

WAS Liberty 最新情報セミナー 2015

～ 次世代ランタイム Libertyで変わる Java EE システム～

3. Libertyプロファイルの基本構成



© 2015 IBM Corporation

このセッションでは、Libertyプロファイルについての基本、すなわち、Libertyプロファイルのインストールバイナリの入手から、導入、基本的な構成方法、開発環境の概要について説明します。

Disclaimer

- この資料は日本アイ・ビー・エム株式会社ならびに日本アイ・ビー・エム システムズ・エンジニアリング株式会社の正式なレビューを受けておりません。
- 当資料は、資料内で説明されている製品の仕様を保証するものではありません。
- 資料の内容には正確を期するよう注意しておりますが、この資料の内容は2015年7月現在の情報であり、製品の新しいリリース、PTFなどによって動作、仕様が変わる可能性があるにご注意下さい。
- 今後国内で提供されるリリース情報は、対応する発表レターなどでご確認ください。
- I B M、I B Mロゴおよびibm.comは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれ I B Mまたは各社の商標である場合があります。現時点での I B Mの商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。
- 当資料をコピー等で複製することは、日本アイ・ビー・エム株式会社ならびに日本アイ・ビー・エム システムズ・エンジニアリング株式会社の承諾なしではできません。
- 当資料に記載された製品名または会社名はそれぞれの各社の商標または登録商標です。
- JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。
- Microsoft, Windows および Windowsロゴは、Microsoft Corporationの米国およびその他の国における商標です。
- Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。
- UNIXはThe Open Groupの米国およびその他の国における登録商標です。

© 2015 IBM Corporation

目次

- Libertyの構造
 - ◆ どんな構造？
- 導入・構成・設定
 - ◆ いざ、導入！
 - ◆ どうやって構成する？
 - ◆ どうやって動かす？
- 開発環境
 - ◆ 開発環境セットアップ

3. Libertyプロファイルの基本構成

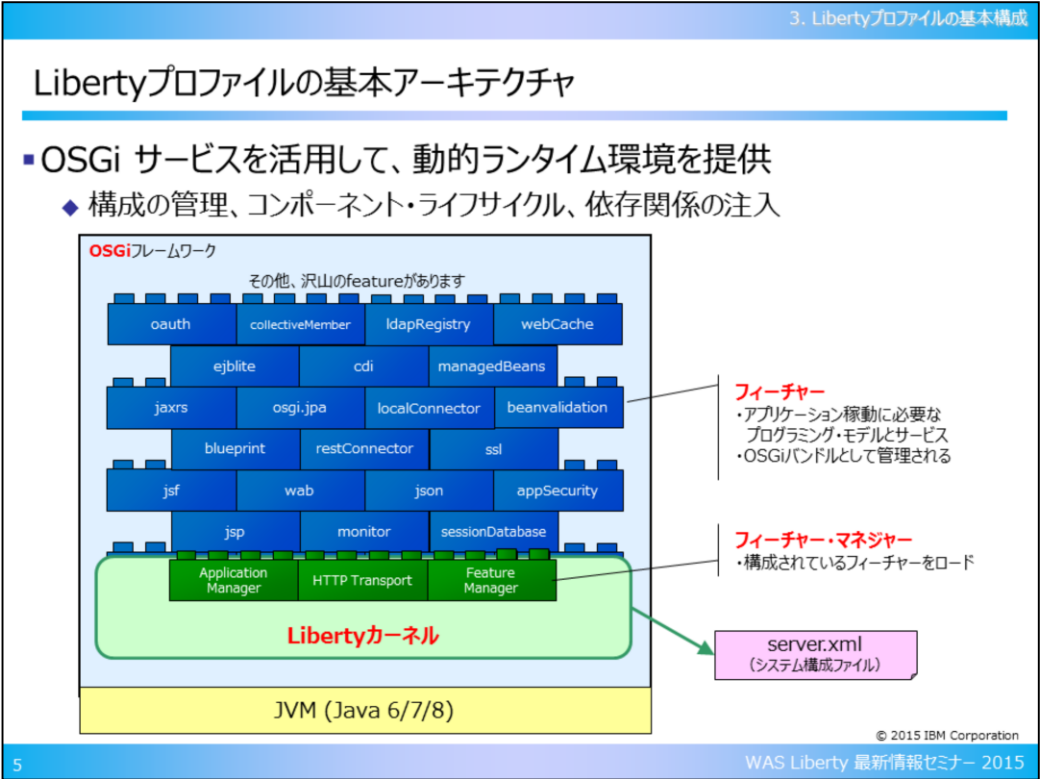
3. Libertyプロファイルの基本構成

どんな構造？

© 2015 IBM Corporation

4WAS Liberty 最新情報セミナー 2015

まずは、Libertyプロファイルはどういう仕組みで動作しているのかを説明します。

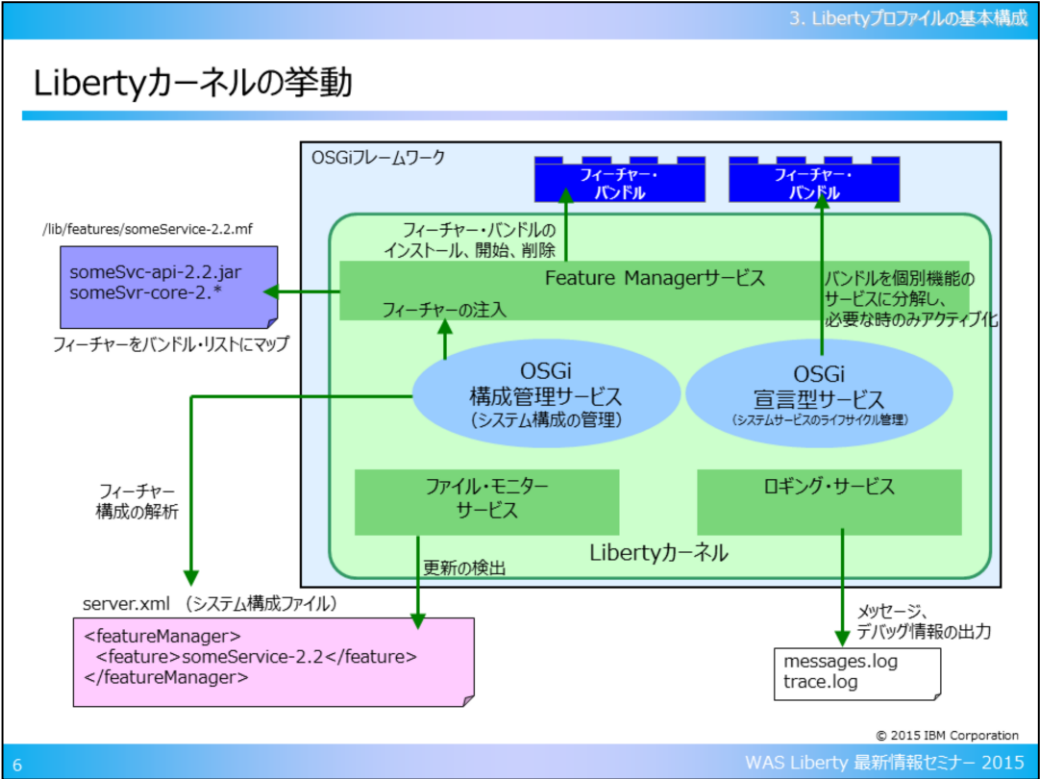


ここでは、まず 個々のアプリケーション機能として提供される「フィーチャー」の利用を可能にしている基本的アーキテクチャーについて確認します。

Libertyプロファイルは、OSGiのフレームワークを基盤として動作します。

Libertyカーネルが起動されると、OSGiのフレームワークが開始されます。OSGiは、古いJavaの世界では欠けていた動的なコンポーネントの追加や実行を可能にしているフレームワークであり、Eclipseなどでも利用されています。

Libertyカーネルの中では「フィーチャー・マネジャー」と呼ばれるコンポーネントが稼動しており、server.xmlの内容を解析して、構成されているフィーチャーをロードします。ロードされる個々のフィーチャーはアプリケーション稼動に必要なプログラミング・モデルやサービスであり、OSGiのコンポーネントの単位であるOSGiバンドルとして管理されています。



Libertyカーネルの挙動について、もう少しだけ詳しく見てみます。

Libertyカーネルで稼動するOSGi 構成管理サービス (OSGi Config Admin Service)が `server.xml`を読み込み、必要とされるフィーチャーの解析を行い、フィーチャー・マネージャーサービスに必要なフィーチャーを伝えます。フィーチャー・マネージャーは伝えられたフィーチャーをOSGiのバンドルにマップし、そのフィーチャーのバンドルを動的にインストールします。

Libertyカーネル上では、ファイル・モニター・サービスも稼動しており、`server.xml`の変更を監視していますので、`server.xml`に変更が行われると、再びフィーチャーの解析が行われ、OSGiの構成管理コンポーネントによりフィーチャーが注入されます。

またOSGiの宣言型サービスを利用することにより、それぞれのサービスの依存性が明確化されています。これにより、バンドルの全てを起動時にアクティブにするのではなく (Late)、依存性のあるサービスのみを必要な時に有効化する (Lazy) ことで、軽量なランタイム環境を実現しています。

またLibertyカーネルにはロギング・サービスが統合されており、メッセージやデバイス情報の出力が行われます。

3. Libertyプロファイルの基本構成

3. Libertyプロファイルの基本構成

Libertyプロファイルの導入

ここでは、テスト環境・本番環境などの実働環境（非開発環境）への Libertyプロファイルの導入について説明します。

7

© 2015 IBM Corporation
WAS Liberty 最新情報セミナー 2015

大まかにではありますが、Libertyプロファイルの仕組みを確認しました。
ここからは、Libertyプロファイルを実働環境に導入する際の手順、考慮点について確認します。



導入時には、まず、何を導入するかを考えます。当然最初はランタイムを導入するわけですが、JRE/JDKをどのように入手するかという点もあわせて考える必要があります。その他、Fixやフィーチャーなども導入の対象になります。

併せてそれらをどこから入手するか検討します。導入対象によって入手先も変わってきますので、選択肢はそれほど多くはありませんが、何をどこから入手するか把握・決定する必要があります。

そして、入手したものをどのようにして導入するのか、を検討します。導入対象が何か、によって導入方法も絞られますのでこれも選択肢は多くありませんが、その後の運用にも影響を与える検討事項です。

ここでは、「何を」、「どこから」、「どうやって」、の観点で、どういう選択肢・方法があるのかの概要をご説明します。

特に紫枠で囲ったものについては、IBM製品、Liberty固有の機能やサイトですので、まずはそれらの説明をします。

また、WASのFixダウンロードページからもダウンロードできます。

8.5.5.6: WebSphere Application Server V8.5.5 Fix Pack 6 : <http://www-01.ibm.com/support/docview.wss?uid=swg24040035>

3. Libertyプロフィールの基本構成

【参考】Liberty Repositoryとは？

WASdev

Liberty Repository

■ Libertyプロフィールに関連する様々なコンテンツ（アセット）の単一のアクセス先、ダウンロード先です。

■ 以下のようなアセットをダウンロードできます。

Addons

導入済みLiberty環境に対して新しい機能を追加するためのものです。

Admin Scripts

標準的な管理タスクを実行するためのサンプルスクリプトです。

Config Snippets

特定のタスクに関するサンプルのサーバー設定ファイルです。

Features

Libertyのランタイムに、個別の単機能を追加するためのものです。

Open Source Integration

標準的に使われるオープンソースとの統合に使われるものです。

Products

Libertyプロフィールのランタイムのアーカイブファイルです。

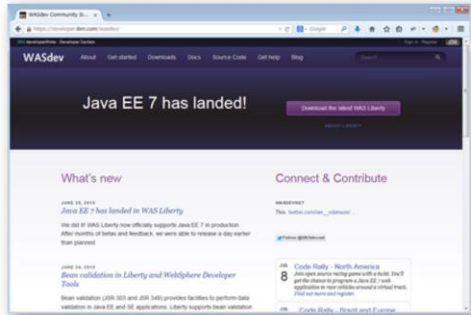
Product Samples

Libertyランタイムの機能をデモするためのサンプルアプリケーションです。

Tools

Libertyのアプリケーションやランタイム拡張を開発・テストするためのツールです。

https://developer.ibm.com/wasdev/



© 2015 IBM Corporation

WAS Liberty 最新情報セミナー 2015

Liberty Repositoryは、Libertyプロフィール関連のアセットのリポジトリです。ここから、各種アセットをダウンロードします。

ランタイム含め、さまざまな種類のアセットがあります。

ブラウザから直接サイトにアクセスし、任意のアセットをダウンロードすることも可能ですし、後述する管理コマンドから接続してダウンロードすることも可能です。

WAS Liberty 最新情報セミナー 2015

9

3. Libertyプロフィールの基本構成

3. Libertyプロフィールの基本構成

Liberty Repositoryからのダウンロード

WASdev
Liberty Repository

Get started with Liberty profile

Download in Eclipse

Install Eclipse plugins to develop, deploy, and debug applications using the Liberty profile. Download and manage Liberty profile installations from Eclipse.

GET IT NOW!

Download just the runtime

Download the Liberty profile runtime to install in build environments, to deploy from the command line, and to develop applications in other IDEs.

GET IT NOW!

Want to try the Beta?

The latest beta updates the Java EE 7 capabilities of Liberty, plus some bug fixes. Let us know if you find any problems so we can fix them in the next beta.

GET IT NOW!

Eclipse上のWDT

GA版の
自己展開圧縮 ZIPファイル

Beta版の
自己展開圧縮 ZIPファイル

Download the WAS Liberty runtime

Get the Liberty profile in four simple steps:

1. Download the runtime from within the tools, or choose from one of these downloads here:

2. Download the runtime from within the tools, or choose from one of these downloads here:

3. Download the runtime from within the tools, or choose from one of these downloads here:

4. Download the runtime from within the tools, or choose from one of these downloads here:

WAS Liberty Beta with Java EE 7 Web Profile with Java 8 (Windows 64-bit)

220Mb

Download Zip

JDK付 Zipアーカイブ、あります。

10

WAS Liberty 最新情報セミナー 2015

Liberty Repositoryでは、GA（Generally Available：正式版）版と、beta版のダウンロード先がそれぞれ用意されていますので注意が必要です。

また、8.5.5.6から、Java8をバンドルしたzipアーカイブ形式のランタイムがダウンロードできるようになっています。

これまで、JDK/JREはLibertyプロフィールとは別に用意しておかなければなりませんでしたので、手間が減り便利になった印象です。

WAS Liberty 最新情報セミナー 2015

10

【参考】LARS (Liberty Asset Repository Service) とLocal Repository

- LARS (Liberty Asset Repository Service) 
 - ◆ Liberty Repository相当のものをオンプレミス環境に構築可能
 - ◆ 社内環境など、Internet越しにLiberty Repositoryにアクセスできない環境において有用
 - ◆ 別途導入・構築する必要があります
- Local directory-based Repository (Local Repository)
 - ◆ ローカルマシンのファイルシステムに直接アセットを配置
 - ◆ ファイルシステムへのアクセスになるので、そのマシン上でのみ利用可能
 - ◆ installUtility downloadコマンドで作成されます

© 2015 IBM Corporation

11

WAS Liberty 最新情報セミナー 2015

次に、Liberty Repositoryに変わるアセットのダウンロード先について説明します。

LARS (Liberty Asset Repository Service) は、Liberty Repository相当のものをオンプレミスの環境に構築し、社内環境などから接続して利用します。Internet上にあるLiberty Repositoryに直接接続できないような環境で重宝します。LARS自体がLibertyプロファイルのサーバーとして稼動し、LARSのクライアントを使用してアセットの登録を行います。登録されたアセットが、LARSから利用可能になります。

もうひとつのLocal Repositoryは、後述のinstallUtilityを使用してローカルマシンのファイルシステムにダウンロードしておいたアセットを保管しておき、後にインストールして使用可能とするローカル限定のRepositoryです。

【参考】Fix Central とは？

Fix Central

- IBM システムのソフトウェア、ハードウェア、およびOS用の Fix および更新をダウンロードできます。
- 製品、バージョン、プラットフォーム、フィックスのタイプなどをウィザード的に選択することで該当Fix が絞られ、効率よい検索が可能です。



※ Fix Central : <http://www.ibm.com/support/fixcentral/>

その他、WAS専用fixダウンロードページ もあります
8.5.5.6 : WebSphere Application Server V8.5.5 Fix Pack 6 : <http://www-01.ibm.com/support/docview.wss?uid=swg24040035>

© 2015 IBM Corporation

Fixは、IBM製品のFixを統合的に提供しているFix Centralというサイトからダウンロードします。
製品、バージョン、プラットフォーム、Fixの種類などを選択することで、該当のFixが一覧表示され、そのままダウンロードすることができます。

【参考】installUtilityとは？

New
FP6

- installUtilityとは、Libertyプロファイル環境にアセットを導入したり、アセットの情報を確認したり出来るコマンドです。
- リポジトリの情報を、`${wlp.install.dir}/etc/repositories.properties` ファイルに定義します。
- 以下の様なことが出来ます。
 - ◆ download
 - アセットをLiberty Repositoryからダウンロードして、ローカルのファイルシステムに保存します。後で参照することが出来ます。
 - ◆ find
 - Liberty Repository / Local Asset Repository Service (LARS) / Local Repositoryにあるアセットを検索します。
 - ◆ install
 - リポジトリにあるアセットをインストールします。
 - ◆ testConnection
 - リポジトリに接続できるか確認します。
 - ◆ uninstall
 - 導入済みアセットをアンインストールします。
 - ◆ viewSettings
 - リポジトリの設定を確認します。

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/t_install_assets_installUtility.html

© 2015 IBM Corporation

installUtilityは、Libertyプロファイル環境にアセットを導入したり、アセットの情報を確認したり、前述のLocal Repositoryにダウンロードしたりするコマンドです。

`${wlp.install.dir}/etc/repositories.properties` に、Liberty Repository使用有無、LARSの接続先、Local Repositoryの保存先を記述しておくことで、透過的に各リポジトリにアクセスできます。

【参考】featureManagerとは？

New
FP2

- featureManagerは、Liberty Repository上のFeatureをLibertyプロファイル環境にインストールするためのコマンドです。
- 以下のようなことができます。
 - ◆ install
 - featureをインストールします。
 - ◆ uninstall
 - 導入済みのfeatureをアンインストールします。
 - ◆ featureList
 - インストールされているfeatureのリストをXML形式で出力します。
 - ◆ find
 - Liberty環境に適用可能なfeatureをLibertyRepositoryから検索します。

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/t_install_assets_cmd.html © 2015 IBM Corporation

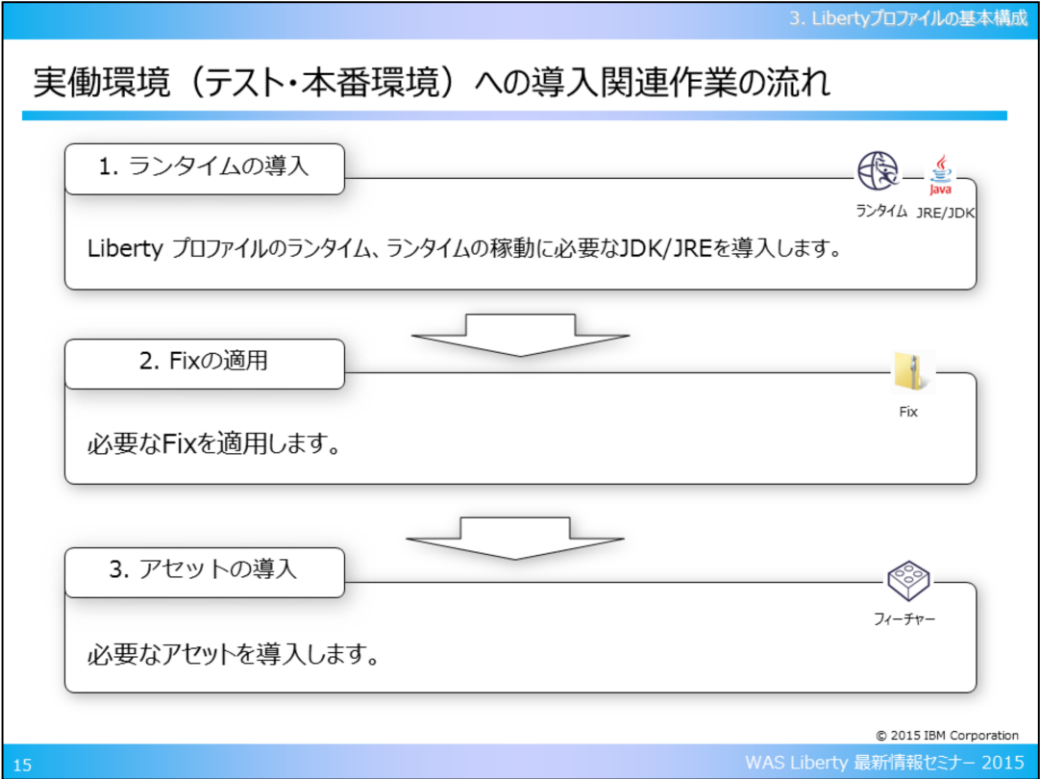
14

WAS Liberty 最新情報セミナー 2015

featureManagerは、Liberty Repository上のFeatureをLibertyプロファイル環境にインストールするためのコマンドです。

installUtilityが複数のリポジトリにアクセス可能であったのに対して、featureManagerはLiberty Repositoryのみにアクセス可能です。また、扱えるものもfeatureのみです。

installUtilityには、インストール済みのフィーチャーのリストを出力する機能がありませんので補完的に使用する事になります。



前頁まで、導入関連作業に必要な基本事項の紹介をしてきました。
ここからは実際に導入を行う際の手順、考慮点を紹介していきます。
大きな流れとしては、ランタイムの導入、Fixの適用、アセットの導入、という3つのステップになります。
導入作業としては、アセットの導入がLiberty固有ですが、流れとしてはこれまでのフル・プロファイルの導入と大きく異なることはありません。

3. Libertyプロファイルの基本構成

導入関連作業の流れー 1. ランタイムの導入

1. ランタイムの導入

ランタイム JRE/JDK

Liberty プロファイルのランタイム、ランタイムの稼動に必要なJDK/JREを導入します。

- 以下の選択肢があります。
 - ◆ **IBM Installation Manager (IIM)**での導入
 - 導入ソフトウェアの統合的な管理を必要とする場合
 - Fixのロールバックなど、厳密な管理をしたい場合
 - ◆ **圧縮Zip/jar展開**での導入
 - シンプルな導入方法
 - より柔軟に管理したい場合
- ここで選択した導入方法で、メンテナンス（Fix適用など）を行っていく事になります。
 - ◆ IIMの場合はIIMでFix適用を、圧縮Zip/jar展開で導入した場合は圧縮Zip/jar展開でFix適用する事になります。

© 2015 IBM Corporation

16

WAS Liberty 最新情報セミナー 2015

まずはランタイムの導入です。

導入方法には、大きく2つの方法があります。

ひとつはIBM Installation Manager（IIM）を使用する方法です。Installation Managerは、IBMソフトウェアの汎用的なインストーラです。Fixの適用、ロールバックなど、IIMが提供する機能でFix適用、履歴管理を行うことができます。



もうひとつは、圧縮Zip/Jarの展開による方法です。圧縮Zip/Jar展開用のインストールモジュールを入手し、展開するだけです。簡単に導入することができます。（ただし、zip解凍用コマンド、jar解凍用のJDKが必要です）より柔軟な管理を行いたい場合などはこちらの方法で対応すると良いでしょう。

いずれの場合でもIIMで導入した場合は、Fix適用もIIMを、Zip/Jar展開導入の場合はFix適用もZip/Jar展開で行うことになり、導入方法によってその後のメンテナンスにも影響ができますので、これらも加味してどのように導入するか検討を行う必要があります。

3. Libertyプロフィールの基本構成

もうひとつ考えないといけない事 - Liberty用のJDK/JRE

1. ランタイムの導入


ランタイム JRE/JDK

Liberty プロファイルのランタイム、ランタイムの稼動に必要なJDK/JREを導入します。

- Libertyランタイムの稼動の前提となるJDK/JREの導入も必要です。選択肢として以下があります。
 - IIMでの導入
 - メディア、PAサイト、Fix Central から、IIMでLiberty稼動用の導入可能なIBM JDKを入手
 - 圧縮Zip/jar展開での導入
 - Liberty Repositoryから、IBM Java 8 付属のLibertyプロフィールをダウンロード可能
 - 自前で用意・導入する
 - Oracle JDK利用などの場合は、自前で用意

17

© 2015 IBM Corporation
WAS Liberty 最新情報セミナー 2015

Libertyランタイム導入時にもう一点考慮しないといけないのが、JDK/JREの導入について、です。

IIMで導入する場合には、IIM導入用のリポジトリパッケージとして提供されているLiberty稼動用のIBM JDKを入手することができます。

圧縮Zip形式での導入の場合は、Java SDK 8付属のZip形式ファイルがダウンロードできますので同時に導入可能です。

以下、対象ファイルダウンロードのポータルです。

8.5.5.6: WebSphere Application Server V8.5.5 Fix Pack 6

<http://www-01.ibm.com/support/docview.wss?uid=swg24040035>

また、Mac Osでの稼動など、IBM Java以外のJDKを必要とする場合は、各自でJDKを用意し導入する必要があります。

但し、製品のテスト等はIBM JDKで行われており、障害発生時などの解析はIBM JDKのほうがスムーズに進捗することもあり、IBM JDKの利用を推奨しています。

WAS Liberty 最新情報セミナー 2015

17

3. Libertyプロファイルの基本構成		
【参考】インストール方法の差異		
	Installation Manager	Archive
エージェントなしでの導入	No	Yes
任意のFixPackレベルに直接上げられるか？	Yes	Yes
FixPackの上書き導入	Yes	No
Interim fixの上書き導入	Yes	Yes
FixPack/iFixのロールバック	Yes	No
製品のエディションのアップグレード	Yes	Yes
組み込み SDK 提供	Yes	Yes
developer toolsとの統合	No	Yes
フィーチャーの追加	Yes	Yes
Liberty Repository の統合	Yes	Yes
z/OS パッケージの提供	Yes	No
パッケージ Minification のサポート	Yes (生成されたイメージは新しいアーカイブとして扱われ、IIMの管理対象にはなりません)	Yes
<div><div>http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/twlp_inst_top.html</div><div>© 2015 IBM Corporation</div></div>		
18	WAS Liberty 最新情報セミナー 2015	

上記は、Installation Managerでの導入とZip/jar解凍での導入の差異をまとめたものです。

導入関連作業の流れー 2. Fixの適用

2. Fixの適用



Fix

必要なFixを適用します。

- FixCentralからFixダウンロードします。
- 導入した方法に合わせてFixを適用します。
 - ◆ IIM導入の場合
 - FixPack/Interim fix共にIIMを使用してFixを適用
 - ◆ 圧縮Zip/jar展開での導入の場合
 - FixPackの場合は、導入時と同様に別ディレクトリに展開し、必要な設定（※1）を移行
 - Intrim fixの場合は、jar展開で導入します。（その他注意点はfixに含まれるreadmeに従う）

※1 WLP_USER_DIRとWLP_OUTPUT_DIR 変数相当のディレクトリ下 が対象（変数については後述）詳細は以下参照
http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/twlp_inst_fixpack_zip.html

© 2015 IBM Corporation

ランタイムの導入後は、最新のFixを適用します。

FixPackの適用は、導入時に採った方法と同じ方法で行います。


IIMでの適用の場合、導入環境にそのままFixが適用されますが、Zip/jarの場合は、新規ランタイムとしてFixPackが提供されますので、これを解凍して新規に導入することになります。

その後必要なユーザーデータ、サーバー設定を移動・コピーすることになります。

3. Libertyプロファイルの基本構成

導入関連作業の流れー 3. フィーチャーの導入

3. フィーチャーの導入


フィーチャー

必要なフィーチャーを導入します。

- 導入方法はいろいろあります。
 - ◆ featureManager
 - ◆ installUtility
- どこからもってくる？
 - ◆ Liberty Repository
 - ・ インターネットアクセス可の場合
 - ◆ LARS
 - ・ インターネットアクセス不可の場合
 - ・ 別途セットアップが必要だが、複数サーバーで共有可能
 - ◆ Local Repository
 - ・ サーバーのファイルシステム上に置いたもの
 - ・ そのサーバーでのみ利用可能

© 2015 IBM Corporation

20

WAS Liberty 最新情報セミナー 2015

稼働環境の準備が整ったら、必要なフィーチャーを導入します。

フィーチャーの導入方法は複数あります。

導入の為のコマンドとしては、featureManagerコマンド、installUtilityコマンドがあります。これらは先に説明したとおりですが、特にinstallUtilityコマンドで導入する場合は、要件・環境に応じてあらかじめ必要なリポジトリを構成して利用可能な状態にしておきます。

3. Libertyプロファイルの基本構成

【参考】どのアセットをどんな方法で導入すればよい？

- 導入方法はいろいろあります。環境、必要なアセットに合わせて方法を選択できます。

アセットの種類	featureManager コマンド	installUtility コマンド	configUtility コマンド	Installation Manager	WDT (WebSphere Developer tools)	WASdev site (Liberty Repository)
Addons	✓	✓		✓	✓	✓
Admin scripts					✓	
Config snippets			✓		✓	✓
Features	✓	✓		✓	✓	
Open source integrations		✓		✓	✓	✓
Products				✓	✓	✓
Product samples		✓		✓	✓	✓
Tools						✓

21

© 2015 IBM Corporation
WAS Liberty 最新情報セミナー 2015

上記は、導入可能なアセットの一覧と、それらの導入にどんな導入方法が選択可能か？の組み合わせを示した表です。

WDTであればTools以外のアセットを網羅的に利用できます。

WDTは開発を行う他、サーバーの構成なども統合的に行えますので、あらかじめWDTにて必要なサーバー構成を作成した後に、パッケージコマンドを利用してランタイム込みのパッケージを作成し、このパッケージで導入まで行う、という方法がLibertyを利用した新しいインフラ構築の方法として考えられると思います。従来のインフラ構築とは大分趣きが異なります。

導入関連のポイントまとめ！

- 何を使って導入するか？
 - ◆ IIM or アーカイブ解凍導入
 - ◆ 運用も考慮して導入方法を選択
- Fixの適用はどうするか？
 - ◆ 導入方法に依存します
- アセットのインストールはどうするか？
 - ◆ featureManager、installUtility、etc が使えます
 - ◆ リポジトリをどう使うか？
 - Liberty repository ? LARS ? Local Repository ?

ここまでのまとめです。

3. Libertyプロフィールの基本構成

3. Libertyプロフィールの基本構成

どう構成する？

© 2015 IBM Corporation

23 WAS Liberty 最新情報セミナー 2015

Libertyプロフィールの導入に続いて、ここからは構成方法について説明していきます。

3. Libertyプロファイルの基本構成

導入すると……Libertyプロファイルのディレクトリ構成

以下のようなディレクトリ構成で導入されます。

wlp

bin

clients

dev

etc

java

lafiles

lib

templates

usr

servers

.logs

defaultServer

apps

configDropins

defaults

overrides

dropins

workarea

shared

apps

config

resources

▪ wlp

- bin

- clients

- dev

- etc

- lib

- templates

- usr

• servers/<サーバー名>

apps

dropins

logs

workarea

• shared

apps

config

resource

：製品ルート（注：導入方法によってはwlpではない）

：Libertyプロファイル 管理スクリプト

：Libertyプロファイルクライアント、シンクライアント

：ユーザーコードから参照可能なライブラリ

：（オプション）全サーバーに適用されるserver.env, jvm.optionsファイルを配置

：プラットフォームのランタイム

：構成のテンプレート、サンプル

：サーバーの構成情報

• servers/<サーバー名>：サーバー定義ディレクトリ

apps：アプリケーション・構成導入用ディレクトリ

dropins：アプリケーション・ドロップイン導入用ディレクトリ

logs：サーバー・ログ・ディレクトリ（ffdc含む）

workarea：サーバー・ワークエリア

• shared：共有ファイル・ディレクトリ

apps：共有アプリケーション

config：共有構成ファイル

resource：共有リソース

24

© 2015 IBM Corporation
WAS Liberty 最新情報セミナー 2015

Libertyプロファイルの導入ディレクトリは、上記のようなディレクトリ構成になっています。
大きくは、usr以下に、個別のサーバー設定やアプリケーションが配置されます。
usr下のserver以下には、作成したサーバーの構成情報が格納されます。
usr下のshared以下には、サーバー内/サーバー間で共有されるアプリケーションやライブラリを配置します。

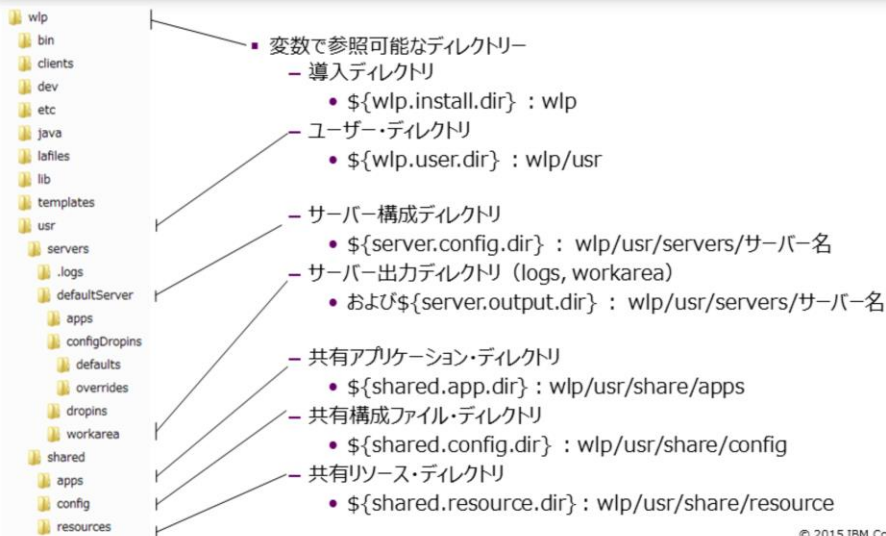
Liberty プロファイル: ディレクトリーのロケーションおよびプロパティ
http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_dirs.html

WAS Liberty 最新情報セミナー 2015

24

【参考】Libertyプロファイルで使用可能な事前定義 環境変数

以下のディレクトリを変数に置き換えて指定できます。



© 2015 IBM Corporation

また、各ディレクトリは、変数として置き換えて指定することも可能です。

3. Libertyプロファイルの基本構成

Libertyサーバーの構成

(基本的には) 1つの構成ファイル (server.xml) に構成を書きます。

▪ server.xml [必須]

◆ Libertyのシステム構成ファイル、変更は動的に反映

◆ サーバー構成を作成した際に自動生成

◆ `${server.config.dir}`に配置

▪ サーバー環境定義ファイル[オプション]

◆ bootstrap.properties : サーバーのブートストラップ・プロパティ

- 初期化用プロパティ、構成ファイル内で使用する変数
- サーバー初期化時にのみ処理される

◆ server.env: サーバー・プロセスで使用する環境変数

- JAVE_HOMEの設定など

◆ jvm.options: VM起動時のオプション

- ヒープサイズの設定、システム・プロパティ、冗長GCなど

✓ `${wlp.install.dir}/etc` および `${server.config.dir}` に配置可能

- 両方存在している場合は、後者の内容が優先される

wlp

...

usr

.settings

servers

.logs

wlp/usr/servers/<サーバー名>

defaultServer

apps

dropins

logs

workarea

bootstrap.properties

jvm.options

server.env

server.xml

server1

server2

shared

Developer Toolsでのサーバー管理画面

サーバー

ローカル・ホストの WebSphere V8.5 Liberty Profile [停止] [defaultServer]

サーバー構成 [server.xml] new server

bootstrap.properties

jvm.options

server.env

26

WAS Liberty 最新情報セミナー 2015

それでは、Libertyプロファイルを構成する構成ファイルについてみていきましょう。
まず唯一必須となる構成ファイルが `server.xml`です。シンプルな構成であればこの`server.xml`のみで構成が完了します。サーバー構成ディレクトリ直下に作成されます。

さらにその他として、追加で作成・利用可能なファイルが3つあります。1つ目が、サーバーの初期化時に読み込まれる`bootstrap.properties`があります。構成ファイル内で使用する変数であったり、トレース・ファイル定義だったりを指定することが可能です。サーバー・プロセスが読み込む環境変数を定義する`server.env`ファイルや、JVM起動時オプションを設定することができる `jvm.options`ファイルがあります。ヒープサイズの設定や`verbose:gc`の設定はここで実施します。これらのファイルはサーバー起動時に読み込まれるため動的変更は行われません。V8.5まではこれらのオプション・ファイルは手で作成する必要がありましたが、V8.5.5からはWebSphere Developer Toolで作成することが可能です。

さらに詳細な内容については以下のKnowledge Centerを参照ください。

Specifying Liberty profile bootstrap properties

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/twlp_inst_bootstrap.html

Customizing the Liberty profile environment

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/twlp_admin_customvars.html

WAS Liberty 最新情報セミナー 2015

26

server.xmlの定義

サーバー構成をXML形式で記述します。

- <server>エレメントの子要素として設定を追加
 - ◆ <featureManager>
 - 利用するフィーチャーを定義
 - ◆ <httpEndpoint>
 - サーバーのホスト名、トランスポートを指定
- システム・デフォルト設定が定義済み
 - ◆ デフォルトを変更する場合のみ記述を追加
 - ◆ 記述がない場合は、デフォルト値が使用される

デフォルトで生成されるserver.xml

```
<server description="new server">
  <featureManager>
    <feature>jsp-2.2</feature>
  </featureManager>
  <httpEndpoint id="defaultHttpEndpoint"
    host="localhost"
    httpPort="9080"
    httpsPort="9443"
  />
</server>
```

© 2015 IBM Corporation

27

WAS Liberty 最新情報セミナー 2015

server.xmlについて、より詳しく見ていきます。

右にあるXMLファイルが、デフォルトで作成される server.xmlファイルです。この構成ファイルでJSPとServlet稼動するWebコンテナについて定義が完了しています。従来のWASの構成ファイルをご存知の方はそのシンプルさに驚くことと思います。

簡単にXMLの要素を見ていきます。<server>エレメントの下に<featureManager>エレメントがありこの中でJSPが有効化されています。さらに<httpEndpoint>エレメントがありWebコンテナがListenするホスト名とポートが指定されています。

それ以外は全てシステムのデフォルト値が暗黙的に定義されています。デフォルトを変更する場合のみ明示的にserver.xmlに記載します。記載がない場合は、デフォルト値が有効になるため、構成ファイルをシンプルに保つことができます。

server.xmlの定義： 使用フィーチャーの定義

<featureManager> および <feature> で使用するfeatureを指定します。

- <featureManager> 下の<feature>で
利用するフィーチャーを定義し、有効化
- フィーチャーには依存関係がある
- 前提となるフィーチャーは暗黙的に有効となる
 - ◆ 例) jsp-2.2はservlet-3.0が前提となっている
- 有効となっていないフィーチャーは、
アプリケーションでは使用できない

server.xml

```
...
<!-- Enable features -->
<featureManager>
  <feature>jsp-2.2</feature>
  <feature>jdbc-4.0</feature>
</featureManager>
...
```



© 2015 IBM Corporation

<featureManager>の下には、<feature>エレメントを追加して構成していきます。

各<feature>には依存関係が定義されていて、前提となるフィーチャーは暗黙的に有効にされます。たとえば、jsp-2.2に関してはservlet-3.0が前提となっているため、jsp-2.2を有効化すると、暗黙的にservlet-3.0が有効化されます。

server.xmlでは、明示的に有効化したフィーチャーしか記述されませんが、WebSphere Developer Toolsのフィーチャー構成画面（後述）では、暗黙的に有効とされているフィーチャーに関しても表示することが可能です。

フィーチャーの一覧と利用可能なエディションの対応については、以下を参照ください。

Liberty features

[http://www-](http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_feat.html)

[01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_feat.html](http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_feat.html)

server.xmlの定義: HTTPエンドポイントの定義

<httpEndpoint>でHTTPサーバーとしての定義を行います。

- <httpEndpoint>
 - ◆ サーバーがLISTENするポート、提供するHTTPサーバー・サービスの定義
 - ◆ 詳細の設定は、子エレメントで指定可能
 - <accessLogging>
 - アクセス・ログのパス、出力フォーマット、ファイル・サイズ、ファイル数
 - <httpOptions>
 - 読み込み/書き込みタイムアウト、KeepAliveの設定
 - <sslOptions>
 - 使用するSSL定義（SSLレパートリー）の設定
 - <tcpOptions>
 - 非活動タイムアウト

server.xml

```
...  
<httpEndpoint  
  host="localhost"  
  httpPort="9080"  
  httpsPort="9443"  
  id="defaultHttpEndpoint" />  
...
```

Developer Toolsの
HTTPエンドポイント構成画面

HTTP エンドポイント

4 child elements are available:
HTTP オプション, TCPオプション, SSL オプション, HTTP アクセス・ロギング. [Add...]

ID	defaultHttpEndpoint
Host	localhost
Port	9080
Security Port	9443
SSL Port	9443
HTTP オプション参照	[Add...]
SSL オプション参照	[Add...]
TCP オプション参照	[Add...]
HTTP アクセス・ロギング参照	[Add...]

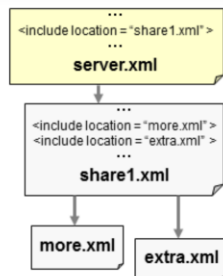
© 2015 IBM Corporation

次に<httpEndpoint>です。サーバーのWebコンテナがListenするポートを指定します。デフォルトではホスト名とポートですが、その他の詳細な設定は、子エレメントで指定していきます。たとえば、Webコンテナのアクセスログのフォーマットやサイズなどは、<accessLogging>エレメントで指定していきます。他にもHTTPやTCPのタイムアウトなども子エレメントの中で指定していきます。

server.xmlの定義：その他Tips

ファイルのインクルード、変数記述など、柔軟な構成が可能です。

- 柔軟な構成：構成ファイルのインクルードが可能
 - 複数サーバーで定義を共有する場合、チーム開発などで有効
 - サーバーインスタンスに依存しない定義（DB定義など）
 - `<include location="dir/file">`で組込むべきファイルを指定
 - dirを相対パスで指定した場合以下の順で検索される
 - `<include>`が書かれた構成ファイルのディレクトリ
 - `•${server.config.dir}`
 - `•${shared.config.dir}`
- 変数を使用した記述（`${〜}`）が可能
 - 製品事前定義 変数（`${server.config.dir}`等）
 - `bootstrap.properties`で指定した値
- 時間を指定する属性値には時間単位をつける
 - 時間単位としては時間（h）、分（m）、または秒（s）が使用可能
 - 一部の設定ではミリ秒（ms）も使用可能
 - たとえば30秒は「30s」と表す
 - 複数の時間単位を使用して「1m30s」というような表記も可能



© 2015 IBM Corporation

30

WAS Liberty 最新情報セミナー 2015

さらにserver.xmlは、構成ファイルの一部を別XMLファイルに外出しにすることが可能です。

たとえば、データ・ソースへのアクセスするための定義など、サーバー・インスタンスごとに変わらない共通する定義は、別のXMLファイルとして外出しし、`<include>`エレメントでserver.xmlに取り込むことが可能です。このように共通する構成ファイルをコンポーネント化することで管理の容易化を実現できます。またチーム開発などの場合も、サーバー・インスタンスに依存しない共通構成ファイルを配ることで、簡単に構成の共通化を実現することが可能です。

その他、`bootstrap.properties`で定義した変数を構成ファイルに利用することができます。従来のWebSphereではタイムアウトを指定する場合は単位を確認する必要がありましたが、Libertyではタイムアウト関連パラメータは、hやm、sなど単位を指定します。たとえば3分は3mと定義することができますし180sと定義することもできます。

Using include elements in configuration files

[https://www-](https://www-01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_includes.html?lang=en)

[01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_includes.html?lang=en](https://www-01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_includes.html?lang=en)

Using variables in configuration files

[https://www-](https://www-01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_vars.html?lang=en)

[01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_vars.html?lang=en](https://www-01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_vars.html?lang=en)

Using Ref tags in configuration files

[https://www-](https://www-01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_reftags.html?lang=en)

[01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_reftags.html?lang=en](https://www-01.ibm.com/support/knowledgecenter/was_beta_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_setup_reftags.html?lang=en)

3. Libertyプロファイルの基本構成

【こんなことも出来ます】 設定のドロップインフォルダ

New FP5

サーバー設定の部分的・動的な上書きが可能です。

- 特定のディレクトリに、変更したい設定を記述した設定ファイルを配置することで、server.xmlに手を加えずにサーバー設定を動的に部分的に上書きすることができます。
- 以下の2つの方法を提供しています。
 - server.xml に定義されていない項目のデフォルト値を変更する
 - server.xml の定義を上書きする

※ overrides > server.xml > defaults > 暗黙的デフォルト値の順に適用されます。

defaultServer

apps

configDropins

defaults

overrides

dropins

workarea

手順

- 以下のディレクトリを作成します。
 - usr/servers/server_name/configDropins/overrides → 設定上書き用
 - usr/servers/server_name/configDropins/defaults → デフォルト値設定用
- 設定ファイルを目的に応じて上記ディレクトリに配置します。
- 動的に配置が検出され設定が適用されます。

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/wlp_setup_dropins.html

© 2015 IBM Corporation

31

WAS Liberty 最新情報セミナー 2015

また、server.xmlの変更はモニターされて自動で再ロードされますが、なんとserver.xmlの内容はそのままに、変更したい部分のみを切り出して構成ファイルを作成し、特定のディレクトリに配置することで、一部分のみを動的に上書きする、という事も可能です。

この特定のディレクトリ、というのは、defaultsと、overridesの2種類用意されており、どちらにファイルを配置するかで効き方が変わってきます。

overridesに配置した場合は、単純にserver.xmlの内容を上書きします。

defaultsに配置した場合は、server.xmlに記述されていない設定のデフォルト値を変更することになります。

server.xmlに設定されていない設定項目は、暗黙的にデフォルト値で動作することになりますが、これらの優先順位は以下のようになります。

overrides > server.xml > defaults > 暗黙的デフォルト値

WAS Liberty 最新情報セミナー 2015

31

3. Libertyプロファイルの基本構成

【こんなときどうする？】通信をSSLで暗号化したい

securityUtilityでSSL証明書の生成を行い、SSLフィーチャーを組み込みます。

- 以下のコマンドでSSL証明書を生成します。

```
securityUtility createSSLCertificate --server=<サーバー名> --password=<パスワード>
```

C:\IBM\WebSphere\Liberty\Archive8556\wlp\bin>securityUtility createSSLCertificate --server=defaultserver --password=password
ステップ C:\IBM\WebSphere\Liberty\Archive8556\wlp\usr\servers\defaultserver\resources\security\key.jks を作成中です
サーバー defaultserver の SSL 証明書を生成しました。証明書は SubjectDN CN=XXXXX.ibm.com, OU=defaultserver, O=ibm, C=us で作成されました。
SSL を使用可能にするには、server.xml に次の行を追加してください。

```
<featureManager>  
<feature>ssl-1.0</feature>  
</featureManager>  
<keyStore id="defaultKeyStore" password="xorjLz4sLCgwLTs" />
```

注意！証明書のデフォルトの有効期限は365日！重要

- 上記出力のとおり、server.xmlを編集します。
- SSLでアクセスしてみると...

https://localhost:9443

Before...

After !

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/twlp_sec_create_certificate.html

© 2015 IBM Corporation

32

WAS Liberty 最新情報セミナー 2015

また、Libertyサーバーとクライアント間の通信を、HTTPSにしたい、という場合は上記の手順をとる事で簡単に構成することが可能です。

ただし、詳細なSSLオプションを設定する必要がある場合は、以下を参照してください。

ここでひとつ大事なポイントがあります。デフォルトで作成した証明書は、有効期限は365日、1年になっています。きちんと認識して運用するか、長めに設定しておくか・・・留意してください。最も、本番運用時には自己署名証明書ではなく厳密な管理を行うことと思います。

Liberty プロファイル: SSL 構成の属性

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_ssl.html

【こんなことも出来ます】オリジナルのフィーチャーの開発

独自の、オリジナルのフィーチャーも開発して組み込みます！

- 作業の流れは以下のようになります。
 1. OSGiバンドルを作る
 2. jarを作成する
 3. フィーチャーマニフェストファイルを作成する
 4. バンドルを`${wlp.user.dir}/extension/lib` に配置
 5. フィーチャーマニフェストファイルを`${wlp.user.dir}/extension/lib/features`に配置
 6. (ローカライズしている場合) `${wlp.user.dir}/extension/lib/features/l10n` にローカライズファイルを配置する

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_feat_example.html

© 2015 IBM Corporation

さらに、Libertyではオリジナルのフィーチャーも作成し、組み込むことが可能です。

3. Libertyプロファイルの基本構成

【こんなことも出来ます】設定サンプルがほしい – configUtility コマンド

Update FP6

設定サンプルを参照しながら簡単にサーバー構成が可能です。

- 設定のサンプルをリポジトリから取得することができます。
- 以下のようなことができます。
 - ◆ find
 - Liberty環境に適用可能なfeatureをLibertyRepositoryから検索します。
 - ◆ install
 - featureをインストールします。
 - 右は例ですが、設定が必要な箇所が<!-- TODO: -->で示されています。

出力例

```
C:\Users\IBM_ADMIN>C:\IBM\Websphere\Liberty\Archive\85556\wlp\bin\configUtility.bat install remoteAdministration

検索された構成スニペットをダウンロードしています...

<server description="Remote Administration Sample Configuration">
  <!-- NOTE: This file is for reference only. -->
  <!-- Enable restConnector-1.0 feature -->
  <featureManager>
    <feature>restConnector-1.0</feature>
  </featureManager>

  <!-- Simple administrative security configuration. -->
  <!-- TODO: Set the security configuration for Administrative access -->
  <quickStartSecurity userNames=" ${adminUser}" userPassword=" ${adminPassword}" />

  <!-- TODO: Set the SSL keystore password -->
  <keyStore id="defaultKeyStore" password=" ${keystorePassword}" />

  <!-- TODO: Set HTTP Endpoint attributes -->
  <httpEndpoint id="defaultHttpEndpoint"
    host="*"
    httpPort="9080"
    httpsPort="9443" />

  <!-- TODO: Use readDir and writeDir to specify directories that remote
  clients are allowed to have read and write access. There can be
  multiple readDir and writeDir elements. Replace writePath and readPath
  variables with your choice of locations or remove them if not needed. -->
  <remoteFileAccess>
    <writeDir>${writePath}</writeDir>
    <readDir>${readPath}</readDir>
  </remoteFileAccess>
</server>
```

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html

© 2015 IBM Corporation

34

WAS Liberty 最新情報セミナー 2015

便利情報が続きますが、server.xmlの設定の更に、設定サンプル/テンプレートをダウンロードして参照することができます。

configUtilityというコマンドを利用して、installすると、サンプルが出力されますので、これをベースにして、<!-- TODO: -->で示された部分を編集することで設定を行うことができる便利コマンドです。

Liberty profile: configUtility command

[http://www-](http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html)

[01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html](http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html)

【こんなことも出来ます】スキーマを確認したい – ws-schemagen.jar ツール

- 以下のコマンドで、Libertyプロファイルの設定ファイル（server.xml）のスキーマファイル出力することができます。

```
java -jar ws-schemagen.jar [options] outputFile
```

- オプションとして、以下が指定できます。
 - ◆ --encoding=charset
 - 出力ファイルのキャラクタセットを指定します。
 - ◆ --locale=language
 - 言語指定が可能です。
 - ISO-639の2文字の言語コードと、ISO-3166のカントリーコードを、"_"で繋いで指定します。

出力例

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ext="http://www.ibm.com/xmlns/dde/schema/annotation/ext">
  <xsd:complexType name="com.ibm.ws.security.authentication.tai.interceptor">
    <xsd:annotation>
      <xsd:documentation>トランス・アノテーション・インターセプターを定義します。 </xsd:documentation>
      <xsd:appinfo>
        <ext:label>トランス・アノテーション・インターセプター</ext:label>
      </xsd:appinfo>
    </xsd:annotation>
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="library" type="com.ibm.ws.classloading.sharedlibrary">
      <xsd:annotation>
        <xsd:documentation>共有ライブラリ構成の ID の参照。 </xsd:documentation>
        <xsd:appinfo>
          <ext:label>共有ライブラリー</ext:label>
        </xsd:appinfo>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
</xsd:schema>
```

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_schema_gen.html

© 2015 IBM Corporation

Libertyプロファイルはいろいろな便利ツールが用意されていますが、さらにはserver.xmlファイルのスキーマファイルまで出力するコマンドが用意されています。

Liberty プロファイル: コマンド・プロンプトから Liberty プロファイル構成スキーマを生成


[http://www-](http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html)

[01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html](http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/rwlp_command_configutil.html)

3. Libertyプロファイルの基本構成

【こんなことも出来ます】アクセス先DBのDDLを生成したい – ddlGen ユーティリティ

データベースへアクセスする設定がされている場合、DDLを生成することができます。

- データベースにアクセスするフィーチャー毎に、DDL（Data Definition Language）を生成します。 
- 前提
 - ◆ ddlGen ユーティリティが検索するサーバーのパスを環境変数 WLP_USER_DIR で指定しておきます。
 - ◆ ddlGenユーティリティーを実行する前に、対象のサーバーを起動しておく必要があります。

ddlGen generate <server_name>

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/twlp_ddlgen.html © 2015 IBM Corporation

36 WAS Liberty 最新情報セミナー 2015

まだまだ続きます。

DBアクセスを行うフィーチャーを利用している場合には、アクセス先DBのDDLを生成するというユーティリティーまで用意されています。

3. Libertyプロファイルの基本構成

3. Libertyプロファイルの基本構成

どう動かす？

© 2015 IBM Corporation

37WAS Liberty 最新情報セミナー 2015

設定が終わったら、いよいよLibertyサーバーを起動し、アプリケーションを導入して動かして見ます。

3. Libertyプロファイルの基本構成

サーバー管理コマンド

■ 基本書式

server <アクション> <サーバー名> [Options]

■ 起動・停止

・ create

: 新規サーバーインスタンスの作成

・ start

: バック・グラウンドでサーバーを起動

・ run

: フォア・グラウンドでサーバーを起動

・ debug

: JVMデバッグオプションで、フォアグラウンドでサーバーを起動

・ stop

: 稼動しているサーバーを停止

■ 管理

・ status

: 指定されたサーバーのステータスを確認

・ package

: サーバー・ランタイムと、サーバー構成・アプリケーションをzipファイルにパッケージング

・ version

: サーバー・ランタイムバージョンを表示

■ 問題判別

・ dump

: サーバーから問題判別資料をダンプし、アーカイブ

・ javadump

: javacoreを出力

・ help

: 詳細オプション含め、コマンドヘルプを表示

まずは作成しないと動きません！

重要

© 2015 IBM Corporation

38

WAS Liberty 最新情報セミナー 2015

Libertyプロファイルを利用する際にもっとも基本的なコマンド群です。従来のフル・プロファイルのWebSphereではstartServer, stopServerなどコマンドが分かれていましたが、Libertyではserverコマンドに様々なコマンドが集約されています。

サーバー・インスタンスの作成は server create <serverName> のように、非常に簡単に作成することが可能です。

server startで起動するとサーバーはバック・グラウンドで起動、server runで起動するとサーバーはフォア・グラウンドで起動します。

server packageコマンドを利用することでLiberty環境をランタイム含めてパッケージすることができます。package機能の詳細については次のセッションで説明があります。

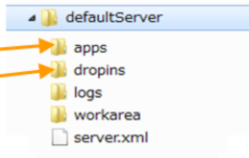
これまでフル・プロファイルではcollector.sh など実施していた問題判別資料の収集も、製品の基本コマンドに統合されています。

3. Libertyプロファイルの基本構成

アプリケーションの導入・構成方法

アプリケーションの導入方法は、静的と動的、2つあります。

- 2つの導入方法
 - ◆ appsディレクトリ下に配置し静的構成
 - server.xmlに<application>要素で構成
 - ◆ ドロップイン・ディレクトリに配置し動的更新
 - アプリケーション・モニターを設定
 - dropinsディレクトリ下に配置
- アプリケーションの配置方法
 - ◆ 圧縮ファイルのまま配置
 - WARファイル, EARファイルをそのままディレクトリー上に配置
 - ◆ ファイルを展開して配置
 - WARファイルの名前でディレクトリーを作成し, WARファイルの中身を展開
 - ファイル単位の更新が可能



© 2015 IBM Corporation

39

WAS Liberty 最新情報セミナー 2015

Libertyでのアプリケーションの導入方法には2つあります。

1つ目は、server.xmlに<application>エレメントを記載していく方法です。もう1つは、ドロップイン・ディレクトリに配置することで、アプリケーションの導入を行う方法です。

配置の方法ですが、圧縮ファイルのまま各ディレクトリに配置することもできますし、ファイルを展開して配置することも可能です。

アプリケーション導入方法1：sevrer.xmlでの構成

- <application>要素により指定
 - ◆ location属性は必須。絶対パス・相対パスやURLで記述
 - ◆ 子要素としてセキュリティー・ロールのバインディングやライブラリー参照を指定することが可能
- location属性を相対パスで指定した場合、
\${server.config.dir}/apps, \${shared.app.dir}から検索される

server.xml

```
<application
  id="SupportTools" location="SupportTools.war"
  name="Support Tools" type="war"
  context-root="/support" />
```

Developer Toolsのアプリケーション構成画面

Application Details

Id:	SupportTools
ロケーション:	SupportTools.war <input type="button" value="Browse..."/>
名前:	Support Tools
アプリケーション・タイプ:	war
アプリケーション・コンテキスト・ルート:	/support
<input checked="" type="checkbox"/> 自動的に開始	



<application>要素にネストして
<application-bnd>要素や
<classloader>要素を定義できる

© 2015 IBM Corporation

まず、<application>エレメントを指定して導入する方法です。

アプリケーションに関して、セキュリティ属性やクラスローダーや共有ライブラリの設定などが必要な場合に利用します。また頻繁な更新が発生しない本番環境ではこちらで明示的に構成を行ってアプリケーションを導入していきます。

アプリケーションのロケーションを指定する際に、相対パスで指定した場合、サーバー構成ディレクトリ下のアプリケーション・ディレクトリや共有アプリケーション・ディレクトリから検索されます。

アプリケーション導入方法2：ドロップイン・ディレクトリ

- コード開発段階でのアプリケーションや、アプリケーション・セキュリティ・ロール等追加構成が必要ないアプリで利用可能
- アプリケーション・モニターを構成
 - ◆ <applicationMonitor>要素で指定
 - ◆ デフォルトは\${server.config.dir}/dropins
- ドロップイン・ディレクトリにファイルをコピーする
 - ◆ 拡張子により自動的にタイプ（EAR/WAR/WAB）を判別
 - WAR/WABは、拡張子を除いたファイル名がアプリケーションのコンテキストルートとなる
例） Snoop.war は http://localhost:9080/Snoop/ でアクセス
- アプリケーションのアンインストール
 - ◆ ファイルを削除すると、自動的にアプリケーションがアンインストールされる



© 2015 IBM Corporation

41

WAS Liberty 最新情報セミナー 2015

次にドロップイン・ディレクトリで指定する方法です。さきほど見たようなセキュリティなど詳細な設定が不要な場合は、こちらを利用できます。特に開発段階などで頻繁なアプリケーションの更新が発生する場合には、こちらの導入方法が適しています。

サーバーの構成の中で、アプリケーション・マネージャーというコンポーネントを有効にし、アプリケーションをドロップインするディレクトリを指定します。

ディレクトリから削除することでアプリケーションはアンインストールされます。

3. Libertyプロファイルの基本構成

アプリケーションから参照するライブラリの管理

■ 共有ライブラリ

◆ 明示的にライブラリを構成

- ファイル・セット、フォルダー、ファイル

◆ クラス・ローダーから明示的に関連づけ

■ グローバル・ライブラリ

◆ 全てのアプリケーションから使用可能なライブラリ

◆ 以下の2箇所に配置可能

- `${shared.config.dir}/lib/global` : 稼動する全サーバー上のアプリ
- `${server.config.dir}/lib/global` : 当該サーバー上のアプリ

◆ 以下の場合にライブラリを参照

- 明示的にクラス・ローダー定義がない場合
- 明示的にクラス・ローダー定義が存在し、関連付けている場合

◆ BestPractice: クラスローダーの分離と依存性明確化のため、明示的にクラスローダー定義を構成

Configuration Structure

Define the main contents of the configuration in this section.

フィルター入力

サーバー構成

- フィーチャー・マネージャー
- HTTP エンドポイント
- 共有ライブラリ
 - ファイル・セット
 - フォルダー
 - ファイル
- Web アプリケーション
 - クラス・ローダー

追加...

削除

上へ

下へ

ファイル・セット

ベース・ディレクトリー: `${server.config.dir}/myLib/`

参照...

☒ 大/小文字の区別

組み込むパターン:

参照...

除外するパターン:

参照...

スキャン深度:

クラス・ローダー

責任:

parentFirst

参照...

ライブラリ参照:

追加

削除

Child element count: 0

共有ライブラリ参照: `shareLib1`

参照...

Child element count: 0

© 2015 IBM Corporation

42

WAS Liberty 最新情報セミナー 2015

アプリケーションが参照するライブラリの管理についてご説明します。

まず共有ライブラリです。明示的にライブラリとして構成し、それを共有ライブラリとしてクラスローダーの管理画面から関連付けます。

もう一つの定義方法が、グローバル・ライブラリです。これは全てのアプリケーションから使用可能となるライブラリであり、スコープにより2つの配置場所があります。稼動する全サーバー上のアプリで共有する場合は、共有構成ディレクトリ下の `${shared.config.dir}/lib/global` に配置します。サーバー・インスタンス内で稼動するアプリケーション内で共有する場合には `${server.config.dir}/lib/global` に配置します。このディレクトリに配置されたライブラリは、アプリケーション内に明示的にクラスローダー定義がないアプリケーションに関しては自動的に参照されます。また明示的なクラスローダー定義が存在する場合も、明示的に関連付けを行っている場合はライブラリを参照します。

なお暗黙的に参照はされますが、クラスローダーの分離を依存性の明確化のために、明示的にクラスローダー定義を構成して頂くことがBest Practiceとされています。

WAS Liberty 最新情報セミナー 2015

42

3. Libertyプロファイルの基本構成

開発環境

ここからは、開発環境へのLibertyプロファイルの導入、
開発環境のセットアップについて説明します。

© 2015 IBM Corporation

43WAS Liberty 最新情報セミナー 2015

WebSphere Developer Toolsで開発！

開発・設定はWebSphere Developer Toolsで行います！

- WebSphere Developer Tools for Eclipseとは？
 - ◆ 無償で提供される WebSphere開発者向けの軽量Eclipseプラグイン
 - JavaEE、OSGi、Web2.0アプリケーション開発、アセンブル、デプロイ
 - Libertyプロファイルのサーバー構成に利用
 - ◆ WDTのサポート環境
 - Eclipse 4.5/4.4/4.3
 - ◆ Eclipseマーケット・プレイスから入手可能



© 2015 IBM Corporation

44

WAS Liberty 最新情報セミナー 2015

開発作業は、WebSphere Developer Tools を利用して行います。

WebSphere Developer Toolsは、無償で提供されるWebSphere開発者向けの軽量Eclipseプラグインです。

WebSphere向けに、Java EEの開発やアセンブルをヘルプする機能が提供されます。さらに今回のLibertyプロファイルのサーバー構成をサポートする機能が豊富に含まれています。

WDT導入方法としては、Eclipseのヘルプ→ Eclipseマーケット・プレイスで WebSphereと検索して頂ければ見つけることが可能です。

またWASDevのサイトを利用した、より簡単な導入方法がありますので、そちらを紹介します。

【参考】WebSphere Developer Tools を導入 (1)

まず、WebSphere Developer Toolsを導入

① WASdevサイトにアクセス <https://developer.ibm.com/wasdev/>

Download Liberty profile in Eclipse

Use Eclipse to develop and deploy applications to WAS Liberty.

1. Download and install Eclipse 4.5 for Java EE Developers (Mars).

2. Install WebSphere Developer Tools into Eclipse using one of the following methods:

• **Online:** Drag and drop the **Install** button on the toolbar in Eclipse, then follow the prompts:

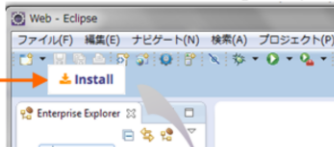
Install

• **Local:** Download the update site to install later.

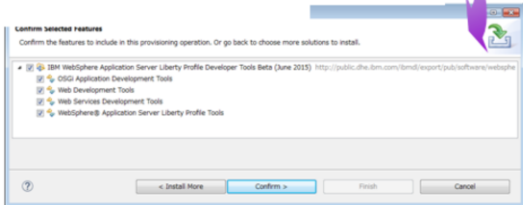
Download

3. In the Servers view of Eclipse, right-click to create a new V to download and install the WAS Liberty runtime.

② エクリプスのtoolbar にDrag&Drop



③ 導入ウィザードが起動されるので、ウィザードに従い導入

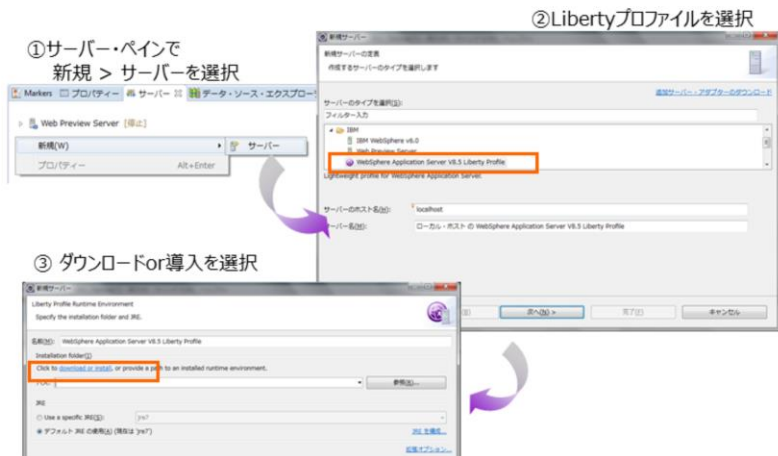


© 2015 IBM Corporation

まずWASdevサイトにアクセスしてください。こちらのダウンロード・ページに WebSphere Application Server Developer Tools for Eclipseというリンクがありますので、こちらでご自身のEclipseのバージョンにあるリンクを開きます。開いた画面において、ブラウザーから「Install」のアイコンを、ご利用中のEclipseのツールバーにDrag&Dropしてください。自動で WebSphere Developer Toolsの導入ウィザードが開始されますので、ウィザードに従い導入を完了させます。

【参考】WebSphere Developer Tools を導入 (2)

Libertyプロファイルをダウンロードします。



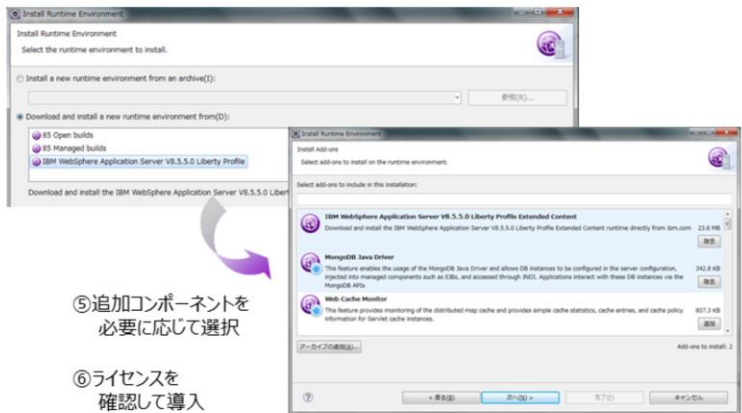
© 2015 IBM Corporation

サーバー・ペインで「新規」>「サーバー」を選択します。サーバーのタイプを選択する画面で WebSphere Application Server V8.5.5 Liberty Profileを選択し、「次へ」をクリックします。
次にLibertyのインストール・イメージを選択する画面になりますので、まだ未導入であれば「download or install」のクリックを選択します。

【参考】WebSphere Developer Tools を導入 (3)

Libertyプロフィールを導入します。

④ 事前ダウンロード済みのパスか、ダウンロードを選択



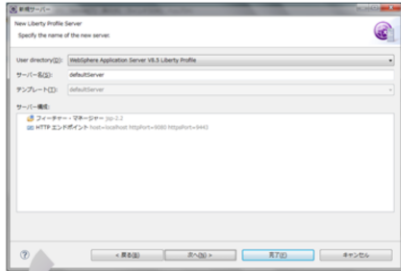
© 2015 IBM Corporation

事前にインストール・イメージがある場合はそのパスを、まだイメージがない場合には「Download and Install」を選んで、Liberty Profileを選択します。「次へ」を押すと、追加コンポーネントを選択する画面が出ます。サンプル・アプリケーションなど必要に応じて追加コンポーネントを選択して「次へ」を押します。あとはライセンスに同意して導入します。

【参考】WebSphere Developer Tools を導入（4）

Libertyプロファイル サーバーを作成します。

⑦ 導入が完了したら、作成するサーバー名を指定



⑧ Liberty プロファイルの導入から
サーバーの構成まで完了！



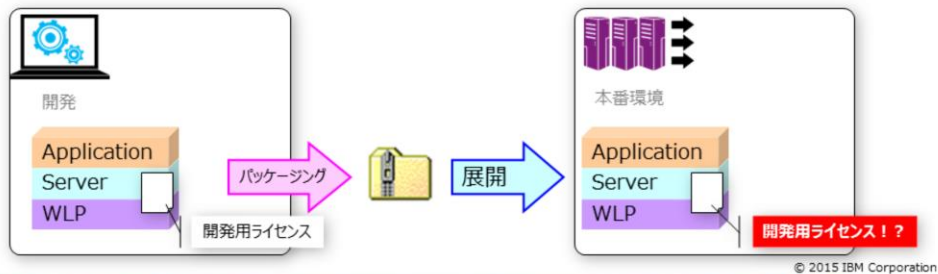
© 2015 IBM Corporation

導入が完了したら作成するサーバー名が確認されますので、指定して構成します。デフォルトでは defaultServer という名前でサーバーが作成されます。

Libertyプロファイル・サーバー・パッケージの実働環境へのデプロイを考える

開発環境のLibertyをそのままパッケージすると・・・

- Libertyプロファイルには、パッケージングの機能があり、ランタイム環境まで含めてアーカイブファイルにパッケージし、これを解凍する事で他環境への導入が可能です。
- この機能を有効活用したLibertyプロファイルの運用を考えると、
「**開発環境で作成したLibertyプロファイルのランタイムを含めてパッケージして、そのまま本番へリリースする**」
といったケースが考えられます。
- しかし、そのままだと・・・実働環境で開発用ライセンスのまま稼動してしまうことになります！



49

WAS Liberty 最新情報セミナー 2015

Libertyプロファイルサーバーは、パッケージング機能によって、ランタイム環境まで含めたアーカイブを作成することができるようになっています。

この機能を活用すると、新しいインフラ運用のカタチが見えてきます。

開発環境においてWDTなどを利用して効率よくサーバー環境まで作成しておき、それをそのままパッケージングして、パッケージしたアーカイブファイルをテスト環境、本番環境などの実働環境にコピーする、というインフラ構築・運用です。

これをビルドやデプロイ・ツールと組み合わせて自動化するというのがImmutable Infrastructureや、Infrastructure as code などの新しいインフラ管理・構築につながっていく訳ですが、その際一点注意が必要が点があります。

開発環境のライセンスで使用していたランタイムをそのままパッケージングして、実働環境で解凍・デプロイして使用すると、開発環境のライセンスのまま稼動してしまうという点です。

実働環境へのデプロイ後のライセンス管理は大丈夫ですか？

パッケージ展開での実働環境へのデプロイ時には、ライセンス管理に注意！

重要

- このような場合、**開発環境用のライセンスを、実働環境のライセンスに置き換える**必要があります。
- 置き換えるためのライセンスファイル（jar）が提供されています。
- 導入方法は、以下のとおり、実働環境に移行した後、ライセンスファイルを展開します。
- この手順をデプロイの手順に組み込んでください！

```
C:\Users\IBM_ADMIN>java -jar C:\Work\wlp-nd-license-8.5.5.jar
IBM WebSphere Application Server Network Deployment V8.5.5.6
を使用、抽出、またはインストールする前に、プログラムのご使用条件の承認と通知のライセンス情報に同意する必要があります。
以下の使用条件をよくお読みください。
?
以下で「同意する」を選択すると、使用許諾契約の契約条件（該当する場合は、IBM 以外の契約条件を含む）を受諾することになります。
同意しない場合は、「同意しない」を選択してください。
```

[1] 同意する、または [2] 同意しないを選択: 1

ライセンス・ファイルのディレクトリを入力するか、ブランクのままにしてデフォルト値を受け入れます。
デフォルトのターゲット・ディレクトリ: C:\Users\IBM_ADMIN

製品ファイルのターゲット・ディレクトリ? C:\IBM\WebSphere\LibertyArchive8556
ライセンス・ファイルが正常に適用されました。

- 現在どのライセンスで稼動しているかは、以下のコマンドで確認できます。

```
C:\IBM\WebSphere\LibertyArchive8556\wlp\bin>C:\IBM\WebSphere\LibertyArchive8556\wlp\bin\productInfo.bat version
製品名: WebSphere Application Server
製品バージョン: 8.5.5.6
製品エディション: BASE_ILAN
ILAN (International License Agreement for Non-Warranted Programs) が付与されていると、保証対象外、すなわち開発用ライセンスと考えられます！
```

- もしくは、起動時のcosole.logに以下のような出力がされます。

```
[15/06/25 20:40:01.446 JST] 000000001 com.ibm.ws.kernel.launch.internal.FrameworkManager A CWWKE0100t この製品は、開発用であり、および限定的な実働使用に限り
にライセンスが交付されています。全ライセンス情報は以下で表示可能です。
https://public.dhe.ibm.com/bmd/export/pub/software/websphere/wasdev/license/base_ilan/8.5.5.5/files/ja.html
```

http://www-01.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/twlp_ins_upgrade_archive.html

© 2015 IBM Corporation

この場合、開発環境ライセンスを、実働環境ライセンスで置き換えます。

メディアやPAオンラインなどで入手した正式版製品には、置き換えるためのライセンスファイルが付属していますので、これを実行することでライセンス・ファイルの適用が行われます。

現在どのライセンスで動いているか、は、起動時のログメッセージや、producInfoコマンドで確認可能です。必要に応じて、ビルド、デプロイのプロセスの中にこれらの処理を組み込んでください。

3. Libertyプロフィールの基本構成

© 2015 IBM Corporation

51

WAS Liberty 最新情報セミナー 2015