



Adobe

LiveCycle™ ES2 アプリケーション サーバークラスタの設定 (WebLogic® 版)

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 アプリケーションサーバークラスタの設定 (WebLogic 版)
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 Builder, FrameMaker, LiveCycle, PageMaker, and PhotoShop are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Intel and Pentium are trademarks of Intel Corporation in the U.S. and other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Microsoft, SharePoint, Windows, Windows Server, and Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Novell and SUSE are registered trademarks of Novell, Inc. in the United States and other countries. Oracle, Java, Sun, and Solaris are 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. UNIX is a registered trademark of The Open Group in the US and other countries. 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 |
| このドキュメントの対象読者 | 6 |
| このドキュメントで使用する表記 | 7 |
| 関連情報 | 8 |
| 1 インストールとデプロイメントの概要 | 9 |
| LiveCycle ES2 のインストールのためのサーバークラスタの準備 | 9 |
| インストール、設定およびデプロイメントプロセス | 9 |
| 設定およびデプロイのためのタスクの選択 | 10 |
| LiveCycle ES2 へのアップグレード | 11 |
| LiveCycle ES2 インストール、設定およびデプロイメントリスト | 11 |
| 自動インストールおよびデプロイメントリスト | 11 |
| 手動インストールおよびデプロイメントリスト | 11 |
| 2 WebLogic Server クラスタの作成 | 13 |
| インストールの準備 | 13 |
| WebLogic Server のインストール | 14 |
| WebLogic ドメインの作成 | 14 |
| WebLogic Server クラスタの作成と設定 | 16 |
| クラスタの作成 | 16 |
| 既存のクラスタへの新しいノードの追加 | 18 |
| サブレットコンテナ用の認証秘密鍵証明書の設定 | 18 |
| クラスタ用のノードマネージャの設定 | 19 |
| ノードマネージャおよび管理対象サーバーの起動 | 20 |
| WebLogic Server クラスタのテスト | 21 |
| 次の手順 | 21 |
| 3 LiveCycle ES2 モジュールのインストール | 22 |
| インストーラの確認 | 22 |
| 製品ファイルのインストール | 23 |
| Linux または UNIX にインストールするための Windows ステージングプラットフォームへの インストール | 24 |
| JAVA_HOME 環境変数の設定 | 24 |
| インストール LiveCycle ES2 | 24 |
| エラーログの表示 | 26 |
| LiveCycle ES2 によってインストールされた JAR ファイルの設定 | 26 |
| キャッシュロケータの設定 (TCP を使用するキャッシュのみ) | 26 |
| TCP ロケータの変更 | 27 |
| TCP ロケータの起動 | 28 |
| フォントディレクトリの設定 | 29 |
| 次の手順 | 29 |

| | | |
|----------|--|-----------|
| 4 | LiveCycle ES2 をデプロイするための設定 | 30 |
| | LiveCycle Configuration Manager の概要 | 30 |
| | LiveCycle Configuration Manager の CLI バージョンと GUI バージョン | 31 |
| | JDBC ドライバの場所 | 31 |
| | LiveCycle ES2 の設定、デプロイ | 31 |
| | EAR ファイルのアンインストール..... | 41 |
| | 次の手順 | 41 |
| 5 | WebLogic Server クラスタの手動設定 | 43 |
| | WebLogic Server の設定 | 43 |
| | タイムアウトの設定 | 43 |
| | サーバーの開始引数の設定 | 44 |
| | 管理対象サーバーのクラスパスを設定するには | 46 |
| | JDBS 接続の設定 | 46 |
| | LiveCycle ES2 データソースの作成と設定..... | 46 |
| | Rights Management ES2 データソースの作成と設定..... | 48 |
| | 次の手順 | 50 |
| 6 | WebLogic Server Cluster への手動デプロイ | 51 |
| | LiveCycle ES2 モジュールのデプロイについて | 51 |
| | WebLogic Server ディレクトリ名 | 51 |
| | デプロイ可能なコンポーネントの概要..... | 51 |
| | Content Services ES2 をデプロイするための必須条件 | 52 |
| | WebLogic Server へのデプロイ | 52 |
| | WebLogic Server の再起動 | 53 |
| | 次の手順 | 54 |
| 7 | デプロイメント完了後の作業 | 55 |
| | アプリケーションサーバーの再起動 | 55 |
| | 日付、時刻およびタイムゾーンの設定 | 55 |
| | デプロイメントの確認..... | 56 |
| | LiveCycle 管理コンソールへのアクセス | 56 |
| | デフォルトのパスワードの変更 | 56 |
| | ログファイルの表示 | 57 |
| | LiveCycle ES2.5 Solution Accelerator のインストール | 57 |
| | WebLogic Server サーバー起動引数の設定..... | 57 |
| | モジュールの Web アプリケーションへのアクセス..... | 58 |
| | Rights Management ES2 へのアクセス | 59 |
| | User Management へのアクセス..... | 60 |
| | LiveCycle PDF Generator ES2 または 3D ES2 の設定 | 60 |
| | 環境変数の設定..... | 61 |
| | Adobe PDF プリンタをデフォルトのプリンタとして設定..... | 62 |
| | Acrobat Professional の設定..... | 62 |
| | 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 | 63 |
| | Windows Server 2003 に東アジア文字をインストールするには..... | 64 |
| | ネットワークプリンタクライアントのインストール | 65 |
| | 監視フォルダのパフォーマンスパラメータの設定 | 66 |
| | LiveCycle Rights Management ES2 の最終設定 | 66 |
| | Content Services ES2 の設定 | 67 |
| | LiveCycle ES2 の LDAP アクセス設定..... | 67 |
| | FIPS モードの有効化..... | 69 |
| | HTML 電子署名の設定 | 69 |
| | Document Management サービスの設定..... | 70 |

| | |
|--|-----------|
| Connector for EMC Documentum サービスの設定..... | 70 |
| Documentum リポジトリでの XDP MIME 形式の作成..... | 73 |
| Connector for IBM FileNet サービスの設定 | 74 |
| SharePoint クライアントアクセスの設定 | 82 |
| share.war ファイルの取得と編集..... | 82 |
| share.war ファイルのデプロイ..... | 83 |
| IPv6 モードでの CIFS の有効化..... | 83 |
| contentservices.war ファイルの編集..... | 83 |
| Connector for IBM Content Manager の設定 | 84 |
| システムイメージバックアップの実行..... | 88 |
| LiveCycle ES2 のアンインストール | 88 |
| 8 ロードバランシングの設定 | 89 |
| Apache サーバープラグインの設定..... | 89 |
| Apache HTTP サーバープラグインのインストール..... | 89 |
| Apache HTTP サーバープラグインの設定 | 90 |
| Apache HTTP サーバープラグインのテスト | 90 |
| 9 高度な実稼働環境の設定 | 91 |
| Output ES2 および Forms ES2 のプールサイズの設定..... | 91 |
| LiveCycle PDF Generator ES2..... | 91 |
| EJB プールサイズの設定 | 91 |
| Windows での CIFS の有効化 | 92 |
| NetBIOS over TCP/IP の有効化 | 93 |
| 他の IP アドレスの追加..... | 93 |
| SMB over NetBIOS レジストリの無効化 (Windows 2003 のみ) | 93 |
| ファイルとプリンタの共有の無効化 (Windows 2008 のみ) | 93 |
| 10 トラブルシューティング | 94 |
| A 付録 - コマンドラインインターフェイスのインストール | 95 |
| LiveCycle ES2 のインストール | 95 |
| エラーログ | 97 |
| コンソールモードでの LiveCycle ES2 のアンインストール | 97 |
| 次の手順..... | 98 |
| B 付録 - LCM コマンドラインインターフェイス | 99 |
| 操作の順序..... | 99 |
| コマンドラインインターフェイスのプロパティファイル | 100 |
| 共通のプロパティ | 100 |
| LiveCycle の構成プロパティ | 102 |
| アプリケーションサーバーの設定および検証のプロパティ | 105 |
| LiveCycle のデプロイプロパティ | 108 |
| LiveCycle の初期化プロパティ | 108 |
| BAM の初期化プロパティ | 108 |
| LiveCycle コンポーネントのデプロイプロパティ | 109 |
| コマンドラインインターフェイスの使用..... | 109 |
| 使用例 | 112 |
| エラーログ | 112 |
| 次の手順..... | 112 |

このドキュメントの内容

このドキュメントは、Adobe® LiveCycle™ ES2 (Enterprise Suite) Update 1 の理解に役立つ資料の 1 つです。LiveCycle ES2 は、顧客、パートナー、構成員および社員間でのビジネスクリティカルな情報のフローの自動化および迅速化に貢献する、柔軟で拡張性の高いプラットフォームです。

このガイドでは、次のクラスタ環境内のモジュールを Microsoft® Windows®、Linux®、および Sun™ Solaris™ でインストールおよび設定する方法と、これらのモジュールを BEA WebLogic Server® にデプロイする方法について説明します。

- Adobe LiveCycle ES2 Connector for EMC Documentum
- Adobe LiveCycle ES2 Connector for IBM FileNet
- Adobe LiveCycle ES2 Connector for IBM Content Manager
- Adobe LiveCycle Content Services ES2
- Adobe LiveCycle Digital Signatures ES2
- Adobe LiveCycle Forms ES2
- Adobe LiveCycle Foundation
- Adobe LiveCycle Output ES2
- Adobe LiveCycle PDF Generator ES2
- Adobe LiveCycle PDF Generator 3D ES2
- Adobe LiveCycle Process Management ES2
- Adobe LiveCycle Reader Extensions ES2
- Adobe LiveCycle Rights Management ES2

このドキュメントの対象読者

このドキュメントは、クラスタ環境で LiveCycle ES2 コンポーネントのインストール、設定、管理またはデプロイを行う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、Java™ 2 Platform Enterprise Edition (J2EE) アプリケーションサーバー、Linux、Windows、または Solaris オペレーティングシステム、Oracle®、DB2® または SQL Server 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 | デフォルト値 | 説明 |
|---------------------|--|--|
| [LiveCycleES2 root] | Windows : C:\Adobe\Adobe LiveCycle ES2\ Linux および UNIX : /opt/adobe/adobe livecycle es2/ | 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、LiveCycle Configuration Manager 用と LiveCycle ES2 SDK 用のサブディレクトリがあります。 |
| [appserver root] | WebLogic Server 10g (Windows) : C:\bea\wlserver_10.3\ WebLogic Server 10g (Linux および Solaris) : /opt/bea/wlserver10.3/ WebLogic Server 11g (Windows) : C:\Oracle\Middleware\wlserver_10.3\ WebLogic Server 11g (Linux および Solaris) : /opt/Oracle/Middleware/wlserver10.3/ | 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。 |
| [server name] | Server-0 | アプリケーションサーバーで設定されているサーバーの名前。 |
| [BEA_HOME] | WebLogic Server 10g (Windows) : C:\bea\ WebLogic Server 10g (Linux および Solaris) : /opt/bea/ WebLogic Server 11g (Windows) : C:\Oracle\Middleware\ WebLogic Server 11g (Linux および Solaris) : /opt/Oracle/Middleware/ | BEA_HOME 環境変数に指定されている、WebLogic Server のインストールディレクトリ。 |

| 名前 | デフォルト値 | 説明 |
|-------------------|---|--|
| [appserverdomain] | <p>WebLogic 10g Server (Windows) : C:\bea\user_projects\domains\%base_domain%</p> <p>WebLogic 11g Server (Windows) : C:\Oracle\Middleware\user_projects\domains\%base_domain%</p> <p>WebLogic 10g Server (Linux および UNIX) : /opt/bea/user_projects/domains/ base_domain/</p> <p>WebLogic 11g Server (Linux および UNIX) : /opt/Oracle/Middleware/user_projects/ domains/base_domain/</p> | WebLogic Server で設定したドメイン。 |
| [APACHE_HOME] | Apache がインストールされている場所。 | Apache HTTP サーバーがインストールされるディレクトリです。Apache をロードバランサとして使用する場合にのみ該当します。 |
| [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載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 (Linux および UNIX® 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 についての詳細情報の参照先を示します。

| 情報 | 参照先 |
|--|---|
| サーバークラスタでの LiveCycle ES2 のインストールの準備 | LiveCycle ES2 のインストールの準備 (サーバークラスタ) |
| LiveCycle ES2 の管理タスクの実行 | LiveCycle 管理ヘルプ |
| LiveCycle Workbench ES2 のインストール | Installing Your Development Environment |
| LiveCycle ES2 に統合できる他のサービスや製品 | www.adobe.com/jp |
| 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、および追加情報 | LiveCycle テクニカルサポート |

1 インストールとデプロイメントの概要

ここでは、LiveCycle ES2 をインストールするためにクラスタを準備する方法と、インストールおよびデプロイメントのために LiveCycle ES2 でどのような作業を実行する必要があるのかをわかりやすく説明し、LiveCycle ES2 のインストール、設定およびデプロイメントのプロセスの理解に役立つ情報を紹介します。

- [「LiveCycle ES2 のインストールのためのサーバークラスタの準備」 \(9 ページ\)](#)
- [「インストール、設定およびデプロイメントプロセス」 \(9 ページ\)](#)
- [「LiveCycle ES2 インストール、設定およびデプロイメントリスト」 \(11 ページ\)](#)

システム要件、データベースの準備、LiveCycle Reader Extensions ES2 の資格情報の設定など、サーバークラスタを作成して LiveCycle ES2 をインストールするためのシステムの準備について詳しくは、『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』を参照してください。

1.1 LiveCycle ES2 のインストールのためのサーバークラスタの準備

LiveCycle ES2 をインストール、設定およびデプロイする前に、サーバークラスタを作成および設定する必要があります ([「WebLogic Server クラスタの作成」 \(13 ページ\)](#) を参照)。

1.2 インストール、設定およびデプロイメントプロセス

LiveCycle ES2 のインストール、設定およびデプロイには、次のプロセスが含まれます。

インストール：インストールプログラムを実行して LiveCycle ES2 をインストールします。LiveCycle ES2 をインストールすると、必要なすべてのファイルが、使用するコンピュータ上の 1 つのインストールディレクトリ構造内に配置されます。デフォルトのインストールディレクトリは C:\Adobe\Adobe LiveCycle ES2 (Windows) または /opt/adobe_livecycle_es2 (Linux および Solaris) ですが、これ以外のディレクトリにファイルをインストールすることもできます。このドキュメントでは、デフォルトのインストールディレクトリを [LiveCycleES2 root] と呼びます ([「LiveCycle ES2 モジュールのインストール」 \(22 ページ\)](#) を参照)。

設定およびアセンブリ：LiveCycle ES2 の設定では、LiveCycle ES2 の動作方法を決定する様々な設定を変更します。製品のアセンブリでは、設定の指示に従って、すべてのインストール済みコンポーネントがデプロイ可能な EAR および JAR ファイルに配置されます。LiveCycle Configuration Manager を実行して、デプロイメント用コンポーネントの設定とアセンブリを行います ([「LiveCycle ES2 をデプロイするための設定」 \(30 ページ\)](#) を参照)。複数の LiveCycle ES2 モジュールに対して同時に設定およびアセンブリを行うこともできます。

デプロイ：製品のデプロイでは、アセンブリされた複数の EAR ファイルといくつかの補助ファイルを、LiveCycle ES2 ソリューションを実行する予定の WebLogic Server クラスタにデプロイします。複数のモジュールを設定およびアセンブリを行った場合は、デプロイ可能なコンポーネントがデプロイ可能な EAR ファイル内にパッケージングされています。また、コンポーネントおよび LiveCycle ES2 アーカイブ (LCA) ファイルは、JAR ファイルとしてパッケージングされています。

LiveCycle ES2 データベースの初期化：LiveCycle ES2 と共に使用するデータベースの初期化では、User Management およびその他のソリューションコンポーネントと共に使用されるテーブルが作成されます。LiveCycle ES2 データベースに接続するモジュールをデプロイする場合は、デプロイメントプロセスの完了後に LiveCycle ES2 データベースを初期化する必要があります。

1.3 設定およびデプロイのためのタスクの選択

インストール完了後には、LiveCycle Configuration Manager を実行して、次のような様々なタスクを実行することができます。

- アプリケーションサーバークラスタにデプロイするために EAR ファイルの LiveCycle ES2 モジュールを設定する
 - LiveCycle ES2 をサポートするようにアプリケーションサーバーのプロパティをクラスタのすべてのサーバーに設定する
 - JDBC モジュールを LiveCycle ES2 EAR にパッケージングする (データソースをセキュリティで保護)
 - アプリケーションサーバークラスタの設定を検証する
 - JDBC モジュールを LiveCycle ES2 EAR ファイルにパッケージする (データソースをセキュリティで保護)
 - LiveCycle ES2 EAR ファイルをデプロイする
 - LiveCycle ES2 データベースを初期化する
 - LiveCycle ES2 コンポーネントをデプロイする
 - LiveCycle ES2 コンポーネントのデプロイメントを検証する
 - LiveCycle ES2 サンプルを LiveCycle ES2 に読み込む (オプション)
- 注意：**読み込むことができる LiveCycle ES2 サンプルに加えて、[Adobe の Web サイト](#)ではさらに多くのサンプルを利用できます。
- LiveCycle ES2 コンポーネントを設定する

Reader Extensions ES2 をインストールする場合は、PDF ドキュメントに使用権限を適用するために必要な Reader Extensions ES2 使用権限資格情報を指定して読み込むこともできます。

Business Activity Monitoring をインストールする場合は、他のすべての LiveCycle ES2 コンポーネントをインストールしてから、WebLogic のシングルサーバーインストールガイドの「LiveCycle Business Activity Monitoring ES2 の設定」を参照してください。

アプリケーションサーバークラスタの設定およびデータベースに対するデータソースの設定は LiveCycle Configuration Manager を使用して実行できますが、これらの手順は手動で実行した方が望ましい場合もあります。

次の理由から、アプリケーションサーバークラスタを手動で設定する必要がある場合があります。

- アプリケーションサーバークラスタで他のアプリケーションを実行しており、設定が競合する可能性がある場合
- 設定管理に関する会社のセキュリティ手順により、より細かい制御が求められている場合
- 自動設定が使用できないデプロイメントの場合

手動で設定する場合は、次のタスクを実行します。

- LiveCycle Configuration Manager を使用して、LiveCycle ES2 コンポーネントを設定し、必要なフォント、一時ディレクトリおよび GDS ディレクトリを設定します。
- アプリケーションサーバークラスタの設定、データソースの設定および LiveCycle ES2 EAR ファイルのデプロイを手動で行います。
- LiveCycle Configuration Manager を実行してデータベースを初期化します。
- LiveCycle Configuration Manager を実行して LiveCycle ES2 コンポーネントをデプロイし、LiveCycle ES2 コンポーネントのデプロイメントを検証します。
- LiveCycle ES2 コンポーネントを設定します。

1.4 LiveCycle ES2 へのアップグレード

LiveCycle 7.x からアップグレードする場合は、『LiveCycle 7.x から LiveCycle ES2 へのアップグレード』ガイドを参照してください。

LiveCycle 8.x 以降から LiveCycle ES2 にアップグレードする場合は、『[LiveCycle ES2 へのアップグレードの準備](#)』に記載されている作業を完了したことを確認してください。また、使用しているアプリケーションサーバー版の [LiveCycle ES から LiveCycle ES2 へのアップグレード](#) ガイドを参照してください。

1.5 LiveCycle ES2 インストール、設定およびデプロイメントリスト

この節では、インストールおよび設定プロセスを進めていく際に使用できるリストを用意しました。インストールと設定のためのリストには、自動オプション用のものと手動オプション用のものがあります。

自動オプション：LiveCycle Configuration Manager を使用して、LiveCycle ES2 EAR ファイルの設定とデプロイ、アプリケーションサーバークラスタの設定、データベースの初期化、サーバークラスタへのモジュールのデプロイを行う方法です。自動オプションは、LiveCycle ES2 のインストール、設定およびデプロイメントのための入力処理を少なくしたい場合に使用します。

手動オプション：LiveCycle Configuration Manager を、LiveCycle ES2 EAR ファイルの設定、データベースの初期化、サーバークラスタへのモジュールのデプロイだけに使用する方法です。アプリケーションサーバークラスタの設定、データベースへの接続、および LiveCycle ES2 EAR ファイルのサーバークラスタへのデプロイは、このマニュアルで後述する手順に従って、管理者が実行します。手動オプションは、LiveCycle ES2 のインストール、設定およびデプロイメントのための厳密な入力を行う場合に使用します。例えば、このオプションは、ロックダウンされたサーバー環境で使用できます。

1.5.1 自動インストールおよびデプロイメントリスト

次のリストに、自動オプションを使用して LiveCycle ES2 モジュールをインストールするために必要な手順を示します。インストールを実行する前に、WebLogic Server クラスタをインストールしておく必要があります。

- 必要なソフトウェアがインストール先の環境にあらかじめインストールされているか確認します（『[LiveCycle ES2 のインストールの準備](#)』および『[WebLogic Server クラスタの作成](#)』（13 ページ）を参照）。
- インストールプログラムを実行します（『[LiveCycle ES2 モジュールのインストール](#)』（22 ページ）を参照）。
- LiveCycle Configuration Manager を実行し、タスク選択画面のすべてのタスクを選択します。この手順により、LiveCycle ES2 EAR ファイルの設定とアセンブリ、アプリケーションサーバーの設定、アプリケーションサーバークラスタへの EAR ファイルおよびその他のコンポーネントのデプロイ、LiveCycle ES2 データベースの初期化およびデプロイメントの検証が行われます（『[LiveCycle ES2 をデプロイするための設定](#)』（30 ページ）を参照）。
- LiveCycle 管理コンソールおよび User Management にアクセスします（『[LiveCycle 管理コンソールへのアクセス](#)』（56 ページ）を参照）。
- (オプション) LDAP アクセスを設定します（『[LiveCycle ES2 の LDAP アクセス設定](#)』（67 ページ）を参照）。

1.5.2 手動インストールおよびデプロイメントリスト

次のリストに、手動オプションを使用して LiveCycle ES2 をインストールする場合に必要な手順を示します。インストールを実行する前に、アプリケーションサーバークラスタをインストールおよび設定しておく必要があります。

- 必要なソフトウェアがインストール先の環境にあらかじめインストールおよび設定されていることを確認します（『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』を参照）。

- インストール先の環境でアプリケーションサーバークラスタが作成および設定されていることを確認します ([「WebLogic Server クラスタの作成」\(13 ページ\)](#) を参照)。
- インストールプログラムを実行します ([「LiveCycle ES2 モジュールのインストール」\(22 ページ\)](#) を参照)。
- LiveCycle Configuration Manager を実行し、LiveCycle ES2 EAR の設定タスクを選択します。このタスクにより、LiveCycle ES2 の設定とアセンブリが行われます ([「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照)。
- WebLogic Application Server を手動で構成します ([「WebLogic Server クラスタの手動設定」\(43 ページ\)](#) を参照)。
- EAR ファイルをアプリケーションサーバークラスタにデプロイします。これは、手動で行うか、LiveCycle Configuration Manager を使用して行うことができます ([「付録 - コマンドラインインターフェイスのインストール」\(95 ページ\)](#) を参照)。
- LiveCycle Configuration Manager を実行して LiveCycle ES2 コンポーネントファイルをデプロイし、LiveCycle ES2 データベースを初期化して、必要に応じて製品サンプルをデプロイします ([「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照)。
- LiveCycle 管理コンソールおよび User Management にアクセスします ([「LiveCycle 管理コンソールへのアクセス」\(56 ページ\)](#) を参照)。
- (オプション) LDAP アクセスを構成します ([「LiveCycle ES2 の LDAP アクセス設定」\(67 ページ\)](#) を参照)。

2

WebLogic Server クラスタの作成

WebLogic Server ソフトウェアは、クラスタ内の各コンピュータにインストールする必要があります。WebLogic Administration Server ソフトウェアもインストールする必要がありますが、クラスタに含まれていない別の管理サーバーにインストールすることもできます。WebLogic Server のインストールについて詳しくは、以下の Web サイトを参照してください。

- (WebLogic Server 10gR3) この [ドキュメントのページ](#) を参照してください。

クラスタ環境で WebLogic Server をインストールおよび設定するには、次の作業を行います。

- クラスタ内のすべてのコンピュータが正しく準備されていることを確認します ([「インストールの準備」\(13 ページ\)](#) を参照)。
- WebLogic Server ソフトウェアをインストールします ([「WebLogic Server のインストール」\(14 ページ\)](#) を参照)。
- WebLogic ドメインを作成します ([「WebLogic ドメインの作成」\(14 ページ\)](#) を参照)。
- WebLogic Server クラスタを設定します ([「WebLogic Server クラスタの作成と設定」\(16 ページ\)](#) を参照)。
- WebLogic Server クラスタをテストします ([「WebLogic Server クラスタのテスト」\(21 ページ\)](#) を参照)。

2.1 インストールの準備

WebLogic Server をクラスタ内のコンピュータにインストールする前に、システムが以下の設定要件を満たしていることを確認してください。

ディスク容量：アプリケーションサーバーをインストールするパーティションに 10 GB 以上の空きディスク容量があることを確認します。製品のインストールに必要な容量に加えて、環境変数 TEMP または TMP が、最低 500 MB の空きディスク容量がある有効な一時ディレクトリを指している必要があります。ダウンロード可能な実行ファイルに約 500 MB、イメージの展開用にさらに 1.0 GB 必要です。

IP アドレスの設定：すべてのコンピュータに、単一の DNS で管理される固定 IP アドレスが必要です。

IP マルチキャスト：すべてのコンピュータが、IP マルチキャストパケットの転送を完全にサポートしている必要があります。つまり、すべてのルータやその他のトンネル用テクノロジーが、クラスタサーバーインスタンスにマルチキャストメッセージを転送するように設定されている必要があります。ネットワーク遅延は、最低でも、ほとんどのマルチキャストメッセージの最終到達先に 200 ~ 300 ミリ秒以内に到達するレベルにあることが必要です。また、クラスタのマルチキャスト time-to-live (TTL) 値は、マルチキャストパケットが最終到達先に届く前にルータによって破棄されることのないよう、十分に大きい値である必要があります。

バージョン：クラスタ内のすべてのコンピュータは、WebLogic Server ソフトウェアの同一バージョンおよび同一サービスパックを使用している必要があります。

水平クラスタ：水平クラスタ設置では、(つまり WebLogic Server のインスタンスが別々のコンピュータにインストールされている場合は) すべてのコンピュータが同一のネットワークサブネット上にあり、コンピュータの時間が同期されている必要があります ([『LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)』](#) を参照)。

アカウントの権限：(Windows) WebLogic Server は、管理者権限を持つユーザーアカウントでインストールおよび実行する必要があります。

共有ネットワークドライブ：クラスタ内のすべてのコンピュータに読み取りと書き込み権限がある、安全な共有ネットワークドライブを作成しておく必要があります ([『LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)』](#) を参照)。

2.2 WebLogic Server のインストール

以下の手順で、WebLogic Server のインストール方法を詳しく説明します。ここでは、インストールファイルをダウンロードしてインストールディレクトリに抽出済みであること、およびシステム端末を開いて、このディレクトリに移動済みであることを想定しています。

WebLogic Server をインストールするには、次の作業を行います。

- WebLogic Server ソフトウェアをインストールします ([「WebLogic Server ソフトウェアをインストールするには」\(14 ページ\)](#) を参照)
- WebLogic ドメインを作成します ([「WebLogic ドメインを作成するには」\(14 ページ\)](#) を参照)
- (オプション) WebLogic Server を有効にする boot.properties ファイルを作成します。これにより、コマンドラインでユーザー名とパスワードを手動で入力することなく、サーバーを起動できます。

注意： WebLogic をインストールする手順は、すべてのクラスタノードについて同じままです。ただし、クラスタの管理サーバーとして任意のノードを指定できます。WebLogic 管理サーバーは、クラスタの管理サーバーでのみ実行できます。その他のノードでは実行できません。

▶ WebLogic Server ソフトウェアをインストールするには

1. WebLogic Server をインストールするコンピュータに、管理者権限のあるユーザーとしてログオンします。
2. オペレーティングシステムに合わせて、次のいずれかのインストールプログラムを実行します。
 - (32 ビット Windows 上の WebLogic Server 10gR3) server103_win32.exe
 - (64 ビットの Windows、Linux または Solaris 上の WebLogic Server 10gR3) server103_generic.jar

注意： 64 ビットバージョンの WebLogic 10 では、64 ビットの JDK (JRockit Java 6) をダウンロードしてインストールする必要があります。WebLogic 10 用の 64 ビット JDK ファイルは、<http://www.oracle.com/bean/index.html?CNT=index.htm&FP=/content/products/weblogic/jrockit/> で入手できます。Solaris 64-bit JDK の場合、まず [Sun Java SE Downloads](#) から 32-bit JDK (Sun Java 6) をインストールします。32 ビット JDK をインストールしてから、その JDK に対応する 64 ビットパッチをインストールします。

3. インストールウィザードで、デフォルトのオプションをそのまま指定して手順を完了します。
4. インストール完了ウィンドウで、「Run Quickstart」の選択を解除して「Done」をクリックします。
5. クラスタ内のコンピュータごとに、手順 1～4 を繰り返します。

2.3 WebLogic ドメインの作成

独自の WebLogic ドメインを作成し、そのドメインのタイムアウト設定を手動で設定する必要があります。boot.properties ファイルは必要に応じて作成できます。

▶ WebLogic ドメインを作成するには

1. WebLogic Server をインストールしたコンピュータに、管理者権限のあるユーザーとしてログオンします。
2. 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、UNIX) ./config.sh

3. 「Create or Extend a Configuration」画面で、「Create a new WebLogic domain」を選択して「Next」をクリックします。
4. 「Generate a domain configured automatically to support the following BEA products」を選択し、「Next」をクリックします。
5. ユーザー名とパスワードを入力し、確認のためにもう一度パスワードを入力して、「Next」をクリックします。
6. 左側のウィンドウで、「Production Mode」を選択します。
7. 右側のウィンドウで、「BEA Supplied SDKs」と適切なプラットフォームを選択して、「Next」をクリックします。
 - (Solaris) Sun SDK
 - (その他すべてのプラットフォーム) JRockit SDK
8. 「Customize Environment and Services Settings」画面で、「No」を選択して、「Next」をクリックします。
9. ドメイン名を入力して「Create」をクリックします。

注意：ドメインは、すべてのクラスタノードで同じ名前を使用して作成する必要があります。クラスタ内のすべてのコンピュータで同じドメイン名を使用する必要があります。このドキュメントでは、この定義済みのドメイン名を [domain name] と表示します。

10. テキストエディタを開いて次の行を入力します。

```
username=[username from weblogic install]
password=[password from weblogic install]
```

11. このテキストファイルを、[BEA_HOME]/user_projects/domains/[domain name]/boot.properties として保存します。

注意：詳しくは、次のサイトを参照してください。

- (WebLogic Server 10gR3) [Managing Server Startup and Shutdown](#)

12. 「Creating Configuration」画面で、コンフィグレーションの作成が完了したら、「Done」をクリックします。
13. クラスタ内のコンピュータごとに、手順 1 ~ 12 を繰り返します。

▶ クラスタのトランザクションタイムアウトを設定するには

1. WebLogic 管理サーバーのホストコンピュータで、コマンドウィンドウを開いて [BEA_HOME]/user_projects/domains/[domain name]/bin に移動し、次のコマンドを入力して WebLogic Administration Server を起動します。
 - (Windows) startWebLogic.cmd
 - (Linux、UNIX) nohup ./startWebLogic.sh&
2. Web ブラウザの URL 行に `http://[Computer1]:7001/console` と入力して WebLogic Administration Console を起動します。
3. WebLogic Server のインストール時に作成したユーザー名とパスワードを入力してログインします。
4. 管理サーバーで WebLogic Administration Console を起動します。
5. 「Domain Structure」で、[「WebLogic ドメインの作成」\(14 ページ\)](#) で定義したドメイン名をクリックします。
6. Change Center で、「Lock & Edit」をクリックします。
7. 「Configuration」タブをクリックし、「JTA」をクリックします。

8. 「Abandon Timeout Seconds」ボックスに 600 と入力します。
9. 「Save」をクリックし、「Activate Changes」をクリックします。

2.4 WebLogic Server クラスタの作成と設定

WebLogic Server クラスタを設定するには、次の作業を行います。

- クラスタのメンバーを作成します ([「クラスタの作成」\(16 ページ\)](#) を参照)。
- サブレットコンテナ用の認証秘密鍵証明書を設定します ([「サブレットコンテナ用の認証秘密鍵証明書の設定」\(18 ページ\)](#) を参照)。
- クラスタ用のノードマネージャを設定します ([「クラスタ用のノードマネージャの設定」\(19 ページ\)](#) を参照)。
- ノードマネージャおよびクラスタのノードを起動します ([「ノードマネージャおよび管理対象サーバーの起動」\(20 ページ\)](#) を参照)。

2.4.1 クラスタの作成

WebLogic Server クラスタを作成するには、次の作業が必要です。

- コンピュータ (マシン) を追加し、コンピュータごとに WebLogic Node Manager を設定し、マシン上でサーバーインスタンス (サーバー) を作成することによって、クラスタのメンバーを作成します ([「クラスタのメンバーを作成するには」\(16 ページ\)](#) を参照)。
- クラスタを作成して、メンバーを追加します ([「クラスタを作成するには」\(17 ページ\)](#) を参照)。

▶ クラスタのメンバーを作成するには

1. WebLogic Administration Console の「Domain Structure」で、Environment / Machines をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 「New」をクリックし、「Name」ボックスにコンピュータ名を入力し、「Machine OS」リストから適切なオペレーティングシステムを選択して、「OK」をクリックします。「Summary of Machines」画面が表示されます。
4. 作成したマシンの名前をクリックして、「Node Manager」タブをクリックします。
5. 「Listen Address」ボックスに、次のいずれかの値 (マシンのノードマネージャが受信接続をリスンする場所) を入力します。
 - (推奨) ノードマネージャをインストールしたコンピュータの DNS 名
 - ノードマネージャをインストールしたコンピュータの静的 IP アドレス

注意： ノードマネージャがデフォルト以外のポートでリスンしている場合は、そのポートを「Listen Port」ボックスに入力して、「Save」をクリックします。
6. 「Domain Structure」で、Environment / Machines をクリックし、クラスタに追加するクラスタノードごとに手順 6 ~ 8 を繰り返します。クラスタのすべてのマシンに関するノードマネージャを追加および設定したら、手順 10 に進みます。
7. 「Domain Structure」で、Environment / Machines をクリックし、「Summary of Machines」画面で、作成したマシンのうちいずれかのマシン名をクリックします。
8. 「Configuration」タブをクリックし、「Servers」をクリックして、「Add」をクリックします。
9. 「Create a new server and associate it with this machine」を選択し、「Next」をクリックします。

10. 「Server Name」ボックスに、サーバー名 (Server-0 など) を入力します。
11. 「Server Listen Address」ボックスに、次のいずれかの場所 (サーバーが受信接続をリスンする場所) を入力します。
 - コンピュータの静的 IP アドレス
 - コンピュータの DNS 名
12. 「Server Listen Port」に、サーバーの一意のポート番号を入力し、「Finish」をクリックします。

注意： 使用するポート番号を決めるときには、以下の情報を考慮してください。

- 管理対象サーバーが管理サーバーと同じコンピュータ上にある場合は、デフォルト値 7001 を使用しないでください。デフォルト値 7001 は、管理サーバーで必要とされる値です。
 - 水平クラスタでは、WebLogic Server を追加するコンピュータ上で使用可能な任意のポートを使用します。クラスタの他のコンピュータ上にある追加の WebLogic Server インスタンスに同じポート番号を再利用することができます。そうすると、クラスタを簡単に管理できる場合があります。
 - 垂直クラスタでは、クラスタのコンピュータ上にある垂直クラスタの WebLogic Server インスタンスには、それぞれ一意のポート番号を使用します (最初のインスタンスにはポート番号 8001、2 番目のインスタンスにはポート番号 9001 など)。
13. クラスタに追加するサーバーノードごとに、手順 11 ~ 15 を繰り返し、手順 17 に進みます。
 14. Change Center で、「Activate Changes」をクリックします。

▶ クラスタを作成するには

1. WebLogic Administration Console の「Domain Structure」で、Environment / Clusters をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 「Clusters」テーブルで、「New」をクリックして、以下のオプションを設定します。
 - 「Name」ボックスに、クラスタ名 (lc9_cluster など) を入力します。
 - 「Messaging Mode」ボックスで、「Multicast」を選択します。
 - 「Multicast Address」ボックスに、224.0.0.0 ~ 239.255.255.255 の範囲内で IPv4 マルチキャストアドレス (239.192.0.1 など) を入力します。IPv6 ベースのクラスタの場合、有効な IPv6 アドレス (ff01::1 など) を入力します。
 - 「Multicast Port」ボックスに、ポート番号を入力します。有効な値は 1 ~ 65535 です。必要に応じて、このボックスのデフォルト値 7001 を他の有効な値に変更できます。

注意： マルチキャストのアドレスとポートの組み合わせは、LiveCycle ES2 クラスタに対して一意である必要があります (つまり、同じネットワーク上の別のクラスタで、同じアドレスとポートの組み合わせを使用することはできません)。

4. 「OK」をクリックし、「Activate Changes」をクリックします。

▶ サーバーをクラスタに割り当てるには

1. WebLogic Administration Console の「Domain Structure」で、Environment / Clusters をクリックします。
2. 「Cluster」テーブルで、[「クラスタを作成するには」 \(17 ページ\)](#) で作成したクラスタを選択します。
3. Change Center で、「Lock & Edit」をクリックします。
4. 「Configuration」タブで、「Servers」をクリックして、「Add」をクリックします。

5. 「Select a server」 リストから、追加するサーバー名を選択して、「Finish」をクリックします。

注意： AdminServer をクラスタに追加しないでください。

6. クラスタに割り当てるサーバーごとに、手順 4～5 を繰り返します。

7. Change Center で、「Activate Changes」をクリックします。

2.4.2 既存のクラスタへの新しいノードの追加

次の手順を実行して既存のクラスタに新しいノードを追加します。

1. WebLogic Server をインストールします。詳しくは、[「WebLogic Server のインストール」 \(14 ページ\)](#) を参照してください。

2. WebLogic ドメインを作成します。詳しくは、[「WebLogic ドメインの作成」 \(14 ページ\)](#) を参照してください。

警告： 作成する WebLogic ドメインは、ノードを追加する既存のクラスタのドメイン名と同じである必要があります。

3. クラスタの新しいメンバーを作成します。次の操作を実行します。

- 新しいノード (マシン) を管理サーバーに追加します。
- 新しいノードの WebLogic Node Manager を設定します。
- ノードにサーバーインスタンスを作成します。

詳しくは、[「クラスタのメンバーを作成するには」 \(16 ページ\)](#) を参照してください。

4. クラスタに新しいメンバーを追加します。詳しくは、[「サーバーをクラスタに割り当てるには」 \(17 ページ\)](#) を参照してください。

2.4.3 サブレットコンテナ用の認証秘密鍵証明書の設定

ここで、サブレットコンテナ用に認証秘密鍵証明書を設定する必要があります。

▶ サブレットコンテナの認証を変更するには

1. WebLogic 管理サーバーは、クラスタの管理サーバー上で実行してください。

2. コマンドプロンプトを開き、次のスクリプトを実行して環境を設定し、WebLogic スクリプトツールを起動します。

- (Windows) [appserver root]/common/bin/wlst.bat
- (Linux、UNIX) [appserver root]/common/bin/wlst.sh

3. WLST で以下のコマンドを入力し、サブレット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] のデフォルト値は 7001 です。

4. WebLogic 管理サーバーを再起動します。

2.4.4 クラスタ用のノードマネージャの設定

クラスタ用のノードマネージャを設定すると、管理サーバーを使用して、WebLogic Administration Console から、クラスタのノード上で起動、停止、監視およびその他の一般的な作業を行うことができます。次の作業を行います。

- クラスタ用のノードマネージャを設定します ([「クラスタ用のノードマネージャを設定するには」\(19 ページ\)](#) を参照)。
- (水平クラスタのみ) 管理対象サーバーをクラスタのノードマネージャに登録します ([「管理対象サーバーを管理サーバーに登録するには」\(19 ページ\)](#) を参照)。
- (水平クラスタのみ) 相互アクセスできるようにクラスタのサーバーを設定します ([「クラスタのサーバー間で相互アクセスを許可するには:」\(20 ページ\)](#) を参照)。

▶ クラスタ用のノードマネージャを設定するには

1. WebLogic Administration Console の「Domain Structure」で、[「WebLogic ドメインを作成するには」\(14 ページ\)](#) で定義したドメイン名をクリックします。
2. 「Security」タブをクリックして、「General」をクリックし、「Advanced」をクリックして詳細を展開します。
3. Change Center で、「Lock & Edit」をクリックします。
4. 「NodeManager Username」ボックスで、ユーザー名を、ドメイン作成時に確立された値に変更します。
5. 「NodeManager Password」ボックスで、パスワードを、ドメイン作成時に確立された値に変更します。
6. 「Save」をクリックし、「Activate Changes」をクリックします。

▶ 管理対象サーバーを管理サーバーに登録するには

1. クラスタに追加する管理対象のノードで、次のタスクのいずれかを実行します。
 - (Windows) [appserver root]¥common¥bin に移動して、コマンド `wlst.cmd` を入力します。
 - (Linux、UNIX) [appserver root]/common/bin に移動して、コマンド `./wlst.sh` を入力します。

注意： クラスタの管理サーバーとして指定するノードでのみ、WebLogic 管理サーバーを起動します。

2. `wlst` コマンドプロンプトで、次のコマンドを入力して AdminServer に接続します。

```
connect (' [adminusername] ', ' [adminpassword] ', ' [adminserverURL] ')
```

説明：

- [adminusername] は、管理サーバーのユーザー名です。
- [adminpassword] は、管理サーバーユーザーのパスワードです。
- [adminserverURL] には、管理サーバーの URL を `t3://hostname:[port]` の形式で指定します。[port] は、通常は 7001 です。

注意： このコマンドのヘルプを表示するには、`wlst` コマンドプロンプトで `help('connect')` と入力します。

3. 管理サーバーに接続したときに、次のコマンドを入力して、セカンダリコンピュータをクラスタに登録します。

```
nmEnroll (' [appserver domain] ')
```

[appserver domain] は、ローカルコンピュータ上のドメインディレクトリへのパスです。例えば、Windows を稼働していて、デフォルトディレクトリに WebLogic Server がインストールされているコンピュータの場合、パスは `C:/bea/user_projects/domains/[domain name]` となります。

4. コマンド `exit()` を入力して、タスクを完了します。
 5. クラスタのサーバーごとに、手順 1～4 を繰り返します。
- 次の手順は、水平クラスタ内のコンピュータにのみ適用されます。

▶ クラスタのサーバー間で相互アクセスを許可するには：

1. テキストエディタを開き、次の例のように、クラスタ内の各コンピュータの IP アドレスまたはホスト名を 1 行ずつに分けて入力します。

```
localhost  
127.0.0.1  
11.11.11.11  
22.22.22.22
```

2. このファイルを、クラスタ内の各コンピュータで、`nodemanager.hosts` という名前で、次のいずれかの場所に保存します。
 - (Windows) [appserver root]¥common¥nodemanager
 - (Linux、UNIX) [appserver root]/common/nodemanager
3. テキストエディタで、クラスタ内の任意のコンピュータの既存のホストファイルを、次のいずれかの場所から開きます。
 - (Windows) C:¥WINDOWS¥system32¥drivers¥etc
 - (Linux、UNIX) /etc
4. クラスタ内のすべてのコンピュータの IP アドレスおよびホスト名を追加します。
5. 編集したファイルを、クラスタ内の各コンピュータの、次のいずれかの場所に保存します。
 - (Windows) C:¥WINDOWS¥system32¥drivers¥etc
 - (Linux、UNIX) /etc

2.4.5 ノードマネージャおよび管理対象サーバーの起動

ノードマネージャと、クラスタの管理対象サーバーを起動するには、次の手順を使用します。

▶ ノードマネージャを起動するには

クラスタの各コンピュータで、コマンドウィンドウを開き、次の適切なディレクトリに移動します。

- (Windows) [appserver root]¥server¥bin に移動して、次のコマンドを入力します。

```
startNodeManager.cmd
```
- (Linux、UNIX) [appserver root]/server/bin に移動して、次のコマンドを入力します。

```
nohup ./startNodeManager.sh&
```

注意：バックグラウンドで実行するには、`startNodeManager.cmd` を Windows サービスとしてインストールする必要があります ([「Using Node Manager」](#) を参照)。

▶ 管理対象サーバーを起動するには：

1. WebLogic Administration Console の「Domain Structure」で、Environment / Clusters をクリックします。
2. 起動するクラスタの名前をクリックします。

3. 「Control」 タブをクリックし、各サーバーのチェックボックスを選択して、「Start」 をクリックします。
4. 「Yes」 をクリックして、サーバーを起動することを確認します。

2.5 WebLogic Server クラスタのテスト

WebLogic Server クラスタをテストして、すべてのメンバーがアクティブであること、クラスタが設計に従って動作することを確認できます。WebLogic Server クラスタが正常に機能することを確認してから、LiveCycle ES2 のインストールおよび設定に進む必要があります。

► WebLogic Server クラスタをテストするには

1. クラスタのすべての WebLogic Server インスタンスが起動していることを確認します。
2. [appserverdomain]/servers/[server name]/logs/[server name].log にある server.log ファイルを表示します。次のようなメッセージで、クラスタのアクティブなメンバーを確認できます。

```
####<Apr 10, 2008 1:58:20 PM CDT> <Info> <Cluster> <hostname> <hostname>  
<[ACTIVE] ExecuteThread: '1' for queue: 'weblogic.kernel.Default  
(self-tuning)'> <<WLS Kernel>> <> <> <1207853900703> <BEA-000111> <Adding  
hostname with ID -5030984338025399S:hostname: [7002,7002,-1,-1,-1,-1,-1]:  
[ドメイン名]:hostname to cluster: lc9_cluster view.>
```

2.6 次の手順

次に、LiveCycle ES2 ソリューションコンポーネントファイルをインストールする必要があります ([「LiveCycle ES2 モジュールのインストール」 \(22 ページ\)](#) を参照)。

3

LiveCycle ES2 モジュールのインストール

ここでは、Windows、Linux および Solaris 上で LiveCycle ES2 インストールプログラムを実行している LiveCycle ES2 システムを設定する最初のフェーズについて説明します。後続のフェーズでは、LiveCycle Configuration Manager を実行して、LiveCycle ES2 の設定およびデプロイを行います。

モジュールをインストールする前に、LiveCycle ES2 の実行に必要なソフトウェアとハードウェアが使用環境に含まれていることを確認してください。また、各インストールオプションについて理解し、必要に応じて環境を整えておく必要があります（『[LiveCycle ES2 のインストールの準備（サーバークラス）](#)』を参照）。

LiveCycle ES2 には、インストールプログラム用にコマンドラインインターフェイス（CLI）も用意されています。CLI の使用方法について詳しくは、『[付録 - コマンドラインインターフェイスのインストール](#)』（95 ページ）を参照してください。LiveCycle Configuration Manager でも CLI を使用できます。『[付録 - LCM コマンドラインインターフェイス](#)』（99 ページ）を参照してください。CLI は、LiveCycle ES2 の上級ユーザーが使用したり、インストールプログラムや LiveCycle Configuration Manager でグラフィカルユーザーインターフェイスがサポートされていないサーバー環境で使用したりすることを前提としています。

この章には、以下のトピックが含まれています。

- [「インストーラの確認」](#)（22 ページ）
- [「製品ファイルのインストール」](#)（23 ページ）
- [「エラーログの表示」](#)（26 ページ）
- [「LiveCycle ES2 によってインストールされた JAR ファイルの設定」](#)（26 ページ）
- [「キャッシュロケータの設定（TCP を使用するキャッシュのみ）」](#)（26 ページ）
- [「フォントディレクトリの設定」](#)（29 ページ）

3.1 インストーラの確認

インストールプロセスを開始する前に、インストーラファイルについて、次のベストプラクティスを確認してください。

▶ DVD インストールメディアの確認：

入手したインストールメディアが破損していないことを確認します。LiveCycle ES2 をインストールするコンピュータのハードディスクにインストーラのメディアコンテンツをコピーする場合は、必ず、すべての DVD コンテンツをハードディスクにコピーしてください。インストールエラーを避けるには、Windows のパスの最大長を超えるディレクトリパスに DVD インストールイメージをコピーしないでください。また、ローカルパスに番号記号（#）などの特殊文字を使用しないでください。ローカルパスに番号記号を使用した場合、適切なライセンス情報がインストール中に表示されないことがあります。

▶ ダウンロードしたファイルの確認：

アドビの Web サイトからインストーラをダウンロードした場合は、MD5 チェックサムを使用してインストーラファイルの整合性を検証してください。次のいずれかを実行し、ダウンロードファイルの MD5 チェックサムを計算して、アドビのダウンロード用 Web ページで公開されているチェックサムと比較します。

- Linux：md5sum コマンドを実行する。
- Solaris：Solaris で digest コマンドを実行する。
- Windows：WinMD5 などのツールを使用する。

▶ ダウンロードしたアーカイブファイルの展開：

アドビの Web サイトから ESD をダウンロードしたら、アーカイブファイルである [appserver]_DVD.zip (Windows) または [appserver]_DVD._unix.tar.gz (Linux または Solaris) 全体を、コンピュータに展開します。Solaris の場合は、gunzip を使用して .gz ファイルを展開します。

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。

3.2 製品ファイルのインストール

正常にインストールするには、インストールディレクトリに対する読み取りと書き込み権限が必要です。デフォルトのインストールディレクトリは以下のとおりですが、必要に応じて、別のディレクトリを指定することもできます。

- (Windows) C:\Adobe\Adobe LiveCycle ES2\
- (Linux または Solaris) /opt/adobe/adobe_lifecycle_es2/

LiveCycle ES2 のインストールパスに国際文字が含まれ、システムに UTF-8 ロケールが設定されていない場合、LiveCycle ES2 では、国際文字を使用した [LiveCycleES2 root] 内のフォントインストールディレクトリを認識しません。このような問題を回避するには、UTF-8 ロケールセットを使用して新しいフォントディレクトリを作成してから、LiveCycle Configuration Manager を UTF-8 ロケールで実行します。このとき、ConfigurationManager.bat または ConfigurationManager.sh スクリプトに `-Dfile.encoding=utf8` 引数を追加します。

警告：LiveCycle ES2 をインストールするときに、インストールパスに 2 バイト文字または拡張ラテン文字 (€ など) を使用しないでください。

Linux でのインストールでは、インストールプログラムにより、ログインしているユーザーのホームディレクトリがファイルを格納するための一時ディレクトリとして使用されます。そのため、次のようなメッセージがコンソールに表示される場合があります。

```
WARNING: could not delete temporary file /home/<username>/ismp001/1556006
```

インストールが完了したら、一時ファイルを手動で削除する必要があります。

警告：オペレーティングシステムの一時的ディレクトリが、『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』に記載されている最小要件を満たしていることを確認してください。一時ディレクトリは次のいずれかの場所にあります。

- (Windows) 環境変数で設定されている TMP または TEMP パス
 - (Linux または Solaris) ログインユーザーのホームディレクトリ
- UNIX 系のシステムでは、root 以外のユーザーは次のディレクトリを一時的ディレクトリとして使用できます。
- (Solaris) /var/tmp

モジュールを UNIX 系のインストールする際に、デフォルトのインストール先である /opt/adobe/adobe_lifecycle_es2 に正常にインストールするには、ルートユーザーでログインする必要があります。ルートユーザー以外でログインした場合は、権限 (読み取り、書き込み、実行権限) を持っている別のディレクトリにインストール先を変更してください。例えば、ディレクトリを /home/[username]/adobe_lifecycle_es2 に変更します。

Windows では、LiveCycle ES2 のインストールには管理者権限が必要です。

一時ファイルは、システムのデフォルトの一時ディレクトリまたは指定したディレクトリに生成されます。生成された一時ファイルが、インストーラの終了後も残る場合があります。これらのファイルは手動で削除することができます。

Windows の場合は、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することにより、インストールに要する時間が短縮されます。

3.2.1 Linux または UNIX にインストールするための Windows ステージングプラットフォームへのインストール

Linux または UNIX プラットフォームにデプロイするために、LiveCycle ES2 を Windows にインストールして設定することができます。この機能を使用して、ロックダウンされた Linux または UNIX 環境にインストールできません。例えば、ロックダウンされた環境にはグラフィカルユーザーインターフェイスはインストールされていません。

Windows でインストールプログラムを実行するときに、LiveCycle ES2 をデプロイするターゲットプラットフォームとして Linux または UNIX オペレーティングシステムを選択することができます。インストールプログラムにより、Linux または Solaris のバイナリがインストールされます。このバイナリは、製品を設定する際にも LiveCycle Configuration Manager で使用されます。

その後、Windows を実行するコンピュータを、デプロイ可能なオブジェクトのステージング場所として使用できます。これらのオブジェクトは、アプリケーションサーバーへのデプロイメント用に Linux または UNIX コンピュータにコピーできます。対象とするアプリケーションサーバーは、オペレーティングシステムに関係なく、インストールおよび設定時に選択したものと同一である必要があります。

3.2.2 JAVA_HOME 環境変数の設定

JAVA_HOME 環境変数は、ご使用のアプリケーションサーバーの Java SDK を指している必要があります。『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』の「サポートされているソフトウェア」の表を参照してください。

3.2.3 インストール LiveCycle ES2

ここでは、LiveCycle ES2 製品ファイルの初期インストールについて説明します。設定およびデプロイメントについて詳しくは、[「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照してください。

注意： デプロイメントの際に権限の問題を回避するため、アプリケーションサーバーを実行する場合と同じユーザーで、LiveCycle ES2 インストーラおよび LiveCycle Configuration Manager を実行してください。

▶ LiveCycle ES2 のインストール：

1. JAVA_HOME 環境変数が、Sun Java (Solaris) または JRockit (Linux/UNIX) JDK がインストールされているディレクトリに設定されていることを確認します。
2. インストールプログラムを起動します。
 - (Windows) 次のいずれかを実行します。
 - インストールメディアの livecycle_server ディレクトリまたはハードディスク上のコピー先フォルダに移動して、run_windows_installer.bat ファイルを起動します。バッチファイルによって、Windows のバージョンに適したインストーラ (32 ビット版または 64 ビット版) が起動されます。

- インストールメディアまたはハードディスク上にあるインストーラのコピー先フォルダ内で、適切なディレクトリに移動して、install.exe ファイルをダブルクリックします。
 - (Windows 32 ビット版) %livecycle_server%9.0%Disk1%InstData%Windows%VM
 - (Windows 64 ビット版) %livecycle_server%9.0%Disk1%InstData%Windows_64bit%VM
 - (Linux、Solaris) 適切なディレクトリに移動して、コマンドプロンプトで ./install.bin と入力します。
 - (Linux) /livecycle_server/9.0/Disk1/InstData/Linux/NoVM
 - (Solaris) /livecycle_server/9.0/Disk1/InstData/Solaris/NoVM
- 注意：** UNIX または Linux にインストールするが、リリース DVD からは直接インストールしない場合は、インストールファイルに実行権限を設定します。
3. プロンプトが表示されたら、インストールプログラムで使用する言語を選択して、「OK」をクリックします。
 4. はじめに画面で「次へ」をクリックします。
 5. インストーラを実行するコンピュータに、LiveCycle ES (8.x) の以前のバージョンがインストールされている場合は、アップグレードの準備画面が表示されます。LiveCycle ES2 へのアップグレードの準備をしたり、LiveCycle ES2 の新規インストールを実行したりできます。「次へ」を選択して、LiveCycle ES2 のインストールを続行します。
 - **既存のインストールを LiveCycle ES2 v9.0 にアップグレードする準備：** インストールプログラムでは、既存の LiveCycle ES (8.x) インストールに基づいて、LiveCycle ES2 へのアップグレードで必要となるデータを準備します。このオプションを選択した場合、インストール後に、LiveCycle 管理コンソールでモジュールライセンスを更新してください。
 6. **LiveCycle ES2 v9.0 をインストール：** インストールプログラムは、LiveCycle ES2 をインストールします。インストールフォルダを選択画面で、表示されたデフォルトのディレクトリをそのまま使用するか、「選択」をクリックして LiveCycle ES2 のインストール先ディレクトリを選択してから、「次へ」をクリックします。
存在しないディレクトリの名前を入力すると、そのディレクトリが作成されます。
- 警告：** Linux または UNIX にインストールする場合は、指定するディレクトリ名にスペースを含めないでください。スペースを含めると、モジュールがインストールされません。
7. **(Windows のみ、および手動インストールが選択されている場合)** 手動インストールオプション画面で、目的のデプロイメントオプションを選択し、「次へ」をクリックします。
 - **Windows (ローカル)：** ローカルサーバーに LiveCycle ES2 をインストールしデプロイする場合は、このオプションを選択します。
 - **ステージ化 (リモートシステムを対象としている Windows にインストールされました)：** デプロイメント用のステージングプラットフォームとして Windows を使用する場合は、このオプションを選択します。その後で、リモートサーバー上のターゲットオペレーティングシステムを選択します。Windows 上でインストールを行っている場合でも、デプロイメント対象として UNIX オペレーティングシステムを選択できます ([「Linux または UNIX にインストールするための Windows ステージングプラットフォームへのインストール」 \(24 ページ\)](#) を参照)。
- 注意：** 保護されたデータソースが WebLogic で使用された場合、adobe-livecycle-weblogic.ear ファイルと adobe-contentservices.ear ファイルは、リモートマシンにデプロイできません。詳しくは、[TechNote](#) を参照してください。
8. LiveCycle ES2 サーバーの使用許諾契約書を読み、「同意します」を選択して使用許諾契約書の条件に同意し、「次へ」をクリックします。使用許諾契約書に同意しない場合は、操作を継続することはできません。
 9. プリインストールの概要画面で、詳細を確認して「インストール」をクリックします。インストールプログラムによりインストールの進行状況が表示されます。

10. リリースノートの情報を確認して「次へ」をクリックします。
11. インストール完了画面で詳細を確認します。
12. 「LiveCycle Configuration Manager を起動」チェックボックスはデフォルトで選択されています。「完了」をクリックして LiveCycle Configuration Manager を実行します。

注意： LiveCycle Configuration Manager を後で実行するには、「完了」をクリックする前に、「LiveCycle Configuration Manager を起動します」オプションの選択を解除します。[LiveCycleES2 root]/configurationManager/bin ディレクトリにある該当するスクリプトを使用して、LiveCycle Configuration Manager を後で起動することができます。[「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照してください。

注意： (PDF Generator ES2 または PDF Generator 3D ES2 Windows 版のみ) クラスタ内のすべてのノードに Acrobat 9.2 がインストールされていない場合は、今すぐインストールしてください。その後で、[「Acrobat Professional の設定」\(62 ページ\)](#) の手順を実行します。

3.3 エラーログの表示

インストール中にエラーが発生した場合は、Adobe_LiveCycle_ES2_InstallLog.log ファイルが作成され、エラーメッセージが記録されます。このログファイルは、[LiveCycleES2_root]/log ディレクトリに作成されます。

3.4 LiveCycle ES2 によってインストールされた JAR ファイルの設定

LiveCycle ES2 によってインストールされた次の JAR ファイルを、WebLogic Server の特定のディレクトリにコピーする必要があります。

- pop3.jar ファイル：このファイルを、クラスタ内のすべてのノードで同じ場所にコピーします。
- データベースドライバ JAR ファイル：使用している LiveCycle ES2 データベースに固有のデータベースドライバファイルをコピーします。

▶ pop3.jar ファイルのコピー：

1. idplib という名前のディレクトリを、[appserver domain] に作成します。
2. pop3.jar ファイルを、[LiveCycleES2 root]/lib/weblogic から WebLogic ノードの [appserver domain]/idplib にコピーします。

▶ 必要なデータベースドライバのコピー：

1. [LiveCycleES2 root]/lib/db でデータベースドライバを探します。
2. 適切なデータベース JAR ファイルを WebLogic ノードの [appserver domain]/idplib にコピーします。

3.5 キャッシュロケータの設定 (TCP を使用するキャッシュのみ)

TCP を使用して LiveCycle ES2 クラスタのキャッシュを実装する場合、LiveCycle ES2 クラスタの他のメンバーを検索するために TCP ロケータを設定します。

注意： このセクションは、UDP を使用して LiveCycle ES2 クラスタのキャッシュを実装する場合は適用されません。UDP を使用して LiveCycle ES2 クラスタのキャッシュを設定するには、[「サーバーの開始引数の設定」\(44 ページ\)](#) を参照してください。

TCP を使用して LiveCycle ES2 クラスタのキャッシュを有効にするには、次の作業を行う必要があります。

- TCP ロケータがインストールおよび設定されていることを確認します。TCP ロケータは、LiveCycle ES2 のインストール時に、デフォルトの設定で、[LiveCycle ES2 root]/lib/caching ディレクトリにインストールされます。デフォルトの設定は変更することができます ([「TCP ロケータの変更」\(27 ページ\)](#) を参照)。
- ロケータを使用するように LiveCycle ES2 クラスタ内の各ノードを設定します ([「サーバーの開始引数の設定」\(44 ページ\)](#) を参照)。
- TCP ロケータが実行されていることを確認します ([「TCP ロケータの起動」\(28 ページ\)](#) を参照)。

3.5.1 TCP ロケータの変更

LiveCycle ES2 インストーラによって、変更せずに使用できる TCP ロケータのデフォルト設定が作成されます。ロケータをネットワーク上の任意のコンピュータに移動し、そのコンピュータで実行できます。ロケータが存在するコンピュータは、LiveCycle ES2 クラスタのメンバーである必要はありません。クラスタで高可用性をサポートするために、フェイルオーバーロケータを追加作成することもできます ([「TCP ロケータをインストールするには：」\(27 ページ\)](#) を参照)。

TCP ロケータを変更して、デフォルトのポート (22345) 以外のポートを使用することもできます ([「デフォルトのロケータポートを変更するには \(Windows\) :」\(27 ページ\)](#) または [「デフォルトのロケータポートを変更するには \(UNIX\) :」\(28 ページ\)](#) を参照)。

▶ TCP ロケータをインストールするには：

1. LiveCycle ES2 をインストールしたコンピュータにログオンし、[LiveCycle ES2 root]/lib/caching ディレクトリに移動します。
2. caching ディレクトリとその内容を、ロケータを実行するコンピュータにコピーします。

▶ デフォルトのロケータポートを変更するには (Windows)：

1. startlocator.bat ファイルをテキストエディタで開きます。デフォルトインストールの startlocator ファイルは、LiveCycle ES2 をインストールしたコンピュータの [LiveCycle ES2 root]/lib/caching ディレクトリにあります。
2. 次のプロパティで、デフォルトのポート番号 (22345) を任意のポート番号に変更します。

```
set port=22345
```

ポート番号は、1 ~ 65535 の使用可能なポートにすることができます。

警告：ここで設定するポート番号が、LiveCycle ES2 クラスタの各ノードの JVM 引数で設定されるポート番号と一致することを確認します ([「サーバーの開始引数の設定」\(44 ページ\)](#) を参照)。

3. (複数のネットワークカードがあるコンピュータのみ) ロケータをホストするコンピュータに複数のネットワークカードがある場合は、スクリプト内の次のプロパティを設定します。

```
set bindaddr=<bind IP address>
```

<bind IP address> は、ロケータがリスンする IP アドレスです。LiveCycle ES2 クラスタ内の各ノードで JVM 引数 adobe.cache.cluster-locators の <bind IP address> を指定する必要があります。

[「サーバーの開始引数の設定」\(44 ページ\)](#) を参照してください。

注意：startlocator スクリプトにバインドアドレスとバインドポートを指定しない場合、スクリプトの実行時にこれらの値を入力するよう求められます。ただし、IPv6 の場合は、startlocator スクリプト自体にバインドアドレスとバインドポートを指定する必要があります。

4. 編集したファイルを保存します。
5. LiveCycle ES2 クラスタのその他のロケータで、手順 1～4 を繰り返します。

▶ デフォルトのロケータポートを変更するには (UNIX) :

1. startlocator.sh ファイルをテキストエディタで開きます。デフォルトインストールの startlocator ファイルは、LiveCycle ES2 をインストールしたコンピュータの [LiveCycle ES2 root]/lib/caching ディレクトリにあります。
2. 次のプロパティで、デフォルトのポート番号 (22345) を任意のポート番号に変更します。

```
GF_PORT=22345
```

ポート番号は、1～65535 の使用可能なポートにすることができます。

警告：ここで設定するポート番号が、LiveCycle ES2 クラスタの各ノードの JVM 引数で設定されるポート番号と一致することを確認します。

[「サーバーの開始引数の設定」\(44 ページ\)](#) を参照してください。

3. (複数のネットワークカードがあるコンピュータのみ) ロケータをホストするコンピュータに複数のネットワークカードがある場合は、次の引数を変更します。

```
GF_BIND_ADDRESS="<bind IP address>"
```

<bind IP address> は、ロケータがリスンする IP アドレスです。LiveCycle ES2 クラスタ内の各ノードで JVM 引数 `adobe.cache.cluster-locators` の `<bind IP address>` を指定する必要があります。

[「サーバーの開始引数の設定」\(44 ページ\)](#) を参照してください。

注意：IPv6 の場合は、startlocator スクリプト自体にバインドアドレスとバインドポートを指定することをお勧めします。

4. 編集したファイルを保存します。
5. LiveCycle ES2 クラスタのその他のロケータで、手順 1～4 を繰り返します。

3.5.2 TCP ロケータの起動

クラスタを起動する前に TCP ロケータを起動する必要があります。LiveCycle ES2 クラスタのメンバーを起動するときに TCP ロケータが実行されていない場合、キャッシュは機能しません。

▶ TCP ロケータを起動するには

1. TCP ロケータがインストールされているコンピュータで、caching ディレクトリに移動します。デフォルトインストールでは、TCP ロケータは、LiveCycle ES2 をインストールしたコンピュータの [LiveCycle ES2 root]/lib/caching ディレクトリにインストールされています。
2. (IPv6 のみ) startlocator.bat (Windows) または startlocator.sh (UNIX) を変更して、次の JVM 引数を追加します。

```
-Djava.net.preferIPv6Stack=true  
-Djava.net.preferIPv6Addresses=true
```

3. 適切なファイルを実行します。
 - (Windows) startlocator.bat
 - (UNIX) startlocator.sh
4. LiveCycle ES2 クラスタのその他のロケータで、手順 1～3 を繰り返します。

▶ TCP ロケータを停止するには

1. TCP ロケータがインストールされているコンピュータで、caching ディレクトリに移動します。デフォルトインストールでは、TCP ロケータは、LiveCycle ES2 をインストールしたコンピュータの [LiveCycle ES2 root]/lib/caching ディレクトリにインストールされています。
2. 適切なファイルを実行します。
 - (Windows) stoplocator.bat
 - (UNIX) stoplocator.sh
3. LiveCycle ES2 クラスタのその他のロケータで、手順 1～3 を繰り返します。

注意： startlocator スクリプトおよび説明した特定の IP アドレスやポートの値にデフォルト値を使用していない場合は、stoplocator スクリプトの値と同じ値を指定します。このように指定しないと、stoplocator スクリプトはロケータの停止に失敗します。

3.6 フォントディレクトリの設定

クラスタ内の各ノードに、[LiveCycleES2 root]¥fonts ディレクトリにインストールされている LiveCycle ES2 のフォントを含めて、フォントディレクトリを設定する必要があります。

フォントはクラスタ内の各ノードで同じパスに存在する必要があり、フォントディレクトリのコンテンツもクラスタ内のすべてのノードで同一になっている必要があります。このように設定するには、次のいずれかの操作を行います。

- クラスタ内のすべてのノードがアクセスできる共有ディレクトリを使用します。
- 同じパスを使用して、クラスタ内の各ノードに [LiveCycle ES2 root]¥fonts ディレクトリをコピーします。

これらの共有ディレクトリを作成した場所を記録しておき、後で LiveCycle Configuration Manager を使用して LiveCycle ES2 を設定するときに使用できるようにします。

警告： フォントディレクトリは、GDS ディレクトリと別にする必要があります。ただし、単一の共有親ディレクトリの個別の兄弟サブディレクトリとして存在させることができます。

3.7 次の手順

9.5 の新機能

デプロイする LiveCycle ES2 を設定する必要があります ([「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照)。LiveCycle ES2.5 Solution Accelerator をインストールする予定がある場合は、LiveCycle ES2 を後から設定することもできます。その場合は、まず LiveCycle ES2 サービスパック 2 以降を適用してから LiveCycle ES2.5 Solution Accelerator をインストールする必要があります。

Solution Accelerator のインストールについて詳しくは、『[LiveCycle ES2.5 Solution Accelerator のインストールおよびデプロイ](#)』を参照してください。

4

LiveCycle ES2 をデプロイするための設定

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。

- アプリケーションサーバーにデプロイするため、EAR ファイルに LiveCycle ES2 モジュールを設定する
- LiveCycle ES2 をサポートするようにアプリケーションサーバーのプロパティを設定する
- アプリケーションサーバーの設定を検証する
- JDBC モジュールを LiveCycle ES2 EAR ファイルにパッケージ化する（データソースをセキュリティで保護）
- LiveCycle ES2 EAR ファイルをデプロイする
- LiveCycle ES2 のデータベースを初期化する
- LiveCycle ES2 コンポーネントをデプロイする
- LiveCycle ES2 コンポーネントを設定する
- LiveCycle ES2 コンポーネントのデプロイメントを検証する
- (オプション) LiveCycle ES2 の Connectors for ECM、Reader Extensions ES2、PDF Generator ES2 および PDF Generator 3D ES2 モジュールを設定する

注意： (オプション) LiveCycle Configuration Manager では、手動インストールによる LiveCycle Business Activity Monitoring ES2 の設定、デプロイメントおよびデータベースの初期化をサポートしていません。BAM をインストールする場合は、使用しているアプリケーションサーバー版の『LiveCycle ES2 のインストールの準備 (シングルサーバー)』ガイドに記載されている「LiveCycle Business Activity Monitoring ES2 の設定」を参照してください。

- (オプション) LiveCycle ES2 のサンプルを LiveCycle ES2 に読み込む

4.1 LiveCycle Configuration Manager の概要

LiveCycle Configuration Manager は、LiveCycle ES2 コンポーネントをアプリケーションサーバーにデプロイするために必要な設定、デプロイおよび検証を行うためのウィザード型のツールです。必要に応じて、LiveCycle Configuration Manager を使用して、アプリケーションサーバーを設定し、製品の EAR ファイルをアプリケーションサーバーにデプロイできます。

LiveCycle Configuration Manager は、LiveCycle ES2 のインストールプログラムの実行時に、モジュールのファイルと共にインストールされます。LiveCycle Configuration Manager の実行時には、設定する LiveCycle ES2 モジュールおよび LiveCycle Configuration Manager で実行するタスクを指定します。

LiveCycle Configuration Manager は、インストールプログラムから実行するか、インストール後にいつでも実行することができます。アプリケーションサーバーの設定またはアプリケーションサーバーへのデプロイを行うために LiveCycle Configuration Manager を使用する場合は、そのアプリケーションサーバーを起動しておく必要があります。

別のコンピュータにインストールされているアプリケーションサーバーも設定できます。ただし、その場合は、そのアプリケーションサーバーのライブラリファイルを LiveCycle Configuration Manager で使用できるようにするために、LiveCycle Configuration Manager を実行しているコンピュータにもアプリケーションサーバーをインストールする必要があります（そのアプリケーションサーバーを実行する必要はありません）。

4.1.1 LiveCycle Configuration Manager の CLI バージョンと GUI バージョン

ここでは、LiveCycle Configuration Manager の GUI バージョンについて説明します。LiveCycle Configuration Manager のコマンドラインインターフェイス (CLI) バージョンの使用方法については、「[付録 - LCM コマンドラインインターフェイス](#)」(99 ページ) を参照してください。

LiveCycle の設定は、GUI や CLI モードで LCM を使用したり、手動モードを利用して実行できます。次の表は、設定手順とそれらの手順に対応する有効なモード (GUI、CLI、手動) をまとめたものです。

| LiveCycle ES2 の設定タスク | LCM GUI | LCM CLI | 手動タスク (LCM 以外) |
|---|---------|---------|----------------|
| LiveCycle ES2 を設定する | 可 | 可 | 不可 |
| アプリケーションサーバーを設定する | 可 | 可 | 可 |
| アプリケーションサーバーの設定を検証する | 可 | 可 | 可 |
| JDBC モジュールを LiveCycle ES2 EAR にパッケージする | 可 | 不可 | 可 |
| LiveCycle ES2 EAR をデプロイする | 可 | 可 | 可 |
| LiveCycle ES2 のデータベースを初期化する | 可 | 可 | 不可 |
| Business Activity Monitoring ES2 を初期化する | 可 | 可 | 可 |
| LiveCycle ES2 サーバー接続を検証する | 可 | 可 | 不可 |
| LiveCycle ES2 コンポーネントをデプロイする | 可 | 可 | 不可 |
| LiveCycle ES2 コンポーネントのデプロイメントを検証する | 可 | 可 | 可 |
| LiveCycle コンポーネントを設定する (次のタスクを含む) <ul style="list-style-type: none"> ECM コネクタを設定する PDF Generator ES2 を設定する Reader Extensions ES2 を設定する | 可 | 可 | 可 |
| サンプルを読み込む | 可 | 可 | 可 |

4.2 JDBC ドライバの場所

設定では、データベースの JDBC ドライバの場所を指定する必要があります。Oracle、SQL Server、DB2 の各ドライバは、[LiveCycle ES2 root]/lib/db/[database] ディレクトリにあります。

4.3 LiveCycle ES2 の設定、デプロイ

LiveCycle Configuration Manager の実行時には、自動的に実行するタスクを選択することができます。

警告：デュアルスタックマシン (IPV6 と IPV4 のサポート) 上で LiveCycle ES2 Configuration Manager を実行する前に、管理サーバー、ノードマネージャおよび管理対象サーバーのリスンアドレスの割り当てが完了していることを確認します。この操作を行っていない場合は、リスンアドレスを割り当てた後、それぞれを再起動します。

注意： LiveCycle Configuration Manager によるリモートサーバーへの LiveCycle ES2 モジュールのデプロイは、ノードで管理されるアプリケーションサーバーでのみサポートされ、スタンドアロンのアプリケーションサーバーではサポートされません。

ヒント： LiveCycle Configuration Manager では、各画面で「次へ」をクリックしたときに、その画面で指定した値が検証されます。検証できない値があると、警告メッセージが表示され、画面上のプロパティは赤色で表示されて、有効な値を入力するまで次の操作に進めなくなります。

ヒント： 前回の実行後、LiveCycle Configuration Manager をもう一度実行している場合、既に設定されているパラメータが表示されますが、編集はできません。これらのフィールドを編集可能にして、変更を行うには「設定を編集」をクリックします。

LiveCycle Configuration Manager によって LiveCycle ES2 EAR ファイルが設定された後、次のファイルが [LiveCycleES2 root]/configurationManager/export ディレクトリに配置されます。

- adobe-livecycle-native-weblogic-[OS].ear
- adobe-livecycle-weblogic.ear
- adobe-workspace-client.ear (if you installed LiveCycle Process Management ES2)
- adobe-contentservices.ear (LiveCycle Content Services ES2 をインストールしている場合)

LiveCycle Configuration Manager を使用して EAR ファイルをデプロイする場合、LiveCycle Configuration Manager が EAR ファイルにアクセスしてアプリケーションサーバーに自動的にデプロイします。LiveCycle ES2 EAR ファイルを手動でデプロイする場合は、このディレクトリ内のファイルにアクセスしてアプリケーションサーバーにデプロイすることができます。

LiveCycle Configuration Manager で LiveCycle ES2 EAR ファイルを設定した後で、EAR ファイルの名前をカスタムファイル名に変更できます (例えば、バージョン ID や、ローカル JDK 環境の標準的な操作に必要な情報をファイル名に指定できます)。

LiveCycle Configuration Manager では、カスタムファイル名を持つ EAR ファイルのデプロイまたはデプロイ解除をサポートしていません。EAR ファイルがカスタムファイル名を使用している場合は、アプリケーションサーバーに手動でデプロイまたはデプロイ解除する必要があります (例えば、最初にカスタム名を持つ EAR ファイルをデプロイしていて、後からサービスパックやバッチなどの変更を適用する場合)。

リモートアプリケーションサーバーを設定する場合は、そのアプリケーションサーバーのライブラリファイルを LiveCycle Configuration Manager で使用できるようにするために、LiveCycle Configuration Manager のコンピュータにもアプリケーションサーバーがインストールされていることを確認してください。

ヒント： LiveCycle Configuration Manager 画面でディレクトリやファイルの名前を選択したり、参照したりしているときに、フォントを上書きできます。次の JVM 引数を ConfigurationManager.bat (Windows) または ConfigurationManager.sh (Linux、UNIX) に追加してください。

```
-Dlcm.font.override=<FONT_FAMILY _NAME>
```

例えば、-Dlcm.font.override=SansSerif のように指定します。

▶ LiveCycle Configuration Manager を使用した設定：

ヒント： LiveCycle Configuration Manager で F1 キーを押すと、表示されている画面のヘルプ情報が表示されます。このヘルプには、このドキュメントに含まれていない詳細な情報や、LiveCycle Configuration Manager の各画面に固有の情報が表示されます。

ヒント： 前回の実行後、LiveCycle Configuration Manager をもう一度実行している場合、既に設定されているパラメータが表示されますが、編集はできません。これらのフィールドを編集可能にして、変更を行うには「設定を編集」をクリックします。

注意： LiveCycle Configuration Manager を使用して、IPv6 ベースのクラスタの設定を行うことはできません。

IPv6 ベースの WebLogic クラスタを使用している場合は、[「WebLogic Server クラスタの手動設定」\(43 ページ\)](#) を参照してください。

1. LiveCycle Configuration Manager がインストールプログラムで自動的に起動しない場合は、[LiveCycleES2 root]/configurationManager/bin ディレクトリに移動し、適切なコマンドを入力して起動します。
 - (Windows) ConfigurationManager.bat
 - (Windows 以外) /ConfigurationManager.sh
2. 言語を選択するよう求められた場合は、選択して「OK」をクリックします。
3. ようこそ画面で「次へ」をクリックします。
4. アップグレードタスクの選択画面で、オプションが選択されていないことを確認し、「次へ」をクリックして続行します。

警告： 既存の LiveCycle ES インストールをアップグレードする場合は、この手順を続行しないでください。アップグレードの情報と手順については、LiveCycle ES2 の現在のバージョン用の『[LiveCycle ES 8.x から LiveCycle ES2 へのアップグレードの準備](#)』ガイドと、使用しているアプリケーションサーバー版の [LiveCycle ES2 へのアップグレードガイド](#) を参照してください。

5. モジュールの選択画面で、LiveCycle ES2 モジュールを選択し、「次へ」をクリックします。
6. タスク選択画面で、実行するすべてのタスクを選択し、「次へ」をクリックします。

注意： データソースを保護するには、「JDBC モジュールを LiveCycle ES2 EAR にパッケージ (データソースをセキュリティで保護)」を選択します。ただし、XML フォームを処理するために LiveCycle ES2 の実装が必要な場合は、このタスクを選択しないでください。または、[この Technote](#) の手順を実行して、WebLogic アプリケーションサーバー上の JNDI アーティファクトへのアクセスを保護します。

アプリケーションサーバーを設定して、LiveCycle Configuration Manager を使用して自動で LiveCycle ES2 をデプロイする予定がない場合は、「アプリケーションサーバーを設定します」、「アプリケーションサーバーの設定を検証します」、「LiveCycle ES2 EAR をデプロイします」の各タスクは選択しないでください。ただし、[「WebLogic Server クラスタの手動設定」\(43 ページ\)](#) の各手順を完了してアプリケーションサーバーを手動で設定する必要があります。

警告： アプリケーションサーバークラスタを以前に手動で設定した場合は、「アプリケーションサーバーを設定します」オプションを選択しないでください。このオプションは、手動で設定したアプリケーションサーバークラスタにより LiveCycle ES2 でエラーが発生する場合に選択してください。

7. LiveCycle ES2 を設定 (1/5) 画面で、「設定」をクリックします。完了したら、「次へ」をクリックします。
8. LiveCycle ES2 を設定 (2/5) 画面で、「次へ」をクリックしてデフォルトのディレクトリをそのまま使用するか、LiveCycle ES2 がフォントへのアクセスに使用するディレクトリをカスタマイズします。その後で「次へ」をクリックします。
 - (オプション) 「Adobe サーバーフォントディレクトリ」のデフォルトの場所を変更するには、パスを入力するかディレクトリを参照します。
 - (オプション) 「カスタマフォントディレクトリ」のディレクトリを指定します。ディレクトリには、ライセンスおよびインストールされている追加のフォントが格納されています。

注意： クラスタ内の各ノードで、同じパスにローカルサーバーフォントとカスタマフォントのディレクトリを配置することをお勧めします。ローカルフォントディレクトリの代わりに共有フォントディレクトリを使用すると、パフォーマンスの問題が発生する可能性があります。

- (オプション) 「システムフォントディレクトリ」のデフォルトの場所を変更するには、パスを入力するか、ディレクトリを参照します。
 - (オプション) FIPS を有効にするには、「FIPS を有効にする」を選択します。このオプションは、連邦情報処理規格 (FIPS) を適用する場合にのみ選択してください。
9. LiveCycle ES2 を設定 (3/5) 画面で、「参照」をクリックし、「一時ディレクトリの場所」を指定します。
- (UNIX 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と、システムで設定されているデフォルトの場所が使用されます。
- クラスタ環境での一時ディレクトリの作成の詳細については、『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』を参照してください。
- 警告：**共有ネットワークディレクトリを一時ディレクトリとして指定した場合、以降のアップグレードが失敗することがあります。
10. LiveCycle ES2 を設定 (4/5) 画面で、GDS ディレクトリパスの場所を追加します。これを行うには、GDS ディレクトリの特定の場所を入力するか、「参照」をクリックして GDS ディレクトリの場所に移動します。
- クラスタの場合、クラスタ内のすべてのノードにアクセスできる共有 GDS ディレクトリパスを指定します。Windows ドライブに共有ネットワークの場所をマップできます。その後で、マップされたドライブを GDS のルートディレクトリとして使用できます。
- GDS ディレクトリのフィールドを空白のままにすると、LiveCycle ES2 によって、アプリケーションサーバーのディレクトリツリーにあるデフォルトの場所にディレクトリが作成されます。構成が完了した後で、LiveCycle 管理コンソール/コア設定をクリックすると、この場所を表示できます。
- 注意：**以前に作成した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を指定する必要があります (『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』ガイドを参照)。
- 警告：**GDS ディレクトリがドライブのルート (D:¥ など) にある場合、Windows でのコンポーネントのデプロイメントは失敗します。GDS については、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あ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。
11. 永続的なドキュメントストレージを設定 (5/5) 画面で、GDS ディレクトリのほかに、永続的なドキュメントストレージのオプションを選択します。次のいずれかを選択します。
- **GDS を使用：**すべての永続的なドキュメントストレージにファイルシステムベースの GDS を使用します。このオプションでは、最高のパフォーマンスを実現し、ストレージの場所として GDS だけを使用します。
 - **データベースを使用：**永続的なドキュメントや長期間有効な成果物の保存に、LiveCycle ES2 データベースを使用します。ただし、ファイルシステムベースの GDS も必要です。データベースを使用することにより、バックアップと復元の手順が簡単になります。
12. 「設定」をクリックし、LiveCycle ES2 EAR にこのディレクトリ情報を設定します。設定が完了したら、「次へ」をクリックします。
13. (**Content Services ES2 のみ**) LiveCycle Content Services ES2 設定画面で、Content Services ES2 で使用する次のパラメータを設定して、「次へ」をクリックします。
- **デプロイの種類：**「シングルサーバー」または「クラスタ」を選択します。
 - **インデックスのルートディレクトリ：**Content Services ES2 で使用するインデックスディレクトリを指定します。このディレクトリは、クラスタノードごとに固有です。すべてのノードで名前と場所を同じにする必要があります。

- **コンテンツ保存場所のルートディレクトリ**：Content Services ES2 で使用するルートディレクトリを指定します。コンテンツ保存場所のルートディレクトリは、クラスタ内のすべてのインスタンスによって共有される場所である必要があります。

注意： インターナショナル文字を含んだコンテンツ保存場所のルートディレクトリを指定する場合は、UTF-8 ロケールを使用して LiveCycle Configuration Manager を実行します。

- (オプション) **CIFS を有効にする**：このオプションを選択すると、LiveCycle ES2 を実行しているサーバーのファイルに Windows ベースのクライアントからアクセスするときに Common Internet File System (CIFS) プロトコルが使用できます。「次へ」をクリックし、以下の詳細を指定します。

- **CIFS サーバー名**：Content Services ES2 リポジトリへアクセスできるようにするための名前を指定します。LiveCycle Configuration Manager のデフォルトでは、LiveCycle ES2 サーバーに「a」を付けたサーバー名が設定されます。例えば、LiveCycle ES2 サーバーの名前が `lcserver` である場合は、CIFS サーバーの名前は `lcservera` に設定されます。指定する CIFS サーバー名は、ネットワーク内で一意である必要があります。

- **CIFS サーバー実装を選択**：サーバーでサポートされている、次のいずれかの CIFS 実装の種類を選択します。

JAVA (ソケットベース)：CIFS サーバーに割り当てられている代替 IP アドレスを指定し、ローカルドメインでのサーバー名の解決方法を指定します。例えば、プライマリ IP が 10.40.68.142 である場合は、10.40.68.143 を代替 IP として割り当てます。この IP がネットワーク上の他のマシンに割り当てられていないことを確認してください。

Windows ネイティブ (DLL ベース)：「参照」をクリックして、LiveCycle Configuration Manager が DLL ファイルをコピーする先のパス (通常は `C:\Windows\system32`) を選択します。LiveCycle Configuration Manager の実行を予定していないクラスタノードの場合、この手順は手動で実行する必要があります。

DLL ファイルのコピー元となるパスは、システムの PATH 環境変数に指定する必要があります。例えば、Windows Server 2003 および 2008 の場合、システム PATH に `C:\Windows\system32` を含める必要があります。

- **WINS サーバーまたはブロードキャストを使用してローカルドメインを解決してください**：ローカルドメインの解決に使用する方法を選択します。

ブロードキャスト：ローカルドメインにおけるネットワークセグメントのブロードキャストアドレス (サブネットマスク) を指定します。例えば、10.40.91.255 などです。ブロードキャストモードでは、CIFS サーバーとクライアントは同じサブネットに属している必要があります。

WINS サーバー：プライマリおよびセカンダリ WINS サーバーの IP アドレスを指定します。例えば、10.40.4.248 などです。WINS サーバーを選択する場合、クライアントはローカルドメイン内のどのサブネットでも使用できます。

Windows のみ

アプリケーションサーバーが Pure IPv6 モードで実行されている場合、CIFS は、Windows ではサポートされません。

CIFS サーバー名および割り当てられた仮想 IP アドレスを使用して、DNS エントリを更新し、Windows クライアントが CIFS サーバーに名前前でアクセスできるようにする必要があります。

UNIX のみ

UNIX マシンの場合、サポートされるのは Java の実装だけです。UNIX マシンでは、仮想インターフェイスを作成し、仮想 IP アドレスを割り当てる必要があります。この IP アドレスは、CIFS の実装で代替 IP アドレスとして使用することができます。これにより、UNIX マシンの場合は、Samba と CIFS を同じマシンで実行することができますが、2 つの異なる IP アドレスで実行されることとなります。これらのサービスは同じポートを使用するためです。

LiveCycle ES2 の IPv6 の実装で CIFS を有効にする場合は、EAR ファイルの設定が完了した後で、contentservices.war ファイルを編集する必要があります。EAR ファイルを更新してから、LiveCycle Configuration Manager の次の手順へ進みます。[「IPv6 モードでの CIFS の有効化」\(83 ページ\)](#) を参照してください。

Windows Server 2003 および Windows Server 2008 では、LiveCycle Configuration Manager でのこれらの手順に加えて、他の設定手順を手動で実行する必要があります。『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』ガイドの「CIFS を有効にするためのサーバー設定」を参照してください。

14. (**Content Services ES2 のみ**) LiveCycle Content Services ES2 モジュール設定画面で、次のタスクを実行してから、「設定」をクリックし、Content Services ES2 の設定を使用して LiveCycle ES2 EAR ファイルを設定します。設定が完了したら、「次へ」をクリックします。詳しくは、Alfresco のドキュメントを参照してください。

- Content Services ES2 に実装する Alfresco Module Packages (AMP) を選択します。デフォルトでは、選択した LiveCycle ES2 モジュールに関連するすべての AMP が実装されます。
- (オプション) 独自の AMP を実装するには、「Content Services で独自の AMP をパッケージングしますか?」ボックスを選択し、「参照」をクリックして、カスタム AMP が保存されているディレクトリを選択します。選択したディレクトリ内のすべての AMP がパッケージングされます。

注意： SharePoint クライアントを Alfresco CMS に移行できるようにするには、SharePoint AMP である [LiveCycleES2 root]¥LiveCycle_ES_SDK¥misc¥ContentServices¥adobe-vti-module.amp を追加する必要があります。

このファイルを追加したら、[「SharePoint クライアントアクセスの設定」\(82 ページ\)](#) で説明されている手順を実行します。

15. (**Windows のみ**) Acrobat を LiveCycle PDF Generator に合わせて設定画面で、「設定」をクリックして、Adobe Acrobat および必要な環境設定を設定するスクリプトを実行します。完了したら「次へ」をクリックします。

この画面は、サーバーコンピュータでローカルに LiveCycle Configuration Manager を実行している場合のみ表示されます。Adobe Acrobat が既にインストールされている必要があります。インストールされていないと、この手順は失敗します。

注意： Linux または Solaris で OpenOffice.org を使用するには、openoffice_PATH 環境変数を設定します。[「環境変数の設定」\(61 ページ\)](#) を参照してください。

16. LiveCycle ES2 の概要を設定画面で、「次へ」をクリックします。設定したアーカイブは [LiveCycleES2 root]/configurationManager/export ディレクトリに配置されます。アプリケーションサーバーが設定され、実行されていることを確認してください。

17. アプリケーションサーバーの設定の詳細画面で、各フィールドの情報を指定して (すべてのフィールドは必須です)、「サーバー接続を検証」をクリックします。検証が正常に完了したら、「次へ」をクリックします。必要な情報について詳しくは、**F1** キーを押してください。

注意： Adobe LiveCycle Configuration Manager のアプリケーションサーバーの設定の詳細画面では、アプリケーションサーバーのホスト名に使用する形式は、WebLogic Administration Console の「Listen Address」ボックスの値によって決まります。「Listen Address」ボックスが空の場合は、デフォルトの「localhost」をそのままホスト名として使用する必要があります。「Listen Address」ボックスに IP アドレスが表示されている場合は、ホスト名にその IP アドレスを使用する必要があります。

注意： LiveCycle Configuration Manager を使用してアプリケーションサーバーを設定し、ホスト名を IP アドレスで入力している場合、そのアプリケーションサーバーへのアクセスには常にその IP アドレスを使用する必要があります。

アプリケーションサーバーの「サーバーインスタンス名」は、大文字と小文字が区別されます。

LiveCycle Configuration Manager では、アプリケーションサーバーを実行しているサーバーの修飾されたホスト名を入力する必要があります。デフォルト値の localhost では動作しません。

警告：「デプロイの種類」ボックスで「クラスタ」を選択していることを確認します。

18. アプリケーションサーバーの設定の選択画面で、LiveCycle Configuration Manager で実行するタスクを選択し、「次へ」をクリックします。必要な情報について詳しくは、F1 キーを押してください。次の 1 つ以上のタスクを選択することができます。デフォルトでは両方のタスクが選択されます。

- サーバー設定を設定
- データソースを設定：次のいずれかのオプションを選択して、データソースを設定します。
 - パッケージされた JDBC モジュール（データソースをセキュリティで保護）
 - グローバルスコープのデータソース

アプリケーションサーバークラスタを手動で設定する場合は、対象のタスクを選択解除し、「次へ」をクリックします。

[「WebLogic Server クラスタの手動設定」\(43 ページ\)](#) の説明に従って、手動による設定タスクを実行する必要があります。

19. (「サーバー設定を設定」を選択した場合のみ) サーバー設定の設定画面で、各フィールドに情報を入力して「次へ」をクリックします。必要な情報について詳しくは、F1 キーを押してください。

サーバー設定を手動で設定している場合は、LiveCycle Configuration Manager を実行したまま、[「WebLogic Server の設定」\(43 ページ\)](#) の節の説明に従ってタスクを実行します。

注意： pop3.jar および JDK のパスは、クラスタ内のすべてのノードで同じにする必要があります。

20. (「パッケージされた JDBC モジュール」を指定して「データソースを設定」を選択した場合) データソース JDBC ドライバのクラスパスを設定画面で、JDBC ドライバのパスを指定します。

21. (「グローバルスコープのデータソース」を指定して「データソースを設定」を選択した場合) データソース設定画面で、各フィールドに情報を入力し、「データベース接続をテスト」をクリックします。接続のテストが正常に終了したら、「次へ」をクリックします。必要な情報について詳しくは、F1 キーを押してください。

データソースを手動で設定している場合は、LiveCycle Configuration Manager を実行したまま、[「JDBS 接続の設定」\(46 ページ\)](#) の節の説明に従ってタスクを実行します。

注意： WebLogic 11g R1 (バージョン 10.3.3) に LiveCycle ES2 をインストールしている場合は、必ず [「JDBS 接続の設定」\(46 ページ\)](#) の手順を完了してから、次の手順に進んでください。

22. アプリケーションサーバーの設定画面で、「設定」をクリックします。プロセスが完了したら、「次へ」をクリックします。

23. (「パッケージされた JDBC モジュール」を指定して「データソースを設定」を選択した場合) JDBC モジュールを LiveCycle ES2 EAR にパッケージ (1/2) 画面で、JDBC 設定の詳細を入力して、「データベース接続をテスト」をクリックします。データソース設定のパラメータについて詳しくは、F1 キーを押してください。

24. (「パッケージされた JDBC モジュール」を指定して「データソースを設定」を選択した場合) JDBC モジュールを LiveCycle ES2 EAR にパッケージ (2/2) 画面で、WebLogic のデータソース設定用に暗号化されたデータベースパスワードを生成するための詳細情報を入力します。次のいずれかのオプションを使用します。

警告： これは、WebLogic アプリケーションサーバーによって暗号化されたデータベースパスワードであり、アプリケーションサーバーのパスワードではありません。

- **既存の WebLogic で暗号化されたパスワードを使用：** データベース用に暗号化されたパスワードが既にある場合は、このオプションを選択します。WebLogic の暗号化ユーティリティを使用して、データベース接続のテスト用に前の画面で入力したパスワードを暗号化することができます。

- **WebLogic で暗号化されたパスワードを生成**：データベース用に暗号化されたパスワードを生成し、必要な詳細を指定する場合は、このオプションを選択します。データベース接続のテスト用に前の画面で入力したプレーンテキストのパスワードが、「パスワード」フィールドに自動的に入力されます。「パスワードを暗号化」をクリックして、暗号化されたデータベースパスワードを生成します。

25. 「設定」をクリックして、JDBC モジュールを LiveCycle ES2 EAR にパッケージし、完了したら、「次へ」をクリックします。
26. アプリケーションサーバーの設定の検証画面で、検証するタスクを選択し、「検証」をクリックします。プロセスが完了したら、「次へ」をクリックします。

注意：(WebLogic のみ) JDBC モジュールが LiveCycle ES2 EAR ファイルにパッケージされていると、LiveCycle Configuration Manager によって、アプリケーションサーバーの設定の検証中にデータソースの検証に失敗したことが報告されます。このメッセージは無視できます。

27. LiveCycle ES2 IVS EAR の対象範囲の確認画面が表示される場合は、この画面で、インストール検証サンプル (IVS) EAR ファイルをデプロイメントセットに含めるオプションを選択します。
28. LiveCycle ES2 EAR をデプロイ画面で、デプロイする EAR ファイルを選択してから「デプロイ」をクリックします。デプロイには数分かかる場合があります。デプロイメントが正常に完了したら、「次へ」をクリックします。

注意：この手順の後、null という名前のディレクトリが [appserverdomain] に作成されているかどうかを確認します。作成されている場合は、管理対象サーバー、ノードマネージャ、管理サーバーを停止し、逆の順序で起動します。再起動した後、**adobe** という名前のディレクトリが [appserverdomain] に作成されていることを確認します。確認できたら、[appserverdomain]/null ディレクトリを削除します。

29. LiveCycle ES2 データベースの初期化画面で、ホストとポートの情報を確認して、「初期化」をクリックします。データベースの初期化タスクによって、データベースにテーブルが作成され、デフォルトのデータがテーブルに追加されて、データベースに基本的なロールが作成されます。初期化が正常に完了したら、「次へ」をクリックします。

この画面のポートは、管理サーバーのポートではなく、管理対象サーバーのポートを示しています。

注意：データベースの初期化を行う必要があるのは、クラスタの 1 つのサーバーに対してのみです。また、このサーバーに対してのみ、後続の手順を実行する必要があります。

30. LiveCycle ES2 サーバー情報画面で、「パスワード」ボックスに password (これはデフォルトの管理者パスワードです。後で変更することをお勧めします) を入力します。
31. 「サーバー接続を検証」をクリックし、完了したら、「次へ」をクリックします。

注意：この画面に表示されるサーバー情報はデプロイメントの既定値です。

サーバー接続の検証は、デプロイメントや検証でエラーが発生した場合に、トラブルシューティングの対象を絞り込むのに役立ちます。接続テストが正常に終了しても以降の段階でデプロイメントや検証のエラーが発生する場合は、接続の問題をトラブルシューティングの対象から除外できます。

32. Central Migration Bridge Service デプロイメント設定画面が表示される場合は、この画面で「Central Migration Bridge Service をデプロイメントに含める」オプションを選択し、「次へ」をクリックします。
33. LiveCycle コンポーネントのデプロイメント画面で、「デプロイ」をクリックします。ここでデプロイされるコンポーネントは、サービスのデプロイ、統合および実行を目的として LiveCycle ES2 サービスコンテナにプラグインされている Java アーカイブファイルです。検証の進行状況を確認するには、「進行状況ログを表示」をクリックします。デプロイメントが正常に完了したら、「次へ」をクリックします。

34. LiveCycle コンポーネントのデプロイメントの検証画面で、「検証」をクリックします。LiveCycle Configuration Manager によって、LiveCycle コンポーネント (Java アーカイブファイル) が LiveCycle ES2 サーバーにデプロイされ、稼働していることが検証されます。検証の進行状況を確認するには、「進行状況ログを表示」をクリックします。検証が正常に完了したら、「次へ」をクリックします。
35. LiveCycle コンポーネントを設定画面で、LiveCycle Configuration Manager で実行するタスクを選択し、「次へ」をクリックします。詳しくは、F1 キーを押してください。
36. **(EMC Documentum、IBM FileNet または IBM Content Manager が設定用に選択されている場合)**
LiveCycle サーバー JNDI 情報画面で、JNDI サーバーのホスト名とポート番号を入力します。詳しくは、F1 キーを押してください。
37. **(オプション - EMC Documentum のみ)** EMC Documentum のクライアントを指定画面で、「Connector for EMC Documentum コンテンツサーバーを設定します。」を選択して、次の情報を指定します。詳細情報を入力して、「検証」をクリックし、完了したら、「次へ」をクリックして次に進みます。
 - **EMC Documentum クライアントのバージョンを選択** : EMC Documentum Content Server で使用するクライアントのバージョンを選択します。
 - **EMC Documentum クライアントのインストールディレクトリのパス** : 「参照」をクリックして、ディレクトリのパスを選択します。
38. **(オプション - EMC Documentum のみ)** EMC Documentum Content Server 設定を指定画面で、EMC Documentum Server の詳細を入力し、「次へ」をクリックします。入力する必要がある情報について詳しくは、F1 キーを押してください。
39. **(オプション - EMC Documentum のみ)** Adobe LiveCycle ES2 Connector for EMC Documentum を設定画面で、「Documentum Connector を設定」をクリックします。完了したら、「次へ」をクリックします。
40. **(オプション - IBM Content Manager のみ)** IBM Content Manager のクライアントを指定画面で、「IBM Content Manager のクライアントを指定」を選択し、「IBM Content Manager クライアントのインストールディレクトリのパス」を入力します。「検証」をクリックし、完了したら、「次へ」をクリックして次に進みます。
41. **(オプション - IBM Content Manager のみ)** IBM Content Manager サーバーの設定を指定画面で、IBM Content Manager Server の詳細情報を入力し、「次へ」をクリックします。詳しくは、F1 キーを押してください。
42. **(オプション - IBM Content Manager のみ)** Adobe LiveCycle ES2 Connector for IBM Content Manager を設定画面で、「IBM Content Manager Connector を設定」をクリックします。完了したら、「次へ」をクリックします。
43. **(オプション - IBM FileNet のみ)** IBM FileNet のクライアントを指定画面で、「IBM FileNet のクライアントを指定」を選択し、次の設定を指定します。詳細情報を入力して、「検証」をクリックし、完了したら、「次へ」をクリックして次に進みます。
 - **IBM FileNet クライアントのバージョンを選択** : EMC Documentum Content Server で使用するクライアントのバージョンを選択します。
 - **IBM FileNet クライアントのインストールディレクトリのパス** : 「参照」をクリックして、ディレクトリのパスを選択します。
44. **(オプション - IBM FileNet のみ)** IBM FileNet Content Server の設定を指定画面で、必要な詳細情報を入力し、「次へ」をクリックします。詳しくは、F1 キーを押してください。
45. **(オプション - IBM FileNet のみ)** IBM FileNet Process Engine のクライアントを指定画面で、必要な詳細情報を入力し、「確認」をクリックします。完了したら、「次へ」をクリックします。詳しくは、F1 キーを押してください。
46. **(オプション - IBM FileNet のみ)** IBM FileNet Process Engine サーバーの設定を指定画面で、必要な詳細情報を入力し、「次へ」をクリックします。詳しくは、F1 キーを押してください。

47. **(オプション - IBM FileNet のみ)** Adobe LiveCycle ES2 Connector for IBM FileNet を設定画面で、「FileNet Connector を設定」をクリックします。完了したら、「次へ」をクリックします。詳しくは、F1 キーを押してください。
48. **(オプション - LiveCycle ES2 Connector for Microsoft SharePoint のみ)** Adobe LiveCycle ES2 Connector for Microsoft SharePoint の画面で、次の詳細情報を入力して「設定」をクリックします。完了したら、「次へ」をクリックします。

注意： LiveCycle 管理コンソールを使用して後で SharePoint Connector を設定する場合は、この手順をスキップできます。

- **ユーザー名およびパスワード：** SharePoint サーバーへの接続に使用するユーザーアカウントの詳細情報を入力します。
- **ホスト名：** SharePoint サーバーのホスト名を <hostname>:<port> の形式で入力します。SharePoint サーバー上の Web アプリケーションのポート番号を指定します。
- **ドメイン名：** SharePoint サーバーが存在しているドメインを入力します。

49. **(PDF Generator ES2 のみ)** LiveCycle サーバーマシン管理者のユーザー資格情報を管理画面で、サーバーコンピュータの管理者権限を持つユーザーのユーザー名とパスワードを入力して、「追加」をクリックします。

注意： Windows Server 2008 の場合は、管理ユーザーを 1 人以上追加する必要があります。Windows Server 2008 では、追加するユーザーのユーザーアカウント制御 (UAC) を無効にする必要があります。UAC を無効にするには、コントロールパネル/ユーザーアカウント/ユーザーアカウント制御の有効化または無効化を順にクリックし、「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」の選択を解除します。次に、「OK」をクリックします。変更を適用するには、コンピュータを再起動します。

Windows Server 2003、Linux および Solaris の場合は、ユーザーの追加は必須ではありません。Linux および Solaris プラットフォームで追加されるユーザーには、sudo 権限が必要です。

詳しくは、この画面で F1 キーを押し、LiveCycle Configuration Manager ヘルプにアクセスしてください。

50. **(LiveCycle Configuration Manager がサーバーマシン上でローカルに実行されている場合の PDF Generator ES2 のみ)** LiveCycle PDFGenerator System Readiness Test 画面で、「開始」をクリックし、システムが PDF Generator ES2 に合わせて適切に設定されているかどうかを検証します。
51. **(PDF Generator ES2 のみ)** System Readiness Tool レポートを確認し、「次へ」をクリックします。
52. **(Reader Extensions ES2 のみ)** LiveCycle Reader Extensions ES2 の秘密鍵証明書の設定画面で、モジュールサービスをアクティブにする Reader Extensions ES2 秘密鍵証明書に関連する詳細を指定します。

- **秘密鍵証明書ファイル：** Reader Extensions ES2 秘密鍵証明書のパスとファイル名 (.pfx または .p12 ファイルタイプ)。
- **秘密鍵証明書のパスワード：** 秘密鍵証明書に関連付けられるパスワード。このパスワードは秘密鍵証明書ファイルと共に提供されています。
- **この秘密鍵証明書のユーザー定義の名前：** 設定時にこの秘密鍵証明書に対して指定した名前 (またはエイリアス)。

この名前は、Reader Extensions ES2 Web インターフェイスに表示され、SDK 呼び出しを通じて秘密鍵証明書の参照に使用されるエイリアスです。Reader Extensions ES2 秘密鍵証明書には任意の一意名を作成できます。

注意： ここで、「LiveCycle 管理コンソールを使用して後から設定」を選択すると、この手順をスキップできません。デプロイメントの完了後に、LiveCycle 管理コンソールで Reader Extensions ES2 秘密鍵証明書を設定できます (LiveCycle 管理コンソールにログインしたら、ホーム/設定/Trust Store の管理/ローカル秘密鍵証明書の順にクリックします)。

「設定」をクリックし、「次へ」をクリックします。

53. (オプション) LiveCycle ES2 サンプルを読み込み画面で、「読み込み」をクリックします。読み込みが正常に完了したら、「次へ」をクリックします。または、「LiveCycle ES2 サンプルの読み込みをスキップ」をクリックしてから「次へ」をクリックし、サンプルを後で読み込みます。

警告： LiveCycle ES2 サンプルは実稼働環境に読み込まないでください。これらのサンプルによりデフォルトパスワードを持つユーザーが作成され、実稼働環境にセキュリティ上の問題が生じる可能性があります。

54. 概要ページで、実行するタスクを確認して、「次へ」をクリックします。
55. 次の手順画面に、URL とログイン情報が表示されます。「完了」をクリックし、LiveCycle Configuration Manager を終了します。

注意： LiveCycle ES2 を設定したら、実際のソリューション実装に該当する設定後の作業を行います。

56. クラスタ内の各アプリケーションサーバーインスタンスを再起動します。

4.4 EAR ファイルのアンインストール

LiveCycle ES2 モジュールのデプロイを解除するには、次の手順を実行します。

▶ EAR ファイルのアンインストール：

1. 管理サーバーと管理対象サーバーを起動します。
2. Web ブラウザの URL 行に `http://localhost:7001/console` と入力して WebLogic Server Administration Console を起動します。
3. 「Domain Structure」で、「Deployments」をクリックします。
4. Change Center で、「Lock & Edit」をクリックします。
5. 「LiveCycle application」ボックスを選択し、LiveCycle アプリケーション (adobe-livecycle-native-weblogic-[OS]、adobe-livecycle-weblogic、adobe-workspace-client、adobe-contentservices など) を選択します。
6. 「Deploy」タブで「Stop」をクリックします。
7. 「Lock & Edit」をクリックし、停止しているアプリケーションを選択して、「Delete」をクリックします。
8. デプロイしたその他の LiveCycle ES2 モジュールに対して、手順 5～7 を繰り返します。
9. 「Save」をクリックし、「Activate Changes」をクリックします。
10. 管理サーバーと管理対象サーバーを停止して起動します。

4.5 次の手順

LiveCycle ES2 の設定とデプロイを完了したら、次の作業を実行できます。

- デプロイメントの確認 ([「デプロイメントの確認」\(56 ページ\)](#) を参照)。
- LiveCycle 管理コンソールへのアクセス ([「LiveCycle 管理コンソールへのアクセス」\(56 ページ\)](#) を参照)。
- PDF Generator ES2 または PDF Generator 3D ES2 の設定 ([「LiveCycle PDF Generator ES2 または 3D ES2 の設定」\(60 ページ\)](#) を参照)。
- Rights Management ES2 の最終設定 ([「監視フォルダのパフォーマンスパラメータの設定」\(66 ページ\)](#) を参照)。

- LDAP にアクセスするための LiveCycle ES2 モジュールの設定 ([「LiveCycle ES2 の LDAP アクセス設定」\(67 ページ\)](#) を参照)。
- PDF Generator ES2 の監視フォルダパフォーマンスの設定 ([「監視フォルダのパフォーマンスパラメータの設定」\(66 ページ\)](#) を参照)。
- FIPS モードの有効化 ([「FIPS モードの有効化」\(69 ページ\)](#) を参照)。
- HTML 電子署名の設定 ([「HTML 電子署名の設定」\(69 ページ\)](#) を参照)。
- Connector for EMC Documentum、Connector for IBM Content Manager または Connector for IBM FileNet の設定 ([「Connector for EMC Documentum サービスの設定」\(70 ページ\)](#)、[「Connector for IBM FileNet サービスの設定」\(74 ページ\)](#) または [「Connector for IBM Content Manager の設定」\(84 ページ\)](#) を参照)。
- PDF Generator ES2 の環境変数の設定 ([「環境変数の設定」\(61 ページ\)](#) を参照)。
- LiveCycle Business Activity Monitoring ES2 のインストールおよびデプロイ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』ガイドを参照してください。
- LiveCycle ES2 のアンインストール ([「LiveCycle ES2 のアンインストール」\(88 ページ\)](#) を参照)。

5

WebLogic Server クラスタの手動設定

この章では、クラスタ環境で LiveCycle ES2 の手動デプロイメントに備えて WebLogic Server クラスタを手動で設定する方法について説明します。この章は、WebLogic Server クラスタを自動設定しないように選択している場合にのみ該当します。アプリケーションサーバーを自動的に設定する方法については、[「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照してください。

インストールのこの段階では、既に LiveCycle ES2 ファイルがインストールされ、デプロイ可能な LiveCycle ES2 アーカイブを設定するための LiveCycle Configuration Manager が実行されています。ここで、以下の作業を手動で実行する必要があります。

- WebLogic Server を設定します ([「WebLogic Server の設定」\(43 ページ\)](#) を参照)。
- JDBS 接続を設定します ([「JDBS 接続の設定」\(46 ページ\)](#) を参照)。

5.1 WebLogic Server の設定

WebLogic Server クラスタの以下の領域を設定します。

- WebLogic Server のタイムアウトの設定 ([「タイムアウトの設定」\(43 ページ\)](#) を参照)。
- WebLogic Server サーバーの開始引数 ([「サーバーの開始引数の設定」\(44 ページ\)](#) を参照)。
- 管理対象となる各 WebLogic Server のクラスパス ([「管理対象サーバーのクラスパスを設定するには」\(46 ページ\)](#) を参照)。

5.1.1 タイムアウトの設定

デプロイメントによっては、LiveCycle ES2 EAR ファイルを大きくすることができます。EAR ファイルをデプロイするときにタイムアウトが発生しないように、WebLogic Server のタイムアウト設定値を大きくする必要があります。

▶ クラスタのトランザクションタイムアウトを設定するには

1. WebLogic Server Administration Console の「Domain Structure」で、[「WebLogic ドメインの作成」\(14 ページ\)](#) で定義したドメイン名をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 「Configuration」タブをクリックし、「JTA」をクリックします。
4. 「Timeout Seconds」ボックスに 300 と入力します。
5. 「Save」をクリックし、「Activate Changes」をクリックします。

▶ クラスタのスタックレッドタイムアウトを設定するには

1. WebLogic Server Administration Console の「Domain Structure」で、Environment / Servers をクリックします。
2. 「Servers」テーブルで、クラスタ内のサーバーの名前をクリックします。
3. Change Center で、「Lock & Edit」をクリックします。

4. 「Configuration」タブ/Tuning をクリックします。
5. 「Stuck Thread Max Time」ボックスに 1200 と入力します。
6. 「Save」をクリックし、「Activate Changes」をクリックします。
7. クラスタ内のサーバーごとに、手順 2～6 を繰り返します。

5.1.2 サーバーの開始引数の設定

LiveCycle ES2 オプションを追加するには、LiveCycle ES2 クラスタの各 WebLogic Server インスタンスでサーバーの開始引数を設定する必要があります。

この手順を開始する前に、クラスタで 32 ビットまたは 64 ビット JVM が使用されているかどうかを確認する必要があります。クラスタ設定に必要な JVM を確認するには、『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』を参照してください。

この手順を開始する前に、クラスタキャッシュのサーバーの開始引数を正しく設定できるように、LiveCycle ES2 クラスタでのクラスタキャッシュの実装方法を確認してください。クラスタキャッシュは、UDP または TCP を使用して実装できますが、両方使用することはできません。クラスタに適切な実装を選択してください。

- クラスタが IPv4 ベースの場合のみ、UDP を使用します。
- クラスタが IPv4 ベースまたは IPv6 ベースの場合は、TCP を使用します。IPv6 ベースのクラスタには、IPv6 に準拠するように TCP を使用します。

TCP を使用してクラスタキャッシュを実装する場合は、TCP ロケータを正しく設定します ([「キャッシュロケータの設定 \(TCP を使用するキャッシュのみ\)」 \(26 ページ\)](#) を参照)。

▶ サーバー起動引数を設定するには：

1. WebLogic Server Administration Console で、「Domain Structure」の Environment / Servers をクリックし、右側のウィンドウで LiveCycle ES2 クラスタのサーバー名をクリックします。
2. 「Configuration」タブ/Server Start をクリックします。
3. Change Center で、「Lock & Edit」をクリックします。
4. 「Arguments」ボックスに、以下の JVM 引数セットのいずれか 1 つを追加します。
 - (32 ビット JVM のみ) Add -XX:MaxPermSize=256m -Xms256m -Xmx1024m
 - (64 ビット JVM のみ) Add -XX:MaxPermSize=512m -Xms256m -Xmx1792m

注意： 上述の適切なテキストブロックをテキストエディタにコピーして、すべての改行が削除されていることを確認してください。

5. (IPv4 のみ) 「Arguments」ボックスに、以下の JVM 引数を追加します。

```
-Dadobeidp.RootDirectory=<appserver domain>  
-Djava.net.preferIPv4Stack=true -Dfile.encoding=utf8  
-Djava.security.policy=<appserver root>/server/lib/  
weblogic.policy
```

6. (IPv6 のみ) 「Arguments」ボックスに、以下の JVM 引数を追加します。

```
-Dadobeidp.RootDirectory=<appserverdomain> -Djava.net.preferIPv6Stack=true  
-Djava.net.preferIPv6Addresses=true -Dfile.encoding=utf8  
-Djava.security.policy=<appserver root>/server/lib/weblogic.policy
```

注意： 64 ビット UNIX プラットフォームを使用している場合、次の JVM 引数を追加します。

```
-d64
```

注意： adobeidp.RootDirectory は、クラスタのすべてのノードで同じ場所に作成する必要があります。

ヒント： 上述のテキストブロックをテキストエディタにコピーして、すべての改行が削除されていることを確認し、以下のテキストをすべて置き換えます。

- <appserver domain> をアプリケーションサーバーのドメインパスに置き換えます。
- <appserver root> をアプリケーションサーバーのルートディレクトリに置き換えます。

7. クラスタキャッシュ用に JVM 引数を設定します。「Arguments」ボックスに、以下のキャッシュ引数の 1 つを追加または設定します。

- (UDP 検索を使用したキャッシュ) マルチキャストポート引数を次の形式で設定します。

```
-Dadobe.cache.multicast-port=<port number>
```

注意： <port number> の値は、1 ~ 65535 の使用可能なポートにすることができます。マルチキャストポートは、LiveCycle ES2 クラスタに対して一意である必要があります (同じネットワーク上の別のクラスタで同じポートを使用することはできません)。次の例のように、LiveCycle ES2 クラスタのすべてのノードに同じ <port number> を設定することをお勧めします。

```
-Dadobe.cache.multicast-port=33456
```

- (TCP 検索を使用したキャッシュ) IPv4 の場合、クラスタロケータ引数を次の形式で設定します。

```
-Dadobe.cache.cluster-locators=<IPaddress> [<port number>] , <IPaddress>  
 [<port number>]
```

IPv6 の場合、クラスタロケータ引数を次の形式で設定します。

```
-Dadobe.cache.cluster-locators=<hostname>@<IPv6 address>  
 [<port number>] , <hostname>@<IPv6 address> [<port number>]
```

注意： クラスタのすべてのノードのロケータを、カンマ区切りのリストで設定します。<IPaddress> の値は、ロケータを稼働しているコンピュータの IP アドレスです。<port number> の値は、1 ~ 65535 の任意の未使用ポートです。次の例に示すように、LiveCycle ES2 クラスタ内のすべてのノードで同じ <port number> を設定することをお勧めします。

```
-Dadobe.cache.cluster-locators=10.20.30.5 [22345] , 10.20.30.6 [22345]
```

```
-Dadobe.cache.cluster-locators=node1@
```

```
2001:1890:110b:7131:7996:3596:2921:364e [22345] , node2@
```

```
2001:1890:110b:7131:9192:a8df:1cc7:75b2 [22345]
```

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

9. クラスタ内のサーバーごとに、手順 2 ~ 8 を繰り返します。

5.1.3 管理対象サーバーのクラスパスを設定するには

LiveCycle ES2 によってインストールされた JAR ファイルを含めるには、クラスタ内の各管理対象 WebLogic Server のクラスパスを設定します。

▶ 管理対象サーバーのクラスパスを変更するには

1. クラスタ内の各管理対象すべての WebLogic Server が起動していることを確認します。停止している管理対象サーバーを起動するには、ノードマネージャを使用します。
2. WebLogic Server Administration Console の「Domain Structure」で、Environment / Servers をクリックします。
3. Change Center で、「Lock & Edit」をクリックし、サーバーの名前をクリックします。
4. 「Configuration」タブ / Server Start をクリックします。
5. 「Class Path」にクラスパスを入力し、pop3.jar ファイル ([appserverdomain]/idplib/pop3.jar)、weblogic.jar ファイル ([appserver root]/server/lib/weblogic.jar)、tools.jar ファイル ([JAVA_HOME]/lib/tools.jar) および JDBC ドライバ ([appserverdomain]/idplib/[JDBC ドライバの .jar ファイル]) の場所とファイル名をクラスパスに追加します。
注意： weblogic.jar ファイルクラスおよび各種 JAR ファイルをコロン (:) (Linux/UNIX) またはセミコロン (;) (Windows) で区切る前に、pop3.jar ファイルがリストされていることを確認してください。
6. 「Save」をクリックします。
7. クラスタ内のすべてのサーバーに関して、手順 2 ~ 6 を繰り返します。
8. Change Center で、「Activate Changes」をクリックします。

5.2 JDBS 接続の設定

クラスタ内の全メンバーが、LiveCycle ES2 データベースの JDBC データソースを共有します。共有の JDBC データソースを作成および設定するには、次の作業を行います。

- LiveCycle ES2 データベースへの接続を設定します ([「LiveCycle ES2 データソースの作成と設定」\(46 ページ\)](#) を参照)。
- (LiveCycle Rights Management ES2 のみ) LiveCycle Rights Management ES2 に使用するデータベースへの接続を設定します ([「Rights Management ES2 データソースの作成と設定」\(48 ページ\)](#) を参照)。

5.2.1 LiveCycle ES2 データソースの作成と設定

クラスタ用のデータソースを作成します。共有の JDBC データソースを作成および設定するには、次の作業を行います。

- データソースを作成します ([「LiveCycle ES2 データソースの作成」\(47 ページ\)](#) を参照)。
- データソースをクラスタに割り当てます ([「クラスタへのデータソースの割り当て」\(49 ページ\)](#) を参照)。
([「クラスタへのデータソースの割り当て」\(47 ページ\)](#) を参照)
- データソースの最大プール容量を設定します ([「データソースの最大プール容量の設定」\(49 ページ\)](#) を参照)。
([「データソースの最大プール容量の設定」\(48 ページ\)](#) を参照)

5.2.1.1 LiveCycle ES2 データソースの作成

WebLogic Server クラスタ内で、まず LiveCycle ES2 データベースにリンクする JDBC データソースを作成します。

▶ LiveCycle ES2 データソースを作成するには

1. WebLogic Server Administration Console の「Domain Structure」で、Services/JDBC/Data Sources をクリックします。
2. Change Center で、「Lock & Edit」をクリックし、「New」をクリックします。
3. (オプション)「Name」ボックスに、データソースの名前 (IDP_DS など) を入力します。
4. 「JNDI Name」ボックスに、データソースの名前として IDP_DS と入力します。
5. 「Database」リストで、使用しているデータベースに一致するデータベースの種類を選択します。
6. 「Database Driver」リストで、データベースに適切なデータベースドライバを選択し、「Next」をクリックします。
7. 「Supports Global Transactions」を選択し、「Emulate Two-Phase Commit」を選択して、「Next」をクリックします。
8. 「Database Name」ボックスに、データベースの名前を入力します。
9. 「Host Name」ボックスに、データベースのホストコンピュータの IP アドレスを入力します。
10. 「Port」ボックスに、データベースのホストコンピュータのポート番号を入力します。
11. 「Database User Name」ボックスに、データベースのユーザー名を入力します。
12. 「Password」ボックスに、データベースのパスワードを入力して、「Confirm Password」ボックスに再入力します。
13. 「Next」をクリックし、「Test Configuration」をクリックします。データベース設定が正しいことを確認する確認応答が表示されます。
14. テストが成功したら、「Finish」をクリックします。
15. Change Center で、「Activate Changes」をクリックします。

5.2.1.2 クラスタへのデータソースの割り当て

データソースをクラスタに割り当てる必要があります。

▶ LiveCycle ES2 データソースをクラスタに割り当てるには

1. WebLogic Server Administration Console の「Domain Structure」で、Services/JDBC/Data Sources をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. クラスタに割り当てるデータソースの名前をクリックします。
4. 「Targets」タブをクリックして、「Clusters」領域でクラスタ名を選択し、「All servers in the cluster」を選択します。
5. 「Save」をクリックし、「Activate Changes」をクリックします。

5.2.1.3 データソースの最大プール容量の設定

データソースの最大プール容量を設定する必要があります。

▶ 最大プール容量を設定するには

1. WebLogic Server Administration Console の「Domain Structure」で、Services / JDBC / Data Sources をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 設定するデータソースの名前をクリックします。
4. 「Configuration」タブ / Connection Pool をクリックします。
5. (Oracle のみ) 「Initial Capacity」ボックスに、1 と入力します。
6. 「Maximum Capacity」ボックスに 30 と入力します。
7. 「Save」をクリックし、「Activate Changes」をクリックします。

5.2.2 Rights Management ES2 データソースの作成と設定

LiveCycle ES2 デプロイメントで Rights Management ES2 を使用する場合は、クラスタ内で Rights Management ES2 用の別個のデータソースを作成および設定します。LiveCycle ES2 データソースの場合と同じ作業を、以下の設定を使用して行います。

- データソースを作成します ([「データソースの作成」\(48 ページ\)](#) を参照)。
- データソースをクラスタに割り当てます ([「クラスタへのデータソースの割り当て」\(49 ページ\)](#) を参照)。
- データソースの最大プール容量を設定します ([「データソースの最大プール容量の設定」\(49 ページ\)](#) を参照)。

5.2.2.1 データソースの作成

WebLogic Server クラスタ内で、まず Rights Management ES2 データベースにリンクする JDBC データソースを作成します。

▶ Rights Management ES2 データソースを作成するには

1. WebLogic Server Administration Console の「Domain Structure」で、Services / JDBC / Data Sources をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 「New」をクリックし、「Name」ボックスに、データソースの名前として RM_DS と入力します。
4. 「JNDI Name」ボックスに、データソースの名前として EDC_DS と入力します。
5. 「Database」リストで、使用しているデータベースに一致するデータベースの種類を選択します。
6. 「Database Driver」リストで、データベースに適切なデータベースドライバを選択し、「Next」をクリックします。
7. 「Supports Global Transactions」を選択して、「One-Phase Commit」を選択し、「Next」をクリックします。
8. 「Database Name」ボックスに、データベースの名前を入力します。
9. 「Host Name」ボックスに、データベースのホストコンピュータの IP アドレスを入力します。
10. 「Port」ボックスに、データベースのホストコンピュータのポート番号を入力します。

11. 「Database User Name」ボックスに、データベースのユーザー名を入力します。
12. 「Password」ボックスに、データベースのパスワードを入力して、「Confirm Password」ボックスに再入力します。
13. 「Next」をクリックし、「Test Configuration」をクリックします。データベース設定が正しいことを確認する確認応答が表示されます。
14. テストが成功したら、「Finish」をクリックします。
15. Change Center で、「Activate Changes」をクリックします。

5.2.2.2 クラスタへのデータソースの割り当て

データソースをクラスタに割り当てる必要があります。

▶ データソースをクラスタに割り当てるには

1. WebLogic Server Administration Console の「Domain Structure」で、Services/JDBC/Data Sources をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. クラスタに割り当てるデータソースの名前をクリックします。
4. 「Targets」タブをクリックして、「Clusters」領域でクラスタ名を選択し、「All servers in the cluster」を選択します。
5. 「Save」をクリックし、「Activate Changes」をクリックします。

5.2.2.3 データソースの最大プール容量の設定

Rights Management ES2 データソースの最大プール容量を設定する必要があります。

▶ 最大プール容量を設定するには

1. WebLogic Server Administration Console の「Domain Structure」で、Services/JDBC/Data Sources をクリックします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 設定するデータソースの名前をクリックします。
4. 「Configuration」タブ/Connection Pool をクリックします。
5. 次のボックスにそれぞれの値を入力します。

Initial Capacity : 1

Maximum Capacity : 20

Capacity Increment : 5.

Statement Cache Size : 100.

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

5.3 次の手順

WebLogic Server クラスタを設定したら、以下の作業を実行します。

- LiveCycle Configuration Manager を使用して LiveCycle ES2 EAR ファイルを設定します ([「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照)。
- 以下のいずれかの方法を選択して、LiveCycle ES2 EAR ファイルを WebLogic Server クラスタにデプロイします。
 - **自動**：LiveCycle Configuration Manager を使用します ([「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#) を参照)。
 - **手動**：[「WebLogic Server Cluster への手動デプロイ」\(51 ページ\)](#) を参照してください。

6

WebLogic Server Cluster への手動デプロイ

この章では、LiveCycle ES2 モジュールを a WebLogic Server cluster に手動でデプロイする方法について説明します。この章の内容は、LiveCycle ES2 を your WebLogic Server cluster に自動的にデプロイしない場合にのみ適用されます。LiveCycle ES2 をアプリケーションサーバーに自動的にデプロイする方法については、[「LiveCycle ES2 をデプロイするための設定」 \(30 ページ\)](#) を参照してください。

インストールのこの段階では、既に LiveCycle ES2 ファイルがインストールされています。LiveCycle Configuration Manager を実行してデプロイ可能な LiveCycle ES2 アーカイブを設定し、WebLogic Server を手動で設定してください cluster。この時点で、デプロイ可能な LiveCycle ES2 アーカイブを手動でデプロイします。

この章には、以下のトピックが含まれています。

- [「LiveCycle ES2 モジュールのデプロイについて」 \(51 ページ\)](#)
- [「WebLogic Server へのデプロイ」 \(52 ページ\)](#)

6.1 LiveCycle ES2 モジュールのデプロイについて

LiveCycle ES2 をデプロイする前に、次のタスクが完了していることを確認してください。

- 必要なソフトウェアとファイルがインストールしてあり、作業を行うディレクトリの場所を確認している。このタスクを完了しなかった場合、『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』を参照してください。
- LiveCycle Configuration Manager を実行し、システムおよびアプリケーションサーバーの要件に従って LiveCycle ES2 モジュールを設定およびアセンブリしている。デプロイメントにモジュールを追加するには、LiveCycle Configuration Manager を実行して変更を行い、更新した EAR ファイルを再デプロイします。

LiveCycle ES2 を初めてデプロイする場合は、製品をデプロイした後に、LiveCycle Configuration Manager を使用してデータベースを初期化してください。

外部 Web サーバーを使用している場合は、Web サーバーのマニュアルを参照して、アプリケーションサーバーへのアクセスに必要な設定について確認してください。

6.1.1 WebLogic Server ディレクトリ名

この章では、WebLogic Server のホームディレクトリを [appserver root] と表します。ホームディレクトリは、WL_INSTALL_ROOT 環境変数に指定されているものと同じです。同様に、[LiveCycleES2 root] は、LiveCycle ES2 コンポーネントがインストールされている場所を表します ([「このドキュメントで使用する表記」 \(7 ページ\)](#) を参照)。

6.1.2 デプロイ可能なコンポーネントの概要

デプロイメントプロセス中に、次の EAR ファイルをデプロイします。

- adobe-livecycle-native-weblogic-[OS].ear
- (Process Management ES2 のみ) adobe-livecycle-weblogic.ear
- adobe-workspace-client.ear
- adobe-contentservices.ear (LiveCycle Content Services ES2 のみ)

LiveCycle Configuration Manager を使用して LiveCycle ES2 を設定すると (必須)、これらのファイルは [LiveCycleES2 root]/configurationManager/export ディレクトリに配置されます。

6.2 Content Services ES2 をデプロイするための必須条件

LiveCycle ES2 インストールで Content Services ES2 が使用されているが、LiveCycle Configuration Manager を利用してアプリケーションサーバーをまだ設定していない場合、デプロイメントの前にアプリケーションサーバーの追加設定を手動で実行する必要があります。[「Content Services ES2 の設定」\(67 ページ\)](#) を参照してください。

6.3 WebLogic Server へのデプロイ

LiveCycle ES2 コンポーネントを WebLogic Server にデプロイするには、WebLogic Administration Console を使用して、コンポーネントの EAR ファイルをアプリケーションサーバーにデプロイします。

WebLogic Server にデプロイする前に、使用しているコンピュータでアプリケーションサーバーを起動します。必須のコンポーネントをデプロイしたら、サービスを起動する前にアプリケーションサーバーを停止して再起動します。

▶ EAR ファイルをデプロイするには

1. Web ブラウザの URL 行に `http://localhost:7001/console` と入力して WebLogic Administration Console にアクセスします。
2. WebLogic Server 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。
3. Change Center で、「Lock & Edit」をクリックします。
4. 「Domain Structure」で、「Deployments」をクリックし、右側のウィンドウで「Install」をクリックします。
5. Install Application Assistant ウィンドウで、インストールするデプロイメント EAR ファイルに移動します。
6. EAR ファイルを選択し、「次へ」をクリックします。
7. 「Install this deployment as an application」を選択し、「Next」をクリックします。
8. デプロイ対象についてクラスタを選択し、「All servers in the cluster」を選択します。
9. 「Next」をクリックして、デフォルトの設定を受け入れて、「Finish」をクリックします。
10. Change Center で、「Activate Changes」をクリックします。
11. 右側のウィンドウで、今インストールしたアプリケーションのチェックボックスを選択します。
12. 「Start」をクリックし、メニューから「Servicing all requests」を選択します。
13. 右側のウィンドウで「Yes」をクリックし、Change Center で「Lock & Edit」をクリックします。
14. [「デプロイ可能なコンポーネントの概要」\(51 ページ\)](#) の各 EAR ファイルについて、手順 5 ~ 13 を繰り返します。
15. デプロイが完了したら、WebLogic を再起動します ([「WebLogic Server の再起動」\(53 ページ\)](#) を参照)。

6.4 WebLogic Server の再起動

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。再起動する必要があるのは、WebLogic 管理対象サーバーと WebLogic 管理サーバーです。ノードマネージャは再起動の必要はありません。

▶ WebLogic 管理対象サーバーを停止するには

1. WebLogic 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 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 管理サーバーを停止するには

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 管理サーバーを再起動するには

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 管理対象サーバーを再起動するには

1. WebLogic Administration Server が起動したら、WebLogic Administration Console にログインします。
2. Change Center で、「Lock & Edit」をクリックします。
3. 「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。

4. 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。
5. 「Start」をクリックして、「Yes」をクリックします。

6.5 次の手順

LiveCycle Configuration Manager を実行して、データベースを初期化し、コンポーネントおよび LiveCycle ES2 アーカイブファイル (LCA) をデプロイする必要があります。必要に応じて、コンポーネントおよびアーカイブファイルのデプロイメントを検証することもできます ([「LiveCycle ES2 の設定、デプロイ」 \(31 ページ\)](#) を参照)。

この章では、LiveCycle 管理コンソールにアクセスしてアプリケーションサーバーのログファイルを確認することにより、デプロイメントの検証を行う方法について説明します。また、インストール、設定およびアプリケーションサーバーへのデプロイを完了した LiveCycle ES2 モジュールおよびサービスを使用して処理を開始する方法についても説明します。

- [「アプリケーションサーバーの再起動」 \(55 ページ\)](#)
- [「日付、時刻およびタイムゾーンの設定」 \(55 ページ\)](#)
- [「デプロイメントの確認」 \(56 ページ\)](#)
- [「LiveCycle ES2.5 Solution Accelerator のインストール」 \(57 ページ\)](#) (オプション)
- [「WebLogic Server サーバー起動引数の設定」 \(57 ページ\)](#)
- [「モジュールの Web アプリケーションへのアクセス」 \(58 ページ\)](#)
- [「User Management へのアクセス」 \(60 ページ\)](#)
- [「LiveCycle PDF Generator ES2 または 3D ES2 の設定」 \(60 ページ\)](#)
- [「Content Services ES2 の設定」 \(67 ページ\)](#)
- [「LiveCycle ES2 の LDAP アクセス設定」 \(67 ページ\)](#)
- [「FIPS モードの有効化」 \(69 ページ\)](#)
- [「HTML 電子署名の設定」 \(69 ページ\)](#)
- [「Document Management サービスの設定」 \(70 ページ\)](#)
- [「Connector for EMC Documentum サービスの設定」 \(70 ページ\)](#)
- [「Connector for IBM FileNet サービスの設定」 \(74 ページ\)](#)
- [「Connector for IBM Content Manager の設定」 \(84 ページ\)](#)
- [「システムイメージバックアップの実行」 \(88 ページ\)](#)
- [「LiveCycle ES2 のアンインストール」 \(88 ページ\)](#)

この章の設定を行った後、LiveCycle ES2 の開発環境および実稼働環境の設定について詳しくは、[LiveCycle ES2 管理ヘルプ](#)を参照してください。

7.1 アプリケーションサーバーの再起動

LiveCycle ES2 を初めてデプロイするとき、サーバーはデプロイメントモードになっています。このモードでは、ほとんどのモジュールがメモリ内に配置されます。このため、メモリの消費量が高く、サーバーは実稼働に適した状態ではありません。アプリケーションサーバーを再起動して、サーバーをクリーンな状態に戻す必要があります。

7.2 日付、時刻およびタイムゾーンの設定

LiveCycle ES2 環境に接続するすべてのサーバーで日付、時刻およびタイムゾーンを設定することで、時間に依存するモジュール (LiveCycle Digital Signatures ES2 や LiveCycle Reader Extensions ES2 など) が正常に機能するようになります。例えば、未来の時間に作成された署名は、有効になりません。

同期を必要とするサーバーは、データベースサーバー、LDAP サーバー、HTTP サーバーおよび J2EE サーバーです。

7.3 デプロイメントの確認

LiveCycle 管理コンソールにログインすることで、デプロイメントの検証を行うことができます。正常にログインできる場合は、LiveCycle ES2 がアプリケーションサーバーで実行されており、データベースにデフォルトのユーザーが作成されています。

アプリケーションサーバーのログファイルを確認して、コンポーネントが正しくデプロイされたことを確認したり、発生する可能性のあるデプロイメントの問題の原因を特定したりすることができます。

7.3.1 LiveCycle 管理コンソールへのアクセス

LiveCycle 管理コンソールは、各種設定ページにアクセスするための Web ベースのポータルです。これらの設定ページでは、LiveCycle ES2 の動作を制御する実行時プロパティを設定できます。LiveCycle 管理コンソールにログインすると、User Management、監視フォルダ、電子メールクライアント設定、および他のサービスの管理設定オプションにアクセスできます。LiveCycle 管理コンソールでは、管理者がアーカイブの管理およびサービスの実稼働環境へのデプロイに使用するアプリケーションおよびサービスにもアクセスできます。

ログインする場合のデフォルトのユーザー名とパスワードは、それぞれ administrator と password です。初回のログイン後は、User Management にアクセスしてパスワードを変更してください。アップグレードした環境の場合は、ユーザー名とパスワードは、管理者が LiveCycle ES (8.x) を設定したときに指定したものと同じです。

LiveCycle 管理コンソールにアクセスするには、デプロイ済みの LiveCycle ES2 がアプリケーションサーバー上で実行されている必要があります。

LiveCycle 管理コンソールの使用について詳しくは、[LiveCycle ES2 管理ヘルプ](#)を参照してください。

▶ LiveCycle 管理コンソールにアクセスするには

1. Web ブラウザに次の URL を入力します。

```
http://[host name]:[port]/adminui
```

WebLogic Server のデフォルトのポート番号は 7001 です (新しい管理対象サーバーを作成した場合は、異なるポートが設定されている可能性があります。管理対象サーバーはデフォルトのポートである 8001 を使用します)。

2. 「ユーザー名」ボックスに administrator と入力し、「パスワード」ボックスに password と入力します。
3. ログイン後、「サービス」をクリックして、サービスの管理ページにアクセスするか、「設定」をクリックして、様々なモジュールの設定を管理できるページにアクセスします。

7.3.2 デフォルトのパスワードの変更

LiveCycle ES2 では、インストール時に 1 つ以上のデフォルトのユーザーが作成されます。これらのユーザーのパスワードは製品資料に記載され、公開されています。セキュリティ要件に応じて、このデフォルトのパスワードを変更する必要があります。

LiveCycle ES2 管理者のユーザーパスワードは、デフォルトで「password」に設定されています。このパスワードは、LiveCycle 管理コンソール/設定/User Management で変更する必要があります。

7.3.3 ログファイルの表示

ログファイルを表示して、LiveCycle ES2 クラスタが正常に機能していることを確認できます。実行時や起動時のエラーなどのイベントは、アプリケーションサーバーのログファイルに記録されます。アプリケーションサーバーへのデプロイ中に何らかの問題が発生した場合には、ログファイルを参照して問題を見つけることができます。ログファイルは、テキストエディタを使用して開くことができます。

▶ LiveCycle ES2 クラスタを確認するには

1. クラスタのすべてのアプリケーションサーバーが起動していることを確認します。
2. Gemfire.log ファイルを表示します。このファイルは、ディレクトリ [system temp]/adobewl_<domain name>_<server name>/Caching にあります。次のようなメッセージで、キャッシュがクラスタのすべてのサーバーに接続されていることを確認できます。

```
[info 2008/01/22 14:24:31.109 EST GemfireCacheAdapter <UDP mcast receiver> nid=0x5b611c24] Membership: received new view [server-0:2916|1] [server-0:2916/2913, server-1:3168/3165] [info 2008/01/22 14:24:31.125 EST GemfireCacheAdapter <View Message Processor> nid=0x7574d1dc] DMMembership: admitting member <server-1:3168/3165>; now there are 2 non-admin member(s)
```

注意： 非管理メンバーの数（上記のログエントリの例では 2）がクラスタ内のメンバーの数と一致することを確認してください。一致しない場合は、クラスタ内の一部のメンバーがキャッシュに接続されていないことを意味します。

7.4 LiveCycle ES2.5 Solution Accelerator のインストール

9.5 の新機能

LiveCycle ES2.5 Solution Accelerator のインストールを予定している場合は、最初に LiveCycle ES2 サービスパック 2 以降を適用してから LiveCycle ES2.5 Solution Accelerator をインストールする必要があります。ただし、LiveCycle ES2.5 Solution Accelerator のインストール後に LiveCycle Configuration Manager を再実行する必要がありますことに注意してください。

Solution Accelerator のインストールについて詳しくは、『[LiveCycle ES2.5 Solution Accelerator のインストールおよびデプロイ](#)』を参照してください。

7.5 WebLogic Server サーバー起動引数の設定

LiveCycle Output ES2 または LiveCycle Forms ES2 をインストールした場合、クラスタの各 WebLogic Server インスタンスでサーバー起動引数の PoolMax 値を設定して、LiveCycle ES2 が正常に起動するようにする必要があります。

▶ サーバー起動引数を設定するには：

1. WebLogic Server Administration Console の「Domain Structure」で、Environment / Servers をクリックし、右側のウィンドウでサーバーの名前をクリックします。
2. 「Configuration」タブをクリックし、「Server Start」をクリックします。
3. Change Center で、「Lock & Edit」をクリックします。

4. 「Arguments」ボックスに、次のテキストを追加します。

```
-Dcom.adobe.convertpdf.bmc.POOL_MAX=<value>  
-Dcom.adobe.xmlform.bmc.POOL_MAX=<value>
```

注意： 上記のテキストブロックをテキストエディタにコピーし、改行をすべて削除して、すべての <value> を PoolMax の適切な値に置き換えます。

PoolMax のデフォルト値は 4 です。使用環境のハードウェア構成と予想される使用量に基づいて値を設定します。一般的に、上限はサーバー上の CPU 数の 2 倍に設定します。PoolMax の下限を使用可能な CPU の数以上にし、上限はサーバーの負荷パターンによって決めることをお勧めします。

5. 「Save」をクリックします。
6. クラスタ内のサーバーごとに、手順 4～5 を繰り返します。

7.6 モジュールの Web アプリケーションへのアクセス

LiveCycle ES2 のデプロイ後には、次のモジュールに関連付けられた Web アプリケーションにアクセスできます。

- LiveCycle Reader Extensions ES2
- LiveCycle Workspace ES2
- LiveCycle Content Services ES2
- LiveCycle Rights Management ES2
- LiveCycle Business Activity Monitoring ES2

デフォルトの管理者権限を使用して Web アプリケーションにアクセスし、そのアプリケーションにアクセス可能であることを確認したら、他のユーザーがログインしてアプリケーションを使用できるように追加のユーザーとロールを作成できます ([LiveCycle ES2 管理ヘルプ](#)を参照)。

▶ Reader Extensions ES2 Web アプリケーションにアクセスするには

注意： Reader Extensions ES2 秘密鍵証明書を適用して、新しいユーザーのユーザーロールを適用する必要があります ([LiveCycle ES2 管理ヘルプ](#)の「秘密鍵証明書を Reader Extensions ES2 で使用するための設定」を参照)。

1. ローカルデプロイメントを使用している場合は、Web ブラウザを開き、次の URL を入力します。

```
http://localhost:[port]/ReaderExtensions
```

WebLogic の場合、[port] は管理対象 WebLogic Server に割り当てられているポートです。

2. デフォルトのユーザー名とパスワードを使用してログインします。

ユーザー名： administrator

パスワード： password

注意： デフォルトのユーザー名とパスワードを使用してログインするには、管理者またはスーパーユーザーの権限が必要です。他のユーザーが Reader Extensions ES2 Web アプリケーションにアクセスできるようにするには、User Management でユーザーを作成し、そのユーザーに Reader Extensions Web アプリケーションロールを付与する必要があります。

▶ **Workspace ES2 にアクセスするには：**

1. ローカルデプロイメントを使用している場合は、Web ブラウザを開き、次の URL を入力します。

`http://localhost:[port]/workspace`

WebLogic の場合、[port] は管理対象 WebLogic Server に割り当てられているポートです。

2. デフォルトのユーザー名とパスワードを使用してログインします。

ユーザー名： administrator

パスワード： password

▶ **Content Services ES2 Web アプリケーションにアクセスするには：**

注意： 新しいユーザーがこの Web アプリケーションにログインできるように、LiveCycle Contentspace 管理者ロールまたは LiveCycle Contentspace ユーザーロールを適用する必要があります。ロールを適用するには、User Management でユーザーを作成して適切なロールを適用します。

1. ローカルデプロイメントを使用している場合は、Web ブラウザを開き、次の URL を入力します。

`http://localhost:[port]/contentspace`

WebLogic の場合、[port] は管理対象 WebLogic Server に割り当てられているポートです。

2. デフォルトのユーザー名とパスワードを使用してログインします。

ユーザー名： administrator

パスワード： password

7.7 Rights Management ES2 へのアクセス

User Management で LiveCycle Rights Management End User ロールのユーザーを作成し、そのユーザーに関連付けられたログイン情報を使用して Rights Management ES2 の管理者またはエンドユーザーアプリケーションにログインする必要があります。

注意： デフォルトの管理者ユーザーは、Rights Management ES2 エンドユーザー Web アプリケーションにはアクセスできません。ただし、このユーザーのプロファイルに必要なロールを追加できます。新しいユーザーを作成したり、既存のユーザーを修正したりするには、LiveCycle 管理コンソールを使用します。

▶ **Rights Management ES2 エンドユーザー Web アプリケーションにアクセスするには：**

1. Web ブラウザを開き、次の URL を入力します。

`http://[server]:[port]/edc/Login.do`

ここで、[port] は WebLogic 管理対象サーバーに割り当てられているポートです。

▶ **Rights Management ES2 管理 Web アプリケーションにアクセスするには：**

1. Web ブラウザを開き、次の URL を入力します。

`http://[server]:[port]/adminui`

ここで、[port] は管理対象 WebLogic Server に割り当てられているポートです。

2. サービス / LiveCycle Rights Management ES2 をクリックします。Rights Management ES2 でのユーザーおよびロールの設定について詳しくは、[LiveCycle ES2 管理ヘルプ](#)を参照してください。

▶ LiveCycle Rights Management End User ロールをアサインするには

1. LiveCycle 管理コンソールにログインします ([「LiveCycle 管理コンソールへのアクセス」\(56 ページ\)](#) を参照)。
2. 設定 / User Management / ユーザーとグループをクリックします。
3. 「キーワード」ボックスにすべてのと入力し、「条件 2」リストで「グループ」を選択します。
4. 「検索」をクリックし、該当するドメインについて、表示されるリストの「すべてのプリンシパル」をクリックします。
5. 「ロールアサイン」タブをクリックし、「ロールを検索」をクリックします。
6. ロールのリストで、「LiveCycle Rights Management End User」の横にあるチェックボックスを選択します。
7. 「OK」をクリックし、「保存」をクリックします。

7.8 User Management へのアクセス

管理者は、User Management を使用して、1 つ以上のサードパーティユーザーディレクトリと同期するすべてのユーザーおよびグループのデータベースを管理できます。User Management には、Reader Extensions ES2、Workspace ES2、Rights Management ES2、Process Management ES2、Forms ES2、PDF Generator ES2、PDF Generator 3D ES2 および Content Services ES2 などの LiveCycle ES2 モジュールのユーザー認証、権限付与およびユーザー管理の機能があります。

▶ User Management にアクセスするには：

1. LiveCycle 管理コンソールにログインします。
2. ホームページで、設定 / User Management をクリックします。

注意： User Management でのユーザー設定について詳しくは、User Management ページの右上隅にある「User Management ヘルプ」をクリックしてください。

7.9 LiveCycle PDF Generator ES2 または 3D ES2 の設定

LiveCycle PDF Generator ES2 または LiveCycle PDF Generator 3D ES2 を LiveCycle ES2 ソリューションの一部としてインストールしている場合は、次のタスクを実行します。

- [「環境変数の設定」\(61 ページ\)](#)
- [「Adobe PDF プリンタをデフォルトのプリンタとして設定」\(62 ページ\)](#)
- [「Acrobat Professional の設定」\(62 ページ\)](#)
- [「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」\(63 ページ\)](#)
- [「Windows Server 2003 に東アジア文字をインストールするには」\(64 ページ\)](#)
- [「ネットワークプリンタクライアントのインストール」\(65 ページ\)](#)
- [「ネットワークプリンタクライアントのインストール」\(65 ページ\)](#)
- [「監視フォルダのパフォーマンスパラメータの設定」\(66 ページ\)](#)

7.9.1 環境変数の設定

PDF Generator ES2 または PDF Generator 3D ES2 モジュールをインストールして、ファイルを PDF に変換するように設定した場合、一部のファイル形式については、環境変数を手動で設定して、対応するアプリケーションの起動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。次の表に、PDF Generator ES2 または PDF Generator 3D ES2 で環境変数を設定する必要があるネイティブアプリケーションを示します。クラスタ設定では、クラスタ内のすべてのノードに、必要なアプリケーションがインストールされていることを確認してください。

| アプリケーション | 環境変数 | 例 |
|-------------------|------------------|---|
| Acrobat | Acrobat_PATH | C:\Program Files\Adobe\Acrobat 9.0\Acrobat\Acrobat.exe 注意： 環境変数 Acrobat_PATH は大文字と小文字が区別されます。 |
| Adobe FrameMaker® | FrameMaker_PATH | C:\Program Files\Adobe\FrameMaker7.1\FrameMaker.exe |
| メモ帳 | Notepad_PATH | C:\WINDOWS\Notepad.exe 注意： Notepad_PATH 変数は空欄でかまいません。 |
| OpenOffice | OpenOffice_PATH | C:\Program Files\OpenOffice.org 3 |
| Adobe PageMaker® | PageMaker_PATH | C:\Program Files\Adobe\PageMaker 7.0\PageMaker.exe |
| WordPerfect | WordPerfect_PATH | C:\Program Files\WordPerfect Office 12\Programs\wpwin12.exe |

注意：これらの環境変数は、クラスタ内のすべてのノードに対して設定する必要があります。

注意：環境変数 OpenOffice_PATH は、実行ファイルのパスではなく、インストールフォルダのパスに設定します。

Word、PowerPoint、Excel、Visio、Project などの Microsoft Office アプリケーションや AutoCAD のパスを設定する必要はありません。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。

▶ 新しい Windows 環境変数を作成するには

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」セクションで、「新規」ボタンをクリックします。
4. 設定が必要な環境変数の名前（例えば、Photoshop_PATH など）を入力します。このフォルダは、実行ファイルを含むフォルダです。例えば、次のコードを入力します。

```
D:\Program Files\Adobe\Adobe Photoshop CS4\Photoshop.exe
```

▶ Linux または UNIX で PATH 変数を設定するには (OpenOffice のみ) :

- 次のコマンドを入力します。

```
export OpenOffice_PATH=/opt/openoffice.org3
```

7.9.2 Adobe PDF プリンタをデフォルトのプリンタとして設定

Adobe PDF プリンタを、サーバーのデフォルトプリンタに設定する必要があります。Adobe PDF プリンタがデフォルトとして設定されていない場合、PDF Generator ES2 ではファイルを変換できません。

クラスタの場合、Adobe PDF プリンタを、すべてのノードのデフォルトプリンタに設定する必要があります。

▶ デフォルトのプリンタを設定するには

1. スタート/プリンタと FAX を選択します。
2. プリンタと FAX ウィンドウで、「Adobe PDF」を右クリックして「通常使うプリンタに設定」を選択します。

7.9.3 Acrobat Professional の設定

この手順は、LiveCycle ES2 のインストールを完了後に Acrobat 9.0 へのアップグレードまたはインストールを行った場合にのみ必要です。Acrobat のアップグレードは、LiveCycle Configuration Manager を実行してアプリケーションサーバーに LiveCycle ES2 をデプロイした後に実行できます。Acrobat Professional のルートディレクトリは、`[Acrobat root]` と表記します。通常、ルートディレクトリは `C:\Program Files\Adobe\Acrobat 9.0\Acrobat` です。

▶ PDF Generator で使用するために Acrobat を設定するには：

1. Acrobat の以前のバージョンがインストールされている場合、Windows コントロールパネルの「プログラムの追加と削除」を使用して Acrobat をアンインストールします。
2. 次のいずれかを実行します。
 - メディアを使用する場合は、Acrobat の CD を挿入します。
 - ESD ダウンロードを使用している場合は、ESD の場所から Acrobat をダウンロードします。
3. AutoPlay.exe ファイルを実行して、Acrobat をインストールします。
4. LiveCycle ES2 インストールメディアの `additional\scripts` フォルダに移動します。
5. LiveCycle ES2 がインストールされているクラスターノード上で、次のバッチファイルを実行します。

```
Acrobat_for_PDFG_Configuration.bat [LiveCycleES2 root]/pdfg_config
```
6. LiveCycle Configuration Manager を実行する予定のないすべてのクラスターノード上で、次の手順を実行します。
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print に、`Sp1WOW64TimeOut` という名前の新しいレジストリ DWORD エントリを追加します。値を 60000 に設定します。
 - LiveCycle ES2 がインストールされているノードの `[LiveCycle ES2 root]/plugins/x86_win32` ディレクトリから、`PDFGen.api` をコピーして、現在設定しているノードの `[Acrobat root]/plug_ins` ディレクトリに追加します。
 - LiveCycle ES2 がインストールされているノードの `[LiveCycle ES2 root]/plugins/x86_win32` ディレクトリから、`PDFG_PDFE-1 (sRGB) .kfp`、`PDFG_PDFA-1a (sRGB) .kfp` および `PDFG_PDFA-1b (sRGB) .kfp` ファイルをコピーして、現在設定しているノードの `[Acrobat root]/plug-ins/Preflight` ディレクトリに追加します。

7. PDF Generator 3D ES2 もインストールしている場合は、LiveCycle Configuration Manager を実行する予定のないすべてのノード上で、次の追加手順を実行します。

- 環境変数 A3DREVIEWER_MULTI の値に 1 を設定します。
- LiveCycle ES2 がインストールされているノードの [LiveCycle ES2 root]/plugins/x86_win32 ディレクトリから、PDFG3dAddin.dll をコピーして、現在設定しているノードの任意のディレクトリに追加します。
- PDFG3dAddin.dll を追加したディレクトリに移動します。
- 次のコマンドを使用して、PDFG3dAddin.dll を登録します。

```
regsvr32 PDFG3dAddin.dll
```

注意： LiveCycle ES2 がインストールされているノードでは、手順 6 および 7 を実行する必要はありません。手順 5 で指定したバッチファイルを実行します。

8. Acrobat を開き、ヘルプ/アップデートの有無をチェック/環境設定を選択します。

9. 「自動的に新しいアップデートを確認する」を選択解除します。

▶ Acrobat のインストールを検証するには：

1. システム上の PDF ファイルに移動し、そのファイルをダブルクリックして Acrobat で開きます。PDF ファイルが正常に開いた場合は、Acrobat が正しくインストールされています。
2. PDF ファイルを正しく開くことができない場合は、Acrobat をアンインストールしてから再インストールします。

注意： Acrobat のインストール完了後に表示される Acrobat のすべてのダイアログボックスを閉じてから、Acrobat の自動アップデートを無効化してください。

環境変数 Acrobat_PATH を、Acrobat.exe を指すように設定してください（例えば、C:\Program Files\Adobe\Acrobat 9.0\Acrobat\Acrobat.exe）。

▶ ネイティブアプリケーションサポートを設定するには

1. 前の手順で説明したように、Acrobat をインストールして検証します。
2. Adobe PDF プリンタをデフォルトのプリンタとして設定します。
3. (PDF Generator 3D ES2) [LiveCycleES2 root]\plugins\x86_win32\にある DLL ファイルを登録します。PDFG3dAddin.dll にある DLL ファイルを登録します。

7.9.4 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

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： マルチスレッドファイル変換は、Microsoft Word 2007 および Microsoft PowerPoint 2007 のみでサポートされています。Microsoft Excel 2003 および Microsoft Excel 2007 では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、『[LiveCycle ES2 のインストールの準備 \(シングルサーバー\)](#)』ガイドの「マルチスレッドファイル変換の有効化」の節で説明されているタスクを実行する必要があります。

Linux および Solaris ユーザーの場合、ユーザーを作成して、パスワードプロンプトが表示されないように追加設定を指定する必要があります。

▶ ユーザーアカウントの追加：

1. LiveCycle Configuration Manager で、サービス/LiveCycle PDF Generator ES2/ユーザーアカウントをクリックします。
2. 「追加」をクリックし、LiveCycle ES2 サーバー上での管理者権限を持つユーザーのユーザー名とパスワードを入力します。OpenOffice のユーザーを設定する場合は、最初に表示される OpenOffice のアクティベート用のダイアログを閉じます。

注意： OpenOffice のユーザーを設定する場合、OpenOffice のインスタンス数を、この手順で指定したユーザーアカウント数よりも大きくすることはできません。

3. LiveCycle ES2 サーバーを再起動します。

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

1. 上記の説明に従って、ユーザーアカウントを追加します。
2. /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。

3. 「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。

4. サーバーを再起動します。

7.9.5 Windows Server 2003 に東アジア文字をインストールするには

PDF Generator ES2 または PDF Generator 3D ES2 を使用して HTML ファイルを PDF に変換すると、日本語、韓国語、中国語などの一部の東アジア言語は、アラビア語、アルメニア語、グルジア語、ヘブライ語、インド系言語、タイ語、ベトナム語などの右から左方向に書く言語同様、PDF ファイルに表示されない可能性があります。

これらの言語を Windows Server 2003 で正常に表示するには、クライアントおよびサーバーに適切なフォントが必要です。

▶ Windows Server 2003 に東アジア文字をインストールするには

1. スタート/コントロールパネル/地域と言語のオプションを選択します。
2. 「言語」タブをクリックし、「東アジア言語のファイルをインストールする」を選択します。
3. 「詳細設定」タブをクリックし、「コードページ変換テーブル」のすべてのオプションを選択します。

変換された PDF ファイルで依然としてフォントが正しく表示されない場合は、C:\WINDOWS\FONTS ディレクトリに Arial Unicode MS (True Type) フォント (ARIALUNI.TTF) があることを確認します。

7.9.6 ネットワークプリンタクライアントのインストール

PDF Generator ES2 には、クライアントコンピュータに PDF Generator ES2 ネットワークプリンタをインストールするための実行ファイルが含まれています。インストールが完了すると、PDF Generator ES2 プリンタが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。

注意： PDF Generator ES2 ネットワークプリンタクライアント (ウィザード) は、32 ビットの Windows プラットフォームのみでサポートされています。

Windows で PDFG ネットワークプリンタのインストールが失敗する場合や、プリンタを UNIX または Linux のプラットフォームにインストールする場合は、各オペレーティングシステムのネイティブのプリンタ追加ユーティリティを使用して、[「Windows でネイティブのプリンタの追加ウィザードを使用して PDFG ネットワークプリンタを設定するには」 \(65 ページ\)](#) の説明に従って設定してください。

▶ PDF Generator ES2 ネットワークプリンタクライアントをインストールするには

1. PDF Generator ES2 をサーバーに正常にインストールしたことを確認します。
2. Windows クライアントコンピュータから、Web ブラウザに次の URL を入力します。[server] は PDF Generator ES2 をインストールしたサーバーの名前、[port] は使用しているアプリケーションサーバーポートです。

```
http:// [server] : [port] /pdfg-ipp/install
```

3. インターネットポートの構成画面で、「指定されたユーザーアカウントを使う」を選択して、PDFG 管理者またはユーザーのロールを持つ LiveCycle ユーザーの資格情報を指定します。このユーザーには電子メールアドレスも必要です。このアドレスは、変換済みのファイルを受信する際に使用できます。このセキュリティ設定をクライアントコンピュータ上のすべてのユーザーに適用するには、「すべてのユーザーに同じセキュリティ設定を使う」を選択して、「OK」をクリックします。

インストールが終了すると、Adobe LiveCycle PDF Generator ES2 が正常にインストールされたことを示すダイアログボックスが表示されます。

4. 「OK」をクリックします。使用可能なプリンタのリストに Adobe LiveCycle PDF Generator ES2 という名前のプリンタが追加されます。

▶ Windows でネイティブのプリンタの追加ウィザードを使用して PDFG ネットワークプリンタを設定するには

1. スタート/プリンタと FAX をクリックし、「プリンタの追加」をダブルクリックします。
2. 「次へ」をクリックし、「ネットワークプリンタ、またはほかのコンピュータに接続されているプリンタ」を選択して、「次へ」をクリックします。
3. 「インターネット上または自宅 / 会社のネットワーク上のプリンタに接続する」を選択し、次の PDFG プリンタの URL を入力します。[server] はサーバー名、[port] はサーバーを実行しているポート番号です。

```
http:// [server] : [port] /pdfg-ipp/printer
```

4. インターネットポートの構成画面で、「指定されたユーザーアカウントを使う」を選択し、ユーザーの有効な資格情報を指定します。
5. 「プリンタドライバの選択」ボックスで、任意の標準的な PostScript ベースのプリンタドライバ (HP Color LaserJet PS など) を選択します。

6. 適切なオプション（このプリンタをデフォルトに設定するなど）を選択してインストールを完了します。
注意： プリンタの追加の際に使用するユーザーの資格情報では、応答を受信するために、有効な電子メール ID を User Management で設定する必要があります。
 7. 電子メールサービスの sendmail サービスを設定します。サービスの設定オプションで有効な SMTP サーバーと認証情報を指定します。
- ▶ プロキシサーバーのポート転送を使用するように PDF Generator ES2 ネットワークプリンタクライアントをインストールして設定するには
1. CC プロキシサーバーで特定のポートについて LiveCycle ES2 サーバーへのポート転送を設定し、プロキシサーバーレベルで認証を無効にします（LiveCycle ES2 で独自の認証を使用するため）。転送を設定したポートでクライアントがこのプロキシサーバーに接続すると、すべての要求が LiveCycle ES2 サーバーに転送されます。
 2. 次の URL を使用して、PDFG ネットワークプリンタをインストールします。
`http://[proxy server]:[forwarded port]/pdfg-ipp/install.`
 3. PDFG ネットワークプリンタの認証に必要な資格情報を指定します。
 4. PDFG ネットワークプリンタがクライアントマシンにインストールされます。これにより、ファイアウォールで保護されている LiveCycle ES2 サーバーを使用した PDF 変換が可能になります。

7.9.7 監視フォルダのパフォーマンスパラメータの設定

監視フォルダを使用した PDF の変換を実行するための十分なディスク容量がないことを示す `java.io.IOException` エラーメッセージが発生しないように、LiveCycle 管理コンソールで PDF Generator の設定を変更できます。

- ▶ PDF Generator のパフォーマンスパラメータを設定するには：
1. LiveCycle 管理コンソールにログインし、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。
 2. サービスのリストで **PDFGConfigService** を探してクリックし、以下の値を設定します。
 - **PDFG Cleanup Scan Seconds** : 1800
 - **Job Expiration Seconds** : 6000
 - **Server Conversion Timeout** : デフォルト値の 270 から 450 などの大きい値に変更します。
 3. 「保存」をクリックして、サーバーを再起動します。

7.10 LiveCycle Rights Management ES2 の最終設定

Rights Management ES2 では、SSL を使用するようにアプリケーションサーバーを設定する必要があります ([LiveCycle ES2 管理ヘルプ](#)を参照)。

7.11 Content Services ES2 の設定

LiveCycle ES2 のインストールで Content Services ES2 を使用し、LiveCycle Configuration Manager でアプリケーションサーバーを設定していない場合、デプロイ前にアプリケーションサーバーの追加のマニュアル設定を実行する必要があります。アプリケーションサーバーで、次の手順を完了します。

注意： この手順を実行する前に LiveCycle ES2 を設定およびデプロイする必要があります ([「LiveCycle ES2 の設定、デプロイ」 \(31 ページ\)](#) を参照)。この手順では、LiveCycle ES2 をデプロイする場合にのみ作成されるディレクトリが設定されます。

▶ Content Services ES2 の設定を行うには：

1. WebLogic Server Administration Console の「Domain Structure」で、Environment / Servers をクリックし、右側のウィンドウでサーバーの名前をクリックします。
2. 「Configuration」タブをクリックし、「Server Start」をクリックします。
3. Change Center で、「Lock & Edit」をクリックします。
4. 「Arguments」ボックスに、次のテキストを入力します。

```
-Dalfresco.tcp.initial_hosts=<host name>[<port value>], <host name>
[<port value>]
-Dalfresco.cluster.name=lc9_cluster
-Dalfresco.tcp.start_port=<port value>
-Dalfresco.tcp.port_range=3
-Dfile.encoding=utf8
```

注意： 上記のテキストをテキストエディタにコピーし、改行をすべて削除します。<host name> をクラスタ内のコンピュータの名前で置き換えます。<port value> をそのコンピュータのポート番号 (7800 ~ 8000 の任意の値) で置き換えます。

注意： IPv6 ベースのクラスタの場合、<host name> は IPv6 アドレスにマップする必要があります。

警告： クラスタ内のすべての WebLogic Server を含める必要があります。ただし、設定するサーバーは除きます。tcp.initial_hosts= <host name>[<port value>] に続く引数のリストでサーバーの名前を区切るには、カンマを使用します。

5. 「Save」をクリックし、「Activate Changes」をクリックします。
6. クラスタのサーバーごとに、手順 1 ~ 5 を繰り返します。

注意： 垂直クラスタは、Content Services ES2 ではサポートされていません。

7.12 LiveCycle ES2 の LDAP アクセス設定

LiveCycle 7.x 製品用に LDAP を設定している場合、これらの設定はアップグレードプロセス中に移行されるため、この節の手順を実行する必要はありません。LDAP を事前に設定していない場合は、次の手順をガイドラインとして、LDAP を使用した認証をサポートするように User Management を設定することができます。

▶ LDAP に対応するように User Management を設定するには (エンタープライズドメイン)

1. Web ブラウザを開き、http://[host]:[port]/adminui に移動してログインします ([「LiveCycle 管理コンソールへのアクセス」 \(56 ページ\)](#) を参照)。
2. 設定 / ユーザー管理 / ドメインの管理をクリックし、「新規エンタープライズドメイン」をクリックします。

3. 「ID」ボックスにドメインの一意の ID を入力し、「名前」ボックスにドメインの識別名を入力します。
注意： LiveCycle ES2 データベースとして DB2 を使用している場合、ID の許容される最大長は、1 バイト (ASCII) 文字で 100、2 バイト文字で 50、4 バイト文字で 25 文字です ([LiveCycle ES2 管理ヘルプ](#)の「エンタープライズドメインの追加」を参照)。
4. 「認証を追加」をクリックし、「認証プロバイダ」リストで「LDAP」を選択します。
5. 「OK」をクリックします。
6. 「ディレクトリを追加」をクリックし、「プロファイル名」ボックスに、LDAP プロファイルの名前を入力します。
7. 「次へ」をクリックします。
8. 「サーバー」、「ポート」、「SSL」、「バインド」の各ボックスに値を指定し、「ページに次の情報を入力」ボックスで、ディレクトリ設定オプション (「Sun ONE のデフォルト値」など) を選択します また、「名前」ボックスと「パスワード」ボックスで、匿名アクセスが無効な場合に LDAP データベースへの接続に使用する値を指定します ([LiveCycle ES2 管理ヘルプ](#)の「ディレクトリ設定」を参照)。
9. (オプション) 設定をテストします。
 - 「テスト」をクリックします。画面に、サーバーのテストが成功したか、または設定エラーが存在することを示すメッセージが表示されます。
10. 「次へ」をクリックして、必要に応じて、「ユーザー設定」を設定します ([LiveCycle ES2 管理ヘルプ](#)の「ディレクトリ設定」を参照)。
11. (オプション) 設定をテストします。
 - 「テスト」をクリックします。
 - 「検索フィルタ」ボックスで、検索フィルタを確認するか新しい検索フィルタを指定してから、「送信」をクリックします。画面に検索条件に一致するエントリのリストが表示されます。
 - 「閉じる」をクリックしてユーザー設定画面に戻ります。
12. 「次へ」をクリックして、必要に応じて、「グループ設定」を設定します ([LiveCycle ES2 管理ヘルプ](#)の「ディレクトリ設定」を参照)。
13. (オプション) 設定をテストします。
 - 「テスト」をクリックします。
 - 「検索フィルタ」ボックスで、検索フィルタを確認するか新しい検索フィルタを指定してから、「送信」をクリックします。画面に検索条件に一致するエントリのリストが表示されます。
 - 「閉じる」をクリックしてグループの設定画面に戻ります。
14. 「完了」をクリックして新規ディレクトリページを閉じ、「OK」をクリックして終了します。

► **User Management を設定するには (ローカルドメイン)：**

1. Web ブラウザを開き、[http://\[host\]:\[port\]/adminui](http://[host]:[port]/adminui) に移動してログインします ([「LiveCycle 管理コンソールへのアクセス」\(56 ページ\)](#) を参照)。
2. 設定/ユーザー管理/ドメインの管理をクリックし、「新規ローカルドメイン」をクリックします。
3. 該当するボックスにドメイン ID とドメイン名を入力します ([LiveCycle ES2 管理ヘルプ](#)の「ローカルドメインの追加」を参照)。
4. (オプション) 「アカウントロックを有効にする」オプションの選択を解除して、アカウントロックを無効にします。
5. 「OK」をクリックします。

7.13 FIPS モードの有効化

LiveCycle ES2 には FIPS モードがあり、RSA BSAFE Crypto-C 2.1 暗号化モジュールを使用して、データ保護を連邦情報処理規格 (FIPS) 140-2 承認アルゴリズムに限定しています。

LiveCycle ES2 の設定中に LiveCycle Configuration Manager を使用してこのオプションを有効化しなかった場合、または有効化した設定を無効化する場合は、LiveCycle 管理コンソールからこの設定を変更できます。

FIPS モードを変更した場合は、サーバーを再起動する必要があります。

FIPS モードでは、Acrobat のバージョン 7.0 以前はサポートされていません。FIPS モードが有効になっており、パスワードを使用した暗号化およびパスワードの削除の各プロセスに Acrobat 5 の設定が含まれている場合、プロセスは失敗します。

通常、FIPS が有効化されていると、Assembler サービスでは、どのドキュメントにもパスワードの暗号化が適用されません。この処理が試行されると、`FIPSMODEException` が発生し、FIPS モードではパスワードを暗号化できないことが示されます。また、ベースドキュメントがパスワードで暗号化されている場合、`PDFsFromBookmarks` エレメントは FIPS モードではサポートされません。

▶ FIPS モードをオンまたはオフにするには

1. LiveCycle 管理コンソールにログインします。
2. 設定 / コアシステム設定 / 設定をクリックします。
3. 「FIPS を有効にする」を選択して FIPS モードを有効化するか、選択を解除して FIPS モードを無効化します。
4. 「OK」をクリックして、アプリケーションサーバーを再起動します。

注意： LiveCycle ES2 ソフトウェアでは、FIPS の互換性を確認するためのコードの検証を行いません。FIPS 操作モードは、FIPS で承認されたライブラリ (RSA) の暗号化サービスで、FIPS で承認されたアルゴリズムが使用されるようにするために提供されています。

7.14 HTML 電子署名の設定

Forms ES2 の HTML 電子署名機能を使用するには、次の手順を実行します。

▶ HTML 電子署名を有効にするには

1. `[LivecycleES2 root]/deploy/adobe-forms-ds.ear` ファイルをアプリケーションサーバーに手動でデプロイします。
2. LiveCycle 管理コンソールにログインし、サービス / LiveCycle Forms ES2 をクリックします。
3. 「HTML 電子署名が有効です」を選択し、「保存」をクリックします。

7.15 Document Management サービスの設定

Content Services ES2 をインストールし、アプリケーションサーバーがデフォルト以外のポートで動作している場合は、Document Management サービスで使用するポートを変更します。

▶ ポートを変更するには

1. LiveCycle 管理コンソールにログインし、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。
2. リストで「DocumentManagementService」を選択します。
3. 「設定」タブの「HTTP ポート」ボックスで、クラスタの各メンバーで使用しているポート番号を（カンマ区切りのリストとして）指定して、「保存」をクリックします。例えば、次のように指定します。
 - WebLogic Cluster : 8001, 8002, 8003

7.16 Connector for EMC Documentum サービスの設定

注意： LiveCycle ES2 でサポートされる EMC Documentum のバージョンは、6.0 および 6.5 のみです。ECM が適切にアップグレードされていることを確認してください。

Connector for EMC Documentum サービスを LiveCycle ES2 ソリューションの一部としてインストールした場合は、次の手順を実行して、Documentum リポジトリに接続するようサービスを設定します。

▶ Connector for EMC Documentum を設定するには

1. [appserverdomain]/bin フォルダにある adobe-component-ext.properties ファイルを開きます（ファイルが存在しない場合は、ファイルを作成します）。
2. 次の Documentum Foundation Classes JAR ファイルを指定する新しいシステムプロパティを追加します。
 - dfc.jar
 - aspectjrt.jar
 - log4j.jar
 - jaxb-api.jar
 - (Connector for EMC Documentum 6.5 のみ)
 - configservice-impl.jar
 - configservice-api.jar

新しいシステムプロパティは、次の形式にする必要があります。

```
[component id].ext=[JAR files and/or folders]
```

例えば、デフォルトの Content Server と Documentum Foundation Classes のインストールを使用して、次のいずれかのシステムプロパティをファイルに追加します。その際、システムプロパティは新しい行に記述し、行中に改行を入れず、末尾は改行にしてください。

- Connector for EMC Documentum 6.0 のみ：

```
com.adobe.livecycle.ConnectorforEMCDocumentum.ext=  
C:/Program Files/Documentum/Shared/dfc.jar,  
C:/Program Files/Documentum/Shared/aspectjrt.jar,  
C:/Program Files/Documentum/Shared/log4j.jar  
C:/Program Files/Documentum/Shared/jaxb-api.jar
```

- Connector for EMC Documentum 6.5 のみ：

```
com.adobe.livecycle.ConnectorforEMCDocumentum.ext=  
C:/Program Files/Documentum/Shared/dfc.jar,  
C:/ProgramFiles/Documentum/Shared/aspectjrt.jar,  
C:/Program Files/Documentum/Shared/log4j.jar,  
C:/Program Files/Documentum/Shared/jaxb-api.jar,  
C:/Program Files/Documentum/Shared/configservice-impl.jar,  
C:/Program Files/Documentum/Shared/configservice-api.jar
```

注意：上記のテキストには、改行が含まれています。このテキストをコピー&ペーストする場合、改行を削除してください。

3. クラスタの各アプリケーションサーバーインスタンスについて、手順 1～2 を繰り返します。
4. Web ブラウザを開き、次の URL を入力します。
http://[host]/:[port]/adminui
5. 次のデフォルトのユーザー名とパスワードを使用してログインします。
ユーザー名： administrator
パスワード： password
6. サービス／LiveCycle ES2 Connector for EMC Documentum／環境設定に移動して、次のタスクを実行します。
 - 必要な Documentum リポジトリ情報のすべてを入力します。
 - Documentum をリポジトリプロバイダとして使用するには、「リポジトリサービスプロバイダ」で「EMC Documentum リポジトリプロバイダ」を選択し、「保存」をクリックします（詳しくは、[LiveCycle ES2 管理ヘルプ](#) ページの右上隅にあるヘルプリンクをクリックしてください）。
7. (オプション) サービス／LiveCycle ES2 Connector for EMC Documentum／リポジトリ証明書の設定に移動して、「追加」をクリックし、Docbase 情報を指定して、「保存」をクリックします（詳しくは、右上隅の「ヘルプ」をクリックしてください）。
8. アプリケーションサーバーが現在実行されていない場合は、サーバーを起動します。実行されている場合は、サーバーを停止し、再起動します。
9. Web ブラウザを開き、次の URL を入力します。
http://[host]/:[port]/adminui
10. 次のデフォルトのユーザー名とパスワードを使用してログインします。
ユーザー名： administrator
パスワード： password
11. サービス／アプリケーションおよびサービス／サービスの管理に移動して、次のサービスを選択します。
 - EMCDocumentumAuthProviderService
 - EMCDocumentumContentRepositoryConnector
 - EMCDocumentumRepositoryProvider
12. 「起動」をクリックします。サービスのいずれかが正常に起動されない場合は、手順 6 で入力した設定を確認します。

13. 次のいずれかのタスクを実行します。

- Documentum Authorization サービス (EMCDocumentumAuthProviderService) を使用して、Workbench ES2 のリソースビューで Documentum リポジトリから内容を表示するには、この手順を続行します。Documentum Authorization サービスを使用すると、デフォルトの LiveCycle ES2 認証が上書きされるため、Documentum 秘密鍵証明書を使用して Workbench ES2 にログインするよう設定する必要があります。
- LiveCycle ES2 リポジトリを使用するには、LiveCycle ES2 の上級管理者の秘密鍵証明書 (デフォルトは「Administrator」と「password」) を使用して Workbench ES2 にログインします。

これで、この手順に必要なステップを完了しました。この場合、手順 6 で指定した秘密鍵証明書を使用してデフォルトリポジトリにアクセスし、デフォルトの LiveCycle ES2 認証サービスを使用します。

14. LiveCycle 管理コンソールにログインし、設定 / User Management / ドメインの管理をクリックします。

15. 「新規エンタープライズドメイン」をクリックして、ドメイン ID と名前を入力します。ドメイン ID は、ドメインの固有の識別子です。名前は、ドメインの識別名です。

注意：(WebSphere および WebLogic のみ) LiveCycle ES2 データベースとして DB2 を使用している場合、ID の許容される最大長は、1 バイト (ASCII) 文字で 100 文字、2 バイト文字で 50 文字、4 バイト文字で 25 文字です ([LiveCycle ES2 管理ヘルプ](#)の「エンタープライズドメインの追加」を参照)。

16. カスタム認証プロバイダを追加します。

- 「認証を追加」をクリックします。
- 「認証プロバイダ」リストで「カスタム」を選択します。
- 「EMCDocumentumAuthProvider」を選択し、「OK」をクリックします。

17. LDAP 認証プロバイダを追加します。

- 「認証を追加」をクリックします。
- 「認証プロバイダ」リストで「LDAP」を選択し、「OK」を選択します。

18. LDAP ディレクトリを追加します。

- 「ディレクトリを追加」をクリックします。
- 「プロファイル名」ボックスに一意の名前を入力し、「次へ」をクリックします。
- 「サーバー」、「ポート」、「SSL」、「バインド」および「ページに次の情報を入力」オプションの値を指定します。「バインド」オプションで「ユーザー」を選択する場合は、「名前」と「パスワード」フィールドにも値を指定する必要があります。
- (オプション) 必要に応じてベースドメイン名を取得するには、「BaseDN を取得」を選択します。
- 「次へ」をクリックし、ユーザー設定を指定して「次へ」をクリックし、必要に応じてグループ設定を指定して「次へ」をクリックします。

設定について詳しくは、ページの右上隅にある「User Management ヘルプ」をクリックします。

19. 「OK」をクリックして「ディレクトリを追加」ページを閉じ、もう一度「OK」をクリックします。

20. 新しいエンタープライズドメインを選択し、「今すぐ同期」をクリックします。LDAP ネットワークのユーザーとグループ数および接続の速度によって、同期処理には数分かかる場合があります。

(オプション) 同期のステータスを確認するには、「更新」をクリックし、「現在の同期の状態」列にステータスを表示します。

21. 設定 / User Management / ユーザーとグループをクリックします。

22. LDAP から同期されたユーザーを検索し、以下のタスクを実行します。

- 1 つ以上のユーザーを選択し、「ロールをアサイン」をクリックします。
- 1 つ以上の LiveCycle ES2 ロールを選択し、「OK」をクリックします。
- 「OK」をもう一度クリックして、ロールアサインを確認します。

ロールをアサインするすべてのユーザーについて、この手順を繰り返します。詳しくは、ページの右上隅にある「User Management ヘルプ」をクリックします。

23. 次の秘密鍵証明書を使用して Workbench ES2 を起動し、ログインします。

ユーザー名: [username]@[repository_name]

パスワード: [password]

これで、Documentum リポジトリが Workbench ES2 のリソースビューに表示されます。

[username]@[repository_name] を使用してログインしない場合、Workbench ES2 では、手順 6 で指定したデフォルトリポジトリへのログインが試行されます。

24. (オプション) Connector for EMC Documentum の LiveCycle ES2 サンプルをインストールする場合、「Samples」という名前の Documentum リポジトリを作成してその中にインストールします。

Connector for EMC Documentum サービスを設定したら、Documentum リポジトリを使用した Workbench ES2 の機能の正しい設定について、[LiveCycle ES2 管理ヘルプ](#)を参照することをお勧めします。

7.17 Documentum リポジトリでの XDP MIME 形式の作成

ユーザーが Documentum リポジトリから XDP ファイルを取得し、保存できるようにするには、次のタスクのいずれかを実行する必要があります。

- ユーザーがアクセスする XDP ファイルが置かれている各リポジトリに、対応する XDP 形式を作成します。
- Documentum リポジトリにアクセスするときに Documentum 管理者アカウントを使用するように、Connector for EMC Documentum サービスを設定します。この場合、Connector for EMC Documentum サービスでは必要に応じて XDP 形式が使用されます。

▶ Documentum 管理者アカウントを使用して Documentum Content Server に XDP 形式を作成するには

1. Documentum 管理者アカウントにログインします。
2. 「形式」をクリックし、ファイル／新規作成／形式を選択します。
3. 次の情報を対応するフィールドに入力します。

名前: xdp

デフォルトのファイル拡張子: xdp

Mime タイプ: application/xdp

4. ユーザーが XDP ファイルを保存する他のすべての Documentum リポジトリについて、手順 1 ~ 3 を繰り返します。

▶ Documentum 管理者アカウントを使用するように Connector for EMC Documentum サービスを設定するには

1. Web ブラウザを開き、次の URL を入力します。

http://[host]/:[port]/adminui

2. 次のデフォルトのユーザー名とパスワードを使用してログインします。

ユーザー名 : administrator

パスワード : password

3. サービス / LiveCycle ES2 Connector for EMC Documentum / 環境設定 をクリックします。
4. 「Documentum プリンシパル秘密鍵証明書に関する情報」領域で、次の情報を更新し、「保存」をクリックします。

ユーザー名 : [Documentum 管理者のユーザー名]

パスワード : [Documentum 管理者のパスワード]

5. 「リポジトリ証明書の設定」をクリックして、リストからリポジトリを選択します。リストにない場合は、「追加」をクリックします。
6. 対応するフィールドで適切な情報を指定して、「保存」をクリックします。

リポジトリ名 : [リポジトリ名]

リポジトリ証明書のユーザー名 : [Documentum 管理者のユーザー名]

リポジトリ証明書のパスワード : [Documentum 管理者のパスワード]

7. ユーザーが XDP ファイルを保存するすべてのリポジトリについて、手順 5 ~ 6 を繰り返します。

7.18 Connector for IBM FileNet サービスの設定

LiveCycle ES2 でサポートされる IBM FileNet のバージョンは、4.0 および 4.5 のみです。ECM が適切にアップグレードされていることを確認してください。

Connector for IBM FileNet サービスを LiveCycle ES2 ソリューションの一部としてインストールした場合は、次の手順を実行して、FileNet のオブジェクトストアに接続するようサービスを設定する必要があります。

FileNet 4.x を使用して Connector for IBM FileNet サービスを設定するには、次のいずれかの手順を選択します。

- [「FileNet 4.x および CEWS トランスポートを使用してコネクタを設定するには」 \(74 ページ\)](#)
- [「FileNet 4.x および EJB トランスポートを使用してコネクタを設定するには」 \(79 ページ\)](#)

▶ FileNet 4.x および CEWS トランスポートを使用してコネクタを設定するには

1. 次の操作を実行します。
2. [appserverdomain]/config/config.xml ファイルを探し、そのバックアップコピーを作成します。
3. WebLogic Server Administration Console の「Domain Structure」で、Environment / Servers をクリックし、右側のウィンドウでサーバーの名前をクリックします。
4. 「Configuration」タブをクリックし、「Server Start」をクリックします。
5. Change Center で、「Lock & Edit」をクリックします。
6. 「Arguments」ボックスに次のテキストを入力し、「Save」をクリックします。

`-Dwasp.location= <configuration files location>`

例えば、デフォルトの FileNet Application Engine インストールを Windows オペレーティングシステムで使用する場合、次の Java オプションを追加します。

`-Dwasp.location=C:/Progra~1/FileNet/AE/CE_API/wsi`

7. [appserverdomain]/bin フォルダにある adobe-component-ext.properties ファイルを開きます (ファイルが存在しない場合は、ファイルを作成します)。
8. 次の FileNet Application Engine JAR ファイルの場所を指定する、新しいシステムプロパティを追加します。
 - javaapi.jar
 - log4j-1.2.13.jar
 - soap.jar
 - wasp.jar
 - builtin_serialization.jar (FileNet 4.0 のみ)
 - wsdl_api.jar
 - jaxm.jar
 - jaxrpc.jar
 - saaj.jar
 - jetty.jar
 - runner.jar
 - p8cjares.jar
 - Jace.jar
 - (オプション) pe.jar

注意： pe.jar ファイルは、デプロイメントで IBMFileNetProcessEngineConnector サービスを使用する場合にのみ追加します。新しいシステムプロパティには、次の構造を反映させる必要があります。

```
[component id].ext=[JAR files and/or folders]
```

例えば、デフォルトの FileNet Application Engine インストールを Windows オペレーティングシステムで使用する場合、次のシステムプロパティをファイルに追加します。その際、システムプロパティは新しい行に記述し、行中に改行を入れず、末尾で改行してください。

注意： 次のテキストには、改行が含まれています。このテキストを、このドキュメント以外の場所にコピーする場合は、新しい場所に貼り付けるときに改行を削除してください。

```
com.adobe.livecycle.ConnectorforIBMFileNet.ext=  
C:/Program Files/FileNet/AE/CE_API/lib2/javaapi.jar,  
C:¥Program Files¥FileNet¥AE¥CE_API¥lib2¥log4j-1.2.13.jar,  
C:/Program Files/FileNet/AE/Workplace/WEB-INF/lib/soap.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/wasp.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/builtin_serialization.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/wsdl_api.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/jaxm.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/jaxrpc.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/saaj.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/jetty.jar,  
C:/Program Files/FileNet/AE/CE_API/wsi/lib/runner.jar,  
C:/Program Files/FileNet/AE/CE_API/lib2/p8cjares.jar,  
C:/Program Files/FileNet/AE/CE_API/lib/Jace.jar,  
C:/Program Files/FileNet/AE/Workplace/WEB-INF/lib/pe.jar
```

注意： C:¥Program Files¥FileNet¥AE¥Workplace¥WEB-INF¥lib¥pe.jar は、実際のデプロイメントで IBMFileNetProcessEngineConnector サービスを使用している場合のみ追加します。

注意： FileNet 4.5 の場合は、C:/Program Files/FileNet/AE/CE_API/wsi/lib/builtin_serialization.jar, の行を削除します。

9. (FileNet Process Engine Connector のみ) 次の手順で、プロセスエンジンの接続プロパティを設定します。

- テキストエディタを使用してファイルを作成し、次のコンテンツを 1 行で入力します。末尾で改行してください。

```
RemoteServerUrl = cemp:http://[contentserver_IP]:[contentengine_port]/
wsi/FNCEWS40DIME/
```

- このファイルを WcmApiConfig.properties という名前で別のフォルダに保存して、そのフォルダの場所を adobe-component-ext.properties ファイルに追加します。

例えば、このファイルを c:\pe_config\WcmApiConfig.properties として保存して、パス c:\pe_config を adobe-component-ext.properties ファイルに追加します。

注意： ファイル名では大文字と小文字が区別されます。

10. カスタム JAAS 設定ファイルが使用されている場合、次の行をカスタム JAAS 設定ファイルに追加します。

```
FileNetP8 {weblogic.security.auth.login.UsernamePasswordLoginModule
  required authOnLogin=true;};
FileNetP8WSI {com.filenet.api.util.WSILoginModule required;};
FileNetP8Engine
  {weblogic.security.auth.login.UsernamePasswordLoginModule required
  authOnLogin=true;};
FileNetP8Server
  {weblogic.security.auth.login.UsernamePasswordLoginModule required
  authOnLogin=true;};
```

ヒント： カスタム JAAS 設定ファイルが使用されているかどうかは、アプリケーションサーバーの start コマンドのプロパティ -Djava.security.auth.login.config の値で識別できます。

11. (FileNet Process Engine Connector のみ) この FileNet Process Engine Connector をデプロイメントで使用する場合、設定に応じて次のいずれかの手順を実行します。

- カスタム JAAS ファイルをデプロイメントで使用する場合、次の行をカスタム JAAS 設定ファイルに追加します。

```
FileNetP8 {com.filenet.api.util.WSILoginModule required;};
```

- カスタム JAAS ファイルをデプロイメントで使わない場合、テキストエディタを使用して次のファイルを作成します。

```
FileNetP8 {com.filenet.api.util.WSILoginModule required;};
```

このファイルを jaas.conf.WSI として保存して、その場所を、WebLogic Server start コマンドに次の Java オプションとして追加します。

```
-Djava.security.auth.login.config=<JAAS file location>
```

例えば、このファイルを C:\pe_config\jaas.conf.WSI として保存して、次の Java オプションを追加します。

```
-Djava.security.auth.login.config=C:/pe_config/jaas.conf.WSI
```

12. config.xml ファイルを開き、管理対象サーバーのユーザードメインの <credential-encrypted> 値を探します。このエレメントに値が設定されていない場合は、手順 2 で作成した config.xml ファイルのバックアップコピーを開き、<credential-encrypted> 値をコピーします。

13. 新しい config.xml ファイルに値を貼り付けてから、このファイルを保存して閉じます。

14. アプリケーションサーバーが実行されていない場合は、サーバーを起動します。実行されている場合は、サーバーを停止し、再起動します。

15. クラスタの各 WebLogic Server インスタンスについて、手順 3 ~ 14 を繰り返します。
16. Web ブラウザを開き、次の URL を入力します。
`http:// [host] : [port] /adminui`
17. デフォルトのユーザー名とパスワードを使用してログインします。
ユーザー名 : administrator
パスワード : password
18. サービス / LiveCycle ES2 Connector for IBM FileNet をクリックします。
19. 必要なすべての FileNet リポジトリ情報を入力し、「リポジトリサービスプロバイダ」の下で「IBM FileNet リポジトリプロバイダ」を選択します。
オプションのプロセスエンジンサービスをデプロイメントで使用する場合、「プロセスエンジン設定」領域で「プロセスエンジンコネクタサービスを使用」を選択し、プロセスエンジンの各設定を指定します。詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。
注意 : この手順で指定する秘密鍵証明書は、IBM FileNet リポジトリサービスを後で起動するときに検証されます。秘密鍵証明書が無効な場合はエラーが発生し、サービスは起動されません。
20. 「保存」をクリックして、サービス / アプリケーションおよびサービス / サービスの管理に移動します。
21. 次の各サービスの横にあるチェックボックスを選択して「開始」をクリックします。
 - IBMFileNetAuthProviderService
 - IBMFileNetContentRepositoryConnector
 - IBMFileNetRepositoryProvider
 - IBMFileNetProcessEngineConnector (設定されている場合)サービスのいずれかが正常に開始しない場合は、手順 19 で入力した設定を確認します。
22. 次のいずれかのタスクを実行します。
 - FileNet Authorization サービス (IBMFileNetAuthProviderService) を使用して Workbench ES2 のリソースビューで FileNet オブジェクトストアからコンテンツを表示するには、この手順を続行します。FileNet Authorization サービスを使用すると、デフォルトの LiveCycle ES2 認証が上書きされるため、FileNet の秘密鍵証明書を使用して Workbench ES2 にログインするよう設定する必要があります。
 - LiveCycle ES2 リポジトリを使用するには、LiveCycle ES2 の上級管理者の秘密鍵証明書 (デフォルトは「Administrator」と「password」) を使用して Workbench ES2 にログインします。この場合、手順 19 で指定した秘密鍵証明書は、デフォルトリポジトリにアクセスするためにデフォルトの LiveCycle ES2 認証サービスを使用します。
23. アプリケーションサーバーを再起動します。
24. LiveCycle 管理コンソールにログインし、設定 / User Management / ドメインの管理をクリックします。
25. 「新規エンタープライズドメイン」をクリックして、ドメイン ID と名前を入力します。ドメイン ID は、ドメインの固有の識別子です。名前は、ドメインの識別名です。
注意 : LiveCycle ES2 データベースとして DB2 を使用している場合、ID の許容される最大長は、1 バイト (ASCII) 文字で 100、2 バイト文字で 50、4 バイト文字で 25 文字です ([LiveCycle ES2 管理ヘルプ](#)の「エンタープライズドメインの追加」を参照)。

26. カスタム認証プロバイダを追加します。
 - 「認証を追加」をクリックします。
 - 「認証プロバイダ」リストで「カスタム」を選択します。
 - 「IBMFileNetAuthProviderService」を選択し、「OK」をクリックします。
27. LDAP 認証プロバイダを追加します。
 - 「認証を追加」をクリックします。
 - 「認証プロバイダ」リストで「LDAP」を選択し、「OK」をクリックします。
28. LDAP ディレクトリを追加します。
 - 「ディレクトリを追加」をクリックし、「プロファイル名」ボックスに一意の名前を入力して、「次へ」をクリックします。
 - 「サーバー」、「ポート」、「SSL」、「バインド」および「ページに次の情報を入力」オプションの値を指定します。「バインド」オプションで「ユーザー」を選択する場合は、「名前」と「パスワード」フィールドにも値を指定する必要があります。
 - (オプション) 必要に応じてベースドメイン名を取得するには、「BaseDN を取得」を選択します。完了したら、「次へ」をクリックします。
 - ユーザー設定を指定し、「次へ」をクリックし、必要に応じてグループ設定を指定して「次へ」をクリックします。

設定について詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックしてください。
29. 「OK」をクリックして「ディレクトリを追加」ページを閉じ、もう一度「OK」をクリックします。
30. 新しいエンタープライズドメインを選択し、「今すぐ同期」をクリックします。LDAP ネットワークのユーザーとグループ数および接続の速度によって、同期処理には数分かかる場合があります。

(オプション) 同期のステータスを確認するには、「更新」をクリックし、「現在の同期の状態」列にステータスを表示します。
31. 設定 / User Management / ユーザーとグループをクリックします。
32. LDAP から同期されたユーザーを検索し、以下のタスクを実行します。
 - 1 つ以上のユーザーを選択し、「ロールをアサイン」をクリックします。
 - 1 つ以上の LiveCycle ES2 ロールを選択し、「OK」をクリックします。
 - 「OK」をもう一度クリックして、ロールアサインを確認します。

ロールをアサインするすべてのユーザーについて、この手順を繰り返します。詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。
33. Workbench ES2 を起動して、IBM FileNet リポジトリ用の次の秘密鍵証明書を使用してログインします。

ユーザー名: [username]@[repository_name]
パスワード: [password]

これで、FileNet オブジェクトストアが Workbench ES2 のリソースビューに表示されます。
[username]@[repository_name] を使用してログインしない場合、Workbench ES2 では、手順 19 で指定したデフォルトリポジトリへのログインが試行されます。
34. (オプション) Connector for IBM FileNet の LiveCycle ES2 サンプルをインストールする場合、Samples という名前の FileNet オブジェクトストアを作成してその中にインストールします。

Connector for IBM FileNet サービスを設定したら、FileNet リポジトリを使用した Workbench ES2 の機能の正しい設定について、[LiveCycle ES2 管理ヘルプ](#)を参照することをお勧めします。

▶ FileNet 4.x および EJB トランスポートを使用してコネクタを設定するには：

1. [appserverdomain]/config/config.xml ファイルを探し、そのバックアップコピーを作成します。
2. [appserverdomain]/bin フォルダにある adobe-component-ext.properties ファイルを開きます (ファイルが存在しない場合は、ファイルを作成します)。
3. 次の FileNet Application Engine JAR ファイルの場所を指定する、新しいシステムプロパティを追加します。
 - javaapi.jar
 - log4j-1.2.13.jar
 - p8cjares.jar
 - Jace.jar
 - (オプション) pe.jar

注意： pe.jar ファイルは、デプロイメントで IBMFileNetProcessEngineConnector サービスを使用する場合にのみ追加します。新しいシステムプロパティには、次の構造を反映させる必要があります。

```
[component id].ext=[JAR files and/or folders]
```

例えば、デフォルトの FileNet Application Engine インストールを Windows オペレーティングシステムで使用する場合、次のシステムプロパティをファイルに追加します。その際、システムプロパティは新しい行に記述し、行中に改行を入れず、末尾で改行してください。

注意： 次のテキストには、改行が含まれています。このテキストを、このドキュメント以外の場所にコピーする場合は、新しい場所に貼り付けるときに改行を削除してください。

```
com.adobe.livecycle.ConnectorforIBMFileNet.ext=  
C:/Program Files/FileNet/AE/CE_API/lib2/javaapi.jar,  
C:¥Program Files¥FileNet¥AE¥CE_API¥lib2¥log4j-1.2.13.jar,  
C:¥Program Files¥FileNet¥AE¥CE_API¥lib2¥p8cjares.jar,  
C:¥Program Files¥FileNet¥AE¥CE_API¥lib¥Jace.jar,  
C:¥Program Files¥FileNet¥AE¥Workplace¥WEB-INF¥lib¥pe.jar
```

注意： C:¥Program Files¥FileNet¥AE¥Workplace¥WEB-INF¥lib¥pe.jar は、実際のデプロイメントで IBMFileNetProcessEngineConnector サービスを使用している場合のみ追加します。

4. (FileNet Process Engine Connector のみ) テキストエディタを使用してファイルを作成し、次のコンテンツを 1 行で入力します。末尾で改行してください。

```
RemoteServerUrl = cemp:http://[contentserver_IP]:[contentengine_port]/  
wsi/FNCEWS40DIME/
```

5. 作成したファイルを WcmApiConfig.properties という名前で別のフォルダに保存して、そのフォルダの場所を adobe-component-ext.properties ファイルに追加します。

例えば、このファイルを c:¥pe_config¥WcmApiConfig.properties として保存して、パス c:¥pe_config を adobe-component-ext.properties ファイルに追加します。

注意： ファイル名では大文字と小文字が区別されます。

6. カスタム JAAS 設定ファイルが使用されている場合、次の行をカスタム JAAS 設定ファイルに追加します。

```
FileNetP8 { weblogic.security.auth.login.  
    UsernamePasswordLoginModule required authOnLogin=true; };  
FileNetP8WSI { com.filenet.api.util.WSILoginModule required; };  
FileNetP8Engine { weblogic.security.auth.login.  
    UsernamePasswordLoginModule required authOnLogin=true; };
```

```
FileNetP8Server { weblogic.security.auth.login.  
    UsernamePasswordLoginModule required authOnLogin=true; };
```

ヒント: カスタム JAAS 設定ファイルが使用されているかどうかは、アプリケーションサーバーの start コマンドのプロパティ `-Djava.security.auth.login.config` の値で識別できます。

7. config.xml ファイルを開き、管理対象サーバーのユーザードメインの `<credential-encrypted>` 値を探します。このエレメントに値が設定されていない場合は、手順 2 で作成した config.xml ファイルのバックアップコピーを開き、`<credential-encrypted>` 値をコピーします。

8. 新しい config.xml ファイルに値を貼り付けてから、このファイルを保存して閉じます。

9. (FileNet Process Engine Connector のみ) この FileNet Process Engine Connector をデプロイメントで使用する場合、設定に応じて次のいずれかの手順を実行します。

- カスタム JAAS ファイルをデプロイメントで使用する場合、次の行をカスタム JAAS 設定ファイルに追加します。

```
FileNetP8 {com.filenet.api.util.WSILoginModule required};
```

- カスタム JAAS ファイルをデプロイメントで使わない場合、テキストエディタを使用して次のファイルを作成します。

```
FileNetP8 {com.filenet.api.util.WSILoginModule required};
```

このファイルを `jaas.conf.WSI` として保存して、その場所を、WebLogic Server start コマンドに次の Java オプションとして追加します。

```
-Djava.security.auth.login.config=<JAAS file location>
```

例えば、このファイルを `C:\pe_config\jaas.conf.WSI` として保存して、次の Java オプションを追加します。

```
-Djava.security.auth.login.config=C:/pe_config/jaas.conf.WSI
```

10. WebLogic Server が実行されていない場合は、サーバーを起動します。実行されている場合は、サーバーを停止し、再起動します。

11. クラスタの各 WebLogic Server インスタンスについて、手順 2 ~ 10 を繰り返します。

12. Web ブラウザを開き、次の URL を入力します。

```
http://localhost:8001/adminui (デフォルトのポートを使用したローカルのデプロイメント)
```

13. 次のデフォルトのユーザー名とパスワードを使用してログインします。

ユーザー名: administrator

パスワード: password

14. サービス / LiveCycle ES2 Connector for IBM FileNet をクリックします。

15. 必要なすべての FileNet リポジトリ情報を入力し、「リポジトリサービスプロバイダ」の下で「IBM FileNet リポジトリプロバイダ」を選択します。

オプションのプロセスエンジンサービスをデプロイメントで使用する場合、「プロセスエンジン設定」領域で「プロセスエンジンコネクタサービスを使用」を選択し、プロセスエンジンの各設定を指定します。詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。

注意: この手順で指定する秘密鍵証明書は、IBM FileNet リポジトリサービスを後で起動するときに検証されます。秘密鍵証明書が無効な場合はエラーが発生し、サービスは起動されません。

16. 「保存」をクリックします。

17. 「FileNet DSC の設定情報」の「ポート番号」ボックスに、Content Engine を実行しているポート番号を入力します。デフォルトのポートは 8001 です。

18. 「保存」をクリックして、サービス/アプリケーションおよびサービス/サービスの管理に移動します。
19. 次の各サービスの横にあるチェックボックスを選択して「開始」をクリックします。
 - IBMFileNetAuthProviderService
 - IBMFileNetContentRepositoryConnector
 - IBMFileNetRepositoryProvider
 - IBMFileNetProcessEngineConnector (設定されている場合)サービスのいずれかが正常に開始しない場合は、手順 15 で入力した設定を確認します。
20. 次のいずれかのタスクを実行します。
 - FileNet Authorization サービス (IBMFileNetAuthProviderService) を使用して Workbench ES2 のリソースビューで FileNet オブジェクトストアからコンテンツを表示するには、この手順を続行します。FileNet Authorization サービスを使用すると、デフォルトの LiveCycle ES2 認証が上書きされるため、FileNet の秘密鍵証明書を使用して Workbench ES2 にログインするよう設定する必要があります。
 - LiveCycle ES2 リポジトリを使用するには、LiveCycle ES2 の上級管理者の秘密鍵証明書 (デフォルトは「Administrator」と「password」) を使用して Workbench ES2 にログインします。この場合、手順 15 で指定した秘密鍵証明書は、デフォルトリポジトリにアクセスするためにデフォルトの LiveCycle ES2 認証サービスを使用します。
21. WebLogic Server を再起動します。
22. LiveCycle 管理コンソールにログインし、設定/User Management/ドメインの管理をクリックします。
23. 「新規エンタープライズドメイン」をクリックして、ドメイン ID と名前を入力します。ドメイン ID は、ドメインの固有の識別子です。名前は、ドメインの識別名です。

注意： LiveCycle ES2 データベースとして DB2 を使用している場合、ID の許容される最大長は、1 バイト (ASCII) 文字で 100、2 バイト文字で 50、4 バイト文字で 25 文字です ([LiveCycle ES2 管理ヘルプ](#)の「エンタープライズドメインの追加」を参照)。
24. カスタム認証プロバイダを追加します。
 - 「認証を追加」をクリックし、「認証プロバイダ」リストで「カスタム」を選択します。
 - 「IBMFileNetAuthProviderService」を選択して、「OK」をクリックします。
25. LDAP 認証プロバイダを追加します。
 - 「認証を追加」をクリックします。
 - 「認証プロバイダ」リストで「LDAP」を選択し、「OK」をクリックします。
26. LDAP ディレクトリを追加します。
 - 「ディレクトリを追加」をクリックし、「プロファイル名」ボックスに一意の名前を入力して、「次へ」をクリックします。
 - 「サーバー」、「ポート」、「SSL」、「バインド」および「ページに次の情報を入力」オプションの値を指定します。「バインド」オプションで「ユーザー」を選択する場合は、「名前」と「パスワード」フィールドにも値を指定する必要があります。
 - (オプション) 必要に応じてベースドメイン名を取得するには、「BaseDN を取得」を選択します。完了したら、「次へ」をクリックします。
 - ユーザー設定を指定し、「次へ」をクリックし、必要に応じてグループ設定を指定して「次へ」をクリックします。詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。

27. 「OK」をクリックして「ディレクトリを追加」ページを閉じ、もう一度「OK」をクリックします。
28. 新しいエンタープライズドメインを選択し、「今すぐ同期」をクリックします。LDAP ネットワークのユーザーとグループ数および接続の速度によって、同期処理には数分かかる場合があります。
(オプション) 同期のステータスを確認するには、「更新」をクリックし、「現在の同期の状態」列にステータスを表示します。
29. 設定 / User Management / ユーザーとグループをクリックします。
30. LDAP から同期されたユーザーを検索し、以下のタスクを実行します。
 - 1 つ以上のユーザーを選択し、「ロールをアサイン」をクリックします。
 - 1 つ以上の LiveCycle ES2 ロールを選択し、「OK」をクリックします。
 - 「OK」をもう一度クリックして、ロールアサインを確認します。ロールをアサインするすべてのユーザーについて、この手順を繰り返します。詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。
31. Workbench ES2 を起動し、次の秘密鍵証明書を使用してログインします。
ユーザー名: [username]@[repository_name]
パスワード: [password]
これで、FileNet オブジェクトストアが Workbench ES2 のリソースビューに表示されます。
[username]@[repository_name] を使用してログインしない場合、Workbench ES2 では、手順 15 で指定したデフォルトリポジトリへのログインが試行されます。
32. (オプション) Connector for IBM FileNet の LiveCycle ES2 サンプルをインストールする場合、Samples という名前の FileNet オブジェクトストアを作成してその中にインストールする必要があります。

Connector for IBM FileNet サービスを設定したら、FileNet リポジトリを使用した Workbench ES2 の機能の正しい設定について、[LiveCycle ES2 管理ヘルプ](#)を参照することをお勧めします。

7.19 SharePoint クライアントアクセスの設定

Microsoft SharePoint クライアントを設定して、LiveCycle ES2 からコンテンツサービスにアクセスできます。そのためには、LiveCycle Configuration Manager を使用して、SharePoint Alfresco Module Package を追加します。SharePoint AMP ファイル (adobe-vti-module.amp) は、[LiveCycleES2 root]¥LiveCycle_ES_SDK¥misc¥ContentServices フォルダにあります。

SharePoint AMP を追加した後で、次の手順を実行します。

7.19.1 share.war ファイルの取得と編集

Alfresco CMS では、share.war ファイルを使用して、Content Services ES2 に接続します。SharePoint クライアントが Content Services ES2 にアクセスできるようにするには、share.war ファイルを変更する必要があります。

1. Alfresco インストールから share.war を取得します。詳しくは、Alfresco のドキュメントを参照してください。
2. ファイルシステム内のディレクトリに share.war ファイルをコピーします。
3. WinRar などのファイルアーカイブユーティリティを使用して、share.war ファイルを開きます。
4. ファイルアーカイブユーティリティのウィンドウから、ファイル WEB-INF/classes/alfresco/webscript-framework-config.xml を抽出し、テキストエディタで開きます。

5. 行 `<endpoint-url>http://localhost:8001/alfresco/s</endpoint-url>` を探して、これを `<endpoint-url>http://localhost:8001/contentspaces/s</endpoint-url>` に変更します。
6. ファイルを保存して閉じます。

7.19.2 share.war ファイルのデプロイ

1. WinRar などのアーカイブユーティリティを使用し、アプリケーションサーバーに適した場所で、アーカイブファイル `adobe-contentservices.ear` を開きます。
 - (WebLogic) : `[appserver domain]¥servers¥<server-name>¥stage¥adobe-contentservices¥`
2. アーカイブユーティリティのウィンドウで開いた `adobe-contentservices.ear` アーカイブに、更新済みの `share.war` ファイルを追加します。
3. ファイルアーカイブユーティリティのウィンドウからローカルファイルシステムのフォルダに、ファイル `application.xml` を抽出して、テキストエディタで開きます。このファイルは、`adobe-contentservices.ear¥META-INF` ディレクトリにあります。

4. `<application>` タグの下に、次の行を追加します。

```
<module id="Share">
  <web>
    <web-uri>share.war</web-uri>
    <context-root>/share</context-root>
  </web>
</module>
```

5. 更新した `application.xml` ファイルを、`adobe-contentservices.ear` アーカイブにコピーして戻します。
6. アーカイブを保存して閉じます。
7. 更新した EAR ファイルをデプロイします。

注意： アプリケーションサーバーの管理コンソールを使用して、更新した EAR ファイルを手動でデプロイする必要があります。

7.20 IPv6 モードでの CIFS の有効化

IPv6 の実装で Content Services ES2 の CIFS を有効にする場合は、LiveCycle ES2 をホストするマシンに補足の IPv6 アドレスを明示的に追加する必要があります。この IPv6 アドレスは、クライアントと同じサブネットに存在する静的 IP アドレスであることが必要です。LiveCycle Configuration Manager を使用して LiveCycle ES2 を設定した後で、次のタスクを実行する必要があります。通常は、EAR ファイルの設定の後で LiveCycle Configuration Manager を一時停止してから、EAR ファイルを編集します。EAR ファイルを編集した後で LiveCycle Configuration Manager に戻り、更新した EAR ファイルを、選択した他の EAR ファイルと共にデプロイできます。

7.20.1 contentservices.war ファイルの編集

1. `[LiveCycleES2 root]¥configurationManager¥export` ディレクトリに移動します。
2. WinRar などのファイルアーカイブユーティリティを使用して、`contentservices.war` ファイルを開きます。
3. ファイルアーカイブユーティリティのウィンドウから、ファイル `contentservices.war¥WEB-INF¥classes¥alfresco¥file-services-custom.xml` を抽出して、テキストエディタで開きます。

4. 次の行を探し、`ipv6="enabled"` を追加して、この行を変更します。

```
<tcpipSMB platforms="linux,solaris,macosx,windows,AIX"/>
```

これを次のように変更します。

```
<tcpipSMB platforms="linux,solaris,macosx,windows,AIX" ipv6="enabled"/>
```

5. ファイルを保存して閉じます。
6. ファイルアーカイブユーティリティのウィンドウからローカルファイルシステムのフォルダに、ファイル `contentservices.war¥WEB-INF¥classes¥alfresco¥extension¥file-servers-properties` を抽出して、テキストエディタで開きます。
7. 行 `cifs.ipv6=disabled` を探して、`cifs.ipv6=enabled` に置き換えます。
8. ファイルを保存して閉じます。
9. 更新した `file-servers-custom.xml` ファイルを、`contentservices.war¥WEB-INF¥classes¥alfresco¥extension¥` にあるアーカイブにコピーします。
10. `contentservices.war` ファイルを保存します。

注意： EAR ファイルを更新した後で、LiveCycle Configuration Manager を使用して、更新した EAR ファイルをデプロイする必要があります。

7.21 Connector for IBM Content Manager の設定

注意： LiveCycle ES2 でサポートされる IBM Content Manager のバージョンは、8.4 のみです。ECM が適切にアップグレードされていることを確認してください。

Connector for IBM Content Manager サービスを LiveCycle ES2 ソリューションの一部としてインストールした場合は、次の手順を実行して、IBM Content Manager データストアに接続するようサービスを設定します。

▶ Connector for IBM Content Manager を設定するには

1. `[appserverdomain]` フォルダで `adobe-component-ext.properties` ファイルを開きます。ファイルが存在しない場合は、ファイルを作成します。
2. 次の IBM II4C JAR ファイル、IBM II4C プロパティファイルを含む Config フォルダ、DB2 Universal Database Client インストールの ZIP ファイルの場所を指定する、新しいシステムプロパティを追加します。
 - `cmb81.jar`
 - `cmbcm81.jar`
 - `cmbicm81.jar`
 - `cmblog4j81.jar`
 - `cmbSDK81.jar`
 - `cmbutil81.jar`
 - `cmbutilicm81.jar`
 - `cmbview81.jar`
 - `cmbwas81.jar`
 - `cmbwcm81.jar`
 - `cmgmt`

注意： cmgmt は JAR ファイルではありません。Windows では、このフォルダはデフォルトで
C:/Program Files/IBM/db2cmv8/ にあります。

- common.jar
- db2jcc.jar
- db2jcc_license_cisuz.jar
- db2jcc_license_cu.jar
- ecore.jar
- ibmjgssprovider.jar
- ibmjseprovider2.jar
- ibmpkcs.jar
- icmrm81.jar
- jcache.jar
- log4j-1.2.8.jar
- xerces.jar
- xml.jar
- xsd.jar

新しいシステムプロパティは、次のような構造になります。

[component id].ext=[JAR files and/or folders]

例えば、デフォルトの DB2 Universal Database Client および I14C インストールを使用する場合、次のシステムプロパティをファイルに追加します。その際、システムプロパティは新しい行に記述し、行中に改行を入れず、末尾で改行してください。

```
C:/Program Files/IBM/db2cmv8/cmgmt,
C:/Program Files/IBM/db2cmv8/java/jre/lib/ibmjseprovider2.jar,
C:/Program Files/IBM/db2cmv8/java/jre/lib/ibmjgssprovider.jar,
C:/Program Files/IBM/db2cmv8/java/jre/lib/ibmpkcs.jar,
C:/Program Files/IBM/db2cmv8/java/jre/lib/xml.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbview81.jar,
C:/Program Files/IBM/db2cmv8/lib/cmb81.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbcm81.jar,
C:/Program Files/IBM/db2cmv8/lib/xsd.jar,
C:/Program Files/IBM/db2cmv8/lib/common.jar,
C:/Program Files/IBM/db2cmv8/lib/ecore.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbicm81.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbwcm81.jar,
C:/Program Files/IBM/db2cmv8/lib/jcache.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbutil81.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbutilicm81.jar,
C:/Program Files/IBM/db2cmv8/lib/icmrm81.jar,
C:/Program Files/IBM/db2cmv8/lib/db2jcc.jar,
C:/Program Files/IBM/db2cmv8/lib/db2jcc_license_cu.jar,
C:/Program Files/IBM/db2cmv8/lib/db2jcc_license_cisuz.jar,
C:/Program Files/IBM/db2cmv8/lib/xerces.jar,
C:/Program Files/IBM/db2cmv8/lib/cmblog4j81.jar,
C:/Program Files/IBM/db2cmv8/lib/log4j-1.2.8.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbsdk81.jar,
C:/Program Files/IBM/db2cmv8/lib/cmbwas81.jar
```

3. アプリケーションサーバーが現在実行されていない場合は、サーバーを起動します。実行されている場合は、サーバーを停止し、再起動します。

これで、IBMCMConnectorService プロパティシートから IBM Content Manager データストアに、「Use User Credentials」をログインモードとして使用して接続できます。

これで、この手順に必要なステップを完了しました。

(オプション) IBMCMConnectorService プロパティシートから IBM Content Manager データストアに、「Use Credentials From Process Context」をログインモードとして使用して接続するには、次の手順を実行します。

▶ 「Use Credentials from process context」 ログインモードを使用して接続するには

1. Web ブラウザを開き、次の URL を入力します。
http://[host]/:[port]/adminui
2. 次のデフォルトのユーザー名とパスワードを使用してログインします。
ユーザー名：administrator
パスワード：password
3. サービス / LiveCycle ES2 Connector for IBM Content Manager / 環境設定をクリックします。
4. 必要なリポジトリ情報のすべてを入力して「保存」をクリックします。IBM Content Manager リポジトリ情報について詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。
5. 次のいずれかのタスクを実行します。
 - IBM Content Manager Authorization サービス (IBMCMProviderService) を使用して IBM Content Manager データストアの内容を Workbench ES2 のプロセスビューで使用するには、この手順を続行します。IBM Content Manager Authorization サービスを使用すると、デフォルトの LiveCycle ES2 認証が上書きされるため、IBM Content Manager 秘密鍵証明書を使用して Workbench ES2 にログインするよう設定する必要があります。
 - Workbench ES2 のプロセスビューで IBM Content Manager データストアからコンテンツを使用するために手順 4 で指定したシステム資格情報を使用するには、LiveCycle ES2 の上級管理者の資格情報 (デフォルトは Administrator と password) を使用して Workbench ES2 にログインします。これで、この手順に必要なステップを完了しました。この場合、手順 4 で指定したシステム秘密鍵証明書は、デフォルトリポジトリにアクセスするためにデフォルトの LiveCycle ES2 認証サービスを使用します。
6. LiveCycle 管理コンソールにログインし、設定 / User Management / ドメインの管理をクリックします。
7. 「新規エンタープライズドメイン」をクリックして、ドメイン ID と名前を入力します。ドメイン ID は、ドメインの固有の識別子です。名前は、ドメインの識別名です。
注意：LiveCycle ES2 データベースとして DB2 を使用している場合、ID の許容される最大長は、1 バイト (ASCII) 文字で 100、2 バイト文字で 50、4 バイト文字で 25 文字です ([LiveCycle ES2 管理ヘルプ](#)の「エンタープライズドメインの追加」を参照)。
8. カスタム認証プロバイダを追加します。
 - 「認証を追加」をクリックします。
 - 「認証プロバイダ」リストで「カスタム」を選択し、「IBMCMAuthProviderService」を選択して、「OK」をクリックします。

9. LDAP 認証プロバイダを追加します。
 - 「認証を追加」をクリックします。
 - 「認証プロバイダ」リストで「LDAP」を選択し、「OK」をクリックします。
10. LDAP ディレクトリを追加します。
 - 「ディレクトリを追加」をクリックします。
 - 「プロファイル名」ボックスに一意の名前を入力し、「次へ」をクリックします。
 - 「サーバー」、「ポート」、「SSL」、「バインド」および「ページに次の情報を入力」オプションの値を指定します。「バインド」オプションで「ユーザー」を選択する場合は、「名前」と「パスワード」フィールドにも値を指定する必要があります。(オプション) 必要に応じてベースドメイン名を取得するには、「BaseDN を取得」を選択します。完了したら、「次へ」をクリックします。
 - ユーザー設定を指定し、「次へ」をクリックし、必要に応じてグループ設定を指定して「次へ」をクリックします。

上記の設定について詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックしてください。

11. 「OK」をクリックして「ディレクトリを追加」ページを閉じ、もう一度「OK」をクリックします。
12. 新しいエンタープライズドメインを選択し、「今すぐ同期」をクリックします。LDAP ネットワークのユーザーとグループ数および接続の速度によって、同期処理には数分かかる場合があります。
13. 同期のステータスを確認するには、「更新」をクリックし、「現在の同期の状態」列にステータスを表示します。
14. 設定 / User Management / ユーザーとグループをクリックします。
15. LDAP から同期されたユーザーを検索し、以下のタスクを実行します。
 - 1 つ以上のユーザーを選択し、「ロールをアサイン」をクリックします。
 - 1 つ以上の LiveCycle ES2 ロールを選択し、「OK」をクリックします。
 - 「OK」をもう一度クリックして、ロールアサインを確認します。

ロールをアサインするすべてのユーザーについて、この手順を繰り返します。詳しくは、ページの右上隅にある「ヘルプ」リンクをクリックします。
16. Workbench ES2 を起動し、IBM Content Manager データストア用の次の秘密鍵証明書を使用してログインします。

ユーザー名: [username]@[repository_name]

パスワード: [password]

これで、IBMCMConnectorService オーケストレーション可能コンポーネントのログインモードが「Use Credentials from process context」と選択されている場合に、Workbench ES2 のプロセスビューで IBM Content Manager データストアを使用できます。

Connector for IBM Content Manager サービスを設定したら、[LiveCycle ES2 管理ヘルプ](#)を参照することをお勧めします。

7.22 システムイメージバックアップの実行

実稼働環境に LiveCycle ES2 をインストールおよびデプロイした後、このシステムを稼働する前に、LiveCycle ES2 を実装したサーバーのシステムイメージバックアップを実行することをお勧めします。このバックアップには、LiveCycle ES2 のデータベース、GDS ディレクトリおよびアプリケーションサーバーを含める必要があります。これは、ハードドライブまたはコンピュータ全体が動作しなくなった場合に、コンピュータの内容の復元に使用できる完全なシステムバックアップです。[LiveCycle ES2 管理ヘルプ](#)の「LiveCycle ES2 のバックアップおよび回復」トピックを参照してください。

7.23 LiveCycle ES2 のアンインストール

[Livecycle ES2 root] ディレクトリにあるアンインストーラでは、LiveCycle ES2 インストーラで作成されたファイルおよびアプリケーションが削除されます。ただし、アンインストーラでは、アプリケーションサーバーにデプロイされた LiveCycle 以外のカスタムアプリケーションのフォルダやファイルは削除されません。アンインストール中に一部のフォルダが削除されない場合は、システムを再起動して、削除されていないフォルダを手動で削除してください。

注意： コマンドラインインターフェイス (CLI) から LiveCycle ES2 をインストールした場合は、CLI を使用して LiveCycle ES2 をアンインストールする必要があります。[「付録 - コマンドラインインターフェイスのインストール」 \(95 ページ\)](#) を参照してください。

警告： アンインストーラを実行すると、製品インストールディレクトリ内のすべての内容が警告メッセージなしに削除されます。操作を進める前に、保存しておく必要があるデータのバックアップを作成してください。

▶ コンピュータからファイルを削除するには

1. アンインストールプログラムを起動します。

- (Windows) 次のいずれかを実行します。
 - Windows のコントロールパネルの「プログラムの追加と削除」を使用して、Adobe LiveCycle ES2 を削除します。
 - 手動アンインストール：
 - アンインストーラが格納されているディレクトリに移動します：`[LiveCycleES2 root]¥ Uninstall_Adobe LiveCycle ES2`
 - `Uninstall Adobe LiveCycle ES2.exe` ファイルをダブルクリックします。
- (UNIX) 次の操作を実行します。
 - 端末から、アンインストールスクリプトが格納されているディレクトリに移動します：`cd /opt/adobe/adobe_lifecycle_es2/Uninstall_Adobe LiveCycle ES2`
 - `./Uninstall Adobe LiveCycle ES2` と入力します (コマンドを `chmod 777` のように入力して、このバイナリを実行ファイルにする必要がある場合があります)。

注意： (UNIX) アンインストールスクリプトが格納されているディレクトリ (`Uninstall_Adobe LiveCycle ES2`) に移動して、そのスクリプトを実行します。ディレクトリ名にスペースが含まれているため、製品をアンインストールするには、ディレクトリの完全パスをコマンドの一部として含める必要があります。

2. アンインストールプログラムの画面の指示に従い、「完了」をクリックします。

3. LiveCycle ES2 を再インストールする場合は、`[LiveCycleES2 root]` ディレクトリに残っているディレクトリおよびファイルをすべて削除してください。

ロードバランシング機能が動作するように WebLogic Server クラスタを設定できます。

ドメインの設定を完了したら、管理サーバーとクラスタ管理対象サーバーを使用して、複数の管理対象サーバーへロードバランシング用の転送を行うためのプロキシ方法を準備する必要があります。WebLogic 用のプロキシサーバーは、次のいずれかのメカニズムを使用して実装できます。

- WebLogic Server のインスタンスおよび HttpClusterServlet ([Using Web Server Plug-Ins with WebLogic Server](#) を参照)。
- Apache、Microsoft IIS、Netscape IPlanet など、WebLogic プロキシプラグインを備えたサードパーティのプロキシサーバー ([Using Web Server Plug-Ins with WebLogic Server](#) を参照)。
- F5 の BigIP ([BIG-IP® Product Family](#) を参照) や、その他の製品 ([Choosing a Hardware Load-Balancing Device](#) を参照) などの、ハードウェアベースのロードバランサ。

注意： クラスタ環境の LiveCycle ES2 でロードバランシング用にサポートされているのは、スティッキーセッションのみです。Apache 用の WebLogic プラグインでは、デフォルトでスティッキーセッションをサポートしています。

WebLogic をロードバランシング用に設定する手順について詳しくは、「[Using Clusters](#)」を参照してください。

8.1 Apache サーバープラグインの設定

Apache HTTP サーバープラグインをインストールおよび設定して、クラスタにロードバランシングを提供することができます。Apache HTTP サーバープラグインは、Apache サーバーで受信した要求を、クラスタの WebLogic Server インスタンスへ転送します。

次の作業を実行します。

- Apache HTTP サーバープラグインをインストールします ([「Apache HTTP サーバープラグインのインストール」\(89 ページ\)](#) を参照)。
- Apache HTTP サーバープラグインを設定します ([「Apache HTTP サーバープラグインの設定」\(90 ページ\)](#) を参照)。
- Apache HTTP サーバープラグインをテストします ([「Apache HTTP サーバープラグインのテスト」\(90 ページ\)](#) を参照)。

8.1.1 Apache HTTP サーバープラグインのインストール

Apache HTTP サーバープラグインは、Solaris、Linux、AIX、Windows および HPUX11 の各プラットフォーム用の共有オブジェクト (.so) として配布されています。WebLogic 10gR3 のインストールファイルに Apache HTTP プラグインは含まれません。これらのプラグインは、Oracle のダウンロードサイトおよびサポートサイトから [個別の .zip ファイル](#) としてダウンロードできます。プラグインのインストール方法について詳しくは、「[Installing the Apache HTTP Server Plug-In as a Dynamic Shared Object](#)」を参照してください。

注意： mod_wl28_20.so ファイルは 128 ビット暗号化に使用します。このプラグインをインストールするには、mod_wl_20.so (または mod_wl28.so) ファイルを [APACHE_HOME]¥modules ディレクトリにコピーします。

8.1.2 Apache HTTP サーバープラグインの設定

Apache HTTP サーバープラグインを設定するには、設定ファイルを変更します。

▶ Apache HTTP サーバープラグインを設定するには

1. テキストエディタを使用して、[APACHE_HOME]¥conf¥httpd.conf を開き、次の行を追加します。

```
LoadModule weblogic_module modules¥mod_wl_20.so
```

2. 次のように IfModule ブロックを追加します。

```
<IfModule mod_weblogic.c>  
WebLogicCluster <Server1>:8001,<Server2>:8001  
MatchExpression *  
</IfModule>
```

注意： コンピュータ名が機能しない場合は、IP アドレスを使用します。コンピュータ名をリストに追加する場合は、それぞれをカンマで区切ります。

3. Apache HTTP サーバーを起動または再起動します。

注意： クラスタに SSL を実装する場合は、[Using Web Server Plug-ins with WebLogic Server](#) を参照し、SSL を使用した Apache プラグインの設定方法に関する情報を確認してください。

8.1.3 Apache HTTP サーバープラグインのテスト

LiveCycle ES2 を使用して Apache HTTP サーバープラグインをテストできます。

▶ Apache HTTP サーバープラグインをテストするには

1. ブラウザを開いて、URL ([Apache サーバーの URL]/adminui) を入力します。
2. LiveCycle ES2 にログインして、クラスタ内の管理対象サーバーのログファイルで、応答成功メッセージを確認し、クラスタのどのサーバーが要求を処理したかを特定します。
3. 新しいブラウザウィンドウと、クラスタの別のサーバーを使用して、手順 1 ~ 2 を繰り返し、要求がラウンドロビン方式でクラスタの別のサーバーに転送されることを確認します。

応答成功メッセージでは、クラスタ内の様々なサーバーにアクセスしたことを確認できます。

ここでは、LiveCycle Output ES2、LiveCycle Forms ES2 および LiveCycle PDF Generator ES2 の高度なチューニングについて説明します。この章に記載されている作業は、上級アプリケーションサーバー管理者が実稼働システムに対してのみ行ってください。

9.1 Output ES2 および Forms ES2 のプールサイズの設定

PoolMax の現在のデフォルト値は 4 です。実際に設定する値は、使用環境のハードウェア構成と予想される使用量によって異なります。

最適な使用方法としては、PoolMax の下限を使用可能な CPU の数以上に設定し、上限はサーバーの負荷パターンによって決めることをお勧めします。一般的に、上限はサーバー上にある CPU コアの数に 2 倍に設定します。

▶ 既存の PoolMax 値を変更するには

1. テキストエディタを使用して、WebLogic 起動スクリプトを編集します。
2. ConvertPdf の以下のプロパティを追加します。
 - `com.adobe.convertpdf.bmc.POOL_MAX=[new_value]`
 - `com.adobe.convertpdf.bmc.MAXIMUM_REUSE_COUNT=5000`
 - `com.adobe.convertpdf.bmc.REPORT_TIMING_INFORMATION=true`
 - `com.adobe.convertpdf.bmc.CT_ALLOW_SYSTEM_FONTS=true`
3. XMLFM の以下のプロパティを追加します。
 - `com.adobe.xmlform.bmc.POOL_MAX=[new_value]`
 - `com.adobe.xmlform.bmc.MAXIMUM_REUSE_COUNT=5000`
 - `com.adobe.xmlform.bmc.REPORT_TIMING_INFORMATION=true`
 - `com.adobe.xmlform.bmc.CT_ALLOW_SYSTEM_FONTS=true`

9.2 LiveCycle PDF Generator ES2

LiveCycle PDF Generator ES2 では、一部の種類の入力ファイルについて、複数の PDF 変換を同時に行うことができます。これは、ステートレスセッションビーンを使用して実行されます。

9.2.1 EJB プールサイズの設定

以下の種類の入力ファイルについて個別のプールサイズを適用するために、4 種類のステートレスセッションビーンがあります。

- Adobe PostScript® および Encapsulated PostScript (EPS) ファイル
- 画像ファイル (BMP、TIFF、PNG、JPEG ファイルなど)
- OpenOffice ファイル
- Microsoft Office ファイル、Photoshop® ファイル、PageMaker® ファイル、FrameMaker® ファイルなど、その他すべての種類のファイル (HTML ファイルを除く)

HTML から PDF への変換時のプールサイズは、ステートレスセッションビーンでは管理されません。

PostScript および EPS ファイルと画像ファイルのデフォルトのプールサイズは 3 に設定され、OpenOffice とその他の種類のファイル (HTML を除く) のデフォルトのプールサイズは 1 に設定されます。

CPU の数や各 CPU 内のコアの数など、使用しているサーバーハードウェア構成に基づいて、PS/EPS と画像のプールサイズを別の値に設定できます。ただし、PDF Generator ES2 を正常に機能させるためには、OpenOffice とその他の種類のファイルのプールサイズを 1 のままにする必要があります。

この節では、サポートされるアプリケーションサーバーのそれぞれについて、PS2PDF (PS から PDF への変換) と Image2PDF (画像から PDF への変換) のプールサイズを設定する方法を説明します。

以下の説明は、次の 2 つの LiveCycle ES2 アプリケーション EAR ファイルがアプリケーションサーバーにデプロイされていることを前提としています。

- adobe-livecycle-weblogic.ear
- adobe-livecycle-native-weblogic-[platform].ear

この [platform] は、オペレーティングシステムに応じて、次のいずれかの文字列に置き換えられます。

- (Windows) x86_win32
- (Linux) x86_linux
- (SunOS™) sparc_sunos

▶ PS2PDF および Image2PDF 用のプールサイズを構成するには：

[LiveCycle ES2 管理ヘルプ](#)の「サービスの管理」の節にある、「Distiller サービスの設定」および「Generate PDF サービスの設定」を参照してください。

9.3 Windows での CIFS の有効化

LiveCycle ES2 をホストする Windows Server 2003 および 2008 マシンを手動で設定する必要があります。Alfresco で CIFS サポートを有効にすると、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できるようになり、ローカルファイルシステムでの場合と同様に各種の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：サーバーには、静的 IP アドレスが必要です。

Windows マシンで、次の作業を行う必要があります。

- [「NetBIOS over TCP/IP の有効化」\(93 ページ\)](#)
- [「他の IP アドレスの追加」\(93 ページ\)](#)
- [「SMB over NetBIOS レジストリの無効化 \(Windows 2003 のみ\)」\(93 ページ\)](#)
- [「ファイルとプリンタの共有の無効化 \(Windows 2008 のみ\)」\(93 ページ\)](#)

9.3.1 NetBIOS over TCP/IP の有効化

LiveCycle ES2 サーバーに接続するクライアントの要求がサーバーホスト名において解決されるように、NetBIOS over TCP/IP を有効にする必要があります。

1. ローカルエリア接続の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、「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 アドレスが使用されてい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、「NetBIOS over TCP/IP を有効にする」を選択します。

9.3.2 他の IP アドレスの追加

1. ローカルエリア接続の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、「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 アドレスが使用されてい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、「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。

9.3.3 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 の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 の SMBDeviceEnabled を 0 に設定します。この値が存在しない場合は、SMBDeviceEnabled という名前の新しい DWORD 値を追加して 0 に設定します。

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

- 「ネットワークの設定」に移動し、「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。

10 | **トラブルシューティング**

LiveCycle ES2 のインストールおよび設定のトラブルシューティングについて詳しくは、『[LiveCycle ES2 のトラブルシューティング](#)』ガイドを参照してください。

A

付録 - コマンドラインインターフェイスのインストール

LiveCycle ES2 には、インストールプログラム用のコマンドラインインターフェイス (CLI) が用意されています。CLI は、LiveCycle ES2 の上級ユーザーが使用したり、インストールプログラムのグラフィカルユーザーインターフェイス (GUI) がサポートされていないサーバー環境で使用したりすることを前提としています。CLI はコンソールモードで実行します。1つのインタラクティブセッションで、すべてのインストール操作を行うことができます。

CLI インストールオプションを使用してモジュールをインストールする前に、次の点を確認してください。

- LiveCycle ES2 の実行に必要なソフトウェアおよびハードウェアが使用環境に含まれている。
- 必要な環境が整っている (『[LiveCycle ES2 のインストールの準備 \(サーバークラスタ\)](#)』を参照)。
- [「製品ファイルのインストール」 \(23 ページ\)](#) の最初のページと [「LiveCycle ES2 モジュールのインストール」 \(22 ページ\)](#) に目を通した。

注意： MySQL はサーバークラスタではサポートされていません。LiveCycle ES2 CLI インストールプログラムで表示される画面で、MySQL オプションを選択しないようにしてください。

この付録には、以下のトピックが含まれています。

- [「LiveCycle ES2 のインストール」 \(95 ページ\)](#)
- [「エラーログ」 \(97 ページ\)](#)
- [「コンソールモードでの LiveCycle ES2 のアンインストール」 \(97 ページ\)](#)
- [「次の手順」 \(98 ページ\)](#)

A.1 LiveCycle ES2 のインストール

ここでは、LiveCycle ES2 の初期のインストールについて説明します。設定およびデプロイメントについては、[「LiveCycle ES2 をデプロイするための設定」 \(30 ページ\)](#) や [「付録 - LCM コマンドラインインターフェイス」 \(99 ページ\)](#) を参照してください。

注意： デプロイメントの際の権限の問題を回避するため、LiveCycle ES2 インストール CLI および LiveCycle Configuration Manager を実行する際には、アプリケーションサーバープロセスを実行するユーザーとしてログインしてください。

インストールプロセスを開始したら、画面の指示に従ってインストールオプションを選択します。各プロンプトに応答しながらインストールを進めてください。前の手順で選択した内容を変更する場合は、back と入力します。quit と入力すれば、いつでもインストールをキャンセルできます。

▶ LiveCycle ES2 をインストールするには

1. コマンドプロンプトを開き、実行可能なインストーラが含まれるインストールメディアまたはハードディスクのフォルダに移動します。
 - (Windows) livecycle_server¥9.0¥Disk1¥InstData¥Windows¥VM
 - (Windows 64 ビット版) livecycle_server¥9.0¥Disk1¥InstData¥Windows_64bit¥VM
 - (Linux) livecycle_server/9.0/Disk1/InstData/Linux/NoVM
 - (Solaris) livecycle_server/9.0/Disk1/InstData/Solaris/NoVM

2. 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。

- (Windows) `install.exe -i console`
- (Linux、Solaris) `./install.bin -i console`

注意： `-i console` オプションを指定せずにコマンドを入力すると、GUI ベースのインストーラが起動します。

3. 次の表の説明に従って、プロンプトに応答します。

| プロンプト | 説明 |
|--|---|
| Choose Locale | <p>インストールで使用するロケールを値 1 ~ 2 を入力して選択します。デフォルト値を選択するには、Enter キーを押します。</p> <p>English、または日本語を選択できます。デフォルトの言語は日本語です。</p> |
| Choose Install Folder | <p>Destination 画面で、Enter キーを押してデフォルトディレクトリを使用するか、新しいインストールディレクトリの場所を入力します。</p> <p>デフォルトのインストールフォルダは次のとおりです。</p> <p>(Windows) : C:\Adobe\Adobe LiveCycle ES2</p> <p>(Linux、Solaris) : /opt/adobe/adobe_livecycle_es2</p> <p>注意： ディレクトリ名にアクセント記号付きの文字を使用しないでください。アクセント記号付きの文字を使用すると、CLI によってアクセントが無視され、アクセント記号付きの文字が変更されてからディレクトリが作成されます。</p> |
| LiveCycle ES2 Server License Agreement | <p>Enter キーを押して、使用許諾契約のページに目を通します。</p> <p>契約に同意する場合は、<code>Y</code> を入力し、Enter キーを押します。</p> |
| Pre-Installation Summary | <p>選択したインストール内容を確認し、その内容でインストールを続行する場合は Enter キーを押します。</p> <p>前の手順に戻って設定を変更するには、<code>back</code> と入力します。</p> |
| Ready To Install | <p>インストーラによりインストールディレクトリが表示されます。</p> <p>Enter キーを押すと、インストールプロセスが開始します。インストール中、進行状況バーによりインストールの進行状況が示されます。</p> <p>設定を変更する場合は <code>back</code> と入力します。<code>quit</code> と入力すると、インストールが終了します。</p> |
| Installing | <p>インストールの進行状況が示されます。</p> |

| プロンプト | 説明 |
|---------------------------------|--|
| LiveCycle Configuration Manager | <p>Enter キーを押すと、LiveCycle ES2 のインストールが完了します。</p> <p>LiveCycle Configuration Manager を実行するには、次のスクリプトを呼び出します。</p> <p>(Windows) :</p> <pre>C:¥Adobe¥Adobe Livecycle ES2¥configurationManager¥bin¥ConfigurationManager.bat</pre> <p>(Linux、Solaris) :</p> <pre>/opt/adobe/adobe_livecycle_es2/ configurationManager/bin/ConfigurationManager.sh</pre> |
| Installation Complete | <p>インストールの完了画面にインストールのステータスと場所が表示されます。</p> <p>Enter キーを押すと、インストーラが終了します。</p> <p>Adobe LiveCycle ES2 を適切にアンインストールするには、<code>-i console</code> フラグを使用してコマンドラインからアンインストーラを実行します。</p> |

A.2 エラーログ

エラーが発生した場合は、次のインストールのログディレクトリで `Adobe_LiveCycle_ES2_InstallLog.log` を確認できます。

- (Windows) `C:¥Adobe¥Adobe LiveCycle ES2¥log`
- (Linux、Solaris) `/opt/adobe/adobe_livecycle_es2/log`

インストール中に発生するおそれのあるエラーについて詳しくは、適切なトラブルシューティングガイドを参照してください。

A.3 コンソールモードでの LiveCycle ES2 のアンインストール

コマンドラインオプションを使用して LiveCycle をインストールした場合は、コマンドラインからアンインストーラを実行するだけで Adobe LiveCycle ES2 をアンインストールできます。サイレントアンインストールを実行する場合は、「`-i console`」フラグを省略します。

次の操作を実行します。

1. コマンドプロンプトを開き、アンインストールスクリプトが含まれるディレクトリに移動します。

注意： UNIX システムの場合は、ディレクトリ名にスペースが含まれているので、アンインストールスクリプトが含まれるディレクトリには手動で移動する必要があります。

- (Windows) `cd C:¥Adobe¥Adobe LiveCycle ES2¥Uninstall_Adobe LiveCycle ES2`
- (Linux、Solaris) `cd /opt/adobe/adobe_livecycle_es2
/Uninstall_Adobe LiveCycle ES2`

2. プロンプトで次のコマンドを入力し、Enter キーを押します。

- (Windows) `Uninstall Adobe LiveCycle ES2.exe -i console`
- (Linux、Solaris) `./Uninstall Adobe LiveCycle ES2 -i console`

注意： `-i console` オプションを指定せずにアンインストールコマンドを入力すると、サイレントアンインストールが実行されます。

3. 画面の指示に従って操作します。

| プロンプト | 説明 |
|---------------------------------------|---|
| Uninstall Adobe LiveCycle ES2 | Enter キーを押すと、アンインストールが続行します。 quit と入力すると、アンインストールプログラムが終了します。 アンインストールプログラムを開始した後に、前の手順に戻って変更を加えるには、 back と入力します。 |
| Uninstalling... Uninstall Complete | アンインストールが開始したら、残りのアンインストールプロセスが完了し、カーソルがプロンプトに戻ります。 一部の項目については削除されない可能性があります。また、LiveCycle ES2 のインストール後に作成されたフォルダは削除されません。これらのファイルやフォルダは手動で削除する必要があります。 |

A.4 次の手順

デプロイする LiveCycle ES2 を設定する必要があります ([「LiveCycle ES2 をデプロイするための設定」\(30 ページ\)](#))
または [「付録 - LCM コマンドラインインターフェイス」\(99 ページ\)](#) を参照)。

LiveCycle ES2 には、LiveCycle Configuration Manager 用のコマンドラインインターフェイス (CLI) が用意されています。LiveCycle ES2 の上級ユーザーを対象としている CLI は、例えば LiveCycle Configuration Manager のグラフィカルユーザーインターフェイス (GUI) がサポートされていないサーバー環境で使用します。この章では、CLI を使用して LiveCycle ES2 を構成する方法について説明します。

- [「操作の順序」 \(99 ページ\)](#)
- [「コマンドラインインターフェイスのプロパティファイル」 \(100 ページ\)](#)
- [「使用例」 \(112 ページ\)](#)
- [「エラーログ」 \(112 ページ\)](#)

B.1 操作の順序

LiveCycle Configuration Manager CLI は、GUI バージョンの LiveCycle Configuration Manager と同じ順序で操作する必要があります。CLI は以下の順序で操作してください。

1. LiveCycle ES2 を構成します。
2. アプリケーションサーバポートポロジを検証します。
3. データベース接続を検証します。
4. アプリケーションサーバーを構成します (WebSphere および WebLogic のみ)。
5. アプリケーションサーバーの構成を検証します。
6. LiveCycle ES2 をデプロイします。
7. LiveCycle ES2 を初期化します。
8. Business Activity Monitoring ES2 を初期化します。
9. LiveCycle ES2 サーバーを検証します。
10. LiveCycle ES2 モジュールをデプロイします。
11. 7.x 互換レイヤーを LiveCycle ES2 モジュールと共にデプロイします。
12. LiveCycle ES2 モジュールのデプロイメントを検証します。
13. PDF Generator ES2 のシステム準備設定を確認します。
14. PDF Generator ES2 の管理者ユーザーを追加します。
15. LiveCycle ES2 Connector for IBM Content Manager を構成します。
16. LiveCycle ES2 Connector for IBM FileNet を構成します。
17. LiveCycle ES2 Connector for EMC Documentum を構成します。
18. LiveCycle ES2 Connectors for ECM の構成をすべてテストします。

19. Content Services ES2 を構成します。

警告： LiveCycle Configuration Manager CLI の操作を完了したら、クラスタ内の各 WebLogic Server を再起動する必要があります。

B.2 コマンドラインインターフェイスのプロパティファイル

LiveCycle Configuration Manager CLI には、LiveCycle 環境用に定義したプロパティを含むファイルが必要です。プロパティファイルのテンプレートである cli_propertyFile_template.txt は、[LiveCycleES2 root]/configurationManager/bin フォルダにあります。コピーを作成して、値を編集する必要があります。このファイルは、使用する LiveCycle Configuration Manager の操作に基づいてカスタマイズできます。次の節で、必要なプロパティとその値について説明します。

プロパティファイルは、インストールの状態に応じて作成する必要があります。次のいずれかの方法を使用します。

- プロパティファイルを作成し、インストールシナリオおよび構成シナリオに応じて値を設定します。
- プロパティファイル cli_propertyFile_template.txt をコピーして、使用する LiveCycle Configuration Manager 操作に基づいて値を編集します。
- LiveCycle Configuration Manager の GUI を使用し、GUI バージョンによって作成されたプロパティファイルを CLI バージョンのプロパティファイルとして使用します。[LiveCycleES2 root]/configurationManager/bin/configurationManager.bat ファイルを実行すると、userValuesForCLI.properties ファイルが [LiveCycleES2 root]/configurationManager/config ディレクトリに作成されます。このファイルを LiveCycle Configuration Manager CLI の入力として使用できます。

注意： CLI プロパティファイルでは、Windows パスのディレクトリセパレータ (\) にエスケープ文字 (\\) を使用する必要があります。例えば、指定する Fonts フォルダが C:\Windows\Fonts である場合、LiveCycle Configuration Manager CLI スクリプトでは C:\\Windows\\Fonts と入力する必要があります。

B.2.1 共通のプロパティ

共通のプロパティは次のとおりです。

WebLogic および WebSphere 固有のプロパティ：アプリケーションサーバーの設定、LiveCycle のデプロイ、アプリケーションサーバートポロジおよびアプリケーションサーバー設定の検証操作に必要です。

LiveCycle サーバー固有のプロパティ：LiveCycle の初期化と LiveCycle コンポーネントのデプロイ操作に必要です。

以下の操作に必要なプロパティは次の表のとおりです。

- LiveCycle ES2 を初期化する
- LiveCycle ES2 コンポーネントをデプロイする。

| プロパティ | 値 | 説明 |
|---------------------------|--------------------|---|
| WebLogic 固有のプロパティ | | |
| targetServer.topologyType | server または cluster | LiveCycle ES2 にデプロイするアプリケーションサーバートポロジの種類。 |
| targetServer.name | 文字列 | アプリケーションサーバーノードまたはクラスタに割り当てられた名前。 |

| プロパティ | 値 | 説明 |
|-------------------------------|---|--|
| targetServer.adminHost | 文字列 デフォルトは localhost です。 | WebLogic アプリケーションサーバーがインストールされているサーバーのホスト名。 |
| targetServer.adminPort | 整数 | WebLogic 管理サーバーが要求をリスンするポートのポート番号。 |
| targetServer.adminUserID | 文字列 | WebLogic アプリケーションサーバーへのアクセスに使用する管理ユーザー ID。 |
| targetServer.adminPassword | 文字列 | WebLogic 管理ユーザー ID に関連付けられているパスワード。 |
| localServer.appServerRootDir | デフォルト : (Windows) WebLogic 10g Server 10.3 C:\bea\wlserver_10.3 (Linux、Solaris) WebLogic 10g Server 10.3 /opt/bea/wlserver_10.3 (Windows) WebLogic 11g Server 10.3 C:\Oracle\Middleware\wlserver_10.3 (Linux、Solaris) WebLogic 10g Server 10.3 /opt/Oracle/Middleware/wlserver_10.3 | ローカルに構成するアプリケーションサーバーインスタンスのルートディレクトリ (LiveCycle ES2 をデプロイしたり、LiveCycle ES2 をデプロイするリモートサーバーと通信するために使用したりするディレクトリ)。 |
| targetServer.appServerRootDir | デフォルト : (Windows) WebLogic 10g Server 10.3 C:\bea\wlserver_10.3 (Windows) WebLogic 11g Server 10.3 C:\Oracle\Middleware\wlserver_10.3 (Linux、Solaris) WebLogic 10g Server 10.3 /opt/bea/wlserver_10.3 (Linux、Solaris) WebLogic 11g Server 10.3 /opt/Oracle/Middleware/wlserver_10.3 | リモートサーバー上に設定するアプリケーションサーバーインスタンスのルートディレクトリ (LiveCycle ES2 をデプロイするディレクトリ)。 |
| LiveCycle サーバー固有のプロパティ | | |
| LCHost | 文字列 | LiveCycle ES2 をデプロイするサーバーのホスト名。 |
| LCPort | 整数 | LiveCycle ES2 のデプロイ先の Web ポート番号。 |

| プロパティ | 値 | 説明 |
|----------------------------|--|---|
| excludedSolutionComponents | 文字列。次の値がサポートされています。 ALC-LFS-Forms、 ALC-LFS-BusinessActivityMonitoring、 ALC-LFS-ConnectorEMCDocumentum、 ALC-LFS-ConnectorIBMFileNet、 ALC-LFS-ConnectorIBMContentManager、 ALC-LFS-ContentServices、 ALC-LFS-DigitalSignatures、 ALC-LFS-DataCapture、 ALC-LFS-Output、 ALC-LFS-PDFGenerator、 ALC-LFS-PDFGenerator3D、 ALC-LFS-ProcessManagement、 ALC-LFS-ReaderExtensions、 ALC-LFS-RightsManagement | (オプション) 構成しない LiveCycle ES2 モジュールを指定します。構成対象から除外するモジュールが複数ある場合はカンマで区切ります。 |

B.2.2 LiveCycle の構成プロパティ

以下のプロパティは LiveCycle の構成操作にのみ適用されます。

| プロパティ | 値 | 説明 |
|------------------|-----|--|
| AdobeFontsDir | 文字列 | Adobe サーバーフォントディレクトリの場所。このパスは、デプロイ先のサーバーからアクセスできるようにする必要があります。 |
| customerFontsDir | 文字列 | カスタマーフォントディレクトリの場所。このパスは、デプロイ先のサーバーからアクセスできるようにする必要があります。 |
| systemFontsDir | 文字列 | システムフォントディレクトリの場所。複数のシステムフォントの場所は、セミコロンで区切って入力します。 このパスは、デプロイ先のサーバーからアクセスできるようにする必要があります。 |
| LCTempDir | 文字列 | 一時ディレクトリの場所。このパスは、デプロイ先のサーバーからアクセスできるようにする必要があります。 |

| プロパティ | 値 | 説明 |
|----------------------------|------------------------------------|--|
| LCGlobalDocStorageDir | 文字列 | <p>グローバルドキュメントを保存するルートディレクトリ。</p> <p>長期間有効なドキュメントを保存したり、それらをすべてのクラスタノードで共有したりするために使用する、NFS 共有ディレクトリのパスを指定します。</p> <p>このプロパティは、LiveCycle ES2 コンポーネントをクラスタ環境にデプロイする場合にのみ指定します。このパスは、デプロイ先のサーバーからアクセスできるようにする必要があります。</p> |
| EnableDocumentDBStorage | true または false デフォルト : false | <p>永続ドキュメントについて、データベースへのドキュメントの保存を有効または無効にします。</p> <p>データベースへのドキュメントの保存を有効にしても、GDS のファイルシステムディレクトリは必要です。</p> |
| enableFIPS | true または false デフォルト : false | <p>「FIPS を有効にする」オプションを選択すると、RSA BSAFE Crypto-J 3.5.2 暗号化モジュールと FIPS 140-2 検証証明書 #590 を使用して FIPS 140-2 承認済みアルゴリズムにデータ保護が制限されます。このオプションは、FIPS を実行する必要がある場合にのみ設定してください。</p> |
| Content Services ES2 のみ | | |
| contentServices.rootDir | 文字列 | <p>[Adobe LiveCycle Content Services ES2 のみ] Content Services ES2 で使用するルートディレクトリを指定します。クラスタ環境で LiveCycle を実行している場合、クラスタ内のすべてのノードによって、すべてのノードにわたって同じパスでこのディレクトリが共有されている必要があります。</p> |
| contentServices.indexesDir | 文字列 | <p>[Adobe LiveCycle Content Services ES2 のみ] Content Services ES2 で使用するインデックスディレクトリを指定します。このディレクトリは各クラスタで一意であり、すべてのクラスタノードで同じ名前と場所にする必要があります。例えば、 contentServices.indexes Dir=C:\¥¥Adobe¥¥ LiveCycle9¥¥lccs_indexes になります。</p> |

| プロパティ | 値 | 説明 |
|-------------------------------------|---|--|
| contentServices.topology | 文字列。SERVER または CLUSTER を指定します。 デフォルト：SERVER | [Adobe LiveCycle Content Services ES2 のみ] シングルノードの場合は SERVER、クラスタ構成の場合は CLUSTER。 |
| contentServices.cifs.enable | true または false デフォルト：false | [Adobe LiveCycle Content Services ES2 のみ] CIFS を有効または無効にします。 |
| contentServices.cifs.servername | 文字列 | [Adobe LiveCycle Content Services ES2 のみ] CIFS サーバーのサーバー名。 |
| contentServices.cifs.implementation | 文字列。次のいずれかを指定します。 <ul style="list-style-type: none"> • NetBIOS • PureJava | [Adobe LiveCycle Content Services ES2 のみ] CIFS サーバーへの Content Services ES2 の接続方法を指定します。 |
| contentServices.cifs.dllpath | 文字列。 NetBIOS DLL のコピー元のパスを指定します。 | [Adobe LiveCycle Content Services ES2 のみ] NetBios DLL のコピー元のパス。「contentServices.cifs.implementation=NetBIOS」の場合は必須。このパスは環境内に存在している必要があります。 |
| contentServices.cifs.alternateIP | 数値 | [Adobe LiveCycle Content Services ES2 のみ] CIFS サーバーの代替 IP アドレス。静的 IP である必要があります。「contentServices.cifs.implementation=PureJava」の場合、これは必須フィールドです。 |
| contentServices.cifs.WinsOrBrdcast | 文字列。次のいずれかを指定します。 <ul style="list-style-type: none"> • winsServer • broadcast | [Adobe LiveCycle Content Services ES2 のみ] DNS 検索方法。「winsServer」または「broadcast」になります。「contentServices.cifs.implementation=PureJava」の場合、これは必須フィールドです。 |
| contentServices.cifs.winsPrmlIP | 数値 | [Adobe LiveCycle Content Services ES2 のみ] プライマリ WINS サーバーの IP アドレス。ipconfig/all コマンドから取得できます。「contentServices.cifs.implementation=PureJava」および「contentServices.cifs.WinsOrBrdcast=winsServer」の場合、これは必須フィールドです。 |

| プロパティ | 値 | 説明 |
|--------------------------------|-----|--|
| contentServices.cifs.winsSecIP | 数値 | [Adobe LiveCycle Content Services ES2 のみ] セカンダリ WINS サーバーの IP アドレス。ipconfig/all コマンドから取得できます。 「contentServices.cifs.implementation=PureJava」および 「contentServices.cifs.WinsOrBrdcast=winsServer」の場合、これは必須フィールドです。 |
| contentServices.cifs.brdCastIP | 数値 | [Adobe LiveCycle Content Services ES2 のみ] ブロードキャスト IP アドレス。 「contentServices.cifs.implementation=PureJava」および 「contentServices.cifs.WinsOrBrdcast=broadCast」の場合、これは必須フィールドです。 |
| contentServices.dbType | 文字列 | [Adobe LiveCycle Content Services ES2 のみ] Content Services のデータベースタイプ。 |

B.2.3 アプリケーションサーバーの設定および検証のプロパティ

LiveCycle Configuration Manager では、LiveCycle ES2 の必要に応じて、WebLogic アプリケーションサーバーを構成および検証できます。

次の操作に適用されるプロパティを、以下の表に示します。

- アプリケーションサーバーの構成
- アプリケーションサーバートポロジの検証
- アプリケーションサーバー構成の検証
- データベース接続の検証

| プロパティ | 値 | 説明 |
|---|----------------|---------------------|
| WebLogic 固有のプロパティセクションを構成する必要があります。詳しくは、 「共通のプロパティ」(100 ページ) を参照してください。 | | |
| jvm.initialHeapSize | デフォルト： 512 | JVM の初期ヒープサイズ (MB)。 |
| jvm.maxHeapSize | デフォルト： 1792 | JVM の最大ヒープサイズ (MB)。 |
| WebLogic および WebSphere クラスタのみ | | |

| プロパティ | 値 | 説明 |
|---|--|---|
| cache.useUDP | true または false | LiveCycle ES2 が UDP を使用してキャッシュを実装する場合は、値を true に設定します。 LiveCycle ES2 が TCP を使用してキャッシュを実装する場合は、値を false に設定します。 |
| cache.udp.port | デフォルト： 33456 | プライマリ WebLogic Server コンピュータが UDP ベースのキャッシュ通信に使用するポートのポート番号。 cache.useUDP=true の場合にのみ設定します。 |
| cache.tcpip.primaryhost | 文字列 | プライマリ WebLogic Server がインストールされているコンピュータのホスト名。 cache.useUDP=true の場合にのみ設定します。 |
| cache.tcpip.primaryport | デフォルト： 22345 | プライマリ WebLogic Server コンピュータが TCP ベースのキャッシュ通信に使用するポートのポート番号。 cache.useUDP!=true の場合にのみ構成します。 |
| cache.tcpip.secondaryhost | 文字列 | セカンダリ WebLogic Server がインストールされているコンピュータのホスト名。 cache.useUDP!=true の場合にのみ構成します。 |
| cache.tcpip.secondaryport | デフォルト： 22345 | セカンダリ WebLogic Server コンピュータが TCP ベースのキャッシュ通信に使用するポートのポート番号。 cache.useUDP!=true の場合にのみ構成します。 |
| contentServices.cluster.cacheListenerArgs | カンマ区切り文字列 myhostA:7800,myhostB:7800 | (クラスタのみ) [Adobe LiveCycle Content Services ES2 のみ] クラスタ内の Content Services ES2 キャッシュリスナーのホスト名または IP アドレスおよびポート。 |
| contentServices.cluster.startPort | 整数 | (クラスタのみ) [Adobe LiveCycle Content Services ES2 のみ] このノードの Content Services ES2 キャッシュリスナーが要求をリスンするために使用するポート。 |

| プロパティ | 値 | 説明 |
|------------------------------------|---|--|
| contentServices.cluster.portRange | 整数。デフォルトは 3 です。 | (クラスタのみ) [Adobe LiveCycle Content Services ES2 のみ] Content Services ES2 キャッシュ範囲。 |
| WebLogic サーバーのコアクラスパス構成 | | |
| classpath.targetServer.javaHome | 文字列 | ターゲットアプリケーションサーバーの構成では、ターゲットアプリケーションサーバーの実行で使用する Java Home の場所を指定する必要があります。 このパスは、構成するサーバーからアクセスできる必要があります。 |
| classpath.targetServer.pop3JarPath | 文字列 | ターゲットアプリケーションサーバーがアクセスできる Pop3 JAR ファイルのパス。このパスは、構成するサーバーからアクセスできる必要があります。 |
| データソース構成 | | |
| datasource.dbType | 次のオプションから選択します。 <ul style="list-style-type: none"> ● oracle ● db2 ● sqlserver | LiveCycle ES2 で使用するよう構成するデータベースの種類。 |
| datasource.dbName | 文字列 | データベースの名前。 |
| datasource.dbHost | 文字列 | データベースがあるサーバーのホスト名または IP アドレス。 |
| datasource.dbPort | 整数 | LiveCycle ES2 がデータベースとの通信に使用するデータベースポート。 |
| datasource.dbUser | 文字列 | LiveCycle ES2 がデータベースへのアクセスに使用するユーザー ID。 |
| datasource.dbPassword | 文字列 | データベースユーザー ID に関連付けられているパスワード。 |
| datasource.target.driverPath | 文字列 | アプリケーションサーバーの lib ディレクトリ内の JDBC ドライバ。このパスは、構成するサーバーからアクセスできる、有効なパスである必要があります。 |
| datasource.local.driverPath | 文字列 | ローカル JDBC ドライバ。この値は、直接データベース接続のテストにのみ使用します。 |

B.2.4 LiveCycle のデプロイプロパティ

以下のプロパティは LiveCycle ES2 のデプロイ操作にのみ適用されます。

| プロパティ | 値 | 説明 |
|--|----------------|--|
| LiveCycle サーバー情報セクションを構成する必要があります。詳しくは、 「共通のプロパティ」(100 ページ) を参照してください。 | | |
| deployment.includeIVS | true または false | IVS EAR ファイルをデプロイメントに含めるかどうかを指定します。 警告： IVS EAR ファイルは実稼働環境に含めないようにすることをお勧めします。 |

B.2.5 LiveCycle の初期化プロパティ

以下のプロパティは LiveCycle ES2 の初期化操作にのみ適用されます。

| プロパティ | 値 | 説明 |
|--|---|----|
| LiveCycle サーバー情報セクションを構成する必要があります。詳しくは、 「共通のプロパティ」(100 ページ) を参照してください。 | | |

B.2.6 BAM の初期化プロパティ

以下のプロパティは BAM の初期化操作にのみ適用されます。

注意： Business Activity Monitoring ES2 は、LiveCycle ES2 のオプションコンポーネントです。

| プロパティ | 値 | 説明 |
|------------------|--|--|
| BAMHost | 文字列 | BAM をデプロイして実行するサーバーのホスト名。 |
| BAMPort | 整数 | BAM サーバーが要求をリスンするポート番号。 |
| BAMAdminUserID | 文字列 | BAM サーバーへの接続に使用する BAM 管理者ユーザー ID。 |
| BAMAdminPassword | 文字列 | BAM サーバーへの接続に使用する BAM 管理者パスワード。 |
| databaseType | 次のオプションから選択します。 <ul style="list-style-type: none">oraclemysqldb2sqlserver | LiveCycle が BAM データのキャプチャに使用するデータベースの種類。 |

B.2.7 LiveCycle コンポーネントのデプロイプロパティ

以下の操作に適用されるプロパティは次の表のとおりです。

- LiveCycle コンポーネントのデプロイ
- LiveCycle コンポーネントのデプロイメントの検証
- LiveCycle サーバーの検証

| プロパティ | 値 | 説明 |
|---|-----|--|
| LiveCycle サーバー情報セクションを構成する必要があります。詳しくは、 「共通のプロパティ」 (100 ページ) を参照してください。 | | |
| LCAdminUserID | 文字列 | LiveCycle 管理者ユーザーに割り当てるユーザー ID。このユーザー ID は、LiveCycle 管理コンソールへのログオンに使用されます。 |
| LCAdminPassword | 文字列 | LiveCycle 管理者ユーザーに割り当てるパスワード。このパスワードは、LiveCycle 管理コンソールへのログオンに使用されます。 |

B.2.8 コマンドラインインターフェイスの使用

プロパティファイルを構成したら、次のフォルダに移動する必要があります。
[LiveCycleES2 root]/configurationManager/bin フォルダ

LiveCycle Configuration Manager CLI のコマンドの詳細な説明を表示するには、
ConfigurationManagerCLI help <command name> と入力します。

B.2.8.1 LiveCycle の設定 CLI の使用

LiveCycle の構成操作では、次の構文を使用する必要があります。

```
configureLiveCycle -f <propertyFile>
```

説明：

- -f <propertyFile>：必要な引数が含まれるプロパティファイル。プロパティファイルの作成について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」 \(100 ページ\)](#)を参照してください。

B.2.8.2 アプリケーションサーバーの構成 CLI の使用

WebLogic のアプリケーションサーバーの構成操作では、次の構文を使用する必要があります。

```
configureApplicationServer -targetServer_AdminPassword <password> -f <propertyFile> [-skip <configurationsToSkipList>]
```

説明：

- -targetServer_AdminPassword <password>：コマンドラインで管理者パスワードを設定できます。この引数を指定すると、プロパティファイルの targetServer_AdminPassword プロパティが上書きされます。

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -skip <configurationsToSkipList> : 構成しないアプリケーションサーバーコンポーネントを指定できるオプションのパラメータです。構成対象から除外するコンポーネントが複数ある場合はカンマで区切ります。有効なオプションは、Datasource および Core です。

B.2.8.3 LiveCycle のデプロイ CLI の使用

LiveCycle のデプロイ操作では、次の構文を使用する必要があります。

```
deployLiveCycle -f <propertyFile>
```

説明 :

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。

B.2.8.4 LiveCycle の初期化 CLI の使用

LiveCycle の初期化操作では、次の構文を使用する必要があります。

```
initializeLiveCycle -f <propertyFile>
```

説明 :

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。

B.2.8.5 Business Activity Monitoring の初期化 CLI の使用

Business Activity Monitoring の初期化操作では、次の構文を使用する必要があります。

```
initializeBAM -f <propertyFile>
```

説明 :

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。

B.2.8.6 LiveCycle コンポーネントのデプロイ CLI の使用

LiveCycle コンポーネントのデプロイ操作では、次の構文を使用する必要があります。

```
deployLiveCycleComponents -f <propertyFile> -targetServer_AdminPassword <password>
```

説明 :

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -targetServer_AdminPassword <password> : コマンドラインで管理者パスワードを設定できます。この引数を指定すると、プロパティファイルの targetServer.adminPassword プロパティが上書きされます。

B.2.8.7 アプリケーションサーバートポロジの検証 CLI の使用

アプリケーションサーバートポロジの検証操作 (オプション) では、次の構文を使用する必要があります。

```
validateApplicationServerTopology -f <propertyFile> -targetServer_AdminPassword <password>
```

説明：

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -targetServer_AdminPassword <password> : コマンドラインで管理者パスワードを設定できます。この引数を指定すると、プロパティファイルの targetServer.adminPassword プロパティが上書きされます。

B.2.8.8 データベース接続の検証 CLI の使用

データベース接続の検証操作 (オプション) では、次の構文を使用する必要があります。

```
validateDBConnectivity -f <propertyFile> -datasource_dbPassword <password>
```

説明：

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -datasource_dbPassword <password> : コマンドラインでデータベースユーザーパスワードを設定できます。この引数を指定すると、プロパティファイルの datasource.dbPassword プロパティが上書きされます。

B.2.8.9 アプリケーションサーバー構成の検証 CLI の使用

アプリケーションサーバー構成の検証操作 (オプション) では、次の構文を使用する必要があります。

```
validateApplicationServerConfigurations -f <propertyFile> -targetServer_AdminPassword <password>
```

説明：

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -targetServer_AdminPassword <password> : コマンドラインで管理者パスワードを設定できます。この引数を指定すると、プロパティファイルの targetServer.adminPassword プロパティが上書きされます。

B.2.8.10 LiveCycle サーバーの検証 CLI の使用

LiveCycle サーバーの検証操作 (オプション) では、次の構文を使用する必要があります。

```
validateLiveCycleServer -f <propertyFile> -targetServer_AdminPassword <password>
```

説明：

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -targetServer_AdminPassword <password> : コマンドラインで管理者パスワードを設定できます。この引数を指定すると、プロパティファイルの targetServer.adminPassword プロパティが上書きされます。

B.2.8.11 LiveCycle コンポーネントのデプロイメントの検証 CLI の使用

LiveCycle コンポーネントのデプロイメントの検証操作 (オプション) では、次の構文を使用する必要があります。

```
validateLiveCycleComponentDeployment -f <propertyFile> -targetServer_AdminPassword <password>
```

説明：

- -f <propertyFile> : 必要な引数が含まれるプロパティファイル。プロパティファイルの作成手順について詳しくは、[「コマンドラインインターフェイスのプロパティファイル」\(100 ページ\)](#) を参照してください。
- -targetServer_AdminPassword <password> : コマンドラインで管理者パスワードを設定できます。この引数を指定すると、プロパティファイルの targetServer.adminPassword プロパティが上書きされます。

B.3 使用例

C:\Adobe\Adobe LiveCycle ES2\configurationManager\bin から次のコマンドを入力します。
ConfigurationManagerCLI configureLiveCycle -f cli_propertyFile.txt

cli_propertyFile.txt には、作成済みのプロパティファイルの名前を指定します。

B.4 エラーログ

エラーが発生した場合は、[LiveCycleES2 root]\configurationManager\log フォルダにある CLI エラーログで確認できます。生成されるログファイルには、命名規則に基づいて lcmCLI.0.log のような名前が付けられます。ファイル名の数字 (ここでは 0) は、ログファイルがロールオーバーされるたびに増加します。

B.5 次の手順

LiveCycle Configuration Manager CLI を使用して LiveCycle ES2 の構成とデプロイを行った場合は、この時点で次の作業を実行できます。

- デプロイメントの確認 ([「監視フォルダのパフォーマンスパラメータの設定」\(66 ページ\)](#) を参照)。
- LiveCycle 管理コンソールへのアクセス ([「LiveCycle 管理コンソールへのアクセス」\(56 ページ\)](#) を参照)。
- LDAP にアクセスするための LiveCycle モジュールの構成 ([「LiveCycle ES2 の LDAP アクセス設定」\(67 ページ\)](#) を参照)。
- LiveCycle ES2 のアンインストール ([「LiveCycle ES2 のアンインストール」\(88 ページ\)](#) を参照)。

デプロイメントのためのアプリケーションサーバーの構成を行っていない場合は、この時点でアプリケーションサーバーを構成する必要があります ([「WebLogic Server クラスタの手動設定」\(43 ページ\)](#) を参照)。