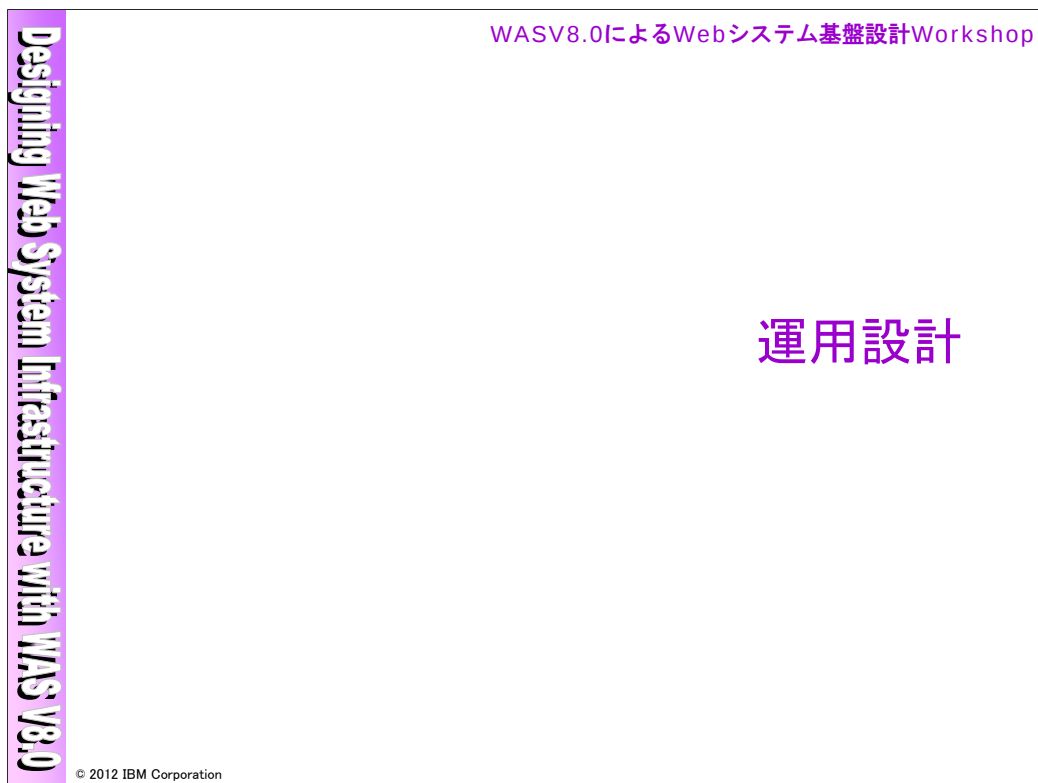




当セッションの目的

- ミッション・クリティカルなWebシステム環境におけるWASV8.0を使用した運用設計に必要な知識の習得
 - ◆ WASが提供している運用管理機能を理解する。
 - ◆ 要件に基づいたWAS運用管理パターン、機能の選択基準、メリット・デメリットを理解する。

Designing Web System Infrastructure with WAS V8.0

Agenda

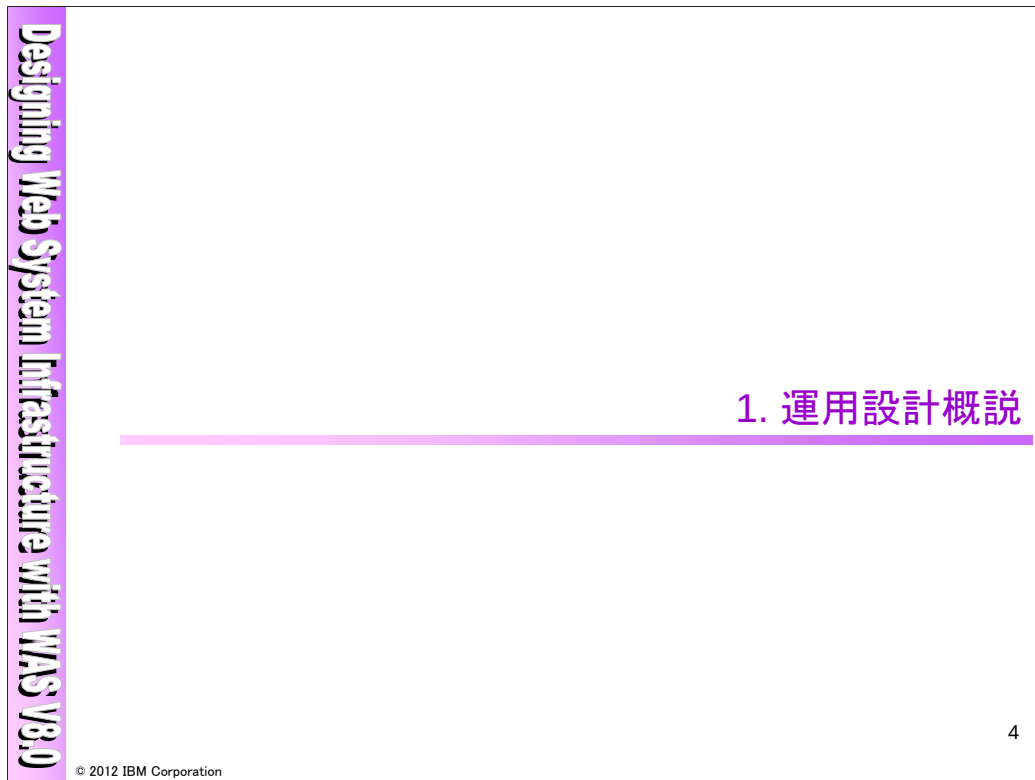
1. 運用設計概説
2. 構成管理
3. 定常運用
4. 非定期運用
5. マイグレーション

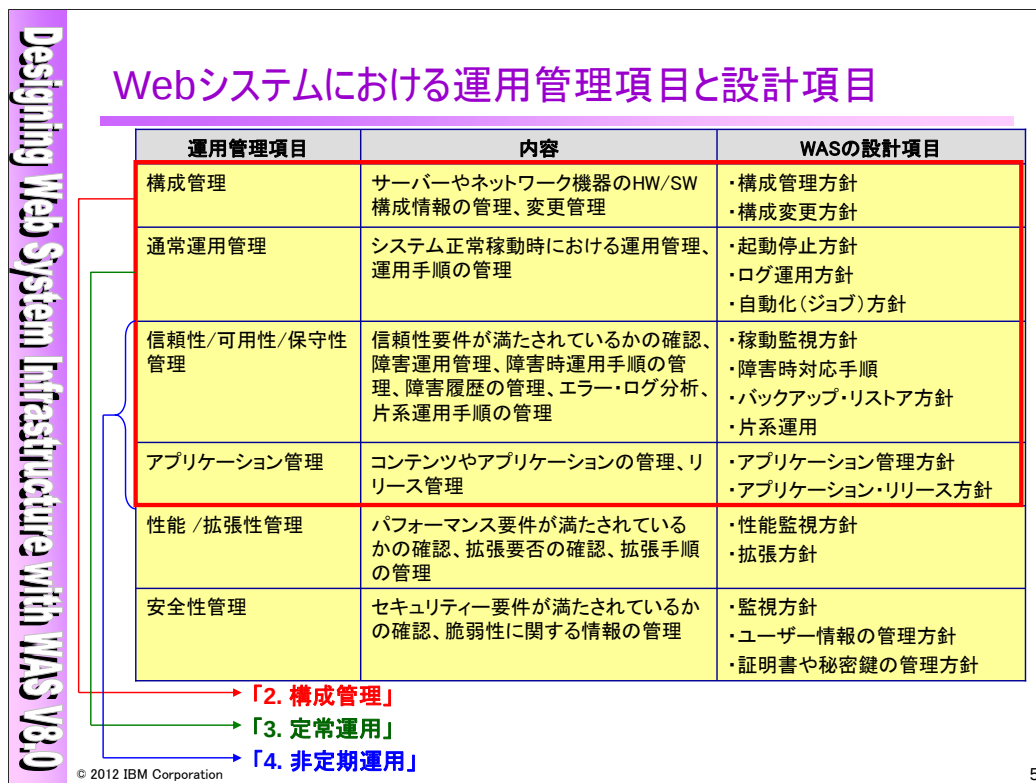
まとめ・参考文献

参考資料

© 2012 IBM Corporation

3





5

運用管理の項目は、必ずしも明確な定義があるわけではありませんが、ITILやGSMMethodでの定義をもとに、IHS/WASを中心とするWebシステムに必要な運用管理項目をまとめると上記のようになります。ここでは、構成管理、通常運用管理、信頼性/可用性/保守性管理、アプリケーション管理、性能/拡張性管理、安全性管理の6つに分類しています。それぞれの運用管理項目における要件に対して、環境/コスト制約などを考慮して、どのレベルまで実現するかを検討します。運用設計の中には、運用業務の自動化や障害検知の仕組みなどシステム設計に取り入れなければならないものもあるので、できるだけ早い段階から検討し、設計を進めていくことが重要です。

当セッションでは、上記の赤い四角で囲まれた部分を対象としています。WASV8.0の既存機能および新機能の説明、それらを使用した運用設計、設計時の考慮事項などについてご紹介します。性能管理に関しては「パフォーマンス設計」セッション、安全性管理に関しては「セキュリティ設計」セッションをご参照ください。

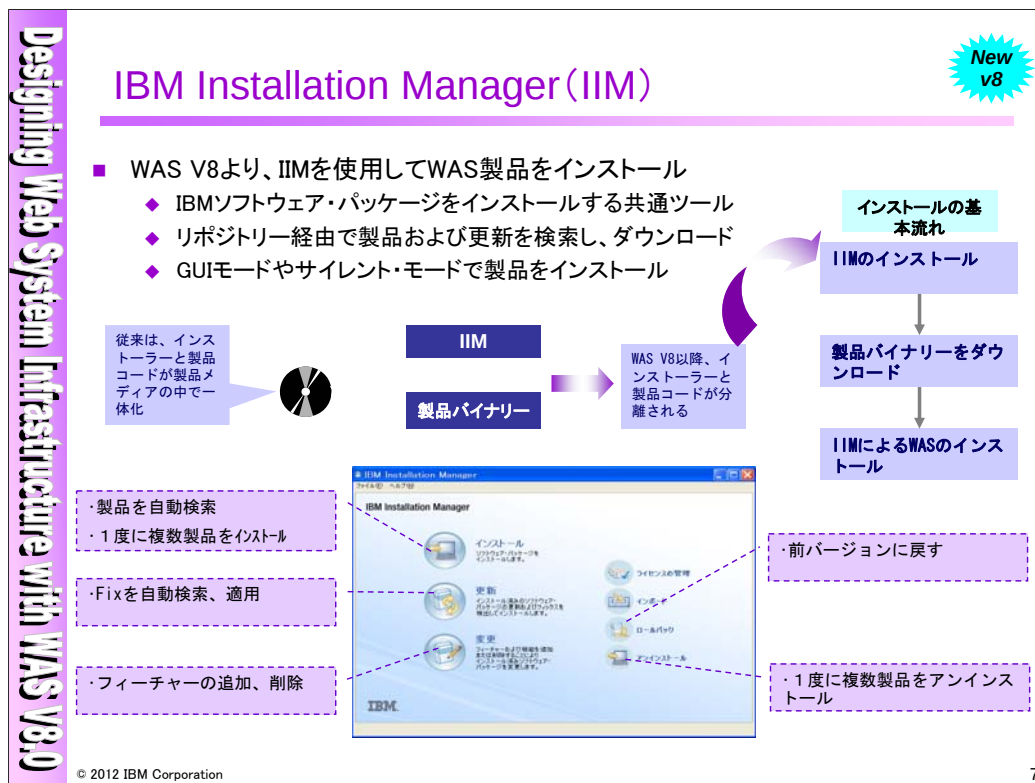
Designing Web System Infrastructure with WAS V8.0

2. 構成管理

- 2-1 製品インストール**
- 2-2 構成情報の管理
- 2-3 WAS管理ツール
- 2-4 構成管理に関する設計ポイント

6

© 2012 IBM Corporation



・WAS V7までは、インストーラーは製品そのもの一部として捉えます。インストーラーは製品イメージに組み込まれ、製品リリースごとに、その製品固有なインストーラーを使用して、製品のインストールを行ってきました。製品やプラットフォームによって、インストーラーもそれぞれ異なります。

・WAS V8より、すべてのWAS製品のインストールは単一のツールであるIBM Installation Manager (IIM) によって行うようになりました。

・もともとIIMはIBM Rational Desktop製品をインストールするためのツールですが、最近ではWASを始め、IBMの複数の製品のインストールおよび保守用に使用されるようになりました。

・IIMは、リモートまたはローカルのリポジトリを使用して、WAS製品をインストール、変更、更新、ロールバック、アンインストールすることができる単一のインストール・プログラムです。

・IIMは、使用可能なパッケージ（製品、FixPack、iFixなど）の判別および表示、前提条件、相互依存性をチェックし、選択されたパッケージのインストールを行います。また、IIMを使用して、インストールされたパッケージを容易にアンインストールすることもできます。

・IIMは、一度に複数製品をインストールすることが出来、一度のインストール・更新でFixを全て適用した状態にすることが可能です。

・WAS V8より、WAS関連製品のインストールには必ずIIMを使用することになります。

・IIMは、集中インストール・マネージャー (CIM) や、IBM Packaging Utility と併用することで、複数台環境など企業向け大規模インストールを可能にした基本ツールとなります。

・IIMの最大な特徴は、インストールとメンテナンスの統合です。

・IIMの入手先

－製品メディア

－Webからダウンロード: <http://www.ibm.com/support/docview.wss?rs=180&uid=swg24023498>

IIMのを使用してインストール可能なWAS製品例:

Application Client for IBM WebSphere Application Server、Edge Components、IBM HTTP Server V8.0、Pluggable Application Client for IBM WebSphere Application Server、WebSphere Customization Toolbox、WebSphere DMZ Secure Proxy Server、Web Server Plug-ins

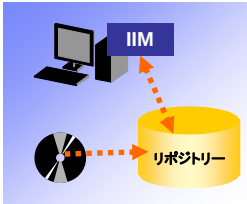
Designing Web System Infrastructure with WAS V8.0

インストール時の環境設定-リポジトリ

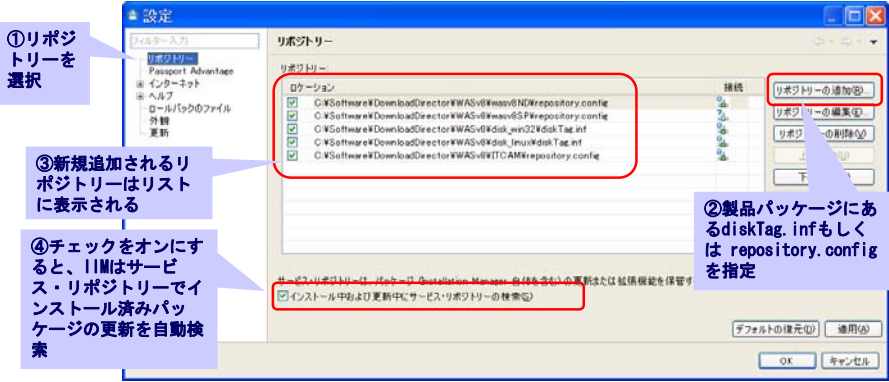
■ リポジトリとは

- ◆ 製品パッケージを格納する場所
 - ローカル・リポジトリ: IIMと同一マシン 上にある
 - リモート・リポジトリ: ネットワーク上識別可能なディレクトリー／Webサイト、FTPサイト
 - サービス・リポジトリ: IBMサイトにあるリモート・リポジトリ

■ リポジトリの設定



New
v8



①リポジトリを選択

③新規追加されるリポジトリはリストに表示される

④チェックをオンにすると、IIMはサービス・リポジトリでインストール済みパッケージの更新を自動検索

②製品パッケージにあるdiskTag.infもしくは repository.configを指定

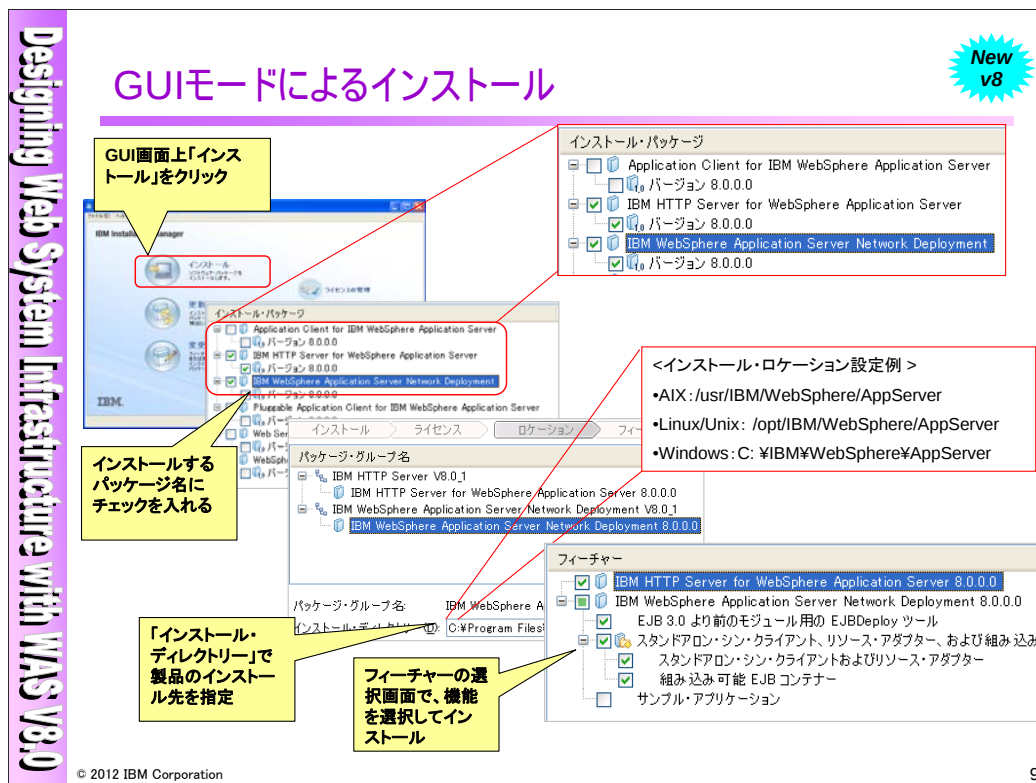
© 2012 IBM Corporation

- ・IIMを使用してインストールできる各ソフトウェア製品は、「パッケージ」と呼んでいます。
- ・リポジトリとは、パッケージのインストール、変更、ロールバック、更新、またはアンインストールを行う場合のデータの格納場所です。
- ・リポジトリに格納されているパッケージは単なるフラット・ファイルですが、製品バイナリーとパッケージ構造を表すメタデータ情報の両方が含まれています。
- ・パッケージをIIMと同一マシン 上に格納したロケーションはローカル・リポジトリです。
- ・パッケージをネットワーク上識別可能なリモート・ディレクトリーやWebサイト／FTPサイトに格納した場合、リモート・リポジトリと呼んでいます。
- ・IBMDメインにあるリモート・リポジトリはサービス・リポジトリと呼んでいます。
- ・サービス・リポジトリは IBMDメイン にあるリポジトリで、製品の更新が含まれています。各 IBM 製品には、製品に固有のサービス・リポジトリへのリンクが組み込まれています。IBM Installation Manager は、サービス・リポジトリと、リポジトリ設定ウィンドウのリポジトリ・テーブルにリストされたリポジトリの両方を検索します。
- サービス・リポジトリのリンクを表示したり、変更したりすることはできません。設定「インストール中および更新中にサービス・リポジトリの検索」をオンにすると、Installation Manager は、IBMDメインにアクセスし、インストール済みの製品について使用可能な更新を検索します。この設定はデフォルトで選択されています。
- ・リポジトリの設定ウィンドウでリポジトリを追加する場合には、diskTag.inf、repository.configのいずれかを選択することができます。

※diskTag.inf は、リポジトリ・ファイルがディスク・イメージからのものであることを示します。

WAS V8.0 によるWebシステム基盤設計ワークショップ

8



リポジトリを設定後、以下の手順でGUIモードによるインストール作業を行います。

1. IBM Installation Manager を開始します。
2. 「インストール」をクリックします。
3. インストール対象のパッケージを選択します。一度に複数のパッケージを選べます。
4. 使用条件の条項に同意し、「次へ」をクリックします。
5. 製品バイナリ のインストール・ルート・ディレクトリを指定します。
6. インストールするフィーチャーを選択します。
7. 要約情報を検討して、「インストール」をクリックします。

Designing Web System Infrastructure with WAS V8.0

サイレント・モードによるインストール(1/2)

New
v8

■ IIMのGUI操作で応答ファイルの記録が可能

▶インストール用応答ファイル作成コマンド発行

▶Unix・Linux(windows)の場合

```
# cd <Installation Managerのインストールディレクトリー> /eclipse
# ./IBMIM -record responseFile -skipInstall agentDataLocation
```

▶Windowsの場合

```
>cd <Installation Managerのインストールディレクトリー> %eclipse
>IBMIM.exe -record responseFile -skipInstall agentDataLocation
```

▶コマンドの発行例(Windows)

```
>cd C:\Program Files\IBM\Installation\%eclipse
>IBMIM.exe -record C:\temp\responsefiles\install_was.xml -skipInstall C:\temp\agent\was
```

▶IIMが起動され、ウィザード形式でインストール手順を進め、完了後IIMを終了させ、指定したロケーションにXML応答ファイルが作成される(実際の製品はインストールされない)

© 2012 IBM Corporation

10

IIMのGUIモードで、サイレント・インストールに必要な応答ファイルを作成できます。

1. コマンド行で、IIMをインストールしたディレクトリーの Eclipse サブディレクトリーに移動します。
2. -record オプションを使用して、コマンド行から IIMを開始します。
3. GUI画面が立ち上がり、適切なりポジトリーを IIM設定に追加します。
4. 「インストール」をクリックします。
5. 以降はGUIモードによるインストール手順と同様になります。

- -skipInstall パラメーターを指定することにより、実際に製品をインストールせず、応答ファイルのみ生成します。
- agentDataLocationには、インストールに必要な作業用ファイルが生成されます。
- responseFileには、応答ファイルが生成されます。

Designing Web System Infrastructure with WAS V8.0



サイレント・モードによるインストール(2/2)

- 記録した応答ファイルを利用し、サイレント・モードでIIMを実行し製品パッケージをインストール
 - ▶サイレント・モードでインストール・コマンド発行
 - ▶Unix・Linuxの場合


```
# cd <Installation Managerのインストール・ディレクトリー> /eclipse/tools
# ./imcl input responseFile -log logFile -acceptLicense
```
 - ▶Windowsの場合


```
>cd <Installation Managerのインストール・ディレクトリー> %eclipse/tools
>imcl.exe input responseFile -log logFile -acceptLicense
```
 - ▶コマンドの発行例(Windows)


```
>cd C:\Program Files\IBM\Installation\%eclipse/tools
>imcl.exe input C:\temp\responsefiles\%install_ihs.xml -log C:\temp\logs\%ihs -acceptLicense
```
 - ▶インストールが正常に終了すると、指定されたログ・ファイルは空になる

C:\IBM\HTTPServer ディレクトリーに com.ibm.websphere.IHS.v80_8.0.0.20110503_0200
がインストールされました。

C:\Program Files\IBM\Installation Manager\%eclipse\tools>

<?xml version="1.0"
encoding="UTF-8"?>

<result>

</result>
 - ▶正常完了しなかった場合、ログ・ファイルにエラー・メッセージが記録される

© 2012 IBM Corporation
11

記録した応答ファイルを指定し、サイレント・モードで製品パッケージをインストールします。

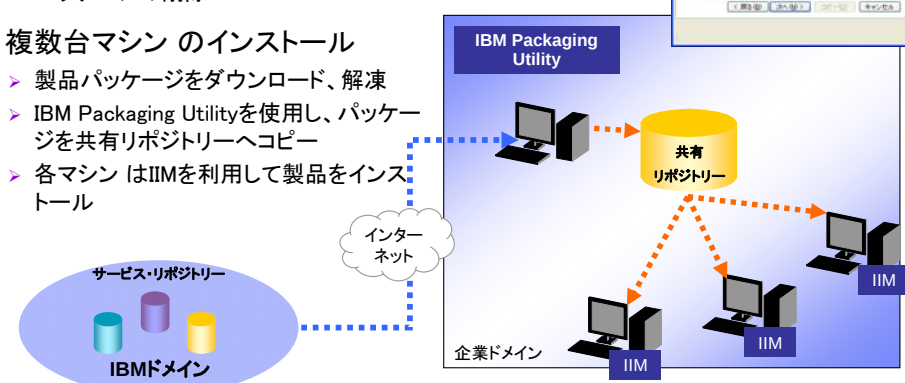
Designing Web System Infrastructure with WAS V8.0

リポジトリを使った複数台のインストール

■ IBM Packaging Utility

- ◆ リポジトリと製品パッケージを一元管理
 - 共有リポジトリ（共有ディレクトリ）を簡単に作成
 - 製品CDイメージやダウンロードしたパッケージをリポジトリにコピー
 - 複数のパッケージをリポジトリにコピー
 - パッケージの削除
- ◆ 複数台マシンのインストール
 - 製品パッケージをダウンロード、解凍
 - IBM Packaging Utilityを使用し、パッケージを共有リポジトリへコピー
 - 各マシンはIIMを利用して製品をインストール





© 2012 IBM Corporation
12

IBM Packaging Utilityはリポジトリとパッケージを一元管理できるツールです。別途入手する必要があります。

<http://www.ibm.com/support/docview.wss?uid=swg24031047>

※最新版はv1.5.1 (2012年1月31日時点)

IBM Packaging Utility では、以下の作業を実行することができます。

- ・パッケージ用の新規リポジトリを生成する。
- ・複数のパッケージを 1 つのリポジトリにコピーする。
- ・複数のバージョンの製品を 1 つのリポジトリにコピーする。ユーザーは、同じリポジトリを指定して、インストール済みの製品を更新します。
- ・コンポーネントを共有している複数のパッケージが 1 つのリポジトリに追加された場合に、ディスク・スペースを節約する。リポジトリに保持されるのは、共有コンポーネントのコピーのうちいずれか 1 つのみです。
- ・不要なパッケージを削除する。
- ・HTTP を介してパッケージをインストールするためのリポジトリを作成する。
- ・CD インストール・イメージまたは IBM リポジトリから、内部サーバーまたはローカル・マシンにあるリポジトリに、パッケージをコピーする。

企業レベルでリポジトリを一元管理するメリット

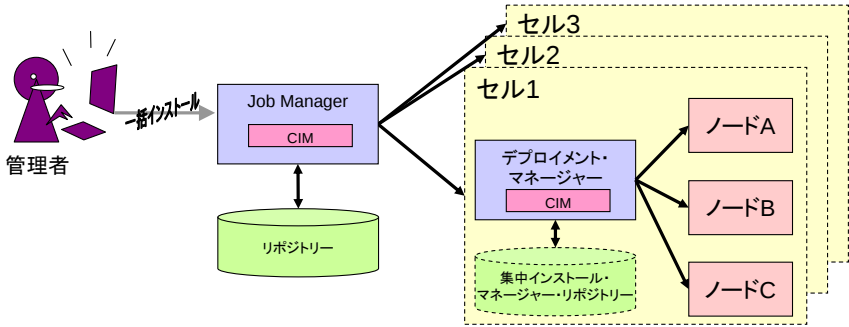
- ・企業の全ユーザーに提供する製品コンテンツを一箇所で集中管理できる
- ・セキュリティをかけやすい

Designing Web System Infrastructure with WAS V8.0



集中インストール・マネージャー (CIM)

- 集中インストール・マネージャー
 - ◆ セルを跨いで複数のリモート・ノードに製品や修正モジュールをPush形式でインストール
 - ◆ IBM Installation Managerでインストール可能な全製品が対象
 - ◆ Job Managerから集中インストール・マネージャー機能が実行可能
 - ◆ インストールのスケジューリングが可能



CIMよりIIMをインストールする手順詳細は「運用設計-参考-」のP.113～116をご参照ください。

© 2012 IBM Corporation
13

WAS V8では、集中インストール・マネージャーの機能が拡張され、製品のインストールおよびFix適用作業がより簡単になりました。V8より集中インストール・マネージャーはセルを跨いで複数のリモート・ノードにもインストール可能となりました。インストール対象は、IBM Installation Managerでインストール可能な全製品となります。Job Managerから集中インストール・マネージャー機能が実行可能であり、インストールのスケジューリングも可能となります。

CIMよりIIMをインストールする手順詳細は「運用設計-参考-」のP.113～116をご参照ください。

Designing Web System Infrastructure with WAS V8.0

WebSphereプロファイルの作成・管理

- プロファイル管理ツール(PMT)
 - ◆ GUIによるプロファイル作成が可能
 - ◆ 64bit版WASに対応 New v8

※プロファイルの削除など作成以外の作業はmanageprofilesコマンドを使用する必要あり

manageprofilesコマンドの詳細は「運用設計-参考-」のP.118をご参照ください。

© 2012 IBM Corporation

14

WASではノードを定義するために、構成に応じたプロファイルを作成する必要があります。

WebSphereプロファイルを管理するため、GUIの『プロファイル管理ツール (Profile Management Tool : PMT)』およびCUIの『manageprofilesコマンド』の2種類の管理方法が提供されています。これらは使用目的に応じて使い分ける必要があります。

まず、プロファイル管理ツールはプロファイルの作成に使用できるウィザードで、『<WAS_ROOT>/bin/ProfileManagement/pmt.<実行ファイル拡張子>』コマンドで起動します。このツールでは、対話式でのプロファイルの作成・拡張と作成済みのプロファイルの確認が可能で、その他プロファイルの削除などの作業はmanageprofilesコマンドを使用する必要があります。

WAS V7までは、64bit版のWASにはプロファイル管理ツールは使用できず、manageprofilesコマンドからプロファイルを作成せざるを得ない背景がありましたが、WAS V8.0より64bit版のWASでも使用できるようになりました。

manageprofilesコマンドの詳細は「運用設計-参考-」のP.118をご参照ください。

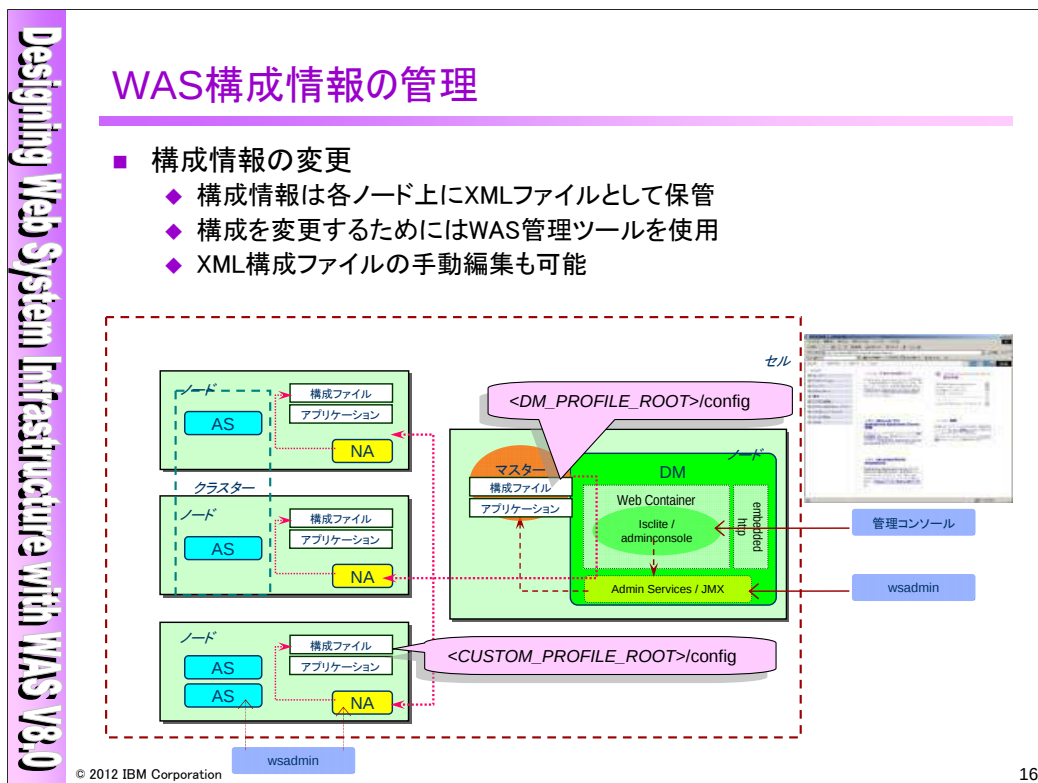
Designing Web System Infrastructure with WAS V8.0

2. 構成管理

- 2-1 製品インストール
- 2-2 構成情報の管理**
- 2-3 WAS管理ツール
- 2-4 構成管理に関する設計ポイント

15

© 2012 IBM Corporation



16

スタンドアロン構成では、`<WAS_PROFILE_ROOT>/config`ディレクトリー以下に全ての構成ファイルが配置されます。セルレベル、ノードレベル、サーバーレベルといった階層構造があり、それぞれの構成がXMLファイルで配置されています。

ND環境では、デプロイメント・マネージャーがセル内のノード全ての構成ファイルを `<DM_PROFILE_ROOT>/config`ディレクトリー以下に保管します。各ノードの構成情報はデプロイメント・マネージャーが保有している構成ファイルをマスター構成として、デプロイメント・マネージャーから各ノードのノード・エージェントを経由して配布されます。この構成ファイルの配布を同期と呼びます。

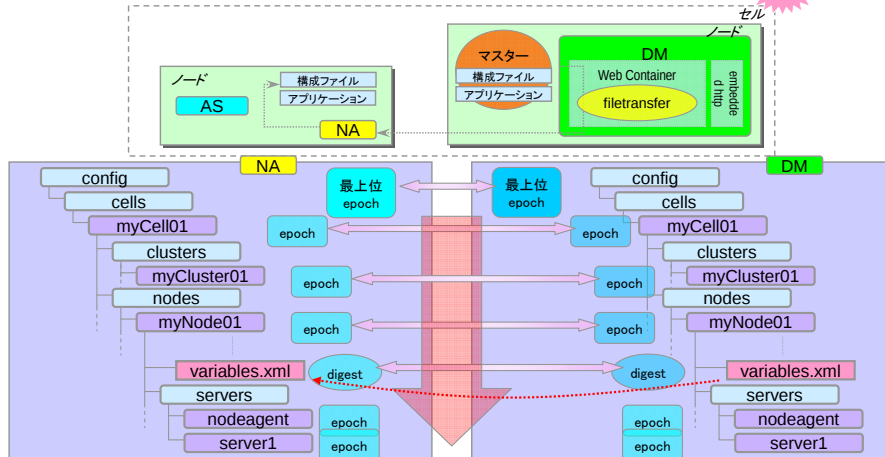
WASのファイル同期化サービスにより、各ノードの `<CUSTOM_PROFILE_ROOT>/config`ディレクトリー以下に、該当ノード分の構成ファイルがコピーされます。デフォルトでは1分間隔で自動同期が行われます。アプリケーション・サーバーは `<CUSTOM_PROFILE_ROOT>/config`ディレクトリー以下の構成ファイルを読み込んで稼動します。

WASの構成を変更するためには、管理コンソールやwsadminといったWAS管理ツールを使用します。XML構成ファイルを手動編集することも可能です。構成ファイル内には自動生成ID番号などが含まれており、手動では構成しきれない箇所があります。

ファイル同期化サービス

■ ファイル同期化サービス

- ◆ DMとNAの構成情報の整合性チェック
 - 整合性が取れていない場合はDMの構成情報をNAに配布
- ◆ 自動同期のON/OFF設定可能
 - 本番稼動環境では自動同期は必須ではない
 - 自動同期をOFFにすると、WASのSSL証明書の自動更新が行われない **注意**



ファイル同期サービスは、デプロイメント・マネージャーが保持する構成情報と各ノードが保持する情報の整合チェックを行います。整合性が取れていない場合には、デプロイメント・マネージャーが持つ構成情報が同期のとれていないノードに対して上書きコピーされます。ファイル同期サービスは、`<CUSTOM_PROFILE_ROOT>/installedApps`ディレクトリー以下に展開されているアプリケーションのファイルも更新します。これは、`config/cells/<cell_name>/applications`ディレクトリー以下のアーカイブされたearの同期がとられ、そのファイルが展開されることで行われます。

デプロイメント・マネージャーと各ノードの同期は、デフォルトでは、1分間隔で定期的に行われますが、管理コンソールやwsadminを使用して実行することもできます。管理コンソールから実行する場合には、管理コンソールで[システム管理]-[ノード]を選択したページで、「同期化」または「完全な再同期」をクリックします。デプロイメント・マネージャーや各ノード上の構成ファイルを手編集で変更した場合なども、デプロイメント・マネージャーの構成ファイルが各ノードに同期化されます。

本番環境では、通常、構成変更は頻繁ではありませんし、アプリケーションの更新などを行った場合に明示的に特定のノードごとの同期を取りたい場合や、勝手に自動更新されるのには問題があるという場合、また、ネットワーク通信による負荷を少しでも下げたいといった場合には、自動同期をOFFにします。

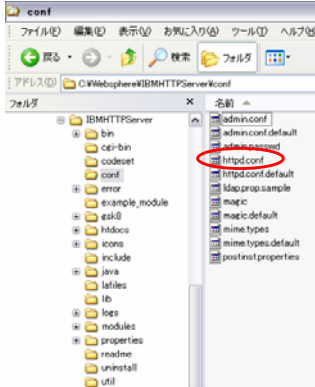
ただし、ND環境でWASのSSL証明書を使用している場合には注意が必要です。「同期化」処理にてDMでマスターとして管理しているSSL証明書の情報も各ノードに同期しているため、自動同期をOFFにした状態で証明書の有効期限が切れた場合、DMのSSL証明書は証明書自動更新の機能によって更新されますが、各ノードのSSL証明書が更新されず有効期限切れとなり、DMと各ノード間でSSL通信が出来ない状態となります。この場合、手動で各ノードの証明書を最新にするといった対応が必要となります。WASのSSL証明書を使用する際の設計方法詳細については「[セキュリティー設計](#)」でもご説明しておりますのでご参照ください。

構成情報の同期を行うコマンド行ツールとして、syncNodeコマンドが提供されています。このコマンドは、構成情報の不具合により、ノード・エージェントが起動できない場合にご利用ください。syncNodeコマンドは、ノード・エージェントが停止された状態でしか実行できません。（セキュリティ設定を誤った場合などに、デプロイメント・マネージャーとノード・エージェントの構成情報の不一致により、ノード・エージェントが起動できなくなる場合があります。そのような場合に、syncNodeコマンドを使用します。）

Designing Web System Infrastructure with WAS V8.0

IHS構成情報の管理

- IHSV8.0構成ファイルの所在
 - ◆ (デフォルト)<IHS_ROOT>/conf/httpd.conf
- 構成情報の変更
 - ◆ IHS構成ファイルの手動編集
 - ◆ WAS管理下の場合はWAS管理コンソールで編集可能



© 2012 IBM Corporation
18

IHS8.0の構成ファイルは、<IHS_ROOT>/conf/httpd.confファイルです。IHS8.0は、Apache HTTP Server 2.2をベースとしていますので、Apache HTTP Server 2.2の構成情報の記述に従って、構成します。

WASV6.0から、Webサーバーを管理コンソールから管理できるようになり、httpd.confファイルも管理コンソール上から編集することができるようになりましたが、httpd.confファイルを事前にバックアップを取ったりすることができませんので、通常は、httpd.confファイルをテキスト・エディターで直接編集します。

IHSV8の管理と構成については、下記のInformation Center の記載を参照ください。

WASV8.0 Information Center 「IBM HTTP Server の管理と構成」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/welc6top_adminconfig_ihs_container.html

Designing Web System Infrastructure with WAS V8.0

Webサーバー・プラグイン構成情報の管理

- V8.0からプラグインのデフォルトのインストール・ディレクトリーが変更 注意
 - ◆ AIXの場合 : /usr/IBM/WebSphere/Plugins
 - ◆ Windowsの場合 : C:\IBM\WebSphere\Plugins
 - ◆ Linux、その他UNIXの場合 : /opt/IBM/WebSphere/Plugins
- V8.0におけるWebサーバー・プラグイン構成ファイルの所在
 - ◆ (ローカル構成のデフォルト) : <PROFILE_ROOT>/config/cells/<cell_name>/nodes/<node_name>/servers/<web_server_name>/plugin-cfg.xml
 - ◆ (リモート構成のデフォルト) : <PLUGIN_ROOT>/config/<web_server_name>/plugin-cfg.xml
- 構成情報の変更
 - ◆ Webサーバー・プラグイン構成ファイルの手動編集
 - ◆ WAS管理コンソール/GenPluginCfgコマンドで構成ファイルを編集/生成
 - ◆ プラグイン構成ファイルの自動生成・自動伝搬
- 考慮事項
 - ◆ プラグイン構成ファイルを手動編集などでカスタマイズしている場合はプラグインの自動生成・自動伝搬によって上書きされてしまうため対応が必要
 - プラグイン構成ファイルを別ディレクトリーに配置
 - プラグイン自動生成・自動伝搬の無効化

《httpd.conf 抜粋》

```
LoadModule was_ap22_module "C:\IBM\HTTPServerV80\Plugins\bin\mod_was_ap22_http.dll"

WebSpherePluginConfig "C:\WebSphereV80\AppServer\profiles\Custom01\config\cells\QUEENMARYCell01\nodes\QUEENMARYNode01\servers\webserv1\plugin-cfg.xml"
```

プラグインの設定や生成の詳細は「運用設計-参考-」のP.119-120をご参照ください。

© 2012 IBM Corporation
19

プラグインのデフォルトのインストール・ディレクトリーはV7.0までは<IHS_ROOT>/Pluginsでしたが、V8.0から変更となっているのでご注意ください。

Webサーバー・プラグイン構成情報のデフォルトの配置ディレクトリーは、WASとWebサーバーが同一OS上で稼動するローカル構成とWASとWebサーバーが異なるOS上で稼動するリモート構成で異なります。ローカル構成の場合は、<PROFILE_ROOT>/config/cells/<cell_name>/nodes/<node_name>/servers/<web_server_name>に配置され、リモート構成の場合は、<PLUGIN_ROOT>/config/<web_server_name>に配置されます。

IHSがどのプラグイン構成ファイルを読み込んでいるかを確認するには、IHSの構成ファイルであるhttpd.confファイルの最後に追加されているWebSpherePluginConfigディレクティブに記載されている構成ファイルのパスを確認するのが確実です。

Webサーバーのローカル構成、リモート構成の詳細については、下記のInformation Center の記載を参照ください。

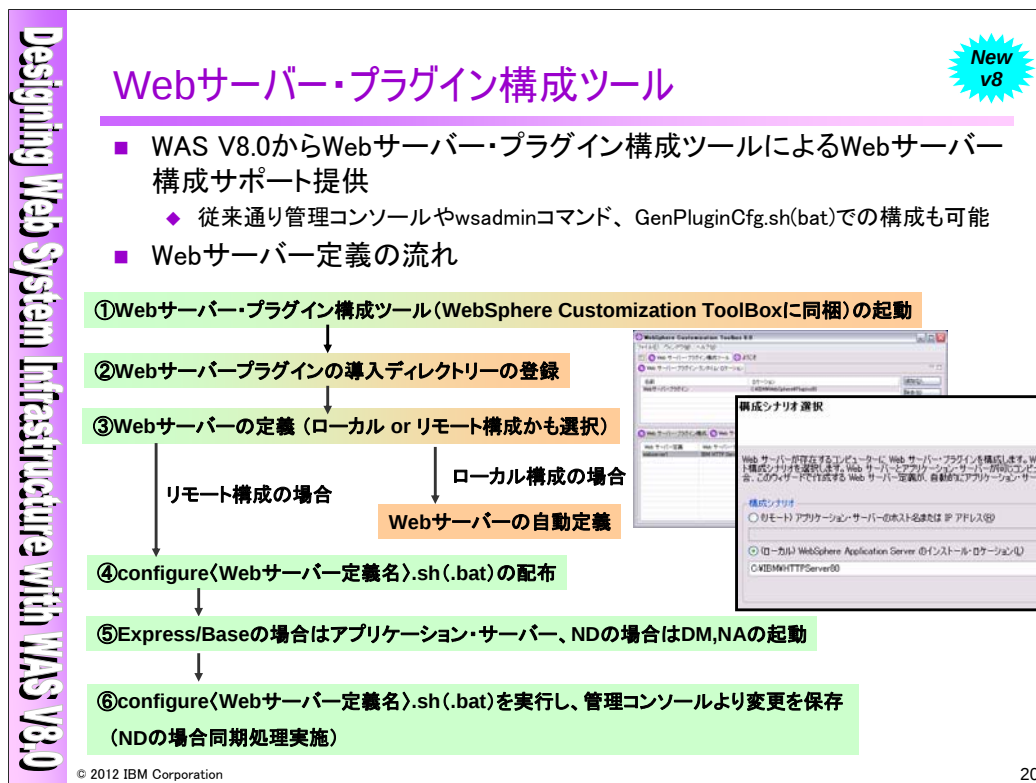
WASV8.0 Information Center 「Web サーバーのトポロジー・ダイアグラムおよびロードマップの選択」
http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tins_road_plugins.html

プラグイン構成ファイルの生成、編集は基本的には、管理コンソールまたはwsadminスクリプト、GenPluginCfgコマンドを使用して行います。IHSとWASの構成を1対1構成とする場合には、GenPluginCfgコマンドの使用が役立ちます。

また、プラグイン構成ファイルには自動生成・自動伝搬するサービスがあり、こちらはデフォルトで設定されています。自動生成・自動伝搬は新規アプリケーションが関連アプリケーション・サーバー上でデプロイされた場合や、Web サーバー定義が保管された場合等の、Web サーバー環境が変更されるたびに、プラグイン構成ファイルが再生成され関連サーバに伝搬するサービスになります。

プラグイン構成ファイルを手編集でカスタマイズしている場合には、プラグイン構成ファイルの配置先をデフォルトと同じにしておくと、構成リポジトリ内のプラグイン構成ファイルによって上書きされてしまう危険があります。そのようなことを回避するために、手編集でカスタマイズしたプラグイン構成ファイルを管理する場合には、GenPluginCfgコマンドで生成したファイルを独自のディレクトリーにコピーし、httpd.confファイル内で指定する読み込むプラグイン構成ファイルをその独自のディレクトリーにポイントします。あるいは、管理コンソールから、Web サーバー > [web_server_name] > プラグイン・プロパティ から、プラグインの自動生成および自動伝搬のチェックをはずします。チェックをはずさない場合、ファイル同期化サービスをOFFにしてもプラグインの伝搬は行われますのでご注意ください。

プラグインの設定や生成の詳細に関しては、「運用設計-別紙-」の P.119-120をご参照ください。



V8.0よりWeb Server Plug-insの構成定義をWebサーバー・プラグイン構成ツールによりGUIで行うことが可能になりました(従来通り管理コンソールやwsadminコマンド、GenPluginCfgからも構成はできます)。Web サーバー・プラグイン構成ツールは、アプリケーション・サーバーとの通信用に Web サーバーを構成し、可能な場合はアプリケーション・サーバーに Web サーバー構成定義を作成します。

上記はWebサーバー・プラグイン構成ツールでのWebサーバーの定義(WAS管理コンソール上でWebサーバーを管理するための設定)の流れを示したものとなります。

Webサーバーの定義の流れは、リモート構成かローカル構成かによって異なります。

Web サーバーとアプリケーション・サーバーがそれぞれ別のシステム上にあるようなトポロジーでは、Web サーバー・プラグイン構成ツールでリモート・システム上のアプリケーション・サーバー構成に直接 Web サーバー定義を作成することはできません。このような場合、Web サーバー・プラグイン構成ツールは、アプリケーション・サーバー・システムにコピーできるスクリプトを作成します。このスクリプトを実行して、アプリケーション・サーバー構成内に Web サーバー構成定義を作成します。

Designing Web System Infrastructure with WAS V8.0

2. 構成管理

- 2-1 製品インストール
- 2-2 構成情報の管理
- 2-3 WAS管理ツール**
- 2-4 構成管理に関する設計ポイント

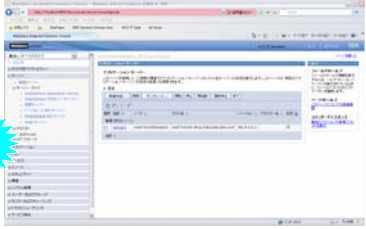
21

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WAS管理ツール

- 構成情報の変更や起動停止オペレーションを行うツール
 - ◆ 管理コンソール
 - ブラウザからアクセスするGUIの管理ツール
 - WASV6.0から Tivoli Performance Viewer も統合
 - Job Managerのジョブの実行が可能
 - Javaダンプ、ヒープ・ダンプの取得が可能
 - ◆ wsadmin
 - JythonとJaclの2種類のスクリプト言語を使用するCUIの管理ツール
 - 新規に開発される場合は、Jythonにて作成することを推奨
 - インタラクティブ・モード(対話形式)もスクリプト実行モードも可能
 - ◆ コマンド行ツール
 - WASに関するオペレーションの一部をスクリプト化したツール
 - 管理プロセスではなくローカルで実行可能
 - ◆ JMXプログラム
 - 管理機能を提供するJavaプログラミングのフレームワーク



各WAS管理ツールの詳細は「運用設計-参考-」のP.122-124をご参照ください。

© 2012 IBM Corporation
22

WASでは、構成情報の変更や起動停止などのオペレーションを行うために4種類の管理ツールが提供されています。

管理コンソールは、ブラウザからアクセスするGUIの管理ツールです。isclite.earというJ2EEエンタープライズ・アプリケーションがスタンドアロン構成の場合はアプリケーション・サーバー上で、ND構成の場合はデプロイメント・マネージャー上で稼動しています。システム管理者は、アプリケーション・サーバーやデプロイメント・マネージャーなどのプロセスに対してブラウザから「http://<hostname>:9060/ibm/console」というURLでアクセスすることができ、管理セキュリティをONにしている場合はSSLポート9043でアクセスします。ポート番号はデフォルトから変更することができます。WASV6.0からTivoli Performance Viewerが管理コンソールに統合され、パフォーマンス・モニタリングを容易に行うことができるようになりました。WAS V8では、デプロイメント・マネージャーの管理コンソールから、V7ではJob Managerから実施していたジョブの実行ができるようになりました。また、管理コンソール上からJavaダンプや、ヒープ・ダンプファイルを取得できるようになりました。

wsadminは、JythonとJaclの2種類のスクリプト言語を使用するCUIの管理ツールです。ノードのホスト名の変更等が行えるAdminTaskオブジェクトやサーバーの起動・停止等を行えるAdminControlオブジェクト等のオブジェクトが提供されています。WAS V7.0において、Jacl言語は“安定化されたフィーチャー”とされました。今後Jacl言語のサポートが除去されることはなく、既存のwsadminスクリプトをJythonに置き換える必要はありませんが、今後のアップデートは、戦略的代替機能であるJythonに重点が置かれることになるため、新規に開発される場合は、Jythonにて作成することを推奨します。<PROFILE_ROOT>/binディレクトリーに実行可能ファイルとして用意されており、アプリケーション・サーバーやデプロイメント・マネージャーなどのプロセスに対してアクセスします。接続方法は-conntypeオプションで指定しますが、SOAP/RMI/JSR160RMI/IPC/NONEから選択することができます。wsadminを使用する際は、インタラクティブ・モード(対話形式)でもスクリプト実行モードでも実行可能です。バッチプログラムなど運用スクリプトに管理系オペレーションを組み込むにはこのwsadminを使用します。

コマンド行ツールは、WASに関するオペレーションの一部をスクリプト化した管理ツールです。<WAS_ROOT>/binまたは<PROFILE_ROOT>/binディレクトリー以下にコマンド行ツールのコマンドが配置されており、アプリケーション・サーバーの起動停止などのコマンドが提供されています。wsadminとは異なり、管理プロセスではなく、ローカルのプロファイル上のアプリケーション・サーバーなどのプロセスに対して操作を実行することができ、プロセスは必ずしも起動している必要はありません。

JMX (Java Management eXtensions) プログラムは、運用管理機能を提供するJavaプログラミングのフレームワークです。JMXのAPIを使用してWAS管理プログラムを開発することができます。

システム運用管理を行う上で、管理プロセスへの接続に使用するプロトコル(SOAP/httpなど)の使用は許可されているか、運用手順としてGUIを使用することが適切か、該当の運用処理を自動化させる必要があるか、などといったお客様の要件や環境制約を考慮し、場面に応じて適切な管理ツールを選択する必要があります。

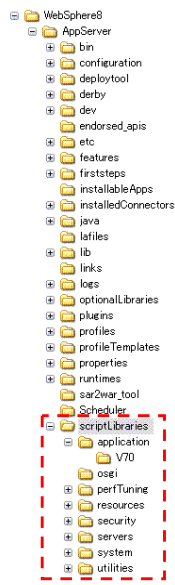
それぞれのWAS管理ツールの詳細に関しては、「運用設計-別紙-」のP.122-124をご参照ください。

Designing Web System Infrastructure with WAS V8.0

スクリプト・ライブラリー



- V7.0以降の管理操作のサンプルJythonスクリプトを提供
- スクリプト・ライブラリーはwsadminが実行されると自動的に読み込まれる
- <WAS_ROOT>/scriptLibraries ディレクトリーに配置
 - ◆ **application** : アプリケーションのインストール・更新・起動・停止・構成変更、など
 - ◆ **osgi** : OSGi Application Console上でOSGiバンドルの開始・停止、など
 - ◆ **perfTuning** : アプリケーション・サーバーやクラスターの設定変更、など
 - ◆ **resources** : リソースのリスト・作成、など
 - ◆ **security** : 許可グループの作成、管理ロールのマップ、など
 - ◆ **servers** : アプリケーション・サーバーやクラスターの作成・削除・起動・停止・構成変更、など
 - ◆ **system** : ノードの停止・同期、ノード・エージェントの停止・再起動、など
 - ◆ **utilities** : ノードの停止・同期、ノード・エージェントの停止・再起動、など



© 2012 IBM Corporation
23

WASV7からの新機能として、管理操作のJythonのサンプル・スクリプトを提供するスクリプト・ライブラリーが提供されました。サンプル・スクリプトでは、管理オブジェクトを組み合わせ、管理操作のスク립トをより簡単な書式で提供しています。スクリプト・ライブラリーでは、Application、osgi、perfTuning、Resources、Security、Servers、System、Utilitiesの8種類のカテゴリ別にサンプル・スクリプトを提供しています。この8種類のカテゴリは<WAS_ROOT>/scriptLibrariesディレクトリー以下にディレクトリー毎に分かれています。例えば<WAS_ROOT>/scriptLibraries/application/V70ディレクトリー以下には、アプリケーション管理に関するJythonサンプル・スクリプトがAdminApplication.pyとAdminBLA.pyというファイルに記述されています。このファイル内でサンプル・スクリプトの引数や使用方法が定義されています。

スクリプト・ライブラリーはwsadminが実行されると自動的に読み込まれます。

Designing Web System Infrastructure with WAS V8.0



【参考】スクリプト・ライブラリーの使用例

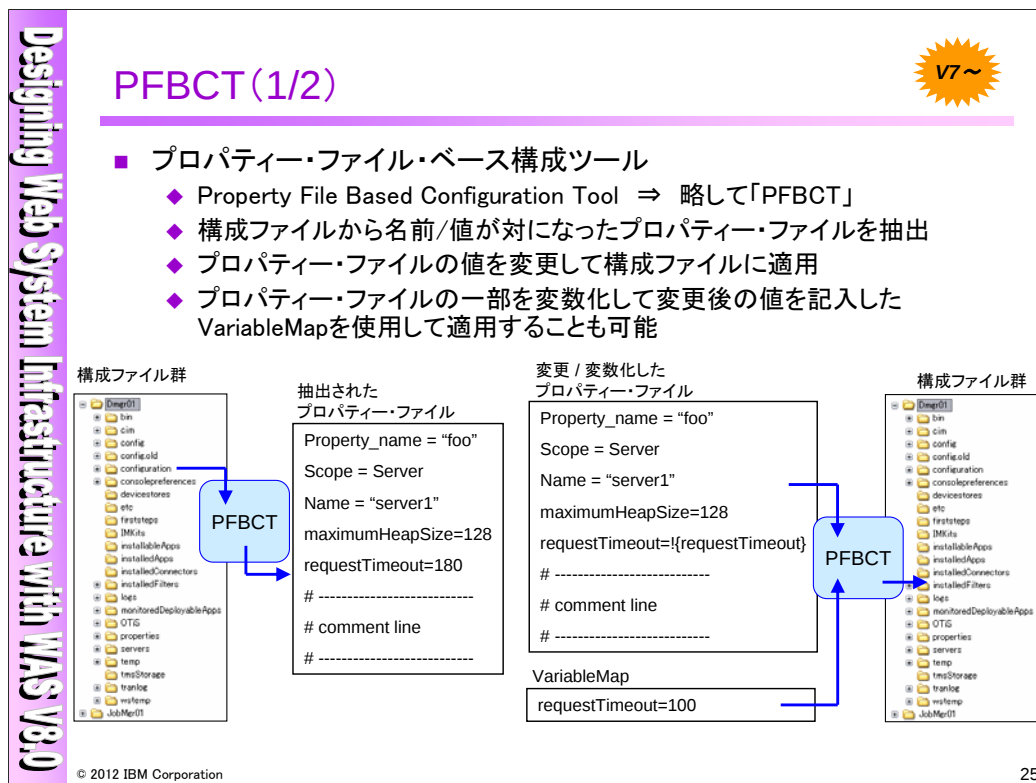
■ 例) アプリケーションのクラスローダー・モードの変更

- ◆ AdminConfigオブジェクト使用の場合
 - 変更するオブジェクトの構成IDを検索してdep変数に設定
`dep = AdminConfig.getid('/Deployment:myApp/')`
 - デプロイ済みオブジェクトを識別してそれをdepObject変数に設定
`depObject = AdminConfig.showAttribute(dep, 'deployedObject')`
 - クラスローダーを識別してそれをclassldr変数に設定
`classldr = AdminConfig.showAttribute(depObject, 'classloader')`
 - modify コマンドを使用して構成オブジェクトの属性を変更
`AdminConfig.modify(classldr, [['mode', 'PARENT_LAST']])`
 - 構成の保管
`AdminConfig.save()`
- ◆ スクリプト・ライブラリー使用の場合
 - 上記の操作がライブラリー内で一つの管理操作として登録

```
AdminApplication.configureClassLoaderLoadingModeForAnApplication
("myApp", "PARENT_LAST")
```

© 2012 IBM Corporation
24

こちらはアプリケーションのクラスローダー・モードを変更する際に、AdminConfigオブジェクトを使用して変更する場合の手順とスクリプト・ライブラリーで提供されているスクリプトを使用した場合の手順を例として挙げています。スクリプト・ライブラリーでは、AdminConfigを使用した場合の手順がAdminApplication.configureClassLoaderLoadingModeForAnApplicationというスクリプトに一つの管理操作として登録されているため、必要な引数を与えるだけで、より簡単な書式で同様の操作を行うことが可能になりました。



プロパティ・ファイル・ベース構成ツール(PFBCT)はV7から登場した管理ツールです。WASの構成ファイルは構成リポジトリ内の複数のサブ・ディレクトリーにXMLとして分散されて配置されており、特定のプロパティ値を構成ファイル群から見つけ出すのは容易なことではありません。PFBCTは、構成ファイル群からプロパティ名と値が対になった、ユーザーにとって現行の構成ファイルよりも分かりやすいプロパティ・ファイルを作成します。ただし、このプロパティ・ファイルは構成ファイルではないため、プロパティ・ファイル内のプロパティ値を変更しただけでは構成変更には反映されません。プロパティ値を変更後にプロパティ・ファイルを構成に適用する、という手順が必要になります。

プロパティ・ファイルの一部のプロパティ値を変数化して変更後の値を記入したVariableMapを使用して構成に適用することもできます。変数は、「{プロパティ名}」の形で記述します。複数のプロパティ値を調整しながら何度も変更するパフォーマンス・チューニングなどの場合は、VariableMapのプロパティ値を変更して構成に適用することで、毎回プロパティ・ファイルの中から該当のプロパティ値を探し出す必要がないので便利です。

Designing Web System Infrastructure with WAS V8.0

PFBCT (2/2)



- PFBCTはAdminTaskオブジェクトのコマンド群
 - ◆ 以下のコマンド・オプションを使用してプロパティ・ファイルで構成を実施する
 - ◆ extractConfigProperties
 - 構成データをプロパティ・ファイルの形式で抽出する
 - ◆ createPropertiesFileTemplates
 - 特定のオブジェクトの作成・削除に使用するテンプレート・プロパティ・ファイルを作成する
 - ◆ applyConfigProperties
 - あるプロパティ・ファイルのプロパティを構成に適用する
 - ◆ deleteConfigProperties
 - プロパティ・ファイルに指定したように構成からプロパティを削除する
 - ◆ validateConfigProperties
 - プロパティ・ファイルのプロパティが有効で構成に安全に適用可能か検証する

PFBCTでサポートされるプロパティの詳細は「運用設計-参考-」のP.125をご参照ください。

© 2012 IBM Corporation
26

PFBCTはAdminTaskオブジェクトのコマンド群であり、以下の5種類が提供されています。

- ・extractConfigProperties・・・構成データをプロパティ・ファイルの形式で抽出します。
- ・createPropertiesFileTemplates・・・特定のオブジェクトの作成・削除に使用するテンプレート・プロパティ・ファイルを作成する。
- ・applyConfigProperties・・・あるプロパティ・ファイルのプロパティを構成に適用します。
- ・deleteConfigProperties・・・プロパティ・ファイルに指定したように構成からプロパティを削除します。
- ・validateConfigProperties・・・プロパティ・ファイルのプロパティが有効で構成に安全に適用可能か検証します。

プロパティ・ファイルには指定したオブジェクトおよびそのオブジェクトにネストされたオブジェクトのすべてのプロパティが抽出されます。例えばサーバーのプロパティが抽出される際には、サーバー、アプリケーション・サーバー、Webコンテナ、EJBコンテナなどのオブジェクトのプロパティも含まれます。

PFBCTでサポートされるプロパティの詳細に関しては、「運用設計-別紙-」のP.125をご参照ください。

Designing Web System Infrastructure with WAS V8.0

拡張機能:PFBCT(1/3)

■ V7.0.0.9 で追加された機能

- ◆ 構成可能なオブジェクト・プロパティの種類が増加
 - 以下のいずれかで構成する
 - - extractConfigPropertiesコマンドでオブジェクトを抽出
 - - InformationCenter にあるテンプレートを修正し、適用することが可能
- ◆ プロパティ・ファイル内で、wsadmin 管理コマンドを指定可能
 - wsadminのコマンドをプロパティ・ファイルで指定し実行させることが可能に

```
wsadmin> AdminTask.createPropertiesFileTemplates(['-propertiesFileName createSiBus.props
-configType GenericType -options [[commandName createSiBus]] '])
```

```
CreateDeleteCommandProperties=true
#SKIP=true
commandName=createSiBus
#
#Properties
#
busSecurity=false #Boolean
```

V7~

◆ 別環境への複製のためのオプションが追加

→環境のコピー、マイグレーション・シナリオへの適用(次ページ)



© 2012 IBM Corporation
27

プロパティ・ファイル・ベース構成ツール(PFBCT)はWAS V7のFixPack 9で機能拡張され、WASの構成において適用範囲が広がりました。

FixPack 9以前は、抽出したプロパティ・ファイルに含まれていたのは、各オブジェクトの主要なプロパティのみでした。しかし、FixPack 9以降、WASの構成オブジェクトのすべてのプロパティを抽出し、変更後に適用できるようになりました。既存の構成からプロパティを抽出する方法の他に、Information Center に各種構成オブジェクトのプロパティ・ファイルのテンプレートが追加されていますので、それをテキスト・エディターにコピーし必要箇所を修正の上、システム構成に適用することができます。

PFBCTで新規にオブジェクトを作成する場合、createPropertiesFileTemplate コマンドを使用することでテンプレート・ファイルが作成され、そのファイルを修正・適用することで作成することができましたが、-configType に指定できるのは、Server、ServerCluster、Application、AuthorizationGroupに限られていました。FixPack 9からは、-configTypeに GenericTypeを指定し、-optionsで管理コマンドを指定できるようになりました。この機能によりこれまで管理コマンドで作成するしかなかったオブジェクトもPFBCTで作成、変更することができるようになります。

もう1つの機能追加はプロパティ・ファイルを別環境へ移植することができるようになった点です。次ページ以降で詳しくご説明します。

Designing Web System Infrastructure with WAS V8.0

拡張機能:PFBCT(2/3)

V7~

■ 別環境への複製

- ◆ V7.0.0.9 まではプロパティ・ファイルの環境固有のIDが含まれており再利用できなかった
- ◆ V7.0.0.9 から環境固有のIDを含めないオプションが追加 → **PortablePropertiesFile**

▶ extractConfigProperties の実行例

```
wsadmin>AdminTask.extractConfigProperties([-
propertiesFileName appserver1.props -
configData Server=appserver1])
```

```
wsadmin>AdminTask.extractConfigProperties([-
propertiesFileName appserver1.props2 -configData
Server=appserver1 -options
[[PortablePropertiesFile true]] ])
```

```
#
# SubSection 1.0.1 # StateManageable Section
#
ResourceType=StateManageable
ImplementingResourceType=Server
ResourceID=Cell=:[cellName];Node=:[nodeName];Server=:[serverName];StateManageable=ID#StateManageableID#1309842541718
AttributeInfo=stateManagement
#
...
```

環境固有のIDが含まれていた

```
#
# SubSection 1.0.1 # StateManageable Section
#
ResourceType=StateManageable
ImplementingResourceType=Server
ResourceID=Cell=:[cellName];Node=:[nodeName];Server=:[serverName];StateManageable=
AttributeInfo=stateManagement
#
...
```

環境固有のIDが無く、他の環境でも使用可能

© 2012 IBM Corporation
28

WAS V7リリース当初は、PFBCTは同一セル・同一環境での構成管理を前提としていたため、抽出したプロパティ・ファイルには、環境固有のXMI IDが含まれていました。

そのため、特定のセルの構成を他の環境へ適用するといった再利用ができませんでした。

WAS V7のFixPack 9からは、extractConfigProperties コマンドを実行する際に、環境固有のIDを含めないための PortablePropertiesFile オプションが追加されました。

このオプションにより、アプリケーション・サーバーやそれより細かいレベルで構成を複製することができます。

また、複製先のセルに対象のサーバーやクラスターなどのオブジェクトがない場合は、GenerateTemplates オプションを true に指定して抽出することで適用時に新規にサーバーを作成することもできます。

WASV8.0 Information Center「複数の環境にわたる移植可能なプロパティ・ファイルの適用」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/txml_portable_prop.html

Designing Web System Infrastructure with WAS V8.0

拡張機能:PFBCT(3/3)

V7~

■ 別環境への複製

- ◆ シナリオ1
 - 既存のWAS V8の構成を利用し、一部の設定情報をコピーした新たなセルを作成する
 - backupConfig/restoreConfig を用いると全く同じ構成となってしまう
 - PFBCTではプロパティー・ファイルの編集により必要な構成情報を繰り返し実行可能

きめ細かく柔軟な構成が可能
- ◆ シナリオ2
 - マイグレーションに使用し、新環境への段階移行を行う
 - マイグレーション・ツールを使う場合、環境をそのまま一度に全て引き継ぐことになる
 - そこから構成情報のみを抽出して、新規作成したWAS8セルに適用する

新規作成したWAS8環境に対して、段階的に旧環境の構成情報の適用が可能

© 2012 IBM Corporation
29

プロパティー・ファイルの移植が可能になったことにより、新たにPFBCTを使用したシステム管理のシナリオが検討可能になりました。

■シナリオ1

作成済みのWAS V8環境から、類似のWAS V8環境を構築する場合に、PFBCTの適用が考えられます。

セル構成やノードの構成を丸ごと複製する場合には、WAS V8の新機能の addNode の -asExistingNode オプションと backupConfig/restoreConfig を組み合わせることで実現できるようになりました。

一方、構成を全て複製するのではなく、例えば、JVMの設定のみ、JDBC関連の構成のみをコピーしたいといった要件がある場合には、PFBCTによって必要なプロパティーのみ抽出し、必要箇所を修正後に適用可能ですので、有用な手段となります。一度抽出したファイルは、パラメーターを適宜修正すれば、他の環境で適用できるため、アセットとして残しておくことができますし、変更管理も容易になります。

■シナリオ2

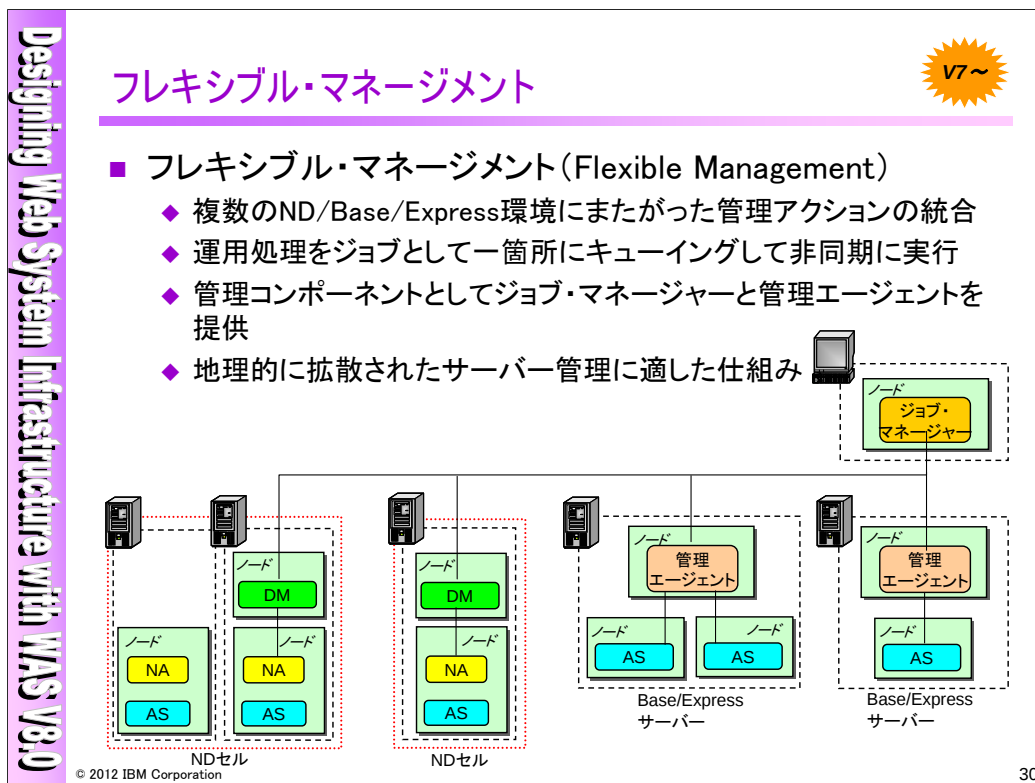
マイグレーションの際にも、PFBCTを活用することで柔軟な構成管理を行うことができます。

WASに付属されるマイグレーション・ツールを使用すると、旧バージョンのWASから構成が全て引き継がれ、WAS V8のセルに移行されます。引き継ぎたくないパラメーターや修正箇所が多くなる場合には、手作業での修正や削除が必要となり工数がかかってしまうことが考えられます。

段階的かつ効率的な移行を行うために次のようなシナリオが考えられます。

まず、一時的にマイグレーション・ツールを使用してWAS V8への移行を実施し、そのセル内で必要なプロパティー・ファイルを抽出します。別途、WAS V8のセルを新規に作成し、抽出したプロパティー・ファイルを必要に応じて修正し、適用することで任意のパラメーターのみを引き継ぐことが可能です。

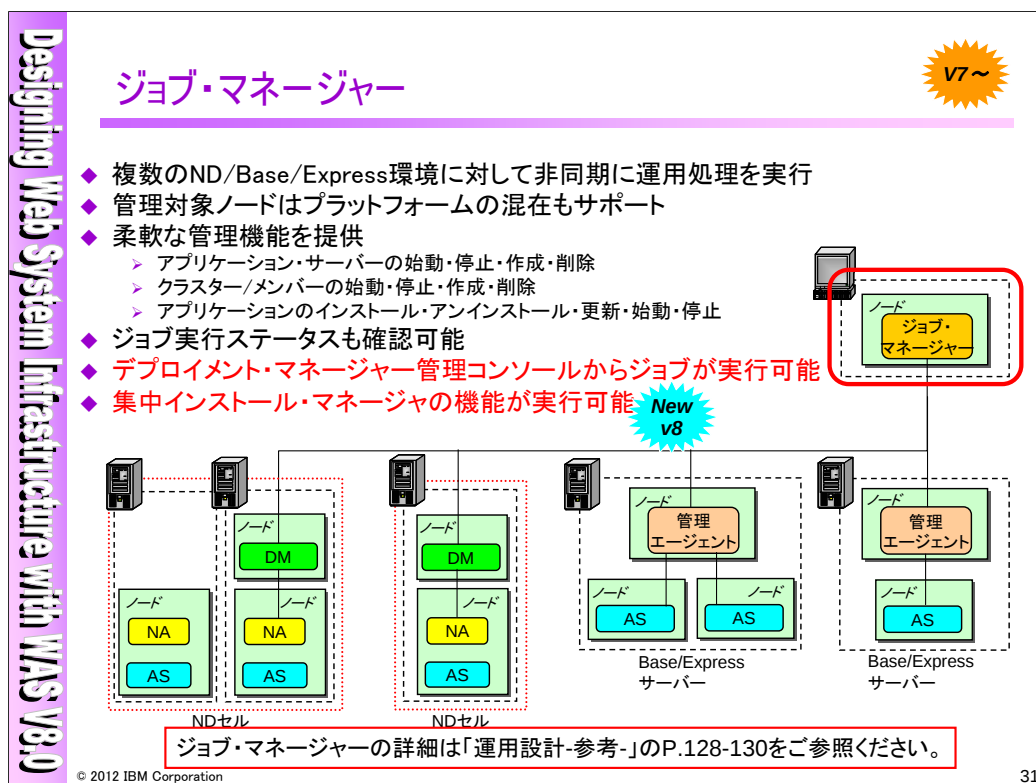
基本的なPFBCTの使い方についても「運用設計-参考-」のP.126-127に記載しておりますのでご参照ください。



WASV6.x以前、スタンドアロン構成では各ノードでサーバーを個々に管理し、またND環境ではデプロイメント・マネージャーがノード・エージェントを通じてノード内のサーバーを管理するしかありませんでした。管理対象は個々のサーバーレベルのため、複数スタンドアロン構成や複数ND環境が存在する場合には、それぞれの環境を個別に管理する必要があり、大規模環境での運用性には限界がありました。

そこで、大規模環境での運用に柔軟性を持たせる、従来の管理概念を拡張したものとして フレキシブル・マネージメントがWASV7から登場しました。既存のデプロイメント・マネージャーやセル・トポロジを置き換えるものではありません。管理トポロジに柔軟性を持たせるための新たな管理コンポーネントとしてジョブ・マネージャーと管理エージェントが提供されます。このフレキシブル・マネージメントによって、複数ND環境や複数Base/Express環境をまたがった一元管理や管理アクションの統合が可能になりました。また、従来の同期型のモデルでは、リモート環境のサーバーを管理するケースなどにおいてネットワークの応答性能が課題でありましたが、フレキシブル・マネージメントでは運用・管理などの処理をジョブとして非同期に複数サーバーに対して実行できるため、支店や拠点などの地理的に拡散されたサーバー管理にも適しています。

ただし、フレキシブル・マネージメントはV7以降の環境のみサポートしており、V6.x以前の環境はサポートされません。



31

ジョブ・マネージャーとは、運用・管理などの処理をジョブとして一箇所にキューイングし、複数 Base/Express環境と複数ND環境に対して非同期で実行する仕組みを提供する管理コンポーネントです。アプリケーション・サーバーを時間を指定して起動させるなど、運用・管理の処理をスケジューリングして実行させたい場合にはジョブ・マネージャーを使用します。また、複数NDセル環境や複数 Base/Express環境が混在する環境においても、ジョブ・マネージャーを使用することで統一的な運用管理が可能になります。ただし、ジョブ・マネージャーから実行できるジョブは限られているため、ジョブ・マネージャーから行えない運用管理については従来どおり、NDセル環境の場合にはデプロイメント・マネージャーで、Base/Express環境では個々のアプリケーション・サーバーで行う必要があります。

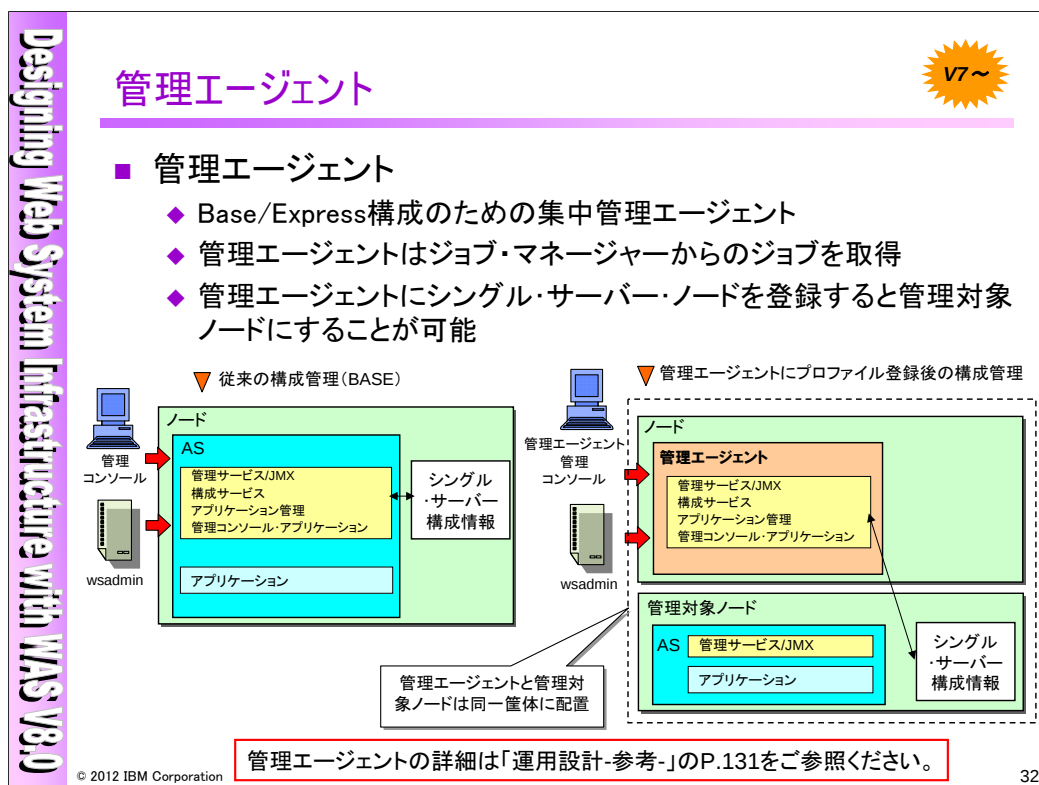
ジョブ・マネージャーが実行できるジョブとしては、アプリケーション・サーバーの始動・停止・作成・削除、クラスター/メンバーの始動・停止・作成・削除、アプリケーションのインストール・アンインストール・更新・始動・停止、などが挙げられます。また、WASV7から登場した管理ツールであるプロパティ・ファイル・ベース構成ツール(PFBCT)から作成するプロパティ・ファイルを、ジョブとして特定のノードに適用させることができます。ジョブ・マネージャーにキューイングされている特定のノードに対するジョブや特定のノードでのジョブの実行結果を確認することもできます。

ジョブの管理はジョブ・マネージャー管理コンソールもしくはwsadminで可能です。

WAS V8からは、デプロイメント・マネージャーの管理コンソールからもジョブの実行が可能になりました。また、集中インストール・マネージャーの機能の利用も可能となり、これによりセルを跨いで複数のリモート・ノードに製品や修正モジュールをインストールすること、インストールのスケジューリングが可能となりました。

ただし、ジョブ・マネージャーで出来る運用管理には限りがあり、構成変更などをジョブ・マネージャーから行うことはできません。

ジョブ・マネージャーの詳細に関しては、「運用設計-別紙-」のP.128-130をご参照ください。



32

管理エージェントとは、WASのBase構成とExpress構成を管理するためにWASV7から加わった管理コンポーネントです。管理エージェントにより、複数のBase構成とExpress構成を集中管理することが可能になっています。また、管理エージェントはシングル・サーバー・ノードの代理としてジョブ・マネージャーからのジョブを取得し、各シングル・サーバー・ノードに対してジョブを実行させることが可能です。シングル・サーバー・ノードを管理エージェントの管理対象にするためにはシングル・サーバー・レベルでの登録が必要です。

管理エージェントからシングル・サーバー・ノードを管理するためには、シングル・サーバー・ノードを管理エージェントに登録する必要があります。その際には管理エージェントと登録するシングル・サーバー・ノードは同一筐体に配置されている必要があります。

シングル・サーバー・ノードが登録されるとそのノードは管理対象ノードとして扱われます。シングル・サーバー・ノードが登録されると構成管理/アプリケーション管理/コマンド・マネージャー/ファイル転送/管理コンソールの機能が管理エージェントに移行され、アプリケーション・サーバーのフットプリントを縮小させ起動時間を短縮させることができます。また、管理エージェントでは、同一ホストの一つ以上のノードにある複数のアプリケーション・サーバプロセスの監視と再起動を行うことができ、これによりアプリケーション・サーバーにおける管理サービス実行のオーバーヘッドを削減することもできます。

管理エージェントに登録されると各シングル・サーバー・ノードの管理コンソールは使用できなくなります。ただし、管理エージェントに登録してもノードの構成情報は管理エージェントに移行されません。各ノードの構成情報はそのまま各ノードが保持します。したがって、管理エージェントはマスター・リポジトリも持ちませんし、管理対象ノード間との同期処理も行いません。そのため、各シングル・サーバー・ノードの構成変更は管理エージェントの管理コンソールから直接シングル・サーバー・ノードの構成ファイルを変更します。また、管理エージェントに複数ノードを登録した場合にも、それぞれのノードの独立性は確保され、各ノードごとに異なるコネクター・ポートや管理ランタイム・インスタンスを持ちます。

また、従来のシングル・サーバー・ノードではwsadminからノード上に複数のアプリケーション・サーバーを作成することは出来ましたが、管理コンソールから複数のサーバーを作成することはできませんでした。管理対象ノードになると、管理コンソールからもノード上に複数のサーバーを作成し、管理することが可能になっております。

管理エージェントの詳細に関しては、「運用設計-別紙-」のP.131をご参照ください。

Designing Web System Infrastructure with WAS V8.0

2. 構成管理

- 2-1 製品インストール
- 2-2 構成情報の管理
- 2-3 WAS管理ツール
- 2-4 構成管理に関する設計ポイント

33

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WASV8.0構成管理に関する設計ポイント

- 製品インストール方針
 - ◆ GUIモード/サイレント・モード、リポジトリ構成、集中インストール・マネージャーの使用
- 構成管理方針
 - ◆ IHS
 - 構成ファイル数(=プロセス数)、構成ファイル名、構成ファイル保管先
 - ◆ プラグイン
 - 構成ファイル数(1:1構成/1:N構成)、構成ファイル名、構成ファイル保管先
 - ◆ WAS
 - マスター構成ファイル(1:1構成/1:N構成)、ジョブ・マネージャー/管理エージェント使用有無
- 構成変更方針
 - ◆ IHS
 - httpd.confを手動編集、WAS管理コンソールにて編集(手動で事前バックアップが必要)
 - ◆ プラグイン
 - plugin-cfg.xmlを手動編集、コマンド/管理コンソールにて生成・配布、自動生成/伝搬要否
 - ◆ WAS
 - 自動化(=スクリプト化)、手動、使用するWAS管理ツール(管理コンソール/wsadmin/PFBCT)、ファイル同期による反映タイミング(自動同期、自動同期間隔、手動同期、同期対象)

© 2012 IBM Corporation
34

ここで、WASV8.0の構成管理に関する設計ポイントをまとめます。

インストール方法についてはGUIにて行うか、サイレント・モードで行うかを検討します。リポジトリについては、ローカル/リモート/サービスのどう構成をとるか、リモートの場合は共有リポジトリを作成するかについて検討します。さらに、サーバー台数が多いようであれば、WASV7からの機能である集中インストール・マネージャー(CIM)を使用すべきかどうかについても検討します。CIMを使用するためには、事前にリポジトリの作成や登録の作業が必要になりますので、事前に検討し、決定しておく必要があります。

構成管理方針として、各コンポーネントの構成管理をどのように行うかを検討します。IHSに関しては、同一筐体上でプロセスを複数起動させる場合はその数分の構成ファイルが必要になりますので、それぞれのファイル名および保管先を決めます。プラグインに関しては、WASクラスタによって複数の系が存在する場合は、IHSとWASを1:1構成にするのか1:N構成にするのかによって、構成ファイルをノード固有にするのか全ノードで共通にするのか異なりますので、構成に応じて構成ファイル名および保管先を決めます。ノード固有にする場合は、WASのファイル同期によって上書きされないように、別ディレクトリに退避させるなどの対応が必要になります。WASに関しては、スタンドアロン構成であれば、そのノードの構成情報がマスターになりますが、セル構成であれば、デプロイメント・マネージャーを配置したノードにマスター構成情報が保管されることになります。Base/Express環境の場合、その環境単体で管理するのか、WASV7からの機能であるジョブ・マネージャーや管理エージェントを使用してデプロイメント・マネージャー配下で管理するのか、についても検討します。

構成変更方針として、各コンポーネントの構成変更をどのように行うかを検討します。IHSに関しては、構成ファイルを手動編集するのか、WAS管理下に置いてWAS管理コンソールにて編集するのかを検討します。既述の通り、WAS管理コンソールにてIHS構成ファイルを編集する場合は、ファイルの事前バックアップ機能などがありますので、手動にてバックアップを取得しておく必要があります。プラグインに関しては、構成ファイルを手動編集するのか、WAS管理コンソールにて生成し、各ノードに配布するのか、を検討します。既述の通り、構成に応じて、GenPluginCfgコマンドを使用するか管理コンソールを使用するか、自動生成/自動伝搬の機能を有効化するか無効化するか、についても検討します。WASに関しては、状況や処理に応じて、自動化するためにスクリプト化するのか、毎回管理コンソールを使用して構成変更を行うのか、WASV7からの機能であるPFBCTを使用するのか、を検討します。また、ファイル同期サービスについても、自動化するのか手動で行うのか、自動化するのであれば間隔はどれくらいにするのか、などを検討します。

Designing Web System Infrastructure with WAS V8.0

3. 定常運用

- 3-1 起動停止**
- 3-2 ログ運用
- 3-3 自動化
- 3-4 定常運用に関する設計ポイント

35

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WASの起動停止

- WASの起動コマンド
 - ◆ デプロイメント・マネージャー
 - `<DM_PROFILE_ROOT>/bin/startManager.sh(.bat)`
 - ◆ ノード・エージェント
 - `<CUSTOM_PROFILE_ROOT>/bin/startNode.sh(.bat)`
 - ◆ アプリケーション・サーバー
 - `<CUSTOM_PROFILE_ROOT>/bin/startServer.sh(.bat) <application_server_name>`

- WASの停止コマンド
 - ◆ デプロイメント・マネージャー
 - `<DM_PROFILE_ROOT>/bin/stopManager.sh(.bat)`
 - ◆ ノード・エージェント
 - `<CUSTOM_PROFILE_ROOT>/bin/stopNode.sh(.bat)`
 - ◆ アプリケーション・サーバー
 - `<CUSTOM_PROFILE_ROOT>/bin/stopServer.sh(.bat) <application_server_name>`

© 2012 IBM Corporation
36

WAS関連プロセスの起動停止時に使用するコマンドです。起動停止の対象となるプロセスは、スタンドアロン構成の場合はアプリケーション・サーバーのみで、ND構成の場合は、デプロイメント・マネージャー、ノード・エージェント、アプリケーション・サーバーです。

アプリケーション・サーバーの起動は、管理コンソールやwsadminコマンドを使用しても行うことができます。管理コンソールやwsadminコマンドを使用してアプリケーション・サーバーを起動する場合は、事前にデプロイメント・マネージャーとノード・エージェントの起動が必要です。デプロイメント・マネージャーとノード・エージェントは、管理コンソールやwsadminからは起動することはできませんので、コマンド行ツールを使用して起動します。また、Windows版を利用の場合は、WASをWindowsのサービスに登録することによって、デプロイメント・マネージャー、ノード・エージェント、アプリケーション・サーバーの各プロセスをWindowsのサービスとして起動を行うことができます。サービスへの登録は、プロファイルの作成中に選択することもできますが、プロファイル作成後にWASServiceコマンドを使用して、登録・解除を変更することができます。

WASServiceコマンドの使用方法的詳細は、下記のInformation Center の記載を参照ください。

WASV8.0 Information Center 「WASService コマンド」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rins_wasservice.html

起動と同様に、管理コンソールやwsadminを使用してアプリケーション・サーバーを停止することもできます。管理コンソールを使用して、ノード・エージェントは停止と再起動、デプロイメント・マネージャーは停止を行うことも可能です。また、起動と同様にWindows版の場合は、Windowsのサービスから管理することも可能です。

Designing Web System Infrastructure with WAS V8.0

IHSの起動停止

- IHSの起動コマンド
 - ◆ HTTP Serverプロセス
 - UNIX/Linux
 - `<IHS_ROOT>/bin/apachectl start`
 - Windows
 - Windowsサービスから起動
 - `<IHS_ROOT>/bin/Apache.exe -k start`

- IHSの停止コマンド
 - ◆ HTTP Serverプロセス
 - UNIX/Linux
 - `<IHS_ROOT>/bin/apachectl stop`
 - Windows
 - Windowsサービスから停止
 - `<IHS_ROOT>/bin/Apache.exe -k stop`

© 2012 IBM Corporation
37

WAS関連プロセスの起動停止時に使用するコマンドです。IHSをWASの管理ノードに統合している場合には、IHSのプロセスを管理コンソールから起動・停止することも可能です。Windows版の場合、Windowsのサービスに登録されますので、Windowsのサービスから管理を行うこともできます。

IHSをリモートWebサーバー構成としていて、WASのセルに統合している場合には、IHSの管理サーバーの起動も必要になります。管理サーバーの起動を行う場合には、`adminctl`コマンドを使用します。Windowsの場合は、Windowsサービスから起動を行うか、Apacheコマンドの引数として、管理サーバーの構成ファイルを `-f admin.conf` のように追加して起動します。

IHS管理サーバーの詳細については、下記のInformation Center の記事を参照ください。

WASV8.0 Information Center 「管理サーバーの管理と構成」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/welc6top_adminconfig_admsserver_ihs_container.html

Designing Web System Infrastructure with WAS V8.0

各コンポーネントの起動停止順序について

- プロセス間の依存関係
 - ◆ ノード・エージェントを起動してからアプリケーション・サーバーを起動
 - ◆ その他のプロセス間の依存関係なし
- プロセス起動停止順序
 - ◆ 起動【バックエンド⇒フロントエンド】
 - 1. DB/WMQ(順不同)
 - 2. ノード・エージェント
 - 3. アプリケーション・サーバー(クラスター)
 - 4. IHS
 - ◆ 停止【フロントエンド⇒バックエンド】
 - 1. IHS
 - 2. アプリケーション・サーバー(クラスター)
 - 3. ノード・エージェント
 - 4. DB/WMQ(順不同)
- ◆ デプロイメント・マネージャーはどのタイミングの起動停止でも影響なし

© 2012 IBM Corporation
38

WASに関連する各コンポーネントのプロセス起動停止順については、プロセス間の依存関係を考慮する必要があります。アプリケーション・サーバーが起動する前に、そのノードのノード・エージェントが起動している必要があります。これは、アプリケーション・サーバーは、ノード・エージェントの提供する機能であるロケーション・サービス・デーモンが利用できないと起動に失敗するからです。一度アプリケーション・サーバーが起動した後であれば、たとえノード・エージェントがダウンしたとしても、アプリケーション・サーバーがそれに伴ってダウンしたり、ユーザー・サービスが提供できなくなるというわけではありません。その他のプロセス間の依存関係はありません。ただし、プロセス間依存関係はありませんが、デプロイメント・マネージャーが停止している場合、ノード・エージェントの起動停止時にDiscovery protocol待ちで固定2分待ちが発生するという問題もあるため注意が必要です。

WASと関連するDBやMQのプロセスとの関連という意味では、通常、DBやMQなどのバックエンドのシステムのプロセスから、順に起動を行います。逆に、停止するときには、フロントエンドのプロセスから順に停止を行います。デプロイメント・マネージャーのプロセスは、管理アプリケーションのランタイム環境であり、業務アプリケーション稼働の観点からは、どのタイミングで起動あるいは停止されても影響がありません。

Designing Web System Infrastructure with WAS V8.0

3. 定常運用

- 3-1 起動停止
- 3-2 ログ運用**
- 3-3 自動化
- 3-4 定常運用に関する設計ポイント

39

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

基本方針

- ログ・ファイルの収集
 - ◆ 定期的なログ収集によりファイル・システムの圧迫を回避
 - サイズが大きくなりすぎるとファイル・システムを圧迫してミドルウェア自身やOSの動作を不安定にする要因となりうる
 - ログ・ファイル出力先となるファイルシステムの使用率を監視する
 - ◆ お客様の要件に沿ったログ保存期間を設定
 - 問題判別が必要なケースを除いてWAS/IHS/プラグインのログは長期間必要なものではない

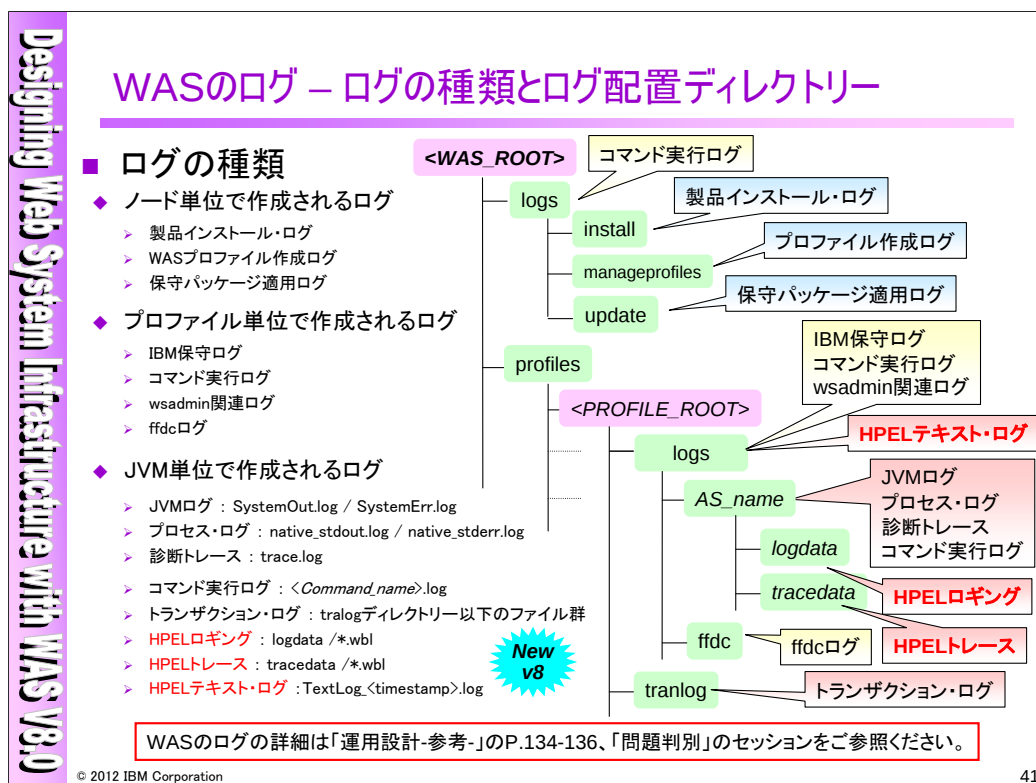
- ログ・ファイルの保管先
 - ◆ ログは製品バイナリーとは別ファイルシステムに配置
 - ◆ WASではWebSphere変数でログ・ディレクトリー名を変更可能
 - LOG_ROOT (ノード・レベル)
 - デフォルトは\${USER_INSTALL_ROOT}/logs
 - SERVER_LOG_ROOT (サーバー・レベル)
 - デフォルトは\${LOG_ROOT}/\${SERVER_NAME}

© 2012 IBM Corporation
40

ログ運用の基本方針として、ログ・ファイルの収集とログ・ファイルの保管先に関して考慮する必要があります。

WASのログ・ファイルは、定期的に記録されるものは循環ログであり、非定期に書き込まれるものは、循環しないものもあります。ログ・ファイルによってファイルシステムを圧迫しないように、定期的に収集を行う必要があります。ログ・ファイルのサイズが大きくなりすぎると、ファイルシステムを圧迫してミドルウェア自身やOSの動作を不安定にする要因にもなります。ログ・ファイル出力先となるファイルシステムの使用率は、監視するなどの対応も検討します。

ログ・ファイルの出力先ディレクトリーを製品インストール・ディレクトリーやプロファイル・ディレクトリーと分けることにより、ファイルの容量監視や、構成のバックアップとログのバックアップ・メンテナンス・ディレクトリーを明確に分離することができます。WASでは、WebSphere変数により、ログ・ディレクトリー名を変更することができます。



WASのログの種類は多岐に渡ります。

ノード単位で作成されるログは、インストール、Fix適用時やプロファイル作成時に記録されるログで、これらは定常的に増加するログではありません。定期的なメンテナンスは不要ですが、問題発生時に必要になる場合がありますので、すべて保管しておくことをお勧めします。

プロファイル単位で作成されるログは、IBM保守ログ、コマンド実行ログ、ffdcログ、wsadmin関連ログがあります。IBM保守ログ、ffdcログは、循環ログですので、必ずしもメンテナンスの必要はありません。

JVM単位で作成されるログは、JVMログ、プロセスログ、診断トレース、コマンド実行ログ、トランザクション・ログがあります。

JVMログは、障害発生時にまず確認するログになります。循環機能が提供されており、ファイルサイズや保管世代数を変更することができます。プロセスログは、通常は、定期的にログが記録されるわけではありませんが、冗長ガーベッジ・コレクションのデフォルトの出力先となりますので、冗長ガーベッジ・コレクションのログを出力している場合には、ログの管理を必ず実施する必要があります。診断トレースも、通常は出力はありませんが、問題判別のためにトレース設定を行った場合には、多くのログが出力されます。ログローテート機能が提供されていますので、問題判別に十分なファイルサイズと保管世代数を設定する必要があります。コマンド実行ログは、コマンド実行ごとに追記されます。トランザクション・ログは、WASのアプリケーション・サーバーがトランザクション・マネージャーとなる場合に、そのトランザクションを管理するためにWASが使用するログです。基本的には、ユーザーが管理する必要はありません。

HPEL (High Performance Extensible Logging) ログはV8.0より追加となったログ方式で、ログ・データをバイナリー形式でリポジトリに蓄積します。初期設定ではロギング方式は基本モード(テキスト出力)となっており、HPELを利用する場合は出力形式を切り替えるために設定を行う必要があります。HPEL形式ではロギングに関する各種設定がありますが、こちらにつきましては後ほど説明いたします。

WASのログの詳細は「運用設計-参考-」のP.134-136、「問題判別」のセッションをご参照ください。

Designing Web System Infrastructure with WAS V8.0

WASのログ設定

- ローテート機能
 - ◆ JVMログはローテート設定可能
 - 時間またはサイズを設定
 - ヒストリー・ログ・ファイル数を設定可能

可能であれば、ファイルのサイズ、
保管世代数を大きく設定することを推奨

- ログ詳細レベルの変更
 - ◆ 出力するロガー・レベルの変更が可能
 - コンポーネント、パッケージ、グループ単位で設定可能
 - ◆ ロガー・レベルの種類

➢ Off(なし)	➢ Config(構成)
➢ Fatal(致命的)	➢ Detail(詳細-低)
➢ Severe(重大)	➢ Fine(詳細-中)
➢ Warning(警告)	➢ Finer(詳細-高)
➢ Audit(監査)	➢ Finest(詳細-最高)
➢ Info(情報)	➢ all(全て)
 - ◆ デフォルトのレベルはInfo(情報)



The image shows two screenshots from the WAS Management Console. The top screenshot, titled '管理コンソール' (Management Console), shows the 'ログおよびトレース' (Log and Trace) section with a red dashed box highlighting the 'ログファイルの回転' (Log file rotation) settings, including file size and history. The bottom screenshot, titled 'ログ詳細レベルの変更' (Change log detail level), shows a tree view of components with a red box highlighting the log level settings for various components, with 'Info' selected as the default.

© 2012 IBM Corporation
42

WASのJVM標準出力ログやJVM標準エラー・ログに関しては、ローテート機能が提供されており、管理コンソールの[トラブルシューティング]-[ログおよびトレース]-[*process_name*]-[JVMログ]で設定します。ファイル・サイズまたは時間をログのローテーションのトリガーにすることができます。また、ヒストリーファイルの保管世代数を設定できます。デフォルトでは、ファイル・サイズは1MBであり、1世代のバックアップファイルがあります。問題判別時に重要なファイルになりますので、ファイル・システムに余裕がある場合には、10MB×10世代のように、ファイル・サイズや保管世代数を大きくしておくことが推奨されます。

WASではコンポーネント、パッケージ、グループ単位で、ログの重大度にて出力するログを変更することが出来ます。管理コンソールの[トラブルシューティング]-[ログおよびトレース]-[*process_name*]-[ログ詳細レベルの変更]で設定します。設定方法はログ詳細レベル・ストリングを直接入力するか、またはグラフィカル・トレース・インターフェースを使用することで設定できます。ロガー・レベルの種類にはOff～allまであり、allにすれば全てのイベントがログに出力されますが、ログレベルを上げるとログサイズが増大して稼動に影響を与える可能性がありますので注意が必要です。デフォルトではInfo(情報)に設定されており、Infoより重大度の高いログが全て出力されます。

ログ・レベル詳細については「問題判別」のセッション、および、InformationCenterを参照してください。

WASV8.0 Information Center 「ログ・レベル」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rtrb_loglevels.html

Designing Web System Infrastructure with WAS V8.0

New
v8

High Performance Extensible Logging (HPEL) (1/3)

- V8ではロギングのモードを選択することが可能に
 - ◆ 通常モード
 - ◆ HPELモード
 - 従来のファイル形式の出力ではなく、ログ・データをバイナリー形式でリポジトリに蓄積するログ方式
 - 必要なデータをフィルタリング、整形してログ分析を実施(管理コンソール、logViewer コマンド)

従来のロギングモードに比べてパフォーマンスが向上、V8ではHPELを推奨

- HPELの構成
 - ◆ HPEL ロギング
 - ◆ HPEL トレース
 - ◆ HPEL テキスト・ログ (オプションでの設定)

© 2012 IBM Corporation

43

WAS V8では、新機能として High Performance Extensible Logging (HPEL) というロギングのモードを選択することができるようになりました。

HPEL モードとは、従来のファイル形式の出力とは異なり、生成されるログ・データをバイナリー形式でリポジトリと呼ばれるディレクトリーに蓄積していくログ方式です。

HPEL モードでは、バイナリー形式でファイルに出力されるため、ログを閲覧する場合には logViewer ツールもしくは管理コンソールで表示させます。

従来のログ方式が置き換わるわけではなく、通常モードとHPELモードを選択することができます。デフォルトは従来の通常モードとなっています。

HPEL モードは、HPELロギング、HPELトレース、HPELテキスト・ログを構成します。HPEL ロギングでは、システム出力、システム・エラーといったログ・データをログ・リポジトリにバイナリー形式で蓄積します。同様にHPEL トレースでは、トレースのデータをトレース・リポジトリに蓄積します。

HPEL モードにすると必ずバイナリー・モードで出力され、デフォルトでテキストでも出力されます。オプションでテキスト・モードの出力有無を変更することも可能です。

Designing Web System Infrastructure with WAS V8.0

New v8

High Performance Extensible Logging (HPEL) (2/3)

■ 通常モードとHPELモードの相違点

- ◆ 通常モードはテキスト形式でのログ書き出し。HPELモードではバイナリー形式（※ただし、HPELでもテキスト・モードでのファイル出力が可能）
- ◆ HPELモードを有効にするとSystemOut.log、SystemErr.logにログは出力されません。
- ◆ HPELの提供するログ・データ（ファイル）の循環方式
 - 日時で新しいファイルを作成する仕組み
 - 古いファイルは設定したポリシーに従いパージする仕組み
- ◆ ログ閲覧方法（HPELではlogViewer経由でログ閲覧）

■ ログおよびトレースの出力先

	通常モード （～V7と同様）	HPELモード （バイナリー・モード）	HPELモード （バイナリー・モード + テキスト・モード）
システム出力 システムエラー	SystemOut.log SystemErr.log	logdata /*.wbl	logdata /*.wbl TextLog <timestamp>.log
診断トレース （詳細とその上位レベル）	SystemOut.log trace.log	logdata /*.wbl	logdata /*.wbl TextLog <timestamp>.log
診断トレース （詳細より下位レベル）	trace.log	tracedata /*.wbl	tracedata /*.wbl TextLog <timestamp>.log

※デフォルトの出力先ディレクトリーは、全て \${SERVER_LOG_ROOT} 以下

© 2012 IBM Corporation

44

通常モードとHPELモードの相違点をまとめています。通常モードはテキスト形式のファイルへの出力、HPELモードではバイナリー形式でlogdataディレクトリーもしくはtracedataディレクトリーへログ出力されます。HPELモードでは、テキスト・モードを有効にすることでテキスト・ファイル形式での出力も可能ですが、システム出力、システム・エラー、トレースデータすべて1つのファイルへ書き出されず。

HPELモードを有効にするとSystemOut.log、SystemErr.logにログは出力されなくなります。

循環方式について、従来のロギング・モードではファイルベースの循環方式でしたが、HPELでは日時で新しいデータファイルを作成し、古いデータファイルは設定したポリシーに従いパージされる仕組みとなっています。設定項目、方法については次頁で詳しく紹介します。

また、閲覧方法も異なります。HPEL モードのバイナリー形式のログはテキスト・エディター等で開けないため、ツールを使用して閲覧する必要があります。

WAS V8.0 によるWebシステム基盤設計ワークショップ

44

Designing Web System Infrastructure with WAS V8.0

High Performance Extensible Logging (HPEL) (3/3)

New
v8

- HPEL設定方法
 - ◆ 管理コンソール
 - 「トラブルシューティング」>「ログおよびトレース」>「アプリケーション・サーバー名」
 - 「HPELモードに切り替え」をクリック後、サーバーを再起動
 - ◆ wsadmin
- 新規ログ・ファイルの開始時間が設定可能
- HPELログギングのページ・ポリシー
 - ◆ 以下の条件でのページが可能
 - ログ・サイズが最大に近づいたとき
 - 最も古いログが存続期限に達したとき
 - 期限またはサイズのいずれかの制約に該当したとき
 - ◆ ファイル・システムの容量不足時にもアクションを指定可能
 - サーバーの停止
 - 古いレコードのページ
 - ログギングの停止
- ログ詳細レベルの変更も可能
- ログの監視方法
 - -logViewer -monitorオプションでリアルタイム監視

アプリケーション・サーバー

アプリケーション・サーバー > appserver1 > appserver1

既存のプロシージャに、ハイパフォーマンス拡張可能ログギング (HPEL) の利用拡張可能ログギングに切り替えることをお勧めします。

[HPEL モードに切り替え](#)

このページを使用して、構成するシステム・ログを選択するか、または、コンポーネント・グループに対するログ詳細レベルを指定します。ログレベルを指定す

ログギングおよびトレース > appserver01 > HPEL ログ構成

このページを使用して、ハイパフォーマンス拡張可能ログギング (HPEL) ログ・オプション、logViewer コマンド (プロファイル bin ディレクトリにあり) を使用するか、「HPEL」して表示できます。

構成 ランタイム

一般プロパティ

* デレクティブパス
\$(SERVER_LOG_ROOT)

☒ ログ・レコード・パージ・ポリシーを使用可能にする

☒ 新規ログ・ファイルの最初の開始時刻: 午前 0 時

ログ・レコードのページ・ポリシー

☒ 最も古いレコードのクリーンアップの開始
ログ・サイズが最大に近づいたとき

ログ・レコードの存続期間
48 時間

最大ログ・サイズ
50 メガバイト

* スペース不足時のアクション
古いレコードのページ

適用 OK リセット キャンセル

© 2012 IBM Corporation 45

HPELを設定する方法は、管理コンソールから設定する方法とwsadminコマンドを使用して設定する方法があります。

管理コンソールから有効化するには、「トラブルシューティング」>「ログおよびトレース」>「アプリケーション・サーバー名」をクリックし、「HPELモードに切り替え」ボタンをクリックします。wsadminで有効化する場合、HighPerformanceExtensibleLogging オブジェクトを有効に設定します。

コマンドの実行例はInformationCenterを参照してください。

WASV8.0 Information Center 「基本モードから HPEL ログギングおよびトレースへの変更」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/ttrb_compToHPEL.html

新規ログ・ファイル開始時刻を1時間単位で設定することが可能です。デフォルトでは午前0時に設定されています。ログ・レコードのページはデフォルトで有効になっています。ページ・ポリシーは、「ログ・サイズが最大に近づいたとき」、「最も古いレコードが存続期限に達したとき」、「期限またはサイズのいずれかの制約に該当したとき」から選択します。スペース不足時のアクションとして、「サーバーの停止」、「古いレコードのページ」、「ログギングの停止」から選択できます。

「サーバーの停止」を選択している際に、ファイル・システムの使用率が100%に達すると、リクエストを処理できなくなり、数分後にサーバー・インスタンスが停止します。

HPEL ログ構成のデフォルトは「ログ・サイズが最大に近づいたとき」で最大ログ・サイズのデフォルトは50メガバイトです。

スペース不足時のアクションはデフォルトで「ログギングの停止」が選択されています。

HPEL トレース構成のページ・ポリシーのデフォルトはHPEL ログ構成と同じく「ログ・サイズが最大に近づいたとき」で、スペース不足時のアクションは、「古いレコードのページ」になっています。ログの詳細レベルは通常モードと同様の手順で変更可能です。logViewerコマンドの-monitorオプションでリアルタイム監視を行うことも可能です。

Designing Web System Infrastructure with WAS V8.0

New
v8

【参考】HPEL の読み方 (1/3)

■ 管理コンソール

The screenshot shows the 'Log and Trace' (ログおよびトレース) management console. It displays a list of server instances with timestamps. Callouts highlight the following features:

- 表示させる内容とレベルでフィルタリングすることが可能** (Filtering by content and level is possible): Points to the '表示内容' (Display Content) section where 'システム出力' (System Output), 'システム・エラー' (System Error), and 'ログおよびトレース' (Log and Trace) are checked. It also shows level dropdowns for '最小レベル' (Minimum Level) set to '警告' (Warning) and '最大レベル' (Maximum Level) set to '致命的' (Fatal).
- サーバーの開始時間ごとにログをフィルタリングして表示可能** (Log can be filtered and displayed by server start time): Points to the 'サーバー・インスタンス' (Server Instance) list.
- ログや文字列で検索、絞りこみが可能** (Search and filter by log or text is possible): Points to the 'フィルター' (Filter) section with input fields for '組み込みログ' (Built-in Log), '除外するログ' (Log to Exclude), and 'メッセージの内容' (Message Content).
- 時刻を指定してイベントを特定することも可能** (It is also possible to specify a time to identify an event): Points to the 'イベントのタイミング' (Event Timing) section with '開始時刻' (Start Time) and '終了時刻' (End Time) fields.

© 2012 IBM Corporation 46

HPELのログ・レコードを閲覧する方法は、管理コンソールで表示させる方法とlogViewerコマンドで出力させる方法の2通りあります。

管理コンソールでは、ナビゲーション・ツリーから「トラブル・シューティング」-「ログおよびトレース」をクリックし、対象のサーバー名を選択します。関連項目から「HPEL ログおよびトレースの表示」をクリックするとログ・ビューアーの画面になります。「内容およびフィルターの詳細」を展開すると、表示させるログの絞り込みのオプションを選択できます。

New v8

【参考】HPEL の読み方 (2/3)

■ 管理コンソール

管理コンソールからログ・ファイルのエクスポートが可能

右クリック

最新表示
選択したスレッドのみ表示
すべてのスレッドを表示
列の選択 ...
エクスポート ...
クリップボードにコピー
サーバー・イン...

サーバー・インスタンスからログ・レコードを表示: 2011/07/11 14:51:16

表示するレコード数: 20

タイムスタンプ	スレッド ID	ロガー	レベル	メッセージ
11/07/11 14:51:19.719	00000000	NSKeyStore	警告	CWPKI0041W: 1 つ以上の鍵がデフォルト
11/07/11 14:51:22.732	00000000	ponentImpl	警告	CWSQ0003W: 適用可
11/07/11 14:51:23.981	000000	oolMgrImpl	警告	VSVR0626W: Object
11/07/11 14:52:18.22		PtpGroup	警告	DCSV1115W: DCS ス
11/07/11 14:52:18.229	00000022	iscoveryRcv	警告	DCSV1115W: DCS ス
11/07/11 15:01:16.439	00000052	ySpaceAlert	警告	com.ibm.ws.logging.h
11/07/11 15:05:16.453	00000052	ySpaceAlert	警告	com.ibm.ws.logging.h

「内容およびフィルターの詳細」の設定

「最大および最小レベル」の設定: 詳細 - 中
「最大レベル」の設定: 詳細 - 中
「最小レベル」の設定: 詳細 - 中
「メッセージの内容」に追加: SessionPl
「組み込みロガー」に追加
「除外するロガー」に追加
「イベントのタイミング」

エクスポートオプションの選択

ログ形式の選択

ログ内容の選択

OK

© 2012 IBM Corporation

47

表示させたログ・レコードの任意の行で右クリックすると、いくつかのオプションを選択できます。特定のログのメッセージやロガーを「内容およびフィルターの詳細」に追加することが可能です。

また、管理コンソールからログ・ファイルをエクスポートすることも可能です。ログ形式の選択で「バイナリー形式」、「基本形式」、「拡張形式」から、ログ内容の選択で「現行のビューのみ」、「リポジトリ全体」から選択します。

Designing Web System Infrastructure with WAS V8.0

New
v8

【参考】HPEL の読み方 (3/3)

■ logViewer コマンド

◆ 構文

```
logViewer.sh (bat) [options]
```

以下のオプションを使用して出力ファイルの指定やフィルタリングを行う

-repositoryDir	-minLevel	-extractToNewRepository
-format	-listInstances	-outLog
-startDate	-thread	-instance
-stopDate	-includeLoggers	-latestInstance
-level	-excludeLoggers	-message
-maxLevel	-monitor	

テキスト・ファイルとして
出力させることができる

-monitor オプションにより
リアルタイムのモニタリングも可能

◆ 実行例

```
logViewer.sh (bat) -outLog /work/waslog/server1_20110711.log  
-startDate 11/7/10 -stopDate 11/7/11
```

```
logViewer.sh (bat) -monitor -includeLoggers SystemErr
```

© 2012 IBM Corporation

48

HPELログを閲覧するための管理コマンドのlogViewerの使用方法です。

logViewerコマンドではオプションを使用してフィルタリングやテキストファイル出力など、自在にログを扱うことができます。上記の各オプションについては、Information Centerもしくはコマンドのhelpオプションを使用して確認してください。

WASV8.0 Information Center 「LogViewer コマンド行 tool」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multipatform.doc/info/ae/ae/rtrb_logviewer.html

Designing Web System Infrastructure with WAS V8.0

IHSのログ

- ログの種類
 - ◆ アクセス・ログ
 - IHSへの全てのリクエストが記録
 - アクセス状況によってはファイルサイズが大きくなるため注意
 - IHSはログ・ファイルに書き込みができない状況になるとリクエスト処理不可
 - 任意のディレクトリーと任意のファイル名を指定可能
(デフォルト)<IHS_ROOT>/logs/access_log
 - ◆ エラー・ログ
 - リクエストの処理が正常に行われなかった場合に記録
 - こちらも状況によってファイルサイズが大きくなるため注意
 - 任意のディレクトリーと任意のファイル名を指定可能
(デフォルト)<IHS_ROOT>/logs/error_log
- ローテート機能
 - ◆ ローテート設定可能
 - サイズもしくは時間をトリガーにローテートさせることが可能
 - ローテートされたファイルは無限に保存されるのでログ収集の仕組みが必要
 - ◆ 例) 24時間ごとアクセス・ログ/エラー・ログをローテートさせる設定


```
httpd.conf
CustomLog "I:/IBMHTTPServer/bin/rotatelog.exe logs/access.log 86400" common
ErrorLog "I:/IBMHTTPServer/bin/rotatelog.exe logs/error_%Y_%m_%d.log 86400"
```

© 2012 IBM Corporation
49

IHSのログには、アクセスログとエラー・ログがあります。デフォルトの設定では、ログ・ファイル・サイズに制限はありませんので、ローテートの機能を使用されることをお勧めします。

IHSのログの詳細は下記のApacheのログの説明をご参照ください。

Apache Document「ログ・ファイル」

<http://httpd.apache.org/docs/2.2/ja/logs.html>

Designing Web System Infrastructure with WAS V8.0

Webサーバー・プラグインのログ

- ログの種類
 - ◆ プラグイン・ログ
 - (デフォルト)<PLUGIN_ROOT>/logs/<web_server_name>/http_plugin.log
 - plugin-cfg.xmlにて配置ディレクトリー、ファイル名、ログ出力レベルを指定可能
- ログ出力レベル
 - ◆ 本番稼動システムでは通常はデフォルト(Error)
 - ◆ ErrorレベルであればIHS起動・停止時とプラグイン障害時にしか出力されないためログ・ファイルはほとんど増えない
 - ◆ ログ・レベルを上げるとログ・サイズが増大して稼動に影響を与えるので注意
 - トレース(Trace)・・・要求プロセスのすべてのステップの詳細がログに記録
 - デバッグ(Debug)・・・要求の処理中に実行された重大なステップがすべてログに記録
 - 詳細(Detail)・・・要求および応答に関するすべての情報がログに記録
 - 状態(Stats)・・・それぞれの要求に対して選択されたサーバー、および要求の処理に関するその他のロード・バランシング情報がログに記録
 - 警告(Warn)・・・異常な要求処理の結果出されたすべての警告およびエラー・メッセージがログに記録
 - エラー(Error)・・・異常な要求処理の結果出されたエラー・メッセージのみがログに記録

© 2012 IBM Corporation
50

Webサーバー・プラグインのログは、デフォルトではhttp_plugin.logです。プラグイン構成ファイルであるplugin-cfg.xmlにて、プラグインログの配置ディレクトリー、ファイル名、ログ出力レベルを指定することができます。管理コンソールでのログ・レベル設定は、「サーバー」>「Web サーバー」>「Web_server_name」>「プラグイン・プロパティー」>「ログ・レベル」から行います。

通常のログレベルがErrorの場合は、ほとんどログの記録はありませんが、ログレベルを変更している場合には、サイズが大きくなりますので、ご注意ください。ログ・レベルは以下の通りで、Error<Warn<Stats<Detail<Debug<Traceの順でより多くの情報が記録されます。

・トレース(Trace)

要求プロセスのすべてのステップの詳細がログに記録されます。

・デバッグ(Debug)

要求の処理中に実行された重大なステップがすべてログに記録されます。

・詳細(Detail)

要求および応答に関するすべての情報がログに記録されます。

・状態(Stats)

それぞれの要求に対して選択されたサーバー、および要求の処理に関するその他のロード・バランシング情報がログに記録されます。

・警告(Warn)

異常な要求処理の結果出されたすべての警告およびエラー・メッセージがログに記録されます。

・エラー(Error)

異常な要求処理の結果出されたエラー・メッセージのみがログに記録されます。

ログの中の特定の文字列を監視する際は、バージョンにより文字列が異なる場合がありますのでご注意ください。

Designing Web System Infrastructure with WAS V8.0

ヒープ・ダンプ、Java コア、システム・ダンプ

■ WAS V8より管理コンソールからヒープ・ダンプ、Javaコア、システム・ダンプを出力可能

New
v8

各ダンプファイルの詳細は「JVM」のセッションをご参照ください。

© 2012 IBM Corporation
51

WAS V8から、ヒープ・ダンプ、Java コア、システム・ダンプを管理コンソールから出力することが可能になりました。

Java関連の問題判別や製品サポート問い合わせ時のデータ収集を容易に行うことができます。

各ダンプファイルの詳細は「JVM」のセッションをご参照ください。

Solaris、HP-UXでは、下記リンクのように制限があります。

<http://www-01.ibm.com/support/docview.wss?uid=swg21503798>

Designing Web System Infrastructure with WAS V8.0

3. 定常運用

- 3-1 起動停止
- 3-2 ログ運用
- 3-3 自動化**
- 3-4 定常運用に関する設計ポイント

52

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

基本方針

- 自動化の対象
 - ◆ 定期的な処理の実行
 - ◆ 煩雑な処理の実行
 - ◆ 時間のかかる処理の実行
- 自動化による効果
 - ◆ 作業負荷の軽減
 - ◆ 作業ミスの軽減



ある特定の時間に

```

AppServ1停止
↓
AppServ1ログ・アーカイブ
↓
AppServ1開始
↓
AppServ2停止
↓
AppServ2ログ・アーカイブ
↓
AppServ2開始
          
```

© 2012 IBM Corporation
53

運用フェーズにおいては、管理コンソール以外の管理ツールが必要になります。理由としては、定期的な処理、煩雑な処理、時間のかかる処理の自動実行を行うためです。定期的な処理の例としては、毎週日曜深夜にアプリケーション・サーバーを停止して、ログを保管し、またサーバーを開始する処理、煩雑な処理の例としては、毎回特定のオプションを指定してのエンタープライズ・アプリケーションのインストールを行う処理、時間のかかる処理の例としては、ND環境でユーザーサービスを継続するために5つのアプリケーション・サーバーを1時間ごとに順に再起動する処理などが考えられます。

管理ツールを用いて自動実行を行うことにより、運用管理作業の作業負荷が軽減するだけでなく、作業ミスも軽減することができ、システムの信頼性の向上が期待できます。

Designing Web System Infrastructure with WAS V8.0

WASにおける処理の自動化

- wsadminスクリプトの作成
 - ◆ Information Center 「wsadmin スクリプト・オブジェクトの使用」
 - 各オブジェクト毎にサンプル・スクリプトを提供
 - Help、AdminApp、AdminControl、AdminConfig、AdminTask
 - http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/txml_configvsoper.html
 - ◆ スクリプト・ライブラリー
 - 管理操作のJythonサンプル・スクリプトを提供
 - http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/txml_7scrib.html
 - ◆ 管理コンソール操作のwsadminコマンド記録
 - 管理コンソールでの操作をJythonコマンドで出力
- ジョブ・マネージャーの使用
 - ◆ 複数のND/Base/Express環境にまたがった運用処理の非同期実行

© 2012 IBM Corporation
54

WASにおいて処理を自動化するためには、WASの管理ツールであるwsadminやコマンド行ツールを用いて処理をバッチファイル化します。

wsadminスクリプトを作成する際には、Information Center に記載されているサンプル・スクリプトやWASV7からの機能として提供されているスクリプト・ライブラリーを参考にします。また、WASV6.1から管理コンソール操作のコマンド記録機能が提供されており、管理コンソールで行った操作をJythonスクリプトのコマンドとして、表示または記録することができます。

また、既述の通り、ジョブ・マネージャーを使用することで、複数のND/Base/Express環境にまたがった管理アクションを統合させることができ、運用処理を各管理対象ノードに対して非同期に実行することができます。一般的には、WASに限らずさまざまな製品の多岐に渡る処理を自動化するためには、専用のジョブスケジューラー・ソフトウェアが使用されますが、テスト環境などジョブスケジューラー・ソフトウェアが導入されていない環境においては、有効に使用することができます。

Designing Web System Infrastructure with WAS V8.0

管理コンソール操作のコマンド記録

- コマンド支援
 - ◆ 管理コンソールでの操作をJythonコマンドで出力
 - 管理コンソール上に表示
 - ログに記録
 - 出力先<WAS_PROFILE_ROOT>/logs/dmgr/commandAssistanceJythonCommands_<loginユーザー名>.log
 - RADに通知

コマンド記録機能の詳細は「運用設計-参考-」のP.139-140をご参照ください。

© 2012 IBM Corporation
55

WASV6.1から、管理コンソールで行った操作をJythonスクリプトのコマンドとして、表示または記録することができるようになりました。管理コンソールに表示するには、操作を実施後、コンソール右側の「最後のアクションの管理スクリプトを表示」のリンクをクリックします。ポップアップで新規ウィンドウが表示され、最後に実行したwsadminコマンドがJython形式で表示されます。「保管」操作を実行した後に、最後の管理スクリプトを表示しても、「保管」操作が最後の操作になります。ポップアップ・ウィンドウを表示したまま、管理コンソール上での操作を行った場合には、自動的に最後に実行したコマンドが随時、画面上に反映されます。また、管理コンソール上に表示するだけでなく、ログに記録したり、RADに送信することが可能です。ログに記録することで、最後の操作だけではなく、一連の操作を記録することができるので大変便利です。ログに記録するには、管理スクリプトの表示画面で、「コマンド・アシスタンスのコマンドをログに記録する」のチェックボックスをONにします。ただし、アプリケーションのデプロイ操作やデータソースの設定変更など多くの操作をwsadminコマンドとして記録できますが、必ずしもすべての操作をwsadminコマンドに変換できるわけではありませんので、ご注意ください。

wsadminコマンドとして出力できる操作の一覧は下記のInformation Center の記載を参照ください。

WASV8.0 Information Center 「管理コンソールのコマンド支援を使用したアクション」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multipatform.doc/info/ae/ae/rcon_command_assist.html

コマンド記録機能の詳細に関しては、「運用設計-別紙-」のP.139-140をご参照ください。

Designing Web System Infrastructure with WAS V8.0

3. 定常運用

- 3-1 起動停止
- 3-2 ログ運用
- 3-3 自動化
- 3-4 定常運用に関する設計ポイント**

56

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WASV8.0定常運用に関する設計ポイント

- 起動停止方針
 - ◆ 起動停止方法、起動停止タイミング、全体の起動停止順序（外部システムも考慮）、デプロイメント・マネージャーの通常時状態
- ログ運用方針
 - ◆ ログ出力先、ログ容量見積もり（ログ・レベルを考慮）、メンテナンス対象、保管世代数、IHS/WASローテート機能使用可否、HPELの使用可否
- 自動化方針
 - ◆ 自動化の対象、自動化の方法、ジョブ・マネージャー使用可否

© 2012 IBM Corporation
57

ここで、WASV8.0の定常運用に関する設計ポイントをまとめます。

起動停止方針として、各コンポーネントの起動停止をどのように行うかを検討します。手動で起動や停止を行う場合に、管理コンソールから行うのか、WASのコマンドを使用するのか、あらかじめ作成されたスクリプトを実行するのか、といった起動停止方法を決めておきます。また、それぞれの起動停止（再起動）を行うタイミングについて、業務要件や環境制約などから、どの程度定期的の実施することができるのかを確認し、決めておきます。そして、製品や連携箇所を考慮し、システムにおける全てのコンポーネントの起動順序と停止順序を検討します。その際、システム内だけでなく、連携している外部システムの状態も考慮する必要があります。また、WASのND構成の場合、業務サービスには関係のないデプロイメント・マネージャーを通常時に起動しておくのか、あるいは通常時は停止しておき、構成変更時に起動する運用とするのか、などについても検討します。

ログ運用方針として、各ログ・ファイルをどのようにメンテナンスしていくかを検討します。各ログの出力先やログレベル、ロギングモードを決め、それぞれ定期的なメンテナンスが必要かどうかを判断し、対象を洗い出します。そして、お客様や業務要件から必要な保管世代数を明確にしておきます。また、各製品が提供しているローテート機能を使用するのか、使用する場合は何世代まで保管されるのかについても明確にしておきます。そして、OSの設計や監視項目にも影響を与えるため、ログの保管世代数やローテート機能などから事前に各ログの容量見積もりも行っておく必要があります。

自動化方針として、各運用処理をどのように自動化させるかを検討します。定期的な処理、煩雑な処理、時間のかかる処理、などについて洗い出し、自動化すべきかどうかを検討します。自動化にあたっては、WASのジョブ・マネージャーを使用するのか、あるいは別のジョブスケジューラー・ソフトウェアを使用するのかを検討します。

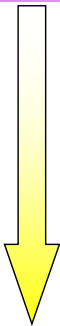
Designing Web System Infrastructure with WAS V8.0

Agenda

1. 運用設計概説
2. 構成管理
3. 定常運用
4. 非定期運用
5. マイグレーション

まとめ・参考文献

参考資料



© 2012 IBM Corporation

58

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

4-1 バックアップ・リストア

4-2 監視

4-3 障害時運用

4-4 片系運用

4-5 Fix Pack/Fix適用

4-6 アプリケーション管理・更新

4-7 リリース・パターン

4-8 非定期運用に関する設計ポイント

59

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

基本方針

- バックアップ取得のタイミング
 - ◆ インストール直後
 - 製品バイナリーも含めたバックアップ取得
 - ◆ 修正モジュール適用前後
 - ◆ 構成変更の前後
- リストアのタイミング
 - ◆ 不慮の問題発生によりシステムが動作しない状態になった場合
 - HW障害、人為的障害、自然災害、など

© 2012 IBM Corporation

60

障害など不慮の問題が発生した時に迅速な復旧が行えるように、普段からバックアップを取得しておくことが重要です。

バックアップ取得は、システム要件やお客様要件にもよりますが、一般的には、製品インストール直後や修正モジュール適用前後、構成変更の前後などに行います。製品バイナリーは、製品再インストールにより復元できるため対象外と考えて良いですが、修正モジュールの再適用や適用後の検証にかかる時間が大きくなることから、製品バイナリーや構成ファイルを併せてバックアップ取得しておく方法も考えられます。

リストアは、不慮の問題発生によりシステムが正常動作しない状態になった場合に行います。不慮の問題は、HW障害や人為的障害、自然災害など、予知不可能なさまざまな原因によって発生するため、想定されるケースに応じて事前にバックアップやリストアの対象を検討しておく必要があります。

Designing Web System Infrastructure with WAS V8.0

WASのバックアップ・リストア

- 最も確実な方法
 - ◆ <WAS_ROOT>ディレクトリー以下を丸ごとアーカイブ化
- WAS構成情報バックアップ・リストアのツール
 - ◆ backupConfig.sh(bat) / restoreConfig.sh(bat)
 - <RPROFILE_ROOT>/config ディレクトリー以下をアーカイブ/リストアするツール
 - バックアップ・ファイル(zip形式)の名称は指定可能
 - (デフォルト)<PROFILE_ROOT>/bin/WebSphereConfig_YYYY-MM-DD.zip
 - ND構成の場合はセル内の全ノード分の構成情報を保持しているDMプロファイル分のみ取得
 - 使用時の考慮点
 - WebSphereセキュリティ/SSL通信を構成している場合はetcディレクトリーのバックアップも取得
 - 各種プロパティー・ファイルを編集している場合はwsadminやffdcなどはプロパティー・ファイルにより動作のカスタマイズが可能なのでpropertiesディレクトリーのバックアップも取得
 - ◆ manageprofiles.sh(bat) -backupProfile / -restoreProfile
 - 対象は<Profile_root> ディレクトリー以下の全てのディレクトリーとファイル
 - バックアップ時には対象プロファイルに関連するプロセスを停止

© 2012 IBM Corporation
61

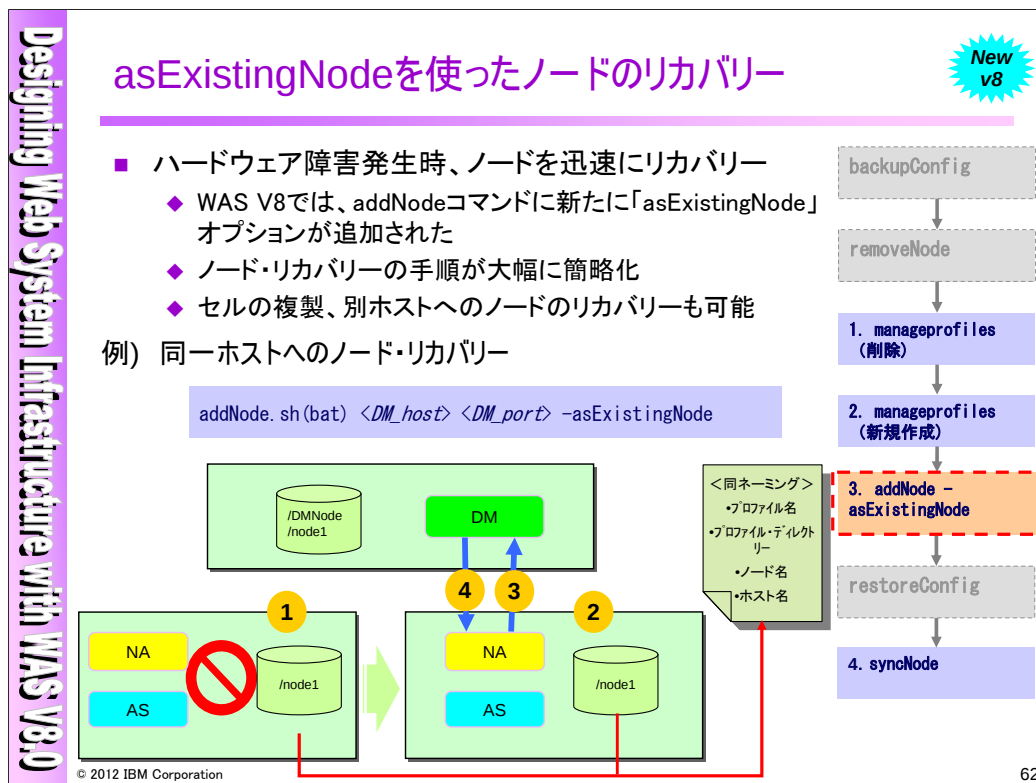
WASのバックアップ方法としては、<WAS_ROOT>ディレクトリー以下をまとめてバックアップ取得しておく方法が、最も確実な方法です。WASには、構成情報バックアップ・リストアのためのツールが提供されていますが、これらのツールでは一部のディレクトリー以下しかバックアップ取得されないため、構成によっては、アプリケーションのインストール先やログ出力先といった他のディレクトリーをバックアップ取得する仕組みが必要となります。

WASの構成情報のバックアップ取得ツールとして、backupConfigコマンドが準備されています。バックアップ・リストアのためのコマンドとしては、backupConfig.sh(bat)、restoreConfig.sh(bat)の2種類のコマンドがあります。このコマンドでは、バックアップ・リストアの対象が<Profile_root>/configディレクトリー以下に限られるため、プロパティー・ファイルを変更している場合やSSLの鍵情報などは、必要に応じて手動で別途バックアップを取得する必要があります。

WASV6.1からは、wasprofileコマンドの拡張版としてmanageprofilesコマンドが提供されており、以下の2つのオプションがあります。

- -backupProfile
- -restoreProfile

manageprofilesコマンドのプロファイルのバックアップ・リストアオプションは、<Profile_root>ディレクトリー以下の全てのディレクトリーとファイルを対象としています。そのため、バックアップ取得作業を短縮する事が出来るようになりました。但し、backupConfig.sh(bat)やrestoreConfig.sh(bat)がプロセスを止めずに行えるオプション(-nostop)を備えているのに対して、プロファイルのバックアップ・リストアは必ず関連するプロセスを停止しなければなりません。状況に応じて、それぞれのコマンド、オプションを使い分ける事が推奨されます。



WAS V8では、addNodeコマンドに新たに「asExistingNode」オプションが追加されました。ノード障害が発生した際、デプロイメント・マネージャーから障害ノードを除去(removeNodeコマンド)、バックアップからデプロイメント・マネージャーの構成情報をリストア(restoreConfigコマンド)といった作業は不要となり、以下の手順でノード・リカバリーを行うことができます。(前提: ノード構成情報はデプロイメント・マネージャーに保存されている)

1. 障害ノードのプロファイルを削除
2. 新規プロファイルを作成
3. 新規プロファイル配下の「bin」ディレクトリで、addNodeコマンドを「asExistingNode」オプション付きで発行し、新規ノードをデプロイメント・マネージャー管理配下に統合
4. ノードの同期化(syncNodeコマンド)

セルの複製や別ホストへのノードのリカバリーも可能です。これらの手順詳細は「運用設計-参考-」のP.142-144をご参照ください。

Designing Web System Infrastructure with WAS V8.0

IHS/プラグイン/IIMのバックアップ・リストア

- IHS/プラグインのバックアップ・リストア
 - ◆ 最も確実な方法
 - インストール・ディレクトリー以下を丸ごとアーカイブ化
 - ◆ バックアップ取得方法
 - 各構成ファイルをコピーして保管
 - ◆ バックアップ取得ファイル
 - httpd.confファイル
 - admin.confファイル(リモートWebサーバー構成時のみ)
 - plugin-cfg.xmlファイル
 - 静的コンテンツ
 - SSL構成時には以下のファイルも取得
 - 鍵データベース・ファイル
 - パスワード・ファイル
- IIMのバックアップ・リストア
 - WAS/IHS等リストア時の、WAS/IHS等とIIMのFix適用情報の不整合を防ぐため
 - インストール・ディレクトリー以下を丸ごとアーカイブ化

© 2012 IBM Corporation
63

IHSとWebサーバー・プラグインのバックアップ方法としては、インストール・ディレクトリー以下をまとめてバックアップ取得しておく方法が、最も確実な方法です。個別にファイルをバックアップする場合には、IHSとプラグインそれぞれの構成情報が格納された構成ファイルをコピーして保管しておきます。IHS構成ファイルとしてhttpd.conf、リモートWebサーバー構成時はIHS管理サーバーの構成ファイルとしてadmin.conf、プラグイン構成ファイルとしてplugin-cfg.xmlをバックアップ取得しておきます。静的コンテンツなどのインストール後に修正を行ったファイルをそれぞれバックアップを取得します。また、SSL構成時は、鍵データベースファイルやパスワードファイルもバックアップ取得の対象に入れます。

またIIMのバックアップについても考慮する必要があります。V7まではUpdateInstallerがWASインストール配下に適用前の情報を格納していましたが、V8よりFixの適用情報はIIMのインストール・ディレクトリー以下に格納されるようになりました。これにより、FixPack適用後にWASやIHSのみバックアップファイルからリストアすると、IIMで管理しているFixの適用情報とWASやIHSに実際に適用されているFixの適用情報が不整合となってしまうため、IIMについてもバックアップを取得しておき、WASやIHSをバックアップファイルからリストアする際には、IIMもバックアップファイルからリストアすることが必要となります。IIMのバックアップ対象につきましては、IIMのインストール・ディレクトリー以下を丸ごとバックアップして下さい。

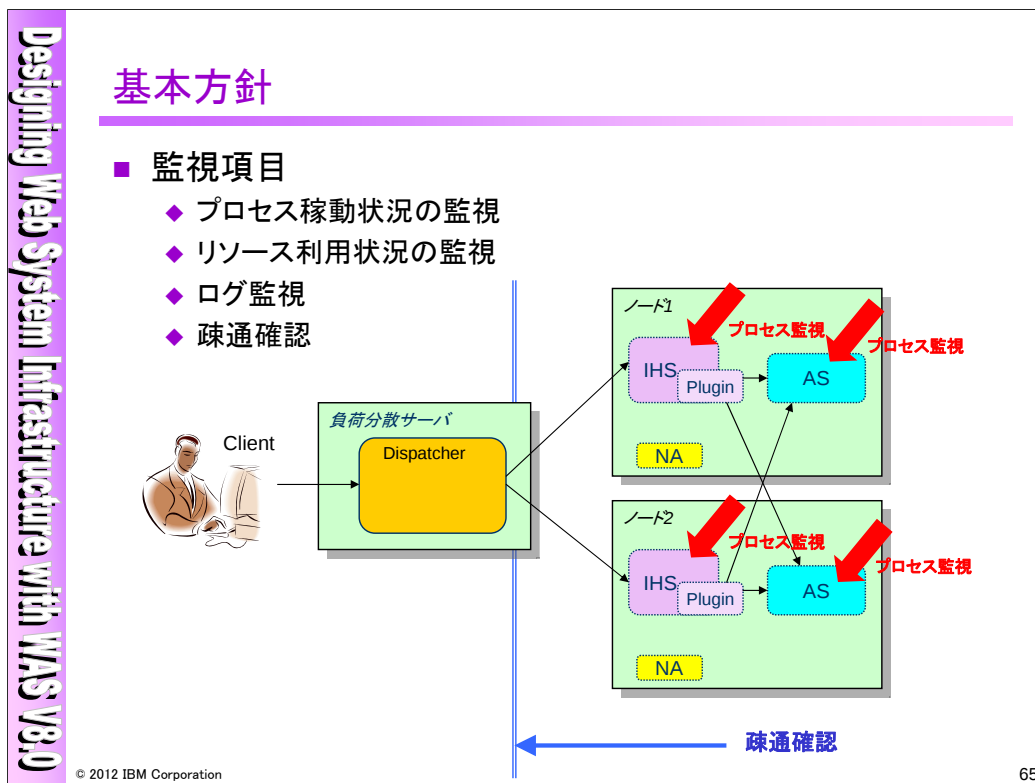
Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視**
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

64

© 2012 IBM Corporation



一般的に、Webシステムの監視項目としては、プロセス稼動状況の監視、リソース利用状況の監視、ログ監視、疎通確認、が挙げられます。プロセス監視だけでは、必ずしもユーザー・サービスの提供状況を把握することができませんので、定期的な疎通確認も重要です。また、監視を行うだけでなく、監視により障害を検知した場合の対応方法を事前に検討しておくことが重要です。

この章では、プロセス稼動状況の監視とログ監視、疎通確認について説明します。
リソース利用状況の監視については、「パフォーマンス設計」のセッションをご参照ください。

Designing Web System Infrastructure with WAS V8.0

プロセス稼動状況の監視

■ 監視プロセスと監視方法

コンポーネント	IHS	WAS(アプリケーション・サーバー)	WAS(ノード・エージェント)	WAS(デプロイメント・マネージャー)
監視プロセス	httpd	java	java	java
監視方法	①psコマンド (Win:tasklistコマンド)	①psコマンド (Win:tasklistコマンド) ②serverStatusコマンド ③ノード・エージェントによる監視	①psコマンド (Win:tasklistコマンド) ②serverStatusコマンド	①psコマンド (Win:tasklistコマンド) ②serverStatusコマンド
備考	Windows2000ではtasklistコマンドを使用できないのでnetstatコマンドでListenしているポートの確認などで代替(WASの場合も同様)	ノード・エージェントによる監視は自動的にプロセスの起動を行うがアラートを上げるものではないので注意		ユーザー・サービスの提供のために必要プロセスではないので通常時の起動や監視は必ずしも必要なし

serverStatusコマンドは別途javaプロセスが起動して結果を出力→負荷が比較的高いので注意

プロセス監視の詳細は「運用設計-参考-」のP.146-148をご参照ください。

© 2012 IBM Corporation
66

上記は、一般的なIHS/WASの監視対象プロセスおよび監視方法をまとめたものです。IHSはhttpdプロセス(親プロセスのみ)、WASの各コンポーネントはjavaプロセスの監視を行います。一般的には、OSのpsコマンドを使用して、プロセス名やアプリケーション・サーバー名をキーにして、出力を確認します。WASが提供するコマンド行ツールのserverStatusコマンドでもプロセスの稼動状況を確認することができますが、serverStatusコマンドを実行した場合には、別途javaプロセスが起動し、psコマンドの実行と比較して負荷が高くなりますので、ご注意ください。また、Windows2003以前のサーバーでは、tasklistコマンドが使用できませんので、netstatコマンドでListenしているポート番号(IHSであれば通常80番、WASのデフォルトは9080番)を監視することなどで代替できます。HAマネージャーによる監視は、シングルトン・サービスのフェイルオーバーなどのために使用され、直接アラートを上げることはできません。

プロセス監視の詳細に関しては、「運用設計-別紙-」のP.146-148をご参照ください。

Designing Web System Infrastructure with WAS V8.0

ログ監視

- WAS
 - ◆ 監視対象ログ・ファイル : SystemOut.log、SystemErr.log
 - ◆ 監視対象メッセージ(例)
 - 事例1 : 一定以上のレベル(イベントタイプFおよびEなど)のメッセージ
 - 事例2 : Exception/Errorの文字列を含むメッセージ
 - アプリケーション・ログはSystemOut.log/SystemErr.logに記録しないこと
 - ◆ 監視対象ログ・ファイル(HPEL) : logdata /*.wbl、TextLog_<timestamp>.log(テキスト)
 - HPELモードの場合 : logviewer -monitorオプションでリアルタイム監視が可能
- IHS
 - ◆ 監視対象ログ・ファイル : error_log
 - ◆ 監視対象メッセージ(例)
 - 事例1 : 一定以上のレベル([error][crit][alert][emrg]など)のメッセージ
 - 事例2 : 500番台のエラー(認証成否にかかわる403エラー等は別途月次バッチで集計するなど)
 - 404 エラーの扱い
 - ユーザ側のURI 指定誤りが原因であるがログレベルとしては [error]
 - "File does not exist*"という文字列は監視対象外とするなど工夫が必要
- プラグイン
 - ◆ 監視対象ログ・ファイル : http_plugin.log
 - ◆ 監視対象メッセージ(例)
 - 事例1 : 全てのエラーメッセージ(IHS起動・停止時の出力メッセージ以外)
 - 事例2 : "ERROR"の文字列を含むメッセージ

WASのログ・フォーマットの詳細は「運用設計-参考-」のP.149をご参照ください。

© 2012 IBM Corporation
67

WASのシステムが出力するログを監視する場合は、SystemOut.logやSystemErr.logを監視対象とすることが一般的です。SystemOut.logやSystemErr.logのエラー・レベルがFatalやErrorのものだけを抽出したり、Warning以上を全て監視したりする場合もあります。アプリケーション・ログはSystemOut.logやSystemErr.logには記録しないようにすることで、WASとアプリケーションでログを完全に分けることができるため、問題判別が効率的になります。HPELモードを利用している場合はバイナリー形式でログの書き出しが行われるので、logviewerのmonitorオプションを利用する必要があります。

IHSの監視対象とするログは、エラー・ログです。ログレベルが一定以上のもの、または、リターンコードが500番台のサーバーエラーに関するものを監視する方法などが考えられます。

プラグインログは、全てのエラーメッセージを監視の対象としたり、"ERROR"の文字列を含むメッセージを監視の対象とする方法が考えられます。

WASのログ・フォーマットの詳細に関しては、「運用設計-別紙-」のP.149をご参照ください。

Designing Web System Infrastructure with WAS V8.0

疎通確認

- WAS構成と疎通確認方法

構成	1:1構成	1:N構成
疎通確認方法	各IHSに対して検査リクエストを行う	各IHSだけではなく各WASサーバーに対しても検査リクエストが必要
理由	IHS:WASが1:1のため、IHSに対するリクエストで、IHS-WASの組み合わせで正常性を確認できる	IHSは、正常なWASに対してリクエストを割り振るため、1台のWASがダウンしていてもIHSへの検査だけでは、ダウンしているWASを検知できない

監視サーバー
監視ツール

ノード1
IHS Plugin AS
NA

ノード2
IHS Plugin AS
NA

監視ツールとしては、外部のツールまたはEdgeコンポーネントのAdvisor機能を使用可能

各構成における疎通確認の詳細は「運用設計-参考-」のP.150-151をご参照ください。

© 2012 IBM Corporation

68

IHS/WASを合わせての疎通確認を行う場合には、IHSとWASの構成が1:1構成か1:N構成をとっているかによって、監視の考慮点が変わってきます。1:1構成の場合、各IHSに対してWAS上で稼動する動的ページのリクエストを行うことで、関連付けられているアプリケーション・サーバーの稼動状況まで含めて監視することができます。1:N構成の場合は、IHSとWASとの関連付けが1:1ではありませんので、IHSとWASとをそれぞれ独自に監視する必要があります。

各構成における疎通確認の詳細に関しては、「運用設計-別紙-」のP.150-151をご参照ください。

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用**
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

69

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

障害発生時の対応

- ① 障害が発生した系の切り離し
 - ◆ 自動での切り離しが望ましい
 - ◆ 自動で切り離した場合も、障害の発生は検知する必要がある

- ② 障害情報の収集
 - ◆ ログの収集・確認
 - ◆ IBMパスポート・アドバンテージ・テクニカル・サポートを利用する場合
 - 問題判別を行うために、その環境の情報を収集
 - WAS コレクター・ツール <全般的な情報収集>
 - WAS MustGather <コンポーネント毎の問題情報収集>
 - OSレベルの障害情報
 - ISA (Lite)

- ③ 問題判別・対応
 - ◆ ②の情報を元に分析を行う

© 2012 IBM Corporation
70

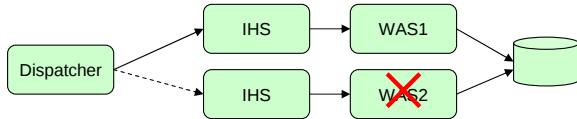
障害が発生した場合には、上記のような作業が必要になります。まず、障害発生時には、冗長化構成をとっている場合には、障害が発生した系の切り離しが必要です。障害が発生した系の切り離しは、自動での切り離しを行うことにより、ユーザー・サービスをほとんど停止することなく、サービスを継続することが可能になります。また、自動で切り離しを行った場合でも、早期の障害検知が重要です。

障害情報の収集や問題判別の詳細については、「問題判別」のセッションをご参照ください。

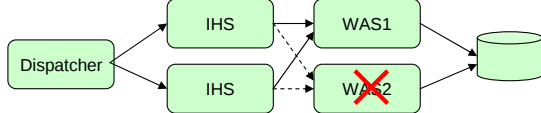
Designing Web System Infrastructure with WAS V8.0

障害が発生した系の切り離し

- IHS:WAS=1:1構成の場合
 - ◆ Dispatcherにより障害検知時に自動切り離し



- IHS:WAS=1:N構成の場合
 - ◆ IHSの障害時
 - Dispatcherにより自動的に切り離される
 - ◆ WAS(アプリケーション・サーバー)の障害時
 - WASプロセス障害の場合
 - Webサーバー・プラグインにより自動的に切り離し
 - WASのハングやOS/Network障害の場合
 - Webサーバー・プラグインでは検知が困難、検知が遅延
 - ✓ プラグインの最大接続数の設定やタイムアウト設定が必要
 - 障害検知時にはプラグイン構成ファイルを置き換えるなどの実装が必要



71

障害が発生した場合の系の切り離しの動きについて説明します。

IHSとWASが1:1構成の場合は、DispatcherのAdvisor機能によって、自動的に切り離されます。ただし、DispatcherのAdvisor機能は、定期的に監視リクエストを行いますので、切り離すまでには、1分程度の時間を要します。

IHSとWASが1:N構成の場合は、障害コンポーネントや、障害の状況によって、切り離しにかかる時間や障害として検知するまでの時間が異なります。IHSに障害が発生した場合には、1:1構成のときと同様にDispatcherが障害検知、切り離しを行います。WASに障害が発生した場合には、障害状況がWASのプロセス・ダウンなのか、WASのハングやNetwork障害などの状況によって異なります。WASのプロセス・ダウンの場合は、プラグインが障害を即座に検知し、エンド・ユーザーにエラーは返りません。WASのハングや、OS障害/Network障害の場合には、プラグインの監視では、検知できなかったり、検知が遅くなったりします。そのような状況をできるだけ避けるため、外部ツールを使用した監視や、プラグイン設定の最大接続数やタイムアウト設定を適切に行うことが重要です。

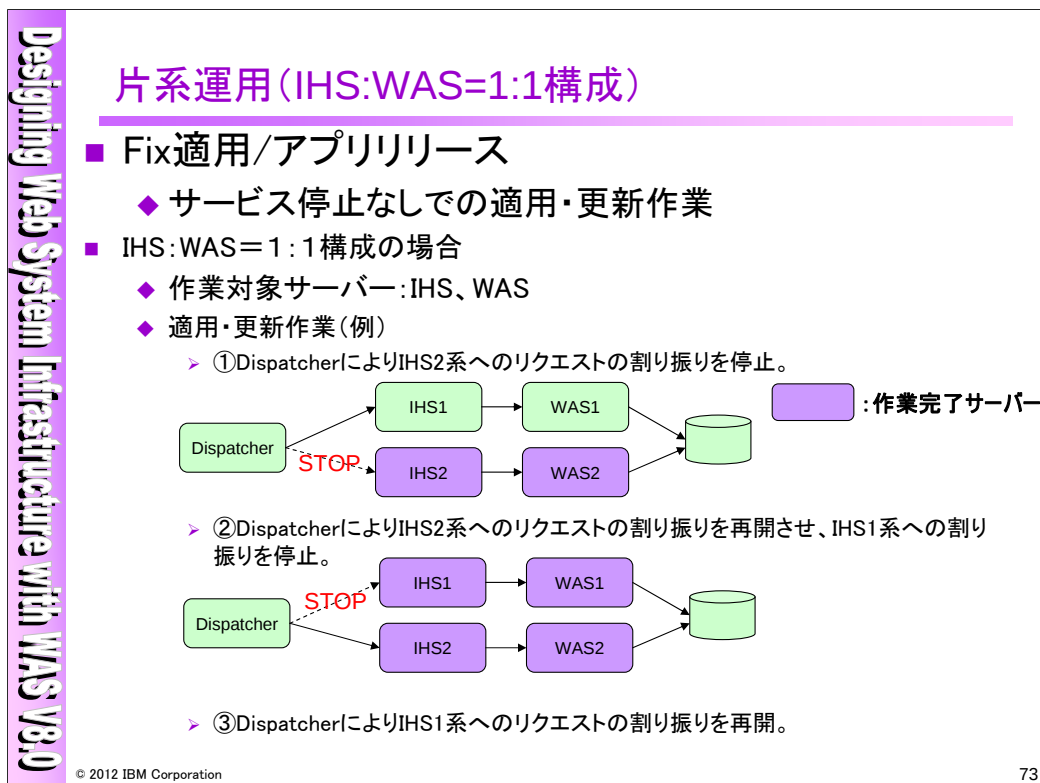
Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用**
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

72

© 2012 IBM Corporation



Fix適用やアプリリリースを行う際、IHS/WASのプロセスを停止する必要がある場合があります。このとき、サービスを停止させずに作業を行いたい場合、冗長化構成をとっていれば、片系運用を実施することが可能です。

実施手順はIHS:WASの構成や作業対象サーバによって異なります。

IHS:WAS=1:1構成の場合の実施手順の一つの例が下記となります。Dispatcherにより、ユーザーからのリクエストを作業対象のサーバを含む系に割り振らないように制御し、順次適用・更新作業を行っていく必要があります。

また作業時には以下の点にご注意ください。

- ・割り振り停止後、作業対象サーバのユーザーからのリクエストに対するプロセスが完了したことを確認してから作業を実施するようご注意ください。
- ・新旧のバージョンが混在してしまう時間帯が生じてしまいますのでご注意ください。
- ・別系に割り振れるとセッションがきれるので、必要に応じてセッション・パーススタンスをご検討ください。

Designing Web System Infrastructure with WAS V8.0

片系運用(IHS:WAS=1:N構成)

- IHS:WAS=1:N構成の場合
 - ◆ 作業対象サーバー:IHS
 - ◆ 適用・更新作業
 - IHS:WAS=1:1構成と同様
 - ◆ 作業対象サーバー:WAS(アプリケーション・サーバー)
 - ◆ 適用・更新作業(例)
 - ①IHS1系のプラグイン設定にてIHS1系からWAS1系へのリクエストの割り振りを停止。
 - ②IHS2系のプラグイン設定にてIHS2系からWAS1系へのリクエストの割り振りを停止。

- ③IHS1系のプラグイン設定にてIHS1系からWAS1系へのリクエストの割り振りを再開し、WAS2系への割り振りを停止。
- ④IHS2系のプラグイン設定にてIHS2系からWAS1系へのリクエストの割り振りを再開し、WAS2系への割り振りを停止。

- ⑤IHS1系のプラグイン設定にてIHS1系からWAS2系へのリクエストの割り振りを再開する。
- ⑥IHS2系のプラグイン設定にてIHS2系からWAS2系へのリクエストの割り振りを再開する。

© 2012 IBM Corporation
74

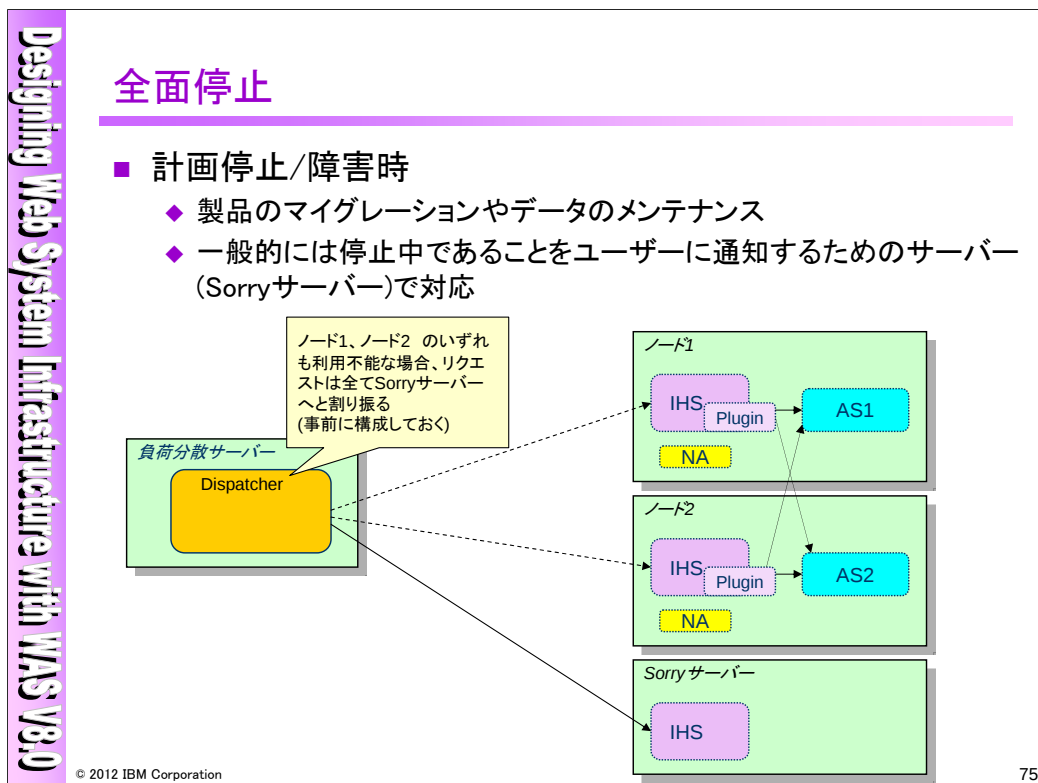
IHS:WAS=1:N構成の場合、作業対象サーバによって手順が異なってきます。

作業対象サーバがIHSの場合の手順は、IHS:WAS=1:1構成の場合と同様となります。

作業対象サーバがWASの場合の実施手順の一つの例が下記となります。IHS:WAS=1:1構成の場合、Dispatcherで割り振りの制御ができませんので、IHSのプラグインでの割り振りの制御を行う必要があります。plugin-cfg.xmlファイルの編集による割り振り制御は、全てのIHSノードでそれぞれ作業を行う必要がありますので筐体数が多い場合は運用負荷が高くなります。あらかじめ複数のplugin-cfg.xmlを別名で用意しておき、シェルである程度自動化しておけば編集ミスなどの問題が発生しにくくなります。

また作業時には以下の点にご注意ください。

- ・割り振り停止後、作業対象サーバのユーザーからのリクエストに対するプロセスが完了したことを確認してから作業を実施するようご注意ください。
- ・新旧のバージョンが混在してしまう時間帯が生じてしまいますのでご注意ください。
- ・別系に割り振れるとセッションがきれるので、必要に応じてセッション・パーシスタンスをご検討ください。



製品のマイグレーションやデータのメンテナンスを行うために、あるいは障害時に、全サーバーを停止する必要がある場合があります。このような全面停止のプランを事前に検討しておくことが重要です。計画停止時にも、ユーザーに不明なエラーページを返すのではなく、ユーザーにシステムが停止中であることを通知するためのサーバー (Sorryサーバー) を準備しておくことにより、ユーザーに適切なエラーページを返すことができます。

Sorryサーバーとは、障害時/計画停止時にユーザーにメンテナンス画面を表示するためのサーバー (Webサーバー) で、どのリクエストに対しても同一コンテンツ (Sorryページ) を返すよう構成したり、状況に応じて、以下のように返すコンテンツを差し替えることもできます。

- ・【計画停止用】メンテナンス中です。明日08時よりサービスを再開いたします。
- ・【障害用】只今問題が発生しております。時間をあけてアクセスしてください。

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用**
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

76

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WASV8.0 Update Strategy

- Release (WASV8.0) ※PTFではなくバージョンアップ
 - ◆ 主要機能の変更・追加
 - ◆ 前リリースからのマイグレーション・ツールを提供
- Feature Pack
 - ◆ 新しい機能追加
 - 2012年1月31日現在「Feature Pack for Web 2.0 and Mobile V1.1.0.1」提供
 - ◆ 適用ツールとしてIBM Installation Managerを使用
- Fix Pack
 - ◆ 累積Fix
 - ◆ 各コンポーネントごとに提供
 - ◆ 適用ツールとしてIBM Installation Managerを使用
- iFix(暫定修正)
 - ◆ 個別Fix
 - ◆ 適用ツールとしてIBM Installation Managerを使用
- Feature Pack/Fix Pack のダウンロード先およびリリース予定日はWeb上で公開
 - ◆ <http://www.ibm.com/support/docview.wss?rs=180&uid=swg27004980>

「Feature Pack for Web 2.0 and Mobile V1.1.0.1」の詳細は「運用設計-参考-」のP.153をご参照ください。

© 2012 IBM Corporation
77

WASV8.0の修正プログラムや機能追加などのUpdate Strategyは上記のようになります。

WASV6.1から、大きな機能追加はFeature Packという形で、プログラム修正とは切り離されて行われるようになっていきます。なお、現在は、「Feature Pack for Web 2.0 and Mobile V1.1.0.1」のFeature Packが提供されています。

WASの一般的な運用方法としては、個別のFixを適宜適用するのではなく、Fix Packという形で個別Fixをまとめたものが提供されますので、Fix Packを適宜適用していく運用がお勧めです。Fix Packはおおよそ、3ヶ月に1回くらいの頻度で定期的に提供されます。非常に緊急の障害などの場合には、個別Fixを適用します。Fix Packは、WAS、IHS、Plugin、Application Client、Java SDKの各コンポーネントごとに提供されます。Java SDKのFix Packの適用もWASのFix Packの適用とは別途行う必要がありますので、ご注意ください。

WASのFix Packは、少なくともサービスイン前は、できるだけ最新のFix Packを適用されることをお勧めいたします。また、サービスイン後も、必ずしも毎回でなくとも、Fix Packを適用することを計画されることをお勧めします。また、WAS、IHS、Pluginなどは、できる限り同じFix Packレベルをご使用になることをお勧めします。(必ずしも同じでなければならないというわけではありません。)

「Feature Pack for Web 2.0 and Mobile V1.1.0.1」の詳細に関しては、「運用設計-別紙-」のP.153をご参照ください。

Designing Web System Infrastructure with WAS V8.0

Fix Pack / Fix – インストール

- Fix Pack / Fixのインストール
 - ◆ Step0. rootまたは管理者ユーザーでOSにログオン
 - ◆ Step1. バックアップの取得
 - マニュアル上はbackupConfig.sh(bat)を使用
 - <WAS_ROOT> 以下をフルバックアップする方法が確実でお奨め
 - ◆ Step2. ディスク空き容量の確保
 - ◆ Step3. IHS/WAS関連プロセス停止
 - ◆ Step4. IBM Installation Managerの起動
 - ◆ Step5. 修正モジュールのダウンロード・解凍
 - ◆ Step6. リポジトリの追加

- 解凍したFix Pack内のrepository.configを指定

}

ローカルのリポジトリを使用する場合のみ
 - ◆ Step7. IBM Installation Manager起動画面にて、「更新」を選択
 - ◆ Step8. Install対象の製品・フィーチャーを選択
 - ◆ Step9. Installation完了

© 2012 IBM Corporation
78

WAS V8より、すべてのWAS製品のインストールは単一なツールであるIBM Installation Manager (IIM) によって行うようになりました。

Fix Pack/FixについてもIIMよりインストールします。Fix Pack/Fix適用にはGUIとサイレント・モードを使用して行う方法がありますがここではGUIを使用した際の流れになります。

V7までは製品メディアによるインストール作業が必要でしたが、V8からはリポジトリから製品パッケージをダウンロードすることが可能となりました。リポジトリはメディアだけでなく、IBM Passport Advantage サイトからダウンロードすることやWebベースのリポジトリを使用することも可能になっています。

Install対象の製品は一度に複数選択することが可能です。WebSphere Application Server Network Deployment (WAS ND)を導入されているお客様は、DeploymentManagerのFixレベルがAppServerより高く(新しく)なければなりません。WAS NDを導入されているお客様で、製品ごとにインストールを行う場合は、まず先にFixをDeploymentManagerへ適用し、その後、AppServerへ適用をお願いします。

Designing Web System Infrastructure with WAS V8.0

Fix Pack / Fix – 適用確認とアンインストール

- WASのバージョン確認方法
 - ◆ コマンドから確認する方法
 - ./versionInfo.bat(sh)
 - ◆ ファイルから確認する方法
 - インストール済みの製品一覧
 - <IM_DATA_ROOT>/installed.xml
 - インストール履歴
 - <IM_DATA_ROOT>/histories/製品名/history.xml
- IHSのバージョン確認方法
 - <IHS_ROOT>%bin%versionInfo.bat(sh)
- Pluginのバージョン確認方法
 - <Plugin_ROOT>%bin%versionInfo.bat(sh)
- アンインストール
 - ◆ 作業手順はインストール時と同様
 - IBM Installation Managerにて、「ロールバック」を選択
 - ◆ モジュールのアンインストール順序に注意
 - インストールと逆の順序でアンインストール
 - Fix → Fix pack の順

© 2012 IBM Corporation
79

V7まではFix Pack適用の際にUpdateInstallerを使用していたため、<WAS_INSTALL_ROOT>/logs/update/<package_name>/updateLogディレクトリーにログが出力され“INSTCONFSUCCESS”などの出力によって、適用成否を確認できました。しかし、V8からはFix Pack適用の際にInstallation Managerを使用するようになったため、Fix Pack適用の際のログはWAS側に出力されず全てInstallation Manager側に出力されるようになっています。インストールしたFix Packが適用され、“インストール済み製品”の“バージョン”が更新されていれば正しくインストールされています。

一度適用したFix Packは、アンインストールすることができます。WAS V8からはロールバックという手順を踏むことで以前のバージョンに戻すことができます。

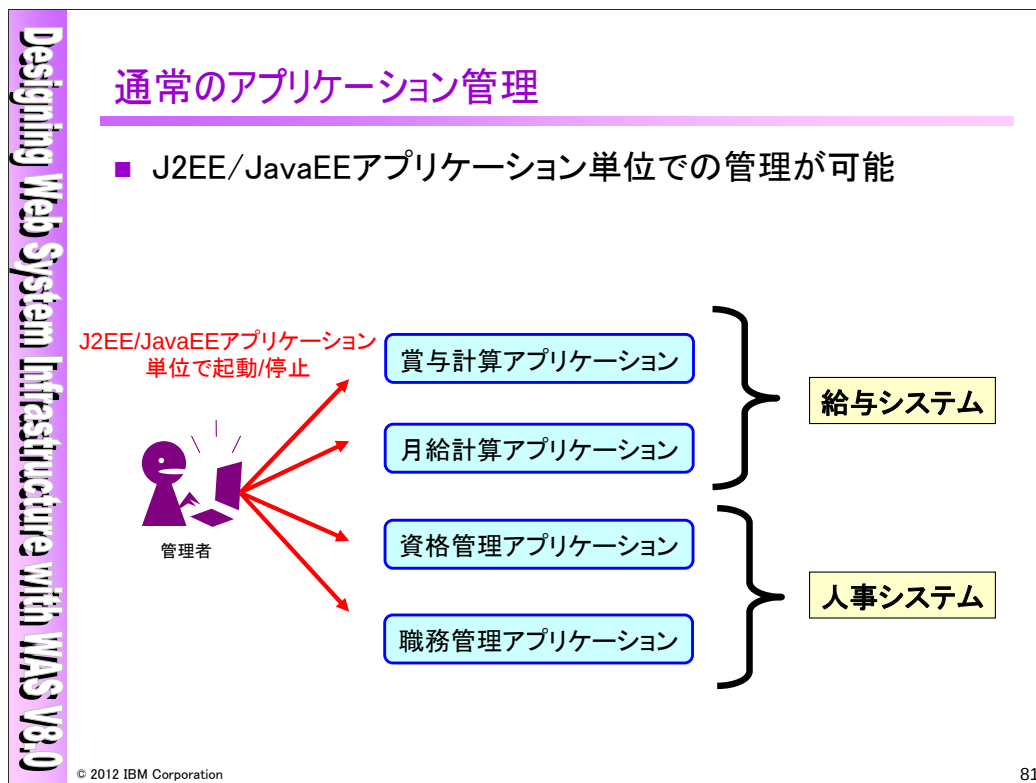
Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新**
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

80

© 2012 IBM Corporation

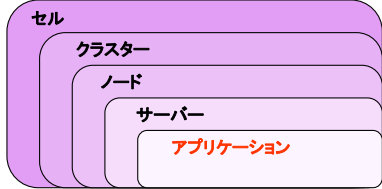


通常のJ2EE/JavaEE単位でのアプリケーションの操作方法は、従来のWASのバージョンの場合、各アプリケーション・モジュール単位での管理となっております。そのため、同一システム内に複数のアプリケーションが存在している場合、同一システムであっても、それぞれのアプリケーションを操作する必要があります。

Designing Web System Infrastructure with WAS V8.0

Enhanced EAR

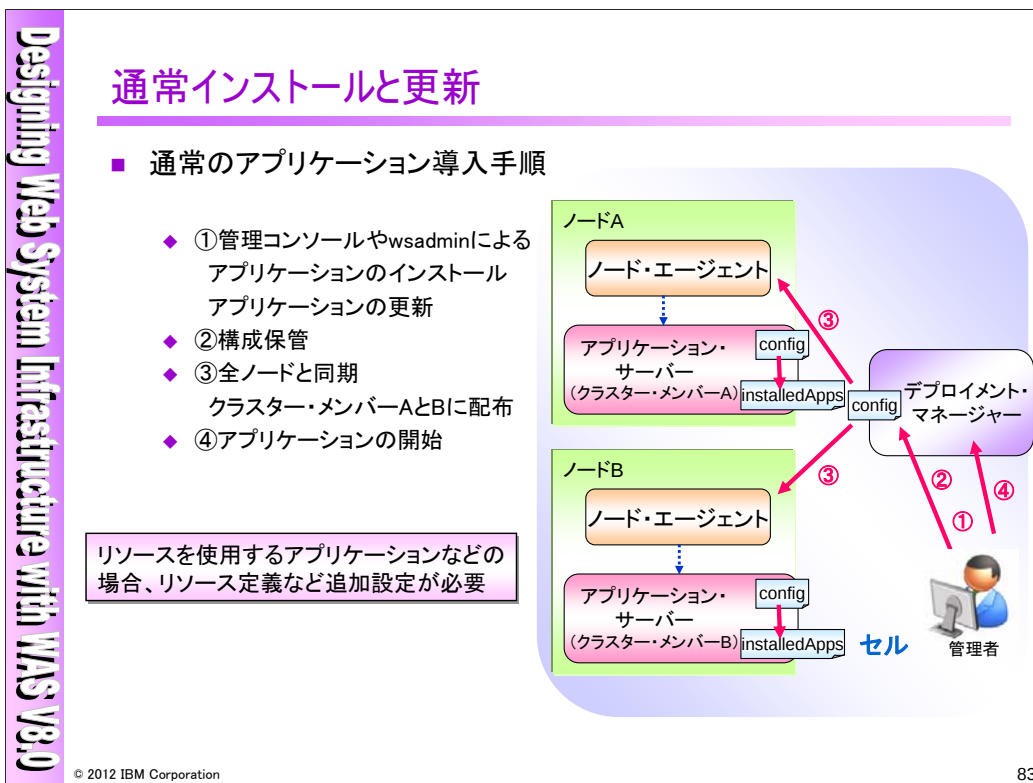
- Enhanced EAR とは
 - ◆ アプリケーション・レベルのリソース情報を含んだEARファイル
 - アプリケーション・レベルでの定義は最優先
 - 注意 ➢ アプリケーションより大きいスコープでのリソース定義変更には注意が必要
 - ◆ RADの機能。設定はRADのEnhanced EAR エディターを使用。
- 以下リソースの定義が可能
 - ◆ JDBCプロバイダー、データ・ソース、JMSリソース、リソース・アダプター（アプリケーション・レベル）
 - ◆ 置換変数（WebSphere変数）（アプリケーション・レベル）
 - ◆ クラスローダー・ポリシー（アプリケーション・レベル）
 - ◆ 共用ライブラリー（サーバー・レベル）
 - ◆ JAAS認証エイリアス（セル・レベル）
 - ◆ 仮想ホスト（セル・レベル）



リソース定義のオーバーライド階層

© 2012 IBM Corporation
82

Enhanced EARは EARファイルにデータソースなどのリソース設定を含めてパッケージングしたものです。WASのリソース設定は、通常、WASの管理コンソールやwsadminを使用して、アプリケーションの導入前に設定しておく必要がありますが、Enhanced EARをデプロイすると、アプリケーション・パッケージにリソース情報などが含まれますので、デプロイと同時にリソース定義も行われるという形になります。Enhanced EARの設定はRAD (Rational Application Developer) のEnhanced EAR エディターを使用します。Enhanced EARで設定したリソース定義のレベルはアプリケーション・レベルとなり、管理コンソール上で定義するクラスターやサーバーといったスコープのリソース定義よりも優先されます。このため、サーバーの設定を変更した際に、Enhanced EARに含まれるリソース情報が適用されてしまい、意図した設定と異なってしまうケースがあるため、使用する際には注意が必要です。



ここでは、通常アプリケーションをインストール・更新するときの、手順が説明されています。通常であれば、単体アプリケーション・サーバー又はデプロイメント・マネージャーの管理コンソール・wsadmin ツール経由でアプリケーションをインストール・更新して、ノードのinstalledAppsディレクトリーにアプリケーションのファイルが展開されます。

Designing Web System Infrastructure with WAS V8.0

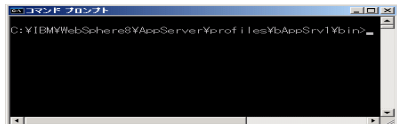
プロパティ・ファイルベース構成ツール(PFBCT)

V7~

- プロパティ・ファイルを使ってアプリケーションのインストール/更新/アンインストールができるようになった

```
#
# Header
#
ResourceType=Application
ImplementingResourceType=Application
CreateDeleteCommandProperties=true
ResourceId=Deployment

# Properties
Name=PerfUtil
#TargetServer={serverName}
#TargetNode={nodeName}
EarFileLocation=
C:/work/Application/PerfUtil.ear
TargetCluster={clusterName}
#
EnvironmentVariablesSection
#
#Environment Variables
cellName=Cell01
nodeName=Node03
serverName=Member1
clusterName=Cluster1
```



```
wsadmin.bat -lang jython

AdminTask.applyConfigProperties(['-
propertiesFileName
C:/work/Application/deployApplication.props -
reportFileName C:/work/Application/report.txt '])

AdminConfig.save()
```

インストールするアプリケーション・ファイルのロケーション

→

デプロイ対象のサーバー、クラスター情報を入力

↓

デプロイ又は更新の反映

アプリケーション・サーバー

注意 ノードとの同期、サーバーの再起動又はアプリケーションの再起動を行う必要がある

© 2012 IBM Corporation

84

ここでは、PFBCTを使ってアプリケーションをインストール・更新・アンインストールする機能(WAS V7.0 FP 9から追加された)をご説明します。

PFBCTとは、バイナリー・ファイル形式で保管されるWASの各種構成情報から、テキスト形式な単純な構成ファイルを抜き出し、設定を変更した構成ファイルを適用したりできるwsadminのコマンド群です。

既存のアプリケーションの場合、アプリケーションのプロパティ・ファイルをextractConfigPropertiesコマンドで抜き出すことができますが、アプリケーションの新規インストールの場合、構成情報が存在しないため、管理コンソールからプロパティ・ファイルのテンプレートをコピーして、編集して使います。

スライドの左側にあるのがアプリケーションをインストールするためのプロパティファイルの例であり、アプリケーションの名前、EARファイルの保存先、インストール対象のサーバー名もしくはクラスター名などを指定します。準備できたプロパティ・ファイルの保存先を指定して、applyConfigPropertiesコマンドで設定を適用します。

PFBCTでアプリケーションをインストールする場合、wsadminのinstallコマンドと同じく変更を保存して、セル環境においてノードとの同期処理を行い、アプリケーションを開始するという追加作業が必要です。

Designing Web System Infrastructure with WAS V8.0

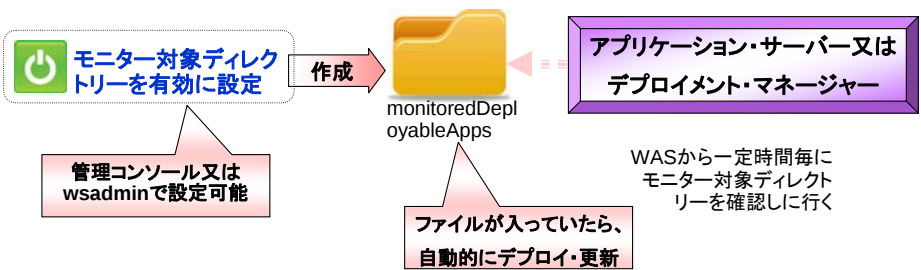


モニター対象ディレクトリー（1/4）

■ モニター対象ディレクトリーによる自動デプロイ

- ◆ 指定したディレクトリーにEARなどのアーカイブ・ファイルをドラッグ・アンド・ドロップすると、変更を認識してWASに自動的にインストール・更新される
- ◆ デフォルト: 無効 → 使用するには有効に設定する必要がある
- ◆ デフォルトのモニター対象ディレクトリーは\${USER_INSTALL_ROOT}/monitoredDeployableAppsであり、デフォルトを変更しても良い
- ◆ 本番環境での適用は非推奨

例: \${USER_INSTALL_ROOT}=C:/IBM/WebSphere/AppServer/profiles/Dmgr01



© 2012 IBM Corporation
85

ここでご紹介するモニター対象ディレクトリー機能は、アプリケーションのアーカイブ・ファイルをモニター対象のディレクトリーにドラッグ・アンド・ドロップすることで、WASに対してアプリケーションのインストール・更新が行われる機能です。

モニター対象ディレクトリー機能はデフォルトで無効に設定されていますので、有効に設定する必要があります。有効に設定した後、モニター対象ディレクトリーmonitoredDeployableApps(デフォルト設定の場合、変更可能)が自動で作成されます。

Designing Web System Infrastructure with WAS V8.0



モニター対象ディレクトリー (2/4)

- モニター対象ディレクトリーを使ってインストールした場合
 - ◆ 同じ名前のアーカイブ・ファイルを(EAR, JAR, WAR, SAR)モニター対象ディレクトリーに上書きすることで、アプリケーションを**更新**できる
 - ◆ ノード・エージェントとサーバーが起動中であれば、更新後にアプリケーションを同期・再起動してくれる



- ◆ モニター対象ディレクトリーにおいてあるモジュールを削除することで**アンインストール**できる



© 2012 IBM Corporation

86

アプリケーションの更新の場合は、同じ名前のアーカイブ・ファイルを(EAR, JAR, WAR, SAR)モニター対象ディレクトリーに上書きすることで、アプリケーションが更新されます。

アンインストールの場合は、モニター対象ディレクトリーに配置されているアーカイブ・ファイルを削除することで実現できます。

Designing Web System Infrastructure with WAS V8.0

New
v8

モニター対象ディレクトリー（3/4）

- **モニター対象ディレクトリーの特徴**
 - ◆ WASプロセス内で動作
 - ◆ 設定を有効にした場合、ディレクトリーが常にモニターされているので、間違ってファイルを上書き・削除すると、アプリケーションが変更・削除される
→対応策: 該当ディレクトリーに対するアクセス制御
 - ◆ 既にDMのモニター対象ディレクトリーにデプロイされているファイルと同名のファイルを別のモニター対象ディレクトリーにデプロイすると、最初にデプロイされたアプリケーションは削除される。(ファイルはディレクトリーに残る。)
 - ◆ サポート:
 - Express版、Base版、ND版
 - Java EE 5.0 以降の EAR、EJB、WAR、SAR
 - ◆ 対象ファイル
 - アプリケーションのアーカイブ・ファイル
 - プロパティー・ファイル
 - PFBCTを使用することでアプリケーションの1つのファイル、1部のファイルなどの部分的な更新も可能
- **モニター対象ディレクトリーのメリット**
 - 頻繁にアプリケーションを更新する開発環境に適する

①ドラッグ・アンド・ドロップ

②削除

③デプロイ

© 2012 IBM Corporation

既に、デプロイメント・マネージャーの対象ディレクトリーにデプロイされているファイルを、デプロイメント・マネージャーの別のモニター対象ディレクトリーにデプロイする場合は、現在のターゲット・モニター対象ディレクトリーからそのファイルを除去した後、新しいターゲット・モニター対象ディレクトリーにそのファイルを追加します。このリリースでは、1つのアプリケーション・ファイルは、デプロイメント・マネージャーの1つのモニター対象ディレクトリーにのみ存在する必要があります。

モニター対象ディレクトリーに配置できるアーカイブ・ファイルはJava EE 5.0より以降のEAR、EJB、WAR、SARモジュールとなります。また、PFBCTを使用することで、アプリケーションの1つのファイル、一部のファイルなどの部分的な更新も可能になります。

Designing Web System Infrastructure with WAS V8.0

New
v8

【参考】モニター対象ディレクトリー（4/4）

■ モニター対象ディレクトリーの有効設定

```

cell1=AdminConfig.getid('/Cell:Cell01/')
md=AdminConfig.showAttribute(cell1, "monitoredDirectoryDeployment ")
AdminConfig.modify(md,['enabled', true],['pollingInterval', 10],['monitoredDirectory',
${USER_INSTALL_ROOT}/monitoredDeployableApps'])
AdminConfig.save()

```

88

モニター対象ディレクトリーという機能ですが、管理コンソールの「グローバル・デプロイメント設定」から設定します。有効に設定するためには、「アプリケーションの自動デプロイのためにディレクトリーをモニター」にチェックを入れます。更に、モニター対象ディレクトリーを監視する、ポーリング間隔はデフォルトで5秒であり、ポーリング間隔この画面から変更できます。

管理コンソールから行う作業をwsadminコマンドで行う手順もあります。

Designing Web System Infrastructure with WAS V8.0

Fine-Grained Update 機能

- 選択した更新オプションの単位で更新
 - ◆ A. 全アプリケーション
 - ◆ B. 単一モジュール(War/Jar)
 - ◆ C. 単一ファイル(.class/.html ...)
 - ◆ D. アプリケーションの一部

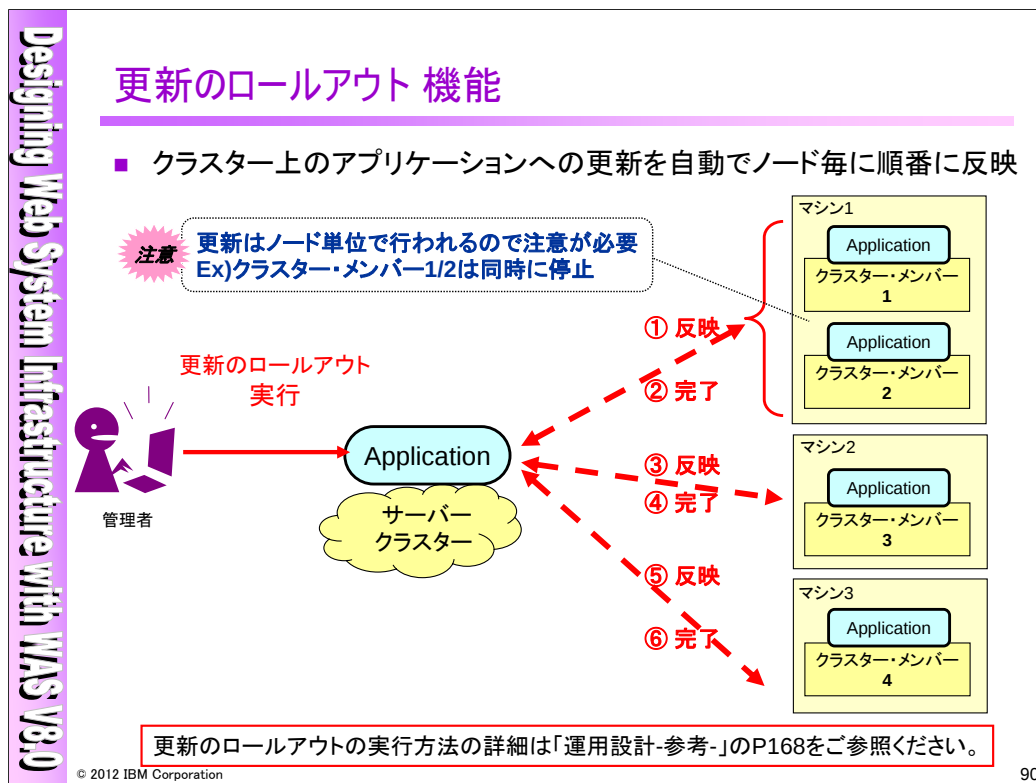
Fine-Grained Updateの設定の詳細は「運用設計-参考-」のP.164-167をご参照ください。

© 2012 IBM Corporation

89

WASのFine-Grained Update機能を使用することで、エンタープライズ・アプリケーション(EAR)単位だけではなく、EAR内の単一モジュール、単一ファイル、アプリケーションの一部(複数のモジュールやファイルの組み合わせ)という単位で更新を行うことができます。

Fine-Grained Updateの設定の詳細に関しては、「運用設計-別紙-」のP.164-167をご参照ください。



WASの更新のロールアウト機能とは、クラスター上のアプリケーションを更新する際、サーバークラスターに対する更新を全てのクラスター・メンバーに同時に反映させるのではなく、ノード毎に順番に反映させていくという機能です。この機能を使用すると、クラスターを構成している複数ノードのうち常にいずれかのノード上のクラスター・メンバーは稼動しているため、サービス全体としては停止することなくアプリケーションの更新を行うことができます。

ただしロールアウト機能はノード単位の更新になるため、垂直クラスター構成の場合は更新中のサービスが停止してしまいます。また、水平クラスターの場合でも複数のノードを順番に更新するようになり、更新済みのノードと更新前のノードが混在することになり、新旧のバージョンが混在してしまう時間帯が生じてしまいます。

更新のロールアウトの実行方法の詳細に関しては、「運用設計-別紙-」のP168をご参照ください。

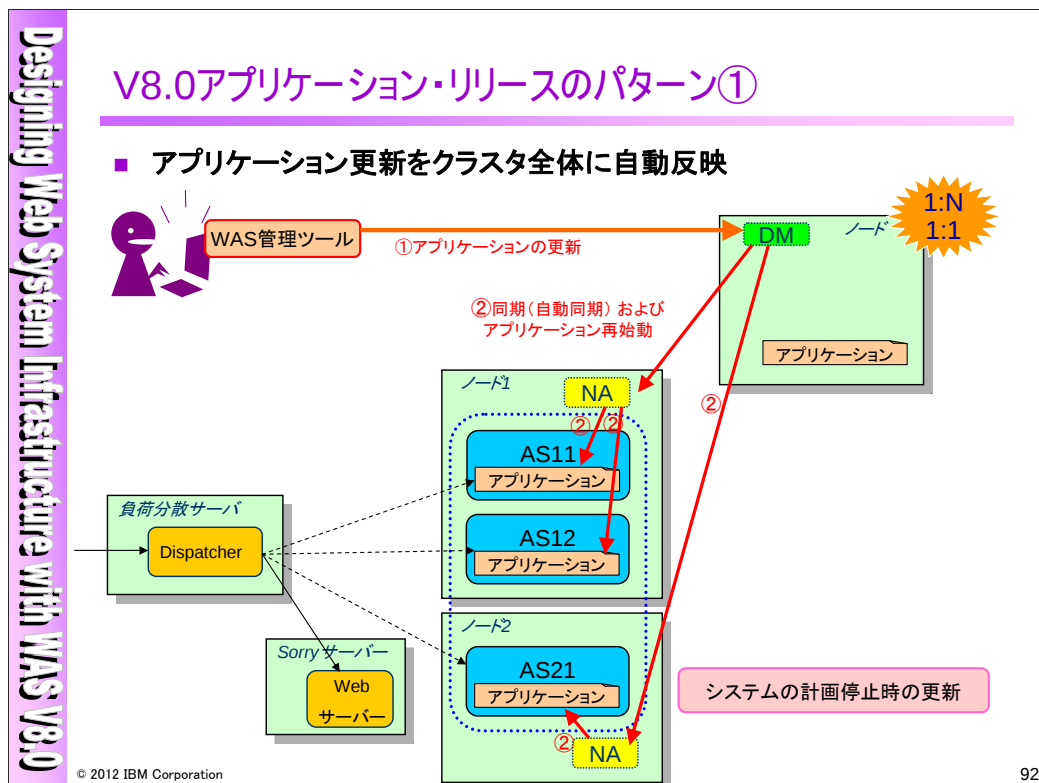
Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン**
- 4-8 非定期運用に関する設計ポイント

91

© 2012 IBM Corporation



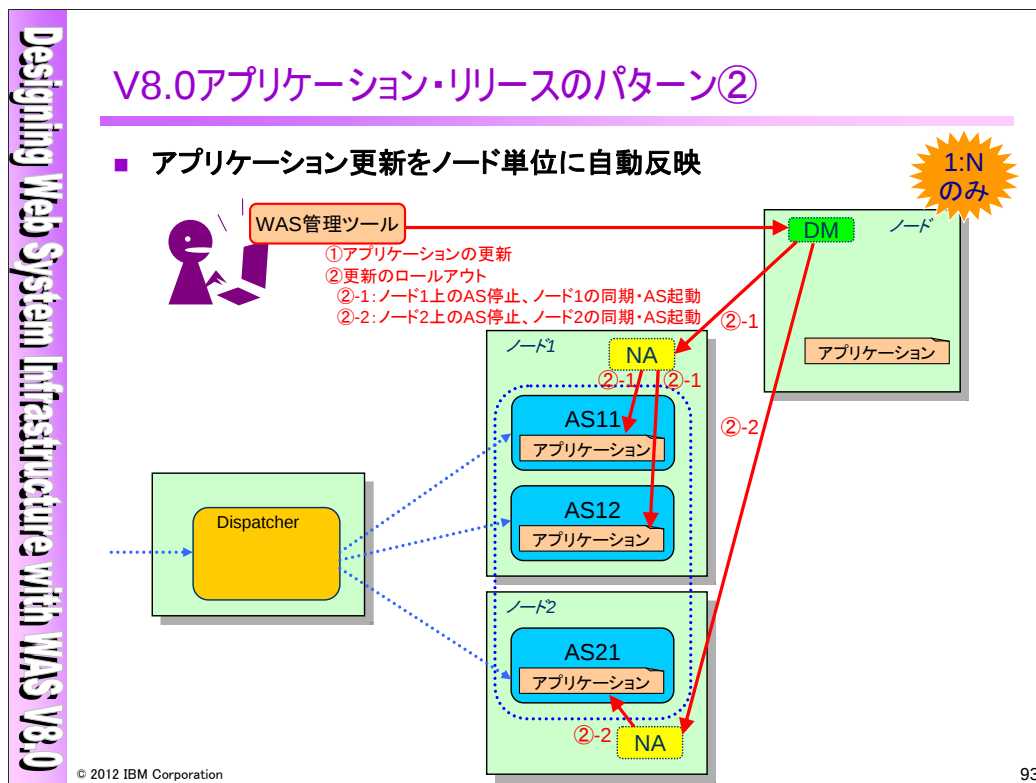
アプリケーション更新をクラスター全体に同時に自動反映させる場合の更新フローです。一般に、定期メンテナンス時間があり、ユーザー・アクセスがない(Sorryサーバへ割り振りを行う)状態で、更新を行う方法です。

ポイント

- 同期やアプリケーションの再起動はWASにより制御されます。
- アプリケーション再起動完了のタイミングがアプリケーション更新処理の完了のタイミングとなります。

考慮点

- システム全面ダウンのタイミングが発生します。
- EARファイルサイズが大きいケースの場合には、同期処理に時間がかかる場合がありますので、Fine-Grained Updateを使用して更新範囲を小さくするなどの工夫が必要です。また、同期処理に時間がかかる場合は、soap.client.propsで設定されているwsadminコマンドの接続に関するタイムアウト値を変更する必要があるか、確認しておく必要があります。



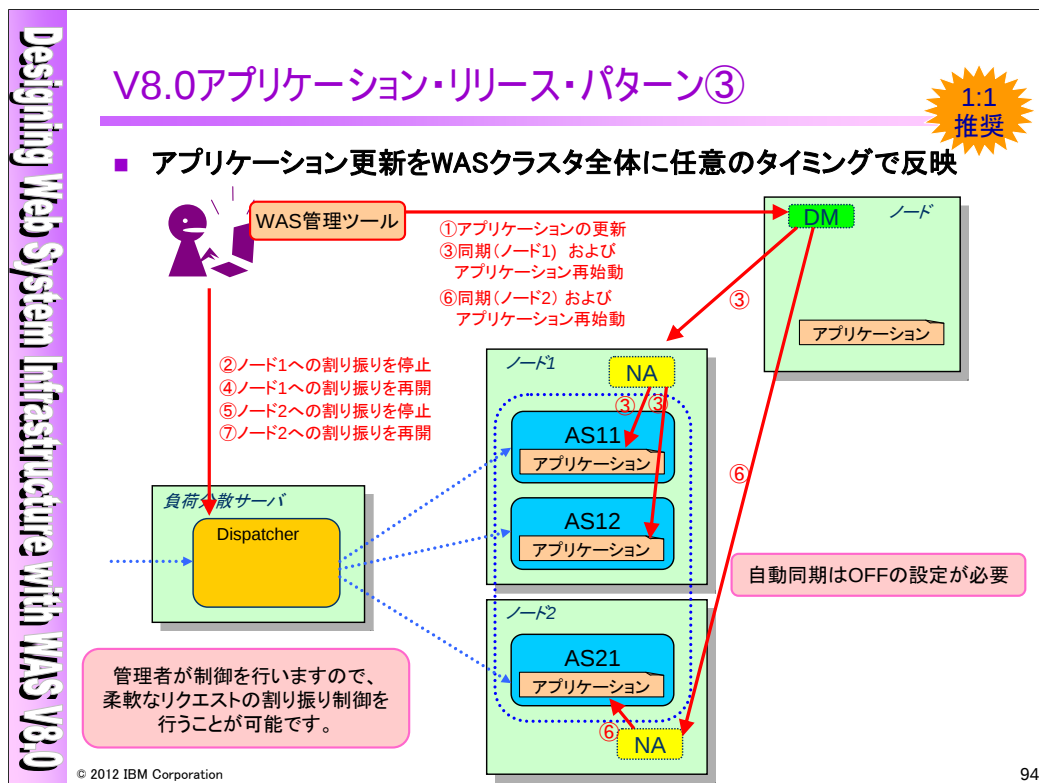
アプリケーション更新をノード単位に自動反映させる場合の更新フローです。更新のロールアウト機能を使用することにより、ユーザー・サービスを無停止で、更新を行うことができます。

ポイント

- ・同期やアプリケーション・サーバーの再起動はWASにより制御されます。
- ・「更新のロールアウト」を行うと、自動的に1ノードずつの同期とアプリケーション・サーバー再起動が行われます。
- ・アプリケーション・サーバーが再起動するまでの間はIHSなどWebサーバーにより割り振りが停止されますので、1対1構成ではなく、1対N構成をとる必要があります。
- ・アプリケーション・リリース時の負荷分散装置による割り振り制御は不要です。

考慮点

- ・更新時には、異なるアプリケーション・サーバーにリクエストが割り振られますので、各アプリケーション・サーバー間でセッション情報を共用するセッション・パーシスタンスを行うことが推奨されます。
- ・新旧アプリケーションが混在するタイミングがありますので、セッション・オブジェクトなどのアプリケーションの更新範囲によって、必ずしもユーザーの処理が正しく完了できない場合があります。
- ・アプリケーション・サーバーの再起動はノードごとに行われますので、垂直クラスターだけでは、更新のロールアウトは使用することはできません。



アプリケーション更新をクラスタ全体に任意のタイミングで反映させる場合の更新フローです。アプリケーションの更新やユーザー・リクエストの割り振りは管理者が制御します。イメージとしては、②の更新のロールアウトの機能を手動で行う方法です。

ポイント

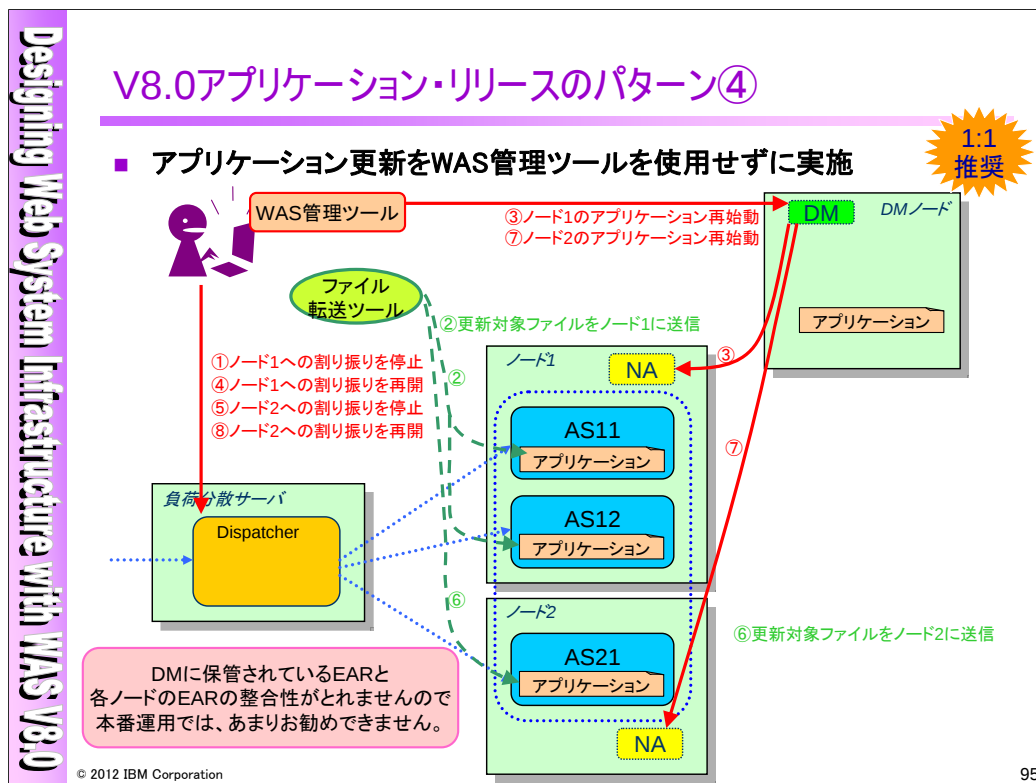
- 同期のタイミングは管理者が制御します。
- アプリケーション更新中のノードには割り振りを行わないように、管理者が割り振り制御を行います。(図中では負荷分散サーバーによる割り振り制御)
- 負荷分散サーバーではなく、IHSのプラグインで割り振り制御を行うことも可能ですが、その場合は、プラグイン構成ファイルの置き換えが必要になりますし、割り振りリクエスト数などを細かく追跡することができませんので、通常、1対1構成をとり、負荷分散装置で割り振りの制御を行います。

考慮点

- パターン②と同様

要設定項目

- ノード・エージェントの自動同期機能をOFFにしておきます。



アプリケーション更新をWAS管理ツールを使用せずに実施する場合の更新フローです。各ノード上のクラスファイルや、JSP、HTMLファイルなどを直接更新する方法です。この方法は、デプロイメント・マネージャーが保持するマスターのEARファイルと、各ノード上に展開されるファイルの整合性が取れなくなりますので、本番環境での使用はあまりお勧めできません。

ポイント

- WAS管理ツールを使用せずにアプリケーション更新を完了できます。
- アプリケーション更新・再始動のタイミングは管理者が制御します。
- アプリケーションの再起動が必要ない場合もありますが、基本的にはアプリケーションまたはアプリケーション・サーバーの再起動が必要になりますので、割り振り制御を行う必要があります。(図中では負荷分散装置による割り振り制御)

考慮点

- 構成リポジトリ内のEARファイルと、ランタイムでのアプリケーションとの整合性が取れない状態となります。
 - アプリケーションのエクスポート機能によるEARの復元はできません。
 - 障害時開発物のバージョン管理システムが確立されている必要があります。

Designing Web System Infrastructure with WAS V8.0

WVE (WebSphere Virtual Enterprise) のアプリケーション管理 - エディション・コントロール

- 同一のターゲット(動的/静的クラスター、アプリケーション・サーバー)に、
 - ◆ 同一のアプリケーションを、
 - ◆ 異なるエディションとして、
 導入することができる
 - ◆ WAS NDでは、同一アプリケーション名もしくは同一コンテキスト・ルートを持つアプリケーションは、導入できなかった
- 同じターゲット上で稼働できるのは、一時点では1つのエディションのみ、異なるターゲットであれば、同時に複数のエディションを稼働することができる

© 2012 IBM Corporation

WVEでは、アプリケーション管理の機能として、エディション・コントロールの機能があります。エディション・コントロール機能のポイントは2つあります。

1つ目のポイントは、

WAS NDでは、1時点で導入可能な同一のアプリケーション(同一の名前、コンテキスト・ルート)は1つのみでした。新しいアプリケーションの導入後は前のバージョンが消えてしまっていました。WVEでは、同一のターゲット(動的/静的クラスター、アプリケーション・サーバー)に、

- ・同一のアプリケーションを、
- ・異なるエディションとして、

導入することができるようになっていきます。

2つ目のポイントは、

- ・同じターゲット上で稼働できるのは、一時点では1つのエディションのみ、異なるターゲットであれば、同時に複数のエディションを稼働することができる

という点です。

Designing Web System Infrastructure with WAS V8.0

WVE (WebSphere Virtual Enterprise)のアプリケーション管理 – ロールアウト

- 無停止でのアプリケーション切り替え
⇒ **ロールアウト 機能**の適用
 - WAS NDのロールアウト機能の場合、サービスの停止、または、新旧のバージョンが混在する時間が発生してしまう。
- ◆ 2つのモード
 - アトミック・ロールアウト – 全体サーバー数の半分ずつ更新
 - グループ・ロールアウト – 指定したサーバー数ずつ更新
- ◆ 更新処理中のリクエストを制御
 - ファイルの転送が発生しないため更新処理が迅速
 - バージョンの戻しも同様の手順でサポート
- 妥当性検査モード
 - ◆ 本番環境にて、現行バージョンを稼働させたまま新バージョンの稼働確認が可能

© 2012 IBM Corporation
97

次は、WVEのロールアウト・アップデート機能についてご説明します。

WAS NDの更新のロールアウト機能はノード単位の更新になるため、垂直クラスター構成の場合は更新中のサービスが停止してしまいます。また、水平クラスターの場合でも複数のノードを順番に更新するような形になりますので、更新済みのノードと更新前のノードが混在することになり、新旧のバージョンが混在してしまう時間帯が生じてしまいます。WVEの、ロールアウトは、迅速でサービスの停止なしでエディションの切り替えを提供しています。

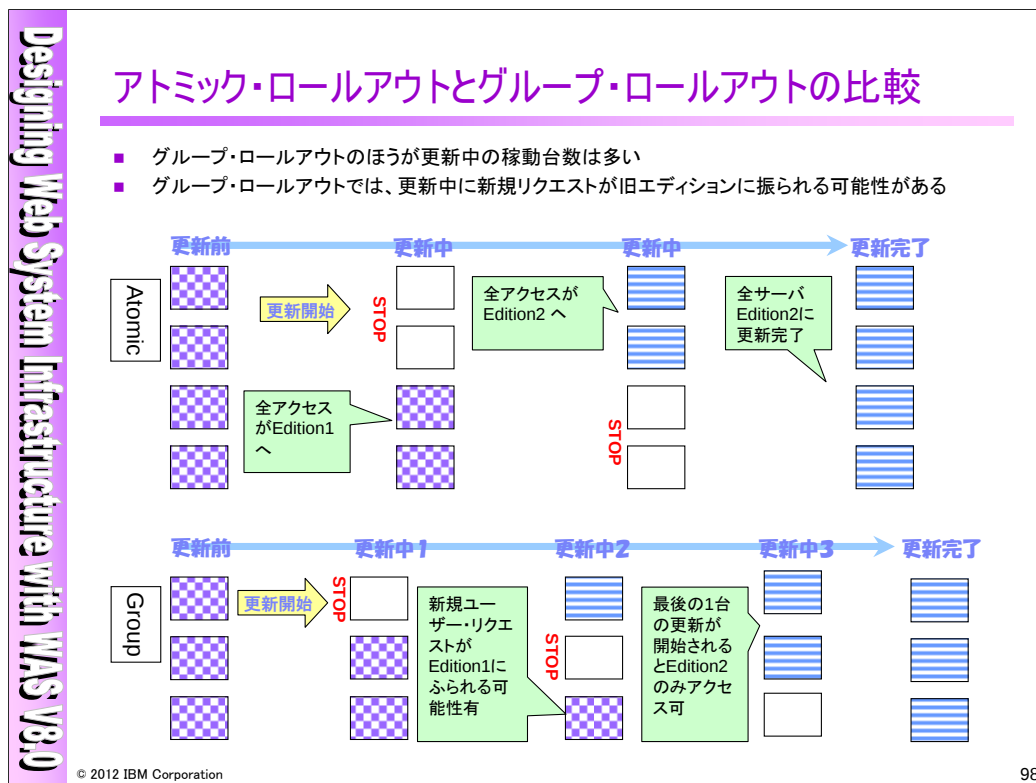
入れ替えの方式には以下の2つの方式があります。

- ・グループ・ロールアウト – 指定したサーバー数ずつ更新
- ・アトミック・ロールアウト – 全体サーバー数の半分ずつ更新

また、ロールアウト機能の使用に際して、以下を考慮する必要があります。

- ・セッションの扱い
 - ・ロールアウト前/後でエディション間でセッション情報は共有される
 - ・セッション・オブジェクトの内容、構造が変わらないことが前提
 - ・維持のためにはセッション・パーシステンス構成は必須
- ・キャッシュの扱い
 - ・ODRまたはその前段でキャッシュを使用している場合は注意が必要
- ・旧エディションの扱い
 - ・旧エディションは「非活動化」状態に設定される
 - ・ファイル自体は各ノードにデプロイされたまま
 - ・必要に応じてそのエディションに該当するアプリケーションをアンインストールする

ロールアウトの際には妥当性検査モード、というものを指定することが出来ます。通常のロールアウトであれば、アプリケーションのエディションを旧⇒新と切り替えるだけですが、妥当性検査モードの場合は自動的に、評価用のクラスタを作成し、そこに新エディションを導入するという動作をしてくれます。この評価用クラスターに対してルーティング・ルールを設定し、リクエスト条件を限定してテストを行います。テストが終わったら、通常のロールアウトを行うことで、本来のクラスターへの新エディションのロールアウト アップデートが行われます。ロールアウト アップデート完了後は、評価用クラスターは自動的に削除されます。



グループ・ロールアウトとアトミック・ロールアウトでは方式に違いはあるものの、どちらも以下に示す特徴、制約は同じです

・短期間に更新が完了。構成に依存しますが、基本的に数分から十数分ぐらいのスコープで更新が完了します。

・エディション間での互換性は必須。これは例えばログイン・リクエストは旧エディションで行われたとしても、その後アクセスしているうちに、サーバー上でエディションが更新されている可能性があるためです。

・セッション・パーシスタンスは必須。あるサーバーにアフィニティを持っていても、そのサーバーに対する更新処理が開始されるとリクエストは別のサーバーに振られるため、クラスター・メンバー間でのセッション情報の共有は必須となります。

2つの方式の違いとしては主に以下の2つがあげられますので、これらを参考に、どちらの方式を選択するかを決定します。

・アトミック・ロールアウトではエディションが混在しないため、全ユーザーのリクエストがある時点では必ずどちらかのエディションにのみ振られますが、グループ・ロールアウトの場合、旧エディションに振られるリクエストと新エディションに振られるリクエストが存在します。したがって、既存ユーザー・リクエストだけでなく、新規ユーザー・リクエストが旧エディションに振られる可能性があります。(上記グループ・ロールアウト図の「更新中2」)

・更新中の稼働台数の違い。

Designing Web System Infrastructure with WAS V8.0

WVEとWASのロールアウト機能比較

	WVE ロールアウト機能	WAS ND ロールアウト更新	WAS ND
更新単位	EAR	・EAR ・モジュール ・ファイル	・EAR ・モジュール ・ファイル
再起動単位	・アプリケーション ・アプリケーション・サーバー (選択可能)	アプリケーション・サーバー (ノード単位)	・アプリケーション ・アプリケーション・サーバー
新旧アプリケーションの混在	・アトミック・ロールアウトであれば発生しない ・グループ・ロールアウトでは発生する	発生する	発生する
リリース時間	短い	比較的長くなり得る	比較的長くなり得る
手戻り時間	短い	比較的長くなり得る	比較的長くなり得る
セッション維持	・セッション・パーシステンスが必須 ・保守モードを使えばアフィニティーだけで対応できる可能性あり。	・セッション・パーシステンスが必須	・セッション・パーシステンスが必須

© 2012 IBM Corporation

99

こちらはWVEのロールアウト機能を用いた場合と、WASのロールアウト機能を用いた場合、そしてWASの1:1構成でのリリース方法を比較した表になります。

WVEを導入することで次のような利点があります。

- ・WVEのアトミックロールアウトなら新旧アプリの混在を防げる。
- ・WVEのロールアウト更新はアプリ単位で再起動できるので、同一JVM上の別アプリに影響を与えない。

ただし、WVE導入には別途ライセンスが必要となります。

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント**

100

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WASV8.0非定期運用に関する設計ポイント①

- バックアップ・リストア方針
 - ◆ バックアップ・リストアの対象、バックアップ・リストアの方法、タイミング
- 監視方針
 - ◆ 監視項目、各監視項目の監視対象、監視方法(1:1構成/1:N構成を考慮)
- 障害対応方針
 - ◆ 障害検知時の対応方法

© 2012 IBM Corporation

101

ここで、WASV8.0の非定期運用におけるバックアップ・リストア方針・監視方針・障害対応方針に関する設計ポイントをまとめます。

バックアップ・リストア方針として、バックアップ・リストア運用をどのように行うかを検討します。まずは、想定される障害ケースと各ケースに対するリカバリー方法を検討します。その上で、バックアップ・リストアの対象やタイミング、方法について検討します。

監視方針として、どのように監視を行うかを検討します。まずは監視項目を列挙し、各監視項目において監視対象を洗い出します。そして、各監視対象をどの機能で監視するのか検討します。

障害対応方針としては、障害検知したらどのような対応を取るのか、対応は自動化させるか手動で行うかについて事前に検討しておきます。

Designing Web System Infrastructure with WAS V8.0

WASV8.0非定期運用に関する設計ポイント②

- 更新タイミング
 - ◆ サービス停止状態での計画的Fix Pack/Fix適用・リリース、サービス提供状態での計画的Fix Pack/Fix適用・リリース、緊急Fix Pack/Fix適用・リリース、
- 更新時間
 - ◆ 更新ファイルサイズに依存した時間、計画停止の中で確保できる時間
- 更新範囲
 - ◆ エンタープライズ・アプリケーション単位 (EAR)、モジュール単位 (WARJAR)、ファイル単位 (HTML/GIF)、BLA使用有無
 - ◆ Fix Pack適用対象サーバー
- 更新方法
 - ◆ 手動リリース、自動リリース、Fine-Grained Update機能の使用、更新のロールアウト機能の使用、Enhanced EARの使用、モニター対象ディレクシー（開発環境推奨）、WebSphere Virtual Enterpriseの導入

© 2012 IBM Corporation
102

WASV8.0の非定期運用におけるFix Pack/Fix適用・アプリケーション更新に関する設計ポイントをまとめます。

まずは、Fix Pack/Fix適用、アプリケーションの更新タイミングを検討します。業務要件に依存しますので、事前にいつ計画的な更新ができるのかを明確にしておきます。一般的に、定期的な計画停止を行っているシステムでは、その時間帯に更新作業を行います。その場合、計画停止中におけるバッチ処理なども考慮し、アプリケーション更新作業にどれくらいの時間が割り当てられるのかを確認しておきます。24時間365日稼動システムでは、夜間などアクセスの少ない時間帯に、サービスを止めることなく、片系運用にて1系ずつアプリケーション更新作業を行います。

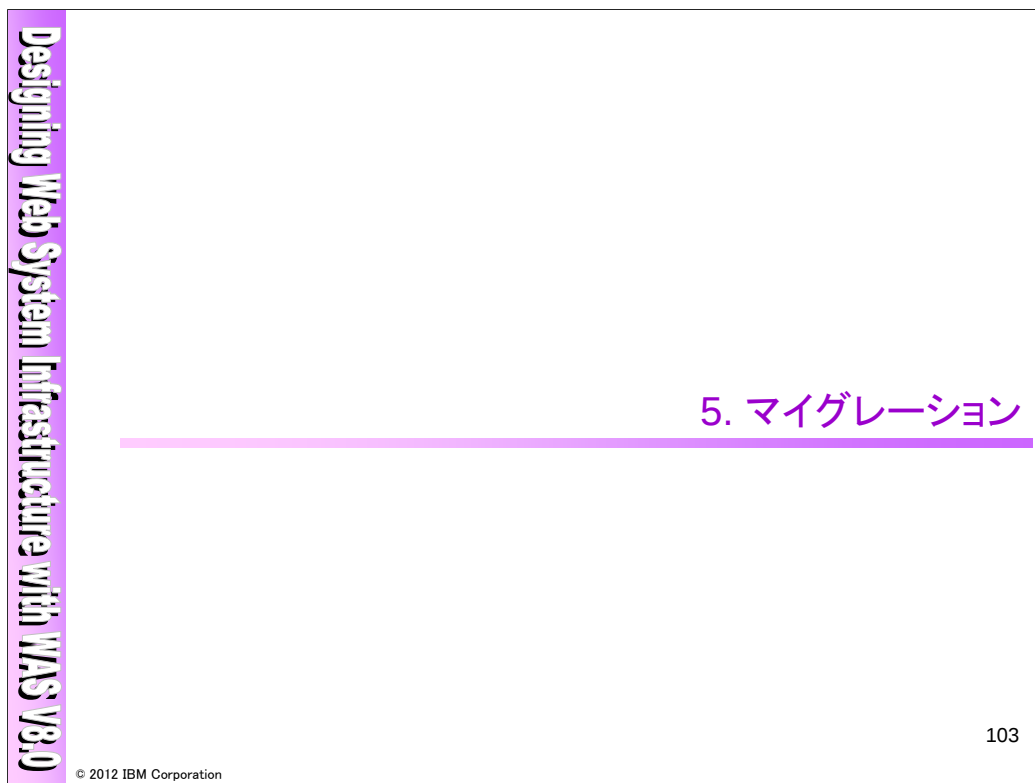
Fix Pack/Fix適用については、定期的にリリースされるFix PackやFixの適用をどのように行うかを検討します。常に最新のFix PackやFixがリリースされるたびに最新のものを適用するのか、リリースされてからある程度の期間を置いてから適用するのか、発生した障害あるいは緊急度の高い脆弱性に関するものだけを適用するのか、などを決めておきます。

アプリケーション更新は、方法によってかかる時間が変わりますので、まずは制約からアプリケーション更新に使用できる時間を割り出し、どのような方法で更新するのかを検討していきます。「毎回手順やスクリプトが変わるのを避けたいため毎回エンタープライズ・アプリケーション単位で更新する」、「アプリケーション更新にかけられる時間はわずかで、エンタープライズ・アプリケーションのファイルサイズが大きく展開先も多くて時間がかかるので、更新範囲を小さくする」、などといったように検討します。

Fix Pack適用はWAS、IHS、Plugin、Application Client、Java SDKの各コンポーネントごとに提供されます。Java SDKのFix Packの適用もWASのFix Packの適用とは別途行う必要がありますのでこちらについても更新時間を考慮する必要があります。

更新方法は、WASだけでなく、負荷分散装置でどこまで制御できるのか、WAS上のアプリケーション以外の運用スクリプトなどとリリース方法をそろえて運用手順を統一化すべきか、などといったWAS以外の要素も考慮した上で最適なアプリケーション更新の方法を検討する必要があります。

また、計画的なリリースだけでなく、障害対応による更新やアプリケーションの前バージョンへの戻しといった緊急リリースもありえますので、その場合においても、正確かつ効率的に更新が行えるように、事前に更新範囲や更新方法についても検討しておき、運用手順化しておくことが重要です。



Designing Web System Infrastructure with WAS V8.0

WASでサポートされている主な仕様

WAS	J2EE/ Java EE	J2SE/ JDK	Servlet	JSP	EJB	DB Connection
3.5	N/A	J2SE 1.2	2.0/2.1 2.2	0.91/1.0 1.1	1.0+ α	ConnMgr JDBC 2.0
4.0	J2EE 1.2	J2SE 1.3	2.2	1.1	1.1	JDBC 2.0
5.0	J2EE 1.3	J2SE 1.3	2.3	1.2	2.0	JDBC 2.0
5.1	J2EE 1.3	J2SE 1.4	2.3	1.2	2.0	JDBC 2.0
6.0	J2EE 1.4	J2SE 1.4	2.4	2.0	2.1	JDBC 3.0
6.1	J2EE 1.4	J2SE 5.0	2.4	2.0	2.1	JDBC 3.0
7.0	J2EE 1.4 Java EE 5	Java SE 6	2.4 2.5	2.0 2.1	2.1 3.0	JDBC 4.0
8.0	Java EE 6	Java SE 6	3.0	2.2	3.1	JDBC 4.0

使用している場合、移行に当たって大きな修正が必要なもの
 使用している場合、移行に当たってある程度の修正が必要となる可能性が高いもの
 使用している場合、移行に当たってある程度の修正が必要となる可能性があるもの

© 2012 IBM Corporation 104

それぞれのWASのバージョンが対応している主な仕様のリストです。

図上で赤字で書かれている仕様如果使用されていると、WASV8へのマイグレーションにあたって大規模なアプリケーション修正が必要となります。黄色字の仕様如果使用されている場合にも、ある程度の修正が必要となる可能性があります。

その他に、黒字で書かれているバージョンについては、基本的な上位互換性がとられているため、一般的には大規模なアプリケーションの改修は必要ありません。ただし、細かい仕様の違いにより、小規模なアプリケーション修正が必要となるケースはあります。

WASのバージョンが4.0以上であれば、アプリケーションの改修はそれほど必要ないことが多いことがわかります。

Designing Web System Infrastructure with WAS V8.0

アプリケーションのマイグレーション

- **V7.0からV8.0への移行**
 - ◆ J2EE1.4、JavaEE 5からJavaEE 6は高い上位互換あり
 - ◆ V7.0で稼動していたアプリケーションはほぼ改修なく稼動
- **V6.1からV8.0への移行**
 - ◆ J2EE1.4からJavaEE 6は高い上位互換あり
 - ◆ V6.1で稼動していたアプリケーションはほぼ改修なく稼動
- **V6.0からV8.0への移行**
 - ◆ J2SE1.4からJavaSE6への対応が必要
 - ◆ 詳細はORACLE(Sun)の文書を参照
 - <http://java.sun.com/j2se/1.5.0/compatibility.html>
 - <http://www.oracle.com/technetwork/java/javase/compatibility-137541.html>
- **V5.1以前からV8.0への移行**
 - ◆ J2EE1.4以降への移行が必要
 - ◆ J2EE1.2以降であればツールを使用してJ2EE1.4へ移行可能

© 2012 IBM Corporation
105

WASV8.0を直近のバージョンであるV7.0と比較すると、非常に高いレベルの上位互換性がとられています。WASV8.0上では、WASV7.0用に作成されたJ2EE 1.4/JavaEE 5準拠のアプリケーションをそのまま稼働させることができます。

WASV6.1をWASV8.0と比較した場合においても、非常に高いレベルの上位互換性がとられています。既述の通りWASV8.0上では、WASV6.1用に作成されたJ2EE 1.4準拠のアプリケーションをそのまま稼働させることができます。WASV6.1より前のバージョンからの移行対象がV6.1/V7.0/V8.0いずれであっても、必要なアプリケーション修正内容は同じになります。

WASV6.0からWASV8.0への移行にあたっては、Java実行環境の仕様がJ2SE 1.4からJava SE 6に変わったことによる対応が必要となります。Java実行環境の仕様は、J2SE 1.4からJ2SE 5.0にあがる際に大きく変更されていますので、これらの変更への対処が必要となります。また、Java SE 6ではJ2SE 5.0にあった機能の一部が削除されています。詳細については、ORACLE(Sun)のサイトで情報が公開されています。

WAS 5.1以前のアプリケーションにおいては、J2EEのバージョンを1.4に、JDK/J2SEのバージョンをJava SE 6にあげる必要があります詳細については、既存のWAS V6.1へのマイグレーションの資料がそのまま使用できるので、そちらを参照ください。Java EE 5/6ではアプリケーションの記述方法が大きく変わっていますので、Java EE 5/6へ移行する必要はありません。実際、Rational Application Developer 7.5で提供されているマイグレーション・ウィザードでも、J2SE 1.2/1.3のアプリケーションを1.4に変換することはできますが、Java EE 5/6へ変換する機能は提供されていません。Java EE 5による開発は、新規アプリケーション開発をおこなうときに使用します。

Designing Web System Infrastructure with WAS V8.0

システムのマイグレーション

- 6.0/6.1/7.0からV8.0への移行
 - ◆ V8.0との共存や相互運用はサポート
 - ◆ ツールによる既存のアプリケーション実行環境の自動移行が可能

- V5.0、V5.1からV8.0への移行
 - ◆ 基本的な管理トポロジは変更されていないため同様の構成が可能
 - ◆ ツールによるマイグレーションがサポートされていないため手動で同様のアプリケーション実行環境を再作成する必要あり

- V4.0以前からV8.0への移行
 - ◆ サーバーの管理トポロジが根本的に変更になっているため従来の構成/設定内容/運用スクリプトなどを流用することは不可
 - ◆ 新規にアプリケーション実行環境を設計し直す必要あり

© 2012 IBM Corporation
106

WASV5.0以降からWASV8.0への移行であれば、構成情報などは新環境に引き継ぐことが可能です。WASV5.0はツールによる自動移行ができませんが、WASV5.1以降ではツールによる移行もサポートされます。

WASV4.0以前とV5.0以降では、サーバーの管理トポロジが根本的に変更されています。これらのバージョン間では、起動しているプロセスの種類やその役割、出力されるログの種類と場所、サーバーを管理するためのコマンドの名前や使用方法、構成情報が保存される場所や方式などが全て異なっています。WASV4.0以前のサーバー構成情報や運用・監視手段は、新しい環境に移行することができません。完全に新しく設計し直す必要があります。

Designing Web System Infrastructure with WAS V8.0

混合セルを使用したマイグレーション

- 混合セルを使用したマイグレーション
 1. 既存システムのデプロイメント・マネージャーをV8.0にマイグレーション
 2. セル内のノードをクラスタ単位でV8.0にマイグレーション
- ◆ 注意事項
 - V8.0のデプロイメント・マネージャーはV6.0/V6.1/V7.0のノードを含むことができるが、V6.0.0.xおよびV6.0.1.xのノードは管理することができないため、必ずFixPack/Fixを適用してV6.0.2.xにあげておく必要がある
 - デプロイメント・マネージャーはセル内で最上位のバージョンでなければならない
 - 混合セルではアプリケーション・サーバーの作成などに制限があるので、長期運用は避け、あくまでもマイグレーション中の一時的な構成として使用する
 - Nodeをマイグレーションした直後はsyncNodeコマンドを使用して手動で同期を行う必要がある場合もある
 - 1つのアプリケーションが実行されているクラスタは全て同じバージョンにする
HttpSessionの内容を新旧バージョン間で共有することはできない

© 2012 IBM Corporation
107

WASV8.0では、1つのセル内に旧バージョンのノードを同居させることができる混合バージョンセルをサポートしています。これにより、複数サーバーで構成された環境を段階的にWASV8.0にマイグレーションしていくことができます。

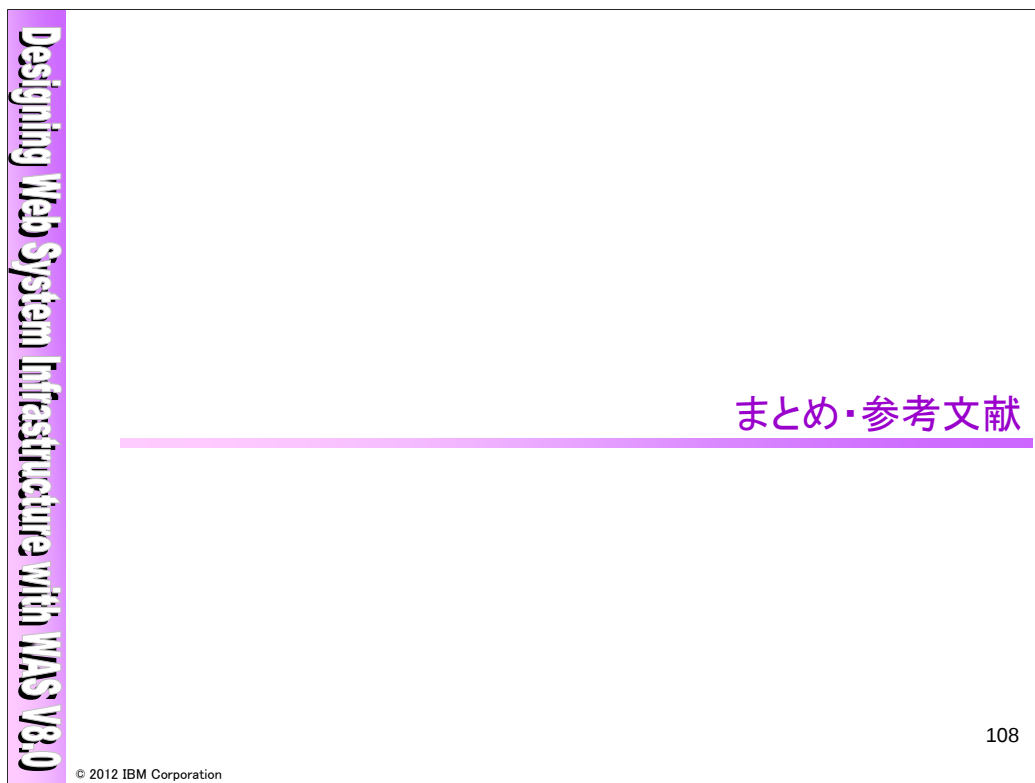
混合セルを使用してマイグレーションを行う場合、まずは、デプロイメント・マネージャーがセル内で最上位のバージョンでなければならないという制約があるため、既存システム内のデプロイメント・マネージャーをV8.0にマイグレーションします。その後で、段階的にセル内のノードをクラスタ単位でマイグレーションしていき、最終的にシステム全体のV8.0へのマイグレーションが完了します。

混合セルを使用したマイグレーションにおいて、いくつか注意事項があります。WASV8.0のデプロイメント・マネージャーはV6.0/V6.1/V7.0のノードを含むことができますが、V6.0.0.xおよびV6.0.1.xのノードは管理することができないため、必ずFixPack/Fixを適用して既存環境のノードのWASはV6.0.2.xにあげておく必要があります。また、既述の通り、デプロイメント・マネージャーはセル内で最上位のバージョンでなければなりません。混合セルではアプリケーション・サーバーの作成などに制限があるので、長期運用は避け、あくまでもマイグレーション中の一時的な構成として使用することとします。ノードをマイグレーションした直後はsyncNodeコマンドを使用して手動で同期を行う必要がある場合もあります。1つのアプリケーションが実行されているクラスタは全て同じバージョンにします。HttpSessionの内容を新旧バージョン間で共有することはできないのでご注意ください。

developerWorksにV8.0へのマイグレーションの詳細について記載した記事もございますので、V8へのマイグレーションをご検討される方はこちらもご参照ください。

developerWorks - 「WebSphere Application Server V8.0へのマイグレーションガイド」

http://www.ibm.com/developerworks/jp/websphere/library/was/was8_migration/



Designing Web System Infrastructure with WAS V8.0

まとめ

- 構成管理
 - ◆ WASV8.0新機能
 - IBM Installation Managerによる製品・Fixインストール・アンインストール等
 - プロファイル作成ツール、Webサーバ・プラグイン構成ツール
 - 集中インストール・マネージャー、管理コンソールの機能拡張
 - ◆ 設計ポイント
 - 構成管理方針、構成変更方針
- 定常運用
 - ◆ WASV8.0新機能
 - HPEL
 - ◆ 設計ポイント
 - 起動停止方針、ログ運用方針、自動化方針
- 非定期運用
 - ◆ WASV8.0新機能
 - モニター対象ディレクトリー
 - ◆ 設計ポイント
 - バックアップ・リストア、監視、障害対応
 - アプリリリース、Fix Pack/Fix適用の更新タイミング、更新時間、更新範囲、更新方法
- マイグレーション
 - ◆ WASV8.0へのマイグレーション

© 2012 IBM Corporation
109

最後に、当セッションで説明した内容をまとめます。

構成管理に関してはWAS管理ツールやフレキシブル・マネージメント等のV7.0までの機能に加えWASV8.0新機能として、製品・Fixのインストール方法等が変更になったこと、プロファイル作成ツール、Webサーバ・プラグイン構成ツール、集中インストール・マネージャー、管理コンソールの機能拡張をご紹介しました。そして、設計ポイントとして、構成管理方針、構成変更方針を明確にしておく、ということをご説明しました。

定常運用ではWASV8.0新機能としてHPELをご紹介いたしました。また、設計ポイントとして、起動停止方針、ログ運用方針、自動化方針を明確にしておく、ということをご説明しました。

非定期運用では設計ポイントとして、バックアップ/リストア、監視、障害時対応方針やFix Pack/Fix適用、アプリケーションの更新タイミング、更新時間、更新範囲、更新方法を明確にしておく、ということをご説明しました。また、WASV8.0新機能として、モニター対象ディレクトリーをご紹介するとともに、WASのHigh-Value製品であるWVE (WebSphere Virtual Enterprise)を利用したアプリケーション管理についてご紹介しました。

マイグレーションに関しては、WASV8.0へのマイグレーションの方法についてご紹介しました。

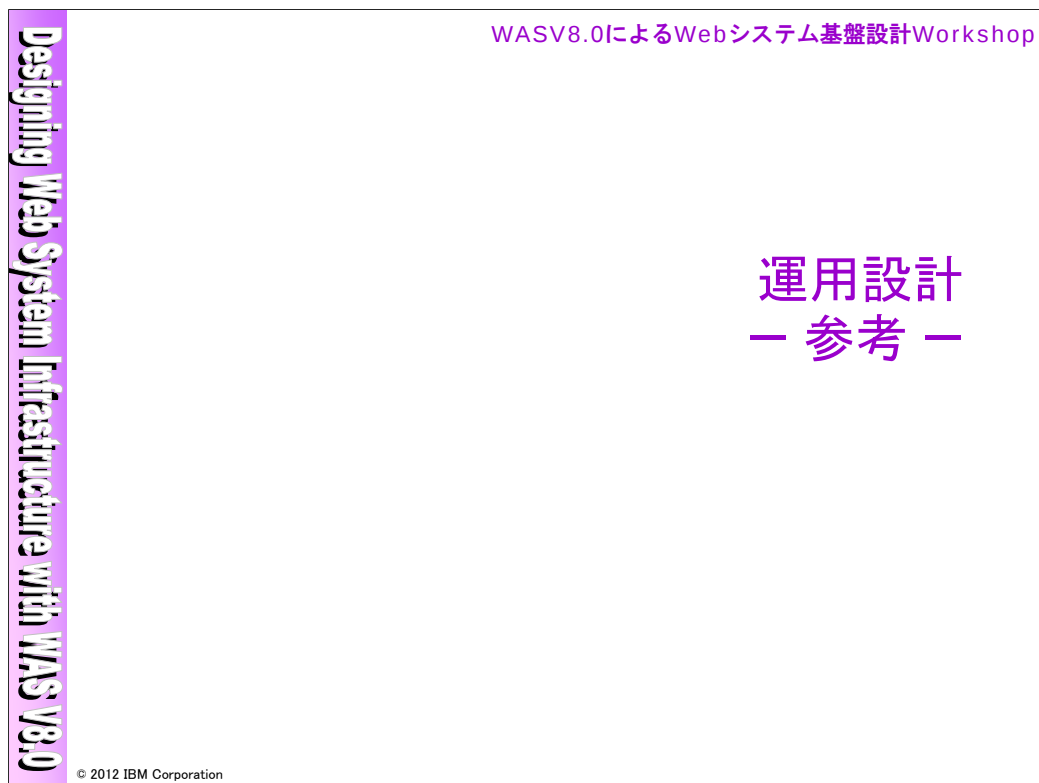
参考文献

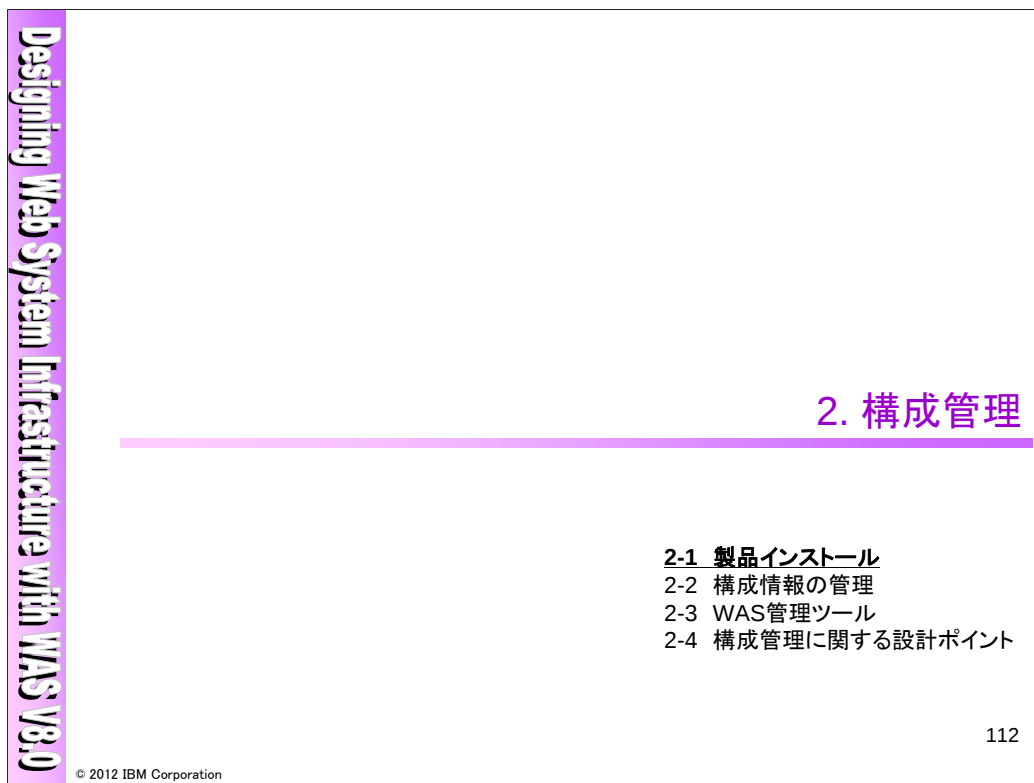
■ Information Center

- ◆ WebSphere Application Server V8.0 Information Center
 - http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.wasphere.home.doc_wasinfo_v8r0/welcome_collab.html
- ◆ IBM WebSphere Virtual Enterprise バージョン 7.0
 - <http://publib.boulder.ibm.com/infocenter/wveinfo/v7r0/index.jsp>

■ developerWorks資料

- ◆ WebSphere Application Server V8 アナウンスメント・ワークショップ
 - http://www.ibm.com/developerworks/jp/websphere/library/was/was8_ws/
- ◆ WebSphere Application Server V8.0へのマイグレーションガイド
 - http://www.ibm.com/developerworks/jp/websphere/library/was/was8_migration/
- ◆ WebSphere Application Server V7.0による基盤設計ワークショップ
 - http://www.ibm.com/developerworks/jp/websphere/library/was/was7_guide/index.html





Designing Web System Infrastructure with WAS V8.0

CIMよりIIMをインストール(1/4)

- 事前準備
 - ◆ IIMインストール・イメージを事前に入手
- Job Managerの管理コンソールからリモート・ホストを登録
 - ◆ 管理コンソールのナビゲーション・ツリーから、「ジョブ」>「ターゲット」の順でクリック
 - ◆ 「新規ホスト」をクリックし、リモート・ホストの情報を入力し、リモート・ホストをJob Manager配下に登録

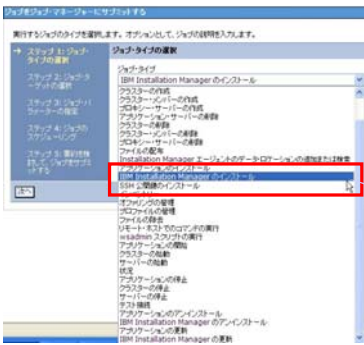
© 2012 IBM Corporation
113

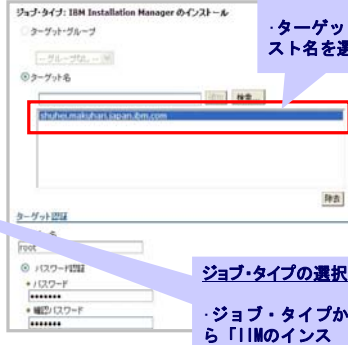
- ・前述の方法で事前に製品メディアかWebサイトよりIIMのインストール・イメージを入手し、Job Managerから認識可能なリポジトリに格納します。
- ・Job Managerを開始し、インストール対象のリモート・ホストをJob Managerのターゲットに設定します。Job Managerのコンソールで、「ジョブ」>「ターゲット」>「新規ホスト」をクリックして、「新規ターゲット」ページのフィールドにリモート・ホストの情報を入力します。

Designing Web System Infrastructure with WAS V8.0

CIMよりIIMをインストール(2/4)

- 管理コンソールのナビゲーション・ツリーから、「ジョブ」>「サブミット」の順でクリック
- 「IBM Installation Manager のインストール」ジョブを選択して、「次へ」をクリック
- ジョブ・ターゲットを選択し、「ターゲット名」を選択した後は、ターゲット名を指定して「追加」をクリックするか、「検索」をクリックして、「ターゲットの検索」ページで選択したターゲットを指定





インストール先の選択

・ターゲットのホスト名を選択

ジョブ・タイプの選択

・ジョブ・タイプから「IIMのインストール」を選択

© 2012 IBM Corporation

114

- ・管理コンソールのナビゲーション・ツリーから、「ジョブ」>「サブミット」の順でクリックします。
- ・「IBM Installation Manager のインストール」ジョブを選択して、「次へ」をクリックします。
- ・インストール対象ターゲットを選択します。
 - ー「ターゲット名」を選択した後は、ターゲット名を指定して「追加」をクリックするか、「検索」をクリックして、「ターゲットの検索」ページで選択したターゲットを指定します。
 - ーユーザー認証が必要な場合は、ユーザー名、パスワード、または必要に応じて他の認証の値を指定します。

WAS V8.0 によるWebシステム基盤設計ワークショップ

114

Designing Web System Infrastructure with WAS V8.0

CIMよりIIMをインストール(3/4)

- 「ジョブ・パラメーターの指定」ページで、以下を指定
 - ◆ IIMのイメージを格納したロケーション
 - ◆ インストール作業用ディレクトリー
 - ◆ リモート・ホストのIIMインストール・ディレクトリー

115

「ジョブ・パラメーターの指定」ページで、以下を指定します。

- ・Job Managerから認識可能なIIMインストール・イメージのパス
- ・インストール対象ターゲットにおけるインストール作業用ディレクトリー
- ・インストール対象ターゲットにおけるIIMインストール・ディレクトリー

Designing Web System Infrastructure with WAS V8.0

CIMよりIIMをインストール(4/4)

- ジョブ・スケジュールを設定し、ジョブを実行
- ジョブ・ステータスに「成功」したメッセージが表示されたことを確認

ジョブの実行と確認

- ・ジョブをジョブ・スケジュールに登録
- ・ジョブを実行し、実行結果を確認

© 2012 IBM Corporation

116

- ・ジョブをスケジュールに入れ、ジョブをサブミットします。
- ・ジョブが実行されて、選択したターゲット上に IIM がインストールされます。

Designing Web System Infrastructure with WAS V8.0

2. 構成管理

- 2-1 製品インストール
- 2-2 構成情報の管理**
- 2-3 WAS管理ツール
- 2-4 構成管理に関する設計ポイント

117

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WebSphereプロファイルの作成・管理 (manageprofiles)

■ manageprofilesコマンド

- ◆ プロファイルの作成・削除・バックアップ・リストア等を実施可能
- ◆ 使用方法:

```
#<WAS_ROOT>/bin/manageprofiles.sh (.bat) -<mode> -<option> <parameter> ...
```

- モードは全18種類
 - create プロファイルの作成
 - delete プロファイルの削除
 - listProfiles プロファイルのリスト

- backupProfile プロファイルのバックアップ
- restoreProfile プロファイルのリストア
- help ヘルプ

etc..

- ◆ 使用例:
- デプロイメント・マネージャー・プロファイルの作成

```
#<WAS_ROOT>/bin/manageprofiles.sh (bat) -create -profileName Dmgr01
-profilePath <WAS_ROOT>/profiles/Dmgr01
-templatePath <WAS_ROOT>/templates/profileTemplates/management
-serverType DEPLOYMENT_MANAGER -cellName iseCell01
-nodeName iseNode01 -hostName iseHost01
```

© 2012 IBM Corporation
118

manageprofilesコマンドは『<WAS_ROOT>/bin/manageprofiles.<実行ファイル拡張子>』で呼び出すコマンドラインツールです。プロファイルの作成など選べるプロファイル操作(モード)は18種類(create, augment, delete, unaugment, unaugmentAll, deleteAll, listProfiles, listAugments, backupProfile, restoreProfile, getName, getPath, validateRegistry, validateAndUpdateRegistry, getDefaultName, setDefaultName, response, help)あります。

例えばデプロイメント・マネージャー・プロファイルを作成する場合は上記コマンドのようにコマンドを発行します。

manageprofilesコマンドの詳細に関しては、下記リンク先をご参照ください。

WASV8.0 Information Center「manageprofiles コマンド」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.migration.nd.doc/info/ae/ae/rxml_manageprofiles.html

Designing Web System Infrastructure with WAS V8.0

Webサーバー・プラグイン構成ファイル – 設定

- プラグイン構成ファイルのプロパティ値の設定箇所
 - ◆ アプリケーション・サーバーの設定
 - ◆ Webサーバーの設定




© 2012 IBM Corporation
119

Webサーバー・プラグイン構成ファイルのプロパティ値の設定箇所は、2箇所にあります。

1つは、リクエストを割り振る側の設定であるWebサーバーのプロパティの設定項目です。管理コンソールで[サーバー]-[Webサーバー]-[web_server_name]-[追加プロパティ: プラグイン・プロパティ]のページで設定します。割り振り先のアプリケーション・サーバーをダウンしていると判定したのちの再実行間隔などは、こちらで設定します。

もう1つは、リクエストが割り振られる先の設定であるアプリケーション・サーバーのプロパティです。管理コンソールで[サーバー]-[アプリケーション・サーバー]-[application_server_name]-[追加プロパティ: Webサーバーのプラグイン・プロパティ]の画面で設定します。接続タイムアウトの設定や、接続の最大数の使用の設定はこちらで設定します。

Designing Web System Infrastructure with WAS V8.0

Webサーバー・プラグイン構成ファイル – 生成

■ プラグイン構成ファイルの生成コマンド

- ◆ <CUSTOM_PROFILE_ROOT>/bin/GenPluginCfg.sh(bat)
 - セル内の全ノード分の情報を含むプラグインを作成したい場合
 - DMプロファイル下のGenPluginCfgコマンドを実行
 - IHS:WAS=1:1構成となる各ノード固有のプラグインを作成したい場合
 - カスタムプロファイル下のGenPluginCfgコマンドを実行

```

cipher@root ./GenPluginCfg.sh -help

使用法: GenPluginCfg [-option.name optionValue]...

有効なオプション:
=====
-config.root configroot_dir
      (デフォルトは環境変数 CONFIG_ROOT)
-wasprofile profileName
      (構成データを含むプロファイルのオプションの名前)
-cell.name cell
      (デフォルトは環境変数 WAS_CELL)
-node.name node
      (デフォルトは環境変数 WAS_NODE)
-webserver.name webserver1
      (特定の Web サーバーのプラグイン構成ファイル作成に必要)
-propagate yes/no
      (webserver.name が指定されている場合にのみ適用可)
      (デフォルトは no)

```

```

-propagateKeyring yes/no
      (webserver.name が指定されている場合にのみ適用可)
      (デフォルトは no)
-cluster.name cluster1,cluster2 | ALL
      (クラスターのオプション・リスト)
      (オプション webserver.name が指定された場合無視される)
-server.name server1,server2
      (サーバーのオプション・リスト)
      (単一サーバー・プラグイン生成の場合に必要)
      (オプション webserver.name が指定された場合無視される)
-output.file.name file_name
      (デフォルトは configroot_dir/plugin-cfg.xml)
      (オプション webserver.name が指定された場合無視される)
-destination.root root
      (構成が使用されるマシンのインストール・ルート)
      (オプション webserver.name が指定された場合無視される)
-destination.operating.system windows/unix
      (構成が使用されるマシンのオペレーティング・システム)
      (オプション webserver.name が指定された場合無視される)
-debug yes/no
      (デフォルトは no)

```

© 2012 IBM Corporation

120

IHSとWASの構成を1対1構成とする場合には、GenPluginCfgコマンドの使用が役立ちます。

DMプロファイル以下のGenPluginCfgコマンドを使用する場合には、基本的には、管理コンソールやwsadminコマンドで生成するプラグイン構成ファイルと同様の構成ファイルを生成することができます。-server.nameオプションや-cluster.nameオプションを使用することで、特定のスタンドアローン・サーバーやクラスターに対して、割り振りを行うプラグイン構成ファイルを作成することができます。

カスタム・プロファイル以下のGenPluginCfgコマンドを使用することで、そのノード上の、スタンドアローン・サーバーやクラスター・メンバーにのみ割り振るプラグイン構成ファイルを生成することができます。割り振り元のWebサーバーが、コマンドを実行するノード上に存在する場合には、-webserver.nameオプションを使用することで、構成情報に設定したプラグイン・プロパティを反映したプラグイン構成ファイルを生成することができます。割り振り元のWebサーバーが、コマンドを実行するノード上に存在しないリモート構成の場合は、-webserver.nameオプションを合わせて使用することはできませんので、そのノード上のスタンドアローン・サーバーやクラスター・メンバーに割り振りを行う一般的なプラグイン構成ファイルを生成します。ここで、一般的と呼んでいるのは、-webserver.nameオプションを指定していませんので、Webサーバー側のプロパティがプラグイン構成ファイルに反映されません。たとえば、プラグイン再ロード間隔やログ・ファイルのディレクトリー、ダウンしたサーバーへの再割り振り間隔などは、デフォルト値が反映されますので、プラグイン構成ファイルを生成後、手編集で該当箇所を修正する必要があります。

GenPluginCfgコマンドの詳細については、下記のInformation Center の記載を参照ください。

WASV8.0 Information Center「GenPluginCfg コマンド」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_genplugincfg1.html

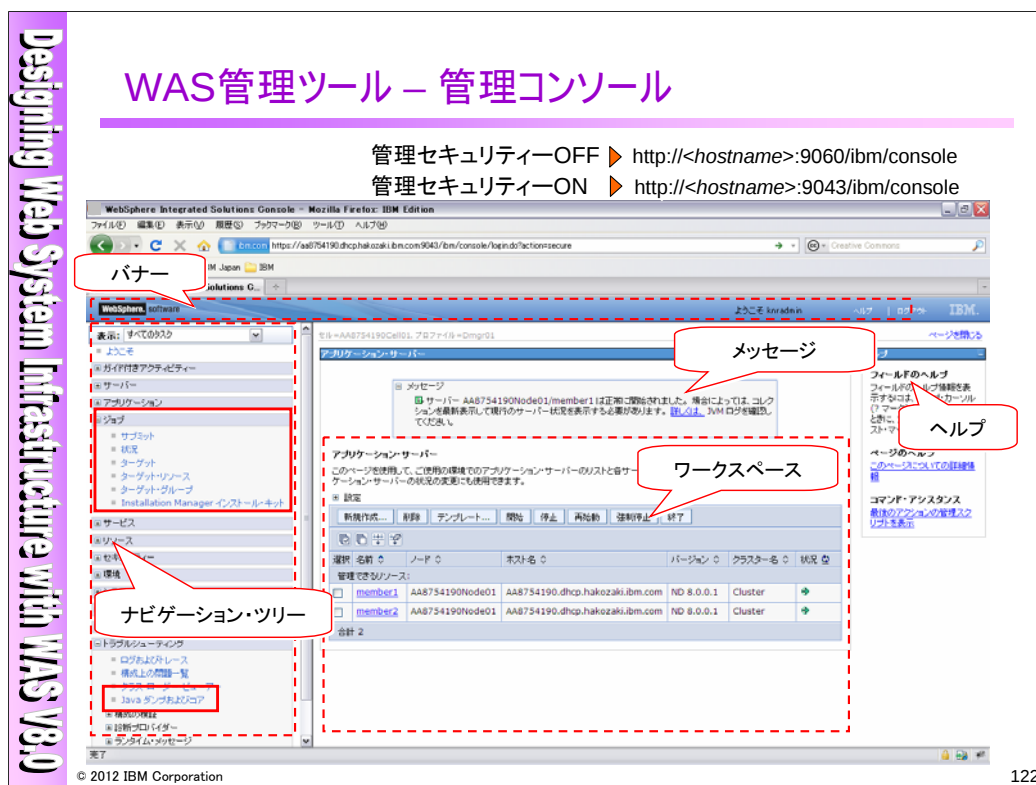
Designing Web System Infrastructure with WAS V8.0

2. 構成管理

- 2-1 製品インストール
- 2-2 構成情報の管理
- 2-3 WAS管理ツール**
- 2-4 構成管理に関する設計ポイント

121

© 2012 IBM Corporation



管理コンソールはGUIベースのWAS管理ツールです。管理コンソールから、アプリケーション・サーバーやアプリケーションの構成情報の変更や、アプリケーション・サーバーの起動停止などの状態の変更を行うことができます。ほとんどのWAS構成情報の変更は管理コンソールから行うことができます。上記画面はデプロイメント・マネージャーの管理コンソールです。この管理コンソールを使用するには、<http://<hostname>:<ポート番号>/ibm/console>にアクセスします。この時のポート番号は、該当のデプロイメント・マネージャーのWC.adminhost（デフォルト9060）を指定します。管理セキュリティを設定している場合は、WC.adminhost_secure（デフォルト9043）を指定します。管理コンソールを使用するには、ND構成の場合には、デプロイメント・マネージャーが起動している必要があります。WASV6.0から、Tivoli Performance Viewerは管理コンソールに統合され、パフォーマンス・モニタリングを容易に行うことが可能になりました。WAS V8では、デプロイメント・マネージャーの管理コンソールからいくつかのJob Manager機能を利用できるようになりました。また、管理コンソール上からJavaダンプや、Javaコアファイルを取得できるようになりました。管理コンソールのユーザー・インターフェースのレイアウトについては、以下の通りです。

・バナー

すべての統合ソリューション・コンソール環境に共通する画像を表示します。バナーには、ログインするユーザーへのグリーティング、コンソールをログアウトするリンク、およびコンソール・ヘルプをオープンするリンクが含まれます。

・ナビゲーション・ツリー

コンソールで使用可能なタスクをリストします。タスクは、「サーバー」や「アプリケーション」など、タスクのカテゴリを表す編成ノードにグループ化されます。編成ノードは、複数のレベルでネストすることがあります。タスクは、アクセスが可能なユーザーのみに表示されます。ナビゲーション内のタスクをクリックすると、ワークスペースにページが表示されます。ナビゲーション・エリアの上部にある「表示」選択リストを使用すれば、状況に合わせてタスクのリストを変更することができます。

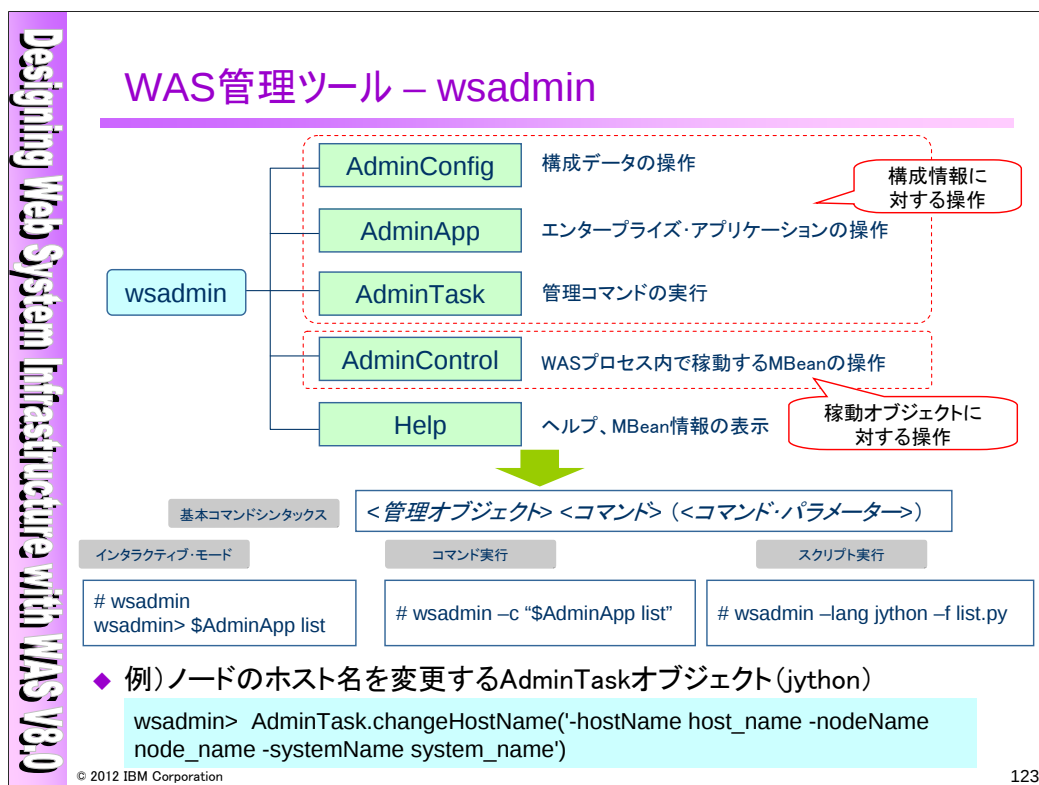
・ワークスペース

タスクをクリックすると、操作や構成情報を変更するページがワークスペースに表示されます。

管理コンソールの詳細については、下記のInformation Center の記載をご参照ください。

WASV8 Information Center「管理コンソールの使用」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tcon_console.html



wsadminはCUIベースの管理ツールで、WAS関連の構成変更はすべてwsadminから実行することができます。wsadminを実行するには、接続先のプロセス(基本的にはデプロイメント・マネージャー)が起動している必要があります。また、JaclとJythonの2種類のスクリプト言語がサポートされます。wsadminコマンド実行時のデフォルトは、Jaclであり、Jythonを用いる場合は-langオプションの指定が必要です。WAS V7.0において、Jacl言語は“安定化されたフィーチャー”とされました。今後Jacl言語のサポートが除去されることはなく、既存のwsadminスクリプトをJythonに置き換える必要はありませんが、今後のアップデートは、戦略的代替機能であるJythonに重点が置かれることになるため、新規に開発される場合は、Jythonにて作成することを推奨します。

wsadminには、インタラクティブ・モード、コマンド実行モード、スクリプト実行モードの3つの実行モードがあります。インタラクティブ・モードは対話形式で、順にコマンドの入力、実行を繰り返します。コマンド実行モードでは、実行したい処理内容を引数で渡します。1回の実行につき、1処理のみの実行になります。スクリプト実行モードでは、処理内容をあらかじめ記述したスクリプトファイルを引数に渡して実行します。

運用局面での、実行コマンドが複数に及ぶ場合には、コマンド実行モードでは毎回JVMが起動され、スクリプト実行モードで同じ処理を実施するよりマシンへの負荷が高くなりますので、スクリプト実行モードがお勧めです。

WASの構成可能なオブジェクトは全てJMXのMBean (Management Bean)と関連付けられています。MBeanはJMX APIを使用して管理するユーザーのための属性のgetter、setterやstart、stopなどをもちます。管理コンソールもwsadminもいずれもこのMBeanを操作してWASの管理を行います。wsadminコマンドの基本シンタックスは、上記のように管理オブジェクトを指定し、その管理オブジェクトに対して実行できるコマンドに必要な応じてパラメーターを指定します。管理オブジェクトは管理対象のMBeanをカテゴライズしたもので、構成データの操作を行う『AdminConfig』、エンタープライズ・アプリケーション関連操作をまとめた『AdminApp』、管理コマンドを実行するための『AdminTask』、スタート/ストップなどオペレーション操作『AdminControl』、ヘルプおよびMBean情報の表示を行う『Help』の計5種類があります。

wsadminの例として、ノードのホスト名を変更するAdminTaskコマンドを記載いたします。

wsadminの詳細については、下記のInformation Center の記載を参照ください。

WASV8 Information Center 「wsadminスクリプトの使用」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/txml_scriptingep.html

Designing Web System Infrastructure with WAS V8.0

WAS管理ツール – コマンド行ツール

- WASオペレーションの一部をシェル(バッチ)スクリプト化したツール
 - ◆ 起動停止など
 - ◆ ジョブ・マネージャーの起動(停止)

start(stop)Server.sh(bat) jobmgr
 - ◆ 管理エージェントの起動(停止)

start(stop)Server.sh(bat) adminagent
- wsadminとの相違点
 - ◆ ローカルでのみ実行可能
 - ◆ 各コマンドの実行ログが取得される
- コマンド行ツールのセキュリティ
 - ◆ コマンドによってユーザーID/
パスワードの認証が必要
 - stopServer、serverStatus、など

《WAS V7のコマンド行ツール》
 manageprofiles
 startServer
 stopServer
 startManager
 stopManager
 startNode
 stopNode
 addNode
 serverStatus
 cleanupNode
 syncNode
 backupConfig
 restoreConfig
 EARExpander
 GenPluginCfg
 versionInfo
 genVersionReport
 historyInfo
 genHistoryReport ...etc

© 2012 IBM Corporation
124

コマンド行ツールは、WAS関連のオペレーションの一部をシェル(バッチ)スクリプト化したツールです。アプリケーション・サーバーの起動停止などのコマンドが準備されています。wsadminとは異なり、ローカルのプロファイル上のアプリケーション・サーバーなどのプロセスに対して、操作を実行することができます。また、プロセスが必ずしも起動している必要はありません。

コマンド行ツールのコマンドは、`<WAS_ROOT>/bin`ディレクトリーまたは
`<WAS_PROFILE_ROOT>/bin`ディレクトリー以下に配置されています。各コマンドは、実行のたびに、下記のファイルにログが記録されます。

`<WAS_PROFILE_ROOT>/logs/<command_name>.log`

`<WAS_PROFILE_ROOT>/logs/<SERVER_NAME>/<command_name>.log`

このログは、追記形式であるため、少しずつですがサイズが大きくなりますので、ご注意ください。

管理セキュリティをONにしている場合は、wsadminのときと同様に、ユーザーIDとパスワードによる認証が必要なコマンドがあります。引数に`-username`、`-password`オプションを指定するか、`soap.client.props`プロパティファイルに、セキュリティONの設定とユーザーID、パスワードを指定する必要があります。

コマンド行ツールの詳細については、下記のInformation Center の記載を参照ください。

WASV8 Information Center 「コマンド行ツールの使用」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/txml_command.html

Designing Web System Infrastructure with WAS V8.0

PFBCT – コマンド・オプションとプロパティー

- PFBCTはAdminTaskオブジェクトのコマンド群
- PFBCTでサポートされるプロパティー

AdminService	JDBCProvider	SIPContainer
Application	JMSProvider	SchedulerProvider
ApplicationServer	JavaVirtualMachine	Security
AuthorizationGroup	Library	Server
AuthorizationTableExt	MailProvider	ServerCluster
Cell	NameServer	TPVService
CoreGroup	Node	TimerManagerProvider
CoreGroupBridgeService	NodeGroup	TransactionService
DynamicCache	ObjectPoolProvider	URLProvider
EJBContainer	ObjectRequestBroker	VariableMap
EventInfrastructureProvider	PMEServerExtension	VirtualHost
EventInfrastructureService	PMIModule	WebContainer
HAManagerService	PMIService	WebserverPluginSettings
J2CResourceAdapter	PortletContainer	WorkManagerProvider

© 2012 IBM Corporation
125

PFBCTはAdminTaskオブジェクトのコマンド群です。

各コマンド・オプションの詳細は以下のInformation Center の記載をご参照ください。

WASV8 Information Center 「wsadmin スクリプトを使用する AdminTask オブジェクトの PropertiesBasedConfiguration コマンド・グループ」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_7propbasedconfig.html

PFBCTでサポートされるプロパティーは上記の通りです。プロパティー・ファイルには、指定したオブジェクトおよびそのオブジェクトにネストされたオブジェクトのすべてのプロパティーが抽出されます。例えば、サーバーのプロパティーが抽出される際には、サーバー、アプリケーション・サーバー、Webコンテナ、EJBコンテナなどのオブジェクトのプロパティーも含まれます。プロパティー・ファイルにはセクションがあり、各セクションが一つのオブジェクトのプロパティー群を保持します。ネストされたオブジェクトのプロパティーはサブ・セクションにリストされます。

Designing Web System Infrastructure with WAS V8.0

PFBCTの使用例①

- server1の構成をカスタマイズ
 - ◆ server1の既存構成からプロパティ・ファイル(config.props)を作成

wsadmin> AdminTask.extractConfigProperties(' -propertiesFileName config.props -configData Server=server1')
 - ◆ テキスト・エディターを使ってプロパティを変更
 - ◆ 変更したプロパティをserver1の構成に適用

wsadmin> AdminTask.applyConfigProperties('[-propertiesFileName config.props]')
 - ◆ 構成の保管

wsadmin> AdminConfig.save()

© 2012 IBM Corporation
126

こちらは最も基本的なPFBCTの使い方です。

初めにAdminTask.extractConfigPropertiesを使用してserver1の構成からプロパティ値を抽出し、config.propsという名前のプロパティ・ファイルを作成します。プロパティ・ファイルを作成したらテキスト・エディターで該当のプロパティ値を変更します。そしてAdminTask.applyConfigPropertiesを使用してプロパティ・ファイルをserver1に適用します。最後に構成を保管すると、server1の構成ファイルにプロパティ値の変更が反映されます。

Designing Web System Infrastructure with WAS V8.0

PFBCTの使用例②

- VariableMapを使用してserver1の構成をカスタマイズ
 - ◆ server1の既存構成からプロパティ・ファイル(config.props)を作成
 - ◆ ファイル内の変更するプロパティ値を変数に変更

▼ 変更前プロパティ・ファイル

```
initialHeapSize=64
:
maximumHeapSize=128
:
```

➡

▼ 変更後プロパティ・ファイル

```
initialHeapSize=!{initialHeapSize}
:
maximumHeapSize=!{maximumHeapSize}
:
```
 - ◆ VariableMap (variableMap.vars)を作成して変更後のプロパティ値を記入

▼ VariableMap

```
initialHeapSize=128
maximumHeapSize=256
```
 - ◆ プロパティ・ファイルをVariableMapを使用してserver1の構成に適用


```
wsadmin> AdminTask.applyConfigProperties( '[--propertiesFileName
config.props --variablesMapFileName variableMap.vars] '
```
 - ◆ 構成の保管


```
wsadmin> AdminConfig.save()
```

© 2012 IBM Corporation
127

こちらはserver1のプロパティ・ファイルのプロパティ値を直接変更するのではなく、VariableMapを使用して変更する手順です。

初めにAdminTask.extractConfigPropertiesを使用してserver1の構成からプロパティ値を抽出し、config.propsという名前のプロパティ・ファイルを作成します。プロパティ・ファイルを作成したらテキスト・エディターで該当のプロパティ値を「!{プロパティ名}」の形で変数化します。そして、VariableMap (variableMap.vars)を作成して、該当のプロパティに対して変更後の値を記入します。最後に、AdminTask.applyConfigPropertiesでVariableMapを使用してプロパティ・ファイルをserver1に適用し、構成を保管すると、server1の構成ファイルにプロパティ値の変更が反映されます。

Designing Web System Infrastructure with WAS V8.0

ジョブ・マネージャー – 管理機能

- ジョブの非同期実行
 - ◆ 指定した時刻に実行
 - ◆ 特定の時間を越えたタイミングでの無効化
 - ◆ 特定の時間間隔での繰り返し実行
 - ◆ ジョブの完了通知の送信 (e-mail)
- 以下のジョブに関して柔軟な管理機能を提供
 - ◆ アプリケーション・サーバーの始動・停止・作成・削除
 - ◆ クラスター(メンバー)の始動・停止・作成・削除
 - ◆ アプリケーションのインストール・アンインストール・更新・始動・停止
 - ◆ ファイルの配布・収集・除去
 - ◆ wsadminスクリプトの実行
 - ◆ プロキシ・サーバーの作成・削除
 - ◆ プロパティの構成
 - ◆ インベントリ
 - ◆ 状況
 - ◆ リモート・サーバーに対し、IIM / WASインストール・Fix適用(CIM機能)
 - ◆ プロファイルの作成・削除
- DM経由でジョブの実行が可能
- ステータスの確認機能
 - ◆ ジョブ実行の成功/失敗/未完了、各ステータスのノード数などの表示

© 2012 IBM Corporation
128

ジョブ・マネージャーが実行できるジョブとしては上記に記述されているものが全てです。この中で名前から推察しづらいもののみ解説いたします。プロパティの構成は、WASV7から登場した管理ツールであるプロパティ・ファイル・ベース構成ツールから作成するプロパティ・ファイルを、ジョブとして特定のノードに適用させることができます。インベントリではジョブ・マネージャーにキューイングされている特定のノードに対するジョブを表示します。状態では特定のノードでのジョブの実行結果を返します。

wsadminでのジョブの管理については、以下のInformation Center の記載をご参照ください。

WASV8 Information Center 「wsadmin スクリプトを使用する管理ジョブ・タイプ」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_7jobtypes.html

WASV8 Information Center 「wsadminスクリプトによる柔軟な管理環境内でのジョブの管理」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/txml_7flexiblemgmt.html

Designing Web System Infrastructure with WAS V8.0

ジョブ・マネージャー – 登録

- ジョブ・マネージャーには個々のサーバー・レベルではなくセル・レベルでRegister(登録)して認識させる
 - ◆ 管理コンソール/wsadminから登録

```
#wsadmin.sh -profileName Dmgr01 -lang jython
wsadmin>AdminTask.registerWithJobManager
('[-host <jobmgr_host> -port 9960
-managedNodeName <dmgr_node_name>]')
```

- セルがRegister(登録)されると、管理者はセル内のサーバー向け、またはデプロイメント・マネージャー向けにジョブ・マネージャーを介してジョブをキューイングすることが可能となる

129

ND環境をジョブ・マネージャーの管理対象にするにはデプロイメント・マネージャー・ノードを登録する必要があります。デプロイメント・マネージャー・ノードを登録する場合には、デプロイメント・マネージャーの管理コンソール、もしくはwsadminから行います。管理コンソールから行う場合には、ナビゲーションツリーの[システム管理]>[デプロイメント・マネージャー]>[ジョブ・マネージャー]をクリックし、「ジョブ・マネージャーに登録」をクリックしてください。登録を解除する場合には「ジョブ・マネージャーの登録抹消」をクリックしてください。wsadminから登録する場合にはAdminTaskオブジェクトのregisterWithJobManagerコマンドオプションを使用します。-portでは管理セキュリティを設定していない場合にはジョブ・マネージャーのWC_adminhost(デフォルト9960)を指定します。管理セキュリティを設定している場合にはWC_adminhost_secure(デフォルト9943)を指定します。またジョブ・マネージャーからデプロイメント・マネージャーの登録を解除するにはAdminTaskオブジェクトでunregisterWithJobManagerコマンドオプションを使用します。

コマンドオプションの引数の詳細などは以下のInformation Center の記載をご参照ください。

WASV8 Information Center 「AdminTask オブジェクトの ManagedNodeAgent コマンド・グループ」
http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_7managednodeagent.html

ジョブ・マネージャーにデプロイメント・マネージャー・ノードを登録すると、ジョブ・マネージャーはデプロイメント・マネージャーに向けてジョブを非同期に実行することが可能です。ただしジョブ・マネージャーは構成情報を管理するものではないため、各ノードの構成情報は持ちません。

ジョブ・マネージャー – ポーリング

- ジョブ・マネージャーの管理対象になるとデプロイメント・マネージャー/管理エージェントはジョブ・マネージャーに対して一定間隔でポーリングしジョブを確認
 - ◆ ポーリング間隔は管理コンソールで変更可能

デプロイメント・マネージャー管理コンソール画面のスクリーンショット。画面には「ポーリング間隔」のフィールドがあり、デフォルト値は30秒です。この間隔はデプロイメント・マネージャー管理コンソールで変更可能です。

▲ DMをジョブ・マネージャーに登録した場合

ジョブ・マネージャーに登録して管理対象になると、デプロイメント・マネージャーと管理エージェントはジョブ・マネージャーに対してジョブの有無を確認するために一定間隔でポーリングを行います。デフォルトではこの間隔は30秒です。この間隔はデプロイメント・マネージャー管理コンソールもしくは管理エージェント管理コンソールで変更可能です。

デプロイメント・マネージャー管理コンソールから変更する場合はナビゲーション・ツリーの[システム管理]>[デプロイメント・マネージャー]>[ジョブ・マネージャー]>[ジョブ・マネージャーUUID]をクリックし、ポーリング間隔を変更します。

管理エージェント管理コンソールから変更する場合は[システム管理]>[管理エージェント]>[ノード]>[ノード名]>[ジョブ・マネージャー]>[ジョブ・マネージャーUUID]をクリックし、変更します。

Designing Web System Infrastructure with WAS V8.0

管理エージェント – 登録

- 管理対象のノードを管理エージェントにRegister(登録)して管理
 - ◆ registerNodeスクリプトによってシングル・サーバー・セルを登録


```
#<ADMINAGENT_PROFILE_ROOT>/bin/registerNode.sh(bat) -profilePath
<WAS_ROOT>/profiles/AppSrv01 -host localhost -connntype SOAP -port 8877
```
- 管理エージェントの管理対象ノードをジョブ・マネージャーに登録することも可能
 - ◆ 管理エージェントからジョブ・マネージャーに対して管理エージェント配下のシングル・サーバー・セルをRegister(登録)して認識させる
 - ◆ 管理エージェントの管理対象ノードはジョブ・マネージャーの管理対象外にすることも可能
 - ◆ 管理エージェントノードを登録することはできない
 - ◆ 管理エージェントは、シングル・サーバー・ノードの代理としてジョブ・マネージャーからのジョブを取得する
 - ◆ 管理エージェント管理コンソール/wsadminからノードを登録


```
#wsadmin.sh -profileName AdminAgent -lang jython
wsadmin>AdminTask.registerWithJobManager ( '[-host <jobmgr_host> -port
9960 -managedNodeName <node_name>]')
```

© 2012 IBM Corporation
131

シングル・サーバー・ノードのプロファイルを管理エージェントに登録するにはregisterNodeスクリプトを使用します。registerNodeスクリプトは<ADMINAGENT_PROFILE_ROOT>/binディレクトリーにあります。管理エージェントからノードの登録を解除する場合にはderegisterNodeスクリプトを使用します。コマンドの引数の詳細などは以下のInformation Center の記載をご参照ください。

WASV8 Information Center 「registerNodeコマンド」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/ragt_registerNode.html

WASV8 Information Center 「deregisterNodeコマンド」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/ragt_deregisterNode.html

また、管理エージェントの管理対象ノードをジョブ・マネージャーに登録することで、管理エージェントは管理対象のシングル・サーバー・ノードの代理としてジョブ・マネージャーからジョブを受け取り、そのノードに対してジョブを実行することが可能です。管理エージェントノードはジョブ・マネージャーに登録できないのでご注意ください。管理エージェントの管理対象ノードをジョブ・マネージャーに登録するには管理コンソールもしくはwsadminから登録が可能です。wsadminで登録する場合にはAdminTaskオブジェクトのregisterWithJobMangerコマンドオプションを使用して登録します。登録を解除する際にはAdminTaskオブジェクトのunregisterWithJobManagerコマンドオプションを使用します。

Designing Web System Infrastructure with WAS V8.0

ND構成とジョブ・マネージャー/管理エージェント構成

アプリケーション・サーバー管理機能	NDセル	Base +JM +AA
リモート管理	√	√
コンソール経由でのFine-Grained管理	√	X
管理プロセスが停止した場合の管理(ローカル接続)	√	X
クラスタリング - ワークロード管理	√	X
クラスタリング - フェールオーバー	√	X
Webサーバー・プラグインに基づくHTTPリクエストのルーティング	√	X
データ複製(メモリー・ベースのセッション複製)	√	X
構成バックアップ/リストアの集中化	√	X
パフォーマンス・モニタリング	√	X
管理機能のスケジュール化	X	√

ジョブ・マネージャーを使用するにはNDのライセンスが必要

© 2012 IBM Corporation
132

上記は従来のNDセル構成でのアプリケーション・サーバー管理と、Base構成にJob ManagerとAdminAgentを加えたアプリケーション・サーバー管理を比較した表です。Job Managerで実現できる管理機能は限られているため、Base構成にJob ManagerとAdminAgentを加えてもNDセル構成と同様のサーバー構成・管理を行うことは出来ません。クラスタリングやデータ複製などサーバーの耐障害性を実現するためにはNDセル構成をとる必要があります。また、Job Managerを使用する場合にはNDのライセンスが必要になります。AdminAgentはBaseのライセンスで使用できます。

Designing Web System Infrastructure with WAS V8.0

3. 定常運用

- 3-1 起動停止
- 3-2 ログ運用**
- 3-3 自動化
- 3-4 定常運用に関する設計ポイント

133

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

WASのログ – ログの種類(ノード単位)

- 製品インストール・ログ
 - ◆ <WAS_ROOT>/logs/install ディレクトリー内のファイル群
 - ◆ 製品インストール時の状況が記録される
- 保守パッケージ適用ログ
 - ◆ <WAS_ROOT>/logs/update ディレクトリー内のファイル群
 - ◆ 保守パッケージ適用成否が記録される
- WASプロファイル作成ログ
 - ◆ <WAS_ROOT>/logs/manageprofiles ディレクトリー内のファイル群
 - ◆ プロファイル作成成否が記録される

定期的にファイル・サイズが増加するものではないので、クリーンアップの必要はない
問題判別のために必要になる可能性があるので、すべて保管しておくことを推奨

© 2012 IBM Corporation
134

ノード単位で作成されるログは、上記の通りです。インストール、Fix適用時やプロファイル作成時に記録されるログで、これらは定常的に増加するログではありません。定期的なメンテナンスは不要ですが、問題発生時に必要になる場合がありますので、すべて保管しておくことをお勧めします。

Designing Web System Infrastructure with WAS V8.0

WASのログ – ログの種類(プロファイル単位)

- IBM保守ログ
 - ◆ ログ・アナライザーもしくはshowlogツールで解析する(バイナリー・ファイル)
 - ◆ 最大サイズを設定可能(デフォルト2MB)
 - activity.log という名称で作成される
- コマンド実行ログ
 - ◆ コマンド実行時毎に追記される
 - <Command_name>.log という名称で作成される
- ffdcログ
 - ◆ ランタイムにて発生したイベント及びエラー情報が即時に収集され作成されるログ・ファイル
 - <AS_name>.Exception.log
 - <AS_name>_<thread>_<date>_<timestamp>_<sequence>.log
 - ◆ 保持期間を指定できる
 - ffdcRun.propertiesファイルのExceptionFileMaximumAgeプロパティに指定する
- wsadmin関連ログ
 - ◆ wsadminの処理内容・実行結果・Exception情報等が出力される
 - wsadmin.traceout
 - wsadmin.valout
 - ◆ wsadminの新規のセッションが開始されるごとにリフレッシュされる

© 2012 IBM Corporation
135

プロファイル単位で作成されるログは、上記の通りです。IBM保守ログ、ffdcログは、循環ログですので、必ずしもメンテナンスの必要はありません。

Designing Web System Infrastructure with WAS V8.0

WASのログ – ログの種類(JVM単位)

- JVMログ
 - ◆ アプリケーション・サーバーの標準出力、標準エラー出力
 - SystemOut.log (JVM標準出力ログ)
 - 出力頻度が高く、サイズも大きくなる傾向
 - デフォルトでは1MB、バックアップ1世代の循環ログ
 - SystemErr.log (JVM標準エラー出力ログ)
 - デフォルトでは1MB、バックアップ1世代の循環ログ
 - ログ・ローテート機能が提供されている
 - 時間指定で循環することも可能
- プロセスログ
 - ◆ ネイティブ・モジュールの標準出力、標準エラー出力
 - native_stdout.log
 - 通常は何も出力されない
 - native_stderr.log
 - 冗長GC (verbose:gc) のデフォルトの出力先ファイル
- 診断トレース
 - ◆ trace.log という名称のログ・ファイルが作成される
 - ◆ ログ・ローテート機能が提供されている
- コマンド実行ログ
 - ◆ コマンド実行時毎に追記される
 - <Command_name>.log という名称で作成される
- トランザクション・ログ
 - ◆ グローバル・トランザクションの実行内容が記録されるログ・ファイル
 - ◆ グローバル・トランザクションのリカバリー時に使用される
 - ◆ ファイルサイズはデフォルトで1MB × 4 = 4MB (一定サイズ)
 - アプリケーション・サーバー起動時にこのサイズが確保される

© 2012 IBM Corporation
136

JVM単位で作成されるログは、上記の通りです。JVMログは、障害発生時にまず確認するログになります。循環機能が提供されており、ファイルサイズや保管世代数を変更することができます。プロセスログは、通常は、定期的にログが記録されるわけではありませんが、冗長ガーベッジ・コレクションのデフォルトの出力先となりますので、冗長ガーベッジ・コレクションのログを出力している場合には、ログの管理を必ず実施する必要があります。診断トレースも、通常は出力はありませんが、問題判別のためにトレース設定を行った場合には、多くのログが出力されます。ログローテート機能が提供されていますので、問題判別に十分なファイルサイズと保管世代数を設定する必要があります。ログの設定は、管理コンソールで、[トラブルシューティング]-[ログおよびトレース]-[process_name]-[診断トレース]で設定します。コマンド実行ログは、コマンド実行ごとに追記されます。トランザクション・ログは、WASのアプリケーション・サーバーがトランザクション・マネージャーとなる場合に、そのトランザクションを管理するためにWASが使用するログです。基本的には、ユーザーが管理する必要はありません。

Secure Proxyのログ

- プロキシ・アクセス・ログ(proxy.log)
 - ◆ Proxy Serverに届いたリクエストの転送処理を記録
- ローカル・アクセス・ログ(local.log)
 - ◆ 転送対象でないリクエストに対するProxy Serverの処理を記録
- キャッシュ・アクセス・ログ(cache.log)
 - ◆ キャッシュにヒットした際に、そのアクセス情報を記録
- JVMログ(SystemOut.log、SystemErr.log)
 - ◆ JVMの標準出力、標準エラー出力を記録し、問題判別で最も重要
- プロセス・ログ(native_stdout.log、native_stderr.log)
 - ◆ JVM単位に出力され、JVM自身を含むネイティブ・コードの標準出力、標準エラー出力を記録
- 診断トレース(trace.log)
 - ◆ JVM単位に出力され、Secure Proxyコンポーネントの処理の詳細を記録

Designing Web System Infrastructure with WAS V8.0

3. 定常運用

- 3-1 起動停止
- 3-2 ログ運用
- 3-3 自動化**
- 3-4 定常運用に関する設計ポイント

138

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

コマンド記録 – ログに記録

■ ログに記録(例)

◆ 出力先 :

< DM_PROFILE_ROOT >/logs/<DMサーバ名>/commandAssistanceJythonCommands_<ユーザ名>.log

アプリケーションのデプロイ

```
# [07/07/17 12:44:47:062 JST] エンタープライズ・アプリケーション > エンタープライズ・アプリケーション
AdminApp install("C:\EntSysSampPro\ear", [ -noreCompileJSPs -distributeApp -nouseMetaDataFromBinary -nodeployejb -appname EntSysSampPro -
createMBeansForResources -noreloadEnabled -nodeployws -validateinstall warn -processEmbeddedConfig -
filepermission "%dl=755#,%so=755#,%a=755#,%sl=755 -noallowDispatchRemoteInclude -noallowServiceRemoteInclude" ])

# [07/07/17 12:45:08:703 JST] エンタープライズ・アプリケーション > エンタープライズ・アプリケーション > 保管
AdminConfig.save()
```

データソースの作成

```
# [07/07/18 22:15:17:468 JST] JDBC プロバイダー > DB2 Universal JDBC Driver Provider > データ・ソース
AdminTask.createDataSource("DB2 Universal JDBC Driver Provider(cells/61Cell01/nodes/AA308866161Node01/resources.xml#JDBCProvider_1184764413046)",
["-name sampleDS -jndiName jdbc/sampleDS -dataStoreHelperClassName com.ibm.websphere.rsadapter.DB2UniversalDataSourceHelper -
componentManagedAuthenticationAlias (なし) -configureResourceProperties [(databaseName java.lang.String sample) (driverType java.lang.Integer 4)
(serverName java.lang.String MyServer) (portNumber java.lang.Integer 50000)]])

# [07/07/18 22:15:18:453 JST] JDBC プロバイダー > DB2 Universal JDBC Driver Provider > データ・ソース
AdminConfig.list("DataSource", AdminConfig.getid( "/Cell:61Cell01/Node:AA308866161Node01/JDBCProvider:DB2 Universal JDBC Driver Provider/"))

# [07/07/18 22:22:13:531 JST] JDBC プロバイダー > DB2 Universal JDBC Driver Provider > データ・ソース
AdminConfig.save()
```

© 2012 IBM Corporation

139

管理コンソールの操作のwsadminコマンドをログに記録した例です。上記では、アプリケーションのデプロイとデータソースの作成を行った場合を抜粋しています。操作を実行した時間と管理コンソール上の画面遷移のルートも記録されます。生成したコマンドはJython形式ですので、実行する場合には、wsadminの-lang jythonオプションを付与する必要があります。また、-cオプションを使用したコマンド実行モードで実行する場合には、生成したコマンドを"(ダブルコーテーション)で囲んでコマンドを記述します。-fオプションを使用したスクリプト実行モードで実行する場合、スクリプト・ファイル名は任意で構いませんが、Jythonの場合は、Pythonで使われる.pyを使用すると良いです。

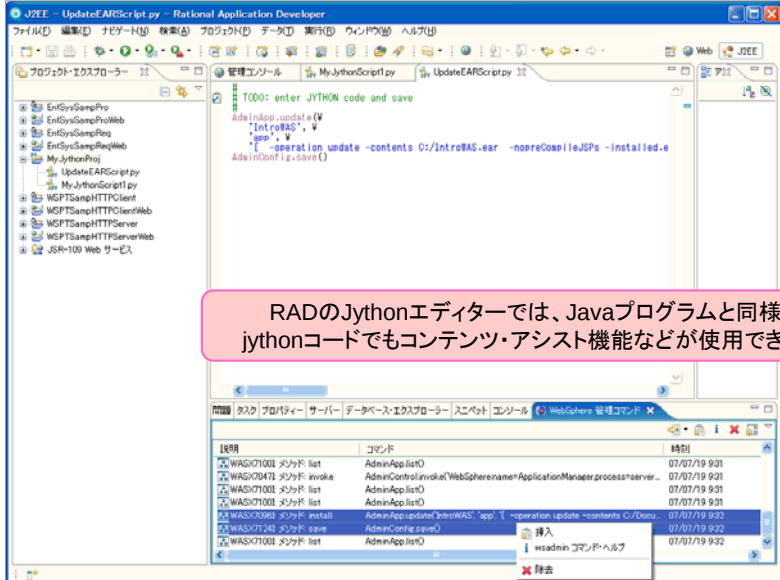
WAS V8.0 によるWebシステム基盤設計ワークショップ

139

Designing Web System Infrastructure with WAS V8.0

コマンド記録 – RADへ送信

- RADへ送信(例)



RADのJythonエディターでは、Javaプログラムと同様にjythonコードでもコンテンツ・アシスト機能などが使用でき便利

説明	コマンド	時刻
WAS071008 スケジュール	AdminApp list	07/07/19 9:21
WAS070472 スケジュール	AdminControl invoke	07/07/19 9:21
WAS071008 スケジュール	AdminApp list	07/07/19 9:21
WAS071008 スケジュール	AdminApp list	07/07/19 9:21
WAS071008 スケジュール	AdminApp update	07/07/19 9:22
WAS071241 スケジュール	AdminApp save	07/07/19 9:22
WAS071008 スケジュール	AdminApp list	07/07/19 9:22

© 2012 IBM Corporation

140

上記は、管理コンソール操作のwsadminコマンドをRADに送信した例です。上記では、アプリケーションの更新のJythonスクリプトを取り込んだコマンドをもとに作成しています。RADに管理コンソールの操作のwsadminコマンドを取り込むには、取り込み対象のサーバー側で、「コマンド・アシスタンス通知を使用可能にする」のチェックボックスをONにします。次に、RAD側で、取り込む対象のサーバーを登録(RADのテストサーバーとしてすでに登録してある場合には不要)し、「WebSphere管理コマンド支援」のウィンドウで取り込む先のサーバーを指定することで、ログに記録すると同様に、実行したコマンドが「WebSphere管理コマンド」ウィンドウに順次登録されます。RAD上で、Jythonプロジェクト、Jythonスクリプト・ファイルを作成し、Jythonスクリプト・ファイルを起動するとJythonエディター上で編集できます。Jythonエディター上のファイルに、取り込んだwsadminコマンドを挿入することができます。また、Jythonエディターでは、javaのプログラムと同様に、jythonコードでも[Ctrl]+[Space]キーで、コードを補完するコンテンツ・アシスト機能などが使用できますので、テキスト・エディターで編集するよりも、容易にJythonスクリプトファイルを作成することが可能です。また、RADのローカルのサーバーであれば、テストを行うこともできます。

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

4-1 バックアップ・リストア

4-2 監視

4-3 障害時運用

4-4 片系運用

4-5 Fix Pack/Fix適用

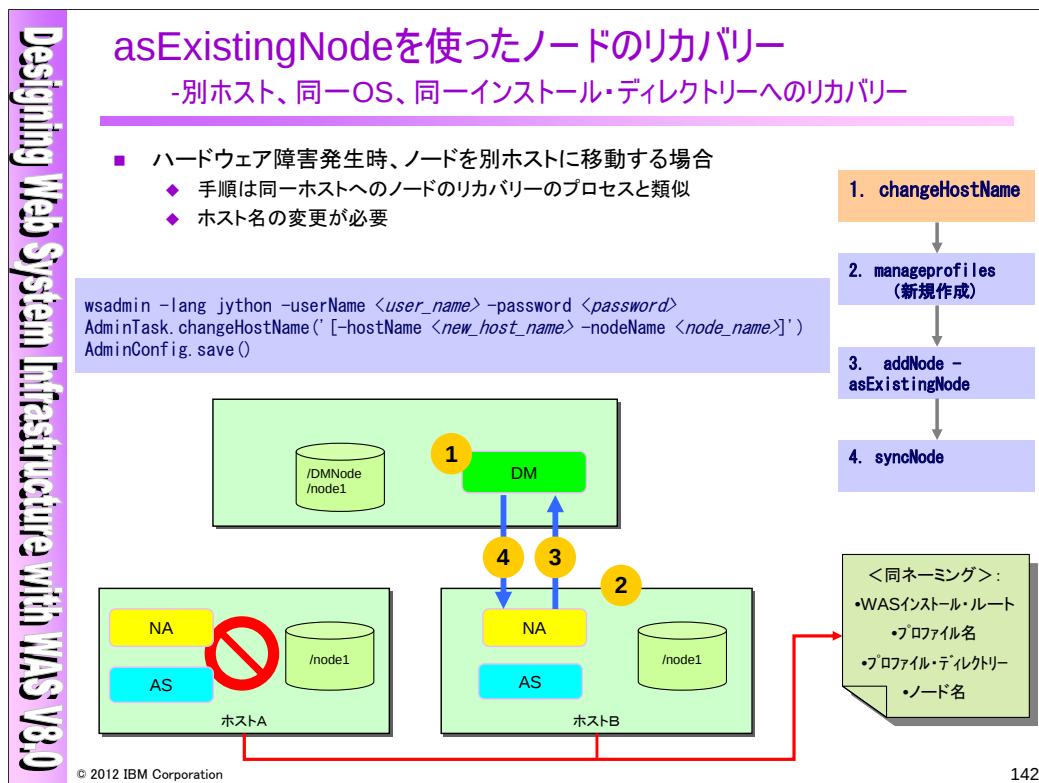
4-6 アプリケーション管理・更新

4-7 リリース・パターン

4-8 非定期運用に関する設計ポイント

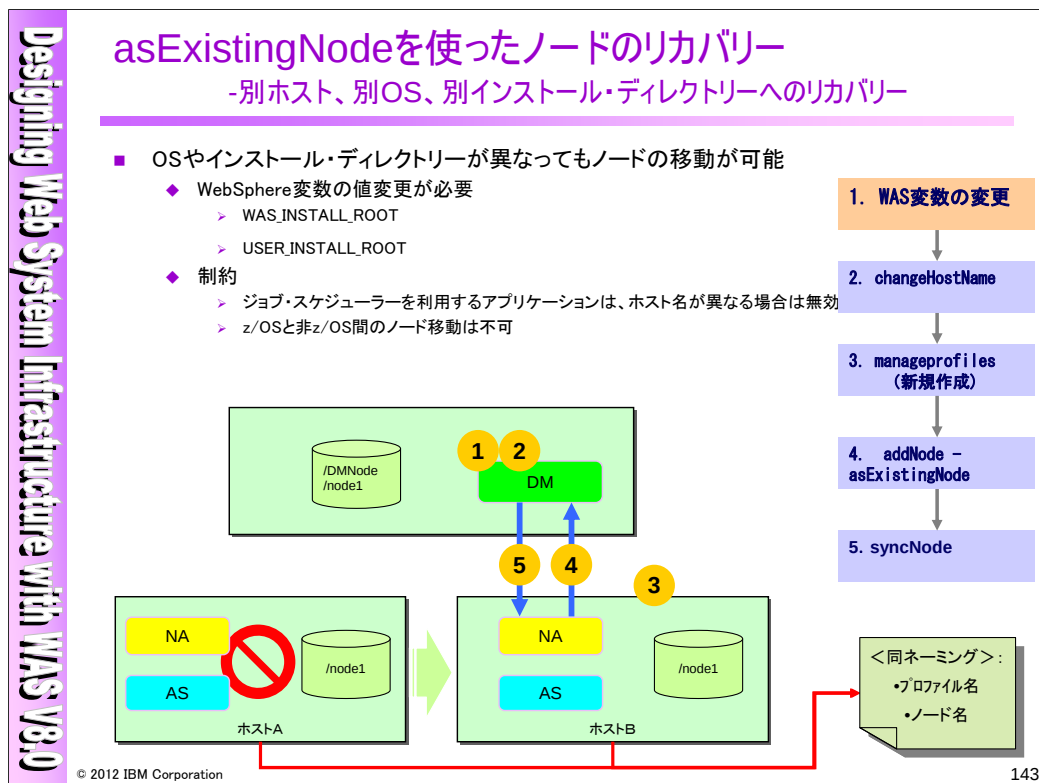
141

© 2012 IBM Corporation



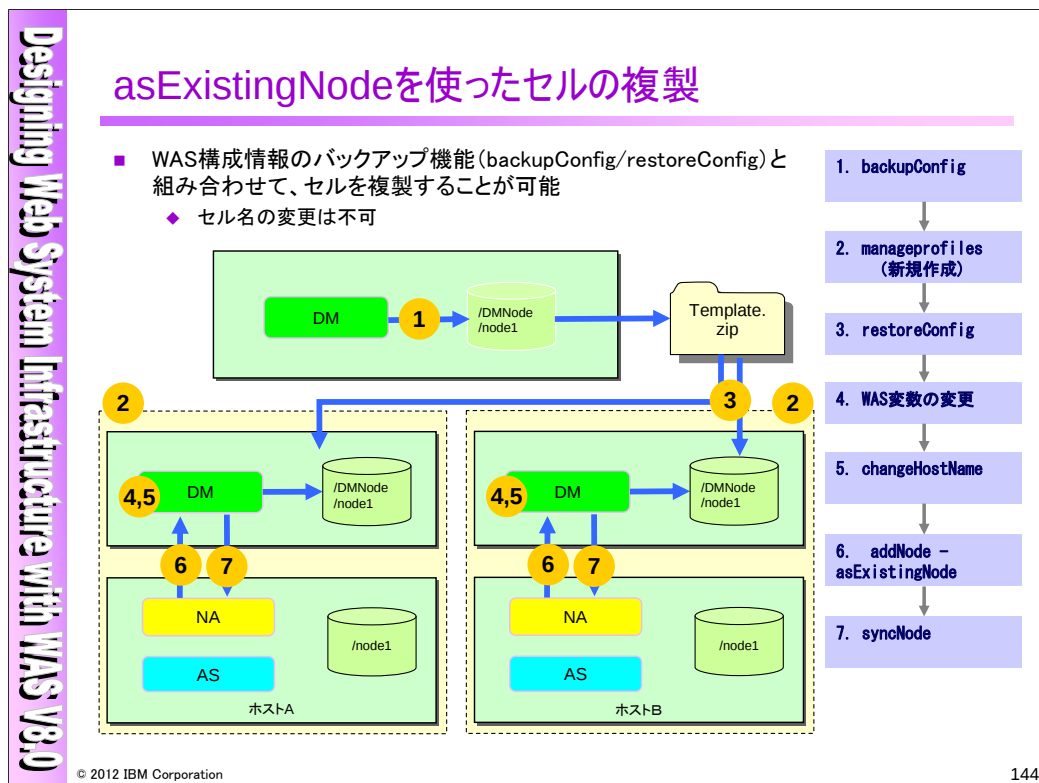
ノードを別ホストに移動する場合の手順は同一ホスト上でのノードのリカバリー手順と類似します。ノード・エージェントとアプリケーション・サーバーのプロセスを停止してから、以下の手順を実施します。

- デプロイメント・マネージャーからAdminTask.changeHostNameコマンドを実行し、ノードのホスト名を変更
- 新しいホストにWASをインストールし、新規プロファイルを作成。WASインストール・ディレクトリー、プロファイル名、プロファイル・ディレクトリー名、ノード名は、もとのノードと同様にします。
- 新規プロファイル配下の「bin」ディレクトリーで、addNodeコマンドを「asExistingNode」オプション付きで発行し、新規ノードをデプロイメント・マネージャー管理配下に統合
- ノードの同期化(syncNodeコマンド)



OSやWASのインストール・ディレクトリーが障害ノードと異なっても、ノードの移動が可能です。ノード・エージェントとアプリケーション・サーバーのプロセスを停止してから、以下の手順を実施します。

1. 管理コンソールから「環境」>「WebSphere変数」の順にクリックし、「WAS_INSTALL_ROOT」と「USER_INSTALL_ROOT」の値を変更。
2. デプロイメント・マネージャーからAdminTask.changeHostNameコマンドを実行し、ノードのホスト名を変更。
3. 新しいホストにWASをインストールし、新規プロファイルを作成。WASインストール・ディレクトリー、プロファイル名、プロファイル・ディレクトリー名、ノード名は、もとのノードと同様にします。
4. 新規プロファイル配下の「bin」ディレクトリーで、addNodeコマンドを「asExistingNode」オプション付きで発行し、新規ノードをデプロイメント・マネージャー管理配下に統合
5. ノードの同期化(syncNodeコマンド)



144

asExistingNodeオプションのもう1つの使い方は、セルの複製です。

セルの複製は以下の手順で実施可能です。

1. 複製元のセル情報をbackupConfigコマンドを使用してバックアップを取得。
2. 複製先のマシン にWASをインストールし、新規プロファイル(デプロイメント・マネージャー、アプリケーション・サーバー・ノード)を作成。
3. バックアップから複製先のデプロイメント・マネージャーの構成情報をリストア(restoreConfigコマンド)
4. 管理コンソールから「 WAS_INSTALL_ROOT 」と「 USER_INSTALL_ROOT 」の値を変更。
5. デプロイメント・マネージャーからAdminTask.changeHostNameコマンドを実行し、ノードのホスト名を変更。
6. 新規プロファイル配下の「bin」ディレクトリーで、addNodeコマンドを「asExistingNode」オプション付きで発行し、新規ノードをデプロイメント・マネージャー管理配下に統合
7. ノードの同期化(syncNodeコマンド)

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

4-1 バックアップ・リストア

4-2 監視

4-3 障害時運用

4-4 片系運用

4-5 Fix Pack/Fix適用

4-6 アプリケーション管理・更新

4-7 リリース・パターン

4-8 非定期運用に関する設計ポイント

145

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

OSコマンドによるプロセス監視 – UNIX系OS

■ psコマンド(UNIX系OS)

- ◆ アプリケーション・サーバー名でgrepすることで特定プロセスの存在確認が可能

```
# ps -ef | grep server1 | grep -v grep
root    27885    1 41 20:33 pts/2    00:03:14 /opt/IBM/WebSphereV70/AppServer/java/bin/java -Declipse.security -
Dwas.status.socket=57323 -Dosgi.install.area=/opt/IBM/WebSphereV70/AppServer -
Dosgi.configuration.area=/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/configuration -Djava.awt.headless=true -
Dosgi.framework.extensions=com.ibm.cds.com.ibm.ws.eclipse.adaptors -
Xshareclasses=name=webspherev70_%g.groupAccess,nonFatal -Xscmx50M -
Xbootclasspath:/opt/IBM/WebSphereV70/AppServer/java/jre/lib/ext/ibmorb.jar:/opt/IBM/WebSphereV70/AppServer/java/jre/lib/ext/ib
mext.jar -classpath
/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/properties:/opt/IBM/WebSphereV70/AppServer/properties:/opt/IBM/WebSphere
V70/AppServer/lib/startup.jar:/opt/IBM/WebSphereV70/AppServer/lib/bootstrap.jar:/opt/IBM/WebSphereV70/AppServer/lib/jsf-
nls.jar:/opt/IBM/WebSphereV70/AppServer/lib/improxy.jar:/opt/IBM/WebSphereV70/AppServer/lib/urlprotocols.jar:/opt/IBM/WebSphere
V70/AppServer/deploytool/itp/batchboot.jar:/opt/IBM/WebSphereV70/AppServer/deploytool/itp/batch2.jar:/opt/IBM/WebSphereV70/Ap
pServer/java/lib/tools.jar -Dibm.websphere.internalClassAccessMode=allow -Xms50m -Xmx256m -
Dws.ext.dirs=/opt/IBM/WebSphereV70/AppServer/java/lib:/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/classes:/opt/IBM/We
bSphereV70/AppServer/classes:/opt/IBM/WebSphereV70/AppServer/lib:/opt/IBM/WebSphereV70/AppServer/installedChannels:/opt/IB
M/WebSphereV70/AppServer/lib/ext:/opt/IBM/WebSphereV70/AppServer/web/help:/opt/IBM/WebSphereV70/AppServer/deploytool/itp/l
ugins/com.ibm.etools.ejbdeploy/runtime -Dderby.system.home=/opt/IBM/WebSphereV70/AppServer/derby -
Duser.install.root=/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01 -
Djava.management.builder.initial=com.ibm.ws.management.PlatformMBeanServerBuilder -
Dwas.install.root=/opt/IBM/WebSphereV70/AppServer -
Dpython.cachedir=/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/temp/cachedir -
Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager -
Dserver.root=/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01 -Dcom.ibm.security.jgss.debug=off -
Dcom.ibm.security.krb5.Krb5Debug=off -
Djava.security.auth.login.config=/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/properties/wsjaas.conf -
Djava.security.policy=/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/properties/server.policy
com.ibm.wsspi.bootstrap.WSPreLauncher -nosplash -application com.ibm.ws.bootstrap.WSLauncher com.ibm.ws.runtime.WsServer
/opt/IBM/WebSphereV70/AppServer/profiles/AppSrv01/config/satsukiCell02/satsukiNode02/server1
```

© 2012 IBM Corporation
146

上記はpsコマンドでjavaプロセスの状況を確認した例です。-fオプションで、完全表示を行っているため、非常に長くなりますが、アプリケーション・サーバー名も含まれますので、アプリケーション・サーバー名をキーにしてgrepすることで、プロセスの稼働を確認することができます。

Designing Web System Infrastructure with WAS V8.0

OSコマンドによるプロセス監視 – Windows

- tasklistコマンド(Windows2003以上)
 - ◆ 複数のjavaプロセスが起動していると見分けることができないためプロセスIDでフィルターするなどの工夫が必要
 - プロセスIDはpidファイルで確認可能
`<PROFILE_ROOT>%logs%\<SERVER_NAME>%\<SERVER_NAME>.pid`

```
D:\>tasklist /FI "PID eq 3284"
```

イメージ名	PID	セッション名	セッション#	メモリ使用量
java.exe	3284	Console	0	298,216 K

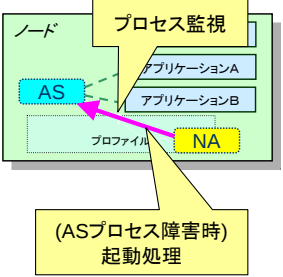
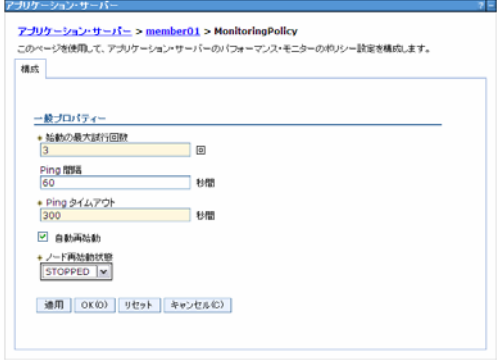
© 2012 IBM Corporation
147

上記はWindows上で、tasklistコマンドにより、javaプロセスの監視を行った例です。複数のjavaプロセスが起動している場合は各プロセスを見分けることができないため、プロセスIDでフィルターするなどの工夫が必要になります。各javaプロセスIDはpidファイルで確認することができます。

Designing Web System Infrastructure with WAS V8.0

ノード・エージェントによるプロセス監視

- ノード・エージェントのプロセス監視機能
 - ◆ アプリケーション・サーバー・プロセスを監視
 - アプリケーション・サーバーが停止操作無しで停止した場合に再起動を試行
 - デフォルト: ON

148

WASの機能として、ND構成の場合、ノード・エージェントがそのノード上のアプリケーション・サーバーをモニタリングし、応答がない場合には、自動的に再起動を行います。この機能はデフォルトでONですが、OFFに変更することも可能です。WASはノード・エージェントを監視する機能が提供していませんので、もしノード・エージェントがダウンした場合に、自動的に再起動が必要な場合には、UNIX/Linuxの場合、inittabにaction:respawnの指定を行ったり、Windowsの場合は、サービス登録を行いダウン時のアクションを設定することで、ダウンした場合に自動的に再起動を行うように設定することもできます。ただ、通常、ノード・エージェントやデプロイメント・マネージャーは、ユーザー・サービスに影響しませんので、必ずしも再起動が必要なわけではありません。

Designing Web System Infrastructure with WAS V8.0

ログ監視 – WASのログ・フォーマット

■ WAS SystemOut.log/SystemErr.logのフォーマット

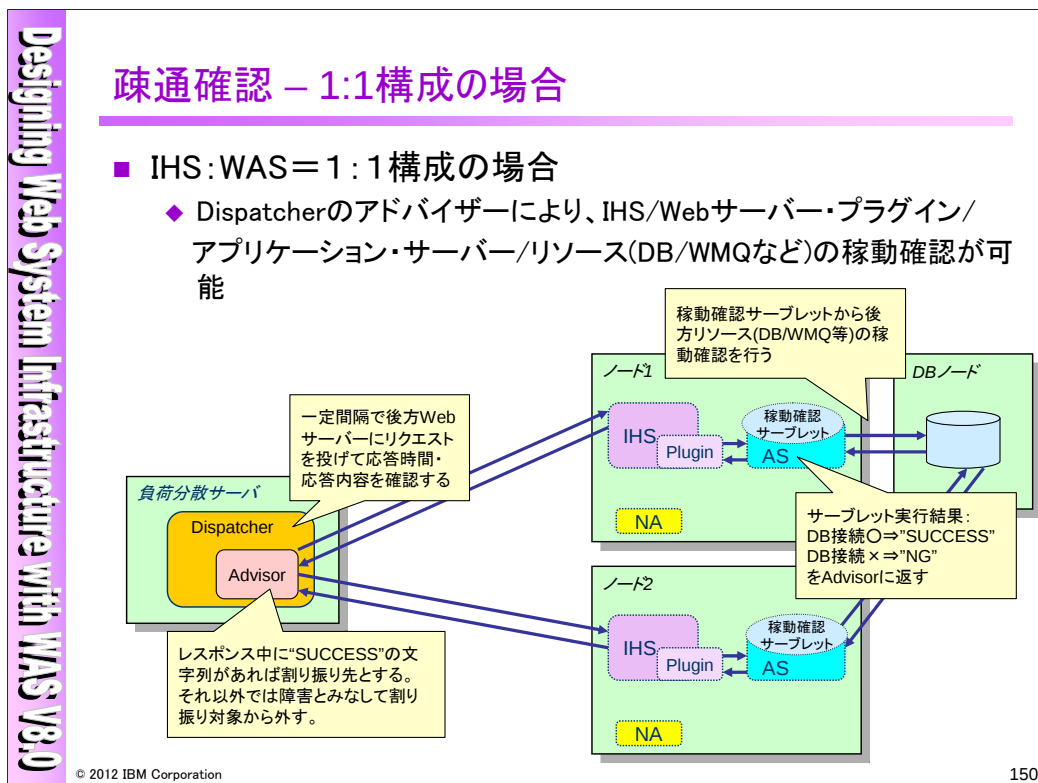
TimeStamp	ThreadId	Threadname	ShortName	LongName	EventType	ClassName	MethodName
F : 致命的メッセージ E : エラー・メッセージ W : 警告メッセージ A : 監査メッセージ I : 通知メッセージ C : 構成メッセージ D : 詳細メッセージ O : ユーザー・アプリケーション/内部コンポーネントにより、System.out に直接書き込まれたメッセージ R : ユーザー・アプリケーション/内部コンポーネントにより、System.err に直接書き込まれたメッセージ Z : タイプが認識されなかったことを示すプレースホルダー							

《実例》

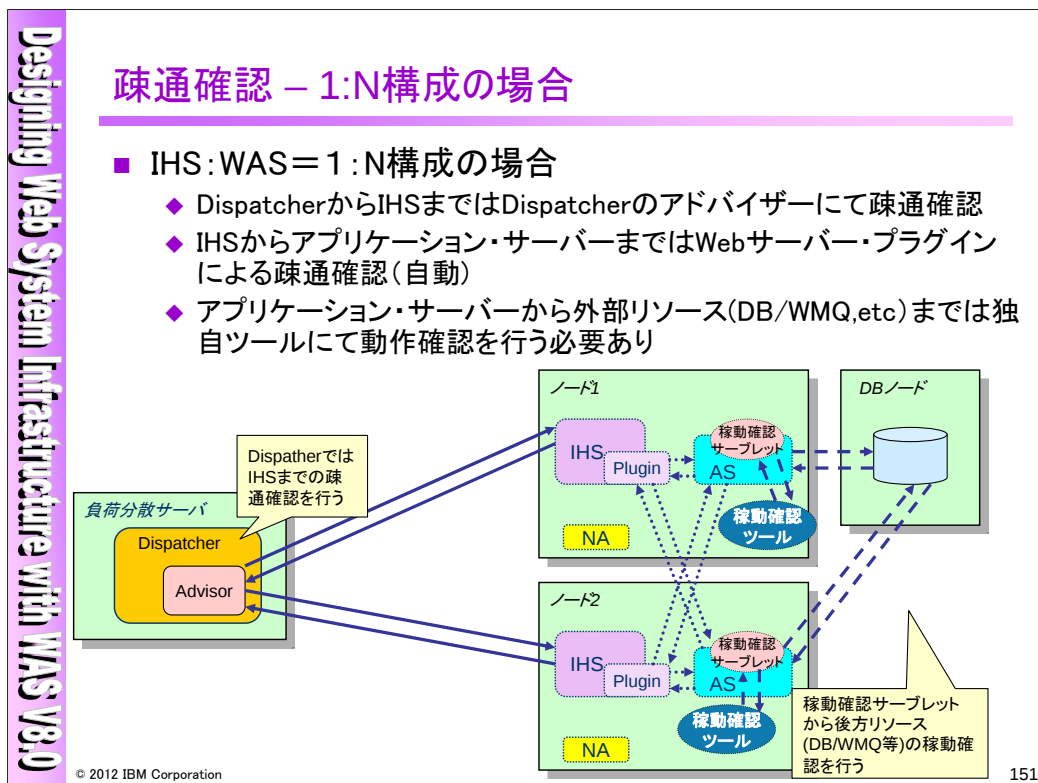
[05/11/13 0:04:18:406 JST] 00000046 CoordinatorCo W HMGR0152W:
CPU 欠乏が検出されました。現行スレッドのスケジューリング遅延は 261 秒です。

© 2012 IBM Corporation
149

WASのSystemOut.log/SystemErr.logのフォーマットは上記のようになります。EventTypeの文字により、そのログの重要度を判別することができます。



IHSとWASが1対1構成の場合の例です。2台のIHSに対して、監視を行うことで1:1で関連付けられているWASまでの稼動確認を行うことができます。



151

IHSとWASが1対N構成の場合の例です。2台のIHSに対して監視を行っても、IHSが2台のWASサーバーのうち正常なサーバーにリクエストの割り振りを行いますので、WASサーバーに障害が発生した場合も検知を行うことができません。WASサーバーについては別途何らかのツールを使用して監視をする必要があります。Dispatcherから直接WASがListenする9080ポートへのリクエストを送信し監視する方法もあります。

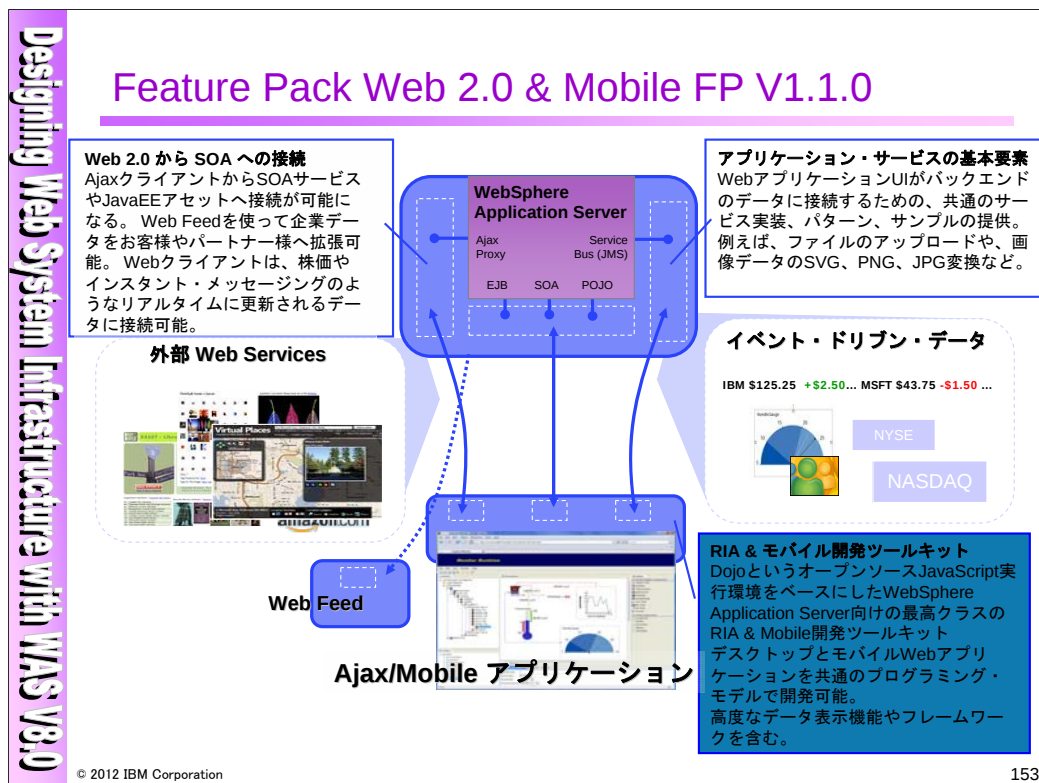
Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用**
- 4-6 アプリケーション管理・更新
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

152

© 2012 IBM Corporation



153

Web 2.0 & Mobile FP (WebSphere Application Server Feature Pack for Web 2.0 and Mobile) V1.1.0は、WASにWeb 2.0とモバイルの機能を追加するFeature Packの最新版です。以前は、Feature Pack for Web 2.0 V1.0.1として公開されていたFeature Packの名称が変更になり、バージョンアップしたものです。その特徴は、次の3点です。

- Web 2.0の機能を使用してクライアントからSOAに容易に接続可能にします。
- REST要求を受け付けるアプリケーション・サービスの機能やサンプルを提供します。
- RIAやモバイルのアプリケーションを開発するためのツールキットを提供します。

Designing Web System Infrastructure with WAS V8.0

4. 非定期運用

- 4-1 バックアップ・リストア
- 4-2 監視
- 4-3 障害時運用
- 4-4 片系運用
- 4-5 Fix Pack/Fix適用
- 4-6 アプリケーション管理・更新**
- 4-7 リリース・パターン
- 4-8 非定期運用に関する設計ポイント

154

© 2012 IBM Corporation

Designing Web System Infrastructure with WAS V8.0

Business Level Application (BLA)

- Business Level Application (BLA) とは
 - ◆ 複数コンポーネント(EAR/WAR/JARなど)をひとつのビジネス・アプリケーションとして関連付ける管理モデル
 - 複数のコンポーネントをまとめて管理可能
 - JAVA EEパッケージング・モデルよりも柔軟な管理が可能
- セッション共有方法
 - ◆ IBMApplicationSession
 - ◆ 共用セッション・コンテキスト: 非推奨
- ビジネス・レベルとしてBLA単位での管理が可能
- Fine-grained管理セキュリティによりBLA単位での管理者設定が可能

© 2012 IBM Corporation
155

WASV7より、Business Level Application (BLA)が新機能として追加されています。BLAは、複数コンポーネント(EAR/WAR/JARなど)を1つのビジネス・アプリケーションとして関連付ける新しい管理モデルです。BLAでは、J2EEパッケージング・モデルより柔軟な管理方法を提供します。これはJ2EEの仕様ではなく、WebSphereの拡張機能となります。また、Servletの仕様では、HTTPセッションはWebモジュール内でしか共用することができませんが、WebSphereではIBM拡張として、共用セッション・コンテキストによって複数EAR間でのセッションを共用する事ができました。但し、これはWAS V7.0より非推奨機能となっています。WASV7.0からは、BLAレベルでのセッションの共用として、IBMApplicationSessionの使用を推奨しています。例えば、BLA内に2つのEARアプリケーションが導入されている場合、それぞれのWebモジュールが同一のセッションを共用する事ができます。

BLAを定義する事で、BLAに登録された複数のアプリケーションをBLA単位で操作する事が可能となります。これにより、管理者はシステム単位(BLA単位)での操作が可能となり、同時にビジネス・レベルでの影響範囲がより明確になります。

また、詳細な(Fine-grained)管理セキュリティを使用することにより、BLA単位での管理者権限を付与する事が可能になります。これにより各システム単位(BLA単位)での管理者設定が可能となり、管理者から他のシステムへの操作を制限する事ができます。

また、BLAに関する詳細は、下記の資料をご参照ください。

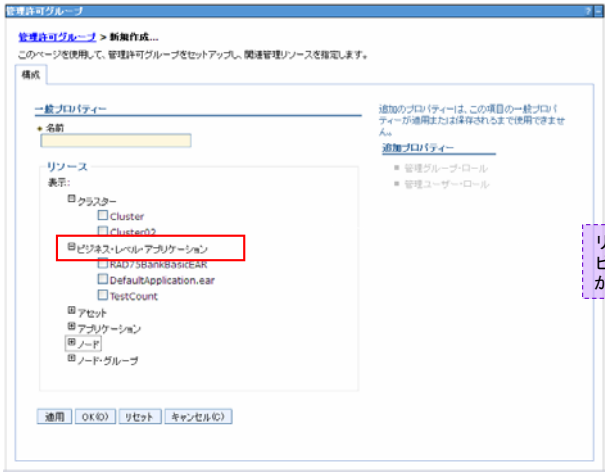
WebSphere Application Server V8 アナウンスメント・ワークショップ資料「アプリケーション管理」

http://www.ibm.com/developerworks/jp/websphere/library/was/was8_ws/

Designing Web System Infrastructure with WAS V8.0

BLA – 詳細な(Fine-grained) 管理セキュリティ設定

- 管理リソースの範囲でビジネス・レベル・アプリケーションを選択



リソースとして
ビジネス・レベル・アプリケーション
が選択可能

© 2012 IBM Corporation 156

管理コンソールでの詳細な(Fine-grained)管理セキュリティの設定です。管理リソースの範囲で「ビジネス・レベル・アプリケーション」を選択します。

Designing Web System Infrastructure with WAS V8.0

OSGiアプリケーションの管理(1/2)

- OSGi(Open Service Gateway Initiative)とは
 - ◆ モジュール化(コンポーネント化)を行うための技術の一つ
 - モジュールは「OSGiバンドル」として構成する
 - マニフェスト・ファイル(MANIFEST.MF)にOSGiで定義されたメタデータを記載
 - バンドルは独立的に管理・更新でき、アプリケーション全体との依存性が決まってい、柔軟にアップデート・再利用可能

The diagram illustrates the OSGi architecture stack. At the top, a box labeled 'OSGiアプリケーション' (OSGi Application) contains three yellow boxes labeled 'OSGi バンドル' (OSGi Bundle). This application layer sits on top of the 'OSGiフレームワーク' (OSGi Framework). Below the framework are three stacked layers: 'JVM' (Java Virtual Machine), 'オペレーティング・システム' (Operating System), and 'ハードウェア' (Hardware).

© 2012 IBM Corporation
157

OSGiアプリケーションとは、モジュール化(コンポーネント化)を行うための技術の一つであり、アプリケーションはOSGiバンドルから構成されています。

バンドルは物理的にはJARファイルそのものですが、マニフェストファイル(MANIFEST.MF)にOSGiで定義されているメタデータを追記することで、OSGiフレームワークにバンドルとして認識されるようになります。

メタデータには、バンドルが外部に公開するAPIを指定する定義(Export-Package)や、バンドルが動作するために必要とするAPIを指定する定義(Import-Package)が記述されており、この情報によってバンドル間のstaticな依存関係を明示することが出来ます。

OSGiを採用するメリットの一つ目は、コンポーネント化を安全に進めることができるということです。

これは、OSGiの提供するバンドルの独立性や、バンドル間の依存関係の明示が役立ちます。

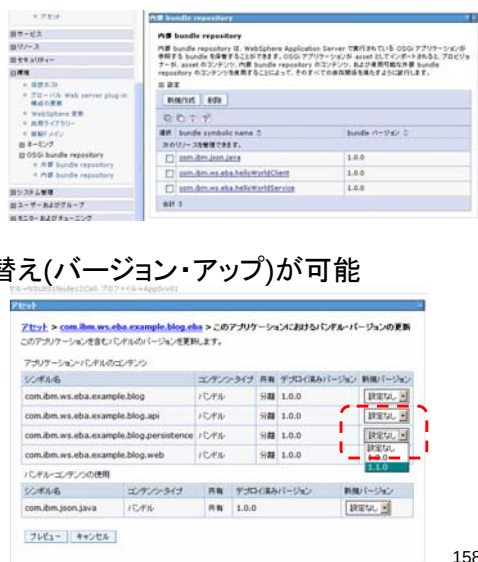
OSGiを採用するもう一つのメリットは、サービス・レジストリーによる間接参照を利用したモジュールの動的更新の実現です。

エンタープライズ・システムにおいては無停止でのアプリケーション更新を行うために様々なテクニックが利用されていますが、OSGiを採用することで解決可能な問題もあるかもしれません。

Designing Web System Infrastructure with WAS V8.0

OSGiアプリケーションの管理 (2/2)

- WASにおけるOSGi
 - ◆ Java EEとOSGi技術を両方使うアプリケーションをデプロイ/稼動/管理できるようになった
 - OSGiアプリケーションが使うバンドルの複数バージョンをバンドル・リポジトリから管理できる
 - 同じ又は異なるアプリケーションから、3rdパーティー・ライブラリーの異なるバージョンを使用することが可能
 - OSGiアプリケーションが使うバンドルの複数バージョンをバンドル・リポジトリから管理できる
 - ◆ JVMを起動したままバンドルの入れ替え(バージョン・アップ)が可能
 - 実際の更新を行う前に、管理コンソールからバンドルの新旧バージョンのプレビューが可能
 - 動的更新機能により、アプリケーションの停止を伴わずにバンドルの更新プロセスが実行可能



158

OSGiアプリケーションのインストール・管理は、WAS V7.0のFeature Pack for OSGiにおいても使用可能でした。WAS V8.0では、Feature Packを導入する必要なく、OSGiのアプリケーション管理機能が含まれています。

WAS V8では、Java EEとOSGi技術を両方使うアプリケーションをデプロイ/稼動/管理できるようになりました。OSGiアプリケーションをWASにインストールする際に、OSGiバンドルをBLAのアセットとして登録する必要があります。OSGiアプリケーションが使うバンドルの複数バージョンを登録したり、バンドル・リポジトリから閲覧・管理できます。

更に、WAS V8.0から追加された新機能として、アプリケーション・サーバーの起動中にOSGiバンドルのアップデートができます。この動的更新機能により、サービス停止を伴わずにバンドルの更新プロセスが実行可能になりました。

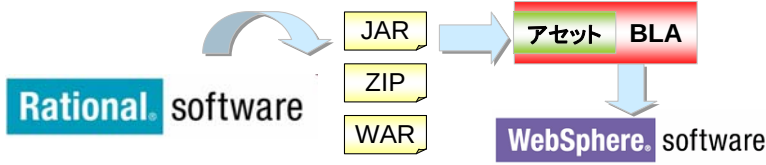
WASV8.0 Information Center「OSGi アプリケーションの管理」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/welc6tech_osgi_adm.html

Designing Web System Infrastructure with WAS V8.0

SCAアプリケーションの構成ファイルのエクスポート (1/3)

- SCAはXMLファイルで定義された、複数のコンポーネントを組み合わせ、Webサービス・アプリケーションを開発するモデル
- SCAアプリケーションをアセットとしてインポートし、BLAに登録
 - ◆ Java EEパッケージングから独立して利用することが可能



- SCAアプリケーションの部品はコンポーネント
- 複数コンポーネントを組合わせてコンポジット
- <コンポジット名>.compositeファイル
 - ◆ 一般的にSCDLという名前と呼ばれる
 - ◆ コンポジット内のコンポーネントやサービスなどの構造、設定を記述



© 2012 IBM Corporation 159

Service Component Architecture (略してSCA) はXMLファイルで定義された、複数のコンポーネントを組み合わせ、Webサービス・アプリケーションを開発するモデルです。SCAアプリケーションをWASに対してインストールするには、複数コンポーネントからなるコンポジットをBLAのアセットとして登録し、SCAアプリケーションをBLAとして動かします。

SCAアプリケーションの部品はコンポーネントであり、複数コンポーネントを組合わせてコンポジットを構成します。コンポジット内のコンポーネントに関する構造、設定などの構成情報は、SCDL(Service Component Definition Languageの略)と呼ばれる構成ファイルに記述されています。

Designing Web System Infrastructure with WAS V8.0

SCAアプリケーションの構成ファイルのエクスポート (2/3)

- SCAコンポジットの構成ファイルをエクスポートできる

AdminTask.exportSCDL(['-cuName <SCA_コンポジション名> -exportFile <ファイル名>'])

結果: SCAコンポジット(SCDL)が正常にエクスポートされています。
- 管理コンソールから、コンポジット構成ファイルを参照できる

管理コンソールから、「アプリケーション」>「アプリケーション・タイプ」>「ビジネス・レベル・アプリケーション」>「<アプリケーション名>」>「<SCAアセット名>」>「コンポジットの表示」をクリックする




© 2012 IBM Corporation 160

WAS V8.0から追加された新機能により、SCAアプリケーションのSCDL構成ファイルをwsadminでエクスポートできるようになりました。更に、管理コンソールからSCDL構成ファイルを閲覧可能になっています。

WASV8.0 Information Center「スクリプトによるSCAコンポジット定義のエクスポート」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tsca_export_compdef_scripting.html

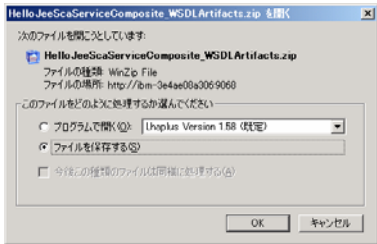
Designing Web System Infrastructure with WAS V8.0

SCAアプリケーションの構成ファイルのエクスポート (3/3)

- デプロイされたSCAのWSDL, XSD構成ファイルのエクスポート可能
SCAビジネス・レベル・アプリケーションは単一又は複数のコンポジション単位からなり、これらの定義は**WSDL**、**XSD構成ファイル**に書かれています。

管理コンソールから：

「アプリケーション」>「アプリケーション・タイプ」>「ビジネス・レベル・アプリケーション」>「<アプリケーション名>」>「<SCAアセット名>」>「Export WSDL and XSD document」のリンクをクリックして、ファイルの出力先ディレクトリを指定する


➡


© 2012 IBM Corporation
161

SCAアプリケーションはWebサービスの呼び出しなどで使用される、単一又は複数WSDL、XSDファイルを持っています。WAS V8.0から追加された新機能により、SCAアプリケーションのWSDLファイルを管理コンソールからエクスポートできるようになりました。WAS V7.0のFeature Pack for SCAにおいては、WSDL、XSDファイルのエクスポートは、wsadminでのみ実施可能でした。

WASV8.0 Information Center「WSDLおよびXSD文書のエクスポート」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tsca_export_wsdl_console.html

Designing Web System Infrastructure with WAS V8.0

SCAアプリケーションの構成ファイルのエクスポート(1/2)

- wsadmin|によるコンポジット構成ファイルのエクスポート

例) AdminTask.exportSCDL('[-cuName HelloJeeScaServiceComposite -exportFile C:/work/CompositeFile.txt]')

```

C:\>コマンド プロンプト - wsadmin.bat -lang jython
wsadmin>AdminTask.exportSCDL('[-cuName HelloJeeScaServiceComposite -exportFile C:/work/CompositeFile.txt]')
CWSAMU5081: The SCA Composite (SCDL) has been exported successfully.
wsadmin>

```

```

<composite targetNamespace="http://foo" name="HelloJeeScaServiceComposite"
xsi:schemaLocation="http://www.osoa.org/xmlns/sca/1.0 http://www.osoa.org/xmlns/sca/1.0" >
  <component name="HelloJeeEnhancedComponent" >
    <implementation.jee archive="HelloJeeEnhancedEar.ear" />
    <service name="HelloJeeEnhancedSBean_HelloJeeEnhancedSBeanRemote" >
      <interface.java interface="sca.jee.HelloJeeEnhancedSBeanRemote" />
    </binding.sca/>
    </service>
    <reference name="sbean2" target="HelloJeeScaComponent/HelloJeeScaService" >
      <interface.java interface="sca.jee.HelloJeeScaService" />
    </reference>
    </component>
    <component name="HelloJeeComponent" >
      <implementation.jee archive="HelloJeeEar.ear" />
    </component>
    <component name="HelloJeeScaComponent" >
      <implementation.java class="sca.jee.impl.HelloJeeScaServiceImpl" />
      <service name="HelloJeeScaService" >
        <interface.java interface="sca.jee.HelloJeeScaService" />
      </service>
    </component>
  </composite>

```

© 2012 IBM Corporation

162

詳細につきまして、WASV8.0 Information Centerをご参照ください。

WASV8.0 Information Center「スクリプトによるSCAコンポジット定義のエクスポート」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tsca_export_compdef_scripting.html

WAS V8.0 によるWebシステム基盤設計ワークショップ

162

Designing Web System Infrastructure with WAS V8.0

SCAアプリケーションの構成ファイルのエクスポート(2/2)

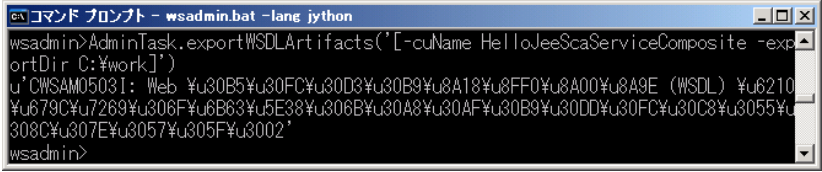
■ wsadminによるWSDL構成ファイルのエクスポート

例) wsadmin

1. print AdminTask.help('exportWSDLArtifacts')
2. AdminTask.exportWSDLArtifacts('[-cuName <CU名> -exportDir <ディレクトリー名>')

結果: 'CWSAM0503I: WSDL 成果物は、正常にエクスポートされました。'

3. exportWSDLArtifacts コマンドを実行すると <コンポジット名>_WSDLArtifacts.zip というファイルが指定されたディレクトリーに保存される (例では、HelloJeeScaServiceComposite_WSDLArtifacts.zipファイル)



WAS v7のFP for SCAIにおいて、wsadminによるエクスポートは可能だったが、管理コンソールからのエクスポートは新しく追加された

© 2012 IBM Corporation
163

詳細につきまして、WASV8.0 Information Centerをご参照ください。

WASV8.0 Information Center「WSDLおよびXSD文書のエクスポート」

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tsca_export_wsdl_console.html

Designing Web System Infrastructure with WAS V8.0

Fine-Grained Update – アプリケーション全体

- 更新オプション「アプリケーション全体を置換する」

- 設定
 - ◆ ファイル・システムの選択
 - ◆ パスの指定

© 2012 IBM Corporation
164

Fine-Grained Update の更新オプションの「アプリケーション全体を置換する」は、アプリケーション (earファイル) 単位で更新や置換したい場合に選択します。例えば、あるアプリケーション・サーバーにインストールされている「ApplicationA.ear」というアプリケーションを「ApplicationB.ear」で置換して更新したい場合は、更新オプションで「アプリケーション全体を置換する」を選択します。

設定は、ファイル・システムの選択とパスの指定です。ファイル・システムは、選択するearファイルが管理コンソールを開いているブラウザと同じノード上に存在している場合には「ローカル・ファイル・システム」を、別のノード上に存在している場合には「リモート・ファイル・システム」を選択します。パスの指定は、選択するearファイルが存在する場所を指定します。

Designing Web System Infrastructure with WAS V8.0

Fine-Grained Update – モジュール

■ 更新オプション「1つのモジュールを置換または追加する」

ApplicationA.ear

WebModuleA.war

EJBModuleC.jar

➡

ApplicationA.ear

WebModuleB.war

EJBModuleC.jar

更新

■ 設定

- ◆ モジュールへの相対パス
- ◆ ファイル・システムの選択
- ◆ パスの指定

Fine-Grained Update の更新オプションの「1つのモジュールを置換または追加する」は、モジュール(warファイル、jarファイル、rarファイル)単位で更新や置換、追加したい場合に選択します。例えば、あるアプリケーション・サーバーにインストールされている「ApplicationA.ear」というアプリケーション内の「WebModuleA.war」というモジュールを「WebModuleB.war」というモジュールで置換して更新したい場合は、更新オプションで「1つのモジュールを置換または追加する」を選択します。

設定は、モジュールへの相対パス、ファイル・システムの選択、パスの指定の指定です。モジュールへの相対パスは旧バージョンのモジュールへの相対パスを指定します。ここに、旧バージョンのモジュールへの相対パスを指定した場合はそのモジュールの置換となり、新バージョンのモジュールへの相対パスを指定した場合はそのモジュールの追加となります。ファイル・システムは、選択するモジュールが管理コンソールを開いているブラウザと同じノード上に存在している場合には「ローカル・ファイル・システム」を、別のノード上に存在している場合には「リモート・ファイル・システム」を選択します。パスの指定は、選択するモジュールが存在する場所を指定します。

WAS V8.0 によるWebシステム基盤設計ワークショップ

165

Designing Web System Infrastructure with WAS V8.0

Fine-Grained Update – ファイル

- 更新オプション「1つのファイルを置換または追加する」

ApplicationA.ear
 WebModuleA.war
 FileA.jpg
 FileB.jpg

➡

ApplicationA.ear
 WebModuleA.war
FileC.jpg
 FileB.jpg

更新

- 設定
 - ◆ ファイルへの相対パス
 - ◆ ファイル・システムの選択
 - ◆ パスの指定

① 1つのファイルを置換または追加する

新規ファイルへのパスがインストール済みアプリケーション内のいずれかのファイルへの既存パスと一致する場合、新規ファイルはその既存ファイルを置き換えます。新規ファイルへのパスがインストール済みアプリケーション内に存在しない場合、その新規ファイルはアプリケーションに追加されます。

置換または追加されるファイルへの、インストール済みアプリケーション・アーカイブ・ファイルで始まるパスを指定します。
 WebModuleA.war/FileA.jpg

ファイルへのパスを指定してください。

☒ ローカル・ファイル・システム

絶対パス
 C:/WAS7/FileC.jpg 参照...

☐ リモート・ファイル・システム

絶対パス 参照...

© 2012 IBM Corporation
166

Fine-Grained Update の更新オプションの「1つのファイルを置換または追加する」は、ファイル（classファイル、jpgファイル）単位で更新や置換したい場合に選択します。例えば、あるアプリケーション・サーバーにインストールされている「ApplicationA.ear」というアプリケーション内の「WebModuleA.war」というモジュール内の「FileA.jpg」というファイルを「FileC.jpg」というファイルで置換して更新したい場合は、更新オプションで「1つのファイルを置換または追加する」を選択します。

設定は、ファイルへの相対パス、ファイル・システムの選択、パスの指定です。ファイルへの相対パスは旧バージョンのファイルへの相対パスを指定します。ここに、旧バージョンのファイルへの相対パスを指定した場合はそのファイルの置換となり、新バージョンのファイルへの相対パスを指定した場合はそのファイルの追加となります。ファイル・システムは、選択するファイルが管理コンソールを開いているブラウザと同じノード上に存在している場合には「ローカル・ファイル・システム」を、別のノード上に存在している場合には「リモート・ファイル・システム」を選択します。パスの指定は、選択するファイルが存在する場所を指定します。

Designing Web System Infrastructure with WAS V8.0

Fine-Grained Update – 一部

- 更新オプション「複数のファイルを置換、追加、または削除する」

ApplicationA.ear
 WebModuleA.war
 Dir1/Dir2/FileA.jpg
 EJBModuleA.jar

更新

Application.zip
 [Application]
 |--[WebModuleA.war]
 | |--[Dir4]
 | | |--[Dir5]--FileD.jpg
 |--EJBModuleB.jar

ApplicationA.ear
 WebModuleA.war
 Dir1/Dir2/FileA.jpg
 Dir4/Dir5/FileD.jpg
 EJBModuleB.jar

- アーカイブの作成
- 設定
 - ◆ ファイル・システムの選択
 - ◆ パスの指定



© 2012 IBM Corporation
167

Fine-Grained Update の更新オプションの「複数のファイルを置換、追加、または削除する」は、複数のファイルやモジュールをアーカイブしたzipファイルやgzipファイルなどの単位で更新や置換、削除したい場合に選択します。例えば、あるアプリケーション・サーバーにインストールされている「ApplicationA.ear」というアプリケーション内の「WebModuleA.war」というモジュール内のDir4/Dir5ディレクトリーに「FileD.jpg」というファイルを追加し、「EJBModuleA.jar」というモジュールを「EJBModuleB.jar」というモジュールで置換して更新したいような場合は、更新オプションで「複数のファイルを置換、追加、または削除する」を選択します。

アーカイブ・ファイルは、追加や更新の場合は新バージョンのファイルやモジュールをディレクトリー構造にまとめてzipやgzipでアーカイブして作成します。このとき、旧バージョンのパスと一致するファイルの場合は追加、一致しないファイルの場合は更新となります。

設定は、ファイル・システムの選択とパスの指定です。ファイル・システムは、選択するアーカイブ・ファイルが管理コンソールを開いているブラウザと同じノード上に存在している場合には「ローカル・ファイル・システム」を、別のノード上に存在している場合には「リモート・ファイル・システム」を選択します。パスの指定は、選択するアーカイブ・ファイルが存在する場所を指定します。

Designing Web System Infrastructure with WAS V8.0

更新のロールアウト – 管理コンソールでの更新

■ 更新ウィザード

◆ 終了後に「更新のロールアウト」を選択

© 2012 IBM Corporation
168

管理コンソールを使用した 更新のロールアウト を行う手順を説明します。管理コンソールから Fine-Grained Update などでするサーバークラスターに対するアプリケーションの更新を行った場合、処理が完了すると「更新のロールアウト」、「マスター構成に保管」、「アプリケーションの管理」の3つの中から次の処理内容を選択することになります。更新のロールアウトを行うには、ここで「更新のロールアウト」をクリックするか、「アプリケーションの管理」をクリックしてから次の画面で「更新のロールアウト」ボタンをクリックします。この時点から、各ノードへの自動反映が順番に始まります。

Designing Web System Infrastructure with WAS V8.0

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本講演資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもののでも、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したもののでもなく、またそのような結果を生むものでもありません。

本講演資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本講演資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本講演資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもののでも、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、AIX、Rational、Tivoli、WebSphere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IBMまたは各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。

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。
Windowsは Microsoft Corporationの米国およびその他の国における商標です。
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。
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。

© 2012 IBM Corporation

169