
2. oracle-ds.xml ファイルを、[appserver root]/docs/examples/jca ディレクトリから [appserver root]/server/all/deploy ディレクトリにコピーします。
3. [appserver root]/server/all/deploy/oracle-ds.xml ファイルをテキストエディタで開き、`<local-tx-datasource>` エレメントを Oracle の接続設定の内容に変更します。

```
<jndi-name>OracleDS</jndi-name>
<connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid
</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>x</user-name>
<password>y</password>
```
4. 太字の値をデータベースに固有の次の値に置き換えます。
 - **OracleDS** : この値を DefaultDS に変更します。
 - **youroraclehost** : この値を Oracle サーバーのホスト名に置き換えます。
 - **1521** : Oracle でデフォルトのポートを使用していない場合は、この値を適切なポート番号に置き換えます。
 - **yoursid** : この値を Oracle システム識別子に置き換えます。
5. `<connection-url>` 設定に続く行で、`user-name` および `password` 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。

6. (Oracle RAC でのみ) 手順 3 で示されている接続設定を次の接続 URL に置き換えます。

```
jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken) (ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=yourhost1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=yourhost2) (PORT=1521)) (LOAD_BALANCE=on) (FAILOVER=on)) (CONNECT_DATA=(SERVER=dedicated) (SERVICE_NAME=service.yourcompany.com) (FAILOVER_MODE=(TYPE=session) (METHOD=basic) (RETRIES=10) (DELAY=3))))
```

注意： このエントリが oracle-ds.xml ファイル内に 1 行で表示されることを確認してください。

7. (Oracle RAC でのみ) 手順 6 で示されている接続 URL に含まれる次のテキストをデータベースに固有の値に置き換えます。

- **yourhost1** : データベースをホストするクラスタ内の最初のノードの名前、IP アドレス、または完全修飾ドメイン名。
- **yourhost2** : データベースをホストするクラスタ内の 2 番目のノードの名前、IP アドレス、または完全修飾ドメイン名。

注意： データベースをホストするクラスタは任意の数のノードで構成することができます。**yourhost1** および **yourhost2** は、クラスタが 2 つのノードで構成される場合の例です。

- **service.yourcompany.com** : Oracle RAC データベースのサービス名。

8. ファイルを保存して閉じます。

A.5.2.4 standardjbosscomp-jdbc.xml ファイルの編集

1. [appserver root]/server/all/conf/standardjbosscomp-jdbc.xml ファイルを開き、次のエレメントを変更します (これらのエレメントは連続した行とは限りません)。

```
<!-- optional since 4.0 <datasource-mapping>Hypersonic SQL  
</datasource-mapping> -->  
  <fk-constraint>>false</fk-constraint>
```

- <datasource-mapping> タグから <!-- optional since 4.0 テキストおよび末尾の --> コメントタグを削除して、コメント化を解除します。

- <datasource-mapping> エレメントで Hypersonic SQL を Oracle10g に置き換えます。

最終的に、<datasource-mapping> タグは次の行のようになります。

```
<datasource-mapping>Oracle10g</datasource-mapping>
```

注意： [appserver root]/server/all/deploy/oracle-ds.xml および [appserver root]/server/all/conf/standardjbosscomp-jdbc.xml で指定した Oracle データベースのバージョンは一致している必要があります。

- <fk-constraint> の false を true に置き換えます。

2. ファイルを保存して閉じます。

A.5.2.5 login-config.xml ファイルの編集

1. [appserver root]/server/all/conf/login-config.xml ファイルをテキストエディタで開き、<policy> エレメント内に次のテキストを追加します。

```
<application-policy name = "OracleDbRealm">  
  <authentication>  
    <login-module code =  
      "org.jboss.resource.security.ConfiguredIdentityLoginModule" flag =  
      "required">
```

```
<module-option name = "principal">adobe</module-option>
<module-option name = "userName">adobe</module-option>
<module-option name = "password">adobe</module-option>
<module-option name = "managedConnectionFactoryName">
  jboss.jca:service=LocalTxCM,name=DefaultDS</module-option>
</login-module>
</authentication>
</application-policy>
```

2. 太字のテキストをデータベースに固有の次の値に置き換えます。

注意： Oracle RAC の場合、**adobe** をデータベースに固有のサービス名に置き換えます。

3. ファイルを保存して閉じます。
4. JBoss を起動します。

A.5.3 SQL Server の設定

JBoss から LiveCycle ES2 データを格納している SQL Server データベースへの接続を有効にするには、次のタスクを実行する必要があります。

- SQL Server JDBC ドライバファイルを取得して、LiveCycle ES2 をデプロイする JBoss のインスタンスにコピーします。
- SQL Server データソースファイルを作成し、LiveCycle ES2 をデプロイする JBoss のインスタンスにデプロイします ([appserver_root]/server/all/deploy など)。
- <http://community.jboss.org/wiki/EncryptingDataSourcePasswords> で説明されているいずれかの方法を使用して、データソースファイル (adobe-ds.xml および mssql-ds.xml) および login-config.xml ファイルのパスワードを暗号化します。
http://blogs.adobe.com/livecycle/2009/10/livecycle_-_encrypting_clearite.html の手順を使用することもできます。

A.5.3.1 SQL Server データソースの設定

SQL Server データソースを設定する前に、LiveCycle ES2 データベースが SQL Server に作成されている必要があります ([「SQL Server データベースの作成」 \(56 ページ\)](#) を参照)。

A.5.3.2 JBoss 用 SQL データベースドライバのインストール

1. Microsoft の Web サイトから SQL Server JDBC 1.2 データベースドライバを取得します。

注意： Microsoft SQL Server 2005 SP3 および Microsoft SQL Server 2008 のどちらのサーバーにも SQL Server JDBC ドライバ 1.2 を使用します。

2. (Windows) *.exe ファイルをダウンロードして実行し、これらのファイルを一時ディレクトリに抽出します (この節の以降の部分では、[SQL_root] ディレクトリと呼びます)。
3. (Linux) *.tar.gz ファイルを一時ディレクトリに抽出します (この節の以降の部分では、[SQL_root] ディレクトリと呼びます)。
4. sqljdbc.jar ファイルを、[SQL_root]/sqljdbc_1.2/enu ディレクトリから [appserver root]/server/all/lib ディレクトリにコピーします。
5. [appserver root]/server/all/lib ディレクトリにある mysql-connector-java-3.1.12-bin.jar ファイルを削除します。

A.5.3.3 adobe-ds.xml ファイルの編集

- [DVD_root]/third_party/datasources/lc_sqlserver/deploy ディレクトリから [appserver root]/server/all/deploy ディレクトリに adobe-ds.xml ファイルをコピーします。
 - adobe-ds.xml ファイルをテキストエディタで開き、<local-tx-datasource> エlement を SQL Server の接続設定の内容に変更します。

```
<connection-url>jdbc:sqlserver://localhost:1433;DatabaseName=adobe
</connection-url>
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
<user-name>adobe</user-name>
<password>adobe</password>
```
 - 次の値をデータベースに固有の値に置き換えます。
 - localhost** : データベースをホストするコンピュータの名前、IP アドレス、または完全修飾パス。デフォルトは localhost です。
 - 1433** : データベースへのアクセスに使用するポート。
 - adobe** : LiveCycle ES2 データを格納しているデータベースの名前。デフォルト値の adobe を、独自のデータベース名に更新する必要があります。
 - <driver-class> Element を次のように変更します。

```
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
```
 - <driver-class> 設定に続く行で、user-name および password 設定を探し、デフォルト値をアプリケーションサーバーがデータベースにアクセスするために使用するユーザー名とパスワードに置き換えます。データソース接続の最小値と最大値を次のように変更します。
 - IDP_DS** :

```
<min-pool-size>1</min-pool-size>
<max-pool-size>30</max-pool-size>
```
 - EDC_DS** :

```
<min-pool-size>1</min-pool-size>
<max-pool-size>20</max-pool-size>
```
- 注意** : LiveCycle ES2 サーバーで処理する負荷が大きい場合は、JDBC 接続の最大数を増やして、すべてのジョブが確実に処理されるようにします。このような場合は、IDP_DS と EDC_DS の両方に関して、<max-pool-size> を 50 以上に増やします。
- ファイルを保存します。

A.5.3.4 mssql-ds.xml ファイルの編集

- [appserver root]/server/all/deploy ディレクトリに移動し、hsqldb-ds.xml ファイルを削除します。
- [appserver root]/docs/examples/jca ディレクトリに移動し、[appserver root]/server/all/deploy ディレクトリに mssql-ds.xml ファイルをコピーします。
- mssql-ds.xml ファイルをテキストエディタで開き、<local-tx-datasource> Element を SQL Server の接続設定の内容に変更します (これらの Element は連続した行とは限りません)。

```
<jndi-name>MSSQLDS</jndi-name>
<datasource-mapping>MS SQLSERVER2000</datasource-mapping>
<connection-url>jdbc:sqlserver://localhost:1433;
DatabaseName=MyDatabase</connection-url>
```

```
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver
  </driver-class>
<user-name>x</user-name>
<password>y</password>
```

- <jndi-name> エlementを DefaultDS に変更
- <datasource-mapping> エlementを MS SQLSERVER2000 に変更
- データベース名 *MyDatabase* を独自のデータベース名に変更
- <user-name>x エlementを独自のユーザー名に変更
- <password>y エlementを独自のパスワードに変更

注意： MS SQL Server 2005 を使用していても、<datasource-mapping> エlementでは常に MS SQLSERVER2000 を指定する必要があります。

4. ファイルを保存して閉じます。

A.5.3.5 standardjbosscmp-jdbc.xml ファイルの編集

1. [appserver root]/server/all/conf/standardjbosscmp-jdbc.xml ファイルを開き、次のElementを変更します (これらのElementは連続した行とは限りません)。

```
<!-- optional since 4.0 <datasource-mapping>Hypersonic SQL
</datasource-mapping> -->
  <fk-constraint>>false</fk-constraint>
```

- <datasource-mapping> タグから <!-- optional since 4.0 テキストおよび末尾の --> コメントタグを削除して、コメント化を解除します。
- <datasource-mapping> エlementで Hypersonic SQL を MS SQLSERVER2000 に置き換えます。

最終的に、<datasource-mapping> タグは次の行のようになります。

```
<datasource-mapping>MS SQLSERVER2000</datasource-mapping>
```

注意： MS SQL Server 2005 を使用していても、<datasource-mapping> エlementでは常に MS SQLSERVER2000 を指定する必要があります。

- <fk-constraint> の false を true に置き換えます。

2. ファイルを保存して閉じます。

A.5.3.6 login-config.xml ファイルの編集

1. [appserver root]/server/all/conf/login-config.xml ファイルをテキストエディタで開き、<policy> Element内に次の行を追加します。

```
<application-policy name = "MSSQLDbRealm">
  <authentication>
    <login-module code =
      "org.jboss.resource.security.ConfiguredIdentityLoginModule"
      flag = "required">
    <module-option name = "principal">adobe</module-option>
    <module-option name = "userName">adobe</module-option>
    <module-option name = "password">adobe</module-option>
    <module-option name =
```

```
"managedConnectionFactoryName">jboss.jca:service=LocalTxCM,name=
  DefaultDS
</module-option>
</login-module>
</authentication>
</application-policy>
```

2. 太字の値をデータベースに固有の次の値に置き換えます。
3. ファイルを保存して閉じます。

A.5.3.7 Windows での統合セキュリティの設定

1. 次の例に示すように、[appserver root]¥server¥all¥deploy にある adobe-ds.xml および mssql-ds.xml ファイルを変更し、`integratedSecurity=true` を接続 URL に追加します。

```
<connection-url>jdbc:sqlserver://<serverhost>:<port>;
  databaseName=<dbname>;integratedSecurity=<true></connection-url>
```

2. JBoss を実行しているコンピュータの Windows システムパス (C:¥Windows) に `sqljdbc_auth.dll` ファイルを追加します。 `sqljdbc_auth.dll` ファイルは、Microsoft SQL JDBC 1.2 ドライバのインストール先 (32 ビットオペレーティングシステムの場合のデフォルトは `[install dir]/sqljdbc_1.2/enu/auth/x86`、64 ビットオペレーティングシステムの場合のデフォルトは `[install dir]/sqljdbc_1.2/enu/auth/x64`) と同じ場所にあります。
3. JBoss for Adobe LiveCycle サービスのプロパティを開き、「ログオン」タブをクリックします。
4. 「アカウント」を選択し、有効なユーザーアカウントの値を入力します。コマンドラインから JBoss を実行する場合は、この変更は必要ありません。
5. SQL Server のセキュリティを混合モードから Windows 認証のみに変更します。

A.6 次の手順

『[LiveCycle ES2 のインストールおよびデプロイ \(JBoss 版\)](#)』の手順に従って、LiveCycle ES2 をインストールします。