

JEUS リファレンスガイド

JEUS v8.0



Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13613, South Korea

Restricted Rights Legend

All TmaxSoft Software (JEUS®) and documents are protected by copyright laws and international convention. TmaxSoft software and documents are made available under the terms of the TmaxSoft License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxSoft Co., Ltd.

Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxSoft trademarks, logos, or any other brand features. This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

このソフトウェア(JEUS®)マニュアルの内容とプログラムは、日本国の著作権法および国際条約によって保護されています。マニュアルの内容とプログラムは、TmaxSoft Co., Ltd.との使用許諾契約書の下でのみ使用することができ、マニュアルは使用許諾契約で許可されている範囲を除いては、配布または複製することができません。TmaxSoftの書面による事前の承諾を得ることなく、このマニュアルの全部または一部を電子的または機械的な方法を問わず、転送、複製、配布したり、または二次的著作物を作成する等の行為を一切禁じます。

このソフトウェアのマニュアルとプログラムの使用許諾契約は、いかなる場合においても、マニュアル及びプログラムと関連する知的財産権(登録の有無を問わず)を譲渡するものと解釈されず、TmaxSoftのブランド、ロゴ、商標等の使用権限を与えるものではありません。マニュアルは、情報を提供する目的でのみ提供しており、これに伴う契約上の直接的ないしは間接的な責任を負わず、マニュアルの内容は法律上もしくは商業的な特定の条件が満たされることを保証しません。マニュアルの内容は、製品のアップグレード及び修正により、その内容が予告なく変更されることがあり、内容上の誤りがないことを保証しません。

Trademarks

JEUS® is registered trademark of TmaxSoft Co., Ltd.

JEUS®は、TmaxSoft Co., Ltd.の登録商標です。

Java and Solaris are registered trademarks of Oracle Corporation and its subsidiaries and affiliates.

Java、Solarisは、Oracle Corporation及びその子会社、関連会社の登録商標です。

Microsoft, Windows, and Windows NT are registered trademarks or trademarks of Microsoft Corporation.

Microsoft、Windows、Windows NTは、Microsoft Corporationの登録商標または商標です。

HP-UX is a registered trademark of Hewlett Packard Enterprise Company.

HP-UXは、Hewlett Packard Enterprise Companyの登録商標です。

AIX is a registered trademark of International Business Machines Corporation.

AIXは、International Business Machines Corporationの登録商標です。

UNIX is a registered trademark of X/Open Company, Ltd.

UNIXは、X/Open Company, Ltd.の登録商標です。

Linux is a registered trademark of Linus Torvalds.

Linuxは、Linus Torvaldsの登録商標です。

Other products and company names are trademarks or registered trademarks of their respective owners.

その他、記載されている会社名、製品名などは、各社の商号、商標または登録商標です。

The names of companies, systems, and products mentioned in this manual may not necessarily be indicated with a trademark symbol (TM, ®).

本マニュアルに記載されている会社名、システム名、製品名などには必ずしも商標表示(TM、®)を付記しておりません。

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/lib/licenses

この製品の一部ファイルまたはモジュールは、APACHE2.0、CDDL1.0、EDL1.0、OPEN SYMPHONY SOFTWARE1.1、TRILEAD-SSH2、Bouncy Castle、BSD、MIT、SIL OPEN FONT1.1のライセンスに準拠します。

文書情報

文書名: JEUS リファレンスガイド

発行日: 2016年10月14日

ソフトウェアバージョン: JEUS v8.0

ガイドバージョン: v2.1.1

目次

このガイドについて	xv
I. システム・プロパティ	1
第1章 システム・プロパティ	3
1.1. 概要	3
1.2. サーバーのシステム・プロパティ	3
1.3. ネットワークのシステム・プロパティ	10
1.4. JNDIのシステム・プロパティ	14
1.5. トランザクションのシステム・プロパティ	18
1.5.1. TM Clientのためのシステム・プロパティ	20
1.5.2. 下位の互換性のためのシステム・プロパティ	22
1.6. Webエンジンのプロパティ	23
1.6.1. Webエンジン・レベルのプロパティ	23
1.6.2. バーチャル・ホスト・レベルのプロパティ	24
1.6.3. Webアプリケーション・レベルのプロパティ	24
1.7. セッション・サーバーのシステム・プロパティ	32
1.8. EJBのシステム・プロパティ	33
1.8.1. EJBのシステム・プロパティ	33
1.8.2. JPAのシステム・プロパティ	37
1.8.3. プロパティ	38
1.9. セキュリティーのシステム・プロパティ	42
1.9.1. Standard Java SE & Java EEセキュリティーのシステム・プロパティ	43
1.9.2. JEUSセキュリティーのシステム・プロパティ	43
1.10. JEUS MQのシステム・プロパティ	45
1.10.1. JEUS MQのクライアント	46
1.10.2. JEUS MQサーバー	48
II. コンソール・コマンドとツール	51
第2章 コマンドの説明規則	53
第3章 JEUSサーバーの実行	55
3.1. 概要	55
3.2. 使用方法	55
3.3. マスター・パスワードの入力	58
3.4. デバッグ・モードのJEUSサーバーの実行	59
第4章 コンソール・ツール	61
4.1. 概要	61
4.2. jeusadmin	62
4.2.1. 起動および終了コマンド	63
4.2.2. ローカル・コマンド	67
4.2.3. サーバー管理関連コマンド	82

4.2.4.	ドメイン設定関連コマンド	133
4.2.5.	スレッド管理関連コマンド	167
4.2.6.	アプリケーション関連コマンド	178
4.2.7.	EJBエンジン関連コマンド	201
4.2.8.	Webエンジン関連コマンド	205
4.2.9.	セッション関連コマンド	243
4.2.10.	JMSエンジン関連コマンド	251
4.2.11.	データソース関連コマンド	273
4.2.12.	コネクション・プールのモニタリングおよび制御コマンド	300
4.2.13.	トランザクション関連コマンド	307
4.2.14.	セキュリティ関連コマンド	313
4.2.15.	ノード管理関連コマンド	328
4.2.16.	ノード・マネージャー関連のコマンド	340
4.2.17.	設定編集コマンド	346
4.2.18.	JSONを使用したコマンドの実行	360
4.2.19.	Scriptモードの使用方法およびスクリプトの作成方法	363
4.3.	appcompiler	366
4.4.	ejbddinit	370
4.5.	webddgen	376
4.6.	encryption	377
4.7.	wsgen	379
4.8.	wsimport	382
4.9.	java2wsdl	385
4.10.	wsdl2java	386
4.11.	wsdl2uddi	389
4.12.	xjc	390
4.13.	schemagen	393
4.14.	tcpmon	394
4.14.1.	実行	394
4.14.2.	Listenerモードの使用	395
4.14.3.	Proxyモードの使用	398
4.14.4.	その他の機能	398
III.	Antタスク	399
第5章	Antタスク	401
5.1.	概要	401
5.2.	環境構成のAntタスク	402
5.2.1.	create-domain	402
5.3.	サーバー制御のAntタスク	404
5.3.1.	boot	404
5.3.2.	down	405
5.3.3.	実行	405
5.4.	EJBのAntタスク	408

5.4.1.	appcompiler	408
5.4.2.	ejbddinit	410
5.5.	WebサービスのAntタスク	412
5.5.1.	java2wsdl	413
5.5.2.	wsdl2java	414
5.5.3.	wsgen	418
5.5.4.	wsimport	420
5.5.5.	xjc	422
5.5.6.	schemagen	423
5.6.	WebサービスのMavenプラグイン	425
5.6.1.	wsgen	434
5.6.2.	wsimport	428
5.7.	jeusadminコマンドの実行	432
5.8.	Mavenプラグイン	434
5.8.1.	インストール	434
5.8.2.	Maven Plugin Goals	435

図目次

[図 4.1]	tcpmonの実行画面	394
[図 4.2]	サーバー(A)が8000番ポートでSOAP要求を受信するシナリオ	395
[図 4.3]	Listenerモードの入力画面	395
[図 4.4]	モニタリング画面	396
[図 4.5]	tcpmonが9000番ポートで受信した要求を受け取り、Aの8000番ポートに送信	396
[図 5.1]	wsdl2javaタスクの構成	416

表目次

[表 4.1]	ejbddinitのプロパティ	373
[表 4.2]	export-nameをサポートするパターン	374
[表 5.1]	create-domainタスクのプロパティ	402
[表 5.2]	boot Taskプロパティ	404
[表 5.3]	downタスクのプロパティ	405
[表 5.4]	appcompiler Antタスクのプロパティ	408
[表 5.5]	ejbddinit Antタスクのプロパティ	410
[表 5.6]	ejbddinitプロパティ・ファイルとejbddinit Antスクリプトにおけるejbddinitプロパティの設定例 ..	411
[表 5.7]	java2wsdl Antタスクのプロパティ	413
[表 5.8]	wsdl2javaタスクのプロパティ	415
[表 5.9]	wsdl2javaタスク - mapping要素のプロパティ	416
[表 5.10]	wsgenタスクのプロパティ	418
[表 5.11]	wsimportタスクのプロパティ	420
[表 5.12]	xjcタスクのプロパティ	422
[表 5.13]	schemagenタスクのプロパティ	423
[表 5.14]	wsgenプラグインのプロパティ	425
[表 5.15]	wsimportプラグインのプロパティ	429

例目次

[例 5.1]	boot & down Antタスクのbuildファイルの例	406
[例 5.2]	appcompiler Antタスクのbuildファイルの例	409
[例 5.3]	ejbddinit Antスクリプトの例	411
[例 5.4]	java2wsdl Antタスクのbuildファイルの例	413
[例 5.5]	wsdl2java Antタスクのbuildファイルの例	417
[例 5.6]	wsgen Antタスクのbuildファイルの例	419
[例 5.7]	wsimport Antタスクのbuildファイルの例	421
[例 5.8]	xjc Antタスクのbuildファイルの例	422
[例 5.9]	schemagen Antタスクのbuildファイルの例	424
[例 5.10]	wsgen plugin pomファイルの例	426
[例 5.11]	wsimport plugin pomファイルの例	430

このガイドについて

対象読者

本書は、JEUS[®](以下、JEUS)を使用するために必要な詳細設定および使用方法について説明します。そのため、JEUSを管理するシステム管理者や開発者を対象としています。

前提知識

JEUSの基本的な使用方法と製品を理解するには、以下のガイドについてあらかじめ熟知することをお勧めします。

- JEUS 紹介ガイド
- JEUS インストール & スタートガイド

本書のすべてのサンプルと環境構成は、UNIXスタイルに準拠します。Microsoft Windows[™](以下、Windows)など他の環境で作業を行う場合は、次のような事項を考慮してください。

たとえば、Windowsプラットフォームでは、ディレクトリー区切り子をUNIXスタイルのスラッシュ(/)からWindowsスタイルのバックスラッシュ(\)に変えて使用してください。また、環境変数もWindowsスタイル(%%)に変更して使用してください。

本書で触れているJEUS_HOMEは、JEUSがインストールされているディレクトリーです。

制限事項

本書の内容は、Java標準に準拠して作成されていますが、本書で触れているJava EEやJava仕様については詳しく取り上げていません。関連内容についてはJava関連ドキュメントを参照してください。

本書では、他のガイドでは詳しく取り上げていない設定および使用方法について説明します。本書ではXML設定ファイルについては扱っていないので、関連内容については『JEUS XMLリファレンスガイド』を参照してください。

本書の構成

本書は、特定の内容をすぐに見つけて参照できるように構成されています。各パートと各章の内容は互いに先行関係がなく、必要に応じて目次などから関連内容を探して参照してください。

本書は、計3章で構成されています。

- [Part I, 「システム・プロパティ」](#)

JEUSのシステム・プロパティについて説明します。

- [Part II, 「コンソール・コマンドとツール」](#)

コンソールのコマンドとツールについて説明します。

- [Part III, 「Antタスク」](#)

JEUSが提供するAntタスクについて説明します。

表記上の規則

表記	意味
<<AaBbCc123>>	プログラム・ソースコードのファイル名
<Ctrl>+C	CtrlキーとCキーを同時に押す
[Button]	GUIのボタン、メニュー名
太字	強調
「」、『』（鍵カッコ）	関連文書、あるいはガイド内の他の章および節の表示
「入力項目」	画面UI上の入力項目
ハイパーリンク	メール・アカウント、Webサイト
>	メニューの実行順
+----	下位ディレクトリー/ファイル有り
----	下位ディレクトリー/ファイル無し
<div>参考</div>	参照/注意事項
<div>注</div>	注意事項
[図 1.1]	図の名前
[表 1.1]	表の名前
AaBbCc123	Javaコード、XMLドキュメント
[<i>command argument</i>]	オプション・パラメータ
< xyz >	「<」と「>」の間の内容は実際に使用される特定の名前または値で置き換えられる
	構文の中の相互に排他的な選択項目の選択肢を示す 例) A B: AとBのいずれかを選択
...	パラメータ、値、または他の情報が繰り返される
\${ }	環境変数

システム要件

	要求事項
プラットフォーム	Solaris 9, 10, 11
	HP-UX 11.x, 11i, 11iV2
	IBM AIX 5L, 6L, AIX 7L
	MS Windows 2008, 2012, Vista, 7, 8
ハードウェア	最小2GB以上、推奨20GBのハードディスク容量
	推奨1GB以上のメモリー容量
JDK	JDK 7, JDK 8

関連文書

ガイド	説明
JEUS 紹介ガイド	JEUSサーバーについて全般的に紹介し、JEUSのアーキテクチャーを含む各構成要素について記述しています
JEUS インストール&スタートガイド	JEUSについて紹介し、JEUSのインストールおよび開始方法について記述しています
JEUS サーバガイド	JEUSシステムおよびサーバーの概要とシステムの管理方法について記述しています
JEUS WebAdminガイド	JEUSのWeb管理ツールであるWebAdminを利用したJEUSの設定および制御、モニタリング、クラスタリング、リソースの設定および管理について記述しています

参考文献

- JEUS API
JEUS_HOME/docs/api/jeusapi/index.html
- JEUS Security SPI
JEUS_HOME/docs/api/security/index.html
- TCP Listener API
JEUS_HOME/docs/api/tcplistener/index.html

- JMX API

JEUS_HOME/docs/api/jmx/index.html

お問合せ先

Korea

TmaxSoft Co., Ltd.
45, Jeongjail-ro, Bundang-gu,
Seongnam-si, Gyeonggi-do, 13613
South Korea
Tel: +82-31-8018-1000
Fax: +82-31-8018-1115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmaxsoft.com>
TechNet: <http://technet.tmaxsoft.com>

USA

TmaxSoft Inc.
101 North Wacker Drive, Suite 2014,
Chicago, IL 60606
U.S.A
Tel: +1-312-525-8330
Email: info@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/us_en/home

Japan

TmaxSoft Japan Co., Ltd.
5F Sanko Bldg, 3-12-16 Mita, Minato-Ku, Tokyo, 108-0073
Japan
Tel: +81-3-5765-2550
Fax: +81-3-5765-2567
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

Beijing TmaxSoft System Software Co., Ltd.
Room103, No.2 Huizhong Building, Seven Street Shangdi,
Haidian District, Beijing, 100085
P.R.China
Tel: +86-10-6298-8827
Email: info@tmaxsoft.com.cn
Web (Chinese): http://www.tmaxsoft.com/cn_en/home_cn_en

Brazil

Tmax Brasil Sistemas e Serviços Ltda.
Av. Copacabana, 177, sala 32~35 Empresarial 18 do Fortel
Alphaville Barueri, Sao Paulo, 06472-001
Brazil
Tel: +55-11-4191-3100
Fax: +55(11) 4191-3705 (extension#112)
Email: info.bra@tmaxsoft.com
Web (Portuguese): http://www.tmaxsoft.com/br_en/home_br_en

Russia

Tmax Rus L.L.C.
Leninsky prospekt, 113/1 (Park Place Moscow),
Office 318e, Moscow, 117198
Russia
Tel: +7(495)970-01-35
Email: info.rus@tmaxsoft.com
Web (Russian): http://www.tmaxsoft.com/ru_ru/home_ru_ru

Singapore

Tmax Singapore Pte. Ltd.
430 Lorong 6, Toa Payoh #10-02,
OrangeTee Building, 319402
Singapore
Tel: +65-6259-7223
Fax: +65-6258-7112
Email: info.sg@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/sg_en/home_sg_en

United Kingdom

TmaxSoft UK Ltd.
215 Knyvett House, Watermans Business Park,
The Causeway, Staines TW18 3BAB
United Kingdom
Tel: +44-1784-895005
Email: info.uk@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/gb_en/home_gb_en

Canada

TmaxSoft Canada, Inc.
2425 Matheson Blvd East, 8th floor,
Unit 824 Mississauga, ON, L4W 5K4
Canada
Tel: +1-905-361-2888
Email: info.canada@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/ca_en/home_ca_en

Australia

TmaxSoft Proprietary Limited
L32, 101 Miller Street, North Sydney 2060
Australia
Tel: +91-9845-330-704
Email: info.aus@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/au_en/home_au_en

India

TmaxSoft Technologies Private Limited
Sobha Alexander Plaza, 3rd Floor,
16/2 Commissariat Road, Bangalore-560025
India
Tel: +91-9845-330-704
Email: info.india@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/in_en/home_in_en

Turkey

TmaxSoft Co., Ltd. Turkey Liaison Office
Windowist Tower. Eski Buyukdere Cad. No:26,
Maslak 34467 Istanbul
Turkey
Tel: +90-544-553-6045
Email: cslee@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/tr_en/home_tr_en

Part I. システム・プロパティ

第1章 システム・プロパティ

1.1. 概要

システム・プロパティは、JVMに-Dオプションで設定する環境変数です。JEUSは基本的にLaunch JVMを利用してサーバーJVMを駆動するため、domain.xmlのサーバー別にシステム・プロパティを設定します。

システム・プロパティを設定する方法は以下のとおりです。

```
<domain>
  <servers>
    <server>
      <name>server1</name>
      <jvm-config>
        <jvm-option>-Xmx512m -XX:MaxPermSize=250m</jvm-option>
      </jvm-config>
      ...
    
```

- 注
1. Domain Administrator Serverまたはサーバー・スクリプトに設定したものは、Launch JVMにのみ適用されるため注意してください。

2. システム・プロパティは、JEUSの全般的な動作とすべてのアプリケーションに影響を与えるため、注意して設定してください。システム・プロパティよりXML設定を使用することをお勧めします。

1.2. サーバーのシステム・プロパティ

本節では、サーバーのシステム・プロパティについて説明します。

- jeus.home

説明	XMLやコードなどを省略します
デフォルト値	JEUSがインストールされているディレクトリー

- jeus.launcher.log.home

説明	Jeus Launcher Logが作成される位置を指定します。指定していない場合はJEUS_SERVER_HOME/logsに残ります
デフォルト値	null

- jeus.boot.printclasspath

説明	システムのクラス・ローダーのクラス・パスを出力するか否かを設定します
デフォルト値	false

- jeus.server.autorestart

説明	SCFによってMS(Managed Server)が障害検知された場合に、自動で再起動するか否かを設定します。例外的にJAVAノード・マネージャーを使用する場合には、障害の検知をSCFに依存しないため、このプロパティと関係なく、ノード・マネージャーのauto restart設定値に従います
デフォルト値	true

- jeus.jdk.home

説明	JDKホーム・ディレクトリーのパスで、javac、tools.jarのパスの設定時に使用します
デフォルト値	指定されていない場合はjava.homeを使用します

- jeus.properties.replicate

説明	ランチャーでサーバーを移動する際に渡すシステム・プロパティのプレフィックスを指定します。 domain.xmlのserver/jvm-config/jvm-optionが優先されます
デフォルト値	jeus

- jeus.server.check-interrupted-status

説明	JEUSでEJB、JDBCオペレーションを実行するスレッドが割り込み信号を受信した場合、例外を発生させるか否かを設定します。 JEUSでは、特定作業を行う場合にスレッドに割り込みが掛かっていると例外が発生するため、該当例外についてはユーザー・アプリケーションで処理する必要があります
デフォルト値	true

- jeus.threadpool.queueSize

説明	JEUSシステムで使用するスレッド・プールの内部キューのサイズを設定します。 スレッド・プールではタスクを実行するために内部キューを使用しますが、このキューのサイズを設定します。システムの負荷が大きい場合や設定スレッドの数が少ない場合、キューイングされるタスクが増加します。この際、設定したキューのサイズのみだけキューイングを許可します。 このシステム・プロパティよりdomain.xmlのserver/system-thread-pool/queue-size値が優先されます
デフォルト値	4096

- jeus.threadpool.keepalive

説明	JEUSシステムで使用するスレッド・プールに適用されるキープ・アライブ時間を設定します。 min以下のスレッドは、設定時間内に使用されなければスレッド・プールから削除されます。 このシステム・プロパティよりdomain.xmlのserver/system-thread-pool/keep-alive-time値が優先されます
デフォルト値	300000 (単位 : ms)

- jeus.rmi.defaultPort

説明	RMIのデフォルトのエクスポート・ポートを指定します。export-portが指定されていないすべてのEJBにも適用されます
デフォルト値	- base port + 7

- jeus.rmi.jeusrmi.defaultPort

説明	JEUSが提供するJEUS RMIを使用する場合、RMIのデフォルトのエクスポート・ポートを指定します
デフォルト値	- base port + 11

- jeus.vm.bits

説明	JVMのビット数を指定します。 サーバーの起動時、64bit JVMの場合は64を設定すると-d64が追加されます。 IBMの場合は-d64が不要です
----	-------------------------------------------------------------------------------------------

- jeus.interop.ssl.keystore

説明	CORBA CSrv2で使用するSSLキーストアを指定します
デフォルト値	jeus.net.ssl.keystoreの値

- jeus.interop.ssl.keypass

説明	CORBA CSrv2で使用するSSLキーストアのパスワードを指定します
デフォルト値	jeus.net.ssl.keypassの値

- jeus.interop.ssl.truststore

説明	CORBA CSrv2で使用するSSLトラストストアを指定します
デフォルト値	jeus.net.ssl.truststoreの値

- jeus.interop.ssl.trustpass

説明	CORBA CSrv2で使用するSSLトラストストアのパスワードを指定します
デフォルト値	jeus.net.ssl.trustpassの値

- jeus.jvm.version

説明	JEUSが使用するJVMのバージョンに関する情報を設定します
デフォルト値	hotspot (-serverを使用)

- jeus.server.cpumonitor.enabled

説明	ホストのCPU使用率を測定するためのモニタリング機能を活性化します
デフォルト値	false

- jeus.server.cpumonitor.interval

説明	ホストのCPU使用率をモニタリングする機能を使用する場合のCPU使用率を測定する周期です
デフォルト値	60000 (単位 : ms)

- jeus.server.process.cpumonitor.enabled

説明	プロセスのCPU使用率を測定するためのモニタリング機能を活性化します
デフォルト値	false

- jeus.server.process.cpumonitor.interval

説明	プロセスのCPU使用率をモニタリングする機能を使用する場合、CPU使用率を測定する周期です
デフォルト値	2000 (単位 : ms)

- jeus.server.cpumonitor.ratio

説明	プロセスのCPU使用率をモニタリングする機能を使用する場合、CPUのオーバーフローの基準値を設定します。 小数点で設定し、CPUの使用量を最大100と考えた場合、最大何パーセントのCPUを使用すればオーバーフローと仮定するかを設定します
デフォルト値	0.8

- jeus.server.enable.restart.in.memory.shortage

説明	メモリーがオーバーフローの場合にサーバーを再起動するか否かを設定します
デフォルト値	true

- jeus.server.memorymonitor.enabled

説明	サーバーのメモリー使用率を測定するためのモニタリング機能を活性化します。モニタリングするとき、スレッド・ダンプがログに残ります
デフォルト値	false

- jeus.server.memorymonitor.interval

説明	メモリーの使用率をモニタリングする機能を使用する場合、メモリーの使用率を測定する周期を指定します
デフォルト値	2000 (単位 : ms)

- jeus.server.memorymonitor.duration

説明	メモリーの使用率をモニタリングする機能を使用する場合、メモリーのオーバーフロー現象の持続時間を設定します。設定時間の間にメモリーのオーバーフロー時間が持続されると、サーバーが停止します
デフォルト値	60000 (単位 : ms)

- jeus.server.memorymonitor.ratio

説明	<p>メモリーの使用率をモニタリングする機能を使用する場合、メモリーのオーバーフローの基準値を設定します。</p> <p>小数点で設定し、メモリーの使用量を最大100とした場合、最大何パーセントのメモリーを使用している場合にオーバーフローと見なすかを設定します</p>
デフォルト値	0.8

- jeus.server.maxdowntime

説明	<p>メモリーの使用率をモニタリングする機能を使用する場合、メモリーのオーバーフローが発生してサーバーが停止する際に、サーバーが停止するまで待機する時間を設定します。</p> <p>設定時間が過ぎてもサーバーが終了しない場合、サーバーは強制終了します</p>
デフォルト値	0 (単位 : ms)

- jeus.server.configfile.backup.size

説明	動的変更時に保存されるbackup domain.xmlの数を指定します
デフォルト値	50

- jeus.print-system-properties

説明	設定されたシステム・プロパティを出力するか否かを設定します
デフォルト値	false

- jeus.management.snmp.dynamic

説明	<p>SNMPサーバーを使用する場合に動的OIDを生成するか否かを設定します。</p> <p>SNMPサーバーを使用する場合、OIDを静的にするか、または動的にするかを設定できます。</p> <ul style="list-style-type: none"> – true : 静的の場合で、MIBファイル・ベースのOIDを生成します – false : 動的の場合で、オブジェクトによって動的に作成されたOIDが静的OIDのサフィックスとして追加されます
デフォルト値	true

- jeus.management.snmp.dynamic.shortoid

説明	<p>動的OIDの長さを選択します。</p> <p>MIBベースのOIDに追加されるOIDの構成要素の桁数を指定します。</p> <ul style="list-style-type: none"> – true : Short OIDの場合、8桁のIDが追加されます – false : Short OIDでない場合、32桁のIDが追加されます
デフォルト値	false

- jeus.domainadminserver.deploy.socket

説明	<p>DAS(Domain Administration Server)が維持するデプロイ用途のソケット・コネクションの数の最大値です。</p> <p>MS(Managed Server)では、起動時間を短縮するために、コネクションを複数確立し、平行してDASからアプリケーションを受け取ることができます。この際、DASは一度に複数のMSから多くのコネクションを確立するため、この最大値を設定できるようにします。</p> <p>DASのデプロイ・コネクションが最大値となる状態で、他のMSからアプリケーション転送要求がある場合、該当MSは1個のコネクションのみを使用して順次的にアプリケーションを受け取るようにします</p>
デフォルト値	100

- jeus.managedserver.deploy.socket

説明	<p>DASからアプリケーションを受け取る際、MSが使用するboot time deploy connectionの最大値です。</p> <p>MSが起動し、DASからアプリケーションを受け取りますが、この際に平行して受け取るために使用するMSのboot time deploy connectionの数の最大値を設定します。</p> <p>1つのMSが非常に多くのコネクションを設定してDASからのコネクション数がフルになった場合、他のMSが影響を受ける可能性があるため、注意して使用する必要があります。</p> <p>アプリケーションの転送が終了すると、該当コネクションはすぐに解除され、DASの使用可能なコネクションの数を再度増加させます。また、コネクション数を大きく設定しても、受け取るアプリケーションの数が小さい場合、内部的に維持するコネクションの数はアプリケーションの数に自動設定されます</p>
デフォルト値	5

- jeus.logging.useAsync

説明	プロパティについての詳細は『JEUS サーバガイド』の「8.1. 概要」を参照してください
デフォルト値	true

- jeus.access.logging.skip.when.busy

説明	jeus.logging.useAsyncがtrueの場合にのみ適用されるオプションです。プロパティについての詳細は『JEUS サーバガイド』の「8.1. 概要」を参照してください
デフォルト値	false

- jeus.jmx.ping.period

説明	JMXコネクションで接続状態をチェックするために使用するピングの周期を設定します
デフォルト値	15000 (単位: ms)

- jeus.jmx.ping.timeout

説明	JMXコネクションで接続状態をチェックするために使用するピングのタイムアウトを設定します。ピングを送った後、タイムアウト時間まで応答が返されない場合、接続に問題があると判断してI/O例外を発生させます
デフォルト値	15000 (単位: ms)

参考

jeus.server.memorymonitor.enabled、jeus.server.memorymonitor.ratio、jeus.server.memorymonitor.interval、jeus.server.memorymonitor.duration、jeus.server.enable.restart.in.memory.shortageプロパティの詳細については『JEUS サーバガイド』の「3.4.2. メモリ使用量による制御」を参照してください。

1.3. ネットワークのシステム・プロパティ

本節では、ネットワークのシステム・プロパティについて説明します。

- jeus.server.checktmout

説明	JEUSを管理するRMIコネクションやJMXコネクタに適用するタイムアウトを設定します
デフォルト値	60 * 1000 (単位 : ms)

- jeus.net.crosswait

説明	2つのJEUSサーバーが互いにコネクションを確立しようとするクロス・コネクション状況の場合、コネクションが拒否されたプロセスにおいて相手は接続要求を行い、コネクションの結合を待機する時間を指定します
デフォルト値	10000 (単位 : ms)

- jeus.net.timewait

説明	CLOSEメッセージの送信を待機する時間です。この時間が短い場合、正常終了でもI/O例外が発生したというトレースが出力されることがあります
デフォルト値	10000 (単位 : ms)

- jeus.net.tcpbuffer

説明	JEUSの運用環境のTCPバッファ・サイズを設定します。 この値は、JEUSサーバー内部のリード・バッファのサイズとして使用されるため、I/Oの性能を最大限発揮するには、正確な値を設定する必要があります
デフォルト値	8*1024 (単位 : byte)

- jeus.net.recvbuffer

説明	JEUSサーバーの環境のTCP Receive Bufferのサイズを設定します
-----------	------------------------------------------

- jeus.net.sendbuffer

説明	JEUSクライアント環境のTCP Send Bufferのサイズを設定します
-----------	----------------------------------------

- jeus.net.busywrite

説明	非ブロッキング・モードで動作する場合、書き込み時にセクターに任せず、引き続き試行する回数を設定します。受信側で読み込まない場合を除いては、この回数内ではselect()を再度呼び出さないのが速いです
デフォルト値	2

- jeus.net.msg.max

説明	JEUSのプロトコルを通じて送信できる1つのメッセージの最大サイズを設定します。 このサイズを超過するメッセージを送信した場合、受信側で例外が発生し、接続が切断されることがあります
-----------	---------------------------------------------------------------------------------------------------

デフォルト値	5 * 1000 * 1000 (単位 : byte)
---------------	-----------------------------

- jeus.net.ping.enable

説明	双方向のPing機能を使用するか否かを設定します
デフォルト値	false

- jeus.net.ping.timeout

説明	Pingメッセージのタイムアウト時間を設定します
デフォルト値	60*1000 (単位 : ms)

- jeus.net.ping.period

説明	Pingメッセージの送信周期を設定します。 値が0の場合、Pingを送信して相手ノードが有効であることを確認しません
デフォルト値	10*60*1000 (単位 : ms)

- jeus.net.nio.select.limit

説明	非ブロッキング・モードで動作する場合、select()の呼び出し回数がこの回数以上失敗した場合、新しいセクターを作成して代替します
デフォルト値	5

- jeus.net.nio.write.limit

説明	非ブロッキング・モードで動作する場合、書き込みスレッドがブロックされるまでに書き込みキューに蓄積されるメッセージのサイズを設定します。書き込めない状況の場合、書き込みキューに入ります。 指定したバイト数を超えた場合は書き込みスレッドをブロックします
デフォルト値	10* 1024 (単位 : byte)

- jeus.net.nio.write.restart

説明	非ブロッキング・モードで動作する場合、書き込みスレッドの再開時に書き込みキューに蓄積されているメッセージのサイズを設定します。書き込みイベントで書き込みキューのバイト数が指定のバイト数より小さくなると、ブロックされている書き込みスレッドがすべて開始されます
-----------	----------------------------------------------------------------------------------------------------------------------------------

デフォルト値	3*1024 (単位 : byte)
---------------	--------------------

- jeus.net.nio.write.repeat.count

説明	1つのメッセージを最後まで書き込むために書き込みを行う回数を設定します。 1つのソケットに大きなメッセージがあり、セクター・スレッドがそれを書き込むことに集中してしまつて他のソケットのメッセージを書き込むことができない場合に使用します。複数のソケットに使用されるメッセージのサイズがすべて大きい場合、このオプションを適用すると応答はむしろ遅れることがあるので注意が必要です
デフォルト値	-1(無限大)

- jeus.net.reuseAddress

説明	サーバー・ソケットのSO_REUSEADDRを設定します
デフォルト値	false

- jeus.ssl.enabledProtocols

説明	SSL認証に使用するSSLプロトコル・バージョンを設定します。それぞれのプロトコル・バージョンはコンマ(,)で区切って記述します
デフォルト値	null

- jeus.net.client.use-ssl

説明	JEUSに接続するクライアントがSSLを使用するか否かを設定します。他の設定は別途のシステム・プロパティで設定します
デフォルト値	false

- jeus.net.client.ssl-properties-file

説明	JEUSに接続するクライアントが使用する設定が含まれているプロパティ・ファイル(Properties File)を設定します
デフォルト値	null

- jeus.ssl.keystoreまたはjavax.net.ssl.keyStore

説明	SSL認証に使用するキーストアのパスを設定します
-----------	--------------------------

- jeus.ssl.keypassまたはjavax.net.ssl.keyStorePassword

説明	SSL認証に使用するキーストアのパスワードを設定します
-----------	-----------------------------

- jeus.ssl.keystorepass

説明	SSL認証に使用するキーストアのキー・パスワードを設定します
-----------	--------------------------------

- jeus.ssl.truststoreまたはjavax.net.ssl.TrustStore

説明	SSL認証に使用するトラストストアのパスを設定します
-----------	----------------------------

- jeus.ssl.trustpassまたはjavax.net.ssl.TrustStorePassword

説明	SSLに使用するトラストストアのパスワードを設定します
-----------	-----------------------------

1.4. JNDIのシステム・プロパティ

本節では、JNDIのシステム・プロパティについて説明します。

- java.naming.factory.initial (必須)

説明	JNDIサービス・プロバイダーのInitialContext Factoryクラス名を設定します。このFactoryクラスを通じてInitialContextを作成します。このプロパティにjeus.jndi.JNSContextFactoryを格納してJEUSのInitialContextを作成します
デフォルト値	jeus.jndi.JNSContextFactory

- java.naming.factory.url.pkgs

説明	JEUSのInitialContextでURLスキーマを使用して検索するために設定します
デフォルト値	jeus.jndi.jns.url

- java.naming.provider.url

説明	JNDIサービスを提供するMSのDNSホスト名とポートを定義します。つまり、JEUSのJNSServerのIPアドレスを設定します。クラスタリングされているMSにJNDIを要求をする際は、クラスタリングされているすべてのサーバーのアドレスをコンマ(,)で区分する必要があります。 例) yohan:9736, yohan1:9836
デフォルト値	127.0.0.1:9736

- java.naming.security.principal

説明	JEUS Naming Serverで認証を受けるためのユーザー名を入力します。ユーザー名が設定されていない場合は「anonymous」と処理されます。 ユーザー名はJEUS Securityドメインに定義されている必要があります
デフォルト値	anonymous

- java.naming.security.credentials

説明	JEUS Naming Serverで認証を受けるためのパスワードを入力します
デフォルト値	N/A

- jeus.jndi.jns.resolution

説明	JNSClientがリソースを管理するチェック時間の周期を指定します
デフォルト値	30 * 1000 (単位 : ms)

- jeus.jndi.jns.connecttimeout

説明	JNSClientがJNSServerに接続する際のタイムアウト時間を指定します。 接続の試行が指定時間を超えた場合はjeus.net.ConnectExceptionが発生します
デフォルト値	0 (無限大を意味し、OSによって異なることがあります) (単位 : ms)

- jeus.jndi.jns.connectionduration

説明	JNSClientがJNSServerに接続したコネクションを維持する時間を設定します。指定時間の間にコネクションを使用しなければコネクションが自動的に終了します。 デフォルト値は0で、コネクションが維持されます
デフォルト値	0 (単位 : ms)

- jeus.jndi.local.tmout

説明	JNDIサービスでクライアントの役割をするJNSClientで要求を送信後、応答が来るまで待機する時間を指定します。指定時間を過ぎてもJNSServerから応答が来ない場合はjeus.util.WaitTimeoutExceptionが発生します
デフォルト値	20 * 1000 (単位 : ms)

- jeus.jndi.cluster.recheckto

説明	JEUSClusterContextの使用時に「failed」になったサーバーが再び使用可能になったかを確認する周期を指定します
デフォルト値	300 * 1000 (単位 : ms)

- jeus.jndi.clusterlink.selection-policy

説明	<p>クラスタリングされたサーバーにバインドされた、あるオブジェクトを検索して使用する場合、どのサーバーからオブジェクトを読み込むかについてのポリシーを設定します。</p> <p>以下のうち1つを選択できます。</p> <ul style="list-style-type: none"> – locallinkPreference : ローカルのMSにあるオブジェクトを使用します – roundrobin : 最初の要求ではランダムに選択したMSのオブジェクトを使用し、それ以降の要求からは1つずつ増やしてサーバーを選択します – random : クラスタリングされたMSからランダムに1つを選択して使用します <p>このシステム・プロパティを設定していない場合、jeus.jndi.clusterlink.prefer-local-nodeの設定内容によってデフォルトが変わります</p>
デフォルト値	locallinkPreference

- jeus.jndi.clusterlink.prefer-local-node

説明	<p>クラスタリングされたMSにバインドされた、あるオブジェクトを検索して使用する場合、ローカルのMSにあるオブジェクトを優先して使用するか否かを設定します。</p> <p>jeus.jndi.clusterlink.selection-policyにlocallinkPreference以外の値が設定されている場合はその値に従います</p>
デフォルト値	true

- jeus.jndi.enable.cache

説明	<p>JNDIクライアント(JNSClient)でキャッシュを使用するか否かを設定します。</p> <ul style="list-style-type: none"> – true : キャッシュを使用します(デフォルト値) – false : キャッシュを使用しません
デフォルト値	true

- jeus.jndi.jns.replicatebindings

説明	<p>バインドするオブジェクトをクラスタリングされたMS全体で共有するか否かを設定します。</p> <ul style="list-style-type: none"> – true : オブジェクトを共有します(デフォルト値) – false : オブジェクトを共有しません
デフォルト値	true

- jeus.jndi.jns.cachebindings

説明	<p>オブジェクトのバインド時にこのオプションを設定すると、JNSServerでバインドした後、JNSClientにそのオブジェクトをキャッシングします。</p> <ul style="list-style-type: none"> – true : オブジェクトをキャッシングします(デフォルト値) – false : オブジェクトをキャッシングしません
デフォルト値	true

- jeus.jndi.jns.localbindings

説明	<p>オブジェクトを自身のJVMにあるJNSClientにのみバインドします。</p> <ul style="list-style-type: none"> – true : オブジェクトをJEUSネーミング・クライアントにのみバインドします。このようにバインドしたオブジェクトはJNDI要求をするクライアントでのみ使用できます – false : オブジェクトをJEUSネーミング・サーバーにバインドします(デフォルト値)
デフォルト値	false

- jeus.jndi.compat

説明	<p>JNDI互換のための設定です。JEUS v7.0 Fix#2以降バージョンと以前バージョンの間でJNDIサービスを提供するには、このプロパティをtrueに設定する必要があります。</p> <ul style="list-style-type: none"> – true : JNDIサーバーがJEUS v7.0 Fix#2以前のバージョンであり、JNDIクライアントがJEUS v7.0 Fix#2を含む以降バージョンである場合、JNDI互換のためにtrueに設定します – false : JEUS v7.0 Fix#2以降バージョンと以前バージョンの間でJNDI互換を維持しません
デフォルト値	false

1.5. トランザクションのシステム・プロパティ

本節では、トランザクションのシステム・プロパティについて説明します。

- jeus.tm.forcedReg

説明	サブ・コーディネーターがルート・コーディネーターに登録される方式を決定します。 <ul style="list-style-type: none">– true : トランザクションが連動されるとすぐに登録します– false : RMが存在しなければ登録しません
デフォルト値	true

- jeus.tm.checkReg

説明	サブ・コーディネーターがルート・コーディネーターに登録されていることを確認するか否かを設定します
デフォルト値	true

- jeus.tm.disableJoin

説明	同じRMで作成されたリソースを結合するか否かを設定します
デフォルト値	false

- jeus.tm.noLogging

説明	トランザクション・マネージャーの復旧のためにトランザクション情報をロギングするか否かを設定します。 <ul style="list-style-type: none">– true : ロギングします– false : ロギングしません
デフォルト値	false

- jeus.tm.ignore.broken.log.file

説明	TMログ・ファイルが破損して復旧が行えない状況である場合、このオプションをtrueに指定すると、JEUSであらかじめファイルが破損したことを検知して全部削除します。ファイルが壊れたまま復旧を試行すると、ブートに失敗します
デフォルト値	false

- jeus.tm.removeIncompleteTx

説明	復旧対象のXAリソースの復旧が終われば復旧リストから削除するか否かを設定します。 手動での復旧が必要な状況に備えるにはfalseと指定します
デフォルト値	false

- jeus.tm.recoveryTrial

説明	復旧に失敗したリソースに対する、復旧の再試行回数を設定します。 復旧の再試行はバックグラウンドで行われます
デフォルト値	30

- jeus.tm.recoveryInterval

説明	復旧に失敗したリソースに対する、復旧の再試行の周期を設定します
デフォルト値	120000 (単位 : ms)

- jeus.tm.destroy-timeout

説明	トランザクションがタイムアウト(アクティブ・タイムアウト)になっても、明示的なコミットとロールバックを受けられなかった場合、トランザクション・マネージャーはタイムアウトした該当リソースを引き続き維持することになります。 これについて、クリーンアップ作業を行うタイムアウトを指定します。つまり、この設定値以上の時間が過ぎると、タイムアウトしたトランザクションに対して内部で消去作業を実行します
デフォルト値	1800000 (単位 : ms)

- jeus.tm.destroy-timeout-check-interval

説明	destroy timeoutをチェックする周期を設定します。 設定周期を通じてjeus.tm.destroy-timeoutの動作可否を確認します
デフォルト値	1800000 (単位 : ms)

- jeus.tm.ots.recoveryInterval

説明	OTSの復旧時にコーディネーターが準備状態になっていない場合の再試行の周期を設定します
デフォルト値	10000 (単位 : ms)

- jeus.tm.ots.decisionTO

説明	伝播されてprepareが呼び出された後に、この値に指定されている時間内にコミット/ロールバックがなければ、リカバリー・コーディネーターのreplay_completionを使用してコミット/ロールバックします
デフォルト値	60000 (単位 : ms)

- jeus.tm.net.address-mapping-properties

説明	JEUS TM間で通信を行う場合に使用するIPアドレスをマッピングさせるプロパティ・ファイルを指定します。詳細については『JEUS サーバガイド』の「7.7. IP帯域が異なるサーバー間のトランザクション通信問題」を参照してください
デフォルト値	プロパティ・ファイルの位置を指定していない場合は、この機能は使用しません

- jeus.tm.profile.classes

説明	TM Profileのために使用されるProfileListenerクラスを指定します。 プロファイル機能についての詳細は『JEUS サーバガイド』の「7.6. トランザクション・プロファイル機能」を参照してください
デフォルト値	なし

1.5.1. TM Clientのためのシステム・プロパティ

クライアントでTMを使用する場合はXMLの設定を使用できないため、システム・プロパティで設定します。

- jeus.tm.not_use

説明	トランザクション・マネージャーを使用するか否かを設定します。 – true : クライアントの場合、JNDIを使用するものの、トランザクションは使用しない場合を意味します – false : クライアントの場合、JNDIとトランザクションを使用する場合を意味します
デフォルト値	false

- jeus.tm.version

説明	使用するTMランタイム・バージョンを設定します。 – client : クライアントのTMを使用します
----	------------------------------------------------------------

	– server : サーバーのTMを使用します
デフォルト値	client

- jeus.tm.port

説明	JEUSTランザクション・マネージャーが使用するTCP/IPのポート番号を設定します
-----------	--------------------------------------------

- jeus.tm.usenio

説明	トランザクション・マネージャー間のI/O方式を設定します。 – true : NIO – false : ブロッキングI/O
デフォルト値	true

- jeus.tm.tmMin

説明	トランザクションで使用するスレッド・プールの最小サイズを設定します
デフォルト値	2

- jeus.tm.tmMax

説明	トランザクションで使用するスレッド・プールの最大サイズを設定します
デフォルト値	30

- jeus.tm.activeto

説明	トランザクションの開始からコミットやロールバックを行うまでの制限時間を設定します。 制限時間を超えた場合は強制的にロールバックします
デフォルト値	600000 (単位 : ms)

- jeus.tm.prepareto

説明	ルート・コーディネーターがサブ・コーディネーターから準備完了信号を受けるまでの制限時間を設定します。 制限時間を超えた場合は強制的にロールバックします
デフォルト値	120000 (単位 : ms)

- jeus.tm.preparedto

説明	サブ・コーディネーターがルート・コーディネーターに準備完了信号を送信後、決定を待つ制限時間を設定します。 制限時間を超えた場合は、決定待機信号をルート・コーディネーターに繰り返し送信します
デフォルト値	60000 (単位 : ms)

- jeus.tm.committo

説明	ルート・コーディネーターがサブ・コーディネーターにコミット信号を送信後、応答を受信しなければならない制限時間を設定します。制限時間を超えた場合、未解決のトランザクションとして記録します
デフォルト値	240000 (単位 : ms)

- jeus.tm.recoveryto

説明	ルート・コーディネーターがサブ・コーディネーターから復旧情報を受信しなければならない制限時間を設定します。制限時間を超えた場合は進行が止まります
デフォルト値	120000 (単位 : ms)

- jeus.tm.incomplete.to

説明	未解決のトランザクションが保存される時間を設定します
デフォルト値	86400000 (単位 : ms)

1.5.2. 下位の互換性のためのシステム・プロパティ

- jeus.tm.propagation.recalculated-timeout

説明	トランザクションが他のコンテナに伝播される際、伝播される前に消費した時間をアクティブ・タイムアウトから減らし、伝播させる設定です。 JEUS v6.0 Fix#9から提供される設定で、下位の互換性のためにはfalseと指定します
デフォルト値	false

1.6. Webエンジンのプロパティ

Webエンジンで提供するすべてのプロパティは、JVMシステムのプロパティだけでなく、バーチャル・ホスト別、Webアプリケーション別に適用できます。適用方法については『JEUS Webエンジンガイド』の「5.4. 仮想ホストの設定」、『JEUS Webエンジンガイド』の「5.4. 仮想ホストの設定」、『JEUS Webエンジンガイド』の「3.3.1. jeus-web-dd.xmlの設定」を参照してください。

参考

JVMシステム・プロパティやWebエンジンのレベルで設定するよりは、バーチャル・ホストまたはWebアプリケーションのレベルで設定することを推奨します。

1.6.1. Webエンジン・レベルのプロパティ

domain.xmlのWebエンジン・レベルで設定できるプロパティです。バーチャル・ホスト、jeus-web-dd.xmlに設定しても無視されます。

- jeus.servlet.prevent-forcibly-threadlocal-memoryleak

説明	Webアプリケーションでスレッド・ローカル変数を使用後、これを解除していない場合に発生するメモリー・リークをWebエンジンで該当Webアプリケーションがリロードされる場合や、再デプロイ、アンデプロイされる場合に、強制的にスレッド・ローカル変数を初期化します。しかし、このオプションを使用した場合、他のWebアプリケーションで設定したスレッド・ローカル変数も初期化されるため、注意してください
デフォルト値	false

- jeus.servlet.webadmin.suspendCmd.tmout

説明	WebtoBに一時停止信号を送信後、すべてのWebtoBリクエスト・プロセッサが再度WebtoBから一時停止信号を受けてsuspend waitを行うまでの最大待機時間を設定します
デフォルト値	120000L

- jeus.servlet.request.enableDns

説明	ServletReqeust.getRemoteHost()を呼び出す際にDNSでホスト名を取得するか否かを設定します。WebエンジンがあるサーバーのDNSアクセス可否を設定することであるため、Webエンジン・レベルのプロパティで提供します
デフォルト値	true

- jeus.servlet.loader.jspcount

説明	Webエンジンで同時にロードできるJSPの数です
デフォルト値	0

- jeus.servlet.useCaseInsensitiveFileSystem

説明	Linux OSとWindowsファイル・サーバーの組み合わせのように、大小文字を区別するOSに大小文字を区別しないファイル・サーバーを使用するといった組み合わせを使用する場合、要求アドレスの大小文字を区別しないため、期待した動作をしないことがあります。このプロパティを使用すると、このような組み合わせでも大小文字を区別して処理すると共に、Windowsでのみ動作する次のようなプロパティも使用できるようになります <ul style="list-style-type: none"> – jeus.servlet.resource.ignore-filename-case – jeus.servlet.jsp.ignore-filename-case
デフォルト値	false

以下は、JVMシステム・プロパティだけで設定するオプションです。

- org.glassfish.web.rfc2109_cookie_names_enforced

説明	クッキーの名前にRFC2109でサポートしていない文字が含まれている場合、IllegalArgumentExceptionが発生します。javax.servlet.http.Cookieに定義されている内容ですが、JEUS 6まではAPI実装で正常に処理されていませんでした
デフォルト値	true

1.6.2. バーチャル・ホスト・レベルのプロパティ

domain.xmlのWebエンジン設定でバーチャル・ホストに設定できるプロパティです。jeus-web-dd.xmlに設定しても無視されます。

参考

現在このレベルのプロパティは存在しません。

1.6.3. Webアプリケーション・レベルのプロパティ

jeus-web-dd.xmlに設定して、Webアプリケーション別に適用できるプロパティです。このプロパティはWebエンジンまたはバーチャル・ホスト・レベルのプロパティで使用可能です。

- jeus.application.login.key

説明	JEUS Login Managerを使用する場合、ログインに使用する属性キーを設定します
デフォルト値	JEUS_LOGIN_KEY

1.6.3.1. 共通またはサーブレット・エンジン関連のプロパティ

サーブレットまたはJSPエンジンに適用されるプロパティです。

- jeus.servlet.response.cookie.version0Style

説明	version0スタイルのクッキー・ヘッダーを使用するか否かを設定します
デフォルト値	false

- jeus.servlet.welcomefile.sendRedirect

説明	<p>HttpServletResponse.sendRedirect()を通じてwelcomeファイルを送信するか否かを設定します。</p> <ul style="list-style-type: none"> – true : HttpServletResponse.sendRedirect()を使用します – false : RequestDispatcher.foward()を使用します
デフォルト値	false

- jeus.servlet.filter.applyOnWelcomeFileRequest

説明	コンテキスト・ルートへの要求に対するwelcomeファイルを処理するときに、フィルターを適用するか否かを設定します
デフォルト値	true

- jeus.servlet.response.header.serverInfo

説明	ヘッダーにJEUSのバージョン情報を含めるか否かを設定します
デフォルト値	false

- jeus.servlet.keep-original-request-on-forward

説明	Tomcatと同じようにオリジナル・リクエストに設定されているリクエストURIなどのようなパス情報を変更しません
デフォルト値	false

- jeus.servlet.keep-original-query-string-on-forward

説明	フォワード時にオリジナルのクエリー・ストリングをRequestオブジェクトにそのままにしておくか、あるいはフォワード時に転送したクエリー・ストリングで上書きするかを設定します
デフォルト値	false

- jeus.servlet.keep-original-query-string-on-include

説明	インクルード時にオリジナルのクエリー・ストリングをRequestオブジェクトにそのままにしておくか、あるいはインクルード時に転送したクエリー・ストリングで上書きするかを設定します
デフォルト値	false

- jeus.servlet.out.ensureContentOrder

説明	クライアントに応答を書き出すresponse.getOutputStream()、response.getWriter()、JspWriter(jsp out)を組み合わせる場合の出力順序を維持するか否かを設定します。ただし、JEUSではこの3つを組み合わせる使用せずに、いずれか1つを使用することをお勧めします
デフォルト値	false

- jeus.servlet.classloader.allowServletAPI

説明	アプリケーションのクラスのうち、サーブレットAPI関連のクラスをロードするか否かを設定します
デフォルト値	false

- jeus.servlet.response.useRequestHttpVersion

説明	HTTP応答ヘッダーのHTTPバージョンをHTTPクライアント・バージョンに設定します。以前提供していたjeus.servlet.response.force-response-10を、この設定に変更してください
デフォルト値	false

- jeus.servlet.context.attribute.serialize

説明	Context Autoのリロード時にServletContext attributeをシリアライズして維持するか否かを設定します
デフォルト値	false

- jeus.servlet.resource.ignore-filename-case

説明	リソース・サーブレットでリソース・ファイル名の太文字と小文字を区別するか否かを設定します。 Windows OS、またはjeus.servlet.useCaseInsensitiveFileSystemがtrueの場合に有効です
デフォルト値	false

- jeus.servlet.response.header.encoding

説明	応答ヘッダーに使用するキャラクタ・エンコーディングを設定します
デフォルト値	null

- jeus.servlet.scan-servlet-container-initializers

説明	Servlet 3.0から追加されたServletContainerInitializerメカニズムを使用するか否かを決定します。サーブレット標準に準拠するために、基本的にスキャン作業を実行しますが、これによってデプロイ時間が長くなることがあります
デフォルト値	true

- jeus.servlet.sortWebinfLibraries

説明	WEB-INF/libに含まれているjarファイルをロードするときにソートするオプションです <ul style="list-style-type: none"> – name_asc : ファイル名を基準に昇順でソート – name_dsc : ファイル名を基準に降順でソート – time_asc : ファイルのLMFを基準に昇順でソート – time_dsc : ファイルのLMFを基準に降順でソート
デフォルト値	null

- jeus.servlet.useMetadataCompleteDeploy

説明	Java EE 5からはWebアプリケーションのデプロイ時に、必ずJava EE標準で定義されたアノテーションが定義されているクラスを探すことになっています。 WEB-INF/classesのクラス・ファイルだけでなく、WEB-INF/libに含まれているクラスも探すため、デプロイ時間が長くなることがあります。クラスの探索を実行しないためには、web.xmlのmetadata-complete設定をtrueに指定する必要があります。アプリケーションに含まれているweb.xmlを修正せずに、jeus-web-dd.xmlを追加してmetadata-complete設定をtrueにできます
-----------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

デフォルト値	false
--------	-------

1.6.3.2. JSPエンジン関連のプロパティ

JSPエンジンでのみ使用するプロパティです。

- jeus.servlet.jsp.reload

説明	JSPリロードをサポートするか否かを設定します
デフォルト値	true

- jeus.servlet.jsp.precompile

説明	JSPプリコンパイルをサポートするか否かを設定します(プリコンパイル要求パラメータによってJSPを呼び出すときにコンパイルのみを実行する機能)
デフォルト値	true

- jeus.servlet.jsp.print.null.as.emptystring

説明	JSPでnullを出力時に空白文字を使用するか否かを設定します
デフォルト値	false

- jeus.servlet.jsp.jspwriter.null

説明	JSPでnullを出力時に代替する文字列を設定します
デフォルト値	"null"

- jeus.servlet.jsp.compile.retrycount

説明	JSPで作成したサーブレットをロードできない場合、再コンパイルを試行する回数を設定します
デフォルト値	2

- jeus.servlet.jsp.compile.delay

説明	FTPのアップロード時のJSP遅延コンパイル時間を設定します
デフォルト値	0L (単位 : ms)

- jeus.servlet.jsp.helpReloadNativeLibrariesByCallingSystemGC

説明	JSPのリロード時にJSPクラス・ローダーが使用したネイティブ・ライブラリーが正常にクリーンアップされないJVM問題を解決するための方法です。JSPクラス・ローダーがGC(Garbage Collection)されるとき、ネイティブ・ライブラリーが処理されるため、JEUSではSystem.gc()を呼び出せるだけであり、100% GCされるという保証はありません。そのため、JSP内でSystem.loadLibrary()などを使用する場合、リロードに失敗することがあります
デフォルト値	false

注

System.gc()とSystem.runFinalization()を呼び出すため、システムの性能を低下させることがあります。したがって、ネイティブ・ライブラリーを使用するJSPを実際にサービスを運用中に修正しないことを推奨します。最もいい方法は、ネイティブ・ライブラリーを使用するWebアプリケーションを別途作成して、そこにdispatchする方法です。そのWebアプリケーションは、コンテキストの自動リロードや再デプロイを行ってはいけません。

- jeus.servlet.jsp.ignore-filename-case

説明	JSPファイル名の大小文字を区別するか否かを設定します。Windows OS、または jeus.servlet.useCaseInsensitiveFileSystemがtrueの場合に有効です
デフォルト値	false

- jeus.servlet.jsp.assure-utf8-file-encoding-detection

説明	JasperでJSPファイルのエンコードがUTF-8であると確信できない場合でも、これをUTF-8と扱います
デフォルト値	false

注意

included JSPファイルにもpageEncodingタグを記述してください。

1.6.3.3. 互換性提供プロパティ

JEUSは基本的に、サーブレットおよびJSP標準を遵守する必要があります。しかし、標準解釈エラー、要求事項の実装欠落など、開発者のミスまたは不十分な説明、標準技術文書とAPI文書間の説明不一致などによる標準文書のエラーによりAPIの動作が異なることがあります。

- jeus.servlet.jsp.modern

説明	falseと設定した場合、JEUS 4互換のJSPエンジンを使用します。この場合、JSP 1.2バージョンのみサポートします。
----	-----------------------------------------------------------------

	該当JSPエンジンは推奨されていないため、使用しないことをお勧めします。この互換性プロパティも次のバージョンまたは次のFixで削除される予定です
デフォルト値	true

- jeus.servlet.request.url.allowNonStandardPercentCharacterUsage

説明	URLに単独で「%」文字の使用を許容します。しかし、RFC 2396, 2.4.3. Excluded US-ASCII 文字に従うと、「%」文字はURLエスケープのための予約語であり、単独文字として使用することはできず、決められた形式でのみ使用可能です。したがって、クライアントでこれを「%25」とエスケープして送信します
デフォルト値	false

- jeus.servlet.request.6CompatibleSetCharacterEncoding

説明	<p>ServletRequest.setCharacterEncoding()で設定したエンコーディングは、必ずリクエスト・ボディにのみ適用する必要があります。</p> <p>しかし、JEUS 6でこれを遵守していないバグが発見され、Query String、Cookieのように、リクエスト・ヘッダーを読み取る際も使用します。JEUS 6の正しくない動作に基づいて作成したアプリケーションを修正なくサポートする必要がある場合は、このオプションを使用してください</p>
デフォルト値	false

- jeus.servlet.response.defaultContentType

説明	コンテンツ・タイプの応答ヘッダーに設定したデフォルト値です。この値がnullではない場合、以下のような状況で応答ヘッダーを設定します。ServletResponse.setCharacterEncoding()やServletResponse.setLocale()の呼び出し時、サーブレットが明示的にServletResponse.setContentType()を呼び出さなかった場合、このプロパティ値をコンテンツ・タイプのヘッダーとして設定します。しかし、このような動作はコンテンツ・タイプの応答ヘッダーに関するサーブレット標準に準拠しません
デフォルト値	null

参考

サーブレットが明示的にServletResponse.setContentType()を呼び出さなかった場合、応答ヘッダーにコンテンツ・タイプのヘッダーを指定してはならないとサーブレット標準で規定しています。また、ServletResponse.setContentType()を呼び出す際はnullを返すようにしています。これはServlet 2.3でも明示されている内容で、JEUSでは遵守せずにいた事項です。

したがって、JEUS v7.0 Fix#1からはこれを遵守するためにデフォルト値をnullに変更しています。ここに依存してアプリケーションが動作していた場合、適切なプロパティ値を設定してください。

- jeus.servlet.response.applyContentLanguageProperlyWithSetLocaleAPI

説明	ServletResponse.getWriter()の呼び出し後にServletResponse.setLocale()を呼び出した場合、Content-Languageの応答ヘッダーは必ず設定されている必要があります。キャラクター・エンコーディングは変わりません。意図しないContent-Languageの応答ヘッダーが設定されている場合、この値をfalseに指定します
デフォルト値	true

- jeus.servlet.response.6CompatibleForcedEncoding

説明	xml設定で適用する<response-encoding><forced>の適用優先順位を、本来の意味と合わないように実装されたJEUS 6方式で動作させます。本来、forcedオプションは優先順位が最も高いです。アプリケーションAPIにより設定し、JSPファイルのpageタグのcontentTypeに設定されたHTTP応答エンコーディングに優先します
デフォルト値	false

- jeus.servlet.response.6CompatibleSetCharacterEncoding

説明	ServletResponse.setCharacterEncoding()で設定したエンコーディングは、forcedより優先順位が低くなければなりません。 しかし、JEUS 6以前まではこれを遵守していないバグが発見され、常にforced設定を無視する問題があります。もしこのような動作に依存して実装されたアプリケーションがあり、この動作を維持しようとする場合は、このオプションをtrueに設定します。 リクエスト側と名前が似ているので、取り違えないように注意してください
デフォルト値	false

- jeus.servlet.session.invalidateBySetMaxInactiveIntervalZero

説明	HttpSession.setMaxInactiveInterval(int timeout)は、仕様に基づいて、0以下の値は削除されてはならないセッションを意味します。 以前の仕様では、タイムアウトを「0」に設定した場合にセッションを削除してはならないということが明示されていなかったため、設定値「0」を設定してセッションを削除するケースが発生しました。 そのように使用していたユーザーの互換性のために設定を提供します
デフォルト値	false

- jeus.servlet.engine.disableRequestFinishCheck

説明	RequestオブジェクトはサーブレットのserviceメソッドまたはフィルターのdoFilterメソッド内でのみ有効であり、すでに処理済みのRequestオブジェクトを使用しようとすると、IllegalStateExceptionを発生させるように仕様に定義されています。しかし、JEUS 7以前のバージョンまではこの確認を行っていなかったため、処理済みのRequestオブジェクトを使用するアプリケーションが正常に動作しました。 既存のアプリケーションを使用するユーザーの互換性のために設定を提供します
デフォルト値	false

- jeus.servlet.ignoreWebComponentFailureWhenDeploying

説明	デプロイの際に、load-on-startupで登録したサーブレットのinit()、Filter.init()、ServletContextListenerで例外が発生してもデプロイは失敗しません
デフォルト値	false

1.7. セッション・サーバーのシステム・プロパティ

本節では、セッション・サーバーのシステム・プロパティについて説明します。

- jeus.sessionmanager.dumpbackup.max

説明	dumpbackupを実行するcount単位を設定します。 バックアップするサーバーが再起動する場合、dumpbackupを実行します。このとき、保持しているセッションが多すぎる場合は、一定の単位に分けてバックアップを実行します。 Session Cluster設定画面のProperties項目に設定する場合はすべてのクラスターに適用され、Session Cluster Config設定画面のProperties項目に設定する場合は該当するクラスターにのみ適用されます
デフォルト値	100 (個)

- jeus.sessionmanager.receive.backup.despite.no.deploy

説明	DOMAIN_WIDEモードを使用するとき、アプリケーションをデプロイしなくても、リモートからバックアップすることができるようにダミー・アプリケーションをデプロイします
デフォルト値	false

1.8. EJBのシステム・プロパティ

本節では、EJBのシステム・プロパティについて説明します。

1.8.1. EJBのシステム・プロパティ

以下は、EJBのシステム・プロパティについての説明です。

- `jeus.ejb.transaction.attribute.default`

説明	Container-managed Transaction Beanのメソッドにトランザクション属性が設定されていない場合に使用する属性を設定します。 以下のうち1つを設定します。 <ul style="list-style-type: none">– NotSupported– Supports– Required– RequiresNew– Mandatory– Nerver 属性の優先順位は以下のとおりです。 <ol style="list-style-type: none">1. <code>ejb-jar/trans-attribute</code>2. <code>jeus-ejb-dd/unspecified-container-transaction</code>3. システム・プロパティ
デフォルト値	<ul style="list-style-type: none">– EJB 3.0以上のBeanの場合 : Required (スペック基準)– EJB 3.0未満のBeanの場合 : Supports (下位互換性を維持するため)– Timer Serviceメソッドの場合 : NotSupported

- `jeus.ejb.sharedcolumn.modify`

説明	CMPエンティティBeanで列共有されているcmp、cmr fieldを1つのトランザクションで両方とも修正することを許可するか否かを決定します
デフォルト値	false

- jeus.ejb.checkTable

説明	CMPエンティティBeanをデプロイする際に、データベースのテーブルの存在有無、列の存在有無を確認するか否かを設定します
デフォルト値	false

- jeus.ejb.enable.configDeleteOption

説明	<p>CMPエンティティBeanをアンデプロイする際に、データベースのテーブルを削除することを許可するか否かを設定します。</p> <ul style="list-style-type: none"> – true : <deleting-table>の設定が適用されます – false : <deleting-table>の設定が無視されます
デフォルト値	false

- jeus.ejb.csi.trusthosts

説明	CORBA CSIV2を使用する際のトラスト・ホストです
-----------	------------------------------

- jeus.client.interop

説明	クライアントJVMでCORBA interoperabilityを使用するか否かを設定します
デフォルト値	false

- jeus.ejb.csi.defaultUser

説明	CORBA CSIV2を使用する際のデフォルト・ユーザーを設定します
-----------	------------------------------------

- jeus.ejb.notreload

説明	<p>sharedクラスローダーを使用するEJBがデプロイ、アンデプロイ、再デプロイされる際、リロードから除外されるWebモジュールの一覧です。</p> <p>以下のフォーマットで設定します</p> <pre>context_name1;context_name2</pre>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------

- jeus.corba.props

説明	EJBで使用するORBのプロパティです。
-----------	----------------------

	key=value[,key=value]*
--	------------------------

- jeus.ejb.cluster.useip

説明	<p>EJBでアクティブ・スタブを使用する場合、サーバーに接続するためのアドレスをIPアドレスにするか、あるいはホスト名にするかを設定します。</p> <p>ここでは、jeus.net.localipで指定されたIPを使用します。</p> <p>このオプションをtrueに指定すると、classftpのアドレスもIPアドレスを使用することになります</p>
デフォルト値	false

- jeus.ejb.all.disconnect

説明	<p>すべてのBeanタイプに適用されるディスコネクト・タイムアウトを設定します。</p> <p>-1と設定した場合はディスコネクト機能を使用しません。</p> <p>[Deprecated] EJB 3.1 specでは、@StatefulTimeoutまたはejb-jar.xmlの<stateful-timetout>と設定可能です</p>
デフォルト値	3600000 (単位 : ms)

- jeus.ejb.entity.disconnect

説明	<p>エンティティーBeanに適用されるディスコネクト・タイムアウトを設定します(単位 : ms)。</p> <p>jeus.ejb.all.disconnect設定より優先されます。</p> <p>-1と設定した場合はディスコネクト機能を使用しません</p>
-----------	------------------------------------------------------------------------------------------------------------------------------------

- jeus.ejb.stateful.disconnect

説明	<p>ステートフル・セッションBeanに適用されるディスコネクト・タイムアウトを設定します(単位 : ms)。</p> <p>jeus.ejb.all.disconnect設定より優先されます。</p> <p>-1と設定した場合はディスコネクト機能を使用しません。</p> <p>[Deprecated] EJB 3.1 specでは、@StatefulTimeoutまたはejb-jar.xmlの<stateful-timetout>と設定可能です</p>
-----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- jeus.ejb.all.passivate

説明	すべてのBeanタイプに適用されるパッシベーション・タイムアウトを設定します。 -1と設定した場合はディスコネクト機能を使用しません
デフォルト値	300000(単位 : ms)

- jeus.ejb.entity.passivate

説明	エンティティーBeanに適用されるパッシベーション・タイムアウトを設定します(単位 : ms)。 jeus.ejb.all.passivate設定より優先されます。 -1と設定した場合はディスコネクト機能を使用しません
-----------	-----------------------------------------------------------------------------------------------------------------------------

- jeus.ejb.stateful.passivate

説明	ステートフル・セッションBeanに適用されるパッシベーション・タイムアウトを設定します(単位 : ms)。 jeus.ejb.all.passivate設定より優先されます。 -1と設定した場合はディスコネクト機能を使用しません
-----------	----------------------------------------------------------------------------------------------------------------------------------

- jeus.ejb.exportPort

説明	EJBで使用するデフォルトのエクスポート・ポートを設定します。 jeus-ejb-dd.xmlのexport portが0のEJBは、この値を使用します。 0であればRMIが提供するデフォルト・ポートを使用します
デフォルト値	Managed Server base port + 7

- jeus.ejb.jeusrmi.exportPort

説明	JEUS RMIを使用するEJBで使用するデフォルトのエクスポート・ポートを指定します。この値を指定すると、jeus-ejb-dd.xmlのエクスポート・ポートが0のEJBはこの値を使用します
デフォルト値	Managed Server base port + 11

- jeus.ejb.entity.lock-wait-timeout

説明	同じエンティティーを使用する際、ロックを待機するタイムアウト値です。SINGLEまたはEXCLUSIVEの場合にのみ適用されます(単位 : ms)
-----------	---------------------------------------------------------------------------

デフォルト値	0(無限待機)
---------------	---------

- jeus.ejb.timer.generate-table-ddl

説明	EJBタイマー・ハンドルを保存する表を作成するためのDDLをファイルに作成します。内部的な用途でのみ使用する必要があります。JEUSを実行したディレクトリーにcreateDDL.jdbcという名前で作成されます
デフォルト値	false

- jeus.ejb.generate-no-interface-view-files

説明	no-interface-viewのEJBのローカル・ビジネス・インターフェースのクラス・ファイルを作成します。「_generated_」フォルダ下位にクラス・ファイルが作成されます
デフォルト値	false

- jeus.ejb.cluster.selection-policy

説明	<p>クラスタリングされたEJBエンジンに属するEJBを呼び出す場合、どのEJBエンジンのBeanを選択するかに関するポリシーを設定します。(EJB 3ステートレスBeanに限る。詳細については、『JEUS EJBガイド』のクラスタリング関連内容を参照)</p> <p>以下のうち1つを選択できます</p> <ul style="list-style-type: none"> – locallinkPreference : ローカルMSにあるオブジェクトを使用するようにします – roundrobin : 最初の要求ではランダムに選択したMSのオブジェクトを使用し、その後の要求からは1ずつ増やしながらサーバーを選択します – random : クラスタリングされたMSのうちランダムに1つを選択して使用します
デフォルト値	locallinkPreference

1.8.2. JPAのシステム・プロパティ

以下は、JPAのシステム・プロパティについての説明です。

- jeus.persistence.defaultProvider

説明	デフォルトのJava Persistence APIのプロバイダー・クラスの名前です
-----------	--------------------------------------------

1.8.3. プロパティ

以下は、RMIのシステム・プロパティについての説明です。

参考

プロパティの説明において、[client]はRMIを呼び出し、[server]はEJBがデプロイされている側に設定する必要があるという意味です。RMI関連の設定と、より多様なオプションについては、以下のサイトを参照してください。

<http://docs.oracle.com/javase/7/docs/technotes/guides/rmi/javarmiproperties.html>

<http://docs.oracle.com/javase/7/docs/technotes/guides/rmi/sunrmiproperties.html>

- java.rmi.server.codebase

説明	[server]RMIの使用時にEJB Beanクラス以外のクラスを渡す必要がある場合、クラス・アノテーションに入力するコードベースを設定します
----	--------------------------------------------------------------------------

- java.rmi.server.useLocalHostName

説明	[server]クライアントがサーバーに接続する際、サーバーのIPを使用せずにホスト名を使用する場合に使用します
デフォルト値	false

- java.rmi.server.hostname

説明	[server]クライアントがサーバーに接続する際に使用するホスト名です。 java.rmi.server.useLocalHostNameより優先順位は低いです
デフォルト値	ip

- jeus.ejb.rmi.jeus

説明	[server]JEUS RMIを使用するか否かを設定します。 jeus-ejb-dd.xml/jeus-bean/jeus-rmiを使用すると、Bean別に適用できます。このプロパティはEJB全体に適用する場合に使用します。DDの値が優先されます
デフォルト値	false

- jeus.ejb.rmi.socketfactory

説明	[client/server]JEUSが使用するRMIソケット・ファクトリー・クラスのfully qualified nameを指定します。指定されたクラスはEJBだけでなく、すべてのRMI通信に使用されます。 クライアントも同じRMIソケット・ファクトリーを使用しなければ通信はできません。このクラスはno argumentの構築子(constructor)を必ず提供する必要があり、JEUS_HOME/lib/applicationに位置します
デフォルト値	jeus.ejb.util.EJBRMISocketFactory

- jeus.ejb.operationTimeout

説明	[client]リモートの呼び出し後のタイムアウト時間です(SoTimeout)。 sun.rmi.transport.tcp.responseTimeout値をJEUSで上書きします
デフォルト値	5 * 60 * 1000

- sun.rmi.transport.connectionTimeout

説明	[client]クライアント側のRMIコネクションがアイドル状態で残っている最大時間を設定します(KeepAliveTimeout)
デフォルト値	15 * 1000

- sun.rmi.transport.tcp.readTimeout

説明	[server]サーバーでクライアントの要求を読み込むのに掛かる最大時間を設定します(SoTimeout)。 クライアントのSoTimeoutであるjeus.ejb.operationTimeout(sun.rmi.transport.tcp.responseTimeout)より大きい値を指定する必要があります
デフォルト値	2 * 60 * 60 * 1000

- sun.rmi.transport.tcp.maxConnectionThreads

説明	[server]サーバーで要求を処理するスレッド数の最大値を設定します。 関連するJEUS設定には、jeus-ejb-dd.xmlにBean別に設定する<thread-max>値があります。<thread-max>は各Beanに割り当てられるスレッド数の最大値で、maxConnectionThreadsの値をいくら大きく設定しても<thread-max>が小さい場合はスレッド数が増加しません。
-----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	たとえば、EJB AとEJB Bの2つのBeanが存在し、<thread-max>を別途設定せずにデフォルト値の100を使用すると、maxConnectionThreadsがInteger.MAX_VALUEであっても200までしか増加しません
デフォルト値	Integer.MAX_VALUE

- sun.rmi.transport.tcp.threadKeepAliveTime

説明	[server]スレッドの要求が終了後(アイドル状態で)、スレッドが他の要求を待つ最大時間を設定します(KeepAliveTimeout)
デフォルト値	1*60*1000

- jeus.ejb.compiler.rmic.fork

説明	[server]IIOPクラスの作成時にRMICコンパイラ・プロセスをフォークするか否かを設定します
デフォルト値	false

- jeus.ejb.compiler.rmic.verbose

説明	[server]IIOPクラスの作成時にverboseログを残すか否かを設定します
デフォルト値	false

- jeus.ejb.compiler.rmic.vcompat

説明	IIOPのコンパイル時、RMI 1.1、1.2 compatible protocolで作成するか否かを設定します
デフォルト値	false

- jeus.ejb.compiler.rmic.enable

説明	IIOPスタブ・ファイルを作成するか否かを設定します。一般的な場合はIIOPスタブ・ファイルが必要ありません
デフォルト値	false

- jeus.ejb.compiler.class.version

説明	[server]EJBで作成されたクラスをコンパイル時のソースとtarget versionを設定します
デフォルト値	1.4

- jeus.ejb.compiler.use-java-compiler-api

説明	JDK 6から提供するJavaコンパイラAPIを使用するようにするオプションです。 基本的には使用するようになっていますが、既存のようにJDKのlib/tools.jar内に含まれているSun内部のクラスを直接利用してコンパイルする方式をとる場合は、このオプションをfalseに設定します。 コンパイルされたクラスの違いによって互換性の問題などが発生した場合に使用できます
デフォルト値	true

JEUS RMIを使用する場合

以下は、JEUS RMIを使用する場合のシステム・プロパティについての説明です。

- jeus.rmi.jeusnet

説明	RMI通信にJEUSネットワーク層を使用するか否かを設定します
デフォルト値	false

- jeus.rmi.io.type

説明	JEUSネットワーク層を使用する際のI/Oタイプを設定します。 ブロッキング通信あるいは非ブロッキング通信のうち、いずれかを選択します(blocking/non-blocking)
デフォルト値	blocking

- jeus.rmi.blocking

説明	JEUSネットワーク層のブロッキング通信を使用する際のブロッキング・タイプを設定します。 ソケットあるいはソケット・チャンネルのうち、いずれかを選択します(socket/channel)
デフォルト値	socket

- jeus.rmi.blocking.client.num

説明	JEUSネットワーク層のブロッキング通信を使用する際のクライアント数の期待値を設定します。スレッド・プールの数を決める際に使用します
デフォルト値	50

- jeus.rmi.channel.buffer.size

説明	チャンネルを使用する場合のバッファ・サイズを設定します。 msgの最大サイズをバッファ・サイズに設定すると、バッファのサイズを最適化できます
デフォルト値	jeus.net.tcpbufferで指定した値

- jeus.rmi.response.timeout

説明	RMIから要求を送信後、応答があるまで待機する時間を設定します
デフォルト値	600*1000 (単位 : ms)

- jeus.rmi.client.enableProcessPool

説明	クライアントでのメッセージ処理を読み込みスレッドと別途行うか否かを設定します
デフォルト値	false

- jeus.rmi.server.enableProcessPool

説明	サーバーでのメッセージ処理を読み込みスレッドと別途行うか否かを設定します
デフォルト値	false

- jeus.rmi.client.threadPool

説明	enableClientProcessPoolをtrueと設定した場合のプロセス・プールのサイズを設定します
デフォルト値	50

- jeus.rmi.server.threadPool

説明	enableServerProcessPoolをtrueと設定した場合のプロセス・プールのサイズを設定します
デフォルト値	50

- jeus.rmi.usebaseport

説明	MSのデフォルト・ポートをRMIのデフォルト・ポートとして使用します (JEUS RMIIは例外)
デフォルト値	false

1.9. セキュリティーのシステム・プロパティ

本節では、セキュリティーのシステム・プロパティについて説明します。

1.9.1. Standard Java SE & Java EEセキュリティのシステム・プロパティ

以下は、Standard Java SE & Java EEセキュリティのシステム・プロパティについての説明です。

- java.security.manager

説明	コード保護によりJEUSサーバーの堅固性を向上させるためにJ2SE SecurityManagerを使用します。ただし、これはパフォーマンスを低下させる場合があります。 基本的にSecurityManagerはJEUSで使用されません
----	------------------------------------------------------------------------------------------------------------------------------

- jeus.security.enable.default.manager

説明	System.exitメソッドが呼び出されてもJEUS JVMプロセスは終了しません。 -Djeus.security.enable.default.manager=trueと設定すると、JDKのJava Security ManagerではないJava Security Managerを拡張したJEUS Security Managerが適用され、アプリケーション・プログラム・コードからSystem.exitメソッドが呼び出されてもJEUS JVMプロセスが終了しません
デフォルト値	false

- java.security.policy

説明	SecurityManagerで使用するJ2SEポリシー・ファイルのパスを設定します
デフォルト値	JAVA_HOME/lib/security/java.security

- javax.security.jacc.policy.provider

説明	JACCポリシーの実装クラス名を設定します
デフォルト値	jeus.security.impl.jacc.JACCPolicyWrapper

- javax.security.jacc.PolicyConfigurationFactory.provider

説明	JACC PolicyConfigurationFactoryの実装クラス名を設定します
デフォルト値	jeus.security.impl.jacc.JACCPolicyConfigurationFactoryImpl

1.9.2. JEUSセキュリティのシステム・プロパティ

以下は、JEUSセキュリティのシステム・プロパティについての説明です。

- jeus.security.enable

説明	JEUSの起動時にセキュリティー・サーバーをインストールするか否かを設定します
デフォルト値	true

- jeus.security.default.configDir

説明	セキュリティー・システムが使用するルート・ディレクトリーを設定します
デフォルト値	JEUS_HOME/domains/<domain-name>/config/security

- jeus.security.default.domainName

説明	デフォルトのドメイン名を設定します
デフォルト値	SYSTEM_DOMAIN

- jeus.security.installer.classname

説明	SecurityInstallerの実装クラス名を設定します
デフォルト値	jeus.security.impl.installer.JeusSecurityDomainInstaller

- jeus.security.jacc.principalRoleMapper

説明	jeus.security.impl.aznrep.JACCPPrincipalRoleMapperインターフェースを実装するクラス名を設定します。このクラスはJACC Providerでprincipal-roleマッピングを担当します
デフォルト値	jeus.security.impl.jacc.JACCDDefaultPrincipalRoleMapper

- jeus.net.ssl.security

説明	セキュリティー通信にSSLを使用するか否かを設定します
デフォルト値	false

- jeus.security.keypath

説明	security.keyファイルのパスを指定します
デフォルト値	JEUS内部で基本的に設定されます

- jeus.security.master

説明	マスター・パスワードを指定します
デフォルト値	セキュリティーを保つために、プロンプトでの入力を推奨します

- jeus.net.logListenPort

説明	サーバー・ソケットを作成する際、ポートのログイン可否を設定します。 jeus.security.enable.default.manager=trueを設定しなければ作動しません
デフォルト値	true

- jeus.net.logConnectInfo

説明	ソケットを作成する際に接続するホスト、ポートのログイン可否を設定します。 jeus.security.enable.default.manager=trueを設定しなければ作動しません
デフォルト値	false

- jeus.net.logListenStack

説明	サーバー・ソケットを作成する際のスレッド・スタック・トレースのログイン可否を設定します。 jeus.security.enable.default.manager=trueを設定しなければ作動しません
デフォルト値	false

- jeus.net.logConnectStack

説明	ソケットを作成し、接続する際のスレッド・スタック・トレースのログイン可否を設定します。 jeus.security.enable.default.manager=trueを設定しなければ作動しません
デフォルト値	false

- jeus.security.keylength

説明	暗号化アルゴリズム (AES/DES/DESe/SEED/Blowfishなど)を使用するとき、キー・サイズを指定します。jeus.security.keylength=256に設定すると、システム全体に統一された256ビットのキー・サイズが適用されます
デフォルト値	256

1.10. JEUS MQのシステム・プロパティ

本節では、JEUS MQのシステム・プロパティについて説明します。

1.10.1. JEUS MQのクライアント

以下は、JEUS MQクライアントのシステム・プロパティについての説明です。

- jeus.jms.client.default-redelivery-limit

説明	JMS_JEUS_RedeliveryLimitメッセージ・プロパティのデフォルト値を設定します。 メッセージの送信側で別途指定していない場合、送信したメッセージは最大でこの回数分のみ復旧し、メッセージ受信側に渡されます
デフォルト値	3

- jeus.jms.file.blocksize

説明	ファイル・メッセージを使用して、サーバーにファイルをブロック単位で送信する際のブロック・サイズを設定します
デフォルト値	4096 (単位 : byte)

- jeus.jms.client.connect.timeout

説明	ConnectionFactory.createConnection()メソッドがサーバーからの応答を待つ制限時間を設定します
デフォルト値	5000 (単位 : ms)

- jeus.jms.client.workdir

説明	ファイル・メッセージを使用してサーバーから受信したファイルが保存されるパスを設定します。 Java EEクライアントの場合、この設定は無視され、JEUS_HOME/logs/jms/BROKER_NAME/client/パスが使用されます
デフォルト値	USER_HOME/jeusmq_client_work/

- jeus.jms.client.maxThread

説明	JEUS MQクライアント・ライブラリーで使用するスレッド・プールのサイズを設定します。 クライアント・ライブラリーがメッセージの送受信など複数の作業を同時に実行するために必要です。サーバーに接続されているコネクションごとに1つずつのスレッド・プールが作成されます
デフォルト値	100

- jeus.jms.client.optimizedProduce

説明	メッセージ送信の性能を向上させるために、サーバーからの承認を受信するか否かを設定します。 trueと設定した場合、セッションの承認モードをJeusSession.NONE_ACKNOWLEDGEと設定しなくても、メッセージの送信時にサーバーからの承認を常に受けません。ただし、ファイル・メッセージの送信時はこの設定に関係なく承認を受けます
デフォルト値	false

- jeus.jms.client.use-single-server-entry

説明	コネクション間に物理的な接続を共有して使用するか否かを設定します。 1つのコネクション・ファクトリーあたり1つの物理的な接続を持ちます
デフォルト値	true

- jeus.jms.client.single-server-entry.shutdown-delay

説明	コネクション間で共有していた物理的な接続を使用しなくなった場合、再使用するために終了が遅延される時間を設定します
デフォルト値	600000 (単位 : ms)

- jeus.jms.client.use-pooled-connection-factory

説明	一度使用したクライアント・ファシリティーを再使用するためにプーリングを実行するか否かを設定します
デフォルト値	true

- jeus.jms.client.pooled-connection.check-period

説明	プーリングされたクライアント・ファシリティーのうち、使用されていないものを削除するためにチェックする周期を設定します
デフォルト値	60000 (単位 : ms)

- jeus.jms.client.pooled-connection.unused-timeout

説明	プーリングされたクライアント・ファシリティーのうち、この時間内に使用されなければプールから削除されます
デフォルト値	120000 (単位 : ms)

- jeus.jms.client.send-by-lpq-only

説明	メッセージを送信するときに、すべてのメッセージを信頼度を高めるためのLPQサービスを利用して送信します
デフォルト値	false

- jeus.jms.client.lpq-configuration-path

説明	LPQサービスを利用するときに使用する設定ファイルの詳細パスを設定します。 <ul style="list-style-type: none"> – 絶対パスの場合：そのパスのファイルを読み込みます – 相対パスの場合：DEPLOYED_HOME/myApp/WEB-INF/、DEPLOYED_HOME/myApp/META-INF/、DEPLOYED_HOME/myApp/の順に見つかるファイルを読み込みます
デフォルト値	120000 (単位 : ms)

1.10.2. JEUS MQサーバー

以下は、JEUS MQサーバーのシステム・プロパティについての説明です。

- jeus.jms.file.blocksize

説明	ファイル・メッセージを使用してクライアントにファイルをブロック単位で送信する際のブロック・サイズを設定します
デフォルト値	4096 (単位 : byte)

- jeus.jms.server.deleteFileMessage

説明	ファイル・メッセージの送信が完了後、該当ファイルをサーバーから削除するか否かを設定します
デフォルト値	true

- jeus.jms.server.discard-stored-on-start

説明	JEUS MQサーバーの起動時に設定されたストレージに存在するすべてのテーブルを削除するか否かを設定します
デフォルト値	false

- jeus.jms.server.rowFetchSize

説明	データベースのストレージからメッセージをクエリーする際に、一度に取得するレコードの数を設定します
デフォルト値	1000

- jeus.jms.server.create-facility-mbeans

説明	コネクション、セッション、メッセージ・プロデューサー、コンシューマのようなクライアント・ファシリティーのMBeanを作成するか否かを設定します
デフォルト値	false

Part II. コンソール・コマンドとツール

第2章 コマンドの説明規則

各コマンドを説明する規則について簡単に紹介します。各節の題名はコマンド名を示しています。本文では、説明、エイリアス、使用方法、パラメータ、使用例に分けて記述します。

- 説明：コマンドの機能について説明します。
- 関連スキーマ：該当コマンドが関連しているスキーマ項目を表記します。
- エイリアス：コマンドの別名を表記します。
- 使用方法：コマンドの文法について説明します。jeusadminでヘルプを実行した場合、usageラインに出力される文法と同一です。
 - コマンド：コマンドは**太字**で表記します。
 - パラメータ：コマンドのオプションを始めた各パラメータが持つ意味、どのようなケースでどう使用するのかを説明します。

パラメータは一般書体とイタリック体を使用します。

 - 一般書体：オプションを表現します。ユーザーはガイドに記述されているまま使用します。
 - イタリック体：ユーザーが入力するパラメータを意味します。該当値をユーザーが直接入力します。
 - []：省略可能な項目を示します。[]で囲まれていない項目は必須です。
 - |または/：「or」を示し、これを区分するパラメータはその中の1つのみを使用できます。
- パラメータ：コマンドに使用できるオプションや引数など、パラメータについて説明します。
- 使用例：jeusadminにおける実際の使用例と出力結果について簡単に説明します。

第3章 JEUSサーバーの実行

本節では、JEUSサーバーを実行するコマンドについて説明します。

3.1. 概要

JEUSサーバーを実行するコマンドは以下の2つに分けられます。

- DAS(Domain Administration Server)を実行する**startDomainAdminServer**コマンド
- MS(Managed Server)を実行する**startManagedServer**コマンド

サーバーの実行コマンドは、該当サーバーのランチャーを実行します。ランチャーはサーバーを実行させるための準備作業が完了後、サーバーJVMを起動します。DASは実際にサービスが行われるMSを管理するサーバーで、管理者のコマンドをMSに渡すエージェントの役割をします。

実行コマンドは以下のディレクトリーに位置します。

```
JEUS_HOME/bin/
```

3.2. 使用方法

本節では、DASとMSの起動方法について説明します。

jeusadminやWebAdminなどのツールを利用してMSを管理するためには、まずDASを起動します。

DASは各MSと通信し、MSは各ホスト・マシンで動作します。ドメインに属するすべてのサーバーは、1つの設定ファイルである次のファイルを共有します。設定ファイルで自身の動作に必要な設定を読み込み、動作します。

```
JEUS_HOME/domains/DOMAIN_NAME/config/domain.xml
```

DASの起動

以下は、DASを起動するためにstartDomainAdminServerコマンドを実行する手順です。

1. DASを起動するには、各OS別に以下のコマンドを実行します。

- UNIXあるいはLinux

```
startDomainAdminServer
```

- Windows

```
startDomainAdminServer.cmd
```

2. コマンド・プロンプトに**startDomainAdminServer**と入力し、以下の設定を追加します。

domain.xmlを修正すると、サーバーはデバッグモードで運用できます。この設定方法は「[3.4. デバッグ・モードのJEUSサーバーの実行](#)」を参照してください。

```
startDomainAdminServer [-h]
                        [-domain <domain-name> -server <das-name>]
                        [-u <username> -p <password>]
                        [-cachellogin]
                        [-f <filename>]
                        [-force]
                        [-standby]
                        [-rolling]
                        [-verbose]
```

以下は、コマンド・オプションについての説明です。

オプション	説明
[-h]	コマンドのヘルプを出力します
[-domain <domain-name>]	起動するDASが属するドメインの名前を設定します
[-server <das-name>]	起動するDASの名前を設定します
[-u <username>]	サーバーを起動する主体を意味します。
[-p <password>]	[-u]および[-p]は管理者IDとパスワードを意味し、サーバーを実行するための必須入力事項です
[-cachellogin]	[-u]、[-p]オプションで入力したログイン情報をキャッシュします。 キャッシュしたログイン情報が存在する場合、オプションが別途なくても自動的にユーザー情報が設定されます。 これは推奨していない機能です。詳細については『JEUS セキュリティガイド』を参照してください
[-f <filename>]	管理者のログイン情報を持つファイルのパスを指定します。 [-u]、[-p]オプションの代替です

オプション	説明
[-force]	登録されているアプリケーションがすべてRUNNING状態でなくても、サーバーの状態を強制的にRUNNINGにします
[-standby]	サーバーをSTANDBY状態で起動します
[-rolling]	ローリング・パッチのための内部オプションです。起動時にユーザーが設定してはなりません
[-verbose]	ランチャーを終了せずに、サーバーで発生するログをランチャー画面に表示するか否かを指定します

MSの起動時のオプションはDASの起動時のオプションと大部分同じです。MSではdasurlオプションが追加で存在し、これはMSの起動時にDASから設定ファイルを受け取るのに使用されます。

MSの起動

以下は、MSを起動するためにstartManagedServerコマンドを実行する手順です。

1. MSを起動するには以下のコマンドを実行します。

- UNIXあるいはLinux

```
startManagedServer
```

- Windows

```
startManagedServer.cmd
```

2. コマンド・プロンプトに**startManagedServer**と入力し、以下の設定を追加します。

```
startManagedServer [-h]
                    [-domain <domain-name> -server <das-name>]
                    [-u <username> -p <password>]
                    [-cachelogin]
                    [-f <filename>]
                    [-force]
                    [-standby]
                    [-rolling]
                    [-verbose]
                    [-dasurl <das-url>]
```

以下は、コマンド・オプションについての説明です。

オプション	説明
[-h]	コマンドのヘルプを出力します
[-domain <domain-name>]	起動するMSが属するドメインの名前を設定します
[-server <das-name>]	起動するMSの名前を設定します。必須入力事項です
[-u <username>]	サーバーを起動する主体を意味します。
[-p <password>]	[-u]および[-p]は管理者IDとパスワードを意味し、サーバーを実行するための必須入力事項です
[-cachellogin]	[-u]、[-p]オプションで入力したログイン情報をキャッシュします。 キャッシュしたログイン情報が存在する場合、オプションが別途なくても自動的にユーザー情報が設定されます。 これは推奨していない機能です。詳細については『JEUS セキュリティガイド』を参照してください
[-f <filename>]	管理者のログイン情報を持つファイルのパスを指定します。 [-u]、[-p]オプションの代替です
[-force]	登録されているアプリケーションがすべてRUNNING状態でなくても、サーバーの状態を強制的にRUNNINGにします
[-standby]	サーバーをSTANDBY状態で起動します
[-rolling]	ローリング・パッチのための内部オプションです。起動時にユーザーが設定してはなりません
[-verbose]	ランチャーを終了せずに、サーバーで発生するログをランチャー画面に表示するか否かを指定します
[-dasurl <das-url>]	自身が属するドメインを管理するDASのURLアドレスを指定します。DASから新しい設定ファイルを受信できるようにURLアドレスを設定することを推奨します

3.3. マスター・パスワードの入力

暗号化ツールを利用してsecurity.keyファイルをマスター・パスワードで暗号化した場合、サーバーの起動時にマスター・パスワードを入力する必要があります。security.keyファイルには対称キーの暗号化アルゴリズムのキーが保存されます。暗号化ツールに関する詳細は[「4.6. encryption」](#)を参照してください。

サーバーの起動時にsecurity.keyファイルが暗号化されている場合、以下のようなコンソール入力画面が現れ、マスター・パスワードの入力を待機します。マスター・パスワードが正しくない場合はサーバーの実行が不可能です。

```
The encryption key file is encrypted. Enter the master password.  
Password>
```

3.4. デバッグ・モードのJEUSサーバーの実行

デバッグ・モードでJEUSサーバーを実行する場合、JPDAがサポートするデバッガーを使用することでサーブレットとEJBアプリケーションをデバッグできます。JEUSサーバーをデバッグ・モードで起動するには、各サーバーのjvm-option設定に以下のようにデバッグ・オプションを追加します。

```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8888
```

「8888」のデバッグ・ポート(デバッグ・ポートはJPDAデバッガーがJVMに接続する際に使用されます)を使用してJEUSサーバーをデバッグ・モードで動作するようにします。

修正された設定でJEUSサーバーを起動し、JPDAをサポートするIDE(Integrated Development Environment)を使用してEJBとサーブレット・アプリケーションをデバッグします。

各サーバーのjvm-option設定にてシステム環境変数などの設定も可能です。JEUSで使用するシステム環境変数については[Part I, 「システム・プロパティ」](#)を参照してください。「[1.1. 概要](#)」ではJVMで使用する「-D」オプションについて説明しています。

参考

jvm-option設定は該当サーバーにのみ適用される点に注意してください。

第4章 コンソール・ツール

本章では、JEUSで使用するコンソール・ツールの種類と使用方法について説明します。

4.1. 概要

以下は、JEUSで使用するコンソール・ツールの一覧です。

- アドミン・ツール

コンソール・ツール	説明
jeusadmin	JEUSを直接管理するために使用するコンソール・ツールです

- その他ツール

コンソール・ツール	説明
appcompiler	アプリケーションのEJBインターフェース、Impl、Skeleton、Stubクラス、JSPをコンパイルし、サーブレット・クラス、WebサービスのEndpointクラスなどを作成します
ejbddinit	ejb-jar.xml情報と事前に作成されたプロパティ・ファイルを利用して、EJBモジュールのDDファイル(jeus-ejb-dd.xml)を作成できます
webddgen	WebモジュールのDDファイル(jeus-web-dd.xml)を作成またはアップデートできます
encryption	暗号化する文字列を暗号化あるいは復号化する機能を提供します
wsген	JAX-WS Webサービスのためのツールで、JavaクラスからWSDLファイルとJAX-RPCマッピング・ファイルを作成します。また、Webサービス・ポリシー設定ファイルを作成できます
wsimport	WSDLファイルからクライアント側のJava Stubソース・ファイルとサーバー側のWebサービス・インターフェースのJavaソース・ファイルを作成します。また、Webサービス・ポリシー設定ファイルを作成できます
java2wsdl	JavaクラスからWSDLファイルとJAX-RPCマッピング・ファイルを作成します
wsdl2java	WSDLファイルからクライアント側のJava Stubソース・ファイルとサーバー側のサービス・インターフェースのJavaソース・ファイルを作成します
wsdl2uddi	WebサービスのWSDLファイルからUDDIに配布できます
xjc	XMLスキーマ・ファイルをJavaプログラミング言語で作成されたJAXB Contentクラスに変換します
schemagen	Javaクラスに現れる各名前空間に対して1つのスキーマ・ファイルを作成します

コンソール・ツール	説明
tcpmon	送受信するTCPパケットを出力します。tcpmonはHTTPのSOAPメッセージを表示できます

4.2. jeusadmin

本節では、JEUSの管理ツールであるjeusadminのコマンドとその使用方法について説明します。

jeusadminは、JEUSを直接管理するために使用するコンソール・ツールです。このツールを使用することにより、JEUSサーバーの起動と停止といった基本的な操作を始め、アプリケーションのデプロイやアンデプロイ、ロガーレベルの変更、JMX MBeanの一覧の表示などを実行できます。

jeusadminで使用するコマンドについての基本文法は以下のとおりです。

```
COMMAND [COMMAND_OPTIONS]
        [GLOBAL_OPTIONS]
(>|>>) FILENAME
```

- COMMAND

コマンド名またはエイリアスを設定します。

- COMMAND_OPTIONS

該当コマンドのオプションを設定します。

- GLOBAL_OPTIONS

すべてのコマンドに対して適用できる共用オプションを設定します。

オプション	説明
-verbose	エラーが発生した場合、スタック・トレースを出力します
-repeat <count>	コマンドを繰り返し実行する回数を指定します
-interval <seconds>	繰り返し実行する場合の実行周期を秒単位で指定します。 -repeatオプションを指定せずに使用した場合はエラーが発生します

- (>|>>) FILENAME

コマンドの実行結果をファイルに保存する際に使用します。保存するファイル名はFILENAMEと指定します。

以下のように使用できます。

```
offline>help > result.txt
または
offline>help >> result.txt
```

4.2.1. 起動および終了コマンド

jeusadminを起動および終了するコマンドについて説明します。

コマンド	説明
jeusadmin	jeusadminを実行します
exit	管理作業を完了し、jeusadminを終了します

4.2.1.1. jeusadmin

jeusadminを実行します。このファイルはJEUS_HOME/binに位置します。

- 使用方法

```
jeusadmin [-u,--username <username>]
          [-p,--password <password>]
          [-f <filename>]
          [-cachellogin]
          [-d,--domain <domain>]
          [-host <server address>]
          [-port <server port>]
          [-verbose]
          [-help]
          [<target command>]
          [-version]
          [-fullversion]
          [-buildversion]
          [-licensedue]
          [-licenseinfo]
          [-debug]
          [-script (<script file>|"<script file> <args> ...")]
```

- パラメータ

パラメータ	説明
<code>[-u, --username <username>]</code>	接続時に使用するユーザー名を指定します。 オプションを指定した場合、実行と共にサーバーへの接続を試行します

パラメータ	説明
[-p, ---password <password>]	接続時に使用するパスワードを指定します。 オプションを指定した場合、実行と共にサーバーへの接続を試行します
[-f <filename>]	接続時に使用するユーザー名とパスワードを格納したファイルを指定します
[-cachelgin]	特定ドメイン(または基本ドメイン)に接続時、使用していたユーザー名とパスワードを記録します
[-d, --domain <domain>]	接続するドメインの名前を指定します
[-host <server address>]	接続時に使用するアドレスを指定します。 [-u]オプションと[-p]オプションを指定した状態でこのオプションを別途指定していない場合、ローカル・ホストへの接続を試行します
[-port <server port>]	接続時に使用するポート番号を指定します。 [-u]オプションと[-p]オプションを指定した状態でこのオプションを別途指定していない場合、9736番ポートに接続を試行します
[-verbose]	実行中にエラーが発生した場合、詳細状況まで出力します
[-help]	jeusadminに関するヘルプを出力します
[<target command>]	<target command>を一度実行します。スクリプトで使用する場合に有効です。 サーバーに接続が必要なコマンドの場合、[-u]、[-p]、[-host]、[-port]など必要なオプションを一緒に指定します
[-version]	JEUSのメジャー・バージョンを出力します
[-fullversion]	JEUSのメジャー・バージョンとマイナー・バージョンを出力します
[-buildversion]	JEUSのメジャー・バージョンとビルド・バージョンを出力します
[-licensedue]	ライセンスの有効期限を出力します。有効期限が0になるとJEUSは動作不可能となるので、TmaxSoftから新しいライセンスを取得する必要があります
[-licenseinfo]	現在インストールされているライセンス情報を出力します
[-debug]	jeusadminでJeus Loggerを使用して出力されるログ・メッセージを表示します
[-script (<script file> "<script file> <args> ...")]	スクリプト・モードでjeusadminを実行します。このとき、ファイルの拡張子は、.py(Python)、.rb(Ruby)である必要があります

- 例

以下は、JEUSのユーザー名を「administrator」、パスワードを「jeus」と仮定したjeusadminの例です。

– jeusadminの実行

以下は、jeusadminを実行する例です。

```
JEUS_HOME/bin$jeusadmin
JEUS8 Administration Tool
To view help, use the 'help' command.
offline>
```

– 実行と同時に即時サーバーに接続 #1

以下は、localhost:9736にあるサーバーに接続する例です。

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
in the domain domain1.
JEUS8 Administration Tool
To view help, use the 'help' command.
[DAS]domain1.adminServer>
```

– 実行と同時に即時サーバーに接続 #2

以下は、IPアドレスとポート番号を指定して接続する例です。

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus -host 192.168.0.1 -port 9746
Attempting to connect to 192.168.0.1:9746.
The connection has been established to Domain Administration Server adminServer
in the domain domain1.
JEUS8 Administration Tool
To view help, use the 'help' command.
[DAS]domain1.adminServer>
```

– ライセンス情報の表示

以下は、JEUSライセンスを表示する例です。

```
JEUS_HOME/bin$jeusadmin -licensedue
Unlimited

JEUS_HOME/bin$jeusadmin -licenseinfo
===== LICENSE INFORMATION =====
=== EDITION : Enterprise (Trial License)
=== ISSUE-DAY : 2016/1/16
=== LICENSE SEQNO : xxx-xxxx-xxx-xxxx
=====
```

- SSLを使用してjeusadminに接続

SSLを使用してjeusadminに接続する場合、jeusadminの実行スクリプト・ファイルを開いて、以下のシステム・プロパティを設定します。(システム・プロパティの値はサーバーのSSL設定によって異なる場合があります)

```
. . .
# execute jeusadmin
"${JAVA_HOME}/bin/java" -classpath "${BOOTSTRAP_CLASSPATH}" "${TOOL_OPTION}

-Djmx.remote.x.request.timeout=600000

-Djava.library.path="${JEUS_LIBPATH}"

-Djeus.home="${JEUS_HOME}"

-Djeus.baseport=${JEUS_BASEPORT}

-Djeus.tool.console.useJLine="false"

-Djava.endorsed.dirs="${JEUS_HOME}/lib/endorsed"

-Djava.naming.factory.initial=jeus.jndi.JEUSContextFactory

-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url

-Djava.util.logging.config.file="${JEUS_HOME}/bin/logging.properties"

-Djeus.net.client.use-ssl=true

-Djavax.net.ssl.trustStore=${JEUS_HOME}/domains/domain1/config/truststore

-Djavax.net.ssl.trustStorePassword=changeit

-Djavax.net.ssl.trustStoreType=JKS

-Dssl.TrustManagerFactory.algorithm=SunX509

${JAVA_ARGS}

jeus.tool.console.ConsoleBootstrapper ${BOOT_PARAMETER}
```

4.2.1.2. exit

管理作業を完了し、jeusadminを終了します。

- エイリアス

quit

- 使用方法

```
exit
```

- 例

```
[DAS]domain1.server1>exit
JEUS_HOME/bin$
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2. ローカル・コマンド

以下は、サーバー接続、ヘルプ、履歴機能などを提供する基本コマンドです。

コマンド	説明
help	コマンドに関するヘルプを出力します
history	以前に使用したコマンドの一覧を出力します
p	入力したコマンドのうち1つを選んで実行します。 オプションを指定せずに実行した場合、最後に実行したコマンドを再実行します
connect	DASやMSへの接続を試行します
disconnect	接続を切断し、オフライン状態に戻ります
verbose	コマンドの実行時に詳細出力モードを有効あるいは無効にします
remove-login-cache	cachelogin機能を通じて、保存されているユーザー・ログイン情報をキャッシュ・ファイルから削除します
set-server-option	serverオプションのデフォルト値を設定します
unset-server-option	set-server-optionで設定したデフォルト値を削除します
start-record	入力するコマンドをスクリプト形式で保存するレコーディング機能を開始します
stop-record	start-recordで設定したレコーディング機能を停止します
set-servers-option	serversオプションのデフォルト値を設定します
unset-servers-option	set-servers-optionで設定したデフォルト値を削除します

4.2.2.1. help

コマンドに関するヘルプを出力します。オプションを別途指定せずに実行した場合、ヘルプと共に、現在使用できるコマンドの一覧、コマンドの基本文法に関する説明を出力します。

- 使用方法

```
help [-g,--group <group name> | <command name> | -l,--list]
```

- パラメータ

パラメータ	説明
[-g,--group <group name>]	指定されたグループ(group name)に属するコマンドの一覧と、簡単な説明を出力します
[<command name>]	特定コマンド(command name)のヘルプを出力します。 コマンド名、エイリアス、使用方法、オプションに関する情報を表示します。 オプションを省略して使用することもできます
[-l,--list]	すべてのコマンドに関する簡単な説明を出力します

- 例

– helpコマンドのみを使用した場合

```
offline>help
LIST OF AVAILABLE COMMANDS

[ Local]_____
  connect                disconnect
  exit                   help
  history                p
  remove-login-cache     set-server-option
  set-servers-option     start-record
  stop-record            unset-server-option
  unset-servers-option   verbose

[ Server Management]_____
  local-shutdown         local-start-server

[ Domain Configuration]_____
  apply-configuration-plan  create-domain
  delete-domain            list-domains
  pack-domain              unpack-domain

[ Application]_____
  install-application      install-deployment-plan
  redeploy-application

[ Web]_____
  precompile-jsp
```

```
[ NodeManager]_____
connect-nodemanager          disconnect-nodemanager
nm-start-server              nm-state-server
nm-stop-server               stop-nodemanager

COMMAND GRAMMAR
  Command Usage
    COMMAND [COMMAND_OPTIONS] [GLOBAL_OPTIONS] [(>|>>) FILENAME]
  Details
    COMMAND: target command to execute.
    [COMMAND_OPTIONS]: Options for target command.
    [GLOBAL_OPTIONS]: Common options for all commands.
      -verbose
        Prints the stack trace for errors.
      -repeat <count>
        Repeats the target command X times.
      -interval <seconds>
        Specifies the interval at which to repeat execution, in
        seconds.
    [(>|>>) FILENAME]
      Prints the result of the target command to the specified file.
  Examples
    help -l -repeat 3 -interval 10 > result.txt

To show detailed information for a command, use 'help [COMMAND_NAME]'.
ex) help connect
offline>
```

– helpコマンドを[-l]オプションと一緒に使用した場合

```
offline>help -l
[ Local]_____
connect          Attempting to authenticate with the target
                  server.
disconnect       Disconnects from the currently connected server.
.....

[ Server Management]_____
local-shutdown   Shuts down the currently connected server.
local-start-server On off-line mode, start managed server via SSH.
                  You should specify -node, -domain, -server, -u,
                  -p, -dasurl.
                  On on-line mode, start standby server.
.....
```

```
[ NodeManager]_____
connect-nodemanager      Connects to the node manager.
disconnect-nodemanager   Closes the connection to the node manager.
nm-start-server          Starts a server using the node manager.
nm-state-server          Shows the state of a server using node manager.
nm-stop-server           Stops a server using the node manager.
stop-nodemanager         Stops the node manager.

To show detailed information for a command, use 'help [COMMAND_NAME]'.
ex) help connect
offline>
```

- helpコマンドを[-g]オプションと一緒に使用した場合

以下は、「Server Management」グループのコマンド一覧を出力する例です。

```
offline>help -g "Server Management"
[ Server Management]_____
local-shutdown           Shuts down the currently connected server.
local-start-server       On off-line mode, start managed server via SSH.
                        You should specify -node, -domain, -server, -u,
                        -p, -dasurl.
                        On on-line mode, start standby server.

To show detailed information for a command, use 'help [COMMAND_NAME]'.
ex) help connect
offline>
```

- [<command name>]オプションを使用して特定コマンドに関するヘルプを出力する場合

以下は、add-serverコマンドの説明を出力する例です。

```
[DAS]domain1.adminServer>help add-server
NAMES
    add-server
        Adds a new server to the domain configuration. If a server name is
        not provided, the current server list in this domain will be shown.
ALIAS
    addserver
USAGE
    add-server <server-name>
        [-target,--duptargetserver <target-server>]
        [-port,--baseport <base-port>]
        [-addr,--baseaddr <base-addr>]
        [-node,--nodeName <node-name>]
        [-jvm,--jvmOptions <jvm-options>]
```

```

[-logdir,--logHomeDirectory <server-log-home-directory>]
[-a,--actionOnResourceLeak <action-on-resource-leak>]
[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
[-m,--mejb <enable-MEJB>]
[-c,--classFtp <enable-class-ftp-service>]
[-f,--forceLock]
[-detail]

```

OPTIONS

<server-name>

the name of server you want to modify configuration of

```
[-target,--duptargetserver <target-server>]
```

duplicate the configuration of the target server

```
[-port,--baseport <base-port>]
```

base service listen port of the server

```
[-addr,--baseaddr <base-addr>]
```

base service listen address of the server

```
[-node,--nodeName <node-name>]
```

name of the node this server is located

```
[-jvm,--jvmOptions <jvm-options>]
```

jvm configurations applied to this server jvm

```
[-logdir,--logHomeDirectory <server-log-home-directory>]
```

the log directory which has all the log files created by this server

```
[-a,--actionOnResourceLeak <action-on-resource-leak>]
```

strategy when resource leak is detected. it must be one of
AutoClose, NoAction, Warning

```
[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
```

[Dynamic] whether stdout is printed in row format or in JEUS Logger
format.

```
[-m,--mejb <enable-MEJB>]
```

[Dynamic] whether using MEJB or not .true, false

```
[-c,--classFtp <enable-class-ftp-service>]
```

[Dynamic] whether using class ftp service or not .true, false

```
[-f,--forceLock]
```

Acquires the configuration lock forcibly.

```
[-detail]
```

```
Show detail results

[DAS]domain1.adminServer>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.2. history

以前に使用したコマンドの一覧を出力します。

コマンド一覧は\${user.home}/.jeusadmin/commandHistoryに保存されます。記録したコマンドを再実行する場合はpコマンドを使用します。最大100個まで記録できます。

- 使用方法

```
history [-clear | <size>]
```

- パラメータ

パラメータ	説明
[-clear]	以前に実行したコマンドの一覧を削除し、初期化します
[<size>]	出力するコマンド数を設定します。最終実行順にsizeの分だけ出力します

- 例

```
offline>history
1 : help history
2 : connect
3 : disconnect

To execute a command in the history, please use '!' command (e.g. ! 7)
offline>history 3
3 : disconnect

To execute a command in the history, please use '!' command (e.g. ! 7)
offline>history -clear
The command history has been cleared.
offline>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.3. p

以前に実行したコマンドを再実行する際に使用します。以前に実行したコマンドの一覧は[history](#)コマンドを使用して確認できます。

- エイリアス

!

- 使用方法

```
p [<history number>]
```

- パラメータ

パラメータ	説明
[<history number>]	コマンド記録で実行するコマンドの番号を指定します

- 例

```
offline>history
1 : connect -u jeus -p jeus
2 : disconnect

To execute a command in the history, please use '!' command (e.g. ! 7)
offline>p 1
connect -u jeus -p jeus
Enter the server address:
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
in the domain domain1.
[DAS]domain1.adminServer>! 2
disconnect
The connection has been closed.
offline>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.4. connect

サーバーへの接続を試行します。オプションを省略して実行した場合、欠落した情報を再度確認してきます。

- エイリアス

login

- 使用方法

```
connect [-u,--username <username>]
        [-p,--password <password>]
        [-f <filename>]
        [-cachellogin]
        [-h,--host <server address>]
        [-port <server port>]
```

- パラメータ

パラメータ	説明
[-u, --username <username>]	接続時に使用するユーザー名を指定します
[-p, --password <password>]	接続時に使用するパスワードを指定します
[-f <filename>]	接続時に使用したユーザー名とパスワードを保存したファイルを指定します
[-cachellogin]	接続時に使用したユーザー名とパスワードをファイルに記録します
[-h, --host <server address>]	接続するサーバーのアドレスを指定します。このオプションを指定せずに接続を試行するとアドレスを再度確認してきますが、この際に何も入力しなければローカルホストに接続を試行します
[-port <server port>]	接続するサーバーのポートを指定します。別途指定していない場合はデフォルト値を使用します(デフォルト値: 9736)

- 例

- オプションを指定せずに実行した場合は、必要な情報を確認してきます。

```
offline>connect
Enter the server address:
User name: administrator
Password:
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
in the domain domain1.
[DAS]domain1.adminServer>
```

- [-u]オプションのみを指定して実行した場合は、欠落した情報を再度確認してきます。

```
offline>connect -u administrator
Enter the server address:
Password:
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
```

```
in the domain domain1.  
[DAS]domain1.adminServer>
```

- [-port]オプションで指定したポートに接続する場合の例は以下のとおりです。

```
offline>connect -u administrator -p jeus -h 127.0.0.1 -port 9736  
Attempting to connect to 127.0.0.1:9736.  
The connection has been established to Domain Administration Server adminServer  
in the domain domain1.  
[DAS]domain1.adminServer>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.5. disconnect

接続を切断し、オフライン状態に戻るコマンドです。

- エイリアス

logout

- 使用方法

```
disconnect
```

- 例

```
offline>connect  
Enter the server address:  
User name: administrator  
Password:  
Attempting to connect to 127.0.0.1:9736.  
The connection has been established to Domain Administration Server adminServer  
in the domain domain1.  
[DAS]domain1.adminServer>disconnect  
The connection has been closed.  
offline>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.6. verbose

コマンドの実行過程の詳細を出力するか否かを設定します。オプションを指定していない場合は、詳細出力が現在有効であるのか否かを通知します。詳細出力が有効の場合、コマンドの実行中に例外が発生するとスタック・トレースを出力します。

- 使用方法

```
verbose [-off | -on]
```

- パラメータ

パラメータ	説明
[-on]	詳細出力を有効にします
[-off]	詳細出力を無効にします

- 例

- 詳細出力が現在有効であるのか否かを確認する例

```
offline>verbose
Verbose output disabled.
offline>
```

- 実際の使用例

```
offline>verbose
Verbose output disabled.
offline>help notexistcommand
Command notexistcommand not found
offline>verbose -on
Verbose output is enabled.
offline>help notexistcommand
Command notexistcommand not found
jeus.tool.console.executor.CommandException: Command notexistcommand not found

.....

offline>verbose -off
Verbose output disabled.
offline>help notexistcommand
Command notexistcommand not found
offline>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.7. remove-login-cache

cachelogin機能を通じて、保存されているユーザーのログイン情報をキャッシュ・ファイルから削除します。

jeusadminによって保存されたログイン情報は<host-name>と<port-number>によって区分されます。また、スクリプトを使用したサーバーを実行する際に保存されたログイン情報は<domain-name>と<server-name>によって区分されます。

- 使用方法

```
remove-login-cache [-host <host-name>]
                  [-port <port-number>]
                  [-domain <domain-name>]
                  [-server <server-name>]
                  [-f <filename>]
```

- パラメータ

パラメータ	説明
[-host <host-name>]	jeusadminによるログイン情報を保存する際に使用していたホスト名を指定します
[-port <port-number>]	jeusadminによるログイン情報を保存する際に使用していたポート番号を指定します
[-domain <domain-name>]	スクリプトによるログイン情報を保存する際に使用していたドメイン名を指定します
[-server <server-name>]	スクリプトによるログイン情報を保存する際に使用していたサーバー名を指定します
[-f <filename>]	ユーザー名とパスワードが保存されているパスワード・ファイル名を指定します

- 例

```
offline>remove-login-cache -host localhost -port 9736
The cached login information [localhost:9736] has been successfully removed from
/home/OSUser/.jeusadmin/.jeuspasswd.

offline>remove-login-cache -domain domain1 -server adminServer
The cached login information [domain1:adminServer] has been successfully removed
from /home/OSUser/.jeusadmin/.jeuspasswd.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.8. set-server-option

-serverオプションのデフォルト値を設定できる機能を提供します。オプションを指定せずに使用した場合、現在設定されているデフォルト値を出力します。

- エイリアス

setserver

- 使用方法

```
set-server-option <server-name>
```

- パラメータ

パラメータ	説明
<server-name>	serverオプションのデフォルト値として使用するサーバー名を設定します

- 例

adminServerをデフォルト値に設定し、**jndi-info**コマンドを実行します。set-server-optionコマンドを使用して<server-name>を指定する前にjndi-infoコマンドを実行すると、-serverオプションが指定されていないというエラーが発生します。しかし、サーバーを指定後は-serverオプションを別途指定しなくても、adminServerに対してjndi-infoコマンドが動作します。

```
[DAS]domain1.adminServer>jndi-info
The following options are missing: server
[DAS]domain1.adminServer>set-server-option
The default server name is not set.
[DAS]domain1.adminServer>set-server-option adminServer
The default server name has been set: adminServer.
[DAS]domain1.adminServer>set-server-option
The current default server name: adminServer.
[DAS]domain1.adminServer>jndi-info
The JNDI list on the adminServer
List of the context /
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| ConnectionFactory | jeus.jms.client.facility.factory.JeusC | false |
| onnectionFactory | | |
```

```
|  
.....  
[DAS]domain1.adminServer>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.9. unset-server-option

set-server-optionコマンドで設定したデフォルト値を削除します。

- エイリアス

unset-server-option

- 使用方法

```
unset-server-option
```

- 例

unset-server-optionを使用して、set-server-optionで設定したデフォルト値を削除しました。解除後に **jndi-info** コマンドを実行すると、serverオプションが必要であるというエラーが発生したことを確認できます。

```
[DAS]domain1.adminServer>set-server-option  
The current default server name: adminServer.  
[DAS]domain1.adminServer>unset-server-option  
The default server name has been unset.  
[DAS]domain1.adminServer>set-server-option  
The default server name is not set.  
[DAS]domain1.adminServer>jndi-info  
The following options are missing: server
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.10. start-record

入力したコマンドをスクリプト・ファイルに記録するレコーディング機能を開始します。

指定したレコーディング・ファイルの拡張子によって、該当するスクリプトにコマンドが保存されます。レコーディングが開始すると、プロンプトに「-Rec*」が追加されます。

- エイリアス

startrecord

- 使用方法

```
start-record [<recfile>]
```

- パラメータ

パラメータ	説明
[<recfile>]	コマンドを記録するファイルを指定します。スクリプト・ファイルに保存されるため、現在サポートしているスクリプトの拡張子である .py、.rbのみ使用できます

- 例

- オプションを指定せずに実行する場合は、record_script_[timestamp].pyファイルに保存します。

```
offline>start-record
Starting recording to file : record_script_20150101123456.py
offline-Rec*>
```

- [<recfile>]を指定した場合は、指定したファイルに保存します。

```
offline>start-record record.py
Starting recording to file : record.py
offline-Rec*>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.11. stop-record

start-recordで設定したレコーディング機能を停止します。

- エイリアス

stoprecord

- 使用方法

```
stop-record
```

- 例


```
offline-Rec*>stop-record  
Stopping recording to file : recrod.py  
offline>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.12. set-servers-option

-serversオプションのデフォルト値を設定します。オプションを指定せずに使用した場合、現在設定されているデフォルト値を出力します。

- エイリアス

setservers

- 使用方法

```
set-servers-option [<server-list>]
```

- パラメータ

パラメータ	説明
[<server-list>]	serversオプションのデフォルト値として使用するサーバー名を指定します。各サーバーの名前はコンマ(,)で区切ります

- 例

set-servers-optionコマンドを使用してデプロイするサーバーの一覧を指定した後、deployコマンドを実行して指定したサーバー・アプリケーションをデプロイする例です。

```
[DAS]domain1.adminServer>set-servers-option  
The default server list is not set.  
[DAS]domain1.adminServer>set-servers-option adminServer,server1  
The default server list has been set: adminServer,server1.  
[DAS]domain1.adminServer>set-servers-option  
The current default server list: adminServer,server1.  
[DAS]domain1.adminServer>deploy sample  
deploy the application for the application [sample] succeeded.  
[DAS]domain1.adminServer>appinfo  
Application information for the domain [domain1].  
=====+-----+-----+-----+-----+-----+-----+  
| Applicati|Application| State | Target | Target | Application Path  
|
```

on ID	Type		Servers	Clusters
sample	WAR	RUNNING	server1,adminServer	\${INSTALL_HOME}/sample/sample.war

```
[DAS]domain1.adminServer>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.2.13. unset-servers-option

set-servers-optionコマンドで指定した -serversオプションのデフォルト値の設定を削除します。

- エイリアス

unsetservers

- 使用方法

```
unset-servers-option
```

- 例

```
[DAS]domain1.adminServer>set-servers-option
The current default server list: adminServer,server1.
[DAS]domain1.adminServer>unset-servers-option
The default server list has been unset.
[DAS]domain1.adminServer>set-servers-option
The default server list is not set.
[DAS]domain1.adminServer>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.3. サーバー管理関連コマンド

Server Managementに関連するコマンドの一覧は以下のとおりです。

コマンド	説明
add-custom-resource-to-servers	サーバーにカスタム・リソースを動的に追加します
add-data-sources-to-server	サーバーにデータベースを動的に登録します
add-external-resource-to-servers	サーバーに外部リソースを動的に追加します
add-jvm-option	サーバーにJVMの設定を追加します
add-listener	サーバーにリスナーを追加します
add-log-handler	サーバーのログにハンドラを動的に追加します
add-logger	サーバーにログを動的に追加します
disable-engine-init-on-boot	サーバーの内部エンジン(Servlet、EJB、JMS)の初期化時点をアプリケーションのデプロイ時点に設定します
disable-engines	サーバーの内部エンジン(Servlet、EJB、JMS)を選択的に使用しないように設定します
disable-json-command	JsonCommandを非活性化します
disable-to-resynchronize-applications	MSがINDEPENDENT状態からDEPENDENT状態に変更されたときに、アプリケーションを同期化しません
disable-webadmin	WebAdminサービスを非活性化します
dump	サーバーまたはクラスターに属するサーバーのスレッド・ダンプを取得します
enable-engine-init-on-boot	サーバーの内部エンジン(Servlet、EJB、JMS)の初期化時点をサーバーの開始時点に設定します
enable-engines	サーバーの内部エンジン(Servlet、EJB、JMS)を選択的に使用するよう設定します
enable-json-command	JsonCommandを活性化します
enable-to-resynchronize-applications	MSがINDEPENDENT状態からDEPENDENT状態に変更されたときに、アプリケーションを同期化します
enable-webadmin	WebAdminサービスを活性化します
jndi-info	サーバーのJNDIコンテキストにバインドされているオブジェクトのバインディング(名前)一覧を確認します
list-jvm-options	サーバーのJVM設定を表示します
list-log-handlers	サーバーのログに登録されているハンドラの情報を出力します
list-loggers	サーバーに登録されているログの情報を出力します
list-server-listeners	サーバーにあるリスナーの設定を出力します
local-shutdown	現在接続されているサーバーを終了します
local-start-server	オンライン・モードでは、現在接続されているサーバーがSTANDBY状態の場合にサーバーを開始します

コマンド	説明
log-level	サーバーのログと該当ログに登録されているハンドラのレベルを確認します
mbean-info	サーバーに登録されているMBeanのリスナーを出力します
memory-info	サーバーまたはクラスターに属するサーバーのメモリー情報を確認します
modify-jvm-option	サーバーのJVM設定を修正します
modify-listener	サーバーにあるリスナーの設定を修正します
modify-log-handler	サーバーのログに登録されているハンドラを修正します
modify-logger	サーバーに登録されているログを動的に修正します
remove-custom-resource-from-servers	サーバーに登録されているカスタム・リソースを動的に削除します
remove-data-sources-from-server	サーバーに登録されているデータソースを動的に削除します
remove-external-resource-from-servers	サーバーに登録されている外部リソースを動的に削除します
remove-jvm-option	サーバーのJVM設定を削除します
remove-listener	サーバーにあるリスナーを削除します
remove-log-handler	サーバーのログに登録されているハンドラを動的に削除します
remove-logger	サーバーに登録されているログを動的に削除します
resume-server	一時停止しているサーバーを再度サービスされるようにします
rollback-config	バックアップされている設定ファイルにロールバックします
run-garbage-collection	サーバーまたはクラスターに属するサーバーにfull gcを実行します
server-info	サーバーの現在の状態を確認します
show-config	現在の設定ファイルとバックアップされている設定ファイルの一覧を表示します
start-cluster	クラスターに属するMSを起動します
start-domain	現在ドメインに属しているすべてのMSを起動します
start-server	MSを起動します
stop-cluster	クラスターに属するMSを停止します
stop-domain	現在ドメインに属しているMSを停止します
stop-server	MSを終了します
suspend-server	サービス中のサーバーを一時停止します
system-info	サーバーのシステム情報を確認します
server-log	特定のサーバーのログをコマンドで確認します

4.2.3.1. add-custom-resource-to-servers

サーバーにカスタム・リソースを動的に追加します。

- エイリアス

add-cr-to-servers

- 使用方法

```
add-custom-resource-to-servers <export-name>
                                -servers <server-list>
                                [-f,--forceLock]
                                [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	サーバーに追加するカスタム・リソースのJNDIバインド名です
-servers <server-list>	カスタム・リソースを与えられたサーバー一覧に登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-custom-resource-to-servers custom/dog -servers
server1
Successfully performed the ADD operation for A custom resource.
Check the results using "list-custom-resources"
```

4.2.3.2. add-data-sources-to-server

サーバーにデータベースを動的に登録します。

- エイリアス

adddstosvr

- 使用方法

```
add-data-sources-to-server -server <server-name>
                           [-ids <data-source-id-list> | -all]
                           [-f,--forceLock]
                           [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	データベースを登録するサーバーの名前です
[-ids <data-source-id-list>]	サーバーに登録するデータソースのID一覧です。1つ以上の場合はコンマ(,)で区分します
[-all]	ドメインに定義されているすべてのデータソースをサーバーに登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-data-sources-to-server -server server1 -all
Successfully performed the ADD operation for data sources to the server [server1
].
Check the results using "add-data-sources-to-server -server server1"
```

4.2.3.3. add-external-resource-to-servers

サーバーに外部リソースを動的に追加します。

- エイリアス

add-er-to-servers

- 使用方法

```
add-external-resource-to-servers <export-name>
    -servers <server-list>
    [-f,--forceLock]
    [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	サーバーに追加する外部リソースの名前です
-servers <server-list>	外部リソースを与えられたサーバー一覧に登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-external-resource-to-servers test/ext -servers
server1
Successfully performed the ADD operation for A external resource.
Check the results using "list-external-resources"
```

4.2.3.4. add-jvm-option

サーバーにJVM設定を追加します。

- エイリアス

add-jvmopt, addjvmopt, add-jvm-config, add-jvmcfg, addjvmcfg

- 使用方法

```
add-jvm-option -server <server-name>
                [-opt,--jvmOption <jvm-option>]
                [-f,--forceLock]
                [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-opt,--jvmOption <jvm-option>]	追加するJVM設定です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	設定変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-jvm-option -server server1 -opt "-Xmx256m
-XX:MaxPermSize=128m"
Successfully performed the ADD operation for JVM configuration for the server
(sever1)., but all changes were non-dynamic. They will be applied after
restarting.
Check the results using "list-jvm-options or add-jvm-option"
```

4.2.3.5. add-listener

サーバーにリスナーを追加します。

- エイリアス

addlistener, createlister

- 使用方法

```
add-listener -server <server-name>
             -name <listener-name>
             [-addr <address>]
             -port <port>
             [-selectors <selectors>]
             [-dual]
             [-backlog <backlog>]
             [-timeout <read-timeout>]
             [-keepaliveTimeout <keepalive-timeout>]
             [-rt,--reservedthreads <reserved-threads>]
             [-f,--forceLock]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-name <listener-name>	追加するリスナーの名前です
[-addr <address>]	追加するリスナーのアドレスです
-port <port>	追加するリスナーのポート番号です。1~65535の値を設定する必要があり、他のリスナーのポート番号と重複しないようにします
[-selectors <selectors>]	使用するセクタの数です
[-dual]	読み取り/書き込み作業にそれぞれ別のセクタを使用します
[-backlog <backlog>]	追加するリスナーのバックログです
[-timeout <read-timeout>]	[動的変更] 追加するリスナーの読み取りタイムアウトです
[-keepaliveTimeout <keepalive-timeout>]	使用しないソケットを閉じるまでの時間です
[-rt,--reservedthreads <reserved-threads>]	追加するリスナーが使用するReserved Threadの数です
[-f,--forceLock]	追加するためのロックを強制的に取得します

- 例

```
[DAS]domain1.adminServer>add-listener -server server1 -name Test -port 10000
Executed Successfully
Check the result using 'list-server-listeners -server server1 -name Test.'
[DAS]domain1.adminServer>list-server-listeners -server server1 -name Test
=====
Test
+-----+-----+
```


Attribute name	Value
address	not-set
port	10000
selectors	1
dual-selector	false
backlog	128
read-timeout	30000
keepalive-timeout	not-set
reserved-threads	0

4.2.3.6. add-log-handler

サーバーのロガーにハンドラを動的に追加します。

- エイリアス

addloghandler, add-handler, addhandler

- 使用方法

```
add-log-handler -server <server-name>
                -logger <logger-name>
                [<handler-name>]
                [-level <handler-level>]
                [-filter <filter-class>]
                [-encoding <encoding>]
                [-filename <file-name>]
                [-enable <enable-rotation>]
                [-count <rotation-count>]
                [-dir <rotation-dir>]
                [-buffer <buffer-size>]
                [-append <append>]
                [-hour <hour> | -day <day> | -size <size>]
                [-f,--forceLock]
                [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-logger <logger-name>	ハンドラを追加するロガーの名前です
[<handler-name>]	追加するハンドラの名前です

パラメータ	説明
[-level <handler-level>]	[動的变化] ハンドラのログ・レベルです(デフォルト値: FINEST)
[-filter <filter-class>]	ログ・メッセージのフィルタ・クラスの名前です
[-encoding <encoding>]	ログ・メッセージのエンコード値です
[-filename <file-name>]	ログを記録するファイル名です (デフォルト値: loggerName.log. jeus loggerの場合は JeusServer.log)
[-enable <enable-rotation>]	ファイルのローテーションを可能にするか否かを指定します(デフォルト値: true)
[-count <rotation-count>]	バックアップするログファイルの最大数です
[-dir <rotation-dir>]	バックアップするログファイルに位置するディレクトリー・パスです
[-buffer <buffer-size>]	ログ・メッセージをファイルに記録する際のバッファ・サイズです
[-append <append>]	再起動時に以前のファイルに引き続きログを残すか否かを設定します (true false、デフォルト値: true)
[-hour <hour>]	入力された時間ごとに新しいログ・ファイルを作成します
[-day <day>]	入力された日ごとに新しいログ・ファイルを作成します
[-size <size>]	入力されたサイズに従って新しいログ・ファイルを作成します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的变化結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-log-handler -server server1 -logger jeus.jndi
fileHandler -day 10 -detail
Successfully performed the ADD operation for The handler for the logger(jeus.jndi)
on the server(server1)., but all changes were non-dynamic. They will be applied
after restarting.
Check the results using "list-log-handlers or add-log-handler"
```

4.2.3.7. add-logger

サーバーにロガーを動的に追加します。

- エイリアス

addlogger

- 使用方法

```
add-logger -server <server-name>
           [<logger-name>]
           [-level <log-level>]
           [-useParentHandlers <use-parent-handlers>]
           [-filter <filter-class>]
           [-formatter <formatter-class>]
           [-f,--forceLock]
           [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[<logger-name>]	サーバーに追加するロガーの名前です
[-level <log-level>]	[動的変更] ロガーのログ・レベルです
[-useParentHandlers <use-parent-handlers>]	[動的変更] 上位ロガーのハンドラの使用可否です(true false)
[-filter <filter-class>]	出力されるログ・メッセージのフィルタ・クラスの名前です
[-formatter <formatter-class>]	ロガーに適用するFormatterクラスの名前です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-logger -server server1 jeus.security -level FINE
Successfully performed the ADD operation for The logger for the server(server1).
, but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-loggers or add-logger"
```

4.2.3.8. disable-engine-init-on-boot

サーバーの内部エンジン(Servlet、EJB、JMS)の初期化時点をアプリケーションのデプロイ時に設定します。設定を適用するには、サーバーを再起動する必要があります。

- 関連スキーマ

jeus-domain.xsd

- エイリアス

lazyengine

- 使用方法

```
disable-engine-init-on-boot <server-list>
                             [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<server-list>	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>disable-engine-init-on-boot adminServer,server1
EngineInitOnBoot was successfully enabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
|                                     +-----+                                     |
+-----+
| Successfully changed only the XML.                                         |
|                                     +-----+                                     |
| Restart the server to apply the changes.                                   |
|                                     +-----+                                     |
+-----+
=====
```

4.2.3.9. disable-engines

サーバーの内部エンジン(Servlet、EJB、JMS)を使用しないように設定します。設定を適用するには、サーバーを再起動する必要があります。

- 関連スキーマ

jeus-domain.xsd

- エイリアス

engineoff

- 使用方法

```
disable-engines <server-list>
                [-all]
```

```
[-web]
[-ejb]
[-jms]
[-f,--forceLock]
```

- パラメータ

パラメータ	説明
<server-list>	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-all]	Web、EJB、JMSエンジンをすべて設定します
[-web]	Webエンジンを設定します
[-ejb]	EJBエンジンを設定します
[-jms]	JMSエンジンを設定します
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>disable-engines adminServer -all
All engine(s) was(were) successfully disabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully changed only the XML.                                           |
|                                                                               |
| Restart the server to apply the changes.                                   |
+-----+
=====
...
[DAS]domain1.adminServer>disable-engines adminServer -web -ejb
Web EJB engine(s) was(were) successfully disabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully changed only the XML.                                           |
|                                                                               |
| Restart the server to apply the changes.                                   |
+-----+
```

```
|
+-----+
=====
...
```

4.2.3.10. disable-json-command

JsonCommandを非活性化します。

- エイリアス

suspend-json-command, jsonoff

- 使用方法

```
disable-json-command [-f,--forceLock]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>disable-json-command
JsonCommand was successfully disabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
|                                     |
+-----+
| Successfully applied the configuration changes. |
|                                     |
+-----+
=====
```

4.2.3.11. disable-to-resynchronize-applications

MSがINDEPENDENT状態からDEPENDENT状態になったとき、アプリケーションを同期化しません

- エイリアス

disalbe-resync-apps, disable-resyncapps

- 使用方法

```
disable-to-resynchronize-applications [-f,--forceLock]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>disable-to-resynchronize-applications
ResyncAppsCommand was successfully disabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully applied the configuration changes.                             |
+-----+
=====
```

4.2.3.12. disable-webadmin

WebAdminサービスを非活性化します。

- エイリアス

suspend-webadmin, webadminoff

- 使用方法

```
disable-webadmin [-f,--forceLock]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>disable-webadmin
WebAdmin was successfully disabled.
The configuration was changed.
```

```

=====
+-----+
|                      Result                      |
+-----+
| Succeeded to apply configuration changes.         |
+-----+
=====

```

4.2.3.13. dump

サーバーまたはクラスターに属するサーバーのスレッド・ダンプを取得します。該当サーバーのサーバー・ログに記録されます。

- 使用方法

```
dump [-clusters <cluster-list> | -servers <server-list>]
```

- パラメータ

パラメータ	説明
[-clusters <cluster-list>]	クラスターの名前です。クラスターが1つ以上の場合はコンマ(,)で区分します
[-servers <server-list>]	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します

- 例

```

[DAS]domain1.adminServer>dump -servers server1
Dumping to the servers.
The result of executing a dump of server processes.
=====
+-----+-----+
| Server |          Result          |
+-----+-----+
| server1| Successfully dumped.     |
+-----+-----+
=====

```

4.2.3.14. enable-engine-init-on-boot

サーバーの内部エンジン(Servlet、EJB、JMS)の初期化時点をサーバーの開始時点に設定します。設定を適用するには、サーバーを再開始する必要があります。

- 関連スキーマ

jeus-domain.xsd

- エイリアス

eagerengine

- 使用方法

```
enable-engine-init-on-boot <server-list>
                             [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<server-list>	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>enable-engine-init-on-boot adminServer
EngineInitOnBoot was successfully enabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully applied part of the changes.                                     |
| Restart the server to apply the remaining changes.                         |
+-----+
=====
```

4.2.3.15. enable-engines

サーバーの内部エンジン(Servlet、EJB、JMS)を使用しないように設定します。設定を適用するには、サーバーを再開始する必要があります。

- 関連スキーマ

jeus-domain.xsd

- エイリアス

engineon

- 使用方法

```
enable-engines <server-list>
                [-all]
                [-web]
                [-ejb]
                [-jms]
                [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<server-list>	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-all]	Web、EJB、JMSエンジンをすべて設定します
[-web]	Webエンジンを設定します
[-ejb]	EJBエンジンを設定します
[-jms]	JMSエンジンを設定します
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>diabile-engines adminServer -all
All engine(s) was(were) successfully enabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
|                                     +-----+                                     |
+-----+
| Successfully applied part of the changes.                                     |
|                                     +-----+                                     |
| Restart the server to apply the remaining changes.                           |
|                                     +-----+                                     |
+-----+
=====
...
[DAS]domain1.adminServer>diabile-engines adminServer -web -ejb
Web EJB engine(s) was(were) successfully enabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
|                                     +-----+                                     |
+-----+
```

```
+-----+
| Successfully applied part of the changes.
|
| Restart the server to apply the remaining changes.
|
+-----+
=====
...
```

4.2.3.16. enable-json-command

JsonCommandを活性化します。

- エイリアス

resume-json-command, jsonon

- 使用方法

```
enable-json-command [-f,--forceLock]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>enable-json-command
JsonCommand was successfully enabled.
The configuration was changed.
=====
+-----+
|                                     Result
|
+-----+
| Successfully applied the configuration changes.
|
+-----+
=====
```

4.2.3.17. enable-to-resynchronize-applications

MSがINDEPENDENT状態からDEPENDENT状態になったときに、アプリケーションを同期化します。

- エイリアス

enable-resynch-apps, enable-resyncapps, enableresyncapps

- 使用方法

```
enable-to-resynchronize-applications [-f,--forceLock]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>enable-to-resynchronize-applications
ResyncAppsCommand was successfully enabled.
The configuration was changed.
=====
+-----+
|                                     Result                                     |
+-----+
| Successfully applied the configuration changes.                             |
+-----+
=====
```

4.2.3.18. enable-webadmin

WebAdminサービスを活性化します。

- エイリアス

resume-webadmin, webadminon

- 使用方法

```
enable-webadmin [-f,--forceLock]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer>enable-webadmin
WebAdmin was successfully enabled.
The configuration was changed.
=====
+-----+
|                      Result                      |
+-----+
| Succeeded to apply configuration changes.         |
+-----+
=====
```

4.2.3.19. jndi-info

サーバーのJNDIコンテキストにバインドされているオブジェクトのバインディング(名前)一覧を確認します。

- エイリアス

jndiinfo, jndilist

- 使用方法

```
jndi-info -server <server-name>
          [<context-name>]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[<context-name>]	JNDIコンテキストの名前です。 <context-name>がコンテキストの場合は該当コンテキストにバインディングされているすべてのオブジェクトを表示し、<context-name>がコンテキストではなくリーフ・ノードの場合は該当オブジェクトの詳細情報を出力します

- 例

- コンテキスト

```
[DAS]domain1.adminServer>jndi-info -server server1
The JNDI list on the server1
List of the context /
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
```

```

|
+-----+-----+-----+
| JEUSMQ_DLQ | jeus.jms.common.destination.JeusQueue          | false
|
| mgmt        | jeus.jndi.JNSContext                                     | false
|
+-----+-----+-----+
=====
[DAS]domain1.adminServer>jndi-info -server server1 mgmt
The JNDI list on the server1
List of the context /mgmt
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| rmbs | jeus.jndi.JNSContext | false |
+-----+-----+-----+
=====

```

リーフ・ノード

```

[DAS]domain1.adminServer>jndi-info -server server1 mgmt/rmbs/adminServer
The JNDI list on the server1
List of the context /mgmt/rmbs/server1
=====
+-----+-----+-----+
| Name | Value | Local Binding |
+-----+-----+-----+
| mgmt/rmbs/server1 | javax.management.remote.JMXConnector | false |
+-----+-----+-----+

Reference class name : javax.management.remote.JMXConnector
Reference Factory class name :
jeus.management.remote.jeusmp.JeusmpConnectorFactory
[0] StringRefAddr : jmxmp, RefAddr type : protocol
[1] StringRefAddr : Administrator, RefAddr type : host
[2] StringRefAddr : 19736, RefAddr type : port
[3] StringRefAddr : /JEUSMP_server1, RefAddr type : urlPath
[4] StringRefAddr : 60000, RefAddr type : checktmout
[5] StringRefAddr : 61.77.153.160, RefAddr type : ipaddr
end of RefAddr
=====

```

4.2.3.20. list-jvm-options

サーバーのJVM設定を表示します。

- エイリアス

list-jvmopt, listjvmopt, list-jvm-config, list-jvmcfg, listjvmcfg

- 使用方法

```
list-jvm-options <server-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>list-jvm-config adminServer
List of JVM Configurations
=====
+-----+
|                                     JVM Configs                                     |
+-----+
| -Xmx256m -XX:MaxPermSize=128m |
+-----+
=====
```

4.2.3.21. list-log-handlers

サーバーのロガーに登録されているハンドラの情報を出力します。

- エイリアス

listloghandlers, listhandlers, list-handlers

- 使用方法

```
list-log-handlers -server <server-name>
                  <logger-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
<logger-name>	ロガーの名前です

- 例

```
[DAS]domain1.adminServer>list-log-handlers -server adminServer jeus.deploy
List of Loggers
=====
+-----+-----+
|      Name      |      Value      |
+-----+-----+
| Handler Name   | fileHandler     |
| Handler Type   | FileHandlerType |
| Handler Level  | FINEST          |
| Filename       | jeus.deploy.log |
| Enable Rotation| true            |
| Rotation Directory| ${SERVER_HOME}/logs
| Valid Day      | 1               |
| Buffer Size    | 1024            |
| Logging to Append| true           |
+-----+-----+
=====
```

4.2.3.22. list-loggers

サーバーに登録されているロガーの情報を出力します。

- エイリアス

listloggers

- 使用方法

```
list-loggers <server-name>
              [-logger<logger-name>]
```


- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です
[-logger<logger-name>]	ロガーの名前です

- 例

```
[DAS]domain1.adminServer>list-loggers server1
List of Loggers
=====
+-----+-----+-----+-----+-----+
|   Logger Name   | Level | Use Parent Handlers | Filter | Formatter |
+-----+-----+-----+-----+-----+
|   jeus           | INFO  | false               |        |            |
| jeus.security    | FINE  | true                |        |            |
+-----+-----+-----+-----+-----+
=====
```

4.2.3.23. list-server-listeners

サーバーにあるリスナーの設定を動的に出力します。

- エイリアス

listener, serverlistener

- 使用方法

```
list-server-listeners -server <server-name>
                        [-name <listener-name>]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-name <listener-name>]	出力するリスナーの名前です

- 例

```
[DAS]domain1.adminServer>list-server-listeners -server adminServer
=====
+-----+-----+-----+
|           listener-name           | address | port |
+-----+-----+-----+
```

```

| base | 0.0.0.0 | 9736 |
| jms | 0.0.0.0 | 9741 |
| http-lisener | not-set | 8088 |
+-----+-----+-----+
=====
[DAS]domain1.adminServer>list-server-listeners -server adminServer -name base
=====
base
+-----+-----+-----+
| address | 0.0.0.0 |
| port | 9736 |
| selectors |
| dual-selector | false |
| backlog | 128 |
| read-timeout | 30000 |
| reserved-threads | 0 |
+-----+-----+-----+
=====

```

4.2.3.24. local-shutdown

現在接続しているサーバーを終了します。

- エイリアス

localdown, localjeusexit

- 使用方法

```
local-shutdown [-to,--shutdowntimeout <timeout> | -f,--force | -g,--graceful]
```

- パラメータ

パラメータ	説明
[-to, --shutdowntimeout <timeout>]	<p>グレースフル・シャットダウンを行う際に使用するタイムアウトを設定します。</p> <p>入力した時間の分だけ、処理中の要求の完了を待機します(単位:second)</p>
[-f, --force]	<p>グレースフル・シャットダウンを行わずに、すぐにサーバーを停止します(デフォルト値)</p>
[-g, --graceful]	<p>グレースフル・シャットダウンを行うか否かを設定します。処理中の要求の停止を無限に待機します</p>

- 例

local-shutdownコマンドを使用して、現在接続しているサーバーを終了します。

```
[DAS]domain1.adminServer>local-shutdown
The server [adminServer] has been shut down successfully.
offline>
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.3.25. local-start-server

オンライン・モードでは、現在接続しているサーバーがSTANDBY状態の場合にサーバーを起動します。オフライン・モードでは、SSHを使用して特定のMSを起動します。この際、-node、-domain、-server、-u、-p、-dasurlは必須項目です。

- エイリアス

localstartserver, local-boot

- 使用方法

以下の使用方法は、オンライン・モードの際の基準です。

```
local-start-server [-node <node-name>]
                  [-domain <domain-name>]
                  [-server <server-name>]
                  [-u <user-name>]
                  [-p <password>]
                  [-dasurl <dasurl>]
                  [-f, --force]
                  [-s, --standby]
```

- パラメータ

パラメータ	説明
[-node <node-name>]	ノードの名前です
[-domain <domain-name>]	ドメインの名前です
[-server <server-name>]	サーバーの名前です
[-u <user-name>]	ユーザーの名前です
[-p <password>]	ユーザーのパスワードです
[-dasurl <dasurl>]	ドメイン運用サーバーのアドレスです
[-f, --force]	登録されているアプリケーションがRUNNING状態でなくても、強制的に起動します

パラメータ	説明
[-s, --standby]	サーバーを準備状態であるSTANDBY状態で起動します

- 例

```
offline>local-start-server -node node1 -domain domain1 -server server1 -u
administrator -p password -dasurl 192.168.34.55:9736
The Server [server1] was successfully started.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.3.26. log-level

サーバーのロガーと該当ロガーに登録されているハンドラのレベルを確認します。

- エイリアス

loglevel

- 使用方法

```
log-level -server <server-name>
          <logger-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
<logger-name>	確認するロガーの名前です

- 例

```
[DAS]domain1.adminServer>log-level -server server1 jeus
The logger[jeus] information for the server [server1]
Information about the logger[jeus].
=====
Logger Name : jeus
Level : INFO
Use Parent Handlers : false

+-----+-----+-----+
| Handler Name| Handler Type|          Handler Level          |
+-----+-----+-----+
```

```
| FileHandler | FileHandler | FINEST |
+-----+-----+-----+
=====
```

4.2.3.27. mbean-info

サーバーに登録されているMBeanの一覧を出力します。

- エイリアス

mbeaninfo

- 使用方法

```
mbean-info -server <server-name>
            [<object-name>]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[<object-name>]	オブジェクト名に対する問い合わせです。 特定のオブジェクト名を持つMBeanの一覧を表示する場合に使用します。 問い合わせを行うオブジェクト名については『JEUS JMXガイド』を参照してください

- 例

```
[DAS]domain1.adminServer>mbean-info -server server1
The object names of MBeans on the server [server1].
=====
+-----+
| JEUS:j2eeType=JeusService,jeusType=ServerNodeManagerService,JMXManager=server |
| 1,J2EEServer=server1,name=server1 |
| |
| JEUS:j2eeType=J2EEServer,JMXManager=server1,isTargetable=true,J2EEDomain=doma |
| in1,name=server1 |
| |
| JEUS:j2eeType=JMSResource,JMXManager=server1,J2EEServer=server1,name=server1_|
| jms |
| |
| JEUS:j2eeType=JeusService,jeusType=SecurityDomain,JMXManager=server1,J2EEDoma |
| in=domain1,SecurityService=SecurityService,name=SYSTEM_DOMAIN |
```

```

|
| JEUS:j2eeType=JeusService,jeusType=JEUSMPConnector,JMXManager=server1,J2EESer
| ver=server1,name=server1
|
| JEUS:j2eeType=JeusService,jeusType=ThreadPool,JMXManager=server1,J2EEServer=s
| erver1,name=threadpool.System
|
| JEUS:j2eeType=JeusService,jeusType=WebListener,JMXManager=server1,WebEngine=s
| erver1_servlet,J2EEServer=server1,name=ADMIN-HTTP
|
| JEUS:j2eeType=JeusService,jeusType=ThreadPool,JMXManager=server1,J2EEServer=s
| erver1,name=EJBTimerService
|
| JEUS:j2eeType=JDBCResource,JMXManager=server1,J2EEServer=server1,name=server1
| JEUS:j2eeType=JeusService,jeusType=ThreadPool_WEBC,JMXManager=server1,WebEngi
| ne=server1_servlet,J2EEServer=server1,WebListener=ADMIN-HTTP,name=ADMIN-HTTP
|
| JEUS:j2eeType=JeusService,jeusType=EJBEngine,JMXManager=server1,J2EEServer=se
| rver1,name=server1_ejb
|
| JEUS:j2eeType=JeusService,jeusType=JNDIResourceService,JMXManager=server1,J2E
| EServer=server1,name=server1
|
| JEUS:j2eeType=JeusService,jeusType=SecurityService,JMXManager=server1,J2EEDom
| ain=domain1,J2EEServer=server1,name=SecurityService
|
| JEUS:j2eeType=JeusService,jeusType=WebEngine,JMXManager=server1,J2EEServer=se
| rver1,name=server1_servlet
|
| JEUS:j2eeType=JeusService,jeusType=SecuritySubject,JMXManager=server1,J2EEDom
| ain=domain1,SecurityDomain=SYSTEM_DOMAIN,name=Subject
|
| JEUS:j2eeType=JeusService,jeusType=ThreadPool,JMXManager=server1,J2EEServer=s
| erver1,name=server1
|
| JEUS:j2eeType=JeusService,jeusType=JMSServiceChannel,JMXManager=server1,J2EES
| erver=server1,JMSResource=server1_jms,name=JMSServiceChannel-default
|
| JEUS:j2eeType=Console,JMXManager=server1,J2EEServer=server1,name=server1
|
| JEUS:j2eeType=JeusService,jeusType=JDBCResourceService,JMXManager=server1,J2E
| EServer=server1,name=server1
|
| JEUS:j2eeType=JVM,JMXManager=server1,J2EEServer=server1,name=server1
|
| JEUS:j2eeType=JTAResource,JMXManager=server1,J2EEServer=server1,name=server1
|

```

```
| JEUS:j2eeType=JeusService,jeusType=JMSEngine,JMXManager=server1,J2EEServer=se|
|rver1,name=server1_jms
|
| JEUS:j2eeType=JeusService,jeusType=JMSDestinationResource,JMXManager=server1,|
|J2EEServer=server1,JMSResource=server1_jms,name=JEUSMQ_DLQ
|
| JEUS:j2eeType=JeusService,jeusType=SecurityPolicy,JMXManager=server1,J2EEDoma|
|in=domain1,SecurityDomain=SYSTEM_DOMAIN,name=Policy
|
+-----+
=====
```

4.2.3.28. memory-info

サーバーまたはクラスターに属するサーバーのメモリー情報を確認します。

- エイリアス

memorystatus, memstatus, memoryinfo, meminfo

- 使用方法

```
memory-info [-clusters <cluster-list> | -servers <server-list>]
```

- パラメータ

パラメータ	説明
[-cluster <cluster-name>]	クラスターの名前です。クラスターが1つ以上の場合はコンマ(,)で区分します
[-server <server-name>]	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します

- 例

```
[DAS]domain1.adminServer>memory-info -servers server1
Domain [domain1] Memory Information
Memory Information
=====
+-----+-----+-----+
| Server | Total Amount of Memory | The Current Amount of Memory |
+-----+-----+-----+
| server1 | 110362624 | 55153456 |
+-----+-----+-----+
=====
```

4.2.3.29. modify-jvm-option

サーバーのJVM設定を修正します。

- エイリアス

modify-jvmopt, modifyjvmopt, modify-jvm-config, modify-jvmcfg

- 使用方法

```
modify-jvm-option -server <server-name>
                  [-old,--oldOption <jvm-option>]
                  [-new,--newOption <jvm-option>]
                  [-f,--forceLock]
                  [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-old,--oldOption <jvm-option>]	修正前のJVM設定です。現在のJVM設定と同じです
[-new,--newOption <jvm-option>]	修正後のJVM設定です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	設定変更結果の詳細情報が出力されます

- 例

```
[DAS]domain1.adminServer>modify-jvm-option -server server1 -old "-Xmx256m -XX:Ma
xPermSize=128m" -new "-Xmx512m -XX:MaxPermSize=128m"
Successfully performed the MODIFY operation for JVM configuration for the server
(server1)., but all changes were non-dynamic. They will be applied after
restarting.
Check the results using "list-jvm-options or modify-jvm-option"
```

4.2.3.30. modify-listener

サーバーにあるリスナーの設定を修正します。

- エイリアス

modifylistener, editlistener

- 使用方法


```

modify-listener -server <server-name>
                  -name <listener-name>
                  [-addr <address>]
                  [-port <port>]
                  [-selectors <selectors>]
                  [-dual]
                  [-backlog <backlog>]
                  [-timeout <read-timeout>]
                  [-keepaliveTimeout <keepalive-timeout>]
                  [-rt,--reservedthreads <reserved-threads>]
                  [-f,--forceLock]

```

● パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-name <listener-name>	修正するリスナーの名前です
[-addr <address>]	修正するリスナーのアドレスです
[-port <port>]	修正するリスナーのポート番号です。1~65535の値を設定する必要があり、他のリスナーのポート番号と重複しないようにします
[-selectors <selectors>]	使用するセクタの数です
[-dual]	読み取り/書き込み作業にそれぞれ別のセクタを使用します
[-backlog <backlog>]	修正するリスナーのバックログです
[-timeout <read-timeout>]	[動的変更] 修正するリスナーの読み取りタイムアウトです
[-keepaliveTimeout <keepalive-timeout>]	使用しないソケットを閉じるまでの時間です
[-rt,--reservedthreads <reserved-threads>]	修正するリスナーが使用するReserved Threadの数です
[-f,--forceLock]	修正するためのロックを強制的に取得します

● 例

```

[DAS]domain1.adminServer>modify-listener -server adminServer -name base -dual -b acklog 250
Executed successfully, but some configurations were not applied dynamically. It might be necessary to restart the server.
Check the result using 'list-server-listeners -server adminServer -name base.'

[DAS]domain1.adminServer>list-server-listeners -server adminServer -name base
=====

```

```

base

+-----+-----+
| address          | 0.0.0.0 |
| port             | 9736    |
| selectors        | 1       |
| dual-selector    | true    |
| backlog          | 250     |
| read-timeout     | 30000   |
| keepalive-timeout | not-set |
| reserved-threads | 0       |
+-----+-----+
=====

```

4.2.3.31. modify-log-handler

サーバーのロガーに登録されているハンドラを修正します。

- エイリアス

modifyloghandler, modify-handler, modifyhandler

- 使用方法

```

modify-log-handler -server <server-name>
                    -logger <logger-name>
                    <handler-name>
                    [-level <handler-level>]
                    [-filter <filter-class>]
                    [-encoding <encoding>]
                    [-filename <file-name>]
                    [-enable <enable-rotation>]
                    [-count <rotation-count>]
                    [-dir <rotation-dir>]
                    [-buffer <buffer-size>]
                    [-append <append>]
                    [-hour <hour> | -day <day> | -size <size>]
                    [-f,--forceLock]
                    [-detail]

```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-logger <logger-name>	ロガーの名前です
<handler-name>	変更するハンドラの名前です

パラメータ	説明
[-level <handler-level>]	[動的变化] ハンドラのログ・レベルです(デフォルト値 : FINEST)
[-filter <filter-class>]	ログ・メッセージのフィルター・クラスの名前です
[-encoding <encoding>]	ログ・メッセージのエンコード値です
[-filename <file-name>]	ログ・メッセージを記録するファイルの名前です (デフォルト値 : loggerName.log. jeus loggerの場合は JeusServer.log)
[-enable <enable-rotation>]	ファイル・ローテーションを可能にするか否かを設定します(デフォルト値 : true)
[-count <rotation-count>]	バックアップするログファイルの最大数です
[-dir <rotation-dir>]	バックアップするログファイルが位置するディレクトリー・パスです
[-buffer <buffer-size>]	ログ・メッセージをファイルに記録する際のバッファ・サイズです
[-append <append>]	再起動時に以前のファイルに引き続きログを記録するか否かを設定します (true false、デフォルト値: true)
[-hour <hour>]	入力された時間ごとに新しいログ・ファイルを作成します
[-day <day>]	入力された日ごとに新しいログ・ファイルを作成します
[-size <size>]	入力されたサイズに従って、新しいログ・ファイルを作成します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的变化結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-log-handler -server server1 -logger jeus newHandler -level CONFIG
Successfully performed the MODIFY operation for The handler(FileHandler) for the
    logger(jeus) in server (server1).
Check the results using "modify-log-handler"
```

4.2.3.32. modify-logger

サーバーに登録されているロガーを修正します。

- エイリアス

modifylogger

- 使用方法

```
modify-logger -server <server-name> <logger-name>
               [-level <log-level>]
               [-useParentHandlers <use-parent-handlers>]
               [-filter <filter-class>]
               [-formatter <formatter-class>]
               [-f,--forceLock]
               [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
<logger-name>	ロガーの名前です
[-level <log-level>]	[動的変更] ロガーのログ・レベルです
[-useParentHandlers <use-parent-handlers>]	[動的変更] 上位ロガーのハンドラを使用するか否かを設定します(true false)
[-filter <filter-class>]	ロガーに適用するフィルターのクラス名です
[-formatter <formatter-class>]	ロガーに適用するフォーマッタのクラス名です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-logger -server server1 jeus.security -level FINE
ST -useParentHandlers false
Successfully performed the MODIFY operation for The logger[jeus.security] inform
ation for the server [server1].
Check the results using "modify-logger"
```

4.2.3.33. remove-custom-resource-from-servers

サーバーに登録されているカスタム・リソースを動的に削除します。

- エイリアス

remove-cr-from-servers

- 使用方法

```
remove-custom-resource-from-servers <export-name>
                                     -servers <server-list>
```

```
[-f,--forceLock]
[-detail]
```

- パラメータ

パラメータ	説明
<export-name>	サーバーで削除するカスタム・リソースのJNDIバインド名です
-servers <server-list>	カスタム・リソースを削除するサーバー一覧を設定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-custom-resource-from-servers custom/dog -servers
server1
Successfully performed the REMOVE operation for A custom resource.
Check the results using "list-custom-resources"
```

4.2.3.34. remove-data-sources-from-server

サーバーに登録されているデータソースを動的に削除します。

- エイリアス

rmdsfromsvr

- 使用方法

```
remove-data-sources-from-server -server <server-name>
                                [-ids <data-source-id-list> | -all]
                                [-f,--forceLock]
                                [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	データソースを削除するサーバーの名前です
[-ids <data-source-id-list>]	サーバーから削除するデータソースのID一覧です。 1つ以上の場合はコンマ(,)で区分します
[-all]	サーバーに登録されているすべてのデータソースを削除します
[-f,--forceLock]	設定を強制的に変更して反映します

パラメータ	説明
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-data-sources-from-server -server server1 -ids data-source1
Successfully performed the REMOVE operation for data sources from the server [server1].
Check the results using "remove-data-sources-from-server -server server1"
```

4.2.3.35. remove-external-resource-from-servers

サーバーに登録されている外部リソースを動的に削除します。

- エイリアス

remove-er-from-servers

- 使用方法

```
remove-external-resource-from-servers <export-name>
                                -servers <server-list>
                                [-f,--forceLock]
                                [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	サーバーで削除する外部リソースの名前です
-servers <server-list>	外部リソースを削除するサーバー一覧を設定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-external-resource-from-servers test/ext -servers server1
Successfully performed the REMOVE operation for A external resource.
Check the results using "list-external-resources"
```

4.2.3.36. remove-jvm-option

サーバーのJVM設定を削除します。

- エイリアス

remove-jvmopt, rm-jvmopt, rmjvmopt, remove-jvm-config, remove-jvmcfg, rm-jvmcfg, rmjvmcfg

- 使用方法

```
remove-jvm-option -server <server-name>
                  [-opt,--jvmOption <jvm-option>]
                  [-f,--forceLock]
                  [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-opt,--jvmOption <jvm-option>]	サーバーから削除するJVM設定です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	設定変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-jvm-option -server server1 -opt "-Xmx256m -XX:MaxPermSize=128m"
Successfully performed the REMOVE operation for JVM configuration for the server (server1)., but all changes were non-dynamic. They will be applied after restarting.
Check the results using "list-jvm-options or remove-jvm-option"
```

4.2.3.37. remove-listener

サーバーにあるリスナーを削除します。baseリスナーや他の設定で参照しているリスナーは削除できません。

- エイリアス

removelistener, deletelistener

- 使用方法

```
remove-listener -server <server-name>
                -name <listener-name>
                [-f,--forceLock]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-name <listener-name>	削除するリスナーの名前です
[-f,--forceLock]	修正するためのロックを強制的に取得します

- 例

```
[DAS]domain1.adminServer>list-server-listeners -server adminServer
=====
+-----+-----+-----+
|          listener-name          | address | port |
+-----+-----+-----+
| base                            | 0.0.0.0 | 9736 |
| jms                            | 0.0.0.0 | 9741 |
| http-lisetner                  | not-set | 8088 |
| Test                           | not-set | 10000 |
+-----+-----+-----+
=====

[DAS]domain1.adminServer>remove-listener -server adminServer -name Test
Executed Successfully, but some configurations were not applied dynamically. It
might be necessary to restart the server.
Check the result using 'list-server-listeners -server adminServer.'
[DAS]domain1.adminServer>list-server-listeners -server adminServer
=====
+-----+-----+-----+
|          listener-name          | address | port |
+-----+-----+-----+
| base                            | 0.0.0.0 | 9736 |
| jms                            | 0.0.0.0 | 9741 |
| http-lisetner                  | not-set | 8088 |
+-----+-----+-----+
=====
```

4.2.3.38. remove-log-handler

サーバーのロガーに登録されているハンドラを動的に削除します。

- エイリアス

removeloghandler, removehandler, remove-handler

- 使用方法

```
remove-log-handler -server <server-name>
                    -logger <logger-name>
                    [<handler-name>]
```



```
[-f,--forceLock]
[-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-logger <logger-name>	ハンドラを削除するロガーの名前です
[<handler-name>]	削除するハンドラの名前です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-log-handler -server server1 -logger jeus
newHandler
Successfully performed the REMOVE operation for The handler for the logger(jeus)

on the server(server1)..
Check the results using "list-log-handlers or remove-log-handler"
```

4.2.3.39. remove-logger

サーバーに登録されているロガーを動的に削除します。

- エイリアス

removelogger

- 使用方法

```
remove-logger -server <server-name>
               [<logger-name>]
               [-f,--forceLock]
               [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[<logger-name>]	削除するロガーの名前です
[-f,--forceLock]	設定を強制的に変更して反映します

パラメータ	説明
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-logger -server server1 jeus.security
Successfully performed the REMOVE operation for The logger for the
server(server1)..
Check the results using "list-loggers or remove-logger"
```

4.2.3.40. resume-server

一時停止していたサーバーまたはクラスターを再度サービスするようにします。

- エイリアス

resume

- 使用方法

```
resume-server [-clusters <cluster-list> | -servers <server-list>]
```

- パラメータ

パラメータ	説明
[-clusters <cluster-list>]	クラスターの名前です。クラスターが1つ以上の場合はコンマ(,)で区分します
[-servers <server-list>]	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します

- 例

```
[DAS]domain1.adminServer>resume-server -servers server1
Successfully resumed the servers.
```

4.2.3.41. rollback-config

バックアップされている設定ファイルにロールバックします。ただし、コマンドを利用してロールバックを行う際は動作が動的に反映されないため、サーバーを再起動する必要があります。ただし、WebAdminでロールバックを行う際は自動的に動的反映がされるため、コマンドよりはWebAdminを使用したロールバックを推奨します。

- 使用方法

```
rollback-config -name <file-name>
                [-des <description>]
```

- パラメータ

パラメータ	説明
-name <file-name>	ロールバックするバックアップ設定ファイルの名前です
[-des <description>]	ロールバックした場合、現在の設定ファイルがバックアップされます。バックアップされる現在の背景ファイルについての説明です

- 例

```
[DAS]domain1.adminServer>rollback-config -name domain_20130420_210914.xml
Rollback of backup domain.xml file is done successfully. You should restart all
servers in domain.
```

4.2.3.42. run-garbage-collection

サーバーまたはクラスターに属するサーバーにFull Garbage Collectionを実行します。

- エイリアス

runcg, gc

- 使用方法

```
run-garbage-collection [-clusters <cluster-list> | -servers <server-list>]
```

- パラメータ

パラメータ	説明
[-clusters <cluster-list>]	クラスターの名前です。クラスターが1つ以上の場合はコンマ(,)で区分します
[-servers <server-list>]	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します

- 例

```
[DAS]domain1.adminServer>run-garbage-collection -servers server1
=====
+-----+-----+-----+
| Server |           Result           |
+-----+-----+-----+
| server1| Succeeded to GC.           |
```

```
+-----+-----+-----+-----+-----+-----+
=====
```

4.2.3.43. server-info

サーバーの現在の状態を確認します。

- エイリアス

svrinfo, serverinfo, si

- 使用方法

```
server-info [-server <server-name>]
            [-state | -pid]
            [-g, --group <group-name> | -c, --cluster <cluster-name> |
            -n, --node <node-name>]
```

- パラメータ

パラメータ	説明
[-server <server-name>]	サーバー情報を出力します
[-state]	サーバーを一緒に指定した場合、指定したサーバーの状態のみを出力します
[-pid]	サーバーを一緒に指定した場合、指定したサーバーのpidのみを出力します
[-g, --group <group-name>]	指定したグループに分類されたサーバー情報を出力します
[-c, --cluster <cluster-name>]	指定したクラスターに分類されたサーバー情報を出力します
[-n, --node <node-name>]	指定したノードに分類されたサーバー情報を出力します

- 例

```
[DAS]domain1.adminServer>server-info
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clus | Latest | Need to | Listen Ports | Running |
|        |        | Name |      | ter | StartTime/Shut | Restart |              | Engines |
|        |        |      |      |    | downTime      |        |              |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| adminSe | RUNNING | N/A | 12154 | N/A | 2016-08-26 | false | base-0.0.0.0 | jms,
| rver(*) | (00:21:46 |      |      |    | (金) 午後 |      | :9736 | ejb, web
```

```
|
|      |)      |      |      |02:29:03 KST      |      | http-server-|
|
|      |      |      |      |      |      |0.0.0.0:8088 |
|
+-----+-----+-----+-----+-----+-----+-----+-----+
| server1 | RUNNING | node1 | 12171 | clus | 2016-08-26      | false | base-192.167 | jms,
|
|      | (00:04:03 |      |      | ter1 | (金) 午後      |      | .0.124:9836 | ejb, web
|
|      |)      |      |      |      |02:36:22 KST      |      |      |
|
+-----+-----+-----+-----+-----+-----+-----+-----+
| server2 | RUNNING | node1 | 12172 | clus | 2016-08-26      | false | base-192.167 | jms
|
|      | (00:04:03 |      |      | ter1 | (金) 午後      |      | .0.124:9936 | ejb, web
|
|      |)      |      |      |      |02:36:22 KST      |      |      |
|
+-----+-----+-----+-----+-----+-----+-----+-----+
=====
```

4.2.3.44. show-config

現在の設定ファイルとバックアップされた設定ファイル一覧を表示します。

- 使用方法

```
show-config
```

- 例

```
[DAS]domain1.adminServer>show-config
=====
show previous configurations.

+-----+-----+
|               fileName               | description |
+-----+-----+
| domain.xml                           |             |
| domain_20160513_170151.xml           |             |
| domain_20160513_165917.xml           |             |
| domain_20160513_163355.xml           |             |
| domain_20160513_160642.xml           |             |
| domain_20160513_160229.xml           |             |
| domain_20160513_154155.xml           |             |
+-----+-----+
=====
```

4.2.3.45. start-cluster

クラスターに属するMSを起動します。SHUTDOWN状態またはSTANDBY状態の場合に使用可能です。サーバーがSHUTDOWN状態の場合にはSSHが正しく設定されていなければ動作しません。

- エイリアス

startcluster

- 使用方法

```
start-cluster <cluster-name>
               [-f, --force]
               [-s, --standby]
               [-seq, --sequential]
```

- パラメータ

パラメータ	説明
<cluster-name>	クラスターの名前です
[-f, --force]	強制的にサーバーを起動します
[-s, --standby]	クラスターを準備状態であるSTANDBY状態で起動します
[-seq, --sequential]	クラスターに属するサーバーを順次起動します

- 例

```
[DAS]domain1.adminServer>start-cluster cluster1
The cluster [cluster1] has been successfully started.
```

4.2.3.46. start-domain

現在ドメインに属しているすべてのMSを起動します。SHUTDOWN状態またはSTANDBY状態の場合に使用可能です。サーバーがSHUTDOWN状態の場合はSSHが正しく設定されていなければ動作しません。

- エイリアス

startdomain, bootall

- 使用方法

```
start-domain [-f, --force]
              [-s, --standby]
              [-seq, --sequential]
```

- パラメータ

パラメータ	説明
[-f, --force]	強制的にサーバーを起動します
[-s, --standby]	サーバーを準備状態であるSTANDBY状態で起動します
[-seq, --sequential]	ドメインに属するサーバーを順次起動します

- 例

```
[DAS]domain1.adminServer>start-domain
All managed servers in the domain have been successfully started.
```

4.2.3.47. start-server

MSを起動します。SHUTDOWN状態またはSTANDBY状態の場合に使用可能です。該当サーバーがSHUTDOWN状態の場合はSSHが正しく設定されていなければ動作しません。

- エイリアス

startserver, boot

- 使用方法

```
start-server <server-name>
          [-f, --force]
          [-s, --standby]
          [-seq,--sequential]
```

- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です
[-f, --force]	強制的にサーバーを起動します
[-s, --standby]	サーバーを準備状態であるSTANDBY状態で起動します
[-seq, --sequential]	与えられたサーバーを順次起動します

- 例

```
[DAS]domain1.adminServer>start-server server1
The server [server1] was successfully started.
```

4.2.3.48. stop-cluster

クラスターに属するMSを停止します。

- エイリアス

stopcluster, shutdowncluster, downcluster

- 使用方法

```
stop-cluster <cluster-name>  
          [-to,--shutdowntimeout <timeout> | -f, --force | -g,--graceful]
```

- パラメータ

パラメータ	説明
<cluster-name>	クラスターの名前です
[-to,--shutdowntimeout <time-out>]	グレースフル・シャットダウンを行う際に使用するタイムアウトを設定します。 入力した時間の分だけ、処理中の要求の終了を待機します(単位: ms)
[-f, --force]	グレースフル・タイムアウトを適用せずに強制的に停止するか否かを設定します
[-g, --graceful]	グレースフル・シャットダウンを行うか否かを設定します。処理中の要求の終了を無限に待機します

- 例

```
[DAS]domain1.adminServer>stop-cluster cluster1  
The cluster [cluster1] was successfully stopped.
```

4.2.3.49. stop-domain

現在ドメインに属しているMSを停止します。DASの停止は[local-shutdown](#)コマンドで可能です。

- エイリアス

stopdomain, shutdownall, downall

- 使用方法

```
stop-domain [-to,--shutdowntimeout <timeout> | -f, --force | -g,--graceful]
```

- パラメータ

パラメータ	説明
[-to,--shutdowntimeout <time-out>]	グレースフル・シャットダウンを行う際に使用するタイムアウトを設定します。 入力した時間の分だけ、処理中の要求の完了を待機します(単位: second)
[-f, --force]	グレースフル・タイムアウトを適用せずに強制的に停止するか否かを設定します
[-g, --graceful]	グレースフル・シャットダウンを行うか否かを設定します。処理中の要求の停止を無限に待機します

- 例

```
[DAS]domain1.adminServer>stop-domain
All managed servers in the domain were successfully stopped.
Unable to shut down Domain Administration Server. To shutdown Domain Administration Server, use the local-shutdown command instead.
```

4.2.3.50. stop-server

入力した単一または多数のMSを終了します。タイムアウトを設定したり、gracefulオプションを指定して安全に終了することができます。

- エイリアス

stopserver, down, shutdown, jeusexit

- 使用方法

```
stop-server <server-list>
          [-to,--shutdowntimeout <timeout> | -f, --force | -g,--graceful]
```

- パラメータ

パラメータ	説明
<server-list>	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-to,--shutdowntimeout <timeout>]	グレースフル・シャットダウンを行う際に使用するタイムアウトを設定します。 入力した時間の分だけ、処理中の要求の完了を待機します(単位: second)

パラメータ	説明
[-f, --force]	グレースフル・シャットダウンを行わずに、すぐにサーバーを停止します(デフォルト値)
[-g, --graceful]	グレースフル・シャットダウンを行うか否かを設定します。処理中の要求の停止を無限に待機します

- 例

```
[DAS]domain1.adminServer>stop-server server1
The server [server1] was successfully stopped.
```

4.2.3.51. suspend-server

サービス中のサーバーを一時停止します。

- エイリアス

suspend

- 使用方法

```
suspend-server [-clusters <cluster-list> / -servers <server-list>]
```

- パラメータ

パラメータ	説明
[-clusters <cluster-list>]	クラスターの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-servers <server-list>]	サーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します

- 例

```
[DAS]domain1.adminServer>suspend-server -servers server1
Successfully suspended server(s).
```

4.2.3.52. system-info

サーバーのシステム情報を確認します。

- エイリアス

systeminfo, sysinfo, sys-info, system-information, system-ststus, systemstatus, systemst

- 使用方法

```
system-info [-process,--process | -jvm,--jvm | -CPU,--cpu |  
            -patch,--patch | -memory,--memory]  
            <server-name>
```

- パラメータ

パラメータ	説明
[-process,--process]	プロセス情報を出力します
[-jvm,--jvm]	JVM情報を出力します
[-CPU,--cpu]	CPU情報を出力します
[-patch,--patch]	サーバーに適用されたパッチ情報を出力します
[-memory,--memory]	メモリー情報を出力します
<server-name>	システム情報を出力するサーバーの名前です

- 例

```
[DAS]domain1.adminServer>system-info -process adminServer  
System Information of Server [adminServer]  
Process Information [adminServer]  
=====
```

Key	Value
Process Id	8412
Thread Count	96
Process Priority	8
Nice	-1
Memory Resident	354,836,480 bytes
Memory Share	-1
Page Faults	484667
CPU Percent	0.0 %
User Name	User

```
=====
```

4.2.3.53. server-log

サーバーのログ情報を確認します。

- エイリアス

serverlog

- 使用方法

```
server-log -server <server-name>
           [-basetime, <base-time>]
           [-endtime, <end-time>]
           [-line, <number-of-line>]
```

- パラメータ

パラメータ	説明
-server <server-name>	ログを取得するサーバーを指定します
[-basetime, <base-time>]	ログの開始位置に該当するポイントを時間で指定します (形式: "yyyy.MM.dd HH:mm:ss" または "HH:mm:ss")
[-endtime, <end-time>]	ログの終了位置に該当するポイントを時間で指定します (形式: "yyyy.MM.dd HH:mm:ss" "HH:mm:ss")
[-line, <number-of-line>]	ログの開始位置に該当するポイントを行で指定します

- 例

```
[DAS]domain1.adminServer>server-log -server adminServer -line 10
[2016.08.26 20:00:39][2] [adminServer-33] [NodeManager-0062] Exceed to count to
connecting to the node manager. Verify that the node manager is functioning
normally.
[2016.08.26 20:00:59][2] [adminServer-33] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:00:59][2] [adminServer-33] [Console-0012] An exception occurred
while processing a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:01:45][2] [adminServer-76] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:01:45][2] [adminServer-76] [Console-0012] An exception occurred
while processing a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:02:26][2] [adminServer-36] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:02:26][2] [adminServer-36] [Console-0014] The command: server-log
from heejin@192.168.15.59 is executed successfully.
[2016.08.26 20:02:40][2] [adminServer-73] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
[2016.08.26 20:02:40][2] [adminServer-73] [Console-0014] The command: server-log
from heejin@192.168.15.59 is executed successfully.
[2016.08.26 20:08:16][2] [adminServer-76] [Console-0010] Attempting to execute
a command: server-log from heejin@192.168.15.59.
```

```
Successfully got the log messages from the server(adminServer).  
[DAS]domain1.adminServer>
```

4.2.4. ドメイン設定関連コマンド

ドメイン設定に関連するコマンドの一覧は以下のとおりです。

コマンド	説明
add-cluster	クラスターを動的に追加します
add-custom-resource	カスタム・リソースを動的に追加します
add-custom-resource-to-clusters	クラスターにカスタム・リソースを動的に追加します
add-data-sources-to-cluster	クラスターにデータソースを動的に登録します
add-external-resource	外部リソースを動的に追加します
add-external-resource-to-clusters	クラスターに外部リソースを動的に追加します
add-server	サーバーを動的に追加します
add-servers-to-cluster	サーバーをクラスターに動的に追加します
add-server-template	複数のサーバーに共通して動的に適用できる設定のテンプレートを追加します
add-dynamic-servers	あらかじめ作成した設定テンプレートに基づいて自動で作成されたサーバーで構成されたクラスターを追加します
apply-configuration-plan	既存のdomain.xmlにconfiguration planを適用した、新しいdomain.xmlを作成するオフライン・コマンドです
create-domain	新しいドメインを作成するオフライン・コマンドです。基本で作成されるユーザーのパスワードは別途入力する必要があります
delete-domain	作成されるドメインを削除するオフライン・コマンドです
list-clusters	現在ドメインに存在するクラスターの情報を確認します
list-custom-resources	現在ドメインに設定されているカスタム・リソースを確認します
list-external-resources	現在ドメインに設定されている外部リソースを確認します
list-domains	現在構成されているドメインの一覧を表示するオフライン・コマンドです
list-servers	現在ドメインに存在するサーバーの情報を確認します
modify-cluster	クラスターの設定を動的に変更します
modify-server	サーバーの設定を動的に変更します
modify-dynamic-servers	Dynamic Servers設定に基づいて作成されたクラスターの設定情報を変更します
remove-cluster	クラスターを動的に削除します

コマンド	説明
remove-custom-resource	カスタム・リソースを動的に削除します
remove-custom-resource-from-clusters	クラスターに登録されているカスタム・リソースを動的に削除します
remove-data-sources-from-cluster	クラスターに登録されているデータソースを動的に削除します
remove-external-resource	外部リソースを動的に削除します
remove-external-resource-from-clusters	クラスターに登録されている外部リソースを動的に削除します
remove-server	サーバーを動的に削除します
remove-servers-from-cluster	クラスターからサーバーを動的に削除します
set-domain-backup	DASの起動時に自動でドメインのバックアップ・ファイルを作成するオンライン・コマンドです
pack-domain	ドメイン・フォルダーを圧縮するオフライン・コマンドです
show-dynamic-servers	動的サーバーの作成により生成されたクラスターの設定情報を表示します
show-server-template	設定されたサーバー・テンプレートの一覧を照会します
unpack-domain	pack-domainコマンドによって圧縮されたファイルを解凍するオフライン・コマンドです

4.2.4.1. add-cluster

クラスターを動的に追加します。

- 使用方法

```
add-cluster [<cluster-name>]
            [-servers <server-list>]
            [-f,--forceLock]
            [-detail]
```

- パラメータ

パラメータ	説明
[<cluster-name>]	追加するクラスターの名前です
[-servers <server-list>]	このクラスターに含めるサーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

- 現在存在するクラスターの確認

```
[DAS]domain1.adminServer>list-clusters
List of Clusters
=====
+-----+-----+-----+-----+-----+
| Cluster | Server List | MEJB | Class FTP | Running |
+-----+-----+-----+-----+-----+
| cluster1 | server1, server2 | false | false | true |
+-----+-----+-----+-----+-----+
=====
```

- 現在存在していないクラスターの名前でクラスターを作成し、他のクラスターに含まれていないサーバーでクラスターを構成する場合

```
[DAS]domain1.adminServer>add-cluster cluster2 -servers server3,server4
Successfully performed the ADD operation for cluster (cluster2).
Check the results using "list-clusters or add-cluster"
```

4.2.4.2. add-custom-resource

カスタム・リソースを動的に追加します。

- エイリアス

addcustomresource, addcr

- 使用方法

```
add-custom-resource [<export-name>]
                    [-resource,--resourceClass <resource-class>]
                    [-factory,--factoryClass <factory-class>]
                    [-p,--properties <properties>]
                    [-servers <server-list>]
                    [-clusters <cluster-list>]
                    [-f,--forceLock]
                    [-detail]
```

- パラメータ

パラメータ	説明
[<export-name>]	追加するカスタム・リソースのJNDIバインドの名前です
[-resource,--resourceClass <resource-class >]	カスタム・リソースのクラス名を指定します
[-factory,--factoryClass <factory-class >]	カスタム・リソースを作成するファクトリー・クラスの名前を指定します

パラメータ	説明
[-p,--properties <properties>]	カスタム・リソースのプロパティを指定します
[-servers <server-list>]	カスタム・リソースをドメインに追加すると同時に、与えられたサーバー一覧に登録します
[-clusters <cluster-list>]	カスタム・リソースをドメインに追加すると同時に、与えられたクラスター一覧に登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-custom-resource custom/dog -resource dog.Dog -factory
dog.DogFactory
Successfully performed the ADD operation for A custom resource.
Check the results using "list-custom-resources or add-custom-resource"
```

4.2.4.3. add-custom-resource-to-clusters

クラスターにカスタム・リソースを動的に追加します。

- エイリアス

add-cr-to-clusters

- 使用方法

```
add-custom-resource-to-clusters <export-name>
    -clusters <cluster-list>
    [-f,--forceLock]
    [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	クラスターに追加するカスタム・リソースのJNDIバインド名です
-clusters <cluster-list>	カスタム・リソースを与えられたクラスター一覧に登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例


```
[DAS]domain1.adminServer>add-custom-resource-to-clusters custom/dog -clusters
cluster1
Successfully performed the ADD operation for A custom resource.
Check the results using "list-custom-resources"
```

4.2.4.4. add-data-sources-to-cluster

クラスターにデータソースを動的に登録します。

- エイリアス

adddstocluster

- 使用方法

```
add-data-sources-to-cluster -cluster <cluster-name>
                             [-ids <data-source-id-list> | -all]
                             [-f,--forceLock]
                             [-detail]
```

- パラメータ

パラメータ	説明
-cluster<cluster-name>	データソースに登録するクラスターの名前です
[-ids <data-source-id-list>]	クラスターに登録するデータソースのID一覧です。1つ以上の場合はコンマ(,)で区分します
[-all]	ドメインに定義されているすべてのデータソースをクラスターに登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-data-sources-to-cluster -cluster cluster1 -ids
datasource1
Successfully performed the ADD operation for data sources to the cluster [cluste
r1].
Check the results using "add-data-sources-to-cluster -cluster cluster1"
```

4.2.4.5. add-external-resource

外部リソースを動的に追加します。

- エイリアス

addexternalresource, adder

- 使用方法

```
add-external-resource [<export-name>]
                        [-class,--resourceClass <resource-class>]
                        [-p,--properties <properties>]
                        [-servers <server-list>]
                        [-clusters <cluster-list>]
                        [-f,--forceLock]
                        [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	追加する外部リソースの名前です
[-class,--resourceClass <resource-class >]	外部リソースのクラス名を指定します
[-p,--properties <properties>]	外部リソースのプロパティを指定します
[-servers <server-list>]	外部リソースをドメインに追加すると同時に、与えられたサーバー一覧に登録します
[-clusters <cluster-list>]	外部リソースをドメインに追加すると同時に、与えられたクラスター一覧に登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-external-resource test/ext -resource
test.ext.TestResourceBootstrapper
Successfully performed the ADD operation for A external resource.
Check the results using "list-external-resources or add-external-resource"
```

4.2.4.6. add-external-resource-to-clusters

外部リソースをクラスターに動的に追加します。

- エイリアス

add-er-to-clusters

- 使用方法

```
add-external-resource-to-clusters <export-name>
    -clusters <cluster-list>
    [-f,--forceLock]
    [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	クラスターに追加する外部リソースの名前です
-clusters <cluster-list>	外部リソースを与えられたクラスター一覧に登録します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的变化結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-external-resource-to-cluster test/ext -clusters
cluster1
Successfully performed the ADD operation for A external resource.
Check the results using "list-external-resources"
```

4.2.4.7. add-server

サーバーを動的に追加します。

- エイリアス

addserver

- 使用方法

```
add-server [<server-name>]
    [-target,--duptargetserver <target-server>]
    [-port,--baseport <base-port>]
    [-addr,--baseaddr <base-addr>]
    [-node,--nodeName <node-name>]
    [-jvm,--jvmOptions <jvm-options>]
    [-logdir,--logHomeDirectory <server-log-home-directory>]
    [-a,--actionOnResourceLeak <action-on-resource-leak>]
    [-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
    [-m,--mejb <enable-MEJB>]
    [-c,--classFtp <enable-class-ftp-service>]
    [-f,--forceLock]
    [-detail]
```

- パラメータ

パラメータ	説明
[<server-name>]	サーバーの名前です
[-target,--duptargetserver <target-server>]	サーバーの設定をコピーするターゲット・サーバーの名前です
[-port,--baseport <base-port>]	サーバーの基本ポート番号です。 デフォルト値として9736が指定されますが、同じマシンに複数のサーバーを作成する場合は区分が必要な設定です
[-addr,--baseaddr <base-addr>]	サーバーの基本アドレスです
[-node,--nodeName <node-name>]	サーバーが実際に動作するマシンを意味するノードの名前です。 list-nodesコマンドを使用してノードの情報を確認できます
[-jvm,--jvmOptions <jvm-options>]	サーバーに適用しているJVMオプションです。 空白が存在する場合は二重引用符(" ")で囲みます
[-logdir,--logHomeDirectory <server-log-home-directory>]	サーバーで生成されるログが含まれるログ・ディレクトリーです
[-a,--actionOnResourceLeak <action-on-resource-leak>]	リソース・リークが発生した場合に行うアクションを設定します。 以下のうち1つを入力します – AutoClose – NoAction – Warning
[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]	[動的変更] stdoutをそのまま出力するのか、JEUSロガーのフォーマットで出力するのかを設定します(true false)
[-m,--mejb <enable-MEJB>]	[動的変更] MEJBを使用するか否かを設定します(true false)
[-c,--classFtp <enable-class-ftp-service>]	[動的変更] class FTPサービスを使用するか否かを設定します(true false)
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例
 - 現在存在するサーバー一覧の確認

```
[DAS]domain1.adminServer>list-servers
List of Editable Servers and Server Templates
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Server| Base |  Base | Node|  JVM  | Action| Stdout| MEJB| Class| Server|
Type|
|      |Listen|Listen |      | Configs| On    |to Raw |      | FTP  | Log   |
|      |      |      |      |      |      |      |      |      |      |
|      |Addres| Port  |      |      |Resourc|Format |      |      | Home  |
|      |      |      |      |      |e Leak |      |      |      |Directo|
|      |      |      |      |      |      |      |      |      |ry     |
|      |      |      |      |      |      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| admin| 0.0.0| 9736  |node1| -Xmx25|Warning| true  |false| true | none  |
serv|
|Server|.0    |      |      |6m     |      |      |      |      |      |er
|      |      |      |      |-XX:Max|      |      |      |      |      |
|      |      |      |      |PermSiz|      |      |      |      |      |
|      |      |      |      |e=128m |      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| serve|      | 9836  |node1| -Xmx51|Warning| true  |false| false| none  |
serv|
|rl    |      |      |      |2m     |      |      |      |      |      |er
|      |      |      |      |-XX:Max|      |      |      |      |      |
|      |      |      |      |PermSiz|      |      |      |      |      |
|      |      |      |      |e=128m |      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====
[DAS]domain1.adminServer>
```

– 現在存在しない名前でサーバーを追加

```
[DAS]domain1.adminServer>add-server server2 -port 10000 -node node1 -m false
Successfully performed the ADD operation for server (server2).
Check the results using "list-servers or add-server"
```

4.2.4.8. add-servers-to-cluster

サーバーをクラスターに動的に追加します。

- 使用方法

```
add-servers-to-cluster <cluster-name>
                        [-servers <server-list>]
                        [-f,--forceLock]
                        [-detail]
```

- パラメータ

パラメータ	説明
<cluster-name>	クラスターの名前です
[-servers <server-list>]	このクラスターに含めるサーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-servers-to-cluster cluster1
-servers server1
Successfully performed the ADD operation for The server list for
cluster(cluster1)..
Check the results using "list-clusters cluster1 or add-servers-to-cluster
cluster1"
```

4.2.4.9. add-server-template

複数のサーバーに共通して動的に適用できる設定のテンプレートを追加します。

- 使用方法

```
add-server-template <server-template-name>
                    [-jvm,--jvmOptions <jvm-options>]
                    [-logdir,--logHomeDirectory <server-log-home-directory>]
                    [-a,--actionOnResourceLeak <action-on-resource-leak>]
                    [-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
                    [-m,--mejb <enable-MEJB>]
                    [-c,--classFtp <enable-class-ftp-service>]
                    [-f,--forceLock]
                    [-detail]
```

- パラメータ

パラメータ	説明
<server-template-name>	追加するサーバー・テンプレートの名前です
[-jvm,--jvmOptions <jvm-options>]	テンプレートに追加するJVMオプションです。 空白が存在する場合は二重引用符(" ")で囲みます
[-logdir,--logHomeDirectory <server-log-home-directory>]	テンプレートで設定されたサーバーで生成されるログが共通して含まれるログ・ディレクトリです
[-a,--actionOnResourceLeak <action-on-resource-leak>]	リソース・リークが発生した場合に行うアクションを設定します。 以下のうち1つを設定します – AutoClose – NoAction – Warning
[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]	[動的変更] stdoutをそのまま出力するのか、JEUSログのフォーマットで出力するのかを設定します(true false)
[-m,--mejb <enable-MEJB>]	[動的変更] MEJBを使用するか否かを設定します(true false)
[-c,--classFtp <enable-class-ftp-service>]	[動的変更] class FTPサービスを使用するか否かを設定します(true false)
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-server-template templ-a Warning
Successfully performed the ADD operation for server template (templ), but all
changes were non-dynamic. They will be applied after restarting.
Check the results using "list-servers or show-server-template or
add-server-template".
```

4.2.4.10. add-dynamic-servers

あらかじめ設定したテンプレートに基づいて自動で作成されたサーバーで構成されたクラスターを作成します。あらかじめ設定情報をテンプレートに保存してから使用する必要があります。

- エイリアス

addservers

- 使用方法

```
add-dynamic-servers <cluster-name>
    -count <the-number-of-dynamic-servers>
    [-nodes,--nodeNames <node-names>]
    [-base <base-port>]
    [-http <http-port>]
    [-jms <jms-port>]
    [-prefix,--serverNamePrefix <server-name-prefix>]
    -template,--serverTemplateName <server-template-name>
    [-step,--listenPortStep <listen-port-step>]
    [-f,--forceLock]
    [-detail]
```

- パラメータ

パラメータ	説明
<cluster-name>	動的設定サーバーで作成するクラスターの名前です
-count <the-number-of-dynamic-servers>	クラスターを作成するサーバーの数を設定します
[-nodes,--nodeNames <node-names>]	追加されるサーバーが位置するノードを設定します。複数のサーバーをコンマ(,)を区切り子に使用して設定できます
[-base <base-port>]	サーバーの基本動作ポートを設定します
[-http <http-port>]	サーバーのHTTPポートを設定します
[-jms <jms-port>]	サーバーのJMSポートを設定します
[-prefix,--serverNamePrefix <server-name-prefix>]	作成されるサーバーの名前の前に付けるプレフィックスを設定します。 プレフィックスが設定されていない場合は、クラスターの名前の後ろに「_server_」が付けられます
-template,--serverTemplateName <server-template-name>	動的サーバーに共通して適用される設定テンプレートを指定します
[-step,--listenPortStep <listen-port-step>]	同じノードに作成されたサーバーのポート番号の増分を指定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>add-dynamic-servers cluster1 -count 2 -nodes node1
-template temp1
Successfully performed the ADD operation for Dynamic servers configuration in
```



```
the cluster (cluster1).  
Check the results using "list-clusters or show-dynamic-servers".
```

4.2.4.11. apply-configuration-plan

既存のdomain.xmlにconfiguration planを適用した、新しいdomain.xmlを作成するコマンドです。

- エイリアス

applycp

- 使用方法

```
apply-configuration-plan -plan,--configPlanPath <configuration-plan-path>  
                           -src,--configSourcePath <configuration-source-path>  
                           -target,--configTargetPath <configuration-target-path>
```

- パラメータ

パラメータ	説明
-plan,--configPlanPath <configuration-plan-path>	適用するconfiguration planのパスを指定します
-src,--configSourcePath <configuration-source-path>	configuration planを適用する既存のdomain.xmlのパスを指定します
-target,--configTargetPath <configuration-target-path>	configuration planが適用された結果ファイルのパスを指定します

- 例

```
offline>apply-configuration-plan -src /home/jeus/domains/domain1/config/domain.xml  
  
-plan /home/plans/jeus-configuration-plan.xml -target  
/home/jeus/domains/domain1/config/domain.xml.new  
Successfully applied configuration plan.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.4.12. create-domain

新しいドメインを作成するコマンドです。基本で作成されるユーザーのパスワードは別途入力する必要があります。

ユーザーのパスワードはAESで暗号化されて保存されます。オプションのデフォルト値はJEUS_HOME/setup/domain-config-template.propertiesの値に従います。

- エイリアス

createdomain

- 使用方法

```
create-domain -domain <domain-name>
               [-das <das-name>]
               [-baseport <das-base-port>]
               [-baseaddress <das-base-address>]
               [-jmsport <jms-port>]
               [-user <user-name>]
               [-httpport <http-port>]
               [-jvmconfig <jvm-config>]
               [-node <node-name>]
               [-pw,--password]
               [-uvm,--usevirtualmulticast]
               [-productionmode,--productionmode]
               [-ha,--heartbeataddress <heart-beat-address>]
               [-hp,--heartbeatport <heart-beat-port>]
```

- パラメータ

パラメータ	説明
-domain <domain-name>	ドメインの名前です
[-das <das-name>]	ドメインを管理するDASの名前です
[-baseport <das-base-port>]	DASの基本動作ポート番号です
[-baseaddress <das-base-address>]	DASの基本動作アドレスです
[-jmsport <jms-port>]	DASのJMSポート番号です
[-user <user-name>]	ドメインのユーザー名です
[-httpport <http-port>]	DASのHTTPポート番号です
[-jvmconfig <jvm-config>]	DASに適用するJVMの設定です
[-node <node-name>]	DASに適用するノード名です
[-pw,--password]	テンプレート・プロパティ・ファイルに関係なく、コンソールから無条件にパスワードの入力を受けます
[-uvm,--usevirtualmulticast]	SCFでTCPを利用したバーチャル・マルチキャストを使用するか否かを設定します

パラメータ	説明
[-productionmode,--productionmode]	作成するJEUSのドメインをプロダクション・モードで使用するかどうかを設定します
[-ha,--heartbeataddress <heart-beat-address>]	SCFでハートビートに使用するアドレスです。 --usevirtualmulticastの設定がされていはいりません
[-hp,--heartbeatport <heart-beat-port>]	SCFでハートビートに使用するポートです。 --usevirtualmulticastの設定がされていはいりません

● 例

```
offline>create-domain -domain domain1 -das adminServer -baseport 9763 -ha
230.30.1.1 -hp 3030 -jvmconfig "-Xmx1024m -XX:MaxPermSize=256m"
Enter the password for [administrator]: *****
Confirm the password: *****
Do you want to encrypt the password? (y/n): y
=====
The domain [domain1] was created successfully.
```

Property	Value
Domain name	domain1
Das Name	adminServer
Das Base Listen Port	9736
Das Base Listen Address	0.0.0.0
JMS Internal Port	9741
HTTP Port	8088
User Name	administrator
Node Name	node1
Password	{AES:256}lBljYlIlBrujFDC7v3NSoHZQprgnncVmdBbkt4oxDRg=
JVM Option	-Xmx1024m -XX:MaxPermSize=256m
Production Mode	true

```
| Heart beat address      | 230.30.1.1
|
| Heart beat port        | 3030
|
+-----+-----+
=====
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.4.13. delete-domain

作成したドメインを削除するコマンドです。

- エイリアス

deletedomain

- 使用方法

```
delete-domain -domain <domain-name>
```

- パラメータ

パラメータ	説明
-domain <domain-name>	ドメインの名前です

- 例

```
offline>delete-domain -domain newDomain
Deleting the domain directory was successfully.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.4.14. list-clusters

現在ドメインに存在するクラスターの情報を確認します。

- 使用方法

```
list-clusters [<cluster-name>]
```

- パラメータ

パラメータ	説明
[<cluster-name>]	クラスターの名前です

- 例

```
[DAS]domain1.adminServer>list-clusters
List of Clusters
=====
+-----+-----+-----+-----+-----+
| Cluster | Server List | MEJB | Class FTP | Running |
+-----+-----+-----+-----+-----+
| cluster1 | newServer, server1 |      |           | true     |
|          |                   |      |           |          |
+-----+-----+-----+-----+-----+
=====
```

4.2.4.15. list-custom-resources

現在ドメインに設定されているカスタム・リソースを確認します。

- エイリアス

listcustomresources, list-cr, listcr

- 使用方法

```
list-custom-resources [-name <export-name>] |
                    [-cluster <cluster-name>] |
                    [-server <server-name>]
```

- パラメータ

パラメータ	説明
[-name <export-name>]	カスタム・リソースの名前です
[-cluster <cluster-name>]	クラスターに登録されているカスタム・リソースを確認するためのクラスター名です
[-server <server-name>]	サーバーに登録されているカスタム・リソースを確認するためのサーバー名です

- 例

```
[DAS]domain1.adminServer>list-custom-resources
List of Custom Resources
=====
```

Export Name	Resource Class	Factory Class	Properties
custom/dog	dog.Dog	dog.DogFactory	[test=1, test1=2]

4.2.4.16. list-external-resources

現在ドメインに設定されている外部リソースを確認します。

- エイリアス

listexternalresources, list-er, lister

- 使用方法

```
list-external-resources [-name <export-name>] |
                        [-cluster <cluster-name>] |
                        [-server <server-name>]
```

- パラメータ

パラメータ	説明
[-name <export-name>]	外部リソースの名前です
[-cluster <cluster-name>]	クラスターに登録されている外部リソースを確認するためのクラスター名です
[-server <server-name>]	サーバーに登録されている外部リソースを確認するためのサーバー名です

- 例

```
[DAS]domain1.adminServer>list-external-resources
List of External Resources
=====
+-----+-----+-----+
| Export Name | Resource Class | Properties |
+-----+-----+-----+
| test/ext    | test.ext.TestResourceBootstrapper | [] |
+-----+-----+-----+
=====
```

4.2.4.17. list-domains

現在構成されているドメインの一覧を表示するコマンドです。

- エイリアス

listdomains

- 使用方法

```
list-domains
```

- 例

```
offline>list-domains
List Domains
=====
+-----+
|                                     Domains                                     |
+-----+
| domain1                             |
+-----+
=====
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.4.18. list-servers

現在ドメインに存在するサーバーの設定情報を確認します。状態情報を確認するには[server-info](#)コマンドを使用します。

- 使用方法

```
list-servers [<server-name>]
```

- パラメータ

パラメータ	説明
[<server-name>]	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>list-servers
List of Editable Servers and Server Templates
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Server|  Base  |  Base  | Node|JVM Configs| Action On | Stdout to| MEJB|Class| Server          | Type
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      | Listen | Listen |     |           | Resource  | Row Format|     | FTP | Log Home       |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      | Address| Port   |     |           | Leak      |           |     |     | Directory      |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| admin| 0.0.0.0| 9736   |     | -Xmx256m | Warning   | true     | false| true| none           | server
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|Server|         |         |     | -XX:MaxPerm|           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | Size=128m  |           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | -Xdebug    |           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | -Xrunjdwp:t|           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | transport=dt|           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | _socket,ser|           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | ver=y,suspe|           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | nd=n,addres|           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     | s=5005     |           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| serve| 192.167.| 9836   | node1|           | Warning   | false    | false| false| none           | server
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| r1   | 0.124    |         |     |           |           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| serve| 192.167.| 9936   |     |           | Warning   | false    | false| false| none           | server
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| r2   | 0.124    |         |     |           |           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| templ| none     | none    | none|           | Warning   | true     | false| false| none           | template
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |         |         |     |           |           |           |     |     |               |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
=====
```

4.2.4.19. modify-cluster

クラスターの設定を動的に変更します。

- エイリアス

modifycluster, clusterconf, clusterconfig

- 使用方法


```

modify-cluster <cluster-name>
    [-m,--mejb <enable-MEJB>]
    [-c,--classFtp <enable-class-ftp-service>]
    [-f,--forceLock]
    [-detail]

```

- パラメータ

パラメータ	説明
<cluster-name>	クラスターの名前です
[-m,--mejb <enable-MEJB>]	[動的変更] MEJBを使用するか否かを設定します(true false)
[-c,--classFtp <enable-class-ftp-service>]	[動的変更] class FTPサービスを使用するか否かを選択します(true false)
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```

[DAS]domain1.adminServer>modify-cluster cluster1 -m true
Successfully performed the MODIFY operation for cluster (cluster1).
Check the results using "list-clusters cluster1 or modify-cluster cluster1"

```

4.2.4.20. modify-server

サーバーの設定を動的に変更します。

- エイリアス

serverconf, serverconfig, modify-server-template

- 使用方法

```

modify-server <server-name>
    [-node,--nodeName <node-name>]
    [-jvm,--jvmOptions <jvm-options>]
    [-a,--actionOnResourceLeak <action-on-resource-leak>]
    [-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
    [-m,--mejb <enable-MEJB>]
    [-c,--classFtp <enable-class-ftp-service>]
    [-detail]

```

- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です
[--remove-log-dir,--removeLogHomeDirectory]	サーバーで生成されるログが共通して含まれるログ・ディレクトリーを削除するか否かを設定します
[-node,--nodeName <node-name>]	サーバーが実際に動作するマシンを意味するノードの名前です。 list-nodesコマンドを使用してノードの情報を確認できます
[-jvm,--jvmOptions <jvm-options>]	サーバーに適用しているJVMオプションです。 空白が存在する場合は二重引用符(" ")で囲みます
[-log-dir,--logHomeDirectory <server-log-home-directory>]	サーバーで生成されるログが共通して含まれるログ・ディレクトリーです
[-a,--actionOnResourceLeak <action-on-resource-leak>]	リソース・リークが発生した場合に行うアクションを設定します。 以下のうち1つを入力します – AutoClose – NoAction – Warning
[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]	[動的変更] stdoutをそのまま出力するのか、JEUSログガーのフォーマットで出力するのかを設定します(true false)
[-m,--mejb <enable-MEJB>]	[動的変更] MEJBを使用するか否かを設定します(true false)
[-c,--classFtp <enable-class-ftp-service>]	[動的変更] class FTPサービスを使用するか否かを設定します(true false)
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-server server1 -a AutoClose -l false
Successfully performed the MODIFY operation for server (server1).
Check the results using "list-servers server1 or modify-server server1"
```

4.2.4.21. modify-dynamic-servers

Dynamic Servers設定に基づいて作成されたクラスターの設定情報を変更します

- エイリアス

modifydservers

- 使用方法

```

modify-dynamic-servers <cluster-name>
    [-count <the-number-of-dynamic-servers>]
    [-nodes,--nodeNames <node-names>]
    [-base <base-port>]
    [-http <http-port>]
    [-jms <jms-port>]
    [-prefix,--serverNamePrefix <server-name-prefix>]
    [-template,--serverTemplateName <server-template-name>]
    [-step,--listenPortStep <listen-port-step>]
    [-f,--forceLock]
    [-detail]

```

- パラメータ

パラメータ	説明
<cluster-name>	動的設定サーバーの設定を変更するクラスターの名前を指定します
[-count <the-number-of-dynamic-servers>]	クラスターに追加するサーバーの数を変更します
[-nodes,--nodeNames <node-names>]	サーバーが位置するノードを変更します。 複数のサーバーをコンマ(,)を区切り子に使用して設定できます。すべてのノードを対象にしたい場合は、-nodesオプションの後ろに引数を渡さないようにします
[-base <base-port>]	サーバーの基本動作ポートを変更します
[-http <http-port>]	サーバーのHTTPポートを変更します
[-jms <jms-port>]	サーバーのJMSポートを変更します
[-prefix,--serverNamePrefix <server-name-prefix>]	サーバーの名前の前に付けるプレフィックスを変更します。 プレフィックスが設定されていない場合は、クラスターの名前の後ろに「_server_」が付けられます
[-template,--serverTemplateName <server-template-name>]	動的サーバーに共通して適用される設定テンプレートを指定します
[-step,--listenPortStep <listen-port-step>]	同じノードに作成されたサーバーのポート番号の増分を変更します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-dynamic-servers cluster1 -count 3
    Successfully performed the MODIFY operation for Dynamic servers
configuration in the cluster (cluster1).
    Check the results using "modify-dynamic-servers or
show-dynamic-servers".
```

4.2.4.22. remove-cluster

クラスターを動的に削除します。

- 使用方法

```
remove-cluster [<cluster-name>]
               [-f,--forceLock]
               [-detail]
```

- パラメータ

パラメータ	説明
<cluster-name>	クラスターの名前です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

- 現在存在するクラスターの確認

```
[DAS]domain1.adminServer>list-clusters
List of Clusters
=====
+-----+-----+-----+-----+-----+
| Cluster | Server List | MEJB | Class FTP | Running |
+-----+-----+-----+-----+-----+
| cluster1 | server1, server2 | false | false | true |
+-----+-----+-----+-----+-----+
| cluster2 | server3, server4 | true | true | true |
+-----+-----+-----+-----+-----+
=====
```

- 現在存在するクラスター一覧で削除するクラスターの名前を確認後、クラスターを削除

```
[DAS]domain1.adminServer>remove-cluster cluster2
Successfully performed the REMOVE operation for cluster (cluster1).
Check the results using "list-clusters or remove-cluster"
```

4.2.4.23. remove-custom-resource

カスタム・リソースを動的に削除します。

- エイリアス

removecr, rmcr

- 使用方法

```
remove-custom-resource [<export-name>]
                        [-f,--forceLock]
                        [-detail]
```

- パラメータ

パラメータ	説明
[<export-name>]	削除するカスタム・リソースのJNDIバインドの名前です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-custom-resource custom/dog
Successfully performed the REMOVE operation for A custom resource.
Check the results using "list-custom-resources or remove-custom-resource"
```

4.2.4.24. remove-custom-resource-from-clusters

クラスターに登録されているカスタム・リソースを動的に削除します。

- エイリアス

remove-cr-from-clusters

- 使用方法

```
remove-custom-resource-from-clusters <export-name>
                                     -clusters <cluster-list>
                                     [-f,--forceLock]
                                     [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	クラスターから削除するカスタム・リソースのJNDIバインド名です
-cluster <cluster-list>	カスタム・リソースを削除するクラスター一覧を設定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-custom-resource-from-clusters custom/dog -clusters
cluster1
Successfully performed the REMOVE operation for A custom resource.
Check the results using "list-custom-resources"
```

4.2.4.25. remove-data-sources-from-cluster

クラスターに登録されているデータソースを動的に削除します。

- エイリアス

rmdatafromcluster

- 使用方法

```
remove-data-sources-from-cluster -cluster <cluster-name>
                                [-ids <data-source-id-list> | -all]
                                [-f,--forceLock]
                                [-detail]
```

- パラメータ

パラメータ	説明
-cluster <cluster-name>	データソースを削除するクラスターの名前です
[-ids <data-source-id-list>]	クラスターから削除するデータソースのID一覧です。1つ以上の場合はコンマ(,)で区分します
[-all]	クラスターに登録されているすべてのデータソースを削除します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-data-sources-from-cluster -cluster cluster1 -all
Successfully performed the REMOVE operation for data sources from the cluster
```

```
[cluster1].  
Check the results using "remove-data-sources-from-cluster -cluster cluster1"
```

4.2.4.26. remove-external-resource

外部リソースを動的に削除します。

- エイリアス

removecr, rmcr

- 使用方法

```
remove-external-resource [<export-name>  
                          [-f,--forceLock]  
                          [-detail]
```

- パラメータ

パラメータ	説明
[<export-name>]	削除する外部リソースの名前です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-external-resource test/ext  
Successfully performed the REMOVE operation for A external resource.  
Check the results using "list-external-resources or remove-external-resource"
```

4.2.4.27. remove-external-resource-from-clusters

クラスターに登録されている外部リソースを動的に削除します。

- エイリアス

remove-er-from-clusters

- 使用方法

```
remove-external-resource-from-clusters <export-name>  
                                       -clusters <cluster-list>  
                                       [-f,--forceLock]  
                                       [-detail]
```

- パラメータ

パラメータ	説明
<export-name>	クラスターから削除する外部リソースの名前です
-clusters <cluster-list>	外部リソースを削除するクラスター一覧を設定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-external-resource-from-clusters test/ext -clusters
cluster1
Successfully performed the REMOVE operation for A external resource.
Check the results using "list-external-resources"
```

4.2.4.28. remove-server

サーバーを動的に削除します。

- 使用方法

```
remove-server [<server-name>]
              [-f,--forceLock]
              [-detail]
```

- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

– 現在存在するサーバー一覧の確認

```
[DAS]domain1.adminServer>server-info
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server| Status|Node | PID | Clu | Latest | Need | Listen |Running|
|       |       |Name |     | ster| Start Time | to | Ports |Engines|
|       |       |     |     |     |           |   |       |       |
|       |       |     |     |     |           |   |       |       |
```


						/ Shutdown	Restart		
						Time			
admin	RUNNING	nod	130	N/A	2016-08-26	false	base-0.0.	jms,	
Server	(00:04:2	e1	83		(金) 午前		0.0:9736	ejb,	
web									
(*)	8)				09:20:44 KST		http-serv		
							er-0.0.0.0		
							:8088		
							jms-0.0.0		
							.0:9741		
serve	RUNNING	nod	132	N/A	2016-08-26	false	base-0.0.	jms,	
r1	(00:02:4	e1	75		(金) 午前		0.0:9836	ejb,	
web									
	5)				09:22:28 KST				
serve	SHUTDOWN	nod	N/A	N/A	N/A	N/A	N/A	N/A	N/A
r2		e1							

– 削除するサーバーが停止しているかを確認後、削除

```
[DAS]domain1.adminServer>remove-server server2
Successfully performed the REMOVE operation for server (server1).
Check the results using "list-servers or remove-server"
```

4.2.4.29. remove-servers-from-cluster

クラスターからサーバーを動的に削除します。

- 使用方法

```
remove-servers-from-cluster <cluster-name>
                        [-servers <server-lists>]
                        [-f,--forceLock]
                        [-detail]
```

- パラメータ

パラメータ	説明
<cluster-name>	クラスターの名前です
[-servers <server-lists>]	このクラスターから削除するサーバーの名前です。サーバーが1つ以上の場合はコンマ(,)で区分します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>remove-servers-from-cluster cluster1 -servers server1
Successfully performed the REMOVE operation for The server list for
cluster(cluster1)..
Check the results using "list-clusters cluster1 or remove-servers-from-cluster
cluster1"
```

4.2.4.30. set-domain-backup

DASの起動時に自動でドメインのバックアップ・ファイルを作成する動的オンライン・コマンドです

- エイリアス

domain-backup

- 使用方法

```
set-domain-backup [-backupOnBoot <backup-on-boot>]
                  [-backupDir <backup-directory>]
                  [-nodeName <nodeName>]
                  [-f,--forceLock]
                  [-detail]
```

- パラメータ

パラメータ	説明
[-backupOnBoot <backup-on-boot>]	DASの起動時にバックアップ・ファイルを作成するか否かを設定します (true/false)

パラメータ	説明
[-backupDir <backup-directory>]	DASのバックアップ・ファイルを作成するディレクトリーです。 ディレクトリーがドメイン内に設定される場合、バックアップの対象から除外されます。NodeNameが設定された場合は、設定されたノードのディレクトリーに作成します
[-nodeName <nodeName>]	バックアップ・ファイルが位置するノードを設定します。 この値を設定した場合は、backupDirがこのノード内のディレクトリーになります。設定したノードへの送信に失敗した場合には、ローカル・ノードのdomains homeディレクトリーに作成されます
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的操作結果についての詳細情報を出力します

- 例

```
offline>set-domain-backup -backupOnBoot true
Successfully performed the MODIFY operation for Domain Backup Policy.
Check the results using "set-domain-backup"
```

4.2.4.31. pack-domain

ドメイン・フォルダーを圧縮するコマンドです。

- エイリアス

packdomain

- 使用方法

```
pack-domain <domain-name>
          [-f <file-path>]
```

- パラメータ

パラメータ	説明
<domain-name>	ドメインの名前です
[-f <file-path>]	圧縮ファイルのパスです (デフォルト値: DOMAIN_HOME/<domain-name>_packed.zip)

- 例

```
offline>pack-domain domain1
Packing the domain [domain1] configuration completed successfully at the path
[D
OMAIN_HOME/domain1_packed.zip].
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.4.32. show-dynamic-servers

動的サーバーの作成により生成されたクラスターの設定情報を表示します。

- エイリアス

showdservers

- 使用方法

```
show-dynamic-servers <cluster-name>
```

- パラメータ

パラメータ	説明
<cluster-name>	動的設定サーバーの情報を表示するクラスターの名前です

- 例

```
[DAS]domain1.adminServer>show-dynamic-servers cluster1
Shows the current configuration.
Dynamic servers configuration in the cluster (cluster1)
=====
+-----+-----+
| Server Count          | 2      |
| Nodes                 | node1  |
| Base Listen Port      | 9736   |
| HTTP Listen Port      | 8088   |
| JMS Listen Port       | 9741   |
| Server Template Name  | temp1  |
| Listen Port Step      | 100    |
+-----+-----+
=====
```

4.2.4.33. show-server-template

設定されたサーバー・テンプレートの一覧を照会します。

- エイリアス

show-template

- 使用方法

```
show-server-template [<server-template-name>]
```

- パラメータ

パラメータ	説明
[<server-template-name>]	設定されたテンプレートの情報を照会します

- 例

```
offline>show-server-template temp1
Shows the current configuration.
server template (temp1)
=====
+-----+-----+-----+
| Action On Resource Leak                | WARNING |
| Stdout to Raw Format                    | true    |
| MEJB                                   | false   |
| Class FTP                              | false   |
| Server Log Home Directory              | none    |
+-----+-----+-----+
=====
```

4.2.4.34. unpack-domain

pack-domainコマンドによって圧縮されたファイルを解凍するコマンドです。該当ドメインの設定にDASの動作アドレスと動作ポートが設定されている場合、変更可否を指定できます。

- エイリアス

unpackdomain

- 使用方法

```
unpack-domain -f <file-path> | <domain-name>
               [-delete]
               [-overwrite]
               [-dasurl <das-url>]
               [-nodename <node-name>]
               [-nodeurl <node-url>]
```

- パラメータ

パラメータ	説明
-f <file-path>	圧縮を解凍するファイルの全体パスです (例: DOMAIN_HOME/<domain-name>_packed.zip)
-f <domain-name>	圧縮を解凍するファイルの名前です。 圧縮を解凍するファイルは必ずDOMAIN_HOMEディレクトリー内にある必要があります (DOMAIN_HOME/<domain-name>_packed.zip)
[-delete]	圧縮を解凍する対象ドメインがDOMAIN_HOMEにすでに存在する場合、既存のドメインを削除してから圧縮を解凍します
[-overwrite]	圧縮を解凍する対象ドメインがDOMAIN_HOMEにすでに存在する場合、圧縮を解凍するドメインが既存のドメインを上書きします
[-dasurl <das-url>]	圧縮を解凍するノードで接続するDASのURLです
[-nodename <node-name>]	圧縮を解凍するノードのノード名です
[-nodeurl <node-url>]	圧縮を解凍するノードのURLです

- 例

```
offline>unpack-domain domain1
The Domain Administration Server listener address is already set to
[192.168.34.55]. Do you want to change it? (y/n): y
Enter the Domain Administration Server base listener address: 192.168.34.56
The Domain Administration Server listener port is already set to [9736]. Do you
want to change it? (y/n): n
The Domain Administration Server nodename is already set to [node1]. Do you want
to change it? (y/n): y
Enter the Domain Administration Server nodename: node2
The nodemanager of DAS machine listener address is already set to [192.168.34.56].
Do you want to change it? (y/n): n
The nodemanager of DAS machine listener port is already set to [7730]. Do you
want to change it? (y/n): n
Unpacking the domain [domain1] configuration completed successfully.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.5. スレッド管理関連コマンド

スレッド管理に関連するコマンドの一覧は以下のとおりです。

コマンド	説明
<code>interrupt-thread</code>	サーブレットまたはEJB RMIスレッドに割り込み信号を送信します
<code>modify-service-thread-pool</code>	サーバーの特定サービスの専用スレッドプールを動的に変更します
<code>modify-system-thread-pool</code>	サーバーのシステム・スレッドプールを動的に変更します
<code>print-stack-trace</code>	サーブレット・スレッドまたはEJB RMIスレッドのスタック・トレースを出力します
<code>show-service-thread-pool</code>	システム・スレッドプールで特定サービスに割り当てられたスレッドの情報を表示します
<code>show-system-thread-pool</code>	システム・スレッドプールの情報を表示します
<code>thread-info</code>	要求を処理するサーブレット・スレッドとEJB RMIスレッドの情報を表示します

4.2.5.1. interrupt-thread

サーブレットまたはEJB RMIスレッドに割り込み信号を送信します。スレッドに割り込み信号を送信した結果についての詳細は『*JEUS サーバガイド*』の「3.3. スレッドのモニタリングと制御」を参照してください。

注

この機能は割り込み信号のみ送信するものであるため、該当スレッドが動作を停止することを保証することはできません。

- エイリアス

`interruptthread`, `interrupt`, `intthread`

- 使用方法

```
interrupt-thread -server <server-name>
                 -li <listener-id> | <thread-id>
```

- パラメータ

パラメータ	説明
<code>-server <server-name></code>	サーバーの名前です
<code>-li <listener-id></code>	Webリスナーの名前です。Webリスナーが保有しているスレッド・プールのスレッド全体に割り込み信号を送信します
<code><thread-id></code>	信号を送信するスレッドのIDです

- 例

```
[DAS]domain1.adminServer>interrupt-thread -server server1 50
Sent an interrupt hint signal to the thread [tid=50] on the server server1.
```

4.2.5.2. modify-service-thread-pool

サービスの専用スレッドプールを動的に変更します。現在統合設定を使用しているネーム・サーバー、スケジューラー、トランザクション・スレッドプールのみが、このコマンドで変更可能です。変更する設定を入力していない場合、現在の設定を表示します。

- エイリアス

modify-svctp, modifysvctp, svctpconfig, svctpconf

- 使用方法

```
modify-service-thread-pool -service <service name>
                               <server-name>
                               [-min <min>]
                               [-max <max>]
                               [-k,--keepalivetime <keep-alive-time>]
                               [-q,--queuesize <queue-size>]
                               [-m,--maxstuckthreadtime <max-stuck-thread-time>]
                               [-a,--actiononstuckthread <action-on-stuck-thread>]
                               [-stcp,--stuckthreadcheckperiod <stuck-thread-check-period>]

                               [-f,--forceLock]
                               [-detail]
```

- パラメータ

パラメータ	説明
-service <service name>	システム・スレッドプールを使用するサービスの名前です。 <service name>に以下のうち1つを選択します。 – transaction – scheduler – namingserver
<server-name>	サーバーの名前です
[-min <min>]	[動的変更] スレッドプールの最小スレッド数です
[-max <max>]	[動的変更] スレッドプールの最大スレッド数です

パラメータ	説明
[-k,--keepalivetime <keep-alive-time>]	[動的変更] 最小スレッド数を超えた場合、アイドル状態が入力時間以上続いているスレッドは削除します
[-q,--queuesize <queue-size>]	実行を待機するワーカーの最大数です
[-m,--maxstuckthreadtime <max-stuck-thread-time>]	[動的変更] 入力時間以上スレッドが実行されている場合はスタック状態と判断します
[-a,--actiononstuckthread <action-on-stuck-thread>]	[動的変更] スタック状態の場合の処理方法です。 <action-on-stuck-thread>に以下のうち1つを選択します。 – IgnoreAndReplace – Interrupt – None
[-stcp,--stuckthreadcheckperiod <stuck-thread-check-period>]	[動的変更] スレッドがスタック状態であることを確認する周期を設定します (単位 : ms)
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-service-thread-pool server1 -service transaction
-min 10 -max 20
Successfully performed the MODIFY operation for The transaction thread pool of
the server (server1)., but all changes were non-dynamic. They will be applied
after restarting.
Check the results using "show-service-thread-pool server1 -service transaction
or modify-service-thread-pool server1 -service transaction"
```

4.2.5.3. modify-system-thread-pool

システム・スレッドプールを動的に変更します。システム・スレッドプールを使用しているサービスが割り当てられたスレッド数も調整できます。

- エイリアス

modify-systp, modifysystp, systpconfig, systpconf

- 使用方法

```

modify-system-thread-pool <server-name>
    [-min <min>]
    [-max <max>]
    [-k,--keepalivetime <keep-alive-time>]
    [-q,--queuesize <queue-size>]
    [-m,--maxstuckthreadtime <max-stuck-thread-time>]
    [-a,--actiononstuckthread <action-on-stuck-thread>]
    [-stcp,--stuckthreadcheckperiod <stuck-thread-check-period>]
    [-service <service name>]
    [-r,--reservednum <reserved-number>]
    [-f,--forceLock]
    [-detail]

```

● パラメータ

パラメータ	説明
<server-name>	サーバーの名前です
[-min <min>]	[動的変更] スレッドプールの最小スレッド数です
[-max <max>]	[動的変更] スレッドプールの最大スレッド数です
[-k,--keepalivetime <keep-alive-time>]	[動的変更] 最小スレッド数を超えた場合、アイドル状態が入力時間以上続いているスレッドは削除します
[-q,--queuesize <queue-size>]	実行を待機するワーカーの最大数です
[-m,--maxstuckthreadtime <max-stuck-thread-time>]	[動的変更] 入力時間以上スレッドが実行されている場合はスタック状態と判断します
[-a,--actiononstuckthread <action-on-stuck-thread>]	[動的変更] スタック状態の場合の処理方法です。 <action-on-stuck-thread>に以下のうち1つを選択します。 <ul style="list-style-type: none"> – IgnoreAndReplace – Interrupt – None
[-stcp,--stuckthreadcheckperiod <stuck-thread-check-period>]	[動的変更] スレッドがSTUCK状態であることを確認する周期を設定します (単位 : ms)
[-service <service name>]	システム・スレッドプールを使用するサービスの名前です。 <service name>に以下のうち1つを選択します。 <ul style="list-style-type: none"> – transaction – scheduler

パラメータ	説明
	- namingserver
[-r,--reservednum <reserved-number>]	[動的変更] システム・スレッドプールを使用するサービスに保証されている最小スレッド数です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-system-thread-pool server1 -min 10 -max 60
-service transaction -reservednum 20 -detail
Successfully performed the MODIFY operation for the system thread pool of the
server (server1).
=====
+-----+-----+-----+-----+
| servers.server.{? name == 'server1' }.systemThreadPool | MODIFY| ACTIVATED|
| servers.server.{? name == 'server1' }.tmConfig          | MODIFY| ACTIVATED|
+-----+-----+-----+-----+
=====
Check the results using "modify-system-thread-pool server1 or show-system-thread
-pool server1"
-----details-----
servers.server.{? name == 'server1' }.tmConfig : ACTIVATED
servers.server.{? name == 'server1' }.tmConfig.pooling : ACTIVATED
servers.server.{? name == 'server1' }.tmConfig.pooling.shared : ACTIVATED
servers.server.{? name == 'server1' }.tmConfig.pooling.shared.reservedThreadNum
: ACTIVATED
previous value : 0, edited value : 20, result value : 20

servers.server.{? name == 'server1' }.systemThreadPool : ACTIVATED
servers.server.{? name == 'server1' }.systemThreadPool.min : ACTIVATED
previous value : 0, edited value : 10, result value : 10
servers.server.{? name == 'server1' }.systemThreadPool.max : ACTIVATED
previous value : 100, edited value : 60, result value : 60
```

4.2.5.4. print-stack-trace

サーブレット・スレッドまたはEJB RMIスレッドのスタック・トレースを出力します。

- エイリアス

stacktrace, strace

- 使用方法

```
print-stack-trace -server <server-name>
                  -li <listener-id> | <thread-id>
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-li <listener-id>]	Webリスナーの名前です。Webリスナーが保有しているスレッド・プールのスレッド全体のスタック・トレースを表示します
<thread-id>	スレッドのIDです

- 例

```
[DAS]domain1.adminServer>print-stack-trace -server server1 50
servlet thread [tid=50] Stack trace of http1-1 [server1-50] tid=50
java.lang.Thread.State: WAITING
    at sun.misc.Unsafe.park(Native Method)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:156)
    at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.
await(AbstractQueuedSynchronizer.java:1987)
    at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:399)

    at jeus.util.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1291)
    at
jeus.servlet.engine.WebThreadPoolExecutor.getTask(WebThreadPoolExecutor.java:68)

    at jeus.util.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:1215)
    at jeus.servlet.engine.WebThreadPoolExecutor$WebRequestWorker.
run(WebThreadPoolExecutor.java:332)
    at java.lang.Thread.run(Thread.java:662)
```

4.2.5.5. show-service-thread-pool

サーバーのシステム・スレッドプールの情報を表示します。

- エイリアス

show-svctp, showsvctp

- 使用方法

```
show-service-thread-pool <server-name>
                        -service <service name>
```

- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です
-service <service name>	サービスの名前です。 <service name>に以下のうち1つを選択します。 - transaction - scheduler - namingserver

- 例

```
[DAS]domain1.adminServer>show-service-thread-pool server1 -service transaction

Shows the current configuration.

=====
+-----+-----+
| Min                | 10    |
| Max                | 20    |
| Keep-Alive Time    | 60000 |
| Queue Size         | 4096  |
| Max Stuck Thread Time | 3600000|
| Action On Stuck Thread | NONE  |
| Stuck Thread Check Period | 300000 |
+-----+-----+
=====
```

4.2.5.6. show-system-thread-pool

サーバーのシステム・スレッドプール情報を表示します。

- エイリアス

show-systp, showsystp

- 使用方法

```
show-system-thread-pool <server-name>
                        [-service <service name>]
```

- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です

パラメータ	説明
<code>[-service <service name>]</code>	<p>サービスの名前です。</p> <p><service name>に以下のうち1つを選択します。</p> <ul style="list-style-type: none"> – transaction – scheduler – namingserver

- 例

```
[DAS]domain1.adminServer>show-system-thread-pool server1
Shows the current configuration.
the system thread pool of the server (server1).
=====
+-----+-----+
| Min                                | 10    |
| Max                                | 60    |
| Keep-Alive Time                    | 300000|
| Queue Size                         | 4096  |
| Max Stuck Thread Time              | 0     |
| Action On Stuck Thread             | NONE  |
| Stuck Thread Check Period          | 300000|
| Reserved Threads for the Service transaction | 0     |
+-----+-----+
=====
```

4.2.5.7. thread-info

要求を処理するサブレット・スレッドとEJB RMIスレッドの情報、およびその他のJEUSスレッド・プールを表示します。サブレットではリスナーのスレッド・プールと各作業のスレッド(ワーカー・スレッド)の情報を表示し、EJBではEJBリモート要求を処理するEJB RMIスレッドの情報を出力します。サーバーでは基本的にサーバー・スレッド・プールの情報のみを表示しますが、オプションによってすべてのプール情報を表示することもできます。

- エイリアス

threadinfo, ti

- 使用方法

```
thread-info -server <server-name>
               [-type <thread-type>]
               [-li <listener-id> | -ctx <context-name>]
               [-r,--request | -a,--all]
```

```
[-s <thread-state>]
[-os,--only-stats]
[-st,--stacktrace]
```

● パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-type <thread-type>]	<p><thread-type>に以下のうち1つを入力します。</p> <ul style="list-style-type: none"> - servlet : サーブレット・スレッド - ejb : EJB RMIスレッド - server : サーブレット・スレッド・プール <p>その他にも、-ctxオプションが指定された場合、以下の値を入力します</p> <ul style="list-style-type: none"> - webasync : Servlet 3.0非同期処理用スレッド・プール - websocket : WebSocketコンテナの非同期送信用スレッド・プール
[-li <listener-id>]	リスナーを指定します。サーブレット・スレッドでのみ有効です
[-ctx <context-name>]	Webコンテキストを指定します。サーブレット・スレッドでのみ有効です。Asynchronous Servletを指定した場合、Asynchronous Processingのためのスレッド・プールの情報を表示します
[-r,--request]	<p>サーブレット・スレッドの要求情報のみを確認します。</p> <p>サーブレット・スレッドに対してのみ有効です</p>
[-a,--all]	サーブレットではサーブレット・スレッドのすべての情報を確認し、サーバーではJEUS内部のすべてのプール情報を確認することを意味します
[-s <thread-state>]	<p>特定の状態のスレッドのみを照会したい場合、その状態を入力します</p> <ul style="list-style-type: none"> - active - idle - blocked - reconn

パラメータ	説明
[-os,--only-stats]	サーブレットでは各リスナー別統計情報のみを出力します。サーバーではプールの統計情報のみを出力します
[-st,--stacktrace]	サーバーで照会したプールに属するすべてのアクティブ・スレッド(実際にタスクが付与され実行中のスレッド)のトレースを照会します。サーバーでのみ有効です

- 例

```
[DAS]domain1.adminServer>thread-info -server server1

Thread information for the server [server1]
There are no EJB RMI threads for the server [server1].
=====
The web container threads for 'ADMIN-HTTP' listener.

+-----+-----+-----+-----+-----+
| tid |          name          | state | elapsed | uri |
+-----+-----+-----+-----+-----+
| 45 | ADMIN-HTTP-w00        | waiting | 48521870 |    |
+-----+-----+-----+-----+-----+

elapsed: Elapsed time (ms)
=====

Thread statistics for the 'ADMIN-HTTP' listener.

+-----+-----+-----+-----+-----+-----+
|          | total | active | idle | blocked | reconn |
+-----+-----+-----+-----+-----+-----+
| The number of threads. | 1 | 0 | 1 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+

total = active + idle, reconn: reconnecting
=====

The web container threads for 'http1' listener.

+-----+-----+-----+-----+-----+
| tid |          name          | state | elapsed | uri |
+-----+-----+-----+-----+-----+
| 46 | http1-w00              | waiting | 48521873 |    |
| 47 | http1-w01              | waiting | 48521868 |    |
```



```

| 48 | http1-w02      | waiting| 48521868| |
| 49 | http1-w03      | waiting| 48521868| |
| 51 | http1-w05      | waiting| 48521862| |
| 52 | http1-w06      | waiting| 48521858| |
| 53 | http1-w07      | waiting| 48521858| |
| 54 | http1-w08      | waiting| 48521857| |
| 55 | http1-w09      | waiting| 48521857| |
| 21203| http1-w10     | waiting| 37629 | |
+-----+-----+-----+-----+-----+

elapsed: Elapsed time (ms)
=====

=====
+-----+-----+-----+-----+-----+
|               | total| active| idle| blocked| reconn|
+-----+-----+-----+-----+-----+
| The number of threads. | 10 | 0 | 10 | 0 | 0 |
+-----+-----+-----+-----+-----+

total = active + idle, reconn: reconnecting
=====

=====
The threads for the 'threadpool.System' thread pool.

+-----+-----+-----+-----+-----+
| tid | name                | thread state | active thread |
+-----+-----+-----+-----+-----+
| 93 | threadpool.System-2 | TIMED_WAITING | false         |
| 92 | threadpool.System-1 | TIMED_WAITING | false         |
| 94 | JMXMP [adminServer-94] | RUNNABLE      | true          |
+-----+-----+-----+-----+-----+
=====

=====
The statistics for the 'threadpool.System' thread pool.

+-----+-----+-----+-----+-----+-----+
| pool name | minimum | maximum | current | work | remaining work |
|           |         |         |         |      |                |
|           | pool size | pool size | pool size | queue size | queue size |
|           |         |         |         |         |         |
+-----+-----+-----+-----+-----+-----+
| threadpool| 0 | 100 | 3 | 4096 | 4096 |
|           |         |         |         |         |         |
|.System    |         |         |         |         |         |

```

+-----+-----+-----+-----+-----+-----+
=====

4.2.6. アプリケーション関連コマンド

アプリケーションに関連するコマンドの一覧は以下のとおりです。

コマンド	説明
add-application-repository	新規のアプリケーション・レポジトリを登録します
add-application-target	デプロイされているアプリケーションのサービス対象にサーバーやクラスターを追加します。該当サーバーやクラスターにアプリケーションをデプロイします
application-info	ドメインに存在するアプリケーションの情報を出力します
deploy-application	インストールしたアプリケーションを対象サーバーにデプロイします。 サービス対象に指定されたサーバーのうち1つでもデプロイに失敗した場合、成功したサーバーにもアンデプロイを実行します
deployment-plan-info	インストールされたデプロイメント・プランの情報を確認します
distribute-application	インストールされたアプリケーションをサーバーやクラスターに配布します。 実際にサービスを開始するためには、 start-application コマンドでアプリケーションを起動します
install-application	アプリケーションをインストールします。アプリケーションのIDを定義できます
install-deployment-plan	デプロイメント・プランをインストールします。デプロイメント・プランの識別子を定義できます
list-application-repositories	登録されているアプリケーションのレポジトリ情報を確認します
redploy-application	アプリケーションを再デプロイします
remove-application-repository	登録されているアプリケーション・レポジトリを削除します
remove-application-target	サーバーやクラスターにデプロイまたは配布されているアプリケーションをサーバーやクラスターから削除します
start-application	配布されたアプリケーションのサービスを開始します
stop-application	サービス中のアプリケーションを停止させます
undeploy-application	デプロイされたアプリケーションをアンデプロイします
uninstall-application	インストールされているアプリケーションを削除します

コマンド	説明
uninstall-deployment-plan	インストールされているデプロイメント・プランを削除します
library-info	インストールおよびデプロイしたライブラリー情報を出力します
install-library	ライブラリーをインストールします
deploy-library	インストールしたライブラリーをサーバーまたはクラスターにデプロイします
undeploy-library	デプロイしたライブラリーをアンデプロイします
uninstall-library	インストールされているライブラリーを削除します

4.2.6.1. add-application-repository

新しいアプリケーション・レポジトリを追加します。

- エイリアス

addapprepo, installapps

- 使用方法

```
add-application-repository [<application-repository-path>]
                           [-f,--forceLock]
                           [-detail]
```

- パラメータ

パラメータ	説明
<application-repository-path>	追加するアプリケーション・レポジトリの絶対パスです
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	詳細情報を表示します

- 例

```
[DAS]domain1.adminServer>add-application-repository /home/user1/apps
Successfully performed the ADD operation for An application repository.
Check the results using "add-application-repository or
list-application-repositories"
```

4.2.6.2. add-application-target

デプロイされているアプリケーションのサービス対象にサーバーやクラスターを追加します。該当サーバーやクラスターにアプリケーションをデプロイします。該当アプリケーションがDEPLOYED、DISTRIBUTED、RUNNING状態の場合にのみ使用可能です。

- エイリアス

addapptarget, addtarget

- 使用方法

```
add-application-target <application-id>
                        [-servers <server-list>]
                        [-clusters <cluster-list>]
                        [-vh,--virtualHost <virtual-host>]
```

- パラメータ

パラメータ	説明
<application-id>	サービス対象を追加するアプリケーションのIDです。 該当アプリケーションがDEPLOYED、DISTRIBUTED、RUNNING状態の場合にのみ使用できます
[-servers <server-list>]	アプリケーションを追加するサーバーの名前です。サーバーの名前はコンマ(,)で区分します
[-clusters <cluster-list>]	アプリケーションを追加するクラスターの名前です。クラスターの名前はコンマ(,)で区分します
[-vh,--virtualHost <virtual-host>]	バーチャル・ホストの名前を指定します

- 例

```
[DAS]domain1.adminServer>add-application-target myApp -servers server1
add a target server or cluster to the application for the application
[deployment_helloear] succeeded.
```

4.2.6.3. application-info

ドメインに存在するアプリケーションの情報を出力します。

- エイリアス

appinfo, list-applications

- 使用方法

```
application-info[-server <server-name>]
                [-id <application-id> | -state <application-state>]
                [-type <application-type>]
                [-module <module-name>]
                [-bean <bean-name>]
                [-jndi <jndi-context>]
```

```

[-mod <jndi-module-name>]
[-comp <jndi-component-name>]
[-detail,--detail]
[-sps,--stateperserver]

```

● パラメータ

パラメータ	説明
[-server <server-name>]	情報を出力するサーバーの名前です。該当サーバーのアプリケーション情報を、より詳しく表示します
[-id <application-id>]	情報を出力するアプリケーションのIDです
[-state <application-state>]	アプリケーションの状態を指定して情報を出力します
[-type <application-type>]	アプリケーションのタイプを指定して情報を確認します。 <application-type>に以下のうち1つを設定します。 – EAR – EJB – WAR – CAR – RAR
[-module <module-name>]	EAR内部のモジュール情報を確認します
[-bean <bean-name>]	EJB Beanの詳細情報を出力するために使用します。 server、id、module情報が必要です
[-jndi <jndi-context>]	アプリケーションのJNDI情報を確認します。 server、id情報が必要です
[-mod <jndi-module-name>]	モジュールのJNDI名前空間を確認します
[-comp <jndi-component-name>]	コンポーネントの名前空間を確認します
[-detail,--detail]	スタンドアロン・モジュールの詳細情報を出力します。 Webモジュールの場合、サーブレット、フィルター、リスナー、WebSocket Server Endpoints、EJB Bean(EJB in a .war)のリストが出力されます
[-sps,--stateperserver]	サーバー別に各アプリケーションの状態を出力します。 対象サーバーが停止しているなどの場合は、「NO_STATE」と出力します

- 例

- 基本出力

以下は、基本出力の結果です。

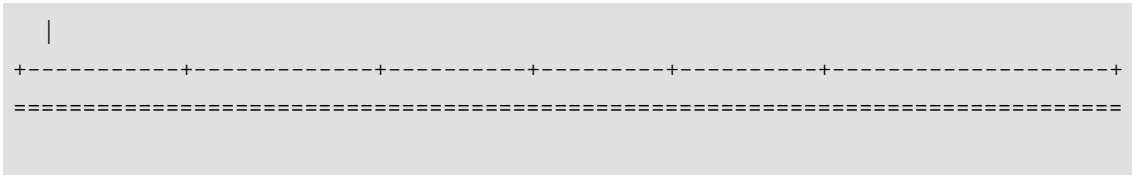
```
[DAS]domain1.adminServer>application-info
Application information for the domain [domain1].
=====
+-----+-----+-----+-----+-----+-----+
| Applicat | Applicati | State | Target | Target | Application Path
|   ID    |   on Type |      | Servers | Clusters |
|         |           |      |         |          |
+-----+-----+-----+-----+-----+-----+
| myApp    | EAR      | RUNNING | adminServer |          | ${INSTALL_HOME}/my
|         |          |         |             |          | App/myApp.ear
|         |          |         |             |          |
+-----+-----+-----+-----+-----+-----+
| myEJB    | EJB      | RUNNING |             | cluster1 | ${INSTALL_HOME}/my
|         |          |         |             |          | EJB/myEJB.jar
|         |          |         |             |          |
+-----+-----+-----+-----+-----+-----+
| myWeb    | WAR      | RUNNING | server1,adm |          | ${INSTALL_HOME}/my
|         |          |         | inServer   |          | Web/myWeb.war
|         |          |         |             |          |
+-----+-----+-----+-----+-----+-----+
=====
[DAS]domain1.adminServer>
```

- stateperserverオプションを使用した場合

サーバー別に各アプリケーションの状態を表示することを確認できます。状態を取得できないサーバーに対しては「NO_STATE」と表示します。

```
[DAS]domain1.adminServer>application-info -sps
Application information for the domain [domain1].
=====
+-----+-----+-----+-----+-----+-----+
| Applicat | Applicati | State | Target | Target | Application Path
|   ID    |   on Type |      | Servers | Clusters |
|         |           |      |         |          |
+-----+-----+-----+-----+-----+-----+
| myApp    | EAR      | RUNNING | adminServer |          | ${INSTALL_HOME}/my
```

						App/myApp.ear
+-----+-----+-----+-----+-----+-----+-----+						
myEJB	EJB	RUNNING		cluster1		\${INSTALL_HOME}/my
						EJB/myEJB.jar
+-----+-----+-----+-----+-----+-----+-----+						
myWeb	WAR	RUNNING	server1,adm			\${INSTALL_HOME}/my
			inServer			Web/myWeb.war
+-----+-----+-----+-----+-----+-----+-----+						
=====						
Server-specific state information for each application in the domain [domain1]						
=====						
+-----+-----+-----+-----+-----+-----+-----+						
Applicati	Application	State	Server	Cluster		Application Path
on ID	Type					
+-----+-----+-----+-----+-----+-----+-----+						
myApp	EAR	RUNNING	adminSe			\${INSTALL_HOME}/my
			rver			App/myApp.ear
+-----+-----+-----+-----+-----+-----+-----+						
myEJB	EJB	RUNNING	server2	cluster1		\${INSTALL_HOME}/my
						EJB/myEJB.jar
+-----+-----+-----+-----+-----+-----+-----+						
myEJB	EJB	NO STATE	server3	cluster1		\${INSTALL_HOME}/my
						EJB/myEJB.jar
+-----+-----+-----+-----+-----+-----+-----+						
myWeb	WAR	NO STATE	server1			\${INSTALL_HOME}/my
						Web/myWeb.war
+-----+-----+-----+-----+-----+-----+-----+						
myWeb	WAR	RUNNING	adminSe			\${INSTALL_HOME}/my
			rver			Web/myWeb.war



4.2.6.4. deploy-application

インストールされたアプリケーションを対象サーバーにデプロイします。サービス対象に指定されているサーバーのうち1つでもデプロイに失敗した場合、成功したサーバーにもアンデプロイを実行します。

- エイリアス

deployapp, deploy

- 使用方法

```
deploy-application [<application-id>]  
    [-path,--applicationPath <path>]  
    [-servers <server-list>]  
    [-clusters <cluster-list>]  
    [-all]  
    [-vh,--virtualHost <virtual-host>]  
    [-type,--applicationType <application-type>]  
    [-cl,--classloading <classloading-mode>]  
    [-auto,--autoRedeployCheckInterval <auto-redeploy-check-interval>]  
  
    [-security,--securityDomainName <security-domain-name>]  
    [-fast,--fastDeploy]  
    [-keep,--keepGenerated]  
    [-shared,--sharedDisk]  
    [-u,--upgrade]  
    [-plan,--deploymentPlanName <deployment-plan-name>]  
    [-ctxp,--contextPath <context-path>]  
    [-staging]  
    [-lib,--libraryId <library-IDs>]  
    [-version,--libraryVersion <library-versions>]  
    [-nj,--nodejava]  
    [-con,--concurrent]
```

- パラメータ

パラメータ	説明
[<i><application-id></i>]	アプリケーションをインストール(install-application)する際に入力したアプリケーションのIDです。1つ以上のアプリケーションをデプロイするには、コンマ(,)でIDを区分します。ファイル・パスにより配布する場

パラメータ	説明
	合も、アプリケーションにIDを付与できます。この場合は、1つのパスと1つのIDのみ使用できます
[-path,--applicationPath <path>]	アプリケーションをインストールせずにファイル・パスを利用する場合に使用します。DASのようなマシンにあるファイルのみ可能です
[-servers <server-list>]	アプリケーションをデプロイするサーバーです。サーバー名はコンマ(,)で区分します
[-clusters <cluster-list>]	アプリケーションをデプロイするクラスターです。クラスター名はコンマ(,)で区分します
[-all]	ドメインに存在するすべてのサーバーにデプロイする場合に指定します
[-vh,--virtualHost <virtual-host>]	バーチャル・ホストのアドレスです。Webエンジンに設定したバーチャル・ホストの名前のうち1つを選択します。Webモジュールでのみ動作します
[-type,--applicationType <application-type>]	<p>アプリケーションのタイプを指定します。</p> <p><application-type>に以下のうち1つを設定します。</p> <ul style="list-style-type: none"> – EAR – EJB – WAR – CAR – RAR <p>オプションを指定していなくても、DDやアノテーションによって自動的に指定されることがあります</p>
[-cl,--classloading <classloading-mode>]	<p>クラス・ローディング形式を指定します。</p> <p><classloading-mode>に以下のうち1つを設定します。</p> <ul style="list-style-type: none"> – SHARED – ISOLATED(デフォルト値)
[-auto,--autoRedeployCheckInterval <auto-redeploy-check-interval>]	<p>アプリケーションを自動的に再デプロイするためのチェック周期です(単位: ms)</p>
[-security,--securityDomainName <security-domain-name>]	<p>アプリケーションに適用されるセキュリティ・ドメインの名前を選択します。デフォルト値にはセキュリティ・ドメインの設定でdefault-application-domainに設定された値が適用されます</p>

パラメータ	説明
[-fast,--fastDeploy]	デプロイ時に作成されるファイルを事前に作成しておいた場合、そのファイルを使用してデプロイ速度を向上させるか否かを設定します
[-keep,--keepGenerated]	デプロイ時に作成されたファイルを維持するか否かを設定します
[-shared,--sharedDisk]	NASを使用する場合に設定します。アプリケーション・ファイルをコピーしないようにするためのオプションです
[-u,--upgrade]	互換性のために、JEUS 8より前のバージョンのDDをJEUS 7バージョンに合う形式に変換するためのオプションです
[-plan,--deploymentPlanName <deployment-plan-name>]	デプロイメント・プラン・ファイルの名前を設定するためのオプションです
[-ctxp,--contextPath <context-path>]	Webアプリケーションの場合、コンテキストを設定するためのオプションです
[-staging]	Exploded module形式のアプリケーションを圧縮して他のマシンに送信するためのオプションです
[-lib,--libraryId <library-IDs>]	アプリケーションが参照するライブラリーを指定します。複数のライブラリーを指定する場合は、コンマ(,)で区切ります
[-version,--libraryVersion <library-versions>]	参照するライブラリーのバージョンを指定します。複数のライブラリーを指定する場合は、コンマ(,)で区切ります
[-nj,--nodejava]	Node JAVAアプリケーションかどうかを設定します
[-con, --concurrent]	複数のアプリケーションをデプロイするとき、アプリケーションを同時にデプロイするか否かを設定します

- 例

```
[DAS]domain1.adminServer>deploy-application myApp -servers server1
deploy the application for the application [myApp] succeeded.
```

4.2.6.5. deployment-plan-info

インストールされたデプロイメント・プランの情報を確認します。

- エイリアス

dpinfo

- 使用方法

```
deployment-plan-info [-name,--deploymentPlanName <deployment-plan-name>]
```

- パラメータ

パラメータ	説明
[-name,--deploymentPlanName <deployment-plan-name>]	詳細情報を確認するデプロイメント・プランの名前(識別子)を指定します

- 例

```
[DAS]domain1.adminServer>deployment-plan-info
The list of deployment plans installed in the domain and the applications to
which each deployment plan applies
=====
+-----+-----+
|          Deployment plan          | Applications |
+-----+-----+
| plan1                             |              |
+-----+-----+
=====
```

4.2.6.6. distribute-application

インストールされたアプリケーションをサーバーやクラスターに配布します。実際にサービスを開始するためには、[start-application](#)コマンドでアプリケーションを起動します。

- エイリアス

distributeapp, distapp, distribute

- 使用方法

```
distribute-application [<application-id>]
    [-path,--applicationPath <path>]
    [-servers <server-list>]
    [-clusters <cluster-list>]
    [-all]
    [-vh,--virtualHost <virtual-host>]
    [-type,--applicationType <application-type>]
    [-cl,--classloading <classloading-mode>]
    [-auto,--autoRedeployCheckInterval <auto-redeploy-check-interval>]

    [-security,--securityDomainName <security-domain-name>]
    [-fast,--fastDeploy]
    [-keep,--keepGenerated]
    [-shared,--sharedDisk]
    [-u,--upgrade]
    [-plan,--deploymentPlanName <deployment-plan-name>]
    [-ctxp,--contextPath <context-path>]
    [-staging]
```

```

[-lib,--libraryId <library-IDs>]
[-version,--libraryVersion <library-versions>]
[-nj,--nodejava]
[-con,--concurrent]

```

● パラメータ

パラメータ	説明
[<application-id>]	アプリケーションをインストール(install-application)する際に入力したアプリケーションのIDです。1つ以上のアプリケーションを配布するには、コンマ(,)でIDを区分します。ファイル・パスにより配布する場合も、アプリケーションにIDを付与できます。この場合は、1つのパスと1つのIDのみ使用できます
[-path,--applicationPath <path>]	アプリケーションをインストールせずにファイル・パスを利用する際に使用します。DASと同じマシンにあるファイルのみ可能です
[-servers <server-list>]	アプリケーションをインストールするサーバーです。サーバー名はコンマ(,)で区分します
[-clusters <cluster-list>]	アプリケーションをインストールするクラスターです。クラスター名はコンマ(,)で区分します
[-all]	ドメインに存在するすべてのサーバーにインストールする際に指定します
[-vh,--virtualHost <virtual-host>]	バーチャル・ホストのアドレスで、Webエンジンに設定したバーチャル・ホストの名前のうち1つを選択します。Webモジュールでのみ動作します
[-type,--applicationType <application-type>]	<p>アプリケーションのタイプを指定します。</p> <p><application-type>に以下のうち1つを設定します。</p> <ul style="list-style-type: none"> – EAR – EJB – WAR – CAR – RAR <p>オプションを指定していなくても、DDやアノテーションによって自動的に指定されることがあります</p>
[-cl,--classloading <classloading-mode>]	<p>クラス・ローディング形式を指定します。</p> <p><classloading-mode>に以下のうち1つを設定します。</p>

パラメータ	説明
	<ul style="list-style-type: none"> – SHARED – ISOLATED (デフォルト値)
<code>[-auto,--autoRedeployCheckInterval <auto-redeploy-check-interval>]</code>	アプリケーションを自動的に再デプロイするためのチェック周期です (単位: ms)
<code>[-security,--securityDomainName <security-domain-name>]</code>	アプリケーションに適用されるセキュリティ・ドメインの名前を選択します。デフォルト値にはセキュリティ・ドメインの設定で default-application-domain に設定された値が適用されます
<code>[-fast,--fastDeploy]</code>	デプロイ時に作成されるファイルを事前に作成しておいた場合、そのファイルを使用してデプロイ速度を向上させるか否かを設定します
<code>[-keep,--keepGenerated]</code>	デプロイ時に作成されたファイルを維持するか否かを設定します
<code>[-shared,--sharedDisk]</code>	NASを使用する場合に設定します。アプリケーション・ファイルをコピーしないようにするためのオプションです
<code>[-u,--upgrade]</code>	互換性のために、JEUS 7より前のバージョンのDDをJEUS 7バージョンに合う形式に変換するためのオプションです
<code>[-plan,-deploymentPlanName <deployment-plan-name>]</code>	デプロイメント・プラン・ファイルの名前を設定するためのオプションです
<code>[-ctxp,--contextPath <context-path>]</code>	Webアプリケーションの場合、コンテキストを設定するためのオプションです
<code>[-staging]</code>	Exploded module形式のアプリケーションを圧縮して、他のマシンに送信するためのオプションです
<code>[-lib,--libraryId <library-IDs>]</code>	アプリケーションが参照するライブラリーを指定します。複数のライブラリーを指定する場合は、コンマ(,)で区切ります
<code>[-version,--libraryVersion <library-versions>]</code>	参照するライブラリーのバージョンを指定します。複数のライブラリーを指定する場合は、コンマ(,)で区切ります
<code>[-nj,--nodejava]</code>	Node JAVAアプリケーションかどうかを設定します
<code>[-con, --concurrent]</code>	複数のアプリケーションを配布するとき、アプリケーションを同時に配布するか否かを設定します

- 例

```
[DAS]domain1.adminServer>distributed-application myApp -servers server1
distributed the application for the application [myApp] succeeded.
```

4.2.6.7. install-application

アプリケーションをインストールします。アプリケーションのIDを定義できます。IDを別途指定していない場合、アプリケーションのファイル名を使用して、終止符(.)はアンダーバー(_)に代替されます。

- エイリアス

installapp, install-app

- 使用方法

```
install-application <application-source-path>
                    [-id,--applicationId <application-id>]
                    [-f,--force]
                    [-u,--upgrade]
```

- パラメータ

パラメータ	説明
<application-source-path>	アプリケーションのパスを指定します
[-id,--applicationId <application-id>]	アプリケーションのIDを指定します
[-f,--force]	同一IDが存在する場合、現在のアプリケーションに上書きするか否かを選択します
[-u,--upgrade]	互換性のために、JEUS 8より前のバージョンのDDをJEUS 7バージョンに合う形式に変換するためのオプションです

- 例

```
[DAS]domain1.adminServer>install-application /home/apps/myApp.ear -id myApp
Successfully installed the application [myApp].
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.6.8. install-deployment-plan

デプロイメント・プランをインストールします。この際、インストールするデプロイメント・プランの識別子を定義できます。

- エイリアス

installdp

- 使用方法

```
install-deployment-plan -path,--deploymentPlanPath <deployment-plan-path>
                        [-name,--deploymentPlanName <deployment-plan-name>]
                        [-f,--force]
```

- パラメータ

パラメータ	説明
-path,--deploymentPlanPath <deployment-plan-path>	インストールするデプロイメント・プランのローカル・パスを指定します
[-name,--deploymentPlanName <deployment-plan-name>]	インストールするデプロイメント・プランのドメイン上での識別子(名前)を定義します
[-f,--force]	同じ名前をもつ既存のデプロイメント・プランを強制的に上書きします

- 例

```
[DAS]domain1.adminServer>install-deployment-plan -path
/home/plans/jeus-deployment-plan.xml -name plan1
Installing the deployment plan [plan1] was successful.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.6.9. list-application-repositories

登録されているアプリケーションのレポジトリ情報を確認します。

- エイリアス

listapprepos

- 使用方法

```
list-application-repositories
```

- 例

```
[DAS]domain1.adminServer>list-application-repositories
Application Repositories
=====
+-----+
|                                     Path of Application Repository                                     |
+-----+
```

```

+-----+
| /home/user1/apps/
|
| //host1/shared/apps
|
+-----+
=====

```

4.2.6.10. redeploy-application

アプリケーションを再デプロイします。

- エイリアス

redapp, redeploy

- 使用方法

```

redeploy-application <application-id>
    [-path,--uploadPath <application-upload-path>]
    [-dasPath,--pathManuallyInstalled
    <application-source-path-in-das>]
    [-to <graceful-timeout>]
    [-f,--force]
    [-distonly,--distributeOnly]
    [-plan,-deploymentPlanName <deployment-plan-name>]
    [-lib,--libraryId <library-IDs>]
    [-version,--libraryVersion <library-versions>]

```

- パラメータ

パラメータ	説明
<application-id>	アプリケーションのIDです。1つ以上のアプリケーションを再デプロイするにはコンマ(,)で区分します
[-path, --uploadPath <application-upload-path>]	変更されたアプリケーションのファイル・パスです。ファイルの再インストールを含みます
[-dasPath, --pathManuallyInstalled <application-source-path-in-das>]	DASマシンでアクセス可能な、変更されたアプリケーションのローカル・ファイル・パスです。インストールする作業を含みません
[-to <graceful-timeout>]	グレースフル・アンデプロイのタイムアウト値です(単位 : 秒)
[-f,--force]	既存のアプリケーションが引き続きサービスされるようにします
[-distonly,--distributeOnly]	新しいアプリケーションをサービスせずに、各サーバーへのインストールまでのみ行います

パラメータ	説明
[-plan,-deploymentPlanName <deployment-plan-name>]	デプロイメント・プラン・ファイルの名前を設定するためのオプションです
[-lib,--libraryId <library-IDs>]	アプリケーションが参照するライブラリーを指定します。複数のライブラリーを指定する場合は、コンマ(,)で区切ります
[-version,--libraryVersion <library-versions>]	参照するライブラリーのバージョンを指定します。複数のライブラリーを指定する場合は、コンマ(,)で区切ります

- 例

```
[DAS]domain1.adminServer>redploy-application myApp
redploy application on das for the application [myApp] succeeded.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.6.11. remove-application-repository

登録されているアプリケーション・レポジトリを削除します。

- エイリアス

rmapprepo, removeapprepo, uninstallapps

- 使用方法

```
remove-application-repository [<application-repository-path>]
                               [-f,--forceLock]
                               [-detail]
```

- パラメータ

パラメータ	説明
[<application-repository-path>]	削除するアプリケーション・レポジトリのパスです
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	詳細情報を表示する際に使用します

- 例

```
[DAS]domain1.adminServer>remove-application-repository /home/user1/apps
Successfully performed the REMOVE operation for An application repository.
```

```
Check the results using "remove-application-repository or  
list-application-repositories"
```

4.2.6.12. remove-application-target

サーバーやクラスターにデプロイまたは配布されたアプリケーションをサーバーやクラスターから削除します。

- エイリアス

remove-app-target, remove-target, rm-app-target, rmapptarget

- 使用方法

```
remove-application-target <application-id>  
                        [-servers <server-list>]  
                        [-clusters <cluster-list>]  
                        [-vh,--virtualHost <virtual-host>]
```

- パラメータ

パラメータ	説明
<application-id>	削除するアプリケーションのIDです
[-servers <server-lists>]	アプリケーションを削除するサーバーの名前を指定します。サーバー名はコンマ(,)で区分します
[-clusters <cluster-lists>]	アプリケーションを削除するクラスターの名前を指定します。クラスター名はコンマ(,)で区分します
[-vh,--virtualHost <virtual-host>]	バーチャル・ホストの名前を指定します

- 例

```
[DAS]domain1.adminServer>remove-application-target myApp -servers server1  
remove server or cluster target from the application for the application [myApp]  
succeeded.
```

4.2.6.13. start-application

配布されたアプリケーションのサービスを開始します。

- エイリアス

startapp, start-app

- 使用方法

```
start-application <application-id>
                [-con,--concurrent]
```

- パラメータ

パラメータ	説明
<application-id>	サービスを開始するアプリケーションのIDです。1つ以上のアプリケーションを開始するにはコンマ(,)で区分します
[-con, --concurrent]	複数のアプリケーションを開始するときに、アプリケーションを同時に開始するか否かを設定します

- 例

```
[DAS]domain1.adminServer>start-application myApp
start the application for the application [myApp] succeeded.
```

4.2.6.14. stop-application

サービス中のアプリケーションを停止させます。

- エイリアス

stopapp, stop-app

- 使用方法

```
stop-application <application-id>
                [-con,--concurrent]
```

- パラメータ

パラメータ	説明
<application-id>	サービスを停止させるアプリケーションのIDです。1つ以上のアプリケーションを停止するにはコンマ(,)で区分します
[-con, --concurrent]	複数のアプリケーションを停止するときに、アプリケーションを同時に停止するか否かを設定します

- 例

```
[DAS]domain1.adminServer>stop-application myApp
stop the application for the application [myApp] succeeded.
```

4.2.6.15. undeploy-application

デプロイされたアプリケーションをアンデプロイします。

- エイリアス

undeployapp, undeploy

- 使用方法

```
undeploy-application <application-id>
                        [-to,--gracefultimeout <graceful-undeploy-timeout>]
                        [-f,--force <non-graceful undeploy>]
                        [-new | -old | -all]
                        [-con,--concurrent]
```

- パラメータ

パラメータ	説明
<application-id>	アプリケーションのIDです。1つ以上のアプリケーションをアンデプロイするためにはコンマ(,)で区分します
[-to,--gracefultimeout <graceful-undeploy-timeout>]	グレースフル・アンデプロイを行う際に適用されるタイムアウト値です。入力した時間の分だけ処理中の要求を待機します(単位: 秒)
[-f,--force <non-graceful undeploy>]	アンデプロイ時にグレースフル・アンデプロイを実行しません。処理中の要求は無視され、アプリケーションはアンデプロイされます
[-new]	グレースフル再デプロイを行う際に新しいアプリケーションをアンデプロイし、既存のアプリケーションをサービスします
[-old]	グレースフル再デプロイを行う際に既存のアプリケーションをアンデプロイします
[-all]	グレースフル再デプロイを行う際に新しいアプリケーションと既存のアプリケーションを両方アンデプロイします
[-con, --concurrent]	複数のアプリケーションをアンデプロイするときに、アプリケーションを同時にアンデプロイするか否かを設定します

- 例

```
[DAS]domain1.adminServer>undeploy-application myApp
Undeploying [myApp] (This may take time due to graceful undeployment) .....
undeploy the application for the application [myApp] succeeded.
Successfully undeployed (elapsed = 2822ms)
```

4.2.6.16. uninstall-application

インストールされているアプリケーションを削除します。

- エイリアス

uninstallapp, uninstall

- 使用方法

```
uninstall-application <application-id> | [-all]
```

- パラメータ

パラメータ	説明
<application-id>	アプリケーションのIDです。1つ以上のアプリケーションを削除するためにはコンマ(,)で区分します
[-all]	インストールされているすべてのアプリケーションを削除します

- 例

```
[DAS]domain1.adminServer>uninstall-application myApp
uninstall the application for the application [myApp] succeeded. : Successfully
deleted [myApp].
```

4.2.6.17. uninstall-deployment-plan

インストールされているデプロイメント・プランを削除します。

- エイリアス

uninstalldp

- 使用方法

```
uninstall-deployment-plan <deployment-plan-name> | [-all]
```

- パラメータ

パラメータ	説明
<deployment-plan-name>	削除するデプロイメント・プランの識別子(名前)を指定します
[-all]	インストールされているすべてのデプロイメント・プランを削除します

- 例

```
[DAS]domain1.adminServer>uninstall-deployment-plan plan1
Uninstalling the deployment plan was successful.
```

4.2.6.18. library-info

現在インストールおよびデプロイされているライブラリーの一覧を出力します。

- エイリアス

libinfo, list-libraries

- 使用方法

```
library-info
```

- 例

```
[DAS]domain1.adminServer>library-info
Library information
=====
+-----+-----+-----+-----+-----+-----+
| Library ID| Version| State | Target Servers| Target Clusters | Applications|
+-----+-----+-----+-----+-----+-----+
| log4j      | 1.2.17 | RUNNING| adminServer   |                  |              |
|            |        |        |               |                  |              |
+-----+-----+-----+-----+-----+-----+
=====
```

4.2.6.19. install-library

ライブラリー・ファイルをインストールします。

- エイリアス

installlib, install-lib

- 使用方法

```
install-library <library-id>
                -path <library-source-path>
                [-version,--libraryVersion <version>]
```

- パラメータ

パラメータ	説明
<library-id>	インストールするライブラリーの識別子を指定します

パラメータ	説明
<code>-path <library-source-path></code>	ライブラリー・ファイルが存在するパスを指定します
<code>[-version,--libraryVersion <version>]</code>	インストールするライブラリーのバージョンを指定します。指定しない場合は、1.0と見なします

- 例

```
[DAS]domain1.adminServer>install-library log4j -path
/home/lib/apache-log4j-1.2.17/log4j-1.2.17.jar -version 1.2.17
Successfully installed the library [log4j] version [1.2.17].
```

4.2.6.20. deploy-library

インストールされているライブラリーをサーバーまたはクラスターにデプロイします。

- エイリアス

deploylib

- 使用方法

```
deploy-library <library-id>
    [-version,--libraryVersion <version>]
    [-all]
    [-clusters <cluster-list>]
    [-servers <server-list>]
```

- パラメータ

パラメータ	説明
<code><library-id></code>	デプロイするライブラリーの識別子を指定します
<code>[-version,--libraryVersion <version>]</code>	インストールするライブラリーのバージョンを指定します
<code>[-all]</code>	すべてのサーバーを対象にデプロイする場合に指定します
<code>[-clusters <cluster-list>]</code>	デプロイ対象になるクラスターを指定します。複数のクラスターを指定する場合は、コンマ(,)で区切ります
<code>[-servers <server-list>]</code>	ライブラリーをデプロイするサーバーを指定します。複数のサーバーを指定する場合は、コンマ(,)で区切ります

- 例

```
[DAS]domain1.adminServer>deploy-library log4j -version 1.2.17 -servers adminServer
deploy the library [log4j] succeeded.
```

4.2.6.21. undeploy-library

サーバーまたはクラスターにデプロイしたライブラリーをアンデプロイします。

- エイリアス

undeploylib

- 使用方法

```
undeploy-library <library-id>
                    [-version,--libraryVersion <version>]
```

- パラメータ

パラメータ	説明
<library-id>	アンデプロイするライブラリーの識別子を指定します
[-version,--libraryVersion <version>]	アンデプロイするライブラリーのバージョンを指定します

- 例

```
[DAS]domain1.adminServer>undeploy-library log4j -version 1.2.17
undeploy the library [log4j] succeeded.
```

4.2.6.22. uninstall-library

ライブラリーを削除します。

- エイリアス

uninstalllib

- 使用方法

```
uninstall-library <library-id>
                   [-version,--libraryVersion <version>]
```

- パラメータ

パラメータ	説明
<library-id>	削除するライブラリーの識別子を指定します
[-version,--libraryVersion <version>]	削除するライブラリーのバージョンを指定します

- 例

```
[DAS]domain1.adminServer>uninstall-library log4j -version 1.2.17
uninstall the library [log4j] succeeded. : Successfully deleted [log4j].
```

4.2.7. EJBエンジン関連コマンド

EJBエンジンに関連するコマンドの一覧は以下のとおりです。

コマンド	説明
cancel-ejb-timer	サーバーのEJBタイマーをキャンセルします
ejb-timer-info	サーバーのEJBタイマー情報を確認します
modify-active-management	サーバーのActive managementを変更します
modify-check-resolution	サーバーのResolution情報を変更します
show-active-management	サーバーのActive management情報を確認します
show-check-resolution	サーバーのResolution情報を確認します

4.2.7.1. cancel-ejb-timer

サーバーのEJBタイマーをキャンセルします。

- エイリアス

cancelejbtimer, canceltimer

- 使用方法

```
cancel-ejb-timer -server <server-name>
                  -module <module-id> | -timer <timer-id>
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-module <module-id>	タイマーをキャンセルするモジュールのIDです
-timer <timer-id>	キャンセルするタイマーのIDです

- 例

```
[DAS]domain1.adminServer>cancel-ejb-timer -server server1 -timer 502
Successfully canceled.
```

4.2.7.2. ejb-timer-info

サーバーのEJBタイマー情報を確認します。

- エイリアス

ejbtimerinfo, timerinfo, scheduler

- 使用方法

```
ejb-timer-info -server <server-name>
                [-module <module-id> | -all]
                [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-module <module-id>]	タイマー情報を確認するEJBモジュールの名前です
[-all]	すべてのEJBモジュールのタイマー情報を確認します
[-detail]	タイマーの詳細情報を表示します

- 例

```
[DAS]domain1.adminServer>ejb-timer-info -server server1 -all -detail
Persistent Timer List: server[server1]
Module ID: Timer
=====
+---+-----+-----+-----+-----+-----+
| ID | BEAN | METHOD | SCHEDULE | INFO | NEXT |
+---+-----+-----+-----+-----+-----+
| 52 | Persistence | print() | sec=*,min=*,hour=* | | 2016-08-29 (Mon) PM
| | TestBean | | ,dayOfMonth=*,dayOf | | 03:50:34 KST
| | | | Week=*,month=*,year | |
| | | | =* | |
+---+-----+-----+-----+-----+-----+
=====
```

4.2.7.3. modify-active-management

サーバーのActive managementを変更します。

- エイリアス

modifyam

- 使用方法

```
modify-active-management -server <server-name>
                        [-mbt,--maxBlockedThread <max-blocked-thread>]
                        [-mit,--maxIdleTime <max-idle-time>]
                        [-f,--forceLock]
                        [-detail]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-mbt,--maxBlockedThread <max-blocked-thread>]	[動的変更] ブロックされたスレッド数の最大値です
[-mit,--maxIdleTime <max-idle-time>]	[動的変更] ブロックされたと判断するアイドル時間の最大値です
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果の詳細情報を出力します

- 例

```
[DAS]domain1.adminServer>modify-active-management -server server1 -mbt 5 -mit
300001
Successfully performed the MODIFY operation for active-management for the server
(server1)..
Check the results using "show-active-management -server server1"
```

4.2.7.4. modify-check-resolution

サーバーのResolutionを変更します。

- エイリアス

modifyresolution, ejbengineresolution

- 使用方法

```

modify-check-resolution -server <server-name>
                        [-r,--resolution <resolution>]
                        [-f,--forceLock]
                        [-detail]

```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-r,--resolution <resolution>]	[動的変更] 変更するEJBエンジンのResolutionです
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	動的変更結果についての詳細情報を出力します

- 例

```

[DAS]domain1.adminServer>modify-check-resolution -server server1 -r 3000001
Successfully performed the MODIFY operation for The EJB engine resolution of th

server (server1)..
Check the results using "show-check-resolution -server server1"

```

4.2.7.5. show-active-management

サーバーのActive management情報を確認します。

- エイリアス

showam

- 使用方法

```

show-active-management -server <server-name>

```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です

- 例

```

[DAS]domain1.adminServer>show-active-management -server server1
Shows the current configuration.
active-management for the server (server1).
=====

```

```

+-----+-----+
| Max Blocked Thread          | 5      |
| Max Idle Time                | 300001 |
+-----+-----+
=====

```

4.2.7.6. show-check-resolution

サーバーのResolution情報を確認します。

- エイリアス

showresolution

- 使用方法

```
show-check-resolution -server <server-name>
```

- パラメータ

パラメータ	説明
-server<server-name>	サーバーの名前です

- 例

```

[DAS]domain1.adminServer>show-check-resolution -server server1
Shows the current configuration.
The EJB engine resolution of the server (server1).
=====
+-----+-----+
| Resolution          | 3000001 |
+-----+-----+
=====

```

4.2.8. Webエンジン関連コマンド

Webエンジンに関連するコマンドは、大きく以下の2つに分けられます。

- モニタリングおよび制御

デプロイされているWebコンテキストおよびWebエンジンの状態を確認あるいは制御するためのコマンドです。

コマンド	説明
resume-web-component	一時停止中のWebコンポーネント(サーブレット)を再開します
suspend-web-component	Webコンポーネント(サーブレット)を一時停止します
reload-web-context	コンポーネントを動的にリロードします
show-web-engine-configuration	Webエンジンに関連する設定情報を表示します
show-web-statistics	Webエンジンに関連する各種統計データを表示します
clear-web-statistics	コンテキスト、サーブレットなどの要求統計データを初期化します
show-request-processing-flow	URLパターンや特定ホスト名で入ってきた要求の処理フローを表示します
precompile-jsp	デプロイされているWebコンテキストのJSPファイルを要求処理前にコンパイルします

- 設定変更関連コマンド

Webエンジンに属する設定を動的に追加、削除、修正できるコマンドです。設定変更事項はXMLには適用しますが、実際に運用中のサービス(メモリ領域)には適用しません。たとえば、バーチャル・ホストの場合は追加および削除が可能ですが、XMLのみ変更されます。そのため、実際にサービスに適用するにはサーバーを再起動する必要があります。

コマンド	説明
add-web-cookie-policy	Cookieポリシーの設定を追加します
modify-web-cookie-policy	Cookieポリシーの設定を変更します
remove-web-cookie-policy	Cookieポリシーの設定を削除します
add-web-encoding	文字セットのエンコーディング設定を追加します
modify-web-encoding	文字セットのエンコーディング設定を変更します
remove-web-encoding	文字セットのエンコーディング設定を削除します
add-web-properties	Webエンジンのプロパティ設定を追加します
modify-web-properties	Webエンジンのプロパティ設定を変更します
remove-web-properties	Webエンジンのプロパティ設定を削除します
add-response-header	HTTPのレスポンス・ヘッダーの設定を追加します
modify-response-header	HTTPのレスポンス・ヘッダーの設定を変更します
remove-response-header	HTTPのレスポンス・ヘッダーの設定を削除します
modify-jsp-engine	JSPエンジンの設定を変更します
add-virtual-host	バーチャル・ホストを追加します
modify-virtual-host	バーチャル・ホストの設定の一部を変更します
remove-virtual-host	バーチャル・ホストを削除します

コマンド	説明
add-webtob-connector	WebtoBコネクタを追加します
modify-webtob-connector	WebtoBコネクタの設定の一部を変更します
remove-webtob-connector	WebtoBコネクタを削除します
add-tmax-connector	Tmax Connectorを追加します
modify-tmax-connector	Tmax Connectorの設定の一部を変更します
remove-tmax-connector	Tmax Connectorを削除します
add-web-listener	HTTP、TCP、AJPリスナーを追加します
modify-web-listener	HTTP、TCP、AJPリスナーの設定の一部を変更します
remove-web-listener	HTTP、TCP、AJPリスナーを削除します
modify-web-engine-configuration	Webエンジンの設定の一部を動的に変更します

4.2.8.1. 共通適用事項

以下は、Webエンジン関連コマンドで共通で適用されるオプションについての説明です。以下のオプションのうち1つのみ設定できます。

- [-cluster <cluster_name>]オプション
 - 各コマンドが適用されるクラスターを指定するためのオプションです。オプションを設定していない場合、現在接続しているサーバーのWebエンジンに適用されます。
 - このオプションは、各コマンドのオプション説明では省略されます。ただし、precompile-jspはこのオプションを提供しません。
- [-server <server_name>]オプション
 - 各コマンドが適用されるサーバーを指定するためのオプションです。オプションを設定していない場合、現在接続しているサーバーのWebエンジンに適用されます。
 - このオプションは、各コマンドのオプション説明では省略されます。

参考

1. サーバーに即時反映されない設定の場合、つまりペンディング状態の設定を最初の設定に再設定するコマンドを実行すると、サーバーの再起動が必要というメッセージ(Restart the Server)が出力されません。
 2. DASの場合、上記のいずれかのオプションを指定する必要があります。しかし、DASではない一般管理サーバーで該当コマンドを実行するとき、いずれのオプションも指定しない場合は、接続しているサーバーに対してコマンドが実行されます。
-

4.2.8.2. add-web-cookie-policy

WebエンジンのHTTP Cookieポリシーの設定を追加します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/cookie-policy

- エイリアス

addcookie

- 使用方法

```
add-web-cookie-policy [-cluster <cluster-name> | -server <server-name>]
                        [-f, -forceLock]
                        [-auer,--apply-url-encoding-rule <apply-url-encoding-rule>]
                        [-enc,--charset-encoding <charset-encoding>]
                        [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f, --forceLock]	設定を強制的に変更して反映します
[-auer,--apply-url-encoding-rule <apply-url-encoding-rule>]	URLエンコーディングの規則の適用可否を設定します
[-enc,--charset-encoding <charset-encoding>]	URLエンコーディングの規則を適用する際に使用する文字セットのエンコーディングです。設定していない場合はRequest Encodingの値に従います
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
add-web-cookie-policy -server server1 -auer true

add-web-cookie-policy -server server1 -enc EUC-KR
```

4.2.8.3. add-web-encoding

Webエンジンのエンコーディングを追加します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。(すでに設定されていたエンコーディングを削除してから追加し、エンコーディングが変更された場合、サーバーを再起動しなければ適用されません)

- 関連スキーマ

web-engine.xsd - web-engine/encoding

- エイリアス

addenc

- 使用方法

```
add-web-encoding [-cluster <cluster-name> | -server <server-name>]
                  [-f, -forceLock]
                  <charset encoding>
                  -co | -default | -forced
                  [-url]
                  [-req]
                  [-res]
                  [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<charset encoding>	設定する文字セットのエンコーディング名です。入力された値が正しくない場合はエラーが出力され、命令が失敗します
-co -default -forced	入力されたエンコーディングをdefaultに設定するか、あるいはforcedに設定するかを決定します。リクエスト・エンコーディングである場合は、この他にclient-overrideも設定できます
[-url]	リクエストURLの文字セットのエンコーディングを設定します
[-req]	リクエストの文字セットのエンコーディングを設定します
[-res]	レスポンスの文字セットのエンコーディングを設定します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
add-web-encoding -server server1 -default -url UTF-8

addenc -server server1 UTF-8 -forced -req -res
```

4.2.8.4. add-web-properties

Webエンジンのプロパティ設定を追加します。追加されるプロパティは大/小文字を区別します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/properties

- alias

addwebpr

- 使用方法

```
add-web-properties [-cluster <cluster-name> | -server <server-name>]
                  [-f, -forceLock]
                  [-p,--properties <properties>]
                  [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-p,--properties <properties>]	追加するWebエンジンのプロパティを設定します。 1つのプロパティは「key=value」形式で、「=」を利用して設定します。 1つ以上のプロパティを追加する場合は各プロパティをコンマ(,)で区分して設定します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
add-web-properties -server server1 -p jeus.servlet.jsp.modern=false

add-web-properties -server server1 -p
jeus.servlet.jsp.compile-java-source-concurrently1=false,
jeus.servlet.jsp.assure-utf8-file-encoding-detection=true
```

4.2.8.5. add-response-header

Webエンジンのレスポンス・ヘッダーの設定を追加します。追加されるレスポンス・ヘッダーは大/小文字を区別しません。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/response-header

- エイリアス

addwebrh

- 使用方法

```
add-response-header [-cluster <cluster-name> | -server <server-name>]
                    [-f, --forceLock]
                    [-n, --name <header-name>]
                    [-v, --value <header-value>]
```

- パラメータ

パラメータ	説明
[-f, --forceLock]	設定を強制的に変更して反映します
[-n, --name <header-name>]	追加するHTTPのレスポンス・ヘッダーの名前を設定します。入力値は大/小文字を区別しません。同じ名前のヘッダーは1つのみ許容されます
[-v, --value <header-value>]	追加するHTTPのレスポンス・ヘッダーの値を設定します。入力値は大/小文字を区別しません

- 例

```
add-response-header -server server1 -n testHeader -v testValue
```

4.2.8.6. add-tmax-connector

Tmaxコネクタを追加します。ただし、XMLにのみ反映し、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/tmax-connector

- エイリアス

addtmaxcon

- 使用方法

```
add-tmax-connector [-cluster <cluster-name> | -server <server-name>]
                   [-f, --forceLock]
                   -name <web-connection-name>
                   -tmin <minimum-thread-num>
                   [-tmax <maximum-thread-num>]
                   -addr <server-address>
                   -port <server-port>
                   -svrg <server-group-name>
```

```
-svr <server-name>
-dcc <dispatcher-config-class>
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
-name <web-connection-name>	Tmaxコネクタの名前です
-tmin <minimum-thread-num>	スレッド・プールの最小スレッド数です
[-tmax <maximum-thread-num>]	スレッド・プールの最大スレッド数です
-addr <server-address>	Tmaxサーバーのアドレスです
-port <server-port>	Tmaxサーバーのポート番号です
-svrg <server-group-name>	Tmaxが属するサーバー・グループの名前です
-svr <server-name>	Tmaxサーバーの名前です
-dcc <dispatcher-config-class>	jeus.servlet.tcp.TCPDispatcherConfigインターフェースを実装したクラスの名前です

- 例

```
add-tmax-connector -name tmax1 -tmin 20 -tmax 20 -addr 5.0.1.2 -port 1024
-svrg group1 -svr server1 -server server1 -dcc serice.DispatcherConfig
```

4.2.8.7. add-virtual-host

バーチャル・ホストを追加します。ただし、XMLにのみ反映し、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/virtual-host

- エイリアス

addvh

- 使用方法

```
add-virtual-host [-cluster <cluster-name> | -server <server-name>]
                  [-f, --forceLock]
                  <virtual-host-name>
                  -list <host-name-list>
```

- パラメータ

パラメータ	説明
[-f, --forceLock]	設定を強制的に変更して反映します
<virtual-host-name>	バーチャル・ホストの名前です
-list <host-name-list>	バーチャル・ホストに登録するドメイン名またはIPアドレスです

- 例

```
add-virtual-host -server server1 host2 -list www.foo.com,192.168.1.2
```

4.2.8.8. add-web-listener

HTTP、TCP、AJPリスナーを追加します。ただし、TCPリスナーはXMLにのみ反映し、TCPリスナーを実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/http-listener, ajp13-listener, tcp-listener

- エイリアス

addwebl

- 使用方法

```
add-web-listener [-cluster <cluster-name> | -server <server-name>]
                 [-f, --forceLock]
                 -name <web-connection-name>
                 -tmin <minimum-thread-num>
                 [-tmax <maximum-thread-num>]
                 [-ajp | -http | -tcp]
                 -slref <server-listener-ref-name>
                 [-dcc <dispatcher-config-class>]
                 [-http2]
                 [-tauto]
                 [-tlimit <thread-num-limit>]
```

- パラメータ

パラメータ	説明
[-f, --forceLock]	設定を強制的に変更して反映します
-name <web-connection-name>	リスナーの名前です
-tmin <minimum-thread-num>	スレッド・プールの最小スレッド数です

パラメータ	説明
[-tmax <maximum-thread-num>]	スレッド・プールの最大スレッド数です
[-ajp -http -tcp]	リスナータイプです。入力していない場合は-httpと仮定します
-slref <server-listener-ref-name>	サーバーに登録されているサービス・リスナーの名前です
[-dcc <dispatcher-config-class>]	jeus.servlet.tcp.TCPDispatcherConfigインターフェースを実装したクラスの名前です。TCPリスナーの場合にのみ必ず必要です
[-http2]	HTTPリスナーがHTTP/2を使用します
[-tauto]	自動チューニングを使用します。このオプションを使用すると、min、max設定を行っても適用されません。HTTPリスナーにのみ適用されます
[-tlimit <thread-num-limit>]	自動チューニングを使用する場合に、スレッド数の制限を設定します

- 例

```
add-web-listener -name http1 -tmin 10 -tmax 20 -server server1 -slref http-server
-http2

add-web-listener -name ajp -ajp -tmin 10 -tmax 20 -server server1 -slref ajp

add-web-listener -name tcp1 -tcp -tmin 10 -tmax 20 -slref tcp -server server1
-dcc tcp.DispatcherConfigImpl
```

4.2.8.9. add-webtob-connector

WebtoBコネクタを追加します。追加したWebtoBコネクタは、実際にサービスで使用可能です。ただし、追加後にWebtoB接続アドレス、ポートなどを変更することはできないため、ご注意ください。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/webtob-connector

- エイリアス

addwebtobcon

- 使用方法

```
add-webtob-connector [-cluster <cluster-name> | -server <server-name>]
[-f, --forceLock]
-name <web-connection-name>
-num <thread-number>
[-ver <<wjp-version>>]
[-addr <server-address>]
-port <server-port> | -dsocket
```

```

-regid <registration-id>
[-wbhome <webtob-home>]
[-ipcport <ipc-base-port>]
[-sndbuf <send-buffer-size>]
[-rcvbuf <receive-buffer-size>]

```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
-name <web-connection-name>	WebtoBコネクタの名前です
-num <thread-number>	スレッド・プールのスレッド数です
-ver <wjp-version>	WebtoBと通信時に使用するプロトコル・バージョンです。1または2を入力します
-addr <server-address>	WebtoBサーバーのアドレスです。-portの使用時には必須オプションです
-port <server-port>	WebtoBサーバーのポート番号です
-regid <registration-id>	WebtoBとの登録IDです
-dsocket	UNIXドメイン・ソケット(Pipe)を利用してJEUSとWebtoB間の通信を行います
-wbhome <webtob-home>	JEUSとWebtoBが同じデバイス内にインストールされている場合、WebtoBがインストールされた絶対パスです
-ipcport <ipc-base-port>	Windows環境でWebtoBの内部プロセス間の通信を行う際に使用するデフォルトのポート番号です
-sndbuf <send-buffer-size>	TCPソケットやUNIXドメイン・ソケットのSO_SNDBUFを設定します。0の場合はOSのデフォルト値を使用します
-rcvbuf <receive-buffer-size>	TCPソケットやUNIXドメイン・ソケットのSO_RCVBUFを設定します。0の場合はOSのデフォルト値を使用します

- 例

```

add-webtob-connector -server server1 -name webtob1 -num 10 -regid MyGroup -port
9999 -addr localhost

```

4.2.8.10. clear-web-statistics

指定したエンティティ(コンテキスト、サブレット)の要求統計データを初期化し、カウンター記録を再設定します。

- 関連スキーマ

web-engine.xsd

- エイリアス

clrstat, clst, clrwebstat

- 使用方法

```
clear-web-statistics [-server <server-name>]
                    [-ctx,--context <context-name>]
                    [-servlet <servlet-name>]
```

- パラメータ

パラメータ	説明
[-ctx <context-name>]	要求回数および処理時間の記録を削除するコンテキストです
[-servlet <servlet-name>]	要求回数および処理時間の記録を削除するサーブレットです

- 例

```
clear-web-statistic -server server1 -ctx ctx1
```

4.2.8.11. modify-web-cookie-policy

WebエンジンのHTTPのCookieポリシーの設定を修正します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/cookie-policy

- エイリアス

modcookie

- 使用方法

```
modify-web-cookie-policy [-cluster <cluster-name> | -server <server-name>]
                        [-auer,--apply-url-encoding-rule <apply-url-encoding-rule>]

                        [-enc,--charset-encoding <charset-encoding>]
                        [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-auer,--apply-url-encoding-rule <apply-url-encoding-rule>]	URLエンコーディングのルール適用可否を修正します
[-enc,--charset-encoding <charset-encoding>]	URLエンコーディングのルールを適用する際に使用する文字セットのエンコーディングを修正します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
modify-web-cookie-policy -server server1 -auer false

modify-web-cookie-policy -server server1 -enc UTF-8
```

4.2.8.12. modify-web-encoding

Webエンジンのエンコーディングを修正します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/encoding

- エイリアス

modenc

- 使用方法

```
modify-web-encoding [-cluster <cluster-name> | -server <server-name>]
                    [-f, --forceLock]
                    <charset encoding>
                    -co | -default | -forced
                    [-url]
                    [-req]
                    [-res]
                    [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<charset encoding>	修正する文字セットのエンコーディング名です。入力された値が正しくない場合はエラーが出力され、命令が失敗します

パラメータ	説明
-co -default -forced	入力されたエンコーディングをdefaultに設定するか、あるいはforcedに設定するかを決定します。リクエスト・エンコーディングである場合は、この他にclient-overrideも設定できます
[-url]	リクエストURLの文字セットのエンコーディングを修正します
[-req]	リクエストの文字セットのエンコーディングを修正します
[-res]	レスポンスの文字セットのエンコーディングを修正します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
modify-web-encoding -server server1 EUC-KR -forced -url

modify-web-encoding -server server1 ISO-8859-1 -default -url -res -req
```

4.2.8.13. modify-web-properties

Webエンジンのプロパティ設定を修正します。修正されるプロパティは大/小文字を区別します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/properties

- エイリアス

modwebpr

- 使用方法

```
modify-web-properties [-cluster <cluster-name> | -server <server-name>]
                        [-f, --forceLock]
                        [-p, --properties <properties>]
                        [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f, --forceLock]	設定を強制的に変更して反映します
[-p, --properties <properties>]	修正するWebエンジンのプロパティを設定します – それぞれのプロパティは「key=value」形式で「=」を利用して設定します

パラメータ	説明
	– 1つ以上のプロパティを修正する場合は、各プロパティをコンマ(,)で区分して設定します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
modify-web-properties -server server1
-p jeus.servlet.jsp.compile-java-source-concurrently=false
```

4.2.8.14. modify-response-header

Webエンジンのレスポンス・ヘッダーの設定を修正します。修正されるレスポンス・ヘッダーは大/小文字を区別しません。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/response-header

- エイリアス

modwebrh

- 使用方法

```
modify-response-header [-cluster <cluster-name> | -server <server-name>]
                        [-f, --forceLock]
                        [-n, --name <header-name>]
                        [-v, --value <header-value>]
```

- パラメータ

パラメータ	説明
[-f, --forceLock]	設定を強制的に変更して反映します
[-n, --name <header-name>]	修正するHTTPのレスポンス・ヘッダーの名前を設定します。入力値は大/小文字を区別しません。 同じ名前のヘッダーは1つのみ許容されます
[-v, --value <header-value>]	修正するHTTPのレスポンス・ヘッダーの値を設定します。入力値は大/小文字を区別しません

- 例

```
modify-response-header -server server1 -n testheader -v testvalue3
```

4.2.8.15. modify-jsp-engine

WebエンジンのJSPエンジンの設定を修正します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/jsp-engine

- エイリアス

modjsp

- 使用方法

```
modify-jsp-engine [-cluster <cluster-name> | -server <server-name>]
  [-f, --forceLock]
  [-rjwd,--remove-jsp-work-dir | -jwd,--jsp-work-dir <jsp-work-dir>]
  [-javac,--java-compiler <java-compiler>]
  [-cod,--compile-output-dir <compile-output-dir> |
  -rcod,--remove-compile-output-dir]
  [-rcopt,--remove-compile-option | -copt,--compile-option <compile-option>]
  [-renc,--remove-compile-encoding | -enc,--compile-encoding <compile-encoding>]

  [-cij,--check-included-jspfile <check-included-jspfile>]
  [-kg,--keep-generated <keep-generated>]
  [-gjr,--graceful-jsp-reloading <graceful-jsp-reloading>]
  [-gjrp,--graceful-jsp-reloading-period <graceful-jsp-reloading-period>]
  [-umc,--use-in-memory-compilation <use-in-memory-compilation>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-rjwd,--remove-jsp-work-dir -jwd,--jsp-work-dir <jsp-work-dir>]	-jwd,--jsp-work-dir <jsp-work-dir>を利用してJSPをJavaで作成したファイルとそのソース・ファイルをコンパイルしたクラス・ファイルが保存される位置を設定します。この際、設定値は絶対パスで入力します。 -rjwd,--remove-jsp-work-dirを利用して設定を削除できます。削除する場合、エンジンの内部デフォルト値が使用されます
[-javac,--java-compiler <java-compiler>]	JSPのJavaソースをサーブレット・クラス・ファイルにコンパイルするためのJavaコンパイラを指定します。

パラメータ	説明
	<p>以下のうち1つを設定します。</p> <ul style="list-style-type: none"> – java6(デフォルト値) – sun.tools.javac – javac – com.sun.tools.javac <p>各値についての説明はスキーマを参照してください。デフォルト値を設定するのが最も効率的であるため、設定は別途行わないことを推奨します</p>
[-cod,--compile-output-dir <compile-output-dir> -rcod,--remove-compile-output-dir]	<p>-cod,--compile-output-dir <compile-output-dir>を利用して、JSP ファイルによって作成されたクラス・ファイルをJSP Work Dirディレクトリー以外のディレクトリーに保存できる設定です。設定していない場合、クラス・ファイルはJSP Work Dirディレクトリーに位置します。</p> <p>-rcod,--remove-compile-output-dirを利用して、設定を削除できます。削除する場合、エンジンの内部デフォルト値が使用されます</p>
[-rcopt,--remove-compile-option -copt,--compile-option <compile-option>]	<p>-copt,--compile-option <compile-encoding>を利用して、サーブレット・コンパイラーで使用されるオプションを設定します。</p> <p>-rcopt,--remove-compile-optionを利用して、設定した値を削除できます</p>
[-renc,--remove-compile-encoding -enc,--compile-encoding <compile-encoding>]	<p>-enc,--compile-encoding <compile-encoding>を利用して、JSP ファイルがパースされて作成されたサーブレット・ソース・ファイルをコンパイルする際に、エンコーディング・オプションに指定する値を設定します。</p> <p>-renc,--remove-compile-encodingを利用して、設定されたエンコーディングを削除できます</p>
[-cij,--check-included-jspfile <check-included-jspfile>]	<p>JSPファイルの変更可否を確認する際にインクルードされたJSP ファイル、Tagファイルの変更有無を確認し、変更されている場合、該当JSPファイルをコンパイルする機能の使用可否を設定します</p>
[-kg,--keep-generated <keep-generated>]	<p>JSPページから作成されたJavaソース・ファイルをコンパイルして、サーブレット・クラス・ファイルを作成した後、Javaソース・ファイルの保存可否を決定します</p>

パラメータ	説明
<code>[-gjr,--graceful-jsp-reloading <graceful-jsp-reloading>]</code>	JSPソースとコンパイルされたクラス・ファイルが共有されており、1つのJVMで独占し、JSPファイルを配置型のJavaファイルに変換してコンパイルすることを設定します。 JSP Work Dir設定によって共有されるフォルダが指定されている場合に限って動作します
<code>[-gjr,--graceful-jsp-reloading-period <graceful-jsp-reloading-period>]</code>	Graceful JSP reloadingが動作する周期を設定します
<code>[-umc,--use-in-memory-compilation <use-in-memory-compilation>]</code>	サービス中のJSPファイルを新しくコンパイルする必要がある場合、.javaおよび.classファイルをメモリー上に作成してコンパイルする機能を設定します。ただし、.classファイルの場合は、今後再起動する際に再コンパイルすることがないように、バックグラウンド・スレッドを通じてファイル・システムに書き込みます。 また、<keep-generated>がtrueの場合、.javaファイルをバックグラウンド・スレッドを通じてファイル・システムに書き込みます

- 例

```
modify-jsp-engine -server server1 -jwd /home/jeus/jsp
```

4.2.8.16. modify-tmax-connector

Tmaxコネクタのスレッド・プールの設定などを変更します。ただし、運用中のスレッド・プールのスレッド数が即時変わるのではなく、スレッド・プールのチェック周期に基づいて変わります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/tmax-connector

- エイリアス

modtmaxcon

- 使用方法

```
modify-tmax-connector [-cluster <cluster-name> | -server <server-name>]
                        [-f, --forceLock]
                        -name <web-connection-name>
                        [-tmin <minimum-thread-num>]
                        [-tmax <maximum-thread-num>]
                        [-tidle <max-idle-time>]
                        [-obuf <output-buffer-size>]
```

```

[-addr <server-address>]
[-port <server-port>]
[-svrg <server-group-name>]
[-svr <server-name>]

```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
-name <web-connection-name>	Tmaxコネクタの名前です
[-tmin <minimum-thread-num>]	[動的変更] スレッド・プールの最小スレッド数です
[-tmax <maximum-thread-num>]	[動的変更] スレッド・プールの最大スレッド数です
[-tidle <max-idle-time>]	スレッド・プールのスレッド別のアイドル時間です
[-obuf <output-buffer-size>]	サーブレット応答バッファのサイズです
[-addr <server-address>]	Tmaxサーバーのアドレスです
[-port <server-port>]	Tmaxサーバーのポート番号です
[-svrg <server-group-name>]	Tmaxサーバーのサーバー・グループの名前です
[-svr <server-name>]	Tmaxサーバーの名前です

- 例

```

modify-tmax-connector -name tmax1 -tmin 30 -server server1 -tmax 30

```

4.2.8.17. modify-virtual-host

バーチャル・ホストのアクセス・ログのフォーマット設定を変更します。バーチャル・ホストのアクセス・ログが有効状態ではない場合に実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/virtual-host

- エイリアス

modvh

- 使用方法

```

modify-virtual-host [-cluster <cluster-name> | -server <server-name>]
                        [-f, --forceLock]
                        <virtual-host-name>
                        [-aluph <access-log-use-parent-handler (true/false)>|
                        -alf <access-log-format> |

```

```
-aluse <use-access-log (true/false)> |
-alhnl <access-log-enable-host-name-lookup> |
-alexext <access-log-excluded-extensions>]
[-hnrm <host-name> | -hnadd <host-name>]
[-ast <attach-stacktrace-on-error>]
```

● パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<virtual-host-name>	バーチャル・ホストの名前です
-aluph <access-log-use-parent-handler>	[動的変更] Webエンジンのアクセス・ログに残すか否かを指定します (true false または t f)
[-alf <access-log-format>]	[動的変更] アクセス・ログのフォーマットです。途中にスペース文字がある場合は二重引用符(" ")で囲みます。 フォーマットに関する詳細は『JEUS Webエンジンガイド』の1.6.10節「アクセス・ログのフォーマット設定」を参照してください
-aluse <use-access-log>	[動的変更] アクセス・ログを使用するか否かを指定します (true false または t f)
-alhnl <access-log-enable-host-name-lookup>	[動的変更] %hフォーマットについてロギング時に、IPアドレスのDNS Resolutionの可否を決定します (true false または t f)
-alexext <access-log-excluded-extensions>	[動的変更] アクセス・ログに残さない拡張子を設定します。 .gif、.jpgのように複数を設定したい場合はコンマ(,)を使用します。 設定しない場合はコンマのみ入力します
-hnrm <host-name>	バーチャル・ホストに登録されているホスト名を削除します
-hnadd <host-name>	バーチャル・ホストに新しいホスト名を登録します
[-ast <attach-stacktrace-on-error>]	JEUSから送られるエラー・ページにスタック・トレースを添付するか否かを設定します

● 例

```
modify-virtual-host -server server1 vhost1 -alf "common %I"
```


4.2.8.18. modify-web-engine-configuration

Webエンジンの設定を動的に変更します。このコマンドは、実際にサービスの運用中にも反映される設定のみ変更を行います。変更可能な情報はモニタリング周期、アクセス・ログのフォーマットです。(qa: . .)

● 関連スキーマ

web-engine.xsd - web-engine

● エイリアス

setwebcfg, setwebconf, set-web-engine-configuration, modwebcfg, modwebconf

● 使用方法

```
modify-web-engine-configuration [-cluster <cluster-name> | -server <server-name>]

                                [-f, --forceLock]
                                [-tpp <monitoring/check-thread-pool>]
                                [-crp <monitoring/check-class-reload>]
                                [-sp <monitoring/check-session>]
                                [-alf <access-log-format>]
                                [-aluse <use-access-log>]
                                [-alhnl <access-log-enable-host-name-lookup>]
                                [-alex <access-log-excluded-extensions>]
                                [-ast <attach-stacktrace-on-error>]
                                [-att <async-timeout-min-threads>]
                                [-rerp | -erp <default-error-page>]
```

● パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-tpp <monitoring/check-thread-pool>]	スレッド・プールのチェック周期です
[-crp <monitoring/check-class-reload>]	Webコンテキストのリロードのチェック周期です
[-sp <monitoring/check-session>]	セッションのチェック周期です
[-alf <access-log-format>]	[動的変更] アクセス・ログのフォーマットです。途中にスペース文字がある場合は二重引用符(" ")で囲みます。 フォーマットに関する詳細は『JEUS Webエンジンガイド』の1.6.10節「アクセス・ログのフォーマット設定」を参照してください

パラメータ	説明
<code>[-aluse <use-access-log (true/false)>]</code>	[動的変更] アクセス・ログを使用するか否かを指定します (true false または t f)
<code>[-alhn1 <access-log-enable-host-name-lookup>]</code>	[動的変更] %hフォーマットについてロギング時に、IPアドレスのDNS Resolutionの可否を決定します (true false または t f)
<code>[-alex1 <access-log-excluded-extensions>]</code>	[動的変更] アクセス・ログに残さない拡張子を設定します。 .gif、.jpgのように複数を設定したい場合はコンマ(,)を使用します。 設定しない場合はコンマのみ入力します
<code>[-ast <attach-stacktrace-on-error>]</code>	JEUSが送信するエラー・ページにスタック・トレースを添付するか否かを設定します
<code>[-att <async-timeout-min-threads>]</code>	Servlet 3.0のAsynchronous Servletを使用する場合、タイムアウト処理をするためのスレッド・プールの最小スレッド数を設定します。 0の場合、タイムアウトが正常に動作しないこともあるため、常に1以上を設定する必要があります
<code>[-rerp -erp <default-error-page>]</code>	-erp <default-error-page>を利用して、Webアプリケーションに別途のエラー・ページを設定していない場合に使用するエラー・ページを設定します。静的なページ(HTML、HTM)のみ設定でき、絶対パスである必要があります。 -rerpを利用して、設定を削除できます

- 例

- `[-alf <access-log-format>]` オプションを使用した例

```
modify-web-engine-configuration -server server1 -alf "common %I"

modify-web-engine-configuration -server server1 -alf "%h %l %u %t \"%r\" %>s %b"
```

- `[-aluse <use-access-log >]` オプションを使用した例

```
modify-web-engine-configuration -server server1 -aluse false
```

- `[-alex1 <access-log-excluded-extensions>]` オプションを使用した例

```
modify-web-engine-configuration -server server1 -alext .gif
modify-web-engine-configuration -server server1 -alext .gif,.jpg
```

- [-alext <access-log-excluded-extensions>]オプションを使用して既存の拡張情報を削除した例

```
modify-web-engine-configuration -server server1 -alext ,
```

- [-tpp <monitoring/check-thread-pool>]、[-crp <monitoring/check-class-reload>]、[-sp <monitoring/check-session>]オプションを使用した例

```
modify-web-engine-configuration -tpp 10000 -crp 20000 -server server1 -sp 25000
```

- [-ast <attach-stacktrace-on-error>]オプションを使用した例

```
modify-web-engine-configuration -server server1 -ast true
```

- [-att <async-timeout-min-threads>]オプションを使用した例

```
modify-web-engine-configuration -server server1 -att 10
```

- [-rerp | -erp <default-error-page>]オプションを使用した例

```
modify-web-engine-configuration -server server1 -erp /home/jeus/error/error.html
```

4.2.8.19. modify-web-listener

HTTP、TCP、AJPリスナーのスレッド・プールの設定などを変更します。ただし、運用中のスレッド・プールのスレッド数が即時変わるのではなく、スレッド・プールのチェック周期に基づいて変わります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/http-listener, ajp13-listener, tcp-listener

- エイリアス

modwebl

- 使用方法

```
modify-web-listener [-cluster <cluster-name> | -server <server-name>]
                    [-f, --forceLock]
                    -name <web-connection-name>
                    [-tmin <minimum-thread-num>]
```

```

[-tmax <maximum-thread-num>]
[-tidle <max-idle-time>]
[-tauto <enable-auto-tuning>]
[-tlimit <thread-num-limit>]
[-obuf <output-buffer-size>]
[-http2 <enable-http2>]

```

● パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
-name <web-connection-name>	リスナーの名前です
[-tmin<minimum-thread-num>]	[動的変更] スレッド・プールの最小スレッド数です
[-tmax <maximum-thread-num>]	[動的変更] スレッド・プールの最大スレッド数です
[-tidle <max-idle-time>]	スレッド・プールの各スレッドの最大のアイドル時間です
[-tauto <enable-auto-tuning>]	自動チューニングを使用します。このオプションを使用すると、min、max設定を行っても適用されません。HTTPリスナーにのみ適用されます
[-tlimit <thread-num-limit>]	自動チューニングを使用する場合に、スレッド数の制限を設定します
-obuf <output-buffer-size>	サーブレットが使用する応答バッファのサイズです
[-http2 <enable-http2>]	HTTPリスナーがHTTP/2を使用します

● 例

```

modify-web-listener -name http1 -tmin 30 -server server1 -tmax 30

modify-web-listener -name ajp1 -server server1 -tmin 20

modify-web-listener -name tcp1 -server server1 -tmax 25

```

4.2.8.20. modify-webtob-connector

WebtoBコネクタースレッド・プールの設定などを変更します。ただし、運用中のスレッド・プールのスレッド数が即時変わるのではなく、スレッド・プールのチェック周期に基づいて変わります。

● 関連スキーマ

web-engine.xsd - web-engine/web-connections/webtob-connector

● エイリアス

modwebtobcon

- 使用方法

```

modify-webtob-connector [-cluster <cluster-name> | -server <server-name>]
    [-f,--forceLock]
    -name <web-connection-name>
    [-num <thread-number>] [-obuf <output-buffer-size>]
    [-ver <wjp-version>]
    [-addr <server-address>]
    [-port <server-port> | -dsocket]
    [-wbhome <webtob-home>|-ipcport <ipc-base-port>]
    [-regid <registration-id>]
    [-sndbuf <send-buffer-size> ]
    [-rcvbuf <receive-buffer-size> ]

```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
-name <web-connection-name>	WebtoBコネクタの名前です
-num <thread-number>	[動的変更] スレッド・プールのスレッド数です。現在のスレッド数より低く設定する場合、アクティブなスレッドが終了される可能性があるため、要求処理が行われている場合は使用しないことを推奨します
[-obuf <output-buffer-size>]	サブレットの応答バッファのサイズです
-ver <wjp-version>	WebtoBと通信時に使用するプロトコルのバージョンです。1または2を入力します
-addr <server-address>	WebtoBサーバーのアドレスです
-port <server-port>	WebtoBサーバーのポート番号です
-regid <registration-id>	WebtoBの登録IDです
-dsocket	UNIXドメイン・ソケット(Pipe)を利用してJEUSとWebtoB間の通信を行います
-wbhome <webtob-home>	JEUSとWebtoBが同じデバイス内にインストールされている場合、WebtoBがインストールされた絶対パスです
-ipcport <ipc-base-port>	Windows環境でWebtoBの内部プロセス間の通信を行う際に使用するデフォルトのポート番号です
-sndbuf <send-buffer-size>	TCPソケットやUNIXドメイン・ソケットのSO_SNDBUFを設定します。0の場合はOSのデフォルト値を使用します
-rcvbuf <receive-buffer-size>	TCPソケットやUNIXドメイン・ソケットのSO_RCVBUFを設定します。0の場合はOSのデフォルト値を使用します

- 例

```
modify-webtob-connector -name webtob1 -num 30 -sndbuf 1200 -rcvbuf 2400 -server  
server1
```

4.2.8.21. precompile-jsp

デプロイされたWebコンテキストのJSPファイルを事前にコンパイルします。

一般的に、WebブラウザからのJSPファイルの要求がWebエンジンに来る際にJSPファイルがコンパイルされます。こうしてJSPファイルの初期リクエストは多くのパースとコンパイル作業を通じて大量のリソースを消費し、応答時間が長くなります。

開発されたJSPファイルが多く、ユーザーの要求が頻繁なWebサイトの場合、「初期リクエストの実行」の所要時間やリソースがサービスに大きく影響を及ぼすことがあります。このような場合、開発されたJSPソース・ファイルをWebエンジンが開始し、サービスの開始前に**precompile-jsp**コマンドでコンパイルをすることで、問題を解決できます。

precompile-jspコマンドは、JEUSが起動した状態でデプロイされたモジュールに限ってコンパイルを実行します。JEUSが起動していない状態で事前コンパイルを実行するためには「[4.3. appcompiler](#)」を使用する必要があります。

参考

1. precompile-jspはWebエンジンの他のコマンドとは異なり、クラスターをオプションに使用しません。
2. JEUS 6まではjspcというコマンドでJEUS_HOME/binの下に別途存在していたバッチ・コンパイラーを、JEUS 7からはjeusadminの内部コマンドに統合しました。

● 関連スキーマ

web-engine.xsd

● エイリアス

jspc

● 使用方法

```
precompile-jsp [-server <server-name>]  
               -ctx <context-name>  
               [-e <excluded-jsp-list-file> | -l <included-jsp-list-file>]
```

● パラメータ

パラメータ	説明
-ctx <context-name>	Webコンテキストの名前を設定します。EAR内にある場合、 ear-name#context-nameと入力します

パラメータ	説明
<code>[-e <excluded-jsp-list-file>]</code>	Webコンテキストを基準にして、コンパイルしないパスを記述したテキスト・ファイルを指定します。JSPのパスはコンテキスト・ルートの相対パスを持つ必要があり、必ずスラッシュ(/)で始めます。 jsp-list-fileの位置はシステムの絶対パスに指定します
<code>[-l <included-jsp-list-file>]</code>	Webコンテキストを基準にして、コンパイルするパスを記述したテキスト・ファイルを指定します。JSPのパスはコンテキスト・ルートの相対パスを持つ必要があり、必ずスラッシュ(/)で始めます。 jsp-list-fileの位置はシステムの絶対パスに指定します

- 例

```
precompile-jsp -server server1 -ctx myctx
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.8.22. reload-web-context

ディスクから指定したエンティティー (Webコンテキスト) をリロードします。既にロードされているサーブレット (JSP) とクラスをリロードします。このコマンドは、サーブレットやクラスのリロード機能が無効化されている場合に有効です。(enable-reload, check-class-reload)

- 関連スキーマ

jeus-web-dd.xsd

- エイリアス

webreload, reloadctx

- 使用方法

```
reload-web-context [-cluster <cluster-name> | -server <server-name>]
                  -ctx <context-name>
```

- パラメータ

パラメータ	説明
<code>-ctx <context-name></code>	リロードするコンテキストです。必ず値を設定します

- 例

```
reload-web-context -server server1 -ctx servlets-examples
```

4.2.8.23. remove-web-cookie-policy

WebエンジンのHTTPのCookieポリシー設定を削除します。ただし、XMLにのみ反映し、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/cookie-policy

- エイリアス

rmcp, rmcookie

- 使用方法

```
remove-web-cookie-policy [-cluster <cluster-name> | -server <server-name>]
                          [-f,--forceLock]
                          [-auer,--apply-url-encoding-rule]
                          [-enc,--charset-encoding]
                          [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-auer,--apply-url-encoding-rule]	URLエンコーディングの規則の適用可否を削除します
[-enc,--charset-encoding]	URLエンコーディングの規則を適用する際に使用する文字セットのエンコーディングを削除します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
remove-web-cookie-policy -server server1 -auer
```

4.2.8.24. remove-web-encoding

Webエンジンのエンコーディングを削除します。ただし、XMLにのみ反映し、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/encoding

- エイリアス

rmenc

- 使用方法

```
remove-web-encoding [-cluster <cluster-name> | -server <server-name>]
                    [-f,--forceLock]
                    [-url]
                    [-req]
                    [-res]
                    [-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-url]	リクエストURLの文字セットのエンコーディングを削除します
[-req]	リクエストの文字セットのエンコーディングを削除します
[-res]	レスポンスの文字セットのエンコーディングを削除します
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
remove-web-encoding -server server1 -url

remove-web-encoding -server server1 -req -res
```

4.2.8.25. remove-web-properties

Webエンジンのプロパティ設定を削除します。削除されるプロパティは大/小文字を区別します。ただし、XMLにのみ反映し、サービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/properties

- エイリアス

rmwebpr

- 使用方法

```
remove-web-properties [-cluster <cluster-name> | -server <server-name>]
                     [-f,--forceLock]
```

```
[-a,--all | -k,--keys <keys>]
[-vh <virtual-host-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-a,--all -k,--keys <keys>]	<ul style="list-style-type: none">– -k,--keys <keys>を利用して、Webエンジンの特定プロパティを削除します。入力値は大/小文字を区別します。1つ以上のプロパティを同時に削除する場合はコンマ(,)で区分し、削除するプロパティのキーを設定します– -a,--allを利用すると、現在設定されているプロパティをすべて削除できます
[-vh <virtual-host-name>]	バーチャル・ホストを指定します

- 例

```
remove-web-properties -server server1 -k
jeus.servlet.jsp.compile-java-source-concurrently1

remove-web-properties -server server1 -a
```

4.2.8.26. remove-response-header

Webエンジンのレスポンス・ヘッダーの設定を削除します。削除されるレスポンス・ヘッダーは大/小文字を区別しません。ただし、XMLにのみ反映し、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/response-header

- エイリアス

rmwebbrh

- 使用方法

```
remove-response-header [-cluster <cluster-name> | -server <server-name>]
                        [-f,--forceLock]
                        [-a,--all | -n,--name <header-name>]
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
[-a,--all -n,--name <header-name>]	<ul style="list-style-type: none"> -n,--name <header-name>を利用して、特定HTTPのレスポンス・ヘッダーを削除します。入力値は大/小文字を区別しません。1つ以上のヘッダーを同時に削除する場合、コンマ (,)で区分して、削除するヘッダーの名前を設定します -a,--allを利用すると、現在設定されているHTTPのレスポンス・ヘッダーをすべて削除できます

- 例

```
remove-response-header -server server1 -n testheader

remove-response-header -server server1 -n testheader2,testheader4

remove-response-header -server server1 -a
```

4.2.8.27. remove-tmax-connector

Tmaxコネクタを削除します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/tmax-connector

- エイリアス

rmtmaxcon

- 使用方法

```
remove-tmax-connector [-cluster <cluster-name> | -server <server-name>]
                        [-f,--forceLock]
                        <web-connection-name>
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<web-connection-name>	Tmaxコネクタの名前です

- 例

```
remove-tmax-connector -server server1 tmax1
```

4.2.8.28. remove-virtual-host

バーチャル・ホストを削除します。ただし、XMLにのみ反映され、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/virtual-host

- エイリアス

rmvh

- 使用方法

```
remove-virtual-host [-cluster <cluster-name> | -server <server-name>]
                    [-f,--forceLock]
                    <virtual-host-name>
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<virtual-host-name>	バーチャル・ホストの名前です

- 例

```
remove-virtual-host -server server1 host2
```

4.2.8.29. remove-web-listener

HTTP、TCP、AJPリスナーを削除します。ただし、XMLにのみ反映され、実際にサービスに適用するには、サーバーを再起動する必要があります。ADMIN-HTTPリスナーは削除されません。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/http-listener, ajp13-listener, tcp-listener

- エイリアス

rmwebl

- 使用方法

```
remove-web-listener [-cluster <cluster-name> | -server <server-name>]
                    [-f,--forceLock]
                    <web-connection-name>
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<web-connection-name>	リスナーの名前です

- 例

```
remove-web-listener -server server1 http1
```

4.2.8.30. remove-webtob-connector

WebtoBコネクタを削除します。ただし、XMLにのみ反映し、実際にサービスに適用するにはサーバーを再起動する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections/webtob-connector

- エイリアス

rmwebtobcon

- 使用方法

```
remove-webtob-connector [-cluster <cluster-name> | -server <server-name>]
                        [-f,--forceLock]
                        <web-connection-name>
```

- パラメータ

パラメータ	説明
[-f,--forceLock]	設定を強制的に変更して反映します
<web-connection-name>	WebtoBコネクタの名前です

- 例

```
remove-webtob-connector -server server1 webtob1
```

4.2.8.31. resume-web-component

一時停止中のWebコンポーネント(サーブレット)を再開します。

参考

JEUS 6ではwebresumeコマンドとwebtob -rコマンドに分離されていましたが、JEUS 7から統合されました。また、JEUS 7からWebコンテキストの開始コマンドは別途提供しません。DASの[start-application](#)を使用して制御する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections

- エイリアス

webresume

- 使用方法

```
resume-web-component [-cluster <cluster-name> | -server <server-name>]
                        [-ctx,--context <context>]
                        -svl,--servlet <servlet> ]
```

- パラメータ

パラメータ	説明
[-ctx, --context <context>]	開始するサーブレットが含まれているWebコンテキストを設定します。 必ず-svlオプションと一緒に設定します。 application-information -typeオプションを使用したコマンドを使用した照会結果のうちモジュール名に該当する値を<context>に設定します
-svl,--servlet <servlet>	停止したサーブレットのうち、再開するサーブレットです。コンテキスト・オプションが設定されている必要があります

- 例

- [-svl,--servlet <servlet>]オプションを使用した例

```
resume-web-component -server server1 -ctx servlets -svl CookieExample
```

4.2.8.32. show-request-processing-flow

URLパターンや特定ホスト名で入ってきた要求の処理フローを表示します。

- 関連スキーマ

web-app_3_0.xsd

- エイリアス

reqflow

- 使用方法

```
show-request-processing-flow [-cluster <cluster-name> | -server <server-name>]
                             -path <url-pattern>
                             [-hostname <host-name>]
```

- パラメータ

パラメータ	説明
-path <url-pattern>	スラッシュ(/)で始まるURLパターンです
[-hostname <host-name>]	HTTPホスト・ヘッダーに明示されているホスト一覧です

- 例

```
show-request-processing-flow -server server1 -path /test/index.jsp

show-request-processing-flow -server server1 -path /test -hostname examples.com
```

4.2.8.33. show-web-engine-configuration

Webエンジンの設定情報を表示します。このコマンドの実行によって表示される構成項目とフィールドの情報は、XMLのWebエンジンの構成リファレンスを参照してください。

空欄で表示されている項目は、XMLに設定されていないという意味です。つまり、値が表示されていれば、実際に設定されているか、デフォルト値が存在する設定です。

- 関連スキーマ

web-engine.xsd - web-gine

- エイリアス

webcfg, webconf, showwebcfg, showwebconf

- 使用方法

```
show-web-engine-configuration [-cluster <cluster-name> | -server <server-name>]

                                [-vh,--virtual host]
                                [-cn,--web-connections <connection-type>]
                                [-al,--access-log]
                                [-sc,--session-config]
                                [-rh,--response-header]
                                [-mo,--monitoring]
                                [-enc,--encoding]
                                [-cp,--cookie-policy]
                                [-jsp,--jsp-engine]
                                [-pr,--properties]
                                [-erp,--error-page]
                                [-att,--async-timeout_thread]
                                [-ast,--attach-stacktrace]
```

● パラメータ

パラメータ	説明
[-vh,--virtual host]	バーチャル・ホストの設定情報を出力します
[-cn,--web-connections <connection-type>]	すべてのWebコネクションの設定情報を出力します。引数を与える場合、該当タイプのコネクション情報のみ出力します
[-al,--access-log]	アクセス・ログの設定情報を出力します
[-sc,--session-config]	Webエンジンのセッション設定情報を出力します
[-rh,--response-header]	レスポンス・ヘッダーの設定情報を出力します
[-mo,--monitoring]	Webエンジンのモニタリングの設定情報を出力します
[-enc,--encoding]	Webエンジンのエンコード設定情報を出力します
[-cp,--cookie-policy]	Cookieのポリシー設定情報を出力します
[-jsp,--jsp-engine]	JSPエンジンの設定情報を出力します
[-pr,--properties]	Webエンジンのプロパティ設定情報を出力します
[-erp,--error-page]	Webエンジンのデフォルトのエラーページ設定情報を出力します
[-att,--async-timeout_thread]	非同期サーブレットのタイムアウト処理のスレッド設定情報を出力します
[-ast,--attach-stacktrace]	エラー発生時のスタック・トレース追加可否の設定情報を出力します

● 例

- Webエンジンの設定をすべて出力

```
show-web-engine-configuration -server server1
```


– オプション別のWebエンジン設定の出力

```
show-web-engine-configuration -erp -att -server server1

show-web-engine-configuration -server server1 -sc
```

4.2.8.34. show-web-statistics

起動中のWebエンジンのJVMで使用するメモリー情報、Webエンジンに属している各リスナー/コネクタースレッド・プールの状態、コンテキストの要求回数および処理時間などの情報を、パラメータの設定に従って出力します。

- 関連スキーマ

web-engine.xsd

- エイリアス

webstats, stat, st

- 使用方法

```
show-web-statistics [-server <server-name>]
                    [-ctx,--context <context-name>]
                    [-t,--thread | -s,--session | -r,--request | -m,--memory]
```

- パラメータ

パラメータ	説明
[-t,--thread]	Webエンジンに属している各リスナーのスレッド・プールの状態を出力します。出力される情報には、現在割り当てられているワーカー・スレッドの数とWait-Queueに蓄積されているクライアントの数、そしてスレッド・プールを維持する最大スレッド数を含みます
[-s,--session]	Webエンジンに構成されているセッション・サーバーのコネクション情報を出力します。 情報は現在接続されているセッション・サーバーの名前、全体セッションの数、プールで使用されているセッション・サーバーとコネクションの数を含みます
[-r,--request]	各コンテキストに対して処理された要求の累積数値と平均処理時間を出力します
[-m,--memory]	起動中のWebエンジンで現在JVMにて使用しているメモリーの状態を出力します

- 例

```
## リスナーのスレッド・プール情報の確認
show-web-statistics -server server1 -t

## セッション・サーバー情報の確認
show-web-statistics -server server1 -s
```

4.2.8.35. suspend-web-component

指定したサーブレットを停止し、クライアントにエラー・ページを表示します。

参考

JEUS 6ではwebsuspendコマンドとwebtob -sコマンドに分離されていましたが、JEUS 7から統合されました。また、JEUS 7からWebコンテキストの停止コマンドは別途提供しません。DASの[stop-application](#)を使用して制御する必要があります。

- 関連スキーマ

web-engine.xsd - web-engine/web-connections

- エイリアス

websuspend

- 使用方法

```
suspend-web-component [-cluster <cluster-name> | -server <server-name>]
                        [-ctx,--context <context>]
                        -svl,--servlet <servlet>]
```

- パラメータ

パラメータ	説明
[-ctx, --context <context>]	停止するサーブレットが含まれているWebコンテキストを設定します。 必ず-svlオプションと一緒に設定する必要があります。 application-information -typeオプションを利用したコマンドを使用した照会結果のうち、Unique Module Nameに該当する値を<context>に設定します
-svl,--servlet <servlet>	開始したサーブレットのうち、停止するサーブレットです。コンテキスト・オプションが設定されている必要があります

- 例

- [-svl,--servlet <servlet>]オプションを使用した例

```
suspend-web-component -svl CookieExample -server server1 -ctx servlets
```

4.2.9. セッション関連コマンド

セッション・コマンドを2種類に分類して説明します。

- モニタリングおよび制御コマンド

セッション状態を確認または制御するコマンドです。

コマンド	説明
list-session	アイドル時間を基準にソートされたセッション情報を表示します
remove-session	入力したタイムアウトの間アクセスのないセッションを削除します。セッション・タイムアウトとは関係がありません

- 設定変更関連コマンド

セッションの設定を追加、削除、変更するためのコマンドです。変更された設定はXMLには適用されますが、運用中のサービス(メモリー領域)には適用されません。サービスに適用するには、サーバーを再開始する必要があります。

コマンド	説明
show-session-configuration	セッション関連の設定情報を確認します
modify-session-configuration	セッション関連の設定情報を確認して変更します

4.2.9.1. 共通適用事項

以下は、セッション関連コマンドで共通で適用されるオプションについての説明です。以下のオプションのうち1つのみ設定できます。

- [-cluster <cluster_name>]オプション

- 各コマンドが適用されるクラスターを指定するためのオプションです。オプションを設定していない場合、現在接続しているサーバーのWebエンジンに適用されます。
 - このオプションは、各コマンドのオプション説明では省略されます。

- [-server <server_name>]オプション

- 各コマンドが適用されるサーバーを指定するためのオプションです。オプションを設定していない場合、現在接続しているサーバーのWebエンジンに適用されます。
- このオプションは、各コマンドのオプション説明では省略されます。

参考

1. サーバーに即時反映されない設定の場合、つまりペンディング状態の設定を最初の設定に再設定するコマンドを実行すると、サーバーの再起動が必要というメッセージ(Restart the Server)が出力されません。
2. DASの場合、上記のいずれかのオプションを指定する必要があります。しかし、DASではない一般管理サーバーで該当コマンドを実行するとき、いずれのオプションも指定しない場合は、接続しているサーバーに対してコマンドが実行されます。

4.2.9.2. list-session

アイドル時間を基準にソートされたセッション情報を表示します。

- 関連スキーマ

jeus-session-server.xsd, jeus-session-domain.xsd

- エイリアス

lisession, lss

- 使用方法

```
list-session [-server <server-name>]
              [-s,--simple ]
              [-c,--count <int count: default=100>]
              [-t,--target <String managerName>]
```

- パラメータ

パラメータ	説明
[-s,--simple]	結果をシンプル・フォーマットで出力します
[-c,--count <int count: default=100>]	モニタリングで確認するセッションの数を設定します。省略する場合、100がデフォルトで適用されます
[-t,--target <String managerName>]	セッション情報を確認するターゲット・セッション・マネージャーを設定します。オプションを指定しない場合、サーバー内のすべてのマネージャーの情報を表示します

- 例

- オプションを使用しない場合の例

```
list-session -server server1
```

- [-s,--simple]オプションを使用した例

```
list-session -server server1 --simple
```

4.2.9.3. remove-session

アイドル時間を基準に設定した時間が過ぎたセッションを削除します。

- 関連スキーマ

jeus-session-server.xsd, jeus-session-domain.xsd

- エイリアス

rmsession, rms

- 使用方法

```
remove-session [-server <server-name>]  
                  [-time,--timeout <int timeout(min)>]
```

- パラメータ

パラメータ	説明
[-time,--timeout <int timeout(min)>]	セッションを削除する基準となるタイムアウトを設定します

- 例

- [-time,--timeout <int timeout(min)>]を使用した例

```
remove-session -server server1 --timeout 3
```

4.2.9.4. show-session-configuration

セッション関連の設定情報を照会します。

- 関連スキーマ

jeus-session-server.xsd, jeus-session-domain.xsd

- エイリアス

getsc, ssc

- 使用方法

```
show-session-configuration [-basic]
                             [-common]
                             [-server <server-name>]
                             [-cluster <cluster-name>] [-specific <cluster-name>]
                             [-h, --help]
```

- パラメータ

パラメータ	説明
[-basic]	ドメイン全体に適用される設定オプションです。 クラスター・モードとプロバイダーなどに対する設定が存在します
[-common]	共通して設定できる分散型セッション・サーバーの設定を確認します
[-server <server-name>]	Webエンジンに設定されているセッション設定を確認します
[-cluster <cluster-name>]	設定されたクラスターの分散型セッション・サーバーの設定を確認します
[-specific <cluster-name>]	特別定義スコープの分散型セッション・サーバーの設定を確認します
[-h, --help]	パラメータのないコマンドと同じであり、全体の設定構造およびパラメータを説明します

- 例

– [-server <server-name>]オプションを使用した例

```
show-session-configuration -server server1
```

– [-cluster <cluster-name>]オプションを使用した例

```
show-session-configuration -cluster cluster1
```

4.2.9.5. modify-session-configuration

セッション関連の設定情報を変更します。

- 関連スキーマ

jeus-session-server.xsd, jeus-session-domain.xsd

- エイリアス

setsc, msc

- 使用方法

```
modify-session-configuration [-basic]
    [-common]
    [-server <server-name>]
    [-cluster <cluster-name>]
    [-specific <cluster-name>]
    [-f,--force]
    [-cm,--cluster-mode <String mode(DEFAULT or DOMAIN_WIDE)>]
    [-sp,--session-provider <String provider>]
    [-ed,--exclusive-das <boolean exclude>]
    [-se,--sticky-encoding <String mode(BASE64 or RAW)>]
    [-to,--timeout <int timeout>]
    [-mc,--max-session-count <int limit count>]
    [-sh,--shared <boolean shared>]
    [-rp,--reload-persistent <boolean reload-persistent>]
    [-tmc,--tracking-mode.cookie <boolean enable-cookie>]
    [-tmu,--tracking-mode.url <boolean enable-url>]
    [-tms,--tracking-mode.ssl <boolean enable-ssl>]
    [-scn,--session-cookie.cookie-name <string cookie-name>]
    [-scv,--session-cookie.version <int version(1 or 0)>]
    [-scd,--session-cookie.domain <string domain-scope>]
    [-scp,--session-cookie.path <string path-scope>]
    [-scm,--session-cookie.max-age <int max-age(sec)>]
    [-scs,--session-cookie.secure <boolean secure>]
    [-sch,--session-cookie.http-only <boolean http-only>]
    [-scc,--session-cookie.comment <string comment>]
    [-tn,--reserved-thread-num <int thread-num>]
    [-ct,--connection-timeout <long connection-timeout (msec)>]
    [-rt,--read-timeout <long read-timeout (msec)>]
    [-fb,--allow-fail-back <boolean allow>]
    [-fc,--ignore-flow-control <boolean ignore>]
    [-bl,--backup-level <string backup-level>]
    [-bu,--backup-unit-size <int unit-size>]
    [-bq,--backup-queue-size <int queue-size>]
    [-pm,--prevent-migration <boolean prevent>]
    [-fd,--failover-delay <long failover delays(s)>]
    [-rd,--restart-delay <long restart delays(s)>]
    [-lmp,--login-manager.primary <string primary login manager>]
    [-lms,--login-manager.secondary <string secondary login manager>]
    [-pfp,--passivation.file-path <string file-path>]
    [-pto,--passivation.timeout <long passivation-timeout (msec)>]
    [-pct,--passivation.count-threshold <int count-threshold>]
```

```

[-pmt,--passivation.memory-threshold <int memory-threshold (byte)>]
[-pfl,--passivation.file-limit <int limit-count>]
[-ppr,--passivation.ratio <float passivation-ratio>]
[-add,--add-config]
[-remove,--remove-config]

```

● パラメータ

パラメータ	説明
[-basic]	ドメイン全体に適用される設定オプションです。 クラスター・モードとプロバイダーなどに対する設定を変更します
[-common]	共通で設定できる分散型セッション・サーバーの設定を変更します
[-server <server-name>]	Webエンジンに設定されているセッション設定を変更します
[-cluster <cluster-name>]	設定されたクラスターの分散型セッション・サーバーの設定を変更します
[-specific <cluster-name>]	特別定義スコープの分散型セッション・サーバーの設定を変更します
[-f, --forceLock]	設定を強制的に変更して反映します
[-cm,--cluster-mode <String mode(DEFAULT or DOMAIN_WIDE)>]	使用するクラスター・モードを選択します <ul style="list-style-type: none"> – DEFAULT – DOMAIN_WIDE
[-sp,--session-provider <String provider>]	使用したセッション・マネージャー・プロバイダーを設定します。 RUNTIMEまたはJEUSを設定するか、使用したパッケージを設定します
[-ed,--exclusive-das <boolean exclude>]	DASをクラスターに参加させないように設定します。DOMAIN_WIDEが選択された場合に適用されます
[-se,--sticky-encoding <String mode(BASE64 or RAW)>]	スティッキー・ルーティングを実行時にIDをエンコーディングする方法を設定します <ul style="list-style-type: none"> – BASE64 – RAW
[-to,--timeout <int timeout>]	session-configのセッション・タイムアウトを設定します
[-mc,--max-session-count <int maxCount>]	保存するセッションの最大数を設定します
[-sh,--shared <boolean shared>]	session-configのSHAREDモードを設定します

パラメータ	説明
<code>[-rp,--reload-persistent <boolean reload-persistent>]</code>	session-configのReload Persistentを設定します
<code>[-tmc,--tracking-mode.cookie <boolean enable-cookie>]</code>	session-configのトラッキング・モードとしてクッキーを設定します
<code>[-tmu,--tracking-mode.url <boolean enable-url>]</code>	session-configのトラッキング・モードとしてURLリライティングを設定します
<code>[-tms,--tracking-mode.ssl <boolean enable-ssl>]</code>	session-configのトラッキング・モードとしてSSLを設定します
<code>[-scn,--session-cookie.cookie-name <string cookie-name>]</code>	セッション・クッキーの名前を指定します
<code>[-scv,--session-cookie.version <int version(1 or 0)>]</code>	セッション・クッキーのコンパイル・バージョンを選択します
<code>[-scd,--session-cookie.domain <string domain-scope>]</code>	セッション・クッキーのドメイン範囲を決定します
<code>[-scp,--session-cookie.path <string path-scope>]</code>	セッション・クッキーのパス範囲を決定します
<code>[-scm,--session-cookie.max-age <int max-age>]</code>	ブラウザでセッション・クッキーの維持時間を指定します
<code>[-scs,--session-cookie.secure <boolean secure>]</code>	ブラウザがクッキーをSSLに制限して転送するか否かを設定します
<code>[-sch,--session-cookie.http-only <boolean http-only>]</code>	ブラウザがHTTP要求によってのみセッション・クッキーを使用するか否かを設定します
<code>[-scc,--session-cookie.comment <string comment>]</code>	クッキー・バージョンが1であるとき、該当セッション・クッキーの説明を設定します
<code>[-tn,--reserved-thread-num <int thread-num>]</code>	分散型セッション・サーバーに割り当てられた要求処理スレッド数の最小値を強制します
<code>[-ct,--connection-timeout <long connection-timeout>]</code>	分散型セッション・サーバーのソケット・コネクションを初期化するタイムアウトを設定します
<code>[-rt,--read-timeout <long read-timeout>]</code>	分散型セッション・サーバーの要求に対する応答タイムアウトを設定します
<code>[-fb,--allow-fail-back <boolean allow>]</code>	分散型セッション・サーバーのフェイルバック機能を使用します
<code>[-fc,--ignore-flow-control <boolean ignore>]</code>	バックアップ・キューがいっぱいになっても、セッションの消失よりサービスが優先されるため、フロー・コントロールを行いません
<code>[-bl,--backup-level <string backup-level>]</code>	分散型セッション・サーバーのアップデート基準を決定します

パラメータ	説明
<code>[-bu,--backup-unit-size <int unit-size>]</code>	負荷がある場合、バックアップを一度に実行する最大のセッション数を設定します
<code>[-bq,--backup-queue-size<int queue-size>]</code>	負荷がある場合、キューに溜められるバックアップ・ユニットの最大数を設定します
<code>[-pm,--prevent-migration <boolean prevent>]</code>	性能が良くなくても、同時要求によるセッションの消失を最小化するためにマイグレーションを実行しないように設定します
<code>[-fd,--failover-delay <long failover delays(s)>]</code>	Webエンジンに障害が発生したとき、該当エンジンを除く残りのエンジンで再度クラスタリング接続を確立するまでのタイムアウト値です
<code>[-rd,--restart-delay <long restart delays(s)>]</code>	Webエンジンを正常に終了したとき、該当エンジンを除く残りのエンジンで再度クラスタリング接続を確立するまでのタイムアウト値です
<code>[-lmp,--login-manager.primary <string primary>]</code>	JEUS Login Managerのプライマリー・マネージャーを設定します
<code>[-lms,--login-manager.secondary <string secondary>]</code>	JEUS Login Managerのセカンダリー・マネージャーを設定します
<code>[-pfp,--passivation.file-path <string file-path>]</code>	パッシベーションを設定するとき、該当ファイルが保存された絶対パスを設定します
<code>[-pto,--passivation.timeout <long passivation-timeout (mesc)>]</code>	パッシベーションを実行するタイムアウトを設定します。アイドル時間がこのタイムアウト値より大きい場合、パッシベーションされます
<code>[-pct,--passivation.count-threshold <int count-threshold>]</code>	パッシベーションを実行するアクティブなセッション数を設定します。セッション数が設定値を超えると、古いセッションからパッシベーションされます
<code>[-pmt,--passivation.memory-threshold <int memory-threshold (byte)>]</code>	パッシベーションを実行するメモリー・サイズを設定します。メモリー・サイズが設定したバイト・サイズを超えると、古いセッションからパッシベーションされます
<code>[-pfl,--passivation.file-limit <int limit-count>]</code>	1つのディレクトリーに保存する最大のファイル数を設定します
<code>[-ppr,--passivation.ratio <float passivation-ratio>]</code>	セッション数やメモリー・サイズを基準にパッシベーションを実行するとき、パッシベーションする割合を設定します
<code>[-add,--add-config]</code>	サーバー・クラスターまたは特別定義スコープのクラスターの分散型セッション・サーバーの設定を新規作成します。この設定を追加する場合、Common設定は無視されます
<code>[-addScope,--add-specific-scope]</code>	特定のクラスターのスコープのみ追加します。特定のクラスターのセッション・サーバーの設定は行わず、Common設定が使用されます
<code>[-removeScope,--remove-specific-scope]</code>	特定のクラスターのスコープを削除します。特定のクラスターのセッション・サーバーの設定の有無に関係なく、該当スコープを削除します

パラメータ	説明
[-remove,--remove-config]	クラスター全体または特定のクラスターの分散型セッション・サーバーの設定を削除します。設定を削除する場合、Common設定が使用されます

- 例

- [-server <server-name>]オプションを使用した例

```
modify-session-configuration -server server1 -to 40 -sh true -scs true
```

- [-cluster <cluster-name>]オプションを使用した例

```
modify-session-configuration -cluster distribute -tn 1 -ct 3000
```

4.2.10. JMSエンジン関連コマンド

JMSエンジンに関連するコマンドの一覧は以下のとおりです。

コマンド	説明
list-jms-connection-factories	コネクション・ファクトリーの一覧、あるいは指定されたコネクション・ファクトリーの情報を出力します
add-jms-connection-factory	新しいコネクション・ファクトリーを追加します
remove-jms-connection-factory	指定されたコネクション・ファクトリーを削除します
list-jms-destinations	デスティネーションの一覧、あるいは指定されたデスティネーションの情報を出力します
add-jms-destination	新しいデスティネーションを追加します
remove-jms-destination	指定されたデスティネーションを削除します
control-jms-destination	指定されたデスティネーションのサービス状態を制御します
list-jms-clients	接続中のクライアント一覧を表示し、情報を出力します
ban-jms-client	指定されたJMSクライアントとの接続を強制的に切断します
list-jms-durable-subscriptions	永続サブスクリプションの一覧を表示します
list-jms-messages	指定されたデスティネーション内のメッセージ情報を出力します
view-jms-message	指定されたメッセージの詳細情報を表示します
move-jms-messages	指定されたデスティネーション内の指定されたメッセージを、指定されたデスティネーションに移動させます

コマンド	説明
<code>delete-jms-messages</code>	指定されたデスティネーション内の指定されたメッセージを削除します
<code>export-jms-messages</code>	指定されたデスティネーション内の指定されたメッセージをXML形式でエクスポートします
<code>import-jms-messages</code>	指定されたデスティネーションにエクスポートされたXML形式のメッセージをインポートします
<code>list-jms-pending-transactions</code>	ペンディング・トランザクションの一覧を出力します
<code>commit-jms-pending-transaction</code>	指定されたペンディング・トランザクションを強制的にコミットします

4.2.10.1. 共通適用事項

以下は、JMSエンジン関連コマンドで共通で適用されるオプションについての説明です。以下のオプションのうち1つのみ設定できます。

- `[-cluster <cluster-name>]`オプション
 - 各コマンドが適用されるクラスターの指定時に使用するオプションです。DASに接続している状態でのみ使用可能です。MSIに接続している状態では、該当サーバーにコマンドが適用されます。
 - デスティネーションに関連するコマンドでのみサポートします。
- `[-server <server-name>]`オプション
 - 各コマンドが適用されるサーバーの指定時に使用するオプションです。DASに接続している状態でのみ使用可能です。MSIに接続している状態では、該当サーバーにコマンドが適用されます。
 - デスティネーションに関連しないコマンドでのみサポートします。

4.2.10.2. list-jms-connection-factories

JMSエンジンに登録されているすべてのコネクション・ファクトリーの一覧、あるいは指定されたコネクション・ファクトリーの情報を出力します。

- エイリアス

`jmscf, jmsconf`

- 使用方法

```
list-jms-connection-factories -server <server-name>
                               [-n, --name <factory-name>]
```

- パラメータ

パラメータ	説明
-server <server-name>	コネクション・ファクトリーが属するサーバーを指定します
[-n, --name <factory-name>]	情報を出力するコネクション・ファクトリーの名前です。指定していない場合はすべてのコネクション・ファクトリーに関する簡略情報が出力されます

● 例

```
[DAS]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory Information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory             | nonxa    |
| XAConnectionFactory           | XAConnectionFactory         | xa       |
+-----+-----+-----+

[DAS]domain1.adminServer>list-jms-connection-factories -server adminServer -name
ConnectionFactory
=====
+-----+-----+-----+
| Factory Name                   | ConnectionFactory             |
| Export Name                    | ConnectionFactory             |
| Type                           | nonxa                        |
| Client ID                     | not-set                      |
| Max Client Session Threads    | 100                          |
| Clustered                     | false                       |
| Broker Selection Policy        | round-robin                  |
+-----+-----+-----+

=====

+-----+-----+-----+
|                               | Addresses                     |
+-----+-----+-----+
| 192.168.0.26:9741(JMSServiceChannel-internal) |
+-----+-----+-----+
```

=====

4.2.10.3. add-jms-connection-factory

JMSエンジンに新しいコネクション・ファクトリーを動的に追加します。キューまたはトピック・コネクション・ファクトリーを作成できます。

- エイリアス

add-connection-factory, addconf, createconf

- 使用方法

```
add-jms-connection-factory -server <server-name>
                             -type <factory-type>
                             -name <factory-name>
                             [-export <export name>]
                             [-clientid <client-id>]
```

- パラメータ

パラメータ	説明
-server <server-name>	対象になるサーバーを設定します
-type <factory-type>	コネクション・ファクトリーのタイプを決定します。 以下のうち1つを設定します。 - nonxa - xa - queue - topic - xaqueue - xatopic
-name <factory-name>	新しく作成するコネクション・ファクトリーの名前を入力します。 この名前はJMSエンジン内で他のコネクション・ファクトリーと重複しないように注意します
[-export <export name>]	新しく作成するコネクション・ファクトリーのJNDIの名前を入力します。

パラメータ	説明
	この名前はJEUS JNDIサーバー内で他のオブジェクトと重複しないように注意します
[-clientid <client-id>]	コネクション・ファクトリーから作成したコネクションに付与するクライアントIDを入力します

- 例

```
[DAS]domain1.adminServer>add-jms-connection-factory -server adminServer -type
nonxa -name MyConnectionFactory
Executed Successfully
View the results using 'list-jms-connection-factories -server adminServer -name
MyConnectionFactory'
[DAS]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory Information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory             | nonxa    |
| MyConnectionFactory          | MyConnectionFactory         | nonxa    |
| XAConnectionFactory           | XAConnectionFactory         | xa       |
+-----+-----+-----+
=====
```

4.2.10.4. remove-jms-connection-factory

JMSエンジンで特定のコネクション・ファクトリーを動的に削除します。

- エイリアス

remove-connection-factory, deleteconf, removeconf

- 使用方法

```
remove-jms-connection-factory -server <server-name>
                             -name <factory-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	コネクション・ファクトリーが属するサーバーを指定します
-name <factory-name>	削除するコネクション・ファクトリーの名前です

- 例

```
[DAS]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory              | nonxa    |
| MyConnectionFactory           | MyConnectionFactory           | nonxa    |
| XAConnectionFactory           | XAConnectionFactory           | xa       |
+-----+-----+-----+
=====

[DAS]domain1.adminServer>remove-jms-connection-factory -server adminServer -name
MyConnectionFactory
Executed Successfully
View the results using 'list-jms-connection-factories -server adminServer'
[DAS]domain1.adminServer>list-jms-connection-factories -server adminServer
Connection Factory information
=====
+-----+-----+-----+
|          Factory Name          |          Export Name          |   Type   |
+-----+-----+-----+
| ConnectionFactory              | ConnectionFactory              | nonxa    |
| XAConnectionFactory           | XAConnectionFactory           | xa       |
+-----+-----+-----+
=====
```

4.2.10.5. list-jms-destinations

JMSエンジンに登録されているすべてのデスティネーションの一覧、あるいは指定されたデスティネーションの情報を出力します。

- エイリアス

jmsdest, dest

● 使用方法

```
list-jms-destinations -cluster <cluster-name> | -server <server-name>
                        [-n, --name <destination-name>]
```

● パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	デスティネーションが属するクラスターまたはサーバーを指定します
[-n, --name <destination-name>]	情報を出力するデスティネーションの名前です。指定していない場合はすべてのデスティネーションに関する簡略情報が出力されます

● 例

```
[DAS]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
| Name | Export | Type | Remaining | Dead Letter | Produce | Consume |
|      |        |      |            |              |         |         |
|      | Name  |      | Messages  | Destination | Suspended | Suspended |
|      |      |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Topic |          0 | JEUSMQ_DLQ | false  | false  |
|      |      |      |            |              |         |         |
| Topic   | Topic   |      |            |              |         |         |
|      |      |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Queue |          0 | JEUSMQ_DLQ | false  | false  |
|      |      |      |            |              |         |         |
| Queue   | Queue   |      |            |              |         |         |
|      |      |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue |          0 | JEUSMQ_DLQ | false  | false  |
|      |      |      |            |              |         |         |
| LQ      | LQ      |      |            |              |         |         |
|      |      |      |            |              |         |         |
+-----+-----+-----+-----+-----+-----+-----+

For detailed information, use the -name option
=====
```

```
[DAS]domain1.adminServer>list-jms-destinations -server adminServer -name
ExamplesQueue
Detailed destination information in Server adminServer
=====
ExamplesQueue

+-----+-----+
| Export Name           | ExamplesQueue |
| Type                  | Queue         |
| Dead Letter Destination | JEUSMQ_DLQ    |
| Consumer Count        | 0             |
| Processed Messages    | 0             |
| Remaining Messages    | 0             |
| Pending Messages      | 0             |
| Dispatched Messages   | 0             |
| Delivered Messages    | 0             |
| Expired Messages      | 0             |
| Moved Messages       | 0             |
| Memory Usage (current) | 0kb           |
| Memory Usage (high mark) | 0kb           |
| Produce Suspended      | false         |
| Consume Suspended      | false         |
+-----+-----+
=====
```

4.2.10.6. add-jms-destination

JMSエンジンに新しいデスティネーションを動的に追加します。

- エイリアス

add-destination, adddest, createdest

- 使用方法

```
add-jms-destination -cluster <cluster-name | -server <server-name>
                    -type <destination-type>
                    -name <destination-name>
                    [-export <export-name>]
```

- パラメータ

パラメータ	説明
-cluster <cluster-name -server <server-name>	デスティネーションが属するクラスターまたはサーバーを指定します
-type <destination-type>	デスティネーションのタイプを決定します(キューまたはトピック)

パラメータ	説明
-name <destination-name>	新しく作成するデスティネーションの名前を入力します。 この名前は、JMSエンジン内で他のデスティネーションと重複しないように注意してください
[-export <export-name>]	新しく作成するデスティネーションのJNDIの名前を入力します。 この名前は、JEUS JNDIサーバー内で他のオブジェクトと重複しないように注意してください。指定していない場合、destination-nameと同じ値が指定されます

● 例

```
[DAS]domain1.adminServer>add-jms-destination -server adminServer -type queue
-name MyQueue
Executed Successfully
View the results using 'list-jms-destinations -server adminServer -name MyQueue'
[DAS]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
| Name | Export | Type | Remaining | Dead Letter | Produce | Consume |
|      |        |      | Messages | Destination | Suspended | Suspended |
|      | Name  |      |           |              |           |           |
+-----+-----+-----+-----+-----+-----+-----+
| MyQueue | MyQueue | Queue | 0 | JEUSMQ_DLQ | false | false |
|      |      |      |   |           |       |       |
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Topic | 0 | JEUSMQ_DLQ | false | false |
|      |      |      |   |           |       |       |
| Topic   | Topic   |      |   |           |       |       |
|      |      |      |   |           |       |       |
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Queue | 0 | JEUSMQ_DLQ | false | false |
|      |      |      |   |           |       |       |
| Queue   | Queue   |      |   |           |       |       |
|      |      |      |   |           |       |       |
+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue | 0 | JEUSMQ_DLQ | false | false |
|      |      |      |   |           |       |       |
| LQ      | LQ      |      |   |           |       |       |
|      |      |      |   |           |       |       |
+-----+-----+-----+-----+-----+-----+-----+
```

For detailed information, use the `-name` option

=====

4.2.10.7. remove-jms-destination

JMSエンジンから特定のデスティネーションを動的に削除します。

- エイリアス

`remove-destination`, `deletedest`, `removedest`

- 使用方法

```
remove-jms-destination -cluster <cluster-name> | -server <server-name>
                        -name <destination-name>
```

- パラメータ

パラメータ	説明
<code>-cluster <cluster-name> -server <server-name></code>	デスティネーションが属するクラスターまたはサーバーを指定します
<code>-name <destination-name></code>	削除するデスティネーションの名前です

- 例

```
[DAS]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
| Name   | Export | Type  | Remaining | Dead Letter | Produce | Consume |
|        |        |       |           |             |         |         |
|        | Name   |       | Messages  | Destination | Suspended | Suspended |
|        |        |       |           |             |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| MyQueue | MyQueue | Queue | 0 | JEUSMQ_DLQ | false | false |
|        |        |       |   |             |       |       |
+-----+-----+-----+-----+-----+-----+-----+
| Examples | Examples | Topic | 0 | JEUSMQ_DLQ | false | false |
|          |          |       |   |             |       |       |
| Topic    | Topic    |       |   |             |       |       |
|          |          |       |   |             |       |       |
+-----+-----+-----+-----+-----+-----+-----+
| Examples | Examples | Queue | 0 | JEUSMQ_DLQ | false | false |
|          |          |       |   |             |       |       |
| Queue    | Queue    |       |   |             |       |       |
|          |          |       |   |             |       |       |
+-----+-----+-----+-----+-----+-----+-----+
```

```

+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue |          0 | JEUSMQ_DLQ | false | false |
|
| LQ      | LQ      |        |          |          |        |        |
|
+-----+-----+-----+-----+-----+-----+-----+

For detailed information, use the -name option
=====
[DAS]domain1.adminServer>
remove-jms-destination -server adminServer -name MyQueue
Executed Successfully
View the results using 'list-jms-destinations -server adminServer'

[DAS]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
|  Name  | Export | Type | Remaining | Dead Letter | Produce | Consume |
|
|        | Name   |      | Messages  | Destination | Suspended | Suspended |
|
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Topic |          0 | JEUSMQ_DLQ | false | false |
|
| Topic   | Topic   |        |          |          |        |        |
|
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Queue |          0 | JEUSMQ_DLQ | false | false |
|
| Queue   | Queue   |        |          |          |        |        |
|
+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue |          0 | JEUSMQ_DLQ | false | false |
|
| LQ      | LQ      |        |          |          |        |        |
|
+-----+-----+-----+-----+-----+-----+-----+

For detailed information, use the -name option
=====

```

4.2.10.8. control-jms-destination

JMSエンジンで特定のデスティネーションの状態を制御します。

- エイリアス

destctrl

- 使用方法

```
control-jms-destination -cluster <cluster-name> | -server <server-name>
                        -dest <destination-name>
                        -suspend | -resume
                        [-produce]
                        [-consume]
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	デスティネーションが属するクラスターまたはサーバーを指定します
-dest <destination-name>	制御するデスティネーションの名前です
-suspend -resume	制御するデスティネーションの状態です – suspend : 該当デスティネーションのサービスが一時的に停止します – resume : 該当デスティネーションのサービスが再開します
[-produce]	指定されたデスティネーションのproduceを制御するか否かを指定します
[-consume]	指定されたデスティネーションのconsumeを制御するか否かを指定します

- 例

```
[DAS]domain1.adminServer>control-jms-destination -server adminServer -dest
ExamplesQueue -suspend -produce
Controlling destination is successful in ExamplesQueue : [production] is suspended
[DAS]domain1.adminServer>control-jms-destination -server adminServer -dest
ExamplesQueue -resume -produce
Controlling destination is successful in ExamplesQueue : [production] is resumed
```

4.2.10.9. list-jms-clients

現在接続している全クライアントの一覧を出力します。

- エイリアス

jmsclient, jmsentry

- 使用方法

```
list-jms-clients -server <server-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	クライアントの一覧を出力するサーバーを指定します

- 例

```
[DAS]domain1.adminServer>list-jms-clients -server adminServer
JEUS MQ client information
=====
+-----+-----+-----+-----+-----+
| Entry Name | Remote Address | Start Time | Connection | Session |
|           |               |           |           |         |
|           |               |           | Count     | Count   |
|           |               |           |           |         |
+-----+-----+-----+-----+-----+
| JMSClient-RE63 | 192.168.34.33/192.168 | Tue Mar 19 | 1 | 1 |
|           |               |           |   |   |
| 675900002      | .34.33:49490        | 16:14:32 KST 2013 |   |   |
|           |               |           |   |   |
+-----+-----+-----+-----+-----+
| JMSClient-RE63 | 192.168.34.33/192.168 | Tue Mar 19 | 1 | 1 |
|           |               |           |   |   |
| 675900001      | .34.33:49484        | 16:14:23 KST 2013 |   |   |
|           |               |           |   |   |
+-----+-----+-----+-----+-----+
=====
```

4.2.10.10. ban-jms-client

JMSクライアントとの接続を強制的に切断します。

- エイリアス

killentry, banentry

- 使用方法

```
ban-jms-client -server <server-name>
               -name <entry-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	対象になるサーバーを指定します
-name <entry-name>	強制的に接続を切断するJMSクライアントの名前を指定します。この名前はlist-jms-clientsコマンドで確認できます

4.2.10.11. list-jms-durable-subscriptions

JMSエンジンに登録されているすべての永続サブスクリプションの情報、あるいは指定された永続サブスクリプションの情報を出力します。

- エイリアス

durable, dur

- 使用方法

```
list-jms-durable-subscriptions -cluster <cluster-name> | -server <server-name>
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	永続サブスクリプションの一覧を出力するサーバーを指定します

- 例

```
[DAS]domain1.adminServer>list-jms-durable-subscriptions -server adminServer
Durable Subscription information
```

```
=====
+-----+-----+-----+-----+
| Durable Name | Client ID | Message Selector | Remaining Messages |
+-----+-----+-----+-----+
| testDurable | testClient | | 5 |
+-----+-----+-----+-----+
=====
```

4.2.10.12. list-jms-messages

デスティネーションのメッセージの情報を出力します。

- エイリアス

message, msg

- 使用方法

```
list-jms-messages -cluster <cluster-name> | -server <server-name>
                    -dest <destination-name> | -durable <<client-id>_<durable-name>>

                    [-s,--selector <message-selector>]
                    [-offset <offset>]
                    [-size <size>]
                    [-id <message-id-pattern>]
                    [-type <message-type>]
                    [-from <YYYY:MM:DD:HH:MM:SS>]
                    [-to <YYYY:MM:DD:HH:MM:SS>]
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	デスティネーションや永続的サブスクライバーが属するクラスターやサーバーを指定します
-dest <destination-name> -durable <<client-id>_<durable-name>>	出力するメッセージがあるデスティネーションや永続的サブスクライバーの名前です
[-s, --selector <message-selector>]	出力するメッセージを選択できるメッセージ・セレクターを指定します
[-offset <offset>]	メッセージを分けて出力する際に、何個目のメッセージから出力するのかを指定します
[-size <size>]	メッセージを分けて出力する際に、1度に出力するメッセージの数を指定します
[-id <message-id-pattern>]	出力するメッセージを選択できるメッセージIDのパターンを指定します
[-type <message-type>]	出力するメッセージを選択できるメッセージのタイプを指定します
[-from <YYYY:MM:DD:HH:MM:SS>]	指定した時点以降に入ってきたメッセージを選択します
[-to <YYYY:MM:DD:HH:MM:SS>]	指定した時点以前に入ってきたメッセージを選択します

- 例

```
[DAS]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
Messages in Destination ExamplesQueue.

=====
+-----+-----+-----+
|      Message ID      | Message Type |      Created Time      |
+-----+-----+-----+
```

ID:0:36386800005:2:2	Text	Mon May 07 20:10:01 KST 2012
ID:0:36386800006:2:2	Text	Mon May 07 20:10:03 KST 2012
ID:0:36386800007:2:2	Text	Mon May 07 20:10:04 KST 2012
ID:0:36386800008:2:2	Text	Mon May 07 20:10:06 KST 2012

=====

4.2.10.13. view-jms-message

デスティネーション内の特定メッセージの詳細情報を出力します。

- エイリアス

viewmsg

- 使用方法

```
view-jms-messages -cluster <cluster-name> | -server <server-name>
                  -dest <destination-name> | -durable <<client-id>_<durable-name>>

                  -id <message-id>
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	デスティネーションや永続的サブスクライバーが属するクラスターやサーバーを指定します
-dest <destination-name> -durable <<client-id>_<durable-name>>	情報を出力するデスティネーションや永続的サブスクライバーの名前です
-id <message-id>	出力するメッセージのmessage idを指定します

- 例

```
[DAS]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
Messages in Destination ExamplesQueue
=====
+-----+-----+-----+
| Message ID | Message Type | Created Time |
+-----+-----+-----+
| ID:1:6927265:1:0 | Text | Tue Mar 19 15:53:17 KST 2013 |
| ID:1:6927265:1:1 | Text | Tue Mar 19 15:53:17 KST 2013 |
```

```

| ID:1:6927265:1:2 | Text          | Tue Mar 19 15:53:17 KST 2013
|
| ID:1:6927265:1:3 | Text          | Tue Mar 19 15:53:17 KST 2013
|
| ID:1:6927265:1:4 | Text          | Tue Mar 19 15:53:17 KST 2013
|
+-----+-----+-----+
=====

[DAS]domain1.adminServer>view-jms-message -server adminServer -dest ExamplesQueue
-id ID:1:6927265:1:0
Message ID:1:6927265:1:0
=====
+-----+-----+-----+
| Attribute name | Value
|
+-----+-----+-----+
| Message ID     | ID:1:6927265:1:0
|
| Destination Name | ExamplesQueue
|
| Delivery Mode    | 2
|
| Message Type     | Text
|
| Correlation ID   |
|
| Created Time     | Tue Mar 19 15:53:17 KST 2013
|
| Expiration Time  | -
|
| Time-to-live     | 0
|
| Priority         | 4
|
| Redelivered      | false
|
| Delivery Time    | Tue Mar 19 15:53:17 KST 2013
|
| Redelivery Limit | 4
|
| Reply To        |
|
| Message Body     | Test Message
|
+-----+-----+-----+
=====

```

4.2.10.14. move-jms-messages

指定されたメッセージをクラスターやサーバー内の他のデスティネーションに移動します。

- エイリアス

mvmsg

- 使用方法

```
move-jms-message -cluster <cluster-name> | -server <server-name>
                  -dest <destination-name> | -durable <<client-id>_<durable-name>>

                  -id <message-id> | -all
                  -target <destination-name>
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	デスティネーションや永続的サブスクライバーが属するクラスターやサーバーを指定します
-dest <destination-name> -durable <<client-id>_<durable-name>>	情報を出力するデスティネーションや永続的サブスクライバーの名前です
-id <message-id> -all	移動するメッセージを指定します。個別のメッセージIDあるいは全体を指定できます
-target <destination-name>	メッセージの移動先となるデスティネーションを指定します

- 例

```
[DAS]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
| Name   | Export | Type  | Remaining | Dead Letter | Produce | Consume |
|        |        |       |           |             |         |         |
|        | Name   |       | Messages  | Destination | Suspended | Suspended |
|        |        |       |           |             |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| MyQueue | MyQueue | Queue | 0         | JEUSMQ_DLQ  | false  | false  |
|        |        |       |           |             |         |         |
+-----+-----+-----+-----+-----+-----+-----+
| Examples | Examples | Topic | 0         | JEUSMQ_DLQ  | false  | false  |
|        |        |       |           |             |         |         |
| Topic   | Topic   |       |           |             |         |         |
```

```

|
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Queue |          5 | JEUSMQ_DLQ | false   | false
|
| Queue   | Queue   |        |          |             |         |
|
+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue |          0 | JEUSMQ_DLQ | false   | false
|
| LQ      | LQ      |        |          |             |         |
|
+-----+-----+-----+-----+-----+-----+-----+

For detailed information, use the -name option
=====
[DAS]domain1.adminServer>move-jms-messages -server adminServer -dest ExamplesQueue
-all -target MyQueue
Successfully moved all messages from ExamplesQueue to MyQueue is successful
[DAS]domain1.adminServer>list-jms-destinations -server adminServer
Destination information in Server adminServer
=====
+-----+-----+-----+-----+-----+-----+-----+
| Name   | Export | Type | Remaining | Dead Letter | Produce | Consume
|
|        | Name   |      | Messages  | Destination | Suspended | Suspended
|
+-----+-----+-----+-----+-----+-----+-----+
| MyQueue | MyQueue | Queue |          5 | JEUSMQ_DLQ | false   | false
|
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Topic |          0 | JEUSMQ_DLQ | false   | false
|
| Topic   | Topic   |        |          |             |         |
|
+-----+-----+-----+-----+-----+-----+-----+
| Examples| Examples| Queue |          0 | JEUSMQ_DLQ | false   | false
|
| Queue   | Queue   |        |          |             |         |
|
+-----+-----+-----+-----+-----+-----+-----+
| JEUSMQ_D| JEUSMQ_D| Queue |          0 | JEUSMQ_DLQ | false   | false
|
| LQ      | LQ      |        |          |             |         |
|
+-----+-----+-----+-----+-----+-----+-----+

```

For detailed information, use the `-name` option

=====

4.2.10.15. delete-jms-messages

キューや永続的サブスクライバーに蓄積している特定のメッセージ、あるいはメッセージ全体を削除します。

- エイリアス

deletemsg, deletemessage, removemsg, removemessage, rmmsg

- 使用方法

```
delete-jms-message -cluster <cluster-name> | -server <server-name>
                    -dest <destination-name> | -durable <<client-id>_<durable-name>>

                    -id <message-id> | -all
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	削除するメッセージがあるデスティネーションや永続的サブスクライバーが属するクラスターやサーバーを指定します
-dest <destination-name> -durable <<client-id>_<durable-name>>	削除するメッセージがあるデスティネーションや永続的サブスクライバーの名前です
-id <message-id> -all	削除するメッセージを指定します。個別のIDあるいはすべてのメッセージを指定できます

4.2.10.16. export-jms-messages

指定されたデスティネーションの特定のメッセージをXML形式でエクスポートします。

- エイリアス

exmsg

- 使用方法

```
export-jms-message -cluster <cluster-name> | -server <server-name>
                    -dest <destination-name> | -durable <<client-id>_<durable-name>>

                    -path <file-path>
                    -id <message-id> | -all
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	エクスポートするメッセージがあるデスティネーションや永続的サブスクライバーが属するクラスターやサーバーを指定します
-dest <destination-name> -durable <<client-id>_<durable-name>>	エクスポートするメッセージがあるデスティネーションや永続的サブスクライバーの名前です
-path <file-path>	エクスポートするメッセージが保存されるファイルのパスを指定します
-id <message-id> -all	エクスポートするメッセージを指定します。個別のIDあるいはすべてのメッセージを指定できます

- 例

```
[DAS]domain1.adminServer>export-jms-messages -server adminServer -dest
ExamplesQueue -all -path exported.xml
Successfully exported 5 messages.
```

4.2.10.17. import-jms-messages

デスティネーションにエクスポートされたXMLファイルからメッセージをインポートします。

- エイリアス

immsg

- 使用方法

```
import-jms-message -cluster <cluster-name> | -server <server-name>
                    -dest <destination-name>
                    -path <file-path>
                    [-overwrite]
```

- パラメータ

パラメータ	説明
-cluster <cluster-name> -server <server-name>	メッセージをインポートするデスティネーションが属するクラスターやサーバーを指定します
-dest <destination-name>	メッセージをインポートするデスティネーションの名前です
-path <file-path>	インポートするメッセージが保存されたファイルのパスを指定します
[-overwrite]	インポートするメッセージのJMSMessageIDを新しく発給しません。同じIDが存在する場合は上書きします

- 例

```
[DAS]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
There are no messages in ExamplesQueue.
[DAS]domain1.adminServer>import-jms-messages -server adminServer -dest
ExamplesQueue -path exported.xml
Successfully imported 5 of 5 messages.
[DAS]domain1.adminServer>list-jms-messages -server adminServer -dest ExamplesQueue
Messages in Destination ExamplesQueue
=====
+-----+-----+-----+
|      Message ID      | Message Type |      Created Time      |
|-----+-----+-----+
| ID:1:1239036099:1:5   | Text         | Tue Mar 19 19:52:54 KST 2013 |
|-----+-----+-----+
| ID:1:1239036099:1:6   | Text         | Tue Mar 19 19:52:54 KST 2013 |
|-----+-----+-----+
| ID:1:1239036099:1:7   | Text         | Tue Mar 19 19:52:54 KST 2013 |
|-----+-----+-----+
| ID:1:1239036099:1:8   | Text         | Tue Mar 19 19:52:54 KST 2013 |
|-----+-----+-----+
| ID:1:1239036099:1:9   | Text         | Tue Mar 19 19:52:54 KST 2013 |
|-----+-----+-----+
=====
```

4.2.10.18. list-jms-pending-transactions

インダウト状態のトランザクション一覧を出力します。

- エイリアス

jmspt, pt

- 使用方法

```
list-jms-pending-transactions -server <server-name>
```

- パラメータ

パラメータ	説明
-server <server-name>	インダウト状態のトランザクションを出力するサーバーを指定します

4.2.10.19. commit-jms-pending-transaction

ペンディング・トランザクションを強制的にコミットします。

- エイリアス

jmsptcommit, ptcommit

- 使用方法

```
commit-jms-pending-transaction -server <server-name>
                                -id <tx-id>
```

- パラメータ

パラメータ	説明
-server <server-name>	対象になるサーバーを指定します
-id <tx-id>	強制的にコミットするトランザクションのIDです。IDは list-jms-pending-transactions コマンドで確認できます

4.2.11. データソース関連コマンド

データソースの管理および設定の変更や確認などの機能を提供します。コマンドの一覧は以下のとおりです。

コマンド	説明
add-data-source	データソースをドメインに動的に追加します
remove-data-source	データソースをドメインから動的に削除します
modify-data-source	データソースの設定を変更します。一部の設定は動的に変更可能です
list-data-sources	ドメインに存在する全データソースの一覧を出力します。データソースのIDを指定した場合は、該当データソースの詳細設定を出力します
test-data-sources-config	データソースの設定が正しいか否かを確認します
add-cluster-data-source	クラスターのデータソースをドメインに動的に追加します
remove-cluster-data-source	クラスターのデータソースをドメインから動的に削除します
modify-cluster-data-source	クラスターのデータソースの設定を変更します。大部分の設定が動的に変更可能です
list-cluster-data-sources	ドメインに存在する全クラスターのデータソースの一覧を出力します。データソースのIDを指定した場合は、該当クラスターのデータソースの詳細設定を出力します

4.2.11.1. add-data-source

データソースをドメインに動的に追加します。

- エイリアス

addds

- 使用方法

```
add-data-source [-id,--dataSourceID <data-source-id>]
                [-ac,--autoCommit <auto-commit-value>]
                [-sqt,--stmtQueryTimeout <statement-query-timeout>]
                [-aocl,--actionOnConnectionLeak <action-on-connection-leak>]
                [-min,--min <minimum-size>]
                [-max,--max <maximum-size>]
                [-step,--step <resizing-step>]
                [-period,--period <resizing-period>]
                [-ew,--enableWait <enable-wait>]
                [-wt,--waitTime <wait-time>]
                [-dds,--delegationDataSource <delegation-data-source>]
                [-muc,--maxUseCount <max-use-count>]
                [-cq,--checkQuery <check-query>]
                [-cqt,--checkQueryTimeout <check-query-timeout>]
                [-nvi,--nonValidationInterval <non-validation-interval>]
                [-cqp,--checkQueryPeriod <check-query-period>]
                [-cqrc,--checkQueryRetrialCount <check-query-retrial-count>]
                [-dpocq,--destroyPolicyOnCheckQuery <destroy-policy-on-check-query>]

                [-scs,--stmtCachingSize <stmt-caching-size>]
                [-sfs,--stmtFetchSize <stmt-fetch-size>]
                [-ct,--connectionTrace <connection-trace>]
                [-gct,--getConnectionTrace <get-connection-trace>]
                [-act,--autoCommitTrace <auto-commit-trace>]
                [-ust,--useSqlTrace <use-sql-trace>]
                [-kcho,--keepConnectionHandleOpen <keep-connection-handle-open>]
                [-en,--exportName <export-name>]
                [-dscn,--dataSourceClassName <data-source-class-name>]
                [-dst,--dataSourceType <data-source-type>]
                [-vendor,--vendor <vendor>]
                [-desc,--description <description>]
                [-sn,--serverName <server-name>]
                [-pn,--portNumber <port-number>]
                [-dn,--databaseName <database-name>]
                [-user,--user <user>]
                [-pw,--password <password>]
                [-lt,--loginTimeout <login-timeout>]
                [-il,--isolationLevel <isolation-level>]
                [-pdt,--poolDestroyTimeout <pool-destroy-timeout>]
                [-prop,--property <property>]
                [-sxe,--supportXaEmulation <support-xa-emulation>]
                [-ddba,--delegationDbName <delegation-dba>]
                [-dbat,--dbaTimeout <dba-timeout>]
```

```

[-cqc,--checkQueryClass <check-query-class>]
[-is,--initSql <init-sql>]
[-f,--forceLock]
[-detail]

```

● パラメータ

パラメータ	説明
[-id,--dataSourceID <data-source-id>]	データソースのIDです。データソースIDは、1つのドメインでデータソースの一意の識別子として動作するように設定します
[-ac,--autoCommit <auto-commit-value>]	<p>データソースの自動コミット設定で、以下のうちから1つを設定します。</p> <ul style="list-style-type: none"> – TRUE – FALSE – DRIVER : DRIVERに設定するとJEUSは自動コミット設定に関与せず、JDBCドライバーの自動コミット設定に従います。XAエミュレーション機能を使用するコネクション・プールのデータソースまたはXAデータソースの場合、トランザクションが連動していない場合にのみ、この設定値が有効です
[-sq,--stmtQueryTimeout <statement-query-timeout>]	<p>データソースから取得したコネクションを通じて作成されたjava.sql.Statementオブジェクトに指定されるクエリー・タイムアウトを設定します。</p> <p>JEUSはJDBC APIを通じて定義され、JDBCドライバーのベンダーで実装したクエリー・タイムアウトの設定メソッド(java.sql.Statement#setQueryTimeout)を呼び出すだけです。JDBCドライバーのベンダー別にクエリー・タイムアウトの適用に若干の違いがあります(単位 : ms)</p>
[-aocl,--actionOnConnectionLeak <action-on-connection-leak>]	<p>アプリケーション(主にステートレス・コンポーネント・サーブレット/JSP、ステートレス・セッションBean、MDB)で使用し、JDBCコネクションのリークを検出した場合のJEUSのアクションを定義します。</p> <p>以下の値のうち1つを設定します。設定していない場合はサーバーの設定に従います。</p> <ul style="list-style-type: none"> – NO_ACTION : 何もアクションを起こしません – WARNING : 返されていないJDBCコネクション情報をログに残します – AUTO_CLOSE : 返されていないJDBCコネクション情報をログに残し、該当JDBCコネクションを回収します

パラメータ	説明
[-min,--min <minimum-size>]	コネクション・プールにプーリングされるコネクション数の最小値を指定します
[-max,--max <maximum-size>]	コネクション・プールにプーリングされるコネクション数の最大値を指定します
[-step,--step <resizing-step>]	コネクション・プールにコネクションが不足した場合、現在のコネクション数が最大値以下であればDBからコネクションを新しく受け取って設定します。この際に新しく受け取るコネクションの数を指定します
[-period,--period <resizing-period>]	コネクション・プールのサイズを最小値に合わせて調整する周期を設定します。コネクション・プールのサイズが最小値を超えた場合、使用しないコネクションは閉じます。また、コネクション・プールのサイズが最小値に及ばない場合、DBに接続するコネクションを新しく受け取って設定します(単位: ms)
[-ew,--enableWait <enable-wait>]	<p>コネクション・プールに使用可能なコネクションがなく、コネクションを増やすこともできない場合にコネクションの要求を処理する方法を決定します。</p> <ul style="list-style-type: none"> – true: 利用可能なコネクションを取得するために待機します – false: 新規コネクションを作成して提供しますが、そのコネクションは返されたときにプーリングされずに破棄されます。これを使い捨て(disposable)コネクションと言います
[-wt,--waitTime <wait-time>]	--enable-waitがtrueの場合にのみ有効な設定で、コネクションを取得するために待機する限界時間を示します。この時間が過ぎてもコネクションを取得できなかった場合、JEUSはタイムアウト例外を発生させます(単位: ms)
[-dds,--delegationDataSource <delegation-data-source>]	<p>トランザクションと連動していない状態では、XAデータソースを通じてコネクションを取得するより、コネクション・プールのデータソースを通じてコネクションを取得することを推奨します。</p> <p>機能上の差がなく、トランザクションを連動するための機能を含むXAコネクションはシステムに負担を掛けやすいためです。そのため、XAデータソースの場合は、この設定を通じてトランザクションと連動していない状態でのコネクション要求を委任するコネクション・プールのデータソースを指定します。</p> <p>一方、Oracle、DB2などでXAコネクションをトランザクションなしでも使用します。トランザクションと連動しながら使用すると、XAを開始できない例外が発生することがあります。しかし、正確な原因を把握できないため、例外を回避するためにもこの設定を利用します</p>

パラメータ	説明
<code>[-muc,--maxUseCount <max-use-count>]</code>	<p>コネクションの最大使用回数です。指定した回数以上になると、新規コネクションに切り替えます。</p> <p>デフォルト値は0で、コネクションを切り替えないという意味です</p>
<code>[-cq,--checkQuery <check-query>]</code>	<p>コネクション状態のチェックに使用されるクエリー（チェック・クエリー）を設定します。通常、DBとの接続の有効性のみを確認すればいいため、簡単な選択クエリーを使用することを推奨します</p>
<code>[-cqt,--checkQueryTimeout <check-query-timeout>]</code>	<p>コネクションをチェックするためにチェック・クエリーを実行した際、DBの応答がなく、ドライバが引き続き待機する状況が発生することがあります。これを避けるために、チェック・クエリーに対してクエリー・タイムアウトを適用します。</p> <p>これは、JDBC APIで定義した<code>java.sql.Statement#setQueryTimeout</code>メソッドを呼び出すことで可能です。1000msより小さい場合は0と設定されるため、注意してください(単位: ms)</p>
<code>[-nvi,--nonValidationInterval <non-validation-interval>]</code>	<p>コネクション・チェックが非常に頻繁で、オーバーヘッドが発生する場合に設定します。チェック直前の時間と最終チェック時間との差が、設定時間の間隔以内の場合、コネクション・チェックを省略します(単位: ms)。</p> <p>たとえば、この設定値が5000msの場合、コネクションの最終チェック時間からまだ5秒過ぎていないのであれば、該当コネクション・チェックは省略されたままアプリケーションに渡されます</p>
<code>[-cqp,--checkQueryPeriod <check-query-period>]</code>	<p>設定した周期ごとにコネクション・プールのコネクションをチェックし、問題のあるコネクションを削除します(単位: ms)。</p> <p>クラスターのデータソースに属するデータソースは自身の状態チェックに使用するため、必ず設定します</p>
<code>[-cqrc,--checkQueryRetrialCount <check-query-retrial-count>]</code>	<p>基本的に、<code>--destroyPolicyOnCheckQuery</code>が<code>FAILED_CONNECTION_ONLY</code>と設定された場合にコネクション・チェックが一度実行されます。</p> <p><code>--destroyPolicyOnCheckQuery</code>が<code>ALL_CONNECTIONS</code>に設定されている場合、最初のコネクション・チェックでコネクションに異常が確認されると、他のコネクションに対してもう一度コネクション・チェックが行われ、合計2回のコネクション・チェックが実行されます。</p> <p>この設定値が基本のコネクション・チェック実行回数に加わり、最終的なコネクション・チェック実行回数が決められます</p>

パラメータ	説明
<code>[-dpocq,--destroyPolicyOnCheckQuery <destroy-policy-on-check-query>]</code>	<p>コネクションが有効でないことが確認された場合、コネクション・プールにある他のコネクションに対する処理ポリシーを設定します。</p> <ul style="list-style-type: none"> – <code>FAILED_CONNECTION_ONLY</code> : 有効でないことが確認されたコネクションのみ削除します – <code>ALL_CONNTECTIONS</code> : 有効でないことが確認されたコネクションを削除し、コネクション・プールにある他のコネクションの有効性を再度確認します。それも有効でないことが確認された場合、コネクション・プールの全コネクションを削除します
<code>[-scs,--stmtCachingSize <stmt-caching-size>]</code>	<p>JDBCドライバは、アプリケーションでプリペアド・ステートメントを要求する度に、パラメータに転送されたSQL文をパースします。このパース作業が性能に影響を与えることがあるため、これを避けるために、JEUS内部でプリペアド・ステートメントをキャッシュする機能を提供します。この設定では、キャッシュするプリペアド・ステートメントの数を指定します。この機能を使用すると、コネクションを常に開いたまま使用するため、コネクションを閉じる際にドライバが行うクリア作業が行われません。</p> <p>たとえば、Oracle JDBCドライバの場合、<code>auto-commit</code>を<code>false</code>に設定して、コミットやロールバックを行わずにコネクションを閉じると、必ずコミットを行うようになっているにもかかわらず処理が行われません。JDBCドライバでステートメント・キャッシュ機能を提供する場合は、それを使用することを推奨します</p>
<code>[-sfs,--stmtFetchSize <stmt-fetch-size>]</code>	JDBCドライバのステートメントのフェッチ・サイズを設定します
<code>[-ct,--connectionTrace <connection-trace>]</code>	<p>コネクションに関連する付加情報を提供するかどうかを決定します</p> <ul style="list-style-type: none"> – <code>false</code> : <code>--getConnectionTrace</code>と<code>--autoCommitTrace</code>はすべて無効です
<code>[-gct,--getConnectionTrace <get-connection-trace>]</code>	アプリケーションが <code>java.sql.DataSource#getConnection</code> を呼び出した際のスタック・トレースを確認できるようにします
<code>[-act,--autoCommitTrace <auto-commit-trace>]</code>	<code>java.sql.Connection#setAutoCommit</code> が呼び出された際の関連ログとスタック・トレースをサーバー・ログに記録します。ただし、 <code>jeus.jdbc.connection-trace</code> ログのログ・レベルを <code>FINE</code> に設定します
<code>[-ust,--useSqlTrace <use-sql-trace>]</code>	<p>コネクション別に使用しているSQLクエリーを表示する機能です。</p> <p><code>jeus.jdbc.sql</code>ログのレベルを<code>FINE</code>に設定した場合、サーバー・ログを通じてSQLクエリーのヒストリーを確認できます。この機能を使用</p>

パラメータ	説明
	した場合、JDBCドライバのステートメントの実装をJEUSのものとして囲むため、JDBCドライバのstatementオブジェクトをキャストして使用するアプリケーションはこの機能を使用できません
[-kcho,--keepConnectionHandleOpen <keep-connection-handle-open>]	<p>trueに設定した場合、XAコネクションを使用してプールに返す際に、ハンドル(または論理的コネクション)は開いておきます。</p> <p>この機能は、DB2ユニバーサル・ドライバの場合に使用する必要があります。この機能を使用するとコネクション・ハンドルが閉じられないため、コネクションを閉じる際にドライバが行うクリア作業が行われません。</p> <p>たとえば、Oracle JDBCドライバの場合、auto-commitをfalseに設定して使用し、コミットやロールバックを行わずにコネクションを閉じると、必ずコミットを行うようになっているにもかかわらず処理が行われません</p>
[-en,--exportName <export-name>]	<p>データソースのJNDIの名前を設定します。</p> <p>互いに異なる2つのデータソースが、互いに異なるサーバーにJNDIバインドされていることを保証できるのであれば、該当データソースは互いに同じJNDI名を持つことができます。これは、任意のサーバーで同じJNDI名を持つ、互いに異なるデータソースを許容しないことを意味します。</p> <p>設定されていない場合、データソースIDをJNDI名に使用します</p>
[-dscn,--dataSourceClassName <data-source-class-name>]	JDBCドライバのデータソース・クラスの名前を設定します。パッケージ名を含む完全な形式で入力します
[-dst,--dataSourceType <data-source-type>]	<p>データソースのタイプを設定します。</p> <ul style="list-style-type: none"> DATA_SOURCE : コネクション・プーリング・サービスは提供されません CONNECTION_POOL_DATA_SOURCE : コネクション・プーリング・サービスが提供されます XA_DATA_SOURCE : コネクション・プーリング・サービスとXAの連動がサポートされます
[-vendor,--vendor <vendor>]	JDBCドライバのベンダー名を設定します
[-desc,--description <description>]	データソースについての説明です
[-sn,--serverName <server-name>]	データベースが実行されるホスト名またはIPを設定します

パラメータ	説明
[-pn,--portNumber <port-number>]	データベースのリスナーのポート番号を設定します
[-dn,--databaseName <database-name>]	データベースの名前を設定します。 Oracleの場合、データベースのSIDを設定します
[-user,--user <user>]	データベース・ユーザーのIDを設定します。トランザクション処理などのためには、十分な権限を持つ必要があります
[-pw,--password <password>]	データベース・ユーザーのパスワードを設定します。パスワードを暗号化して保存する場合は「{algorithm}ciphertext」のような形式にします
[-lt,--loginTimeout <login-timeout>]	データベースとコネクションを接続する際に、ログイン段階で待機する最大時間を設定します(単位: 秒)
[-il,--isolationLevel <isolation-level>]	java.sql.Connectionで定義するトランザクション分離レベルを設定します
[-pdt,--poolDestroyTimeout <pool-destroy-timeout>]	コネクション・プールの破棄が完了するまで待機する時間です(単位: ms)。 データソースを定義したアプリケーションをアンデプロイした際や、サーバーが停止した際にコネクション・プールは破棄されます。しかし、コネクションを削除しながらDBとネットワーク通信を行う場合はブロックされる可能性が存在し、コネクション・プールの破棄をひたすら待機するという問題が発生することがあります。こういった問題を避けるために、この設定を使用します。設定時間の分だけコネクション・プールの破棄が完了するまで待機し、アンデプロイまたはサーバーの停止を続けます
[-prop,--property <property>]	JDBCドライバー別に異なり得るプロパティの設定をすべて受け入れるために提供される、統一された設定方法です。 1つのプロパティに対して「名前:タイプ=値」の形式で入力します。 複数のプロパティが設定される場合はコンマ(,)で区分します
[-sxe,--supportXaEmulation <support-xa-emulation>]	ConnectionPoolDataSourceタイプのデータソースにのみ有効な設定です。この設定を適用した場合、ConnectionPoolDataSourceのコネクションがグローバル・トランザクション(XA)に参加するようにエミュレーションします。 JEUS 6までのLocalXADataSourceの代替オプションとして、ConnectionPoolDataSourceタイプのコネクション・プールに使用します。1つのトランザクションには1つのConnectionPoolDataSourceのみ参加できる点に注意してください

パラメータ	説明
[-ddba,--delegationDb <delegation-dba>]	<p>データベースのセッションを強制的に切断できる権限 (DBA権限) を持つデータソース (以下、委任DBAデータソース) のJNDIの名前を設定します。この設定を行ったデータソースから取得されたコネクションを利用したクエリーの実行が一定時間以上遅延すると、JEUSは委任DBAデータソースを通じて、該当コネクションと関連するDBセッションを強制的に削除するクエリーをDBに送信します。その後、アプリケーションが使用不可能になったコネクションによって発生した例外を処理し、コネクションを閉じるとJEUSはそのコネクションを削除し、DBから新しいコネクションを取得してコネクション・プールに入れます。</p> <p>現在、Tibero、Oracle、Sybaseに対してこの機能をサポートします。この機能は、JDBC 2.0以下のJDBCドライバでクエリーの実行に時間が掛かる場合に実行を中断させる方法として考案されました。しかし、JDBC 3.0またはそれ以上のバージョンを実装したJDBCドライバはjava.sql.Statement#setQueryTimeoutを実装するため、この機能を通じてDBセッションを強制的に削除するよりは、ステートメントのクエリー・タイムアウトの設定を利用することを推奨します。</p> <p>特にXAデータソースの場合は、XAの実行中にDBセッションが削除されると、XA処理に問題が発生することがあります。そのため、ステートメントのクエリー・タイムアウトとトランザクション・タイムアウトを適切に設定して使用します</p>
[-dbat,--dbaTimeout <dba-timeout>]	<p>この設定で指定した時間の間のみ、委任DBAデータソースはコネクションのクエリー実行を待機します。設定時間が過ぎると、該当コネクションに関連するDBセッションを強制的に削除するクエリーをDBに送信します。</p> <p>--delegationDbが設定された場合にのみ有効です (単位: ms)</p>
[-cqc,--checkQueryClass <check-query-class>]	<p>ユーザーや開発者がコネクション・チェック機能をカスタマイズする場合、そのために実装したクラスのパッケージ名を含んだ名前を入力します。</p> <p>この場合、そのクラスは必ずjeus.jdbc.connectionpool.JEUSConnectionCheckerインターフェースを実装します</p>
[-is,--initSql <init-sql>]	コネクションを作成後、最初に行うSQLクエリーを設定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	変更結果についての詳細情報が出力されます

- 例

```
[DAS]domain1.adminServer>addds -id ds1 -dst ConnectionPoolDataSource -dscn
oracle.jdbc.pool.OracleConnectionPoolDataSource -sn 61.77.153.4 -pn 1521 -
dn orcl -user scott -password tiger -property driverType;java.lang.String;thin
Successfully performed the ADD operation for data source [ds1] to domain.
Check the results using "add-data-source"
```

4.2.11.2. remove-data-source

データソースをドメインから動的に削除します。

- エイリアス

rmds

- 使用方法

```
remove-data-source [-id,--dataSourceID <data-source-id>]
                  [-f,--forceLock]
                  [-detail]
```

- パラメータ

パラメータ	説明
[-id,--dataSourceID <data-source-id>]	データソースIDを設定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	変更結果に関する詳細情報が出力されます。

- 例

```
[DAS]domain1.adminServer>rmds -id ds1
Successfully performed the REMOVE operation for data source [ds1] from the domain.
Check the results using "remove-data-source"
```

4.2.11.3. modify-data-source

データソースの設定を変更します。一部の設定は動的に変更可能です。

- エイリアス

modifyds

- 使用方法

```

modify-data-source -id,--dataSourceID <data-source-id>
    [-ac,--autoCommit <auto-commit-value>]
    [-sqt,--stmtQueryTimeout <statement-query-timeout>]
    [-aocl,--actionOnConnectionLeak <action-on-connection-leak>]
    [-min,--min <minimum-size>]
    [-max,--max <maximum-size>]
    [-step,--step <resizing-step>]
    [-period,--period <resizing-period>]
    [-ew,--enableWait <enable-wait>]
    [-wt,--waitTime <wait-time>]
    [-dds,--delegationDataSource <delegation-data-source>]
    [-muc,--maxUseCount <max-use-count>]
    [-cq,--checkQuery <check-query>]
    [-cqt,--checkQueryTimeout <check-query-timeout>]
    [-nvi,--nonValidationInterval <non-validation-interval>]
    [-cqp,--checkQueryPeriod <check-query-period>]
    [-cqrc,--checkQueryRetrialCount <check-query-retrial-count>]
    [-dpocq,--destroyPolicyOnCheckQuery <destroy-policy-on-check-query>]

    [-scs,--stmtCachingSize <stmt-caching-size>]
    [-sfs,--stmtFetchSize <stmt-fetch-size>]
    [-ct,--connectionTrace <connection-trace>]
    [-gct,--getConnectionTrace <get-connection-trace>]
    [-act,--autoCommitTrace <auto-commit-trace>]
    [-ust,--useSqlTrace <use-sql-trace>]
    [-kcho,--keepConnectionHandleOpen <keep-connection-handle-open>]
    [-en,--exportName <export-name>]
    [-dscn,--dataSourceClassName <data-source-class-name>]
    [-dst,--dataSourceType <data-source-type>]
    [-vendor,--vendor <vendor>]
    [-desc,--description <description>]
    [-sn,--serverName <server-name>]
    [-pn,--portNumber <port-number>]
    [-dn,--databaseName <database-name>]
    [-user,--user <user>]
    [-pw,--password <password>]
    [-lt,--loginTimeout <login-timeout>]
    [-il,--isolationLevel <isolation-level>]
    [-pdt,--poolDestroyTimeout <pool-destroy-timeout>]
    [-prop,--property <property>]
    [-sxe,--supportXaEmulation <support-xa-emulation>]
    [-ddba,--delegationDbas <delegation-dba>]
    [-dbat,--dbaTimeout <dba-timeout>]
    [-cqc,--checkQueryClass <check-query-class>]
    [-is,--initSql <init-sql>]
    [-f,--forceLock]
    [-detail]

```

● パラメータ

パラメータ	説明
<code>[-id,--dataSourceID <data-source-id>]</code>	データソースのIDを設定します。データソースIDは、1つのドメインでデータソースの一意の識別子として動作するように設定します
<code>[-ac,--autoCommit <auto-commit-value>]</code>	<p>[動的変更]データソースの自動コミット設定で、以下のうち1つを設定します。</p> <ul style="list-style-type: none"> – TRUE – FALSE – DRIVER : JEUSは自動コミット設定に関与せず、JDBCドライバの自動コミット設定に従います。 <p>XAエミュレーション機能を使用するコネクション・プールのデータソースまたはXAデータソースの場合、トランザクションが連動していない場合にのみ、この設定値が有効です</p>
<code>[-sq,--stmtQueryTimeout <statement-query-timeout>]</code>	<p>[動的変更]データソースから取得したコネクションを通じて作成された <code>java.sql.Statement</code> オブジェクトに指定されるクエリー・タイムアウトを設定します。</p> <p>JEUSはJDBC APIを通じて定義され、JDBCドライバのベンダーで実装したクエリー・タイムアウトの設定メソッド (<code>java.sql.Statement#setQueryTimeout</code>) を呼び出すだけです。JDBCドライバのベンダー別にクエリー・タイムアウトの適用に若干の違いがあります(単位 : ms)</p>
<code>[-aocl,--actionOnConnectionLeak <action-on-connection-leak>]</code>	<p>[動的変更]アプリケーション(主にステートレス・コンポーネント・サーブレット/JSP、ステートレス・セッションBean、MDB)で使用し、正常に返していないJDBCコネクションを検出した場合のJEUSのアクションを定義します。</p> <p>以下の値のうち1つを設定します。</p> <ul style="list-style-type: none"> – NO_ACTION : 何もアクションを起こしません – WARNING : 返されていないJDBCコネクション情報をログに残します – AUTO_CLOSE : 返されていないJDBCコネクション情報をログに残し、該当JDBCコネクションを回収します <p>設定していない場合はサーバーの設定に従います。設定を解除するには、オプション値に「unset」を入力します</p>

パラメータ	説明
[-min,--min <minimum-size>]	[動的変更] コネクション・プールにプーリングされるコネクションの最小値を指定します
[-max,--max <maximum-size>]	[動的変更] コネクション・プールにプーリングされるコネクションの最大値を指定します
[-step,--step <resizing-step>]	[動的変更] コネクション・プールにコネクションが不足した場合、現在のコネクション数が最大値以下であればDBからコネクションを新しく受け取って設定します。この際に新しく受け取るコネクションの数を指定します
[-period,--period <resizing-period>]	[動的変更] コネクション・プールのサイズを最小値に合わせて調整する周期を設定します。コネクション・プールのサイズが最小値を超えた場合、使用しないコネクションは閉じます。また、コネクション・プールのサイズが最小値に及ばない場合、DBに接続するコネクションを新しく受け取って設定します(単位: ms)
[-ew,--enableWait <enable-wait>]	[動的変更] コネクション・プールに使用可能なコネクションがなく、コネクションを増やすこともできない場合にコネクションの要求を処理する方法を決定します。 <ul style="list-style-type: none"> – true : 利用可能なコネクションを取得するために待機します – false : 新規コネクションを作成して提供しますが、そのコネクションは返されたときにプーリングされずに破棄されます。これを使い捨て(disposable)コネクションと言います
[-wt,--waitTime <wait-time>]	[動的変更] --enable-waitがtrueの場合にのみ有効な設定で、コネクションを取得するために待機する限界時間を示します。この時間が過ぎてもコネクションを取得できなかった場合、JEUSはタイムアウト例外を発生させます(単位 : ms)
[-dds,--delegationDataSource <delegation-data-source>]	[動的変更] トランザクションと連動していない状態では、XAデータソースを通じてコネクションを取得するより、コネクション・プールのデータソースを通じてコネクションを取得することを推奨します。 <p>機能上の差がなく、トランザクションを連動するための機能を含むXAコネクションはシステムに負担を掛けやすいためです。そのため、XAデータソースの場合は、この設定を通じてトランザクションと連動していない状態でのコネクション要求を委任するコネクション・プールのデータソースを指定します。</p> <p>一方、Oracle、DB2などでXAコネクションをトランザクションなしでも使用し、トランザクションと連動しながらも使用すると、XAを開始できない例外が発生することがあります。しかし、正確な原因を把握できないため、例外を回避するためにもこの設定を利用します。設定を解除するには、オプション値に「unset」を入力します</p>

パラメータ	説明
<code>[-muc,--maxUseCount <max-use-count>]</code>	[動的変更] コネクションの最大使用回数です。指定した回数以上になると、新規コネクションに切り替えます。デフォルト値は0で、コネクションを切り替えないという意味です
<code>[-cq,--checkQuery <check-query>]</code>	[動的変更] コネクション状態のチェックに使用されるクエリー(チェック・クエリー)を設定します。通常、DBとの接続の有効性のみを確認すればいいため、簡単な選択クエリーを使用することを推奨します。設定を解除するには、オプション値に「unset」を入力します
<code>[-cqt,--checkQueryTimeout <check-query-timeout>]</code>	[動的変更] コネクションをチェックするために、チェック・クエリーを実行した際、DBの応答がなく、ドライバーが引き続き待機する状況が発生することがあります。これを避けるために、チェック・クエリーに対してクエリー・タイムアウトを適用します。これは、JDBC APIで定義したjava.sql.Statement#setQueryTimeoutメソッドを呼び出すことで可能です。 1000msより小さい場合は0と設定されるため、注意してください(単位 : ms)
<code>[-nvi,--nonValidationInterval <non-validation-interval>]</code>	[動的変更] コネクション・チェックが非常に頻繁で、オーバーヘッドが発生する場合に設定します。 チェック直前の時間と最終チェック時間との差が、設定時間の間隔以内の場合、コネクション・チェックを省略するようにする設定です(単位 : ms)。 たとえば、この設定値が5000msの場合、コネクションの最終チェック時間からまだ5秒過ぎていないのであれば、そのコネクション・チェックは省略されたままアプリケーションに渡されます
<code>[-cqp,--checkQueryPeriod <check-query-period>]</code>	[動的変更] 設定した周期ごとにコネクション・プールのコネクションをチェックし、問題のあるコネクションを削除します(単位 : ms)。 クラスターのデータソースに属するデータソースは自身の状態チェックに使用するため、必ず設定します
<code>[-cqrc,--checkQueryRetrialCount <check-query-retrial-count>]</code>	[動的変更] 基本的に、--destroyPolicyOnCheckQueryがFAILED_CONNECTION_ONLYと設定された場合にコネクション・チェックが一度実行されます。 --destroyPolicyOnCheckQueryがALL_CONNECTIONSに設定されている場合、最初のコネクション・チェックでコネクションに異常が確認されると、他のコネクションに対してもう一度コネクション・チェックが行われ、合計2回のコネクション・チェックが実行されます。この設定値が基本のコネクション・チェック実行回数に加わり、最終的なコネクション・チェック実行回数が決められます

パラメータ	説明
<code>[-dpocq,--destroyPolicyOnCheckQuery <destroy-policy-on-check-query>]</code>	<p>[動的変更]コネクションが有効でないことが確認された場合、コネクション・プールにある他のコネクションに対する処理ポリシーを設定します。</p> <ul style="list-style-type: none"> – <code>FAILED_CONNECTION_ONLY</code> : 有効でないことが確認されたコネクションのみ削除します – <code>ALL_CONNTECTIONS</code> : 有効でないことが確認されたコネクションを削除し、コネクション・プールにある他のコネクションの有効性を再度確認します。それも有効でないことが確認された場合、コネクション・プールの全コネクションを削除します
<code>[-scs,--stmtCachingSize <stmt-caching-size>]</code>	<p>[動的変更]JDBCドライバは、アプリケーションでプリペアド・ステートメントを要求する度に、パラメータに転送されたSQL文をパースします。このパース作業が性能に影響を与えることがあるため、これを避けるために、JEUS内部でプリペアド・ステートメントをキャッシュする機能を提供します。この設定では、キャッシュするプリペアド・ステートメントの数を指定します。この機能を使用すると、コネクションを常に開いたまま使用するため、コネクションを閉じる際にドライバが行うクリア作業が行われません。</p> <p>たとえば、Oracle JDBCドライバの場合、<code>auto-commit</code>を<code>false</code>に設定して使用し、コミットやロールバックを行わずにコネクションを閉じると、必ずコミットを行うようになっているにもかかわらず処理が行われません。JDBCドライバでステートメント・キャッシュ機能を提供する場合は、それを使用してください</p>
<code>[-sfs,--stmtFetchSize <stmt-fetch-size>]</code>	<p>[動的変更]JDBCドライバのステートメントのフェッチ・サイズを設定します</p>
<code>[-ct,--connectionTrace <connection-trace>]</code>	<p>[動的変更]コネクションに関連する付加情報を提供するか否かを決定します。</p> <ul style="list-style-type: none"> – <code>false</code> : <code>--getConnectionTrace</code>と<code>--autoCommitTrace</code>はすべて無効になります
<code>[-gct,--getConnectionTrace <get-connection-trace>]</code>	<p>[動的変更]アプリケーションが<code>java.sql.DataSource#getConnection</code>を呼び出した際のスタック・トレースを確認できるようにします</p>
<code>[-act,--autoCommitTrace <auto-commit-trace>]</code>	<p>[動的変更]<code>java.sql.Connection#setAutoCommit</code>が呼び出された際の関連ログとスタック・トレースをサーバー・ログに記録します。ただし、<code>jeus.jdbc.connection-trace</code>ログのログ・レベルを<code>FINE</code>に設定します</p>

パラメータ	説明
[-ust,--useSqlTrace <use-sql-trace>]	[動的变化] コネクション別に使用しているSQLクエリーを表示する機能です。jeus.jdbc.sqlロガーのレベルをFINEに設定した場合、サーバー・ログを通じてSQLクエリーのヒストリーを確認できます。この機能を使用した場合、JDBCドライバーのステートメントの実装をJEUSのものとして囲むため、JDBCドライバーのstatementオブジェクトをキャストして使用するアプリケーションはこの機能を使用できません
[-kcho,--keepConnectionHandleOpen <keep-connection-handle-open>]	[動的变化] trueに設定した場合、XAコネクションを使用してプールに返す際に、ハンドル(または論理的コネクション)は開いておきます。この機能は、DB2ユニバーサル・ドライバーの場合に使用する必要があります。この機能を使用するとコネクション・ハンドルが閉じられないため、コネクションを閉じる際にドライバーが行うクリア作業が行われません。 たとえば、Oracle JDBCドライバーの場合、auto-commitをfalseに設定して使用し、コミットやロールバックを行わずにコネクションを閉じると、必ずコミットを行うようになっているにもかかわらず処理が行われません
[-en,--exportName <export-name>]	データソースのJNDIの名前を設定します。互いに異なる2つのデータソースが、互いに異なるサーバーにJNDIバインドされていることを保証できるのであれば、該当データソースは互いに同じJNDI名を持つことができます。これは、任意のサーバーで同じJNDI名を持つ、互いに異なるデータソースを許容しないことを意味します。 設定されていない場合、データソースIDをJNDI名に使用します。設定を解除するには、オプション値に「unset」を入力します
[-dscn,--dataSourceClassName <data-source-class-name>]	JDBCドライバーのデータソース・クラスの名前を設定します。パッケージ名を含む完全な形式で入力します
[-dst,--dataSourceType <data-source-type>]	データソースのタイプを設定します。 – DATA_SOURCE : コネクション・プーリング・サービスは提供されません – CONNECTION_POOL_DATA_SOURCE : コネクション・プーリング・サービスが提供されます。XA_DATA_SOURCEと設定した場合、コネクション・プーリング・サービスとXAの連動がサポートされます
[-vendor,--vendor <vendor>]	JDBCドライバーのベンダー名を設定します。 設定を解除するには、オプション値に「unset」を入力します

パラメータ	説明
[-desc,--description <description>]	<p>データソースについての説明です。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>
[-sn,--serverName <server-name>]	<p>データベースが実行されるホスト名またはIPを設定します。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>
[-pn,--portNumber <port-number>]	<p>データベースのリスナーのポート番号を設定します。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>
[-dn,--databaseName <database-name>]	<p>データベースの名前を設定します。Oracleの場合、データベースのSIDを設定します。設定を解除するには、オプション値に「unset」を入力します</p>
[-user,--user <user>]	<p>データベース・ユーザーのIDを設定します。トランザクション処理などのためには、十分な権限を持つ必要があります。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>
[-pw,--password <password>]	<p>データベース・ユーザーのパスワードを設定します。</p> <p>パスワードは「{algorithm}ciphertext」という形式で暗号化して保存します。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>
[-lt,--loginTimeout <login-timeout>]	<p>データベースとコネクションを接続する際に、ログイン段階で待機する最大時間を設定します(単位: 秒)</p>
[-il,--isolationLevel <isolation-level>]	<p>java.sql.Connectionで定義するトランザクション分離レベルを設定します。設定していない場合はJDBCドライバーの基本設定に従います。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>
[-pdt,--poolDestroyTimeout <pool-destroy-timeout>]	<p>コネクション・プールの破棄が完了するまで待機する時間です(単位: ms)。</p> <p>データソースを定義したアプリケーションをアンデプロイした際や、サーバーが停止した際にコネクション・プールは破棄されます。しかし、コネクションを削除しながらDBとネットワーク通信を行う場合はブロックされる可能性が存在し、コネクション・プールの破棄をひたすら待機するという問題が発生することがあります。こういった問題を避けるために、この設定を使用します。</p>

パラメータ	説明
	設定時間の分だけコネクション・プールの破棄が完了するまで待機し、アンデプロイまたはサーバーの停止を続けます
<code>[-prop,--property <property>]</code>	<p>JDBCドライバーごとのプロパティ設定について、統一された設定方法を提供します</p> <ul style="list-style-type: none"> – 1つのプロパティは「名前:タイプ=値」の形式で入力します – 複数のプロパティを設定する場合はコンマ(,)で区分します
<code>[-sxe,--supportXaEmulation <support-xa-emulation>]</code>	<p>ConnectionPoolDataSourceタイプのデータソースにのみ有効な設定です。この設定を適用した場合、ConnectionPoolDataSourceのコネクションがグローバル・トランザクション(XA)に参加するようにエミュレーションします。</p> <p>JEUS 6までのLocalXADataSourceの代替オプションとして、ConnectionPoolDataSourceタイプのコネクション・プールに使用します。1つのトランザクションには1つのConnectionPoolDataSourceのみ参加できる点に注意してください</p>
<code>[-ddba,--delegationDb <delegation-dba>]</code>	<p>データベースのセッションを強制的に切断できる権限 (DBA権限) を持つデータソース (以下、委任DBAデータソース) のJNDIの名前を設定します。この設定を行ったデータソースから取得されたコネクションを利用したクエリーの実行が一定時間以上遅延すると、JEUSは委任DBAデータソースを通じて、該当コネクションと関連するDBセッションを強制的に削除するクエリーをDBに送信します。その後、アプリケーションが使用不可能になったコネクションによって発生した例外を処理し、コネクションを閉じるとJEUSはそのコネクションを削除し、DBから新しいコネクションを取得してコネクション・プールに入れます。</p> <p>現在、Tibero、Oracle、Sybaseに対してこの機能をサポートします。この機能は、JDBC 2.0以下のJDBCドライバーでクエリーの実行に時間が掛かる場合に実行を中断させる方法として考案されました。しかし、JDBC 3.0またはそれ以上のバージョンを実装したJDBCドライバーはjava.sql.Statement#setQueryTimeoutを実装するため、この機能を通じてDBセッションを強制的に削除するよりは、ステートメントのクエリー・タイムアウトの設定を利用することを推奨します。特にXAデータソースの場合は、XAの実行中にDBセッションが削除されると、XA処理に問題が発生することがあります。そのため、ステートメントのクエリー・タイムアウトとトランザクション・タイムアウトを適切に設定して使用します。</p> <p>設定を解除するには、オプション値に「unset」を入力します</p>

パラメータ	説明
[-dbat,--dbaTimeout <dba-timeout>]	この設定で指定した時間の間のみ、委任DBAデータソースはコネクションのクエリー実行を待機します。設定時間が過ぎると、該当コネクションに関連するDBセッションを強制的に削除するクエリーをDBに送信します。 --delegationDbaが設定された場合にのみ有効です(単位: ms)
[-cqc,--checkQueryClass <check-query-class>]	ユーザーや開発者がコネクション・チェック機能をカスタマイズする場合、そのために実装したクラスのパッケージ名を含んだ名前を入力します。この場合、そのクラスは必ず jeus.jdbc.connectionpool.JEUSConnectionChecker インターフェースを実装します。 設定を解除するには、オプション値に「unset」を入力します
[-is,--initSql <init-sql>]	コネクションを作成後、最初に行うSQLクエリーを設定します。 設定を解除するには、オプション値に「unset」を入力します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	変更結果についての詳細情報が出力されます

- 例

```
[DAS]domain1.adminServer>modifyds -id ds1 -min 10 -max 50
Successfully performed the MODIFY operation for configuration of the data source
[ds1].
Check the results using "modify-data-source -id ds1"
```

4.2.11.4. list-data-sources

ドメインに存在するすべてのデータソースの一覧を出力します。データソースのIDを指定した場合、該当データソースの詳細設定を出力します。

- エイリアス

listds, lsds

- 使用方法

```
list-data-sources [-id,--dataSourceID <data-source-id>]
```

- パラメータ

パラメータ	説明
[-id,--dataSourceID <data-source-id>]	データソースのIDです

- 例

```
[DAS]domain1.adminServer>lsds
The list of data sources
=====
+-----+-----+-----+
| Data source ID | JNDI export name | Data source type |
+-----+-----+-----+
| ds1            | ds1              | ConnectionPoolDataSource |
| ds2            | ds2              | ConnectionPoolDataSource |
| ds3            | ds3              | ConnectionPoolDataSource |
+-----+-----+-----+
=====

[DAS]domain1.adminServer>lsds -id ds1
The configuration of the data source [ds1]
=====
+-----+-----+
| Configuration name | Configuration value |
+-----+-----+
| id                 | ds1                  |
| export-name        | ds1                  |
| data-source-class-name | oracle.jdbc.pool.OracleConnectionPoolDataSource |
| data-source-type    | ConnectionPoolDataSource |
| vendor              | oracle               |
| server-name         | 61.77.153.4          |
| port-number         |                       | 1521
| database-name       | orcl                 |
| user                | scott                |
+-----+-----+
```

password	tiger	
login-timeout		0
auto-commit	DRIVER	
stmt-query-timeout		0
pool-destroy-timeout		10000
property	[driverType; java.lang.String; thin]	
action-on-connection-leak	Warning	
support-xa-emulation	false	
min		10
max		50
step		1
period		3600000
enable-wait	false	
wait-time		10000
max-use-count		0
dbaTimeout		-1
stmt-caching-size		-1
stmt-fetch-size		-1
connection-trace	false	
get-connection-trace	true	
auto-commit-trace	false	
use-sql-trace	false	
keep-connection-handle-open	false	

```
+-----+-----+
=====
```

4.2.11.5. test-data-source-config

データソースの設定が正しいか否かを確認します。

- エイリアス

testdsconf, testdscfg

- 使用方法

```
test-data-source-config -id <data-source-id>
```

- パラメータ

パラメータ	説明
-id <data-source-id>	データソースのIDです

- 例

```
[DAS]domain1.adminServer>testdscfg -id ds1
The data source [ds1] is properly configured and can be used.
```

4.2.11.6. add-cluster-data-source

クラスターのデータソースをドメインに動的に追加します。

- エイリアス

addcds

- 使用方法

```
add-cluster-data-source [-id,--dataSourceID <data-source-id>]
                        [-en,--exportName <export-name>]
                        [-dss,--dataSourceSelector <data-source-selector>]
                        [-lb,--loadBalance <load-balance>]
                        [-ipc,--isPreConn <is-pre-conn>]
                        [-ufb,--useFailback <use-failback>]
                        [-cds,--componentDataSources <component-data-sources>]
                        [-f,--forceLock]
                        [-detail]
```

- パラメータ

パラメータ	説明
[-id,--dataSourceID <data-source-id>]	クラスターのデータソースのIDです。クラスターのデータソースIDは、1つのドメインでクラスターのデータソースの一意の識別子として動作するように設定します
[-en,--exportName <export-name>]	<p>クラスターのデータソースのJNDIの名前を設定します。互いに異なる2つのデータソースが、互いに異なるサーバーにJNDIバインドされていることを保証できるのであれば、該当データソースは互いに同じJNDI名を持つことができます。これは、任意のサーバーで同じJNDI名を持つ、互いに異なるデータソースを許容しないことを意味します。</p> <p>設定されていない場合、クラスターのデータソースIDをJNDI名に使用します</p>
[-dss,--dataSourceSelector <data-source-selector>]	<p>クラスターのデータソースからコネクションを取得する際、特定のコンポーネントのデータソースの選択に関するポリシーをユーザーや開発者が直接定義できます。jeus.jdbc.helper.DataSourceSelector抽象クラスを継承して実装し、その実装クラスのパッケージ名を含む名前を入力します。これを設定すると、--loadBalance設定は意味がなく、フェイルオーバーとフェイルバックを無条件で行います。</p> <ul style="list-style-type: none"> – フェイルオーバーは、この設定を通じて最初に選択されたコンポーネントのデータソースの次の索引のコンポーネントのデータソースを開始とし、一度のみ端数処理する方式で行われます。 – フェイルバックは、Use Failbackを設定した場合にフェイルバックが行われる方式と似ています。ポリシーを定義する際は同期化を考慮する必要があり、これは実装者の担当です
[-lb,--loadBalance <load-balance>]	<p>ロード・バランシングの可否を設定します。</p> <ul style="list-style-type: none"> – true : --isPreConn設定と--useFailback設定は無意味となります
[-ipc,--isPreConn <is-pre-conn>]	クラスターのデータソースに属するコンポーネントのデータソースのコネクション・プールを事前に作成するか否かを決定します。コンポーネントのデータソースのコネクション・プールを事前に作成しておく、性能上のメリットはありますが、リソースが浪費し、効率が落ちます
[-ufb,--useFailback <use-failback>]	<p>JEUSの以前のバージョンではフェイルオーバーのみを行っていたため、互換性のために提供するオプションです。</p> <p>補助データソースにフェイルオーバーした後、メインのデータソースにフェイルバックするか否かを設定します。デフォルトではフェイルバックを行います。フェイルバックを行うためには、必ずメインのデータソースに対して--checkQueryおよび--checkQueryPeriodを設定します</p>

パラメータ	説明
[-cds,--componentDataSources <component-data-sources>]	クラスターのデータソースに属するコンポーネントのデータソースのデータソースIDを指定します
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	変更結果についての詳細情報が出力されます

- 例

```
[DAS]domain1.adminServer>addcds -id cds1 -cds ds1,ds2
Successfully performed the ADD operation for cluster data source [cds1] to domain.
Check the results using "add-cluster-data-source"
```

4.2.11.7. remove-cluster-data-source

クラスターのデータソースをドメインから動的に削除します。

- エイリアス

rmcds

- 使用方法

```
remove-cluster-data-source [-id,--dataSourceID <data-source-id>]
                           [-f,--forceLock]
                           [-detail]
```

- パラメータ

パラメータ	説明
[-id,--dataSourceID <data-source-id>]	クラスターのデータソースIDです
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	変更結果に関する詳細情報が出力されます

- 例

```
[DAS]domain1.adminServer>rmcds -id cds1
Successfully performed the REMOVE operation for cluster data source [cds1] from
domain.
Check the results using "remove-cluster-data-source"
```

4.2.11.8. modify-cluster-data-source

クラスターのデータソースの設定を変更します。大部分の設定が動的に変更可能です。

- エイリアス

modifycds

- 使用方法

```
modify-cluster-data-source -id,--dataSourceID <data-source-id>
                             [-en,--exportName <export-name>]
                             [-dss,--dataSourceSelector <data-source-selector>]
                             [-lb,--loadBalance <load-balance>]
                             [-ipc,--isPreConn <is-pre-conn>]
                             [-ufb,--useFailback <use-failback>]
                             [-cds,--componentDataSources <component-data-sources>]

                             [-oc,--onsConfig <ons-config>]
                             [-xa,--xaAffinity <xa-affinity>]
                             [-f,--forceLock]
                             [-detail]
```

- パラメータ

パラメータ	説明
-id,--dataSourceID <data-source-id>]	クラスターのデータソースのIDを設定します。 クラスターのデータソースIDは、1つのドメインでクラスターのデータソースの一意の識別子として動作するように設定します
[-en,--exportName <export-name>]	クラスターのデータソースのJNDIの名前を設定します。互いに異なる2つのデータソースが、互いに異なるサーバーにJNDIバインドされていることを保証できるのであれば、該当データソースは互いに同じJNDI名を持つことができます。これは、任意のサーバーで同じJNDI名を持つ、互いに異なるデータソースを許容しないことを意味します。 設定されていない場合、クラスターのデータソースIDをJNDI名に使用します。設定を解除するには、オプション値に「unset」を入力します
[-dss,--dataSourceSelector <data-source-selector>]	[動的変更] クラスターのデータソースからコネクションを取得する際、特定のコンポーネントのデータソースの選択に関するポリシーをユーザーや開発者が直接定義できます。 jeus.jdbc.helper.DataSourceSelector抽象クラスを継承して実装し、その実装クラスのパッケージ名を含む名前を入力します。これを設定すると、ロード・バランス設定は意味がなく、フェイルオーバーとフェイルバックを無条件で行います。

パラメータ	説明
	<ul style="list-style-type: none"> – フェイルオーバーは、この設定を通じて最初に選択されたコンポーネントのデータソースの次の索引のコンポーネントのデータソースを開始とし、一度のみ端数処理する方式で行われます – フェイルバックは、Use Failbackを設定した場合にフェイルバックが行われる方式と似ています。ポリシーを定義する際は同期化を考慮する必要があり、これは実装者の担当です <p>設定を解除するには、オプション値に「unset」を入力します</p>
<code>[-lb,--loadBalance <load-balance>]</code>	<p>[動的変更]ロード・バランシングの可否を設定します。</p> <ul style="list-style-type: none"> – true : --isPreConn設定と--useFailback設定は無意味となります
<code>[-ipc,--isPreConn <is-pre-conn>]</code>	<p>[動的変更]クラスターのデータソースに属するコンポーネントのデータソースのコネクション・プールを事前に作成するか否かを決定します。コンポーネントのデータソースのコネクション・プールを事前に作成しておく、性能上のメリットはありますが、リソースの節約面では良くありません</p>
<code>[-ufb,--useFailback <use-failback>]</code>	<p>[動的変更]JEUSの以前のバージョンではフェイルオーバーのみを行っていたため、互換性のために提供するオプションです。</p> <p>補助データソースにフェイルオーバーした後、メインのデータソースにフェイルバックするか否かを設定します。デフォルトではフェイルバックを行います。フェイルバックを行うためには、必ずメインのデータソースに対して--checkQueryおよび--checkQueryPeriodを設定します</p>
<code>[-cds,--componentDataSources <component-data-sources>]</code>	<p>[動的変更]クラスターのデータソースに属するコンポーネントのデータソースのデータソースIDを指定します</p>
<code>[-oc,--onsConfig <ons-config>]</code>	<p>ONSが含まれたクラスター・データソースを使用する場合に設定します。ONS上の各RACノードがONS通信に使用するIPアドレス、ポート番号を記述します。クラスター・データソースは設定されたIP、ポートにソケット接続を確立し、ONSクライアントとして動作します。以下のような形式で記述します</p> <p>例) nodes=host1:6200,host2:6200</p>
<code>[-xa,--xaAffinity <xa-affinity>]</code>	<p>グローバル・トランザクションのアフィニティ設定の可否を決定します。この設定を有効にすると、グローバル・トランザクションの処理が1つのRACインスタンスに限定して行われるため、グローバル・トランザクションの処理性能の向上を図れます</p>
<code>[-f,--forceLock]</code>	<p>設定を強制的に変更して反映します</p>
<code>[-detail]</code>	<p>変更結果についての詳細情報が出力されます</p>

- 例

```
[DAS]domain1.adminServer>modifycds -id cds1 -cds ds1
Successfully performed the MODIFY operation for configuration of the cluster
data source [cds1].
Check the results using "modify-cluster-data-source -id cds1"
```

4.2.11.9. list-cluster-data-sources

ドメインに存在するすべてのクラスターのデータソースの一覧を出力します。データソースのIDを指定した場合、該当クラスターのデータソースの詳細設定を出力します。

- エイリアス

listcds, lscds

- 使用方法

```
list-cluster-data-sources [-id,--dataSourceID <data-source-id>]
```

- パラメータ

パラメータ	説明
<code>[-id,--dataSourceID <data-source-id>]</code>	クラスターのデータソースIDです

- 例

```
[DAS]domain1.adminServer>lscds
The list of cluster data sources
=====
+-----+-----+-----+
| Data source ID | JNDI export name | Component data sources |
+-----+-----+-----+
| cds1           | cds1             | [ds1, ds2]            |
| cds2           | cds2             | [ds3, ds4]            |
+-----+-----+-----+
=====

[DAS]domain1.adminServer>lscds -id cds1
The configuration of cluster data source [cds1]
=====
+-----+-----+
| Configuration name | Configuration value |
+-----+-----+
```

+-----+-----+	
id	cds1
export-name	cds1
load-balance	false
is-pre-conn	false
use-failback	true
component-data-sources	[ds1, ds2]
+-----+-----+	
=====	

4.2.12. コネクション・プールのモニタリングおよび制御コマンド

JDBC/JCAコネクション・プールのモニタリングおよび制御機能を提供します。コマンドの一覧は以下のとおりです。

コマンド	説明
create-connection-pool	JDBCコネクション・プールを作成します。JCAコネクション・プールに対しては動作しません
enable-connection-pool	JDBCコネクション・プールを活性化します。JCAコネクション・プールに対しては動作しません
disable-connection-pool	JDBCコネクション・プールを非活性化します。JCAコネクション・プールに対しては動作しません
refresh-connection-pool	JDBCコネクション・プールのコネクションを新規コネクションに切り替えます。JCAコネクション・プールに対しては動作しません
shrink-connection-pool	JDBCコネクション・プールのコネクション数を、設定されているコネクションの最小値に調整します。JCAコネクション・プールに対しては動作しません
control-connection-pool	JDBCやJCAコネクション・プールを制御します。コネクション・プールの作成、活性化、非活性化、リフレッシュ、シュリンクが可能です
connection-pool-info	JDBCやJCAコネクション・プールをモニタリングします。コネクション・プールのIDを指定した場合、該当コネクション・プールの詳細情報を出力します

コマンド	説明
<code>control-cluster-data-source</code>	クラスターのデータソースを制御します。クラスターのデータソースに属するコンポーネントのデータソース一覧を表示するか、フェイルバックを実行します

4.2.12.1. create-connection-pool

JDBCコネクション・プールを作成します。JCAコネクション・プールに対しては動作しません。

- エイリアス

`createcp`

- 使用方法

```
create-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- パラメータ

パラメータ	説明
<code>-id <connection-pool-id></code>	JDBCコネクション・プールのIDです
<code>[-server <server-name>]</code>	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>createcp -id ds1 -server server1
Servers that successfully created a connection pool : server1
Servers that failed to create a connection pool : none.
```

4.2.12.2. enable-connection-pool

JDBCコネクション・プールを活性化します。JCAコネクション・プールに対しては動作しません。

- エイリアス

`enablecp`

- 使用方法

```
enable-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- パラメータ

パラメータ	説明
-id <connection-pool-id>	JDBCコネクション・プールのIDです
[-server <server-name>]	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>enablecp -id ds1 -server server1
Servers that successfully enabled a connection pool : server1
Servers that failed to enable a connection pool : none.
```

4.2.12.3. disable-connection-pool

JDBCコネクション・プールを非活性化します。JCAコネクション・プールに対しては動作しません。

- エイリアス

disablecp

- 使用方法

```
disable-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- パラメータ

パラメータ	説明
-id <connection-pool-id>	JDBCコネクション・プールのIDです
[-server <server-name>]	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>disablecp -id ds1 -server server1
Servers that successfully disabled a connection pool : server1
Servers that failed to disable a connection pool : none.
```

4.2.12.4. refresh-connection-pool

JDBCコネクション・プールのコネクションを新規コネクションに切り替えます。JCAコネクション・プールに対しては動作しません。

- エイリアス

refreshcp

- 使用方法

```
refresh-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- パラメータ

パラメータ	説明
-id <connection-pool-id>	JDBCコネクション・プールのIDです
[-server <server-name>]	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>refreshcp -id ds1 -server server1
Servers that successfully refreshed a connection pool : server1
Servers that failed to refresh a connection pool : none.
```

4.2.12.5. shrink-connection-pool

JDBCコネクション・プールのコネクション数を、設定されているコネクションの最小値に調整します。JCAコネクション・プールに対しては動作しません。

- エイリアス

shrinkcp

- 使用方法

```
shrink-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
```

- パラメータ

パラメータ	説明
-id <connection-pool-id>	JDBCコネクション・プールのIDです
[-server <server-name>]	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>shrinkcp -id ds1 -server server1
Servers that successfully shrank a connection pool : server1
Servers that failed to shrink a connection pool : none.
```

4.2.12.6. control-connection-pool

JDBC/JCAコネクション・プールを制御します。コネクション・プールの作成、活性化、非活性化、リフレッシュ、シュリンクが可能です。

- エイリアス

controlcp, ctrlcp

- 使用方法

```
control-connection-pool -id <connection-pool-id>
                        [-server <server-name>]
                        -enable | -shrink | -disable | -refresh | -create
```

- パラメータ

パラメータ	説明
-id <connection-pool-id>	コネクション・プールのIDです
[-server <server-name>]	サーバーの名前です
-enable	コネクション・プールを活性化します
-shrink	コネクション・プールのコネクション数を、設定されているコネクションの最小値に調整します
-disable	コネクション・プールを非活性化します
-refresh	コネクション・プールのコネクションを新規コネクションに切り替えます
-create	コネクション・プールを作成します

- 例

```
[DAS]domain1.adminServer>ctrlcp -id ds1 -server server1 -enable
Servers that successfully enabled a connection pool : server1
Servers that failed to enable a connection pool : none.

[DAS]domain1.adminServer>ctrlcp -id ds1 -server server1 -shrink
Servers that successfully shrank a connection pool : server1
Servers that failed to shrink a connection pool : none.

[DAS]domain1.adminServer>ctrlcp -id ds1 -server server1 -disable
Servers that successfully disabled a connection pool : server1
Servers that failed to disable a connection pool : none.

[DAS]domain1.adminServer>ctrlcp -id ds1 -server server1 -refresh
Servers that successfully refreshed a connection pool : server1
Servers that failed to refresh a connection pool : none.
```



```
[DAS]domain1.adminServer>ctrlcp -id ds1 -server server1 -create
Servers that successfully created a connection pool : server1
Servers that failed to create a connection pool : none.
```

4.2.12.7. connection-pool-info

JDBC/JCAコネクション・プールをモニタリングします。コネクション・プールのIDを指定した場合、該当コネクション・プールの詳細情報を出力します。

- エイリアス

cpinfo

- 使用方法

```
connection-pool-info -server <server-name>
                    [-id <connection-pool-id>]
                    [-jca | -jdbc]
                    [-jndi]
                    [-active]
                    [-t]
                    [-stmt]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-id <connection-pool-id>]	コネクション・プールのIDです
[-jca]	JCAコネクション・プールの情報のみ表示します
[-jdbc]	JDBCコネクション・プールの情報のみ表示します
[-jndi]	コネクション・プールのJNDI名を表示します
[-active]	現在サーバーに作成されているコネクション・プールのみ表示します
[-t]	コネクションを最後に使用したスレッドの名前を表示します。 -idオプションが設定されている場合にのみ有効です
[-stmt]	各コネクションのステートメント・キャッシュ情報を表示します

- 例

```
[DAS]domain1.adminServer>cpinfo -server server1 -jndi -jdbc
The connection pool information on the server [server1].
=====
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Connection | JNDI export| Min | Max | Acti| Idle | Disposa|Total| Wait| Enabl|
| Pool ID   | name      |     |     | ve  |      | ble    |     |     | ed   |
|           |           |     |     |     |     |        |     |     |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ds1       | ds1       | 2   | 30  | 0   | 2   | 0      | 2   | false| true |
|           |           |     |     |     |     |        |     |     |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ds2 *     | ds2       | 2   | 30  | 0   | 0   | 0      | 0   | false| false|
+-----+-----+-----+-----+-----+-----+-----+-----+

* : has not been created, total = active + idle + disposable
=====

[DAS]domain1.adminServer>cpinfo -server server1 -id ds1 -t -jdbc
Information about connections in the server [server1]'s connection pool [ds1].
=====
+-----+-----+-----+-----+-----+-----+-----+-----+
| Connection ID | State | State time(sec.) | Use count | Type | Thread name |
|               |       |                   |           |      |              |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ds1-1         | active| 1105.954         | 1         | pooled | http-wl    |
|               |       |                   |           |      |              |
| ds1-2         | idle  | 1105.923         | 0         | pooled |              |
|               |       |                   |           |      |              |
+-----+-----+-----+-----+-----+-----+-----+-----+
=====

[DAS]domain1.adminServer>cpinfo -server server1 -id ds1 -stmt -jdbc
Information about statement caches in the server [server1]'s connection pool [ds1].
=====
+-----+-----+-----+-----+-----+-----+-----+-----+
| Connection ID| Access | Hit | Miss | Add | Remove | Current |
|               | Count | Count | Count | Count | Count | Size    |
|               |       |       |       |       |       |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ds1-1        | 5      | 0    | 5     | 5     | 0     | 5       |
|               |       |       |       |       |       |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ds1-2        | 5      | 2    | 3     | 3     | 0     | 3       |
|               |       |       |       |       |       |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
=====

```

4.2.12.8. control-cluster-data-source

クラスターのデータソースを制御します。クラスターのデータソースに属するコンポーネントのデータソース一覧を表示するか、フェイルバックを実行します。

- エイリアス

controlcds, ctrlcds

- 使用方法

```
control-cluster-data-source -server <server-name>
                             -id <data-source-id>
                             [-failback]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
-id <data-source-id>	クラスターのデータソースのIDです
[-failback]	クラスターのデータソースに対してフェイルバックを実行します

- 例

```
[DAS]domain1.adminServer>ctrlcds -server server1 -id cds1
Component data sources of the cluster data source [cds1] : ds1, ds2

[DAS]domain1.adminServer>ctrlcds -server server1 -id cds1 -failback
The failback for the cluster data source [cds1] succeeded.
```

4.2.13. トランザクション関連コマンド

トランザクションに関連するコマンドの一覧は以下のとおりです。

コマンド	説明
modify-transaction-manager	指定サーバーのトランザクション・マネージャーの設定を変更します
recover-transactions	指定サーバーのリソース・マネージャーを復旧します。復旧するリソースを選択できます
show-transaction-manager	指定サーバーのトランザクション・マネージャーの設定を確認します
transaction-info	指定サーバーのトランザクションを状態別に確認します

4.2.13.1. modify-transaction-manager

指定サーバーのトランザクション・マネージャーの設定を変更します。

- エイリアス

modify-tm, modifytm, tmconfig, tmconf

- 使用方法

```
modify-transaction-manager <server-name>

    [-at,--activeTimeout <active timeout>]
    [-pt,--prepareTimeout <prepare timeout>]
    [-pdt,--preparedTimeout <prepared timeout>]
    [-ct,--commitTimeout <commit timeout>]
    [-rt,--recoveryTimeout <recovery timeout>]
    [-it,--incompleteTimeout <incomplete timeout>]
    [-d,--txLogDir <transaction log directory>]
    [-a,--automaticRecovery <automatic recovery enable/disable>]

    [-f,--forceLock]
    [-detail]
```

- パラメータ

パラメータ	説明
<server-name>	トランザクションの設定を変更するサーバーの名前です
[-at,--activeTimeout <active timeout>]	トランザクションを開始し、コミットするまでの最大時間です。時間を超過したトランザクションは強制的にロールバックされます
[-pt,--prepareTimeout <prepare timeout>]	ルート・ディレクトリーがprepareを送信し、下位コーディネーターからの応答を待機する最大時間です
[-pdt,--preparedTimeout <prepared timeout>]	下位コーディネーターがprepareの応答を送信し、ルート・コーディネーターのGlobal decisionを待機する最大時間です。時間を超過した場合、ルートにprepareの応答を再度送信します
[-ct,--commitTimeout <commit timeout>]	ルート・コーディネーターが下位コーディネーターにコミットを送信し、その応答を待機する最大時間です。時間を超過した場合、Incompleteトランザクションとして扱います
[-rt,--recoveryTimeout <recovery timeout>]	リカバリー情報を取得するまでに待機する最大時間です。時間を超過した場合、ユーザーにリカバリーを任せます
[-it,--incompleteTimeout <incomplete timeout>]	Incompleteトランザクションがリカバリーされるまでに待機する最大時間です。時間を超過した場合、Incompleteトランザクションという情報を削除します

パラメータ	説明
[-d,--txLogDir <transaction log directory>]	リカバリーのためのトランザクション・ログを残すディレクトリーです。詳細については『JEUS サーバガイド』の「7.5.2. リカバリー関連のログ・ファイル」を参照してください
[-a,--automaticRecovery <automatic recovery enable/disable>]	[動的変更] サーバーに異常がある場合、他のサーバーが該当サーバーのIncompleteトランザクションをリカバリーするか否かを設定します。詳細については『JEUS サーバガイド』の「7.5. トランザクションのリカバリー」を参照してください
[-f,--forceLock]	設定を強制的に変更して反映します
[-detail]	結果の詳細を表示します

- 例

- 現在の設定確認

```
[DAS]domain1.adminServer>modify-transaction-manager server1
Shows the current configuration.
transaction of server (server1)
=====
+-----+-----+
| Active Timeout      | 600000 | |
| |                  |        |
| Prepare Timeout     | 120000 |
| |                  |        |
| Prepared Timeout    | 60000  |
| |                  |        |
| Commit Timeout      | 240000 |
| |                  |        |
| Recovery Timeout    | 120000 |
| |                  |        |
| Incomplete Timeout  | 8640000|
| |                  |        |
| Tx Log Dir          | ${SERVER_HOME}/.workspace/tmlog |
| |                  |        |
| Automatic Recovery  | false  |
| |                  |        |
+-----+-----+
=====
```

- 動的設定のトランザクション自動復旧機能のオン

```
[DAS]domain1.adminServer>modify-transaction-manager server1 -automaticRecovery true
Successfully performed the MODIFY operation for transaction of server (server1).
```

```
Check the results using "show-transaction-manager server1 or  
modify-transaction-manager server1"
```

- サーバーの再起動によって反映されるタイムアウト設定の変更

```
[DAS]domain1.adminServer>modify-transaction-manager server1 -activeTimeout  
20000  
Successfully performed the MODIFY operation for transaction of server (server1),  
but all changes were non-dynamic. They will be applied after restarting.  
Check the results using "show-transaction-manager server1 or  
modify-transaction-manager server1"
```

- 動的に反映される設定とサーバーの再起動によって反映される設定を一緒に変更

```
[DAS]domain1.adminServer>modify-transaction-manager server1 -automaticRecovery  
true -activeTimeout 20000  
Successfully performed the MODIFY operation for transaction of server (server1),  
but some changes were non-dynamic. They will be applied after restarting.  
Check the results using "show-transaction-manager server1 or  
modify-transaction-manager server1"
```

4.2.13.2. recover-transactions

指定サーバーのリソース・マネージャーを復旧します。復旧するリソースを選択できます。

- エイリアス

recovertx, tmresync

- 使用方法

```
recover-transactions -server <server-name>  
                     <resource-type>  
                     <export-name>  
                     [<connector-module-name>]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
<resource-type>	復旧するリソース・タイプを指定します - sql : JDBC XAデータソース - jms : JMSコネクション・ファクトリー - connector : コネクタのリソース・アダプター

パラメータ	説明
<export-name>	リソースの<export-name>を指定します
[<connector-module-name>]	リソースがコネクタの場合にのみ、該当コネクタ・モジュールの名前を指定します

- 例

```
[DAS]domain1.adminServer>recover-transactions -server server1 sql
jdbc/derby/XADataSource
The recover-transactions command was executed on the server1 server.
```

4.2.13.3. show-transaction-manager

指定サーバーのトランザクション・マネージャーの設定を確認します。

- エイリアス

show-tm, showtm

- 使用方法

```
show-transaction-manager <server-name>
```

- パラメータ

パラメータ	説明
<server-name>	サーバーの名前です

- 例

```
[DAS]domain1.adminServer>show-transaction-manager server1
Shows the current configuration.
transaction of server (server1)
=====
+-----+-----+
| Active Timeout      | 600000 |
|-----|-----|
| Prepare Timeout     | 120000 |
|-----|-----|
| Prepared Timeout    | 60000  |
|-----|-----|
| Commit Timeout      | 240000 |
|-----|-----|
| Recovery Timeout    | 120000 |
|-----|-----|
|
```

```
| Incomplete Timeout | 86400000 |
| Tx Log Dir         | ${SERVER_HOME}/.workspace/tmlog
|
| Automatic Recovery | false
|
+-----+-----+
=====
```

4.2.13.4. transaction-info

指定サーバーのトランザクションを状態別に確認します。

- エイリアス

transactioninfo, txinfo

- 使用方法

```
transaction-info -server <server-name>
                    [-i,--inflight]
                    [-d,--indoubt]
                    [-r,--retrying]
```

- パラメータ

パラメータ	説明
-server <server-name>	サーバーの名前です
[-i,--inflight]	実行中のトランザクションを確認します。コミットまたはロールバック段階まで実行されていない、メモリー上のトランザクションです
[-d,--indoubt]	未完了のトランザクションを確認します。まだ終了しておらず、復旧が必要であるか、どう扱うかがまだ決まっていないトランザクションです
[-r,--retrying]	再試行するトランザクションを確認します。リソース・マネージャーの問題により、1フェーズ・コミットまたは2フェーズ・コミットを正常に処理できず、JEUS内部で再試行中のトランザクションです

- 例

```
[DAS]domain1.adminServer>transaction-info -server server1
<transactions of server1>

In-Flight Transaction Information
=====
+-----+-----+-----+-----+-----+-----+-----+
|      XID      | Coordina| Extern| Status | Timeout | Elapse | XAResources|
|                | tor    | al XID |        |         |        |            |
+-----+-----+-----+-----+-----+-----+-----+
```



```

|
+-----+-----+-----+-----+-----+-----+
| 049FE5.00C0A80011| 192.168.|      | STATUS | 5000(ms)| 7418(ms)| jdbc/derby/|
| 2608221CC3D6B40000| 0.17:9736|      | _MARKED_|      | XADataSource|
| 000000000011.00   |      |      | ROLLBACK|      |      |
|
+-----+-----+-----+-----+-----+-----+
=====

In-doubt Transaction Information
=====
+---+-----+-----+-----+-----+-----+
| XID| Coordinator | External XID | Status| Decision | Sub-XID|
+---+-----+-----+-----+-----+-----+
(No data available)
=====

Retrying Transaction Information
=====
+---+-----+-----+-----+-----+-----+
| XID| Name | Failuere Count | Status| Decision|
+---+-----+-----+-----+-----+-----+
(No data available)
=====

```

4.2.14. セキュリティー関連コマンド

本節では、セキュリティーに関連するコマンドについて説明します。セキュリティー関連コマンドは、ユーザー、グループ、およびポリシー管理コマンドに分類されます。コマンドの一覧は以下のとおりです。

- ユーザー管理コマンド

ユーザー管理コマンドは、ユーザーを追加、修正、削除する際に使用します。

コマンド	説明
add-user	新規ユーザーを追加します
remove-user	ユーザーを削除します
list-user-names	ログインしたユーザー名の一覧と各ユーザーのロック状態を出力します
show-user	ユーザーに関する情報を出力します
set-password	パスワードを設定します
lock-user	ユーザーにロックを設定します
unlock-user	ユーザーに設定されているロックを解除します

- グループ管理コマンド

グループ管理コマンドは、グループを追加、修正、削除する際に使用します。

コマンド	説明
add-goup	新規グループを追加します
add-user-to-group	グループにユーザーを追加します
remove-user-from-group	グループのユーザーを削除します
remove-group	グループを削除します
list-group-names	グループの一覧を出力します
show-group	グループに関する情報を出力します
check-group-member	グループにユーザーが登録されているか否かを出力します

- ポリシー管理コマンド

ポリシー管理コマンドは、セキュリティ・システムでポリシーに関連する事項を管理するために使用します。このコマンドを使用することで、Principal-RoleマッピングやRole-Resourceマッピングを追加および削除することが可能です。

ポリシー管理コマンドは、セキュリティ・システムで権限チェックに関わる全般的な事項を管理することができます。

コマンド	説明
show-policy	コンテキストIDを含むポリシーを出力します
assign-role	ロールにプリンシパルを割り当てます
unassign-role	ロールからプリンシパルを削除します
assign-resource	リソースにロールを割り当てます
unassign-resource	リソースを削除します

注

本節で紹介するコマンドはセキュリティに関わるデリケートなものなので、このコマンドを実行するには適切な権限を持つユーザーでログインする必要があります。

4.2.14.1. 共通適用事項

以下は、セキュリティ関連コマンドにおいて共通で適用されるオプションについての説明です。

- [-domain <domain-name>]オプション

- JEUSでセキュリティが適用される範囲のセキュリティ・ドメインを指定します(デフォルト値: SYSTEM_DOMAIN)。セキュリティ・ドメインについての詳細は『JEUS セキュリティガイド』を参照してください。
- 各コマンドのオプションの説明では、このオプションについての説明は省略します。

4.2.14.2. add-user

セキュリティ・システムに新規ユーザーを動的に追加します。指定されたユーザー名は、現在のドメイン内で一意である必要があります(「list-user-names」コマンドで確認可能)。ユーザー名は一連の呼び出し過程でユーザーを参照するために使用します。つまり、ユーザーのIDに該当します。

- エイリアス

adduser

- 使用方法

```
add-user <user-name>
        [-domain <domain-name>]
        [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<user-name>	ユーザー名です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> add-user user1
The user [user1] has been successfully added.
```

4.2.14.3. remove-user

現在のドメインに存在しているユーザーを動的に削除します。

- エイリアス

removeuser

- 使用方法

```
remove-user <user-name>
          [-domain <domain-name>]
          [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<user-name>	ユーザー名です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> remove-user user1
The user [user1] has been successfully removed.
```

4.2.14.4. list-user-names

現在ドメインにログインしているすべてのユーザー名の一覧と各ユーザーのロック状態を出力します。

- エイリアス

listusernames, getusernames

- 使用方法

```
list-user-names [-domain <domain-name> ]
```

- 例

```
[DAS]domain1.adminServer> list-user-names
=====
+-----+-----+
| User Name | Lock State |
+-----+-----+
| jeus      | unlocked  |
| admin     | locked    |
+-----+-----+
=====
```

4.2.14.5. show-user

指定されたユーザーについての詳細情報を出力します。

- エイリアス

showuser, getuser

- 使用方法

```
show-user <user-name>
        [-domain <domain-name>]
```

- パラメータ

パラメータ	説明
<user-name>	ユーザー名です

- 例

```
[DAS]domain1.adminServer> show-user user1
[SUBJECT]
Description: No description
Domain: SYSTEM_DOMAIN
Main principal: Principal user1
Principals: [Principal user1,Administrators(members:Principal admin,Principal
user1)]
Public credentials: []
```

4.2.14.6. set-password

指定したユーザーのパスワードを動的に設定します。パスワードは、暗号化やエンコードのない一般文字で設定する必要があります。-algorithmオプションを使用してパスワードを暗号化できます。設定した後は、新しいパスワードで再度ログインする必要があります。

注

このコマンドは、該当ユーザーに以前設定されていたパスワードを削除します。

- エイリアス

setpassword

- 使用方法

```
set-password <user-name>
            <password>
            [-algorithm <algorithm>]
            [-domain <domain-name>]
            [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<user-name>	ユーザー名です
<password>	[動的変更] ユーザーのパスワードです
[-algorithm <algorithm>]	[動的変更] パスワードの暗号化に使用するアルゴリズムの名前です。 以下のアルゴリズムを選択できます – AES – DES – DESede – Blowfish – SEED
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> set-password user1 pass1 -algorithm AES
The password is set for [user1].
```

4.2.14.7. lock-user

ユーザーにロックを動的に設定します。「SubjectLockoutValidationService」がサーバーに正しく構成されていれば、「unlock-user」が呼び出される前に該当ユーザーのすべてのログイン試行が失敗します。

Subject validationについては『*JEUS セキュリティガイド*』の「5.4.1. SubjectValidationService SPI」を参照してください。

- エイリアス

lockuser

- 使用方法

```
lock-user <user-name>
        [-domain <domain-name>]
```

- パラメータ

パラメータ	説明
<user-name>	ユーザー名です

- 例

```
[DAS]domain1.adminServer> lock-user user2
The user [user2] has been locked out.
```

4.2.14.8. unlock-user

`lock-user`で設定したロックを動的に解除します。

- エイリアス

unlockuser

- 使用方法

```
unlock-user <user-name>
           [-domain <domain-name>]
```

- パラメータ

パラメータ	説明
<user-name>	ユーザー名です

- 例

```
[DAS]domain1.adminServer> unlock-user user2
The user [user2] has been successfully unlocked.
```

4.2.14.9. add-group

セキュリティー・システムに新規グループを動的に追加します。設定されたグループ名は、現在のドメイン内で一意である必要があります(`list-group-names`コマンドで確認可能)。グループ名は一連の呼び出し過程でグループを参照するために使用します。つまり、グループのIDに該当します。

- エイリアス

addgroup

- 使用方法

```
add-group <group-name>
          [-domain <domain-name>]
          [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<group-name>	グループ名です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> add-group group1
The group [group1] has been successfully added.
```

4.2.14.10. add-user-to-group

セキュリティー・システムに登録されている特定グループに新規ユーザーを動的に追加します。

- エイリアス

addusertogroup

- 使用方法

```
add-user-to-group <group-name>
                  <user-name>
                  [-domain <domain-name>]
                  [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<group-name>	グループ名です
<user-name>	ユーザー名です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> add-user-to-group group1 user1
The user [user1] has been successfully added to the group [group1].
```

4.2.14.11. remove-user-from-group

現在ドメインに存在している特定グループの特定ユーザーを動的に削除します。

- エイリアス

removeuserfromgroup

- 使用方法

```
remove-user-from-group <group-name>
                        <user-name>
                        [-domain <domain-name>]
                        [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<group-name>	グループ名です
<user-name>	ユーザー名です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> remove-user-from-group group1 user1
The user [user1] has been successfully removed from the group [group1].
```

4.2.14.12. remove-group

現在ドメインに存在しているグループを動的に削除します。グループに属するユーザーが存在しない場合にのみ削除可能です。

- エイリアス

removegroup

- 使用方法

```
remove-group <group-name>
             [-domain <domain-name>]
             [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<group-name>	グループ名です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> remove-group group1
The group [group1] has been successfully removed.
```

4.2.14.13. list-group-names

現在ドメインに存在している全グループの一覧を出力します。

- エイリアス

listgroupnames, getgroupnames

- 使用方法

```
list-group-names [-domain <domain-name> ]
```

- 例

```
[DAS]domain1.adminServer> list-group-names
=====
+-----+
|                                     Group Name                                     |
|                                     +-----+                                     |
+-----+-----+
| Administrators                                     |
|                                     +-----+                                     |
+-----+-----+
| group1                                     |
|                                     +-----+                                     |
+-----+-----+
=====
```

4.2.14.14. show-group

特定グループの詳細情報を出力します。

- エイリアス

showgroup, getgroup

- 使用方法

```
show-group <group-name>
          [-domain <domain-name>]
```

- パラメータ

パラメータ	説明
<group-name>	グループ名です

- 例

```
[DAS]domain1.adminServer> show-group Administrators
Administrators(members:Principal admin,Principal user1)
```

4.2.14.15. check-group-member

特定グループに該当ユーザーが登録されているか否かを出力します。

- エイリアス

checkgroupmember, ismember

- 使用方法

```
check-group-member <group-name>
                        <user-name>
                        [-domain <domain-name>]
```

- パラメータ

パラメータ	説明
<group-name>	グループ名です
<user-name>	ユーザー名です

- 例

```
[DAS]domain1.adminServer> check-group-member group1 user1
The user [user1] is a member of the group [group1].
```

4.2.14.16. show-policy

特定コンテキストIDを含むポリシーを出力します。context-idが設定されていない場合は「default」が設定されます。

- エイリアス

showpolicy, getpolicy

- 使用方法

```
show-policy [-contextid <context-id>]
              [-domain <domain-name>]
```

- パラメータ

パラメータ	説明
[-contextid <context-id>]	ポリシーを区分するコンテキスト名です

- 例

```
[DAS]domain1.adminServer> show-policy
_____POLICY_____
ROLE PERMISSIONS:
  PERMISSION MAP
  -----
  Excluded permissions:
    (none)
  Unchecked permissions:
    (jeus.security.resource.RolePermission jndiUser)

  Checked permissions:
    Permission owner:
    Principal user1
    Permissions of owner:
    (jeus.security.resource.RolePermission AdministratorsRole)

    Permission owner:
    Principal anonymous
    Permissions of owner:
    (jeus.security.resource.RolePermission jndiUser)

RESOURCE PERMISSIONS:
Context id: default
  PERMISSION MAP
  -----
  Excluded permissions:
    (none)
  Unchecked permissions:
    (none)

  Checked permissions:
    Permission owner:
    Role jndiUser
    Permissions of owner:
    (jeus.security.resource.ResourcePermission jeus.jndi.* lookup)

    Permission owner:
    Role AdministratorsRole
    Permissions of owner:
    (jeus.security.resource.ResourcePermission jeus.* *)
```

4.2.14.17. assign-role

特定ロールをプリンシパルに動的に割り当てます。-classnameで設定したクラスはjava.security.Permissionの抽象クラスを拡張したJavaクラスである必要があります、少なくとも1つの文字列タイプの「role」を取得する構築子を持っている必要があります。デフォルトのクラスはjeus.security.resource.RolePermissionで、contextidは「default」以外のコンテキストを指定する際に使用します。ただし、anonymousの場合は、ターゲット・ロールがunchecked permissionを持っている場合にのみ割り当てが可能です。

- エイリアス

assignrole

- 使用方法

```
assign-role <principal>
           <role-name>
           [-classname <classname>]
           [-domain <domain-name>]
           [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<principal-name>	プリンシパルの名前です
<role-name>	[動的変更] ロールの名前です
[-classname <classname>]	[動的変更] パーミッション・クラスの名前です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> assign-role user1 AdministratorsRole
The role [AdministratorsRole] is assigned to the principal [user1].
```

4.2.14.18. unassign-role

プリンシパルに割り当てられたロールを動的に削除します。

- エイリアス

unassignrole

- 使用方法

```
unassign-role <principal>
              <role-name>
```

```
[-domain <domain-name>]
[-f,--forceLock]
```

- パラメータ

パラメータ	説明
<principal>	プリンシパルの名前です
<role-name>	[動的変更] ロールの名前です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> unassignrole jeus deploy -contextid login -classname
jeus.security.resourc
The role [jndiuser] has been successfully unassigned from the principal [jeus].
```

4.2.14.19. assign-resource

ロールに特定のリソースとアクションを動的に割り当てます。

- エイリアス

assignresource, assignres

- 使用方法

```
assign-resource <role-name>
                 <resource-name>
                 -actions <actions>
                 [-contextid <context-id>]
                 [-classname <classname>]
                 [-domain <domain-name>]
                 [-f,--forceLock]
```

- パラメータ

パラメータ	説明
<role-name>	ロールの名前です
<resource-name>	[動的変更] リソースの名前です。リソース名は通常「jeus.server」のようなリソースを示すJavaクラスの名前です
[-actions <actions>]	[動的変更] ロールに与えられたアクションの一覧です。

パラメータ	説明
	<p>Actionsはリソースに対するアクション一覧で、各アクションはスペースで区切ります。たとえば、jeus.serverリソースの場合、「boot」、「down」というアクションがあります。</p> <p>-actionsパラメータを設定した場合、-classnameで設定したクラスの2番目の構築子パラメータで「actions」を渡す際に使用します。デフォルトのクラス名はjeus.security.resource.ResourcePermissionで、これは「resource」と「actions」を構築子パラメータとして順番に取得します</p>
[-contextid <context-id>]	[動的変更] ポリシーを識別するコンテキストの名前です。contextidは「default」以外のコンテキストを指定する際に使用します
[-classname <classname>]	<p>[動的変更]パーミッション・クラスの名前です。</p> <p>-classnameで設定したクラスはjava.security.Permission抽象クラスを拡張したJavaクラスである必要があり、少なくとも1つの文字タイプの「resource」を受け取る構築子を持っている必要があります。</p> <p>リソース・パーミッションの詳細情報はjeus.security.resource.ResourcePermissionクラスを参照してください。JEUSがチェックするリソース・パーミッションについては『<i>JEUSセキュリティガイド</i>』の「付録B. JEUSサーバーのパーミッション」を参照してください</p>
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> assign-resource deployRole jeus.* -actions
deploy-applications
The resource [jeus.*] is assigned to the role [deployRole].
```

4.2.14.20. unassign-resource

ロールからリソースを動的に削除する際に使用します。ロールは該当リソースにアクセスできなくなります。

- エイリアス

unassignresource, usassignres

- 使用方法

```
unassign-resource <role-name>
                  <resource-name>
```

```
-actions <actions>
[-contextid <context-id>]
[-domain <domain-name>]
[-f,--forceLock]
```

- パラメータ

パラメータ	説明
<role-name>	ロールの名前です
<resource-name>	[動的変更] リソースの名前です
-actions <actions>	[動的変更] ロールに与えられたアクションの一覧です
[-contextid <context-id>]	[動的変更] ポリシーを識別するコンテキストの名前です
[-f,--forceLock]	設定を強制的に変更して反映します

- 例

```
[DAS]domain1.adminServer> unassign-resource deployRole jeus.* -actions
deploy-applications
The resource [jeus.*] has been successfully unassigned from the role [deployRole].
```

4.2.15. ノード管理関連コマンド

本節では、ノード管理に関連するコマンドについて説明します。ノードは1つのマシンに存在する1つのJEUSのインスタレーションを意味します。DASのリモート・サーバーを管理するための機能で、DASに接続した時のみコマンドを使用できます。ノードに関する詳細は『JEUS ノードマネージャガイド』を参照してください。

ノード管理に関連するコマンドの一覧は以下のとおりです。

コマンド	説明
add-java-node	Javaノードを動的に追加します
add-ssh-node	SSHノードを動的に追加します
apply-patch	ノードにパッチを適用します
check-jeus-installed	ノードにJEUSがインストールされているか否かを確認します
check-ssh-node	SSHノードのSSH設定が正常にされているか否かを確認します。
install-jeus	ノードに現在のドメイン構成を含めてJEUSをインストールします
list-nodes	ノードの一覧を確認します
modify-java-node	保存されているJavaノードを修正します
modify-node	保存されているノードを修正します

コマンド	説明
modify-ssh-node	保存されているSSHノードを修正します
patch-info	ノードに適用されているパッチ情報を確認します
remove-node	保存されているノードを削除します
remove-patch	ノードに適用されているパッチを削除します
show-node	ノードの設定を確認します
uninstall-jeus	ノードからJEUSを削除します

4.2.15.1. add-java-node

新しいJavaノードを動的に追加します。各オプションに対して適切な値を入力し、ノードに設定されているJavaノード・マネージャーの接続に問題がないようにします。

- エイリアス

addjavanode

- 使用方法

```
add-java-node <node-name>
               -host <host-name>
               [-port <port-number>]
               [-ssl]
               [-truststore <truststore-path>]
               [-truststorepassword <truststore-password>]
```

- パラメータ

パラメータ	説明
<node-name>	追加するノードの名前です。IDとして使用される値であるため、一意である必要があります
-host <host-name>	該当ノードのノード・マネージャーで使用するホスト・アドレスです
[-port <port-number>]	該当ノードのノード・マネージャーで使用するポート番号です(デフォルト値: 7730)
[-ssl]	SSLを使用するか否かを設定します
[-truststore <truststore-path>]	トラストストア・ファイルのパスを設定します
[-truststorepassword <truststore-password>]	トラストストア・ファイルのパスワードを設定します

- 例

```
[DAS]domain1.adminServer> add-java-node node1 -host 192.168.34.55 -port 7730
The node [node1] was successfully added.
```

4.2.15.2. add-ssh-node

新規のSSHノードを動的に追加します。各オプションに対して適切な値を入力し、SSH接続に問題がないようにします。SSH接続が可能か否かは[check-ssh-node](#)コマンドを使用して確認できます。また、作成されたSSHノードは[modify-ssh-node](#)コマンドを使用して修正できます。

- エイリアス

addsshnode

- 使用方法

```
add-ssh-node <node-name>
             -host <host-name>
             [-dir <install-dir> ]
             [-port <port-number> ]
             [-user <user-name> ]
             [-privatekey <file-path> ]
```

- パラメータ

パラメータ	説明
<node-name>	追加するリモート・ノードの名前です。IDとして使用される値であるため、一意である必要があります
-host <host-name>	リモート・ノードのホスト・アドレスです
[-dir <install-dir>]	リモート・ノードのJEUSインストール・ディレクトリーです
[-port <port-number>]	SSH接続のポート番号です(デフォルト値: 22)
[-user <user-name>]	SSH接続のユーザー名です
[-privatekey <file-path>]	SSH接続のための秘密鍵のファイル・パスです

- 例

```
[DAS]domain1.adminServer> add-ssh-node node1 -host 192.168.34.55 -dir /home/sshU
ser/jeus -user sshUser
The node [node1] was successfully added.
```

4.2.15.3. apply-patch

設定したノードにパッチ・ファイルを適用します。

- エイリアス

applypatch

- 使用方法

```
apply-patch [<file-name>]
            [-nodes <node-name>]
            [-rolling]
            [-nodas]
            [-action <action-on-error>]
```

- パラメータ

パラメータ	説明
[<file-name>]	適用するパッチ・ファイルを設定します。設定していない場合、JEUS_HOME/lib/jextディレクトリ全体を転送します
[-nodes <node-name>]	パッチを適用するノード・リストを設定します。設定していない場合、すべてのドメイン内に存在するノードに適用します
[-rolling]	パッチ・ファイルの転送後、サーバーを再起動して適用するか否かを設定します
[-nodas]	DASは再起動しません
[-action <action-on-error>]	パッチ・ファイルの適用に失敗した場合の動作について設定します。 以下のうち1つを設定します – CONTINUE – ROLLBACK – STOP

- 例

```
[DAS]domain1.adminServer>apply-patch -nodes node2 jext.jar
The patch was successfully applied to the nodes [node1].
```

4.2.15.4. check-jeus-installed

特定ノードにJEUSがインストールされているか否かを確認します。結果値として該当ノードのJEUSバージョン情報が出力されます。

- エイリアス

checkjeusinstalled, isjeusinstalled

- 使用方法

```
check-jeus-installed <node-name>
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です

- 例

```
[DAS]domain1.adminServer> check-jeus-installed node1
The JEUS version on the node [node1] : JEUS 8.0 (Fix#0)
```

4.2.15.5. check-ssh-node

特定のSSHノードにJavaコマンドを実行すると、SSH設定が正常にされているか否かを確認します。

- エイリアス

checksshnode, checknode

- 使用方法

```
check-ssh-node <node-name>
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です

- 例

```
[DAS]domain1.adminServer> check-ssh-node node1
The Domain Administration Server can execute the "java" process via SSH.
```

4.2.15.6. install-jeus

特定ノードに現在のドメイン設定を含めてJEUSをインストールします。ファイルの圧縮およびコピー作業が行われるため、環境によって長時間必要な場合もあります。

- エイリアス

installjeus

- 使用方法

```
install-jeus <node-name>
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です

- 例

```
[DAS]domain1.adminServer> install-jeus node1
JEUS was successfully installed on the node [node1].
```

4.2.15.7. list-nodes

現在構成されているノードの一覧を出力します。ノードの名前、タイプ、制御機能の可否、JEUSのインストール・バージョンが結果値として出力されます。

- エイリアス

listnodes, nodelist

- 使用方法

```
list-nodes [-onlyname]
```

- パラメータ

パラメータ	説明
[-onlyname]	ノード名のリストのみを表示します

- 例

```
[DAS]domain1.adminServer> list-nodes
=====
+-----+-----+-----+-----+
| Node name | Type | Under control | JEUS version |
+-----+-----+-----+-----+
| node1     | SSH | Y              | -             |
| node2     | SSH | Y              | JEUS 8.0 (Fix#0) |
| node3     | JAVA | N              | -             |
+-----+-----+-----+-----+
=====
```

4.2.15.8. modify-java-node

Javaノードの設定を変更します。

- エイリアス

modifyjavanode

- 使用方法

```
modify-java-node <node-name>
    [-host <host-name>]
    [-port <port-number>]
    [-ssl <enable-SSL>]
    [-truststore <truststore-path>]
    [-truststorepassword <truststore-password>]
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です
[-host <host-name>]	ノードのノード・マネージャーで使用するホスト・アドレスです
[-port <port-number>]	ノードのノード・マネージャーで使用するポート番号です(デフォルト値: 7736)
[-ssl <enable-SSL>]	SSLを使用するか否かを設定します
[-truststore <truststore-path>]	トラストストア・ファイルのパスを設定します
[-truststorepassword <truststore-password>]	トラストストア・ファイルのパスワードを設定します

- 例

```
[DAS]domain1.adminServer> modify-java-node node1 -port 7731
The node [node1] was modified successfully. Check the results using "show-node"
```

4.2.15.9. modify-node

ノードの設定を変更します。ノード・マネージャーのタイプを変更できます。

- エイリアス

modifyjavanode

- 使用方法

```

modify-node <node-name>
    [-type <node-type>]
    [-host <host-name>]
    [-port <port-number>]
    [-ssl]
    [-truststore <truststore-path>]
    [-truststorepassword <truststore-password>]
    [-dir <install-dir>]
    [-user <user-name>]
    [-privatekey <file-path>]

```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です
[-type <host-name>]	ノードのタイプを変更します。javaまたはsshのうち1つの値を入力できます
[-host <host-name>]	該当するノードのノード・マネージャーで使用するホスト・アドレスです
[-port <port-number>]	該当するノードのノード・マネージャーで使用するポート番号です (デフォルト値: JAVAタイプの場合は7730、SSHタイプの場合は22)
[-ssl]	SSLを使用するか否かを設定します
[-truststore <truststore-path>]	トラストストア・ファイルのパスを設定します
[-truststorepassword <truststore-password>]	トラストストア・ファイルのパスワードを設定します
[-dir <install-dir>]	該当するノードにJEUSがインストールされたパスを設定します
[-user <user-name>]	SSH接続ユーザー名です
[-privatekey <file-path>]	SSH接続のための個人キー・ファイルのパスです

- 例

```

[DAS]domain1.adminServer> modify-node node1 -type ssh -port 23
The node [node1] was modified successfully. Check the results using "show-node"

```

4.2.15.10. modify-ssh-node

SSHノードの設定を変更します。

- エイリアス

modifysshnode, modifynode

- 使用方法

```
modify-ssh-node <node-name>
    [-host <host-name>]
    [-dir <install-dir>]
    [-port <port-number>]
    [-user <user-name>]
    [-privatekey <file-path>]
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です
[-host <host-name>]	リモート・ノードのホスト・アドレスです
[-dir <install-dir>]	リモート・ノードのJEUSインストール・ディレクトリーです
[-port <port-number>]	SSH接続のポート番号です(デフォルト値: 22)
[-user <user-name>]	SSH接続のユーザー名です
[-privatekey <file-path>]	SSH接続のための秘密鍵のファイル・パスです

- 例

```
[DAS]domain1.adminServer> modify-ssh-node node1 -port 23
The node [node1] was modified successfully. Check the results using "show-node"
```

4.2.15.11. patch-info

ノードに適用されたパッチを確認します。

- エイリアス

patchinfo, patch-list, patchlist, list-patch, listpatch

- 使用方法

```
patch-info [<node-names>]
```

- パラメータ

パラメータ	説明
[<node-names>]	ノードの名前です。ノードが1つ以上の場合はコンマ(,)で区分します

- 例


```
[DAS]domain1.adminServer>patch-info
Successfully got patch information for the nodes.
=====
Node Name[node1]

+-----+-----+
|           Patch File Name           | Patch Contents |
+-----+-----+
(No data available)

Unable to get patch information to SSH NodeManager
=====

=====
Node Name[node2]

+-----+-----+
|           Patch File Name           | Patch Contents |
+-----+-----+
| jext_patch_test.jar                 | patch test     |
+-----+-----+
=====
```

4.2.15.12. remove-node

ノードを削除します。

- エイリアス

removenode, rmnode, rm-node

- 使用方法

```
remove-node <node-name>
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です

- 例

```
[DAS]domain1.adminServer> remove-node node1
The node [node1] was successfully removed.
```

4.2.15.13. remove-patch

設定したノードに適用されたパッチ・ファイルを削除します。

- エイリアス

removepatch, rmpatch, rm-patch

- 使用方法

```
remove-patch [<file-name>]
              [-nodes <node-name>]
              [-rolling]
              [-nodas]
              [-action <action-on-error>]
```

- パラメータ

パラメータ	説明
[<file-name>]	削除するパッチ・ファイルを設定します。設定していない場合、JEUS_HOME/lib/jextディレクトリ全体を削除します
[-nodes <node-name>]	パッチを削除するノード・リストを設定します。設定していない場合、すべてのドメイン内に存在するノードから削除します
[-rolling]	パッチ・ファイルの削除後、サーバーを再起動する際に、削除内容を適用するか否かを設定します
[-nodas]	DASは再起動しません
[-action <action-on-error>]	パッチ・ファイルの削除に失敗した場合の動作について設定します。 以下のうち1つを設定します – CONTINUE – STOP – ROLLBACK

- 例

```
[DAS]domain1.adminServer>remove-patch -nodes node2 jext.jar
The nodes [node2] have successfully removed patch file.
```

4.2.15.14. show-node

ノードの情報を出力します。ノードの基本情報以外に、マッピングされたサーバーの情報も出力します。

基本のSSHノード情報は`modify-ssh-node`コマンドを使用して変更可能です。マッピングされたサーバーの情報は`modify-server`コマンドを使用して変更可能です。

- エイリアス

`shownode, nodeinfo`

- 使用方法

```
show-node <node-name>
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です

- 例

```
[DAS]domain1.adminServer> show-node node1
=====
+-----+-----+
| Property | Value |
+-----+-----+
| Node Name | node1 |
| Host      | 192.168.34.55 |
| Mapped Servers | adminServer server1 |
| Node Type | JAVA |
| NodeManager Port | 7730 |
| Use SSL   | false |
+-----+-----+
=====
```

4.2.15.15. uninstall-jeus

ノードにインストールされているJEUSを削除します。

- エイリアス

`uninstalljeus`

- 使用方法

```
uninstall-jeus <node-name>
```

- パラメータ

パラメータ	説明
<node-name>	ノードの名前です

- 例

```
[DAS]domain1.adminServer> uninstall-jeus node1
JEUS was successfully uninstalled from the node [node1].
```

4.2.16. ノード・マネージャー関連のコマンド

本節では、ノード・マネージャーに接続して使用できるコマンドについて説明します。ノードは1つのマシンに存在する1つのJEUSインストール版を意味します。Javaタイプのノード・マネージャーに接続してサーバーを起動するか状態を確認します。Javaタイプのノード・マネージャーの詳細については『JEUS ノードマネージャガイド』の『JEUS ノードマネージャガイド』を参照してください。

ノード・マネージャーに関連するコマンドの一覧は以下のとおりです。

コマンド	説明
connect-nodemanager	ノード・マネージャーに接続します
disconnect-nodemanager	ノード・マネージャーとの接続を切断します。ノード・マネージャーと接続した状態でのみ使用可能なコマンドです
nm-start-server	ノード・マネージャーを使用してサーバーを起動します。ノード・マネージャーと接続した状態でのみ使用可能なコマンドです
nm-state-server	ノード・マネージャーを使用して起動したサーバーの状態を確認します。ノード・マネージャーと接続した状態でのみ使用可能なコマンドです
nm-stop-server	ノード・マネージャーを使用して起動したサーバーを終了します。ノード・マネージャーと接続した状態でのみ使用可能なコマンドです
stop-nodemanager	ノード・マネージャーを終了します。ノード・マネージャーと接続した状態でのみ使用可能なコマンドです

4.2.16.1. connect-nodemanager

ノード・マネージャーに接続します。

- エイリアス

connect-nm, connectnm, nm-connect, nmconnect

- 使用方法

```
connect-nodemanager [-domain <domain-name>]
                    [-h,--host <host-address>]
                    [-p,--port <port>]
                    [-t,--type <connect-type>]
                    [-ts,--truststoreFile <truststore-file>]
                    [-tsp,--truststorePass <truststore-password>]
```

- パラメータ

パラメータ	説明
[-domain <domain-name>]	ノード・マネージャーで扱うドメインの名前です
[-h,--host <host-address>]	接続するノード・マネージャーのホスト・アドレスです
[-p,--port <port>]	接続するノード・マネージャーのポート番号です
[-t,--type <connect-type>]	ノード・マネージャーに接続する際にSSLを使用するか否かを設定します。 以下のうち1つを設定します – ssl – plain(デフォルト値)
[-ts,--truststoreFile <truststore-file>]	SSLを使用する際のトラストストア・ファイルのパスを設定します
[-tsp,--truststorePass <truststore-password>]	SSLを使用する際のトラストストア・ファイルのパスワードを設定します

- 例

```
offline>connect-nodemanager -host 192.168.34.55 -port 7730 -domain domain1
The connection to the node manager domain1 has been established.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.16.2. disconnect-nodemanager

ノード・マネージャーとの接続を切断します。

- エイリアス

disconnect-nm, disconnectnm, disconnnm, nm-disconnect, nmdisconnect

- 使用方法

```
disconnect-nodemanager
```

- 例

```
[NodeManager]domain1>disconnect-nodemanager  
disconnect to node manager.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.16.3. nm-start-server

ノード・マネージャーを使用してサーバーを起動します。

- エイリアス

nmstart, nmstartserver

- 使用方法

```
nm-start-server [-domain <domain-name>]  
                -server <server-name>  
                -u,--user <user-name>  
                -p,--password <password>  
                [-dasurl <dasurl>]  
                [-f,--force]  
                [-s,--standby]  
                [-host <nm-host>]  
                [-port <nm-port>]  
                [-t,--type <connect-type>]  
                [-ts,--truststoreFile <truststore-file>]  
                [-tsp,--truststorePass <truststore-password>]
```

- パラメータ

パラメータ	説明
[-domain <domain-name>]	起動するサーバーが属するドメインの名前です
-server <server-name>	起動するサーバーの名前です
-u,--user <user-name>	サーバーを起動する際に必要なアカウント情報です

パラメータ	説明
<code>-p,--password <password></code>	サーバーを起動する際に必要なアカウントのパスワードです
<code>[-dasurl <dasurl>]</code>	MSを起動する際に必要なDASのURL情報です
<code>[-f, --force]</code>	MSを起動する際にアプリケーションのデプロイが失敗しても、RUNNING状態にするための設定です
<code>[-s, --standby]</code>	MSを起動する際にSTANDBY状態までのみにするための設定です
<code>[-host <nm-host>]</code>	ノード・マネージャーに接続されていない状態の場合、ノード・マネージャーのホスト情報をオプションとして指定し、ノード・マネージャーに接続した後、サーバーを起動できます
<code>[-port <nm-port>]</code>	ノード・マネージャーに接続されていない状態の場合、ノード・マネージャーのポート情報をオプションに指定し、ノード・マネージャーに接続した後、サーバーを起動できます
<code>[-t --type <connect-type>]</code>	ノード・マネージャーに接続する際にSSLを使用するか否かを設定します。 以下のうち1つを設定します – ssl – plain(デフォルト値)
<code>[-ts,--truststoreFile <truststore-file>]</code>	SSLを使用する際のトラストストア・ファイルのパスを設定します
<code>[-tsp,--truststorePass <truststore-password>]</code>	SSLを使用する際のトラストストア・ファイルのパスワードを設定します

- 例

```
[NodeManager]domain1>nm-start-server -server server1 -u jeus -p jeus
succeed to start server[server1].
RUNNING
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.16.4. nm-state-server

ノード・マネージャーを使用して起動したサーバーの状態を確認します。

- エイリアス

nmstate-server, nmstateserver

- 使用方法

```
nm-state-server [-domain <domain-name>]
                 -server <server-name>
                 -u,--user <user-name>
                 -p,--password <password>
```

- パラメータ

パラメータ	説明
[-domain <domain-name>]	状態を確認するサーバーが属するドメインの名前です
-server <server-name>	状態を確認するサーバーの名前です
-u,--user <user-name>	サーバーの状態を確認するためにサーバーに接続した場合のアカウント情報です
-p,--password <password>	サーバーの状態を確認するためにサーバーに接続した場合のアカウントのパスワードです

- 例

```
[NodeManager]domain1>nm-state-server -server server1 -u jeus -p jeus
server[server1] : RUNNING
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.16.5. nm-stop-server

ノード・マネージャーを使用して起動したサーバーの状態を確認します。

- エイリアス

nmstop-server, nmstopserver

- 使用方法

```
nm-stop-server [-domain <domain-name>]
                -server <server-name>
                -u,--user <user-name>
                -p,--password <password>
```

- パラメータ

パラメータ	説明
[-domain <domain-name>]	終了させるサーバーが属するドメインの名前です
-server <server-name>	終了させるサーバーの名前です
-u,--user <user-name>	サーバーを終了させるためにサーバーに接続した場合のアカウント情報です
-p,--password <password>	サーバーを終了させるためにサーバーに接続した場合のアカウントのパスワードです

- 例

```
[NodeManager]domain1>nm-stop-server -server server1 -u jeus -p jeus
succeed to stop server[server1].
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.16.6. stop-nodemanager

ノード・マネージャーを終了します。

- エイリアス

stop-nm, stopnm, nm-stop, nmstop

- 使用方法

```
stop-nodemanager [-host <host-address>]
                  [-port <port>]
                  [-p --properties <properties-file>]
                  [-t, --type <connect-type>]
                  [-ts, --truststoreFile <truststore-file>]
                  [-tsp, --truststorePass <truststore-password>]
                  [-stopServer]
                  [-verbose]
```

- パラメータ

パラメータ	説明
[-host <host-address>]	終了するノード・マネージャーのホスト情報です
[-port <port>]	終了するノード・マネージャーのポート情報です
[-p --properties <properties-file>]	コマンドの実行に必要な設定値を含むファイルを設定します

パラメータ	説明
<code>[-t,--type <connect-type>]</code>	ノード・マネージャーに接続する際にSSLを使用するか否かを設定します。 以下のうち1つを設定します – ssl – plain(デフォルト値)
<code>[-ts,--truststoreFile <truststore-file>]</code>	SSLを使用する際のトラストストア・ファイルのパスを設定します
<code>[-tsp,--truststorePass <truststore-password>]</code>	SSLを使用する際のトラストストア・ファイルのパスワードを設定します
<code>[-stopServer]</code>	ノード・マネージャーが管理しているサーバーも一緒に終了します
<code>[-verbose]</code>	実行結果を詳細出力します

- 例

```
[NodeManager]domain1>stop-nodemanager
Succeed to stop the node manager.
```

- 参考

このコマンドは、サーバーに接続していない状態でも使用できます。

4.2.17. 設定編集コマンド

本節では、JEUSの設定を編集するコマンドについて説明します。設定編集コマンドを使用すると、コンソールを通じてJEUSの動的設定変更機能を利用できます。

参考

動的設定変更機能についての基本的な内容は、『*JEUSドメインガイド*』の「第3章ドメインの設定変更」を参照してください。

個別コマンドについて説明する前に、先に知っておくべき事項について以下で説明します。

- 設定編集コマンドで編集するJEUSの設定について

設定編集コマンドを使って編集する対象は、JEUSサーバーが設定を保存するために使用するJavaオブジェクトです。このオブジェクトはJEUSの設定を記述するdomain.xmlを定義したXMLスキーマを、JAXB

を使用してJavaオブジェクトに変換したものです。そのため、オブジェクトの構造や各項目の名前などがdomain.xmlのそれと似てはいますが、一部については異なって記述されることもあります。

- 単純型(Simple Type)

設定項目のうち子項目を持たないものを指します。

- 複雑型(Complex Type)

子項目を持つ設定項目です。子項目として単純型要素と複雑型要素、単純型要素を含むリストと複雑型要素を含むリストを持つことができます。

- リスト(List)

単純型や複雑型の要素を持つリスト形式の設定項目です。

- パス(Path)

設定編集コマンドではJEUS設定の各項目の位置をパスで表現します。パスは、設定項目の名前や複雑型要素の識別子、リストで対象の位置を表すための索引番号(index)で構成されます。各項目はスラッシュ(/)で区分します。

1. ルート・パス(root)は、1つのスラッシュで表現します。
2. 現在の位置は、1つの点(dot)で表現します。
3. 現在の位置の親項目は、2つの点(dot)で表現します。たとえば、「/servers」項目の親項目のパス名は「/servers/..」で表現します。
4. ほとんどの場合、設定項目の名前が該当項目のパス名になります。たとえば、ルート・パスrootの配下にあるserversという項目のパスは、「/servers」で表現します。servers項目が子項目を持ち、その項目のうち1つの名前がserverであるとしたら、server項目を表すパスは「/servers/server」です。
5. 複雑型要素を持つリストの場合、該当複雑型要素の識別子の値をパス名として使用します。識別子は、複数の同じ形式の複雑型要素が存在する場合に、特定の複雑型要素を識別できる固有の値を持つ子項目をいいます。たとえば、「/servers/server」はdomain設定で各サーバーの設定を含む複数の複雑型要素で構成されたリストです。サーバーの設定を含む複雑型要素で、識別子はサーバー名を使用するように指定されています。したがって、server1の設定項目のパス名は「/servers/server/server1」に表現できます。
6. 単純型要素を持つリストの場合、複雑型要素とは違って識別子が存在しないため、該当リストで何番目の要素であるかを表す索引番号(index)をパス名として使用します。索引番号は角かっこの中に0以上の整数を記述する形式で指定します。たとえば、「/a/b」というパス名があり、bが単純型要素で構成されたリストであるとしたら、該当リストの最初の要素は「/a/b/[0]」で表現します。

参考

単純型リストのパスを表現する際に、最初の要素を[1]ではなく[0]に表現することに注意してください。

7. それぞれの設定編集コマンドは、現在のパスを指定または変更する場合があります。現在のパスはサーバーに保存されるため、複数のjeusadminを実行して同時に複数の場所で設定の編集を行う場合、現在のパスが意図せずに変更されてコマンドが正常に実行されないことがあります。したがって、設定を編集するときは、なるべく1つのjeusadminを使用することをお勧めします。

以下は、設定編集コマンドをまとめた表です。すべてのコマンドはサーバーに接続している状態でのみ使用できます。

コマンド	説明
start-configuration-editing	サーバーに動的設定変更のためのロックを要求します
change-current-path	現在のパスを変更または確認します
show-element	指定したパスに存在する要素の情報を出力します
create-empty-element	指定した要素に空の複雑型要素を作成したり、複雑型要素を持つリストに新しい要素を追加します
set-element-value	指定した単純型要素の値を変更したり、単純型要素を持つリストに新しい要素を追加します
delete-element	指定したパスに存在する要素を削除します
activate-configuration	変更した設定をサーバーに反映して結果を出力します
cancel	動的設定変更のためのロックを解除し、サーバーに保存されている変更内容を削除します。このコマンドはロックを持っていない状態でも使用できます

4.2.17.1. start-configuration-editing

サーバーに要求して動的設定変更のためのロックを取得します。ロックを正常に取得したら、サーバーの現在のパスを初期化します。ロックを取得できなかった場合は、以降cancel以外のコマンドは動作しないため、設定を編集するためには、必ずこのコマンドを先に実行してロックを取得する必要があります。

- エイリアス

lock-and-edit, edit

- 使用方法

```
start-configuration-editing [-f]
```

- パラメータ

パラメータ	説明
[-f]	他のユーザーが動的設定変更のためのロックを持っている場合も、強制的にロックの取得を試みます

- 例

```
[DAS]domain1.adminServer>start-configuration-editing
Successfully acquired a configuration lock from Domain Administration Server.
```

4.2.17.2. change-current-path

指定したパスを現在のパスに変更します。変更するパスを指定しない場合、現在のパスを出力します。

- エイリアス

```
cd
```

- 使用方法

```
change-current-path [<path>]
```

- パラメータ

パラメータ	説明
<path>	変更するパスを指定します。相対パスと絶対パスの両方を使用できます

- 例

```
[DAS]domain1.adminServer>change-current-path servers/server/adminServer
Current path is changed to /servers/server/adminServer
[DAS]domain1.adminServer>change-current-path ..
Current path is changed to /servers/server
[DAS]domain1.adminServer>change-current-path .
Current path is changed to /servers/server
[DAS]domain1.adminServer>change-current-path /
Current path is changed to /
[DAS]domain1.adminServer>change-current-path
servers/server/server1/../../../../server/adminServer
Current path is changed to /servers/server/adminServer
[DAS]domain1.adminServer>cd
/servers/server/adminServer
```

4.2.17.3. show-element

指定したパスに存在する要素の情報を出力します。出力情報は、要素の名前、値、形式(type)などです。パスを指定しない場合、現在のパスを出力します。

- エイリアス

ls

- 使用方法

```
show-element [<path>]
```

- パラメータ

パラメータ	説明
[<path>]	情報を出力する要素のパスを指定します

- 例

- 複雑型要素の情報を出力した例です。要素の名前、値(存在する場合)、値の形式を出力することが確認できます。

```
[DAS]domain1.adminServer>show-element

name                adminServer
nodeName            node1
listeners            complex type
logHome              Not specified (string      )
group                Not specified (string      )
jvmConfig            complex type
userInterceptor      Not specified (complex type)
userLogging          Not specified (complex type)
systemLogging        list of [systemLogging]
logStdoutToRawFormat true
actionOnResourceLeak WARNING (enum[NO_ACTION, WARNING,
AUTO_CLOSE])
tmConfig             complex type
externalResource     list of [externalResource]
managedExecutorService list of [string]
managedScheduledExecutorService list of [string]
contextService       list of [string]
managedThreadFactory list of [string]
scheduler            Not specified (complex type)
namingServer         complex type
jmxManager           Not specified (complex type)
systemThreadPool     complex type
```

```

dataSourceRemoteLookup      false
engineInitOnStartup         true
useWebEngine                 true
useEjbEngine                 true
useJmsEngine                 true
webEngine                   complex type
ejbEngine                    complex type
jmsEngine                    complex type
useMEJB                      false
classFtp                     true
enableInterop                Not specified (complex type)
lifecycleInvocation          list of [lifecycleInvocation]
resRef                       complex type
jmsResource                  complex type
dataSources                  Not specified (complex type)
customResourceRefs           Not specified (complex type)
externalResourceRefs         Not specified (complex type)
autoGenerated                Not specified (boolean      )

[DAS]domain1.adminServer>

```

- 複雑型要素を持つリストの情報を出力した例です。各項目の名前は識別子の値で出力され、該当リストに含められる値の形式と識別子が出力されることが確認できます。

```

[DAS]domain1.adminServer>show-element /servers/server
  adminServer
  server1

Current list can contain the following type of elements:
  [server]
The name of the identifier element of [server] element is [name].

```

4.2.17.4. create-empty-element

指定した場所に新しい複雑型要素を作成します。生成される複雑型要素は空の状態であり、追加作業により値を与えます。複雑型要素を持つリストの場合は、識別子の値を要素名として使用すると、指定した値を識別子の値とする複雑型要素を作成してリストに追加します。

- エイリアス

create

- 使用方法

```

create-empty-element <name>
                      [<type>]

```

- パラメータ

パラメータ	説明
<name>	作成する要素の名前を指定します。複雑型要素を持つリストに新しい複雑型要素を追加する場合、要素名として指定の値を識別子の値とする複雑型要素を作成します
[<type>]	作成する要素の形式を指定します。一般的には指定しなくても構いませんが、複数の形式を要素として持つリストの場合は、可能な形式のうち作成する形式を指定する必要があります

- 例

- 空の複雑型要素を作成する例です。

```
[DAS]domain1.adminServer>cd /
[DAS]domain1.adminServer>create-empty-element passwordValidator
Created an empty complex type of passwordValidator at path /
```

- 複雑型要素を持つリストに新しい複雑型要素を追加する例です。「server2」を識別子のサーバー名として使用する新しいserver項目が作成され、リストに追加されていることが確認できます。

```
[DAS]domain1.adminServer>cd /servers/server
Current path is changed to /servers/server
[DAS]domain1.adminServer>ls
  adminServer
  server1

Current list can contain the following type of elements:
[server]
The name of the identifier element of [server] element is [name].

[DAS]domain1.adminServer>create-empty-element server2
Added a new element of type server with id [name == server2] to the list at
path /servers/server
[DAS]domain1.adminServer>ls
  adminServer
  server1
  server2

Current list can contain the following type of elements:
[server]
The name of the identifier element of [server] element is [name].

[DAS]domain1.adminServer>
```


- typeオプションを使用して作成する複雑型要素の形式を指定する例です。show-elementコマンドの結果に現在のリストが複数の形式の項目を要素として持つことが確認できます。形式を指定しない場合、案内メッセージが出力されることも確認できます。

```
[DAS]domain1.adminServer>cd
/servers/server/adminServer/systemLogging/jeus/handler/
fileHandlerOrSmtpHandlerOrSocketHandler
[DAS]domain1.adminServer>ls
    fileHandler

Current list can contain the following type of elements:
    [fileHandler, userHandler, smtpHandler, socketHandler]
The name of the identifier element of [fileHandler, userHandler,
smtpHandler, socketHandler] element is [name].

[DAS]domain1.adminServer>create-empty-element handler1
Must specify an element type to create a new element. Available types are
fileHandler, userHandler, smtpHandler, socketHandler.
[DAS]domain1.adminServer>create-empty-element handler1 userHandler
Added a new element of type userHandler with id [name == handler1] to the list
at path
/servers/server/adminServer/systemLogging/jeus/handler/fileHandlerOrSmtpHandlerOrSocketHandler
[DAS]domain1.adminServer>ls
    fileHandler
    handler1

Current list can contain the following type of elements:
    [fileHandler, userHandler, smtpHandler, socketHandler]
The name of the identifier element of [fileHandler, userHandler,
smtpHandler, socketHandler] element is [name].
```

4.2.17.5. set-element-value

指定した単純型要素の値を変更したり、単純型要素を持つリストに値を追加するときに使用します。このコマンドで値を削除することはできません。

- エイリアス

set

- 使用方法

```
set-element-value <path><value>
```

- パラメータ

パラメータ	説明
<path>	値を変更する要素のパスを指定します
<value>	変更する値を指定します

- 例

- set-element-valueコマンドを使用してdescription項目に「Hello, world」という値を設定する例です。

```
[DAS]domain1.adminServer>cd /
Current path is changed to /
[DAS]domain1.adminServer>ls
description                                Not specified (string      )
productionMode                             true
id                                           982883233
securityManager                             complex type
passwordValidator                           Not specified (complex type)
adminServerName                             adminServer
domainLogHome                               Not specified (string      )
systemClusteringFramework                  complex type
domainBackup                               complex type
enableWebadmin                             true
enableJsonCommand                           false
enableToResynchronizeApplications          false
servers                                    complex type
sessionCluster                             complex type
clusters                                   Not specified (complex type)
serverTemplates                             Not specified (complex type)
lifecycleInvocation                         list of [lifecycleInvocation]
applicationRepositories                     Not specified (complex type)
deployedApplications                        Not specified (complex type)
deployedLibraries                           Not specified (complex type)
resources                                  Not specified (complex type)

[DAS]domain1.adminServer>set-element-value description "Hello,world"
The value Hello,world is set at the specified location /description.
[DAS]domain1.adminServer>ls
description                                Hello,world
productionMode                             true
id                                           982883233
securityManager                             complex type
passwordValidator                           Not specified (complex type)
adminServerName                             adminServer
domainLogHome                               Not specified (string      )
systemClusteringFramework                  complex type
domainBackup                               complex type
enableWebadmin                             true
```

enableJsonCommand	false
enableToResynchronizeApplications	false
servers	complex type
sessionCluster	complex type
clusters	Not specified (complex type)
serverTemplates	Not specified (complex type)
lifecycleInvocation	list of [lifecycleInvocation]
applicationRepositories	Not specified (complex type)
deployedApplications	Not specified (complex type)
deployedLibraries	Not specified (complex type)
resources	Not specified (complex type)

- 単純型要素を持つリストに値を追加する例です。値を追加するリストのパスを要素名として使用することが確認できます。

```
[DAS]domain1.adminServer>start-configuration-editing
Successfully acquired a configuration lock from Domain Administration Server.
[DAS]domain1.adminServer>cd lifecycleInvocation
Current path is changed to /lifecycleInvocation
[DAS]domain1.adminServer>create-empty-element com.tmax.sample
Added a new element of type lifecycleInvocation with id [className ==
com.tmax.sample] to the list at path /lifecycleInvocation
[DAS]domain1.adminServer>cd com.tmax.sample
Current path is changed to /lifecycleInvocation/com.tmax.sample
[DAS]domain1.adminServer>cd invocation
Current path is changed to /lifecycleInvocation/com.tmax.sample/invocation
[DAS]domain1.adminServer>create-empty-element invocation
Added a new element of type invocation to the list at path
/lifecycleInvocation/com.tmax.sample/invocation
[DAS]domain1.adminServer>cd [0]
Current path is changed to /lifecycleInvocation/com.tmax.sample/invocation/[0]
[DAS]domain1.adminServer>ls
    invocationMethod          Not specified (complex type)
    invocationArgument        list of [string]
    invocationType             Not specified (enum[BOOT,
                                BEFORE_DEPLOY, AFTER_DEPLOY, READY,
                                BEFORE_UNDEPLOY, AFTER_UNDEPLOY])

[DAS]domain1.adminServer>set-element-value invocationArgument arg0
The value arg0 added to the list in the specified location:
/lifecycleInvocation/com.tmax.sample/invocation/[0]/invocationArgument.
[DAS]domain1.adminServer>cd invocationArgument
Current path is changed to
/lifecycleInvocation/com.tmax.sample/invocation/[0]/invocationArgument
[DAS]domain1.adminServer>ls
    [0]                        arg0
```

```

Current list can contain the following type of elements:
    [string]

[DAS]domain1.adminServer>set-element-value . arg1
The value arg1 added to the list in the specified location:
/lifecycleInvocation/com.tmax.sample/invocation/[0]/invocationArgument.
[DAS]domain1.adminServer>ls
    [0]                                arg0
    [1]                                arg1

Current list can contain the following type of elements:
    [string]

```

4.2.17.6. delete-element

指定したパスに存在する要素を削除します。

- エイリアス

delete, rm

- 使用方法

```
delete-element <path>
```

- パラメータ

パラメータ	説明
<path>	値を削除する要素のパスを指定します

- 例

```

[DAS]domain1.adminServer>cd /
Current path is changed to /
[DAS]domain1.adminServer>ls
description                Hello,world
productionMode              true
id                           982883233
securityManager              complex type
passwordValidator            Not specified (complex type)
adminServerName              adminServer
domainLogHome                Not specified (string      )
systemClusteringFramework   complex type
domainBackup                 complex type
enableWebadmin               true

```

```

enableJsonCommand           false
enableToResynchronizeApplications false
servers                     complex type
sessionCluster              complex type
clusters                   Not specified (complex type)
serverTemplates             Not specified (complex type)
lifecycleInvocation         list of [lifecycleInvocation]
applicationRepositories     Not specified (complex type)
deployedApplications        Not specified (complex type)
deployedLibraries           Not specified (complex type)
resources                   Not specified (complex type)

```

```
[DAS]domain1.adminServer>delete-element description
```

```
Successfully delete the target element at /description.
```

```
[DAS]domain1.adminServer>ls
```

```

description           Not specified (string      )
productionMode         true
id                     982883233
securityManager        complex type
passwordValidator       Not specified (complex type)
adminServerName        adminServer
domainLogHome          Not specified (string      )
systemClusteringFramework complex type
domainBackup           complex type
enableWebadmin         true
enableJsonCommand      false
enableToResynchronizeApplications false
servers                complex type
sessionCluster         complex type
clusters               Not specified (complex type)
serverTemplates        Not specified (complex type)
lifecycleInvocation    list of [lifecycleInvocation]
applicationRepositories Not specified (complex type)
deployedApplications   Not specified (complex type)
deployedLibraries      Not specified (complex type)
resources              Not specified (complex type)

```

4.2.17.7. activate-configuration

変更した内容をサーバーに反映して結果を出力します。

- エイリアス

```
activate
```

- 使用方法

● 例

```
[DAS]domain1.adminServer>cd
/
[DAS]domain1.adminServer>ls
description                Not specified (string      )
productionMode              true
id                          982883233
securityManager             complex type
passwordValidator           Not specified (complex type)
adminServerName             adminServer
domainLogHome               Not specified (string      )
systemClusteringFramework  complex type
domainBackup                complex type
enableWebadmin              true
enableJsonCommand           false
enableToResynchronizeApplications false
servers                     complex type
sessionCluster              complex type
clusters                    Not specified (complex type)
serverTemplates             Not specified (complex type)
lifecycleInvocation         list of [lifecycleInvocation]
applicationRepositories     Not specified (complex type)
deployedApplications        Not specified (complex type)
deployedLibraries           Not specified (complex type)
resources                   Not specified (complex type)

[DAS]domain1.adminServer>set description "Hello, world"
The value Hello, world is set at the specified location /description.
[DAS]domain1.adminServer>ls
description                Hello, world
productionMode              true
id                          982883233
securityManager             complex type
passwordValidator           Not specified (complex type)
adminServerName             adminServer
domainLogHome               Not specified (string      )
systemClusteringFramework  complex type
domainBackup                complex type
enableWebadmin              true
enableJsonCommand           false
enableToResynchronizeApplications false
servers                     complex type
sessionCluster              complex type
clusters                    Not specified (complex type)
```

```

serverTemplates                Not specified (complex type)
lifecycleInvocation            list of [lifecycleInvocation]
applicationRepositories        Not specified (complex type)
deployedApplications           Not specified (complex type)
deployedLibraries              Not specified (complex type)
resources                      Not specified (complex type)

```

```

[DAS]domain1.adminServer>activate
The activation completed successfully.

```

```

[Details]
domain.xml : ACTIVATED
domain : ACTIVATED
domain.description : ACTIVATED

```

4.2.17.8. cancel

変更した内容を削除し、動的設定変更のためのロックを解除します。コマンドを実行するユーザーがロックを持っていなくても使用できます。動的設定変更のためのロックを強制的に解除するために使用するコマンドです。

- 使用方法

```
cancel
```

- 例

```

[DAS]domain1.adminServer>edit
Successfully acquired a configuration lock from Domain Administration Server.
[DAS]domain1.adminServer>ls
description                Hello, world
productionMode              true
id                          982883233
securityManager             complex type
passwordValidator           Not specified (complex type)
adminServerName             adminServer
domainLogHome               Not specified (string      )
systemClusteringFramework  complex type
domainBackup                complex type
enableWebadmin              true
enableJsonCommand           false
enableToResynchronizeApplications false
servers                    complex type
sessionCluster              complex type
clusters                   Not specified (complex type)
serverTemplates             Not specified (complex type)
lifecycleInvocation         list of [lifecycleInvocation]

```

```
applicationRepositories      Not specified (complex type)
deployedApplications        Not specified (complex type)
deployedLibraries           Not specified (complex type)
resources                   Not specified (complex type)

[DAS]domain1.adminServer>cancel
Current configuration editing status is cleared, and the configuration lock is
released.
[DAS]domain1.adminServer>ls
Current user does not have the configuration lock.
```

4.2.18. JSONを使用したコマンドの実行

JEUS v7.0 Fix#1から、JSONを使用してコマンドを実行できる機能が追加されました。

HTTP POSTのリクエストを使用してコマンドの実行を要求するJSON文を送信すると、サーバーでリクエストを処理後、コマンドの実行結果を格納したJSON文をHTTPレスポンスとして送信する構造です。

参考

JSONについての基本的な内容はwww.json.orgを参照してください。

4.2.18.1. 一般コマンドの実行

以下はコマンドの実行例です。

```
serverinfo -server server1 -state
```

JSONを使用して上記のコマンドを実行する手順は以下のとおりです。

1. コマンドの実行を要求するJSON文を作成します。

上記のコマンドを実行するために作成が必要なJSON文は以下のとおりです。

```
{
  "jeusadmin": {
    "command": "server-info",
    "options": [
      "-server server1",
      "-state"
    ],
    "argument": null
  }
}
```


区分	説明
command	実行するコマンド名
options	コマンドに適用するオプションとオプションのパラメータ
argument	コマンドに適用するパラメータ

- 作成したJSON文をサーバーに送信してコマンドの実行を要求します。送信時に使用するプロトコルはHTTPで、メソッドはPOSTを使用します。

JEUSサーバーでJSONコマンドを実行するためのリクエストを受信するURLは以下のとおりです。

```
http://${SERVER_HOST}:${SERVER_BASE_PORT}/jsonCommand/command.json
```

参考

JEUSサーバーでコマンドを実行するにはユーザーの認証が必要です。ユーザーの認証はHTTP Basic Authenticationを使用します。

- サーバーが要求したJSON文を解釈して実行後、実行結果を格納したJSON文をHTTPレスポンスとして送信します。実行結果はコマンド別に異なります。

以下の結果は、上記例のコマンドを実行した場合、サーバーでレスポンスとして受信する例です。JEUSドメインの構成によって結果は異なることがあります。

```
{
  "jeusadmin-result": {
    "message": "",
    "data": [
      {
        "title": "Information of Domain (domain1)",
        "header": null,
        "column-names": [
          "Server", "Status", "Node Name", "PID", "Cluster",
          "Latest StartTime/ShutdownTime", "Need to Restart",
          "Listen Ports"
        ],
        "rows": [
          {
            "row-key": "0",
            "values": [
              "server1(*)", "RUNNING (284sec)", "N/A", "2151", "N/A",
              ... (省略)
            ]
          }
        ]
      }
    ]
  }
}
```

```

        "footer": null
    }
    [ ,
        "post-message": " "
    ]
}

```

レスポンス結果を格納しているJSON文は以下の要素で構成されています。

- message

上段に出力するメッセージです。messageが複数の場合はリスト形式で表現します。コマンドによって存在しない場合もあります。

- data

実行結果を格納しているオブジェクトで、コマンドによって存在しない場合もあります。dataは基本的に表形式のデータ構造を表現しており、構成要素は以下のとおりです。

区分	説明
title	表のタイトルです
header	表のヘッダーです
column-names	表を構成する各列の名前です
rows	表を構成する各行の内容です。 以下の2つの要素で構成されています。 <ul style="list-style-type: none"> – row-key: 該当行の名前です – values: 該当行に指定する値です
footer	表のフッターです

- post-message

下段に出力するメッセージです。post-messageが複数の場合はリスト形式で表現します。コマンドによって存在しない場合もあります。

4.2.18.2. アプリケーションのデプロイ

本節では、JSONを使用してアプリケーションをインストールあるいはデプロイする方法について説明します。アプリケーションをインストールあるいはデプロイする作業は一般コマンドとは異なり、実行するJSONコマンドとデプロイするアプリケーションを一緒に渡す必要があります。

以下は、JSONを使用してアプリケーションをインストールあるいはデプロイする手順についての説明です。

1. コマンドの実行を要求するJSON文を作成します。Hello.warというアプリケーションを「server1」にデプロイする場合、JSONコマンドは以下のように作成します。

```
{
  "jeusadmin": {
    "command": "deploy-application",
    "options": [
      "-servers server1",
      "-path Hello.war"
    ],
    "argument": null
  }
}
```

2. 作成したJSON文とデプロイするファイルをマルチパート形式でサーバーに送信し、コマンドの実行を要求します。マルチパートは、1で作成したJSONコマンドを格納したコマンド・パートと、デプロイするアプリケーションのアーカイブを指定したファイル・パートで構成されます。

以下は、JEUSサーバーでJSONコマンドによるアプリケーションのデプロイ要求を受信するURLです。

```
http://${SERVER_HOST}:${SERVER_BASE_PORT}/jsonCommand/install.json
```

重要

JEUSサーバーでコマンドを実行するにはユーザーの認証が必要です。ユーザーの認証はHTTP Basic Authenticationを使用します。

3. サーバーが要求したJSON文を解釈して実行後、実行結果を格納したJSON文をHTTPレスポンスとして送信します。この部分は一般コマンドを実行した場合と同じであるため、[「4.2.18.1. 一般コマンドの実行」](#)を参照してください。

参考

1. アプリケーションのインストールまたはデプロイに関連するコマンドは、[install-application](#)、[deploy-application](#)、[distribute-application](#)があります。
2. 一般コマンドをJSONで使用するについては[「4.2.18.1. 一般コマンドの実行」](#)を参照してください。

4.2.19. Scriptモードの使用方法およびスクリプトの作成方法

本節では、Scriptモードの使用方法およびスクリプトの作成方法について説明します。

4.2.19.1. Scriptモードの使用方法

Scriptモードでjeusadminを動作します。

- 例

以下は、JEUSのユーザー名は「administrator」、パスワードは「jeus」に設定して、test.pyスクリプトを実行するjeusadminのScriptモードの例です。

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus -script test.py
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
in the domain domain1.
JEUS8 Administration Tool
To view help, use the 'help' command.
```

以下のように、スクリプトに引数を渡すこともできます。

```
JEUS_HOME/bin$jeusadmin -u administrator -p jeus -script "test.py arg1 arg2"
```

4.2.19.2. スクリプトの作成方法

jeusadminは、Scriptモードで動作するとき、コマンドを実行するためのメソッドと結果データを提供します。

- コマンドの実行

以下のような方法でスクリプト言語ごとにコマンドを実行できます。

- Python

```
result = command("server-info")
```

- Ruby

```
result = $command.call "server-info"
```

- 結果データ

コマンドの実行結果としてJeusResultオブジェクトが返されます。結果が表形式を含んでいる場合、JeusResultからJeusTabularDataオブジェクトを取得できます。

各データ型は、以下のメソッドを提供します。

- JeusResult

メソッド	説明
isComplete()	コマンドの実行が正常に完了したかを返します
getMessage()	結果メッセージを返します

メソッド	説明
getData()	結果の表データのJeusTabularDataのリストを返します

– JeusTabularData

メソッド	説明
getTitle()	表のタイトルを返します
getHeader()	表のヘッダーを返します
getFooter()	表のフッターを返します
getColumnNames()	表の各列のタイトルのみをリスト形式で返します
getRows()	表データの各行のリストを返します

● 例

– SHUTDOWN状態のサーバーを見つけて出力し、再開始します。

```
result = command("server-info")
tables = result.getData();
table = tables[0];
rows = table.getRows();

shutdown_servers = []
for row in rows:
    if "SHUTDOWN" in row[1]:
        shutdown_servers.append(row[0])
        print("Server : %s, Node : %s" % (row[0], row[2]))

if shutdown_servers:
    command("start-server %s" % (" ".join(shutdown_servers)))
```

– スクリプトの引数としてアプリケーションのパスとデプロイするサーバー・リストを取得してから、アプリケーションをインストール、デプロイした後、サーバーを再開始します。

```
import time
import sys
import os

path = sys.argv[0]
servers = sys.argv[1]

direc, app = os.path.split(path)
apptype = os.path.splitext(app)[1][1:].upper()
```

```
command("undeploy -f %s " % app)
command("uninstall-application %s" % app)
command("install-application %s -id %s" % (path, app))
command("deploy %s -servers %s -type %s" % (app, servers, apptype))

# restart servers
command("stop-server %s" % servers)
time.sleep(5)
command("start-server %s" % servers)
```

参考

現在jeusadminでは、PythonとRubyの2つのスクリプト言語についてScriptモードを提供し、拡張子で区分しています。拡張子の.pyと.rbに対してスクリプトを認識して動作します。

4.3. appcompiler

appcompilerとは、EJBインターフェースのImplクラス、Skeletonクラス、Stubクラス、JSPをコンパイルして、サーブレット・クラス、WebサービスのEndpointクラスなどを作成するアプリケーション・コンパイラです。

appcompilerは、EJB 2.1インターフェースのImplクラス、RMI StubとSkeletonクラス、WebモジュールのJSPをコンパイルして、サーブレット・クラスを事前に作成するために使用できます。EJB 3.0インターフェースのみ構成されているEJBの場合や、JAX-WSクライアント、サーバー・アプリケーションは、このツールを実行する必要がありません。

このツールは、以下の場合に使用すると効果的です。

1. EJB 2.x標準に準ずるBeanが多く、最初のデプロイ時間が長く掛かる場合
2. サービス・タイムにJSPコンパイルする時間が長く掛かり、事前にコンパイルする必要がある場合

EJB 2.xモジュールの場合、デプロイ時にアプリケーションにfast-deploy設定を行います。または、jeusadminコンソール・ツールの**deploy**コマンドに[-fast]オプションを追加します。このオプションを通じて、モジュールのランタイムのデプロイをする場合、appcompilerが自動で実行されることを防ぐことができます。EJBエンジンは、fast-deployオプションが設定されている場合、インターフェースのImplクラス、Skeletonクラス、Stubクラスを作成し、コンパイルする過程を実行せず、事前に作成されたクラスを使用してデプロイする際に時間を減らすことができます。

Webエンジンは、JSPに対してコンパイルされたサーブレット・クラス・ファイルが存在する場合は、サービス・タイプにコンパイルしません。このツールには、ファイルを作成するたびにコンパイルするEachモードと、すべてのファイルを一括でコンパイルするBatchモードがあります。BatchモードはEachモードに比べてコンパイル速度は速いものの、エラーが発生した場合に原因となるファイルを見付けにくいというデメリットがあります。

基本動作はEachモードであるため、Batchモードを使用する場合は、[-batch]オプションを利用するか jeus.app.compiler.mode=batchを設定します。ただし、BatchモードはEJB 2.xモジュールにのみ適用されます。

注意

WindowsにてWARファイルに対してappcompilerを実行する場合、一時的に作成したディレクトリーが削除されないことがあります。これは、java.net.URLClassLoaderがWARファイル内部に含まれているWEB-INF/lib/*.jarファイルを明示的に閉じることができないためです。java.net.URLClassLoaderの場合、Java 7からcloseメソッドが追加されているため、今後のバージョンでは問題は生じません。しかし、現在は一時ディレクトリーを手動で削除する必要があります。

以下は、ツールの使用方法、パラメータおよび例についての説明です。

● 使用方法

```
appcompiler [-h]
             [-verbose]
             [-clp <class-path>]
             [-keep]
             [-jspmap]
             [-batch]
             [-q]
             [-client <clientview_filename>]
             [-noaddfile]
             [-deloldgen]
             [-ejbjar <ejb-jar.xml_path>]
             [-jeusejbdd <jeus-ebb-dd.xml_path>]
             [-D <property=value>]
             [-property <file-name>]
             [-target <application-path>]
             [-j concurrency-level]
             [-ejbonly]
             [-genjavaonly]
             [-webonly]
```

● パラメータ

パラメータ	説明
[-h]	appcompilerのヘルプを出力します
[-verbose]	verboseモードを設定して詳細手順を出力します(ログ・レベルをFINESTに設定します)

パラメータ	説明
[-clp]	コンパイルの際に必要なライブラリーをクラス・パスに登録します。ファイルとディレクトリーを両方使用でき、それぞれのライブラリーを区分するときは、Windowsはセミコロン(;)、Unixはコロン(:)を使用します
[-keep]	コンパイル中に作成されたソースを維持するか否かを設定します
[-jspmap]	servlet-mapping table(jeus_jspmap.xml)を作成するか否かを設定します。ただし、このオプションを使用する場合はJSPファイルがあるディレクトリー名にJava予約語(例. if、enum、classなど)があってはなりません
[-batch]	すべてのファイルを作成後、一括してコンパイルするBATCHモードで動作します。 このオプションを使用しない場合、1つのファイルを作成後にコンパイルするEachモードで動作します。このオプションはEJB 2.xモジュールにのみ適用されます
[-q]	複数のモジュールを持つEARアプリケーションをコンパイルする際、Webモジュールのコンパイルに失敗すると、残りのモジュールはコンパイルせずにエラーメッセージを出力してコンパイルを終了します。 このオプションを使用しない場合、Webモジュールのコンパイルに失敗しても、エラー・メッセージのみ出力して、残りのモジュールをコンパイルします
[-client <clientview_filename>]	作成されるStubクラスやインターフェース・クラスが含まれるclientviewファイルの名前を設定します。 <ul style="list-style-type: none"> - <clientview_filename>はEJBアーカイブ・ファイル形式のJARのみサポートし、コンパイル対象(archived or exploded application)が存在するディレクトリーに作成されます。このオプションは、スタンドアロンEJBモジュールとEJBモジュールを持つEARアプリケーションにのみ適用されます - <clientview_filename>は<application_file_or_directory_name>と同じ場合はサポートしません - <clientview_filename>と同一名のファイルが存在する場合、clientviewのためのインターフェース・クラスと作成されたStubクラスを、既に存在する同一名のファイルに追加します。clientviewファイルは、サーバーにインストールされているEJBを利用するクライアント・アプリケーションを開発するためにクライアントに提供します

パラメータ	説明
[-noaddfile]	<p>[-client]オプションと組み合わせて使用する場合に有効で、clientviewファイルを作成します。しかし、コンパイル対象になるファイルはコンパイルせず、元の内容を維持します。</p> <p>たとえば、calc.jarをappcompilerでコンパイルする際、clientviewファイルを作成するために[-client]オプションを使用します。ここでclientviewファイルとは、Stub、Remote、Homeインターフェース・クラスのみを集めたファイルを意味します。このとき、calc.jarをappcompilerでコンパイルしてからclientviewファイルを作成するので、calc.jarをコンパイルします。コンパイルされたファイルには、元のファイルをコンパイルして作成されたImpl、Skel、Stubなどのクラス・ファイルが追加されています。calc.jarファイルをコンパイルせず、clientviewファイルのみを作成したい場合は、[noaddfile]オプションを使用します</p>
[-deloldgen]	<p>EJBモジュールをコンパイルする場合、以前のバージョンのJEUSで作成されたImpl、Skel、Stubファイルを削除し、コンパイルを行います。</p> <p>【参考】</p> <p>これは、JEUS 4とJEUS 5では、Impl、Skeleton、Stubクラスの名前を作成する方法がJEUS 8とは異なるためです。以前のJEUS 4とJEUS 5バージョンで作成したEJBモジュールをappcompilerでコンパイルしてJEUS 8で使用する場合、既存バージョンのImpl、Skeleton、Stubファイルが残っているため、JEUS 8でEJBが正常に実行されなかった問題点を解決できます</p>
[-ejbjar <ejb-jar.xml_path>]	コンパイル時に使用するejb-jar.xmlファイルを設定します
[-jeusejbdd <jeus-ejb-dd.xml_path>]	コンパイル時に使用するjeus-ejb-dd.xmlファイルを設定します
[-D <property=value>]	システム・プロパティを指定します。このオプションは1回以上使用できます
[-property <file-name>]	ファイルでシステム・プロパティを設定します
[<application_file_or_directory_name>]	<p>コンパイル対象のアプリケーション・ファイル(archived ear、jar、war)またはディレクトリー(exploded EAR、JAR、WAR)を設定します。</p> <p>EARだけでなく、standalone JAR、WARも1つのアプリケーションとして見なします。コンパイルして作成されたImpl、Skel、Stubクラス・ファイルは、コンパイル対象アプリケーションのファイルに追加されます</p>

パラメータ	説明
[-j concurrency-level]	JSPをコンパイルする際に使用する値です。JSPファイルが多い場合、この設定を通じて複数のスレッドに同時にコンパイルを行うことが効率的です(デフォルト値: 1)
[-ejbonly]	EAR内部にあるEJBモジュールのみコンパイルします
[-genjavaonly]	Javaファイルのみを作成したい場合に使用します
[-webonly]	EAR内部にあるWebモジュールのみコンパイルします

- 例

appcompilerは、JEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

- ejb.jarモジュールの、Home、Remote interface Impl、Skeleton、Stubクラスを作成してJARファイルに追加します。

```
JEUS_HOME/bin$ appcompiler ejb.jar
```

- web.warモジュールのすべてのJSPファイルからサーブレット・クラスを作成してWARファイルに追加します。

```
JEUS_HOME/bin$ appcompiler web.war
```

- Batchモードでモジュールをコンパイルします。

```
JEUS_HOME/bin$ appcompiler -batch app.jar
```

- ejb.jarモジュールの、home、remoteインターフェースのImpl、Skeleton、Stubクラスを作成してejb.jarファイルに追加します。そして、Home、Remoteインターフェースと、作成されたStubクラスを含むclinetview.jarファイルを作成します。

```
JEUS_HOME/bin$ appcompiler -client clientview.jar ejb.jar
```

4.4. ejbddinit

ejbddinitとは、ejb-jar.xml情報と既に作成されているプロパティ・ファイルおよびユーザーの入力値などを利用して、JEUS EJB DD(jeus-ejb-dd.xml)を自動作成するツールです。

EJBのデプロイ時に必要な項目はデフォルト値として設定するため、JEUS EJB DDは必ずしも存在する必要はありません。したがって、EJBを新しく開発する場合に不要な場合もありますが、JEUS EJB DDを使用していた既存の方法でアプリケーションを作成する場合や、他の製品からマイグレーションする場合に、簡単にテンプレートを作成する用途で使用できます。

参考

Java EE 6のannotation-based方式は、ソースファイル内に設定できるためサポートしません。

以下は、ツールの使用方法、パラメータおよび例についての説明です。

● 使用方法

```
ejbddinit [-property property_file_path]
          [-level log_level]
          [-source application_file_or_directory_name]
```

● パラメータ

パラメータ	説明
<code>[-property property_file_path]</code>	<p>プロパティ・ファイルのパスを指定します。jeus-ejb-dd.xmlを目的に合わせて作成するために、このオプションを必ず設定する必要があります。プロパティをJEUS EJB DDのタグに対応させて値を設定できます。設定可能なプロパティ一覧は4.4節「プロパティ一覧」を参照してください</p> <p>– 例1)</p> <pre>export-port=40001</pre> <p>すべてのEJBのexport-portを40001と設定して、jeus-ejb-dd.xmlファイルを作成します</p> <p>– 例2)</p> <pre>SampleBean.export-port=40004</pre> <p>ejb.jarでEJB名がSampleBeanと設定されているEJBのexport-portを40004と設定して、jeus-ejb-dd.xmlファイルを作成します。このように、各プロパティの名前の前に終止符(.)を区切り子とし、特定のEJBの名前を明示できます。同じ設定について特定のEJBの名前を明示したプロパティとそうでないプロパティが存在する場合、前者が優先的に適用されます</p>
<code>[-level log_level]</code>	ejbddinitの実行時に画面に出力されるログ・レベルを設定します
<code>[-source application_file_or_directory_name]</code>	<p>対象EJBモジュールのパスを設定します。EJBモジュールは、「.jar」拡張子を持つ圧縮ファイルやディレクトリです。</p> <p>JEUS EJB DDを作成すると、EJBモジュールがディレクトリの場合、META-INFフォルダの下位にjeus-ejb-dd.xmlを作成します。EJB</p>

パラメータ	説明
	モジュールが圧縮ファイルの場合、本来のファイル名の最後に「.new」を付けた名前で新しい圧縮ファイルを作成し、圧縮ファイル内部のMETA-INFディレクトリーの下位にjeus-ejb-dd.xmlを作成します

- 例

- usageを出力します。

```
$ ejbddinit
```

- ejbddinit.propertiesに定義されているプロパティに基づいて、EJBモジュールのejb.jarのejb.jar.newファイルを作成します。作成されたejb.jar.newファイル内部のMETA-INFディレクトリーの下位にjeus-ejb-dd.xmlが位置します。

```
$ ejbddinit -property ejbddinit.properties -source ejb.jar
```

- ejbddinit.propertiesに定義されているプロパティに基づいて、EJBモジュールのejb_dirのMETA-INFディレクトリーの下位にjeus-ejb-dd.xmlを作成します。

```
$ ejbddinit -property ejbddinit.properties -source ejb_dir
```

- ejbddinit.propertiesに定義されているプロパティに基づいて、jeus-ejb-dd.xmlを作成します。sourceオプション値を入力していないため、ejbddinit.propertiesにEJBモジュール・パスが設定されている必要があります。

```
$ ejbddinit -property ejbddinit.properties
```

- ejbddinit.propertiesに定義されているプロパティに基づいて、EJBモジュールのejb.jarのejb.jar.newファイルを作成します。作成されたejb.jar.newファイル内部のMETA-INFディレクトリーの下位にjeus-ejb-dd.xmlが位置します。FINEレベルのログが画面に出力されます。

```
$ ejbddinit -property ejbddinit.properties -level FINE -source ejb.jar
```

参考

ejbddinitはAntタスクをサポートします。ejbddinit antタスクについては「[5.4.2. ejbddinit](#)」を参照してください。

プロパティ一覧

以下は、ejbddinitでサポートするプロパティの一覧です。各プロパティの詳細についてはjeus-ejb-dd.xmlタグのdescriptionを参照してください。

[表 4.1] ejbddinitのプロパティ

プロパティ	タイプ	備考
source	String	対象EJBモジュールのパスです。jeus-ejb-dd.xmlのタグとは関連がありません
logging-level	String	ejbddinitの実行時に画面に出力されるログのレベルです。jeus-ejb-dd.xmlのタグとは関連がありません
export-name	String	jeus-ejb-dd.xmlの<export-name>タグに対応します。特定のEJBに対してのみ使用を推奨します。使用する場合はパターンをサポートします。詳細については export-name プロパティのパターンのサポート を参照してください
local-export-name	String	jeus-ejb-dd.xmlの<local-export-name>タグに対応します。特定のEJBに対してのみ使用を推奨します
export-port	int (not negative)	jeus-ejb-dd.xmlの<export-port>タグに対応します
export-iiop	boolean	jeus-ejb-dd.xmlの<export-iiop>タグに対応します
thread-max	int (not negative)	jeus-ejb-dd.xmlの<thread-max>タグに対応します
bean-pool-min	int (not negative)	jeus-ejb-dd.xmlの<bean-pool>/<pool-min>タグに対応します
bean-pool-max	int (not negative)	jeus-ejb-dd.xmlの<bean-pool>/<pool-max>タグに対応します
connect-pool-min	int (not negative)	jeus-ejb-dd.xmlの<connect-pool>/<pool-min>タグに対応します
connect-pool-max	int (not negative)	jeus-ejb-dd.xmlの<connect-pool>/<pool-max>タグに対応します
capacity	int (not negative)	jeus-ejb-dd.xmlの<capacity>タグに対応します
passivation-timeout	long (not negative or -1 for disable)	jeus-ejb-dd.xmlの<passivation-timeout>タグに対応します
disconnect-timeout	long (not negative or -1 for disable)	jeus-ejb-dd.xmlの<disconnect-timeout>タグに対応します
engine-type	String (defined)	jeus-ejb-dd.xmlの<engine-type>タグに対応します。 以下のうち1つを設定します – EXCLUSIVE_ACCESS – SINGLE_OBJECT

プロパティ	タイプ	備考
		– MULTIPLE_OBJECT
subengine-type	String (defined)	jeus-ejb-dd.xmlの<subengine-type>タグに対応します。 以下のうち1つを設定します – ReadLocking – WriteLocking – WriteLockingFind
fetch-size	int (not negative)	jeus-ejb-dd.xmlの<fetch-size>タグに対応します
init-caching	boolean	jeus-ejb-dd.xmlの<init-caching>タグに対応します
table-name	String	jeus-ejb-dd.xmlの<table-name>タグに対応します
creating-table	String (defined)	jeus-ejb-dd.xmlの<creating-table>タグに対応します。 以下のうち1つを設定します – none – use-existing – force-create
deleting-table	boolean	jeus-ejb-dd.xmlの<deleting-table>タグに対応します
db-vendor	String	jeus-ejb-dd.xmlの<db-vendor>タグに対応します
datasource-name	String	jeus-ejb-dd.xmlの<data-source-name>タグに対応します
enable-instant-ql	boolean	jeus-ejb-dd.xmlの<enable-instant-ql>タグに対応します

export-nameプロパティのパターンのサポート

export-nameは、EJBモジュールの名前や、ejb-jar.xmlに定義されているEJBコンポーネントの<ejb-name>、<ejb-class>タグの値を利用して指定する場合があります。そのため、これらの値を手軽に利用できるようにパターンをサポートします。

[表 4.2] export-nameをサポートするパターン

パターン名	パターン値
%{module-name}	EJBモジュールの名前です。JARファイルの場合、「.jar」拡張子を削除したファイルの名前です

パターン名	パターン値
%{ejb-name}	EJBコンポーネントの名前です。ejb-jar.xmlに定義されているEJBコンポーネントの<ejb-name>タグの値です
%{ejb-fqn}	EJBコンポーネントのEJBクラスの完全修飾名です。ejb-jar.xmlに定義されているEJBコンポーネントの<ejb-class>タグの値です
%{ejb-class}	%{ejb-fqn}値でパッケージ名を除いたクラス名です

以下は、名前が「sample-ejb」で、EJBクラスの完全修飾名が「sample.SampleBean」のEJBコンポーネントが、名前が「myejb」のEJBモジュールにパッケージングされていると仮定した場合のパターンの使用例です。

- export-nameを「myejb_sample-ejb」と指定します。

```
export-name=%{module-name}_{ejb-name}
```

- export-nameを「PREFIX_sample.SampleBean」と指定します。

```
export-name=PREFIX_%{ejb-fqn}
```

- export-nameを「SampleBean_POSTFIX」と指定します。

```
export-name=%{ejb-class}_POSTFIX
```

export-nameの設定時に必要な(PREFIXやPOSTFIXのような)他文字列をパターンに付けることができ、パターン間の組み合わせも可能です。パターン名は大文字と小文字の区別をせずに使用できます。

プロパティ・ファイルの例

以下は、ejbddinitプロパティ・ファイルの作成例です。JEUS EJB DDを作成するEJBに対して適切に動作するように、以下の例を修正して使用することを推奨します。

```
# PROPERTY FILE SAMPLE
# JEUS EJB DD Generation Option

# target file or directory path
source=/home/workspace/ejb.jar

# log-level for EJB DD init
logging-level=DEBUG

# JEUS EJB DD configuration tag and value pairs for all EJBs
db-vendor=mysql
datasource-name=jdbc/__default
creating-table=use-existing
deleting-table=false
```

```

engine-type=EXCLUSIVE_ACCESS
subengine-type=ReadLocking
fetch-size=1111
enable-instant-ql=true
export-port=9999
export-iiop=false
thread-max=100
bean-pool-min=10
bean-pool-max=100
capacity=10000
passivation-timeout=300000
disconnect-timeout=-1
connect-pool-min=10
connect-pool-max=100
init-caching=false

# JEUS EJB DD configuration tag and value pairs for BookBean EJB
BookBean.export-name=BookBeanFromProperty
BookBean.local-export-name=LocalBookBeanFromProperty
BookBean.export-port=55555
BookBean.thread-max=100

```

4.5. webddgen

webddgenは、ユーザー入力値をもとに、Webアプリケーション内のWEB-INFディレクトリーにjeus-web-dd.xmlを作成またはアップデートするツールです。Webモジュール内部のWEB-INFディレクトリーにjeus-ejb-dd.xmlを作成します。圧縮ファイルの場合は、その圧縮ファイル内にjeus-web-dd.xmlを含めます。

- 使用方法

```

webddgen -file <war-file-path>
          [-ctx <context-path>]
          [-prop <context-level-property>]
          [-verbose]

```

- パラメータ

パラメータ	説明
-file <war-file-path>	Webモジュールのパスを設定します。 .war拡張子を持つJAR形式の圧縮ファイルまたはディレクトリーです
[-ctx <context-path>]	jeus-web-dd.xmlに設定するコンテキスト・パス情報です
[-prop <context-level-property>]	Webモジュール(Web Context)単位で設定するプロパティです。

パラメータ	説明
	たとえば、JEUS 4で作成したJSPを修正せずに使用するには、そのJSPが含まれているWebモジュールのjeus.servlet.jsp.modernプロパティをfalseに設定します。プロパティ形式はname=valueです (例: jeus.servlet.jsp.modern=false)
[-verbose]	動作に問題があるとき、デバッグ用に使えます。そのため、webddgen内部で作成するディレクトリーを削除しません。ただし、webddgenを再実行する場合には削除します

- 例

- 使用方法を出力します。

```
$ webddgen -help
```

- web.warにjeus-web-dd.xmlを作成します。

```
$ webddgen -file web.war
```

- jeus-web-dd.xmlを作成時にコンテキスト・パスを追加します。

```
$ webddgen -file web.war -ctx /web1
```

- JEUS 4で作成したJSPが含まれているWebモジュールにjeus.servlet.jsp.modernプロパティをfalseに設定します。

```
$ webddgen -file oldjsps.war -prop jeus.servlet.jsp.modern=false
```

4.6. encryption

encryptionは暗号化ツール(Encryption Tool)で、文字列を暗号化あるいは復号化する機能を提供します。文字列はBase64フォーマットでエンコードするか、Hash結果を提供します。

秘密鍵が必要なアルゴリズム(例: AES、DES、DESEDE、BLOWFISH、SEED)の場合、security.keyファイルを作成して使用します。security.keyファイルは基本的にJEUS_HOME/domains/<domain name>/config/securityにあります。変更したい場合は、システム・プロパティのjeus.security.keypathを利用してパスを指定します。

参考

JEUS 6 Fix#3からは、マスター・パスワードを利用してsecurity.keyファイルを保護することが可能です。

JEUS 6 Fix#4からは暗号化方式が少し変更されました。それにより、以前のフォーマットのciphertextを、変更したフォーマットのciphertextlに変更するconvertオプションが、Fix#5にて暗号化ツールに追加されました。

encryptionは、以下のディレクトリーに位置するencryptionスクリプトで実行できます。

```
JEUS_HOME/bin/
```

例

以下は、JEUS_HOMEシステム環境変数が設定されており、システム・パスにJEUS_HOME/binディレクトリーが設定されていると仮定した場合のencryptionの呼び出しについての例です。

- 呼び出し後、ツールは単純にエンコードまたはデコードされた文字列を出力します。

ツールにいかなるパラメータも提供していない場合や、呼び出し文法が間違っている場合、簡単なヘルプ・メッセージが出力されます。

```
$ encryption base64 mypassword
[BASE64] : [mypassword] --> [encode:bXlwYXNzd29yZA==]
[BASE64] : [bXlwYXNzd29yZA==] --> [decode:mypassword]
```

以下は、文字列「bXlwYXNzd29yZA==」は「mypassword」をBase 64フォーマットでエンコードしたものです。

```
$ encryption DESede mypassword
[DESEDE] : [mypassword] --> [encode:8JLoskMPHkwwLKi+TJeOgQZBDO15PBQ=]
[DESEDE] : [8JLoskMPHkwwLKi+TJeOgQZBDO15PBQ=] --> [decode:mypassword]
```

文字列「8JLoskMPHkwwLKi+TJeOgQZBDO15PBQ=」は「mypassword」をDESedeの暗号化アルゴリズム(cipher algorithm)で暗号化したパスワードです。この暗号化を実行するために作成した秘密鍵は、JEUS_HOME/domains/<domain name>/config/securityにあるsecurity.keyファイルに保存されます。(最初の暗号化ではない場合、security.keyファイルに保存されている鍵を利用します)

- -protectkeyオプションを利用すると、マスター・パスワードの入力によりsecurity.keyファイルを暗号化できます。

```
$ encryption -protectkey AES mypassword
Input the master password For key encryption>
Confirm the master password For key encryption>
[AES] : [mypassword] --> [encode:qshQdjgh8aAr3fWPYWbU0/VkFbs9y1lZeRJaK5xSiuo=]
[AES] : [qshQdjgh8aAr3fWPYWbU0/VkFbs9y1lZeRJaK5xSiuo=] --> [decode:mypassword]
```

上記例と似ていますが、マスター・パスワードを入力する点が異なります。

ここで入力されるパスワードを利用して、秘密鍵をDESedeアルゴリズムで暗号化します。上記コマンドの実行により作成されたsecurity.keyファイルを使用するには、常にマスター・パスワードの入力が必要があります。したがって、暗号化されたsecurity.keyファイルを利用して暗号化する場合、以下のように実行されます。

```
$ encryption -protectkey AES mypassword
The encryption key file is encrypted. Enter the master password.
Password>
[AES] : [mypassword] --> [encode:qsHQdjgh8aAr3fWPYWbU0/VkFbs9yllZeRJaK5xSiuo=]
[AES] : [qsHQdjgh8aAr3fWPYWbU0/VkFbs9yllZeRJaK5xSiuo=] --> [decode:mypassword]
```

- JEUS 6 Fix#4から暗号化方式が強化され、ciphertextのフォーマットが少し変更されました。そのため、Fix#4より前のmバージョンで作成した暗号文を、それ以降のバージョンではそのまま使用できなくなりました。それにより、Fix#5からencryptionでciphertextを変換できる機能が追加されました。これは、以下のように-convertプションを使用します。

```
$ encryption -convert AES i06wYRz3u60/Gqun2sKtXH1u=
Decryption was successful.
[before:i06wYRz3u60/Gqun2sKtXH1u=] --> [after:ET7c/P21Qx1Io8UI6Ss2NvZ0G=]
```

- security.keyファイルが暗号化されている場合、以下のように実行します。

```
$ encryption -convert -protectkey AES i06wYRz3u60/Gqun2sKtXH1u=
The encryption key file is encrypted. Enter the master password.
Password>

Decryption succeed.
[before:i06wYRz3u60/Gqun2sKtXH1u=] --> [after:ET7c/P21Qx1Io8UI6Ss2NvZ0G=]
```

4.7. wsgen

wsgenツールは、JAX-WSのWebサービスのためのツールです。JAX-WS Webサービスのためのツールで、JavaクラスよりWSDLファイルとJAX-RPCマッピング・ファイルを作成し、Webサービスのポリシー設定ファイルも作成できます。

wsgenツールは、サービス・エンドポイント・インターフェースのクラス（そして、任意のJavaで実装したクラス）から、以下のものを作成します。

- ポータブル・アーティファクト
- WebサービスのWSDLファイル（オプション）
- Webサービスのポリシー・ファイルのwsit-<endpoint classname>.xmlファイル（オプション）

以下は、ツールの使用方法、パラメータおよび例についての説明です。

● 使用方法

```
Usage: wsgen [options] <SEI>

where [options] include:

  -classpath <path>          specify where to find input class files
  -cp <path>                 same as -classpath <path>
  -d <directory>            specify where to place generated output files
  -extension                  allow vendor extensions - functionality not specified
                             by the specification. Use of extensions may
                             result in applications that are not portable or
                             may not interoperate with other implementations
  -help                      display help
  -keep                      keep generated files
  -r <directory>            resource destination directory, specify where to
                             place resouce files such as WSDLs
  -s <directory>            specify where to place generated source files
  -verbose                   output messages about what the compiler is doing
  -version                   print version information
  -wsdl[:protocol]          generate a WSDL file. The protocol is optional.
                             Valid protocols are [soap1.1, Xsoap1.2], the default
                             is soap1.1.
                             The non stanadard protocols [Xsoap1.2] can only be
                             used
                             in conjunction with the -extension option.
  -inlineSchemas            inline schemas in the generated wsdl.
                             Must be used in conjunction with the -wsdl option.
  -servicename <name>       specify the Service name to use in the generated WSDL
                             Used in conjunction with the -wsdl option.
  -portname <name>          specify the Port name to use in the generated WSDL
                             Used in conjunction with the -wsdl option.

Examples:
  wsgen -cp . example.Stock
  wsgen -cp . example.Stock -wsdl -servicename {http://mynamespace}MyService
```

● パラメータ

パラメータ	説明
-classpath <path>	Javaクラス・ファイルを検索するためのパスを指定します
-cp <path>	-classpath <path>と同じです

パラメータ	説明
-d <directory>	結果ファイルが作成されるディレクトリを指定します
-extension	ベンダー特定の拡張性を明示する際に使用します。相互運用および移植性を低下させることがあります
-help	ヘルプを出力します
-keep	作成されるファイルを保存します
-r <directory>	genwsdlプロパティと一緒に使用します。WSDLファイルを作成する位置を設定します
-s <directory>	作成されるソース・ファイルの位置を指定します
-verbose	Verboseメッセージを出力します
-version	バージョン情報を画面に出力します
-wsdl[:protocol]	基本的にwsngenはWSDLファイルを作成しません。このオプションを使用すると、開発者はデプロイされる前にWSDLファイルを確認することができます。 [:protocol]もオプションで、デフォルト値はsoap1.1です。Xsoap1.2も使用でき、-extensionオプションと一緒に使用します
-inlineSchemas	作成されたWSDLのスキーマをインラインします。 -wsdlオプションと一緒に使用します
-servicename <name>	-wsdlオプションと一緒に使用します。作成されるWSDLファイルの特定wsdl:service要素の名前を設定します
-portname <name>	-wsdlオプションと一緒に使用します。作成されるWSDLファイルの特定wsdl:portname要素の名前を設定します

- 例

wsngenは、JEUS_HOME/bin/ディレクトリに位置する一般的なスクリプト・ファイルです。

以下は、wsngenツールの実行例です。

```
~$ wsngen -classpath build/classes -d build/classes -wsdl
fromjava.server.AddNumbersImpl
```

上記コマンドは、サービス・エンドポイントの実装クラスであるfromjava.server.AddNumbersImplから、JAX-WS Webサービスを構成します。fromjava.server.AddNumbersImplのクラス・パスは「build/classes」で、WSDL文書と一緒にJavaBeanファイルがbuild/classesディレクトリに作成されます。

4.8. wsimport

wsimportとは、JAX-WS Webサービスのためのツールです。WSDLファイルからクライアント側のJava Stubソース・ファイルとサーバー側のWebサービス・インターフェースのJavaソース・ファイル、Webサービス・ポリシーの設定ファイルを作成します。

wsimportツールは、WebサービスのWSDLから以下のものを作成します。

- SEI、サービス・クラスとその他のJAXB関連のJavaクラス
- Webサービス・ポリシーの設定が格納されているWSDLファイル、wsit-client.xmlファイル

以下は、ツールの使用方法、パラメータおよび例についての説明です。

- 使用方法

```
Usage: wsimport [options] <WSDL_URI>

where [options] include:
  -b <path>                specify jaxws/jaxb binding files or additional schemas
                             (Each <path> must have its own -b)
  -B <jaxbOption>          Pass this option to JAXB schema compiler
  -catalog <file>          specify catalog file to resolve external entity
                             references supports TR9401, XCatalog,
                             and OASIS XML Catalog format
  -d <directory>           specify where to place generated output files
  -extension                allow vendor extensions - functionality not specified
                             by the specification. Use of extensions may
                             result in applications that are not portable or
                             may not interoperate with other implementations
  -help                    display help
  -httpproxy:<host>:<port> specify a HTTP proxy server (port defaults to 8080)

  -keep                    keep generated files
  -p <pkg>                 specifies the target package
  -quiet                   suppress wsimport output
  -s <directory>          specify where to place generated source files
  -target <version>        generate code as per the given JAXWS spec version
                             Defaults to 2.2, Accepted values are 2.0, 2.1 and
                             2.2
                             e.g. 2.0 will generate compliant code for JAXWS 2.0
                             spec
  -verbose                 output messages about what the compiler is doing
```

```

-version                                print version information
-wsdllocation <location> @WebServiceClient.wsdlLocation value
-clientjar <jarfile>                   Creates the jar file of the generated artifacts along
with the                               WSDL metadata required for invoking the web service.

-generateJWS                           generate stubbed JWS implementation file
-implDestDir <directory>               specify where to generate JWS implementation file
-implServiceName <name>                local portion of service name for generated JWS
implementation
-implPortName <name>                  local portion of port name for generated JWS
implementation

Extensions:
-XadditionalHeaders                     map headers not bound to request or response
message to                             Java method parameters
-Xauthfile                             file to carry authorization information in
the format                             http://username:password@example.org/stock?wsdl

-Xdebug                                print debug information
-Xno-addressing-databinding            enable binding of W3C EndpointReferenceType
to Java
-Xnocompile                            do not compile generated Java files
-XdisableAuthenticator                 disable Authenticator used by JAX-WS RI,
-Xauthfile option will be ignored if set
-XdisableSSLHostnameVerification       disable the SSL Hostname verification while
fetching                               wsdl

Examples:
wsimport stock.wsdl -b stock.xml -b stock.xjb
wsimport -d generated http://example.org/stock?wsdl

```

● パラメータ

パラメータ	説明
-b <path>	外部のJAX-WSまたはJAXBバインディング・ファイルを指定します
-B <jaxbOption>	このオプションをJAXBスキーマ・コンパイラーに渡すことを指定します
-catalog <file>	TR9401、XCatalog、OASIS XML Catalog形式のような外部のエンティティー参照値を設定できます。また、ant xmllcatalogタイプを使用することもできます
-d <directory>	結果ファイルが作成されるディレクトリーを指定します

パラメータ	説明
-extension	ベンダー特定の拡張性を明示する際に使用します。相互運用および移植性を低下させることがあります
-help	ヘルプを出力します
-httpproxy:<host>:<port>	HTTPプロキシ・サーバーを指定します(デフォルト値: 8080)
-keep	作成されるファイルを保存します
-p <pkg>	WSDLのすべての名前空間URIのJavaパッケージ名を指定します
quiet	出力メッセージを表示しません
-s <directory>	作成されたソース・ファイルが格納される位置を指定します。このプロパティを指定すると、keepプロパティが自動設定されます
-target <version>	ここに指定されたJAXWSスペックのバージョンに従ってコードを作成します
-verbose	「true」を指定した場合、Verboseが出力されます(デフォルト値: false)
-version	バージョン情報を画面に出力します
-wsdllocation <location>	WSDL URIをこのプロパティに明示すると、そのURI値が作成されるSEI (Service Endpoint Interface)とサービス・インターフェースの @WebService.wsdlLocation Annotationと @WebServiceImpl.wsdlLocation Annotationに設定されます
-clientjar <jarfile>	Webサービスの呼び出しに必要なWSDLメタデータと一緒に生成されたアーティファクトのjarファイルを作成します
-generateJWS	スタブされたJWS実装ファイルを作成します
-implDestDir <directory>	JWS実装ファイルを作成する位置を指定します
-implServiceName <name>	生成されたJWS実装のサービス名のローカル部分です
-implPortName <name>	生成されたJWS実装のポート名のローカル部分です

- 例

wsimportは、JEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

以下は、wsimportツールの実行例です。

```
~$ wsimport -d build/classes -keep -p fromwsdl.server src/conf/AddNumbers.wsdl
```

上記のコマンドは、AddNumbers.wsdl WSDL文書よりパッケージ名がfromwsdl.serverであるポータブル・アーティファクトを作成します。[-keep]オプションによって作成されるJavaクラスのソースは削除されません。

4.9. java2wsdl

java2wsdlは、JAX-RPC Webサービスのためのツールで、JavaクラスからWSDLファイルとJAX-RPCマッピング・ファイルを作成します。

また、サービス・エンドポイント・インターフェースのJavaクラス(または実装クラス)からWebサービスのWSDLファイル、JAX-RPCマッピング・ファイルのようなリソースを作成します。

以下は、ツールの使用方法、パラメータおよび例についての説明です。

● 使用方法

```
Usage: java2wsdl <options> <input file>

where <input file> include:
    configuration file           service configuration xml file (by default)

where <options> include:
    -classpath <path>           specify where to find input class files
    -cp <path>                  same as -classpath <path>
    -d <directory>             specify where to place generated output files
    -level <log-level>         specify a log level
    -verbose                    [optional] turn verbose mode on
```

● パラメータ

パラメータ	説明
-classpath <path>	Javaクラス・ファイルを検索するためのパスを指定します
-cp <path>	-classpath <path>と同じです
-d <directory>	結果ファイルが作成されるディレクトリーを指定します
-level	ログ・レベルを指定します
-verbose	Verboseメッセージを出力します

● 例

java2wsdlは、JEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

以下は、java2wsdlツールの実行例です。

```
~$ java2wsdl -classpath build/classes -d build/web/WEB-INF service-config.xml
```

上記のコマンドは、service-config.xml設定ファイルとbuild/classesディレクトリーのクラスを使用して、build/web/WEB-INFディレクトリーにWSDL文書およびJAX-RPCマッピング・ファイルを作成します。

4.10. wsdl2java

wsdl2javaは、JAX-RPC Webサービスのためのツールで、WSDLファイルから以下のファイルを作成します。

- クライアントのWebサービスのスタブのJavaソース・コード
- サーバーのWebサービス・インターフェースのJavaソース・コード

以下は、ツールの使用方法、パラメータおよび例についての説明です。

- 使用方法

```
Usage: wsdl2java mode [options] wsdlURI

where mode include:
-gen:client          generate all client artifacts
-gen:server          generate all server artifacts
-gen                same as -gen:client
-import:client       generate client JSR-109 portable artifacts only
-import:server       generate server JSR-109 portable artifacts only
-import             same as -import:client

where [options] include:
* destination directory
-d <directory>      specify where to put output files
-cd <directory>     specify where to put compiled class files
                    If not specified, the compile class files will be
                    put in where '-d' specifies

* WSDL and Java mapping
-inputmapping <filename> specify the input JSR-109 JAX-RPC mapping file
                        (used for generating Java artifacts)
-package <package_name> specify the java package name to which
                        all namespaceURI in the WSDL map
-ns2pkg NS=PKG       specify the namespaceURI and java package name mapping

                        (NS : namespaceURI, PKG : java package name)
                        This option can be used serveral times

* output file
-outputmapping <filename> specify the output JSR-109 JAX-RPC mapping file
                        for the input WSDL
                        This option can not be used with '-inputmapping'
-compile            compile generated Java source files
                    ('tools.jar' must be in the classpath)
```

```

-nokeepsrc                delete generated java source files

* artifact generation options
-nowrapped                disable wrapped mode detection for the WSDL
-datahandleronly          force javax.activation.DataHandler for MIME types
-nodatabinding            force javax.xml.soap.SOAPElement for all WSDL parts
-soapver <VER>            specify SOAP version used in stub/tie class.
                           VER : '11' for SOAP 1.1, '12' for SOAP 1.2
-resolvedir directory     specify directory where to be put local copies
                           of include/import schemas and import WSDL

* other options
-username                 username to access the WSDL-URI
-password                 password to access the WSDL-URI

* verbose options
-level <LEVEL>            specify log level.
                           LEVEL : SEVERE, WARNING, INFO, FINE, FINER, FINEST
-verbose                  same as -level FINE

```

● パラメータ

パラメータ	説明
-gen:client	クライアント側のJavaクラスを作成します
-gen:server	サーバー側のJavaクラスを作成します
-gen	-gen:clientと同じです
-import:client	クライアント側のポータブル・アーティファクトを作成します
-import:server	サーバー側のポータブル・アーティファクトを作成します
-import	-import:clientと同じです
-d <directory>	結果ファイルが作成されるディレクトリーを指定します
-cd <directory>	コンパイルされたクラス・ファイルが作成されるディレクトリーを指定します
-inputmapping <filename>	Javaクラスを作成するために使用されるJAX-RPCマッピング・ファイルを指定します
-package <package_name>	WSDLのすべての名前空間URIのJavaパッケージ名を指定します
-ns2pkg NS=PKG	名前空間URIのJavaパッケージ名を指定します。 – NS: 名前空間URI – PKG: Javaパッケージ名

パラメータ	説明
-outputmapping <filename>	入力WSDLの出力JAX-RPCマッピング・ファイルを指定します
-compile	作成されたJavaファイルをコンパイルします
-nokeepsrc	作成されたJavaソース・ファイルを維持せずに削除します
-nowraped	WSDLのwrappedモード検索を禁止します
-datahandleronly	MIMEタイプに対してjavax.activation.DataHandlerを適用します
-nodatabinding	すべてのWSDLメッセージ・パートに対してjavax.xml.soap.SOAPElementを適用します
-soapver <VER>	stub/tieクラスで使用するSOAPバージョンを指定します。 – 11: SOAP 1.1 – 12: SOAP 1.2
-resolvedir <directory>	リモートのWSDLファイル(スキーマ・ファイルを含む)をアクセス可能なレポジトリに保存したい場合に使用します
-username	WSDL-URIIIにアクセスするためのユーザー名です
-password	WSDL-URIIIにアクセスするためのパスワードです
-level <LEVEL>	ログ・レベルを指定します
-verbose	ログを画面に出力します

● 例

wsdl2javaはJEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

以下は、wsdl2javaツールの実行例です。

```
~$ wsdl2java -gen:client -d build/classes
-outputmapping build/classes/web/WEB-INF/jaxrpcmapping.xml
-compile http://localhost:8088/AddNumbers/addnumbers?wsdl
```

上記コマンドは、リモートのhttp://localhost:8088/AddNumbers/addnumbers?wsdlのWSDL文書を使用してWebサービス・クライアントのためのスタブおよびサービス・インターフェースを作成します。

ここで、スタブおよびサービス・インターフェースはbuild/classesディレクトリーに作成され、JAX-RPCマッピング・ファイルはbuild/classes/web/WEB-INFに作成されます。[-compile]オプションで作成されるスタブとサービス・インターフェースはコンパイルされます。

4.11. wsdl2uddi

WSDLは、抽象的なインターフェースと任意のネットワーク・サービスのプロトコル・バインディングを提供することでUDDI標準を補完できます。JEUS UDDI WSDL PublishingはWSDL2UDDIを提供します。

wsdl2uddiツールはWebサービスのWSDLファイルからUDDIにパブリッシュできます。

使用方法

以下は、ツールの使用方法、パラメータおよび例についての説明です。

● 使用方法

```
Usage: wsdl2uddi UDDIVersion WSDLURI [wsdl-options]
        -uddiInquiry <UDDIInquiryURI> -uddiPublish <UDDIPublish>
        -uddiUsername <UDDIUsername> -uddiPassword <UDDIPassword>

[options]

* wsdl-options
    -wsdlUsername          username to access the WSDL-URI
    -wsdlPassword          password to access the WSDL-URI

* options
    -level <LEVEL>         specify log level.
                           LEVEL : SEVERE, WARNING, INFO, FINE, FINER, FINEST
    -verbose               same as -level FINE
```

● パラメータ

パラメータ	説明
UDDIVersion	UDDIのバージョンを設定します。入力値はv2またはv3です
WSDLURI	WSDLが存在する位置のURI値です
-uddiInquiry <UDDIInquiryURI>	UDDIのInquiry URI値を指定します
-uddiPublish <UDDIPublish>	UDDIのpublish URI値を指定します
-uddiUsername <UDDIUsername>	UDDIにアクセスする際に必要なユーザー名を設定します
-uddiPassword <UDDIPassword>	UDDIにアクセスする際に必要なパスワードを設定します
-wsdlUsername	WSDL URLにアクセスする際に必要なユーザー名を設定します
-wsdlPassword	WSDL URLにアクセスする際に必要なパスワードを設定します

パラメータ	説明
-level <LEVEL>	ログ・レベルを指定する場合に使用します
-verbose	ログ・レベルの「FINE」と同じです

- 例

wsdl2uddiはJEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

以下は、wsdl2uddiツールの実行例です。

```
~$ wsdl2uddi v3 http://localhost:8088/AddNumbers/addnumbers?wsdl
-uddiInquiry http://localhost:8088/uddi/inquiry
-uddiPublish http://localhost:8088/uddi/publish
-uddiUsername jeus
-uddiPassword jeus
-verbose
```

上記のコマンドは、リモートのhttp://localhost:8088/AddNumbers/addnumbers?wsdlのWSDL文書を使用してUDDIにパブリッシュします。ここで、UDDIのInquiryおよびPublish URIとユーザー名、パスワードを指定できます。

4.12. xjc

xjcツールは、XMLスキーマ・ファイルをJavaプログラミング言語で作成されたJAXB Contentクラスに変換します。

以下は、ツールの使用方法、パラメータおよび例についての説明です。

- 使用方法

```
Usage: xjc [-options ...] <schema file/URL/dir/jar> ... [-b <bindinfo>] ...
If dir is specified, all schema files in it will be compiled.
If jar is specified, /META-INF/sun-jaxb.episode binding file will be compiled.
Options:
  -nv                : do not perform strict validation of the input schema(s)

  -extension          : allow vendor extensions - do not strictly follow the
                        Compatibility Rules and App E.2 from the JAXB Spec

  -b <file/dir>       : specify external bindings files
                        (each <file> must have its own -b)
                        If a directory is given, **/*.xjb is searched

  -d <dir>            : generated files will go into this directory

  -p <pkg>            : specifies the target package

  -httpproxy <proxy>  : set HTTP/HTTPS proxy. Format is
                        [user[:password]@]proxyHost:proxyPort

  -httpproxyfile <f> : Works like -httpproxy but takes the argument in a file
```

```

to protect password
-classpath <arg>      : specify where to find user class files
-catalog <file>       : specify catalog files to resolve external entity
                       references support TR9401, XCatalog,
                       and OASIS XML Catalog format.
-readonly             : generated files will be in read-only mode
-npa                  : suppress generation of package level annotations
                       (**/package-info.java)
-no-header            : suppress generation of a file header with timestamp
-target 2.0           : behave like XJC 2.0 and generate code that doesnt use
any
                       2.1 features.
-xmlschema            : treat input as W3C XML Schema (default)
-relaxng              : treat input as RELAX NG (experimental,unsupported)
-relaxng-compact      : treat input as RELAX NG compact syntax
                       (experimental,unsupported)
-dtd                  : treat input as XML DTD (experimental,unsupported)
-wsdl                 : treat input as WSDL and compile schemas inside it
                       (experimental,unsupported)
-verbose              : be extra verbose
-quiet                : suppress compiler output
-help                 : display this help message
-version              : display version information

Extensions:
-Xlocator              : enable source location support for generated code
-Xsync-methods         : generate accessor methods with the 'synchronized' keyword

-mark-generated        : mark the generated code as @javax.annotation.Generated
-episode <FILE>       : generate the episode file for separate compilation

```

● パラメータ

パラメータ	説明
-nv	スキーマの有効性チェックを行うか否かを設定します(true false)
-extension	XJCバインディング・コンパイラは拡張モードで実行するか否かを設定します(true false、デフォルト値: false)
-b <file/dir>	処理する1つ以上の外部バインディング・ファイルを設定します
-d <dir>	XJCバインディング・コンパイラはデフォルト値で現在のディレクトリーにJava Contentクラスを作成します。このオプションを使用すると、他のディレクトリーにクラスを作成できます
-p <pkg>	ここで明示した対象パッケージ名は、他のCustomizationも上書きします

パラメータ	説明
-httpproxy <proxy>	HTTP/HTTPSプロキシを設定します
-httpproxyfile <f>	-httpproxyオプションと似ていますが、暗号を保護するためにファイルでパラメータを使用します
-classpath <arg>	<jxb:javaType>と<xjc:superClass>Customizationに登録されているクライアント・アプリケーション・クラスを検索するパスを設定します
-catalog <file>	TR9401、XCatalog、OASIS XML Catalogのフォーマットに関する外部カタログ・ファイルを設定します
-readOnly	trueの場合、Javaソース・ファイルは読み込み可能なモードです（デフォルト値: false）
-npa	package-info.javaというファイルにパッケージ・レベルのアノテーションの作成を要約します
-no-header	タイム・スタンプがあるファイル・ヘッダーの作成を抑制します
-target (2.0 2.1)	XJC 2.0または2.1と共に動作し、2.2の機能を使用しないコードを作成します
-xmlschema	W3C XMLスキーマで入力処理するように指定します -relaxng、-relaxng-compact、-dtd、-wsdl、-xmlschemaの5つのうち -xmlschemaがデフォルト値です
-relaxng	RELAX NGで入力処理するように指定します
-relaxng-compact	RELAX NGの簡単な構文で入力処理するように指定します
-dtd	XML DTDファイルをコンパイラの入カスキーマとして使用します
-wsdl	WSDLファイルを入力し、その中に使用されているスキーマ・ファイルをコンパイルします
-verbose	Verboseメッセージを出力します
-quiet	進行情報あるいは警告メッセージのようなコンパイラ出力メッセージを表示しません
-help	コンパイラの使用方法を出力します
-version	コンパイラのバージョンを出力します

- 例

xjcはJEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

以下は、xjcツールの実行例です。

```
~$ xjc -d build/classes -p com.tmaxsoft src/conf/ts.xsd
```

上記のコマンドは、ts.xsdスキーマ・ファイルからパッケージ名がcom.tmaxsoftである、コンパイルされたJavaクラス・ファイルをbuild/classesに作成します。

4.13. schemagen

schemagenツールは、Javaクラスの各名前空間に対して、1つのスキーマ・ファイルを作成します。

以下は、ツールの使用方法、パラメータおよび例についての説明です。

- 使用方法

```
Usage: schemagen [-options ...] <java files>
Options:
    -d <path>                : specify where to place processor and javac generated
                              class files
    -cp <path>                : specify where to find user specified files
    -classpath <path>        : specify where to find user specified files
    -encoding <encoding>     : specify encoding to be used for apt/javac invocation
    -episode <file>          : generate episode file for separate compilation
    -version                  : display version information
    -fullversion              : display full version information
    -help                     : display this usage message
```

- パラメータ

パラメータ	説明
-d <path>	プロセッサとクラス・ファイルのパスを設定します
-cp <path>	ユーザーが明示するファイルを検索するパスを設定します
-classpath <path>	-cp <path>と同じです
-encoding <encoding>	apt/javacの呼び出しで使用されるエンコードを設定します
-episode <file>	分離してコンパイルを行うためのepisodeファイルを作成します
-version	バージョン情報を出力します
-fullversion	全体のバージョン情報を出力します
-help	使用方法を出力します

- 例

schemagenは、JEUS_HOME/bin/ディレクトリーに位置する一般的なスクリプト・ファイルです。

以下は、schemagenツールの実行例です。

```
~$ schemagen -d build/classes src/java/cardfile/Address.java
```

上記のコマンドは、ルート・エレメントに該当するAddress.javaソース・ファイルからbuild/classesディレクトリーにスキーマを作成します。

4.14. tcpmon

本節では、tcpmonツールについて説明します。

4.14.1. 実行

tcpmonプログラムは、送受信するTCPパケットを出力するプログラムで、HTTPのSOAPメッセージを照会する際に使用します。

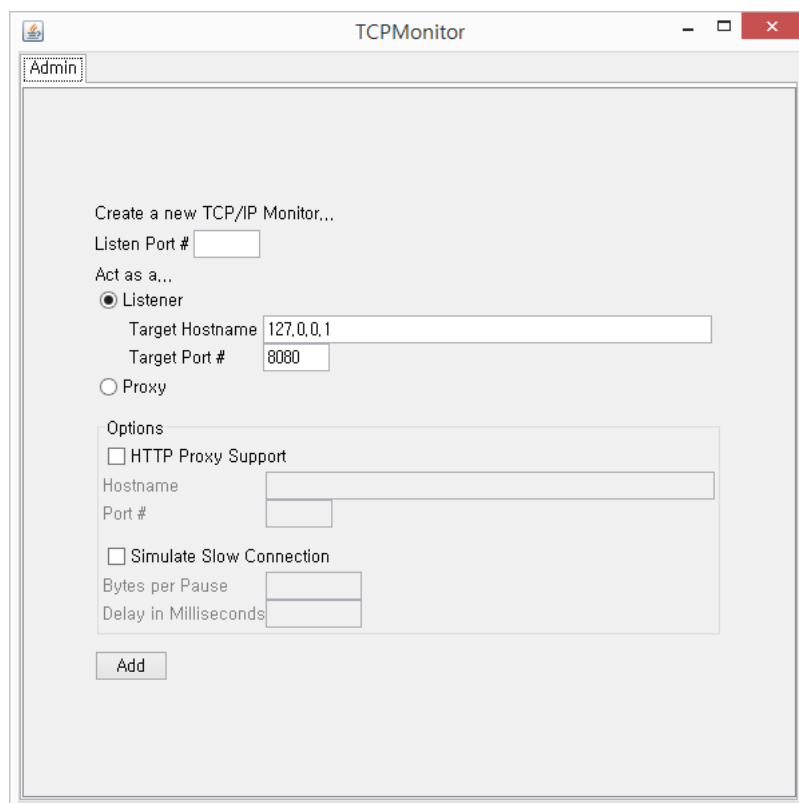
tcpmonは、以下の2つのモードで動作します。

モード	説明
Listener	tcpmonがListenポートで要求を受けて、指定された対象ホストと対象ポートに再送信します。そして、送信した要求に対する応答を受信して、その応答を要求の発信元の接続に転送します
Proxy	tcpmonを一般的なHTTPプロキシのように作動させます

以下のようにtcpmonを実行すると、[\[図 4.1\]](#)画面が表示されます。

```
~/jeus/bin$ tcpmon
```

[図 4.1] tcpmonの実行画面

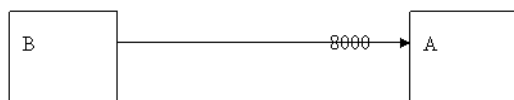


4.14.2. Listenerモードの使用

Listenerモードを使用するには、以下のようなシナリオを仮定します。

- サーバー(A)は8000番ポートでSOAP要求を受信します。
- クライアント(B)のWebサービス・クライアントが送受信するSOAPメッセージをモニタリングしようとしています。

[図 4.2] サーバー(A)が8000番ポートでSOAP要求を受信するシナリオ

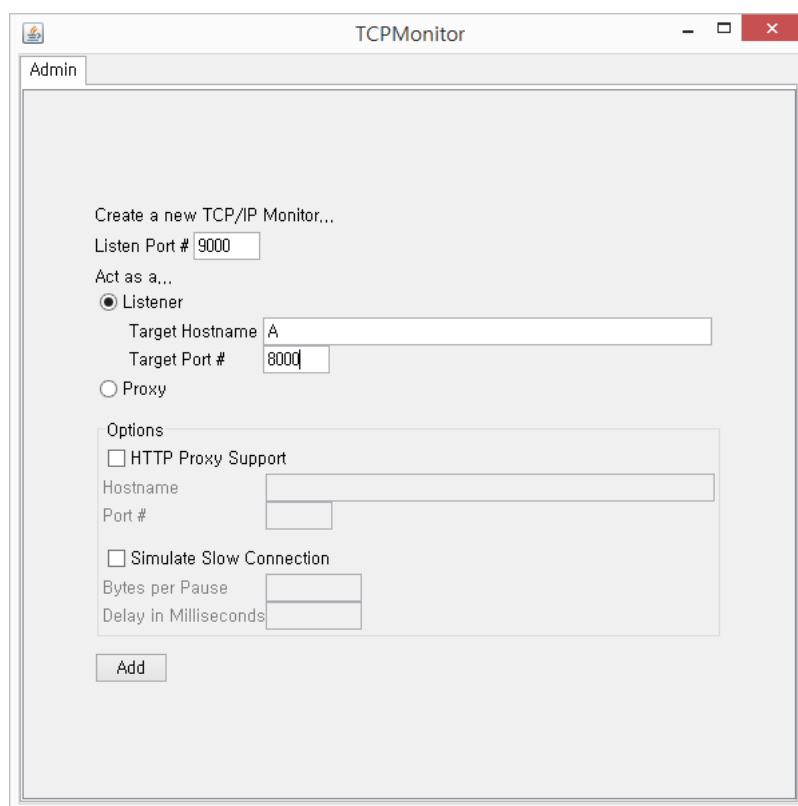


この場合、Webサービス・クライアント・プログラムではtcpmonを使用してSOAPメッセージをモニタリングできます。

クライアント(B)でtcpmonを実行後、以下のように設定します。

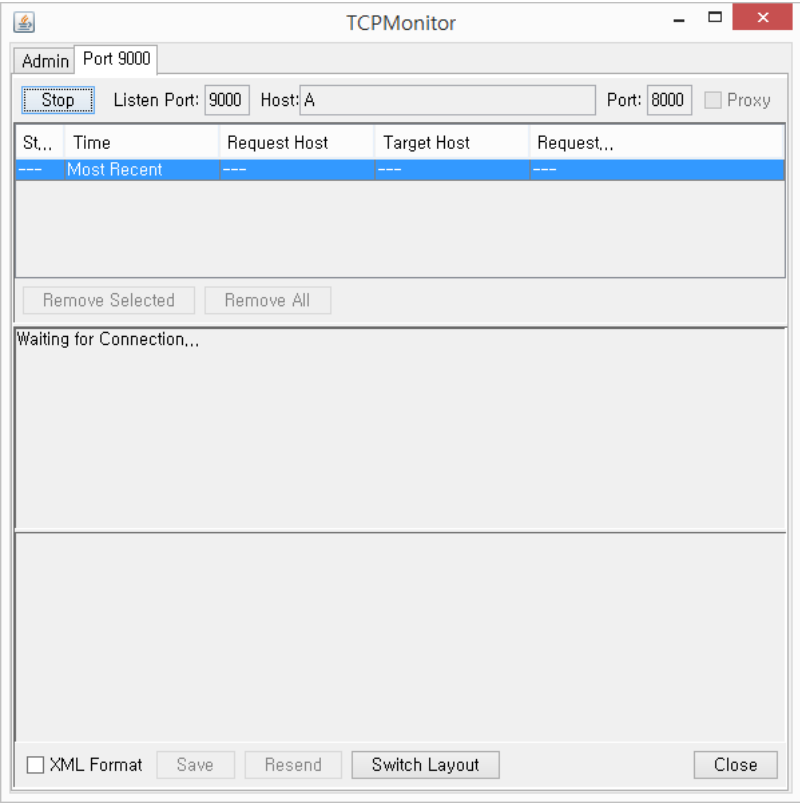
1. **[Admin]**タブで「**Listen Port**」は「9000」と入力し、「**Act as a**」は「**Listener**」を選択します。「**Listener**」の下位項目の「**Target Hostname**」は「A」、「**Target Port**」は「8000」と入力します。各値を入力後、**[Add]**ボタンをクリックします。

[図 4.3] Listenerモードの入力画面



2. 以下のように[Port 9000]タブが作成されます。このタブをクリックすると、以下のような画面が表示されます。

[図 4.4] モニタリング画面

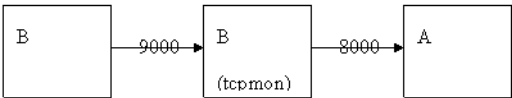


画面のチェック・ボックスと各ボタンについての説明です。

項目	説明
XML Format	XMLを閲覧しやすい形態にソートし、TCPMON画面に出力します
[Save]ボタン	現在のSOAPメッセージを保存します
[Resend]ボタン	現在のRequest SOAPメッセージを再送信します
[Switch Layout]ボタン	メッセージ出力画面の縦/横の分割モードを変更します

3. 上記のように設定後、クライアント・プログラムでサーバー(A)の8000番ポートに接続する代わりにクライアント(B)の9000番ポートに接続します。その後、クライアント・プログラムを実行します。すると、tcpmonが9000番ポートで要求を受信し、サーバー(A)の8000番ポートに送信して応答を待ちます。応答を受け取り、クライアントがtcpmonに接続したコネクションの応答を返し、送受信したメッセージをtcpmonに出力します。

[図 4.5] tcpmonが9000番ポートで受信した要求を受け取り、Aの8000番ポートに送信



Webサービスのエンドポイント・アドレスの変更

tcpmonをListenerモードで使用する場合、Webサービスのエンドポイント・アドレスを変更するために、クライアント・プログラムやクライアントの設定を変更する必要があります。

- J2SE JAX-WS Webサービス・クライアントの場合

以下の例のようにコードを修正し、接続するWebサービスのエンドポイント・アドレスを変更します。

```
(( javax.xml.ws.BindingProvider)port).getRequestContext().put(
    javax.xml.ws.BindingProvider.ENDPOINT_ADDRESS_PROPERTY,
    "http://localhost:9000/ws/AddNumbersService");
```

– port

サービス・エンドポイント・インターフェースのJAX-WS BindingProviderオブジェクトです。

– javax.xml.ws.BindingProvider.ENDPOINT_ADDRESS_PROPERTY

エンドポイント・アドレスを指定するためのJAX-WS仕様で定義した標準プロパティです。

- J2SE JAX-RPC Webサービス・クライアントの場合

以下の例のようにコードを修正し、接続するWebサービスのエンドポイント・アドレスを変更します。

```
(( javax.xml.rpc.Stub)port)._setProperty(
    javax.xml.rpc.Stub.ENDPOINT_ADDRESS_PROPERTY,
    "http://localhost:9000/ws/AddressBookService");
```

– port

サービス・エンドポイント・インターフェースのJAX-RPC Stubオブジェクトです。

– javax.xml.rpc.Stub.ENDPOINT_ADDRESS_PROPERTY

エンドポイント・アドレスを指定するためのJAX-RPC仕様で定義した標準プロパティです。

- J2EE Webサービス・クライアントの場合

コード修正無しでJEUSの設定ファイル(jeus-web-dd.xmlまたはjeus-ejb-dd.xml)を<service-client>に<stub-property>を追加して、上記のプロパティを適用できます。

```
<service-client>
  <port-info>
    <stub-property>
      <name>javax.xml.rpc.service.endpoint.address</name>
      <value>
        http://localhost:9000/ws/AddressBookService
      </value>
    </stub-property>
  </port-info>
</service-client>
```

```
</stub-property>
</port-info>
</service-client>
```

4.14.3. Proxyモードの使用

Proxyモードは、tcpmonを一般的なHTTPプロキシのように動作させます。Proxyモードを使用するには、**[Admin]**タブで「Listenポート」を入力し、「Act as a」のProxyを選択して**[Add]**ボタンをクリックします。

Proxyモードの場合、アプリケーションを修正せずに実行する際に、以下のオプションを設定してtcpmonを使用できます。

オプション	説明
-Dhttp.proxyHost	tcpmonが実行されているホスト名またはIPアドレスを入力します
-Dhttp.proxyPort	tcpmonのListenポートを入力します

クライアント・アプリケーションがcom.acme.AddressBookClientクラスの場合、以下のようにオプションを設定してjavaを実行します。

```
java -Dhttp.proxyHost=B -Dhttp.proxyPort=9000 com.acme.AddressBookClient
```

4.14.4. その他の機能

各ポート別のタブでSOAPメッセージをフォーマットするか、保存または再送信することができます。

Part III. Antタスク

第5章 Antタスク

本章では、Antタスクについて説明します。

5.1. 概要

JEUSでは、アプリケーションの開発時に特定タスクを自動化するため、以下のようなAntタスクを提供します。ユーザーはAntで実行するタスクを記述したBuildファイルを作成する必要があります。

本節では、各機能のBuildファイルに記述する必要がある内容について説明し、そのファイルを使用して各Antタスクを実行するプロセスについて説明します。Antの設定方法と使用方法については<http://ant.apache.org>を参照してください。

Buildファイル : build.xml

build.xmlは、Antタスクを記述したXML形式のAnt Buildファイルです。Buildファイルのクリティカルな部分は<project>に含まれており、一般的に1つのプロジェクトに複数の<target>が存在します。<target>は実際に実行する必要のあるAntタスクを定義します。

JEUSで提供するAntタスクを使用するには、以下のようにタスク定義をbuild.xmlに追加する必要があります。

```
<taskdef resource="jeus/util/ant/jeusant.properties">
  <classpath>
    <path refid="jeus.libraries"/>
  </classpath>
</taskdef>
```

プロジェクトのプロパティは<project>の下位に定義し、「\${property-name}」の表現式によってプロジェクト内でプロパティ値を使用できます。

注

JEUSで提供するAntタスクを利用する際は、なるべくBuildファイルのプロジェクトに「jeus.home」を設定するのが安全です。これは、1つのプロジェクトで多数のターゲットを実行する際に、特定のターゲットはJEUS_HOME情報を必要とする場合があるためです。

JEUS_HOMEはstatic final変数であるため、1つのJVMで初めに設定された値で固定されるという特徴があります。JEUS_HOME情報を必要としないターゲットが先に実行されると、JEUS_HOMEにnull値が

設定されます。その後、JEUS_HOMEが必要なターゲットが実行される際に、先のターゲットが設定したJEUS_HOMEの値を利用するため、目的の動作と異なる場合があります。

5.2. 環境構成のAntタスク

本節では、JEUS環境を構成するAntタスクについて説明します。

5.2.1. create-domain

create-domain Antタスクを使用して、新しいドメイン環境を構成できます。

JEUS_HOME/setupの下位に提供されるbuild.xmlに存在するタスクで、以下のファイルでデフォルト設定値を確認できます。

```
JEUS_HOME/setup/domain-config-template.properties
```

以下は、create-domainタスクのプロパティについての説明です。

[表 5.1] create-domainタスクのプロパティ

プロパティ	説明
domain	環境を構成するドメインの名前です(デフォルト値: jeus_domain)
servername	DASの名前です(デフォルト値: adminServer)
nodename	DASで使用するノード名です(デフォルト値: node1)
jeus.address	DASのデフォルト・アドレスです(デフォルト値: 0.0.0.0)
jeus.port	DASのデフォルト・ポートです(デフォルト値: 9736)
jms.port	DASのJMSポートです(デフォルト値: 9741)
http.port	DASのHTTPポートです(デフォルト値: 8808)
jvm.config	DASのJVM設定です(デフォルト値: -Xmx1024m -XX:MaxPermSize=128m)
jeus.username	ドメインの管理アカウントの名前です。ドメインの管理アカウント名に必ず変更することを推奨します(デフォルト値: administrator)
jeus.password	ドメインの管理アカウントのパスワードで、平文ではjeusで、base64でエンコードされた値です。Ant設定あるいはjeusadminのset-passwordコマンドを通じて変更することを推奨します (デフォルト値: {base64}amV1cw==)
usevirtualmulticast	SCF(System Clustering Framework)の使用時にTCP通信を使用した仮想マルチキャストを使用するか否かを設定します(true false、デフォルト値: false)
heartbeataddress	SCFに使用されるheartbeatのアドレスです。

プロパティ	説明
	uservirtualmulticast値がtrueの場合は適用されません(デフォルト値: 230.30.1.1)
heartbeatport	SCFに使用されるheartbeatのポートです。 uservirtualmulticast値がtrueの場合は適用されません(デフォルト値: 3030)

例

以下は、Antタスクを使用したドメインの作成例です。

ドメインの名前は「domain1」、DASの名前は「adminServer」、DASのjvmは「-Xmx512m -XX:MaxPermSize=256m」と設定した例です。すべてデフォルト値が存在するため、オプションは必ずしも指定する必要はありませんが、以下の例では設定しています。

```
JEUS_HOME/setup$ ant create-domain -Ddomain=domain1 -Dservername=adminServer
-Djvm.config="-Xmx512m -XX:MaxPermSize=256m" -Dheartbeataddress=230.30.1.1
-Dheartbeatport=3030
Buildfile: JEUS_HOME/setup/build.xml

create-domain:
[echo] Creating a domain configuration: domain1dir="JEUS_HOME/domains/domain1 domain
= domain1, server-name = adminServer, admin password={base64}amVlcw==, server
base port=9736, server base listen address=${jeus.address}
[mkdir] Created dir: JEUS_HOME/domains/domain1
[mkdir] Created dir: JEUS_HOME/domains/domain1/.applications
[mkdir] Created dir: JEUS_HOME/domains/domain1/.deploymentplans
[mkdir] Created dir: JEUS_HOME/domains/domain1/bin
[mkdir] Created dir: JEUS_HOME/domains/domain1/config
[mkdir] Created dir: JEUS_HOME/domains/domain1/lib
[mkdir] Created dir: JEUS_HOME/domains/domain1/lib/application
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers
[mkdir] Created dir: JEUS_HOME/domains/domain1/config/servlet
[mkdir] Created dir: JEUS_HOME/domains/domain1/config/security
[copy] Copying 2 files to JEUS_HOME/domains/domain1/config
[copy] Copying 1 file to JEUS_HOME/domains/domain1/config/servlet
[copy] Copying 1 file to JEUS_HOME/domains/domain1/config/security
[copy] Copied 1 empty directory to 1 empty directory under
JEUS_HOME/domains/domain1/config/security
[copy] Copying 3 files to JEUS_HOME/domains/domain1/config/security
[echo] productionmode = true
[java] JVM args ignored when same JVM is used.
[copy] Copying 3 files to JEUS_HOME/domains/domain1/bin
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer
```

```

[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/bin
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/.workspace
[mkdir] Created dir:
JEUS_HOME/domains/domain1/servers/adminServer/.workspace/deployed
[mkdir] Created dir:
JEUS_HOME/domains/domain1/servers/adminServer/.workspace/deployed/deploymentplans
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/logs
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/lib
[mkdir] Created dir: JEUS_HOME/domains/domain1/servers/adminServer/lib/application

BUILD SUCCESSFUL
Total time: 2 seconds
JEUS_HOME/setup$

```

5.3. サーバー制御のAntタスク

本節では、サーバーを制御するAntタスクについて説明します。

5.3.1. boot

bootタスクは、サーバーを起動します。bootタスクを通じて、サーバーを起動する際はノード・マネージャーに命令を送信し、ノード・マネージャーで該当サーバーを起動します。したがって、必ずノード・マネージャーが起動した状態である必要があります。

以下は、bootタスクのプロパティについての説明です。

[表 5.2] boot Taskプロパティ

プロパティ	説明
domainName	起動するサーバーが属するドメインの名前を設定します(必須入力項目、文字列型)
serverName	起動するサーバーの名前を設定します(必須入力項目、文字列型)
host	ノード・マネージャーに接続するホストを設定します(必須入力項目、文字列型)
port	ノード・マネージャーに接続するポートを設定します(必須入力項目、文字列型)
dasUrl	MSを起動時にDASのURLを設定します(文字列型)
connectionType	ノード・マネージャーが接続するコネクション・タイプを設定します。ssl、plainのうちどちらかを設定します(文字列型)
truststorePath	ノード・マネージャーがsslタイプで起動した場合、ノード・マネージャーに接続するための認証に必要なTruststoreファイルのパスを設定します(文字列型)
truststorePassword	ノード・マネージャーがsslタイプで起動した場合、ノード・マネージャーに接続するための認証に必要なTruststoreファイルのパスワードを設定します(文字列型)

プロパティ	説明
adminServer	起動するサーバーがDASであるのかMSであるのかを設定します(必須入力項目、文字列型)
force	サーバーの起動時にデプロイの失敗などの理由でサーバーがSTANDBY状態になった場合、サーバーを強制的にRUNNING状態にするか否かを設定します(デフォルト値: false)
standby	サーバーをSTANDBY状態までのみ起動させます(デフォルト値: false)
username	サーバーを起動するユーザー名を設定します(必須入力項目、文字列型)
password	サーバーを起動するパスワードを設定します(必須入力項目、文字列型)

5.3.2. down

downタスクは、JEUSを終了します。

以下は、downタスクのプロパティについての説明です。

[表 5.3] downタスクのプロパティ

プロパティ	説明
domainName	終了するサーバーが属するドメイン名を設定します(必須入力項目、文字列型)
serverName	終了するサーバーの名前を設定します(必須入力項目、文字列型)
host	ノード・マネージャーに接続するホストを設定します(必須入力項目、文字列型)
port	ノード・マネージャーに接続するポートを設定します(必須入力項目、文字列型)
connectionType	ノード・マネージャーに接続するコネクション・タイプを設定します。ssl、plainのうちどちらかを設定します(文字列型)
truststorePath	ノード・マネージャーがsslタイプで起動した場合、ノード・マネージャーに接続するための認証に必要なTruststoreファイルのパスを設定します(文字列型)
truststorePassword	ノード・マネージャーがsslタイプで起動した場合、ノード・マネージャーに接続するための認証に必要なTruststoreファイルのパスワードを設定します(文字列型)
username	サーバーを終了するユーザー名を設定します(必須入力項目、文字列型)
password	サーバーを終了するパスワードを設定します(必須入力項目、文字列型)

5.3.3. 実行

build.xmlを作成した例と、該当ファイルを通じてサーバーの起動、終了を実行する例について説明します。

build.xmlの作成例

以下は、domain1に「adminServer」という名前を持つDASを起動および終了するbuild.xmlの例です。

[例 5.1] boot & down Antタスクのbuildファイルの例

```
<?xml version="1.0"?>

<project name="example" default="boot" basedir=". ">
  <property environment="env"/>
  <!-- jeus.home project property is required
        when you run the various tasks in an ant jvm and
        one of the various tasks requirs JEUS_HOME information -->
  <property name="jeus.home" value="${env.JEUS_HOME}" />

  <!-- set properties to be needed for boot & down task -->
  <property name="domain.name" value="domain1"/>
  <property name="server.name" value="adminServer"/>
  <property name="das.url" value="" />
  <property name="force" value="false"/>
  <property name="standby" value="false"/>
  <property name="host" value="host1"/>
  <property name="port" value="9736"/>
  <property name="connect.type" value="plain"/>
  <property name="truststore.path" value="" />
  <property name="truststore.password" value="" />
  <property name="username" value="administrator"/>
  <property name="password" value="jeus"/>
  <property name="isAdminServer" value="false"/>

  <!-- set the library-classpath or to run the task class -->
  <path id="jeus.libraries">
    <fileset dir="${jeus.home}/lib/system" includes="*.jar"/>
  </path>

  <!-- include the task definition resource file -->
  <taskdef resource="jeus/util/ant/jeusant.properties">
    <classpath>
      <path refid="jeus.libraries"/>
    </classpath>
  </taskdef>

  <target name="init">
  </target>

  <!-- boot task -->
  <target name="boot" depends="init">
    <boot domainName="${domain.name}"
          serverName="${server.name}"
          host="${host}"
          port="${port}"
```

```

        connectionType="${connect.type}"
        truststorePath="${truststore.path}"
        truststorePassword="${truststore.password}"
        adminServer="${isAdminServer}"
        dasUrl="${das.url}"
        force="${force}"
        standby="${standby}"
        username="${username}"
        password="${password}" />
    </target>

    <!-- down task -->
    <target name="down" depends="init">
        <down domainName="${domain.name}"
            serverName="${server.name}"
            host="${host}"
            port="${port}"
            connectionType="${connect.type}"
            truststorePath="${truststore.path}"
            truststorePassword="${truststore.password}"
            username="${username}"
            password="${password}" />
    </target>

</project>

```

実行例

Antを通じてサーバーを起動するためには、ノード・マネージャーが「startNodeManager」スクリプトを通じて起動した状態である必要があります。この状態でbootコマンドを実行するとサーバーが起動し、downコマンドを実行するとサーバーが終了します。

- boot

```

$ ant boot
Buildfile: build.xml

init:

boot:
    [boot] Succeed to start server [adminServer]

BUILD SUCCESSFUL
Total time: 11 seconds

```

- down

```

$ ant down
Buildfile: build.xml

init:

down:
    [down] Succeed to stop server [adminServer]

BUILD SUCCESSFUL
Total time: 0 seconds

```

5.4. EJBのAntタスク

JEUSベースでEJBコンポーネントを開発する際にタスクを効率的に実行できるように、いくつかのAntタスクを提供します。

5.4.1. appcompiler

appcompiler Antタスクは、プレ・デプロイメント作業を行った後、EJBモジュールまたは個別のEJB Beanに対し、必要なRMIスタブとSkeletonクラスを任意で作成する場合に使用します。また、これはクライアントからリモートにあるデプロイ済みのBeanとの通信に必要なクラス・ファイル、つまりEJBクライアントのJARを作成できます。

以下は、appcompiler Antタスクのプロパティについての説明です。

[表 5.4] appcompiler Antタスクのプロパティ

プロパティ	説明
jeusHome	JEUS_HOMEを設定します(Stringタイプ)
client	作成されるStubクラスが格納されるクライアント・ビュー・ファイルを設定します(Stringタイプ)
keep	コンパイル中に作成されたソースを維持するか否かを設定します(デフォルト値 : false)
jspmap	servlet-mapping table(jeus_jspmap.xml)を作成するか否かを設定します (デフォルト値 : false)
ejbJar	コンパイル時に使用するejb-jar.xmlファイルを設定します(Stringタイプ)
jeusEjbDd	コンパイル時に使用するjeus-ejb-dd.xmlファイルを設定します(Stringタイプ)
name	デプロイに使用する名前を設定します
target	コンパイル対象のアプリケーションとstand-aloneモジュール・ファイルを設定します (必須入力項目。Stringタイプ)

例

以下は、build.xmlの作成例とbuild.xmlを使用したappcompilerの実行例です。

- build.xmlの作成例

[例 5.2] appcompiler Antタスクのbuildファイルの例

```
<?xml version="1.0"?>

<project name="example" default="appcompiler" basedir=".">
  <property environment="env"/>
  <!-- jeus.home project property is required
       when you run the various tasks in an ant jvm and
       one of the various tasks requirs JEUS_HOME information -->
  <property name="jeus.home" value="${env.JEUS_HOME}"/>

  <!-- set properties to be needed for appcompiler task -->
  <property name="client" value="client_view.jar"/>
  <property name="keep" value="false"/>
  <property name="jspmap" value="false"/>
  <property name="ejbjar" value="ejb-jar.xml"/>
  <property name="jeusejbdd" value="jeus-ejb-dd.xml"/>
  <property name="targetfile" value="ejb"/>

  <!-- set the library-classpath or to run the task class -->
  <path id="jeus.libraries">
    <fileset dir="${jeus.home}/lib/system" includes="*.jar"/>
  </path>

  <!-- include the task definition resource file -->
  <taskdef resource="jeus/util/ant/jeusant.properties">
    <classpath>
      <path refid="jeus.libraries"/>
    </classpath>
  </taskdef>

  <target name="init">
  </target>

  <!-- appcompiler task -->
  <target name="appcompiler" depends="init">
    <appcompiler jeusHome="${jeus.home}"
      client="${client}"
      keep="${keep}"
      jspmap="${jspmap}"
      ejbJar="${ejbjar}"
```

```

        jeusEjbDd="${jeusejbdd}"
        target="${targetfile}"/>
    </target>

</project>

```

• 実行例

```

$ ant appcompiler
Buildfile: build.xml

init:

appcompiler:

BUILD SUCCESSFUL
Total time: 3 minutes 7 seconds

```

5.4.2. ejbddinit

ejbddinit Antタスクは、JEUS EJB DDの作成時に使用します。ejbddinitについての詳細は「[4.4. ejbddinit](#)」を参照してください。

以下は、ejbddinit Antタスクのプロパティについての説明です。

[表 5.5] ejbddinit Antタスクのプロパティ

プロパティ	説明
propertyFile	ejbddinitの実行時に参照するプロパティ・ファイルを指定します(Stringタイプ)
logginginglevel	ejbddinitの実行時に画面に出力されるログのレベルを指定します。 このレベルはJava SE Logging APIのレベルに従います(デフォルト値 : INFO)
target	ejbddintの実行対象となるEJBモジュールのパスを指定します。EJBモジュールは、JARアーカイブあるいはディレクトリーの場合があります。 (必須入力項目。プロパティ・ファイルが実行対象情報を持っている場合は省略可能)

ejbddinit Antタスクのスクリプトに制限的にejbddinitプロパティを設定できます。

ejbddinitプロパティ・ファイルのプロパティは、ejbddinit Antスクリプトで以下の規則で設定します。ejbddinitプロパティの詳細については[4.4節「プロパティ一覧」](#)を参照してください。

[表 5.6] ejbddinitプロパティ・ファイルとejbddinit Antスクリプトにおけるejbddinitプロパティの設定例

ejbddinitプロパティ・ファイル	ejbddinit Antスクリプト
export-name=%{ejb-class}	<property name="export-name" value="%{ejb-class}"/>
thread-max=100	<property name="thread-max" value="100"/>
HelloBean.export-port=55555	build.xmlでは特定のEJBコンポーネント(HelloBean)に対するejbddinitプロパティの設定が不可能

ejbddinitプロパティ・ファイルのように、Antスクリプトでもexport-nameプロパティに「%{ejb-class}」のようなパターンを使用できます。export-nameに関連するパターンの使用については[4.4節「export-nameプロパティのパターンのサポート」](#)を参照してください。

Antスクリプトでは、特定のEJBコンポーネントに対するejbddinitプロパティの設定はサポートしないため、特定のEJBコンポーネントに対するejbddinitプロパティの設定が必要な場合は、ejbddinitプロパティ・ファイルを別途作成する必要があります。特定のEJBコンポーネントに対する設定は、ejbddinitプロパティ・ファイルとAntスクリプトの共通設定よりも優先されます。ejbddinitプロパティ・ファイルとejbddinit Antスクリプトに同じ設定が存在する場合は、Antスクリプトの設定を優先します。

例

ejbddinit Antスクリプトの作成例と、それを使用してejbddinit Antタスクを実行する例は以下のとおりです。

● ejbddinit Antスクリプトの作成例

[例 5.3] ejbddinit Antスクリプトの例

```
<?xml version="1.0"?>
<project name="example" default="ejbddinit" basedir=".">
  <property environment="env"/>
  <!-- jeus.home project property is required
       when you run the various tasks in an ant jvm and
       one of the various tasks requirs JEUS_HOME information -->
  <property name="jeus.home" value="${env.JEUS_HOME}"/>

  <!-- set properties to be needed for ejbddinit task -->
  <property name="targetfile" value="ejb.jar"/>
  <property name="logginglevel" value="FINE"/>
  <property name="propertyfile" value="ejbddinit.properties"/>

  <!-- set properties to be needed for ejbddinit properties -->
  <property name="export-name" value="%{ejb-class}"/>
  <property name="thread-max" value="100"/>

  <!-- set the library-classpath or to run the task class -->
  <path id="jeus.libraries">
    <fileset dir="${jeus.home}/lib/system" includes="*.jar"/>
  </path>
</project>
```

```

</path>

<!-- include the task definition resource file -->
<taskdef resource="jeus/util/ant/jeusant.properties">
  <classpath>
    <path refid="jeus.libraries"/>
  </classpath>
</taskdef>

<target name="init">
</target>

<!-- ejbddinit task -->
<target name="ejbddinit" depends="init">
  <ejbddinit loggingLevel="${logginglevel}"
    property="${propertyfile}"
    target="${targetfile}"
    exportName="${export-name}"
    threadMax="${thread-max}">
  </ejbddinit>
</target>
</project>

```

● ejbddinit Antタスクの実行例

```

$ ant ejbddinit
Buildfile: build.xml

init:

ejbddinit:
[ejbddinit] LoadFile: /jeus/sample/ejbddinit.properties
[ejbddinit] Source=/jeus/sample/ejbddinit/ejb.jar
[ejbddinit] Successfully configured the parameters.
[ejbddinit] Deployment descriptor initialization started.
[ejbddinit] Creating JEUS descriptors.
[ejbddinit] Deployment descriptor initialization finished.

BUILD SUCCESSFUL
Total time: 2 seconds

```

5.5. WebサービスのAntタスク

本節では、JEUSでWebサービスとWebサービス・クライアントを作成するために提供するAntタスクについて説明します。

5.5.1. java2wsdl

java2wsdlタスクは、SET(Service Endpoint Interface)クラスと任意のJavaで実装したクラスから以下のファイルを作成します。

- WebサービスのWSDLファイル
- JAX-RPCマッピング・ファイル

参考

java2wsdlのAntタスクを定義するクラスは「jeus.util.ant.webservices.Java2WsdITask」です。

以下は、java2wsdlタスクのプロパティについての説明です。

[表 5.7] java2wsdl Antタスクのプロパティ

プロパティ	説明
configfilepath	Webサービスの設定ファイルのパスを設定します(必須入力項目。文字列型)
classpath	Webサービスを構成するコンパイルされたJavaクラスのクラス・パスを設定します(必須入力項目。文字列型)
destDir	作成されたWSDLファイルが格納されるディレクトリーの絶対パスを設定します(文字列型)
level	ログ・レベルを設定します(文字列型)
verbose	Verboseメッセージを出力するか否かを設定します

ネストした要素

<java2wsdl>はAntの<classpath>要素を持っています。

例

以下は、build.xmlの作成例と、build.xmlを使用したjava2wsdlの実行例です。

- build.xmlの作成例

[例 5.4] java2wsdl Antタスクのbuildファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="java2wsdl" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
```

```

<taskdef name="java2wsdl"
    classname="jeus.util.ant.webservices.Java2Wsd1Task">
    <classpath refid="jeus.libraries.classpath" />
</taskdef>
<target name="-post-compile">
    <java2wsdl destDir="${build.classes.dir}"
        verbose="true"
        configfilepath="${src.conf}/service-config.xml">
        <classpath refid="classpath" />
    </java2wsdl>
</target>
</project>

```

● 実行例

```

$ jant

...

[java2wsdl] Building Web Services : DocLitEchoService
[java2wsdl] Generating WSDL File - jeus/build/classes/DocLitEchoService.wsdl
[java2wsdl] Generating JAX-RPC Mapping File -
jeus/build/classes/DocLitEchoService-mapping.xml

...

BUILD SUCCESSFUL
Total time: 11 seconds

```

5.5.2. wsdl2java

wsdl2javaタスクは、WebサービスのWSDLから、以下の2つのうち1つを作成します。

- クライアント側のWebサービスのスタブのJavaソースコード
- サーバー側のWebサービスのインターフェースのJavaソースコード

参考

wsdl2javaのAntタスクを定義するクラスは「jeus.util.ant.webservices.Wsdl2JavaTask」です。

以下は、wsdl2javaタスクのプロパティについての説明です。

[表 5.8] wsdl2javaタスクのプロパティ

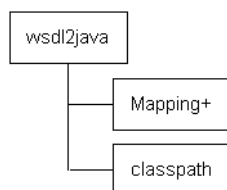
プロパティ	説明
wsdl	Javaソース・ファイルの作成に使用するWSDLのURLまたは絶対パスを設定します (必須入力項目。文字列型)
mode	Javaソース・ファイルの作成に使用するモードを設定します(必須入力項目。文字列型) 以下のうち1つを設定します <ul style="list-style-type: none"> – gen:client – gen:server – gen – import:client – import:server – import
destDir	作成されたJavaファイルが格納されるディレクトリーの絶対パスを設定します (必須入力項目。文字列型)
classDestDir	コンパイルされたクラス・ファイルが作成されるディレクトリーを設定します(文字列型)
inputMapping	Javaクラスの作成に使用する入力JAX-RPCマッピング・ファイルを設定します(文字列型)
package	WSDLのすべての名前空間URIのJavaパッケージ名を設定します (文字列型)
outputMapping	入力WSDLに対する出力JAX-RPCマッピング・ファイルを設定します(文字列型)
doCompile	trueを指定した場合、作成されたJavaソース・ファイルがコンパイルされます(デフォルト値 : true)
username	WSDLのURLへのアクセスに必要なユーザー名を設定します(文字列型)
password	WSDLのURLへのアクセスに必要なパスワードです(文字列型)
keepSrc	trueを指定した場合、作成されたJavaソース・ファイルは削除されません(デフォルト値 : true)
nowraped	trueを指定した場合、WSDLに対するwrappedモード検索を使用できません (デフォルト値 : false)
dataHandlerOnly	trueを指定した場合、MIME型に対してjavax.activation.DataHandlerを適用します (デフォルト値 : false)

プロパティ	説明
nodatabinding	すべてのWSDL message partに対してjavax.xml.soap.SOAPElementを適用します (デフォルト値 : false)
soapver	stub/tieクラスで使用するSOAPバージョンを設定します <ul style="list-style-type: none"> – 11: SOAP 1.1 (デフォルト値) – 12: SOAP 1.2
resolveDir	リモートのWSDLファイル(スキーマを含む)をローカルのレポジトリに保存する際の値を設定します(文字列型)
DDGen	JEUSの基本的な配備記述子(webservices.xmlファイルとモードがwebの場合はweb.xml、ejbの場合はejb-jar.xmlファイル)を作成する際に使用します(文字列型)
level	ログ・レベルを設定します(文字列型)
verbose	trueを指定した場合、Verboseが出力されます(デフォルト値 : false)

ネストした要素

<wsdl2java>は、ネストしたAntの<classpath>と<mapping>要素を持っています。<wsdl2java>のAntタスクの構造は以下のとおりです。(「+」は1つ以上の要素を持つことができるという意味です)

【図 5.1】 wsdl2javaタスクの構成



<mapping>

JavaパッケージとWSDLの名前空間をマッピングします。ユーザーが<mapping>を省略すると、すべての名前空間URIは<wsdl2java>要素であるパッケージのプロパティに記述されたパッケージにマッピングされます。

以下は、<mapping>のプロパティについての説明です。

【表 5.9】 wsdl2javaタスク - mapping要素のプロパティ

プロパティ	説明
package	Javaパッケージの名前です(必須入力項目。文字列型)

プロパティ	説明
namespace	WSDLの名前空間URIです(必須入力項目。文字列型)

参考

ユーザーが<wsdl2java>要素のパッケージのプロパティを明示した場合、このプロパティ値は<mapping>要素で設定された値より優先されます。すなわち、<mapping>要素に指定された値は無効となります。

例

以下は、build.xmlの作成例と、build.xmlを使用したwsdl2javaの実行例です。

● build.xmlの作成例

[例 5.5] wsdl2java Antタスクのbuildファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="wsdl2java" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="wsdl2java"
    classname="jeus.util.ant.webservices.Wsdl2JavaTask">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <wsdl2java destDir="${build.classes.dir}"
      verbose="true"
      mode="import:server"
      doCompile="true"
      noDataBinding="true"
      package="sample.nodatabinding.service"

      outputmapping="${build.classes.dir}/BookQuoteService-mapping.xml"
      wsdl="${src.conf}/BookQuoteService.wsdl">
      <classpath refid="classpath" />
    </wsdl2java>
  </target>
</project>
```

● 実行例

```
$ jant
Buildfile: build.xml
...
```

```
[wsdl2java] Compiling generated sources(1)...  
...  
BUILD SUCCESSFUL  
Total time: 6 seconds
```

5.5.3. wsgen

wsgenタスクは、service endpoint interfaceクラス(そして、任意のJavaで実装したクラス)から以下を作成します。

- ポータブル・アーティファクト
- WebサービスのWSDLファイル(オプション)

参考

wsgen Antタスクを定義するクラスは「jeus.webservices.jaxws.tools.WsGen2」です。

以下は、wsgenタスクのプロパティについての説明です。

[表 5.10] wsgenタスクのプロパティ

プロパティ	説明
sei	サービス・エンドポイント・インターフェース・クラスの名前を設定します(必須入力項目。文字列型)
destdir	作成されたクラス・ファイルが格納されるディレクトリーの絶対パスを設定します(必須入力項目。文字列型)
classpath	入力されたクラス・ファイルを検索する位置を設定します(文字列型)
cp	classpathプロパティと同じです(文字列型)
resourcedestdir	genwsdlプロパティと一緒に使用します。WSDLファイルを作成する位置を指定します(文字列型)
sourcedestdir	作成されるソース・ファイルの位置を設定します(文字列型)
keep	作成されるファイルを保存するか否かを設定します(デフォルト値: false)
verbose	Verboseメッセージを出力するか否かを設定します(デフォルト値: false)
genwsdl	WSDLファイルを作成するか否かを設定します(デフォルト値: false)
protocol	genwsdlプロパティと一緒に使用します。

プロパティ	説明
	wsdl:binding要素に使用されるプロトコルを設定します。デフォルト値はsoap1.1であり、Xsoap1.2のような値も可能です。Xsoap1.2は標準ではないため、extensionプロパティと一緒に使用できます(文字列型)
servicename	genwsdlプロパティと一緒に使用します。作成されるWSDLファイルの特定のwsdl:service要素の名前を設定する際に使用します(文字列型)
portname	genwsdlプロパティと一緒に使用します。作成されるWSDLファイルの特定のwsdl:portname要素の名前を設定する際に使用します(文字列型)
extension	ベンダー特定の拡張性を明示する際に使用します。相互運用および移植性が低下する可能性があります(ブーリアン型)
policy	Webサービス・ポリシーの設定ファイルを読み込みます(文字列型)

実行

以下は、build.xmlの作成例と、build.xmlを使用したwsngenの実行例です。

● build.xmlの作成例

[例 5.6] wsngen Antタスクのbuildファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="wsngen" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="wsngen" classname="com.sun.tools.ws.ant.WsGen">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-post-compile">
    <wsngen sei="fromjava.server.AddNumbersImpl"
      destdir="${build.classes.dir}"
      classpath="${build.classes.dir}"
      resourcedestdir="${build.classes.dir}"
      sourcedestdir="${build.classes.dir}"
      genwsdl="true" />
  </target>
</project>
```

● 実行例

```
~/jeus$ jant
Buildfile: build.xml
...
```

```
[echo] Compiling wsgen...
...

BUILD SUCCESSFUL
Total time: 6 seconds
```

5.5.4. wsimport

wsimportタスクは、WebサービスのWSDLから、以下の2つうち1つを作成します。

- クライアント側のWebサービスのスタブのJavaソース・コード
- サーバー側のWebサービス・インターフェースのJavaソース・コード

参考

wsimport Antタスクを定義するクラスは「jeus.webservices.jaxws.tools.WsImport2」です。

以下は、wsimportタスクのプロパティについての説明です。

[表 5.11] wsimportタスクのプロパティ

プロパティ	説明
wsdl	Javaソース・ファイルの作成に使用されるWSDLのURLまたは絶対パスを設定します（必須入力項目。文字列型）
destDir	作成されたJavaファイルが格納されるディレクトリーの絶対パスを設定します（文字列型）
sourcedestdir	作成されたソース・ファイルが格納される位置を設定します。このプロパティを指定すると、keepプロパティも自動設定されます（文字列型）
keep	作成されるファイルを保存するか否かを設定します（デフォルト値：false）
verbose	trueを指定した場合、Verboseが出力されます（デフォルト値：false）
binding	外部のJAX-WSまたはJAXBバインディング・ファイルを設定します（文字列型）
extension	ベンダー特定の拡張性を明示する際に使用します。相互運用および移植性が低下する可能性があります（ブーリアン型）
wsdllocation	WSDL URIをこのプロパティに明示すると、そのURI値が作成されるサービス・エンドポイント・インターフェースとサービス・インターフェースの@WebService.wsdlLocation Annotationと@WebServiceClient.wsdlLocation Annotationに設定されます（文字列型）
catalog	TR9401、XCatalog、OASIS XML Catalogの形式のような外部のEntity参照値を設定できます。また、ant xmncatalogタイプを使用できます（文字列型）

プロパティ	説明
package	WSDLのすべての名前空間URIのJavaパッケージ名を設定します(文字列型)
target	ここに指定されたJAXWSスペック・バージョンに従ってコードを作成します(文字列型)
quiet	メッセージを出力するか否かを設定します(デフォルト値 : false)
policy	Webサービス・ポリシーの設定ファイルを読み込みます(文字列型)

例

以下は、build.xmlの作成例と、build.xmlを使用したwsimportの実行例です。

● build.xmlの作成例

[例 5.7] wsimport Antタスクのbuildファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="wsimport" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="wsimport" classname="com.sun.tools.ws.ant.WsImport">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <wsimport wsdl="${src.conf}/AddNumbers.wsdl"
              destDir="${build.classes.dir}"
              sourcedestdir="${build.classes.dir}"
              package="fromwsdl.server" />
  </target>
</project>
```

● 実行例

```
$ jant
Buildfile: build.xml
...

[wsimport] Consider using <depends>/<produces> so that wsimport won't do
unnecessary compilation
[wsimport] parsing WSDL...
[wsimport]
[wsimport]
[wsimport] generating code...
...
```

```
BUILD SUCCESSFUL
Total time: 6 seconds
```

5.5.5. xjc

xjcタスクは、XMLスキーマ・ファイルをJavaプログラミング言語で構成されているJAXB Contentクラスに変換します。

以下は、xjcタスクのプロパティについての説明です。

[表 5.12] xjcタスクのプロパティ

プロパティ	説明
wsdl	コンパイルするスキーマ・ファイルです(必須入力項目。文字列型)
binding	スキーマ・ファイルに適用される外部の追加バインディング・ファイルです(文字列型)
package	このオプションを明示すると、作成されるコードは、このJavaパッケージの下に配置されます。このオプションはコマンド・ラインの-pオプションと同じ効果があります(文字列型)
destdir	作成されたコードは、このディレクトリーの下に配置されます(必須入力項目。文字列型)
readonly	trueを指定した場合、Javaソース・ファイルは読み取り可能なモードで実行されます(デフォルト値 : false)
extension	trueを指定した場合、XJCバインディング・コンパイラは拡張モードで実行されます(デフォルト値 : false)
catalog	TR9401、XCatalog、OASIS XML Catalogのフォーマットのような外部カタログ・ファイルを明示する際に使用します(文字列型)
removeOldOutput	yesを指定した場合、produces要素が指すファイルはXJCバインディング・コンパイラがソース・ファイルを再コンパイルする前にすべて削除されます(文字列型。"yes"/"no")
source	使用するスキーマ・コンパイラのバージョンを明示します(文字列型。1.0/2.0)

例

以下は、build.xmlの作成例と、build.xmlを使用したxjcの実行例です。

● build.xmlの作成例

[例 5.8] xjc Antタスクのbuildファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="xjc" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
```

```

<import file="../../common/common-build.xml" />
<taskdef name="xjc" classname="com.sun.tools.xjc.XJCTask">
    <classpath refid="jeus.libraries.classpath" />
</taskdef>
<target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <xjc schema="${src.conf}/ts.xsd"
        package="com.tmaxsoft"
        destdir="${build.classes.dir}">
        <produces dir="${build.classes.dir}/com/tmaxsoft"
            includes="**/*.java" />
        <classpath refid="jeus.libraries.classpath" />
        <classpath refid="classpath" />
    </xjc>
</target>
</project>

```

● 実行例

```

$ jant
Buildfile: build.xml

...

[xjc] jeus/build/classes/com/tmaxsoft is not found and thus excluded from
the dependency check
[xjc] Compiling file:/jeus/src/conf/ts.xsd
[xjc] Writing output to jeus/build/classes

...

BUILD SUCCESSFUL
Total time: 4 seconds

```

5.5.6. schemagen

schemagenタスクは、Javaクラスに表示されるそれぞれの名前空間に対し、1つのスキーマ・ファイルを作成します。

以下は、schemagenタスクのプロパティについての説明です。

[表 5.13] schemagenタスクのプロパティ

プロパティ	説明
destdir	作成されるスキーマ・ファイルを配置するディレクトリを設定します(文字列型)

プロパティ	説明
classpath	クラス・パスを設定します(文字列型)

例

以下は、build.xmlの作成例と、build.xmlを使用したschemagenの実行例です。

● build.xmlの作成例

[例 5.9] schemagen Antタスクのbuildファイルの例

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="schemagen" default="build" basedir=".">
  <property name="is.app-client.module" value="true" />
  <import file="../../common/common-build.xml" />
  <taskdef name="schemagen" classname="com.sun.tools.jxc.SchemaGenTask">
    <classpath refid="jeus.libraries.classpath" />
  </taskdef>
  <target name="-pre-compile">
    <mkdir dir="${build.classes.dir}" />
    <schemagen destdir="${build.classes.dir}">
      <src path="${src.dir}" />
      <classpath refid="jeus.libraries.classpath" />
      <classpath refid="classpath" />
    </schemagen>
  </target>
</project>
```

● 実行例

```
$ jant
Buildfile: build.xml

...

[schemagen] Generating schema from 2 source files
[schemagen] Note: Writing jeus/schemal.xsd

...

BUILD SUCCESSFUL
Total time: 5 seconds
```


5.6. WebサービスのMavenプラグイン

本節では、JEUSでJAX-WSベースのWebサービス作成とWebサービス・クライアントのために提供するMavenプラグインについて説明します。

5.6.1. wsgen

wsgenプラグインは、サービス・エンドポイント・インターフェースのクラス(そして、任意のJavaで実装したクラス)から、以下のものを作成します。

- ポータブル・アーティファクト
- WebサービスのWSDLファイル(オプション)

参考

wsgenプラグインはJEUS_HOME/lib/client/jeus-ws-maven-plugin.jarで提供されます。wsgenプラグインを使用するためには、以下の作業を実行します。

1. jeus-ws-maven-plugin.jarファイルをローカルのレポジトリにインストールします。
2. JDK 7、JDK 8環境では、Mavenオプションに「-Djava.endorsed.dirs=JEUS_HOME/lib/endorsed」を追加します。

以下は、wsgenプラグインのプロパティについての説明です。

[表 5.14] wsgenプラグインのプロパティ

プロパティ	説明
sei	SEIクラスの名前を設定します(必須入力項目。文字列型)
destDir	作成されたクラス・ファイルが置かれるディレクトリーの絶対パスを設定します (デフォルト値: \${project.build.outputDirectory})
resourceDestDir	genwsdlプロパティと一緒に使用します。WSDLファイルを作成する位置を設定します (デフォルト: \${project.build.directory}/generated-sources/wsdl)
sourceDestDir	作成されるソース・ファイルの位置を設定します (デフォルト: \${project.build.directory}/generated-sources/wsgen)
keep	作成されるファイルを保存するか否かを設定します(デフォルト: true)
verbose	Verboseメッセージを出力するか否かを設定します(デフォルト: false)
genWsdl	WSDLファイルが作成されるか否かを設定します(デフォルト: false)

プロパティ	説明
protocol	genwsdlプロパティと一緒に使用され、wsdl:binding Elementに使用されるプロトコルを設定します。デフォルト値はsoap1.1で、Xsoap1.2のような値が可能です。Xsoap1.2は標準ではなく、extensionプロパティと一緒に使用されることがあります(文字列型)
servicename	genwsdlプロパティと一緒に使用します。作成されるWSDLファイルの特定wsdl:service要素の名前を設定する際に使用します(文字列型)
portname	genwsdlプロパティと一緒に使用します。作成されるWSDLファイルの特定wsdl:portname要素の名前を設定する際に使用します(文字列型)
extension	ベンダー特定の拡張性を設定する際に使用します。相互運用および移植性を低下させることがあります(デフォルト : false)

例

pom.xmlの作成例と、このpom.xmlを通じてwsгенプラグインを実行する例について説明します。

- pom.xmlの作成例

[例 5.10] wsген plugin pomファイルの例

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>jeus.webservices.maven.sample</groupId>
  <artifactId>wsген_sample</artifactId>
  <version>0.0.1</version>
  <packaging>war</packaging>

  <name>${project.artifactId}</name>

  <build>
    <plugins>
      <plugin>
        <groupId>jeus.ws</groupId>
        <artifactId>jeus-ws-maven-plugin</artifactId>
        <version>0.0.1</version>
        <executions>
          <execution>
            <id>wsген_test</id>
            <goals>
              <goal>wsген</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>
```

```

        <configuration>
            <sei>jeus.webservices.sample.EchoService</sei>
            <extension>true</extension>
            <genWsdL>true</genWsdL>
            <keep>true</keep>
            <inlineSchemas>true</inlineSchemas>
        </configuration>
    </execution>
</executions>

<dependencies>
    <dependency>
        <groupId>com.sun</groupId>
        <artifactId>tools</artifactId>
        <version>1.6</version>
        <scope>system</scope>
        <systemPath>${java.home}/../lib/tools.jar</systemPath>
    </dependency>
    <dependency>
        <groupId>com.sun.xml.ws</groupId>
        <artifactId>jaxws-tools</artifactId>
        <version>2.2</version>
        <scope>system</scope>
        <systemPath>${jeus.home}/lib/system/jaxws-tools.jar</systemPath>
    </dependency>
</dependencies>
</plugin>

<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.3.2</version>
    <configuration>
        <source>1.6</source>
        <target>1.6</target>
        <encoding>UTF-8</encoding>
        <compilerArguments>
            <endorseddirs>${jeus.home}/lib/endorsed</endorseddirs>
        </compilerArguments>
    </configuration>
</plugin>
</plugins>
</build>

</project>

```

- 実行例

```
$ mvn -Djeus.home=${JEUS_HOME} process-classes
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building wsgen_sample 0.0.1
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ wsgen_sample ---
[INFO] skip non existing resourceDirectory /jeus/wsgen/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ wsgen_sample ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- jeus-ws-maven-plugin:0.0.1:wsgen (wsgen_test) @ wsgen_sample ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.773s
[INFO] -----
```

5.6.2. wsimport

wsimportプラグインは、WebサービスのWSDLから以下の2つのうち1つを作成します。

- クライアントのWebサービスのスタブのJavaソース・コード
- サーバーのWebサービスのインターフェースのJavaソース・コード

参考

wsimportプラグインはJEUS_HOME/lib/client/jeus-ws-maven-plugin.jarで提供されます。wsimportプラグインを使用するためには、以下の作業が先行される必要があります。

1. jeus-ws-maven-plugin.jarファイルをローカルのレポジトリにインストールします。
2. JDK 7環境では、mavenオプションに「-Djava.endorsed.dirs=JEUS_HOME/lib/endorsed」を追加します。

以下は、wsimportプラグインのプロパティについての説明です。

[表 5.15] wsimportプラグインのプロパティ

プロパティ	説明
wsdlDirectory	Javaソース・ファイルを作成するために使用されるWSDLファイルを含むディレクトリー・パスを設定します (デフォルト値 : <code>\${basedir}/src/wsdl</code>)
wsdlFiles	ローカルのWSDLファイル・リストを設定します (文字列型)
wsdlUrls	外部のWSDL URLリストを設定します (文字列型)
destDir	作成されたJavaファイルが置かれるディレクトリーのパスを設定します (デフォルト値 : <code>\${project.build.outputDirectory}</code>)
sourceDestDir	作成されたソース・ファイルが置かれる位置を設定します。このプロパティを指定すると、keepプロパティが自動設定されます (デフォルト値 : <code>\${project.build.directory}/generated-sources/wsimport</code>)
keep	作成されるファイルを保存するか否かを設定します (デフォルト値 : true)
verbose	trueを指定した場合、Verboseが出力されます (デフォルト値 : false)
bindingDirectory	JAX-WSまたはJAXBバインディング・ファイルを含むディレクトリー・パスを設定します (デフォルト値 : <code>\${basedir}/src/jaxws</code>)
bindingFiles	JAX-WSまたはJAXBバインディング・ファイルのリストを設定します (文字列型)
extension	ベンダー特定の拡張性を設定する際に使用します。相互運用および移植性が低下する可能性があります (デフォルト値 : false)
wsdlLocation	WSDL URIをこのプロパティに明示すると、そのURI値が作成されるサービス・エンドポイント・インターフェースとサービス・インターフェースの@WebService.wsdlLocation Annotationと@WebServiceClient.wsdlLocation Annotationに設定されます
catalog	TR9401、XCatalog、OASIS XML Catalogの形式のような外部のEntity参照値を設定できます。また、Ant xmncatalogタイプを使用できます
packageName	WSDLのすべての名前空間URIのJavaパッケージ名を設定します (文字列型)
target	ここに指定されたJAXWSスペック・バージョンに従ってコードを作成します (文字列型)
quiet	メッセージを出力するか否かを設定します (デフォルト値 : false)

例

以下は、pom.xmlの作成例と、pom.xmlを通じてwsimportプラグインを実行する例についての説明です。

- pom.xmlの作成例

[例 5.11] wsimport plugin pomファイルの例

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>jeus.webservices.maven.sample</groupId>
  <artifactId>wsimport_sample</artifactId>
  <version>0.0.1</version>
  <packaging>war</packaging>

  <name>${project.artifactId}</name>

  <build>
    <plugins>
      <plugin>
        <groupId>jeus.ws</groupId>
        <artifactId>jeus-ws-maven-plugin</artifactId>
        <version>0.0.1</version>
        <executions>
          <execution>
            <id>wsimport_test</id>
            <goals>
              <goal>wsimport</goal>
            </goals>
            <configuration>

<packageName>jeus.webservices.sample.test</packageName>
              <wsdlDirectory>${basedir}/src/wsdl</wsdlDirectory>
              <wsdlFiles>
                <wsdlFile>EchoService.wsdl</wsdlFile>
              </wsdlFiles>
            </configuration>
          </execution>
        </executions>

        <dependencies>
          <dependency>
            <groupId>com.sun</groupId>
            <artifactId>tools</artifactId>
            <version>1.6</version>
            <scope>system</scope>
            <systemPath>${java.home}/../lib/tools.jar</systemPath>
          </dependency>
        </dependencies>
      </plugin>
    </plugins>
  </build>
</project>
```

```

        <groupId>com.sun.xml.ws</groupId>
        <artifactId>jaxws-tools</artifactId>
        <version>2.2</version>
        <scope>system</scope>

<systemPath>${jeus.home}/lib/system/jaxws-tools.jar</systemPath>
    </dependency>
</dependencies>
</plugin>

<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.3.2</version>
    <configuration>
        <source>1.6</source>
        <target>1.6</target>
        <encoding>UTF-8</encoding>
        <compilerArguments>
            <endorseddirs>${jeus.home}/lib/endorsed</endorseddirs>
        </compilerArguments>
    </configuration>
</plugin>
</plugins>
</build>

</project>

```

● 実行例

```

$ mvn -Djeus.home=${JEUS_HOME} generate-sources
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building wsimport_sample 0.0.1
[INFO] -----
[INFO]
[INFO] --- jeus-ws-maven-plugin:0.0.1:wsimport (wsimport_test) @ wsimport_sample ---
[INFO] Processing: file:/jeus/wsimport/src/wsd/EchoService.wsdl
[INFO] jaxws:wsimport args: [-keep, -s,
/jeus/wsimport/target/generated-sources/wsimport,
-Xnocompile, -p, jeus.webservices.sample.test,
file:/jeus/wsimport/src/wsd/EchoService.wsdl]
parsing WSDL...

```

```
Generating code...
```

```
[ INFO] -----  
[ INFO] BUILD SUCCESS  
[ INFO] -----  
[ INFO] Total time: 2.052s  
[ INFO] -----
```

5.7. jeusadminコマンドの実行

Ant Buildファイル(一般的にbuild.xml)を編集してjeusadminコマンドを実行する方法を紹介します。例を通じてAnt Targetを作成する方法について説明します。

例

以下は、Antを使用してserver-infoコマンドを実行する例です。

```
<project name="check-server-state-example" default="check-server-state">  
  <property environment="env"/>  
  <property name="jeus.home" value="${env.JEUS_HOME}"/>  
  <property name="jeus.home.bin" value="${jeus.home}/bin"/>  
  <property name="unix.jeusadmin" value="${jeus.home.bin}/jeusadmin"/>  
  <property name="windows.jeusadmin" value="${jeus.home.bin}/jeusadmin.cmd"/>  
  <property name="jeusadmin.args" value="-u administrator -p jeus -host host1  
-port 9736 -verbose"/>  
  <property name="server.name" value="adminServer"/>  
  <property name="cmd.target" value="server-info -server ${server.name} -state"/>  
  
  <condition property="isWindows">  
    <os family="windows"/>  
  </condition>  
  <condition property="isUnix">  
    <os family="unix"/>  
  </condition>  
  
  <target name="check-server-state">  
    <antcall target="check-server-state-unix"/>  
    <antcall target="check-server-state-windows"/>  
  </target>
```



```

<target name="check-server-state-windows" if="isWindows">
    <echo>${windows.jeusadmin} ${jeusadmin.args} ${cmd.target}</echo>
    <exec executable="${windows.jeusadmin}" osfamily="windows" spawn="false"

        failonerror="true">
            <arg line="${jeusadmin.args}" />
            <arg value="${cmd.target}" />
        </exec>
</target>

<target name="check-server-state-unix" if="isUnix">
    <echo>${unix.jeusadmin} ${jeusadmin.args} ${cmd.target}</echo>
    <exec executable="${unix.jeusadmin}" osfamily="unix" spawn="false"

        failonerror="true">
            <arg line="${jeusadmin.args}" />
            <arg value="${cmd.target}" />
        </exec>
</target>
</project>

```

- jeus.homeは、JEUSがインストールされている位置を指します。上記例では、システム環境変数のJEUS_HOMEに該当位置が保存されていると仮定しています。

参考

JEUSで提供するjantスクリプトを使用してantを実行した場合、JEUS_HOMEを別途指定しなくても構いません。

-
- unix.jeusadminとwindows.jeusadminは、jeusadminツールを実行するためのスクリプトの位置を指します。一般的にjeusadminスクリプトはJEUS_HOME/binの下に位置します。UNIXとWindowsでスクリプト名が異なるため、プロパティを2つ使用しています。
 - jeusadmin.argsは、jeusadminツールを実行する際に使用するパラメータ値です。ユーザー名、パスワード、ホスト・アドレス、ポート番号、詳細出力(verbose)の可否など、必要なオプションを設定します。オプションについての詳細は[「4.2.1.1. jeusadmin」](#)を参照してください。
 - cmd.targetは、実行するコマンドを指定します。例では、server-infoコマンドを使用してサーバー状態を確認しようとしているため、「server-info -server \${server.name} -state」と設定しています。jeusadminで利用できるコマンドの詳細については[Part II, 「コンソール・コマンドとツール」](#)を参照してください。
 - check-server-state-windowsとcheck-server-state-unix targetでは、exec taskを使用してjeusadminを実行します。osfamilyプロパティを使用して、OSの種類に適切なスクリプトが実行されるようにしています。<exec>タグでspawnプロパティをfalseに指定しなければ、Antを通じて実行結果を確認できません。コマンドの実行が失敗した際にAnt buildが失敗した場合、failonerrorプロパティをtrueに設定します。

例をUNIXで実行した結果は以下のとおりです。

```
JEUS_HOME/bin$jant check-server-state
Buildfile: JEUS_HOME/bin/build.xml

check-server-state:

check-server-state-unix:
    [echo] JEUS_HOME/bin/jeusadmin -u administrator -p jeus -host host1 -port
9736 -verbose server-info -server adminServer -state
    [exec] Verbose output is enabled.
    [exec] Attempting to connect to localhost:9736.
    [exec] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
    [exec] RUNNING
    [exec]

check-server-state-windows:

BUILD SUCCESSFUL
Total time: 1 second
JEUS_HOME/bin$
```

5.8. Mavenプラグイン

本節では、JEUS Mavenプラグインについて説明します。JEUS Mavenプラグインを使用すると、Apache Mavenを利用してJEUSサーバーを起動したり、アプリケーションを配布したりする作業が実行できます。

参考

本節では、Apache Mavenに関する基本的な内容は記述していません。ユーザーがすでにApache Mavenについての十分な知識を備えており、Apache Mavenがすでに正常にインストールされていると仮定して説明します。Apache Mavenについては<http://maven.apache.org>を参照してください。

5.8.1. インストール

JEUS Mavenプラグインを使用するためには、プラグインを先にインストールする必要があります。

JEUS Mavenプラグインは以下のパスで検索できます。

```
JEUS_HOME/lib/systemapps/jeus-maven-plugin.jar
```

インストールの手順は以下のとおりです。

1. jeus-maven-plugin.jarを任意のディレクトリーにコピーします。(ディレクトリー名は任意に設定します。)

```
mkdir /home/user1/jeus-maven-plugin
cd /home/user1/jeus-maven-plugin
cp JEUS_HOME/lib/systemapps/jeus-maven-plugin.jar /home1/user/jeus-maven-plugin
```

2. コピー先のディレクトリーに移動した後、jeus-maven-plugin.jar圧縮ファイルに含まれているpom.xmlを解凍します。

```
jar xvf /home/user1/jeus-maven-plugin/jeus-maven-plugin
META-INF/maven/jeus.tool/jeus-maven-plugin/pom.xml
```

3. 下記のコマンドを利用してJEUS Mavenプラグインをユーザーのmaven local repositoryにインストールします。

```
mvn install:install-file
-Dfile=/home/user1/jeus-maven-plugin/jeus-maven-plugin.jar
-DpomFile=META-INF/maven/jeus.tool/jeus-maven-plugin/pom.xml
```

5.8.2. Maven Plugin Goals

JEUS Mavenプラグインで提供するgoalについて説明します。

5.8.2.1. start-das

DAS(start-das)を起動します。内部ではJEUS_HOME/binの配下にあるスクリプトを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:start-das
```

- パラメータ

下のパラメータのうち、user、password、filenameは必須入力項目ではありませんが、user/passwordまたはfilenameのうち1つを必ず設定する必要があります。

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	DASのホスト・アドレスを設定します (必須入力項目)

パラメータ名	プロパティ名	タイプ	説明
port	jeus.server.port	java.lang.Integer	DASのポート番号を設定します(必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します
domain	jeus.domain.name	java.lang.String	DASが属するドメイン名を設定します (必須入力項目)
server	jeus.server.name	java.lang.String	DAS名を設定します (必須入力項目)
filename	jeus.auth.file	java.lang.String	JEUSを認証時に使用するアカウント情報ファイルを設定します。startDomainAdminServerスクリプトで提供するfオプションと同じです
waitingTime	jeus.boot.timeout	java.lang.Integer	JEUSを起動した後、プラグインでは一定時間の間待ちながらサーバーが正常に起動されたかをチェックしてgoalが正常に実行されたか否かを判断します。その際の待機時間を設定する値です(単位: 秒、デフォルト値: 10秒)
cachelogin	jeus.server.start.cachelogin	boolean	起動時に使用した認証情報を保存するか否かを設定します(デフォルト値: false)
force	jeus.server.start.forced	boolean	起動時に、設定されているアプリケーションがすべてRUNNING状態にならなくても、起動を完了させます(デフォルト値: false)
rolling	jeus.server.start.rolling	boolean	ローリング・パッチ機能を使用するか否かを設定します (デフォルト値: false)
verbose	jeus.server.start.verbose	boolean	詳細出力の使用の可否を設定します (デフォルト値: false)
standby	jeus.server.start.standby	boolean	スタンバイ起動の使用の可否を設定します (デフォルト値: false)

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - start-das -->
<plugin>
```

```

<groupId>jeus.tool</groupId>
<artifactId>jeus-maven-plugin</artifactId>
<version>8.0.0.0</version>
<configuration>
  <jeusHome>/home/user1/jeus</jeusHome>
  <host>host1</host>
  <port>9736</port>
  <user>administrator</user>
  <password>jeus</password>
  <domain>domain1</domain>
  <server>adminServer</server>
</configuration>
</plugin>

```

goalを実行したときに出力される結果は以下のとおりです。

```

$ mvn jeus.tool:jeus-maven-plugin:start-das
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:start-das (default-cli) @ maven-sample ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.414s
[INFO] Finished at: Mon Feb 17 16:16:39 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----

```

5.8.2.2. start-ms

MSを起動します。内部ではJEUS_HOME/binの配下にあるスクリプトを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:start-ms
```

- パラメータ

下のパラメータのうち、user、password、filenameは必須入力項目ではありませんが、user/passwordまたはfilenameのうち1つを必ず設定する必要があります。

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	DASのホスト・アドレスを設定します (必須入力項目)
port	jeus.server.port	java.lang.Integer	DASのポート番号を設定します(必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します
domain	jeus.domain.name	java.lang.String	DASが属するドメイン名を設定します (必須入力項目)
server	jeus.server.name	java.lang.String	DAS名を設定します(必須入力項目)
filename	jeus.auth.file	java.lang.String	JEUSを認証時に使用するアカウント情報ファイルを設定します。startDomainAdminServerスクリプトで提供するfオプションと同じです
waitingTime	jeus.boot.timeout	java.lang.Integer	JEUSを起動した後、プラグインでは一定時間の間待ちながらサーバーが正常に起動されたかをチェックしてgoalが正常に実行されたか否かを判断します。その際の待機時間を設定する値です(単位: 秒、デフォルト値: 10秒)
cachelogin	jeus.server.start.cachelogin	boolean	起動時に使用した認証情報を保存するか否かを設定します(デフォルト値: false)
force	jeus.server.start.forced	boolean	起動時に、設定されているアプリケーションがすべてRUNNING状態にならなくても、起動を完了させます(デフォルト値: false)
rolling	jeus.server.start.rolling	boolean	ローリング・パッチ機能を使用するか否かを設定します (デフォルト値: false)
verbose	jeus.server.start.verbose	boolean	詳細出力の使用の可否を設定します(デフォルト値: false)
standby	jeus.server.start.standby	boolean	スタンバイ起動の使用の可否を設定します (デフォルト値: false)

パラメータ名	プロパティ名	タイプ	説明
dasurl	jeus.server.start.dasurl	java.lang.String	MSが属するドメインを管理するDASのURLを設定します(必須入力項目)

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - start-ms -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <domain>domain1</domain>
    <server>server1</server>
    <dasurl>host1:9736</dasurl>
  </configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:start-ms
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:start-ms (default-cli) @ maven-sample ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.161s
[INFO] Finished at: Mon Feb 17 16:36:16 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----
```

5.8.2.3. stop-jeus

接続したサーバーを終了します。内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用してlocal-shutdownコマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:stop-jeus
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です(デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します (必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。argとargsを両方設定した場合、argが優先します。pom.xmlに設定する場合に使用することを推奨します
arg	jeus.jeusadmin.command.arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。argとargsを両方設定した場合、argが優先します

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。


```
<!-- JEUS Maven Plugin - stop-jeus -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
  </configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:stop-jeus
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:stop-jeus (default-cli) @ maven-sample ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [local-shutdown]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] The server [adminServer] has been shut down successfully.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.690s
[INFO] Finished at: Mon Feb 17 17:19:10 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----
```

5.8.2.4. deploy

アプリケーションをデプロイします。内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用して**deploy-application**コマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:deploy
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です(デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します(必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します。 pom.xmlに設定する場合に使用することを推奨します
arg	jeus.jeusadmin.command.arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```

<!-- JEUS Maven Plugin - deploy -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
  
```

```

    <args>
      <param>myWeb</param>
      <param>-servers adminServer</param>
    </args>
  </configuration>
</plugin>

```

goalを実行したときに出力される結果は以下のとおりです。

```

$ mvn jeus.tool:jeus-maven-plugin:deploy
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:deploy (default-cli) @ maven-sample ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [deploy-application myWeb -servers
adminServer ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] deploy the application for the application [myWeb] succeeded.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.116s
[INFO] Finished at: Mon Feb 17 16:50:32 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----

```

5.8.2.5. undeploy

デプロイしたアプリケーションをアンデプロイします。内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用して**undeploy-application**コマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:undeploy
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です (デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します (必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します。 pom.xmlに設定する場合に使用することを推奨します
arg	jeus.admin.command arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - undeploy -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
```

```

        <param>myWeb</param>
    </args>
</configuration>
</plugin>

```

goalを実行したときに出力される結果は以下のとおりです。

```

$ mvn jeus.tool:jeus-maven-plugin:undeploy
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:undeploy (default-cli) @ maven-sample ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [undeploy-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] Undeploying [myWeb] (This may take time due to graceful undeployment)
.....
[INFO] undeploy the application for the application [myWeb] succeeded.
[INFO] successfully undeployed (elapsed = 147ms)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.814s
[INFO] Finished at: Mon Feb 17 17:06:40 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----

```

5.8.2.6. start-app

内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用して**start-application**コマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:start-app
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です(デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します (必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します。 pom.xmlに設定する場合に使用することを推奨します
arg	jeusadmin.command.arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - start-app -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>
```

```
</configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:start-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:start-app (default-cli) @ maven-sample ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [start-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] start the application for the application [myWeb] succeeded.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.678s
[INFO] Finished at: Mon Feb 17 17:14:07 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----
```

5.8.2.7. stop-app

サービス中のアプリケーションを停止します。内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用して**stop-application**コマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:stop-app
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します(必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです(必須入力項目)

パラメータ名	プロパティ名	タイプ	説明
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号 (デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します (必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します。 pom.xmlに設定する場合に使用することを推奨します
arg	jeusadmin.command.arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します

- 使用例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - stop-app -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:stop-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----
```



```

[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:stop-app (default-cli) @ maven-sample ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [stop-application myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] stop the application for the application [myWeb] succeeded.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.635s
[INFO] Finished at: Mon Feb 17 17:16:48 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----

```

5.8.2.8. install-app

アプリケーションをDASにインストールします。内部ではコンソール・ツール(jeusadmin)が提供するワンステップ実行機能を使用して**install-application**コマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:install-app
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です(デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します

パラメータ名	プロパティ名	タイプ	説明
			(必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します。 pom.xmlに設定する場合に使用することを推奨します
arg	jeus.admin.command.arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - install-app -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>/home/user/myWeb.war</param>
      <param>-id myWeb</param>
    </args>
  </configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:install-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:install-app (default-cli) @ maven-sample ---
```

```

[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [install-application
/home/user/myWeb.war -id myWeb ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] Successfully installed the application [myWeb].
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.019s
[INFO] Finished at: Mon Feb 17 17:22:57 KST 2014
[INFO] Final Memory: 2M/77M
[INFO] -----

```

5.8.2.9. uninstall-app

DASにインストールされているアプリケーションを削除します。内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用して**uninstall-application**コマンドを実行します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:uninstall-app
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です(デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します (必須入力項目)

パラメータ名	プロパティ名	タイプ	説明
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します。 pom.xmlに設定する場合に使用することを推奨します
arg	jeusadmin.command arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先します

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下は、pom.xmlに設定した例です。

```
<!-- JEUS Maven Plugin - uninstall-app -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <args>
      <param>myWeb</param>
    </args>
  </configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:uninstall-app
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:uninstall-app (default-cli) @ maven-sample ---
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [uninstall-application myWeb ]
```

```

[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] uninstall the application for the application [myWeb] succeeded. :
Successfully deleted [myWeb].
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.654s
[INFO] Finished at: Mon Feb 17 17:24:42 KST 2014
[INFO] Final Memory: 2M/61M
[INFO] -----

```

5.8.2.10. jeusadmin

指定したjeusadminコマンドを実行します。内部ではコンソール・ツール(jeusadmin)で提供するワンステップ実行機能を使用します。

- 使用方法

```
jeus.tool:jeus-maven-plugin:jeusadmin
```

- パラメータ

パラメータ名	プロパティ名	タイプ	説明
jeusHome	jeus.home	java.lang.String	JEUSがインストールされているパスを設定します (必須入力項目)
host	jeus.server.host	java.lang.String	jeusadminで接続するサーバーのアドレスです (必須入力項目)
port	jeus.server.port	java.lang.Integer	jeusadminで接続するサーバーのポート番号です(デフォルト値: 9736、必須入力項目)
user	jeus.user.name	java.lang.String	JEUSユーザー名を設定します (必須入力項目)
password	jeus.user.password	java.lang.String	JEUSユーザーのパスワードを設定します (必須入力項目)
args	なし	java.util.List	コマンドを実行時に渡す引数値を設定します。 argとargsを両方設定した場合、argが優先しま

パラメータ名	プロパティ名	タイプ	説明
			す。pom.xmlに設定する場合に使用することを推奨します
arg	jeusadmin.command.arguments	java.lang.String	コマンドを実行時に渡す引数値を設定します。argとargsを両方設定した場合、argが優先します
command	jeusadmin.command	java.lang.String	実行するjeusadminコマンド名を設定します (必須入力項目)

- 例

パラメータはpom.xmlに設定するか、またはコマンドラインで実行時に指定できます。

以下はpom.xmlに設定した例であり、実行するコマンドは**server-info -server adminServer -state**です。

```
<!-- JEUS Maven Plugin - jeusadmin -->
<plugin>
  <groupId>jeus.tool</groupId>
  <artifactId>jeus-maven-plugin</artifactId>
  <version>8.0.0.0</version>
  <configuration>
    <jeusHome>/home/user1/jeus</jeusHome>
    <host>host1</host>
    <port>9736</port>
    <user>administrator</user>
    <password>jeus</password>
    <command>server-info</command>
    <args>
      <param>-server adminServer</param>
      <param>-state</param>
    </args>
  </configuration>
</plugin>
```

goalを実行したときに出力される結果は以下のとおりです。

```
$ mvn jeus.tool:jeus-maven-plugin:jeusadmin
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-sample-project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jeus-maven-plugin:8.0.0.0:jeusadmin (default-cli) @ maven-sample ---
```

```
[INFO] jeus.home = /home/user1/jeus
[INFO] jeusadmin.path = /home/user1/jeus/bin/jeusadmin
[INFO] jeusadmin.one-step-execution-command = [server-info -server adminServer
-state ]
[INFO] Attempting to connect to host1:9736.
[INFO] The connection has been established to Domain Administration Server
adminServer in the domain domain1.
[INFO] RUNNING
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.642s
[INFO] Finished at: Mon Feb 17 17:29:13 KST 2014
[INFO] Final Memory: 2M/77M
[INFO] -----
```

