

Hitachi Application Server V10  
コマンドリファレンス (UNIX®用)

3021-3-419-30

## 前書き

### ■ 対象製品

適用 OS : Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)、Red Hat Enterprise Linux 5 (AMD/Intel 64)、Red Hat Enterprise Linux Server 6 (64-bit x86\_64)、Red Hat Enterprise Linux Server 7 (64-bit x86\_64)

P-9W43-5KA1 Hitachi Application Server 10-11

注 このマニュアルで「AIX の場合」と表記している個所については、この製品ではサポートしていません。

適用 OS については、上記以外にもご利用になれる場合があります。詳細は『リリースノート』でご確認ください。

### ■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

### ■ 商標類

HITACHI、Cosminexus、HA モニタ、HiRDB、JP1、uCosminexus、Virtage は、株式会社日立製作所の商標または登録商標です。

AMD は、Advanced Micro Devices, Inc.の商標です。

IBM、AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM、POWER は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM、WebSphere は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Intel は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft および Excel は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Hyper-V は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Visual C++ は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

RC4 は、米国 EMC コーポレーションの米国およびその他の国における商標または登録商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

RSA および BSAFE は、米国 EMC コーポレーションの米国およびその他の国における商標または登録商標です。



UNIX は、The Open Group の米国ならびに他の国における登録商標です。

W3C は、World Wide Web Consortium の商標（多数の国において登録された）です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

本製品は、米国 EMC コーポレーションの RSA BSAFE<sup>®</sup>ソフトウェアを搭載しています。

This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project.

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This product includes software developed by the University of California, Berkeley and its contributors.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc (Bellcore).

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The original software is available from <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

This product includes software developed by Ralf S. Engelschall <[rse@engelschall.com](mailto:rse@engelschall.com)> for use in the mod\_ssl project (<http://www.modssl.org/>).

This product includes software developed by IAIK of Graz University of Technology.

This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (<http://relaxngcc.sf.net/>).

This product includes software developed by Andy Clark.

Java is a registered trademark of Oracle and/or its affiliates.

**HITACHI**  
Inspire the Next

株式会社 日立製作所



## ■ 発行

2017 年 9 月 3021-3-419-30

## ■ 著作権

Copyright (C) 2014, 2017, Hitachi, Ltd.

Copyright (C) 2013, Oracle and/or its affiliates. All rights reserved.

## 変更内容

### 変更内容(3021-3-419-30)

Hitachi Application Server 10-11

追加・変更内容	変更箇所
リリースノートのマニュアル訂正を反映した。	-

単なる誤字・脱字などはお断りなく訂正しました。

## はじめに

### ■ マニュアルの目的

このマニュアルは、Hitachi Application Server のシステム基盤の構築および運用で使用するコマンドについて説明したものです。このマニュアルを利用することで、ユーザーが Hitachi Application Server のシステム基盤に対して詳細な設計や設定ができることを目的とします。

### ■ 対象読者

このマニュアルは次の方を対象読者としています。

- システム構築者

前提知識を次に示します。

- Windows または UNIX の操作に関する知識
- Application Server の構築に関する知識
- Java EE 標準仕様に関する知識
- システム構築で使用する周辺環境（データベース、ネットワーク、ジョブ管理など）に関する知識

### ■ 関連マニュアルの表記

関連マニュアル、およびこのマニュアルで使用している関連マニュアル名の表記を次の表に示します。

Application Server 関連

表記	正式名称	資料番号
『ユーザーズガイド』	『Hitachi Application Server V10 ユーザーズガイド』 (UNIX <sup>®</sup> 用)	3021-3-415
『GUI リファレンス』	『Hitachi Application Server V10 GUI リファレンス』 (UNIX <sup>®</sup> 用)	3021-3-417
『コマンドリファレンス』	『Hitachi Application Server V10 コマンドリファレンス』 (UNIX <sup>®</sup> 用)	3021-3-419
『定義リファレンス』	『Hitachi Application Server V10 定義リファレンス』 (UNIX <sup>®</sup> 用)	3021-3-421
『メッセージリファレンス』	『Hitachi Application Server V10 メッセージリファレンス』	3021-3-422
『API リファレンス』	『Hitachi Application Server V10 API リファレンス』	3021-3-423

## HiRDB 関連

表記	正式名称	資料番号
『HiRDB システム導入・設計ガイド』	『ノンストップデータベース HiRDB Version 9 システム導入・設計ガイド』(UNIX <sup>®</sup> 用)	3000-6-452
	『ノンストップデータベース HiRDB Version 9 システム導入・設計ガイド』(Windows <sup>®</sup> 用)	3020-6-452
『HiRDB メッセージ』	『ノンストップデータベース HiRDB Version 9 メッセージ』	3020-6-458

## ■ 製品名と機能名の表記

このマニュアルでは、製品名と機能名を次のように表記しています。

表記	製品名と機能名	
Application Server	Hitachi Application Server	
Application Server - Base	Hitachi Application Server - Base	
Application Server - Optional License for Java	Hitachi Application Server - Optional License for Java	
Application Server for Developers	Hitachi Application Server for Developers	
APV	IBM Advanced POWER Virtualization	
DAS	Domain Administration Server	
ドメイン管理サーバ		
Developer's Kit for Java	Hitachi Developer's Kit for Java	
Excel	Microsoft <sup>®</sup> Excel	
Firefox	Firefox <sup>®</sup>	
HiRDB	HiRDB Version 9	HiRDB Server Version 9
	HiRDB/Single Server	HiRDB/Single Server Version 9
HWS	Hitachi Web Server	
Web Server		
Internet Explorer	Windows <sup>®</sup> Internet Explorer <sup>®</sup>	
Java EE Server	Hitachi Java EE Server	
JP1/AJS3	JP1 Version 10 JP1/Automatic Job Management System 3 - Agent	
	JP1 Version 10 JP1/Automatic Job Management System 3 - Manager	

表記		製品名と機能名	
		JP1 Version 10 JP1/Automatic Job Management System 3 - View	
JP1/IM		JP1 Version 10 JP1/Integrated Management - Manager	
		JP1 Version 10 JP1/Integrated Management - View	
Microsoft IIS	Microsoft IIS 7.5	Microsoft® Internet Information Services 7.5	
	Microsoft IIS 8.0	Microsoft® Internet Information Services 8.0	
	Microsoft IIS 8.5	Microsoft® Internet Information Services 8.5	
Microsoft Visual C++		Microsoft® Visual C++®	
Oracle	Oracle 11g	Oracle Database 11g	
		Oracle Database 11g R2	
	Oracle 12c	Oracle Database 12c	
SQL Server		Microsoft® SQL Server	
UNIX	AIX	AIX V6.1	
		AIX V7.1	
	Linux	Linux (AMD64 & Intel EM64T)	Red Hat Enterprise Linux® 5 Advanced Platform (AMD/Intel 64)
			Red Hat Enterprise Linux® 5 (AMD/Intel 64)
			Red Hat Enterprise Linux® Server 6 (64-bit x86_64)
			Red Hat Enterprise Linux® Server 7 (64-bit x86_64)
	Virtage		Hitachi Virtage
VMware ESX		VMware vSphere® ESX	
VMware vSphere ESXi		VMware vSphere® ESXi™	
Windows	Windows Server 2008 R2	Microsoft® Windows Server® 2008 R2 Standard 日本語版	
		Microsoft® Windows Server® 2008 R2 Enterprise 日本語版	
		Microsoft® Windows Server® 2008 R2 Datacenter 日本語版	
	Windows Server 2012	Microsoft® Windows Server® 2012 Standard 日本語版	
		Microsoft® Windows Server® 2012 Datacenter 日本語版	
	Windows Server 2012 R2	Microsoft® Windows Server® 2012 R2 Standard 日本語版	
		Microsoft® Windows Server® 2012 R2 Datacenter 日本語版	
	Windows 7	Windows 7 x86	Microsoft® Windows® 7 Professional 日本語版(32ビット版)



表記		製品名と機能名		
		Windows 7 x64	Microsoft® Windows® 7 Enterprise 日本語版(32 ビット版)	
			Microsoft® Windows® 7 Ultimate 日本語版(32 ビット版)	
			Microsoft® Windows® 7 Professional 日本語版(64 ビット版)	
			Microsoft® Windows® 7 Enterprise 日本語版(64 ビット版)	
			Microsoft® Windows® 7 Ultimate 日本語版(64 ビット版)	
	Windows 8	Windows 8 x86	Windows 8 x86	Windows® 8 Pro 日本語版(32 ビット版)
				Windows® 8 Enterprise 日本語版(32 ビット版)
		Windows 8 x64	Windows 8 x64	Windows® 8 Pro 日本語版(64 ビット版)
				Windows® 8 Enterprise 日本語版(64 ビット版)
		Windows 8.1 x86	Windows 8.1 x86	Windows® 8.1 Pro 日本語版(32 ビット版)
				Windows® 8.1 Enterprise 日本語版(32 ビット版)
		Windows 8.1 x64	Windows 8.1 x64	Windows® 8.1 Pro 日本語版(64 ビット版)
				Windows® 8.1 Enterprise 日本語版(64 ビット版)
	Windows 10	Windows 10 x64	Windows 10 x64	Windows® 10 Pro 日本語版(64 ビット版)
Windows® 10 Enterprise 日本語版(64 ビット版)				
Windows Server Failover Cluster		Windows Server® Failover Cluster		
クラス別統計		日立クラス別統計		
パフォーマンストレーサ		Performance Tracer		
パフォーマンストレーサー				

Linux に関しては、バージョンごとに次のように表記することがあります。

表記		OS名
Red Hat Enterprise Linux 5	Red Hat Enterprise Linux 5 Advanced Platform (AMD/Intel 64)	Red Hat Enterprise Linux® 5 Advanced Platform (AMD/Intel 64)
	Red Hat Enterprise Linux 5 (AMD/Intel 64)	Red Hat Enterprise Linux® 5 (AMD/Intel 64)
Red Hat Enterprise Linux Server 6	Red Hat Enterprise Linux Server 6 (64-bit x86_64)	Red Hat Enterprise Linux® Server 6 (64-bit x86_64)
Red Hat Enterprise Linux Server 7	Red Hat Enterprise Linux Server 7 (64-bit x86_64)	Red Hat Enterprise Linux® Server 7 (64-bit x86_64)

## ■ 英略語

このマニュアルで使用している英略語を次に示します。

英略語	英字での表記
ACC	Application Client Container
ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BLOB	Binary Large Object
CA	Certificate Authority
CDI	Contexts and Dependency Injection
CGI	Common Gateway Interface
CMP	Container-Managed Persistence
CMT	Container-Managed Transaction
CopyGC	Copy Garbage Collection
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSR	Certificate Signing Request
CSV	Comma Separated Value
CVS	Concurrent Versions System
DBMS	Database Management System
DCOM	Distributed Component Object Model
DD	Deployment Descriptor
DDE	Dynamic Data Exchange
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DI	Dependency Injection
DLL	Dynamic Link Library
DMZ	Demilitarized Zone

英略語	英字での表記
DN	Distinguished Name
DNS	Domain Name System
DoS	Denial of Service
DSO	Dynamic Shared Object
DTD	Document Type Definition
EAR	Enterprise Archive
ear	
EIS	Enterprise Information System
EJB	Enterprise JavaBeans™
EJB QL	EJB™ Query Language
EL	Expression Language
EUC	Extended UNIX Code
FQDN	Fully Qualified Domain Name
FullGC	Full Garbage Collection
G1GC	Garbage First Garbage Collection
GC	Garbage Collection
GMS	Group Management Service
GMT	Greenwich Mean Time
GUI	Graphical User Interface
HA	High Availability
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Security
I/O	Input/Output
ID	Identifier
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IIOP	Internet Inter-Orb Protocol
IIS	Internet Information Services

英略語	英字での表記
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
J2EE	J2EE™
	Java™ 2 Platform, Enterprise Edition
JAAS	Java™ Authentication and Authorization Service
JACC	Java™ Authorization Service Provider Contract for Containers
JAF	JavaBeans™ Activation Framework Specification
JAR	Java™ Archive
jar	
JASPIC	Java™ Authentication Service Provider Interface for Containers
Java	Java™
Java EE	Java™ Platform, Enterprise Edition
Java EE RI	Java EE Reference Implementation
Java HotSpot Client VM	Java HotSpot™ Client Virtual Machine
Java Platform Debugger Architecture	Java™ Platform Debugger Architecture
JPDA	
Java SE	Java™ Platform, Standard Edition
Java VM	Java™ Virtual Machine
JVM	
JavaMail	JavaMail™
JAX-RPC	Java™ API for XML-based RPC
JAX-RS	Java™ API for RESTful Web Services
JAX-WS	Java™ API for XML-based Web Services
JAXB	Java™ Architecture for XML Binding
JAXP	Java™ API for XML Processing
JAXR	Java™ API for XML Registries

英略語	英字での表記
JCA	J2EE™ Connector Architecture
JDBC	Java™ Database Connectivity
	JDBC™
JDK	Java™ Development Kit
	JDK™
JIS	Japanese Industrial Standards
JMS	Java™ Message Service
JMX	Java™ Management Extensions
JNDI	Java Naming and Directory Interface™
JNI	Java™ Native Interface
JPA	Java™ Persistence API
JSF	JavaServer™ Faces
	JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS
JSON-P	Java™ API for JSON Processing
JSP	JavaServer Pages™
	JSP™
JST	Japan Standard Time
JSTL	JavaServer Pages™ Standard Tag Library
JTA	Java™ Transaction API
JVMPI	Java™ Virtual Machine Profiler Interface
JVMTI	Java™ Virtual Machine Tool Interface
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MIME	Multipurpose Internet Mail Extensions
MPM	Multi-Processing Module
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group

英略語	英字での表記
ORB	Object Request Broker
OS	Operating System
OTS	Object Transaction Service
QNAME	Qualified Name
REST	Representational State Transfer
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RSA	Rivest, Shamir and Adleman
SAAJ	SOAP with Attachments API for Java™
SAX	Simple API for XML
SEI	Service Endpoint Interface
Servlet	Java™ Servlet
SHA	Secure Hash Algorithm
SMAP	Source Map
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSH	Secure Shell
ssh	
SSL	Secure Sockets Layer
StAX	Streaming API for XML
TCP	Transmission Control Protocol
TLD	Tag Library Descriptor
TLS	Transport Layer Security
UCS	Universal multi-octet coded Character Set
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time

英略語	英字での表記
UTF	UCS Transformation Format
UTF-8	8-bit UCS Transformation Format
VM	Virtual Machine
VTL	Velocity Template Language
W3C	World Wide Web Consortium
WAR	Web Archive
war	
WBEM	Web-Based Enterprise Management
WSDL	Web Services Description Language
XML	Extensible Markup Language

## ■ KB (キロバイト) などの単位表記について

1KB (キロバイト)、1MB (メガバイト)、1GB (ギガバイト)、1TB (テラバイト)、1PB (ペタバイト) はそれぞれ  $1,024$  バイト、 $1,024^2$  バイト、 $1,024^3$  バイト、 $1,024^4$  バイト、 $1,024^5$  バイトです。

# 目次

前書き	2
変更内容	5
はじめに	6

<b>1</b>	<b>コマンドリファレンスについて</b>	<b>24</b>
1.1	コマンドの入力方法	25
1.2	コマンドリファレンスの見方	27
<b>2</b>	<b>Java EE サーバで使用するコマンド</b>	<b>29</b>
2.1	Java EE サーバで使用するコマンドの一覧	30
2.2	Java EE サーバで使用するコマンド	31
2.2.1	appclient	31
2.2.2	asadmin	35
2.2.3	imqcmd list dst	40
2.2.4	jspc	41
2.2.5	schemagen	46
2.2.6	wscompile	48
2.2.7	wsdeploy	52
2.2.8	wsgen	54
2.2.9	wsimport	58
2.2.10	xjc	63
2.3	Java EE サーバで使用する asadmin コマンドのサブコマンド一覧	69
2.4	ドメインの管理で使用するコマンド	82
2.4.1	backup-domain	82
2.4.2	change-admin-password	84
2.4.3	create-domain	85
2.4.4	delete-domain	94
2.4.5	list-backups	95
2.4.6	list-domains	97
2.4.7	restart-domain	98
2.4.8	restore-domain	100
2.4.9	start-domain	103
2.4.10	stop-domain	106
2.5	ノードの管理で使用するコマンド	109
2.5.1	create-node-config	109



2.5.2	create-node-ssh	111
2.5.3	delete-node-config	115
2.5.4	delete-node-ssh	116
2.5.5	list-nodes	117
2.5.6	list-nodes-config	119
2.5.7	list-nodes-ssh	120
2.5.8	ping-node-ssh	121
2.5.9	setup-ssh	123
2.5.10	update-node-config	127
2.5.11	update-node-ssh	128
2.6	サーバの管理で使用するコマンド	133
2.6.1	cleanup-prf	133
2.6.2	cleanup-webserver	135
2.6.3	copy-config	137
2.6.4	create-cluster	141
2.6.5	create-instance	148
2.6.6	create-prf	152
2.6.7	create-webserver	156
2.6.8	delete-cluster	159
2.6.9	delete-config	160
2.6.10	delete-instance	161
2.6.11	delete-prf	163
2.6.12	delete-webserver	165
2.6.13	export-sync-bundle	166
2.6.14	get-health	169
2.6.15	import-sync-bundle	170
2.6.16	list-clusters	172
2.6.17	list-instances	174
2.6.18	list-prfs	177
2.6.19	list-webservers	180
2.6.20	start-cluster	183
2.6.21	start-instance	185
2.6.22	start-prf	188
2.6.23	start-servers	189
2.6.24	start-webserver	190
2.6.25	stop-cluster	191
2.6.26	stop-instance	193
2.6.27	stop-prf	195
2.6.28	stop-servers	196

- 2.6.29 stop-webserver 197
- 2.6.30 validate-multicast 199
- 2.7 サーバ間関連設定で使用するコマンド 203
  - 2.7.1 create-relation 203
  - 2.7.2 delete-relation 207
  - 2.7.3 list-relations 208
- 2.8 アプリケーション管理で使用するコマンド 214
  - 2.8.1 create-application-ref 214
  - 2.8.2 delete-application-ref 216
  - 2.8.3 deploy 218
  - 2.8.4 disable 225
  - 2.8.5 enable 227
  - 2.8.6 get-client-stubs 229
  - 2.8.7 list-application-refs 230
  - 2.8.8 list-applications 232
  - 2.8.9 list-containers 235
  - 2.8.10 list-modules 236
  - 2.8.11 list-sub-components 237
  - 2.8.12 list-timers 239
  - 2.8.13 list-web-context-param 240
  - 2.8.14 list-web-env-entry 242
  - 2.8.15 redeploy 244
  - 2.8.16 set-web-context-param 250
  - 2.8.17 set-web-env-entry 253
  - 2.8.18 show-component-status 256
  - 2.8.19 undeploy 258
  - 2.8.20 unset-web-context-param 260
  - 2.8.21 unset-web-env-entry 263
- 2.9 セキュリティー管理で使用するコマンド 265
  - 2.9.1 change-master-password 265
  - 2.9.2 create-auth-realm 267
  - 2.9.3 create-file-user 272
  - 2.9.4 create-jacc-provider 275
  - 2.9.5 create-password-alias 278
  - 2.9.6 delete-auth-realm 279
  - 2.9.7 delete-file-user 281
  - 2.9.8 delete-jacc-provider 282
  - 2.9.9 delete-password-alias 284
  - 2.9.10 disable-secure-admin 285

- 2.9.11 enable-secure-admin 286
- 2.9.12 list-auth-realms 288
- 2.9.13 list-file-groups 289
- 2.9.14 list-file-users 291
- 2.9.15 list-jacc-providers 293
- 2.9.16 list-password-aliases 294
- 2.9.17 list-supported-cipher-suites 295
- 2.9.18 update-file-user 296
- 2.9.19 update-password-alias 298
- 2.10 ログ管理で使用するコマンド 301
  - 2.10.1 collect-local-snapshot 301
  - 2.10.2 collect-snapshot 305
  - 2.10.3 list-log-attributes 309
  - 2.10.4 set-log-attributes 311
  - 2.10.5 set-log-attributes コマンドで使用するパラメーター一覧 313
- 2.11 パラメーター設定で使用するコマンド 320
  - 2.11.1 create-jvm-options 320
  - 2.11.2 create-system-properties 322
  - 2.11.3 delete-jvm-options 324
  - 2.11.4 delete-system-property 326
  - 2.11.5 get 328
  - 2.11.6 list 330
  - 2.11.7 list-configs 331
  - 2.11.8 list-jvm-options 332
  - 2.11.9 list-system-properties 334
  - 2.11.10 set 335
  - 2.11.11 set サブコマンドおよび get サブコマンドで使用するパラメーター一覧 336
- 2.12 システム管理で使用するコマンド 399
  - 2.12.1 add-library 399
  - 2.12.2 create-profiler 401
  - 2.12.3 delete-profiler 403
  - 2.12.4 list-commands 405
  - 2.12.5 list-libraries 406
  - 2.12.6 login 408
  - 2.12.7 remove-library 409
  - 2.12.8 version 411
- 2.13 インターネット接続管理で使用するコマンド 413
  - 2.13.1 create-protocol 413
  - 2.13.2 create-ssl 414

- 2.13.3 delete-protocol 419
- 2.13.4 delete-ssl 421
- 2.13.5 list-network-listeners 422
- 2.13.6 list-protocols 424
- 2.13.7 list-transports 425
- 2.13.8 list-virtual-servers 426
- 2.14 ORB 管理機能で使用するコマンド 429
  - 2.14.1 create-iiop-listener 429
  - 2.14.2 delete-iiop-listener 431
  - 2.14.3 list-iiop-listeners 433
- 2.15 リソース管理で使用するコマンド 435
  - 2.15.1 add-resources 435
  - 2.15.2 create-resource-ref 437
  - 2.15.3 delete-resource-ref 439
  - 2.15.4 list-resource-refs 440
- 2.16 データベースコネクション管理で使用するコマンド 442
  - 2.16.1 create-jdbc-connection-pool 442
  - 2.16.2 create-jdbc-resource 455
  - 2.16.3 delete-jdbc-connection-pool 457
  - 2.16.4 delete-jdbc-resource 459
  - 2.16.5 list-jdbc-connection-pools 460
  - 2.16.6 list-jdbc-resources 461
- 2.17 JavaMail サービス管理で使用するコマンド 463
  - 2.17.1 create-javamail-resource 463
  - 2.17.2 delete-javamail-resource 466
  - 2.17.3 list-javamail-resources 468
- 2.18 JNDI サービス管理で使用するコマンド 470
  - 2.18.1 create-custom-resource 470
  - 2.18.2 create-jndi-resource 472
  - 2.18.3 delete-custom-resource 475
  - 2.18.4 delete-jndi-resource 476
  - 2.18.5 list-custom-resources 478
  - 2.18.6 list-jndi-resources 479
- 2.19 JMS 管理で使用するコマンド 481
  - 2.19.1 create-jms-host 481
  - 2.19.2 create-jms-resource 483
  - 2.19.3 create-jmsdest 488
  - 2.19.4 delete-jms-host 493
  - 2.19.5 delete-jms-resource 495

- 2.19.6 delete-jmsdest 496
- 2.19.7 flush-jmsdest 498
- 2.19.8 list-jms-hosts 500
- 2.19.9 list-jms-resources 501
- 2.19.10 list-jmsdest 503
- 2.20 EIS 接続管理で使用するコマンド 505
  - 2.20.1 create-admin-object 505
  - 2.20.2 create-connector-connection-pool 508
  - 2.20.3 create-connector-resource 514
  - 2.20.4 create-connector-security-map 517
  - 2.20.5 create-connector-work-security-map 519
  - 2.20.6 create-resource-adapter-config 521
  - 2.20.7 delete-admin-object 523
  - 2.20.8 delete-connector-connection-pool 525
  - 2.20.9 delete-connector-resource 526
  - 2.20.10 delete-connector-security-map 527
  - 2.20.11 delete-connector-work-security-map 529
  - 2.20.12 delete-resource-adapter-config 530
  - 2.20.13 flush-connection-pool 531
  - 2.20.14 list-admin-objects 533
  - 2.20.15 list-connector-connection-pools 535
  - 2.20.16 list-connector-resources 536
  - 2.20.17 list-connector-security-maps 537
  - 2.20.18 list-connector-work-security-maps 539
  - 2.20.19 list-resource-adapter-configs 540
  - 2.20.20 ping-connection-pool 541
  - 2.20.21 update-connector-security-map 544
  - 2.20.22 update-connector-work-security-map 546
- 2.21 スレッドプール管理で使用するコマンド 549
  - 2.21.1 create-threadpool 549
  - 2.21.2 delete-threadpool 551
  - 2.21.3 list-threadpools 553
- 2.22 トランザクション管理で使用するコマンド 555
  - 2.22.1 recover-transactions 555
- 2.23 バッチジョブ管理で使用するコマンド 558
  - 2.23.1 list-batch-job-executions 558
  - 2.23.2 list-batch-job-steps 561
  - 2.23.3 list-batch-jobs 563
  - 2.23.4 list-batch-runtime-configuration 566

- 2.23.5 set-batch-runtime-configuration 568
- 2.24 並行リソース管理で使用するコマンド 570
  - 2.24.1 create-context-service 570
  - 2.24.2 create-managed-executor-service 572
  - 2.24.3 create-managed-scheduled-executor-service 576
  - 2.24.4 create-managed-thread-factory 580
  - 2.24.5 delete-context-service 582
  - 2.24.6 delete-managed-executor-service 584
  - 2.24.7 delete-managed-scheduled-executor-service 585
  - 2.24.8 delete-managed-thread-factory 587
  - 2.24.9 list-context-services 588
  - 2.24.10 list-managed-executor-services 590
  - 2.24.11 list-managed-scheduled-executor-services 591
  - 2.24.12 list-managed-thread-factories 592
- 2.25 メッセージセキュリティープロバイダーの管理で使用するコマンド 595
  - 2.25.1 create-message-security-provider 595
  - 2.25.2 delete-message-security-provider 599
  - 2.25.3 list-message-security-providers 601
- 2.26 イベント検知時コマンド実行で使用するコマンド 604
  - 2.26.1 create-event-hook 604
  - 2.26.2 delete-event-hook 607
  - 2.26.3 list-event-hooks 609

### **3 Web サーバで使用するコマンド 611**

- 3.1 Web サーバで使用するコマンドの一覧 612
- 3.2 Web サーバの運用で使用するコマンド 614
  - 3.2.1 htpasswd 614
  - 3.2.2 httpsd 616
  - 3.2.3 httpsdctl 618
  - 3.2.4 hwstraceinfo 620
  - 3.2.5 logresolve 622
- 3.3 SSL による認証、暗号化で使用するコマンド 624
  - 3.3.1 hwscertutil cert -hash 624
  - 3.3.2 hwscertutil cert -outform 625
  - 3.3.3 hwscertutil cert -text 626
  - 3.3.4 hwscertutil req 627
  - 3.3.5 hwscertutil reqgen 628
  - 3.3.6 hwskeygen 629
  - 3.3.7 sslpasswd 631

3.4 一般ユーザーによる運用 633

## 4 Java VM で使用するコマンド 636

4.1 Java VM で使用するコマンドの一覧 637

4.2 システムの運用で使用するコマンド 638

4.2.1 car\_tar\_gz 638

4.2.2 car\_tar\_Z 640

4.2.3 Eheapprof 642

4.2.4 java\_hras 649

4.2.5 javacore 652

4.2.6 javagc 664

4.2.7 javatrace 675

4.2.8 jheapprof 678

4.2.9 jheapprofanalyzer 684

## 5 PRF で使用するコマンド 689

5.1 PRF で使用するコマンドの一覧 690

5.2 PRF デーモンの起動や停止で使用するコマンド 691

5.2.1 cprfgetpid 691

5.2.2 cprfstart 693

5.2.3 cprfstop 699

5.3 トレース情報の出力で使用するコマンド 702

5.3.1 cprfed 702

5.3.2 cprfflush 708

5.3.3 cprflevel 710

## 索引 717

# 1

## コマンドリファレンスについて

Application Server のシステム基盤の構築および運用で使用するコマンドについて、コマンドの書式、機能などについて説明します。



## 1.1 コマンドの入力方法

---

コマンドの入力方法について説明します。

### コマンドの入力形式

コマンドの入力形式を次に示します。

```
コマンド名 [オプション...]
```

各項目について説明します。なお、ここではコマンドプロンプトを「\$」、コマンド名を「cmd」と表記します。

### コマンド名

実行するコマンドのファイル名を指定します。

スペースを含むパスを指定してコマンドを実行する場合、パス全体をダブルクォーテーション (") で囲む必要があります。

- 誤った指定例：\$ /opt/program path/bin/cmd
- 正しい指定例：\$ "/opt/program path/bin/cmd"

注：指定例に記載のパスは実際の製品で使用するパスとは異なります。

### オプションの入力形式

オプションは、ハイフン (-) で始まる文字列です。オプションの入力形式には、オプション引数を指定しない形式と、1つのオプション引数を指定する形式があります。

- オプション引数を指定しない形式

```
$ cmd -オプションフラグ
```

- 1つのオプション引数を指定する形式

```
$ cmd -オプションフラグスペースまたはタブオプション引数
```

(凡例)

オプションフラグ：1文字の半角英数字です。大文字と小文字が区別されます。

オプション引数：オプションフラグに対する引数です。

### オプションの指定規則

- オプションフラグは、1つのハイフン (-) にまとめて指定できません。

誤った指定例：\$ cmd -abc

正しい指定例：`$ cmd -a -b -c`

- オプション引数を必要とするオプションフラグのオプション引数は、省略できません。
- オプションフラグとオプション引数の間には、スペースまたはタブが必要です。

誤った指定例：`$ cmd -afile`

正しい指定例：`$ cmd -a file`

- 同じオプションフラグは、複数指定できません。

誤った指定例：`$ cmd -a 1 -a 2`

- オプション引数にスペースを含む場合、オプション引数全体を””で囲む必要があります。

誤った指定例：`$ cmd -a file 1`

正しい指定例：`$ cmd -a "file 1"`

## 1.2 コマンドリファレンスの見方

---

コマンドリファレンスを読むために必要な情報について説明しています。

### 記述形式

コマンドについて次の形式で説明します。各コマンドの説明では、次の項目のうち必要な項目についてだけ説明しています。また、次の項目以外に、各コマンドの固有情報を記載している場合があります。

#### 書式

コマンドの入力形式を示します。

#### 格納先

コマンドが格納されているパスを示します。

#### 機能

コマンドの機能について説明します。

#### 実行権限

コマンドの実行権限を示します。

#### 前提条件

コマンドを実行する上で前提となる条件を示します。

#### 引数

コマンドの引数およびオプションについて説明します。

オプションでは、そのオプションに対するオプション引数が必要な場合があります。

Java EE サーバで使用するコマンドのデフォルト値は、オプションに必要なオプション引数だけを省略した場合ではなく、オプション自体を指定しなかった場合の値を示します。

#### 出力形式

コマンドの出力形式を示します。

#### 使用例

コマンドの入力例、出力例などを示します。

#### 戻り値

コマンドの戻り値について説明します。

#### 注意事項

コマンドを実行する上での注意事項について説明します。

### 使用している記号

コマンドの文法では、次の表に示す記号を使用して記述しています。

記号	意味
	横に並べられた複数の項目に対する項目間の区切りを示し、「または」を意味します。 (例) A   B A または B を指定することを示します。
{ }	この記号で囲まれている複数の項目のうちから 1 つを選択することを示します。項目が横に並べられ、記号 で区切られている場合は、そのうちの 1 つを選択します。 (例) {A B C} A、B または C のどれかを指定することを示します。
[ ]	この記号で囲まれている項目は省略してもよいことを示します。複数の項目が横に並べて記述されている場合には、すべてを省略するか、記号{ }と同じくどれか 1 つを選択します。 (例 1) [A] 「何も指定しない」か「A を指定する」ことを示します。 (例 2) [B C] 「何も指定しない」か「B または C を指定する」ことを示します。
...	この記号の直前に示す項目を繰り返し、複数個指定できることを示します。 (例) [プロパティ]... プロパティは複数個、繰り返して指定できます。
項目	斜体で表記されている項目は、該当する要素やファイルなどを指定したり、該当する要素が表示されたりすることを示します。 (例 1) <i>プロパティ</i> プロパティを記述します。またはプロパティが表示されます。 (例 2) <i>ファイル名</i> ファイル名を指定します。

## 使用している構文要素

コマンドの説明では、次の表に定義した構文要素を使用して記述しています。

構文要素	定義
英字	A~Z a~z
英小文字	a~z
英大文字	A~Z
数字	0~9
英数字	A~Z a~z 0~9
記号	! " # \$ % & ' ( ) + , _ . / : ; < = > @ [ ] ^ - { } ` タブ スペース

### 注

すべて半角文字を使用します。

# 2

## Java EE サーバで使用するコマンド

この章では、Java EE サーバで使用するコマンドの書式、機能などについて説明します。

## 2.1 Java EE サーバで使用するコマンドの一覧

Java EE サーバで使用するコマンドの一覧を次の表に示します。

### Java EE サーバで使用するコマンド

コマンド名称	要約	概要説明
appclient	クライアントアプリケーションの起動	アプリケーションクライアントコンテナを起動し、アプリケーション JAR ファイル内に通常パッケージされているクライアントアプリケーションを起動します。
asadmin	管理タスクの実行	Java EE Server の管理タスクを実行します。
imqcmd list dst	物理格納先の一覧表示	メッセージブローカーの物理格納先を一覧表示します。
jspc	JSP ファイルの事前コンパイル	コマンドラインで JSP ファイルを事前にコンパイルできます。
schemagen	スキーマジェネレーターの起動	Java ソースファイルまたはクラスファイルに対してスキーマジェネレーターの処理を実行します。
wscompile	Java インターフェースクラスからの WSDL 生成、WSDL からサービスエンドポイントインターフェースやスタブなどの Java クラスの生成	JAX-RPC のクライアントやサービスで使用するスタブ、タイ、シリアライザーおよび WSDL ファイルを生成します。
wsdeploy	JAX-RPC の Web サービスとしてデプロイできる WAR ファイルの生成	JAX-RPC の Web サービスとしてデプロイできる WAR ファイルを生成します。
wsgen	JAX-WS Web サービスで使用する JAX-WS ポータブルのアーティファクトの生成	JAX-WS Web サービスで使用する JAX-WS ポータブルのアーティファクトを生成します。
wsimport	JAX-WS ポータブルのアーティファクトの生成	JAX-WS ポータブルのアーティファクトを生成します。
xjc	XML Schema から Java ヘバインディングするコンパイラー	コンパイル時に使用するバインドオプションの設定を基に、Java ソースファイルと <code>jaxb.properties</code> ファイルを含んだパッケージの集合を作成します。

#### ❗ 重要

- コマンドの引数には、各コマンドで指定できる値を指定してください。それ以外の値が指定された場合、動作は保証されません。
- コマンドが正常に終了しても、エラーが発生している場合があります。メッセージログやスタックトレースログを確認してください。

## 2.2 Java EE サーバで使用するコマンド

Java EE サーバで使用するコマンドの書式、機能などについて説明します。

### 2.2.1 appclient

クライアントアプリケーションの起動

#### 書式

```
appclient
  [client_application_classfile | -client client_application_jar]
  [-mainclass main_class_name | -name display_name]
  [-xml glassfish-acc.xml_file] [-textauth]
  [-targetserver host[:port] [,host[:port]...]] [-user username]
  [-passwordfile password_file] [application-options]

appclient [jvm-options]
  [-mainclass main_class_name | -name display_name]
  [-xml client_config_xml_file] [-textauth]
  [-targetserver host[:port] [,host[:port]...]] [-user username]
  [-passwordfile password_file]
  class-selector [application-options]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

appclient コマンドは、アプリケーションクライアントコンテナを起動し、アプリケーション JAR ファイル内に通常パッケージされているクライアントアプリケーションを起動します。また、asadmin get-client-stubs コマンドを使用して、クライアント JAR ファイルを取得できます。

アプリケーションクライアントコンテナとは、Java VM 上で動く第 1 層アプリケーションクライアントのプログラムの実行に必要な Java のクラス、ライブラリー、その他ファイルの集合です。アプリケーションクライアントコンテナは、RMI-IIOP を使用してサーバと通信します。

-client オプションまたは-jar オプションを使用して、アプリケーションのデプロイ後に取得したクライアント JAR ファイルをappclient コマンドの起動中に渡すことができます。

クライアント JAR ファイル名は「アプリケーション名 Client.jar」となります。

EAR ファイルにアプリケーションクライアントが複数存在する場合は、-mainclass または-name オプションを使用して、どのクライアントを起動するか指定する必要があります。

アプリケーションクライアントがスタンドアロンモジュールである場合、または EAR ファイルに 1 つしか存在しない場合は、`-mainclass` オプションまたは `-name` オプションを指定しなくても、アプリケーションクライアントコンテナはクライアントを検索できます。

`-mainclass` オプションまたは `-name` オプションに指定した値がクライアントの値と一致しない場合は、アプリケーションクライアントコンテナはそのクライアントを起動を試みますが、警告メッセージが出力されます。その警告メッセージにはクライアントの実際のメインクラスと名称の値が表示されます。

絶対パス位置を指定してください。絶対パスを指定しない場合は、相対パスが使用されます。相対パスはコマンドを実行したディレクトリーからの相対パスとなります。

## 実行権限

スーパーユーザー

## 環境変数

- `appclient` で PRF トレースを出力する場合には、次の環境変数の設定が必要です。
  - `PRFSP00L`  
ドメイン管理サーバによって設定される環境変数 `PRFSP00L` を設定してください。  
環境変数 `PRFSP00L` は、次の形式で設定されます。  
*Java EEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/nodes/ノード名/PRF名*
- `client_application_classfile` オプションには、ほかのユーザークラスによってクラスが決定し、クラスパス `JVM` オプションが使用されていない場合は、`CLASSPATH` が必要です。
- `-xml` オプションには、クライアント設定 XML ファイル (`glassfish-acc.xml`) が必要です。

## ファイル

- アプリケーション名 `Client.jar` を使用してクライアントアプリケーションを起動します。
- `-xml` オプションを使用する場合はクライアント設定 XML ファイル (`glassfish-acc.xml`) が必要です。

## 引数

### *client\_application\_classfile*

クライアントアプリケーションのクラスファイルのパスを指定します。相対パスを指定する場合は、現在のディレクトリーからの相対パスで記述します。

このクラスファイルには、アプリケーションクライアントコンテナが呼び出す `main()` メソッドが含まれている必要があります。

*client\_application\_classfile* を使用し、クラスがほかのユーザークラスによって決まる場合は、クラスパスを設定する必要があります。

型：String

指定できる値を次に示します。



- クライアントアプリケーションのクラスファイルのパス

デフォルト値：なし

`-client client_application_jar`

クライアント JAR ファイルのパスを指定します。

型：String

指定できる値を次に示します。

- クライアント JAR ファイルのパス

デフォルト値：なし

`-mainclass main_class_name`

MANIFEST.MF ファイルの Main-Class エントリーに指定されているメインクライアントアプリケーションの完全クラス名を指定します。複数のクライアントアプリケーションに対して使用できます。

型：String

指定できる値を次に示します。

- メインクライアントアプリケーションの完全クラス名

デフォルト値：

クライアントアプリケーションが複数存在する場合は、クライアント JAR ファイル内で指定されているクラス

`-name display_name`

クライアントアプリケーションの表示名を示します。複数のクライアントアプリケーションに対して使用できます。

型：String

指定できる値を次に示します。

- クライアントアプリケーションの表示名

デフォルト値：

クライアント JAR ファイル内の *application-client.xml* の *display-name* 属性に設定されている値

`-xml glassfish-acc.xml file`

クライアント設定 XML ファイルのパスを示します。ドメイン、インスタンス、および名称 (*glassfish-acc.xml*) をデフォルトのまま使用している場合、このオプションの指定は任意です。変更している場合は、この値の設定は必須です。

型：String

指定できる値を次に示します。

- クライアント設定 XML ファイルのパス

デフォルト値：

*domain-dir/config* ディレクトリ内の *glassfish-acc.xml* ファイル

#### `-textauth`

認証が必要な場合に、テキスト形式の認証を指定します。

#### `-targetserver host:port`

ORB エンドポイントに対してサーバをコンマで区切ったリストを指定します。指定する各サーバにはホストが含まれます。各サーバのポートを指定することもできます。

サーバの指定からポートを省略した場合、デフォルトとして 3700 が使用されます。

型：String

指定できる値を次に示します。

- コンマで区切られたサーバのホスト名およびポート

デフォルト値：なし

#### `-user username`

EJB コンポーネントなどのアプリケーションで保護されている特定のコンポーネントへのアクセス権限があるユーザーを指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：なし

#### `-passwordfile password_file`

PASSWORD=*appclient-password* の形式でアプリケーションクライアントのパスワードを含むファイル名をフルパスで指定します。

セキュリティ上の理由から、環境変数として指定したパスワードは `appclient` コマンドでは読み取りません。

型：String

指定できる値を次に示します。

- パスワードファイルのパス

デフォルト値：なし

#### *jvm-options*

クライアントアプリケーションに JVM オプションを設定できます。 `-client` と `-jar` を除く有効な Java コマンドオプションを指定できます。JVM オプションとほかの `appclient` コマンドのオプションを *class-selector* の前に記述すれば、この 2 種類のオプションを組み合わせ使用できます。

#### *class-selector*

クライアントアプリケーションのクラスに使用する次のどちらかのクラスセレクターを指定します。

型：String

指定できる値を次に示します。

- `-jar jar ファイル名`

クライアント JAR ファイルの名称を指定します。クライアント JAR ファイルは `deploy` サブコマンドによってデプロイ時に作成されます。この値を指定すると、クライアント JAR ファイルのマニフェスト内の `Class-Path` の設定が適用され、`-classpath` オプションの設定は無視されます。

- **クラス名**

アプリケーションクライアントのメインクラスの完全修飾名を指定します。アプリケーションクライアントコンテナはこのクラスのメインメソッドを呼び出し、クライアントを起動します。

デフォルト値：なし

#### *application-options*

クライアントアプリケーションの引数を指定します。

型：String

指定できる値を次に示します。

- **クライアントアプリケーションの引数**

デフォルト値：なし

## 使用例

次の例では、`myclientapp.jar` という名称のクライアントアプリケーション JAR ファイルを起動します。

`glassfish-acc.xml` はクライアント設定 XML ファイルの名称です。`scott` と `sample` はアプリケーションへ渡される引数です。`glassfish-acc.xml` と `myclientapp.jar` がカレントディレクトリーにない場合は、絶対パスでファイルの場所を指定します。絶対パスを指定しない場合は、相対パスが使われます。相対パスは、コマンドを実行しているディレクトリーからの相対パスとなります。

```
appclient -xml glassfish-acc.xml -jar myclientapp.jar scott sample
```

## 2.2.2 asadmin

管理タスクの実行

### 書式

```
asadmin [--host host] [--port port] [--user admin-user]  
  [--passwordfile filename] [--terse={true|false}]  
  [--secure={false|true}] [--echo={true|false}]  
  [--interactive={true|false}] [--detach={true|false}]  
  [--help|subcommand [options] [operands]]
```

### 格納先

*Application Server* インストールディレクトリー/`javaee/glassfish/bin`

## 機能

asadmin ユーティリティーコマンドを使用して、Java EE Server の管理タスクを実行します。

asadmin ユーティリティーコマンドには次のオプションがあります：

- asadmin オプション

asadmin ユーティリティーコマンドの動作を制御しますが、サブコマンドの動作は制御しません。asadmin オプションはサブコマンドの前後に指定できますが、サブコマンドのあとにasadmin オプションを指定することは推奨しません。また、asadmin オプションはサブコマンドの前または後ろのどちらか一方に統一して指定しなければなりません。サブコマンドの前後ばらばらにasadmin オプションを指定すると、エラーが発生します。

- サブコマンドオプション

サブコマンドの動作を制御しますが、asadmin ユーティリティーコマンドの動作は制御しません。サブコマンドオプションはサブコマンドの後ろに指定します。

asadmin オプションと一部のサブコマンドオプションには長形式と短形式があります。

- 長形式の場合

2つのハイフン (--) の後にオプション文字列が続きます。

- 短形式の場合

1つのハイフン (-) の後に単一文字が続きます。

Boolean オプション（機能の有効・無効を切り替えるオプション）を除く、ほとんどのオプションでは引数値が必要となります。

オペランドにはサブコマンドが処理する項目を指定します。オペランドはサブコマンドオプションの引数値に続いて指定し、スペース、タブ、または2つのハイフン (--) で区切ります。asadmin ユーティリティーはサブコマンドオプションやその値に続く文字列をオペランドとして処理します。

asadmin サブコマンドには、サーバ間の通信またはコマンド処理の時間に対して、タイムアウトが設定されています。すべてのサブコマンドに共通のタイムアウトとしては「リードタイムアウト」および「リクエストタイムアウト」があります。

asadmin プロセスに適用するリードタイムアウトの値は、環境変数 AS\_ADMIN\_READTIMEOUT に指定できます。リクエストタイムアウトの値は、asadmin ユーティリティーのset サブコマンドおよびget サブコマンドの次のプロパティーで設定および確認できます。

- configs.config.name.network-config.protocols.protocol.admin-listener.http.request-timeout-seconds
- configs.config.name.network-config.protocols.protocol.sec-admin-listener.http.request-timeout-seconds

コロン (:) を区切り文字として使用するオプションでは、シングルクォーテーション (') やダブルクォーテーション (") の前に円記号付きのエスケープ文字 (\) が必要です。

asadmin ユーティリティでは、コロン (:) を区切り文字として使用するオプションがあります。コロン (:) が Java VM のプロパティまたはオプションの一部である場合は、円記号付きのエスケープ文字 (\) が必要です。円記号付きのエスケープ文字 (\) の代わりに、ダブルクォーテーション (") やシングルクォーテーション (') を使用できます。これら 2 種類の文字での円記号 (\) の扱いを次に示します。

- ダブルクォーテーション (") で囲まれている場合、円記号 (\) は特殊文字として扱われます。
- シングルクォーテーション (') で囲まれている場合、円記号 (\) は特殊文字として扱われません。

## 実行権限

スーパーユーザー

## 前提条件

ローカルモードの場合、サブコマンドを実行してインストールディレクトリーやドメインディレクトリーにアクセスするには、ユーザーはそのドメインのホストマシンにログインする必要があります。

## 環境変数

次に示す環境変数を使って、asadmin オプションのデフォルト値を変更できます。

- AS\_ADMIN\_TERSE
- AS\_ADMIN\_ECHO
- AS\_ADMIN\_INTERACTIVE
- AS\_ADMIN\_HOST
- AS\_ADMIN\_PORT
- AS\_ADMIN\_SECURE
- AS\_ADMIN\_USER
- AS\_ADMIN\_PASSWORDFILE
- AS\_ADMIN\_HELP
- AS\_ADMIN\_READTIMEOUT

## ファイル

--passwordfile オプションにはパスワードエントリーを含んだファイルが必要です。

## 引数

`--help | -?`

asadmin ユーティリティのヘルプテキストを表示します。

`--host host | -H host`

ドメイン管理サーバが起動しているマシンの名称を指定します。

型：String

指定できる値を次に示します。

- ドメイン管理サーバが起動しているマシン名

デフォルト値：localhost

`--port port | -p port`

管理用 HTTP ポートまたは HTTPS ポートを指定します。

型：String

指定できる値を次に示します。

- 管理用 HTTP ポートまたは HTTPS ポート

デフォルト値：4848

`--user admin-user | -u admin-user`

ドメイン管理サーバの権限を持つ管理ユーザーのユーザー名を示します。login サブコマンドを使用してドメイン認証が完了している場合は、ドメイン上の後続のオペレーションに対して `--user` オプションを指定する必要はありません。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：なし

`--passwordfile filename | -W filename`

特定の形式のパスワードエントリーを含んでいるファイルの名称をフルパスで指定します。これらのパスワードエントリーは平文でパスワードファイルに格納されています。セキュリティに関する追加設定をする場合は、`create-password-alias` サブコマンドを使用して、リモートサブコマンドが使用するパスワードのエイリアスを作成します。エイリアス作成済みのパスワードは暗号化されて格納されます。パスワードに対しエイリアスが存在する場合は、次に示すように、そのパスワードエントリーにエイリアスを指定します。

```
AS_ADMIN_password-name=${ALIAS=password-alias-name}
```

型：String

指定できる値を次に示します。

- パスワードファイル名のフルパス

デフォルト値：なし

`--terse={true|false} | -t={true|false}`

`true` を設定した場合、簡潔で、スクリプトでの利用に最適化した形式のデータを出力します。この形式で出力したデータは、ユーザーが参照するには適していません。通常、説明文と詳細なステータスメッセージはこの出力データから省略されます。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--secure={true|false} | -s={true|false}`

`true` を設定した場合、ドメイン管理サーバとの通信で SSL/TLS を使用します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--echo={true|false} | -e={true|false}`

`true` を設定した場合、標準出力時にコマンドライン文を出力します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--interactive={true|false} | -I={true|false}`

必要なオプションだけをプロンプト表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：

- `asadmin` ユーティリティーをコンソール画面から起動している場合：`true`
- `asadmin` ユーティリティーをスクリプト内から起動している場合：`false`

`--detach={true|false}`

`true` を設定した場合、指定した `asadmin` サブコマンドをバックグラウンドでデタッチおよび実行します。この処理はデタッチモードで行われます。`--detach` オプションは長時間実行のサブコマンドを使用する場合に活用できます。また、1つのコンソールまたはスクリプトから複数の独立したサブコマンドを実行できます。`--detach` オプションはサブコマンドの前に指定します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

## 使用例

次の例では、ローカルホストにデプロイされているすべてのアプリケーションを表示します。

```
asadmin list-applications
```

## 関連項目

- [2.11.11 set サブコマンドおよび get サブコマンドで使用するパラメーター一覧](#)

## 2.2.3 imqcmd list dst

物理格納先の一覧表示

### 形式

```
imqcmd list dst -b hostName:portNumber  
[-t destType][-tmp]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/mq/bin*

### 機能

`imqcmd list dst` サブコマンドは、指定した種類の物理格納先を一覧表示します。

### 引数

`-b hostName:portNumber`

メッセージブローカーのホスト名とポート番号を指定します。



リテラル表記の IPv4 または IPv6 アドレスをホスト名として利用できます。リテラル表記の IPv6 アドレスを利用する場合、その形式は RFC2732 に準拠している必要があります。

型：String

指定できる値を次に示します。

- コロンで区切られたホスト名およびポート番号

デフォルト値：localhost:7676

#### -t *destType*

物理格納先の種類を指定します。

型：String

指定できる値を次に示します。

- t  
トピックの物理格納先を指定します。
- q  
キューの物理格納先を指定します。

デフォルト値：なし

#### -tmp

一時物理格納先も一覧表示します。

## 使用例

次の例では、ホスト名：myHost かつポート番号：4545 で稼働しているメッセージブローカーのすべての物理格納先を一覧表示します。

```
imqcmd list dst -b myHost:4545
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.2.4 jspc

JSP ファイルの事前コンパイル

## 書式

```
jspc OPTION [--] jsp_files
```

*OPTION*:

```
[-webapp dir]  
[-help]  
[-v]  
[-d dir]  
[-l]  
[-s]  
[-p name]  
[-c name]  
[-mapped]  
[-die[#]]  
[-uribase dir]  
[-uriroot dir]  
[-compile]  
[-genclass]  
[-webinc file]  
[-webxml file]  
[-classpath path]  
[-xpoweredBy]  
[-trimSpaces]  
[-smap]  
[-dumpsmap]  
[-validate]  
[-compilerSourceVM jdk_release]  
[-compilerTargetVM jdk_release]  
[-ignoreJspFragmentErrors]  
[-disablePooling]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

jspc コマンドは、コマンドラインで JSP ファイルを事前にコンパイルできます。

JSP ファイルは実行時に自動的にコンパイルされます。

## 実行権限

スーパーユーザー

## ファイル

このツールは、.jsp 形式を入力にして、サーブレットを生成します。

## 引数

### *jsp files*

JSP ページとして解析するファイルを指定します。

型：String

指定できる値を次に示します。

- *JSP ページとして解析するファイル名*

デフォルト値：なし

### *-webapp dir*

Web アプリケーションが含まれるディレクトリーを指定します。指定したディレクトリー内の JSP ページを再帰的に処理します。

型：String

指定できる値を次に示します。

- *Web アプリケーションが含まれるディレクトリー*

デフォルト値：なし

### *-help*

このコマンドのヘルプテキストを表示します。

### *-v*

詳細情報を表示します。

### *-d dir*

出力ディレクトリーを指定します。

型：String

指定できる値を次に示します。

- *出力ディレクトリー*

デフォルト値：-Djava.io.tmpdir

### *-l*

処理に失敗した JSP ページの名称を出力します。

### *-s*

処理に成功した JSP ページの名称を出力します。

### *-p name*

対象パッケージの名称を指定します。

型：String

指定できる値を次に示します。

- *パッケージ名*

デフォルト値：org.apache.jsp

**-c *name***

対象クラスの名称を指定します。最初の JSP ページにだけ適用されます。

型：String

指定できる値を次に示します。

- 対象クラスの名称

デフォルト値：なし

**-mapped**

JSP 中の 1 行ごとに `write()` コールを生成します。

**-die[#]**

致命的なエラー発生時にエラーリターンコード (#) を生成します。

デフォルト値：1

**-uribase *dir***

コンパイル時に相対 URI 参照の基準となる URI (ディレクトリー) を示します。

型：String

指定できる値を次に示します。

- ディレクトリー名称

デフォルト値：/

**-uriroot *dir***

Web アプリケーションを含んでいるディレクトリーを指定します。指定したディレクトリー内の JSP ページを再帰的に処理します。

型：String

指定できる値を次に示します。

- ディレクトリー名

デフォルト値：なし

**-compile**

生成したサーブレットをコンパイルします。

**-genclass**

生成したサーブレットをコンパイルします。

**-webinc *file***

ファイル内に部分的なサーブレットマッピングを作成します。

型：String

指定できる値を次に示します。

- パスを含むファイル名

デフォルト値：なし

**-webxml *file***

ファイル内に完全なweb.xmlを作成します。

型：String

指定できる値を次に示します。

- Web アプリケーション名

デフォルト値：なし

**-classpath *path***

java.class.path システムプロパティにクラスパスを追加します。

型：String

指定できる値を次に示します。

- クラスパス

デフォルト値：なし

**-xpoweredBy**

X-Powered-By レスponsヘッダーを追加します。

**-trimSpaces**

アクションや指示子の間のテンプレートテキストからスペースを削除します。

**-smap**

JSR45 デバッグの SMAP 情報を生成します。

**-dumpsmap**

JSR45 デバッグの SMAP 情報をファイル内にダンプします。

**-validate**

スキーマや DTD を使用して、tld ファイルおよびweb.xml ファイルの妥当性をチェックします。

**-compilerSourceVM *jdk\_release***

指定した JDK リリースとソース互換性を持たせます。

型：float

指定できる値を次に示します。

- 1.5 または5
- 1.6 または6
- 1.7 または7
- 1.8 または8

デフォルト値：1.5

**-compilerTargetVM *jdk\_release***

指定した VM バージョンのクラスファイルを生成します。

型：float

指定できる値を次に示します。

- 1.5 または5
- 1.6 または6
- 1.7 または7
- 1.8 または8

デフォルト値：1.5

`-ignoreJspFragmentErrors`

JSP フラグメントのコンパイルエラーを無視します。

`-disablePooling`

カスタムタグプーリング機能を無効にします。

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.2.5 schemagen

スキーマジェネレーターの起動

### 書式

```
schemagen [OPTION]... <java files>
```

*OPTION:*

```
[-d path]  
[-cp path]  
[-classpath path]  
[-episode file]  
[-version]  
[-help]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`schemagen` コマンドは、Java ソースファイルまたはクラスファイルに対してスキーマジェネレーターの処理を実行します。

現在のスキーマジェネレーターは Java クラスで参照されている名称空間ごとにスキーマ定義ファイルを作成します。

## 実行権限

スーパーユーザー

## 環境変数

Java ソースおよびクラスファイルがほかのクラスを参照する場合、システムのCLASSPATH 環境変数でそれらのクラスにアクセスできるか、`-classpath` オプションまたは`-cp` オプションを使用してそれらのクラスを指定する必要があります。それ以外の場合、スキーマの作成中にエラーが発生します。

## ファイル

Java ソースファイルまたはクラスファイルをこのコマンドで指定します。

スキーマ定義ファイルはこのコマンドによって出力されます。

## 引数

`-d path`

プロセッサおよび `javac` コマンドで生成したクラスファイルの出力先ディレクトリーを指定します。

型：String

指定できる値を次に示します。

- 生成されたクラスファイルの出力先ディレクトリー

デフォルト値：なし

`-cp path | -classpath path`

ユーザー指定ファイルの場所を指定します。

型：String

指定できる値を次に示します。

- ユーザー指定ファイルのパス

デフォルト値：なし

`-episode file`

JAXB カスタマイズファイルであるエピソードファイルを生成します。

型：String

指定できる値を次に示します。

- 生成するエピソードファイル名

デフォルト値：なし

-version

バージョン情報を表示します。

-help

このコマンドのヘルプテキストを表示します。

## 使用例

次の例では、Foo.java、Bar.java に対してスキーマジェネレーターの処理を実行します。

```
schemagen Foo.java Bar.java
```

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.2.6 wscompile

Java インターフェースクラスからの WSDL 生成、WSDL からサービスエンドポイントインターフェースやスタブなどの Java クラスの生成

### 書式

```
wscompile [OPTION] configuration_file
```

#### OPTION:

```
[-classpath path]
[-cp path]
[-d directory]
[-define servicename]
[-f:features]
[-features:features]
[-g]
[-gen]
[-gen:client]
[-gen:server]
[-help]
[-httpproxy:host:port]
[-import]
[-keep]
[-model file]
[-nd directory]
[-O]
[-s directory]
[-verbose]
[-version]
```



```
[-mapping file]
[-security file]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`wscpile` コマンドは、JAX-RPC のクライアントやサービスで使用するスタブ、タイ、シリアライザーおよび WSDL ファイルを生成します。このツールは設定ファイルを読み込みます。設定ファイルには WSDL ファイル、モデルファイル、またはコンパイル済みのサービスエンドポイントインターフェースのどれかを指定します。

## 実行権限

スーパーユーザー

## ファイル

このツールは設定ファイルを読み込みます。設定ファイルには WSDL ファイル、モデルファイル、またはコンパイル済みのサービスエンドポイントインターフェースのどれかを指定します。

## 引数

`-classpath path | -cp path`

入力クラスファイルの場所を指定します。

型：String

指定できる値を次に示します。

- 入力クラスファイルのパス

デフォルト値：なし

`-d directory`

生成した出力ファイルの場所を指定します。

型：String

指定できる値を次に示します。

- 出力先ディレクトリーのパス

デフォルト値：なし

`-define servicename`

サービスを定義します。

型：String

指定できる値を次に示します。

- サービス名

デフォルト値：なし

`-f:features` | `-features:features`

指定した機能を有効にします。

型：String

指定できる値を次に示します。

- datahandleronly
- documentliteral
- rpcliteral
- explicitcontext
- infix:<name>
- infix=<name>
- jaxbenumtype
- nodatabinding
- nonencodetypes
- nomultirefs
- norpcstructures
- novalidation
- resolveidref
- searchschema
- serializeinterfaces
- strict
- useonewayoperations
- wsi
- unwrap
- donotoverride
- donotunwrap

デフォルト値：なし

`-g`

デバッグ情報を生成します。

`-gen`

クライアントのアーティファクト（スタブなど）を生成します。

**-gen:client**

クライアントのアーティファクト（スタブなど）を生成します。

**-gen:server**

クライアントのアーティファクト（タイなど）を生成します。

**-help**

ヘルプテキストを表示します。

**-httpproxy:host:port**

HTTP プロキシサーバを指定します。

*host* には、プロキシサーバのホスト名を指定します。

*port* には、プロキシサーバのポート番号を指定します。

ポート番号の型：Integer

ポート番号に指定できる値を次に示します。

- 1～65535

ポート番号のデフォルト値：8080

**-import**

インターフェースと値タイプだけを生成します。

**-keep**

生成したファイルを保持します。

**-model file**

指定したファイルに内部モデルを書き込みます。

型：String

指定できる値を次に示します。

- ファイル名

デフォルト値：なし

**-nd directory**

生成したファイル（クラスファイル以外）の格納場所を指定します。

型：String

指定できる値を次に示します。

- 出力先ディレクトリーのパス

デフォルト値：なし

**-O**

生成したコードを最適化します。

**-s directory**

生成したソースファイルの場所を指定します。

型：String

指定できる値を次に示します。

- 出力先ディレクトリーのパス

デフォルト値：なし

**-verbose**

コンパイラーの動作に関するメッセージを出力します。

**-version**

バージョン情報を出力します。

**-mapping *file***

指定したファイルに JSR-109 マッピングファイルを書き込みます。

型：String

指定できる値を次に示します。

- ファイル名

デフォルト値：なし

**-security *file***

セキュリティー設定のファイルを表示します。

型：String

指定できる値を次に示します。

- ファイル名

デフォルト値：なし

## 使用例

```
wscompile -gen -f:infix:Name -d generated config.xml
```

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.2.7 wsdeploy

JAX-RPC の Web サービスとしてデプロイできる WAR ファイルの生成

## 書式

```
wsdeploy [OPTION] war file
```

*OPTION*:

[-classpath *path*]

[-keep]

[-o *output war file*]

[-tmpdir *directory*]

[-verbose]

[-version]

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

wsdeploy コマンドは WAR ファイルを入力し、JAX-RPC の Web サービスとしてデプロイできる WAR ファイルを生成します。

## 実行権限

スーパーユーザー

## ファイル

wsdeploy コマンドは WAR ファイルを入力し、デプロイできる WAR ファイルを生成します。

## 引数

-classpath *path*

入力クラスファイルのパスを指定します。

型：String

指定できる値を次に示します。

- クラスファイルのパス

デフォルト値：なし

-keep

一時ファイルを保存します。

-o *output war file*

生成した WAR ファイルの場所を指定します。

型：String

指定できる値を次に示します。

- WAR ファイルのパス

デフォルト値：なし

**-tmpdir** *directory*

使用する一時ディレクトリーを指定します。

型：String

指定できる値を次に示します。

- パスを含む一時ディレクトリー名

デフォルト値：なし

**-verbose**

コンパイラーの動作に関するメッセージを出力します。

**-version**

バージョン情報を出力します。

## 使用例

```
wsdeploy -o target.war myapp.war
```

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.2.8 wsgen

JAX-WS Web サービスで使用する JAX-WS ポータブルのアーティファクトの生成

### 書式

```
wsgen [OPTION]... <SEI>
```

*OPTION*:

**[-classpath** *path*]

**[-cp** *path*]

**[-d** *directory*]

**[-extension**]

**[-help**]

**[-keep**]

**[-r** *directory*]

**[-s** *directory*]

**[-encoding** *encoding*]

**[-verbose**]

**[-version**]

**[--fullversion]**

```
[-wsdl[:protocol]]  
[-servicename name]  
[-portname name]  
[-inlineSchemas]  
[-x path]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`wsgen` コマンドは、JAX-WS Web サービスで使用する JAX-WS ポータブルのアーティファクトを生成します。このコマンドは Web サービスのエンドポイントクラスを読み込み、Web サービスのデプロイや呼び出しに必要なすべてのアーティファクトを生成します。

## 実行権限

スーパーユーザー

## ファイル

このコマンドは Web サービスのエンドポイントクラスを読み込み、JAX-WS Web サービスで使用する JAX-WS ポータブルのアーティファクトを生成します

## 引数

`-classpath path | -cp path`

入力クラスファイルの場所を指定します。

型：String

指定できる値を次に示します。

- 入力クラスファイルのパス

デフォルト値：なし

`-d directory`

生成した出力ファイルの格納場所を指定します。

型：String

指定できる値を次に示します。

- 出力先ディレクトリーのパス

デフォルト値：なし

`-extension`

ベンダー拡張（仕様に指定されていない機能）を許可します。拡張を使用すると、アプリケーションの移植性が失われたり、ほかの実装との相互運用ができなくなったりすることがあります。

**-help**

ヘルプテキストを表示します。

**-keep**

生成したファイルを保持します。

**-r *directory***

生成したリソースファイル (WSDL など) の格納場所を指定します。

常に **-wsdl** オプションと組み合わせて使用します。

型: String

指定できる値を次に示します。

- 出力先ディレクトリーのパス

デフォルト値: なし

**-s *directory***

生成したソースファイルの格納場所を指定します。

型: String

指定できる値を次に示します。

- 出力先ディレクトリーのパス

デフォルト値: 作業中のディレクトリー

**-encoding *encoding***

作成したソースの文字エンコード (EUC-JP、UTF-8 など) を設定します。 **-encoding** オプションを指定しない場合、プラットフォームのデフォルトの文字エンコードが使用されます。

型: String

指定できる値を次に示します。

- UTF-8
- EUC-JP

デフォルト値: なし

**-verbose**

コンパイラーの動作に関するメッセージを出力します。

**-version**

バージョン情報を出力します。

**-fullversion**

フルバージョンの情報を出力します。

**-wsdl[:*protocol*]**

WSDL ファイルを生成する場合はこのオプションを必ず指定します。このオプションの指定がない場合は WSDL ファイルを生成しません。



通常の場合、`wsgen` コマンドは開発者がエンドポイントのデプロイ前に WSDL を参照する場合にだけ使用されます。

`protocol` には `wSDL:binding` に使用するプロトコルを指定します。

`protocol` に `Xsoap1.2` を指定する場合は `-extension` オプションの指定も必要です。

型：String

指定できる値を次に示します。

- `soap1.1`
- `Xsoap1.2`

デフォルト値：`soap1.1`

#### `-servicename name`

WSDL 内に生成される特定の `wSDL:service` 名を指定します。

指定例：`-servicename "{http://mynamespace/}MyService"`

常に `-wsdl` オプションと組み合わせて使用します。

型：String

指定できる値を次に示します。

- サービス名

デフォルト値：なし

#### `-portname name`

WSDL 内に生成される特定の `wSDL:port` 名を指定します。

型：String

指定できる値を次に示します。

- ポート名

デフォルト値：なし

例：`-portname "{http://mynamespace/}MyPort"`

常に `-wsdl` オプションと組み合わせて使用します。

#### `-inlineSchemas`

生成した WSDL にスキーマを埋め込みます。

`-wsdl` オプションと組み合わせて使用する必要があります。

#### `-x path`

使用する外部 Web サービスのメタデータ XML 記述子を指定します。

記述子ファイルが複数存在する場合は、各識別子ファイルの前にこのオプションを指定する必要があります。

型：String

指定できる値を次に示します。

- メタデータ XML 記述子ファイルのパス

デフォルト値：なし

## 使用例

次の例では、stock という名称のディレクトリー内に@WebService で注釈付けされた StockService に必要なラッパークラスを生成します。

```
wsgen -d stock -cp myclasspath stock.StockService
```

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.2.9 wsimport

JAX-WS ポータブルのアーティファクトの生成

### 書式

```
wsimport [OPTION]... <WSDL>
```

*OPTION:*

- `[-d directory]`
- `[-b path]`
- `[-B jaxbOption]`
- `[-catalog catalogfile]`
- `[-extension]`
- `[-help]`
- `[-httpproxy:host:port]`
- `[-keep]`
- `[-p pkg]`
- `[-s directory]`
- `[-encoding encoding]`
- `[-verbose]`
- `[-version]`
- `[-fullversion]`
- `[-clientjar jarfile]`
- `[-wsdllocation location]`
- `[-target version]`
- `[-quiet]`
- `[-XadditionalHeaders]`
- `[-Xauthfile authfile]`
- `[-Xdebug]`
- `[-XdisableAuthenticator]`
- `[-Xno-addressing-databinding]`
- `[-Xnocompile]`
- `[-XdisableSSLHostnameVerification]`

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`wsimport` コマンドは次に示す JAX-WS ポータブルのアーティファクトを生成します。

- サービスエンドポイントインターフェース (SEI)
- サービス
- `wsdl:fault` からマッピングする例外クラス (存在する場合)
- 応答 `wsdl:message` から派生する非同期応答 Bean (存在する場合)
- JAXB が生成する値タイプ (スキーマ型からマッピングする Java クラス)

これらのアーティファクトは、WSDL ドキュメント、スキーマ定義ファイル、およびデプロイするエンドポイントの実装とともに WAR ファイル内にパッケージ化されます。

## 実行権限

スーパーユーザー

## ファイル

- このコマンドは WSDL ファイルを入力とします。
- このコマンドは JAX-WS ポータブルのアーティファクトを生成します。
- `-b` オプションを指定している場合は、このコマンドはバインディングファイルまたはスキーマ定義ファイルを入力とします。

## 引数

`-d` *directory*

生成される出力ファイルの格納場所を指定します。

型：String

指定できる値を次に示します。

- ディレクトリー名

デフォルト値：なし

`-b` *path*

外部の JAX-WS バインディングファイルまたは JAXB バインディングファイル、もしくは追加のスキーマ定義ファイルを指定します。複数のファイルを指定する場合は、ファイル (<file>) ごとに `-b` オプションで指定する必要があります。

型：String

指定できる値を次に示します。

- バインディングファイルのパス

デフォルト値：なし

#### **-B *jaxbOption***

このオプションを JAXB スキーマコンパイラーに渡します。

型：String

指定できる値を次に示します。

- *JAXB* オプション名

デフォルト値：なし

#### **-catalog *catalogfile***

外部エンティティー参照を解決するためのカタログファイルのパスを指定します。TR9401、XCatalog、OASIS XML Catalog の形式をサポートしています。

型：String

指定できる値を次に示します。

- カタログファイルのパス

デフォルト値：なし

#### **-extension**

ベンダー拡張（仕様に規定されていない機能）を許可します。拡張を使用すると、アプリケーションの移植性が失われたり、ほかの実装との相互運用ができなくなったりすることがあります。

#### **-help**

ヘルプテキストを表示します。

#### **-httpproxy:*host:port***

HTTP プロキシサーバを指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：*port* のデフォルトは8080

#### **-keep**

作成したソースファイル、またはファイルを保持します。

-s オプションを設定している場合に有効になります。

#### **-p *pkg***

パッケージを指定します。このコマンドラインオプション経由で対象パッケージを指定した場合、パッケージ名に対するwsdl/スキーマカスタムバインディングや、仕様で規定されているデフォルトのパッケージ名アルゴリズムよりも優先されます。

型：String

指定できる値を次に示します。

- 対象パッケージ名

デフォルト値：JAXB 2.1 で規定されたマッピングをパッケージする WSDL/スキーマ対象の名称空間

#### **-s *directory***

生成されるソースコードの格納場所を指定します。

このオプションを指定した場合、**-keep** オプションは有効になります。

型：String

指定できる値を次に示します。

- 出力先ディレクトリーパス

デフォルト値：作業中のディレクトリー

#### **-encoding *encoding***

作成したソースの文字エンコード（EUC-JP、UTF-8 など）を設定します。**-encoding** オプションを指定しない場合、プラットフォームのデフォルトの文字エンコードが使用されます。

型：String

指定できる値を次に示します。

- UTF-8
- EUC-JP

デフォルト値：プラットフォームのデフォルトの文字エンコード

#### **-verbose**

コンパイラーが実行している処理に関するメッセージを出力します。

#### **-version**

バージョン情報を出力します。

#### **-fullversion**

フルバージョンの情報を出力します。

#### **-clientjar *jarfile***

Web サービスを起動するのに必要な WSDL メタデータに加えて、生成したアーティファクトの JAR ファイルを作成します。

型：String

指定できる値を次に示します。

- 生成する JAR ファイル名

デフォルト値：なし

#### **-wsdllocation *location***

@WebServiceClient.wsdlLocation の値を指定します。

型：String

指定できる値を次に示します。

- *@WebServiceClient.wsdlLocation* の値

デフォルト値：*wsdl* 属性に渡される WSDL の URL

#### **-target version**

指定した JAX-WS バージョンの仕様に従ってコードを生成します。例えば、**-target** オプションのパラメーターに 2.0 を指定した場合は、JAX-WS 2.0 仕様でコードを生成します。

型：String

指定できる値を次に示します。

- 最新の JAX-WS バージョンまで

デフォルト値：2.2

#### **-quiet**

wsimport 出力を抑止します。

#### **-XadditionalHeaders**

要求メッセージまたは応答メッセージにバインドされないヘッダーを Java メソッドのパラメーターにマッピングします。

#### **-Xauthfile authfile**

認証情報を伝達するファイルを次の形式で指定します。

`http://ユーザー名:パスワード@example.org/stock?wsdl`

複数の URL パターンと一致させる場合は、アスタリスク (\*) を使用します。

型：String

指定できる値を次に示します。

- 認証情報を伝達するファイルの URL

デフォルト値：`$HOME/.metro/auth`

#### **-Xdebug**

デバッグ情報を出力します。

#### **-XdisableAuthenticator**

JAX-WS RI が使用するオーセンティケーターを無効にします。

このオプションが設定されている場合、**-Xauthfile** オプションは無視されます。

#### **-Xno-addressing-databinding**

W3C EndpointReference 型の Java へのバインディングを有効にします。

#### **-Xnocompile**

生成した Java ファイルをコンパイルしません。

#### **-XdisableSSLHostnameVerification**

wsdl のフェッチ中に SSL のホスト名を検証しません。

## 使用例

```
wsimport -p stockquote http://stockquote.xyz/quote?wsdl
```

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.2.10 xjc

XML Schema から Java へバインディングするコンパイラ

## 書式

```
xjc [OPTION]... <schema file/URL/dir/jar> [-b <binding>...]
```

### OPTION:

- [-nv]
- [-extension]
- [-d *dir*]
- [-p *pkg*]
- [-httpproxy *proxy*]
- [-httpproxyfile *file*]
- [-classpath *arg*]
- [-catalog *file*]
- [-readOnly]
- [-npa]
- [-no-header]
- [-target 2.0|2.1]
- [-encoding *encoding*]
- [-enableIntrospection]
- [-disableXmlSecurity]
- [-contentForWildcard]
- [-xmlschema]
- [-verbose]
- [-quiet]
- [-help]
- [-version]
- [-Xinject-code]
- [-Xlocator]
- [-Xsync-methods]
- [-mark-generated]
- [-episode *file*]
- [-Xpropertyaccessors]

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`xjc` コマンドは、コンパイル時に使用するバインドオプションの設定を基に、Java ソースファイルと `jaxb.properties` ファイルを含んだパッケージの集合を作成します。

パッケージを生成したときに、`jaxb.properties` ファイルはコンパイル済みのソースコードとともに保持されていて、クライアントアプリケーションの実行時に利用できる状態である必要があります。

## 実行権限

スーパーユーザー

## ファイル

- `xjc` コマンドの入力として渡される `xsd` スキーマ定義ファイルが必要です。`xjc` コマンドは Java コンテンツクラスを作成します。
- `xjc` コマンドは Java ソースファイルと `jaxb.properties` ファイルを含んだパッケージの集合を作成します。

## 引数

`-nv`

スキーマの妥当性検査のレベルを変更します。このオプションを指定すると、厳密性の低い妥当性検査が実施されます。

`-extension`

JAXB 仕様で規定されていない機能のベンダー拡張を許可する場合に指定します。

`-d dir`

Java コンテンツクラスを代替出力ディレクトリーに作成します。

このオプションを指定する場合は、代替出力ディレクトリーが事前に作成されている必要があります。XJC スキーマコンパイラーはこのディレクトリーを作成しません。

型：String

指定できる値を次に示します。

- 出力先ディレクトリー名

デフォルト値：なし

`-p pkg`

パッケージを指定します。`-p` オプションでパッケージを指定した場合、パッケージ名に対するカスタムバインディングや、仕様で規定されているデフォルトのパッケージ名アルゴリズムよりも優先されます。



型：String

指定できる値を次に示します。

- パッケージ名

デフォルト値：なし

#### **-httpproxy *proxy***

HTTP/HTTPS プロキシを指定します。

書式は次のとおりです：

`[user[:password]@]proxyHost[:proxyPort]`

型：Integer

指定できる値を次に示します。

- HTTP/HTTPS プロキシ

デフォルト値：なし

#### **-httpproxyfile *file***

HTTP/HTTPS プロキシのファイルパスを指定します。ファイル内の<proxy>パラメーターを使用して、パスワードを保護できます。

書式は次のとおりです：

`[user[:password]@]proxyHost[:proxyPort]`

型：String

指定できる値を次に示します。

- HTTP/HTTPS プロキシのファイルパス

デフォルト値：なし

#### **-classpath *arg***

<jxb:javaType>や<xjc:superClass>のカスタマイズで使われるクライアントアプリケーションのクラスファイルの場所を指定します。

型：String

指定できる値を次に示します。

- クラスパス

デフォルト値：なし

#### **-catalog *file***

外部エンティティ参照を解決するカタログファイルのパスを指定します。TR9401、XCatalog および OASIS XML Catalog の形式をサポートします。

型：String

指定できる値を次に示します。

- カタログファイルのパス

デフォルト値：なし

#### -readOnly

このオプションを指定すると、XJC スキーマコンパイラーは生成された Java ソースを強制的に読み取り専用とします。

#### -npa

\*\*/package-info.java にパッケージレベルの注釈を作成することを抑止します。

このオプションを指定して生成されたコードでは、生成済みのほかのクラスにこれらの注釈が取り込まれません。

#### -no-header

注釈やタイムスタンプを含むファイルヘッダーコメントの生成を抑止します。

このオプションを指定すると、生成されたコードの差分が取りやすくなります。

#### -target 2.0|2.1

JAXB 2.1 または JAXB 2.2 の機能に依存するコードを生成しないようにします。このオプションを指定すると、生成されたコードが JAXB 2.0 ランタイム (JavaSE 6 など) で実行できるようになります。

型: Integer

指定できる値を次に示します。

- JAXB のバージョン番号

デフォルト値: なし

#### -encoding *encoding*

作成したソースファイルの文字エンコードを指定します。

型: String

指定できる値を次に示します。

- JDK がサポートする文字エンコード

デフォルト値: OS のデフォルトの文字エンコード

#### -enableIntrospection

Boolean getters/setters を正しく生成して、Introspection API を有効にします。

#### -disableXmlSecurity

XML ドキュメントの解析時に XML のセキュリティー機能を無効にします。

#### -contentForWildcard

複数の xs:any 派生要素を持つタイプのコンテンツプロパティを生成します。

#### -xmlschema

入力スキーマを W3C XML Schema として処理します。このオプションを指定しない場合も、入力スキーマは W3C XML Schema として処理されます。

#### -verbose

エラー発生時に詳細な情報メッセージを出力したり、スタックトレースを表示したりします。

-quiet

進捗情報や警告などのコンパイラーによる出力を抑止します。

-help

コマンドラインオプションの概要を表示します。

-version

コンパイラーのバージョン情報を表示します。

-Xinject-code

指定した Java コードのフラグメントを生成したコードに埋め込みます。

-Xlocator

このオプションを指定すると、生成した Java コードに、基の XML の SAX の Locator 情報を出力します。

-Xsync-methods

このオプションを指定すると、生成したメソッドのシグニチャーすべてに synchronized キーワードが含まれます。

-mark-generated

このオプションを指定すると、生成したコードに注釈 (@Generated [http://java.sun.com/javaee/5/docs/api/jaxb-2-0-overview?javax/annotation/Generated.html]) を付与します。

-episode *file*

コンパイル時にエピソードファイルを生成します。このエピソードファイルに依存するほかのスキーマ定義ファイルのコンパイル後、エピソードファイルをコンパイル時に生成されたクラスに依存させることができます。作成したエピソードファイルはベンダー拡張を含んだ JAXB カスタマイズファイルとなります。

型：String

指定できる値を次に示します。

- 生成するエピソードファイル名

デフォルト値：なし

-Xpropertyaccessors

生成したクラスの@XmlAccessorType に、XmlAccessType.FIELD の代わりにXmlAccessType PROPERTY を付与します。

*schema file/URL/dir*

コンパイル対象のスキーマ定義ファイルを 1 つ以上指定します。ディレクトリーを指定した場合、xjc はそのディレクトリーを走査し、検出したスキーマ定義ファイルすべてをコンパイルします。

型：String

指定できる値を次に示します。

- スキーマ定義ファイルまたはディレクトリーのパス

デフォルト値：なし

## -b <binding>

処理する外部バインディングファイルを指定します。

バインディングファイルごとに**-b** オプションを指定します。複数のスキーマのカスタマイズを1つのバインディングファイルに含めたり、複数のバインディングファイルに分割したりできます。コマンドラインに指定するスキーマ定義ファイルとバインディングファイルの順番は任意です。

型：String

指定できる値を次に示します。

- 外部バインディングファイル名

デフォルト値：なし

## 使用例

例 1：1つのバインディングファイル (bindings123.xjb) を使用して、schema1.xsd、schema2.xsd、schema3.xsd から Java クラスを生成する場合

```
xjc schema1.xsd schema2.xsd schema3.xsd -b
bindings123.xjb
```

例 2：複数のバインディングファイル (bindings1.xjb、bindings2.xjb、bindings3.xjb) を使用して、schema1.xsd、schema2.xsd、schema3.xsd から Java クラスを生成する場合

```
xjc schema1.xsd schema2.xsd schema3.xsd -b
bindings1.xjb -b bindings2.xjb -b bindings3.xjb
```

## 戻り値

戻り値	説明
0	コマンドの実行に成功しました。
1	コマンドの実行中にエラーが発生しました。

## 2.3 Java EE サーバで使用する asadmin コマンドのサブコマンド一覧

Java EE サーバで使用する asadmin コマンドのサブコマンド一覧を次の表に示します。

### ドメインの管理で使用するコマンド

コマンド名称	要約	概要説明
backup-domain	ドメインのバックアップ	名前付きドメイン内のファイルをバックアップします。
change-admin-password	管理ユーザーパスワードの変更	管理ユーザーのパスワードを変更します。
create-domain	Java EE Server ドメインの作成	Java EE Server ドメインを作成します。
delete-domain	ドメインの削除	指定したドメインを削除します。
list-backups	ドメインバックアップの一覧表示	ドメインバックアップに関する情報を一覧表示します。
list-domains	ドメインディレクトリー内のドメインの一覧表示	指定したドメインディレクトリー内のドメインを一覧表示します。
restart-domain	ドメイン管理サーバの再起動	指定したドメインのドメイン管理サーバを停止した後、再起動します。
restore-domain	バックアップからのファイルの復元	バックアップディレクトリーからドメイン内にファイルを復元します。
start-domain	ドメイン管理サーバの起動	指定したドメインのドメイン管理サーバを起動します。
stop-domain	ドメイン管理サーバの停止	指定したドメインのドメイン管理サーバを停止します。

### ノードの管理で使用するコマンド

コマンド名称	要約	概要説明
create-node-config	リモート通信が有効化されていないノードの作成	リモート通信が有効化されていないノードを作成します。
create-node-ssh	SSH が有効なノードの作成	SSH を介した通信が有効なノードを作成します。
delete-node-config	リモート通信が無効なノードの削除	ドメインからリモート通信が無効なノードを削除します。
delete-node-ssh	SSH 通信が有効なノードの削除	ドメインからの SSH 通信が有効なノードを削除します。
list-nodes	Java EE Server ノードの一覧表示	ドメイン内のすべての Java EE Server ノードを一覧表示します。
list-nodes-config	リモート通信をサポートしないドメイン内の Java EE Server ノードの一覧表示	リモート通信をサポートしない、ドメイン内のすべての Java EE Server ノードを一覧表示します。
list-nodes-ssh	SSH 経由の通信をサポートするドメイン内の Java EE Server ノードの一覧表示	SSH 経由の通信をサポートするドメイン内すべての Java EE Server ノードを一覧表示します。

コマンド名称	要約	概要説明
ping-node-ssh	SSH 通信によるノード使用のテスト	SSH 経由で、ノードが表すホスト上で <code>asadmin</code> ユーティリティコマンドを実行できるかどうかをテストします。
setup-ssh	SSH キーのセットアップ	指定したホストに SSH キーをセットアップします。
update-node-config	ノード構成データの更新	ノードの構成データを更新します。ノードを更新するために、SSH を設定する必要がありません。
update-node-ssh	SSH を使ったノード構成データの更新	ノードの構成データを更新します。ノードを更新するために、ドメイン管理サーバが稼働しているマシンおよびノードが存在するマシンで、SSH を設定する必要があります。

## サーバの管理で使用するコマンド

コマンド名称	要約	概要説明
cleanup-prf	PRF 環境のクリーンアップ	サブコマンドを実行したホスト上に構築されている PRF を停止して、PRF 環境を削除 (クリーンアップ) します。
cleanup-webserver	Web サーバ環境のクリーンアップ	サブコマンドを実行したホスト上に構築されている Web サーバを停止して、Web サーバ環境を削除 (クリーンアップ) します。
copy-config	既存の名前付き構成をコピーして、別の構成を作成	既存の構成をコピーしてドメイン管理サーバに構成を作成します。
create-cluster	Java EE Server クラスターの作成	Java EE Server クラスターを作成します。
create-instance	サーバインスタンスの作成	サーバインスタンスを作成します。
create-prf	PRF の構築	ドメインに PRF を構築します。
create-webserver	Web サーバの構築	ドメインに Web サーバを構築します。
delete-cluster	Java EE Server クラスターの削除	Java EE Server クラスターを削除します。
delete-config	名前付き既存構成の削除	ドメイン管理サーバの構成から既存の名前付き構成を削除します。
delete-instance	サーバインスタンスの削除	サーバインスタンスを削除します。
delete-prf	PRF の削除	指定した PRF をドメインから削除します。
delete-webserver	Web サーバの削除	Web サーバをドメインから削除します。
export-sync-bundle	クラスターまたはスタンドアロンサーバインスタンスの構成データのアーカイブファイルへのエクスポート	クラスターまたはスタンドアロンサーバインスタンスの構成データをアーカイブファイルにエクスポートします。
get-health	クラスターの状態表示	クラスターの状態に関する情報を取得し、表示します。
import-sync-bundle	クラスター化されたインスタンスまたはスタンドアロンサーバインスタンスの構成データのインポート	<code>export-sync-bundle</code> サブコマンドで作成されたアーカイブファイルから、クラスター化されたインスタンスまたはスタンドアロンサーバインスタンスの構成データをインポートします。

コマンド名称	要約	概要説明
list-clusters	ドメイン内の既存クラスターの一覧表示	ドメイン内の既存のクラスターを一覧表示します。
list-instances	サーバインスタンスの一覧表示	ドメイン内のサーバインスタンスを一覧表示します。
list-prfs	PRF の一覧表示	ドメイン内の PRF を一覧表示します。
list-webservers	Web サーバの一覧表示	ドメインに構築された Web サーバを一覧表示します。
start-cluster	クラスターの起動	クラスター内でまだ実行されていないすべてのサーバインスタンスを起動します。
start-instance	サーバインスタンスの起動	サーバインスタンスを起動します。
start-prf	PRF の起動	指定した PRF を起動します。
start-servers	サーバの一括起動	ドメイン内の各種サーバを一括で起動します。
start-webserver	Web サーバの起動	指定した Web サーバを起動します。
stop-cluster	Java EE Server のクラスターの停止	クラスター内で実行されているすべてのサーバインスタンスを停止することによって、そのクラスターを停止します。
stop-instance	実行中のサーバインスタンスの停止	実行中のサーバインスタンスを停止します。
stop-prf	PRF の停止	指定した PRF を停止します。
stop-servers	サーバの一括停止	ドメイン内の各種サーバを一括で停止します。
stop-webserver	Web サーバの停止	指定した Web サーバを停止します。
validate-multicast	マルチキャストトランスポートの検証	クラスターに対してマルチキャストトランスポートが使用できるかどうかを検証します。

## サーバ間関連設定で使用するコマンド

コマンド名称	要約	概要説明
create-relation	サーバ間関連を作成	サーバ間の関連を作成します。
delete-relation	サーバ間関連を削除	サーバ間の関連をドメインから削除します。
list-relations	サーバ間関連を一覧表示	サーバ間の関連を一覧表示します。

## アプリケーション管理で使用するコマンド

コマンド名称	要約	概要説明
create-application-ref	アプリケーション参照の作成	クラスターまたはクラスター構成ではないサーバインスタンスから、すでにデプロイされているアプリケーションへの参照を作成します。
delete-application-ref	アプリケーション参照の削除	クラスターまたはクラスター構成ではないサーバインスタンスからアプリケーション参照を削除します。
deploy	コンポーネントのデプロイ	アプリケーションをサーバにデプロイします。

コマンド名称	要約	概要説明
disable	コンポーネントの無効化	指定したデプロイ済みコンポーネントをただちに無効にします。
enable	コンポーネントの有効化	指定したデプロイ済みコンポーネントを有効にします。
get-client-stubs	クライアントスタブの JAR ファイルの取得	クライアントスタブ JAR ファイルを、サーバマシンからローカルディレクトリーに取得します。
list-application-refs	既存のアプリケーション参照の一覧表示	クラスターまたはクラスター構成ではないサーバインスタンスにあるすべてのアプリケーション参照を一覧表示します。
list-applications	デプロイ済みアプリケーションの一覧表示	デプロイ済みの Java EE アプリケーションと各アプリケーションの種類を一覧表示します。
list-containers	アプリケーションコンテナの一覧表示	アプリケーションコンテナのリストを一覧表示します。
list-modules	Java EE Server モジュールの一覧表示	Java EE Server からアクセス可能なモジュールの一覧を表示します。
list-sub-components	デプロイされたモジュール内またはデプロイされたアプリケーションモジュール内の EJB またはサーブレットのコンポーネントの一覧表示	デプロイされたモジュール内、またはデプロイされたアプリケーションのモジュール内の EJB またはサーブレットのコンポーネントを一覧表示します。
list-timers	サーバインスタンスが所有する永続タイマーの一覧表示	指定したサーバインスタンスまたはサーバインスタンスのクラスターが所有する永続タイマーを一覧表示します。
list-web-context-param	デプロイされた Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの一覧表示	デプロイされた Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターを一覧表示します。
list-web-env-entry	デプロイされた Web アプリケーションまたはモジュールの環境エントリーの一覧表示	デプロイされた Web アプリケーションまたはモジュールの環境エントリーを一覧表示します。
redeploy	コンポーネントのリデプロイ	デプロイ済みまたは既存のコンポーネント（エンタープライズアプリケーション、Web アプリケーション、EJB モジュール、リソースアダプターアーカイブ、またはアプリケーションクライアントのモジュール）をリデプロイします。
set-web-context-param	Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの設定	デプロイ済みの Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターを設定します。
set-web-env-entry	Web アプリケーションまたはモジュールの環境エントリーの設定	デプロイ済みの Web アプリケーションまたはモジュールの環境エントリーを設定します。
show-component-status	デプロイ済みのコンポーネントの状態の表示	デプロイ済みのコンポーネントの状態を取得します。
undeploy	デプロイ済みのコンポーネントの削除	デプロイ済みのコンポーネント（アプリケーションまたはモジュール）をアンインストールして、リポジトリから削除します。

## 2. Java EE サーバで使用するコマンド



コマンド名称	要約	概要説明
unset-web-context-param	Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの設定の解除	デプロイ済みの Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの設定を解除します。
unset-web-env-entry	Web アプリケーションまたはモジュールの環境エントリーの設定の解除	デプロイ済みの Web アプリケーションまたはモジュールの環境エントリーの設定を解除します。

## セキュリティー管理で使用するコマンド

コマンド名称	要約	概要説明
change-master-password	マスターパスワードの変更	マスターパスワードを変更します。
create-auth-realm	名前付き認証レルムの追加	名前付き認証レルムを追加します。
create-file-user	ファイルユーザーの作成	キーファイルのエントリーを作成します。
create-jacc-provider	任意の認証モジュールで使用できる JACC プロバイダーの作成	JSR-115 準拠の JACC プロバイダーを作成します。
create-password-alias	パスワードエイリアスの作成	パスワードに対するエイリアスを作成します。
delete-auth-realm	名前付きの認証レルムの削除	名前付き認証レルムを削除します。
delete-file-user	ファイルユーザーの削除	指定したユーザー名に対するキーファイルのエントリーを削除します。
delete-jacc-provider	JACC プロバイダーの削除	ドメインに定義された JACC プロバイダーを削除します。
delete-password-alias	パスワードエイリアスの削除	パスワードエイリアスを削除します。
disable-secure-admin	secure admin 機能の無効化	secure admin 機能を無効にします。
enable-secure-admin	secure admin 機能の有効化	secure admin 機能を有効にします。
list-auth-realms	認証レルムの一覧表示	認証レルムを一覧表示します。
list-file-groups	ファイルグループの一覧表示	ファイルレルム認証によってサポートされるファイルユーザーとグループを一覧表示します。
list-file-users	ファイルユーザーの一覧表示	ファイルレルム認証でサポートされるファイルユーザーの一覧を表示します。
list-jacc-providers	JACC プロバイダーの一覧表示	ドメインに定義された JACC プロバイダーを一覧表示します。
list-password-aliases	パスワードエイリアスの一覧表示	すべてのパスワードエイリアスを一覧表示します。
list-supported-cipher-suites	Java EE Server で利用できる暗号化方式群の一覧表示	指定した Java EE Server で利用できる暗号化方式群を一覧表示します。
update-file-user	現在のファイルユーザーの更新	キーファイル内の既存のエントリーを更新します。

コマンド名称	要約	概要説明
update-password-alias	パスワードエイリアスの更新	指定したターゲットでパスワードエイリアス ID を更新します。

## ログ管理で使用するコマンド

コマンド名称	要約	概要説明
collect-local-snapshot	ドメイン管理サーバ非稼働時のシステム情報の収集	Application Server を構成する製品のシステム情報を収集します。ドメイン管理サーバが稼働していない状態でも実行できます。
collect-snapshot	ドメイン管理サーバ稼働時のシステム情報の収集	Application Server を構成する製品のシステム情報を収集します。ドメイン管理サーバが稼働しているときだけ実行できます。
list-log-attributes	ログ属性の一覧表示	指定した Java EE Server ドメインまたはドメイン内の対象に現在定義されているログ属性を一覧表示します。
set-log-attributes	1 つ以上のログのロギング属性の設定	1 つ以上のログのロギング属性を設定します。

## パラメーター設定で使用するコマンド

コマンド名称	要約	概要説明
create-jvm-options	Java VM オプションの作成	Java EE Server 起動時に Java VM に渡されるコマンドラインの各オプションを作成します。
create-system-properties	構成のほかの部分から参照できるシステムプロパティの追加	サーバの各所から参照できる各システムプロパティを追加または更新します。
delete-jvm-options	Java VM オプションの削除	Java VM のコマンドラインオプションを削除します。
delete-system-property	システムプロパティ（ドメイン／構成／クラスター／サーバインスタンス）の削除	システムプロパティ（ドメイン、構成、クラスターまたはサーバインスタンス）を削除します。
get	構成属性値の取得	ドット表記名を使用して Java EE Server の構成属性値を取得します。
list	構成属性値の一覧表示	Java EE Server の構成属性値を一覧表示します。
list-configs	ドメイン管理サーバの構成中の、名前付き構成の一覧表示	ドメイン管理サーバの構成中の名前付き構成を一覧表示します。
list-jvm-options	Java VM オプションの一覧表示	Java EE Server 起動時に Java VM に渡されるコマンドラインオプションを一覧表示します。
list-system-properties	ドメイン、構成、クラスターまたはサーバインスタンスのシステムプロパティの一覧表示	システムプロパティ（ドメイン、構成、クラスターまたはサーバインスタンス）を一覧表示します。
set	構成属性値の設定	1 つ以上の構成属性値を変更します。

## システム管理で使用するコマンド

コマンド名称	要約	概要説明
add-library	1 つ以上のライブラリーのアーカイブを Java EE Server に追加	1 つ以上のライブラリーのアーカイブを Java EE Server に追加します。
create-profiler	プロファイラー要素の作成	プロファイラー要素を作成します。
delete-profiler	プロファイラー要素の削除	Java 構成のプロファイラー要素を削除します。
list-commands	使用できるコマンドの一覧表示	asadmin コマンドのサブコマンドを一覧表示します。
list-libraries	ライブラリーのアーカイブの一覧表示	Java EE Server のライブラリーのアーカイブを一覧表示します。
login	ドメインへのログイン	特定のドメインにログインできる状態にして、ドメイン管理を簡略化します。
remove-library	1 つ以上のライブラリー JAR ファイルの Java EE Server からの除去	1 つ以上のライブラリー JAR ファイルを Java EE Server から除去します。
version	Java EE Server のバージョン情報の表示	Java EE Server のバージョン情報を表示します。

## インターネット接続管理で使用するコマンド

コマンド名称	要約	概要説明
create-protocol	新規プロトコルの追加	プロトコルを作成します。
create-ssl	セキュア通信の構成	選択した HTTP リスナー、IIOP リスナーまたは IIOP サービスに SSL 要素を作成して、それらのリスナーまたはサービスでセキュア通信を有効にするように構成します。
delete-protocol	プロトコルの削除	指定したプロトコルを削除します。
delete-ssl	SSL 設定の削除	選択した HTTP リスナー、IIOP リスナーまたは IIOP サービスの SSL 要素を削除します。
list-network-listeners	ネットワークリスナーの一覧表示	既存のネットワークリスナーを一覧表示します。
list-protocols	プロトコルの一覧表示	既存のプロトコルを一覧表示します。
list-transports	トランスポートの一覧表示	既存のトランスポートを一覧表示します。
list-virtual-servers	仮想サーバの一覧表示	既存の仮想サーバを一覧表示します。

## ORB 管理機能で使用するコマンド

コマンド名称	要約	概要説明
create-iiop-listener	IIOP リスナーの追加	IIOP リスナーを作成します。
delete-iiop-listener	IIOP リスナーの削除	指定した IIOP リスナーを削除します。
list-iiop-listeners	IIOP リスナーの一覧表示	既存の IIOP リスナーを一覧表示します。

## リソース管理で使用するコマンド

コマンド名称	要約	概要説明
add-resources	XML ファイルに指定されたリソースの追加	指定した XML ファイル内で指名されているリソースを追加します。
create-resource-ref	リソースへの参照の作成	クラスターまたはクラスター化されていないサーバインスタンスから前に作成されたリソースへの参照を作成します。
delete-resource-ref	リソース参照の削除	クラスターまたはクラスター化されていないサーバインスタンスから、リソース参照を削除します。
list-resource-refs	リソース参照の一覧表示	クラスターまたはクラスター化されていないサーバインスタンス内のリソース参照を一覧表示します。

## データベースコネクション管理で使用するコマンド

コマンド名称	要約	概要説明
create-jdbc-connection-pool	JDBC コネクションプールの登録	指定した JDBC コネクションプール名を持つ JDBC ソフトウェアを新規登録します。
create-jdbc-resource	JDBC リソースの作成	JDBC リソースを新規に作成します。
delete-jdbc-connection-pool	JDBC コネクションプールの削除	JDBC コネクションプールを削除します。
delete-jdbc-resource	JNDI 名を持つ JDBC リソースの削除	JNDI 名を指定した JDBC リソースを削除します。
list-jdbc-connection-pools	JDBC コネクションプールの一覧表示	現在の JDBC コネクションプールを一覧表示します。
list-jdbc-resources	JDBC リソースの一覧表示	既存の JDBC リソースを一覧表示します。

## JavaMail サービス管理で使用するコマンド

コマンド名称	要約	概要説明
create-javamail-resource	JavaMail セッションリソースの作成	JavaMail セッションリソースを作成します。
delete-javamail-resource	JavaMail セッションリソースの削除	指定した JavaMail セッションリソースを削除します。
list-javamail-resources	JavaMail セッションリソースの一覧表示	既存の JavaMail セッションリソースを一覧表示します。

## JNDI サービス管理で使用するコマンド

コマンド名称	要約	概要説明
create-custom-resource	カスタムリソースの作成	カスタムリソースを作成します。

コマンド名称	要約	概要説明
create-jndi-resource	JNDI リソースの登録	JNDI リソースを登録します。
delete-custom-resource	カスタムリソースの削除	カスタムリソースを削除します。
delete-jndi-resource	JNDI リソースの削除	指定した JNDI リソースを削除します。
list-custom-resources	カスタムリソースの取得	カスタムリソースを一覧表示します。
list-jndi-resources	JNDI リソースの一覧表示	既存のすべての JNDI リソースを一覧表示します。

## JMS 管理で使用するコマンド

コマンド名称	要約	概要説明
create-jms-host	JMS ホストの作成	JMS サービス内に JMS ホストを作成します。
create-jms-resource	JMS リソースの作成	JMS コネクションファクトリーリソースまたは JMS Destination リソースを作成します。
create-jmsdest	JMS の物理送信先の作成	JMS の物理送信先を作成します。
delete-jms-host	JMS ホストの削除	JMS ホストを削除します。
delete-jms-resource	JMS リソースの削除	JMS コネクションファクトリーリソースまたは JMS Destination リソースを削除します。
delete-jmsdest	JMS 物理送信先の削除	JMS 物理送信先を削除します。
flush-jmsdest	JMS 送信先内のメッセージの削除	サーバの JMS 構成にある物理送信先からメッセージを削除します。
list-jms-hosts	JMS ホストの一覧表示	JMS サービスの既存の JMS ホストを一覧表示します。
list-jms-resources	JMS リソースの一覧表示	既存の JMS リソース（送信先とコネクションファクトリーリソース）を一覧表示します。
list-jmsdest	JMS 物理送信先の一覧表示	JMS 物理送信先を一覧表示します。

## EIS 接続管理で使用するコマンド

コマンド名称	要約	概要説明
create-admin-object	管理対象オブジェクトの作成	指定した JNDI 名で管理対象オブジェクト、およびリソースアダプターのインターフェース定義を作成します。
create-connector-connection-pool	コネクションプールの追加	エンタープライズ情報システム (EIS) への接続のプールを定義します。
create-connector-resource	コネクタリソースの登録	指定した JNDI 名でコネクタリソースを登録します。
create-connector-security-map	コネクタコネクションプールのセキュリティマップの作成	指定したコネクタコネクションプールのセキュリティマップを作成します。

コマンド名称	要約	概要説明
create-connector-work-security-map	リソースアダプターのワークセキュリティーマップの作成	リソースアダプター EIS プリンシパルまたは EIS ユーザーグループによってサブミットされたワークの呼び出し側識別を、Java EE Server セキュリティドメインの対応するプリンシパルまたはユーザーグループにマップします。
create-resource-adapter-config	リソースアダプターアーカイブ構成情報の作成	リソースアダプターアーカイブ構成情報を作成します。
delete-admin-object	管理対象オブジェクトの削除	指定した JNDI 名の管理対象オブジェクトを削除します。
delete-connector-connection-pool	コネクタークoneksiンプールの削除	指定したコネクタークoneksiンプールを削除します。
delete-connector-resource	コネクタースリソースの削除	指定した JNDI 名のコネクタースリソースを削除します。
delete-connector-security-map	コネクタークoneksiンプールのセキュリティーマップの削除	指定したコネクタークoneksiンプールのセキュリティーマップを削除します。
delete-connector-work-security-map	リソースアダプターのワークセキュリティーマップの削除	指定したリソースアダプターに関連づけられているワークセキュリティーマップを削除します。
delete-resource-adapter-config	リソースアダプターアーカイブ構成情報の削除	リソースアダプターアーカイブの構成情報を削除します。
flush-connection-pool	コネクションプールで確立されたコネクションの再初期化	JDBC コネクションプールまたはコネクタークoneksiンプールを初期状態にリセットします。
list-admin-objects	管理された全オブジェクトの取得	管理された全オブジェクトの一覧を表示します。
list-connector-connection-pools	コネクタークoneksiンプールの一覧表示	作成されたコネクタークoneksiンプールを一覧表示します。
list-connector-resources	コネクタースリソースの一覧表示	すべてのコネクタースリソースを一覧表示します。
list-connector-security-maps	コネクタークoneksiンプールに属するセキュリティーマップの一覧表示	指定されたコネクタークoneksiンプールに属するセキュリティーマップを一覧表示します。
list-connector-work-security-maps	リソースアダプターに属するワークセキュリティーマップの一覧表示	指定されたリソースアダプターに属するワークセキュリティーマップを一覧表示します。
list-resource-adapter-configs	リソースアダプターアーカイブ構成の名称の一覧表示	リソースアダプターアーカイブに対する domain.xml の構成情報を一覧表示します。
ping-connection-pool	JDBC またはコネクタークoneksiンプールの検証	既存の JDBC またはコネクタークoneksiンプールが使用できるかどうかをテストします。
update-connector-security-map	コネクタークoneksiンプールのセキュリティーマップの変更	指定したコネクタークoneksiンプールのセキュリティーマップを変更します。
update-connector-work-security-map	リソースアダプターのワークセキュリティーマップの変更	指定したリソースアダプターのセキュリティーマップを変更します。

## スレッドプール管理で使用するコマンド

コマンド名称	要約	概要説明
create-threadpool	スレッドプールの追加	指定名でスレッドプールを作成します。
delete-threadpool	スレッドプールの削除	指定した ID のスレッドプールを削除します。
list-threadpools	スレッドプールの一覧表示	Java EE Server のスレッドプールを一覧表示します。

## トランザクション管理で使用するコマンド

コマンド名称	要約	概要説明
recover-transactions	実行待ちトランザクションの手動での回復	サーバ障害後にトランザクションを回復します。

## バッチジョブ管理で使用するコマンド

コマンド名称	要約	概要説明
list-batch-job-executions	バッチジョブ実行の一覧表示	バッチジョブ実行と実行の詳細情報を一覧表示します。
list-batch-job-steps	バッチジョブの実行ステップの一覧表示	特定のバッチジョブの実行ステップを一覧表示します。
list-batch-jobs	バッチジョブの一覧表示	バッチジョブとジョブの詳細を一覧表示します。
list-batch-runtime-configuration	バッチ実行時の構成の表示	バッチ実行時の構成を表示します。
set-batch-runtime-configuration	バッチランタイムの設定	バッチランタイムを設定します。

## 並行リソース管理で使用するコマンド

コマンド名称	要約	概要説明
create-context-service	コンテキストサービスのリソースの作成	コンテキストサービスのリソースを作成します。
create-managed-executor-service	ManagedExecutorService リソースの作成	ManagedExecutorService リソースを作成します。
create-managed-scheduled-executor-service	ManagedScheduledExecutorService リソースの作成	ManagedScheduledExecutorService リソースを作成します。
create-managed-thread-factory	ManagedThreadFactory リソースの作成	ManagedThreadFactory リソースを作成します。
delete-context-service	コンテキストサービスのリソースの削除	指定した JNDI 名のコンテキストサービスのリソースを削除します。

コマンド名称	要約	概要説明
delete-managed-executor-service	ManagedExecutorService リソースの削除	指定した JNDI 名の ManagedExecutorService リソースを削除します。
delete-managed-scheduled-executor-service	ManagedScheduledExecutorService リソースの削除	指定した JNDI 名の ManagedScheduledExecutorService リソースを削除します。
delete-managed-thread-factory	ManagedThreadFactory リソースの削除	指定した JNDI 名の ManagedThreadFactory リソースを削除します。
list-context-services	コンテキストサービスのリソースの一覧表示	コンテキストサービスのリソースを一覧表示します。
list-managed-executor-services	ManagedExecutorService リソースの一覧表示	ManagedExecutorService リソースを一覧表示します。
list-managed-scheduled-executor-services	ManagedScheduledExecutorService リソースの一覧表示	ManagedScheduledExecutorService リソースを一覧表示します。
list-managed-thread-factories	ManagedThreadFactory リソースの一覧表示	ManagedThreadFactory リソースを一覧表示します。

## メッセージセキュリティプロバイダーの管理で使用するコマンド

コマンド名称	要約	概要説明
create-message-security-provider	メッセージセキュリティプロバイダーの作成	メッセージセキュリティプロバイダーを作成します。
delete-message-security-provider	メッセージセキュリティプロバイダーの削除	メッセージセキュリティプロバイダーを削除します。
list-message-security-providers	メッセージレイヤーのメッセージセキュリティプロバイダーの一覧表示	指定したメッセージレイヤー (domain.xml の message-security-config 要素) のすべてのメッセージセキュリティプロバイダー (provider-config 子要素) を一覧表示します。

## イベント検知時コマンド実行で使用するコマンド

コマンド名称	要約	概要説明
create-event-hook	イベント検知時コマンドの登録	ドメイン管理サーバでイベントが発生したときに、自動で実行されるイベント検知時コマンドを登録します。
delete-event-hook	イベント検知時コマンドの登録解除	イベント検知時コマンドの登録を解除します。
list-event-hooks	イベント検知時コマンドの一覧表示	登録されているイベント検知時コマンドを一覧表示します。



**!** 重要

- コマンドの引数には、各コマンドで指定できる値を指定してください。それ以外の値が指定された場合、動作は保証されません。
- コマンドが正常に終了しても、エラーが発生している場合があります。メッセージログやスタックトレースログを確認してください。

## 2.4 ドメインの管理で使用するコマンド

---

ドメインの管理で使用するコマンドの書式、機能などについて説明します。

### 2.4.1 backup-domain

ドメインのバックアップ

#### 書式

```
asadmin [asadmin-options] backup-domain [--help]
  [--long[={false|true}]] [--description description-text]
  [--domaindir domain-root-dir] [--backupdir backup-directory]
  [domain_name]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

backup-domain サブコマンドは、名前付きドメイン内のファイルをバックアップします。

このサブコマンドはローカルモードだけサポートします。

#### 前提条件

バックアップするドメインは、停止する必要があります。

#### ファイル

backup-domain サブコマンドを実行すると、バックアップ対象ドメインのすべてのファイルを含む ZIP アーカイブファイルが、--backupdir オプションで指定したディレクトリーに作成されます。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--long[={false|true}] | -l

バックアップ操作の詳細情報を表示します。

型：Boolean

指定できる値を次に示します。

- true

詳細情報を出力する

- `false`

詳細情報を出力しない

デフォルト値：`false`

`--description description-text`

バックアップファイルに設定する説明文を指定します。説明文は、バックアップファイルに関する情報の一部として表示されます。

型：`String`

指定できる値を次に示します。

- バックアップについて説明するテキスト

デフォルト値：次に示す形式

[*user-name* が、*YYYY\_MM\_DD* に作成した *domain-name* のバックアップ]

`--domaindir domain-root-dir`

ドメインルートディレクトリー（バックアップするドメインの親ディレクトリー）を指定します。

型：`String`

指定できる値を次に示します。

- ドメインのディレクトリーパス

デフォルト値：`Application Server`インストールディレクトリー/`javaee/glassfish/domains`

`--backupdir backup-directory`

バックアップファイルの保存先のディレクトリーを指定します。

型：`String`

指定できる値を次に示します。

- バックアップファイルが格納されるディレクトリーのパス

デフォルト値：`domain-dir/backups`

`domain_name`

バックアップするドメインの名称を指定します。

型：`String`

指定できる値を次に示します。

- ドメイン名

デフォルト値：なし

## 使用例

次の例では、デフォルトの `domain1` のバックアップを行い、バックアップファイルを `/net/backups.example.com/glassfish` に格納します。

```
asadmin backup-domain --backupdir /net/backups.example.com/glassfish domain1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.2 change-admin-password

管理ユーザーパスワードの変更

### 書式

```
asadmin [asadmin-options] change-admin-password [--help]
        [--domaindir domain-root-dir [--domain_name domain-name]]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

change-admin-password サブコマンドは、管理ユーザーのパスワードを変更します。このサブコマンドでは、古いパスワードと新しいパスワードの入力、および新しいパスワードの確認が対話型で求められます。新しいパスワードは 8 文字以上必要です。

次の場合、このサブコマンドはエラーになります。

- パスワードを指定していない匿名ユーザーの場合。
- 空白のパスワードを指定した場合 (secure admin 機能が有効な場合)。

Java EE Server に対して複数の管理ユーザーが設定されている場合、特定の管理ユーザーのパスワードを変更するには、asadmin コマンドに --user オプションを指定して実行する必要があります。

このサブコマンドはローカルモードとリモートモードをサポートします。

### 前提条件

Java EE Server に、2 人以上の管理ユーザーが設定されている場合、--user オプションを指定して asadmin コマンドを実行し、特定のユーザーのパスワードを変更する必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--domaindir domain-root-dir`

`--domain_name` オプションで指定したドメインの親ディレクトリーを指定します。このオプションを指定すると、`change-admin-password` サブコマンドはローカルモードで動作します。

型：String

指定できる値を次に示します。

- ドメインディレクトリーのパス

デフォルト値：なし

`--domain_name domain-name`

管理ユーザーのドメインを指定します。`--domaindir` オプションで指定したディレクトリーにドメインが1つだけの場合、このオプションは不要です。

型：String

指定できる値を次に示します。

- ドメイン名のパス

デフォルト値：なし

## 使用例

次の例では、シングルモードで1人の管理ユーザーのパスワードを変更します。

```
asadmin --user admin change-admin-password
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.3 create-domain

Java EE Server ドメインの作成

## 書式

```
asadmin [asadmin-options] create-domain [--help]  
      [--adminport adminport] [--instanceport instanceport]
```

```
[--portbase portbase] [--template template-name]  
[--domaindir domaindir] [--savemasterpassword={false|true}]  
[--usemasterpassword={false|true}]  
[--domainproperties name=value[:name=value]....]  
[--keytooloptions name=value[:name=value]....]  
[--savelogin={false|true}] [--checkports={true|false}]  
[--nopassword={false|true}] domain-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`create-domain` サブコマンドは、Java EE Server ドメインを作成します。

ドメインとは、Java EE 標準仕様に準拠した、管理上の名前空間です。ドメインごとに構成を持っており、その構成はファイルに保存されます。

ドメインは一意的管理 ID を持っており、Application Server をインストールした場所にはドメインをいくつでも作成できます。個々のドメインは、その他のドメインとは独立して存在します。

構築したシステム上の `asadmin` ユーティリティーにアクセスできるユーザーは、ドメインを作成し、ユーザーが指定したディレクトリーに構成を保存することができます。デフォルトでは、ドメインの構成はドメインのデフォルトディレクトリーに作成されます。デフォルトディレクトリーを上書きして、構成をほかの場所に保存することができます。

このサブコマンドは、`asadmin` ユーティリティーの `--user` オプションで指定した、1 人の管理ユーザーを持つドメインを作成します。

`--user` オプションが指定されていなくて、`--nopassword` オプションが `true` に設定されている場合、デフォルトの管理ユーザー `admin` を使用します。

`--nopassword` オプションが `false` に設定されている場合、ユーザー名が必要です。この場合、`--user` オプションでユーザー名を指定していないと、指定するように対話型で求められます。

新規ドメインで実行するアプリケーションに応じてドメインの適切なプロファイルを選択します。作成したドメインの開発者、クラスター、またはエンタープライズプロファイルを選択できます。

このサブコマンドはローカルモードだけサポートします。

## 実行権限

次に示す情報は、`--domainproperties` オプションだけに適用されます。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--adminport *adminport*

管理用の HTTP ポートまたは HTTPS ポートを指定します。このポートは、ユーザーが Web ブラウザーでドメインを管理するときに URL に指定するポートです。例えば、`http://localhost:4949` のような形式で指定します。

--adminport オプションは、--portbase オプションと同時に指定することはできません。

--adminport オプションは、--domainproperties オプションの `domain.adminPort` プロパティを上書きします。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：4848

--instanceport *instanceport*

デプロイ時に、アプリケーションの実行が行えるよう、サービスを提供するドメインを指定します。

この HTTP ポートは、Web ブラウザーが接続のために Web アプリケーションのコンテキストルートはどこで利用できるかを指定します。このポートは正の整数であり、ドメイン作成時に利用できるようになっている必要があります。

--instanceport オプションは、--portbase オプションと同時に指定することはできません。

--instanceport オプションは、--domainproperties オプションの `domain.instancePort` プロパティを上書きします。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：8080

--portbase *portbase*

ポート割当を開始する番号を指定します。

ドメインは、静的に割り当てられた一定数のポートを使用します。*portbase* の値は、割り当てをどこから始めるかを決定します。ポートの値は次のように算出されます。

- 管理ポート: `portbase + 48`
- HTTP リスナーポート: `portbase + 80`
- HTTPS リスナーポート: `portbase + 81`
- JMS ポート: `portbase + 76`
- IIOP リスナーポート: `portbase + 37`

- セキュア IIOP リスナーポート: portbase + 38
- 相互認証ポートのあるセキュア IIOP: portbase + 39
- JMX ポート: portbase + 86
- JPDA デバッガーポート: portbase + 9
- OSGi モジュール管理用 Felix シェルサービスポート: portbase + 66

--portbase オプションが指定されているとき、このサブコマンドの出力内容には、使用されたすべてのポートのリストが含まれます。

--portbase オプションは、--adminport、--instanceport、または--domainproperties オプションと同時に指定することはできません。

型: Integer

指定できる値を次に示します。

- -8~65449

デフォルト値: なし

#### --template *template-name*

相対パスまたは絶対パスを含む、ドメイン構成テンプレートのファイル名を指定して、ドメイン作成時に使用します。

相対パスが指定された場合、このサブコマンドは *Application Server* インストールディレクトリー/*javaee/glassfish/lib/templates* ディレクトリーにパスを付加してファイルを検索します。絶対パスが指定された場合、サブコマンドは指定されたパスでファイルを特定します。

このオプションを使うことによって、さまざまなタイプのドメインの作成やドメイン構成ファイルが利用できるようになります。

jar ファイルは、このオプションの入力ファイルで、ドメイン構成ファイルを含んでいます。

型: String

指定できる値を次に示します。

- ドメイン構成のファイル名

デフォルト値: なし

#### --domaindir *domaindir*

ドメインを作成するディレクトリーを指定します。指定した場合、ファイルシステム内でそのパスにアクセスする必要があります。指定しない場合、ドメインはデフォルトのドメインディレクトリー *Application Server* インストールディレクトリー/*javaee/glassfish/domains* に作成されます。

型: String

指定できる値を次に示します。

- ドメインのディレクトリーパス

デフォルト値: *Application Server* インストールディレクトリー/*javaee/glassfish/domains*



`--savemasterpassword={false|true}`

このオプションが設定されている場合、マスターパスワードをファイルシステムに書き込むことを許可します。

マスターパスワードは、実際にはセキュアキーストアのパスワードです。ドメインは、ドメイン作成時に作成される自身の証明書を、構成内の安全な場所に保存するよう設計されています。この証明書は、ドメインの SSL サーバ証明書と呼ばれます。Web ブラウザーが安全なチャンネル (HTTPS) でドメインにアクセスすると、この証明書がドメインによって提示されます。マスターパスワードは、この証明書を含むストア (ファイル) を保護します。このファイルは `keystore.jks` と呼ばれ、作成されたドメインの構成ディレクトリーに作成されます。しかし、もしこのオプションが選択された場合、マスターパスワードはドメインの構成内のディスクに保存されます。マスターパスワードは `master-password` というファイルに格納されますが、このファイルは Java の JCEKS タイプのキーストアです。

`--savemasterpassword` オプションはシステムブートの自動化に使用されます。マスターパスワードを保存しておく、この `keystore.jks` ファイルからパスワードが抽出されるので、ドメインの開始時にマスターパスワードの入力を求められることはありません。マスターパスワードは `start-domain` サブコマンドによって利用されるため、ドメイン作成時にはマスターパスワードを作成した方がよいでしょう。セキュリティー目的では、デフォルトの設定は `false` とすべきです。もし、ファイルシステムの権限が適切に設定されていない場合、マスターパスワードをディスクに保存していることは安全とは言えません。マスターパスワードが保存されている場合、`start-domain` はそのパスワードの入力を促しません。マスターパスワードはシステム環境のセキュリティーレベルに影響を与えます。

このオプションが `true` の場合、コマンドラインで指定した値に関係なく、`--usemasterpassword` オプションも `true` になります。

型: Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値: `false`

`--usemasterpassword={false|true}`

システムに組み込まれているマスターパスワードまたはユーザー定義マスターパスワードでキーストアを暗号化するかどうかを指定します。

`false` の場合、キーストアは、システムに組み込まれている `well-known` パスワードで暗号化されません。システムに組み込まれているパスワードでキーストアを暗号化する場合、セキュリティーの向上はありません。

`true` の場合、このサブコマンドはパスワードファイルの `AS_ADMIN_MASTERPASSWORD` エントリーからマスターパスワードを取得するか、またはマスターパスワードの入力を促します。パスワードファイルは、`asadmin` ユーティリティーの `--passwordfile` オプションに指定されています。

`--savemasterpassword` オプションが `true` の場合には、コマンドラインで指定された値に関係なく、このオプションも `true` となります。

型: Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--domainproperties name=value[:name=value]...`

オプションの名前と値のペアを設定します。作成するドメインのプロパティのデフォルト値を上書きします。

指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

`--portbase` オプションは、`--domainproperties` オプションと同時に指定することはできません。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

プロパティ名と `value` に指定できる値を次に示します。

`domain.adminPort=value`

このプロパティは、管理する HTTP または HTTPS ポートのポート番号を指定します。このポートは Web ブラウザーでインスタンスを管理するために指定する URL のポートです (例：`http://localhost:4949`)。

`domain.adminPort` プロパティは、`--adminport` オプションで上書きされます。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`domain.instancePort=value`

このプロパティは、HTTP 要求をリッスンするのに使用するポートのポート番号を指定します。

`domain.instancePort` プロパティは、`--instanceport` オプションで上書きされます。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`domain.jmxPort=value`

このプロパティは、JMX コネクターがリッスンするポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`http.ssl.port=value`

このプロパティは、HTTPS 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`java.debugger.port=value`

このプロパティは、Java Platform Debugger Architecture (JPDA) (<http://java.sun.com/javase/technologies/core/toolsapis/jpda/>) デバッガーへの接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`jms.port=value`

このプロパティは、JMS プロバイダーのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`orb.listener.port=value`

このプロパティは、IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`orb.mutualauth.port=value`

このプロパティは、クライアント認証を持つセキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`orb.ssl.port=value`

このプロパティは、セキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`osgi.shell.Telnet.port=value`

このプロパティは、Apache Felix Remote Shell (<http://felix.apache.org/site/apache-felix-remote-shell.html>) への接続に使用するポートのポート番号を指定します。

このシェルは Felix シェルサービスを使用して、OSGi モジュール管理サブシステムと対話します。

`--portbase` オプションは `--domainproperties` オプションと同時に使用はできません。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`--keytooloptions name=value[:name=value]...`

自己署名証明書のキーツールオプションの名前と値のペアを指定します。証明書は、ドメインの作成中に生成されます。

指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じオプション名を複数指定した場合は、最後に指定したオプションの値が有効となります。

オプション名と `value` に指定できる値を次に示します。

`CN=value`

自己署名証明書に使用されるホストの共通名を指定します。このオプションの名前は、大文字小文字を区別しません。

型：String

デフォルト値：サブコマンドが実行されているホストの完全修飾名

`--savelogin={false|true}`

このオプションが `true` に設定されている場合、管理ユーザー名とパスワードを保存します。

ユーザー名とパスワードは、ユーザーのホームディレクトリーの `.asadminpass` ファイルに格納されています。ドメインはローカルだけで作成できます。そのため、`--savelogin` オプションの利用時には、`.asadminpass` に保存されているホスト名は常に `localhost` です。ユーザーがドメインの作成中にデフォルトの管理ポートを指定した場合、その後の `asadmin` リモートコマンドで `--user`、`--passwordfile`、`--host`、または `--port` を指定する必要はありません。これらの値は自動的に取得されます。

同じホスト上またはホームディレクトリーを NFS マウントしているホスト上に、同じの管理ポート番号を持つ複数ドメインを、同じユーザーが作成する場合であっても、サブコマンドはパスワードを上書きするかどうかを確認しません。パスワードは常に上書きされます。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：false

`--checkports={true|false}`

管理、HTTP、JMS、JMX、IIOP の各ポートが利用できることを確認するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--nopassword={false|true}`

管理ユーザーがパスワードを持つかどうかを指定します。

false の場合、パスワードは、asadmin パスワードファイルの `AS_ADMIN_PASSWORD` エントリーによって設定されます。asadmin パスワードファイルは `--passwordfile` オプションで設定します。false で `AS_ADMIN_PASSWORD` が設定されていない場合は、パスワードの入力が求められます。

true の場合、パスワードのない管理ユーザーが作成されます。`--user` オプションでドメインのユーザー名の指定がなく、`--nopassword` オプションが true に設定されている場合には、デフォルトのユーザー名である `admin` が使われます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

*domain-name*

作成するドメインの名称を指定します。

型：String

指定できる値を次に示します。

- ドメイン名

ドメイン名は、次に示す条件を満たしている必要があります。

- ASCII 文字であること。
- ドメインが作成されるホストの OS で有効な名称であること。

デフォルト値：なし

## 使用例

次の例では、ドメインを作成します。

```
asadmin create-domain --adminport 4848 domain4
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.4 delete-domain

ドメインの削除

### 書式

```
asadmin [asadmin-options] delete-domain [--help]
        [--domaindir domaindir] domain-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-domain` サブコマンドは、指定したドメインを削除します。ドメインはすでに存在し、停止している必要があります。

このサブコマンドはローカルモードだけサポートします。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--domaindir domaindir`

削除対象ドメインが位置するディレクトリーを指定します。

型：String

指定できる値を次に示します。

- ドメインが位置するディレクトリー名

デフォルト値：*Application Server*インストールディレクトリー/*javaee/glassfish/domains*

`domain-name`

ドメインの固有の名称を指定します。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：なし

## 使用例

次の例では、デフォルトのドメインディレクトリーからmydomain4 という名称のドメインを削除します。

```
asadmin delete-domain mydomain4
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.5 list-backups

ドメインバックアップの一覧表示

### 書式

```
asadmin [asadmin-options] list-backups [--help]
  [--long[={false|true}]]
  [--domaindir domain-root-dir]
  [--backupdir backup-directory]
  [domain-name]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

*list-backups* サブコマンドは、ドメインバックアップに関する情報を表示します。

このサブコマンドはローカルモードだけサポートします。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--long={false|true} | -l={false|true}

バックアップごとの詳細情報を表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--domaindir domain-root-dir`

ドメインルートディレクトリーおよびコマンドを実行するドメインの親ディレクトリーを指定します。

型：String

指定できる値を次に示します。

- ディレクトリー名

デフォルト値：Application Serverインストールディレクトリー/javaee/glassfish/domains

`--backupdir backup-directory`

バックアップファイルが格納されているディレクトリーを指定します。

型：String

指定できる値を次に示します。

- ディレクトリー名

デフォルト値：Application Serverインストールディレクトリー/javaee/glassfish/domains/domain-dir/backups (ドメインがデフォルトの場所がない場合はdomain-dir/backups)

`domain-name`

バックアップを一覧表示するドメインを指定します。

Java EE Server ドメインが1つだけの場合は省略できます。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：なし

## 使用例

次の例では、/net/backups.example.com/glassfish ディレクトリーに格納されているデフォルトのドメインdomain1のバックアップを一覧表示します。

```
asadmin list-backups --backupdir /net/backups.example.com/glassfish domain1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。



戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.6 list-domains

ドメインディレクトリー内のドメインの一覧表示

### 書式

```
asadmin [asadmin-options] list-domains [--help]
  [--domaindir domaindir]
  [--long={false|true}]
  [--header={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-domains` サブコマンドは、指定したドメインディレクトリーにあるドメインを一覧表示します。ドメインディレクトリーが指定されていない場合、デフォルトのディレクトリーにあるドメインが一覧表示されます。

このサブコマンドはローカルモードだけサポートします。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--domaindir domaindir`

一覧表示されるドメインディレクトリーを指定します。指定した場合、ファイルシステムでパスにアクセスできることが必要です。指定されていない場合、デフォルトの*Application Server*インストールディレクトリー/*javaee/glassfish/domains* ドメインルートディレクトリーのドメインが一覧表示されます。

型：String

指定できる値を次に示します。

- ディレクトリー名

デフォルト値：*Application Server*インストールディレクトリー/*javaee/glassfish/domains*

`--long={false|true} | -l={false|true}`

一覧表示されたドメイン管理サーバに関する詳細情報（ホスト名とポート番号を含む）が表示されます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--header={false|true} | -h={false|true}`

`--long` オプションを使用したときにヘッダーを表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

## 使用例

次の例では、デフォルトのディレクトリーのドメインを一覧表示します。

```
asadmin list-domains
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.7 restart-domain

ドメイン管理サーバの再起動

### 書式

```
asadmin [asadmin-options] restart-domain [--help]
  [--debug={true|false}]
  [--domaindir domaindir]
  [--force={true|false}] [--kill={false|true}]
  [domain-name]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`restart-domain` サブコマンドは、指定したドメインのドメイン管理サーバを停止した後、再起動します。

ドメインが指定されていない場合、デフォルトのドメインが使用されます。domains ディレクトリーにドメインが複数ある場合、`domain-name` を指定する必要があります。ドメイン管理サーバが稼働していない場合、このサブコマンドはドメイン管理サーバの起動を試みます。

このサブコマンドはローカルモードとリモートモードをサポートします。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--debug={true|false}`

Java Platform Debugger Architecture (JPDA) デバッグを有効にしてドメインを再起動するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`

JPDA デバッグを有効にしてドメインを再起動し、JPDA デバッグのポート番号を表示します。

- `false`

JPDA デバッグを無効にしてドメインを再起動します。

デフォルト値：再起動するドメインに対するこのオプションの現在の設定

`--domaindir domaindir`

再起動するドメインのディレクトリーを含むルートディレクトリーを指定します。

型：String

指定できる値を次に示します。

- ドメインルートディレクトリーのパス

デフォルト値：*Application Server* インストールディレクトリー/*javaee/glassfish/domains*

`--force={true|false}`

ドメインを再起動する直前に、ドメインを強制停止するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`

ドメインはただちに強制停止されます。

- `false`

ドメインに関連づけられているすべてのスレッドが終了するまで待機してから、ドメインを停止します。

デフォルト値：true

`--kill={false|true}`

ドメインプロセスを停止する OS の機能を使用して、ドメインを再起動する前に強制終了するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- **false**

ドメインを強制終了しません。Java プラットフォームの機能を使用してドメインプロセスを停止します。

- **true**

ドメインを強制終了します。OS の機能を使用してドメインプロセスを停止します。

デフォルト値：false

*domain-name*

再起動するドメインの名称を指定します。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：インストール時に指定した名称で、通常はdomain1

## 使用例

次の例では、デフォルトのdomains ディレクトリー内のmydomain4 を再起動します。

```
asadmin restart-domain mydomain4
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.8 restore-domain

バックアップからのファイルの復元

## 書式

```
asadmin [asadmin-options] restore-domain [--help]
  [--long[={false|true}]] [--filename backup-filename]
  [--domaindir domain-root-dir] [--backupdir backup-directory]
  [--force[={false|true}]] [domain-name]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

restore-domain サブコマンドは、バックアップディレクトリーからドメイン内にファイルを復元します。

このサブコマンドはローカルモードだけサポートします。

なお、異なる OS でバックアップしたバックアップファイルは復元できません。

## 前提条件

ドメインを停止する必要があります。

## ファイル

指定されたドメインのディレクトリー内のファイルおよびサブディレクトリー

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--long[={false|true}] | -l

復元操作の詳細情報を表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

--filename *backup-filename*

復元に使用するバックアップファイルの名称を指定します。

型：String

指定できる値を次に示します。

- ファイル名

デフォルト値：なし

`--domaindir domain-root-dir`

ドメインルートディレクトリー（復元するドメインの親ディレクトリー）を指定します。

型：String

指定できる値を次に示します。

- ディレクトリーのパス

デフォルト値：*Application Server*インストールディレクトリー/javaee/glassfish/domains

`--backupdir backup-directory`

バックアップファイルが格納されているディレクトリーを指定します。

型：String

指定できる値を次に示します。

- ディレクトリーのパス

デフォルト値：*Application Server*インストールディレクトリー/javaee/glassfish/domains/domain-dir/backups ドメインがデフォルトの場所がない場合は、ロケーションはdomain-dir/backups です。

`--force[={false|true}]`

復元するドメインの名称がバックアップファイルに格納されているドメインの名称と一致しなくても復元操作を継続できるようにします。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

*domain-name*

復元するドメインの名称を指定します。

Java EE Server にドメインが 1 つだけの場合、この *domain-name* は省略できます。

指定したドメイン名とバックアップファイルに格納されているドメイン名が一致していない場合、`--force` オプションを指定していないとエラーが発生します。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：Java EE Server で有効な単一のドメイン

## 使用例

次の例では、指定されたバックアップディレクトリーに格納されている最新バックアップから、デフォルトドメインのdomain1 のファイルを復元します。

```
asadmin restore-domain --backupdir /net/backups.example.com/glassfish domain1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.9 start-domain

ドメイン管理サーバの起動

### 書式

```
asadmin [asadmin-options] start-domain  
  [--help] [--debug={true|false}] [--domaindir domain-dir]  
  [--dry-run={true|false}] [--upgrade={true|false}]  
  [--verbose={true|false}] [--watchdog={true|false}] [domain-name]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`start-domain` サブコマンドは、指定したドメインのドメイン管理サーバを起動します。ドメインを指定しない場合、デフォルトのドメインが使用されます。`domains` ディレクトリーにドメインが複数ある場合、`domain-name` を指定する必要があります。

このサブコマンドはローカルモードだけサポートします。

`start-domain` サブコマンドは、管理ユーザーにパスワードが設定されていない場合、新たな管理ユーザーパスワードの入力を促すプロンプトを表示します。また、セキュア管理が有効な場合、管理ユーザーパスワードを必ず指定してください。空白の場合、`start-domain` サブコマンドは失敗します。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`{--debug|-d}={true|false}`

Java Platform Debugger Architecture (JPDA) デバッグを有効にしてドメインを起動するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

JPDA デバッグを有効にしてインスタンスを起動し、JPDA デバッグのポート番号を表示します。

- false

JPDA デバッグを無効にしてインスタンスを起動します。

デフォルト値：false

`--domaindir domain-dir`

再起動するドメインのディレクトリーを含むルートディレクトリーを指定します。指定した場合、このパスはファイルシステム内でアクセスできなければなりません。

型：String

指定できる値を次に示します。

- ドメインルートディレクトリーのパス

デフォルト値：*Application Server*インストールディレクトリー/*javaee/glassfish/domains*

`{--dry-run|-n}={true|false}`

ドメインの実際の起動を抑止します。代わりに、`start-domain` サブコマンドは、ドメインを起動するために使用される Java コマンド全体をすべてのオプションを含めて表示します。このサブコマンドを見直すと、JVM オプションを確認する場合や起動に関する問題をトラブルシューティングする場合に役立ちます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--upgrade={true|false}`

Hitachi Application Server の上書きインストールでバージョンまたはリビジョンが上がった場合に、ドメイン管理サーバの設定をアップグレードするために使用します。

型：Boolean

指定できる値を次に示します。

- true

ドメインが起動されると、Hitachi Application Server の現在のリリースに合うように設定が変更され、ドメイン管理サーバのプロセスが停止します。

- false

ドメイン管理サーバの設定は更新されません。

デフォルト値：false



`{--verbose|-v}={true|false}`

ドメインに関する詳細情報を、サブコマンドを実行するコンソール画面に表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- **true**

ドメインに関する詳細なメッセージおよびログメッセージが、このサブコマンドを実行するコンソール画面に表示されます。別のコンソール画面から `restart-domain` サブコマンドを実行してドメインをあとで再起動する場合は、メッセージは元のコンソール画面に引き続き表示されます。コンソール画面で [CTRL] + [C] を入力すると、Java EE Server のプロセスを強制終了できます。コンソール画面で次のキーの組み合わせを入力すると、Java EE Server のプロセスを強制終了したりサーバのスレッドダンプを取得したりできます。

[CTRL] + [¥]

- **false**

詳細情報は表示されません。

デフォルト値：false

`{--watchdog|-w}={true|false}`

ドメインに関する限定された情報を、このサブコマンドを実行するコンソール画面に表示するかどうかを指定します。

`--watchdog` オプションは `--verbose` オプションと似ていますが、詳細なメッセージおよびログメッセージを表示しません。このオプションは、`asadmin` ユーティリティーをバックグラウンドで実行する場合やコンソールを取り付けずに実行する場合に役立ちます。

型：Boolean

指定できる値を次に示します。

- **true**

限定された情報がコンソールに表示されます。

- **false**

限定された情報はコンソールに表示されません。

デフォルト値：false

### *domain-name*

起動するドメインの名称を指定します。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：なし

## 使用例

次の例では、デフォルトのdomains ディレクトリーのmydomain4 を起動します。

```
asadmin start-domain mydomain4
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.4.10 stop-domain

ドメイン管理サーバの停止

### 書式

```
asadmin [asadmin-options] stop-domain [--help]
        [--domaindir domaindir]
        [--kill={false|true}] [domain-name]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

stop-domain サブコマンドは、指定したドメインのドメイン管理サーバを停止します。ドメインディレクトリーが指定されていない場合、デフォルトのdomains ディレクトリーにあるドメインが停止されます。domains ディレクトリーにドメインが複数ある場合、*domain-name* を指定する必要があります。

このサブコマンドはローカルモードとリモートモードをサポートします。

ホスト名を指定した場合、このサブコマンドはリモートモードでの操作を仮定するため、リモートサーバに対して正しく認証する必要があります。ローカルモードでは、サーバを起動したユーザーと同じユーザーでこのサブコマンドを実行する限り、通常サーバに対する認証は必要ありません。

サーバ停止時に、標準エラーおよびlogger.log ファイルにKDKD45000-E のメッセージを出力することがありますが、終了処理に影響はありません。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--domaindir={true|false}`

停止するドメインのディレクトリーを指定します。指定した場合、このパスはファイルシステム内でアクセスできなければなりません。指定しない場合、デフォルトの *Application Server* インストールディレクトリー/`javaee/glassfish/domains` ディレクトリーにあるドメインは停止します。

型：String

指定できる値を次に示します。

- ドメインルートディレクトリーのパス

デフォルト値：*Application Server* インストールディレクトリー/`javaee/glassfish/domains`

`--kill={false|true}`

ドメインプロセスを停止する OS の機能を使用して、ドメインを強制終了するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `false`

ドメインを強制終了しません。Java プラットフォームの機能を使用してドメインプロセスを停止します。

- `true`

ドメインを強制終了します。OS の機能を使用してドメインプロセスを停止します。

デフォルト値：`false`

*domain-name*

停止するドメインの名称を指定します。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：`domain1`

## 使用例

次の例では、デフォルトの `domains` ディレクトリー内の `sampleDomain` という名称のドメインを停止します。

```
asadmin stop-domain sampleDomain
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5 ノードの管理で使用するコマンド

---

ノードの管理で使用するコマンドの書式、機能などについて説明します。

### 2.5.1 create-node-config

リモート通信が有効化されていないノードの作成

#### 書式

```
asadmin [asadmin-options] create-node-config [--help]
  [--nodehost node-host] [--installdir as-install-parent]
  [--nodedir node-dir] node-name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

create-node-config サブコマンドは、リモート通信が有効化されていないノードを作成します。

このサブコマンドでノードを作成する場合、SSH が設定されている必要がありません。

ノードは *Application Server* がインストールされたホストを表します。

サーバインスタンスが存在するホストすべてに対してノードが存在する必要があります。

リモート通信が有効化されていないノードのインスタンスの管理は、すべてそのノードが表すホスト上で行われる必要があります。

リモートホストのドメイン管理サーバはノードにアクセスできません。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

サーバインスタンスがある、すべてのホストにノードが存在する必要があります。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

### `--nodehost node-host`

ノードが表すホスト名を指定します。このオプションが省略されると、ノードに対するホストが指定されないこととなります。

型：String

指定できる値を次に示します。

- ホスト名

デフォルト値：なし

### `--installdir as-install-parent`

*Application Server*インストールディレクトリー/*javaee* の絶対パスを指定します。このオプションが省略されると、ノードに対して*Application Server*インストールディレクトリー/*javaee* が指定されないこととなります。

型：String

指定できる値を次に示します。

- *Application Server*インストールディレクトリー/*javaee* の絶対パス

デフォルト値：なし

### `--nodedir node-dir`

ノードに作成されたサービンスタンスを含むディレクトリーのパスを指定します。相対パスが指定されると、パスはインストール時のディレクトリーからの相対となります。このオプションが省略されると、ノードに対してインスタンスのディレクトリーが指定されないこととなります。

型：String

指定できる値を次に示します。

- ノードが存在するディレクトリーのファイルパス

デフォルト値：なし

### *node-name*

ノード名を指定します。

型：String

指定できる値を次に示します。

- ノード名

名称は、次の条件を満たす必要があります。

ASCII 文字だけで構成する。

文字、数字、またはアンダースコアで開始する必要があります。

次の文字だけを使用すること。

- 小文字
- 大文字
- 数字

- ハイフン
- ピリオド
- アンダースコア

マシン内で一意である必要があり、同一マシン上に作成されている別のノード名、クラスター名、指定された JavaEE サーバの構成名、またはサーバインスタンスの名称と重複できません。

domain、server、または Java EE Server が予約したその他のキーワードは使用できません。

デフォルト値：なし

## 使用例

次の例では、ノードを作成します。

```
asadmin create-node-config --nodehost sj03.example.com sj03
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.2 create-node-ssh

SSH が有効なノードの作成

### 書式

```
asadmin [asadmin-options] create-node-ssh [--help]
--nodehost node-host [--installdir as-install-parent]
[--nodedir node-dir] [--sshport ssh-port] [--sshuser ssh-user]
[--sshkeyfile ssh-keyfile] [--force={false|true}]
node-name
```

### 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

### 機能

create-node-ssh サブコマンドは、SSH を介した通信が有効なノードを作成します。

ノードとは、Application Server がインストールされたホストのことを示します。ノードは、サーバインスタンスが常駐するホストごとに必要です。

ドメイン管理サーバは、SSH コネクタ経由で SSH ノードのホストと通信して、ノードに常駐するサーバインスタンスを管理します。しかし、ドメイン管理サーバは、すべての `asadmin` サブコマンドをローカルで実行できるため、SSH コネクタを使用しないでドメイン管理サーバが稼働するホストと通信します。

このサブコマンドは、リモートモードだけサポートします。

SSH 経由でホストと通信できない場合でも、ドメイン管理サーバの設定内にノードを強制的に作成するには、`--force` オプションに `true` を設定します。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

ノードは、サーバインスタンスが常駐するホストごとに必要です。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--nodehost node-host`

ノードが示すホストの名前を指定します。ホスト名は、必ず指定してください。指定しない場合は、エラーが発生します。

型：String

指定できる値を次に示します。

- ノードのホスト名

デフォルト値：なし

`--installdir as-install-parent`

*Application Server* インストールディレクトリー/*javaee* の絶対パスを指定します。

型：String

指定できる値を次に示します。

- *Application Server* インストールディレクトリー/*javaee* の絶対パス

デフォルト値：ドメイン管理サーバの *Application Server* インストールディレクトリー/*javaee* ディレクトリー。

`--nodedir node-dir`

ノードに作成されたサーバインスタンスを格納するディレクトリーへのパスを指定します。相対パスを指定した場合、パスはインストール時のディレクトリーからの相対になります。

型：String

指定できる値を次に示します。

- サーバインスタンスを格納するディレクトリーへのパス



デフォルト値：*Application Server*インストールディレクトリー/*javaee/glassfish/nodes*

#### `--sshport ssh-port`

このノードのホストへの SSH コネクションに使用するポートを指定します。--nodehost オプションに *localhost-domain* が設定されている場合、--sshport オプションは無視されます。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：22

#### `--sshuser ssh-user`

SSH 経由でホストに接続するプロセスを実行する、このノードのホスト上のユーザーを指定します。このユーザーの SSH 秘密鍵ファイルをドメイン管理サーバが確実に読み込めるようにするには、ドメイン管理サーバのプロセスを実行中のユーザーを指定します。--nodehost オプションに *localhost* ドメインが設定されている場合、--sshuser オプションは無視されます。

型：String

指定できる値を次に示します。

- sshuser 名

デフォルト値：ドメイン管理サーバのプロセスを稼働中のユーザー

#### `--sshkeyfile ssh-keyfile`

--sshuser オプションに指定されたユーザーの SSH 秘密鍵ファイルへの絶対パスを指定します。このファイルは、ノードのホスト上にある SSH 化されたデーモンへの認証に使用されます。

型：String

指定できる値を次に示します。

- ユーザーの SSH 秘密鍵ファイルへのパス

デフォルト値：このサブコマンドを実行しているホストにあるユーザーの *.ssh* ディレクトリー内のキーファイル。

ユーザーの *.ssh* ディレクトリーに複数のキーファイルがある場合、このサブコマンドは次の優先順序で使用します。

1. *id\_rsa*
2. *id\_dsa*
3. *identity*

#### `--force={false|true}`

ノードのパラメーターの検証に失敗した場合でも、ノードがドメイン管理サーバの設定に作成されるかどうかを指定します。

ノードのパラメーターを検証するには、ドメイン管理サーバは、SSH 経由でノードのホストと通信できる必要があります。

型：Boolean

指定できる値を次に示します。

- `true`

ノードのパラメーターの検証に失敗した場合でも、ノードは作成されます。

- `false`

ノードのパラメーターの検証に失敗した場合、ノードは作成されません。

デフォルト値：`false`

#### *node-name*

ノード名を指定します。

型：`String`

指定できる値を次に示します。

- ノード名

名称は、次の条件を満たす必要があります。

ASCII 文字だけで構成する。

文字、数字、またはアンダースコアで開始する必要があります。

次の文字だけを使用すること。

- 小文字
- 大文字
- 数字
- ハイフン
- ピリオド
- アンダースコア

マシン内で一意である必要があり、同一マシン上に作成されている別のノード名、クラスター名、指定された JavaEE サーバの構成名、またはサーバインスタンスの名称と重複できません。

`domain`、`server`、または `Java EE Server` が予約したその他のキーワードは使用できません。

デフォルト値：なし

## 使用例

次の例では、ノードを作成します。

```
asadmin create-node-ssh --nodehost adc.example.com --installdir /export/glassfish3 adc
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.3 delete-node-config

リモート通信が無効なノードの削除

### 書式

```
asadmin [asadmin-options] delete-node-config [--help]
node-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-node-config` サブコマンドは、ドメインからリモート通信が無効なノードを削除します。このサブコマンドでは、SSH を構成する必要がありません。削除するノードにサーバインスタンスが存在する場合、サブコマンドの実行に失敗します。このサブコマンドを実行する前に、ノードに存在するサーバインスタンスを削除してください。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*node-name*

ノード名を指定します。

ノードは SSH 上での通信が無効である必要があります。該当しない場合は、エラーが発生します。

型：String

指定できる値を次に示します。

- ノード名

デフォルト値：なし

## 使用例

次の例では、リモート通信が無効なノードsj03を削除します。

```
asadmin delete-node-config sj03
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.4 delete-node-ssh

SSH 通信が有効なノードの削除

### 書式

```
asadmin [asadmin-options] delete-node-ssh [--help]  
node-name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-node-ssh サブコマンドは、ドメインからの SSH 通信が有効なノードを削除します。このサブコマンドでは、SSH を構成する必要がありません。

このサブコマンドは、SSH 経由での通信が有効なノードだけを削除できます。

削除するノードにサーバインスタンスが存在する場合、サブコマンドの実行に失敗します。このサブコマンドを実行する前に、ノードに存在するサーバインスタンスを削除してください。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*node-name*

ノード名を指定します。

型：String

指定できる値を次に示します。

- ノード名

デフォルト値：なし

## 使用例

次の例では、SSH 上の通信が有効なノードeg1 を削除します。

```
asadmin delete-node-ssh eg1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.5 list-nodes

Java EE Server ノードの一覧表示

### 書式

```
asadmin [asadmin-options] list-nodes [--help]
        [--long={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-nodes サブコマンドはドメインにあるすべての Java EE Server ノードを一覧表示します。

デフォルトでは、このサブコマンドは一覧表示された各ノードの次の情報を表示します。

- ノード名
- 次のノードタイプのどれか1つ  
CONFIG：リモート通信をサポートしないノード  
SSH：SSH 経由の通信をサポートするノード
- ノードが表すホストの名称

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--long={false|true} | -l={false|true}`

ノードを長形式で一覧表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
ノードを長形式で一覧表示します。
- `false`  
ノードを短形式で一覧表示します。

デフォルト値：`false`

## 使用例

次の例では、Java EE Server ノードを長形式で一覧表示します。

```
asadmin list-nodes --long=true
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.6 list-nodes-config

リモート通信をサポートしないドメイン内の Java EE Server ノードの一覧表示

### 書式

```
asadmin [asadmin-options] list-nodes-config [--help]
        [--long={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-nodes-config` サブコマンドは、リモート通信をサポートしない、ドメイン内すべての Java EE Server ノードを一覧表示します。

デフォルトでは、一覧表示される各ノードについて次の情報が表示されます。

- ノード名
- ノードのタイプ (CONFIG)
- ノードが表すホストの名称

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--long={false|true} | -l={false|true}`

ノードを長形式で一覧表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
ノードを長形式で一覧表示します。
- `false`  
ノードを短形式で一覧表示します。

デフォルト値：`false`

## 使用例

次の例では、ドメインにあるリモート通信をサポートしない Java EE Server ノードを長形式で一覧表示します。

```
asadmin list-nodes-config --long=true
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.7 list-nodes-ssh

SSH 経由の通信をサポートするドメイン内の Java EE Server ノードの一覧表示

### 書式

```
asadmin [asadmin-options] list-nodes-ssh [--help]
        [--long={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-nodes-ssh サブコマンドは、SSH 経由の通信をサポートするドメイン内すべての Java EE Server ノードを一覧表示します。

デフォルトでは、一覧表示される各ノードについて次の情報が表示されます。

- ノード名
- ノードのタイプ (SSH)
- ノードが表すホストの名称

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。



## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--long={false|true} | -l={false|true}`

ノードを長形式で一覧表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`

ノードを長形式で一覧表示します。

- `false`

ノードを短形式で一覧表示します。

デフォルト値：`false`

## 使用例

次の例では、SSH 経由の通信をサポートするドメイン内すべての Java EE Server ノードの名称、タイプ、およびホストを表示します。

```
asadmin list-nodes-ssh
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.8 ping-node-ssh

SSH 通信によるノード使用のテスト

### 書式

```
asadmin [asadmin-options] ping-node-ssh [--help]  
      [--validate={false|true}] node-name
```

## 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

## 機能

ping-node-ssh サブコマンドは、SSH 経由で、ノードが表すホスト上でasadmin ユーティリティーコマンドを実行できるかどうかをテストします。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- ドメイン管理サーバが稼働しているマシンと、ノードが存在するマシンに SSH が設定されている必要があります。
- ノードで SSH 経由の通信が有効になっている必要があります。それ以外の場合は、エラーが発生します。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--validate={false|true}

サブコマンドがノードを検証するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true  
ノードを検証します。
- false  
ノードを検証しません。

デフォルト値：false

*node\_name*

テストするノードの名称を指定します。

型：String

指定できる値を次に示します。

- ノードの名称

デフォルト値：なし

## 使用例

次の例では、SSH 経由の通信が有効化されているノードsj03-node が使用できるかどうかをテストします。

```
asadmin ping-node-ssh sj03-node
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.9 setup-ssh

SSH キーのセットアップ

### 書式

```
asadmin [asadmin-options] setup-ssh [--help]
  [--sshport ssh-port] [--sshuser ssh-user]
  [--sshkeyfile ssh-keyfile] [--sshpublickeyfile ssh-public-keyfile]
  [--generatekey={false|true}] host-list
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

setup-ssh サブコマンドは、指定されたホストに SSH キーをセットアップします。このキーによって、Java EE Server は、リモートホストへのユーザーの SSH ログイン時の認証に、公開鍵認証を使用できるようになります。

SSH によって、複数ホストにおよぶ Java EE Server クラスターが一括管理できるようになります。

SSH は、ドメイン管理サーバおよびリモートホスト間で交換されるデータの機密性および安全性をもたらします。

公開鍵認証では、次のキーから構成される SSH キーペアを使用します。

- ドメイン管理サーバホストの安全な場所に格納されていて、パスフレーズで保護されている可能性のある秘密鍵
- ドメイン管理サーバが通信するすべてのリモートホストに格納されている公開鍵

setup-ssh サブコマンドは、ドメイン管理サーバからの設定情報を要求しません。また、ドメイン管理サーバの設定を修正しません。

このサブコマンドはローカルモードだけサポートします。

## 前提条件

- ssh クライアントは、ドメイン管理サーバホストにインストールされ、ドメイン管理サーバユーザーのパス経由でアクセスできる必要があります。
- sshd デーモンは、SSH キーがセットアップされるすべてのホストにインストールされ、常駐している必要があります。
- ssh-keygen ユーティリティーは、デフォルトの場所またはドメイン管理サーバユーザーのパスに定義された場所にあるドメイン管理サーバホストにインストールされている必要があります。
- --sshuser に指定されたユーザーは、SSH キーがセットアップされるすべてのホストに対して SSH 経由でログインできる必要があります。

## ファイル

ドメイン管理サーバホストおよびリモートホスト両方の *user-home/.ssh/* にある次に示すファイルが参照されます。

- authorized\_keys
- id\_rsa
- id\_rsa.pub
- id\_dsa
- id\_dsa.pub
- identity
- identity.pub

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--sshport *ssh-port*

SSH をセットアップ中のホストへの SSH コネクションに使用するポートを指定します。

型：Integer

指定できる値を次に示します。

- 1-65535

デフォルト値：22

--sshuser *ssh-user*

リモートホスト上の SSH ユーザーを指定します。この SSH ユーザーは、リモートホスト上で SSH をセットアップするプロセスを実行します。

ドメイン管理サーバがこのユーザーの SSH 秘密鍵ファイルを読めるようにするには、ドメイン管理サーバのプロセスを実行中のユーザーを指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：コマンドを実行中のユーザー

#### `--sshkeyfile ssh-keyfile`

`--sshuser` オプションで指定したユーザーの SSH 秘密鍵ファイルへの絶対パスを指定します。このファイルは、ホスト上の `sshd` デーモンへの認証に使用されます。このサブコマンドを実行中のユーザーは、キーファイルへのパスにアクセスしてキーファイルを読める必要があります。

型：String

指定できる値を次に示します。

- ユーザーの SSH 秘密鍵ファイルへのパス

デフォルト値：サブコマンドを実行しているホストにあるユーザーの `.ssh` ディレクトリー内のキーファイル。

ユーザーの `.ssh` ディレクトリーに複数のキーファイルがある場合、サブコマンドは次の優先順序で使用します。

1. `id_rsa`
2. `id_dsa`
3. `identity`

#### `--sshpublickeyfile ssh-public-keyfile`

`--sshuser` オプションで指定したユーザーの SSH 公開鍵ファイルへの絶対パスを指定します。公開鍵ファイルの内容は、SSH をセットアップ中の各ホスト上にあるユーザーの `.ssh/authorized_keys` ファイルに追加されます。

`.ssh/authorized_keys` ファイルがホスト上に存在しない場合は、サブコマンドがファイルを作成します。このサブコマンドを実行中のユーザーはキーファイルへのパスにアクセスしてキーファイルを読める必要があります。

型：String

指定できる値を次に示します。

- ユーザーの SSH 公開鍵ファイルへのパス

デフォルト値：サブコマンドを実行しているホストにあるユーザーの `.ssh` ディレクトリー内のキーファイル。

ユーザーの `.ssh` ディレクトリーに複数のキーファイルがある場合、サブコマンドは次の優先順序で使用します。

1. `id_rsa.pub`
2. `id_dsa.pub`
3. `identity.pub`

`--generatekey={false|true}`

ユーザープロンプトなしで SSH キーファイルを作成するかどうかを指定します。

型：Boolean

次の値が有効です。

- `true`

ユーザープロンプトなしで SSH キーファイルを作成します。

- `false`

動作は、SSH キーファイルの有無によって決まります。

SSH キーファイルが存在する場合、ファイルを作成しません。

SSH キーファイルが存在しない場合、動作は、`asadmin` ユーティリティーの `--interactive` オプションの値によって決まります。

`--interactive` オプションが `true` の場合、ユーザーにファイルを作成するよう促します。

`--interactive` オプションが `false` の場合、このサブコマンドは失敗します。

デフォルト値：`false`

`host-list`

SSH キーをセットアップするホストの名称がスペースで区切られたリストを作成します。

型：String

指定できる値を次に示します。

- リモートホスト名

デフォルト値：なし

## 使用例

次の例では、ホスト `sj03` および `sj04` 上で、ユーザー `gfuser` の SSH キーをセットアップします。キーファイルは作成されませんが、サブコマンドが実行中のホストにあるユーザーの `.ssh` ディレクトリーからコピーされます。

```
asadmin setup-ssh sj03 sj04
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.10 update-node-config

ノード構成データの更新

### 書式

```
asadmin [asadmin-options] update-node-config [--help]
  [--nodehost node-host] [--installdir as-install-parent]
  [--nodedir node-dir] node-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

update-node-config サブコマンドは、ノードの構成データを更新します。このサブコマンドは、ノードがリモート通信に対して有効になっているかどうかに関係なく、任意のノードを更新できます。リモート通信のために有効化されたノードを更新した場合、更新後のノードはリモート通信に対して無効になります。

このサブコマンドでは、ノードを更新するために、SSH を設定する必要がありません。ドメイン管理サーバにアクセスできる任意のマシンから、このサブコマンドを実行できます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ノードが存在している必要があります。

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--nodehost *node-host*

ノードの更新後にノードが表すホストの名称を指定します。

型：String

指定できる値を次に示します。

- ホスト名

デフォルト値：なし

--installdir *as-install-parent*

*Application Server*インストールディレクトリー/*javaee* の絶対パスを指定します。

型：String

指定できる値を次に示します。

- *Application Server*インストールディレクトリー/*javaee* の絶対パス

デフォルト値：なし

#### `--nodedir node-dir`

ノード上に作成されるサーバインスタンスを入れるディレクトリーのパスを指定します。相対パスを指定する場合、パスは*Application Server*インストールディレクトリー/*javaee/glassfish* ディレクトリーからの相対パスであり、*Application Server*インストールディレクトリー/*javaee/glassfish* はホスト上の *Application Server* のインストールディレクトリーです。

型：String

指定できる値を次に示します。

- ノードのサーバインスタンスを含むディレクトリーへのパス

デフォルト値：なし

#### `node-name`

更新するノードの名称を指定します。

型：String

指定できる値を次に示します。

- ノード名

デフォルト値：なし

## 使用例

次の例では、ノード `sj04` が表すホストを `hsj04` に更新します。

```
asadmin update-node-config --nodehost hsj04 sj04
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.5.11 update-node-ssh

SSH を使ったノード構成データの更新



## 書式

```
asadmin [asadmin-options] update-node-ssh [--help]
  [--nodehost node-host] [--installdir as-install-parent]
  [--nodedir node-dir] [--sshport ssh-port]
  [--sshuser ssh-user] [--sshkeyfile ssh-keyfile]
  [--force={false|true}] node-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

update-node-ssh サブコマンドは、ノードの構成データを更新します。

このサブコマンドでは、ドメイン管理サーバが稼働しているマシンおよびノードが存在するマシンで、SSHを設定する必要があります。

ドメイン管理サーバにアクセスできる任意のマシンから、このサブコマンドを実行できます。

このサブコマンドは、ノードがリモート通信に対して有効になっているかどうかに関係なく、任意のノードを更新できます。

ノードがリモート通信に対して有効になっていない場合、このサブコマンドはノードのSSH通信を有効にし、そのほかの指定された構成データを更新します。ノードのSSH通信を有効にするためにこのサブコマンドを実行する場合、--sshport、--sshuser および--sshkeyfile が省略されたときには、デフォルト値が適用されます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

- ドメイン管理サーバが稼働しているホストおよびノードが表すホストでSSHが設定されている必要があります。
- 更新するホストが存在している必要があります。
- ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--nodehost *node-host*

ノードの更新後にノードが表すホストの名称を指定します。

型：String

指定できる値を次に示します。

- ホスト名

デフォルト値：なし

`--installdir as-install-parent`

*Application Server* インストールディレクトリー/*javaee* の絶対パスを指定します。

型：String

指定できる値を次に示します。

- *Application Server* インストールディレクトリー/*javaee* の絶対パス

デフォルト値：なし

`--nodedir node-dir`

ノード上に作成されるサーバインスタンスを含むディレクトリーのパスを指定します。相対パスを指定する場合、パスは*Application Server* インストールディレクトリー/*javaee/glassfish* ディレクトリーからの相対パスであり、*Application Server* インストールディレクトリー/*javaee/glassfish* はホスト上の *Application Server* のインストールディレクトリーです。

型：String

指定できる値を次に示します。

- ノードのサーバインスタンスを含むディレクトリーへのパス

デフォルト値：なし

`--sshport ssh-port`

このノードのホストに対する SSH 接続に使用するポートを指定します。

デフォルトは、このサブコマンドをノードの SSH 通信を有効にするために実行するかどうかによって異なります。

ノードが SSH 経由の通信に対してすでに有効になっている場合、デフォルトではポートは変更されません。

ノードの SSH 通信を有効にするためにこのサブコマンドを実行する場合、デフォルトポートは 22 です。--nodehost オプションを *localhost* に設定した場合、--sshport オプションは無視されます。

型：Integer

指定できる値を次に示します。

- 1-65535

デフォルト値：22

`--sshuser ssh-user`

SSH 経由でホストに接続する処理を実行するためのこのノードのホスト上のユーザーを指定します。

デフォルトは、このサブコマンドをノードの SSH 通信を有効にするために実行するかどうかによって異なります。

1. ノードが SSH 経由の通信に対してすでに有効になっている場合、デフォルトではユーザーは変更されません。

2. ノードの SSH 通信を有効にするためにこのサブコマンドを実行する場合、デフォルトはドメイン管理サーバのプロセスを実行するユーザーです。--nodehost オプションを localhost に設定した場合、--sshuser オプションは無視されます。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：ドメイン管理サーバのプロセスを実行しているユーザー

#### --sshkeyfile *ssh-keyfile*

--sshuser オプションで指定するユーザーの SSH 秘密鍵ファイルの絶対パスを指定します。このファイルはノードのホスト上の sshd デーモンに対する認証に使用されます。

SSH 秘密鍵ファイルがパスフレーズで保護されている場合、パスワードファイルに AS\_ADMIN\_SSHKEYPASSPHRASE エントリーを含める必要があります。鍵ファイルは、ドメイン管理サーバから到達できるパスにあり、ドメイン管理サーバが読み取りする必要があります。

デフォルトは、このサブコマンドをノードの SSH 通信を有効にするために実行するかどうかによって異なります。

ノードが SSH 経由の通信に対してすでに有効になっている場合、デフォルトでは鍵ファイルは変更されません。

ノードの SSH 通信を有効にするためにこのサブコマンドを実行する場合、デフォルトはユーザーの .ssh ディレクトリーにある鍵ファイルです。複数の鍵ファイルが見つかった場合、サブコマンドは id\_rsa、id\_dsa、identity の順序で使用します。

型：String

指定できる値を次に示します。

- ユーザーの SSH 秘密鍵ファイルを含むディレクトリーへのパス

デフォルト値：SSH 通信を有効にするためにこのサブコマンドを実行する場合、ユーザーの .ssh ディレクトリーにある鍵ファイル

#### --force={false|true}

ノードのパラメーターの検証に失敗した場合でも、ノードを更新するかどうかを指定します。ノードのパラメーターを検証するには、ドメイン管理サーバが SSH 経由でノードのホストにアクセスする必要があります。

型：Boolean

指定できる値を次に示します。

- false  
ノードのパラメーターの検証に失敗した場合、ノードは更新されません。
- true  
ノードのパラメーターの検証に失敗した場合でも、ノードが更新されます。

デフォルト値：false

### *node-name*

更新するノードの名称を指定します。

型：String

指定できる値を次に示します。

- ノード名

デフォルト値：なし

## 使用例

次の例では、ノード `lssh` が表すホストを `sj04` に更新します。

```
asadmin update-node-ssh --nodehost sj04 lssh
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6 サーバの管理で使用するコマンド

---

サーバの管理で使用するコマンドの書式、機能などについて説明します。

### 2.6.1 cleanup-prf

PRF 環境のクリーンアップ

#### 書式

```
asadmin [asadmin-options] cleanup-prf [--nodedir node-dir]  
      [--node node-name] prf-name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

cleanup-prf サブコマンドは、サブコマンドを実行したホスト上に構築されている PRF を停止し、PRF 環境を削除（クリーンアップ）します。ドメインの PRF 情報は削除しません。

このコマンドは、ドメイン管理サーバの停止によってドメインを再作成したために、リモートのノード上に PRF 環境が残っている場合など、ドメイン管理サーバで管理されていない PRF に対して使用してください。

このコマンドはローカルモードだけサポートします。

#### 実行権限

一般ユーザー

#### 前提条件

対象のノード上の PRF がドメイン管理サーバで管理されていないこと。

#### 引数

--nodedir *node-dir*

PRF が構築されているノードディレクトリーを指定します。

型：String

指定できる値を次に示します。

- 存在するノードディレクトリーのパス

デフォルト値：*Application Server*インストールディレクトリー/*javaee/glassfish/nodes*

`--node node-name`

PRF が構築されているノード名を指定します。

型：String

指定できる値を次に示します。

- 存在するノード名

デフォルト値：なし

`--nodedir` で指定したディレクトリーに存在するノードディレクトリーが 1 つだけの場合は、このオプションを省略できます。

*prf-name*

削除する PRF のサーバ名を指定します。

型：String

指定できる値を次に示します。

- 存在する PRF のサーバ名

デフォルト値：なし（必須）

## 使用例

*/temp/nodes* に構築した *localhost-domain1* のノードに属する PRF1 という名称の PRF を削除する場合

```
asadmin cleanup-prf --nodedir /temp/nodes --node localhost-domain1 PRF1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 注意事項

- `stop-prf` サブコマンドで PRF を停止できる環境では、このコマンドを使用しないでください。使用した場合、ドメイン管理サーバが PRF の障害として検知する場合があります。
- ドメイン管理サーバから PRF を削除できる場合は、このコマンドを使用しないでください。使用してドメインに PRF の情報が残ったときは、`delete-prf` サブコマンドの `--force` オプションを指定して削除します。また、ドメインに PRF の情報が残った状態で `start-prf` を実行すると、ノード上に PRF の情報が作成されることがあります。

## 2.6.2 cleanup-webserver

Web サーバ環境のクリーンアップ

### 書式

```
asadmin [asadmin-options] cleanup-webserver [--graceful={false|true}]  
        [--nodedir node-dir] [--node node-name] webserver-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`cleanup-webserver` サブコマンドは、サブコマンドを実行したホスト上に構築されている Web サーバを停止し、Web サーバ環境を削除（クリーンアップ）します。ドメインの Web サーバ情報は削除しません。

このコマンドは、ドメイン管理サーバの停止によってドメインを再作成したために、リモートのノード上に Web サーバ環境が残っている場合など、ドメイン管理サーバで管理されていない Web サーバに対して使用してください。

このコマンドはローカルモードだけサポートします。

### 実行権限

一般ユーザー

### 前提条件

対象のノード上の Web サーバがドメイン管理サーバで管理されていないこと。

### 引数

`--graceful={false|true}`

Web サーバを計画停止するかどうかを指定します。プロセスが停止済みの場合、この指定は無視されます。

型：Boolean

指定できる値を次に示します。

- `true`  
計画停止します。
- `false`  
通常停止します。

デフォルト値：`true`

`--nodedir node-dir`

Web サーバが構築されているノードディレクトリーを指定します。

型：String

指定できる値を次に示します。

- 存在するノードディレクトリーのパス

デフォルト値：*Application Server* インストールディレクトリー/*javaee/glassfish/nodes*

`--node node-name`

Web サーバが構築されているノード名を指定します。

型：String

指定できる値を次に示します。

- 存在するノード名

デフォルト値：なし

`--nodedir` で指定したディレクトリーに存在するノードディレクトリーが 1 つだけの場合は、このオプションを省略できます。

*webserver-name*

削除する Web サーバのサーバ名を指定します。

型：String

指定できる値を次に示します。

- 存在する Web サーバのサーバ名

デフォルト値：なし (必須)

## 使用例

Web1 という名称の Web サーバをクリーンアップする場合

```
asadmin cleanup-webserver --nodedir /temp/nodes --node localhost-domain1 Web1
```

/temp/nodes に構築した localhost-domain1 のノードに属する Web1 という名称の Web サーバをクリーンアップする場合

```
asadmin cleanup-webserver --nodedir /temp/nodes --node localhost-domain1 Web1
```

Web1 という名称の Web サーバを計画停止してからクリーンアップする場合

```
asadmin cleanup-webserver --graceful true --nodedir /temp/nodes  
--node localhost-domain1 Web1
```



## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 注意事項

- `stop-webserver` サブコマンドで Web サーバを停止できる環境では、このコマンドを使用しないでください。使用した場合、ドメイン管理サーバが Web サーバの障害として検知する場合があります。
- ドメイン管理サーバから Web サーバを削除できる場合は、このコマンドを使用しないでください。使用してドメインに Web サーバの情報が残ったときは、`delete-webserver` サブコマンドの `--force` オプションを指定して削除します。また、ドメインに Web サーバの情報が残った状態で `start-webserver` を実行すると、ノード上に Web サーバの情報が作成されることがあります。

## 2.6.3 copy-config

既存の名前付き構成をコピーして、別の構成を作成

### 書式

```
asadmin [asadmin-options] copy-config [--help]
  [--systemproperties name=value[:name=value]...]
  source-configuration-name destination-configuration-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`copy-config` サブコマンドは、既存の構成をコピーしてドメイン管理サーバの構成に名前付き構成を作成します。新しい構成は、`--systemproperties` オプションで指定したプロパティ以外、コピーした構成と同じです。スタンドアロンのサーバインスタンスまたはスタンドアロンのクラスターの作成時には `default-config` 構成がコピーされます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

- 構成をコピーするドメイン管理サーバは、稼働中である必要があります。
- `copy-config` サブコマンドを実行する前に、コピー元の構成が存在している必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--systemproperties *name=value[:name=value]...*

--systemproperties オプションで任意の属性名と値のペアを指定できます。ここで設定したプロパティは構成の中のポート設定を上書きします。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

プロパティ名と *value* に指定できる値を次に示します。

**ASADMIN\_LISTENER\_PORT=*value***

このプロパティは、ドメイン管理サーバがインスタンスを管理する際に接続する HTTP または HTTPS ポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

**HTTP\_LISTENER\_PORT=*value***

このプロパティは、HTTP 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

**HTTP\_SSL\_LISTENER\_PORT=*value***

このプロパティは、HTTPS 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

**IIOP\_LISTENER\_PORT=*value***

このプロパティは、IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`IIOP_SSL_LISTENER_PORT=value`

このプロパティは、セキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`IIOP_SSL_MUTUALAUTH_PORT=value`

このプロパティは、クライアント認証を持つセキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JAVA_DEBUGGER_PORT=value`

このプロパティは、Java Platform Debugger Architecture (JPDA) (<http://java.sun.com/javase/technologies/core/toolsapis/jpda/>) デバッガーへの接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JMS_PROVIDER_PORT=value`

このプロパティは、JMS プロバイダーのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JMX_SYSTEM_CONNECTOR_PORT=value`

このプロパティは、JMX コネクターがリッスンするポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

### *source-configuration-name*

コピーする構成の名称を表示します。

型：String

指定できる値を次に示します。

- コピー元構成の名称

デフォルト値：なし

### *destination-configuration-name*

コピー元の構成をコピーして作成する構成の名称を表示します。

型：String

指定できる値を次に示します。

- 作成される構成の名称

*destination-configuration-name* の値は、次に示す条件を満たしている必要があります。

ASCII 文字だけで構成されていること。

文字、数字、またはアンダースコアで開始されていること。

次の文字だけを使用すること。

- 小文字
- 大文字
- 数字
- ハイフン
- ピリオド
- アンダースコア

ドメイン内で一意である必要があり、ほかの構成名、クラスター名、サーバインスタンスの名称、または別のノード名と重複できません。

domain、server、または Java EE Server が予約したその他のキーワードは使用できません。

デフォルト値：なし

## 使用例

次の例では、デフォルトの構成をコピーして、名前付きの構成 `clusterpresets-config` を作成します。

```
asadmin copy-config default-config clusterpresets-config
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.4 create-cluster

Java EE Server クラスターの作成

### 書式

```
asadmin [asadmin-options] create-cluster [--help]
  [--config config-name]
  [--systemproperties name=value[:name=value]...]
  [--properties name=value[:name=value]...]
  [--gmsenabled={true|false}] [--multicastport multicast-port]
  [--multicastaddress multicast-address]
  [--bindaddress bind-address] cluster-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-cluster` サブコマンドは、Java EE Server クラスターを作成します。

クラスターは、そのクラスターに追加するすべてのインスタンスの構成を定義する名前付き構成への参照を必要とします。

クラスターの構成がほかのクラスターやスタンドアロンのサーバインスタンズと共有されていないため、作成されるクラスターはスタンドアロンのクラスターです。

このサブコマンドはリモートモードだけサポートします。

### 実行権限

次に示す情報は、`--systemproperties` オプションだけに適用されます。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--config config-name`

クラスターが参照する名前付き構成を指定します。`--config` オプションを指定して、共有クラスターを作成します。このオプションを省略した場合、スタンドアロンのクラスターが作成されます。

型：String

指定できる値を次に示します。

- 構成名

デフォルト値：なし

`--systemproperties name=value[:name=value]...`

`--systemproperties` オプションで任意の属性名と値のペアを指定できます。ここで設定したプロパティは `default-config` 構成の中の設定を上書きします。

指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

プロパティ名と `value` に指定できる値を次に示します。

`ASADMIN_LISTENER_PORT=value`

このプロパティは、ドメイン管理サーバがインスタンスを管理する際に接続する HTTP または HTTPS ポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`HTTP_LISTENER_PORT=value`

このプロパティは、HTTP 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`HTTP_SSL_LISTENER_PORT=value`

このプロパティは、HTTPS 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`IIOP_LISTENER_PORT=value`

このプロパティは、IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`IIOP_SSL_LISTENER_PORT=value`

このプロパティは、セキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`IIOP_SSL_MUTUALAUTH_PORT=value`

このプロパティは、クライアント認証を持つセキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JAVA_DEBUGGER_PORT=value`

このプロパティは、Java Platform Debugger Architecture (JPDA) (<http://java.sun.com/javase/technologies/core/toolsapis/jpda/>) デバッガーへの接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JMS_PROVIDER_PORT=value`

このプロパティは、JMS プロバイダーのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

**JMX\_SYSTEM\_CONNECTOR\_PORT=value**

このプロパティは、JMX コネクターがリッスンするポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

**--properties name=value[:name=value]...**

クラスターのプロパティを定義します。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

プロパティ名と *value* に指定できる値を次に示します。

**GMS\_DISCOVERY\_URI\_LIST=value**

クラスターを検出するために使用されるクラスター内のサーバインスタンスの場所を一覧表示します。このプロパティは、メッセージをブロードキャストするためにグループ管理サービスでマルチキャストを使用しない場合にだけ必要です。

型：String

指定できる値を次に示します。

Uniform Resource Identifier (URI) のコンマ区切りリスト。各 URI はサーバインスタンスまたはドメイン管理サーバを特定する必要があります。複数のサーバインスタンスが同じホストで実行されている場合は、この形式が必須です。

リスト内の各 URI の形式は次のとおりです。

**scheme://host-name-or -IP-address:port**

- **scheme** は URI のスキームで、**tcp** です。
- **host-name-or -IP-address** はインスタンスが実行されているホストのホスト名または IP アドレスです。
- **port** はインスタンスがグループ管理サービスからのメッセージをリッスンするポートのポート番号です。システムプロパティ **GMS\_LISTENER\_PORT-cluster-name** をインスタンスに対して設定する必要があります。

ドメイン管理サーバまたはインスタンスが稼働しているホストの IP アドレスまたはホスト名のコンマ区切りリスト。リストには、IP アドレスとホスト名を混在させることができます。この形式は、クラスター化された 1 つのインスタンスが各ホストで実行されている場合にだけ使用できます。

**GMS\_LISTENER\_PORT** プロパティの値は、ドメイン内の各クラスターで一意である必要があります。

キーワード **generate**。この形式は、クラスター内の 1 つのインスタンスが各ホストで実行されていてドメイン管理サーバが別のホストで稼働している場合にだけ使用できます。同じホスト上の複数のインスタンスは、同じクラスターのメンバーにできません。**GMS\_LISTENER\_PORT** プロパティの値は、ドメイン内の各クラスターで一意である必要があります。

デフォルト値：なし



**GMS\_LISTENER\_PORT=value**

クラスターがグループ管理サービスからのメッセージをリッスンするポートのポート番号を指定します。デフォルト値はGMS\_LISTENER\_PORT-cluster-name システムプロパティへの参照です。デフォルトでは、このシステムプロパティは設定されていません。この状況では、グループ管理サービスはGMS\_TCPSTARTPORT プロパティおよびGMS\_TCPENDPORT プロパティで定義された範囲から空きポートを選びます。デフォルトでは、範囲は9090~9200です。

メッセージをブロードキャストするためにグループ管理サービスでマルチキャストを使用しない場合、GMS\_LISTENER\_PORT プロパティは、クラスター内のすべてのサーバインスタンスに対して有効なポート番号を指定する必要があります。

この要件を満たすデフォルト値を使用するには、システムプロパティを使用してインスタンスごとに個別にポート番号を設定します。

例えば、create-system-properties サブコマンドを使用して、GMS\_LISTENER\_PORT-cluster-name システムプロパティをドメイン管理サーバに対して作成します。次に、クラスター内の各インスタンスについて、GMS\_LISTENER\_PORT-cluster-name システムプロパティを、そのインスタンスがグループ管理サービスからのメッセージをリッスンするポート番号に設定します。クラスターのGMS\_LISTENER\_PORT プロパティのデフォルト値は、このシステムプロパティを参照します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

**GMS\_LOOPBACK=value**

インスタンスがクラスターにブロードキャストしたアプリケーションレベルのメッセージを、そのインスタンスが自身から受信できるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

インスタンスは自身からメッセージを受信できます。インスタンスがクラスター内の唯一のインスタンスである場合に、インスタンスをテストするときに、この設定を使用します。

- false

インスタンスは自身からメッセージを受信できません。

デフォルト値：false

**GMS\_MULTICAST\_TIME\_TO\_LIVE=value**

次のタイプのイベントのマルチキャストメッセージを反復処理または伝送する最大数を指定します。この数を超えるとメッセージは破棄されます。

- グループの検出
- メンバーのハートビート
- メンバーシップの変更

ドメイン管理サーバおよびクラスター化されたインスタンスがデプロイされているネットワークの構成に一致させるには、この値をできる限り小さく設定します。システムに設定できる最小値を判定するには、`validate-multicast` サブコマンドを使用します。値に0を指定すると、マルチキャストメッセージはブロードキャスト元のホストから送信されません。

値に1を指定すると、スイッチまたはルーターで接続された同じサブネット上のホスト間でメッセージがブロードキャストされなくなる場合があります。

デフォルト値の4を指定すると、ホストがスイッチまたはルーターで接続されているネットワークのすべてのクラスター構成のインスタンスにメッセージが正常にブロードキャストされます。

型：Integer

指定できる値を次に示します。

- 0～255

デフォルト値：4

`GMS_TCPENDPORT=value`

`GMS_LISTENER_PORT-cluster-name` システムプロパティが設定されていない場合に、グループ管理サービスが空きポートを選択する範囲の最大のポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：9200

`GMS_TCPSTARTPORT=value`

`GMS_LISTENER_PORT-cluster-name` システムプロパティが設定されていない場合に、グループ管理サービスが空きポートを選択する範囲の最小のポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：9090

`--gmsenabled={true|false}`

グループ管理サービスがクラスターに有効であるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

グループ管理サービスはクラスターに有効です。グループ管理サービスがクラスターに有効な場合、クラスターおよびドメイン管理サーバで、グループ管理サービスは各サーバインスタンスを始動します。ドメイン管理サーバは、このオプションが True に設定されている各クラスターに関与します。

- false

グループ管理サービスはクラスターに無効です。

デフォルト値：true

`--multicastaddress` *multicast-address*

グループ管理サービスがグループイベントをリッスンするアドレスを指定します。

型：String

指定できる値を次に示します。

- 224.0.0.0～239.255.255.255

デフォルト値：228.9.XX.YY

ただし、XX およびYY は、0～255 の範囲で自動生成される独立した値

`--multicastport` *multicast-port*

グループ管理サービスがグループイベントをリッスンする通信ポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 2048～49151

デフォルト値：指定範囲内で自動生成される値

`--bindaddress` *bind-address*

グループ管理サービスがバインドするネットワークインターフェースの IP アドレスを指定します。

型：String

デフォルト値：すべてのパブリックネットワークインターフェースの IP アドレス

*cluster-name*

クラスター名を指定します。

型：String

指定できる値を次に示します。

- クラスター名

デフォルト値：なし

## 使用例

次の例では、クラスターを作成します。

```
asadmin create-cluster
--systemproperties IIOP_SSL_LISTENER_PORT=1169
ltscluster
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.5 create-instance

サーバインスタンスの作成

### 書式

```
asadmin [asadmin-options] create-instance [--help] --node node-name
  [--config config-name | --cluster cluster-name]
  [--portbase=port-number] [--checkports={true|false}]
  [--systemproperties name=value[:name=value]...]
  [--prf prf-name] instance-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-instance` サブコマンドはサーバインスタンスを作成します。

このサブコマンドは、ドメイン管理サーバが動作中のホストや、インスタンスが存在する`--node` オプションで指定されたホストに、SSH が配置されている必要があります。

サーバインスタンスは、Java EE Server が動作している 1 つのノード上の 1 つの Java VM です。

ノードは、サーバインスタンスが存在するホストを定義します。Java VM は、Java EE との互換性が必要です。

このサブコマンドはリモートモードだけサポートします。

インスタンスが、ローカルホストを表すタイプ`CONFIG`のノードに存在している場合、SSH は必要ありません。

タイプ`CONFIG`のノードは、SSH のリモート通信に対して有効化されていません。

### 実行権限

次に示す情報は、`--systemproperties` オプションだけに適用されます。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--node node-name`

インスタンスが作成される対象のホストを定義するノード名を指定します。

型：String

指定できる値を次に示します。

- ノード名

デフォルト値：localhost-domain

`--config config-name`

インスタンスから参照される名前付き構成を指定します。名前がdefault-configまたはserver-configではない構成が必要です。

型：String

指定できる値を次に示します。

- 構成名

デフォルト値：なし

`--cluster cluster-name`

インスタンスが構成を継承する元のクラスターを指定します。

型：String

指定できる値を次に示します。

- クラスター名

デフォルト値：なし

`--portbase=port-number`

ポート割当を開始する番号を確認します。

型：Integer

指定できる値を次に示します。

- -8~65449

デフォルト値：なし

`--checkports={true|false}`

管理、HTTP、JMS、JMX、IIOP の各ポートが利用できることを確認するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--systemproperties name=value[:name=value]...`

`--systemproperties` オプションで任意の属性名と値のペアを指定できます。ここで設定したプロパティは構成の中のポート設定を上書きします。

指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。

プロパティ名と `value` に指定できる値を次に示します。

`ASADMIN_LISTENER_PORT=value`

このプロパティは、ドメイン管理サーバがインスタンスを管理する際に接続する HTTP または HTTPS ポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`HTTP_LISTENER_PORT=value`

このプロパティは、HTTP 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`HTTP_SSL_LISTENER_PORT=value`

このプロパティは、HTTPS 要求をリッスンするのに使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`IIOP_LISTENER_PORT=value`

このプロパティは、IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`IIOP_SSL_LISTENER_PORT=value`

このプロパティは、セキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`IIOP_SSL_MUTUALAUTH_PORT=value`

このプロパティは、クライアント認証を持つセキュアな IIOP 接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JAVA_DEBUGGER_PORT=value`

このプロパティは、Java Platform Debugger Architecture (JPDA) (<http://java.sun.com/javase/technologies/core/toolsapis/jpda/>) デバッガーへの接続に使用するポートのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JMS_PROVIDER_PORT=value`

このプロパティは、JMS プロバイダーのポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

`JMX_SYSTEM_CONNECTOR_PORT=value`

このプロパティは、JMX コネクターがリッスンするポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1～65535

デフォルト値：なし

同じホスト上で2つのクラスター構成のインスタンスがある場合、プリデファインドポートの設定は上書きされます。この状況では、1つのインスタンスのポート設定が優先される必要があります。両方のインスタンスで同じ構成を共有するからです。

#### `--prf prf-name`

Java EE サーバが使用する PRF のサーバ名を指定します。 `--node` オプションに指定したノードに存在する PRF を指定してください。

型：String

指定できる値を次に示します。

- PRF のサーバ名

デフォルト値：なし

#### `instance-name`

作成するインスタンスの名称を指定します。

型：String

指定できる値を次に示します。

- インスタンス名

デフォルト値：なし

## 使用例

次の例では、サーバインスタンスを作成します。

```
asadmin create-instance --node localhost-domain1 pmdsainst
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.6 create-prf

PRF の構築

### 書式

```
asadmin [asadmin-options] create-prf --node node-name  
  [--properties name=value[:name=value]....]  
  [--template prf-templatedir] prf-name
```



## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`create-prf` サブコマンドは、ドメインに PRF を構築します。

このコマンドはリモートモードだけサポートします。

## 実行権限

一般ユーザー

## 前提条件

ノードを作成済みであること。

## ファイル

PRF のサーバテンプレート (オプション指定可能)

ドメインルートディレクトリー/*ドメイン名*/server\_templates/prf

## 引数

`--node node-name`

PRF を構築するノードを指定します。

型: String

指定できる値を次に示します。

- ドメインに存在するノード名

デフォルト値: なし (必須)

`--properties name=value[:name=value]...`

PRF の標準プロパティーまたは拡張プロパティーを、プロパティー名と値のペアで指定します。指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティー名を複数指定した場合は、最後に指定したプロパティーの値が有効となります。

*name*

型: String

指定できる値を次に示します。

- PRF の標準プロパティーまたは拡張プロパティー(ex\_から始まるプロパティー)。

*value*

指定できる値を次に示します。

- name に指定したプロパティの指定範囲に従います。

デフォルト値：このオプションで指定していない標準プロパティについては、PRF のデフォルトコンフィグ（コンフィグ名default-prf-config）の値が適用されます。

#### `--template prf-templatedir`

PRF のサーバテンプレートが格納されているディレクトリーパスを、絶対パスで指定します。

型：String

デフォルト値：hitachi-prf-configs.hitachi-prf-config.default-prf-config.hitachi-manage-info.template-path 属性に設定されているパスが適用されます。

上記属性のデフォルト値を示します。

ドメインルートディレクトリー/ドメイン名/server\_templates/prf

#### `prf-name`

新規に構築する PRF サーバの名称を指定します。

型：String

指定できる値を次に示します。

- 31 文字以内の ASCII 文字であること。
- 名称は次の文字で構成すること。
  - 英小文字<a~z>
  - 英大文字<A~Z>
  - 数字<0~9>
  - アンダースコア<\_>
- 先頭文字は英大文字または英小文字を使用すること。ただし、次の文字列は大文字、小文字ともに先頭に使用できません。

TSC

CTM

- ドメインに存在する次の名称は指定できません。

ノード名

JavaEE サーバ名

Web サーバ名

PRF のサーバ名

クラスター名

サーバ間関連名

JavaEE サーバの構成名

- 次に示す名称はシステムで使用済みまたは予約語のため指定できません。

domain

server

default

server-config  
default-config  
default-webserver-config  
default-prf-config  
javaee  
webserver  
prf  
cluster  
redirect  
prf-relation  
HJES\_で始まる名称  
構築済み Web サーバ名-config  
構築済み PRF 名-config  
デフォルト値：なし（必須）

## 使用例

PRF1 という名称の PRF を構築する場合

```
asadmin create-prf --node localhost-domain1 PRF1
```

標準プロパティを指定して PRF を構築する場合

```
asadmin create-prf --node localhost-domain1 --properties trace-file-size=65536  
PRF1
```

サーバテンプレートのディレクトリーを指定して PRF を構築する場合

```
asadmin create-prf --node localhost-domain1 --template C:¥temp¥PRF1 PRF1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 注意事項

- 構築先ノードのホストに、ほかのドメインが PRF を構築済みの場合、その PRF と同じ名称の PRF を構築しないでください。

## 2.6.7 create-webserver

Web サーバの構築

### 書式

```
asadmin [asadmin-options] create-webserver --node node-name
  [--properties name=value[:name=value]...]
  [--template webserver-templatedir]
  [--prf prf-name] webserver-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-webserver` サブコマンドは、ドメインに Web サーバを構築します。

このコマンドはリモートモードだけサポートします。

### 実行権限

一般ユーザー

### 前提条件

ノードを作成済みであること。

### ファイル

Web サーバのサーバテンプレート (オプション指定可能)

ドメインルートディレクトリー/*ドメイン名*/server\_templates/webserver

### 引数

`--node node-name`

Web サーバを構築するノードを指定します。

型：String

指定できる値を次に示します。

- ドメインに存在するノード名

デフォルト値：なし (必須)

`--properties name=value[:name=value]...`

Web サーバの標準プロパティまたは拡張プロパティを、プロパティ名と値のペアで指定します。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

#### *name*

型：String

指定できる値を次に示します。

- Web サーバの標準プロパティまたは拡張プロパティ (ex\_から始まるプロパティ)。

デフォルト値：このオプションで指定していない標準プロパティについては、Web サーバのデフォルトコンフィグ (コンフィグ名default-webserver-config) の値が適用されます。

#### *value*

指定できる値を次に示します。

- *name* に指定したプロパティの指定範囲に従います。

#### `--template webserver-templatedir`

Web サーバのサーバテンプレートが格納されているディレクトリーパスを絶対パスで指定します。

型：String

デフォルト値：hitachi-webserver-configs.hitachi-webserver-config.default-webserver-config.hitachi-manage-info.template-path 属性に設定されているパスが適用されます。この属性のデフォルト値は次のパスです。

ドメインルートディレクトリー/ドメイン名/server\_templates/webserver

#### `--prf prf-name`

Web サーバが使用する PRF のサーバ名を指定します。

型：String

指定できる値を次に示します。

- `--node` に指定したノードに存在する PRF

デフォルト値：なし (指定しない)

#### *webserver-name*

新規に構築する Web サーバの名称を指定します。

型：String

指定できる値を次に示します。

- 128 文字以内の ASCII 文字であること。

- 名称は次の文字で構成すること。

英小文字<a~z>

英大文字<A~Z>

数字<0~9>

ハイフン<->

アンダースコア<\_>

- 先頭文字は英大文字または英小文字を使用すること。
- ドメインに存在する次の名称は指定できません。
  - ノード名
  - JavaEE サーバ名
  - Web サーバ名
  - PRF のサーバ名
  - クラスター名
  - サーバ間関連名
  - JavaEE サーバの構成名
- 次に示す名称はシステムで使用済みまたは予約語のため指定できません。

domain  
 server  
 default  
 server-config  
 default-config  
 default-webserver-config  
 default-prf-config  
 javaee  
 webserver  
 prf  
 cluster  
 redirect  
 prf-relation  
 HJES\_で始まる名称  
 構築済み Web サーバ名-config  
 構築済み PRF 名-config

デフォルト値：なし（必須）

## 使用例

サーバ名Web1、関連する PRFPRF1 の Web サーバを構築する場合

```
asadmin create-webserver --node localhost-domain1 --prf PRF1 Web1
```

デフォルトから変更したい標準プロパティを指定して Web サーバを構築する場合

```
asadmin create-webserver --node localhost-domain1 --properties  
listen-port=88:server-name=www.webserver.com --prf PRF1 Web1
```

サーバテンプレートを指定して Web サーバを構築する場合

```
asadmin create-webserver --node localhost-domain1 --template /user_template/Web1
--prf PRF1 Web1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 注意事項

- 構築先ノードのホストに、ほかのドメインが Web サーバを構築済みの場合、その Web サーバと同じ名称の Web サーバを構築しないでください。

## 2.6.8 delete-cluster

Java EE Server クラスターの削除

### 書式

```
asadmin [asadmin-options] delete-cluster [--help] cluster-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-cluster` サブコマンドは、Java EE Server クラスターを削除します。クラスターは、サーバインスタンスがクラスターに含まれていない場合にだけ削除できます。削除しようとするクラスターにインスタンスが含まれている場合は、クラスターを削除する前に、インスタンスを停止して削除してください。

クラスターの名前付き構成が自動的に生成された場合、かつ、ほかのクラスターまたはクラスター構成ではないサーバインスタンスがその構成を参照していない場合、クラスター削除時に構成も削除されます。

クラスターの構成が自動的に生成された場合、構成名は `cluster-name-config` になります。クラスターの名称は `cluster-name` に指定します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*cluster-name*

クラスターの名称を指定します。

型：String

指定できる値を次に示します。

- クラスター名

デフォルト値：なし

## 使用例

次の例では、Java EE Server クラスターの `adccluster` を削除します。

```
asadmin delete-cluster adccluster
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.9 delete-config

名前付き既存構成の削除

### 書式

```
asadmin [asadmin-options] delete-config [--help] configuration-name
```

### 格納先

*Application Server*インストールディレクトリー/`javaee/glassfish/bin`



## 機能

`delete-config` サブコマンドは、ドメイン管理サーバの構成から既存の名前付き構成を削除します。構成は、サーバインスタンスまたはクラスターから参照されていない場合にだけ削除できます。スタンドアロン構成は、その構成を参照するインスタンスまたはクラスターが削除されると自動的に削除されます。スタンドアロン構成を作成するためにコピーされる `default-config` 構成は削除できません。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

*configuration-name*

構成の名称を指定します。

型：String

指定できる値を次に示します。

- 構成名

デフォルト値：なし

## 使用例

次の例では、名前付きの構成 `pmdconfig` を削除します。

```
asadmin delete-config pmdconfig
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.10 delete-instance

サーバインスタンスの削除

## 書式

```
asadmin [asadmin-options] delete-instance [--help] instance-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-instance` サブコマンドは、サーバインスタンスを削除します。このサブコマンドの実行には、SSH が次に示すどちらかのホスト上で構成されている必要があります。

- ドメイン管理サーバが動作しているホスト。
- インスタンスが存在する `create-instance` コマンドの `--node` オプションで指定されたホスト。

ただし、削除対象となるインスタンスが `config` タイプのノード上に構築されている場合、SSH の設定は不要です。

ドメイン管理サーバにアクセスできるどのホストからでも、このサブコマンドは実行できます。

サブコマンドでは任意のサーバインスタンスを削除できます。インスタンスが作成された方法に関係ありません。

削除するインスタンスは、動作していないことが必要です。インスタンスが動作している場合は、エラーが発生します。

このサブコマンドでは、次の処理でインスタンスを削除します。

- ドメイン管理サーバ構成からインスタンスを削除する。
- ファイルシステムからインスタンスのファイルを削除する。

削除対象のインスタンスが、ノードディレクトリーを使用する唯一のインスタンスである場合、そのディレクトリーも削除されます。

スタンドアロンサーバインスタンスを削除する場合、インスタンスのスタンドアロン構成も削除されます。スタンドアロン構成とは、スタンドアロンサーバインスタンスが参照し、かつ、ほかのクラスターまたはクラスター化されていないインスタンスが参照していない、`instance-name-config` という名前付きの構成のことです。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*instance-name*

インスタンス名を指定します。

型：String

指定できる値を次に示します。

- インスタンス名

デフォルト値：なし

## 使用例

次の例では、サーバインスタンスの `pmdsainst` を削除します。

```
asadmin delete-instance pmdsainst
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.11 delete-prf

PRF の削除

### 書式

```
asadmin [asadmin-options] delete-prf [--force={false|true}] prf-name
```

### 格納先

*Application Server*インストールディレクトリー/`javaee/glassfish/bin`

### 機能

`delete-prf` サブコマンドは、指定した名称の PRF をドメインから削除します。PRF がほかのサーバに関連している場合（PRF が関連元である関連が存在する場合）は、その関連も削除します。

このコマンドはリモートモードだけサポートします。

## 実行権限

一般ユーザー

## 前提条件

PRF を構築済みであること。

PRF を停止済みであること。

## 引数

`--force={false|true}`

強制削除するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`

強制削除します。ノード上の PRF が停止状態以外の場合や、PRF の削除に失敗した場合でも、ドメインから PRF の情報を削除します。

- `false`

強制削除しません。ノード上の PRF の削除に失敗した場合、ドメインから PRF の情報を削除しないで、エラー終了します。

デフォルト値：`false`

*prf-name*

削除する PRF のサーバ名を指定します。

型：String

指定できる値を次に示します。

- ドメインに存在する PRF のサーバ名

デフォルト値：なし（必須）

## 使用例

PRF1 という名称の PRF を削除する場合

```
asadmin delete-prf PRF1
```

PRF を強制削除する場合

```
asadmin delete-prf --force true PRF1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.12 delete-webserver

Web サーバの削除

### 書式

```
asadmin [asadmin-options] delete-webserver [--force={false|true}]  
webserver-name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-webserver サブコマンドは、Web サーバをドメインから削除します。Web サーバがほかのサーバに関連している場合（Web サーバが関連元である関連が存在する場合）は、その関連も削除します。

このコマンドはリモートモードだけサポートします。

### 実行権限

一般ユーザー

### 前提条件

Web サーバを構築済みであること。

Web サーバを停止済みであること。

### 引数

--force={false|true}

Web サーバを強制削除するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

強制削除します。ノード上の Web サーバが停止状態以外の場合や、Web サーバの削除に失敗した場合でも、ドメインから Web サーバの情報を削除します。

- false

強制削除しません。ノード上の Web サーバの削除に失敗した場合、ドメインから Web サーバの情報を削除しないで、エラー終了します。

デフォルト値：false

#### *webservice-name*

削除する Web サーバのサーバ名を指定します。

型：String

指定できる値を次に示します。

- ドメインに存在する Web サーバ名

デフォルト値：なし（必須）

## 使用例

Web1 という名称の Web サーバを削除する場合

```
asadmin delete-webservice Web1
```

Web サーバを強制削除する場合

```
asadmin delete-webservice --force true Web1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.13 export-sync-bundle

クラスターまたはスタンドアロンサーバインスタンスの構成データのアーカイブファイルへのエクスポート

### 書式

```
asadmin [asadmin-options] export-sync-bundle [--help]
  --target target
  [--retrieve={false|true}]
  [file-name]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`export-sync-bundle` サブコマンドは、クラスターまたはスタンドアロンサーバインスタンスの構成データをアーカイブファイルにエクスポートします。このアーカイブファイルは、`import-sync-bundle` サブコマンドで構成データを復元するときに使用できます。

インスタンスの構成データをインポートすると、インスタンスがドメイン管理サーバと通信できる状態にしなくても、インスタンスのホストにデータが転送されます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

このサブコマンドでは、次のファイルとディレクトリーを含むアーカイブを現在のドメインディレクトリー内に作成します。

次のディレクトリーに含まれるすべてのファイル

- `config`
- `docroot`

次のディレクトリーとサブディレクトリーの全内容

- `applications`
- `config/target` (`target` は構成データをエクスポートするクラスターまたはスタンドアロンサーバインスタンス)
- `generated`
- `lib`

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

エクスポートする構成データの対象を指定します。

このオプションには、クラスター化したサーバインスタンスを指定できません。このオプションにクラスター化されたインスタンスを指定すると、エラーが発生します。クラスター化されたインスタンスの構成データをエクスポートするには、そのインスタンスではなく、インスタンスが所属するクラスターの名称を指定します。

型：String

指定できる値を次に示します。

- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *standalone\_instance\_name*  
指定した名称のスタンドアロンのサーバインスタンスを対象とします。

デフォルト値：なし

`--retrieve={false|true}`

ドメイン管理サーバのホストから、このサブコマンドが実行されるホストにアーカイブファイルをダウンロードするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
このサブコマンドが実行されるホストにアーカイブファイルがダウンロードされます。
- `false`  
アーカイブファイルはダウンロードされずに、ドメイン管理サーバのホストに保持されます。

デフォルト値：`false`

*file-name*

データをエクスポートするアーカイブファイルのファイル名と場所を指定します。

型：String

指定できる値を次に示します。

- ファイル名

デフォルト値：

1. `--retrieve` オプションが `false` の場合  
現在のドメインディレクトリー内の `sync/target-sync-bundle.zip`
2. `--retrieve` オプションが `true` の場合  
現在の作業ディレクトリー内の `target-sync-bundle.zip`

## 使用例

次の例では、`pmdcluster` という名称のクラスターの構成データをエクスポートします。



```
asadmin export-sync-bundle --target pmdcluster
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.14 get-health

クラスターの状態表示

### 書式

```
asadmin [asadmin-options] get-health [--help]  
      cluster_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

get-health サブコマンドは、クラスターの状態に関する情報を取得し、表示します。グループ管理サービスが有効ではない場合、このクラスターのサーバインスタンスが動作しているかどうかに関する基本情報が返されないことに注意してください。サーバインスタンスごとに、次のどれかの状態がレポートされます。

- not started
- started
- stopped
- rejoined
- failed

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*cluster\_name*

状態の情報を表示するクラスターの名称を指定します。クラスター名を指定しないと、クラスター名の入力を要求するプロンプトが表示されます。

クラスターはドメイン管理サーバに存在する必要があります。

型：String

指定できる値を次に示します。

- クラスター名

デフォルト値：なし

## 使用例

次の例では、クラスター内のサーバインスタンスの状態に関する情報を表示します。

```
asadmin get-health cluster1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.15 import-sync-bundle

クラスター化されたインスタンスまたはスタンドアロンサーバインスタンスの構成データのインポート

## 書式

```
asadmin [asadmin-options] import-sync-bundle [--help]
  --instance instance-name
  [--nodedir node-dir] [--node node-name]
  file-name
```

## 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

`import-sync-bundle` サブコマンドは、`export-sync-bundle` サブコマンドで作成されたアーカイブファイルから、クラスター化されたインスタンスまたはスタンドアロンサーバインスタンスの構成データをインポートします。インスタンスの構成データをインポートすると、インスタンスがドメイン管理サーバと通信できる状態でも、インスタンスのホストにデータを転送できます。

このサブコマンドはローカルモードだけサポートします。

このサブコマンドは、インスタンスが存在するホストで実行する必要があります。ドメイン管理サーバにアクセスするには、このサブコマンドでは、ドメイン管理サーバが稼働しているホストの名称が必要です。管理サーバでデフォルト以外のポートを使用している場合、ポート番号も指定する必要があります。

## 前提条件

`--instance` オプションでは、ドメイン管理サーバの設定内の既存のインスタンスを指定する必要があります。また、データのインポート元のアーカイブファイルには、指定したインスタンスのデータが含まれている必要があります。

## ファイル

`import-sync-bundle` サブコマンドは、クラスター化されたインスタンスまたはスタンドアロンサーバインスタンスの構成データを`export-sync-bundle` サブコマンドで作成したアーカイブファイル (`sync-bundle` アーカイブファイル) からインポートします。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--instance instance-name`

構成データをインポートする対象のインスタンスを指定します。ドメイン管理サーバ構成内の既存のインスタンスを指定する必要があります。また、データのインポート元のアーカイブファイルには、指定したインスタンスのデータが含まれている必要があります。

型：String

指定できる値を次に示します。

- インスタンス名

デフォルト値：なし

`--nodedir node-dir`

インスタンスのノードディレクトリーが含まれるディレクトリーを指定します。インスタンスのファイルは、インスタンスのノードディレクトリーに格納されます。

型：String

指定できる値を次に示します。

- ノードディレクトリーのパス

デフォルト値：*Application Server*インストールディレクトリー/*javaee/glassfish/nodes*

`--node node-name`

インスタンスが存在するノードを指定します。このオプションを省略した場合、アーカイブファイルのドメイン管理サーバ構成からノードが判断されます。

型：String

指定できる値を次に示します。

- ノード名

デフォルト値：ドメイン管理サーバ構成のノード

*file-name*

インポートするアーカイブファイルが含まれているファイル名（パスを含む）を指定します。

型：String

指定できる値を次に示します。

- パスを含むファイル名

デフォルト値：なし

## 使用例

次の例では、アーカイブファイル/*export/glassfish3/glassfish/domains/domain1/sync/ymlcluster-sync-bundle.zip* から、ノード*sj02* のクラスター化されたインスタンス*yml2* の構成をインポートします。

コマンドは、ノード*sj02* で表されるホスト*sj02* 上で実行されます。ドメイン管理サーバはホスト*sr04* 上で稼働していて、管理用のデフォルトの HTTP ポートを使用しています。

```
sj02# asadmin --host sr04 import-sync-bundle --node sj02 --instance yml2
/export/glassfish3/glassfish/domains/domain1/sync/ymlcluster-sync-bundle.zip
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.16 list-clusters

ドメイン内の既存クラスターの一覧表示

## 書式

```
asadmin [asadmin-options] list-clusters [--help]
        [target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-clusters` サブコマンドは、ドメイン内の既存のクラスターを一覧表示します。リストは、クラスター、インスタンス、ノードまたは構成でフィルター処理できます。表示された各クラスターに対し、サブコマンドはクラスターが実行中かどうかを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するクラスターの対象を指定します。

型：String

指定できる値を次に示します。

- *domain*  
ドメインを対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。
- *node\_name*  
指定した名称のノードに存在するインスタンスを含むクラスターを一覧表示します。例えば、クラスター *pmdc* にインスタンス *pmdi1*、およびクラスター *ymlc* にインスタンス *yml1* がノード *n1* に存在する場合、*pmdc* および *ymlc* が一覧表示されます。
- *configuration\_name*  
指定した名称の構成を対象とします。

デフォルト値：domain

## 使用例

次の例では、現在のドメイン内のクラスターをすべて一覧表示します。

```
asadmin list-clusters
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.17 list-instances

サーバインスタンスの一覧表示

### 書式

```
asadmin [asadmin-options] list-instances [--help]
  [--timeoutmsec timeout]
  [--long={false|true} | --nostatus={false|true}]
  [--standaloneonly={false|true} | target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-instances` サブコマンドは、ドメインにあるサーバインスタンスを一覧表示します。一覧は、クラスター、サーバインスタンス、ノードまたは構成でフィルター処理できます。

このサブコマンドは、サーバインスタンスを作成した方法に関係なく、指定した対象にあるすべてのサーバインスタンスを表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--timeoutmsec timeout`

サーバインスタンスのステータスを判定するタイムアウト（ミリ秒単位）を指定します。

型：Integer

指定できる値を次に示します。

- -2147483648～2147483647

デフォルト値：60000

`--long={false|true} | -l={false|true}`

一覧表示されたサーバインスタンスごとに詳細情報を表示するかどうかを指定します。

`--long` オプションと `--nostatus` オプションは同時に指定できません。両方のオプションを指定した場合、エラーが発生します。

型：Boolean

指定できる値を次に示します。

- `true`

一覧表示されたサーバインスタンスごとに、次の詳細情報が表示されます。

- サーバインスタンス名
- サーバインスタンスのノードが存在するホスト名
- サーバインスタンスが管理要求をリスニングする HTTP ポート
- サーバインスタンスのプロセスのプロセス ID (PID)、またはプロセスが動作していない場合は-1
- サーバインスタンスがクラスタのメンバーの場合はクラスタ名
- `running` または `not running` のどちらかのサーバインスタンス状態

サーバインスタンスが一覧表示された場合、サーバインスタンスに対するドメイン管理サーバの構成変更がサーバインスタンスに適用されないことがあります。このような状況の場合、変更を反映する必要があるコマンドがサーバインスタンス状態の隣に一覧表示されます。1つのサーバインスタンスに対して一覧表示されるコマンドの最大数は10です。

- `false`

サーバインスタンス名およびサーバインスタンスが実行中であるかどうかだけが表示されます（デフォルト）。サーバインスタンスの実行時間は表示されません。

デフォルト値：`false`

`--nostatus={false|true}`

サーバインスタンスの実行状態を表示させないようにするかどうかを指定します。

`--long` オプションと `--nostatus` オプションは同時に指定できません。両方のオプションを指定すると、エラーが発生します。

型：Boolean

指定できる値を次に示します。

- **true**

サーバインスタンスの実行状態を表示しません。各サーバインスタンスの名称だけが表示されます。

- **false**

サーバインスタンスの実行状態を表示します。

デフォルト値：**false**

**--standaloneonly={false|true}**

スタンドアロンサーバインスタンスだけを一覧表示するかどうかを指定します。

**--standaloneonly** オプションと *target* は同時に指定できません。同時に指定した場合はエラーが発生します。

型：**Boolean**

指定できる値を次に示します。

- **true**

スタンドアロンサーバインスタンスだけを一覧表示します。

- **false**

指定した対象のすべてのサーバインスタンスを一覧表示します。

デフォルト値：**false**

**target**

一覧表示するサーバインスタンスの対象を指定します。

**target** と **--standaloneonly** オプションは同時に指定できません。同時に指定した場合はエラーが発生します。

型：**String**

指定できる値を次に示します。

- **domain**

ドメインを対象とします。

- **cluster\_name**

指定した名称のクラスターを対象とします。

- **instance\_name**

指定した名称のサーバインスタンスを対象とします。

- **node\_name**

指定した名称のノードに存在するサーバインスタンスを対象とします。

- **configuration\_name**

指定した名称の構成を対象とします。

デフォルト値：**domain**



## 使用例

次の例では、現在のドメインにあるすべてのサーバインスタンスに関する詳細情報を一覧表示します。

```
asadmin list-instances --long=true
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.18 list-prfs

PRF の一覧表示

### 書式

```
asadmin [asadmin-options] list-prfs [--long={false|true} |  
--nostatus={false|true}] [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-prfs サブコマンドは、ドメインに構築された PRF を一覧表示します。PRF の構成情報や稼働状態を確認できます。

このコマンドはリモートモードだけサポートします。

### 実行権限

一般ユーザー

### 引数

--long={false|true}

一覧表示する PRF について、詳細情報を表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

PRF のサーバ名とその詳細情報（ホスト名、プロセス ID、ステータス）を表示します。ヘッダー情報も表示します。

- `false`

PRF のサーバ名とステータスだけを表示します。ヘッダー情報は表示しません。

デフォルト値：`false`

`--nostatus={false|true}`

PRF のステータスを表示するかどうかを指定します。

型：`Boolean`

指定できる値を次に示します。

- `true`

ステータスを表示しません。

- `false`

ステータスを表示します。

デフォルト値：`false`

*target*

情報を表示する対象名を指定します。指定された名称から表示する PRF をフィルタリングします。

型：`String`

指定できる値を次に示します。

- `domain`

ドメイン内のすべての PRF の情報を表示します。

- *PRF のサーバ名*

指定した PRF の情報を表示します。

- *ノード名*

指定したノードに属する PRF の情報を表示します。

デフォルト値：`domain`

## 出力形式

1 行目は出力情報のヘッダー部となり、固定文字列で出力形式を示します。

カラム間は、カラム内で最も長い文字列と次のカラムの先頭までを、2 つの半角スペースで区切ります。

ステータスは、半角スペース 1 つを先頭に付けて表示します。

NAME	HOST	PID	STATE
<i>PRF のサーバ名</i>	<i>ホスト名</i>	<i>プロセスID</i>	<i>ステータス</i>

*PRF のサーバ名*：PRF の名称を表示します。

ホスト名：PRF が構築されているホスト名を表示します。

プロセス ID：稼働中の PRF のプロセス ID を表示します。PRF が稼働していない場合は、--を表示します。

ステータス：PRF のステータスを表示します。表示されるステータスを次に示します。

- not running (停止)
- running (稼働)

## 使用例

PRF のサーバ名とステータスだけを表示する場合

```
asadmin list-prfs
```

[出力例]

```
PRF1    running
PRF2    not running
PRF4    running
PRF3    not running
```

PRF のすべての情報を表示する場合

```
asadmin list-prfs --long true
```

[出力例]

```
NAME  HOST      PID  STATE
PRF1  localhost 258  running
PRF2  localhost --    not running
PRF4  WINHOST1  547  running
PRF3  WINHOST1  --   not running
```

PRF のステータスを表示しない場合

```
asadmin list-prfs --nostatus true
```

[出力例]

```
PRF1
PRF2
PRF4
PRF3
```

PRF1 の情報だけを表示する場合

```
asadmin list-prfs PRF1
```

[出力例]

```
PRF1    running
```

ノードwinhost1-domain2（ホストWINHOST1）に属する PRF の情報を表示する場合

```
asadmin list-prfs --long true winhost1-domain2
```

[出力例]

```
NAME  HOST      PID  STATE
PRF4  WINHOST1  547  running
PRF3  WINHOST1  --   not running
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.19 list-webservers

Web サーバの一覧表示

### 書式

```
asadmin [asadmin-options] list-webservers [--long={false|true} |  
--nostatus={false|true}] [target]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-webservers サブコマンドは、ドメインに構築された Web サーバを一覧表示します。Web サーバの構成情報や稼働状態を確認できます。

このコマンドはリモートモードだけサポートします。

--long と--nostatus は形式上、同時に指定できませんが、両方またはどちらか一方を明示的にfalse に設定する場合は、同時指定できます。

## 実行権限

一般ユーザー

## 引数

`--long={false|true}`

一覧表示する Web サーバについて、詳細情報を表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
Web サーバ名とその詳細情報（ホスト名、プロセス ID、ステータス）を表示します。ヘッダー情報も表示します。
- `false`  
Web サーバ名とステータスだけを表示します。ヘッダー情報は表示しません。

デフォルト値：`false`

`--nostatus={false|true}`

Web サーバのステータスを表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
ステータスを表示しません。
- `false`  
ステータスを表示します。

デフォルト値：`false`

*target*

情報を表示する対象名を指定します。指定された名称から表示する Web サーバをフィルタリングします。

型：String

指定できる値を次に示します。

- `domain`  
ドメイン内のすべての Web サーバの情報を表示します。
- *Web サーバ名*  
指定した Web サーバの情報を表示します。
- *ノード名*  
指定したノードに属する Web サーバの情報を表示します。

デフォルト値：domain

## 出力形式

1 行目は出力情報のヘッダー部となり、固定文字列で出力形式を示します。

カラム間は、カラム内で最も長い文字列と次のカラムの先頭までを、2つの半角スペースで区切ります。

ステータスは、半角スペース1つを先頭に付けて表示します。

NAME	HOST	PID	STATE
Webサーバ名	ホスト名	プロセスID	ステータス

Webサーバ名：Webサーバの名称を表示します。

ホスト名：Webサーバが構築されているホスト名を表示します。

プロセスID：稼働中のWebサーバのプロセスIDを表示します。Webサーバが稼働していない場合は、--を表示します。

ステータス：Webサーバのステータスを表示します。表示されるステータスを次に示します。

- not running（停止）
- running（稼働）

## 使用例

Webサーバ名とステータスだけを表示する場合

```
asadmin list-webservers
```

[出力例]

```
Web1 running
Web2 not running
Web4 running
Web3 not running
```

Webサーバのすべての情報を表示する場合

```
asadmin list-webservers --long true
```

[出力例]

NAME	HOST	PID	STATE
Web1	localhost	258	running
Web2	localhost	--	not running

```
Web4 WINHOST1 526 running
Web3 WINHOST1 -- not running
```

Web サーバのステータスを表示しない場合

```
asadmin list-webservers --nostatus true
```

[出力例]

```
Web1
Web2
Web4
Web3
```

Web1 の情報だけを表示する場合

```
asadmin list-webservers Web1
```

[出力例]

```
Web1 running
```

ノードwinhost1-domain2（ホストWINHOST1）に属する Web サーバの情報を表示する場合

```
asadmin list-webservers winhost1-domain2
```

[出力例]

```
Web4 running
Web3 not running
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.20 start-cluster

クラスターの起動

## 書式

```
asadmin [asadmin-options] start-cluster [--help]
        [--verbose={false|true}] cluster-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

start-cluster サブコマンドは、クラスター内でまだ実行されていないすべてのサーバインスタンスを起動します。

このサブコマンドはリモートモードだけサポートします。

なお、このサブコマンドには、起動のタイムアウトがあります。起動のタイムアウトまたは asadmin リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、起動処理は継続します。起動に成功したかどうかは、KDKD20031-I のログまたは GUI のサーバインスタンスの状態により確認できます。

## 前提条件

ドメイン管理サーバが稼働しているホストおよびクラスター内のインスタンスが存在するすべてのホストで、SSH を設定する必要があります。ただし、クラスター内のインスタンスがすべて config タイプのノード上に構築されている場合、SSH の設定は不要です。

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--verbose={false|true}

クラスターの起動時に、追加のステータス情報を表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

クラスター内の各インスタンスを起動するコマンドおよび各インスタンスの起動が成功したかどうかを表示します。

- false

追加のステータス情報を表示しません。

デフォルト値：false



*cluster-name*

起動するクラスターの名称を指定します。

型：String

指定できる値を次に示します。

- クラスター名

デフォルト値：なし

## 使用例

次の例では、クラスター `ymlcluster` を起動します。クラスターの起動時に、追加のステータス情報が表示されます。

```
asadmin start-cluster --verbose ymlcluster
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.21 start-instance

サーバインスタンスの起動

### 書式

```
asadmin [asadmin-options] start-instance [--help]  
        [--debug={false|true}] [--sync={normal|full|none}] instance-name
```

### 格納先

*Application Server*インストールディレクトリー/`javaee/glassfish/bin`

### 機能

`start-instance` サブコマンドは、サーバインスタンスを起動します。ドメイン管理サーバにアクセスできる任意のマシンから、このサブコマンドを実行できます。このサブコマンドは、任意のサーバインスタンスを起動できます。インスタンスが作成された方法はありません。

このサブコマンドはリモートモードだけサポートします。

ノードルートディレクトリー/ノード名/Java EEサーバ名/config に出力される次のファイルは、ログ識別子をデフォルトから変更した場合やstart-instance コマンドに"--sync=full"オプションを指定した場合、Java EE サーバ再起動のときに削除されます。必要に応じてこのサブコマンドの実行前に退避しておいてください。

- サーバインスタンスのスレッドダンプログ (デフォルトのログ識別子 "javacore")
- メモリーダンプ (デフォルトのログ識別子 "core"、".core"、"\*.dmp")
- エラーレポートファイル (デフォルトのログ識別子 "hs\_err\_pid")
- コンパイラーリプレイファイル (デフォルトのログ識別子 "replay\_pid")

なお、このサブコマンドには、起動のタイムアウトがあります。ドメイン管理サーバのリクエストタイムアウトが発生した場合、起動処理は継続します。起動のタイムアウトまたは asadmin リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、起動処理は継続します。起動に成功したかどうかは、KDKD20031-I のログまたは GUI のサーバインスタンスの状態により確認できます。

## 前提条件

インスタンスが存在するリモートのノードがリモート通信に対して有効になっている必要があります。

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--debug={false|true}

Java Platform Debugger Architecture (JPDA) デバッグを有効にしてインスタンスを起動するかどうかを指定します。

型: Boolean

指定できる値を次に示します。

- true

JPDA デバッグを有効にしてインスタンスを起動し、JPDA デバッグのポート番号を表示します。

- false

JPDA デバッグを無効にしてインスタンスを起動します。

デフォルト値: false

--sync={normal|full|none}

インスタンス起動時のドメイン管理サーバとインスタンスのファイル間の同期のタイプを指定します。

型: String

指定できる値を次に示します。

- none

ドメイン管理サーバはインスタンスのファイルと変更を同期しません。このタイプの同期では、インスタンスを起動するのに必要な時間が最小限に抑えられます。

- **normal**

ドメイン管理サーバは次のようにインスタンスと最後の同期以降の変更を同期します。

`config` ディレクトリーの場合、ドメイン管理サーバはインスタンスとすべての変更を同期します。  
`applications` ディレクトリーおよび `docroot` ディレクトリーの場合、最上位のサブディレクトリーに対する変更があった場合だけ、ドメイン管理サーバはそのサブディレクトリー下のすべてのファイルを同期します。

最上位のサブディレクトリー内のファイルには変更がなく、その最上位サブディレクトリーの下のファイルに変更があった場合、完全な同期が必要です。通常の操作では、これらのディレクトリーの最上位のサブディレクトリーの下のファイルは変更されません。アプリケーションのデプロイ後にアンデプロイする場合、変更に合わせてインスタンスを更新するための完全な同期は不要です。

- **full**

ドメイン管理サーバは、インスタンスとすべてのインスタンスファイルを同期します。最後の同期以降ファイルが変更されているかどうかは関係ありません。このタイプの同期では、ドメイン管理サーバがインスタンスのディレクトリー内のすべてのファイルを更新している間、インスタンスの起動が遅くなることがあります。

デフォルト値：normal

#### *instance-name*

起動するサーバインスタンスの名称を指定します。

型：String

指定できる値を次に示します。

- サーバインスタンス名

デフォルト値：なし

## 使用例

次の例では、サーバインスタンスの `pmdsa1` を起動します。

```
asadmin start-instance pmdsa1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.22 start-prf

PRF の起動

### 書式

```
asadmin [asadmin-options] start-prf prf-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

start-prf サブコマンドは、指定した名称の PRF を起動します。

このコマンドはリモートモードだけサポートします。

なお、このサブコマンドには、起動のタイムアウトがあります。ドメイン管理サーバのリクエストタイムアウトが発生した場合、起動処理は継続します。起動のタイムアウトまたは asadmin リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、起動処理は継続します。起動に成功したかどうかは、list-prfs サブコマンドまたは GUI の PRF の状態により確認できます。

### 実行権限

一般ユーザー

### 前提条件

PRF を構築済みであること。

### 引数

*prf-name*

起動する PRF のサーバ名を指定します。

型：String

指定できる値を次に示します。

- ドメインに存在する PRF のサーバ名

デフォルト値：なし（必須）

### 使用例

PRF1 という名称の prf を起動する場合

```
asadmin start-prf PRF1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.23 start-servers

サーバの一括起動

### 書式

```
asadmin [asadmin-options] start-servers
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

start-servers サブコマンドは、ドメイン内に存在する各種サーバを一括で起動します。サーバ間に関連（依存関係）が設定されている場合は、その依存先のサーバを先に起動します。依存関係が多い場合は、依存先の先端に当たるサーバから順に起動します。サーバ間に関連（依存関係）が設定されていない場合は、すべてのサーバに対し、並行して処理します。

すでに稼働状態のサーバに対しては、起動処理を実行しません。

このコマンドはリモートモードだけサポートします。

ノードルートディレクトリー/*ノード名/Java EEサーバ名/config*に出力される次のファイルは、ログ識別子をデフォルトから変更した場合やstart-instance コマンドに"--sync=full"オプションを指定した場合、Java EE サーバ再起動のときに削除されます。必要に応じてこのサブコマンドの実行前に退避しておいてください。

- サーバインスタンスのスレッドダンプログ（デフォルトのログ識別子 "javacore"）
- メモリーダンプ（デフォルトのログ識別子 "core"、".core"、".dmp"）
- エラーレポートファイル（デフォルトのログ識別子 "hs\_err\_pid"）
- コンパイラーリプレイファイル（デフォルトのログ識別子 "replay\_pid"）

### 実行権限

一般ユーザー

## 前提条件

各種サーバを構築済みであること。

## 使用例

ドメイン内のすべてのサーバを起動する場合

```
asadmin start-servers
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.24 start-webserver

Web サーバの起動

### 書式

```
asadmin [asadmin-options] start-webserver webserver-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`start-webserver` サブコマンドは、指定した名称の Web サーバを起動します。

このコマンドはリモートモードだけサポートします。

なお、このサブコマンドには、起動のタイムアウトがあります。ドメイン管理サーバのリクエストタイムアウトが発生した場合、起動処理は継続します。起動のタイムアウトまたは `asadmin` リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、起動処理は継続します。起動に成功したかどうかは、`list-webservers` サブコマンドまたは GUI の Web サーバの状態により確認できます。

### 実行権限

一般ユーザー

## 前提条件

Web サーバを構築済みであること。

## 引数

### *webserver-name*

起動する Web サーバのサーバ名を指定します。

型：String

指定できる値を次に示します。

- ドメインに存在する Web サーバ名

デフォルト値：なし（必須）

## 使用例

Web1 という名称の Web サーバを起動する場合

```
asadmin start-webserver Web1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.25 stop-cluster

Java EE Server のクラスターの停止

## 書式

```
asadmin [asadmin-options] stop-cluster [--help]  
        [--verbose={false|true}] [--kill={false|true}] cluster-name
```

## 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`stop-cluster` サブコマンドは、クラスター内で実行されているすべてのサーバインスタンスを停止することによって、そのクラスターを停止します。

このサブコマンドはリモートモードだけサポートします。

なお、このサブコマンドには、停止のタイムアウトがあります。停止のタイムアウトまたは `asadmin` リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、停止処理は継続します。停止に成功したかどうかは、KDKD20032-I のログまたは GUI のサーバインスタンスの状態により確認できます。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--verbose={false|true}`

クラスターを停止するときに、クラスター内のインスタンスの状態に関する追加の進捗メッセージを表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
クラスター内のインスタンスの状態に関する追加の進捗メッセージが表示されます。
- `false`  
クラスター内のインスタンスの状態に関するメッセージは表示されません。

デフォルト値：`false`

`--kill={false|true}`

サーバインスタンスのプロセスを停止する OS の機能を使用して、クラスター内の各インスタンスを強制終了するかどうかを指定します。

型：String

指定できる値を次に示します。

- `false`  
インスタンスを強制終了しません。Java プラットフォームの機能を使用して各サーバインスタンスのプロセスを停止します。
- `true`  
各インスタンスを強制終了します。OS の機能を使用して各サーバインスタンスのプロセスを停止します。

デフォルト値：`false`

*cluster-name*

停止するクラスターの名称を指定します。



型：String

指定できる値を次に示します。

- クラスター名

デフォルト値：なし

## 使用例

次の例では、クラスター `pmdcluster` を停止します。

```
asadmin stop-cluster pmdcluster
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.26 stop-instance

実行中のサーバインスタンスの停止

### 書式

```
asadmin [asadmin-options] stop-instance [--help]  
        [--kill={false|true}] instance-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`stop-instance` サブコマンドは、実行中のサーバインスタンスを停止します。任意のサーバインスタンスを停止できます。インスタンスが作成された方法はありません。

このサブコマンドはリモートモードだけサポートします。

サーバ停止時に、標準エラーおよび `logger.log` ファイルに `KDKD45000-E` のメッセージを出力することがありますが、終了処理に影響はありません。

サーバインスタンスを停止させる場合、ロードバランサー、Web サーバなど、フロントエンドのサーバから先に停止させてください。フロントエンドのサーバを先に停止させないで、サーバインスタンス内にリクエストが滞留している場合、予期しない例外が発生する場合があります。

なお、このサブコマンドには、停止のタイムアウトがあります。ドメイン管理サーバのリクエストタイムアウトが発生した場合、停止処理は継続します。停止のタイムアウトまたは asadmin リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、停止処理は継続します。停止に成功したかどうかは、KDKD20032-I のログまたは GUI のサーバインスタンスの状態により確認できます。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--kill={false|true}`

サーバインスタンスのプロセスを停止する OS の機能を使用して、インスタンスを強制終了するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `false`

インスタンスを強制終了しません。Java プラットフォームの機能を使用してサーバインスタンスのプロセスを停止します。

- `true`

インスタンスを強制終了します。OS の機能を使用してサーバインスタンスのプロセスを停止します。

デフォルト値：`false`

*instance-name*

停止するサーバインスタンスの名称を指定します。

型：String

指定できる値を次に示します。

- サーバインスタンス名

デフォルト値：なし

## 使用例

次の例では、サーバインスタンスの `yml-i-sj01` を停止します。

```
asadmin stop-instance yml-i-sj01
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.6.27 stop-prf

PRF の停止

### 書式

```
asadmin [asadmin-options] stop-prf prf-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

stop-prf サブコマンドは、指定した名称の PRF を停止します。

このコマンドはリモートモードだけサポートします。

なお、このサブコマンドには、停止のタイムアウトがあります。ドメイン管理サーバのリクエストタイムアウトが発生した場合、停止処理は継続します。停止のタイムアウトまたは asadmin リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、停止処理は継続します。停止に成功したかどうかは、list-prfs サブコマンドまたは GUI の PRF 状態により確認できます。

### 実行権限

一般ユーザー

### 前提条件

PRF を起動済みであること。

### 引数

*prf-name*

停止する PRF のサーバ名を指定します。

型：String

指定できる値を次に示します。

- ドメインに存在する PRF のサーバ名

デフォルト値：なし（必須）

## 使用例

PRF1 という名称の PRF を停止する場合

```
asadmin stop-prf PRF1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.28 stop-servers

サーバの一括停止

### 書式

```
asadmin [asadmin-options] stop-servers [--graceful {false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`stop-servers` サブコマンドは、ドメイン内の各種サーバを一括で停止します。サーバ間に関連（依存関係）が設定されている場合は、その依存元のサーバを先に停止します。依存関係が多い場合は、依存元の末端に当たるサーバから順に停止します。サーバ間に関連（依存関係）が設定されていない場合は、すべてのサーバに対し、並行して処理します。

すでに停止状態のサーバに対しては、停止処理を実行しません。

このコマンドはリモートモードだけサポートします。

WebSocket を使用している場合、クライアント・サーバ間のコネクションが残存した状態になりやすいため、Web サーバが停止できないことがあります。コネクションを強制的に切断して Web サーバを停止したい場合は、`--graceful` オプションに `false` を指定して実行してください。

## 実行権限

一般ユーザー

## 前提条件

各種サーバを起動済みであること。

## 引数

`--graceful {false|true}`

Web サーバを計画的に停止するかどうかを指定します。Web サーバ以外のサーバには適用されません。

型：Boolean

指定できる値を次に示します。

- `true`  
計画停止します。
- `false`  
通常停止します。

デフォルト値：`true`

## 使用例

ドメイン内のすべてのサーバを停止する場合

```
asadmin stop-servers
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.29 stop-webserver

Web サーバの停止

## 書式

```
asadmin [asadmin-options] stop-webserver [--graceful {false|true}]  
webserver-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`stop-webserver` サブコマンドは、指定した名称の Web サーバを停止します。

このコマンドはリモートモードだけサポートします。

なお、このサブコマンドには、停止のタイムアウトがあります。ドメイン管理サーバのリクエストタイムアウトが発生した場合、停止処理は継続します。停止のタイムアウトまたは `asadmin` リードタイムアウトが発生した場合、サブコマンドの実行は失敗しますが、停止処理は継続します。停止に成功したかどうかは、`list-webservers` サブコマンドまたは GUI の Web サーバの状態により確認できます。

WebSocket を使用している場合、クライアント・サーバ間のコネクションが残存した状態になりやすいため、Web サーバが停止できないことがあります。コネクションを強制的に切断して Web サーバを停止したい場合は、`--graceful` オプションに `false` を指定して実行してください。

## 実行権限

一般ユーザー

## 前提条件

Web サーバを起動済みであること。

## 引数

`--graceful {false|true}`

Web サーバを計画的に停止するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
計画停止します。
- `false`  
通常停止します。

デフォルト値：`true`

*webserver-name*

停止する Web サーバのサーバ名を指定します。

型：String

指定できる値を次に示します。

- ドメインに存在する Web サーバ名

デフォルト値：なし（必須）

## 使用例

```
asadmin stop-webserver Web1
```

```
asadmin stop-webserver --graceful true Web1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.6.30 validate-multicast

マルチキャストトランスポートの検証

### 書式

```
asadmin [asadmin-options] validate-multicast  
  [--help] [--multicastport multicastport]  
  [--multicastaddress multicastaddress] [--bindaddress bindaddress]  
  [--sendperiod sendperiod] [--timeout timeout]  
  [--timetolive timetolive] [--verbose={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`validate-multicast` サブコマンドは、クラスターに対してマルチキャストトランスポートが使用できるかどうかを検証します。

検証するそれぞれのホストで同時にこのサブコマンドを実行します。すべてのマシンが相互に確認できれば、それらのマシンでマルチキャストは正常に動作しています。

マシンが相互に確認できない場合は、`--bindaddress` オプションを設定してすべてのマシンが同じサブネット上のインターフェースを使用していることを確認するか、`--timetolive` オプションの値を増やします。

これらの変更を行ってもマルチキャストの問題が解決しない場合は、ネットワーク管理者に依頼して、クラスターの実行に使用されているすべてのマシンで、マルチキャストメッセージを受信できるようにネットワークが構成されているかどうかを確認してください。

このサブコマンドはローカルモードだけサポートします。

ドメイン管理サーバおよびクラスターの実行中は、ドメイン管理サーバおよびクラスターのマルチキャストアドレスとポートの値を使用してこのサブコマンドを実行しないでください。実行すると、エラーが発生します。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--multicastport multicastport`

グループ管理サービスがグループイベントをリッスンするマルチキャストソケットのポートを指定します。2048 から 32000 までの標準の UDP ポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 2048～32000

デフォルト値：2048

`--multicastaddress multicastaddress`

グループ管理サービスがグループイベントをリッスンするマルチキャストソケットのアドレスを指定します。クラス D の IP アドレスを指定します。クラス D の IP アドレスは 224.0.0.0 から 239.255.255.255 までです。アドレス 224.0.0.0 は予約済みのため使用できません。

型：IP

指定できる値を次に示します。

- 224.0.0.1～239.255.255.255

デフォルト値：228.9.3.1

`--bindaddress bindaddress`

グループ管理サービスのマルチキャストデータグラムパケットを受信するローカルインターフェースの IP アドレスを指定します。

型：String

指定できる値を次に示します。

- ローカルインターフェースの IP アドレス

デフォルト値：使用できるすべてのローカルインターフェースの IP アドレス

`--sendperiod sendperiod`

ノード間でテストメッセージを送信する間隔（ミリ秒）を指定します。

型：Integer

指定できる値を次に示します。

- 0～2147483647



デフォルト値：2000

`--timeout timeout`

サブコマンドがタイムアウトし終了するまでの秒数を指定します。このサブコマンドは、[CTRL] + [C] を使用して終了することもできます。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：20

`--timetolive timetolive`

マルチキャストのスコープを制御するために、マルチキャストソケットから送信されるマルチキャストパケットのデフォルトの生存期間を指定します。生存期間の値は、0 から 255 までの範囲にする必要があります。

型：Integer

指定できる値を次に示します。

- 0~255

デフォルト値：JDK のデフォルトまたはグループ管理サービスサブシステムの定数のどちらか小さい方

`--verbose={false|true}`

true に設定した場合、追加のデバッグ情報を表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

## 使用例

次の例では、c1 という名称のクラスターに対してマルチキャストトランスポートが使用できるかどうかを確認します。

ホスト sr1 から実行する場合：

```
asadmin validate-multicast
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.7 サーバ間関連設定で使用するコマンド

サーバ間関連設定で使用するコマンドの書式、機能などについて説明します。

### 2.7.1 create-relation

サーバ間関連を作成

#### 書式

```
asadmin [asadmin-options] create-relation
  --relationtype relation-type-name --from server-name
  --to server-name [--order order]
  [--properties name=value[:name=value]...] relation-name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

create-relation サブコマンドは、サーバ間の関連を作成します。関連種別に対応した関連元と関連先のサーバ名を指定することで、関連を作成できます。このコマンドは関連情報を、--from に指定したサーバの関連情報として作成します。

このコマンドはリモートモードだけサポートします。

リダイレクト関連では、次の--properties オプションを使用して、必要な情報を設定します。

- Java EE サーバと関連する場合

プロパティ	指定内容
path	リダイレクターの動作の定義で、マッピング定義のローカルの仮想パスの名前を指定します。
network-listener	リダイレクト先となるネットワークリスナー名を指定します。
proxy-pass-option	リダイレクターの動作の定義で、マッピング定義のコネクションプーリングの調整パラメーターを指定します。
set-proxy-pass-reverse-cookie	リダイレクターの動作の定義で、Set-Cookie ヘッダーの再割り当てをするかどうかを指定します。

- クラスタと関連する場合

プロパティ	指定内容
stickysession	リダイクターの動作の定義で、Sticky を有効にするかどうかを指定します。

- 特定の URL のパスをリダイレクトさせないようにする場合

プロパティ	指定内容
negative	リダイクターの動作の定義で、マッピング定義に対するリクエストを除外するかどうかを指定します。

## 実行権限

一般ユーザー

## 前提条件

各種サーバを構築済みであること。

## 引数

`--relationtype relation-type-name`

関連元と関連先のサーバを関連付ける関連種別を指定します。

型：String

指定できる値を次に示します。

- `prf-relation`  
PRF 関連
- `redirect`  
リダイレクト関連

デフォルト値：なし（必須）

`--from server-name`

関連元となるサーバ名を指定します。関連種別の関連元サーバ種別に設定されているサーバ種別のサーバを指定します。

型：String

指定できる値を次に示します。

- `--relationtype` が `prf-relation` の場合  
ドメインに存在する JavaEE サーバのサーバ名  
ドメインに存在する Web サーバのサーバ名
- `--relationtype` が `redirect` の場合  
ドメインに存在する Web サーバのサーバ名

デフォルト値：なし（必須）

#### `--to server-name`

関連先となるサーバ名を指定します。関連種別の関連先サーバ種別に設定されているサーバ種別のサーバを指定します。

型：String

指定できる値を次に示します。

- `--relationtype` が `prf-relation` の場合  
ドメインに存在する PRF のサーバ名
- `--relationtype` が `redirect` の場合  
ドメインに存在する JavaEE サーバのサーバ名またはクラスター名

デフォルト値：なし（必須）

#### `--order order`

1つの Web サーバに複数のリダイレクト関連を作成する場合に、リダイレクト先の優先順位を付けるための順序番号を指定します。サーバの起動順序および停止順序には影響しません。

型：Integer

指定できる値を次に示します。

- 0～65535 の整数

デフォルト値：0

#### `--properties name=value[:name=value]...`

サーバ間の関連情報に適用するプロパティを、プロパティ名と値のペアで指定します。指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

##### *name*

型：String

指定できる値を次に示します。

- PRF の標準プロパティまたは拡張プロパティ(ex\_から始まるプロパティ)。

##### *value*

指定できる値を次に示します。

- `name` に指定したプロパティの指定範囲に従います。

デフォルト値：`--relationtype` に `redirect` が指定されている場合は、`hitachi-relation-types.hitachi-relation-type.redirect.property.標準プロパティキー` の値が適用されます。`prf-relation` が指定されている場合は、なし。

#### *relation-name*

サーバ間関連に設定する名称を指定します。

型：String

指定できる値を次に示します。

- ASCII 文字であること。
  - 名称は次の文字で構成すること。
    - 英小文字<a~z>
    - 英大文字<A~Z>
    - 数字<0~9>
    - ハイフン<->
    - アンダースコア<\_>
  - 先頭文字は英大文字または英小文字を使用すること。
  - ドメインに存在する次の名称は指定できません。
    - ノード名
    - JavaEE サーバ名
    - Web サーバ名
    - PRF のサーバ名
    - クラスター名
    - サーバ間関連名
    - JavaEE サーバの構成名
  - 次に示す名称はシステムで使用済みまたは予約語のため指定できません。
    - domain
    - server
    - default
    - server-config
    - default-config
    - default-webserver-config
    - default-prf-config
    - javaee
    - webserver
    - prf
    - cluster
    - redirect
    - prf-relation
    - HJES\_で始まる名称
    - 構築済み Web サーバ名-config
    - 構築済み PRF 名-config
- デフォルト値：なし (必須)

## 使用例

JavaEE サーバJavaEE1 と PRFPRF1 に対して、PRF 関連（関連種別prf-relation、関連名redirect1）を作成する場合

```
asadmin create-relation --relationtype prf-relation --from JavaEE1
--to PRF1 redirect1
```

Web サーバWeb1 と JavaEE サーバJavaEE1 に対して、リダイレクト関連（関連種別redirect、関連名redirect3）を作成する場合

```
asadmin create-relation --relationtype redirect --from Web1 --to JavaEE1
--properties path=/aaa/:network-listener=http-listener-1 redirect3
```

順序番号を指定して複数のリダイレクト関連を作成する場合

```
asadmin create-relation --relationtype redirect --from Web1 --to JavaEE1
--order 20 --properties path=/aaa/:network-listener=http-listener-1 redirect3
```

```
asadmin create-relation --relationtype redirect --from Web1 --to JavaEE2
--order 10 --properties path=/:network-listener=http-listener-1 redirect4
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.7.2 delete-relation

サーバ間関連を削除

### 書式

```
asadmin [asadmin-options] delete-relation relation-name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-relation サブコマンドは、サーバ間の関連をドメインから削除します。

このコマンドはリモートモードだけサポートします。

## 実行権限

一般ユーザー

## 前提条件

サーバ間関連を作成済みであること。

## 引数

*relation-name*

削除するサーバ間関連の名称を指定します。

型：String

指定できる値を次に示します。

- 存在するサーバ間関連の名称

デフォルト値：なし（必須）

## 使用例

関連redirect1 を削除する場合

```
asadmin delete-relation redirect1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.7.3 list-relations

サーバ間関連を一覧表示

## 書式

```
asadmin [asadmin-options] list-relations [--long={false|true} |  
--output output] [--header={false|true}]]  
[--relationtype relation-type-name] [--from server-name]  
[--to server-name] [target]
```



## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-relations` サブコマンドは、サーバ間の関連を一覧表示します。

このコマンドはリモートモードだけサポートします。

## 実行権限

一般ユーザー

## 引数

`--long={false|true}`

一覧表示するサーバ間関連について、詳細情報を表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`

関連名とその詳細情報（関連種別、関連元サーバ名、関連先サーバ名、順序番号）を表示します。ヘッダー情報も表示します。なお、このオプションに `true` を指定した場合は、`--output` オプションは指定できません。

- `false`

`--output` オプションおよび `--header` オプションの指定に従います。

デフォルト値：`false`

`--output output`

リスト表示する項目およびその順序をコンマ区切りのリストで指定します。指定値の大文字、小文字は区別されません。

リスト表示する項目は、このオプションで指定した順番に左から表示します。また、リストの行は一番左の項目でソートされて表示されます。文字列でソートするため、一番左の項目に `order` を指定した場合、数値の昇順にはなりません。

このオプションを省略した場合は、関連名、関連種別、関連元サーバ名、関連先サーバ名が表示されます。ただし、`--long` オプションに `true` を指定した場合は、その指定に従います。

型：String

指定できる値を次に示します。

- `name`

関連名

- `type`

関連種別名

- `from`

関連元サーバ名

- `to`

関連先サーバ名

- `order`

順序番号

デフォルト値：なし

`--header={false|true}`

ヘッダー情報を表示するかどうかを指定します。ただし、`--long` オプションに `true` を指定した場合はこのオプションの指定は無視されます。

型：Boolean

指定できる値を次に示します。

- `true`

ヘッダー情報を表示します。

- `false`

ヘッダー情報を表示しません。

デフォルト値：なし

`--relationtype relation-type-name`

フィルタリングする関連種別を指定します。

型：String

指定できる値を次に示します。

- `prf-relation`

PRF 関連

- `redirect`

リダイレクト関連

デフォルト値：なし

指定がない場合はすべての関連種別の関連を表示します。

`--from server-name`

フィルタリングする関連元のサーバ名を指定します。

型：String

指定できる値を次に示します。

- 存在する関連元のサーバ名

デフォルト値：なし

指定がない場合はすべての関連元のサーバ名を表示します。

`--to server-name`

フィルタリングする関連先のサーバ名を指定します。

型：String

指定できる値を次に示します。

- 存在する関連先のサーバ名

デフォルト値：なし

指定がない場合はすべての関連先のサーバ名を表示します。

`target`

情報を表示する対象名を指定します。

型：String

指定できる値を次に示します。

- `domain`  
ドメイン内のすべての関連の情報を表示します。
- *関連名*  
指定した関連の情報を表示します。

デフォルト値：`domain`

## 出力形式

1行目は出力情報のヘッダー部となり、固定文字列で出力形式を示します。

カラム間は、カラム内で最も長い文字列と次のカラムの先頭までを、2つの半角スペースで区切ります。

NAME	TYPE	FROM	TO	ORDER
<i>関連名</i>	<i>関連種別名</i>	<i>関連元サーバ名</i>	<i>関連先サーバ名</i>	<i>順序番号</i>

*関連名*：関連の名称を表示します。

*関連種別名*：関連種別の名称を表示します。

*関連元サーバ名*：関連元となるサーバの名称を表示します。

*関連先サーバ名*：関連先となるサーバの名称を表示します。

*順序番号*：関連の順序番号を表示します。

## 使用例

すべてのサーバ間関連を表示する場合

```
asadmin list-relations
```

[出力例]

```
relation1 prf-relation      JavaEE1      PRF1
relation2 prf-relation      Web1         PRF1
relation3 redirect           Web1         JavaEE1
relation4 redirect           Web1         JavaEE2
relation5 ex_relation_type1 custom_type1 Web1
```

すべてのサーバ間関連とヘッダーを表示する場合

```
asadmin list-relations --long=true
```

[出力例]

NAME	TYPE	FROM	TO	ORDER
relation1	prf-relation	JavaEE1	PRF1	0
relation2	prf-relation	Web1	PRF1	0
relation3	redirect	Web1	JavaEE1	20
relation4	redirect	Web1	JavaEE2	10
relation5	ex_relation_type1	custom_type1	Web1	0

すべてのサーバ間関連とヘッダー情報なしで表示する場合

```
asadmin list-relations --long=true --header=false
```

[出力例]

relation1	prf-relation	JavaEE1	PRF1	0
relation2	prf-relation	Web1	PRF1	0
relation3	redirect	Web1	JavaEE1	20
relation4	redirect	Web1	JavaEE2	10
relation5	ex_relation_type1	custom_type1	Web1	0

関連種別がredirect の関連だけを表示する場合

```
asadmin list-relations --relationtype redirect
```

[出力例]

```
relation3 redirect Web1 JavaEE1
relation4 redirect Web1 JavaEE2
```

関連元のサーバがWeb1 の関連だけを表示する場合

```
asadmin list-relations --from Web1
```

[出力例]

```
relation2 prf-relation Web1 PRF1
relation3 redirect Web1 JavaEE1
relation4 redirect Web1 JavaEE2
```

関連先のサーバがPRF1 の関連だけを表示する場合

```
asadmin list-relations --to PRF1
```

[出力例]

```
relation1 prf-relation JavaEE1 PRF1
relation2 prf-relation Web1 PRF1
```

関連名relation3 だけの情報を表示する場合

```
asadmin list-relations relation3
```

[出力例]

```
relation3 redirect Web1 JavaEE1
```

関連元のサーバがWeb1、関連種別がredirect の関連の、順序番号、関連名、および関連先サーバの情報だけを順序番号でソートして表示する場合

```
asadmin list-relations --relationtype redirect --from Web1
--output order,name,to
```

[出力例]

```
10 relation4 JavaEE2
20 relation3 JavaEE1
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 2.8 アプリケーション管理で使用するコマンド

アプリケーション管理で使用するコマンドの書式、機能などについて説明します。

### 2.8.1 create-application-ref

アプリケーション参照の作成

#### 書式

```
asadmin [asadmin-options] create-application-ref [--help]
  [--target target] [--virtualservers virtual_servers]
  [--enabled={true|false}] reference_name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

create-application-ref サブコマンドは、クラスターまたはクラスター構成ではないサーバインスタンスからすでにデプロイされているアプリケーション（Java EE アプリケーション、Web モジュール、Enterprise Bean モジュール）への参照を作成します。

これによってアプリケーション要素が実際にデプロイされ、ターゲットのインスタンスまたはクラスターで使用できるようになります。

対象のインスタンスやクラスターのインスタンスが、起動していなかったり、使用できなくても、このサブコマンドは正常に実行されます。1 つ以上のインスタンスがアプリケーション参照を作成できない場合、これらのインスタンスは次の起動時にアプリケーション要素を受け取ります。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- このサブコマンドを使用して参照を作成するアプリケーションまたはモジュールは、事前にデプロイしておく必要があります。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

### `--target target`

作成するアプリケーション参照の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

### `--virtualservers virtual_servers`

コマンド区切りの仮想サーバ ID リストをデプロイ先に指定します。このオプションは Web モジュール（スタンドアロンまたは Java EE アプリケーション内のどれか）にだけ適用されます。このオプションを指定しない場合、アプリケーションは管理対象サーバ、`__asadmin` 以外のすべての仮想サーバにデプロイされます。

型：String

指定できる値を次に示します。

- 指定したコマンド区切りの仮想サーバ ID リストの名称

デフォルト値：なし

### `--enabled={true|false}`

アプリケーションを有効にする（ロードする）必要があるかどうかを示します。この値は、アプリケーションがグローバルレベルで有効な場合にだけ有効になります。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`true`

### `reference_name`

アプリケーション名またはモジュール名を指定します。次のものが含まれます。名称にはオプションでバージョン識別子を入れることができます。この場合、名称の後にコロンの(:)で区切って追加します。次に示すアプリケーション名またはモジュール名を指定できます。

型：String

指定できる値を次に示します。

- `JavaEE` アプリケーション

- Web モジュール
- EJB モジュール
- リソースアダプターアーカイブ
- アプリケーションクライアントのモジュール
- ライフサイクルモジュール

デフォルト値：なし

## 使用例

次の例では、クラスター構成ではないサーバインスタンス上の Web モジュールMyWebApp へのアプリケーション参照を作成します。

```
asadmin create-application-ref --target NewServer MyWebApp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.2 delete-application-ref

アプリケーション参照の削除

### 書式

```
asadmin [asadmin-options] delete-application-ref [--help]
  [--target target] [--cascade={false|true}] reference_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-application-ref サブコマンドは、クラスターまたはクラスター構成ではないサーバインスタンスから、アプリケーション参照を削除します。これによって、アプリケーション要素は対象のインスタンス上またはクラスター上でアンデプロイされ、使用できなくなります。

このサブコマンドを実行するとき、クラスターを構成する対象のインスタンスが、実行中または使用できる状態である必要はありません。



1 つ以上のインスタンスが使用できない場合、それらのインスタンスは次回の起動時にアプリケーションをロードしなくなります。

アプリケーション参照を削除しても、アプリケーションがドメインから削除されることはありません。アプリケーションを削除できるのは `undeploy` サブコマンドだけです。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除するアプリケーション参照の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値： `server`

`--cascade={false|true}`

リソースアダプターアーカイブに関し、モジュールに依存するリソースも再帰的に削除するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値： `false`

`reference_name`

次に示すアプリケーションまたはモジュールの名称を指定します。

- Java EE アプリケーションアーカイブ

- Web モジュール
- EJB モジュール
- リソースアダプターアーカイブ
- アプリケーションクライアントのモジュール
- ライフサイクルモジュール

型：String

指定できる値を次に示します。

- アプリケーション名、またはモジュール名

デフォルト値：なし

## 使用例

次の例では、クラスター構成ではないサーバインスタンスNewServer から、Web モジュールMyWebApp へのアプリケーション参照を削除します。

```
asadmin delete-application-ref --target NewServer MyWebApp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.3 deploy

コンポーネントのデプロイ

### 書式

```
asadmin [asadmin-options] deploy [--help]
  [--force={false|true}]
  [--virtualservers virtual_servers]
  [--contextroot context_root]
  [--precompilejsp={false|true}]
  [--name component_name]
  [--upload={true|false}]
  [--retrieve local_dirpath]
  [--deploymentplan deployment_plan]
  [--altd alternate_deploymentdescriptor]
  [--runtimealtd runtime_alternate_deploymentdescriptor]
  [--deploymentorder deployment_order]
  [--enabled={true|false}]
  [--generateterminstubs={false|true}]
```

```
[--availabilityenabled={false|true}]
[--libraries jar_file[,jar_file]...]
[--target target]
[--type pkg-type]
[--properties name=value[:name=value]...]
file_archive|filepath
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`deploy` サブコマンドは、アプリケーションをサーバにデプロイします。アプリケーションには、エンタープライズアプリケーション、Webアプリケーション、EJBモジュール、リソースアダプターアーカイブ、およびアプリケーションクライアントのモジュールがあります。コンポーネントがすでにデプロイされている場合、`--force` オプションに `true` を設定すると、強制的にリデプロイされます。

このサブコマンドはリモートモードだけサポートします。

アプリケーションのデプロイの実行時にドメイン管理サーバで `OutOfMemoryError` が発生した場合、ドメイン管理サーバのJavaヒープが不足している可能性があります。ドメイン管理サーバのJavaヒープサイズは、デプロイするアプリケーションアーカイブのサイズを考慮したサイズにしてください。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

- `file_archive` または `filepath` には、コマンドでデプロイされる入力ファイルを与えます。
- `--retrieve` オプションを使用すると、`--retrieve` オプションで指定したファイルパスにクライアントスタブのJARファイルを生成します。
- `--deploymentplan` オプションには、Java EE RIのDDを含む入力JARファイルを与えます。
- `--altd` オプションには、標準DDを含む入力JARファイルを与えます。
- `--runtimealtd` オプションには、Java EE RIのDDを与えます。
- `--generateterminstubs` オプションを使用すると、静的RMI-IIOPスタブが生成され、`client.jar` に含まれます。
- `--libraries` オプションには、デプロイされたアプリケーションで利用できる入力ライブラリーJARファイルを与えます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--force={false|true}`

指定したコンポーネントがすでにデプロイされている場合に、コンポーネントをリデプロイするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--virtualservers virtual_servers`

仮想サーバ ID を指定します。複数の ID はコンマ (,) で区切ります。

型：String

指定できる値を次に示します。

- *仮想サーバ ID*

デフォルト値：なし

`--contextroot context_root`

アプリケーションのコンテキストルート指定します。アーカイブが Web モジュールの場合にだけ有効です。

型：String

指定できる値を次に示します。

- *Web モジュールのコンテキストルート*

デフォルト値：Web モジュールのアーカイブファイル (拡張子なし)

`--precompilejsp={false|true}`

Web アプリケーションのデプロイ中に、Web アプリケーションが同梱する JSP ファイルを、Java EE サーバが事前コンパイルするかどうかを指定します。事前コンパイルでは、Web アプリケーションに含まれる次の拡張子のファイルは、すべてコンパイルされます。

- jsp
- jspix

なお、JSP セグメントが Web アプリケーションに含まれる場合、JSP セグメントのファイル名に上記の拡張子を指定しないでください。JSP セグメントは JSP 構文として成立しないことがあります。そのため、JSP セグメントのファイル名の拡張子に"jsp"または"jspx"を指定すると、JSP 構文として成立しない JSP セグメントも事前コンパイルされ、コンパイルエラーとなる場合があります。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

#### `--name component_name`

デプロイするコンポーネントの名称を指定します。

名称にはオプションでバージョン識別子を入れることができます。この場合、名前の後にコロン (:) で区切って追加します。バージョン識別子は、文字または数値で始まる必要があります。使用できる文字は、英数字、アンダースコア (`_`)、ハイフン (`-`) およびピリオド (`.`) です。

型：`String`

指定できる値を次に示します。

- コンポーネント名

デフォルト値：なし

#### `--upload={true|false}`

ファイルをドメイン管理サーバにアップロードするかどうかを指定します。

`filepath` にディレクトリが指定された場合、このオプションは無視されます。

型：`Boolean`

指定できる値を次に示します。

- `false`

ファイルをアップロードしないで、指定したファイル名でファイルへのアクセスを試みます。ドメイン管理サーバがファイルにアクセスできない場合、このサブコマンドは失敗します。

例えば、ドメイン管理サーバが管理ユーザーとは異なるユーザーとして稼働している可能性があり、ファイルへのアクセス権がドメイン管理サーバにない場合があります。この状況では、`--upload` オプションを `false` に設定すると、サブコマンドは失敗します。

- `true`

ネットワーク接続経由でファイルをドメイン管理サーバにアップロードします。

デフォルト値：

- サブコマンドを実行するホスト上にドメイン管理サーバがある場合  
`false`
- リモートホスト上にドメイン管理サーバがある場合  
`true`

#### `--retrieve local_dirpath`

サーバマシンからローカルディレクトリにクライアントスタブの JAR ファイルを取得します。

型：`String`

指定できる値を次に示します。

- クライアントスタブのJAR ファイルの格納先ディレクトリーパス

デフォルト値：なし

#### `--deploymentplan deployment_plan`

Java EE RI の DD を含む JAR ファイルである、デプロイメントプランをデプロイします。

型：String

指定できる値を次に示します。

- JAR ファイル名

デフォルト値：なし

#### `--altddd alternate_deploymentdescriptor`

アプリケーションアーカイブの外にある標準 DD を使用してアプリケーションをデプロイします。

代替となる DD は、アーカイブに含まれる最高位の DD をオーバーライドします。例えば EAR の場合は、`--altddd` オプションは `application.xml` ファイルをオーバーライドします。スタンドアロンモジュールの場合、`--altddd` オプションは `web.xml` などの最上位のモジュール記述子をオーバーライドします。

型：String

指定できる値を次に示します。

- 標準 DD

デフォルト値：なし

#### `--runtimealtddd runtime_alternate_deploymentdescriptor`

アプリケーションアーカイブの外にある Java EE RI の DD を使用してアプリケーションをデプロイします。

代替となる DD は、アーカイブに含まれる最高位の DD をオーバーライドします。

スタンドアロンモジュールの場合、`-runtimealtddd` オプションは `glassfish-web.xml` などの最上位のモジュール記述子をオーバーライドします。Java EE Server の DD だけに適用されます (`glassfish-*.xml`)。代替となる DD ファイルの名称は、`glassfish-`で始まる必要があります。非推奨の `sun-*.xml` DD には適用されません。

型：File

指定できる値を次に示します。

- Java EE RI の DD のファイル名

デフォルト値：なし

#### `--deploymentorder deployment_order`

アプリケーションのデプロイ順序を指定します。

小さい数値のアプリケーションは、大きい数値のアプリケーションの前にロードされます。

2つのアプリケーションのデプロイ順序が同じ場合、最初にデプロイされたアプリケーションがサーバ起動時に先にロードされます。

型：Integer

指定できる値を次に示します。

- 1~2147483647

デフォルト値：100

`--enabled={true|false}`

ユーザーがアプリケーションにアクセスできるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--generateterminstubs={true|false}`

スタブを生成するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--availabilityenabled={false|true}`

Stateful Session Bean のチェックポイントイングと潜在的なパッシベーション、および Web セッションに対して、高可用性を有効にするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--libraries jar_file[,jar_file]...`

`add-library` サブコマンドで Java EE Server に追加したライブラリー JAR ファイルをコンマ区切りで指定します。ライブラリー JAR ファイルは、`domain-dir/lib/applibs` に対する相対パスで指定します。

型：String

デフォルト値：なし

`--target target`

デプロイする対象を指定します。

型：String

指定できる値を次に示します。

- server

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- **domain**

ドメインを対象とします。

ドメインが初期デプロイの対象の場合、アプリケーションはドメインにデプロイされます。このとき、サーバインスタンスまたはクラスターはアプリケーションを参照しません。ドメインがリデプロイの対象 (--force オプションでtrueを設定) で、アプリケーションを参照するクラスターまたはサーバインスタンスに対して動的再構成が有効な場合、参照しているクラスターまたはサーバインスタンスは自動的に新しいバージョンのアプリケーションを取得します。ドメインがリデプロイの対象で動的再構成が無効の場合、参照しているクラスターまたはサーバインスタンスは、クラスターまたはスタンドアロンのサーバインスタンスが再起動するまで新しいバージョンのアプリケーションを取得しません。

- **cluster\_name**

指定した名称のクラスターを対象とします。

- **instance\_name**

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--type pkg-type**

デプロイする対象コンポーネントのアーカイブの形式を指定します。

型：String

指定できる値を次に示します。

- **car**

コンポーネントは CAR ファイルとしてパッケージされます。

- **ear**

コンポーネントは EAR ファイルとしてパッケージされます。

- **ejb**

コンポーネントは JAR ファイルとしてパッケージされた EJB になります。

- **rar**

コンポーネントは RAR ファイルとしてパッケージされます。

- **war**

コンポーネントは WAR ファイルとしてパッケージされます。

デフォルト値：なし

**--properties name=value | --property name=value**

デプロイメントに追加するプロパティをキーと値のペア (キー = 値) で指定します。

型：String

デプロイメントには次のプロパティを指定できます。

- **preserveAppScopedResources={false|true}**



true を設定した場合、アプリケーションスコープのすべてのリソースを保存し、リデプロイのあいだに復元します。

型：Boolean

指定できる値を次に示します。

true

false

デフォルト値：false

これら以外に利用できるプロパティは、リデプロイするコンポーネントの実装によって決まります。

デフォルト値：なし

*file\_archive|filepath*

デプロイするアプリケーションを含むアーカイブへのパスを指定します。

型：String

- アプリケーションを含むアーカイブファイルへのパス

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、Cart.ear ファイルに含まれるエンタープライズアプリケーションをデプロイします。

```
asadmin deploy --target instance1 Cart.ear
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.4 disable

コンポーネントの無効化

### 書式

```
asadmin [asadmin-options] disable [--help]  
      [--target target] component_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`disable` サブコマンドは、指定したデプロイ済みコンポーネントをただちに無効にします。コンポーネントがデプロイされていない場合は、エラーメッセージが返ります。

このサブコマンドはリモートモードだけサポートします。

このサブコマンドはリモートでコンポーネントを無効にするとき、サーバと通信します。この通信に対するタイムアウト時間を、`asenv` ファイルの `AS_ADMIN_READ_TIMEOUT` パラメーターで設定できます。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

サブコマンドのヘルプテキストを表示します。

`--target target`

無効にするコンポーネントの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
スタンドアロンのサーバインスタンス、もしくは、クラスター内のサーバインスタンスを単体で、対象とします。

デフォルト値：`server`

`component_name`

無効にするコンポーネントの名称を指定します。

型：String

- コンポーネント名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスにデプロイされたコンポーネント sampleApp を無効にします。

```
asadmin disable --target instance1 sampleApp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 注意事項

- コンポーネントをアンデプロイ、または参照を削除するとメモリーからインスタンスが解放されますが、コンポーネントを無効化 (disable コマンドを実行) した状態ではインスタンスが解放されません。多量のコンポーネントを無効化した場合、使用中のメモリーが増えて OutOfMemoryError が発生しやすくなる場合があるため注意してください。

## 2.8.5 enable

コンポーネントの有効化

### 書式

```
asadmin [asadmin-options] enable [--help]  
      [--target target] component_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

enable サブコマンドは、指定したデプロイ済みコンポーネントを有効にします。コンポーネントがすでに有効である場合には、再度有効化されます。デプロイされていない場合、エラーメッセージが返ります。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

有効にするコンポーネントの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
スタンドアロンのサーバインスタンス、もしくは、クラスター内のサーバインスタンスを単体で、対象とします。

デフォルト値：`server`

`component_name`

オプションのバージョン ID を含むコンポーネントの名称を指定します。ID はコロン (:) で区切って名称に続けます。

バージョン ID の先頭は、文字か数字である必要があります。使用できる文字は、英数字のほかに、アンダースコア (\_)、ハイフン (-) およびピリオド (.) です。

型：String

指定できる値を次に示します。

- コンポーネント名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンス上の、無効になっている `sampleApp` という名称のコンポーネントを有効にします。

```
asadmin enable --target instance1 sampleApp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.6 get-client-stubs

クライアントスタブの JAR ファイルの取得

### 書式

```
asadmin [asadmin-options] get-client-stubs [--help]
  --appname application_name
  local_directory_path
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

get-client-stubs サブコマンドは、アプリケーションクライアントのスタンドアロンモジュールまたはアプリケーションクライアントのモジュールを含むアプリケーション用のクライアントスタブ JAR ファイルを、サーバマシンからローカルディレクトリーに取得します。生成されたクライアントスタブ JAR ファイルは、サポートするすべての JAR ファイルとともに取得されます。クライアントスタブ JAR ファイル名は、**アプリケーション名Client.jar** の形式です。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- get-client-stubs サブコマンドを実行する前に、アプリケーションまたはモジュールをデプロイしておく必要があります。

### ファイル

get-client-stubs サブコマンドは、アプリケーションクライアントのスタンドアロンモジュールまたはアプリケーションクライアントのモジュールを含むアプリケーション用のクライアントスタブ JAR ファイルを、サーバマシンからローカルディレクトリーに取得します。

クライアントスタブ JAR ファイル名は、**アプリケーション名Client.jar** の形式です。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--appname application_name`

アプリケーションクライアントのモジュールまたはスタンドアロンのアプリケーションクライアントのモジュールの名称を指定します。

名称には任意のバージョン ID を指定できます。この場合、名称の後にコロン (:) で区切って追加します。バージョン ID の先頭は文字か数字である必要があります。使用できる文字は、英数字のほか、アンダースコア (\_)、ハイフン (-) およびピリオド (.) です。

型: String

指定できる値を次に示します。

- アプリケーション名

デフォルト値: なし

`local_directory_path`

クライアントスタブの JAR ファイルを格納するディレクトリーのパスを指定します。

型: String

指定できる値を次に示します。

- ディレクトリーのパス

デフォルト値: なし

## 使用例

次の例では、デプロイ済みである `myapplication` という名称のアプリケーションのクライアントスタブを取得します。

```
asadmin get-client-stubs --appname myapplication
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.7 list-application-refs

既存のアプリケーション参照の一覧表示

## 書式

```
asadmin [asadmin-options] list-application-refs [--help]
        [--long={false|true}] [target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-application-refs` サブコマンドは、クラスターまたはクラスター化されていないサーバインスタンスにあるすべてのアプリケーション参照を一覧表示します。指定した対象に配備されているモジュールはすべて一覧表示されます。複数のバージョンのモジュールまたはアプリケーションが配備されている場合、このサブコマンドはすべてのバージョンを一覧表示します。

このサブコマンドを正常に実行するために、クラスターを構成している対象のインスタンスが実行中または使用できる状態である必要はありません。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--long={false|true}`

`true` を指定した場合、一覧表示される各モジュールまたはアプリケーションが有効になっているかどうかが表示されます。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

*target*

一覧表示するアプリケーション参照の対象を指定します。

型：String

指定できる値を次に示します。

- `server`

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、クラスター化されていないサーバインスタンスNewServer のアプリケーション参照を一覧表示します。

```
asadmin list-application-refs NewServer
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.8 list-applications

デプロイ済みアプリケーションの一覧表示

### 書式

```
asadmin [asadmin-options] list-applications [--help]  
        [--long={false|true}] [--resources] [--subcomponents]  
        [--type type] [target]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-applications サブコマンドは、デプロイされた Java EE アプリケーションと各アプリケーションの種類を一覧表示します。複数のバージョンのモジュールまたはアプリケーションがデプロイされている場合、このサブコマンドはすべてのバージョンを一覧表示します。

--type オプションが指定されていない場合、すべてのアプリケーションが一覧表示されます。--type オプションを指定する場合、種類を指定する必要があります。



このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--long={false|true}`

`true` を指定した場合、一覧表示される各モジュールまたはアプリケーションが有効になっているかどうかが表示されます。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--resources`

各アプリケーションのアプリケーションスコープのリソースを一覧表示します。

`--subcomponents` オプションも使用した場合、アプリケーション内のコンポーネントごとにアプリケーションスコープのリソースを一覧表示します。

`--subcomponents`

各アプリケーションのサブコンポーネントを一覧表示します。

一覧表示されるサブコンポーネントはアプリケーションの種類によって異なります。例えば、Java EE アプリケーション（EAR ファイル）ではモジュールが一覧表示されます。Web アプリケーションではサーブレットと JSP ページが一覧表示されます。EJB モジュールでは EJB サブコンポーネントが一覧表示されます。

`--type type`

一覧表示するアプリケーションの種類を指定します。

型：String

指定できる値を次に示します。

- `application`
- `appclient`
- `connector`
- `ejb`
- `web`

- webservice

デフォルト値：なし（すべてのアプリケーションが一覧表示されます）

#### *target*

このサブコマンドを実行する対象を指定します。

型：String

指定できる値を次に示します。

- server  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- domain  
ドメインを対象とします。
- cluster\_name  
指定した名称のクラスターを対象とします。
- instance\_name  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、Web アプリケーションを一覧表示します。

```
asadmin list-applications --type web
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 注意事項

Java EE Server で提供している GUI やオンラインヘルプがこのコマンドの結果として表示されます。これらは製品の動作に必要なアプリケーションのため、アンデプロイなどの操作をしないでください。

`__internal_admin` をアンデプロイしてしまった場合、Administration Console にアクセスできなくなります。回復するためには、コマンドプロンプトから手動で次のコマンドを実行してください。

```
asadmin deploy --name __internal_admin --contextroot
admin "Application Serverインストールディレクトリ¥javaee¥glassfish¥admin¥admin"
```

`__internal_manual` をアンデプロイした場合には、GUI のオンラインヘルプやメッセージ ID が表示されなくなります。回復するためには、コマンドプロンプトから手動で次のコマンドを実行してください。

```
asadmin deploy --name __internal_manual --contextroot
manual "Application Serverインストールディレクトリー¥common¥docs"
```

## 2.8.9 list-containers

アプリケーションコンテナの一覧表示

### 書式

```
asadmin [asadmin-options] list-containers [--help]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-containers` サブコマンドは、アプリケーションコンテナのリストを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

### 使用例

次の例では、現在のアプリケーションコンテナを一覧表示します。

```
asadmin list-containers
```

### 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.10 list-modules

Java EE Server モジュールの一覧表示

### 書式

```
asadmin [asadmin-options] list-modules [--help]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-modules サブコマンドは、Java EE Server からアクセス可能なモジュールの一覧を表示します。サブコマンドを実行すると各モジュールのバージョンが表示されます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

### 使用例

次の例では、Java EE Server からアクセス可能なモジュールの一覧を表示します。

```
asadmin list-modules
```

### 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.11 list-sub-components

デプロイされたモジュール内またはデプロイされたアプリケーションモジュール内の EJB またはサーブレットのコンポーネントの一覧表示

### 書式

```
asadmin [asadmin-options] list-sub-components [--help]
  [--type type]
  [--appname appname] [--resources]
  modulename
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-sub-components` サブコマンドは、デプロイされたモジュール内、またはデプロイされたアプリケーションのモジュール内の EJB またはサーブレットのコンポーネントを一覧表示します。アプリケーション内の特定のモジュールを表示するには、`--appname` オプションでモジュール名を指定する必要があります。モジュールが指定されていない場合、すべてのモジュールが一覧表示されます。`--appname` オプションは、指定されたモジュールがスタンドアロンの場合だけ機能します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- `--appname` オプションは、指定されたモジュールがスタンドアロンの場合にだけ機能します。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--type type`

一覧表示するコンポーネントのタイプを指定します。指定がない場合、すべてのコンポーネントが一覧表示されます。

型：String

指定できる値を次に示します。

- `ejb`
- `servlet`

デフォルト値：すべてのコンポーネント

`--appname appname`

アプリケーション名を指定します。

- このオプションは、デプロイされたアプリケーションの組み込みモジュールのサブコンポーネントを出力する場合には必須です。
- アプリケーション名にはオプションのバージョン識別子を入れることができます。このバージョン識別子は、アプリケーション名のあとにコロン (:) で区切ります。バージョン識別子は、文字または数値で始まる必要があります。使用できる文字は、英数字のほかに、アンダースコア ( \_ )、ハイフン ( - )、およびピリオド ( . ) です。

型：String

指定できる値を次に示します。

- アプリケーション名

デフォルト値：なし

`--resources`

各サブコンポーネントのアプリケーションスコープのリソースを一覧表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`modulename`

サブコンポーネントを含むモジュールの名称を指定します。

型：String

指定できる値を次に示します。

- モジュール名

デフォルト値：なし

## 使用例

次の例では、`mejb.jar` モジュール内のMEjbApp アプリケーションのサブコンポーネントを一覧表示します。

```
asadmin list-sub-components --appname MEjbApp mejb.jar
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.12 list-timers

サーバインスタンスが所有する永続タイマーの一覧表示

### 書式

```
asadmin [asadmin-options] list-timers [--help] [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-timers` サブコマンドは、指定されたサーバインスタンスまたはサーバインスタンスのクラスターが所有する永続タイマーを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するタイマーの対象を指定します。

対象がスタンドアロンのサーバインスタンスの場合、インスタンスが所有するタイマーの数が一覧表示されます。対象がクラスターの場合、クラスターにある各インスタンスが所有するタイマーの数が一覧表示されます。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `standalone_instance_name`  
指定した名称のスタンドアロンのサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、指定したスタンドアロンのサービンスタンスにある永続タイマーを一覧表示します。

```
asadmin list-timers server
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.13 list-web-context-param

デPLOYされた Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの一覧表示

## 書式

```
asadmin [asadmin-options] list-web-context-param [--help]
        [--name=context-param-name] application-name[/module]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

list-web-context-param サブコマンドは、次のどちらかについて ServletContext 初期化パラメーターを一覧表示します。

- デPLOYされた Web アプリケーション
- デPLOYされた Java EE アプリケーションの Web モジュール

list-web-context-param サブコマンドは、set-web-context-param サブコマンドで事前に設定されたパラメーターだけを一覧表示します。アプリケーションの DD だけで設定されたパラメーターは一覧表示しません。

## 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- アプリケーションがすでにDEPLOYされている必要があります。それ以外の場合は、エラーが発生します。



## ファイル

モジュールへのパスは、アプリケーションの `application.xml` ファイルの `module` 要素に指定されています。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--name=context-param-name`

一覧表示する ServletContext 初期化パラメーターの名称を指定します。

型：String

指定できる値を次に示します。

- *ServletContext* 初期化パラメーターの名称

デフォルト値：事前に設定されたアプリケーションのすべてのパラメーター

*application-name/module*

アプリケーションの名称を指定します。アプリケーションの名称は、`list-applications` サブコマンドを使用して取得できます。

- *module* は、ServletContext 初期化パラメーターが Java EE アプリケーションの Web モジュールに適用される場合だけ必要です。
- アプリケーション名に加えてモジュール名を指定する場合は、`myApp/myWebModule.war` となります。
- *module* オプションは、スラッシュ (/) で区切って *application-name* オプションのあとに指定する必要があります。

型：String

指定できる値を次に示します。

- *application-name*  
アプリケーション名を指定します。
- *module*  
モジュール名を指定します。

デフォルト値：なし

## 使用例

次の例では、`set-web-context-param` サブコマンドで設定された Web アプリケーション `basic-ezcomp` のすべての ServletContext 初期化パラメーターを一覧表示します。

```
asadmin list-web-context-param basic-ezcomp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

### 2.8.14 list-web-env-entry

デプロイされた Web アプリケーションまたはモジュールの環境エントリーの一覧表示

#### 書式

```
asadmin [asadmin-options] list-web-env-entry [--help]
        [--name=env-entry-name] application-name[/module]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

list-web-env-entry サブコマンドは、次の項目のどちらかについて環境エントリーを一覧表示します。

- デプロイされた Web アプリケーション
- デプロイされた Java EE アプリケーションの Web モジュール

list-web-env-entry サブコマンドは、set-web-env-entry サブコマンドで事前に設定されたパラメーターだけを一覧表示します。アプリケーションの DD だけで設定されたパラメーターは一覧表示しません。

#### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- アプリケーションがデプロイ済みである必要があります。それ以外の場合は、エラーが発生します。
- *module* オプションは、スラッシュ (/) で区切って *application-name* オプションのあとに指定する必要があります。

#### ファイル

モジュールへのパスは、アプリケーションの *application.xml* ファイルの *module* 要素に指定されています。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--name=*env-entry-name*

一覧表示する環境エントリーの名称を指定します。

型：String

指定できる値を次に示します。

- 環境エントリーの名称

デフォルト値：アプリケーションに対して事前に設定されたすべての環境エントリー

*application-name/module*

アプリケーションの名称を指定します。名称は、list-applications サブコマンドを使用して取得できます。

- *module* は、環境エントリーが Java EE アプリケーションの Web モジュールに適用される場合だけ必要です。
- アプリケーション名に加えてモジュール名を指定する場合は、myApp/myWebModule.war となります。

型：String

指定できる値を次に示します。

- *application-name*  
アプリケーション名を指定します。
- *module*  
モジュール名を指定します。

デフォルト値：なし

## 使用例

次の例では、set-web-env-entry サブコマンドを使用して Web アプリケーションhello に設定されたすべての環境エントリーを一覧表示します。

```
asadmin list-web-env-entry hello
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.15 redeploy

コンポーネントのリデプロイ

### 書式

```
asadmin [asadmin-options] redeploy [--help]
  --name component_name
  [--upload={true|false}]
  [--retrieve local_dirpath]
  [--deploymentplan deployment_plan]
  [--altd alternate_deploymentdescriptor]
  [--runtimealtd runtime_alternate_deploymentdescriptor]
  [--deploymentorder deployment_order]
  [--enabled={true|false}]
  [--generateterminstubs={false|true}]
  [--contextroot context_root]
  [--precompilejsp={true|false}]
  [--virtualservers virtual_servers]
  [--libraries jar_file [, jar_file]...]
  [--target target]
  [--type pkg-type]
  [--properties name=value [:name=value]...]
  file_archive | filepath
```

### 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

### 機能

redeploy サブコマンドは、デプロイ済みまたは既存のエンタープライズアプリケーション、Web アプリケーション、EJB モジュール、リソースアダプターアーカイブ、またはアプリケーションクライアントのモジュールをリデプロイします。

redeploy サブコマンドは、アプリケーションが最初にデプロイされたときの設定およびほかのオプションを保存します。

このサブコマンドはリモートモードだけサポートします。

アプリケーションのデプロイの実行時にドメイン管理サーバで OutOfMemoryError が発生した場合、ドメイン管理サーバの Java ヒープが不足している可能性があります。ドメイン管理サーバの Java ヒープサイズは、デプロイするアプリケーションアーカイブのサイズを考慮したサイズにしてください。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

- *file\_archive* または *filepath* には、コマンドでデプロイされる入力ファイルを与えます。
- `--retrieve` オプションを使用すると、`--retrieve` オプションで指定したファイルパスにクライアントスタブの JAR ファイルを生成します。
- `--deploymentplan` オプションには、Java EE RI の DD を含む入力 JAR ファイルを与えます。
- `--altdd` オプションには、標準 DD を含む入力 JAR ファイルを与えます。
- `--runtimealtdd` オプションには、Java EE RI の DD を与えます。
- `--generatemitstubs` オプションを使用すると、静的 RMI-IIOP スタブが生成され、`client.jar` に包含されます。
- `--libraries` オプションには、デプロイされたアプリケーションで利用できる入力ライブラリー JAR ファイルを与えます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--virtualservers virtual_servers`

仮想サーバ ID を指定します。複数の ID はコンマ (,) で区切ります。

型：String

指定できる値を次に示します。

- *仮想サーバ ID*

デフォルト値：なし

`--contextroot context_root`

アプリケーションのコンテキストルート指定します。アーカイブが Web モジュールの場合にだけ有効です。

型：String

指定できる値を次に示します。

- *Web モジュールのコンテキストルート*

デフォルト値：Web モジュールのアーカイブファイル (拡張子なし)

`--precompilejsp={true|false}`

デプロイ中に JSP を事前コンパイルするかどうかを指定します。

コンパイルエラー発生時、エラー内容によっては原因のファイル名が表示されない場合があります。

この場合は、`jspc` コマンドによって `jsp` コマンドを一つ一つコンパイルすることで、原因のファイル特定してください。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--name component_name`

デプロイするコンポーネントの名称を指定します。

名称にはオプションでバージョン識別子を入れることができます。この場合、名前の後にコロン (:) で区切って追加します。バージョン識別子は、文字または数値で始まる必要があります。使用できる文字は、英数字、アンダースコア (\_)、ハイフン (-) およびピリオド (.) です。

型：String

指定できる値を次に示します。

- コンポーネント名

デフォルト値：なし

`--upload={true|false}`

ファイルをドメイン管理サーバにアップロードするかどうかを指定します。

`filepath` にディレクトリーが指定された場合、このオプションは無視されます。

型：Boolean

指定できる値を次に示します。

- false

ファイルをアップロードしないで、指定したファイル名でファイルへのアクセスを試みます。ドメイン管理サーバがファイルにアクセスできない場合、このサブコマンドは失敗します。

例えば、ドメイン管理サーバが管理ユーザーとは異なるユーザーとして稼働している可能性があり、ファイルへのアクセス権がドメイン管理サーバにない場合があります。この状況では、`--upload` オプションを `false` に設定すると、サブコマンドは失敗します。

- true

ネットワーク接続経由でファイルをドメイン管理サーバにアップロードします。

デフォルト値：

- サブコマンドを実行するホスト上にドメイン管理サーバがある場合  
false
- リモートホスト上にドメイン管理サーバがある場合  
true

`--retrieve local_dirpath`

サーバマシンからローカルディレクトリーにクライアントスタブの JAR ファイルを取得します。

型：String

指定できる値を次に示します。

- クライアントスタブの JAR ファイルの格納先ディレクトリーパス

デフォルト値：なし

#### `--deploymentplan deployment_plan`

Java EE RI の DD が含まれた JAR ファイルであるデプロイメントプランをデプロイします。

純粹な EAR ファイルをデプロイするときにこのオプションを指定します。純粹な EAR ファイルとは、Java EE Server の記述子がない EAR です。

型：String

指定できる値を次に示します。

- JAR ファイル名

デフォルト値：なし

#### `--altdtd alternate_deploymentdescriptor`

アプリケーションアーカイブの外にある標準 DD を使用してアプリケーションをデプロイします。

代替となる DD は、アーカイブに含まれる最高位の DD をオーバーライドします。例えば EAR の場合は、`--altdtd` オプションは `application.xml` ファイルをオーバーライドします。スタンドアロンモジュールの場合、`--altdtd` オプションは `web.xml` などの最上位のモジュール記述子をオーバーライドします。

型：String

指定できる値を次に示します。

- DD ファイルへのパス

デフォルト値：なし

#### `--runtimealtdtd runtime_alternate_deploymentdescriptor`

アプリケーションアーカイブの外にある Java EE RI の DD を使用してアプリケーションをデプロイします。

代替となる DD は、アーカイブに含まれる最高位の DD をオーバーライドします。

スタンドアロンモジュールの場合、`-runtimealtdtd` オプションは `glassfish-web.xml` などの最上位のモジュール記述子をオーバーライドします。Java EE Server の DD だけに適用されます (`glassfish-*.xml`)。代替となる DD ファイルの名称は、`glassfish-`で始まる必要があります。非推奨の `sun-*.xml` DD には適用されません。

型：String

指定できる値を次に示します。

- 代替となる DD ファイルへの絶対パスまたは相対パスを指定します。

相対パスは、*Application Server* インストールディレクトリー/`javaee/glassfish/bin` です。

デフォルト値：なし

#### `--deploymentorder deployment_order`

アプリケーションのデプロイ順序を指定します。

小さい数値のアプリケーションは、大きい数値のアプリケーションの前にロードされます。

2つのアプリケーションのデプロイ順序が同じ場合、最初にデプロイされたアプリケーションがサーバ起動時に先にロードされます。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：100

`--enabled={true|false}`

ユーザーがアプリケーションにアクセスできるかどうかを指定します。

対象ドメインにデプロイする場合、特定のインスタンスまたはクラスターへのデプロイはしないため、このオプションは無視されます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--generatermistubs={false|true}`

true に設定すると、静的 RMI-IIOP スタブが生成され、`client.jar` に追加されます。false に設定すると、スタブは生成されません。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--libraries jar_file[,jar_file]...`

`add-library` サブコマンドで Java EE Server に追加したライブラリー JAR ファイルをコンマ区切りで指定します。ライブラリー JAR ファイルは、`domain-dir/lib/applibs` に対する相対パスで指定します。

型：String

指定できる値を次に示します。

- コンマ区切りのライブラリー JAR ファイルのリスト

デフォルト値：なし

`--target target`

デプロイ先の対象を指定します。

ドメインが初回デプロイの対象の場合、アプリケーションはドメインにデプロイされますが、サーバーインスタンスおよびクラスターはアプリケーションを参照しません。

型：String

指定できる値を次に示します。

- server



ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- **domain**

ドメインを対象とします。

ドメインが初回デプロイの対象の場合、アプリケーションはドメインにデプロイされますが、サーバインスタンスおよびクラスターはアプリケーションを参照しません。

ドメインがリデプロイの対象で、アプリケーションを参照するクラスターまたはサーバインスタンスの動的構成が有効になっている場合、参照しているクラスターまたはサーバインスタンスはアプリケーションの新しいバージョンを自動的に取得します。

動的構成が無効な状態でリデプロイする場合、参照しているクラスターまたはサーバインスタンスは、クラスター化されたまたはスタンドアロンのサーバインスタンスが再起動するまでアプリケーションの新しいバージョンを取得しません。

- *cluster\_name*

指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--type pkg-type**

デプロイするコンポーネントのアーカイブの形式を指定します。

型：String

指定できる値を次に示します。

- car

コンポーネントは CAR ファイルとしてパッケージされます。

- ear

コンポーネントは EAR ファイルとしてパッケージされます。

- ejb

コンポーネントは JAR ファイルとしてパッケージされた EJB です。

- rar

コンポーネントは RAR ファイルとしてパッケージされます。

- war

コンポーネントは WAR ファイルとしてパッケージされます。

デフォルト値：なし

**{--properties | --property} name=value[:name=value]...**

デプロイのプロパティをプロパティ名と値のペアで指定します。

使用できるプロパティは、デプロイまたはリデプロイされているコンポーネントの実装で決まります。--properties オプションと--property オプションは同じです。指定するプロパティの数に関係なく、どちらのオプションも利用できます。

型：String

デプロイには次のプロパティを指定できます。

- `preserveAppScopedResources={false|true}`

`true` に設定すると、アプリケーションスコープのリソースを保持し、リデプロイ中に復元されます。その他の利用できるプロパティは、リデプロイされているコンポーネントの実装で決まります。その他のプロパティは修飾プロセスで使用されます。

型：Boolean

指定できる値を次に示します。

`true`

`false`

デフォルト値：`false`

`file_archive|filepath`

リデプロイするアプリケーションを含むアーカイブへのパスを指定します。相対パスまたは絶対パスのどちらでも指定できます。相対パスは、`Application Server` インストールディレクトリー/`javaee/glassfish/bin` です。

ディレクトリーまたはアーカイブファイルを指定できます。

型：String

指定できる値を次に示します。

- アーカイブへのパス

デフォルト値：なし

## 使用例

次の例では、Web アプリケーション `hellodir` をリデプロイします。アプリケーションは、`instance1` という名称のサーバインスタンス上のディレクトリーから初期設定でリデプロイされます。

```
asadmin redeploy --target instance1 --name hellodir
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.16 set-web-context-param

Web アプリケーションまたはモジュールの `ServletContext` 初期化パラメーターの設定

## 書式

```
asadmin [asadmin-options] set-web-context-param
  [--help] --name=context-param-name
  [--value=value | --ignoredescriptoritem={false|true}]
  [--description description] application-name[/module]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

set-web-context-param サブコマンドは、次のどちらかの項目の ServletContext 初期化パラメーターを設定します。

- デプロイ済みの Web アプリケーション
- デプロイ済みの Java EE アプリケーション内の Web モジュール

このサブコマンドによって、デプロイ済みのアプリケーションの DD を変更したり、アプリケーションを再パッケージおよびリデプロイしたりすることなく、アプリケーションの設定を変更できます。

デプロイ済みのアプリケーションの ServletContext 初期化パラメーターを設定した場合、ドメイン管理サーバまたはサーバインスタンスを再起動する必要があります。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

アプリケーションがデプロイされている必要があります。

## ファイル

モジュールへのパスは、アプリケーションの *application.xml* ファイルの *module* 要素に指定されます。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--name=*context-param-name*

設定する ServletContext 初期化パラメーターの名称を指定します。

型：String

指定できる値を次に示します。

- パラメーター名

デフォルト値：なし

`--value=value`

ServletContext 初期化パラメーターに設定する値を指定します。

`--value` オプションまたは `--ignoredescriptoritem` オプションを設定する必要があります。

型：String

指定できる値を次に示します。

- パラメーター値

デフォルト値：なし

`--ignoredescriptoritem={false|true}`

ServletContext 初期化パラメーターがアプリケーションの DD に設定されている場合、そのパラメーターを無視するかどうかを指定します。パラメーターを無視する場合、アプリケーションは DD にパラメーターが設定されたことがないかのように動作します。この状況でのアプリケーションの動作は、アプリケーションによって異なります。

型：Boolean

指定できる値を次に示します。

- `false`

値は無視されません。

- `true`

値は無視されます。`--value` オプションまたは `--ignoredescriptoritem` オプションを設定する必要があります。

`set-web-context-param` サブコマンドで以前に設定した ServletContext 初期化パラメーターの設定を取り消す場合、`--ignoredescriptoritem` オプションを使用しないでください。代わりに `unset-web-context-param` サブコマンドを使用してください。

デフォルト値：`false`

`--description description`

設定するコンテキストパラメーターの説明となるテキストを指定します。

型：String

指定できる値を次に示します。

- 環境エントリーの説明

デフォルト値：なし

`application-name`

アプリケーションの名称を指定します。この名称は、`list-applications` サブコマンドで取得できます。

型：String

指定できる値を次に示します。

- アプリケーション名

デフォルト値：なし

#### *module*

アプリケーションの EAR ファイル内のモジュールへの相対パスを指定します。モジュールへのパスは、アプリケーションの `application.xml` ファイルの `module` 要素に指定されます。`module` は、`ServletContext` 初期化パラメーターを Java EE アプリケーションの Web モジュールに適用する場合にだけ必要です。モジュールは、`myApp/myWebModule.war` の形式で指定します。

モジュールは、`application-name` の後にスラッシュ (/) で区切って指定する必要があります。

型：String

指定できる値を次に示します。

- アプリケーション内のモジュール名

デフォルト値：なし

## 使用例

次の例では、Web アプリケーション `basic-ezcomp` の `ServletContext` 初期化パラメーター `javax.faces.STATE_SAVING_METHOD` を `client` に設定します。このパラメーターには、「The location where the application's state is preserved」という説明が設定されています。

```
asadmin set-web-context-param --name=javax.faces.STATE_SAVING_METHOD
--description "The location where the application's state is preserved"
--value=client basic-ezcomp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.17 set-web-env-entry

Web アプリケーションまたはモジュールの環境エントリーの設定

### 書式

```
asadmin [asadmin-options] set-web-env-entry [--help]
--name=env-entry-name --type=env-entry-type
{--value=value | --ignoredescriptoritem={true|false}}
[--description description] application-name[/module]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`set-web-env-entry` サブコマンドは、次のどちらかの項目の環境エントリーを設定します。

- デプロイ済みの Web アプリケーション
- デプロイ済みの Java EE アプリケーション内の Web モジュール

このサブコマンドによって、デプロイ済みのアプリケーションの DD を変更したり、アプリケーションを再パッケージおよびリデプロイしたりすることなく、アプリケーションの設定を変更できます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

アプリケーションがデプロイされている必要があります。

## ファイル

モジュールへのパスは、アプリケーションの `application.xml` ファイルの `module` 要素に指定されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--name=env-entry-name`

設定する環境エントリーの名称を指定します。この名称は `java:comp/env` コンテキストに相対する JNDI 名です。この名称はデプロイ機能内で一意でなければなりません。

型：String

指定できる値を次に示します。

- 環境エントリー名

デフォルト値：なし

`--type=env-entry-type`

アプリケーションのコードが求める環境エントリー値の完全修飾 Java タイプを指定します。

型：String

指定できる値を次に示します。

- `java.lang.Boolean`

- java.lang.Byte
- java.lang.Character
- java.lang.Double
- java.lang.Float
- java.lang.Integer
- java.lang.Long
- java.lang.Short
- java.lang.String

デフォルト値：なし

**--value=*value***

環境エンタリーに設定する値を指定します。

--type が java.lang.Character の場合、値は 1 文字でなければなりません。そうでない場合は、指定したタイプのコンストラクターに有効な文字列でなければなりません。

--value オプションまたは --ignoredescriptoritem オプションを設定する必要があります。

型：String

指定できる値を次に示します。

- 環境エンタリー値

デフォルト値：なし

**--ignoredescriptoritem={true|false}**

環境エンタリーがアプリケーションの DD に設定されている場合、そのエンタリーを無視するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- false

値は無視されません。

- true

値は無視されます。--value オプションまたは --ignoredescriptoritem オプションを設定する必要があります。

set-web-env-entry サブコマンドで以前に設定した環境エンタリーの設定を取り消す場合、--ignoredescriptoritem オプションを使用しないでください。その用途では、代わりに unset-web-env-entry サブコマンドを使用してください。

デフォルト値：false

**--description *description***

設定する環境エンタリーの説明となるテキストを指定します。

型：String

指定できる値を次に示します。

- パラメーターの説明

デフォルト値：なし

#### *application-name*

アプリケーションの名称を指定します。この名称は、`list-applications` サブコマンドで取得できます。

型：String

指定できる値を次に示します。

- アプリケーション名

デフォルト値：なし

#### *module*

アプリケーションの EAR ファイル内のモジュールへの相対パスを指定します。モジュールへのパスは、アプリケーションの `application.xml` ファイルの `module` 要素に指定されます。`module` は、環境エントリを Java EE アプリケーションの Web モジュールに適用する場合にだけ必要です。モジュールは `myApp/myWebModule.war` の形式で指定します。

モジュールは、`application-name` の後にスラッシュ (/) で区切って指定する必要があります。

型：String

指定できる値を次に示します。

- アプリケーション内のモジュール名

デフォルト値：なし

## 使用例

次の例では、アプリケーション `hello` の環境エントリ `Hello User` を `techscribe` に設定します。このエントリの Java タイプは `java.lang.String` です。

```
asadmin set-web-env-entry --name="Hello User"  
--type=java.lang.String --value=techscribe  
--description "User authentication for Hello application" hello
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.18 show-component-status

デプロイ済みのコンポーネントの状態の表示



## 書式

```
asadmin [asadmin-options] show-component-status [--help]
        [--target target] component-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

show-component-status サブコマンドは、デプロイ済みのコンポーネントの状態（有効または無効）を取得します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

表示するコンポーネントの状態の対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *domain*  
ドメインを対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

*component-name*

状況を一覧表示するコンポーネントの名前を指定します。

名前にはオプションでバージョン識別子を入れることができます。このバージョン識別子は名前の後に続き、コロン(:)で区切ります。バージョン識別子は、文字または数値で始まる必要があります。

使用できる文字は、英数字のほかに、アンダースコア (`_`)、ハイフン (`-`) およびピリオド (`.`) です。複数のバージョンを一覧表示する場合、アスタリスク (`*`) をワイルドカードとして使用できます。

型：String

指定できる値を次に示します。

- コンポーネント名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサービンスタンスについて、MEjbApp コンポーネントの状態を取得します。

```
asadmin show-component-status --target instance1 MEjbApp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.19 undeploy

デプロイ済みのコンポーネントの削除

### 書式

```
asadmin [asadmin-options] undeploy [--help]
  [--target target]
  [--cascade={false|true}] name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

`undeploy` サブコマンドは、デプロイ済みのアプリケーションまたはモジュールをアンインストールし、リポジトリから削除します。

このサブコマンドはリモートモードだけサポートします。

タイムアウトなどでデータベースへのアクセスが失敗することによって、EJB タイマーアプリケーションのアンデプロイに失敗した場合、そのアプリケーションがデプロイできなくなります。この場合は、次の手順でEJB タイマーアプリケーションをデプロイし直してください。

1. サーバインスタンスを再起動します。
2. EJB タイマーアプリケーションをアンデプロイします。
3. EJB タイマーアプリケーションをデプロイします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--cascade={false|true}`

関連リソースを削除するかどうかを指定します。

`true` に設定した場合、アンデプロイするリソースアダプターに関連づけられたすべての接続プールやコネクタリソースを削除します。

`false` に設定した場合、リソースアダプターに関連づけられているプールやリソースが残っていると、アンデプロイが失敗します。その場合、それらのプールやリソースを明示的に削除するか、オプションを `true` に設定する必要があります。

このオプションを `false` に設定し、リソースアダプターに関連づけられているプールやリソースが残っていない場合、リソースアダプターはアンデプロイされます。

このオプションはコネクタ（リソースアダプター）とアプリケーションに適用されます。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--target target`

削除するデプロイの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`

ドメインを対象とします。

- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### *name*

デプロイ済みのコンポーネントの名前を指定します。

名前にはオプションでバージョン識別子を入れることができます。このバージョン識別子は名前の後に続き、コロン (:) で区切ります。バージョン識別子は、文字または数値で始まる必要があります。使用できる文字は、英数字のほかに、アンダースコア (\_)、ハイフン (-) およびピリオド (.) です。複数のバージョンを削除する場合、アスタリスク (\*) をワイルドカードとして使用できます。

型：String

指定できる値を次に示します。

- コンポーネント名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンス上の、Cart.ear という名称のエンタープライズアプリケーションをアンデプロイします。

```
asadmin undeploy --target instance1 Cart
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.20 unset-web-context-param

Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの設定の解除

### 書式

```
asadmin [asadmin-options] unset-web-context-param [--help]  
--name=context-param-name application-name[/module]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`unset-web-context-param` サブコマンドは、次のどちらかの項目の `ServletContext` 初期化パラメーターの設定を解除します。

- デプロイ済みの Web アプリケーション
- デプロイ済みの Java EE アプリケーション内の Web モジュール

パラメーターの設定が解除されると、アプリケーションの DD に設定された値があれば、パラメーターの値はその値に戻ります。

このサブコマンドによって、デプロイ済みのアプリケーションの DD を変更したり、アプリケーションを再パッケージおよびリデプロイしたりすることなく、アプリケーションの設定を変更できます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

- 環境エントリーが設定されている必要があります。
- アプリケーションがデプロイされている必要があります。
- ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

モジュールへのパスは、アプリケーションの `application.xml` ファイルの `module` 要素に指定されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--name=context-param-name`

設定を解除する `ServletContext` 初期化パラメーターの名称を指定します。

型：String

指定できる値を次に示します。

- 変数名

デフォルト値：なし

`application-name`

アプリケーションの名称を指定します。

型：String

指定できる値を次に示します。

- アプリケーション名

デフォルト値：なし

#### *module*

アプリケーションの EAR ファイル内のモジュールへの相対パスを指定します。モジュールへのパスは、アプリケーションの `application.xml` ファイルの `module` 要素に指定されます。`module` は、`ServletContext` 初期化パラメーターを Java EE アプリケーションの Web モジュールに適用する場合にだけ必要です。

指定する場合、`module` は `application-name` の後にスラッシュ (/) で区切って指定する必要があります。例えば、`myApp` アプリケーションの `application.xml` ファイルには、Web モジュールを次のように指定します。

```
<module>
<web>
<web-uri>myWebModule.war</web-uri>
</web>
</module>
```

モジュールは、`myApp/myWebModule.war` のような形式となります。

型：String

指定できる値を次に示します。

- アプリケーションモジュール

デフォルト値：なし

## 使用例

次の例では、Web アプリケーション `basic-ezcomp` の `ServletContext` 初期化パラメーター `javax.faces.STATE_SAVING_METHOD` の設定を解除します。パラメーターは、アプリケーションの DD に定義された値があれば、その値に戻ります。

```
asadmin unset-web-context-param --name=javax.faces.STATE_SAVING_METHOD
basic-ezcomp
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.8.21 unset-web-env-entry

Web アプリケーションまたはモジュールの環境エントリーの設定の解除

### 書式

```
asadmin [asadmin-options] unset-web-env-entry [--help]
        --name=env-entry-name application-name[/module]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

unset-web-env-entry サブコマンドは、次のどちらかの項目の環境エントリーの設定を解除します。

- デプロイ済みの Web アプリケーション
- デプロイ済みの Java EE アプリケーション内の Web モジュール

エントリーの設定が解除されると、アプリケーションの DD に設定された値があれば、エントリーの値はその値に戻ります。

このサブコマンドによって、デプロイ済みのアプリケーションの DD を変更したり、アプリケーションを再パッケージおよびリデプロイしたりすることなく、アプリケーションの設定を変更できます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

アプリケーションがデプロイされている必要があります。

エントリーが設定済みである必要があります。

ドメイン管理サーバは、稼働状態である必要があります。

### 環境変数

モジュールへのパスは、アプリケーションの *application.xml* ファイルの *module* 要素に指定されます。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

`--name=env-entry-name`

設定を解除する環境エントリーの名称を指定します。この名前はjava:comp/env コンテキストに相対する JNDI 名です。この名前はデプロイ機能内で一意でなければなりません。

型：String

指定できる値を次に示します。

- 環境エントリー名

デフォルト値：なし

*application-name*

アプリケーションの名称を指定します。

型：String

指定できる値を次に示します。

- アプリケーション名

デフォルト値：なし

*module*

アプリケーションの EAR ファイル内のモジュールへの相対パスを指定します。モジュールへのパスは、アプリケーションの application.xml ファイルの module 要素に指定されます。

*module* は、環境エントリーを Java EE アプリケーションの Web モジュールに適用する場合にだけ必要です。

指定する場合、*module* は application-name の後にスラッシュ (/) で区切って指定する必要があります。例えば、myApp/myWebModule.war のような形式です。

型：String

指定できる値を次に示します。

- アプリケーションモジュール名

デフォルト値：なし

## 使用例

次の例では、アプリケーション hello の環境エントリー Hello User の設定を解除します。エントリーは、アプリケーションの DD に定義された値があれば、その値に戻ります。

```
asadmin unset-web-env-entry --name="Hello User" hello
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.9 セキュリティー管理で使用するコマンド

---

セキュリティー管理で使用するコマンドの書式、機能などについて説明します。

### 2.9.1 change-master-password

マスターパスワードの変更

#### 書式

```
asadmin [asadmin-options] change-master-password [--help]
  [--nodedir node-dir] [--domaindir domain-dir]
  [--savemasterpassword={false|true}] [domain-name |node-name]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

change-master-password サブコマンドは、マスターパスワードを変更するのに使用します。このサブコマンドでは、古いマスターパスワードと新しいマスターパスワードの入力が対話型で求められます。

このサブコマンドは、サーバが停止しているときだけ動作します。

分散環境では、ドメイン内の各マシン上でコマンドを実行する必要があります。

#### 前提条件

ドメイン管理サーバまたはサーバインスタンスは、change-master-password サブコマンドの実行前に停止する必要があります。

#### 環境変数

AS\_DEF\_DOMAINS\_PATH

#### ファイル

change-master-password サブコマンドの--savemasterpassword オプションがtrue に設定されている場合、マスターパスワードが変更され、domains/domain-dir ディレクトリー内のmaster-password ファイルに書き込まれます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--nodedir node-dir`

パスワードを変更するノードのインスタンスを含むディレクトリーの名称を指定します。このオプションを実行すると、変更がドメイン全体に適用されます。

型：String

指定できる値を次に示します。

- ノードが存在するディレクトリーのパス

デフォルト値：なし

`--domaindir domain-dir`

この操作で使用するドメインディレクトリーの名称を指定します。

型：String

指定できる値を次に示します。

- ドメインディレクトリー名

デフォルト値：asenv.bat ファイルまたはasenv.conf ファイルに定義される環境変数

`$SAS_DEF_DOMAINS_PATH`

`--savemasterpassword={false|true}`

ファイルシステムへのマスターパスワードの書き込みをユーザーに要求するかどうかを確認します。`start-domain` サブコマンドを使うと、ユーザーにパスワード入力を要求しないでサーバを起動できるようになります。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`domain-name | node-name`

パスワードを変更するドメインまたはノードの名称を指定します。ドメインが1つだけの場合、このオプションは任意です。

型：String

指定できる値を次に示します。

- ドメイン名、またはノード名

デフォルト値：なし

## 使用例

次の例では、domain44ps ドメインのマスターパスワードを変更します。

```
asadmin change-master-password domain44ps
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.2 create-auth-realm

名前付き認証レルムの追加

### 書式

```
asadmin [asadmin-options] create-auth-realm [--help]
  --classname realm_class [--property name=value[:name=value]...]
  [--target target_name] auth_realm_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

create-auth-realm サブコマンドは、名前付き認証レルムを追加します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### ファイル

各レルムに対して、ユーザー名、パスワード、グループ名を保管するキーファイルを指定できます。デフォルトファイルはdomain-dir/config/keyfile です。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target_name`

作成するレルムの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--classname realm_class`

このレルムを実装する Java クラスを指定します。

型：String

指定できる値を次に示します。

- `com.sun.enterprise.security.auth.realm.file.FileRealm`
- `com.sun.enterprise.security.auth.realm.certificate.CertificateRealm`
- `com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm`
- `com.sun.enterprise.security.auth.realm.ldap.LDAPRealm`
- `com.sun.appserv.security.AppservPasswordLoginModule` クラスを実装する java クラスの名称

デフォルト値：なし

`--property name=value[:name=value]...`

認証レルムを構成するためのオプションのプロパティ名と値のペアを指定します。

指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

プロパティ名と `value` に指定できる値を次に示します。

次のプロパティは、FileRealm、CertificateRealm、JDBCRealm、および LDAPRealm を含むサポートされているすべてのレルムに共通です。

`jaas-context=value`

Java Authentication and Authorization Service (JAAS) コンテキストを指定します。

型：String

デフォルト値：なし

**assign-groups=*value***

このプロパティを指定する場合、*value* にはグループ名がコンマで区切られたリストを記述します。有効な証明書を提示するすべてのクライアントには、Web および EJB コンテナでの認可決定のために、これらのグループに対するメンバーシップが割り当てられます。

型：String

デフォルト値：なし

次のプロパティは、FileRealm に指定できます。

**file=*value***

ユーザー名、パスワードおよびグループ名を保存するファイルを指定します。

型：String

デフォルト値：domain-dir/config/keyfile

次のプロパティは、CertificateRealm に指定できます。

**LoginModule=*value***

認証の実行に使用する JAAS LoginModule の名称を指定します。JAAS LoginModule を使用するには、最初に `javax.security.auth.spi.LoginModule` インターフェースの実装を作成してから、モジュールを `jaas-context` にプラグインする必要があります。

型：String

デフォルト値：なし

次のプロパティは、JDBCRealm に指定できます。

**datasource-jndi=*value***

データベースの `jdbc-resource` の `jndi-name` を指定します。

型：String

デフォルト値：なし

**user-table=*value***

データベース内のユーザーテーブルの名称を指定します。

型：String

デフォルト値：なし

**user-name-column=*value***

データベースのユーザーテーブル内のユーザー名列の名称を指定します。

型：String

デフォルト値：なし

**password-column=*value***

データベースのユーザーテーブル内のパスワード列の名称を指定します。

型：String

デフォルト値：なし

`group-table=value`

データベース内のグループテーブルの名称を指定します。

JDBCRealm クラスの認証レルムのグループテーブルを指定します。

型：String

デフォルト値：なし

`group-name-column=value`

データベースのグループテーブル内のグループ名列の名称を指定します。

型：String

デフォルト値：なし

`db-user=value`

`jdbc-connection-pool` の代わりに、レルムにデータベースユーザー名を指定できます。これによって、ほかのアプリケーションによるデータベースの検索、接続の取得、ユーザーテーブルの参照を回避できます。デフォルトでは、`jdbc-connection-pool` 構成が使用されます。

型：String

デフォルト値：なし

`db-password=value`

`jdbc-connection-pool` の代わりに、レルムにデータベースパスワードを指定できます。これによって、ほかのアプリケーションによるデータベースの検索、接続の取得、ユーザーテーブルの参照を回避できます。デフォルトでは、`jdbc-connection-pool` 構成が使用されます。

型：String

デフォルト値：なし

`digest-algorithm=value`

メッセージダイジェストアルゴリズムを指定します。デフォルトのアルゴリズムは SHA-256 です。JDK でサポートされているアルゴリズム、または `none` を使用できます。アプリケーションが MD5 アルゴリズムに依存している場合は、`asadmin set` サブコマンドを使用してデフォルトの SHA-25 アルゴリズムを上書きできます。

```
asadmin set
server.security-service.property.default-digest-algorithm=MD5
```

`asadmin` の `get` サブコマンドを使用して、どのアルゴリズムが現在使用されているか確認できます。

```
asadmin get
server.security-service.property.default-digest-algorithm
```

型：String

デフォルト値：SHA-256

`digestrealm-password-enc-algorithm=value`

データベースに格納されたパスワードを暗号化するためのアルゴリズムを指定します。パスワード暗号化アルゴリズムを指定しないと、セキュリティー上のリスクが発生します。

型：String

デフォルト値：なし

**encoding=***value*

エンコーディングを指定します。

型：String

指定できる値を次に示します。

- Hex
- Base64

デフォルト値：digest-algorithm が指定された場合、デフォルトでは Hex が指定されます。その他の場合は、デフォルトでは文字コードは指定されません。

**charset=***value*

メッセージダイジェストアルゴリズムの charset を指定します。

型：String

デフォルト値：なし

次のプロパティは、LDAPRealm に指定できます。

**directory=***value*

サーバに対する LDAP URL を指定します。

型：String

デフォルト値：なし

**base-dn=***value*

ユーザーデータの位置の LDAP ベース DN を指定します。ツリー範囲検索が実行されるため、このベース DN はユーザーデータよりも上の任意のレベルになります。検索ツリーが小さくなるほど、パフォーマンスが向上します。

型：String

デフォルト値：なし

**search-filter=***value*

ユーザーの検索に使用する検索フィルターを指定します。

型：String

デフォルト値：uid=%s (%s はサブジェクト名に展開されます)

**group-base-dn=***value*

グループデータの位置のベース DN を指定します。デフォルトでは base-dn と同じですが、必要に応じて調整できます。

型：String

デフォルト値：なし

**group-search-filter=***value*

ユーザーのグループを検索する検索フィルターを指定します。

型：String

デフォルト値：unique-member=%d (%d はユーザーの要素 DN に展開されます)

`group-target=value`

グループ名のエントリーを含む LDAP 属性名を指定します。デフォルト値は CN です。

型：String

デフォルト値：CN

`search-bind-dn=value`

`search-filter` 検索を実行する際にディレクトリーの認証で使用するオプションの DN を指定します。匿名検索を許可しないディレクトリーにだけ必要です。

型：String

デフォルト値：なし

`search-bind-password=value`

`search-bind-dn` で指定された DN の LDAP パスワードを指定します。

型：String

デフォルト値：なし

`auth_realm_name`

レルムの短縮名を指定します。この名称はレルムの参照に使用します。

型：String

指定できる値を次に示します。

- レルム名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、認証レルムを新規に作成します。

```
asadmin create-auth-realm --target instance1
--classname com.sun.enterprise.security.auth.realm.file.FileRealm
--property file=${com.sun.aas.instanceRoot}/config/
admin-keyfile:jaas-context=fileRealm file
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.3 create-file-user

ファイルユーザーの作成



## 書式

```
asadmin [asadmin-options] create-file-user [--help]
  [--authrealmname auth_realm_name] [--target target]
  [--groups user_groups[:user_groups]...] user_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

create-file-user サブコマンドは、次のようなキーファイルのエントリーを作成します。

- 指定のユーザー名
- パスワード
- グループ

複数のグループは、コロン (:) で区切って作成できます。

--authrealmname オプションが指定されていない場合、エントリーはデフォルトのレルムに対してキーファイルに作成されます。--authrealmname オプションが指定されている場合、エントリーはauth\_realm\_name の値を使用してキーファイルに作成されます。

asadmin コマンドの--passwordfile オプションを使用すると、ユーザーのパスワードを指定できます。パスワードファイルのエントリーは、AS\_ADMIN\_USERPASSWORD=user-password の形式にする必要があります。セキュア管理が有効で、作成するユーザーが管理ユーザーの場合、パスワードを指定しないと、このサブコマンドは失敗します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

create-file-user サブコマンドを--passwordfile オプションを指定して実行する場合、パスワードファイルのエントリーはAS\_ADMIN\_USERPASSWORD=user-password の形式にする必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

`--target target`

このサブコマンドが動作する対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--groups user_groups[:user_groups]...`

このファイルユーザーに関連づけるグループを指定します。

型：String

指定できる値を次に示します。

- グループ名

デフォルト値：なし

`--authrealmname auth_realm_name`

新しいユーザーが作成されるレルム名を指定します。このオプションを指定しない場合、ユーザーはレルム名「file」に作成されます。

型：String

指定できる値を次に示します。

- レルム名

デフォルト値：file

`user_name`

作成するファイルユーザーの名称を指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、ファイルユーザーを作成します。

```
asadmin create-file-user --target instance1 --groups staff:manager
--authrealmname auth-realm1 sample_user
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.4 create-jacc-provider

任意の認証モジュールで使用できる JACC プロバイダーの作成

### 書式

```
asadmin [asadmin-options] create-jacc-provider [--help]
--policyproviderclass pol-provider-class
--policyconfigfactoryclass pc-factory-class
[--property name=value[:name=value]...]
[--target target] jacc-provider-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-jacc-provider` サブコマンドは、JSR-115 準拠の Java Authorization Contract for Containers (JACC) プロバイダーを作成します。JSR-115 準拠の JACC プロバイダーは、Java EE Server で実行するアプリケーションの認証に使用できます。

*Application Server* のデフォルトのインストール環境には、`default` および `simple` という名前の 2 つの JACC プロバイダーが含まれています。`create-jacc-provider` サブコマンドによって、任意の認証モジュールを追加できます。

このサブコマンドはリモートモードだけサポートします。

JACC プロバイダーは、`domain.xml` ファイルの `security-service` 要素内に `jacc-provider` 要素として作成されます。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

- JACC プロバイダーは、ドメインのdomain.xml ファイルのsecurity-service 要素内にjacc-provider 要素として作成されます。
- jacc-provider-name には、domain.xml ファイルのjacc-provider 要素を参照するために使用されるプロバイダーの名称を使用します。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--policyproviderclass *pol-provider-class*

java.security.Policy を実装するjavax.security.jacc.policy.provider の完全修飾クラス名を指定します。

型：String

指定できる値を次に示します。

- java.security.Policy インターフェースを実装するクラスの名称

デフォルト値：なし

--policyconfigfactoryclass *pc-factory-class*

プロバイダー固有のjavax.security.jacc.PolicyConfigurationFactory を実装するjavax.security.jacc.PolicyConfigurationFactory.provider の完全修飾クラス名を指定します。

型：String

指定できる値を次に示します。

- javax.security.jacc.PolicyConfigurationFactory インターフェースを実装するクラスの名称

デフォルト値：なし

--property *name=value[:name=value]...*

JACC プロバイダー構成時のオプションをプロパティ名と値のペアで指定します。指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

*name*

型：String

指定できる値を次に示します。

- repository

JACC ポリシーファイルを格納するディレクトリーを指定します。このプロパティは、Java EE Server の JACC プロバイダーではデフォルトで定義されません。

デフォルト値：なし

*value*

型：String

指定できる値を次に示します。

- JACC ポリシーファイルを格納するディレクトリーへのパス

デフォルト値：`${com.sun.aas.instanceRoot}/generated/policy` (default Java EE Server の JACC プロバイダーのデフォルトのディレクトリー)

`--target target`

作成する JACC プロバイダーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサービンスタンスを対象とします。

デフォルト値：server

`jacc-provider-name`

domain.xml の jacc-provider 要素を参照するために使用するプロバイダーの名称を指定します。

型：String

指定できる値を次に示します。

- プロバイダーの名称

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、JACC プロバイダーを作成します。

```
asadmin create-jacc-provider --target instance1
--policyproviderclass com.sun.enterprise.security.provider.PolicyWrapper
--policyconfigfactoryclass
com.sun.enterprise.security.provider.PolicyConfigurationFactoryImpl
testJACC
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.5 create-password-alias

パスワードエイリアスの作成

### 書式

```
asadmin [asadmin-options] create-password-alias [--help] aliasname
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-password-alias` サブコマンドは、パスワードに対するエイリアスを作成します。

エイリアスは、`#{ALIAS=aliasname}`という形式で指定します。エイリアス名に対応するパスワードは暗号化されて格納されます。

このサブコマンドは対話形式でも非対話形式でも実行できます。

- 対話形式で実行すると、エイリアスパスワードの入力と確認入力が必要です。
- 非対話形式で実行すると、`asadmin`ユーティリティーの`--passwordfile`オプションで渡されるファイルからエイリアスパスワードをコマンドが読み込みます。このファイルには、`AS_ADMIN_ALIASSPASSWORD=alias-password`という形式のエントリが必要です (`alias-password`はエイリアスパスワード)。このサブコマンドの非対話形式は、スクリプトでの使用が適しています。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### ファイル

`--passwordfile` オプションで渡されるファイルからエイリアスパスワードをサブコマンドが読み込みます。このファイルには、`AS_ADMIN_ALIASSPASSWORD=alias-password`という形式のエントリが必要です (`alias-password`はエイリアスパスワード)。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*aliasname*

パスワードエイリアスの名称を指定します。

型：String

指定できる値を次に示します。

- パスワード

デフォルト値：なし

## 使用例

次の例では、パスワードエイリアスを対話形式で作成します。

```
asadmin create-password-alias jmspassword-alias
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 注意事項

- 作成したパスワードエイリアスを適用するには、ドメイン管理サーバを再起動する必要があります。

## 2.9.6 delete-auth-realm

名前付き認証レルムの削除

## 書式

```
asadmin [asadmin-options] delete-auth-realm [--help] [--target target]  
auth_realm_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-auth-realm` サブコマンドは、名前付き認証レルムを削除します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストが表示されます。

`--target` *target*

削除する認証レルムの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`auth_realm_name`

レルムの名称を指定します。

型：String

指定できる値を次に示します。

- レルム名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、認証レルム `db` を削除します。

```
asadmin delete-auth-realm --target instance1 db
```



## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.7 delete-file-user

ファイルユーザーの削除

### 書式

```
asadmin [asadmin-options] delete-file-user [--help]
        [--authrealmname auth_realm_name] [--target target] username
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-file-user` サブコマンドは、指定したユーザー名に対するキーファイルのエントリーを削除します。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストが表示されます。

`--authrealmname` *auth\_realm\_name*

ユーザーの作成に使用された認証レルムの名称を指定します。

型：String

指定できる値を次に示します。

- 認証レルム名

デフォルト値：file

`--target` *target*

削除するファイルユーザーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *username*

ファイルユーザーの名称を指定します。

型：`String`

指定できる値を次に示します。

- ファイルユーザー名

デフォルト値：なし

## 使用例

次の例では、デフォルトのレルムfile から、`sample_user` という名称のユーザーを削除します。

```
asadmin delete-file-user sample_user
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.8 delete-jacc-provider

JACC プロバイダーの削除

### 書式

```
asadmin [asadmin-options] delete-jacc-provider [--help]  
        [--target target] jacc-provider-name
```

### 格納先

*Application Server*インストールディレクトリー/`javaee/glassfish/bin`

## 機能

`delete-jacc-provider` サブコマンドは、ドメインに対して定義された JACC プロバイダーを削除します。このサブコマンドは管理ユーザーだけが使用できます。JACC プロバイダーは、該当するドメインの `domain.xml` ファイルの `security-service` 要素内に `jacc-provider` 要素として定義されます。JACC プロバイダーは `create-jacc-provider` サブコマンドを使用して作成できます。

デフォルトの Java EE Server には、`default` および `simple` という名前の 2 つの JACC プロバイダーが含まれています。これらのデフォルトプロバイダーは削除できません。

Java EE Server で承認に使用する JACC プロバイダーは、`domain.xml` 内の `security-service` の `jacc-provider` 要素によって識別されます。したがって、`jacc-provider` プロバイダーを削除する場合は、`jacc-provider` を必ず `security-service` に存在するほかの JACC プロバイダーの名称に変更してください。

`jacc-provider` 要素を変更して別の JACC プロバイダーをポイントするようにした場合は、Java EE Server の再起動が必要です。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

JACC プロバイダーは、該当するドメインの `domain.xml` ファイルの `security-service` 要素内に `jacc-provider` 要素として定義されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除する JACC プロバイダーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`

指定した名称のサービンスタンスを対象とします。

デフォルト値：server

#### *jacc-provider-name*

JACC プロバイダーの名称を指定します。

型：String

指定できる値を次に示します。

- JACC プロバイダー名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスのデフォルトのドメインから、testJACC という名称の JACC プロバイダーを削除します。

```
asadmin delete-jacc-provider --target instance1 testJACC
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.9 delete-password-alias

パスワードエイリアスの削除

### 書式

```
asadmin [asadmin-options] delete-password-alias [--help]  
aliasname
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-password-alias サブコマンドは、パスワードエイリアスを削除します。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

*aliasname* には、`domain.xml` に記載されているパスワードエイリアスの中から、削除するパスワードエイリアスを指定します。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*aliasname*

`domain.xml` に記載されているパスワードエイリアスの中から、削除するパスワードエイリアスを指定します。

型：String

指定できる値を次に示します。

- パスワード

デフォルト値：なし

## 使用例

次の例では、`jdbcpassword-alias` という名称のパスワードエイリアスを削除します。

```
asadmin delete-password-alias
jdbcpassword-alias
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.10 disable-secure-admin

secure admin 機能の無効化

## 書式

```
asadmin [asadmin-options] disable-secure-admin [--help]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`disable-secure-admin` サブコマンドは、`secure admin` 機能がすでに有効な場合に、`secure admin` 機能を無効にします。

`secure admin` 機能を有効または無効にしたあとにドメインで動作しているサーバを再起動する必要があります。ドメイン管理サーバだけが動作しているときに `secure admin` 機能を有効または無効にし、ドメイン管理サーバを再起動した後で、ほかのインスタンスを起動するとより簡単です。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

## 使用例

次の例では、ドメインの `secure admin` 機能を無効にします。

```
asadmin disable-secure-admin
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

### 2.9.11 enable-secure-admin

`secure admin` 機能の有効化

## 書式

```
asadmin [asadmin-options] enable-secure-admin [--help]  
        [--adminalias=alias]  
        [--instancealias=alias]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`enable-secure-admin` サブコマンドは、`secure admin` 機能を有効にします（すでに有効になっていない場合）。

また、オプションでドメイン管理サーバからインスタンスへの管理メッセージに使用するエイリアス、またはインスタンスからドメイン管理サーバへの管理メッセージに使用するエイリアスを変更します。

ドメイン内のドメイン管理サーバとインスタンスで、相互に送信するメッセージの暗号化に SSL 証明書が使用されるようにします。

このサブコマンドを使用すると、`asadmin` コマンドや IDE など、リモート管理クライアントからの管理メッセージもドメイン管理サーバで受け入れることができます。

デフォルトでは、`secure admin` 機能が有効であるとき、ドメイン管理サーバとインスタンスは、これらの SSL 証明書を使用して、双方をセキュリティーのプリンシパルとして認証し、管理アクセスを許可します。

`--adminalias` オプションの値は、ドメイン管理サーバがインスタンスに対して自身を識別するために使用する SSL 証明書をドメイン管理サーバに指示します。

`--instancealias` オプションの値は、インスタンスがドメイン管理サーバに対して自身を識別するために使用する SSL 証明書を決定します。

`enable-secure-admin` サブコマンドは、ドメイン内のどれかの管理ユーザーのパスワードが空白の場合は失敗します。

## 前提条件

- ドメイン管理サーバが稼働し、インスタンスは稼働していない必要があります。
- ドメイン管理サーバの HTTPS 通信ポートは、開いている必要があります。
- インスタンスの HTTPS 通信ポートは、開く設定である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--adminalias=alias`

ドメイン管理サーバで SSL/TLS 証明書を参照するエイリアスを指定します。このエイリアスは、インスタンスがドメイン管理サーバを識別できるようにドメイン管理サーバ自体で使用します。

型：String

指定できる値を次に示します。

- *SSL/TLS 証明書のエイリアス名*

デフォルト値：s1as

--instancealias=*alias*

インスタンス上で SSL/TLS 証明書を参照するエイリアスを 指定します。このエイリアスは、ドメイン管理サーバ が各インスタンスを識別できるようにインスタンス自体で使用します。

型：String

指定できる値を次に示します。

- *SSL/TLS 証明書のエイリアス名*

デフォルト値：glassfish-instance

## 使用例

次の例では、adtest という名称のadmin のエイリアスおよびintest という名称のインスタンスのエイリアスを使用してドメインの secure admin 機能を有効にします。

```
asadmin enable-secure-admin --adminalias adtest --instancealias intest
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.12 list-auth-realms

認証レルムの一覧表示

### 書式

```
asadmin [asadmin-options] list-auth-realms [--help]  
[target]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-auth-realms サブコマンドは、認証レルムを一覧表示します。



このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する認証レルムの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、認証レルムを一覧表示します。

```
asadmin list-auth-realms
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.13 list-file-groups

ファイルグループの一覧表示

## 書式

```
asadmin [asadmin-options] list-file-groups [--help]
  [--name username] [--authrealmname auth_realm_name]
  target
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-file-groups` サブコマンドは、ファイルレルム認証によってサポートされるファイルユーザーとグループを一覧表示します。このサブコマンドはファイルユーザーに利用されているグループを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

`--name` オプションを指定しない場合、すべてのグループが一覧表示されます。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--name` *username*

グループを一覧表示するファイルユーザーの名称を指定します。

型：String

指定できる値を次に示します。

- ファイルユーザー名

デフォルト値：なし

`--authrealmname` *auth\_realm\_name*

グループを一覧表示する認証レルムの名称を指定します。

型：String

指定できる値を次に示します。

- 認証レルム名

デフォルト値：なし

*target*

一覧表示するファイルグループの対象を指定します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、サーバに定義されたファイルグループをすべて一覧表示します。

```
asadmin list-file-groups
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.14 list-file-users

ファイルユーザーの一覧表示

### 書式

```
asadmin [asadmin-options] list-file-users [--help]  
        [--authrealmname auth_realm_name] [target]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-file-users サブコマンドは、ファイルレルム認証でサポートされるファイルユーザーの一覧を表示します。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--authrealmname auth_realm_name`

指定した認証レルムのユーザーだけを一覧表示します。

型：String

指定できる値を次に示します。

- *認証レルム名*

デフォルト値：なし

*target*

一覧表示するファイルユーザーの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：*server*

## 使用例

次の例では、デフォルトの file レルムファイルのファイルユーザーを一覧表示します。

```
asadmin list-file-users
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.15 list-jacc-providers

JACC プロバイダーの一覧表示

### 書式

```
asadmin [asadmin-options] list-jacc-providers [--help]
        [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-jacc-providers サブコマンドで、管理者は、ドメインに定義された JACC プロバイダーを一覧表示できます。

JACC プロバイダーは、ドメインの domain.xml ファイルにある security-service 要素に jacc-provider 要素として定義されます。JACC プロバイダーは create-jacc-provider サブコマンドで作成できます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### ファイル

JACC プロバイダーは、ドメインの domain.xml ファイルにある security-service 要素に jacc-provider 要素として定義されます。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する JACC プロバイダーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのドメインの JACC プロバイダーを一覧表示します。

```
asadmin list-jacc-providers
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.16 list-password-aliases

パスワードエイリアスの一覧表示

### 書式

```
asadmin [asadmin-options] list-password-aliases [--help]
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-password-aliases` サブコマンドは、すべてのパスワードエイリアスを一覧表示します。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

## 使用例

次の例では、すべてのパスワードエイリアスを一覧表示します。

```
asadmin list-password-aliases
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.17 list-supported-cipher-suites

Java EE Server で利用できる暗号化方式群の一覧表示

## 書式

```
asadmin [asadmin-options] list-supported-cipher-suites [--help]  
[--target target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-supported-cipher-suites` サブコマンドを使用すると、管理者は、サポートされている暗号化方式群のうち、指定した Java EE Server で利用できる暗号化方式群を一覧表示できます。

Java EE Server パッケージに含まれているデフォルトの SSL/TLS プロバイダーのほかに利用できる暗号化方式群は、プロバイダー（他社製）によって異なります。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

一覧表示する暗号化方式群の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、デフォルトのドメインの暗号化方式群を一覧表示します。

```
asadmin list-supported-cipher-suites
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.18 update-file-user

現在のファイルユーザーの更新

## 書式

```
asadmin [asadmin-options] update-file-user [--help]
  [--groups user_groups[:user_groups]...] [--target target]
  [--authrealmname authrealm_name] username
```



## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`update-file-user` サブコマンドは、指定したユーザー名、パスワード、グループを使用して、キーファイル内の既存のエントリーを更新します。複数のグループをコロン (:) で区切って入力できます。

新たなパスワードを指定しない場合、セキュア管理が有効で、更新するユーザーが管理ユーザーなら、このサブコマンドは失敗します。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

*domain-dir/domain\_name/config* ディレクトリーの *keyfile* ファイルが更新されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--groups user_groups[:user_groups]...`

ファイルユーザーが所属するグループの名称を指定します。

型: String

指定できる値を次に示します。

- グループ名

デフォルト値: なし

`--target target`

更新するファイルユーザーの対象を指定します。

型: String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値: `server`

`--authrealmname authrealm_name`

更新するユーザーが存在するユーザー認証レルムを指定します。

型：String

指定できる値を次に示します。

- 認証レルム名

デフォルト値：file

*username*

更新するファイルユーザーの名称を指定します。

型：String

指定できる値を次に示します。

- ファイルユーザー名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンス上の、sample\_user という名称のファイルレルムユーザーの情報を更新します。

```
asadmin update-file-user --target instance1
--groups staff:manager:engineer sample_user
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.9.19 update-password-alias

パスワードエイリアスの更新

### 書式

```
asadmin [asadmin-options] update-password-alias [--help]
aliasname
```

## 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

## 機能

update-password-alias サブコマンドは、指定されたターゲットでパスワードエイリアス ID を更新します。

エイリアスは、`${ALIAS=password-alias-password}` という形式で指定します。エイリアス名に対応するパスワードは暗号化されて格納されます。

update-password-alias サブコマンドでは、次に示す両方の形式を使用できます。

- ユーザーがすべての情報の入力を求められる安全な対話形式。
- パスワードがコマンドラインで伝達されるが、スクリプトの処理しやすい形式。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

domain.xml に記録されているパスワードの名称。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*aliasname*

domain.xml ファイルに記録されているパスワードの名称を指定します。

型：String

指定できる値を次に示します。

- パスワードエイリアス名

デフォルト値：なし

## 使用例

次の例では、jmspassword-alias エイリアスのパスワードを更新します。

```
asadmin update-password-alias jmspassword-alias
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。

戻り値	説明
1	サブコマンドの実行中にエラーが発生しました。

## 注意事項

- 変更したパスワードエイリアスを適用するには、ドメイン管理サーバを再起動する必要があります。

## 2.10 ログ管理で使用するコマンド

ログ管理で使用するコマンドの書式、機能などについて説明します。

### 2.10.1 collect-local-snapshot

ドメイン管理サーバ非稼働時のシステム情報の収集

#### 書式

```
asadmin [asadmin-options] collect-local-snapshot  
  [--infotype info-type1 [, info-type2]...]  
  [--domaindir domain-root-dir]  
  [--domain domain-name] [--nodedir node-root-dir]  
  --targettype={node|server}  
  [--targetname target-name1 [, target-name2]...]  
  [--archivefile archive-file] [--timeout timeout]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`collect-local-snapshot` サブコマンドは、*Application Server* を構成する製品のシステム情報を収集します。集めた情報は、システム情報アーカイブファイルとして出力します。

このサブコマンドの実行中に強制終了した場合、システム情報アーカイブファイルの出力先ディレクトリーに、次に示すファイルやディレクトリーが残ることがあります。

- 拡張子が"zip"の、サブコマンドが一時的に生成したファイル
- 拡張子が"tmp"の、サブコマンドが一時的に生成したファイル
- サブコマンドが一時的に生成したディレクトリー（ディレクトリー名は実行ごとに異なります）

出力先ディレクトリーに生成されるシステム情報アーカイブファイルの数は上限値を設定できるため、拡張子が"zip"のファイルが残り続けることはありません。これ以外のファイルおよびディレクトリーは適宜削除してください。削除するときは、`collect-snapshot` サブコマンドおよび`collect-local-snapshot` サブコマンドが動作中でないことをメッセージログで確認してください。

#### ❗ 重要

- `collect-local-snapshot` サブコマンドを実行するごとにスレッドダンプファイルが生成されます。スレッドダンプファイルを残しておく必要がない場合は、サブコマンドの実行完了後に削除してください。スレッドダンプファイルは、デフォルトではカレントディレクト

リーに出力されます。ドメイン管理サーバのカレントディレクトリーは「*Application Server* インストールディレクトリー/*javaee/glassfish/domains/ドメイン名/config*」、Java EE サーバのカレントディレクトリーは「*Application Server* インストールディレクトリー/*javaee/glassfish/nodes/ノード名/サーバインスタンス名/config*」になります。

- スレッドダンプファイルの出力先をカレントディレクトリーから変更した場合は、システム情報アーカイブファイルにスレッドダンプファイルが含まれなくなります。
- タイムアウトによってシステム情報の収集が中断された場合は、`collect-local-snapshot` サブコマンドが内部で呼び出したプロセスの実行も中断されます。ただし、`collect-local-snapshot` サブコマンドが呼び出したプロセスがさらに別のプロセスを呼び出している場合、このプロセスを中断できない場合があります。

## 実行権限

一般ユーザー

## 前提条件

- このサブコマンドを、同一のサーバに対して複数同時に実行しないこと

## 引数

`--infotype info-type1[, info-type2]...`

収集するシステム情報の種類を指定します。複数の種類をコンマで区切って記述できます。

型：String

指定できる値を次に示します。

- `initinfo`  
問題点の切り分けに最低限必要な、初期調査用のシステム情報を収集します。
- `sysinfo`  
原因の調査に必要な、すべてのシステム情報を収集します。初期調査用のシステム情報も含まれます。

デフォルト値：`initinfo, sysinfo`

`--domaindir domain-root-dir`

ドメインルートディレクトリーの絶対パスを指定します。ドメインルートディレクトリーを *Application Server* のデフォルト値から変更していない場合や、ドメイン管理サーバが存在しないホストで実行する場合は省略できます。

型：String

指定できる値を次に示します。

- ドメインルートディレクトリーの絶対パス

デフォルト値：*Application Server* インストールディレクトリー/*javaee/glassfish/domains*

### `--domain domain-name`

同じ名称のノードやサーバが複数のドメインに存在する場合は、システム情報を収集したいノードやサーバが所属するドメインの名称を指定します。それ以外の場合は省略できます。

このオプションに指定した名称のドメインが複数あり、各ドメインに存在する同じ名称のサーバを `--targetname` オプションに指定した場合は、ドメイン名とサーバ名が一致するすべてのサーバのシステム情報を収集します。

型：String

指定できる値を次に示します。

- ドメイン名

デフォルト値：なし

### `--nodedir node-root-dir`

サーバ (Java EE サーバ、Web サーバ、PRF) が構築されているノードルートディレクトリーの絶対パスを指定します。サーバが構築されているノードルートディレクトリーを Application Server のデフォルト値から変更していない場合や、ノードが存在しないホストで実行する場合は省略できます。

このオプションは、ノードルートディレクトリーを1つだけ指定できます。異なるノードルートディレクトリー下にあるノードやサーバのシステム情報を取得する場合は、それぞれのノードルートディレクトリーで `collect-local-snapshot` サブコマンドを実行してください。

型：String

指定できる値を次に示します。

- ノードルートディレクトリーの絶対パス

デフォルト値：*Application Server* インストールディレクトリー/*javaee/glassfish/nodes*

### `--targettype={node|server}`

システム情報を収集する対象を指定します。

型：String

指定できる値を次に示します。

- *node*

`--targetname` で指定したノードを構成する全サーバのシステム情報を収集します。

- *server*

`--targetname` で指定したサーバ (Java EE サーバ、Web サーバ、PRF) のシステム情報を収集します。

デフォルト値：なし

### `--targetname target-name1 [, target-name2] ...`

システム情報を収集するノードまたはサーバの名称を指定します。複数の名称をコンマで区切って記述できます。

型：String

指定できる値を次に示します。

- ノード名またはサーバ名

デフォルト値：server

#### `--archivefile archive-file`

システム情報アーカイブファイルの末尾に付く文字列を指定します。文字列の形式を次に示します。

`snapshot-info-type-archive-file`

`info-type`：--infotype オプションの指定値。

`archive-file`：--archivefile オプションの指定値。

--infotype オプションに複数の値を指定した場合は、それぞれの指定値に対応するシステム情報アーカイブファイルを作成します。

システム情報アーカイブファイルの出力先ディレクトリーに同じ名前のファイルが存在する場合は、既存のファイルが上書きされます。

型：String

指定できる値を次に示します。

- システム情報アーカイブファイルの末尾に付く文字列

次の半角文字が使用できます。

英小文字

英大文字

数字

ハイフン

ピリオド

アンダースコア

デフォルト値：`domain-name-date-process-id-thread-id.zip`

`date` の形式は”`yyyyMMddHHmmss`”。

`process-id` は`collect-local-snapshot` サブコマンドのプロセス ID の 16 進数表記。

`thread-id` は`collect-local-snapshot` サブコマンドのスレッド ID の 16 進数表記。

#### `--timeout timeout`

システム情報の収集を中止してコマンド実行を終了するまでの時間を指定します。

システム情報の収集が中止された場合、システム情報アーカイブファイルの出力先ディレクトリーに生成途中のファイルが残ることがあるため、適宜削除してください。

型：Integer

指定できる値を次に示します。

- 1~2147483647

デフォルト値：なし (`collect-local-snapshot` サブコマンドはタイムアウトしない)

## 使用例

次の例では、サーバ名`server` のシステム情報を収集します。



```
asadmin collect-local-snapshot --targettype=server
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 出力形式

このサブコマンドは、`--infotype` オプションで指定したシステム情報を収集して、それぞれに対応するシステム情報アーカイブファイルを生成します。

システム情報アーカイブファイルの名称は`--archivefile` オプションの説明を参照してください。

システム情報アーカイブファイルは、サブコマンドを実行したホストの、次に示すディレクトリーに出力されます。

*Application Server*インストールディレクトリー/`javaee/snapshot`

出力先ディレクトリーの場所や出力先ディレクトリーに生成できるシステム情報アーカイブファイル数は、`asadmin` ユーティリティーの`set` サブコマンドおよび`get` サブコマンドの次のプロパティーで設定および確認できます。

- `hitachi-domain.hitachi-snapshot.output-dir`※
- `hitachi-domain.hitachi-snapshot.max-num`
- `nodes.node.ノード名.hitachi-node.hitachi-node-snapshot.output-dir`※
- `nodes.node.ノード名.hitachi-node.hitachi-node-snapshot.max-num`

### 注※

アクセス権のないディレクトリーを出力先に指定した場合は、エラーメッセージ (KDKD10211-E) を出力してシステム情報収集処理が中断されます。

プロパティーの詳細は、「[set サブコマンドおよび get サブコマンドで使用するパラメーター一覧](#)」を参照してください。

## 2.10.2 collect-snapshot

ドメイン管理サーバ稼働時のシステム情報の収集

## 書式

```
asadmin [asadmin-options] collect-snapshot  
  [--infotype info-type1 [, info-type2]...]  
  --targettype={domain|node|server}  
  [--targetname target-name1 [, target-name2]...] [--timeout timeout]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

collect-snapshot サブコマンドは、Application Server を構成する製品のシステム情報を収集します。ドメインを構成するノードやサーバが存在する複数のホストで収集します。収集した情報は各ホスト内にシステム情報アーカイブファイルとして出力されます。

このサブコマンドの実行中に強制終了した場合、システム情報アーカイブファイルの出力先ディレクトリーに、次に示すファイルやディレクトリーが残ることがあります。

- 拡張子が"zip"の、サブコマンドが一時的に生成したファイル
- 拡張子が"tmp"の、サブコマンドが一時的に生成したファイル
- サブコマンドが一時的に生成したディレクトリー（ディレクトリー名は実行ごとに異なります）

出力先ディレクトリーに生成されるシステム情報アーカイブファイルの数は上限値を設定できるため、拡張子が"zip"のファイルが残り続けることはありません。これ以外のファイルおよびディレクトリーは適宜削除してください。削除するときは、collect-snapshot サブコマンドおよびcollect-local-snapshot サブコマンドが動作中でないことをメッセージログで確認してください。

### ❗ 重要

- collect-snapshot サブコマンドを実行するごとにスレッドダンプファイルが生成されます。スレッドダンプファイルを残しておく必要がない場合は、サブコマンドの実行完了後に削除してください。スレッドダンプファイルは、デフォルトではカレントディレクトリーに出力されます。ドメイン管理サーバのカレントディレクトリーは「*Application Server*インストールディレクトリー/*javaee/glassfish/domains/ドメイン名/config*」、Java EE サーバのカレントディレクトリーは「*Application Server*インストールディレクトリー/*javaee/glassfish/nodes/ノード名/サーバインスタンス名/config*」になります。
- スレッドダンプファイルの出力先をカレントディレクトリーから変更した場合は、システム情報アーカイブファイルにスレッドダンプファイルが含まれなくなります。
- タイムアウトによってシステム情報の収集が中断された場合は、collect-snapshot サブコマンドが内部で呼び出したプロセスの実行も中断されます。ただし、collect-snapshot サブコマンドが呼び出したプロセスがさらに別のプロセスを呼び出している場合、このプロセスを中断できない場合があります。

- ドメイン管理サーバのあるホストから通信できないホストがある場合は、そのホストにあるノードおよびサーバを収集対象から除外して、`collect-snapshot` サブコマンドを実行してください。

## 実行権限

一般ユーザー

## 前提条件

- ドメイン管理サーバが稼働していること
- システム情報アーカイブファイルを収集するホストと正常に通信できること
- このサブコマンドを、同一のサーバに対して複数同時に実行しないこと

## 引数

`--infotype info-type1[,info-type2]...`

収集するシステム情報の種類を指定します。複数の種類をコンマで区切って記述できます。

型：String

指定できる値を次に示します。

- `initinfo`  
問題点の切り分けに最低限必要な、初期調査用のシステム情報を収集します。
- `sysinfo`  
原因の調査に必要な、すべてのシステム情報を収集します。初期調査用のシステム情報も含まれます。

デフォルト値：`initinfo,sysinfo`

`--targettype={domain|node|server}`

システム情報を収集する対象を指定します。

型：String

指定できる値を次に示します。

- `domain`  
ドメインを構成する全サーバのシステム情報を収集します。`--targetname` の指定値は使用されません。
- `node`  
`--targetname` で指定したノードを構成する全サーバのシステム情報を収集します。
- `server`  
`--targetname` で指定したサーバ (Java EE サーバ、Web サーバ、PRF) のシステム情報を収集します。

デフォルト値：なし

`--targetname target-name1[,target-name2]...`

システム情報を収集するノードまたはサーバの名称を指定します。複数の名称をコンマで区切って記述できます。

`--targettype` に `domain` を指定した場合は、このオプションの指定値は使用されません。

型：String

指定できる値を次に示します。

- ノード名またはサーバ名

デフォルト値：server

`--timeout timeout`

このサブコマンドが呼び出した `collect-local-snapshot` サブコマンドが、システム情報の収集を中止してコマンド実行を終了するまでの時間を指定します。

システム情報の収集が中止された場合、システム情報アーカイブファイルの出力先ディレクトリーに生成途中のファイルが残ることがあるため、適宜削除してください。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：なし (`collect-local-snapshot` サブコマンドはタイムアウトしない)

## 使用例

次の例では、ドメインを構成する全サーバのシステム情報を収集します。

```
asadmin collect-snapshot --targettype=domain
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 出力形式

このサブコマンドは、`--infotype` オプションで指定したシステム情報を収集して、それぞれに対応するシステム情報アーカイブファイルを生成します。システム情報アーカイブファイルは次の名称で生成されます。

```
snapshot-info-type※1-収集対象名※2-日時※3-プロセスID※4-スレッドID※5.zip
```

※1

--infotype オプションの指定値です。複数の値を指定した場合は、各指定値に対応する複数のシステム情報アーカイブファイルが生成されます。

※2

ドメイン管理サーバがあるホストに出力するファイルではドメイン名、ドメイン管理サーバがないノードに出力するファイルではドメイン名-ホスト名となります。

※3

形式は *yyyyMMddHHmmss* です。--targetname オプションで複数のホスト上にあるノードやサーバを指定した場合、各ホストで出力するファイル名の日時は、すべてcollect-snapshot サブコマンドを実行した時点の日時になります。

※4

ドメイン管理サーバのプロセス ID の 16 進数表記です。

※5

ドメイン管理サーバ上で動作しているcollect-snapshot サブコマンドのスレッド ID の 16 進数表記です。

システム情報アーカイブファイルは、--targetname オプションで指定したサーバが存在するホストの、次に示すディレクトリーに出力されます。

*Application Server*インストールディレクトリー/javaee/snapshot

出力先ディレクトリーの場所や出力先ディレクトリーに生成できるシステム情報アーカイブファイル数は、asadmin ユーティリティーのset サブコマンドおよびget サブコマンドの次のプロパティーで設定および確認できます。

- hitachi-domain.hitachi-snapshot.output-dir<sup>※</sup>
- hitachi-domain.hitachi-snapshot.max-num
- nodes.node.ノード名.hitachi-node.hitachi-node-snapshot.output-dir<sup>※</sup>
- nodes.node.ノード名.hitachi-node.hitachi-node-snapshot.max-num

注※

アクセス権のないディレクトリーを出力先に指定した場合は、エラーメッセージ (KDKD10211-E) を出力してシステム情報収集処理が中断されます。

プロパティーの詳細は、「set サブコマンドおよび get サブコマンドで使用するパラメーター一覧」を参照してください。

## 2.10.3 list-log-attributes

ログ属性の一覧表示

## 書式

```
asadmin [asadmin-options] list-log-attributes [--help]
        [target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-log-attributes` サブコマンドは、指定された Java EE Server ドメインまたはドメイン内の対象に現在定義されているログ属性をすべて一覧表示します。一覧表示された値はドメインの `logging.properties` ファイルの値に対応します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

ログ属性は指定した Java EE Server ドメインまたはドメイン内の対象に対して定義されています。一覧表示された値はドメインの `logging.properties` ファイルの値に対応します。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するログ属性の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、デフォルトのドメインのすべてのログ属性を一覧表示します。

```
asadmin list-log-attributes
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.10.4 set-log-attributes

1つ以上のロガーのロギング属性の設定

### 書式

```
asadmin [asadmin-options] set-log-attributes [--help]
  [--target=target]
  attribute-name=attribute-value[:attribute-name=attribute-value]...
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

set-log-attributes サブコマンドは、1つ以上のロガーのロギング属性を設定します。設定する属性は、ドメインのlogging.properties ファイルで使用できる属性に対応しています。設定した属性によっては、サーバの再起動が必要になることがあります。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### ファイル

設定できる属性は、ドメインのlogging.properties ファイルで使用できる属性に対応しています。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target=target`

設定するロギング属性の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

*attribute-name*

ロギング属性の完全スコープ名を指定します。

指定できる属性名については「[set-log-attributes コマンドで使用するパラメーター一覧](#)」を参照してください。

*attribute-value*

指定された属性に適用する値を指定します。

指定できる属性値については「[set-log-attributes コマンドで使用するパラメーター一覧](#)」を参照してください。

## 使用例

次の例では、サーバインスタンス `instanceA` のメッセージログ切り替えサイズを `16777216` に設定します。

```
asadmin set-log-attributes --target instanceA
ServerInstance.message_log.rotation-size=16777216
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.10.5 set-log-attributes コマンドで使用するパラメーター一覧

ここでは、set-log-attributes サブコマンドの *attribute-name* オプションに指定できるパラメーターについて説明します。

### set-log-attributes コマンドで使用するパラメーター

項番	パラメーター名	説明	指定できる値※1	デフォルト値	-target オプションに指定する値	指定例
1	ServerInstance.message_log.enabled	サーバインスタンスのメッセージログ出力の有効/無効を切り替えます。	true false	true	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.message_log.enabled=true
2	ServerInstance.message_log.rotation-size	サーバインスタンスのメッセージログの切り替えサイズを指定します。	4096~2147483647	16777216	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.message_log.rotation-size=4096
3	ServerInstance.message_log.rotation-time	サーバインスタンスのメッセージログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.message_log.rotation-time=000000
4	ServerInstance.message_log.file-number	サーバインスタンスのメッセージログの保持面数を指定します。	1~16	8	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.message_log.file-number=8
5	ServerInstance.message_log.log-level	サーバインスタンスのメッセージログのログレベルを指定します。※2	1~3	2	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.message_log.log-level=2
6	ServerInstance.stacktrace_log.enabled	サーバインスタンスのスタックトレースログ出力の有効/無効を切り替えます。	true false	true	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.stacktrace_log.enabled=true

項番	パラメーター名	説明	指定できる値※1	デフォルト値	-target オプションに指定する値	指定例
7	ServerInstance. stacktrace_log.rotation-size	サーバインスタンスのスタックトレースログの切り替えサイズを指定します。	4096~2147483647	16777216	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance. stacktrace_log.rotation-size=4096
8	ServerInstance. stacktrace_log.rotation-time	サーバインスタンスのスタックトレースログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance. stacktrace_log.rotation-time=000000
9	ServerInstance. stacktrace_log.file-number	サーバインスタンスのスタックトレースログの保持面数を指定します。	1~16	8	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance. stacktrace_log.file-number=8
10	ServerInstance. websocket_access_log.enabled	サーバインスタンスの WebSocket アクセスログ出力の有効/無効を切り替えます。	true false	true	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance. websocket_access_log.enabled=false
11	ServerInstance. websocket_access_log.format	サーバインスタンスの WebSocket アクセスログの出力フォーマットを指定します。 WebSocket アクセスログにペイロードデータを出力する場合、%PAYLOADDATA% と %PAYLOADDATA(文字数)% は同時に指定できません。	<ul style="list-style-type: none"> <li>• %TS%</li> <li>• %CLIENTADDR%</li> <li>• %SERVERADDR%</li> <li>• %IO%</li> <li>• %URI%</li> <li>• %SESSIONID%</li> <li>• %OPCODE%</li> <li>• %FIN%</li> <li>• %PAYLOADDATALEN%</li> <li>• %MASK%</li> <li>• %ROOTAP%</li> <li>• %MASKKEY%</li> <li>• %ISEXTENDED%</li> <li>• %RSV%</li> <li>• %FRAMEMAINTYPE%</li> <li>• %CLOSEREASON%</li> </ul>	%TS% %IO% %OPCODE% %URI% %FIN% %CLOSEREASON% %PAYLOADDATALEN% %ROOTAP% %CLIENTAP%	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance. websocket_access_log.format=%TS% %IO% %OPCODE% %URI% %FIN% %CLOSEREASON% %PAYLOADDATALEN% %ROOTAP%

項番	パラメーター名	説明	指定できる値※1	デフォルト値	-target オプションに指定する値	指定例
			<ul style="list-style-type: none"> <li>• %PAYLOADDATA%</li> <li>• %PAYLOADDATA(文字数)%</li> <li>• %CLIENTAP%</li> </ul>			
12	ServerInstance.websocket_access_log.rotation-size	サーバインスタンスの WebSocket アクセスログの切り替えサイズを指定します。	4096~2147483647	314572800	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.websocket_access_log.rotation-size=16777216
13	ServerInstance.websocket_access_log.rotation-time	サーバインスタンスの WebSocket アクセスログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.websocket_access_log.rotation-time=000000
14	ServerInstance.websocket_access_log.file-number	サーバインスタンスの WebSocket アクセスログの保持数を指定します。	1~16	8	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.websocket_access_log.file-number=2
15	ServerInstance.maintenance_log.enabled	サーバインスタンスの保守ログ出力の有効/無効を切り替えます。	true false	true	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.maintenance_log.enabled=true
16	ServerInstance.maintenance_log.rotation-size	サーバインスタンスの保守ログの切り替えサイズを指定します。	4096~2147483647	16777216	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.maintenance_log.rotation-size=4096
17	ServerInstance.maintenance_log.rotation-time	サーバインスタンスの保守ログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.maintenance_log.rotation-time=000000

## 2. Java EE サーバで使用するコマンド

項番	パラメーター名	説明	指定できる値*1	デフォルト値	-target オプションに指定する値	指定例
18	ServerInstance.maintenance_log.file-number	サーバインスタンスの保守ログの保持面数を指定します。	1~16	8	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.maintenance_log.file-number=8
19	ServerInstance.maintenance_log.maintenance-log-level	サーバインスタンスの保守ログのログレベルを指定します。	1~2	1	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.maintenance_log.maintenance-log-level=1
20	ServerInstance.glassfish_debug_log.enabled	サーバインスタンスのデバッグログ出力の有効/無効を切り替えます。	true false	false	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.glassfish_debug_log.enabled=false
21	ServerInstance.glassfish_debug_log.rotation-size	サーバインスタンスのデバッグログの切り替えサイズを指定します。	4096~2147483647	2147483647	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.glassfish_debug_log.rotation-size=4096
22	ServerInstance.glassfish_debug_log.rotation-time	サーバインスタンスのデバッグログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.glassfish_debug_log.rotation-time=000000
23	ServerInstance.glassfish_debug_log.file-number	サーバインスタンスのデバッグログの保持面数を指定します。	1~16	8	サーバインスタンス名またはクラスター名	asadmin set-log-attributes --target=instanceA ServerInstance.glassfish_debug_log.file-number=8
24	DAS.message_log.enabled	ドメイン管理サーバのメッセージログ出力の有効/無効を切り替えます。	true false	true	server	asadmin set-log-attributes --target=server

項番	パラメーター名	説明	指定できる値※1	デフォルト値	-target オプションに指定 する値	指定例
						DAS.message_log.enabled=true
25	DAS.message_log.rotation-size	ドメイン管理サーバのメッセージログの切り替えサイズを指定します。	4096~2147483647	16777216	server	asadmin set-log-attributes --target=server DAS.message_log.rotation-size=4096
26	DAS.message_log.rotation-time	ドメイン管理サーバのメッセージログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	server	asadmin set-log-attributes --target=server DAS.message_log.rotation-time=000000
27	DAS.message_log.file-number	ドメイン管理サーバのメッセージログの保持面数を指定します。	1~16	8	server	asadmin set-log-attributes --target=server DAS.message_log.file-number=8
28	DAS.message_log.log-level	ドメイン管理サーバのメッセージログのログレベルを指定します。※2	1~3	2	server	asadmin set-log-attributes --target=server DAS.message_log.log-level=2
29	DAS.stacktrace_log.enabled	ドメイン管理サーバのスタックトレースログ出力の有効/無効を切り替えます。	true false	true	server	asadmin set-log-attributes --target=server DAS.stacktrace_log.enabled=true
30	DAS.stacktrace_log.rotation-size	ドメイン管理サーバのスタックトレースログの切り替えサイズを指定します。	4096~2147483647	16777216	server	asadmin set-log-attributes --target=server DAS.stacktrace_log.rotation-size=4096
31	DAS.stacktrace_log.rotation-time	ドメイン管理サーバのスタックトレースログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	server	asadmin set-log-attributes --target=server DAS.stacktrace_log.rotation-time=000000
32	DAS.stacktrace_log.file-number	ドメイン管理サーバのスタックトレース	1~16	8	server	asadmin set-log-attributes --

項番	パラメーター名	説明	指定できる値※1	デフォルト値	-target オプションに指定 する値	指定例
		ログの保持面数を指定します。				target=server DAS.stacktrace_log. file-number=8
33	DAS.maintenance_log.enabled	ドメイン管理サーバの保守ログ出力の有効/無効を切り替えます。	true false	true	server	asadmin set-log-attributes --target=server DAS.maintenance_log.enabled=true
34	DAS.maintenance_log.rotation-size	ドメイン管理サーバの保守ログの切り替えサイズを指定します。	4096~2147483647	16777216	server	asadmin set-log-attributes --target=server DAS.maintenance_log.rotation-size=4096
35	DAS.maintenance_log.rotation-time	ドメイン管理サーバの保守ログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	server	asadmin set-log-attributes --target=server DAS.maintenance_log.rotation-time=000000
36	DAS.maintenance_log.file-number	ドメイン管理サーバの保守ログの保持面数を指定します。	1~16	8	server	asadmin set-log-attributes --target=server DAS.maintenance_log.file-number=8
37	DAS.maintenance_log.maintenance-log-level	ドメイン管理サーバの保守ログのログレベルを指定します。	1~2	1	server	asadmin set-log-attributes --target=server DAS.maintenance_log.maintenance-log-level=1
38	DAS.glassfish_debug_log.enabled	ドメイン管理サーバのデバッグログ出力の有効/無効を切り替えます。	true false	false	server	asadmin set-log-attributes --target=server DAS.glassfish_debug_log.enabled=false
39	DAS.glassfish_debug_log.rotation-size	ドメイン管理サーバのデバッグログの切り替えサイズを指定します。	4096~2147483647	2147483647	server	asadmin set-log-attributes --target=server DAS.glassfish_debug_log.rotation-size=2147483647

## 2. Java EE サーバで使用するコマンド

項番	パラメーター名	説明	指定できる値 <sup>※1</sup>	デフォルト値	-target オプションに指定する値	指定例
						ug_log.rotation-size=4096
40	DAS.glassfish_debug_log.rotation-time	ドメイン管理サーバのデバッグログの切り替え時間を指定します。	HHMMSS の形式でラップ時刻を指定します。	000000	server	asadmin set-log-attributes --target=server DAS.glassfish_debug_log.rotation-time=000000
41	DAS.glassfish_debug_log.file-number	ドメイン管理サーバのデバッグログの保持面数を指定します。	1~16	8	server	asadmin set-log-attributes --target=server DAS.glassfish_debug_log.file-number=8

注※1

範囲外の値を指定した場合は、コマンドの実行に失敗します。

注※2

標準出力に出力するメッセージをメッセージログに出力するためには、ログレベルを3に設定してください。標準エラー出力に出力するメッセージはデフォルトの設定 (2) でメッセージログに出力します。

## 2.11 パラメーター設定で使用するコマンド

パラメーター設定で使用するコマンドの書式、機能などについて説明します。

### 2.11.1 create-jvm-options

Java VM オプションの作成

#### 書式

```
asadmin [asadmin-options] create-jvm-options [--help] [--target target]  
  [--profiler={true|false}]  
  option-name[=option-value][:option-name[=option-value]]...
```

#### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

#### 機能

create-jvm-options サブコマンドは、Java EE Server が起動されるときに、Java VM に渡されるコマンドラインの各オプションを作成します。このサブコマンドで作成されるオプションは、Java EE Server で事前設定されているオプションに追加されます。

Java VM のオプションは、Java 構成 (domain.xml ファイルのprofiler 要素/java-config 要素) に格納されます。domain.xml ファイルのprofiler 要素、java-config 要素に表示される順序でオプションはコマンドラインに送られます。

このサブコマンドを使用して次のタイプのオプションを作成できます。

- Java システムプロパティ：これらのオプションの前に-D オプションで設定します。
- Java VM のスタートアップパラメーター：これらのオプションの前にはハイフン (-) をつけます。

このサブコマンドはリモートモードだけサポートします。

ドメイン管理サーバの Java VM オプションに不適切な値を指定した場合、ドメイン管理サーバが起動しないで、ドメインの再構築が必要になることがあります。このような事態を避けるために、ドメイン管理サーバの Java VM オプションを変更するときには、あらかじめ、backup-domain コマンドを使用してドメインのバックアップを採取しておくことをお勧めします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。



## ファイル

Java VM の各オプションは、Java 構成 (domain.xml ファイルの `profiler` 要素/`java-config` 要素) に格納されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

作成する Java VM オプションの対象を指定します。

型: String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値: `server`

`--profiler={true|false}`

Java VM オプションがプロファイラー用かどうかを指定します。

型: Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値: `false`

`option-name [=option-value] [:option-name [=option-value]]...`

オプションを指定します。複数の場合は、コロン (:) で区切ります。

オプションの形式は、次のとおりです。

オプションに名前と値がある場合は、形式は `option-name=value` です。

オプションが名前だけの場合は、形式は `option-name` です。

型: String

指定できる値を次に示します。

- jvm-option 名、または値を持つ名称

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、Java VM オプションを作成します。

```
asadmin create-jvm-options --target instance1 -Dunixlocation=/root/example:
-Dvariable=%$HOME:-Dwindowslocation=d%:%$sun%$appserver:-Doption1=-value1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 注意事項

このコマンドのオプションに次のシステムプロパティを指定しないでください。Message Queue のログの設定 (config.properties) に影響を与えます。

- java.util.logging.FileHandler.count
- java.util.logging.FileHandler.limit
- java.util.logging.FileHandler.level
- java.util.logging.FileHandler.pattern

## 2.11.2 create-system-properties

構成のほかの部分から参照できるシステムプロパティの追加

### 書式

```
asadmin [asadmin-options] create-system-properties [--help]
  [--target target] [name=value[:name=value]...]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

`create-system-properties` サブコマンドは、サーバの各所から参照できる各システムプロパティを追加または更新します。

Java EE Server には、各トークン（システムプロパティ）を指定できるフックが用意されています。Java EE Server には複数サーバ要素がないので、任意のレベルで特定のトークンを指定できます。ドメインが複数サーバをサポートするとき、優先原則が悪用されるおそれがあります。

ドメインが起動または再起動される時、すべての `system-property` 要素は `System.setProperty()` 呼び出しによって解決され、Java VM で利用できるようになります（要素の名前と値は、対応する要素属性から抽出されます）。この機能は、Java のコマンドラインを使用して、要素を `-D` パラメーターとして送信する機能に類似しています。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

作成するシステムプロパティの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`name=value[:name=value]...`

指定した対象に追加するシステムプロパティのプロパティ名と値のペアを指定します。

複数システムプロパティ記述する場合コロン (:) で区切る必要があります。コロン (:) がシステムプロパティの名前や値に含まれる場合、円記号 (¥) でエスケープする必要があります。

型：String

指定できる値を次に示します。

- システムプロパティのプロパティ名と値のペア

デフォルト値：なし

## 使用例

次の例では、`myserver` という名称のサーバインスタンス上に、HTTP リスナーに関連したシステムプロパティを作成します。

```
asadmin create-system-properties --target myserver http-listener-port=1088
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 注意事項

このコマンドのオプションに次のシステムプロパティを指定しないでください。Message Queue のログの設定 (`config.properties`) に影響を与えます。

- `java.util.logging.FileHandler.count`
- `java.util.logging.FileHandler.limit`
- `java.util.logging.FileHandler.level`
- `java.util.logging.FileHandler.pattern`

### 2.11.3 delete-jvm-options

Java VM オプションの削除

## 書式

```
asadmin [asadmin-options] delete-jvm-options [--help]
  [--target target] [--profiler={true|false}]
  option-name[=option-value] [:option-name[=option-name]]...
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-jvm-options` サブコマンドは、Java VM のコマンドラインオプションを削除します。これらのオプションは、`domain.xml` での、Java に関する構成である `java-config` 要素、またはプロファイラーに関する構成である `profiler` 要素から削除されます。

削除するオプションによっては、このコマンドの実行後にドメイン管理サーバの再起動が必要です。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

指定したオプションは Java 構成の `java-config` 要素または `domain.xml` ファイルの `profiler` 要素から削除されます。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

削除する Java VM オプションの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--profiler={true|false}`

Java VM オプションがプロパティに対して存在するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`option-name=option-value`

削除する Java VM オプションを次の形式で指定します。

- オプションに名称と値がある場合は、「option-name=value」の形式で指定します。
- オプションが名称だけの場合は、「option-name」を指定します（例：-Xmx2048m）。

複数のオプションを指定する場合はコロン（:）で区切ります。コロン（:）がオプションの名称や値に含まれる場合、円記号（¥）で名称や値のコロンをエスケープする必要があります。

型：String

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、複数の Java VM のオプションを削除します。

```
asadmin delete-jvm-options --target instance1 -Doption1=value1
"-Doption1=value1:-Doption2=value2"
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.4 delete-system-property

システムプロパティ（ドメイン／構成／クラスター／サーバインスタンス）の削除

## 書式

```
asadmin [asadmin-options] delete-system-property [--help]
  [--target target] property_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-system-property` サブコマンドは、ドメイン、構成、クラスターまたはサーバインスタンスのシステムプロパティを削除します。このサブコマンドを実行する前に、システムプロパティが構成内のどこからも参照されていないことを確認してください。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するシステムプロパティの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

### *property\_name*

削除対象のシステムプロパティの名称を指定します。

型：String

指定できる値を次に示します。

- システムプロパティ名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、http-listener-port という名称のシステムプロパティを削除します。

```
asadmin delete-system-property --target instance1 http-listener-port
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.5 get

構成属性値の取得

### 書式

```
asadmin [asadmin-options] get [--help]  
      attribute-name
```

### 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

### 機能

get サブコマンドは、設定対象識別子を使用して Application Server の設定値を取得します。

設定対象識別子として指定できるパラメーターについては「set サブコマンドおよび get サブコマンドで使用するパラメーター一覧」を参照してください。



PRF 関連のパラメーターでは、「hitachi-prfs.hitachi-prf.*PRF名*.で始まるパラメーター」と「hitachi-prf-configs.hitachi-prf-config.*PRFのコンフィグ名*で始まるパラメーター」の両方に値が設定されている場合、「hitachi-prfs.hitachi-prf.*PRF名*.で始まるパラメーター」の値が有効になります。

Web サーバ関連のパラメーターでは、「hitachi-webservers.hitachi-webserver.*Webサーバ名*で始まるパラメーター」と「hitachi-webserver-configs.hitachi-webserver-config.*Webサーバのコンフィグ名*で始まるパラメーター」の両方に値が設定されている場合、「hitachi-webservers.hitachi-webserver.*Webサーバ名*で始まるパラメーター」の値が有効になります。

サーバインスタンス関連のパラメーターでは、「servers.server.*Java EEサーバ名*で始まるパラメーター」と「configs.config.*Java EEサーバの構成名*で始まるパラメーター」の両方に値が設定されている場合、「servers.server.*Java EEサーバ名*で始まるパラメーター」の値が有効になります。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*attribute-name*

設定対象識別子を指定します。少なくとも 1 つの設定対象識別子を指定してください。設定対象識別子は、Application Server の設定を参照するために使用されます。

また、設定対象識別子にはワイルドカード"\*"を利用することもできます。ワイルドカード"\*"を使用するときには設定対象識別子をダブルクォーテーション (") で囲んでください。

型：String

指定できる値を次に示します。

- *設定対象識別子*

## 使用例

次の例では、listener.http-listener-1 の属性を取得します。

```
asadmin get server.http-service.http-listener.http-listener-1.*
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.6 list

構成属性値の一覧表示

### 書式

```
asadmin [asadmin-options] list [--help]  
      dotted-parent-element-name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list サブコマンドは、Java EE Server の構成属性値を一覧表示します。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*dotted-parent-element-name*

構成属性値の名称を指定します。

型：String

指定できる値を次に示します。

- ドット表記名で指定した要素名

デフォルト値：なし

### 使用例

次の例では、構成属性値を一覧表示します。

```
asadmin list *
```

### 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.7 list-configs

ドメイン管理サーバの構成中の、名前付き構成の一覧表示

### 書式

```
asadmin [asadmin-options] list-configs [--help]
        [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-configs` サブコマンドは、ドメイン管理サーバの構成中の、名前付き構成を一覧表示します。一覧は、クラスター、インスタンス、または構成でフィルター処理できます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する構成の対象を指定します。

型：String

指定できる値を次に示します。

- *domain*  
ドメインを対象とします。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：*domain*

## 使用例

次の例では、現在のドメイン上にあるすべての構成を一覧表示します。

```
asadmin list-configs
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.8 list-jvm-options

Java VM オプションの一覧表示

### 書式

```
asadmin [asadmin-options] list-jvm-options [--help]
  [--target target]
  [--profiler={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-jvm-options サブコマンドは、Java EE Server 起動時に Java VM に渡されるコマンドラインオプションを一覧表示します。

オプションは、create-jvm-options サブコマンドおよびdelete-jvm-options サブコマンドを使用して管理されます。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

`--target target`

一覧表示する Java VM のオプションの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--profiler={false|true}`

一覧表示する Java VM のオプションがプロファイラー用かどうかを指定します。

このオプションは、プロファイラーが設定されている場合だけ、`true` に設定します。このオプションを `true` に設定してもプロファイラーが設定されていない場合、エラーが発生します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：false

## 使用例

次の例では、`instance1` という名称のサーバインスタンスについて、Java VM オプションを一覧表示します。

```
asadmin list-jvm-options --target instance1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.9 list-system-properties

ドメイン、構成、クラスターまたはサーバインスタンスのシステムプロパティの一覧表示

### 書式

```
asadmin [asadmin-options] list-system-properties [--help]
        [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-system-properties` サブコマンドは、ドメイン、構成、クラスターまたはサーバインスタンスのシステムプロパティを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するシステムプロパティの対象を指定します。

型：String

指定できる値を次に示します。

- *domain*  
ドメインを対象とします。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。
- *server*

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

デフォルト値：server

## 使用例

次の例では、localhost のシステムプロパティを一覧表示します。

```
asadmin list-system-properties
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.10 set

構成属性値の設定

### 書式

```
asadmin [asadmin-options] set [--help]
      attribute-name=value
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

set サブコマンドは、設定対象識別子を使用して、Application Server の設定値を変更します。

設定対象識別子として指定できるパラメーターについては「[set サブコマンドおよび get サブコマンドで使用するパラメーター一覧](#)」を参照してください。

アスタリスク (\*) のようなシェルやコマンドインタプリターに特別な意味を持つ文字は、例えば引数をダブルクォーテーションで囲むなど、シェルに応じてダブルクォーテーションで囲むかエスケープする必要があります。

設定値を変更することによって、サービスを有効にしたり無効にしたり、既存の要素がどのように機能するかカスタマイズしたりできます。asadmin ユーティリティーのサブコマンドを使用した変更は、関連づけられたドメイン構成ファイルに自動的に適用されます。

設定値を反映するため、このサブコマンドの実行後には、設定対象のサーバ（ドメイン管理サーバ、Webサーバ、Java EEサーバ、PRFデーモン）を再起動してください。

ファイル書き込みの失敗などによって設定値の永続化に失敗した場合、コマンドの実行には成功しますが、再起動後に設定値が反映されないことがあります。永続化に失敗した場合はドメイン管理サーバのメッセージログにエラーが出力されます。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*attribute-name=value*

設定対象識別子とその値を指定します。

型：String

指定できる値を次に示します。

- 設定対象識別子とその値

デフォルト値：なし

## 使用例

次の例では、DerbyPool コネクションプールの通常プールサイズを9に変更します。

```
asadmin set resources.jdbc-connection-pool.DerbyPool.steady-pool-size=9
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.11.11 set サブコマンドおよび get サブコマンドで使用するパラメーター一覧

ここでは、set サブコマンドおよび get サブコマンドの設定対象識別子に指定できるパラメーターについて説明します。



## リソース関連

applications.application.アプリケーション名.module.モジュール名.resources で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
1	connector-connection-pool.コネクタコネクションプール名.hitachi-connection-sharing-component-enabled	コネクションシェアリングのコンポーネント内シェアリングの有効/無効を設定します。モジュールスコープのコネクタコネクションプールに設定します。	true false	なし	false	asadmin set applications.application.myapplication.module.mymodule.resources.connector-connection-pool.myconnectorconnectionpool.hitachi-connection-sharing-component-enabled=false
2	jdbc-connection-pool.JDBCコネクションプール名.hitachi-connection-sharing-component-enabled	コネクションシェアリングのコンポーネント内シェアリングの有効/無効を設定します。モジュールスコープの JDBC コネクションプールに設定します。	true false	なし	false	asadmin set applications.application.myapplication.module.mymodule.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-connection-sharing-component-enabled=false
3	jdbc-connection-pool.JDBCコネクションプール名.hitachi-jdbc-transaction-rollback-enabled	コネクションクローズ時の JDBC トランザクションのロールバックの有効/無効を設定します。モジュールスコープの JDBC コネクションプールに設定します。	true false	なし	true	asadmin set applications.application.myapplication.module.mymodule.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-jdbc-transaction-rollback-enabled=true
4	jdbc-connection-pool.JDBCコネクションプール名.hitachi-validation-timeout	コネクション障害検知のタイムアウトの時間を、秒単位で指定します。モジュールスコープの JDBC コネクションプールに設定します。	1~2147483647	なし	5	asadmin set applications.application.myapplication.module.mymodule.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-validation-timeout=5

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
5	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.hitachi-validation-timeout-enabled	コネクション障害検知のタイムアウトの有効/無効を設定します。モジュールスコープのJDBCコネクションプールに設定します。	true false	なし	true	asadmin set applications.application.myapplication.module.mymodule.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-validation-timeout-enabled=true

applications.application.*アプリケーション名*.resources で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
1	connector-connection-pool. <i>コネクターコネクションプール</i> 名.hitachi-connection-sharing-component-enabled	コネクションシェアリングのコンポーネント内シェアリングの有効/無効を設定します。アプリケーションスコープのコネクターコネクションプールに設定します。	true false	なし	false	asadmin set applications.application.myapplication.resources.connector-connection-pool.myconnectorconnectionpool.hitachi-connection-sharing-component-enabled=false
2	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.hitachi-connection-sharing-component-enabled	コネクションシェアリングのコンポーネント内シェアリングの有効/無効を設定します。アプリケーションスコープのJDBCコネクションプールに設定します。	true false	なし	false	asadmin set applications.application.myapplication.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-connection-sharing-component-enabled=false
3	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.hitachi-jdbc-transaction-rollback-enabled	コネクションクローズ時のJDBCトランザクションのロールバックの有効/無効を設定します。アプリケーションスコープのJDBCコネクションプールに設定します。	true false	なし	true	asadmin set applications.application.myapplication.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-jdbc-transaction-rollback-enabled=true

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
4	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.hitachi-validation-timeout	コネクション障害検知のタイムアウトの時間を、秒単位で指定します。アプリケーションスコープのJDBCコネクションプールに設定します。	1～ 2147483647	なし	5	asadmin set applications.application.myapplication.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-validation-timeout=5
5	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.hitachi-validation-timeout-enabled	コネクション障害検知のタイムアウトの有効/無効を設定します。アプリケーションスコープのJDBCコネクションプールに設定します。	true false	なし	true	asadmin set applications.application.myapplication.resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-validation-timeout-enabled=true

resources で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
1	connector-connection-pool.コネクターコネクションプール名.hitachi-connection-sharing-component-enabled	コネクションシェアリングのコンポーネント内シェアリングの有効/無効を設定します。ドメインに作成したコネクターコネクションプールに設定します。	true false	なし	false	asadmin set resources.connector-connection-pool.myconnectorconnectionpool.hitachi-connection-sharing-component-enabled=false
2	connector-connection-pool.コネクターコネクションプール名.idle-timeout-in-seconds	アイドル状態のコネクションがタイムアウトするまでの時間を、秒単位で設定します。	1～ 2147483647	300	なし	asadmin set resources.connector-connection-pool.ConnectionPool1.idle-timeout-in-seconds=200
3	connector-connection-pool.コネクターコネクションプール名.max-pool-size	クライアントのリクエストに対応するために作成できるコネクションの最大数を設定します。	1～ 2147483647	24	なし	asadmin set resources.connector-connection-pool.ConnectionPool1.max-pool-size=60
4	connector-connection-pool.コ	コネクションが使用できない場合、コネ	0～ 2147483647	60000	なし	asadmin set resources.connector

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
	ネクターコネクションプール名.max-wait-time-in-millis	クッションが作成されるまでに呼び出し側が待機しなければならない時間をミリ秒単位で設定します。				r-connection-pool.ConnectionPool1.max-wait-time-in-millis=700000
5	connector-connection-pool.コネクターコネクションプール名.pool-resize-quantity	プールがコネクション数をスケールアップまたはスケールダウンするときの単位量を設定します。	1～ 2147483647	2	なし	asadmin set resources.connector-connection-pool.ConnectionPool1.pool-resize-quantity=4
6	connector-connection-pool.コネクターコネクションプール名.property.プロパティ名	プールを構成するときのプロパティ名およびその値を設定します。	Type: String	なし	なし	asadmin set resources.connector-connection-pool.ConnectionPool1.property.User=User1
7	connector-connection-pool.コネクターコネクションプール名.steady-pool-size	プールに保持されるコネクション数の最小値を設定します。	0～ 2147483647	24	なし	asadmin set resources.connector-connection-pool.ConnectionPool1.steady-pool-size=30
8	jdbc-connection-pool.JDBCコネクションプール名.hitachi-connection-sharing-component-enabled	コネクションシェアリングのコンポーネント内シェアリングの有効/無効を設定します。ドメインに作成した JDBC コネクションプールに設定します。	true false	なし	false	asadmin set resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-connection-sharing-component-enabled=false
9	jdbc-connection-pool.JDBCコネクションプール名.hitachi-jdbc-transaction-rollback-enabled	コネクションクローズ時の JDBC トランザクションのロールバックの有効/無効を設定します。ドメインに作成した JDBC コネクションプールに設定します。	true false	なし	true	asadmin set resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-jdbc-transaction-rollback-enabled=true
10	jdbc-connection-pool.JDBCコネクションプール名.hitachi-validation-timeout	コネクション障害検知のタイムアウトの時間を、秒単位で指定します。ドメインに作成した JDBC コ	1～ 2147483647	なし	5	asadmin set resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-

## 2. Java EE サーバで使用するコマンド

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
		ネクションプールに設定します。				validation-timeout=5
11	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.hitachi-validation-timeout-enabled	コネクション障害検知のタイムアウトの有効/無効を設定します。ドメインに作成した JDBC コネクションプールに設定します。	true false	なし	true	asadmin set resources.jdbc-connection-pool.myjdbcconnectionpool.hitachi-validation-timeout-enabled=true
12	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.idle-timeout-in-seconds	アイドル状態のコネクションがタイムアウトするまでの時間を、秒単位で設定します。 アプリケーション内に使用できないコネクションが蓄積しないように、このタイムアウト値にはデータベース側のタイムアウト値よりも常に小さい値を設定してください。	0～ 2147483647	300	なし	asadmin set resources.jdbc-connection-pool.ConnectionPool1.idle-timeout-in-seconds=200
13	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.max-pool-size	作成できるコネクションの最大数を設定します。	1～ 2147483647	24	なし	asadmin set resources.jdbc-connection-pool.ConnectionPool1.max-pool-size=60
14	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.max-wait-time-in-millis	JDBC コネクションタイムアウトが送信されるまでの、呼び出し側が待機する時間をミリ秒単位で設定します。 0 を指定した場合、呼び出し側の待機時間は無制限となります。	0～ 2147483647	60000	なし	asadmin set resources.jdbc-connection-pool.ConnectionPool1.max-wait-time-in-millis=70000
15	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.pool-resize-quantity	idle-timeout-in-seconds タイマーがタイムアウトしたときに削除されるコネ	1～ 2147483647	2	なし	asadmin set resources.jdbc-connection-pool.ConnectionPool

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
		クッション数を設定します。				l1.pool-resize-quantity=4
16	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.property.プロパティ名	プールを構成するときのプロパティ名およびその値を設定します。	Type: String	なし	なし	asadmin set resources.jdbc-connection-pool.ConnectionPool1.property.User=User1
17	jdbc-connection-pool. <i>JDBC</i> コネクションプール名.steady-pool-size	プールに保持されるコネクション数の最小値を設定します。	0~2147483647	24	なし	asadmin set resources.jdbc-connection-pool.ConnectionPool1.steady-pool-size=30
18	mail-resource. <i>JavaMail</i> リソース名.property.プロパティ名	JavaMail リソースを構成するときのプロパティ名およびその値を設定します。	Type: String	なし	なし	asadmin set resources.mail-resource.mymailresource.property.mail-from=xyz@abc.com

## PRF 関連

hitachi-prfs.hitachi-prf.*PRF*名.で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	property.拡張プロパティ名	PRF に追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。 追加した拡張プロパティは、PRF のサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.ex_timeout=600
2	property.env_環境変数名	PRF 起動時に設定する環境変数を指定します。複数の環境変数を指定する場合は、この標準プロパ	Type: String 半角英数字と次の記号だけが指定	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.env_TZ=JST-9

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>ティーを複数指定します。</p> <p>環境変数の指定値に <code>\${apserver.home}</code> が含まれている場合は、<i>Application Server</i> インストールディレクトリーに置換されます。</p> <p>なお、このプロパティー値に環境変数値を引き継ぐ指定 (<code>%PATH%;C:¥temp</code> や <code>\${PATH}:/temp</code> という指定) をしても、その環境変数の値は引き継がれません。指定する場合は、次のどれかの場所で定義してください。</p> <ul style="list-style-type: none"> <li>ノード上 OS のシステム環境変数</li> <li>ノード上の <code>asenv</code> の定義</li> <li>サーバテンプレートの起動用バッチまたはシェルスクリプト</li> </ul> <p>このプロパティーで指定できない文字をサーバの環境変数に設定する必要がある場合は、次の方法で環境変数を設定してください。</p> <ul style="list-style-type: none"> <li>環境変数設定用にサーバに拡張プロパティーを設け、この拡張プロパティーに環境変数に設定する値を設定する。</li> </ul> <p>[環境変数設定用の拡張プロパティーの例]            拡張プロパティー名：<code>ex_myenv</code></p>	<p>できます。</p> <p>パーセント (%)、単価記号 (@)、プラス (+)、ハイフン (-)、アンダースコア (_)、コンマ (,)、ピリオド (.)。</p>			

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		拡張プロパティの値：環境変数に設定する値 <ul style="list-style-type: none"> <li>サーバテンプレートの中のサーバを起動するスクリプト中に、拡張プロパティの値を環境変数に設定する処理を追加する。 [追加する行の例] MY_ENV=\$ {property.ex_my env.value} export MY_ENV</li> </ul>				
3	property.log-file-count	PRF が出力するログファイルの面数を指定します。 このプロパティは、PRF の cprfstart コマンドの -PrfLogFileCount オプションに対応します。	1~32	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.log-file-count=16
4	property.log-file-size	PRF が出力するログファイルサイズの上限を、メガバイト単位で指定します。 このプロパティは、PRF の cprfstart コマンドの -PrfLogFileSize オプションに対応します。	1~100	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.log-file-size=20
5	property.log-shift-time	PRF が出力するログファイルのシフト時刻を指定します。 このプロパティは、PRF の cprfstart コマンドの -PrfLogShiftTime オプションに対応します。	HHMMSS (000000~235959)	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.log-shift-time=121530
6	property.prf-spool	PRF トレース出力ディレクトリーを指定します。	Type: String	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		デフォルトの出力先は、HJES_LOGSDIR/nodes/ノード名/PRF 識別子です。				prf.PRf1.property.prf-spool=C:¥PRFSP00L¥PRF1
7	property.trace-buffer-size	共用メモリーに確保する PRF トレースファイルのバッファサイズを、キロバイト単位で指定します。 このプロパティは、PRF の cprfstart コマンドの -PrfTraceBufferSize オプションに対応します。	512~102400	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.trace-buffer-size=16384
8	property.trace-count	PRF トレースファイルの面数を指定します。 このプロパティは、PRF の cprfstart コマンドの -PrfTraceCount オプションに対応します。	3~256	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.trace-count=8
9	property.trace-file-size	PRF トレースファイルの 1 ファイル当たりの最大容量を、キロバイト単位で指定します。 このプロパティは、PRF の cprfstart コマンドの -PrfTraceFileSize オプションに対応します。	1024~1048576	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.trace-file-size=16384
10	property.trace-level	PRF トレース取得レベルを、4 バイトの 16 進数 (8 桁の値) で指定します。先頭に 0x を付けられますが、0x は無視されます。各インデックス番号のレベル値を左からコンマ (,) で区切ることで複数指定できます。	4 バイトの 16 進数 (8 桁の値) [4 バイトの 16 進数 (8 桁の値) ...]	プロパティ削除	なし	asadmin set hitachi-prfs.hitachi-prf.PRf1.property.trace-level=0x44445555

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		このプロパティは、PRF の <code>cprfstart</code> コマンドの <code>PrfTraceLevel</code> オプションに対応します。				
11	<code>hitachi-prf-config-ref</code>	PRF の設定要素を指定します。この属性の値は <code>set</code> サブコマンドで変更できません。	ドメインに存在する PRF のコンフィグ名	変更不可	なし	変更不可

#### 注※

デフォルト値欄に「プロパティ削除」と記載されているパラメーターは、空文字を指定することでプロパティ自体が削除されます。

#### hitachi-prf-configs.hitachi-prf-config.PRFのコンフィグ名で始まるパラメーター

パラメーター名が `property.` で始まるパラメーターを `hitachi-prfs.hitachi-prf.PRF名.` で始まるパラメーターでも定義している場合は、`hitachi-prfs.hitachi-prf.PRF名.` で始まるパラメーターの値が優先されます。

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	<code>hitachi-manage-info.running-watch-interval-in-seconds</code>	PRF の動作確認間隔を、秒単位で指定します。	1~86400	省略不可	10	<code>asadmin set hitachi-prf-configs.hitachi-prf-config.PRF1-config.hitachi-manage-info.running-watch-interval-in-seconds=20</code>
2	<code>hitachi-manage-info.start-timeout-in-seconds</code>	PRF の起動監視時間を、秒単位で指定します。0 を指定した場合は、監視しません。	0~3600	省略不可	60	<code>asadmin set hitachi-prf-configs.hitachi-prf-config.PRF1-config.hitachi-manage-info.start-timeout-in-seconds=180</code>
3	<code>hitachi-manage-info.start-type</code>	PRF の起動方法を指定します。 <ul style="list-style-type: none"> <li><code>direct</code> : 直接起動</li> </ul>	<code>direct</code>	省略不可	<code>direct</code>	<code>asadmin set hitachi-prf-configs.hitachi-prf-config.PRF1-config.hitachi-</code>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
						manage-info.start-type=direct
4	hitachi-manage-info.starting-watch-interval-in-seconds	PRF の起動時監視間隔を、秒単位で指定します。	1~86400	省略不可	1	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.hitachi-manage-info.starting-watch-interval-in-seconds=3
5	hitachi-manage-info.starting-watch-start-time-in-seconds	PRF で起動コマンドを実行してから動作確認を開始するまでの時間を、秒単位で指定します。	0~86400	省略不可	0	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.hitachi-manage-info.starting-watch-start-time-in-seconds=10
6	hitachi-manage-info.stop-timeout-in-seconds	PRF の停止監視時間を、秒単位で指定します。0 を指定した場合は、監視しません。	0~1800	省略不可	60	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.hitachi-manage-info.stop-timeout-in-seconds=180
7	hitachi-manage-info.stopping-watch-interval-in-seconds	PRF の停止時監視間隔を、秒単位で指定します。	1~86400	省略不可	1	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.hitachi-manage-info.stopping-watch-interval-in-seconds=3
8	hitachi-manage-info.template-path	PRF のサーバテンプレートのパスを絶対パスで指定します。	存在するディレクトリーパス	省略不可	ドメイン ルートディレクトリー/ ドメイン名/ server_templates/ サーバ種別	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.hitachi-manage-info.template-path=C:¥temp¥prf

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
9	property.拡張プロパティ	PRF に追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。  追加した拡張プロパティは、PRF のサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティ削除	なし	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.ex_timeout=600
10	property.env_環境変数名	PRF 起動時に設定する環境変数を指定します。複数の環境変数を指定する場合は、この標準プロパティを複数指定します。  環境変数の指定値に \${apserver.home} が含まれている場合は、 <i>Application Server</i> のインストールディレクトリに置換されます。  なお、このプロパティ値に環境変数値を引き継ぐ指定 (%PATH%;C:¥temp や \${PATH}:/temp という指定) をしても、その環境変数の値は引き継がれません。指定する場合は、次のどれかの場所で定義してください。  <ul style="list-style-type: none"> <li>• ノード上 OS のシステム環境変数</li> <li>• ノード上の asenv の定義</li> <li>• サーバテンプレートの起動用バッチ</li> </ul>	Type: String  半角英数字と次の記号だけが指定できます。 パーセント (%)、単価記号 (@)、プラス (+)、ハイフン (-)、アンダースコア (_)、コンマ (,), ピリオド (.)。	プロパティ削除	なし	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.env_TZ=JST-9

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>またはシェルスクリプト</p> <p>このプロパティーで指定できない文字をサーバの環境変数に設定する必要がある場合は、次の方法で環境変数を設定してください。</p> <ul style="list-style-type: none"> <li>環境変数設定用にサーバに拡張プロパティーを設け、この拡張プロパティーに環境変数に設定する値を設定する。 [環境変数設定用の拡張プロパティーの例] 拡張プロパティー名: <code>ex_myenv</code> 拡張プロパティーの値: <i>環境変数に設定する値</i></li> <li>サーバテンプレートの中のサーバを起動するスクリプト中に、拡張プロパティーの値を環境変数に設定する処理を追加する。 [追加する行の例] <code>MY_ENV=\${property.ex_myenv.value}</code> <code>export MY_ENV</code></li> </ul>				
11	<code>property.log-file-count</code>	<p>PRF が出力するログファイルの面数を指定します。</p> <p>このプロパティーは、PRF の <code>cprfstart</code> コマンドの <code>PrfLogFileCount</code> オプションに対応します。</p>	1~32	プロパティー削除	8	<code>asadmin set hitachi-prf-configs.hitachi-prf-config.PRF1-config.property.log-file-count=16</code>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
12	property.log-file-size	PRF が出力するログファイルサイズの上限を、メガバイト単位で指定します。 このプロパティは、PRF のcprfstart コマンドの-PrfLogFileSize オプションに対応します。	1~100	プロパティ削除	10	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.log-file-size=20
13	property.log-shift-time	PRF が出力するログファイルを時刻でシフトさせる場合のシフト時刻を指定します。 このプロパティは、PRF のcprfstart コマンドの-PrfLogShiftTime オプションに対応します。	HHMMSS (000000~235959)	プロパティ削除	000000	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.log-shift-time=121530
14	property.prf-spool	PRF トレース出力ディレクトリーを指定します。 デフォルトの出力先は、HJES_LOGSDIR/nodes/ノード名/PRF 識別子です。	Type: String	プロパティ削除	なし	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.prf-spool=C:¥PRFSPool¥PRF1
15	property.trace-buffer-size	共用メモリーに確保する PRF トレースファイルのバッファサイズを、キロバイト単位で指定します。 このプロパティは、PRF のcprfstart コマンドの-PrfTraceBufferSize オプションに対応します。	512~102400	プロパティ削除	8192	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.trace-buffer-size=16384
16	property.trace-count	PRF トレースファイルの面数を指定します。 このプロパティは、PRF のcprfstart コマンドの-	3~256	プロパティ削除	4	asadmin set hitachi-prf-configs.hitachi-prf-config.PRf1-config.property.trace-count=8

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		PrfTraceCount オプションに対応します。				
17	property.trace-file-size	PRF トレースファイルの 1 ファイル当たりの最大容量を、キロバイト単位で指定します。 このプロパティは、PRF の cprfstart コマンドの PrfTraceFileSize オプションに対応します。	1024～1048576	プロパティ削除	262144	asadmin set hitachi-prf-configs.hitachi-prf-config.PRF1-config.property.trace-file-size=16384
18	property.trace-level	PRF トレース取得レベルを、4 バイトの 16 進数 (8 桁の値) で指定します。先頭に 0x を付けられますが、0x は無視されます。各インデックス番号のレベル値を左からコンマ (,) で区切ることで複数指定できます。 このプロパティは、PRF の cprfstart コマンドの PrfTraceLevel オプションに対応します。	4 バイトの 16 進数 (8 桁の値) [,4 バイトの 16 進数 (8 桁の値) ...]	プロパティ削除	0	asadmin set hitachi-prf-configs.hitachi-prf-config.PRF1-config.property.trace-level=0x44445555

#### 注※

デフォルト値欄に「プロパティ削除」と記載されているパラメーターは、空文字を指定することでプロパティ自体が削除されます。

## Web サーバ関連

hitachi-webservers.hitachi-webserver.*Web*サーバ名で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	hitachi-relations.hitachi-relation-type-ref.hitachi-	サーバ間関連の順序番号を指定します。	0～65535	省略不可	なし	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
	relation.関連名.order					relations.redirect.hitachi-relation.redirect1.order=10
2	hitachi-relations.prf-relation.hitachi-relation.関連名.property.拡張プロパティ	Web サーバの PRF 関連に追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。  追加した拡張プロパティは、サーバ間関連の関連元である Web サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティ削除	なし	hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.prf-relation.hitachi-relation.prfref2.property.ex_timeout=600
3	hitachi-relations.redirect.hitachi-relation.関連名.property.拡張プロパティ	リダイレクト関連に追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。  追加した拡張プロパティは、サーバ間関連の関連元である Web サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティ削除	なし	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect.hitachi-relation.redirector1.property.ex_timeout=600
4	hitachi-relations.redirect.hitachi-relation.関連名.property.negative	リダイレクターの動作の定義で、マッピング定義に対するリクエストを除外するかどうかを指定します。  このプロパティは、Web Server の ProxyPass ディレク	true false	プロパティ削除	false	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect.hitachi-relation.redirector1.property.ex_timeout=600



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		タイプの!オプションに対応します。				r1.property.negative=true
5	hitachi-relations.redirect.hitachi-relation. <i>関連</i> 名.property.network-listener	リダイレクト先となるネットワークリスナー名を指定します。 このプロパティは、Web Server の ProxyPass ディレクティブの URL に対応します。URL には、指定したネットワークリスナー名から特定したホスト名とポート番号が設定されます。	ネットワークリスナー名の文字列	プロパティ削除	http-listener-1	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect.hitachi-relation.redirector1.property.network-listener=http-listener-1
6	hitachi-relations.redirect.hitachi-relation. <i>関連</i> 名.property.path	リダイレクターの動作の定義で、マッピング定義のローカルの仮想パスの名前を指定します。 このプロパティは、Web Server の ProxyPass ディレクティブのパス名指定に対応します。	httpsd.conf ファイルの ProxyPass に定義する path (ローカルの仮想パスの名前) の形式	プロパティ削除	/	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect.hitachi-relation.redirector1.property.path=/news/
7	hitachi-relations.redirect.hitachi-relation. <i>関連</i> 名.property.proxy-pass-option	リダイレクターの動作の定義で、マッピング定義のコネクションプーリングの調整パラメーターを指定します。 このプロパティは、Web Server の ProxyPass ディレクティブの「キー=値」指定に対応します。	httpsd.conf ファイルの ProxyPass に定義するコネクションプーリングの調整パラメーターの形式 ([key=value key=value ...])	プロパティ削除	空文字	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect.hitachi-relation.redirector1.property.proxy-pass-option=timeout=300
8	hitachi-relations.redirect.hitachi-relation. <i>関連</i> 名.property.set-proxy-pass-reverse-cookie	リダイレクターの動作の定義で、Set-Cookie ヘッダーの再割り当てを行うかどうかを指定します。 このプロパティは、Web Server の	true false	プロパティ削除	true	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		HWSProxyPassReverseCookie ディレクティブに対応します。				.hitachi-relation.redirector1.property.set-proxy-pass-reverse-cookie=true
9	hitachi-relations.redirect. hitachi-relation. <i>関連</i> 名.property.sticky session	リダイレクターの動作の定義で、Sticky を有効にするかどうかを指定します。 このプロパティは、Web Server の ProxyPass ディレクティブの stickysession キーに対応します。 また、このプロパティにtrue を指定すると、必要な Header ディレクティブが追加されます。	true false	プロパティ削除	true	asadmin set hitachi-webservers.hitachi-webserver.Web1.hitachi-relations.redirect.hitachi-relation.redirector1.property.sticky session=true
10	hitachi-webserver-config-ref	Web サーバのコンフィグ名を指定します。 この属性の値はset サブコマンドで変更できません。	ドメインに存在する Web サーバのコンフィグ名	変更不可	なし	編集不可
11	property. <i>拡張プロパティ</i>	Web サーバに追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_ から始まる名称で指定します。 追加した拡張プロパティは、Web サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティ削除	なし	asadmin set hitachi-webservers.hitachi-webserver.Web1.property.ex_timeout=600
12	property.user	Web Server のサーバプロセスが動作するときのユーザー名	Type: String	プロパティ削除	なし	asadmin set hitachi-webservers.hitachi

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		を指定します。このプロパティーは、Web Server のUser ディレクティブに対応します。				- webserver.Web1.property.user=user01
13	property.group	Web Server のサーバプロセスが動作するときのグループ名を指定します。このプロパティーは、Web Server の Group ディレクティブに対応します。	Type: String	プロパティー削除	なし	asadmin set hitachi-webserver.hitachi-webserver.Web1.property.group=group01
14	property.env_環境変数名	Web サーバ起動時に設定する環境変数を指定します。複数の環境変数を指定する場合は、この標準プロパティーを複数指定します。 環境変数の指定値に \${apserver.home} が含まれている場合は、Application Server のインストールディレクトリーに置換されます。 なお、このプロパティー値に環境変数値を引き継ぐ指定 (%PATH%;C:¥temp や \${PATH}:/temp という指定) をしても、その環境変数の値は引き継がれません。指定する場合は、次のどれかの場所で定義してください。 <ul style="list-style-type: none"> <li>ノード上 OS のシステム環境変数</li> <li>ノード上の asenv の定義</li> <li>サーバテンプレートの起動用シェルスクリプト</li> </ul>	Type: String 半角英数字と次の記号だけが指定できます。 パーセント (%)、単価記号 (@)、プラス (+)、ハイフン (-)、アンダースコア (_)、コンマ (,)、ピリオド (.)。	プロパティー削除	なし	asadmin set hitachi-webserver.hitachi-webserver.Web1.property.env_TZ=JST-9

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>このプロパティーで指定できない文字をサーバの環境変数に設定する必要がある場合は、次の方法で環境変数を設定してください。</p> <ul style="list-style-type: none"> <li>環境変数設定用にサーバに拡張プロパティーを設け、この拡張プロパティーに環境変数に設定する値を設定する。 [環境変数設定用の拡張プロパティーの例] 拡張プロパティー名：<code>ex_myenv</code> 拡張プロパティーの値：<i>環境変数に設定する値</i></li> <li>サーバテンプレートの中のサーバを起動するスクリプト中に、拡張プロパティーの値を環境変数に設定する処理を追加する。 [追加する行の例] <code>MY_ENV=\$ {property.ex_myenv.value} export MY_ENV</code></li> </ul>				
15	<code>property.listen-add-count</code>	<p>複数の Listen ディレクティブを指定する場合に、追加する Listen ディレクティブの数を指定します。このプロパティーは、プロパティー <code>listen-port</code> および <code>listen-host</code> に指定した Listen ディレクティブ設定以外の追加数を指定するため、Listen ディレクティ</p>	0～ 2147483647	プロパ ティー削除	なし	<code>asadmin set hitachi- webservers.hitachi- webserver.Web1.pro perty.listen-add- count=2</code>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>プの総数は、「このプロパティの指定値+1」になります。デフォルトは0です。</p> <p>このプロパティに指定した数だけ、次のプロパティを指定する必要があります。</p> <ul style="list-style-type: none"> <li>listen-add-port<math>n</math>*</li> <li>listen-add-host<math>n</math>*</li> </ul> <p>注※</p> <p><math>n</math> : 1~このプロパティの指定値</p>				
16	property.listen-add-host $n$	<p>追加 Listen ディレクティブの IP アドレスを指定します。プロパティ名の <math>n</math> には、1~listen-add-count の指定値の値を指定します。<math>n</math> が同じ数値のlisten-add-port<math>n</math> と対で、Listen ディレクティブが指定されます。</p> <p>このプロパティは省略できます。</p> <p>1~listen-add-count の指定値の範囲外の値を <math>n</math> に指定した場合、そのポート番号は Web サーバに適用されません。</p> <p>このプロパティは、Web Server の Listen ディレクティブに対応します。</p>	<p>ホスト名  IPv4 ドット記法  IPv6 ドット記法 (前後に[]を指定)</p>	プロパティ削除	なし	<pre>asadmin set hitachi-webserver.hitachi-webserver.Web1.property.listen-add-host1=192.168.1.102</pre>
17	property.listen-add-port $n$	<p>追加 Listen ディレクティブのポート番号を指定します。プロパティ名の <math>n</math> には、1~listen-add-count の指定値の値を指定します。<math>n</math> が</p>	1~65535	プロパティ削除	なし	<pre>asadmin set hitachi-webserver.hitachi-webserver.Web1.property.listen-add-port1=82</pre>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		同じ数値の <code>listen-add-hostn</code> と対で、 <code>Listen</code> ディレクティブが指定されます。 1～ <code>listen-add-count</code> の指定値の範囲外の値を <i>n</i> に指定した場合、そのポート番号は Web サーバに適用されません。 このプロパティは、Web Server の <code>Listen</code> ディレクティブに対応します。				
18	<code>property.listen-host</code>	リクエストを受け付ける IP アドレスを指定します。 このプロパティは省略できます。 このプロパティは、Web Server の <code>Listen</code> ディレクティブに対応します。	ホスト名  <i>IPv4</i> ドット記法  <i>IPv6</i> ドット記法 (前後に[]を指定)	プロパティ削除	なし	<code>asadmin set hitachi-webservers.hitachi-webserver.Web1.property.listen-host=192.168.1.101</code>
19	<code>property.listen-port</code>	リクエストを受け付けるポート番号を指定します。 このプロパティは、Web Server の <code>Listen</code> ディレクティブに対応します。	1～65535	プロパティ削除	なし	<code>asadmin set hitachi-webservers.hitachi-webserver.Web1.property.listen-port=81</code>
20	<code>property.max-request-workers</code>	同時に接続できるクライアントの最大数を指定します。 このプロパティは、Web Server の <code>MaxRequestWorkers</code> ディレクティブに対応します。	1～1024	プロパティ削除	なし	<code>asadmin set hitachi-webservers.hitachi-webserver.Web1.property.max-request-workers=300</code>
21	<code>property.server-name</code>	Web Server のサーバ名およびポート番号を指定します。 このプロパティは、Web Server の <code>ServerName</code> ディレクティブに対応します。	Type: String	プロパティ削除	なし	<code>asadmin set hitachi-webservers.hitachi-webserver.Web1.property.server-</code>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
						name=www.host1.co.jp
22	property.start-servers	Web サーバ起動時のサーバプロセス数を指定します。 このプロパティは、Web Server の StartServers ディレクティブに対応します。	1~1024	プロパティ削除	なし	asadmin set hitachi-webserver.hitachi-webserver.Web1.property.start-servers=10

#### 注※

デフォルト値欄に「プロパティ削除」と記載されているパラメーターは、空文字を指定することでプロパティ自体が削除されます。

hitachi-webserver-configs.hitachi-webserver-config.*Web*サーバのコンフィグ名で始まるパラメーター  
パラメーター名がproperty.で始まるパラメーターをhitachi-webserver.hitachi-webserver.*Web*サーバ名で始まるパラメーターでも定義している場合は、hitachi-webserver.hitachi-webserver.*Web*サーバ名で始まるパラメーターの値が優先されます。

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	hitachi-manage-info.running-watch-interval-in-seconds	Web サーバの動作確認間隔を、秒単位で指定します。	1~86400	省略不可	10	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.hitachi-manage-info.running-watch-interval-in-seconds=20
2	hitachi-manage-info.start-timeout-in-seconds	Web サーバの起動監視時間を、秒単位で指定します。0を指定した場合は、監視しません。	0~3600	省略不可	60	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.hitachi-manage-info.start-timeout-in-seconds=180
3	hitachi-manage-info.start-type	Web サーバの起動方法を指定します。 <ul style="list-style-type: none"> <li>direct: 直接起動</li> </ul>	direct	省略不可	direct	asadmin set hitachi-webserver-configs.hitachi-webserver-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
						config.Web1- config.hitachi- manage-info.start- type=direct
4	hitachi-manage- info.starting- watch-interval-in- seconds	Web サーバの起動 時監視間隔を、秒単 位で指定します。	1~86400	省略不可	1	asadmin set hitachi-webserver- configs.hitachi- webserver- config.Web1- config.hitachi- manage- info.starting- watch-interval-in- seconds=3
5	hitachi-manage- info.starting- watch-start-time- in-seconds	Web サーバの起動 で、起動コマンドを 実行してから動作確 認を開始するまでの 時間を、秒単位で指 定します。	0~86400	省略不可	0	asadmin set hitachi-webserver- configs.hitachi- webserver- config.Web1- config.hitachi- manage- info.starting- watch-start-time- in-seconds=20
6	hitachi-manage- info.stop-timeout- in-seconds	Web サーバの停止 監視時間を、秒単位 で指定します。0を 指定した場合は、監 視しません。	0~1800	省略不可	60	asadmin set hitachi-webserver- configs.hitachi- webserver- config.Web1- config.hitachi- manage-info.stop- timeout-in- seconds=180
7	hitachi-manage- info.stopping- watch-interval-in- seconds	Web サーバの停止 時監視間隔を、秒単 位で指定します。	1~86400	省略不可	1	asadmin set hitachi-webserver- configs.hitachi- webserver- config.Web1- config.hitachi- manage- info.stopping- watch-interval-in- seconds=3



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
8	hitachi-manage-info.template-path	Web サーバのサーバテンプレートのパスを絶対パスで指定します。	存在するディレクトリーパス	省略不可	ドメイン ルートディレクトリー/ ドメイン名/ server_templates/ サーバ種別	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.hitachi-manage-info.template-path=C:¥temp ¥webserver
9	property.拡張プロパティー	Web サーバに追加する拡張プロパティーとして、任意のプロパティー名およびその値を指定します。プロパティー名はex_から始まる名称で指定します。追加した拡張プロパティーは、Web サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティー削除	なし	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.ex_timeout=600
10	property.user	Web Server のサーバプロセスが動作するときのユーザー名を指定します。このプロパティーは、Web Server のUser ディレクティブに対応します。	Type: String	プロパティー削除	なし	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.user=user01
11	property.group	Web Server のサーバプロセスが動作するときのグループ名を指定します。このプロパティーは、Web Server のGroup ディレクティブに対応します。	Type: String	プロパティー削除	なし	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.group=group01
12	property.env_環境変数名	Web サーバ起動時に設定する環境変数を指定します。複数の環境変数を指定する場合は、この標準	Type: String 半角英数字と次の記号だけが指定	プロパティー削除	なし	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>プロパティを複数指定します。</p> <p>環境変数の指定値に <code>\${apserver.home}</code> が含まれている場合は、<i>Application Server</i> のインストールディレクトリーに置換されます。</p> <p>なお、このプロパティ値に環境変数値を引き継ぐ指定 (<code>%PATH%;C:¥temp</code> や <code>{PATH}:/temp</code> という指定) をしても、その環境変数の値は引き継がれません。指定する場合は、次のどれかの場所で定義してください。</p> <ul style="list-style-type: none"> <li>• ノード上 OS のシステム環境変数</li> <li>• ノード上の <code>asenv</code> の定義</li> <li>• サーバテンプレートの起動用シェルスクリプト</li> </ul> <p>このプロパティで指定できない文字をサーバの環境変数に設定する必要がある場合は、次の方法で環境変数を設定してください。</p> <ul style="list-style-type: none"> <li>• 環境変数設定用にサーバに拡張プロパティを設け、この拡張プロパティに環境変数に設定する値を設定する。</li> </ul> <p>[環境変数設定用の拡張プロパティの例] 拡張プロパティ名: <code>ex_myenv</code></p>	<p>できます。</p> <p>パーセント (%)、単価記号 (@)、プラス (+)、ハイフン (-)、アンダースコア (_)、コンマ (,), ピリオド (.)。</p>			<pre>config.property.env_TZ=JST-9</pre>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>拡張プロパティーの値：環境変数に設定する値</p> <ul style="list-style-type: none"> <li>サーバテンプレートの中のサーバを起動するスクリプト中に、拡張プロパティーの値を環境変数に設定する処理を追加する。</li> </ul> <p>[追加する行の例]</p> <pre>MY_ENV=\$ {property.ex_my env.value} export MY_ENV</pre>				
13	property.listen-add-count	<p>複数の Listen ディレクティブを指定する場合に、追加する Listen ディレクティブの数を指定します。このプロパティーは、プロパティー listen-port および listen-host に指定した Listen ディレクティブ設定以外の追加数を指定するため、Listen ディレクティブの総数は、「このプロパティーの指定値+1」になります。デフォルトは0です。</p> <p>このプロパティーに指定した数だけ、次のプロパティーを指定する必要があります。</p> <ul style="list-style-type: none"> <li>listen-add_port<math>n</math>*</li> <li>listen-add_host<math>n</math>*</li> </ul> <p>注※</p> <p><math>n</math> : 1~このプロパティーの指定値</p>	0~ 2147483647	プロパ ティー削除	なし（未定義）	asadmin set hitachi-webserver- configs.hitachi- webserver- config.Web1- config.property.li sten-add-count=2

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
14	property.listen-add-hostn	追加 Listen ディレクティブの IP アドレスを指定します。プロパティ名の <i>n</i> には、1～listen-add-count の指定値の値を指定します。 <i>n</i> が同じ数値のlisten-add-portn と対で、Listen ディレクティブが指定されます。このプロパティは省略できます。 1～listen-add-count の指定値の範囲外の値を <i>n</i> に指定した場合、そのポート番号は Web サーバに適用されません。このプロパティは、Web Server の Listen ディレクティブに対応します。	ホスト名 IPv4 ドット記法 IPv6 ドット記法 (前後に[]を指定)	プロパティ削除	なし (未定義)	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.listen-add-host1=192.168.1.102
15	property.listen-add-portn	追加 Listen ディレクティブのポート番号を指定します。プロパティ名の <i>n</i> には、1～listen-add-count の指定値の値を指定します。 <i>n</i> が同じ数値のlisten-add-hostn と対で、Listen ディレクティブが指定されます。 1～listen-add-count の指定値の範囲外の値を <i>n</i> に指定した場合、そのポート番号は Web サーバに適用されません。このプロパティは、Web Server の Listen ディレクティブに対応します。	1～65535	プロパティ削除	なし (未定義)	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.listen-add-port1=82
16	property.listen-host	リクエストを受け付ける IP アドレスを指定します。	ホスト名 IPv4 ドット記法 IPv6	プロパティ削除	なし (未定義)	asadmin set hitachi-webserver-configs.hitachi-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		このプロパティは省略できます。 このプロパティは、Web Server の Listen ディレクティブに対応します。	ドット記法 (前後に[]を指定)			webserver-config.Web1-config.property.listen-host=192.168.1.101
17	property.listen-port	リクエストを受け付けるポート番号を指定します。 このプロパティは、Web Server の Listen ディレクティブに対応します。	1~65535	プロパティ削除	80	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.listen-port=81
18	property.max-request-workers	同時に接続できるクライアントの最大数を指定します。 このプロパティは、Web Server の MaxRequestWorkers ディレクティブに対応します。	1~1024	プロパティ削除	150	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.max-request-workers=300
19	property.server-name	Web Server のサーバ名およびポート番号を指定します。 このプロパティは、Web Server の ServerName ディレクティブに対応します。	Type: String	プロパティ削除	www.example.com	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.server-name=www.host1.co.jp
20	property.start-servers	Web サーバ起動時のサーバプロセス数を指定します。 このプロパティは、Web Server の StartServers ディレクティブに対応します。	1~1024	プロパティ削除	20	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.start-servers=10
21	property.threads-per-child	サーバとして起動するスレッド数を指定します。 このプロパティは、Web Server の ThreadsPerChild	1~1024	プロパティ削除	50	asadmin set hitachi-webserver-configs.hitachi-webserver-config.Web1-config.property.th

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		ディレクティブに対応します。				reads-per-child=100

注※

デフォルト値欄に「プロパティー削除」と記載されているパラメーターは、空文字を指定することでプロパティー自体が削除されます。

## サーバインスタンス関連

servers.server.Java EEサーバ名で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	hitachi-relations.hitachi-relation-type-ref.hitachi-relation.関連名.order	サーバ間関連の順序番号を指定します。	0~65535	省略不可	なし	asadmin set servers.server.JavaEE1.hitachi-relations.redirect.hitachi-relation.redirect1.order=10
2	hitachi-relations.prf-relation.hitachi-relation.関連名.property.拡張プロパティー	Java EE サーバの PRF 関連に追加する拡張プロパティーとして、任意のプロパティー名およびその値を指定します。プロパティー名はex_から始まる名称で指定します。 追加した拡張プロパティーは、サーバ間関連の関連元である Java EE サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。	Type: String	プロパティー削除	なし	asadmin set servers.server.JavaEE1.hitachi-relation.prfref1.property.ex_timeout=600
3	hitachi-session-config.server-id	Web Server で負荷分散を行う場合に、負荷分散先の Java EE サーバを特定するためのサーバ ID を設定します。 このプロパティーで指定したサーバ ID が、セッション ID	Type : String	自動生成された最大 52 桁の英数字	なし	asadmin set servers.server.JavaEE1.hitachi-session-config.server-id=server1

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>の 33 文字目以降に設定されます。</p> <p>サーバ ID を明示的に設定しない場合は、Java EE サーバが動作しているホスト名、ドメイン名、サーバインスタンス名を基に、自動生成された最大 52 桁の英数字が設定されます。</p> <p>セッション ID を一意にするために、自動生成された値を使用することを推奨します。</p>				
4	property.拡張プロパティー	<p>Java EE サーバに追加する拡張プロパティーとして、任意のプロパティー名およびその値を指定します。プロパティー名はex_から始まる名称で指定します。</p> <p>追加した拡張プロパティーは、Java EE サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。</p>	Type: String	プロパティー削除	なし	asadmin set servers.server.JavaEE1.property.ex_timeout=600
5	property.balancer-member-root-id	<p>Web サーバからクラスターを介してこのサーバにリダイレクトする場合に使用する cookie-id を指定します。</p>	Type: String	プロパティー削除	なし	asadmin set servers.server.JavaEE1.property.balancer-member-root-id=route01
6	property.env_環境変数名	<p>Java EE サーバ起動時に設定する環境変数を指定します。複数の環境変数を指定する場合は、この標準プロパティーを複数指定します。</p> <p>環境変数の指定値に \${apserver.home} が</p>	Type: String	プロパティー削除	なし	asadmin set servers.server.JavaEE1.property.env_TZ=JST-9

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>含まれている場合は、<i>Application Server</i> のインストールディレクトリーに置換されます。</p> <p>なお、このプロパティ値に環境変数値を引き継ぐ指定 (%PATH%;C:¥temp や \${PATH}:/temp という指定) をしても、その環境変数の値は引き継がれません。指定する場合は、次のどちらかの場所で定義してください。</p> <ul style="list-style-type: none"> <li>ノード上 OS のシステム環境変数</li> <li>ノード上の asenv の定義</li> </ul>				

注※

デフォルト値欄に「プロパティ削除」と記載されているパラメーターは、空文字を指定することでプロパティ自体が削除されます。

configs.config.Java EEサーバの構成名で始まるパラメーター

パラメーター名がproperty. で始まるパラメーターをservers.server.Java EEサーバ名で始まるパラメーターでも定義している場合は、servers.server.Java EEサーバ名で始まるパラメーターの値が優先されます。また、構成名がクラスターで使用される場合は、クラスターに属するすべてのJava EEサーバに対して適用されます。

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	admin-service.jmx-connector.system.port	JMX のネットワークポートを設定します。UNIX 上でポート番号 1~1024 をリッスンするソケットを作成するには、スーパーユーザー権限が必要です。	1~65535	なし	\${JMX_SYSTEM_CONNECTOR_PORT}	asadmin set configs.config.JavaEE1-config.admin-service.jmx-connector.system.port=8686
2	availability-service.availability-enabled	特定のクラスター、またはasadmin set サブコマンドを使用する特定の Web、EJB、JMS コンテ	true false	true	なし	asadmin set configs.config.Cluster1-config.availability-



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		ナーの可用性を有効にするかどうかを設定します。				<code>service.availability-enabled=true</code>
3	<code>connector-service.shutdown-timeout-in-seconds</code>	Application Server のシャットダウン時に、すべてのリソースアダプターが停止されます。シャットダウンはリソースを集中的に使用することが多い動作のため、シャットダウン中にリソースアダプターがハングアップする可能性があります。 このような状況を回避するため、リソースアダプターのシャットダウンを中断するためのタイムアウト値を設定できます。デフォルトのタイムアウト値は、リソースアダプターモジュール単位で 30 秒です。	1～ 2147483647	30	なし	<code>asadmin set configs.config.JavaEE1-config.connector-service.shutdown-timeout-in-seconds=60</code>
4	<code>ejb-container.max-cache-size</code>	キャッシュできる最大インスタンス数を指定します。	0～ 2147483647	512	2000	<code>asadmin set configs.config.JavaEE1-config.ejb-container.max-cache-size=1000</code>
5	<code>ejb-container.max-pool-size</code>	プールできる最大サイズを指定します。値が 0 の場合は、プールのサイズに制限がないことを示します。サイズが無制限のプールは、 <code>pool-resize-quantity</code> の定義に従って、 <code>steady-pool-size</code> へ収縮していきます。	0～ 2147483647	32	24	<code>asadmin set configs.config.JavaEE1-config.ejb-container.max-pool-size=30</code>
6	<code>ejb-container.steady-pool-size</code>	通常プールに保持される bean インスタンスの数を指定します。	0～ 2147483647	0	24	<code>asadmin set configs.config.JavaEE1-config.ejb-</code>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
						container.steady-pool-size=20
7	group-management-service.group-discovery-timeout-in-millis	グループのほかのメンバーを見つけるためにインスタンスを起動しているとき、インスタンスのグループ管理サービスモジュールが待つ時間を、ミリ秒単位で指定します。 group-discovery-timeout-in-millis タイムアウトには、デフォルト値、またはそれより高い値を設定する必要があります。デフォルトは5000です。	1000～120000	5000	なし	asadmin set configs.config.Cluster1-config.group-management-service.group-discovery-timeout-in-millis=8000
8	hitachi-eheap.httpsession-enabled	HTTPセッションに対して明示管理ヒープ機能を適用するかどうかを設定します。	true false	true	true	asadmin set configs.config.JavaEE1-config.hitachi-eheap.httpsession-enabled=false
9	hitachi-jaxrs-config.connect-timeout	jersey が提供する Client configuration properties の jersey.config.client.connectTimeout の初期値を、ミリ秒単位で設定します。	整数値 (Type : Integer)	なし	なし	asadmin set configs.config.JavaEE1-config.hitachi-jaxrs-config.connect-timeout=60000
10	hitachi-jaxrs-config.read-timeout	jersey が提供する Client configuration properties の jersey.config.client.readTimeout の初期値を、ミリ秒単位で設定します。	整数値 (Type : Integer)	なし	なし	asadmin set configs.config.JavaEE1-config.hitachi-jaxrs-config.read-timeout=300000
11	hitachi-jaxws-config.connect-timeout	JAX-WS 仕様での BindingProvider properties の、JAX-WS RI が提供	整数値 (Type : Integer)	なし	なし	asadmin set configs.config.JavaEE1-config.hitachi-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		するプロパティ <code>com.sun.xml.ws.connect.timeout</code> の初期値を、ミリ秒単位で設定します。				<code>jaxws-config.connect-timeout=60000</code>
12	<code>hitachi-jaxws-config.request-timeout</code>	JAX-WS 仕様での <code>BindingProvider</code> <code>properties</code> の、JAX-WS RI が提供するプロパティ <code>com.sun.xml.ws.request.timeout</code> の初期値を、ミリ秒単位で設定します。	整数値 (Type : Integer)	なし	なし	<code>asadmin set configs.config.JavaEE1-config.hitachi-jaxws-config.request-timeout=300000</code>
13	<code>hitachi-jaxws-config.security-auth-password</code>	JAX-WS 仕様での <code>BindingProvider</code> <code>properties</code> の <code>javax.xml.ws.security.auth.password</code> の初期値を設定します。	Type: String	なし	なし	<code>asadmin set configs.config.JavaEE1-config.hitachi-jaxws-config.security-auth-password=pass</code>
14	<code>hitachi-jaxws-config.security-auth-username</code>	JAX-WS 仕様での <code>BindingProvider</code> <code>properties</code> の <code>javax.xml.ws.security.auth.username</code> の初期値を設定します。	Type: String	なし	なし	<code>asadmin set configs.config.JavaEE1-config.hitachi-jaxws-config.security-auth-username=user</code>
15	<code>hitachi-jaxws-config.session-maintain</code>	JAX-WS 仕様での <code>BindingProvider</code> <code>properties</code> の <code>javax.xml.ws.session.maintain</code> の初期値を設定します。	true false	なし	なし	<code>asadmin set configs.config.JavaEE1-config.hitachi-jaxws-config.session-maintain=true</code>
16	<code>hitachi-jca.connection-auto-close-enabled</code>	コネクション自動クローズの有効または無効を設定します。サーブレットおよび JSP でユーザーが生成したスレッドで取得したコネクションは、自動クローズされません。	true false	なし	true	<code>asadmin set configs.config.JavaEE1-config.hitachi-jca.connection-auto-close-enabled=true</code>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		サーブレットおよび JSP の各種コールバックメソッド (init メソッドなど)、ならびに EJB の各種コールバックメソッド (ejbCreate メソッド、PostConstruct メソッドなど) で取得したコネクションが明示的にクローズされていない場合、自動クローズされることがあります。				
17	hitachi-manage-info.running-watch-interval-in-seconds	Java EE サーバの動作確認間隔を、秒単位で指定します。	1~86400	省略不可	10	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.running-watch-interval-in-seconds=20
18	hitachi-manage-info.start-timeout-in-seconds	Java EE サーバの起動監視時間を、秒単位で指定します。0 を指定した場合は、監視しません。	0~3600	省略不可	60	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.start-timeout-in-seconds=180
19	hitachi-manage-info.starting-watch-interval-in-seconds	Java EE サーバの起動時監視間隔を、秒単位で指定します。	1~86400	省略不可	1	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.starting-watch-interval-in-seconds=3
20	hitachi-manage-info.starting-watch-start-time-in-seconds	Java EE サーバの起動で、起動コマンドを実行してから動作確認を開始するまでの時間を、秒単位で指定します。	0~86400	省略不可	0	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.starting-watch-start-time-in-seconds=60

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
21	hitachi-manage-info.stop-timeout-in-seconds	Java EE サーバの停止監視時間を、秒単位で指定します。0を指定した場合は、監視しません。	0~1800	省略不可	60	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.stop-timeout-in-seconds=180
22	hitachi-manage-info.stopping-watch-interval-in-seconds	Java EE サーバの停止時監視間隔を、秒単位で指定します。	1~86400	省略不可	1	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.stopping-watch-interval-in-seconds=3
23	hitachi-manage-info.template-path	Java EE サーバのサーバテンプレートのパスを絶対パスで指定します。	存在するディレクトリーパス	省略不可	ドメイン ルートディレクトリー/ ドメイン名/ server_templates/ サーバ種別	asadmin set configs.config.JavaEE1-config.hitachi-manage-info.template-path=C:\%temp%\javaee
24	hitachi-monitoring.hitachi-statistics-files.base-time	稼働情報ファイルを切り替える際の基点時刻を、ローカルタイムの1970年1月1日0時0分0秒からの増分として分単位で指定します。 h時m分を基点にする場合には、h×60+mを指定します。	0~1439の整数 (0~23時間59分)	0	0	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.base-time=360
25	hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.jvm-memory-extensions	Java EE Serverの稼働情報収集機能が収集するJava VMメモリー拡張稼働情報をファイルに蓄積するかどうかを指定します。	true false	true	true	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.jvm-memory-extensions=true

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
26	hitachi-monitoring.hitachi-statistics-files.interval	稼働情報ファイルの蓄積間隔を、秒単位で指定します。	1～86400の整数（1秒～1日）	60	60	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.interval=300
27	hitachi-monitoring.hitachi-statistics-files.output-dir	稼働情報ファイルの出力先を変更したいときに出力先ディレクトリーを絶対パスで指定します。 複数のサーバインスタンスで同じ出力先ディレクトリーを指定すると複数のサーバインスタンスが稼働情報を同じファイルに出力する可能性があるため、サーバインスタンスごとに異なる出力先ディレクトリーを指定してください。	半角英数字、-（ハイフン）、_（アンダースコア）、:（コロン）、¥（円記号）、/（スラッシュ）	空文字	空文字	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.output-dir=D:/monitoring_dir
28	hitachi-monitoring.hitachi-statistics-files.num	稼働情報ファイルのファイル面数を指定します。	2～168の整数	8	8	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.num=14
29	hitachi-monitoring.hitachi-statistics-files.period	稼働情報ファイルの切り替え時間を、時間単位で指定します。	1～744の整数（1時間～31日）	24（1日）	24（1日）	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.period=1
30	http-service.virtual-server.id.property.errorReportValve.value	この仮想サーバ上のアプリケーションのデフォルトエラーページを生成するカスタムバルブの完全修飾クラス名を指定します。	次のどちらかを指定します。 • org.apache.catalina.valves.Error	org.apache.catalina.valves.ErrorReportValve	空文字	<ul style="list-style-type: none"> <li>仮想サーバのデフォルトエラーページの方式を有効にする場合 asadmin set configs.config.JavaEE1-config.http-</li> </ul>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>この仮想サーバのデフォルトエラーページの方式を有効にするには、 org.apache.catalina.valves.ErrorReportValve を指定してください。</p> <p>この仮想サーバのデフォルトエラーページの方式を無効にするには、空文字を指定してください。</p>	<p>ReportValve</p> <ul style="list-style-type: none"> <li>空文字</li> </ul>			<pre>service.virtual- server.server.p roperty.errorRe portValve.value =org.apache.cat alina.valves.Er rorReportValve</pre> <ul style="list-style-type: none"> <li>仮想サーバのデフォルトエラーページの方式を無効にする場合</li> </ul> <pre>asadmin set configs.config. JavaEE1- config.http- service.virtual- server.server.p roperty.errorRe portValve.value =""</pre>
31	iiop-service.iiop-listener.id.port	<p>リスナーのポート番号を指定します。</p> <p>UNIX 上でポート番号 1~1024 をリスンするソケットを作成するには、スーパーユーザー権限が必要です。</p>	1~65535	1072	<p>3700 &lt;Java EEサーバの構成名 ="server-config", id="orb-listener-1" の場合&gt;</p> <pre>\$ {IIOP_LISTENER_PORT}&lt;サーバインスタンスの場合&gt;</pre>	<pre>asadmin set configs.config.Jav aEE1-config.iiop- service.iiop- listener.orb- listener-1.port=99 99</pre>
32	java-config.native-library-path-prefix	<p>ネイティブライブラリーへのパスのプレフィックスを設定します。</p> <p>パス区切り文字は、":"です。</p> <p>このパラメーターを変更したあとに、サーバを再起動する必要があります。</p>	Type: String	空文字	なし	<pre>asadmin set configs.config.Jav aEE1-config.java- config.native- library-path- prefix=D:¥usr ¥nativelib</pre>

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
33	jms-service.jms-host. <i>JMS host name</i> .port	JMS プロバイダーのネットワークポートを指定します。 UNIX 上でポート番号 1~1024 をリスンするソケットを作成するには、スーパーユーザー権限が必要です。	1~65535	なし	\$ {JMS_PROVIDER_PORT}	asadmin set configs.config.JavaEE1-config.jms-service.jms-host.default_JMS_host.port=7677
34	jms-service.reconnect-interval-in-seconds	再接続を試行する間隔を、秒単位で指定します。 この間隔は、AddressList の各 JMS ホストへの試行、および AddressList 中の連続するアドレスに対して適用されます。この間隔が短過ぎると、JMS ホストが回復する時間がなくなります。また、長過ぎると、再コネクが遅くなります。	1~2147483647	5	なし	asadmin set configs.config.JavaEE1-config.jms-service.reconnect-interval-in-seconds=5
35	mdb-container.max-pool-size	クライアントのリクエストに対応するために作成できるコネクションの最大数を指定します。	0~2147483647	32	なし	asadmin set configs.config.JavaEE1-config.mdb-container.max-pool-size=20
36	monitoring-service.module-monitoring-levels.hitachi-jvm-memory-extensions	Java VM メモリ拡張稼働情報の監視レベルを指定します。	LOW HIGH OFF	OFF	HIGH	asadmin set configs.config.JavaEE1-config.monitoring-service.module-monitoring-levels.hitachi-jvm-memory-extensions=OFF
37	monitoring-service.module-monitoring-levels.http-service	HTTP サービスの監視レベルを指定します。 監視レベルに HIGH または LOW を設定すると、HTTP サービスの監視が有効に	OFF LOW HIGH	OFF	HIGH	asadmin set configs.config.JavaEE1-config.monitoring-service.module-monitoring-



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>なります。監視レベルに OFF を設定すると、HTTP サービスの監視が無効になります。</p> <ul style="list-style-type: none"> <li>• OFF：監視は行いません。パフォーマンスへの影響はありません。</li> <li>• LOW：作成数やバイト数などの簡易的な統計を行います。</li> <li>• HIGH：簡易的な統計に加えて、メソッド数や期間などのメソッド統計を行います</li> </ul>				levels.http-service=HIGH
38	monitoring-service.module-monitoring-levels.jdbc-connection-pool	<p>JDBC コネクションプールの監視レベルを指定します。</p> <p>監視レベルに HIGH または LOW を設定すると、JDBC コネクションプールの監視が有効になります。監視レベルに OFF を設定すると、JDBC コネクションプールの監視が無効になります。</p> <ul style="list-style-type: none"> <li>• OFF：監視は行いません。パフォーマンスへの影響はありません。</li> <li>• LOW：作成数やバイト数などの簡易的な統計を行います。</li> <li>• HIGH：簡易的な統計に加えて、メソッド数や期間などのメソッド統計を行います。</li> </ul>	OFF LOW HIGH	OFF	HIGH	asadmin set configs.config.JavaEE1-config.monitoring-service.module-monitoring-levels.jdbc-connection-pool=HIGH

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
39	monitoring-service.module-monitoring-levels.web-container	<p>Web コンテナの監視レベルを指定します。</p> <p>監視レベルに HIGH または LOW を設定すると、Web コンテナの監視が有効になります。監視レベルに OFF を設定すると、Web コンテナの監視が無効になります。</p> <ul style="list-style-type: none"> <li>• OFF：監視は行いません。パフォーマンスへの影響はありません。</li> <li>• LOW：作成数やバイト数などの簡易的な統計を行います。</li> <li>• HIGH：簡易的な統計に加えて、メソッド数や期間などのメソッド統計を行います。</li> </ul>	OFF LOW HIGH	OFF	HIGH	asadmin set configs.config.JavaEE1-config.monitoring-service.module-monitoring-levels.web-container=HIGH
40	network-config.network-listeners.network-listener.name.enabled	リスナーをアクティブにするかどうかを指定します。	true false	true	なし	asadmin set configs.config.JavaEE1-config.network-config.network-listeners.network-listener.http-listener-1.enabled=false
41	network-config.network-listeners.network-listener.name.port	<p>ネットワークリスナーのネットワークポートを指定します。</p> <p>UNIX 上でポート番号 1~1024 をリスンするソケットを作成するには、スーパーユーザー権限が必要です。</p>	1~65535	なし	<p>8080&lt;ドメイン管理サーバの http-listener-1 の場合&gt;</p> <p>8181&lt;ドメイン管理サーバの http-listener-2 の場合&gt;</p>	asadmin set configs.config.JavaEE1-config.network-config.network-listeners.network-listener.http-listener-1.port=8080

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
					\$ {HTTP_LISTENER_PORT}<サーバインスタンスのhttp-listener-1の場合> \$ {HTTP_SSL_LISTENER_PORT}<サーバインスタンスのhttp-listener-2の場合>	
42	network-config.protocols.protocol.admin-listener.http.request-timeout-seconds	<p>リクエストがタイムアウトするまでの時間を、秒単位で指定します。</p> <p>リクエストの処理時間（リクエストの処理の開始から完了まで）が、このパラメーターに設定した時間を越えた場合、リクエストタイムアウトが発生します。</p> <p>0を指定した場合は、タイムアウトしません。</p> <p>タイムアウト値に0（タイムアウト無効）を指定することは推奨しません。0を指定した場合、admin-listenerのリクエストの処理時間が長期化すると、そのコマンド処理がasadminサブコマンドおよびユーザーリクエストを処理するスレッドを専有してしまうおそれがあります。</p>	0～ 2147483647	900	900	asadmin set configs.config.JavaEE1-config.network-config.protocols.protocol.admin-listener.http.request-timeout-seconds=600

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
43	network-config.protocols.protocol.http-listener-1.http.max-connections	サーバによってコネクションが閉じられるまでにパイプライン化できる HTTP リクエストの最大数を指定します。 このプロパティに 1 を設定すると、HTTP/1.0 keep-alive、HTTP/1.1 keep-alive、およびパイプライン化が無効になります。	1～ 2147483647	256	なし	asadmin set configs.config.JavaEE1-config.network-config.protocols.protocol.http-listener-1.http.max-connections=120
44	network-config.protocols.protocol.http-listener-1.http.request-timeout-seconds	リクエストがタイムアウトするまでの時間を、秒単位で指定します。 リクエストの処理時間（リクエストの処理の開始から完了まで）が、このパラメーターに設定した時間を越えた場合、リクエストタイムアウトが発生します。 0 を指定した場合は、タイムアウトしません。単位は秒です。 WebSocket を使用する場合も、このタイムアウトは有効です。WebSocket の場合は、受信した WebSocket フレームに関連する ServerEndpoint の処理時間に対してのタイムアウトに使用されます。	0～ 2147483647	900	190	asadmin set configs.config.JavaEE1-config.network-config.protocols.protocol.http-listener-1.http.request-timeout-seconds=600
45	network-config.protocols.protocol.http-listener-1.http.timeout-seconds	keep-alive 状態およびアイドル状態であるコネクションが、タイムアウトするまでの時間を、秒単位で指定します。 0 未満の値を指定した場合は、タイムア	-1～ 2147483647	30	-1	asadmin set configs.config.JavaEE1-config.network-config.protocols.protocol.http-listener-1.http.timeout-seconds=20

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		ウトしません。単位は秒です。				
46	network-config.protocols.protocol.sec-admin-listener.http.request-timeout-seconds	<p>リクエストがタイムアウトするまでの時間を、秒単位で指定します。</p> <p>リクエストの処理時間（リクエストの処理の開始から完了まで）が、このパラメーターに設定した時間を越えた場合、リクエストタイムアウトが発生します。</p> <p>0を指定した場合は、タイムアウトしません。</p> <p>タイムアウト値に0（タイムアウト無効）を指定することは推奨しません。0を指定した場合、admin-listenerのリクエストの処理時間が長期化すると、そのコマンド処理がasadminサブコマンドおよびユーザーリクエストを処理するスレッドを専有してしまうおそれがあります。</p>	0～ 2147483647	900	900	asadmin set configs.config.JavaEE1-config.network-config.protocols.protocol.sec-admin-listener.http.request-timeout-seconds=600
47	property.拡張プロパティ	<p>Java EE サーバに追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。</p> <p>追加した拡張プロパティは、Java EE サーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。</p>	Type: String	プロパティ削除	なし	asadmin set configs.config.JavaEE1-config.property.ex_timeout=600

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
48	property.balancer-member-root-id	Web サーバからクラスターを介してこのサーバにリダイレクトする場合に使用する cookie-id を指定します。	Type: String	プロパティ削除	なし	asadmin set configs.config.JavaEE1- config.property.balancer-member- root-id=route01
49	property.env_環境変数名	Java EE サーバ起動時に設定する環境変数を指定します。複数の環境変数を指定する場合は、この標準プロパティを複数指定します。 環境変数の指定値に <code>\${apserver.home}</code> が含まれている場合は、 <i>Application Server</i> のインストールディレクトリに置換されます。 なお、このプロパティ値に環境変数値を引き継ぐ指定 ( <code>%PATH%;C:¥temp</code> や <code>{PATH}:/temp</code> という指定) をしても、その環境変数の値は引き継がれません。指定する場合は、次のどちらかの場所で定義してください。 <ul style="list-style-type: none"> <li>ノード上 OS のシステム環境変数</li> <li>ノード上の <code>asenv</code> の定義</li> </ul>	Type: String	プロパティ削除	なし	asadmin set configs.config.JavaEE1- config.property.env_TZ=JST-9
50	security-service.activate-default-principal-to-role-mapping	アプリケーション固有のマッピングが定義されていないロールマッピングに対して、デフォルトプリンシパルを適用します。各ロールは、 <code>mapped-principal-class</code> で定義された <code>java.security.Principal</code> 実装クラスのインスタンスへマッピ	true false	false	なし	asadmin set configs.config.JavaEE1- config.security-service.activate- default-principal- to-role- mapping=true

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		ングされます。このクラスは、ロールと同じ名称を持っています。				
51	security-service.default-principal	プリンシパルが提供されていない場合に、必要に応じてデフォルトセキュリティーコンテキストの識別に使用します。この属性は、通常のサーバ運用では設定する必要はありません。	Type: String	なし	なし	asadmin set configs.config.JavaEE1-config.security-service.default-principal=dsmith
52	security-service.default-principal-password	デフォルトプリンシパルのパスワードを指定します。この属性は、通常のサーバ運用では設定する必要はありません。	Type: String	なし	なし	asadmin set configs.config.JavaEE1-config.security-service.default-principal-password=secret
53	security-service.mapped-principal-class	activate-default-principal-to-role-mapping に true が設定された場合に使用される、java.security.Principal インプリメンテーションクラスをカスタマイズします。	Type: String	なし	なし	asadmin set configs.config.JavaEE1-config.security-service.mapped-principal-class=CustomPrincipalImplClass
54	thread-pools.thread-pool.name.idle-thread-timeout-seconds	プールからアイドルスレッドが削除されるまでの時間を、秒単位で指定します。	1～2147483647	900	なし	asadmin set configs.config.JavaEE1-config.thread-pools.thread-pool-1.idle-thread-timeout-seconds=1000
55	thread-pools.thread-pool.name.max-queue-size	スレッドプールのキューに格納できる最大タスク数を指定します。 -1 の場合、最大数のチェックは実施されません。	-1 または 1～2147483647	4096	80 <http-thread-pool の場合>	asadmin set configs.config.JavaEE1-config.thread-pools.thread-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
						pool-1.max-queue-size=1000
56	thread-pools.thread-pool.name.max-thread-pool-size	プールに含めることができるスレッドの最大数を指定します。iiop-listener で使用するためにこのスレッドプールを最適化するには、この値を200 に設定します。	1～ 2147483647	5	200 <thread-pool-1 の場合> 24 <http-thread-pool の場合>	asadmin set configs.config.JavaEE1-config.thread-pools.thread-pool.http-thread-pool.max-thread-pool-size=8
57	thread-pools.thread-pool.name.min-thread-pool-size	リクエストを処理する最小のスレッド数を指定します。このスレッドプールがインスタンス化されると、これらのスレッドは事前に作成されます。	1～ 2147483647	2	24 <http-thread-pool の場合>	asadmin set configs.config.JavaEE1-config.thread-pools.thread-pool-1.min-thread-pool-size=20
58	transaction-service.automatic-recovery	サービスの起動時に保留中のトランザクションを完了させるかどうかを指定します。	true false	false	true	asadmin set configs.config.JavaEE1-config.transaction-service.automatic-recovery=false
59	transaction-service.heuristic-decision	ほかのトランザクション参加者が到達できないために分散トランザクションの結果を決定できない場合、このプロパティが結果を決定します。	commit rollback	rollback	なし	asadmin set configs.config.JavaEE1-config.transaction-service.heuristic-decision=rollback
60	transaction-service.keypoint-interval	ログ内のキーポイント処理間のトランザクション数を指定します。キーポイント処理はトランザクションのステータスファイルを圧縮することでサイズ容量を減少させます。 この属性に対して大きな値を指定すると、	0～ 2147483647	65536	なし	asadmin set configs.config.JavaEE1-config.transaction-service.keypoint-interval=2048



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		より大きなトランザクションのステータスファイルになって、キーポイント処理が少なく、潜在的に優れた性能になります。小さな値を指定すると、より小さなログファイルになって、キーポイント処理の頻度が高いため若干性能が低下します。				
61	transaction-service.property.d b-logging-resource	トランザクションサービスを構築するために使用します。このプロパティーにはデフォルト値がありません。また、デフォルトでは設定されていません。	Type: String	なし	なし	asadmin set configs.config.JavaEE1- config.transaction- service.property.d b-logging- resource="jdbc/ TxnDS"
62	transaction-service.property.o racle-xa-recovery- workaround	XAResource.commitメソッドに問題がある場合にサーバの回避方法を無効にするには、このプロパティー値にfalseを設定します。	true false	true	なし	asadmin set configs.config.JavaEE1- config.transaction- service.property.o racle-xa-recovery- workaround=true
63	transaction-service.property.p ending-txn- cleanup-interval	トランザクションサービスを構築するために使用します。このプロパティーに-1を指定した場合は、定期回復スレッドが実行されません。単位は秒です。定期自動回復は、このプロパティーに正数の値が設定されている場合に、スレッドのバックグラウンドで実行されます。	-1または1~ 2147483647	なし（プロパティーがあって値が空の場合は60）	なし	asadmin set configs.config.JavaEE1- config.transaction- service.property.p ending-txn- cleanup- interval=-1
64	transaction-service.property.u	トランザクションサービスを構築するために使用します。	true false	true	なし	asadmin set configs.config.JavaEE1-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
	se-last-agent-optimization	このプロパティはデフォルトでtrueが設定されています。 trueの場合、トランザクション処理能力を向上させる最終エージェント最適化を有効にします。同じトランザクション内で非XAリソースとXAリソースが使用された場合、非XAリソースが最後のエージェントになります。				config.transaction-service.property.use-last-agent-optimization=true
65	transaction-service.property.wait-time-before-recovery-insec	トランザクションサービスを構築するために使用します。	1～2147483647	60	なし	asadmin set configs.config.JavaEE1-config.transaction-service.property.wait-time-before-recovery-insec=60
66	transaction-service.property.xaresource-txn-timeout	トランザクションサービスを構築するために使用します。 このプロパティに0を指定した場合は、タイムアウトしません。単位は秒です。 XAResource タイムアウトを変更します。 場合によっては、XAResource のデフォルトのタイムアウトはトランザクションが中止される可能性があるため、値の変更を推奨します。	0～2147483647	120 (XAResource に依存します)	180	asadmin set configs.config.JavaEE1-config.transaction-service.property.xaresource-txn-timeout=0
67	transaction-service.retry-timeout-in-seconds	次の場合でのリトライ時間を決定します。 • トランザクションリカバリータイムでリソースへの到達が不可能な場合	-1～2147483647	600	なし	asadmin set configs.config.JavaEE1-config.transaction-service.retry-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<ul style="list-style-type: none"> <li>2相コミットプロトコルの第2相に一時的な例外がある場合</li> </ul> <p>-1を指定した場合は、無限にリトライします。</p> <p>0を指定した場合は、リトライしません。</p> <p>正の値は、リトライが試みられるまでの時間を意味します。</p>				timeout-in-seconds=600
68	transaction-service.timeout-in-seconds	トランザクションが中止されるまでの時間を指定します。0を指定した場合、トランザクションはタイムアウトしません。	0～ 2147483647	0	180	asadmin set configs.config.JavaEE1-config.transaction-service.timeout-in-seconds=0
69	web-container.session-config.session-manager.manager-properties.max-sessions	キャッシュに格納できるセッションの最大数を指定します。制限を設けない場合は、-1を指定します。	-1または1～ 2147483647	-1	なし	asadmin set configs.config.JavaEE1-config.web-container.session-config.session-manager.manager-properties.max-sessions=10
70	web-container.session-config.session-properties.timeout-in-seconds	セッションがタイムアウトするまでの時間を、秒単位で指定します。	0～ 2147483647	1800	7200	asadmin set configs.config.JavaEE1-config.web-container.session-properties.timeout-in-seconds=5000
71	hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.network-connection-queue	Java EE Serverの稼働情報収集機能が収集するネットワークコネクションキュー稼働情報をファイルに蓄積するかどうかを指定します。	true false	true	true	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.hitachi-output-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
						enabled.network-connection-queue=true
72	hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.network-thread-pool	Java EE Server の稼働情報収集機能が収集するネットワークスレッドプール稼働情報をファイルに蓄積するかどうかを指定します。	true false	true	true	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.network-thread-pool=true
73	hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.jdbc-connection-pool	Java EE Server の稼働情報収集機能が収集するリソース稼働情報（接続プール）のうち、JDBC 接続プールの稼働情報をファイルに蓄積するかどうかを指定します。	true false	true	true	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.jdbc-connection-pool=true
74	hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.web-session	Java EE Server の稼働情報収集機能が収集する Web セッション稼働情報をファイルに蓄積するかどうかを指定します。	true false	true	true	asadmin set configs.config.JavaEE1-config.hitachi-monitoring.hitachi-statistics-files.hitachi-output-enabled.web-session=true

注※

デフォルト値欄に「プロパティ削除」と記載されているパラメーターは、空文字を指定することでプロパティ自体が削除されます。

clusters.cluster. クラスター名で始まるパラメーター

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
1	config-ref	サービンスタンスが使用する config の名称を参照します。	Type: String	なし	なし	asadmin get clusters.cluster.Cluster1.config-ref

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
2	gms-bind-interface-address	グループ管理サービスがバインドするネットワークインターフェースの IP アドレスです。	Type: String	なし	なし	asadmin set clusters.cluster.Cluster1.gms-bind-interface-address=%GMS-BIND-INTERFACE-ADDRESS-mycluster%

## 運用管理関連

### 運用管理関連

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	hitachi-domain.hitachi-snapshot.max-num	システム情報アーカイブファイルの出力先ディレクトリーに生成できるファイルの数を指定します。ドメインや複数のノードのアーカイブファイル出力先に同じディレクトリーを指定する場合は、このパラメーターに同じ数値を指定してください。 指定したファイル数を超える場合は、古いファイルから削除されます。 このパラメーターの設定値は、ドメイン全体に適用されます。設定値を反映するため、設定後には、ドメイン内のすべての Web サーバ、Java EE サーバ、および PRF デーモンを再起動してください。	5～ 2147483647	10	10	asadmin set hitachi-domain.hitachi-snapshot.max-num=20
2	hitachi-domain.hitachi-snapshot.output-dir	システム情報アーカイブファイルの出力先ディレクトリーを、絶対パスで指定します。出力先ディレク	半角英数字、- (ハイフン)、_ (アンダースコア)、:	<i>Application Server</i> インストールディレクトリー/	<i>Application Server</i> インストールディレクトリー/	asadmin set hitachi-domain.hitachi-snapshot.output-dir=/logs/snapshot

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		トリーには、ローカルドライブ上の作成済みディレクトリだけを指定できます。 このパラメーターの設定値は、ドメイン全体に適用されます。設定値を反映するため、設定後には、ドメイン内のすべての Web サーバ、Java EE サーバ、および PRF デーモンを再起動してください。	(コロン)、¥ (円記号)、/ (スラッシュ)	javaee/snapshot	javaee/snapshot	
3	hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.イベント名.enabled	このイベント検知時コマンドを実行するかどうかを指定します。	true false	省略不可	true	asadmin set hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.Event1.enabled=false
4	hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.イベント名.monitoring-max-count	hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.イベント名.monitoring-time-span-in-seconds で設定した時間内に、同じ種類のイベントに対するイベント検知時コマンドの実行を、最大何回許すかを指定します。 0 を指定した場合は、常にイベント検知時コマンドの実行を許可します。 1 以上を指定した場合は、hitachi-hook-policy.hitachi-event-	0~256	省略不可	0	asadmin set hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.Event1.monitoring-max-count=5

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		hooks.hitachi-event-hook. イベント名.monitoring-time-span-in-seconds で設定した時間内に指定回数を超えるイベントが発生すると、指定回数を超えた分のイベントに対するイベント検知時コマンドは実行されません。				
5	hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook. イベント名.monitoring-time-span-in-seconds	<p>最大許容実行回数*を設定する時間間隔を、秒単位で指定します。</p> <p>0 を指定した場合は、監視しません。</p> <p>1 以上を指定した場合は、指定時間の間監視します。</p> <p>注※</p> <p>最大許容実行回数は、hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook. イベント名.monitoring-max-count で設定します。</p>	0～ 2147483647	省略不可	0	asadmin set hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.Event1.monitoring-time-span-in-seconds=60
6	hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook. イベント名.path	<p>イベント検知時コマンドを絶対パスで指定します。ファイルの区切りには/を使用します。</p> <p>イベント検知時コマンドをシェルスクリプトで作成した場合、コマンド名の前に/bin/sh を追加して指定してください。/bin/sh が指定されていない場合、コマンドが正常に動作しないことがあります。</p>	Type: String	省略不可	<p><i>Application Server</i>インストールディレクトリ/ glassfish/config/manager/snapshot_event-hook</p> <p>このスクリプトを正常に稼働させるためには、スクリプトファイルの</p>	asadmin set hitachi-manage-configs.hitachi-hook-policy.hitachi-event-hooks.hitachi-event-hook.Event1.path=/temp/event-hook.sh

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>指定例</p> <pre>/bin/sh /home/ user1/ javaeeabnormal.e nd.sh</pre>			<p>次の変数を設定する必要があります。</p> <ul style="list-style-type: none"> <li>• ADMIN_HOST</li> <li>• UID</li> <li>• PWDFILE</li> </ul>	
7	hitachi-relation-types.hitachi-relation-type.prf-relation.property.拡張プロパティ	<p>PRF 関連に追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。</p> <p>追加した拡張プロパティは、サーバ間関連の関連元サーバを設定するサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。</p>	Type: String	プロパティ削除	なし	asadmin set hitachi-relation-types.hitachi-relation-type.prf-relation.property.ex_timeout=600
8	hitachi-relation-types.hitachi-relation-type.redirect.property.拡張プロパティ	<p>リダイレクト関連に追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。</p> <p>追加した拡張プロパティは、サーバ間関連の関連元である Web サーバを設定するサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。</p>	Type: String	プロパティ削除	なし	asadmin set hitachi-relation-types.hitachi-relation-type.redirect.property.ex_timeout=600
9	hitachi-relation-types.hitachi-relation-type.redirect.property.negative	<p>リダイレクターの動作の定義で、マッピング定義に対するリクエストを除外する</p>	true false	プロパティ削除	false	asadmin set hitachi-relation-types.hitachi-relation-



項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		かどうかを指定します。 このプロパティーは、Web Server の ProxyPass ディレクティブの!オプションに対応します。				type.redirect.property.negative=true
10	hitachi-relation-types.hitachi-relation-type.redirect.property.network-listener	リダイレクト先となるネットワークリスナー名を指定します。 このプロパティーは、Web Server の ProxyPass ディレクティブのURL に対応します。URL には指定したネットワークリスナー名から特定されたホスト名とポート番号が設定されます。	ネットワークリスナー名の文字列	プロパティー削除	http-listener-1	asadmin set hitachi-relation-types.hitachi-relation-type.redirect.property.network-listener=http-listener-1
11	hitachi-relation-types.hitachi-relation-type.redirect.property.path	リダイレクターの動作の定義で、マッピング定義のローカルの仮想パスの名前を指定します。 このプロパティーは、Web Server の ProxyPass ディレクティブのパス名指定に対応します。	Type: String (httpsd.conf ファイルの ProxyPass に定義するパス名 (ローカルの仮想パスの名前) の形式)	プロパティー削除	/	asadmin set hitachi-relation-types.hitachi-relation-type.redirect.property.path=/news/
12	hitachi-relation-types.hitachi-relation-type.redirect.property.proxy-pass-option	リダイレクターの動作の定義で、マッピング定義のコネクションプーリングの調整パラメーターを指定します。 このプロパティーは、Web Server の ProxyPass ディレクティブの「キー=値」指定に対応します。	httpsd.conf ファイルの ProxyPass に定義するコネクションプーリングの調整パラメーターの形式 ([key=value key=value ...])	プロパティー削除	空文字	asadmin set hitachi-relation-types.hitachi-relation-type.redirect.property.proxy-pass-option=timeout=300
13	hitachi-relation-types.hitachi-relation-	リダイレクターの動作の定義で、Set-Cookie ヘッダーの	true false	プロパティー削除	true	asadmin set hitachi-relation-

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
	type.redirect.property.set-proxy-pass-reverse-cookie	再割り当てをするかどうかを指定します。 このプロパティーは、Web Server の HWSProxyPassReverseCookie ディレクティブに対応します。				relation-type.redirect.property.set-proxy-pass-reverse-cookie=true
14	hitachi-relation-types.hitachi-relation-type.redirect.property.stickysession	リダイレクターの動作の定義で、Sticky を有効にするかどうかを指定します。 このプロパティーは、Web Server の ProxyPass ディレクティブの stickysession キーに対応します。 また、このプロパティーにtrueを指定すると、必要な Header ディレクティブが追加されます。	true false	プロパティー削除	true	asadmin set hitachi-relation-types.hitachi-relation-type.redirect.property.stickysession=true

注※

デフォルト値欄に「プロパティー削除」と記載されているパラメーターは、空文字を指定することでプロパティー自体が削除されます。

ドメイン管理サーバ関連

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
1	configs.config.server-config.network-config.network-listeners.network-listener.name.port	ネットワークリスナーのネットワークポートを指定します。 UNIX 上でポート番号 1~1024 をリスンするソケットを作成するには、スーパーユーザー権限が必要です。	1~65535	なし	8080<http-listener-1の場合> 8181<http-listener-2の場合> 4848<admin-listenerの場合>	asadmin set configs.config.server-config.network-config.network-listeners.network-listener.http-listener-1.port=8080
2	configs.config.server-config.network-config.network-listeners.network-	リスナーをアクティブにするかどうかを指定します。	true false	true	なし	asadmin set configs.config.server-config.network-config.network-

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
	listener.name.enabled					listener.http-listener-1.enabled=false
3	configs.config.server-config.network-config.protocols.protocol.admin-listener.http.request-timeout-seconds	<p>リクエストがタイムアウトするまでの時間を、秒単位で指定します。</p> <p>リクエストの処理時間（リクエストの処理の開始から完了まで）が、このパラメーターに設定した時間を超えた場合、リクエストタイムアウトが発生します。</p> <p>0を指定した場合は、タイムアウトしません。</p> <p>タイムアウト値に0（タイムアウト無効）を指定することは推奨しません。0を指定した場合、admin-listenerのリクエスト処理時間が長期化すると、そのコマンド処理がasadminサブコマンドのスレッドを占有し、asadminコマンドが使用できなくなるおそれがあります。</p>	0～ 2147483647	900	900	asadmin set configs.config.server-config.network-config.protocols.protocol.admin-listener.http.request-timeout-seconds=600
4	configs.config.server-config.network-config.protocols.protocol.sec-admin-listener.http.request-timeout-seconds	<p>リクエストがタイムアウトするまでの時間を、秒単位で指定します。</p> <p>リクエストの処理時間（リクエストの処理の開始から完了まで）が、このパラメーターに設定した時間を超えた場合、リクエストタイムアウトが発生します。</p> <p>0を指定した場合は、タイムアウトしません。</p>	0～ 2147483647	900	900	asadmin set configs.config.server-config.network-config.protocols.protocol.sec-admin-listener.http.request-timeout-seconds=600

項番	パラメーター名	説明	指定できる値	デフォルト値	製品出荷時の値	指定例
		タイムアウト値に0 (タイムアウト無効) を指定することは推奨しません。0を指定した場合、sec-admin-listenerのリクエスト処理時間が長期化すると、そのコマンド処理がasadmin サブコマンドのスレッドを占有し、asadmin コマンドが使用できなくなるおそれがあります。本パラメーターはsecure admin 機能が有効な場合に使用します。				
5	configs.config.server-config.admin-service.jmx-connector.system.port	JMX のネットワークポートを設定します。UNIX 上でポート番号1~1024をリスンするソケットを作成するには、スーパーユーザー権限が必要です。	1~65535	なし	8686	asadmin set configs.config.server-config.admin-service.jmx-connector.system.port=8686

## ノード関連

### ノード関連

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
1	nodes.node.ノード名.node-host	ノードが表すホストの名前を指定します。	Type: String	なし	localhost	asadmin set nodes.nodenode1.node-host=HostA
2	nodes.node.ノード名.hitachi-node.property.拡張プロパティ	ノードに追加する拡張プロパティとして、任意のプロパティ名およびその値を指定します。プロパティ名はex_から始まる名称で指定します。 追加した拡張プロパティは、ノードに	Type: String	プロパティ削除	なし	asadmin set nodes.node.node1.hitachi-node.property.ex_timeout=600

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		存在するサーバのサーバテンプレートの各ファイルに記述する置き換え文字列として使用できます。				
3	nodes.node.ノード名.hitachi-node.hitachi-node-snapshot.output-dir	<p>各ノードのシステム情報アーカイブファイルの出力先ディレクトリーを絶対パスで指定します。ドメイン管理サーバのシステム情報の出力に関する設定は、デフォルトノード (localhost-ドメイン名) のパラメーターを指定することで変更できます。</p> <p>出力先ディレクトリーには、ローカルドライブ上の作成済みディレクトリーだけを指定できます。</p> <p>設定値が空文字の場合は、hitachi-domain.hitachi-snapshot.output-dir パラメーターの設定値が適用されます。</p> <p>空文字以外の値を設定した場合は、hitachi-domain.hitachi-snapshot.output-dir パラメーターの値より優先して適用されます。</p> <p>このパラメーターは、値を設定したノードだけに適用されます。</p>	半角英数字、- (ハイフン)、_ (アンダースコア)、: (コロン)、¥ (円記号)、/ (スラッシュ)	hitachi-domain.hitachi-snapshot.output-dir パラメーターの設定値	空文字	asadmin set nodes.node.node1.hitachi-node.hitachi-node-snapshot.output-dir=/logs/node1/snapshot
4	nodes.node.ノード名.hitachi-node.hitachi-node-snapshot.max-num	各ノードのシステム情報アーカイブファイルの出力先ディレクトリーに生成できるファイルの数を指定します。ドメイン	5~2147483647	hitachi-domain.hitachi-snapshot.max-num パラ	空文字	asadmin set nodes.node.node1.hitachi-node.hitachi-node-snapshot.max-num=20

項番	パラメーター名	説明	指定できる値	デフォルト値*	製品出荷時の値	指定例
		<p>管理サーバのシステム情報の出力に関する設定は、デフォルトノード (localhost-ドメイン名)のパラメーターを指定することで変更できます。</p> <p>ドメインや複数のノードのアーカイブファイル出力先に同じディレクトリーを指定する場合は、このパラメーターに同じ数値を指定します。</p> <p>指定したファイル数を超える場合は、古いファイルから削除されます。</p> <p>設定値が空文字の場合は、hitachi-domain.hitachi-snapshot.max-num パラメーターの設定値が適用されます。</p> <p>空文字以外の値を設定した場合は、hitachi-domain.hitachi-snapshot.max-num パラメーターの値より優先して適用されます。</p> <p>このパラメーターは、値を設定したノードだけに適用されます。</p>		メーターの設定値		

注※

デフォルト値欄に「プロパティー削除」と記載されているパラメーターは、空文字を指定することでプロパティー自体が削除されます。

## 2.12 システム管理で使用するコマンド

システム管理で使用するコマンドの書式、機能などについて説明します。

### 2.12.1 add-library

1 つ以上のライブラリーのアーカイブを Java EE Server に追加

#### 書式

```
asadmin [asadmin-options] add-library [--help]
  [--type={common|ext|app}] [--upload={false|true}]
  library-file-path [library-file-path ... ]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

add-library サブコマンドは、1 つ以上のライブラリーのアーカイブを Java EE Server に追加します。

このサブコマンドはリモートモードだけサポートします。

サーバのランタイムによってライブラリーのアーカイブが確実に取り込まれるように、ドメイン管理サーバまたはサーバインスタンスを再起動する必要があります。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### ファイル

ドメイン管理サーバまたはサーバインスタンスに追加するライブラリーのアーカイブは、add-library サブコマンドの実行時に入力データとして使用されます。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--type={common|ext|app}

ライブラリータイプと、ライブラリーのアーカイブの追加先のディレクトリーを指定します。

型：String

指定できる値を次に示します。

- **common**

Common クラスローダーディレクトリー、`domain-dir/lib` にライブラリーのアーカイブを追加します。

- **ext**

Java オプションパッケージディレクトリー、`domain-dir/lib/ext` にライブラリーのアーカイブを追加します。

- **app**

アプリケーション固有のクラスローダーディレクトリー、`domain-dir/lib/applibs` にライブラリーのアーカイブを追加します。Applib クラスローダーでロードするには、追加したライブラリーを `deploy` または `redeploy` サブコマンドの `--libraries` オプションに指定してください。

デフォルト値：`common`

`--upload={false|true}`

ライブラリーのアーカイブをドメイン管理サーバにアップロードするかどうかを指定します。ディレクトリーが指定されている場合、このオプションは無視されます。

型：`Boolean`

指定できる値を次に示します。

- **true**

ドメイン管理サーバにライブラリーのアーカイブをアップロードします。

- **false**

ドメイン管理サーバにライブラリーのアーカイブをアップロードしません。`add-library` サブコマンドは、指定のファイル名でファイルアクセスを試行します。

ドメイン管理サーバが管理ユーザーと異なるユーザーで動作していて、ファイルの読み取りアクセス権がないなど、ドメイン管理サーバがライブラリーのアーカイブにアクセスできない場合は、このサブコマンドはエラーになります。

デフォルト値：

デフォルト値は、ドメイン管理サーバがサブコマンド実行中のホスト上にあるか、リモートホスト上にあるかに依存します。

ドメイン管理サーバがサブコマンドが動作するホスト上にある場合、`false`。

ドメイン管理サーバがリモートホスト上にある場合、`true`。

### *library-file-path*

追加するライブラリーのアーカイブのパスを指定します。複数のパスを指定する場合は、各パスをスペースで区切ります。絶対パスでも相対パスでも指定できます。相対パスは、ディレクトリー `Application Server` インストールディレクトリー/`javaee/glassfish/bin` からの相対パスを指定します。

型：`String`

指定できる値を次に示します。

- スペースで区切られたライブラリーのアーカイブのパス

デフォルト値：なし



## 使用例

次の例では、mylib.jar ライブラリーのアーカイブ内のライブラリーを、デフォルトのサーバインスタンス上にある、アプリケーション固有のクラスローダーディレクトリーに追加します。

```
asadmin add-library --type app /tmp/mylib.jar
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.12.2 create-profiler

プロファイラー要素の作成

### 書式

```
asadmin [asadmin-options] create-profiler [--help]
  [--target target] [--classpath classpath]
  [--nativelibrarypath native_library_path] [--enabled={false|true}]
  profiler_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

create-profiler サブコマンドでは、プロファイラー要素を作成します。サーバインスタンスは、Java 構成のプロファイラー要素によってプロファイラーに結び付けられます。一度に存在できるのは1つのプロファイラーだけです。プロファイラーがすでにあるときに作成しようとする、エラーメッセージが表示されます。変更を有効にするには、サーバを再起動する必要があります。

このサブコマンドはリモートモードだけサポートします。

一度に存在できるのは1つのプロファイラーだけです。プロファイラーがすでにあるときに作成しようとする、エラーメッセージが表示されます。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

作成するプロファイラーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--classpath classpath`

プロファイラーに必要な Java クラスを指定します。

型：String

指定できる値を次に示します。

- プロファイラーに必要なクラスパス

デフォルト値：なし

`--nativelibrarypath native_library_path`

自動的に構築されるパスを指定します。このパスは次の要素を連結したものです。

- Java EE Server インストールディレクトリからの相対パス（ネイティブ共有ライブラリー用）
- 標準 JRE ネイティブライブラリーへのパス
- シェル環境設定（UNIX での LD\_LIBRARY\_PATH）
- プロファイル要素で指定した任意のパス

型：String

指定できる値を次に示します。

- ネイティブライブラリーへのパス

デフォルト値：なし

`--enabled={false|true}`

プロファイラーがデフォルトで有効であるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

*profiler\_name*

プロファイラーの名称を指定します。

型：String

指定できる値を次に示します。

- プロファイラー名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、プロファイラーを作成します。

```
asadmin create-profiler --target instance1 --classpath /home/appserver/  
--nativelibrarypath /u/home/lib --enabled=false sample_profiler
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

### 2.12.3 delete-profiler

プロファイラー要素の削除

## 書式

```
asadmin [asadmin-options] delete-profiler [--help]  
[--target target] profiler_name
```

## 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

## 機能

`delete-profiler` サブコマンドは、Java 構成のプロファイラー要素を削除します。

一度に存在できるのは 1 つのプロファイラーだけです。プロファイラーが存在しないときに削除しようとすると、エラーメッセージが表示されます。

このサブコマンドによる変更を有効にするには、サーバを再起動する必要があります。

このコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するプロファイラーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`profiler_name`

プロファイラー名を指定します。

型：String

指定できる値を次に示します。

- プロファイラー名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスのプロファイラーを削除します。

```
asadmin delete-profiler --target instance1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.12.4 list-commands

使用できるコマンドの一覧表示

### 書式

```
asadmin [asadmin-options] list-commands [--help]
  [--localonly={false|true}] [--remoteonly={false|true}]
  [pattern-list]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-commands サブコマンドはasadmin のサブコマンドを一覧表示します。デフォルトでは、list-commands サブコマンドはローカルサブコマンドの一覧と、それに続いてリモートサブコマンドの一覧を表示します。リモートサブコマンドまたはローカルサブコマンドだけを表示したり、指定したテキスト文字列を名称に含むサブコマンドだけを表示したりするように指定することもできます。

このサブコマンドはローカルモードとリモートモードをサポートします。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--localonly={false|true}

true を設定した場合、ローカルコマンドだけが一覧表示されます。

なお、true を設定した場合、--remoteonly オプションはfalse を設定する必要があります。それ以外の場合は、エラーが発生します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--remoteonly={false|true}`

true を設定した場合、リモートコマンドだけが一覧表示されます。

なお、true を設定した場合、`--localonly` オプションはfalse を設定する必要があります。それ以外の場合は、エラーが発生します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

### *pattern-list*

サブコマンドの一覧をフィルター処理するためのテキスト文字列の空白区切りのリストを指定します。指定したテキスト文字列のどれかが含まれるサブコマンドだけが一覧表示されます。

型：String

指定できる値を次に示します。

- サブコマンド名

デフォルト値：なし

## 使用例

次の例では、ローカルサブコマンドだけを一覧表示します。

```
asadmin list-commands --localonly=true
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.12.5 list-libraries

ライブラリーのアーカイブの一覧表示

## 書式

```
asadmin [asadmin-options] list-libraries [--help]
        [--type={common|ext|app}]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

list-libraries サブコマンドは、Java EE Server のライブラリーのアーカイブを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

--type オプションには、ライブラリータイプを指定することで、対応するディレクトリー内のライブラリーのアーカイブを一覧表示します。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--type={common|ext|app}

ライブラリータイプを指定することで、対応するディレクトリー内のライブラリーのアーカイブを一覧表示します。

型：String

指定できる値を次に示します。

- **common**  
Common クラスローダーディレクトリーdomain-dir/lib のライブラリーのアーカイブを一覧表示します。
- **ext**  
Java オプションパッケージディレクトリーdomain-dir/lib/ext のライブラリーのアーカイブを一覧表示します。
- **app**  
アプリケーション固有のクラスローダーディレクトリーdomain-dir/lib/applibs のライブラリーのアーカイブを一覧表示します。

デフォルト値：common

## 使用例

次の例では、デフォルトのサーバインスタンスにあるアプリケーション固有のクラスローダーディレクトリー内のライブラリーのアーカイブを一覧表示します。

```
asadmin list-libraries --type app
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.12.6 login

ドメインへのログイン

### 書式

```
asadmin [asadmin-options] login [--help]
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

login サブコマンドは、特定のドメインへログインできる状態にし、ドメイン管理を簡略化します。

Java EE Server ドメインが複数のマシンに作成されている場合、そのどのマシンからでも asadmin ユーティリティーコマンドを実行してほかの場所にあるドメインをリモート管理できます。

複数のドメインおよびサーバの管理クライアントとして、特定の 1 台のマシンが設定されている場合に有用です。

### ファイル

ユーザーのホームディレクトリーにある `.asadminpass` ファイル

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。



## 使用例

次の例では、別のマシンにあるドメインにログインします。login サブコマンドの前にホストおよびポートを指定しています。

```
asadmin --host foo --port 8282 login
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.12.7 remove-library

1 つ以上のライブラリー JAR ファイルの Java EE Server からの除去

### 書式

```
asadmin [asadmin-options] remove-library [--help]
  [--type={common|ext|app}]
  library-name [library-name ...]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

remove-library サブコマンドは、1 つ以上のライブラリー JAR ファイルを Java EE Server から除去します。Common クラスローダーディレクトリー、Java オプションパッケージディレクトリー、またはアプリケーション固有のクラスローダーディレクトリーからライブラリーを除去するかどうかを指定できます。

このサブコマンドはリモートモードだけサポートします。

ライブラリーのアーカイブがドメイン管理サーバから除去されます。共通ライブラリーおよび拡張ライブラリーの場合、サーバ実行時にライブラリーの除去が反映されるようにドメイン管理サーバを再起動する必要があります。ほかのサーバインスタンスからライブラリーを除去するには、そのインスタンスを再起動してドメイン管理サーバと同期させます。

### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- ライブラリーがすでに Java EE Server に追加されている必要があります。

## ファイル

- **common** タイプの場合  
Common クラスローダーディレクトリ `domain-dir/lib` からライブラリーファイルを除去します。
- **ext** タイプの場合  
Java オプションパッケージディレクトリ `domain-dir/lib/ext` からライブラリーファイルを除去します。
- **app** タイプの場合  
アプリケーション固有のクラスローダーディレクトリ `domain-dir/lib/applibs` からライブラリーファイルを除去します。  
上記のディレクトリーのうち、指定されたライブラリー JAR ファイルがドメイン管理サーバから削除されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--type={common|ext|app}`

ライブラリータイプと、ライブラリーの除去元である Java EE Server のディレクトリーを指定します。

型：String

指定できる値を次に示します。

- **common**  
Common クラスローダーディレクトリ `domain-dir/lib` からライブラリーファイルを除去します。
- **ext**  
Java オプションパッケージディレクトリ `domain-dir/lib/ext` からライブラリーファイルを除去します。
- **app**  
アプリケーション固有のクラスローダーディレクトリ `domain-dir/lib/applibs` からライブラリーファイルを除去します。

デフォルト値：common

*library\_name*

除去するライブラリーを含む JAR ファイルの名称を指定します。

型：String

指定できる値を次に示します。

- アーカイブの名称

デフォルト値：なし

## 使用例

次の例では、アーカイブファイルmylib.jar内のライブラリーを、デフォルトのサービンスタンス上にある、アプリケーション固有のクラスローダーディレクトリーから除去します。

```
asadmin remove-library --type app mylib.jar
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.12.8 version

Java EE Server のバージョン情報の表示

### 書式

```
asadmin [asadmin-options] version [--help]
        [--verbose={false|true}] [--local={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

version サブコマンドは、Java EE Server のバージョン情報を表示します。このサブコマンドがドメイン管理サーバにアクセスできない場合、デフォルトでは、ローカルでバージョンを取得し、警告メッセージを表示します。

このサブコマンドはローカルモードとリモートモードをサポートします。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

{--verbose|-v}={false|true}

true に設定した場合、サーバが稼働している Java Runtime Environment (JRE) のバージョンを表示します。

型: Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--local={false|true}`

バージョン情報をローカルで取得するかどうかを指定します。

true に設定した場合、このサブコマンドを実行するホスト上の Java EE Server をインストールした環境から、ローカルでバージョンを取得します。

false に設定した場合、このサブコマンドは、ドメイン管理サーバにアクセスしてバージョンの取得を試みます。ドメイン管理サーバへのアクセスに失敗した場合、ローカルでバージョンを取得し、警告メッセージを表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

## 使用例

次の例では、稼働しているドメイン管理サーバからバージョン情報を取得します。

```
asadmin version
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13 インターネット接続管理で使用するコマンド

インターネット接続管理で使用するコマンドの書式、機能などについて説明します。

### 2.13.1 create-protocol

新規プロトコルの追加

#### 書式

```
asadmin [asadmin-options] create-protocol [--help]
  [--securityenabled={false|true}] [--target target] protocol-name
```

#### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

#### 機能

create-protocol サブコマンドはプロトコルを作成します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--securityenabled={false|true}

プロトコルで SSL を実行するかどうかを指定します。この値に true を設定すると、プロトコルで SSL を実行します。

さらに SSL2 または SSL3 を有効または無効にすると、ssl 要素を使用して暗号を設定できます。セキュリティ設定で SSL をグローバルレベルで有効または無効にするには、サーバインスタンスで利用できる証明書を作成します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

**--target** *target*

作成するプロトコルの対象を指定します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- **configuration\_name**  
指定した名称の構成を対象とします。
- **cluster\_name**  
指定した名称のクラスターを対象とします。
- **standalone\_instance\_name**  
指定した名称のスタンドアロンのサーバインスタンスを対象とします。

デフォルト値：server

**protocol-name**

プロトコルの名称を指定します。

型：String

指定できる値を次に示します。

- プロトコル名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、プロトコルを作成します。

```
asadmin create-protocol --target instance1 --securityenabled=true http-1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.2 create-ssl

セキュア通信の構成

## 書式

```
asadmin [asadmin-options] create-ssl [--help] [--target target]  
  --type listener_or_service_type --certname cert_name  
  [--ssl2enabled={false|true}] [--ssl2ciphers ssl2ciphers]  
  [--ssl3enabled={true|false}] [--tlseabled={true|false}]  
  [--ssl3tlsciphers ssl3tlsciphers]  
  [--clientauthenabled={false|true}] [listener_id]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`create-ssl` サブコマンドは、選択した HTTP リスナー、IIOP リスナーまたは IIOP サービスに SSL 要素を作成し、それらリスナー、サービスでセキュア通信を有効にするように構成します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

`--type` オプションでは、タイプが `iiop-service` であると、組み込み SSL 要素とともに `ssl-client-config` が `domain.xml` に作成されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

SSL 要素の構成対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### `--type listener_or_service_type`

SSL を作成するサービスやリスナーのタイプを指定します。

`--type` が `iiop-service` である場合、`listener_id` は必要ありません。タイプが `iiop-service` の場合、組み込み SSL 要素と一緒に `ssl-client-config` が `domain.xml` に作成されます。

型：String

指定できる値を次に示します。

- http-listener
- iiop-listener
- iiop-service
- jmx-connector
- network-listener

デフォルト値：なし

#### `--certname cert_name`

証明書名または PKCS#11 トークンを指定します。

証明書名のフォーマットは、"トークン名:ニックネーム"です。このプロパティ中の、"トークン名:"の指定は任意です。

型：String

指定できる値を次に示します。

- 証明書名

デフォルト値：なし

#### `--ssl2enabled={false|true}`

SSL2 を有効にするかどうかを指定します。仮想サーバに対して SSL2 と SSL3 の両方を有効にした場合、サーバは最初に SSL3 暗号化を試行します。SSL3 暗号化が使用できない場合、サーバは SSL2 暗号化を試行します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

#### `--ssl2ciphers ssl2ciphers`

使用する SSL2 暗号をコンマ区切りリストで指定します。



指定できる値に記載のない暗号は、ユーザーが利用している特定の暗号スイートにおいて利用できる暗号であっても使用できません。

型：String

指定できる値を次に示します。

- rc4
- rc4export
- rc2
- rc2export
- idea
- des
- desede3

デフォルト値：

- rc4
- rc4export
- rc2
- rc2export
- idea
- des
- desede3

`--ssl3enabled={true|false}`

SSL3 を有効にするかどうかを指定します。

仮想サーバに対して SSL2 と SSL3 の両方を有効にした場合、サーバは最初に SSL3 暗号化を試行します。SSL3 暗号化が使用できない場合、サーバは SSL2 暗号化を試行します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--tlseabled={true|false}`

TLS を有効にするかどうかを指定します。

SSL は TLS を有効にすると、よりセキュアになります。

型：Boolean

指定できる値を次に示します。

- true

- false

デフォルト値：true

`--ssl3tlsciphers ssl3tlsciphers`

使用する SSL3 暗号や TLS 暗号をコンマ区切りリストで指定します。

指定できる値に記載のない暗号は、ユーザーが特定の暗号スイートで利用している場合でも、使用できません。

このオプションの指定がない場合、サポートされた暗号のすべてが有効になります。

`--type` オプションに `iioop-listener` または `iioop-service` を指定した場合は、`--ssl3tlsciphers` オプションで指定する暗号アルゴリズムの前に '+' を指定してください。

型：String

指定できる値を次に示します。

- SSL\_RSA\_WITH\_RC4\_128\_MD5
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_EXPORT\_WITH\_RC4\_40\_MD5
- SSL\_RSA\_WITH\_NULL\_MD5
- SSL\_RSA\_WITH\_RC4\_128\_SHA
- SSL\_RSA\_WITH\_NULL\_SHA

デフォルト値：

- SSL\_RSA\_WITH\_RC4\_128\_MD5
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_RSA\_WITH\_DES\_CBC\_SHA
- SSL\_RSA\_EXPORT\_WITH\_RC4\_40\_MD5
- SSL\_RSA\_WITH\_NULL\_MD5
- SSL\_RSA\_WITH\_RC4\_128\_SHA
- SSL\_RSA\_WITH\_NULL\_SHA

`--clientauthenable={false|true}`

ACL ベースのアクセス制御に関係なく、すべての要求に対して SSL3 クライアント認証を有効にするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

## *listener\_id*

SSL 要素を作成する、HTTP または IIOP リスナーの ID を指定します。

--type が *iiop-service* の場合、*listener\_id* は必要ありません。

型：String

指定できる値を次に示します。

- HTTP リスナーの ID、または IIOP リスナーの ID

デフォルト値：なし

## 使用例

次の例では、*instance1* という名称のサーバインスタンス上の、*http-listener-1* という名称の HTTP リスナーに SSL 要素を作成する方法を示します。

```
asadmin create-ssl --type http-listener --target instance1
--certname sampleCert http-listener-1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.3 delete-protocol

プロトコルの削除

## 書式

```
asadmin [asadmin-options] delete-protocol [--help]
        [--target target] protocol-name
```

## 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

## 機能

*delete-protocol* サブコマンドは指定されたプロトコルを削除します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するプロトコルの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `standalone_instance_name`  
指定した名称のスタンドアロンのサーバインスタンスを対象とします。

デフォルト値：`server`

`protocol-name`

プロトコルの名称を指定します。

型：String

指定できる値を次に示します。

- `プロトコル名`

デフォルト値：なし

## 使用例

次のコマンドは、`instance1` という名称のサーバインスタンスから、`http-1` という名称のプロトコルを削除します。

```
asadmin delete-protocol --target instance1 http-1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.4 delete-ssl

SSL 設定の削除

### 書式

```
asadmin [asadmin-options] delete-ssl [--help]
  [--target target] --type listener_or_service_type
  listener_id
```

### 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

### 機能

delete-ssl サブコマンドは、選択した HTTP リスナー、IIOP リスナー、または IIOP サービスの SSL 要素を削除します。--type オプションで iop-service を指定した場合、*listener\_id* の指定は不要です。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target

削除する SSL の対象を指定します。指定した対象の、リスナーまたは IIOP サービスから SSL が削除されます。

型：String

指定できる値を次に示します。

- server

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### --type

SSL 要素を削除するサービスやリスナーのタイプを指定します。

型：String

指定できる値を次に示します。

- http-listener
- iiop-listener
- iiop-service

デフォルト値：なし

#### *listener\_id*

削除対象の SSL 要素があるリスナーの ID を指定します。

--type オプションで iiop-service を指定した場合、リスナー ID の指定は不要です。

型：String

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンス上の、http-listener-1 という名称の HTTP リスナーから SSL 要素を削除します。

```
asadmin delete-ssl --target instance1 --type http-listener http-listener-1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.5 list-network-listeners

ネットワークリスナーの一覧表示

## 書式

```
asadmin [asadmin-options] list-network-listeners [--help]
        [target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-network-listeners` サブコマンドは、既存のネットワークリスナーを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するネットワークリスナーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、サーバインスタンスのすべてのネットワークリスナーを一覧表示します。

```
asadmin list-network-listeners
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.6 list-protocols

プロトコルの一覧表示

### 書式

```
asadmin [asadmin-options] list-protocols [--help]
        [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-protocols サブコマンドは、既存のプロトコルを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するプロトコルの対象を指定します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。



- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、デフォルトのサーバインスタンスのすべてのプロトコルを一覧表示します。

```
asadmin list-protocols
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.7 list-transport

トランスポートの一覧表示

### 書式

```
asadmin [asadmin-options] list-transport [--help]
        target
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-transport サブコマンドは、既存のトランスポートを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するトランスポートの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：なし

## 使用例

次の例では、デフォルトのサーバインスタンスのすべてのトランスポートを一覧表示します。

```
asadmin list-transport server
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.13.8 list-virtual-servers

仮想サーバの一覧表示

## 書式

```
asadmin [asadmin-options] list-virtual-servers [--help]  
[--target target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-virtual-servers` サブコマンドは、既存の仮想サーバを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- *target* に指定した対象に仮想サーバが存在している必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

一覧表示する仮想サーバの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：*server*

## 使用例

次の例では、*instance1* という名称のサーバインスタンスについて、すべての仮想サーバを一覧表示します。

```
asadmin list-virtual-servers --target instance1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.14 ORB 管理機能で使用するコマンド

---

ORB 管理機能で使用するコマンドの書式、機能などについて説明します。

### 2.14.1 create-iiop-listener

IIOP リスナーの追加

#### 書式

```
asadmin [asadmin-options] create-iiop-listener [--help]
  --listeneraddress address [--iiopport iiop-port-number]
  [--securityenabled={false|true}] [--enabled={true|false}]
  [--target target] listener_id
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

create-iiop-listener サブコマンドは IIOP リスナーを作成します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--listeneraddress *address*

IP アドレスまたは DNS で解決できるホスト名を指定します。

型：String

指定できる値を次に示します。

- IP アドレスまたはホスト名

デフォルト値：なし

--iiopport *iiop-port-number*

IIOP ポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：1072

`--securityenabled={false|true}`

`true` に設定の場合、IIOP リスナーが SSL を実行するよう指定します。SSL2 または SSL3 を有効または無効にすると、SSL 要素を使用して暗号を設定できます。

セキュリティー設定によって、証明書をサーバインスタンスで利用できるまたはできないようにすることで、SSL をグローバルレベルで有効または無効にします。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--enabled={true|false}`

`true` に設定の場合、IIOP リスナーを実行時に有効にします。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`true`

`--target target`

作成する IIOP リスナーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `stand-alone_instance_name`  
指定した名称のスタンドアロンのサーバインスタンスを対象とします。

デフォルト値：`server`

## *listener\_id*

IIOP リスナーの一意的識別子を作成します。

型：String

指定できる値を次に示します。

- リスナー ID

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、IIOP リスナーを作成します。

```
asadmin create-iiop-listener --target instance1 --listeneraddress 192.168.1.100
--iiopport 1400 sample_iiop_listener
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.14.2 delete-iiop-listener

IIOP リスナーの削除

### 書式

```
asadmin [asadmin-options] delete-iiop-listener [--help]
        [--target target] listener_id
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-iiop-listener サブコマンドは、指定した IIOP リスナーを削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除する IIOP リスナーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`listener_id`

削除対象である IIOP リスナーの一意的識別子を指定します。

型：String

指定できる値を次に示します。

- IIOP リスナーの一意的 ID

デフォルト値：なし

## 使用例

次のコマンドは、`instance1` という名称のサーバインスタンスから、`sample_iiop_listener` という名称の IIOP リスナーを削除します。

```
asadmin delete-iiop-listener --target instance1 sample_iiop_listener
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.14.3 list-iiop-listeners

IIOP リスナーの一覧表示

### 書式

```
asadmin [asadmin-options] list-iiop-listeners [--help] [target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-iiop-listeners サブコマンドは、既存の IIOP リスナーを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する IIOP リスナーの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：*server*

## 使用例

次の例では、サーバインスタンスのすべての IIOP リスナーを一覧表示します。

```
asadmin list-iiop-listeners
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.15 リソース管理で使用するコマンド

---

リソース管理で使用するコマンドの書式、機能などについて説明します。

### 2.15.1 add-resources

XML ファイルに指定されたリソースの追加

#### 書式

```
asadmin [asadmin-options] add-resources [--help] [--target target]  
      [--upload={false|true}] xml-file-name
```

#### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

#### 機能

add-resources サブコマンドは、指定した XML ファイル内で指名されているリソースを追加します。DOCTYPE は、resources.xml ファイルで [http://glassfish.org/dtds/glassfish-resources\\_1\\_5.dtd](http://glassfish.org/dtds/glassfish-resources_1_5.dtd) として指定されている必要があります。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### ファイル

絶対パスで任意の場所の XML ファイルを指定してください。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

作成するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- server

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- `domain`  
ドメインを対象とします。
- `cluster-name`  
指定した名称のクラスターを対象とします。
- `instance-name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--upload={false|true}`

ファイルをドメイン管理サーバにアップロードするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`  
ドメイン管理サーバへファイルをアップロードします。
- `false`  
ドメイン管理サーバにファイルをアップロードしません。  
ドメイン管理サーバが管理ユーザーと異なるユーザーで動作していて、ファイルの読み取りアクセス権がないなど、ドメイン管理サーバがファイルにアクセスできない場合は、このサブコマンドはエラーになります。

デフォルト値：

デフォルト値は、ドメイン管理サーバがサブコマンド実行中のホスト上にあるか、リモートホスト上にあるかに依存します。

ドメイン管理サーバがサブコマンドが動作するホスト上にある場合、`false`。

ドメイン管理サーバがリモートホスト上にある場合、`true`。

`xml-file-name`

リソースを含む xml ファイルの名称を指定します。

型：String

指定できる値を次に示します。

- xml ファイル名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、resource.xml ファイルの内容を使用しているリソースを追加します。

```
asadmin add-resources --target instance1
domain-dir/domain_name/config/resource.xml
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.15.2 create-resource-ref

リソースへの参照の作成

### 書式

```
asadmin [asadmin-options] create-resource-ref [--help]
        [--target target] [--enabled={false|true}] reference_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

`create-resource-ref` サブコマンドは、クラスターまたはサーバインスタンスから、あらかじめ作成されたりソースへの参照を作成します。このサブコマンドは、クラスターまたはサーバインスタンスが開始状態でない場合でも成功します。新たに作成したりソースが反映されるのは、次の起動時です。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

作成するリソース参照の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`--enabled={false|true}`

リソースを有効にするかどうかを指定します。

この値が有効になるのは、リソースがグローバルレベルで有効な場合だけです。

型：`Boolean`

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`true`

`reference_name`

リソースの名称、または JNDI 名を指定します。

型：`String`

指定できる値を次に示します。

- リソース名、またはリソースの JNDI 名

デフォルト値：なし

## 使用例

次の例では、クラスター `Cluster1` 上にある `jdbc/DerbyPool` という名称の JDBC リソースへの参照を作成します。

```
asadmin create-resource-ref --target Cluster1 jdbc/DerbyPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.15.3 delete-resource-ref

リソース参照の削除

### 書式

```
asadmin [asadmin-options] delete-resource-ref [--help]
        [--target target] reference_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-resource-ref` サブコマンドは、クラスターまたはクラスター化されていないサーバインスタンスから、リソース (JDBC リソースなど) 参照を削除します。このサブコマンドを実行すると、対象インスタンスまたはクラスターの JNDI ツリーからリソース参照が削除されます。このサブコマンドの実行には、クラスターを構成する対象でサーバインスタンスが実行されていないことが必要です。

このサブコマンドでリソース参照を削除しても、ドメインからリソースは削除されません。リソースを削除したい場合は、リソース用の `delete` サブコマンド (`delete-jdbc-resource` など) を使用してください。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するリソース参照の対象を指定します。

型: String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

*reference\_name*

リソースの JNDI 名を指定します。

型：String

デフォルト値：なし

## 使用例

次の例では、cluster1 クラス上の jdbc/DerbyPool という名称の JDBC リソースへの参照を削除します。

```
asadmin delete-resource-ref --target cluster1 jdbc/DerbyPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.15.4 list-resource-refs

リソース参照の一覧表示

### 書式

```
asadmin [asadmin-options] list-resource-refs [--help]  
[target]
```

### 格納先

*Application Server* インストールディレクトリー/javaee/glassfish/bin

### 機能

list-resource-refs サブコマンドは、クラスターまたはクラスター化されていないサーバインスタンスにあるすべてのリソース参照を一覧表示します。指定された対象の JNDI ツリーにある有効なすべてのリソース (JDBC リソースなど) を一覧表示できます。

クラスターを構成する 1 つ以上の対象のインスタンスは実行されていなくても、また使用できなくても、このサブコマンドは正常に実行されます。

このサブコマンドはリモートモードだけサポートします。



## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するリソース参照の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、クラスター`cluster1`のリソース参照を一覧表示します。

```
asadmin list-resource-refs cluster1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.16 データベースコネクション管理で使用するコマンド

データベースコネクション管理で使用するコマンドの書式、機能などについて説明します。

### 2.16.1 create-jdbc-connection-pool

JDBC コネクションプールの登録

#### 書式

```
asadmin [asadmin-options] create-jdbc-connection-pool [--help]
  [--datasourceclassname=datasourceclassname]
  [--restype=resourcetype]
  [--steadypoolsize=poolsize]
  [--maxpoolsize=maxpoolsize]
  [--maxwait=maxwaittime]
  [--poolresize=poolresizelimit]
  [--idletimeout=idletimeout]
  [--isolationlevel=isolationlevel]
  [--isolationguaranteed={true|false}]
  [--isconnectvalidatereq={false|true}]
  [--validationmethod=validationmethod]
  [--validationtable=validationtable]
  [--nontransactionalconnections={false|true}]
  [--validateatmostonceperiod=validationinterval]
  [--leaktimeout=leaktimeout]
  [--leakreclaim={false|true}]
  [--creationretryattempts=creationretryattempts]
  [--creationretryinterval=creationretryinterval]
  [--statementtimeout=statementtimeout]
  [--lazyconnectionenlistment={false|true}]
  [--lazyconnectionassociation={false|true}]
  [--driverclassname=jdbcdriverclassname]
  [--matchconnections={false|true}]
  [--maxconnectionusagecount=maxconnectionusagecount]
  [--ping={false|true}]
  [--pooling={false|true}]
  [--statementcachesize=statementcachesize]
  [--validationclassname=validationclassname]
  [--description description]
  [--property name=value[:name=value]...]
  connectionpoolid
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

create-jdbc-connection-pool サブコマンドは、指定された JDBC コネクションプール名を持つ Java Database Connectivity (JDBC) ソフトウェアを新規登録します。

次のようにして、認証付きの JDBC コネクションプールは作成できます。

- --property オプションを使用して、ユーザー、パスワードなどのコネクション情報を指定します。
- XML ディスクリプターファイルでコネクション情報を指定します。

接続するデータベースに応じて次に示すオプションを指定します。

- HiRDB に接続する場合に指定するオプション

オプション名	値	備考	
--datasourceclassname	JP.co.Hitachi.soft.HiRDB.JDBC.PrdbDataSource	※1	
	JP.co.Hitachi.soft.HiRDB.JDBC.PrdbXADataSource	※2	
--restype	javax.sql.DataSource	※1	
	javax.sql.XADataSource	※2	
--validationmethod	custom-validation	※3	
--validationclassname	com.hitachi.software.javaee.jca.connectionvalidation.HiRDBConnectionValidation	※3	
--property	description	ポート番号	※1
		環境変数グループ識別子	※2
	DBHostName	HiRDBのホスト名	-
	XALocalCommitMode	true	※2、 ※4
	XAOpenString	環境変数グループ識別子+HiRDB.iniファイルのパス	※2
	user	ユーザー名	-
	password	パスワード	-
HiRDBINI	HiRDB.INIファイルが存在するディレクトリーの絶対パス	※5	

- Oracle に接続する場合に指定するオプション

オプション名	値	備考
--datasourceclassname	oracle.jdbc.pool.OracleDataSource	※1
	oracle.jdbc.xa.client.OracleXADataSource	※2
--restype	javax.sql.DataSource	※1
	javax.sql.XADataSource	※2

オプション名		値	備考
--validationmethod		custom-validation	※3
--validationclassname		org.glassfish.api.jdbc.validation.OracleConnectionValidation	※3
--property	databaseName	Oracle SID	※6
	serverName	Oracleのホスト名称、またはIPアドレス	※6
	portNumber	ポート番号	※6
	user	ユーザー名	-
	password	パスワード	-
	driverType	thin	※6
	url	jdbc:oracle:thin:@Oracleのホスト名称、またはIPアドレス:ポート番号:Oracle SID	※6

- SQL Server に接続する場合に指定するオプション

オプション名		値	備考
--datasourceclassname	com.microsoft.sqlserver.jdbc.SQLServerDataSource		※1
	com.microsoft.sqlserver.jdbc.SQLServerXADataSource		※2
--restype	javax.sql.DataSource		※1
	javax.sql.XADataSource		※2
--validationmethod		custom-validation	※3
--validationclassname		org.glassfish.api.jdbc.validation.MSSQLConnectionValidation	※3
--property	databaseName	SQL Serverデータベース名	-
	serverName	SQL Serverのホスト名称、またはIPアドレス	-
	portNumber	ポート番号	-
	user	ユーザー名	-
	password	パスワード	-

注※1

XA トランザクションを使用しない場合に指定します。

注※2

XA トランザクションを使用する場合に指定します。

注※3

コネクション障害検知を行う場合だけ指定します。

#### 注※4

このオプションは、トランザクションがグローバルトランザクションでないときに、オートコミット機能を有効にするかどうかを設定します。この設定がtrueでない場合、コネクション障害検知機能が正常に動作しません。コネクション障害検知機能を使用する場合、trueを指定します。

#### 注※5

XA トランザクションを使用しない場合、かつHiRDB.ini ファイルに指定されている設定を使用する場合に指定します。

#### 注※6

databaseName、serverName、portNumber、driverType の4つのプロパティか、url のどちらかを指定します。両方を指定した場合、url が有効になり、それ以外の4つのプロパティは無視されます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

JDBC コネクションプールを作成するのに--property オプションを使用しない場合は、XML ディスクリプターでコネクション情報を指定します。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--datasourceclassname=*datasourceclassname*

ベンダー供給の JDBC データソースのリソースマネージャーの名称を指定します。グローバルトランザクションを使用する場合は、javax.sql.XADataSource インターフェースを実装したクラスを指定します。

型：String

指定できる値を次に示します。

- datasource クラス名

デフォルト値：なし

--restype=*resourcetype*

このオプションは次のどちらかの場合に必要になります。

- DataSource クラスが、javax.sql.DataSource、javax.sql.XADataSource または javax.sql.ConnectionPoolDataSource のうち2つ以上のインターフェースを実装する時
- Driver クラスの提供が必須の時

--restype オプションにjava.sql.Driver を指定する場合、--driverclassname オプションも必要です。

--restype オプションに `javax.sql.DataSource`、`javax.sql.XADataSource`、または `javax.sql.ConnectionPoolDataSource` を指定する場合、--datasourceclassname オプションも必要です。  
--restype オプションを指定しない場合、--driverclassname オプションか--datasourceclassname オプションのどちらか一方を必ず指定してください。

型：String

指定できる値を次に示します。

- `java.sql.Driver`
- `javax.sql.DataSource`
- `javax.sql.XADataSource`
- `javax.sql.ConnectionPoolDataSource`

デフォルト値：なし

--steadypoolsize=*poolsize*

プールに保持する初期コネクション数と最小コネクション数を指定します。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：24

`glassfish-resources.xml` の `jdbc-connection-pool` 要素の `steady-pool-size` 属性のデフォルト値も、このオプションのデフォルト値と同じです。

--maxpoolsize=*maxpoolsize*

作成できる最大コネクション数を指定します。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：24

`glassfish-resources.xml` の `jdbc-connection-pool` 要素の `max-pool-size` 属性のデフォルト値も、このオプションのデフォルト値と同じです。

--maxwait=*maxwaittime*

コネクションタイムアウトが送信されるまでに、呼び出し側が待機する必要がある時間をミリ秒単位で指定します。

値0は、呼び出し側を無期限に待機させます。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：60000

`--poolresize=poolresizelimit`

`idle-timeout-in-seconds` タイマーが終了の際、解除されるコネクション数を指定します。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：2

`--idletimeout=idletimeout`

コネクションをアイドル状態で維持できる最長時間を秒単位で指定します。

このタイムアウト値は、アプリケーションのコネクションに使えない時間の蓄積を避けるため、データベースサーバ側のタイムアウト値よりも短くする必要があります。

0 を指定した場合、無効になります。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：300

`--isolationlevel=isolationlevel`

プール化データベースコネクションに設定されたトランザクションアイソレーションレベルを指定します。

プールされたコネクションに対して分離レベルをプログラマ的に変更するようなアプリケーションを利用した場合、プールに不整合が起きる危険があります。これはプログラムエラーにつながります。

型：String

指定できる値を次に示します。

- read-uncommitted
- read-committed
- repeatable-read
- serializable

デフォルト値：JDBC ドライバーのプロバイダーが設定する独立性レベル

`--isolationguaranteed={true|false}`

特定の分離レベルがトランザクションアイソレーションレベルに指定された場合にだけ適用されます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--isconnectvalidatereq={false|true}`

コネクション障害検知機能を使用するかどうかを指定します。true の場合、コネクション障害検知機能を使用します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--validationmethod=validationmethod`

`--isconnectvalidatereq` オプションが true の場合、実行する検証のタイプを指定します。

- auto-commit を指定した場合：  
Connection#setAutoCommit(Connection#getAutoCommit())を実行します。
- meta-data を指定した場合：  
Connection#getMetaData()を実行します。
- table を指定した場合：  
`--validationtable` オプションで指定したテーブルに対して、SELECT 文を実行します。
- custom-validation を指定した場合：  
`--validationclassname` オプションで指定したユーザ定義のクラスを使用します。

型：String

指定できる値を次に示します。

- auto-commit
- meta-data
- table
- custom-validation

デフォルト値：table

`--validationtable=validationtable`

使用する検証テーブルの名称を指定し、SELECT 文を実行してコネクションを検証します。

`--isconnectvalidatereq` オプションが true の場合、`--validationmethod` オプションが table に設定されていれば、このオプションの設定は必須です。

型：String

指定できる値を次に示します。

- 検証テーブル名

デフォルト値：なし



`--nontransactionalconnections={false|true}`

このプロパティを持つプールがtrueに設定の場合、非トランザクションコネクションを返します。このコネクションは、トランザクションマネージャーに自動的に登録されません。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--validateatmostonceperiod=validationinterval`

連続するリクエスト間で、コネクションを検証する間隔を秒単位で指定します。この属性を適切な値に設定すると、コネクションの検証リクエスト数を最小限にします。

0を指定した場合、常にコネクションを検証します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--leaktimeout=leaktimeout`

コネクションプール内のコネクションリークを追跡する時間を秒単位で指定します。指定の時間内にアプリケーションによってコネクションがプールに戻されない場合は、潜在的リークがあるものと見なされ、呼び出し側のスタックトレースが記録されます。0を指定した場合はコネクションリークを追跡しません。

コネクションリークがある場合にだけ、このオプションで検知できます。

`--leakreclaim` オプションがtrueに設定される場合だけ、コネクションは再開できます。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--leakreclaim={false|true}`

リークコネクションの追跡が完了した後で、リークコネクションをコネクションプールに復元するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true  
リークコネクションはコネクションプールに復元されます。
- false  
リークコネクションはコネクションプールに復元されません。

デフォルト値：false

`--creationretryattempts=creationretryattempts`

Java EE Server が、最初の接続の確立に失敗した場合に、最大何回まで接続再確立を試行するかを指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--creationretryinterval=creationretryinterval`

接続確立のリトライ間隔を秒単位で指定します。

`--creationretryattempts` が0の場合、`--creationretryinterval` オプションは無視されます。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：10

`--statementtimeout=statementtimeout`

完了していないクエリーが終了するまでの時間を秒単位で指定します。

型：Integer

指定できる値を次に示します。

- -1~2147483647

デフォルト値：-1

`--lazyconnectionenlistment={false|true}`

メソッドがリソースを実際に使用する場合、リソースをトランザクションに参加させるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true

メソッドがリソースを実際に使用する場合に限り、リソースをトランザクションに参加させます。

- false

リソースはトランザクションに常時参加させます。メソッドがリソースを実際に使用する場合に限りません。

デフォルト値：false

`--lazyconnectionassociation={false|true}`

物理コネクションを使用する際に、その物理コネクションを論理コネクションに関連づけ、および、トランザクションが完了した際には関連づけないことを指定します。関連づけの有無は物理コネクションの再使用を有効にします。

型：Boolean

指定できる値を次に示します。

- **true**

物理コネクションは、物理コネクションを使用するときだけに関連づけられ、トランザクションが完了した際には関連づけられません。

`--lazyconnectionenlistment` オプションも **true** に設定する必要があります。

- **false**

物理コネクションは、物理コネクションを使用する前でも論理コネクションに関連づけられ、トランザクションが完了した際には関連づけられません。

デフォルト値：false

`--driverclassname=jdbcdriverclassname`

ベンダー供給の JDBC ドライバーのクラス名を指定します。このドライバーは `java.sql.Driver` インターフェースを実装している必要があります。

型：String

指定できる値を次に示します。

- `java.sql.Driver` インターフェースを実装する Java クラス

デフォルト値：なし

`--matchconnections={false|true}`

プールから選択したコネクションがリソースアダプターと一致する必要があるかどうかを指定します。

プール内のすべてのコネクションが同一の場合、コネクションとリソースアダプターが一致する必要はありません。

型：Boolean

指定できる値を次に示します。

- **true**

コネクションはリソースアダプターと一致する必要があります。

- **false**

コネクションはリソースアダプターと一致する必要はありません。

デフォルト値：false

`--maxconnectionusagecount=maxconnectionusagecount`

コネクションを再使用できる最大回数を指定します。最大数に達すると、コネクションが閉じられます。コネクションを再使用できる最大回数を制限することで、ステートメントのリークが回避できます。0 を指定した場合はコネクションを再使用できる回数の上限がなくなります。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：0

`--ping={false|true}`

true の場合、コネクションプール作成時または再構成時にコネクションプールが使用できるかどうかテストします。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--pooling={false|true}`

プールに対してコネクションのプーリングが有効かどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--statementcachesize=statementcachesize`

デフォルトのキャッシュ機構を使用してキャッシュされる SQL ステートメント数を指定します。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：0

`--validationclassname=validationclassname`

`--validationmethod` オプションが `custom-validation` のときに独自の検証を提供するクラス名を指定します。このクラスは、`org.glassfish.api.jdbc.ConnectionValidation` インターフェースを実装する必要があります、さらに Java EE Server にアクセスできるようにする必要があります。

`--validationmethod` オプションが `custom-validation` に設定されている場合、このオプションは必須です。

型：String

指定できる値を次に示します。

- `org.glassfish.api.jdbc.ConnectionValidation` インターフェースを実装したクラス

デフォルト値：なし

## `--description description`

指定された JDBC コネクションプールの詳細を説明するテキストを指定します。

型：String

指定できる値を次に示します。

- テキストでの説明

デフォルト値：なし

## `--property name=value[:name=value]...`

プールを構成するためのオプションの属性名と値のペアを指定します。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

属性名または属性値にコロンが含まれる場合、そのコロンをエスケープするために円記号 (¥) を使用する必要があります。別の文字でも、エスケープ文字が必要になることがあります。

HiRDB または Oracle に接続するためのプロパティは、それぞれ「HiRDB に接続する場合に指定するオプション」または「Oracle に接続する場合に指定するオプション」を参照してください。そのほかに指定できるプロパティについては、使用するデータベースのマニュアルを参照してください。

プロパティ名と *value* に指定できる値を次に示します。

### `user=value`

データベースに接続するために必要なユーザー名を指定します。

型：String

デフォルト値：なし

### `password=value`

型：String

デフォルト値：なし

データベースに接続するために必要なパスワードを指定します。

### `databaseName=value`

このコネクションプールのデータベースを指定します。

型：String

デフォルト値：なし

### `serverName=value`

このコネクションプールのデータベースサーバを指定します。

型：String

デフォルト値：なし

### `portNumber=value`

データベースサーバがリクエストをリッスンするポートを指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：なし

`networkProtocol=value`

通信プロトコルを指定します。

型：String

デフォルト値：なし

`roleName=value`

初期 SQL ロール名を指定します。

型：String

デフォルト値：なし

`datasourceName=value`

コネクションプーリングが動作する XADataSource または ConnectionPoolDataSource を指定します。

型：String

デフォルト値：なし

`description=value`

テキストの説明を指定します。

型：String

デフォルト値：なし

`url=value`

このコネクションプールの URL を指定します。これは標準プロパティーではありませんが、共通に使用されます。

型：String

デフォルト値：なし

`dynamic-reconfiguration-wait-timeout-in-seconds=value`

コネクションプールの動的再構成を、コネクションプールを使用中のアプリケーションに透過的に有効にするために使用されます。これによって、プールへの属性またはプロパティーの変更を有効にするためにアプリケーションを再起動する必要がなくなります。

処理中のトランザクションのコネクション取得要求は、タイムアウト期間内のコネクション取得要求であるかぎり、トランザクションを完了するために古いプール構成のまま実行することが許可されます。新しいコネクション取得要求はプールの再構成が完了するまで待機し、コネクションは変更されたプール構成を使用して取得されます。

型：Long

指定できる値を次に示します。

- $x \sim y$  ( $x$  の値  $\times 1000 \geq -9223372036854775808$  および  $y$  の値  $\times 1000 \leq 9223372036854775807$ )

デフォルト値：0

## *connectionpoolid*

作成する JDBC コネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクションプール ID

デフォルト値：なし

## 使用例

次の例は、JDBC コネクションプールを作成します。

```
asadmin create-jdbc-connection-pool
--datasourceclassname org.apache.derby.jdbc.ClientDataSource
--restype javax.sql.XADataSource
--property portNumber=1527:password=APP:user=APP:serverName=
localhost:databaseName=sun-appserv-samples:connectionAttributes=¥;
create¥¥=true sample_derby_pool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.16.2 create-jdbc-resource

JDBC リソースの作成

### 書式

```
asadmin [asadmin-options] create-jdbc-resource [--help]
--connectionpoolid connectionpoolid [--enabled={false|true}]
[--description description][--target target] jndi_name
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-jdbc-resource` サブコマンドは新しい JDBC リソースを作成します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--connectionpoolid connectionpoolid`

JDBC コネクションプールの名称を指定します。2つ以上の JDBC リソース要素が同じコネクションプール要素をポイントしている場合、それらは実行時に同じプール接続を使用します。

型：String

指定できる値を次に示します。

- コネクションプール ID

デフォルト値：なし

`--enabled={false|true}`

JDBC リソースを実行時に有効にするかどうかを確認します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--description description`

JDBC リソースの詳細な説明をするテキストを記述します。

型：String

指定できる値を次に示します。

- JDBC リソースの詳細な説明

デフォルト値：なし

`--target target`

作成する JDBC リソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソースの `resource-ref` は指定した対象に対してだけ作成されるため、リソースがドメインで定義されても、リソースは指定した対象だけで使用できるようになります。複数の対象でリソースを参照する場合は、`create-resource-ref` サブコマンドを使用します。

型：String

指定できる値を次に示します。

- server



ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### *jndi\_name*

この JDBC リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、jdbc/DerbyPool という名称の JDBC リソースを作成します。

```
asadmin create-jdbc-resource --target instance1
--connectionpoolid sample_derby_pool jdbc/DerbyPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.16.3 delete-jdbc-connection-pool

JDBC コネクションプールの削除

### 書式

```
asadmin [asadmin-options] delete-jdbc-connection-pool [--help]
[--cascade={false|true}] jdbc_connection_pool_id
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-jdbc-connection-pool` サブコマンドは JDBC コネクションプールを削除します。このサブコマンドを実行する前に、必ずこの JDBC コネクションプールへのすべての関連づけを削除してください。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--cascade={false|true}`

コネクションプールとそのプールに関連づけられているすべての JDBC リソースを削除します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`jdbc_connection_pool_id`

JDBC リソースの名称を指定します。

型：String

指定できる値を次に示します。

- JDBC リソース名

デフォルト値：なし

## 使用例

次の例では、`sample_derby_pool` という名称の JDBC コネクションプールを削除します。

```
asadmin delete-jdbc-connection-pool --cascade=false sample_derby_pool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.16.4 delete-jdbc-resource

JNDI 名を持つ JDBC リソースの削除

### 書式

```
asadmin [asadmin-options] delete-jdbc-resource [--help]
        [--target target] jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-jdbc-resource` サブコマンドは JDBC リソースを削除します。このサブコマンドを実行する前に、必ずこの JDBC リソースへのすべての関連づけを削除してください。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除する JDBC リソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソース作成時に `--target` オプションを使用して、`resource-ref` が作成された対象だけがアクティブになります。

リソースの削除は、`--target` オプションで指定した対象の `resource-ref` だけを削除します。削除の対象としてドメインが指定されない限り、ドメイン全体からそのリソースは削除されません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *jndi\_name*

JDBC リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JDBC リソースの JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`jdbc/DerbyPool` という名称の JDBC リソースを削除します。

```
asadmin delete-jdbc-resource --target instance1 jdbc/DerbyPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.16.5 list-jdbc-connection-pools

JDBC コネクションプールの一覧表示

### 書式

```
asadmin [asadmin-options] list-jdbc-connection-pools [--help]
```

## 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

list-jdbc-connection-pools サブコマンドは、現在の JDBC コネクションプールを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

## 使用例

次の例では、現在の JDBC コネクションプールを一覧表示します。

```
asadmin list-jdbc-connection-pools
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.16.6 list-jdbc-resources

JDBC リソースの一覧表示

## 書式

```
asadmin [asadmin-options] list-jdbc-resources [--help]  
[target]
```

## 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

`list-jdbc-resources` サブコマンドは、既存の JDBC リソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する JDBC リソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、現在の JDBC リソースを一覧表示します。

```
asadmin list-jdbc-resources
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.17 JavaMail サービス管理で使用するコマンド

---

JavaMail サービス管理で使用するコマンドの書式、機能などについて説明します。

### 2.17.1 create-javamail-resource

JavaMail セッションリソースの作成

#### 書式

```
asadmin [asadmin-options] create-javamail-resource [--help]
  [--target target] --mailhost hostname --mailuser username
  --fromaddress address [--storeprotocol storeprotocol]
  [--storeprotocolclass storeprotocolclass]
  [--transprotocol transprotocol]
  [--transprotocolclass transprotocolclass]
  [--enabled={true|false}] [--description resource-description]
  [--property name=value[:name=value]...] jndi-name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

create-javamail-resource サブコマンドは、JavaMail セッションリソースを作成します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

作成する JavaMail セッションリソースの対象を指定します。

型：String

指定できる値を次に示します。

- **server**

ドメイン管理サーバを対象とします。**server** はドメイン管理サーバの名称です。

- *domain*  
ドメインを対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--mailhost *hostname***

デフォルトメールサーバの DNS 名を指定します。プロトコル固有のホストプロパティが指定されていない場合、Store および Transport オブジェクトの接続メソッドは、この値を使用します。

型：String

指定できる値を次に示します。

- メールホスト名

デフォルト値：なし

**--mailuser *username***

メールサーバへの接続時に使用されるメールアカウントユーザーの名前を指定します。

型：String

指定できる値を次に示します。

- メールサーバのユーザー名

デフォルト値：なし

**--fromaddress *address***

デフォルトユーザーのメールアドレスを `username@host.domain` 形式で指定します。

型：String

指定できる値を次に示します。

- デフォルトユーザーのメールアドレス

デフォルト値：なし

**--storeprotocol *storeprotocol***

メールサーバのストアプロトコルを指定します。

型：String

指定できる値を次に示します。

- *imap*
- *pop3*

デフォルト値：imap



`--storeprotocolclass storeprotocolclass`

メールサーバのストアプロトコルクラス名を指定します。

型：String

指定できる値を次に示します。

- `com.sun.mail.imap.IMAPStore`
- `com.sun.mail.pop.POPStore`

デフォルト値：`com.sun.mail.imap.IMAPStore`

`--transprotocol transprotocol`

メールサーバのトランスポートプロトコルを指定します。

型：String

指定できる値を次に示します。

- `smtp`
- `lmt`

デフォルト値：`smtp`

`--transprotocolclass transprotocol`

メールサーバのトランスポートプロトコルのクラス名を指定します。

型：String

指定できる値を次に示します。

- `com.sun.mail.smtp.SMTPTransport`
- `com.sun.mail.lmt.LMTPTTransport`

デフォルト値：`com.sun.mail.smtp.SMTPTransport`

`--enabled={true|false}`

`true` を指定した場合、リソースは実行時に有効になります。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`true`

`--description resource-description`

JavaMail リソースの詳細の一部を記述するテキストを指定します。

型：String

指定できる値を次に示します。

- JavaMail リソースの説明

デフォルト値：なし

`--property name=value[:name=value]...`

JavaMail リソース構成時のオプションをプロパティ名と値のペアで指定します。Java EE Server 固有の mail-プレフィックスは標準のメールプレフィックスに変換されます。指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

*name* に指定できる値については、Java Mail 標準仕様の JavaDoc(<http://docs.oracle.com/javaee/7/api/javax/mail/package-summary.html>)から、`javax.mail` のパッケージサマリーを参照してください。デフォルト値はありません。

*value* には、プロパティ名に指定したプロパティの指定範囲に従う値が指定できます。

### *jndi-name*

作成する JavaMail リソースの JNDI 名を指定します。JavaMail リソースには、ネーミングサブコンテキストのプレフィックス `mail/` の使用を推奨します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサービンスタンスに、JavaMail セッションリソースを作成します。

```
asadmin create-javamail-resource --target instance1 --mailhost localhost
--mailuser sample --fromaddress sample@sun.com mail/MyMailSession
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.17.2 delete-javamail-resource

JavaMail セッションリソースの削除

### 書式

```
asadmin [asadmin-options] delete-javamail-resource [--help]
[--target target] jndi_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-javamail-resource` サブコマンドは指定した JavaMail セッションリソースを削除します。このサブコマンドを実行する前に、必ずこのリソースへのすべての参照を削除してください。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除する JavaMail セッションリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`jndi_name`

JavaMail セッションリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- リソースの JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、mail/MyMailSession という名称の JavaMail セッションリソースを削除します。

```
asadmin delete-javamail-resource --target instance1 mail/MyMailSession
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.17.3 list-javamail-resources

JavaMail セッションリソースの一覧表示

### 書式

```
asadmin [asadmin-options] list-javamail-resources [--help]  
[target]
```

### 格納先

*Application Server* インストールディレクトリー/javaee/glassfish/bin

### 機能

list-javamail-resources サブコマンドは、既存の JavaMail セッションリソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する JavaMail セッションリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのサーバインスタンスの JavaMail セッションリソースを一覧表示します。

```
asadmin list-javamail-resources
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.18 JNDI サービス管理で使用するコマンド

JNDI サービス管理で使用するコマンドの書式、機能などについて説明します。

### 2.18.1 create-custom-resource

カスタムリソースの作成

#### 書式

```
asadmin [asadmin-options] create-custom-resource [--help]
  --restype type --factoryclass classname
  [--enabled={true|false}] [--description text]
  jndi-name [--target target]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`create-custom-resource` サブコマンドは、カスタムリソースを作成します。カスタムリソースは、`javax.naming.spi.ObjectFactory` インターフェースを実装するサーバ全体のカスタムリソースのオブジェクトファクトリーを指定します。

カスタムリソースの作成後は、Java EE Server を再起動してください。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

作成するカスタムリソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソースの `resource-ref` は指定した対象に対してだけ作成されるため、リソースがドメインで定義されても、リソースは指定した対象だけで使用できるようになります。複数の対象でリソースを参照する場合は、`create-resource-ref` サブコマンドを使用します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- **domain**  
ドメインを対象とします。
- **cluster\_name**  
指定した名称のクラスターを対象とします。
- **instance\_name**  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--restype** *type*

作成するカスタムリソースのタイプを指定します。

型：String

指定できる値を次に示します。

- カスタムリソース型

デフォルト値：なし

**--factoryclass** *classname*

カスタムリソースのファクトリークラス名を指定します。

型：String

指定できる値を次に示します。

- javax.naming.spi.ObjectFactory インターフェースを実装するクラスの名称

デフォルト値：なし

**--enabled**={true|false}

カスタムリソースをランタイムに有効にするかどうかを確認します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

**--description** *text*

カスタムリソースの詳しい説明文を記述します。説明文の長さは最大 250 文字です。

型：String

指定できる値を次に示します。

- テキストでの説明

デフォルト値：なし

### *jndi-name*

このリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、カスタムリソースを作成します。

```
asadmin create-custom-resource --target instance1 --restype topic
--factoryclass com.imq.topic mycustomresource
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.18.2 create-jndi-resource

JNDI リソースの登録

### 書式

```
asadmin [asadmin-options] create-jndi-resource [--help]
  [--target target] --restype restype --factoryclass factoryclass
  --jndilookupname jndilookupname [--enabled={true|false}]
  [--description description] jndi-name
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-jndi-resource` サブコマンドでは JNDI リソースを登録します。

このサブコマンドはリモートモードだけサポートします。



## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

作成する JNDI リソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソースの `resource-ref` は指定した対象に対してだけ作成されるため、リソースがドメインで定義されても、リソースは指定した対象だけで使用できるようになります。複数の対象でリソースを参照する場合は、`create-resource-ref` サブコマンドを使用します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`--restype restype`

JNDI リソースのタイプを指定します。

型：String

指定できる値を次に示します。

- `topic`
- `queue`

デフォルト値：なし

`--factoryclass factoryclass`

この JNDI リソースを作成するクラスを指定します。

型：String

指定できる値を次に示します。

- `javax.naming.spi.InitialContextFactory` インターフェースを実装するクラスの名称

デフォルト値：なし

`--jndilookupname jndilookupname`

外部コンテナが使用するルックアップ名を指定します。

型：String

指定できる値を次に示します。

- JNDI ルックアップ名

デフォルト値：なし

`--enabled={true|false}`

リソースが実行時に有効であるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--description description`

JNDI リソースに関する詳細情報を表すテキストを指定します。

型：String

指定できる値を次に示します。

- テキストでの説明

デフォルト値：なし

*jndi-name*

作成対象である JNDI リソースの一意の名称を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサービンスタンスに、`my-jndi-resource` という名称の JNDI リソースを作成します。

```
asadmin create-jndi-resource --target instance1
--restype com.example.jndi.MyResourceType
--factoryclass com.example.jndi.MyInitialContextFactoryClass
--jndilookupname remote-jndi-name --description "sample JNDI resource"
my-jndi-resource
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

### 2.18.3 delete-custom-resource

カスタムリソースの削除

#### 書式

```
asadmin [asadmin-options] delete-custom-resource [--help]
          [--target target] jndi-name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`delete-custom-resource` サブコマンドはカスタムリソースを削除します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストが表示されます。

`--target` *target*

削除するカスタムリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`

ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。

- `domain`

ドメインを対象とします。

- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### *jndi\_name*

リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- リソースの JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、mycustomresource という名称のカスタムリソースを削除します。

```
asadmin delete-custom-resource --target instance1 mycustomresource
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.18.4 delete-jndi-resource

JNDI リソースの削除

### 書式

```
asadmin [asadmin-options] delete-jndi-resource [--help]
        [--target target] jndi_name
```

### 格納先

*Application Server* インストールディレクトリー/javaee/glassfish/bin

## 機能

`delete-jndi-resource` サブコマンドは、指定した JNDI リソースを削除します。このサブコマンドを実行する前に、必ずこの JNDI リソースへのすべての関連づけを削除してください。このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除する JNDI リソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソース作成時に `--target` オプションを使用して、`resource-ref` が作成された対象だけがアクティブになります。

リソースの削除は、`--target` オプションで指定した対象の `resource-ref` の JNDI リソースだけを削除します。削除の対象として `domain` が指定されない限り、ドメイン全体からそのリソースは削除されません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`jndi_name`

JNDI リソースの名称を指定します。

型：String

指定できる値を次に示します。

- `JNDI 名`

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、sample\_jndi\_resource という名称の既存の JNDI リソースを削除します。

```
asadmin delete-jndi-resource --target instance1 sample_jndi_resource
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.18.5 list-custom-resources

カスタムリソースの取得

### 書式

```
asadmin [asadmin-options] list-custom-resources [--help]  
[target]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-custom-resources サブコマンドは、カスタムリソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するカスタムリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのサーバインスタンス上のカスタムリソースを一覧表示します。

```
asadmin list-custom-resources
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.18.6 list-jndi-resources

JNDI リソースの一覧表示

### 書式

```
list-jndi-resources [--help] [target]
```

### 格納先

*Application Server* インストールディレクトリー/`javaee/glassfish/bin`

### 機能

`list-jndi-resources` サブコマンドは、既存のすべての JNDI リソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する JNDI リソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのサーバインスタンス上の JNDI リソースを一覧表示します。

```
asadmin list-jndi-resources
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.19 JMS 管理で使用するコマンド

---

JMS 管理で使用するコマンドの書式、機能などについて説明します。

### 2.19.1 create-jms-host

JMS ホストの作成

#### 書式

```
asadmin [asadmin-options] create-jms-host [--help] --mqhost mq-host  
--mqport mq-port --mquser mq-user --mqpassword mq-password  
[--target target] [--force={false|true}] jms_host_name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

JMS サービス内に JMS ホストを作成します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--mqhost mq-host`

JMS サービスのホスト名を指定します。

型：String

指定できる値を次に示します。

- ホスト名

デフォルト値：localhost

`--mqport mq-port`

JMS サービスが使用するポート番号を指定します。

型：Integer

指定できる値を次に示します。

- 1~65535

デフォルト値：7676

**--mquser** *mq-user*

JMS サービスのユーザー名を指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：admin

**--mqpassword** *mq-password*

JMS サービスのパスワードを指定します。

型：String

指定できる値を次に示します。

- パスワード

デフォルト値：admin

**--target** *target*

作成する JMS ホストの対象を指定します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。**server** はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサービインスタンスを対象とします。

デフォルト値：server

**--force**={false|true}

このサブコマンドが既存の同名 JMS ホストを上書きするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

### *jms\_host\_name*

作成する JMS ホストの固有識別子を指定します。

型：String

指定できる値を次に示します。

- JMS ホストの固有の名称

デフォルト値：なし

## 使用例

次のコマンドは、instance1 という名称のサーバインスタンス上の、システムpigeon に、MyNewHost という名称の JMS ホストを作成します。

```
asadmin create-jms-host --target instance1 --mqhost pigeon.example.com --mqport 7677
--mquser admin --mqpassword admin MyNewHost
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.2 create-jms-resource

JMS リソースの作成

### 書式

```
asadmin [asadmin-options] create-jms-resource [--help] --restype type
  [--target target] [--enabled={true|false}]
  [--property name=value[:name=value]...] [--force={false|true}]
  jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

`create-jms-resource` サブコマンドは、Java Message Service (JMS) コネクションファクトリーリソースまたは JMS 送信先リソースを作成します。

このサブコマンドはリモートモードだけサポートします。

`asadmin create-jms-resource` コマンドの `addresslist` プロパティ ( `host:mqport,host2:mqport,host3:mqport` 形式) を指定する場合、`¥¥` を使用して : 文字をエスケープします。例えば、`host1¥¥:mqport,host2¥¥:mqport,host3¥¥:mqport` のようにします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--restype type`

JMS リソースのタイプを指定します。

型 : String

指定できる値を次に示します。

- `javax.jms.Topic`
- `javax.jms.Queue`
- `javax.jms.ConnectionFactory`
- `javax.jms.TopicConnectionFactory`
- `javax.jms.QueueConnectionFactory`

デフォルト値 : なし

`--target target`

作成する JMS リソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソースの `resource-ref` は指定した対象に対してだけ作成されるため、リソースがドメインで定義されても、リソースは指定した対象だけで使用できるようになります。複数の対象でリソースを参照する場合は、`create-resource-ref` サブコマンドを使用します。

型 : String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。 `server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--enabled={true|false}`

リソースを実行時に有効にするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--property name=value[:name=value]...`

JMS リソースの構成時のオプションをプロパティ名と値のペアで指定します。指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

プロパティ名と *value* に指定できる値を次に示します。

コネクションファクトリーリソースのプロパティ

**ClientId=*value***

永続化サブスクリバが使用するコネクションファクトリーのクライアント ID を指定します。

型：String

指定できる値を次に示します。

- JMS リソースの ID

デフォルト値：なし

**AddressList=*value***

Message Queue のアドレスのコンマ区切りリストを指定します。アプリケーションが通信する 1 つまたは複数のメッセージブローカーのインスタンスのホスト名（およびオプションでポート番号）を指定します。

例えば、値には *earth* または *earth:7677* を指定できます。メッセージブローカーがデフォルト（7676）以外のポートで動作している場合は、ポート番号を指定します。サーバの JMS サービス構成で定義された JMS ホストから構成されるデフォルト値は *localhost* で、デフォルトのポート番号は 7676 です。クライアントは、ローカルホストのポート 7676 のブローカーへの接続を試みます。

型：String

指定できる値を次に示します。

- コンマで区切られたホスト名およびポート番号

デフォルト値：なし

**UserName=*value***

コネクションファクトリーのユーザー名を指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：guest

Password=*value*

コネクションファクトリーのパスワードを指定します。

型：String

指定できる値を次に示します。

- パスワード

デフォルト値：guest

ReconnectEnabled=*value*

値をtrueに設定すると、接続が失われたときに、クライアントのランタイムはメッセージサーバ（またはAddressListで指定したアドレスのリスト）への再接続を試みます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

ReconnectAttempts=*value*

クライアントのランタイムがリストの次のアドレスを試行する前に、AddressListに指定した各アドレスへの接続（または再接続）を試みる回数。値を-1に設定すると、再接続の試行回数が無制限になります（クライアントのランタイムは、接続が成功するまで最初のアドレスへの接続を試みます）。

型：Integer

指定できる値を次に示します。

- -1～2147483647

デフォルト値：6

ReconnectInterval=*value*

再接続を試みる間隔（ミリ秒単位）を指定します。これは、AddressListで指定した各アドレスでの試行およびリストの連続するアドレスへの再接続に適用されます。間隔が短すぎると、ブローカーが回復できません。また間隔が長すぎると、処理遅延が発生することがあります。

型：Long

指定できる値を次に示します。

- -1～9223372036854775807L

デフォルト値：30000（ミリ秒）

AddressListIterations=*value*

接続を確立（または再確立）するために、クライアントのランタイムがAddressList 経由で接続を試みる回数を指定します。値を-1に設定すると、接続回数が無制限になります。コネクタリソースのプロパティとしてconnector-connection-pool 属性を指定できます。

型：Integer

指定できる値を次に示します。

- -1～2147483647

デフォルト値：-1

送信先リソースのプロパティ

**Name=***value*

リソースが参照する物理送信先の名称。物理送信先は、送信先リソースを使用するアプリケーションを実行するときに自動的に作成されます。また、create-jmsdest サブコマンドで物理送信先を作成することもできます。このプロパティを指定しない場合、JMS サービスは送信先リソースと同じ名称で物理送信先を作成します（JNDI 名のスラッシュはアンダースコアに置換されます）。

型：String

指定できる値を次に示します。

- 物理送信先の名称

デフォルト値：\_ replacing /を含む送信先リソースの JNDI 名

**Description=***value*

物理送信先の説明を記述します。

型：String

指定できる値を次に示します。

- 物理送信先の説明

デフォルト値：なし

**--force={false|true}**

このサブコマンドが既存の同名の JMS リソースを上書きするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

*jndi\_name*

作成する JMS リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、JNDI 名が `jms/DurableConnectionFactory` である型 `javax.jms.ConnectionFactory` のコネクションファクトリーリソースを作成します。

`ClientId` プロパティは、クライアント ID をコネクションファクトリー上に設定し、コネクションファクトリーが持続性のあるサブスクリプションに使用できるようにします。JMS リソースの JNDI 名は、習慣的に、`jms/ネーミングサブコンテキスト` を含んでいます。

```
asadmin create-jms-resource --restype javax.jms.ConnectionFactory --target instance1
--property ClientId=MyID jms/DurableConnectionFactory
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.3 create-jmsdest

JMS の物理送信先の作成

### 書式

```
asadmin [asadmin-options] create-jmsdest [--help] --desttype dest_type
[--property name=value[:name=value]...] [--target target]
[--force={false|true}] dest_name
```

### 格納先

*Application Server* インストールディレクトリー/`javaee/glassfish/bin`

### 機能

`create-jmsdest` サブコマンドは JMS の物理送信先を作成します。

このサブコマンドはリモートモードだけサポートします。

`create-jms-resource` サブコマンドを使用しても、`Name` プロパティに物理送信先が指定された JMS 送信先リソースを作成できます。この物理送信先は、ユーザーが送信先リソースを使用するアプリケーション



を実行するときに自動的に作成されます。プロパティにデフォルト値がない物理送信先を作成するには、`create-jmsdest` サブコマンドを使用します。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--desttype dest_type`

JMS 送信先の種類を指定します。

型：String

指定できる値を次に示します。

- `topic`
- `queue`

デフォルト値：なし

`--property name=value[:name=value]...`

物理送信先を構成するためのプロパティ名と値のペアを指定します。指定形式は、`name=value` です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

これらのプロパティの値を変更するには、`Application Server` インストールディレクトリー/`javaee/glassfish/mq/bin/imqcmd` コマンドを使用できます。

プロパティ名と `value` に指定できる値を次に示します。

**`MaxNumMsgs=value`**

送信先で格納できる未配信メッセージの最大数を選択できます。値に `-1` を指定すると、メッセージの数は制限されません。`limitBehavior` プロパティが `FLOW_CONTROL` に設定されている場合、ブローカーはメッセージの受信処理をすぐに停止できないため、指定されたメッセージ制限を超過することがあります。

型：Integer

指定できる値を次に示します。

- `-1~2147483647`

デフォルト値：デッドメッセージキューの場合は、`1000`。その他の場合は、`-1`。

**`MaxBytesPerMsg=value`**

単一メッセージの最大サイズ (バイト単位) を指定します。永続メッセージが拒否されると、メッセージを生成しているクライアントに例外が報告されます。非永続メッセージの場合、例外はクライアントに報告されません。

値は、次のサフィックスを使用して、バイト単位、キロバイト単位、またはメガバイト単位で指定することができます。サフィックスのない値はバイト単位になります。値に-1を設定した場合、メッセージのサイズは制限されません。

- b: バイト
- k: キロバイト (1024 バイト)
- m: メガバイト (1024 x 1024 = 1,048,576 バイト)

型: Integer

指定できる値を次に示します。

- -1 ~ 2147483647

デフォルト値: -1

**MaxTotalMsgBytes=value**

未配信メッセージの最大合計サイズ (バイト単位) を指定します。構文はmaxBytesPerMsg の場合と同じです。

型: Integer

指定できる値を次に示します。

- -1 ~ 2147483647

デフォルト値: デッドメッセージキューの場合は、10000000。その他の場合は、-1

**LimitBehavior=value**

メモリー制限のしきい値に達したときのブローカーの動作を指定します。

型: String

指定できる値を次に示します。

- REJECT\_NEWEST  
最新のメッセージを拒否し、メッセージを保持する場合だけ、メッセージを生成しているクライアントに例外を通知します。
- FLOW\_CONTROL  
メッセージプロデューサーがメッセージを送信する頻度を下げます。
- REMOVE\_OLDEST  
最も古いメッセージを廃棄します。
- REMOVE\_LOW\_PRIORITY  
優先度が最も低いメッセージを、メッセージを生成しているクライアントに通知することなく、古さに応じて削除します。  
値がREMOVE\_OLDEST またはREMOVE\_LOW\_PRIORITY に設定され、さらにuseDMQ プロパティーがtrue に設定されている場合は、制限を超えたメッセージはデッドメッセージキューに移動されます。  
デッドメッセージキュー自体では、デフォルトの制限動作はREMOVE\_OLDEST であり、値にFLOW\_CONTROL は設定できません。

デフォルト値: REJECT\_NEWEST

### MaxNumProducers=*value*

送信先のメッセージプロデューサーの最大数を指定します。この制限に達すると、新しいプロデューサーを作成できなくなります。値に-1を指定すると、プロデューサーの数は制限されません。デフォルト値は100です。このプロパティはデッドメッセージキューには適用されません。

型：Integer

指定できる値を次に示します。

- -1 および1~2147483647

デフォルト値：100

### ConsumerFlowLimit=*value*

単一のバッチでコンシューマーに配信できるメッセージの最大数を指定します。値に-1を指定すると、メッセージの数は制限されません。デフォルト値は1000です。クライアントのランタイムは、より小さい値をコネクションファクトリーオブジェクトに指定することで、この制限を上書きできます。

キュー送信でロードバランスを使用する場合、この値はロードバランスが始まる前に、キューにあるアクティブなコンシューマー宛てメッセージの初期数です。

型：Integer

指定できる値を次に示します。

- -1~2147483647

デフォルト値：1000

### UseDMQ=*value*

trueに設定した場合、デッドメッセージはデッドメッセージキューに移動します。falseに設定した場合、デッドメッセージは破棄されます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

### ValidateXMLSchemaEnabled=*value*

trueに設定した場合、送信先に対するXMLスキーマ検証が有効になります。

XML検証が有効になっている場合、MessageQueueクライアントのランタイムは、XMLメッセージをブローカーに送信する前に、指定されたXSD（XSDが指定されていない場合はDTD）に対するメッセージの妥当性の検証を試みます。指定したスキーマが見つからないかメッセージを検証できない場合、メッセージは送信されないで例外がスローされます。

このプロパティは、送信先が非アクティブである、つまりコンシューマーまたはプロデューサーがない場合、および送信先にメッセージが存在しない場合に設定するようにしてください。それ以外の場合、プロデューサーが再接続する必要があります。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

XMLSchemaURIList=*value*

XML スキーマ定義ファイル URI 文字列の空白区切りのリストを指定します。validateXMLSchemaEnabled がtrue に設定されている場合、URL は XML スキーマ検証に使用するために 1 つ以上の XSD の場所を指します。

複数の URI を指定する場合は、次の例に示すように、この値をダブルクォーテーションで囲みます。

```
"http://foo/flap.xsd http://test.com/test.xsd"
```

このプロパティが設定されていないか、またはnull の場合に XML 検証が有効になっていると、XML ドキュメント内で指定された DTD を使用して XML 検証が実行されます。

アプリケーションの要件が変更された結果 XSD が変更された場合は、変更後の XSD に基づいて XML メッセージを生成するすべてのクライアントアプリケーションをブローカーに再接続する必要があります。

型：String

指定できる値を次に示します。

- スペースで区切られた XSD の URI 文字列のリスト

デフォルト値：null

**--target** *target*

作成する JMS の物理送信先の対象を指定します。

create-jmsdest サブコマンドは各リソースに関連付けられていますが、物理送信先は、構成の一部の JMS サービス（ブローカー）を使用して作成されます。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--force**={false|true}

同じ名称の JMS 物理送信先が存在する場合に、このサブコマンドで上書きするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

#### *dest\_name*

作成対象である JMS 送信先の一意的識別子を指定します。

型：String

指定できる値を次に示します。

- 物理送信先の名称

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、デフォルト値以外のプロパティ値を持つ、PhysicalQueue という名称の JMS 物理キューを作成します。

```
asadmin create-jmsdest --target instance1 --desttype queue
--property maxNumMsgs=1000:maxBytesPerMsg=5k PhysicalQueue
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.4 delete-jms-host

JMS ホストの削除

### 書式

```
asadmin [asadmin-options] delete-jms-host [--help] [--target target]
jms_host_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

`delete-jms-host` サブコマンドは JMS ホストを削除します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除する JMS ホストの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`jms_host_name`

削除するホストの名称を指定します。

型：String

指定できる値を次に示します。

- ホスト名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`MyNewHost` という名称の JMS ホストを削除します。

```
asadmin delete-jms-host --target instance1 MyNewHost
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.5 delete-jms-resource

JMS リソースの削除

### 書式

```
asadmin [asadmin-options] delete-jms-resource [--help]
        [--target target]
        jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-jms-resource` サブコマンドは、Java Message Service (JMS) コネクションファクトリーリソースまたは JMS 送信先リソースを削除します。

このサブコマンドはリモートモードだけサポートします。このサブコマンドをリモートモードで実行するには、ドメイン管理サーバが稼働している必要があります。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

削除する JMS リソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソース作成時に `--target` オプションを使用して、`resource-ref` が作成された対象だけがアクティブになります。

リソースの削除は、`--target` オプションで指定した対象の `resource-ref` の JMS リソースだけを削除します。削除の対象としてドメインが指定されない限り、ドメイン全体からそのリソースは削除されません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *jndi\_name*

削除する JMS リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`.jms/MyQueue` という名称の JMS 送信先リソースを削除します。

```
asadmin delete-jms-resource --target instance1 .jms/MyQueue
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.6 delete-jmsdest

JMS 物理送信先の削除



## 書式

```
asadmin [asadmin-options] delete-jmsdest [--help]
  --desttype dest_type
  [--target target]
  dest_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-jmsdest` サブコマンドは JMS 物理送信先を削除します。

このサブコマンドはリモートモードだけサポートします。このサブコマンドをリモートモードで実行するには、ドメイン管理サーバが稼働している必要があります。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--desttype dest_type`

JMS 送信先のタイプを指定します。

型：string

指摘できる値を次に示します。

- `topic`
- `queue`

デフォルト値：なし

`--target target`

削除する JMS の物理送信先の対象を指定します。

型：String

指摘できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。

- *cluster\_name*

指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### *dest\_name*

削除対象となる JMS 送信先の一意の識別子を指定します。

型：String

指定できる値を次に示します。

- 物理送信先の名称

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、PhysicalQueue という名称のキューを削除します。

```
asadmin delete-jmsdest --target instance1 --desttype queue PhysicalQueue
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.7 flush-jmsdest

JMS 送信先内のメッセージの削除

### 書式

```
asadmin [asadmin-options] flush-jmsdest [--help]
--desttype {topic|queue} [--target target] destname
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

`flush-jmsdest` サブコマンドは、サーバの JMS 構成にある物理送信先からメッセージを削除します。  
このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--desttype {topic|queue}`

メッセージを削除する物理送信先を指定します。

型：String

指定できる値を次に示します。

- `topic`
- `queue`

デフォルト値：なし

`--target target`

メッセージを削除する物理送信先の対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`dest_name`

削除対象である JMS 送信先の一意的識別子を指定します。

型：String

指定できる値を次に示します。

- JMS 送信先の識別子

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンス上の、PhysicalQueue という名称のキューからメッセージを削除します。

```
asadmin flush-jmsdest --target instance1 --desttype queue PhysicalQueue
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.8 list-jms-hosts

JMS ホストの一覧表示

### 書式

```
asadmin [asadmin-options] list-jms-hosts [--help] [--target target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-jms-hosts サブコマンドは、JMS サービスの既存の JMS ホストを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

`--target target`

一覧表示する JMS ホストの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、`instance1` という名称のサーバインスタンスについて、JMS サービスの JMS ホストを一覧表示します。

```
asadmin list-jms-hosts --target instance1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.9 list-jms-resources

JMS リソースの一覧表示

### 書式

```
asadmin [asadmin-options] list-jms-resources [--help]  
        [--restype type] [target]
```

## 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-jms-resources` サブコマンドは、既存の JMS リソース（送信先とコネクションファクトリーリソース）を一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--restype type`

一覧表示する JMS リソースのタイプを指定します。

型：String

指定できる値を次に示します。

- `javax.jms.Topic`
- `javax.jms.Queue`
- `javax.jms.ConnectionFactory`
- `javax.jms.TopicConnectionFactory`
- `javax.jms.QueueConnectionFactory`

デフォルト値：なし

`target`

一覧表示する JMS リソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、すべての `javax.jms.ConnectionFactory` リソースを一覧表示します。

```
asadmin list-jms-resources --restype javax.jms.ConnectionFactory
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.19.10 list-jmsdest

JMS 物理送信先の一覧表示

### 書式

```
asadmin [asadmin-options] list-jmsdest [--help]
        [--desttype type]
        target
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-jmsdest` サブコマンドは、JMS 物理送信先を一覧表示します。

このサブコマンドはリモートモードだけサポートします。`asadmin` サブコマンドをリモートモードで実行するには、ドメイン管理サーバが動作していることが必要です。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--desttype` *type*

一覧表示する JMS 物理送信先のタイプを指定します。

型：String

指定できる値を次に示します。

- topic
- queue

デフォルト値：queue

#### *target*

一覧表示する JMS 物理送信先の対象を指定します。

型：String

指定できる値を次に示します。

- server  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- configuration\_name  
指定した名称の構成を対象とします。
- cluster\_name  
指定した名称のクラスターを対象とします。
- instance\_name  
指定した名称のサービンスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、物理トピックをすべて一覧表示します。

```
asadmin list-jmsdest --desttype topic
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.20 EIS 接続管理で使用するコマンド

EIS 接続管理で使用するコマンドの書式、機能などについて説明します。

### 2.20.1 create-admin-object

管理対象オブジェクトの作成

#### 書式

```
asadmin [asadmin-options] create-admin-object [--help]
  [--target target] --restype restype [--classname classname]
  --raname raname [--enabled={true|false}]
  [--description description] [--property name=value[:name=value]...]
  jndi_name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`create-admin-object` サブコマンドは、指定した JNDI 名で管理対象オブジェクト、およびリソースアダプターのインターフェース定義を作成します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- `create-admin-object` サブコマンドを実行する前に、リソースアダプターをデプロイしておく必要があります。

#### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

作成する管理対象オブジェクトの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソースの `resource-ref` は指定した対象に対してだけ作成されるため、リソースがドメインで定義されても、リソースは指定した対象だけで使用できるようになります。複数の対象でリソースを参照する場合は、`create-resource-ref` サブコマンドを使用します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。**server** はドメイン管理サーバの名称です。
- **configuration\_name**  
指定した名称の名前付き構成を対象とします。
- **cluster\_name**  
指定した名称のクラスターを対象とします。
- **instance\_name**  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### **--restype *restype***

管理対象オブジェクトのインターフェース定義を指定します。リソース型は、リソースアダプターの **ra.xml** ファイルに指定されるインターフェース定義である必要があります。

型：String

指定できる値を次に示します。

- リソース型

デフォルト値：なし

#### **--classname *classname***

管理対象オブジェクトのクラス名を指定します。複数の管理対象オブジェクトが同じインターフェース定義を使用する場合は必須です。

型：String

指定できる値を次に示します。

- 管理対象オブジェクトのクラス名

デフォルト値：なし

#### **--raname *raname***

この管理対象オブジェクトに関連づけるリソースアダプターの名称を指定します。

型：String

指定できる値を次に示します。

- リソースアダプター名

デフォルト値：なし

#### **--enabled={true|false}**

このオブジェクトを有効にするかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--description description`

管理対象オブジェクトを説明するテキスト文字列を記述します。

型：String

指定できる値を次に示します。

- 作成される管理対象オブジェクトを説明するテキスト

デフォルト値：なし

`--property name=value[:name=value]...`

リソースの構成にプロパティ名と値のペアを記述します。リソースアダプターに依存します。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

*name*

型：String

指定できる値を次に示します。

- リソースアダプターがサポートするプロパティ名

デフォルト値：なし

*value*

指定できる値を次に示します。

- *name* に指定したプロパティの指定範囲に従います。

*jndi\_name*

作成する管理対象オブジェクトの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- 作成される管理対象オブジェクトの JNDI 名

デフォルト値：なし

## 使用例

次の例では、*jmsra* は、管理対象オブジェクトのインターフェースである *javax.jms.Queue* および *javax.jms.Topic* を持つ、Message Queue のリソースアダプターを作成します。

この *jmsra* リソースアダプターは、*instance1* という名称のサービンスタンスに作成されます。

```
asadmin create-admin-object --target instance1 --restype javax.jms.Queue
--rename jmsra --description "sample administered object"
--property Name=sample_jmsqueue jms/samplequeue
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.2 create-connector-connection-pool

コネクションプールの追加

### 書式

```
asadmin [asadmin-options] create-connector-connection-pool [--help]
--rename raname
--connectiondefinition connectiondefinitionname
[--steadypoolsize steadypoolsize]
[--maxpoolsize maxpoolsize]
[--maxwait maxwait]
[--poolresize poolresize]
[--idletimeout idletimeout]
[--isconnectvalidatereq={false|true}]
[--leaktimeout=timeout]
[--leakreclaim={false|true}]
[--creationretryattempts=attempts]
[--creationretryinterval=interval]
[--lazyconnectionenlistment={false|true}]
[--lazyconnectionassociation={false|true}]
[--matchconnections={true|false}]
[--maxconnectionusagecount=count]
[--validateatmostonceperiod=interval]
[--transactionssupport transactionssupport]
[--description description]
[--ping {false|true}]
[--pooling {true|false}]
[--property name=value[:name=value]...]
poolname
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`create-connector-connection-pool` サブコマンドは、エンタープライズ情報システム (EIS) への接続のプールを定義します。名前付きプールは複数のコネクタリソースから参照できます。

定義された各プールは、サーバの起動時にインスタンス化され、最初にアクセスされたときに読み込まれます。

同じコネクタ接続プールに 2 つ以上のコネクタリソースがポイントする場合、実行時に同じ接続プールが使用されます。

1 つのリソースアダプター内の接続定義に対して複数のプールを指定できます。

認証付きのコネクタ接続プールは、XML ディスクリプターファイルで接続情報を指定することによって作成できます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--connectiondefinition connectiondefinitionname`

接続定義の名称を指定します。

型: String

指定できる値を次に示します。

- コネクション定義名

デフォルト値: なし

`--creationretryattempts=attempts`

サーバが最初の接続の確立に失敗した場合に、最大何回まで接続確立を再試行するかを指定します。

型: Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値: 0

`--creationretryinterval=interval`

接続確立を連続して試行するときの間隔を秒単位で指定します。

--creationretryattempts が 0 の場合、--creationretryinterval オプションは無視されます。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：10

--description *description*

コネクターコネクションプールの詳しい説明文を記述します。

型：String

指定できる値を次に示します。

- テキストでの説明

デフォルト値：なし

--idletimeout *idletimeout*

コネクションをアイドル状態で維持できる最長時間を秒単位で指定します。

型：Integer

指定できる値を次に示します。

- 1~2147483647

デフォルト値：300

--isconnectvalidatereq={false|true}

コネクション障害検知機能を使用するかどうかを指定します。true の場合、コネクション障害検知機能を使用します。一定間隔のコネクション障害検知については、指定値に関係なく常に有効になります。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

--lazyconnectionenlistment={false|true}

メソッドがリソースを実際に使用する場合、そのリソースをトランザクションに参加させるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--lazyconnectionassociation={false|true}`

物理コネクションを使用する場合、その物理コネクションを論理コネクションに関連づけるべきかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- **true**

物理コネクションは、物理コネクションを使用する場合にだけ論理コネクションに関連づけられます。トランザクションが完了すると、関連づけは行われません。`--lazyconnectionenlistment` オプションも **true** に設定される必要があります。

- **false**

物理コネクションは、物理コネクションを使用する前から論理コネクションに関連づけられます。トランザクションが完了すると、関連づけは行われません。

デフォルト値：false

`--leakreclaim={false|true}`

リークコネクションの追跡が完了した後で、リークコネクションをコネクションプールに復元するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- **true**

リークコネクションはコネクションプールに復元されます。

- **false**

リークコネクションはコネクションプールに復元されません。

デフォルト値：false

`--leaktimeout=timeout`

コネクションプール内のコネクションリークを追跡する時間を秒単位で指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--matchconnections={false|true}`

プールから選択したコネクションがリソースアダプターと適合する必要があるかどうかを指定します。

プール内のすべてのコネクションが同一の場合、コネクションとリソースアダプターが適合する必要はありません。

型：Boolean

指定できる値を次に示します。

- **true**

コネクションはリソースアダプターと適合する必要があります。

- `false`

コネクションはリソースアダプターと適合する必要はありません。

デフォルト値：`true`

`--maxconnectionusagecount=count`

コネクションを再使用できる最大回数を指定します。

型：`Integer`

指定できる値を次に示します。

- `0~2147483647`

デフォルト値：`0`

`--maxpoolsize maxpoolsize`

クライアントのリクエストを満たすために作成できるコネクションの最大数を指定します。

型：`Integer`

指定できる値を次に示します。

- `1~2147483647`

デフォルト値：`24`

`--maxwait maxwait`

コネクションが使用できない場合、呼び出し側がコネクション確立までに待機する必要がある時間をミリ秒単位で指定します。

値`0`は、呼び出し側を無期限に待機させます。

型：`Integer`

指定できる値を次に示します。

- `0~2147483647`

デフォルト値：`60000`

`--ping {false|true}`

作成時、この属性が`true`に設定されているプールが使用できるかどうかをテストします。

型：`Boolean`

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--pooling {true|false}`

`false`に設定されている場合にコネクションプーリングを無効にします。

型：`Boolean`



指定できる値を次に示します。

- true
- false

デフォルト値：true

`--poolresize poolresize`

プールがコネクション数を増加する量または減少する量を指定します。

型：Integer

指定できる値を次に示します。

- 1~2147483647

デフォルト値：2

`--property name=value[:name=value]...`

プールを構成するためのオプションのプロパティ名と値のペアを指定します。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

このオプションを使用することで、ManagedConnectionFactory クラスが保持している設定値を、JavaBeans 仕様に基き上書きできます。プロパティ名は、ra.xml のmanagedconnectionfactory-class 要素に指定したManagedConnectionFactory クラスのsetter に対応する JavaBeans 仕様に従ったプロパティ名を指定します。ra.xml のconfig-property 要素とこのオプションに同じプロパティを設定した場合、このオプションの設定が優先されます。

型：String

デフォルト値：なし

`--raname raname`

リソースアダプターの名称を指定します。

型：String

指定できる値を次に示します。

- リソースアダプター名

デフォルト値：なし

`--steadypoolsize steadypoolsize`

プールに保持する初期コネクション数と最小コネクション数を指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：24

`--transactionsupport transactionsupport`

このプールでのトランザクションサポートのレベルを確認します。

型：String

指定できる値を次に示します。

- XATransaction
- LocalTransaction
- NoTransaction

`--validateatmostonceperiod=interval`

連続するリクエスト間で、コネクションを検証する間隔を秒単位で指定します。

0 を指定した場合、常にコネクションを検証します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

*poolname*

作成するコネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- プール名

デフォルト値：なし

## 使用例

次の例では、コネクタコネクションプールを作成します。

```
asadmin create-connector-connection-pool --raname jmsra
--connectiondefinition javax.jms.QueueConnectionFactory --steadypoolsize 20
--maxpoolsize 100 --poolresize 2 --maxwait 60000 jms/qConnPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.3 create-connector-resource

コネクタリソースの登録

## 書式

```
asadmin [asadmin-options] create-connector-resource [--help]
  --poolname connectorConnectionPoolName [--enabled={true|false}]
  [--description description] [--objecttype objecttype]
  [--target target] jndi_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`create-connector-resource` サブコマンドは、指定した JNDI 名でコネクタリソースを登録します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--poolname connectorConnectionPoolName`

コネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクションプール名

デフォルト値：なし

`--enabled={true|false}`

実行時にリソースを有効にするかどうかを確認します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--objecttype objecttype`

コネクタリソースのタイプを定義します。

型：String

指定できる値を次に示します。

- **system-all**  
すべてのサーバインスタンスおよびドメイン管理サーバに対して有効なシステムリソース。
- **system-admin**  
ドメイン管理サーバにだけ有効なシステムリソース。
- **system-instance**  
すべてのサーバインスタンスにだけ有効なシステムリソース。
- **user**  
ユーザーリソース。

デフォルト値：user

#### **--description** *description*

コネクターリソースの詳しい説明文を表示します。

型：String

指定できる値を次に示します。

- テキストでの説明

デフォルト値：なし

#### **--target** *target*

作成するコネクターリソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソースの **resource-ref** は指定した対象に対してだけ作成されるため、リソースがドメインで定義されても、リソースは指定した対象だけで使用できるようになります。複数の対象でリソースを参照する場合は、**create-resource-ref** サブコマンドを使用します。

型：String

指定できる値を次に示します。

- **server**  
ドメイン管理サーバを対象とします。 **server** はドメイン管理サーバの名称です。
- **domain**  
ドメインを対象とします。
- **cluster\_name**  
指定した名称のクラスターを対象とします。
- **instance\_name**  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

#### *jndi\_name*

このコネクターリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、コネクタリソースを作成します。

```
asadmin create-connector-resource --target instance1 --poolname jms/qConnPool
--description "sample connector resource" jms/qConnFactory
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.4 create-connector-security-map

コネクタコネクションプールのセキュリティーマップの作成

### 書式

```
asadmin [asadmin-options] create-connector-security-map [--help]
--poolname connector_connection_pool_name
[--principals principal-name1[,principal-name2]...]
[--usergroups user-group1[,user-group2]...]
--mappedusername user-name mapname
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`create-connector-security-map` サブコマンドは、指定したコネクタコネクションプールのセキュリティーマップを作成します。セキュリティーマップがない場合、新しいマップが作成されます。

このサブコマンドは、アプリケーションの呼び出し側識別（プリンシパルまたはユーザーグループ）をコンテナー管理による認証シナリオで、対応するエンタープライズ情報システム（EIS）プリンシパルにマップすることもできます。

コネクタコネクションプールには 1 つ以上の名前付きセキュリティーマップを関連づけることができます。

コネクターセキュリティーマップ構成では、ワイルドカードのアスタリスク (\*) を使用して、すべてのユーザーまたはすべてのユーザーグループを示すことができます。

EIS パスワードを指定する場合、AS\_ADMIN\_MAPPEDPASSWORD エントリーをパスワードファイルに追加してから --passwordfile asadmin オプションを使用して、ファイルを指定できます。

このサブコマンドはリモートモードだけサポートします。

このサブコマンドを正常に実行するには、コネクターコネクションプールを create-connector-connection-pool サブコマンドを使用して作成する必要があります。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--poolname *connector\_connection\_pool\_name*

セキュリティーマップが属すコネクターコネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクションプール名

デフォルト値：なし

--principals *principal-name1[,principal-name2]...*

バックエンド EIS プリンシパルのリストを指定します。

同じコマンドでは --principals オプションまたは --usergroups オプションのどちらかを使用し、両方のオプションは使用しません。

複数のプリンシパルはコンマ区切りのリストで指定できます。

型：String

指定できる値を次に示します。

- バックエンド EIS 主体名

デフォルト値：なし

--usergroups *user-group1[,user-group2]...*

バックエンド EIS ユーザーグループのリストを指定します。

同じコマンドでは --principals オプションまたは --usergroups オプションのどちらかを使用し、両方のオプションは使用しません。

複数のユーザーグループはコンマ区切りのリストで指定できます。

型：String

指定できる値を次に示します。

- EIS ユーザーグループ

デフォルト値：なし

`--mappedusername user-name`

EIS ユーザー名を指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：なし

*mapname*

作成するセキュリティーマップの名称を指定します。

型：String

指定できる値を次に示します。

- マップ名

デフォルト値：なし

## 使用例

次の例では、コネクタセキュリティーマップを作成します。

```
asadmin create-connector-security-map --poolname connector-pool1
--principals principal1,principal2 --mappedusername backend-username
securityMap1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.5 create-connector-work-security-map

リソースアダプターのワークセキュリティーマップの作成

### 書式

```
asadmin [asadmin-options] create-connector-work-security-map [--help]
--raname raname [--principalsmap eis-principal1=principal_name1
[,eis-principal2=principal_name2]. . . ]
```

```
--groupsmap eis-group1=server-group1[,eis-group2=server-group2]...]  
mapname
```

## 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

## 機能

create-connector-work-security-map サブコマンドは、リソースアダプター EIS プリンシパルまたは EIS ユーザーグループによってサブミットされたワークの呼び出し側識別を、Java EE Server セキュリティードメインに適しているプリンシパルまたはユーザーグループにマップします。

リソースアダプターには 1 つ以上のワークセキュリティーマップを関連づけることができます。

コネクタワークセキュリティーマップの構成では、ワイルドカードのアスタリスク (\*) を使用して、すべてのユーザーまたはすべてのユーザーグループを示すことができます。

エンタープライズ情報システム (EIS) は、組織のデータを保持するあらゆるシステムです。メインフレーム、メッセージングシステム、データベースシステム、またアプリケーションなどが該当します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--groupsmap eis-group1=server-group1[,eis-group2=server-group2]...

バックエンド EIS ユーザーグループの Java EE Server ユーザーグループへのマッピングを指定します。複数のマッピングを指定する場合は、コンマ区切りのリストを使用します。

--principalsmap オプションまたは--groupsmap オプションのどちらかを使用し、両方のオプションは使用しません。

型: String

指定できる値を次に示します。

- バックエンド EIS のマップ名

デフォルト値: なし

--principalsmap eis-principal1=principal\_name1[,eis-principal2=principal\_name2]...

バックエンド EIS プリンシパルの Java EE Server のプリンシパルへのマッピングを指定します。複数のマッピングを指定する場合は、コンマ区切りのリストを使用します。



--principalsmap オプションまたは--groupsmap オプションのどちらかを使用し、両方のオプションは使用しません。

型：String

指定できる値を次に示します。

- 主体マップ名

デフォルト値：なし

--rename *raname*

リソースアダプターの名称であるリソースアダプターアーカイブ名を指定します。

型：String

指定できる値を次に示します。

- リソースアダプターアーカイブ名

デフォルト値：なし

*mapname*

作成するワークセキュリティーマップの名称を指定します。

型：String

指定できる値を次に示します。

- マップ名

デフォルト値：なし

## 使用例

次の例では、コネクタワークセキュリティーマップを作成します。

```
asadmin create-connector-work-security-map --rename my-resource-adapter
--principalsmap eis-principal-1=server-principal-1,eis-principal-2
=server-principal-2,eis-principal-3=server-principal-1 workSecurityMap1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.6 create-resource-adapter-config

リソースアダプターアーカイブ構成情報の作成

## 書式

```
asadmin [asadmin-options] create-resource-adapter-config [--help]
  [--objecttype object-type] raname
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`create-resource-adapter-config` サブコマンドはリソースアダプターアーカイブ構成情報を作成します。このサブコマンドは、リソースアダプターを配置する前に実行できます。つまり構成情報は、デプロイ時に利用できます。リソースアダプターアーカイブ構成は、リソースアダプターが配置された後でも作成できます。この場合、リソースアダプターは新しい構成で再起動されます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

*raname* は、`domain.xml` ファイルの `resource-adapter-name` の値です。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--objecttype object-type`

オブジェクトのタイプを指定します。

型：String

指定できる値を次に示します。

- オブジェクトの型名

デフォルト値：user

*raname*

リソースアダプターアーカイブ名を指定します。`domain.xml` ファイルの `resource-adapter-name` の値です。

型：String

指定できる値を次に示します。

- リソースアダプターアーカイブ名

デフォルト値：なし

## 使用例

次の例では、ra1 のリソースアダプターアーカイブ構成情報を作成します。

```
asadmin create-resource-adapter-config ra1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.7 delete-admin-object

管理対象オブジェクトの削除

### 書式

```
asadmin [asadmin-options] delete-admin-object [--help]  
        [--target target] jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-admin-object` サブコマンドは、指定した JNDI 名の管理対象オブジェクトを削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除する管理対象オブジェクトの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソース作成時に `--target` オプションを使用して、`resource-ref` が作成された対象だけがアクティブになります。

リソースの削除は、`--target` オプションで指定した対象の `resource-ref` だけを削除します。削除の対象としてドメインが指定されない限り、ドメイン全体からそのリソースは削除されません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *jndi\_name*

管理対象オブジェクトの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- 管理対象オブジェクトの JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`jdbc/samplequeue` という名称の管理対象オブジェクトを削除します。

```
asadmin delete-admin-object --target instance1 jdbc/samplequeue
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.8 delete-connector-connection-pool

コネクタコネクションプールの削除

### 書式

```
asadmin [asadmin-options] delete-connector-connection-pool [--help]
        [--cascade={false|true}] poolname
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-connector-connection-pool` サブコマンドは、指定したコネクタコネクションプールの削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--cascade={false|true}`

コネクションプールとそのプールに関連づけられているすべてのコネクタリソースを削除します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

*poolname*

コネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクションプール名

デフォルト値：なし

## 使用例

次の例では、jdbc/qConnPool という名称のコネクターコネクションプールを削除します。

```
asadmin delete-connector-connection-pool --cascade=false jdbc/qConnPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.9 delete-connector-resource

コネクターリソースの削除

### 書式

```
asadmin [asadmin-options] delete-connector-resource [--help]  
        [--target target] jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-connector-resource サブコマンドは、指定した JNDI 名のコネクターリソースを削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストが表示されます。

--target *target*

削除するコネクターリソースの対象を指定します。

リソースは常にドメイン全体に対して作成されますが、リソース作成時に--target オプションを使用して、resource-ref が作成された対象だけがアクティブになります。

リソースの削除は、`--target` オプションで指定した対象の `resource-ref` だけを削除します。削除の対象としてドメインが指定されない限り、ドメイン全体からそのリソースは削除されません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *jndi\_name*

コネクタリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- コネクタリソースの JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`jdbc/qConnFactory` という名称のコネクタリソースを削除します。

```
asadmin delete-connector-resource --target instance1 jdbc/qConnFactory
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.10 delete-connector-security-map

コネクタコネクションプールのセキュリティーマップの削除

## 書式

```
asadmin [asadmin-options] delete-connector-security-map [--help]
--poolname connector_connection_pool_name mapname
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-connector-security-map` サブコマンドは、指定したコネクタコネクションプールのセキュリティマップを削除します。

このサブコマンドを正常に実行するには、最初に`create-connector-connection-pool` サブコマンドを使用してコネクタコネクションプールを作成する必要があります。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

`--poolname` オプションを使用するには、コネクタコネクションプールを作成しておく必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--poolname` *connector\_connection\_pool\_name*

削除するセキュリティマップが属するコネクタコネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクタコネクションプール名

デフォルト値：なし

*mapname*

セキュリティマップの名称を指定します。

型：String

指定できる値を次に示します。

- セキュリティマップ名

デフォルト値：なし



## 使用例

次の例では、connector-pool1 という名称の既存のコネクションプールに属するsecurityMap1 を削除します。

```
asadmin delete-connector-security-map --poolname connector-pool1 securityMap1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.11 delete-connector-work-security-map

リソースアダプターのワークセキュリティーマップの削除

### 書式

```
asadmin [asadmin-options] delete-connector-work-security-map [--help]
--raname raname mapname
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-connector-work-security-map サブコマンドは、指定したリソースアダプターに関連づけられているワークセキュリティーマップを削除します。このサブコマンドを正常に実行するには、最初に指定したリソースアダプターを作成してデプロイする必要があります。

エンタープライズ情報システム (EIS) は組織のデータを保持するシステムです。メインフレーム、メッセージングシステム、データベースシステム、またアプリケーションなどが該当します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

--raname オプションを使用するには、指定したリソースアダプターを作成し、かつデプロイしておく必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--rename raname`

ワークセキュリティーマップが関連づけられているリソースアダプターアーカイブ名を確認します。

型：String

指定できる値を次に示します。

- リソースアダプターアーカイブ名

デフォルト値：なし

*mapname*

ワークセキュリティーマップの名称を指定します。

型：String

指定できる値を次に示します。

- セキュリティーマップ名

デフォルト値：なし

## 使用例

次の例では、`ra_name` という名称のリソースアダプターに関連する `work_security_map_name` という名称のワークセキュリティーマップを削除します。

```
asadmin delete-connector-work-security-map --rename ra_name
work_security_map_name
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.12 delete-resource-adapter-config

リソースアダプターアーカイブ構成情報の削除

## 書式

```
asadmin [asadmin-options] delete-resource-adapter-config [--help]
raname
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-resource-adapter-config` サブコマンドは、リソースアダプターアーカイブの構成情報を削除します。

このコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*raname*

リソースアダプターアーカイブ名を指定します。

型：String

指定できる値を次に示します。

- リソースアダプターアーカイブ名

デフォルト値：なし

## 使用例

次の例では、`ra1` のリソースアダプターアーカイブの構成情報を削除します。

```
asadmin delete-resource-adapter-config ra1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.13 flush-connection-pool

コネクションプールで確立されたコネクションの再初期化

## 書式

```
asadmin [asadmin-options] flush-connection-pool [--help]
  [--appname application [--modulename module]]
  [--target target] pool_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`flush-connection-pool` サブコマンドは、JDBC コネクションプールまたはコネクタコネクションプールを初期状態にリセットします。このサブコマンドを実行すると、未使用のコネクションは削除されます。そのあと、プールの初期コネクションが再作成され、プールは通常プールサイズに復元されます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--appname application`

アプリケーションスコープのリソースが定義されているアプリケーション名を指定します。

型：String

指定できる値を次に示します。

- アプリケーション名

デフォルト値：なし

`--modulename module`

モジュールスコープのリソースが定義されているモジュール名を指定します。

型：String

指定できる値を次に示します。

- モジュール名

デフォルト値：なし

`--target target`

再初期化するコネクションプールの対象を指定します。

サーバインスタンス作成後、一度も起動していないサーバインスタンス名を指定した場合、コマンドの実行に成功します。

引数で指定するプール名に依存しません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *pool\_name*

再初期化するコネクシオンプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクシオンプール名

## 使用例

次の例では、`instance1` という名称のサーバインスタンス上の、`__TimerPool` という名称の JDBC コネクシオンプールを再初期化します。

```
asadmin flush-connection-pool --target instance1 __TimerPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.14 list-admin-objects

管理された全オブジェクトの取得

### 書式

```
asadmin [asadmin-options] list-admin-objects [--help]  
[target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-admin-objects` サブコマンドは、管理された全オブジェクトの一覧を表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する管理対象オブジェクトの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：*server*

## 使用例

次の例では、すべての管理対象オブジェクトを一覧表示します。

```
asadmin list-admin-objects
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.15 list-connector-connection-pools

コネクタコネクションプールの一覧表示

### 書式

```
asadmin [asadmin-options] list-connector-connection-pools [--help]
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-connector-connection-pools サブコマンドは、作成されたコネクタコネクションプールを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

### 使用例

次の例では、既存のコネクタコネクションプールを一覧表示します。

```
asadmin list-connector-connection-pools
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.16 list-connector-resources

コネクタリソースの一覧表示

### 書式

```
asadmin [asadmin-options] list-connector-resources [--help]  
[target]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-connector-resources` サブコマンドはすべてのコネクタリソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するコネクタリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。



- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では既存のコネクターリソースをすべて一覧表示します。

```
asadmin list-connector-resources
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.17 list-connector-security-maps

コネクターコネクションプールに属するセキュリティーマップの一覧表示

### 書式

```
asadmin [asadmin-options] list-connector-security-maps [--help]
  [--securitymap securitymap]
  [--verbose={false|true}]
  pool-name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

list-connector-security-maps サブコマンドは、指定されたコネクターコネクションプールに属するセキュリティーマップを一覧表示します。このサブコマンドを正常に実行するには、先にcreate-connector-connection-pool サブコマンドを使用してコネクターコネクションプールを作成する必要があります。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--securitymap securitymap`

ID とプリンシパルを一覧表示する、コネクタコネクションプール内に含まれるセキュリティーマップの名称を指定します。

このオプションを指定した場合、`--verbose` オプションの指定に関係なく、詳細情報が表示されます。

型：String

指定できる値を次に示します。

- セキュリティーマップ名

デフォルト値：なし

`--verbose={false|true}`

ID、プリンシパルおよびセキュリティー名を含む詳細情報を一覧表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`pool-name`

セキュリティーマップを一覧表示する、コネクタコネクションプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクタコネクションプール名

デフォルト値：なし

## 使用例

次の例では、`connector-Pool1` というプール名のコネクタコネクションプールに属するセキュリティーマップを一覧表示します。

```
asadmin list-connector-security-maps connector-Pool1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.18 list-connector-work-security-maps

リソースアダプターに属するワークセキュリティーマップの一覧表示

### 書式

```
asadmin [asadmin-options] list-connector-work-security-maps [--help]
  [--securitymap securitymap]
  resource_adapter_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-connector-work-security-maps` サブコマンドは、指定されたリソースアダプターに属するワークセキュリティーマップを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--securitymap securitymap`

ID とプリンシパルを一覧表示する、リソースアダプター内に含まれるセキュリティーマップの名称を指定します。

型：String

指定できる値を次に示します。

- セキュリティーマップ名

デフォルト値：なし

### *resource\_adapter\_name*

セキュリティーマップを一覧表示する、リソースアダプターの名称を指定します。

型：String

指定できる値を次に示します。

- リソースアダプター名

デフォルト値：なし

## 使用例

次の例では、`my_resource_adapter` という名称のリソースアダプターについて、現在のワークセキュリティーマップを一覧表示します。

```
asadmin list-connector-work-security-maps my_resource_adapter
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.19 list-resource-adapter-configs

リソースアダプターアーカイブ構成の名称の一覧表示

### 書式

```
asadmin [asadmin-options] list-resource-adapter-configs [--help]  
        [--raname raname] [--verbose={false|true}]
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-resource-adapter-configs` サブコマンドは、`domain.xml` 内のリソースアダプターアーカイブ構成情報を一覧表示します。`domain.xml` ファイルの `resource-adapter-config` というエントリーが一覧表示されます。`--raname` オプションを指定した場合、指定したリソースアダプターアーカイブ構成だけが一覧表示されます。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--rename raname`

リソースアダプターアーカイブ名を指定します。

型：String

指定できる値を次に示します。

- リソースアダプターアーカイブ名

デフォルト値：なし

`--verbose={false|true}`

構成されたプロパティを一覧表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

## 使用例

次の例では、現在のリソースアダプターアーカイブ構成情報を一覧表示します。

```
asadmin list-resource-adapter-configs
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.20 ping-connection-pool

JDBC またはコネクタコネクションプールの検証

## 書式

```
asadmin [asadmin-options] ping-connection-pool [--help]
  [--appname application [--modulename module]]
  [--target target] pool_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

ping-connection-pool サブコマンドは、既存の JDBC またはコネクタコネクションプールが使用できるかどうかをテストします。ping-connection-pool サブコマンドは、glassfish-resources.xml で定義したアプリケーションスコープまたはモジュールスコープのリソースも対象にできます。

このサブコマンドはリモートモードだけサポートします。

アプリケーションスコープまたはモジュールスコープのリソースの JNDI 名は、java:app/jdbc/myDataSource または java:module/jdbc/myModuleLevelDataSource の形式を使用して指定されます。この命名範囲は、Java EE 6 仕様で定義されています。

## 前提条件

- ドメイン管理サーバは、稼働状態である必要があります。
- コネクションプールが使用できるかどうかをテストする前に、認証されたコネクションプールを作成し、サーバまたはデータベースが起動されていることを確認する必要があります。

## ファイル

ping-connection-pool サブコマンドは、glassfish-resources.xml で定義したアプリケーションスコープまたはモジュールスコープのリソースも対象にできます。

## 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--appname *application*

アプリケーションスコープのリソースが定義されているアプリケーションの名称を指定します。

アプリケーションスコープのリソースの JNDI 名を参照するには、java:app プレフィックスを使用してルックアップを実行します。

型：String

指定できる値を次に示します。

- アプリケーションの名称

デフォルト値：なし

**--modulename *module***

モジュールスコープリソースが定義されているモジュールの名称を指定します。

モジュールスコープリソースの JNDI 名を参照するには、`java:module` プレフィックスを使用してルックアップを実行します。

型：String

指定できる値を次に示します。

- モジュールの名称

デフォルト値：なし

**--target *target***

このサブコマンドの対象を指定します。

サーバインスタンス作成後、一度も起動していないサーバインスタンス名を指定した場合、コマンドの実行に成功します。

引数で指定するプール名に依存しません。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

***pool\_name***

テストするコネクシオンプールの名称を指定します。

型：String

指定できる値を次に示します。

- コネクシオンプールの名称

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンス上の、`DerbyPool` という名称のコネクシオンプールが使用できるかどうかをテストします。

```
asadmin ping-connection-pool --target instance1 DerbyPool
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.21 update-connector-security-map

コネクタコネクションプールのセキュリティーマップの変更

### 書式

```
asadmin [asadmin-options] update-connector-security-map [--help]
  --poolname connector_connection_pool_name
  [--addprincipals principal_name1[,principal_name2]...]
  [--addusergroups user_group1[,user_group2]...]
  [--removeprincipals principal_name1[,principal_name2]...]
  [--removeusergroups user_group1[,user_group2]...]
  [--mappedusername username] mapname
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

update-connector-security-map サブコマンドは、指定したコネクタコネクションプールのセキュリティーマップを変更します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

コネクタコネクションプールが存在している必要があります。

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--poolname *connector\_connection\_pool\_name*

更新するセキュリティーマップが属すコネクタコネクションプールの名称を指定します。

型：String



指定できる値を次に示します。

- コネクションプール名

デフォルト値：なし

`--addprincipals principal_name1[,principal_name2]....`

追加する EIS 固有のプリンシパルをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- プリンシパル名

デフォルト値：なし

`--addusergroups user_group1[,user_group2]....`

追加する EIS ユーザーグループをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- グループ名

デフォルト値：なし

`--removeprincipals principal_name1[,principal_name2]....`

削除する EIS 固有のプリンシパルをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- プリンシパル名

デフォルト値：なし

`--removeusergroups user_group1[,user_group2]....`

削除する EIS ユーザーグループをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- グループ名

デフォルト値：なし

`--mappedusername username`

EIS ユーザー名を指定します。

型：String

指定できる値を次に示します。

- ユーザー名

デフォルト値：なし

*mapname*

更新するセキュリティーマップの名称を指定します。

型：String

指定できる値を次に示します。

- アプリケーションモジュール名

デフォルト値：なし

## 使用例

次の例では、`securityMap1` という名称の既存のセキュリティーマップにプリンシパルを追加します。

```
asadmin update-connector-security-map --poolname connector-pool1
--addprincipals principal1,principal2 securityMap1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.20.22 update-connector-work-security-map

リソースアダプターのワークセキュリティーマップの変更

### 書式

```
asadmin [asadmin-options] update-connector-work-security-map [--help]
--rename raname
[--addprincipals eis-principal1=server-principal1
[, eis-principal2=server-principal2]....]
--addgroups eis-group1=server-group1
[, eis-group2=server-group2]....]
[--removeprincipals eis-principal1[,eis-principal2]....]
[--removegroups eis-group1[, eis-group2]....] mapname
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`update-connector-work-security-map` サブコマンドは、指定したリソースアダプターのセキュリティーマップを変更します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

コネクターコネクションプールが存在している必要があります。

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--addgroups eis-group1=server-group1 [, eis-group2=server-group2]...`

追加する EIS グループをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- グループ名

デフォルト値：なし

`--addprincipals eis-principal1=server-principal1 [, eis-principal2=server-principal2]...`

追加する EIS 固有のプリンシパルをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- プリンシパル名

デフォルト値：なし

`--removegroups eis-group1 [, eis-group2]...`

削除する EIS グループをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- グループ名

デフォルト値：なし

`--removeprincipals eis-principal1 [, eis-principal2]...`

削除する EIS 固有のプリンシパルをコンマ区切りのリストで指定します。

型：String

指定できる値を次に示します。

- プリンシパル名

デフォルト値：なし

`--rename raname`

ワークセキュリティーマップが関連づけられているリソースアダプターアーカイブ名を指定します。

型：String

指定できる値を次に示します。

- リソースアダプターアーカイブ名

デフォルト値：なし

`mapname`

更新するワークセキュリティーマップの名称を指定します。

型：String

指定できる値を次に示します。

- ワークセキュリティーマップ名

デフォルト値：なし

## 使用例

次の例では、ワークセキュリティーマップからプリンシパルを削除します。

```
asadmin update-connector-work-security-map --rename generic-ra
--removeprincipals eis-foo generic-ra-principals-map
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.21 スレッドプール管理で使用するコマンド

スレッドプール管理で使用するコマンドの書式、機能などについて説明します。

### 2.21.1 create-threadpool

スレッドプールの追加

#### 書式

```
asadmin [asadmin-options] create-threadpool [--help] [--target target]  
  [--maxthreadpoolsize maxthreadpoolsize]  
  [--minthreadpoolsize minthreadpoolsize]  
  [--idletimeout idletimeout] [--maxqueueusize maxqueueusize]  
  threadpool-id
```

#### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`create-threadpool` サブコマンドは、指定名でスレッドプールを作成します。プールの最小および最大スレッド数、メッセージ数、およびスレッドのタイムアウト時間を指定できます。

作成したスレッドプールは、IIOP リクエスト処理と、リソースアダプター処理で利用します。スレッドプールは複数リソースアダプターで使用できます。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

作成するスレッドプールの対象を指定します。

型：String

指定できる値を次に示します。

- `server`

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--maxthreadpoolsize** *maxthreadpoolsize*

プールに保持できる最大スレッド数を指定します。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：5

**--minthreadpoolsize** *minthreadpoolsize*

プールの最小スレッド数を指定します。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：2

**--idletimeout** *idletimeout*

使われていないスレッドをプールから削除するまでの時間を秒で指定します。

型：Integer

指定できる値を次に示します。

- 1～2147483647

デフォルト値：900

**--maxqueue size** *maxqueue size*

ネットワークリスナーまたは IIOP リスナーのスレッドがメッセージを処理できるまで、キューに保持できるメッセージの最大数を指定します。値に-1を指定すると、メッセージの数は制限されません。

型：Integer

指定できる値を次に示します。

- -1 および1～2147483647

デフォルト値：4096

### *threadpool-id*

作業キューの ID を指定します。

型：String

指定できる値を次に示します。

- スレッドプール ID

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、threadpool-1 という ID のスレッドプールを新規に作成します。

```
asadmin create-threadpool --target instance1 --maxthreadpoolsize 100
--minthreadpoolsize 20 --idletimeout 2 threadpool-1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.21.2 delete-threadpool

スレッドプールの削除

### 書式

```
asadmin [asadmin-options] delete-threadpool [--help]
        [--target target] threadpool-id
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

delete-threadpool サブコマンドは、指定した ID のスレッドプールを削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するスレッドプールの対象を指定します。

指定できる値を次に示します。

型：String

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `configuration_name`  
指定した名称の構成を対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`threadpool-id`

スレッドプールの ID を指定します。

型：String

- スレッドプールの ID

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`threadpool-1` という名称のスレッドプールを削除します。

```
asadmin delete-threadpool --target instance1 threadpool-1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.21.3 list-threadpools

スレッドプールの一覧表示

### 書式

```
asadmin [asadmin-options] list-threadpools [--help]
        target
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-threadpools サブコマンドは Java EE Server のスレッドプールを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するスレッドプールの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *configuration\_name*  
指定した名称の構成を対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、デフォルトのサーバインスタンスにある現在のスレッドプールを一覧表示します。

```
asadmin list-threadpools server
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.22 トランザクション管理で使用するコマンド

トランザクション管理で使用するコマンドの書式、機能などについて説明します。

### 2.22.1 recover-transactions

実行待ちトランザクションの手動での回復

#### 書式

```
asadmin [asadmin-options] recover-transactions [--help]
  [--transactionlogdir transaction_log_dir]
  [--target target] server_name
```

#### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

#### 機能

recover-transactions サブコマンドは、サーバ障害後にトランザクションを回復します。

複数サーバインスタンス環境の場合、残存しているサーバインスタンスにrecover-transactions サブコマンドを実行できます。このサブコマンドを使用するには、次の条件を満たす必要があります。

- 代理トランザクション回復が無効化されている。
- トランザクションサービスログが、すべてのサーバインスタンスがアクセスできる共有ファイルシステムまたはデータベースに格納されている。

スタンドアロンのサーバインスタンスの場合、サーバ障害後にトランザクションを回復するためにこのサブコマンドを使用しないでください。

このサブコマンドはリモートモードだけサポートします。

スタンドアロンのサーバインスタンスの場合、リソースに障害が発生してもサーバが稼働中の場合にだけ、recover-transactions サブコマンドがトランザクションを回復できます。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

複数サーバインスタンス環境の場合、サーバ障害後に残存しているサーバインスタンスにrecover-transactions サブコマンドを実行してトランザクションを回復できます。このように使用するには、次の条件を満たす必要があります。

- 代理トランザクション回復が無効化されている。
- トランザクションサービスログが、すべてのサーバインスタンスがアクセスできる共有ファイルシステムまたはデータベースに格納されている。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--transactionlogdir transaction_log_dir`

トランザクションの回復が要求されたサーバのトランザクションのステータスファイルの場所を指定します。

このオプションは、共有ファイルシステム上にトランザクションのステータスファイルが格納されている場合にだけ適用されます。

型：String

指定できる値を次に示します。

- ログの場所

デフォルト値：なし

`--target target`

*server\_name* で指定されたサーバの回復を実行する対象サーバを指定します。

型：String

指定できる値を次に示します。

- 対象サーバインスタンスの名称

デフォルト値：なし

*server\_name*

サーバ名を指定します。スタンドアロンのサーバインスタンス用です。

このサーバが稼働している場合は、同じサーバによって回復が実行されます。この場合、`--transactionlogdir` および `--target` オプションを省略する必要があります。

サーバが稼働していない場合、`--target` オプションが必要です。また、共有ファイルシステム上にトランザクションのステータスファイルが格納されている場合は `--transactionlogdir` オプションも必要です。

型：String

指定できる値を次に示します。

- サーバインスタンスの名称

デフォルト値：

スタンドアロンのサーバインスタンスの場合、`server` です。

複数サーバインスタンス環境の場合、回復が必要なサーバの名称です。

## 使用例

次の例では、稼働していないサーバのトランザクションを回復します。

```
asadmin recover-transactions --transactionlogdir /logs/tx
--target server1 server2
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.23 バッチジョブ管理で使用するコマンド

バッチジョブ管理で使用するコマンド書式、機能などについて説明します。

### 2.23.1 list-batch-job-executions

バッチジョブ実行の一覧表示

#### 書式

```
asadmin [asadmin-options] list-batch-job-executions [--help]
  [--target target]
  [--executionid execution-id]
  [--long={false|true}]
  [--output output]
  [--header={false|true}]
  [instance_ID]
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

list-batch-job-executions サブコマンドは、バッチジョブ実行と実行の詳細情報を一覧表示します。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

一覧表示するバッチジョブ実行の対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *cluster\_name*  
指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--executionid execution-id | -x execution-id`

バッチジョブ実行の実行 ID を指定します。

型：String

指定できる値を次に示します。

- ジョブインスタンス ID

デフォルト値：なし

`--long={false|true} | -l={false|true}`

バッチジョブ実行に関する詳細情報を表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--output output | -o output`

バッチジョブ実行の特定の詳細情報を表示します。表示する詳細とその順序を指定するには、コンマ区切りのリストを使用します。値では大文字と小文字が区別されません。

型：String

指定できる値を次に示します。

- *jobname*  
ジョブの名称を表示します。
- *executionid*  
バッチジョブ実行に割り当てられた ID を表示します。ジョブが初めて開始したときと既存のジョブ実行が再開されるたびに、新しいジョブ実行が作成されます。
- *starttime*  
ジョブ実行の開始時間を表示します。
- *endtime*  
ジョブ実行の終了時間を表示します。
- *batchstatus*  
バッチ実行時に設定されたジョブの実行状態を表示します。
- *exitstatus*

ジョブのジョブ XML またはバッチアプリケーションによって設定されたジョブの実行状態を表示します。デフォルトでは、`exitstatus` が明示的にオーバーライドされないかぎり、`exitstatus` および `batchstatus` は同じです。

- `jobparameters`

バッチ実行時に設定されたジョブ実行のためのプロパティのキーと値のペア一覧を表示します。

- `stepcount`

バッチジョブ実行のステップ数を表示します。

デフォルト値：表示できるすべてのヘッダーのサブセット

`--header={false|true} | -h={false|true}`

`--long` オプションを使用したときに列ヘッダーを表示するかどうかを指定します。ヘッダーを表示しないようにするには、`--header` オプションに `false` を設定します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`true`

#### *instance\_ID*

実行の詳細を一覧表示するジョブインスタンスの ID を指定します。

型：Integer

指定できる値を次に示します。

- ジョブインスタンス ID

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスに対して実行されたバッチジョブを一覧表示し、特定の詳細情報を表示します。

```
asadmin list-batch-job-executions --target instance1
-o=jobname,executionid,batchstatus,exitstatus
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。



## 2.23.2 list-batch-job-steps

バッチジョブの実行ステップの一覧表示

### 書式

```
asadmin [asadmin-options] list-batch-job-steps [--help]
  [--long={false|true}]
  [--target target]
  [--output output]
  [--header={false|true}]
  execution_id
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-batch-job-steps サブコマンドは、特定のバッチジョブの実行ステップを一覧表示します。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

一覧表示するバッチジョブ実行ステップの対象を指定します。

型：String

指定できる値を次に示します。

- *server*  
ドメイン管理サーバを対象とします。*server* はドメイン管理サーバの名称です。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：*server*

--long={false|true} | -l={false|true}

バッチジョブの実行ステップの詳細情報を表示します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--output output | -o output`

バッチジョブの実行ステップに関する特定の情報を表示します。表示する詳細とその順序を指定するには、コンマ区切りのリストを使用します。値では大文字と小文字が区別されません。

型：String

指定できる値を次に示します。

- `stepname`  
ステップの名称を表示します。
- `stepid`  
ステップ ID を表示します。
- `starttime`  
ステップの開始時間を表示します。
- `endtime`  
ステップの終了時間を表示します。
- `batchstatus`  
バッチ実行時に設定されたステップの状態を表示します。
- `exitstatus`  
ジョブのジョブ XML またはバッチアプリケーションによって設定されたジョブの実行状態を表示します。デフォルトでは、`exitstatus` が明示的にオーバーライドされないかぎり、`exitstatus` および `batchstatus` は同じです。

デフォルト値：表示できるすべてのヘッダーのサブセット

`--header={false|true} | -h={false|true}`

`--long` オプションを使用したときに列ヘッダーを表示するかどうかを指定します。ヘッダーを表示しないようにするには、`--header` オプションに `false` を設定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`execution_id`

バッチジョブのステップおよび詳細を一覧表示する実行 ID を指定します。

型：Integer

指定できる値を次に示します。

- ジョブインスタンス ID

デフォルト値：なし

## 使用例

次の例では、バッチジョブの実行ステップおよび実行 ID が 7 のジョブ実行について、特定のステップ詳細を一覧表示します。対象は、instance1 という名称のサーバインスタンスです。

```
asadmin list-batch-job-steps --target instance1 o=stepname,stepid,batchstatus 7
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.23.3 list-batch-jobs

バッチジョブの一覧表示

### 書式

```
asadmin [asadmin-options] list-batch-jobs [--help]
  [--target target]
  [--long={false|true}]
  [--output output]
  [--header={false|true}]
  [job_name]
```

### 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

### 機能

list-batch-jobs サブコマンドは、バッチジョブとジョブの詳細を一覧表示します。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

一覧表示するバッチジョブの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`--long={false|true} | -l={false|true}`

バッチジョブに関する詳細情報を表示します。

型：Boolean

指定できる値を次に示します。

- `true`
- `false`

デフォルト値：`false`

`--output output | -o output`

バッチジョブに関する特定の情報を表示します。表示する詳細とその順序を指定するには、コンマ区切りのリストを使用します。値では大文字と小文字が区別されません。

型：String

指定できる値を次に示します。

- `jobname`  
ジョブの名称を表示します。
- `appname`  
アプリケーションの名称を表示します。
- `instancecount`  
ジョブインスタンス数を表示します。
- `instanceid`  
ジョブインスタンスに割り当てられた ID を表示します。

- **executionid**  
バッチジョブ実行に割り当てられた ID を表示します。ジョブが初めて開始したとき、および既存のジョブ実行が再開されるたびに新しいジョブ実行が作成されます。
- **batchstatus**  
バッチ実行時に設定されたジョブ状態を表示します。
- **starttime**  
ジョブの開始時間を表示します。
- **endtime**  
ジョブの終了時間を表示します。
- **exitstatus**  
ジョブのジョブ XML またはバッチアプリケーションによって設定されたジョブの状態を表示します。デフォルトでは、**exitstatus** が明示的にオーバーライドされないかぎり、**exitstatus** および **batchstatus** は同じです。

デフォルト値： **jobname**、 **instancecount**

**--header={false|true} | -h={false|true}**

**--long** オプションを使用したときに列ヘッダーを表示するかどうかを指定します。ヘッダーを表示しないようにするには、**--header** オプションに **false** を設定します。

型： Boolean

指定できる値を次に示します。

- **true**
- **false**

デフォルト値： **true**

### *job\_name*

詳細を一覧表示するジョブの名称を指定します。

型： Integer

指定できる値を次に示します。

- ジョブ名

デフォルト値： なし

## 使用例

次の例では、**instance1** という名称のサーバインスタンスのバッチジョブを一覧表示します。

```
asadmin list-batch-jobs --target instance1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.23.4 list-batch-runtime-configuration

バッチ実行時の構成の表示

### 書式

```
asadmin [asadmin-options] list-batch-runtime-configuration [--help]
  [--target target]
  [--output output]
  [--header={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

list-batch-runtime-configuration サブコマンドは、バッチ実行時の構成を表示します。バッチ実行時の構成データはdomain.xml ファイルのconfig 要素に格納されます。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### ファイル

バッチ実行時の構成データはdomain.xml ファイルのconfig 要素に格納されます。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

一覧表示するバッチ実行時の構成の対象を指定します。

型：String

指定できる値を次に示します。

- server

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- *cluster\_name*

指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

`--output output | -o output`

バッチ実行時の構成に関する特定の情報を表示します。

表示する詳細とその順序を指定するには、コンマ区切りのリストを使用します。値では大文字と小文字が区別されません。

型：String

指定できる値を次に示します。

- *datasourcelookupname*

ジョブ情報の格納に使用されるデータソースの JNDI ルックアップ名です。デフォルトでは、バッチ実行時はデフォルトのデータソース `jdbc/_BatchPool` が使用されます。

- *executorservicelookupname*

ジョブへのスレッドの提供に使用される管理対象の `ManagedExecutorService` の JNDI ルックアップ名です。デフォルトでは、バッチ実行時はデフォルトの管理対象の `ManagedExecutorService` の `concurrent/_defaultManagedExecutorService` が使用されます。

デフォルト値：datasourcelookupname、executorservicelookupname

`--header={false|true} | -h={false|true}`

`--long` オプションを使用したときに列ヘッダーを表示するかどうかを指定します。ヘッダーを表示しないようにするには、`--header` オプションに `false` を設定します。

型：Boolean

指定できる値を次に示します。

- *true*

- *false*

デフォルト値：true

## 使用例

次の例では、instance1 という名称のサーバインスタンスについて、バッチ実行時の構成を一覧表示します。

```
asadmin list-batch-runtime-configuration --target instance1
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.23.5 set-batch-runtime-configuration

バッチランタイムの設定

### 書式

```
asadmin [asadmin-options] set-batch-runtime-configuration [--help]
  [--target target]
  [--datasourcelookupname datasource-lookup-name]
  [--executorservicelookupname executor-service-lookup-name]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

set-batch-runtime-configuration サブコマンドは、バッチランタイムを設定します。ランタイムはデータソースとManagedExecutorService を使用してバッチジョブを実行します。バッチランタイムの構成データは domain.xml ファイルのconfig 要素に格納されます。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### ファイル

バッチランタイムの構成データはdomain.xml ファイルのconfig 要素に格納されます。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--target *target*

設定するバッチランタイムの対象を指定します。

型：String

指定できる値を次に示します。



- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`{--datasourcelookupname | -d} datasource-lookup-name`

ジョブ情報の格納に使用されるデータソースの JNDI ルックアップ名を指定します。

データソースを変更する必要がある場合、ドメインを停止してから再起動し、ジョブが起動または再起動される前に変更します。なお、データソースを変更すると、以前のデータソースに格納されていた情報にはアクセスできなくなります。

型：String

指定できる値を次に示します。

- JNDI ルックアップ名

デフォルト値：`jdbc/_BatchPool`

`{--executorservicelookupname | -x} executor-service-lookup-name`

ジョブにスレッドを提供するために使用される `ManagedExecutorService` の JNDI ルックアップ名を指定します。

型：String

指定できる値を次に示します。

- `ManagedExecutorService` の JNDI ルックアップ名

デフォルト値：`concurrent/_defaultManagedExecutorService`

## 使用例

次の例では、`concurrent/myExecutor` という名称の既存の `ManagedExecutorService` を使用するために、`instance1` という名称のサーバインスタンスのバッチランタイムを設定します。

```
asadmin set-batch-runtime-configuration --target instance1
--executorservicelookupname concurrent/myExecutor
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24 並行リソース管理で使用するコマンド

並行リソース管理で使用するコマンドの書式、機能などについて説明します。

### 2.24.1 create-context-service

コンテキストサービスのリソースの作成

#### 書式

```
asadmin [asadmin-options] create-context-service [--help]
  [--enabled={false|true}] [--contextinfoenabled={false|true}]
  [--contextinfo={ClassLoader|JNDI|Security|WorkArea}]
  [--description description] [--target target]
  jndi_name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`create-context-service` サブコマンドは、コンテキストサービスのリソースを作成します。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--enabled={false|true}`

実行時にリソースを有効にするかどうかを決定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--contextinfoenabled={false|true}`

コンテナのコンテキストをスレッドに伝えるかどうかを決定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--contextinfo={ClassLoader|JNDI|Security|WorkArea}`

各スレッドに伝える個別のコンテナコンテキストを指定します。

型：String

指定できる値を次に示します。

- Classloader
- JNDI
- Security
- WorkArea

デフォルト値：ClassLoader、JNDI、Security およびWorkArea

`--description description`

リソースについて詳細な説明を記述します。

型：String

指定できる値を次に示します。

- リソースについての詳細説明

デフォルト値：なし

`--target target`

作成するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- server  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- domain  
ドメインを対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

### *jndi\_name*

このリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、コンテキストサービスを作成します。

```
asadmin create-context-service --target instance1 concurrent/myContextService
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.2 create-managed-executor-service

ManagedExecutorService リソースの作成

### 書式

```
asadmin [asadmin-options] create-managed-executor-service [--help]
  [--enabled={false|true}] [--contextinfoenabled={false|true}]
  [--contextinfo={ClassLoader|JNDI|Security|WorkArea}]
  [--threadpriority threadpriority] [--longrunningtasks={false|true}]
  [--hungafterseconds hungafterseconds] [--corepoolsize corepoolsize]
  [--maximumpoolsize maximumpoolsize]
  [--keepaliveseconds keepaliveseconds]
  [--threadlifetimeseconds threadlifetimeseconds]
  [--taskqueuecapacity taskqueuecapacity]
  [--target target] jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

`create-managed-executor-service` サブコマンドでは、`ManagedExecutorService` リソースを作成します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--enabled={false|true}`

リソースが実行時に有効であるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--contextinfoenabled={false|true}`

コンテナのコンテキストをスレッドに伝えるかどうかを決定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--contextinfo={ClassLoader|JNDI|Security|WorkArea}`

スレッドに伝える個々のコンテナコンテキストを指定します。デフォルトでは、すべてのコンテキストが伝えられます。

型：String

指定できる値を次に示します。

- Classloader
- JNDI
- Security
- WorkArea

デフォルト値：Classloader、JNDI、Security および WorkArea

`--threadpriority threadpriority`

作成済みスレッドに割り当てる優先順位を指定します。

型：Integer

指定できる値を次に示します。

- 1～10

デフォルト値：5

`--longrunningtasks={false|true}`

リソースを長時間動作タスクで使用する必要があるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--hungafterseconds hungafterseconds`

応答なしと見なされてタスクを実行できなくなるまでの秒数を指定します。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：0

`--corepoolsize corepoolsize`

スレッドプールに保持するスレッド数（アイドル状態も含む）を指定します。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：0

`--maximumpoolsize maximumpoolsize`

スレッドプールに保持できる最大スレッド数を指定します。

型：Integer

指定できる値を次に示します。

- 0～2147483647

デフォルト値：2147483647

`--keepaliveseconds keepaliveseconds`

`corepoolsize` よりスレッド数が多いとき、スレッドがアイドル状態でいられる秒数を指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：60

**--threadlifetimeseconds *threadlifetimeseconds***

利用しなくなったスレッドをプールから削除するまでの秒数を指定します。スレッド数が `corepoolsizesize` より多い、またはスレッドがアイドル状態であるかに関係ありません。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

**--taskqueuecapacity *taskqueuecapacity***

実行を待っているタスクキューに格納できる、サブミットされたタスク数を指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：2147483647

**--target *target***

作成するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

***jndi\_name***

このリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスに、ManagedExecutorService を作成します。

```
asadmin create-managed-executor-service --target instance1 concurrent/myExecutor
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.3 create-managed-scheduled-executor-service

ManagedScheduledExecutorService リソースの作成

### 書式

```
asadmin [asadmin-options] create-managed-scheduled-executor-service  
  [--help] [--enabled={false|true}]  
  [--contextinfoenabled={false|true}]  
  [--contextinfo={ClassLoader|JNDI|Security|WorkArea}]  
  [--threadpriority threadpriority] [--longrunningtasks={false|true}]  
  [--hungafterseconds hungafterseconds] [--corepoolsize corepoolsize]  
  [--keepaliveseconds keepaliveseconds]  
  [--threadlifetimeseconds threadlifetimeseconds]  
  [--description description] [--target target] jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

### 機能

create-managed-scheduled-executor-service サブコマンドは、ManagedScheduledExecutorService リソースを作成します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。



## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--enabled={false|true}`

リソースが実行時に有効であるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--contextinfoenabled={false|true}`

コンテナのコンテキストをスレッドに伝えるかどうかを決定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

`--contextinfo={ClassLoader|JNDI|Security|WorkArea}`

各スレッドに伝える個別のコンテナコンテキストを指定します。

型：String

指定できる値を次に示します。

- Classloader
- JNDI
- Security
- WorkArea

デフォルト値：ClassLoader、JNDI、Security および WorkArea

`--threadpriority threadpriority`

作成済みスレッドに割り当てる優先順位を指定します。

型：Integer

指定できる値を次に示します。

- 1～10

デフォルト値：5

`--longrunningtasks={false|true}`

リソースを長時間動作タスクで使用する必要があるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`--hungafterseconds hungafterseconds`

応答なしと見なされてタスクを実行できなくなるまでの秒数を指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--corepoolsize corepoolsize`

スレッドプールに保持するスレッド数（アイドル状態も含む）を指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--keepaliveseconds keepaliveseconds`

`corepoolsize` よりスレッド数が多いとき、スレッドがアイドル状態でいられる秒数を指定します。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：60

`--threadlifetimeseconds threadlifetimeseconds`

利用しなくなったスレッドをプールから削除するまでの秒数を指定します。スレッド数が `corepoolsize` より多い、またはスレッドがアイドル状態であるかに関係ありません。

型：Integer

指定できる値を次に示します。

- 0~2147483647

デフォルト値：0

`--description description`

リソースに関する詳細情報を指定します。

型：String

指定できる値を次に示します。

- リソースの説明

デフォルト値：なし

`--target target`

作成するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`jndi_name`

このリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスに、`ManagedScheduledExecutorService` を作成します。

```
asadmin create-managed-scheduled-executor-service --target instance1
concurrent/myScheduledExecutor
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.4 create-managed-thread-factory

ManagedThreadFactory リソースの作成

### 書式

```
asadmin [asadmin-options] create-managed-thread-factory [--help]
  [--enabled={false|true}] [--contextinfoenabled={false|true}]
  [--contextinfo={ClassLoader|JNDI|Security|WorkArea}]
  [--threadpriority threadpriority] [--description description]
  [--target target] jndi_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

create-managed-thread-factory サブコマンドでは、ManagedThreadFactory リソースを作成します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

--help | -?

このサブコマンドのヘルプテキストを表示します。

--enabled={false|true}

ManagedThreadFactory が実行時に有効であるかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：true

--contextinfoenabled={false|true}

コンテナのコンテキストをスレッドに伝えるかどうかを決定します。

型：Boolean

指定できる値を次に示します。

- true

- false

デフォルト値：true

--contextinfo={ClassLoader|JNDI|Security|WorkArea}

各スレッドに伝える個別のコンテナコンテキストを指定します。

型：String

指定できる値を次に示します。

- Classloader
- JNDI
- Security
- WorkArea

デフォルト値：ClassLoader、JNDI、Security およびWorkArea

--threadpriority *threadpriority*

作成済みスレッドに割り当てる優先順位を指定します。

型：Integer

指定できる値を次に示します。

- 1~10

デフォルト値：5

--description *description*

リソースに関する詳細情報を指定します。

型：String

指定できる値を次に示します。

- リソースの説明

デフォルト値：なし

--target *target*

作成するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- server  
ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。
- domain  
ドメインを対象とします。
- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*

指定した名称のサービンスタンスを対象とします。

デフォルト値：server

#### *jndi\_name*

このリソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- JNDI 名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサービンスタンスに、ManagedThreadFactory を作成します。

```
asadmin create-managed-thread-factory --target instance1
concurrent/myThreadFactory
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.5 delete-context-service

コンテキストサービスのリソースの削除

### 書式

```
asadmin [asadmin-options] delete-context-service [--help]
        [--target target] context-service-name
```

### 格納先

Application Serverインストールディレクトリー/javaee/glassfish/bin

### 機能

delete-context-service サブコマンドは指定した JNDI 名のコンテキストサービスのリソースを削除します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストが表示されます。

`--target target`

削除するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`context-service-name`

リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- リソースの JNDI 名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`concurrent/myContextService` という名称のコンテキストサービスのリソースを削除します。

```
asadmin delete-context-service --target instance1 concurrent/myContextService
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.6 delete-managed-executor-service

ManagedExecutorService リソースの削除

### 書式

```
asadmin [asadmin-options] delete-managed-executor-service [--help]
  [--target target]
  managed_executor_service_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-managed-executor-service` サブコマンドは、指定された JNDI 名を持つ ManagedExecutorService リソースを削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

削除するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`



ドメインを対象とします。

- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

*managed\_executor\_service\_name*

リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- *JNDI 名*

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンスから、concurrent/myExecutor という名称の ManagedExecutorService リソースを削除します。

```
asadmin delete-managed-executor-service --target instance1 concurrent/myExecutor
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.7 delete-managed-scheduled-executor-service

ManagedScheduledExecutorService リソースの削除

### 書式

```
asadmin [asadmin-options] delete-managed-scheduled-executor-service  
  [--help] [--target target]  
  managed_scheduled_executor_service_name
```

## 格納先

*Application Server* インストールディレクトリー/javaee/glassfish/bin

## 機能

`delete-managed-scheduled-executor-service` サブコマンドは、指定された JNDI 名を持つ `ManagedScheduledExecutorService` リソースを削除します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`managed_scheduled_executor_service_name`

リソースの JNDI 名を指定します。

型：String

指定できる値を次に示します。

- `JNDI 名`

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`concurrent/myScheduledExecutor` という名称の `ManagedScheduledExecutorService` リソースを削除します。

```
asadmin delete-managed-scheduled-executor-service --target instance1
concurrent/myScheduledExecutor
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.8 delete-managed-thread-factory

ManagedThreadFactory リソースの削除

### 書式

```
asadmin [asadmin-options] delete-managed-thread-factory [--help]
  [--target target]
  managed_thread_factory_name
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`delete-managed-thread-factory` サブコマンドは、指定された JNDI 名を持つ `ManagedThreadFactory` リソースを削除します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target` *target*

削除するリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

#### *managed\_thread\_factory\_name*

リソースの JNDI 名を指定します。

型：`String`

指定できる値を次に示します。

- *JNDI 名*

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサーバインスタンスから、`concurrent/myThreadFactory` という名称の `ManagedThreadFactory` リソースを削除します。

```
asadmin delete-managed-thread-factory --target instance1
concurrent/myThreadFactory
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.9 list-context-services

コンテキストサービスのリソースの一覧表示

### 書式

```
asadmin [asadmin-options] list-context-services [--help]
[target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-context-services` サブコマンドは、コンテキストサービスのリソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示するコンテキストサービスのリソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのサーバインスタンス上のコンテキストサービスのリソースを一覧表示します。

```
asadmin list-context-services
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.10 list-managed-executor-services

ManagedExecutorService リソースの一覧表示

### 書式

```
asadmin [asadmin-options] list-managed-executor-services [--help]  
[target]
```

### 格納先

*Application Server* インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-managed-executor-services` サブコマンドでは、ManagedExecutorService リソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する ManagedExecutorService リソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`

ドメインを対象とします。

- *cluster\_name*  
指定した名称のクラスターを対象とします。
- *instance\_name*  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、デフォルトのサーバインスタンスにある ManagedExecutorService リソースを一覧表示します。

```
asadmin list-managed-executor-services
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.11 list-managed-scheduled-executor-services

ManagedScheduledExecutorService リソースの一覧表示

## 書式

```
asadmin [asadmin-options] list-managed-scheduled-executor-services  
[--help] [target]
```

## 格納先

*Application Server*インストールディレクトリー/javaee/glassfish/bin

## 機能

list-managed-scheduled-executor-services サブコマンドは、ManagedScheduledExecutorService リソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する ManagedScheduledExecutorService リソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのサーバインスタンス上の ManagedScheduledExecutorService リソースを一覧表示します。

```
asadmin list-managed-scheduled-executor-services
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.24.12 list-managed-thread-factories

ManagedThreadFactory リソースの一覧表示



## 書式

```
asadmin [asadmin-options] list-managed-thread-factories [--help]
        [target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-managed-thread-factories` サブコマンドでは、`ManagedThreadFactory` リソースを一覧表示します。

このサブコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

*target*

一覧表示する `ManagedThreadFactory` リソースの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

## 使用例

次の例では、デフォルトのサーバインスタンス上の `ManagedThreadFactory` リソースを一覧表示します。

```
asadmin list-managed-thread-factories
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.25 メッセージセキュリティープロバイダーの管理で使用するコマンド

メッセージセキュリティープロバイダーの管理で使用するコマンドの書式、機能などについて説明します。

### 2.25.1 create-message-security-provider

メッセージセキュリティープロバイダーの作成

#### 書式

```
asadmin [asadmin-options] create-message-security-provider [--help]
  [--target target] --classname provider_class
  --layer message_layer [--providertype provider_type]
  [--requestauthsource request_auth_source]
  [--requestauthrecipient request_auth_recipient]
  [--responseauthsource response_auth_source]
  [--responseauthrecipient response_auth_recipient]
  [--isdefaultprovider]
  [--property name=value[:name=value]...]
  provider_name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`create-message-security-provider` サブコマンドでは、管理者がセキュリティーサービスのメッセージセキュリティープロバイダーを作成できます。これで SOAP メッセージを保護する方法を指定できます。

このサブコマンドはリモートモードだけサポートします。

#### 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

#### ファイル

`--property` オプションで、セキュリティー構成ファイル (`security.config`) を指定できます。

#### 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

作成するメッセージセキュリティープロバイダーの対象を指定します。

型：String

指定できる値を次に示します。

- `server`  
ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：`server`

`--classname provider_class`

プロバイダーの Java 実装クラスを定義します。クライアント認証の各プロバイダーは `com.sun.enterprise.security.jauth.ClientAuthModule` インターフェースを実装する必要があります。サーバ側プロバイダーは、`com.sun.enterprise.security.jauth.ServerAuthModule` インターフェースを実装する必要があります。プロバイダーに両方のインターフェースを実装できますが、プロバイダーのタイプに応じたインターフェースを実装する必要があります。

型：String

指定できる値を次に示します。

- `com.sun.enterprise.security.jauth.ClientAuthModule`
- `com.sun.enterprise.security.jauth.ServerAuthModule`

デフォルト値：なし

`--layer message_layer`

`message-security-config` 要素の `auth-layer` 属性の値を定義するために使用する `message-layer` エンティティーを指定します。

型：String

指定できる値を次に示します。

- `HttpServlet`
- `SOAP`

デフォルト値：`HttpServlet`

`--providertype provider_type`

使用されるプロバイダーのタイプ（クライアント認証プロバイダー、サーバ認証プロバイダー、あるいは両方）を指定します。

型：String

指定できる値を次に示します。

- client
- server
- client-server

デフォルト値：なし

**--requestauthsource** *request\_auth\_source*

message-layer センダー認証（ユーザー名、パスワードなど）または内容認証（デジタル署名など）を要求メッセージに適用する要件を定義するauth-source 属性を指定します。

型：String

指定できる値を次に示します。

- sender
- content

デフォルト値：なし

**--requestauthrecipient** *request\_auth\_recipient*

メッセージのレシーバーがセンサーをmessage-layer 認証（XML 暗号による場合など）する要件を定義するauth-recipient 属性を指定します。

型：String

指定できる値を次に示します。

- before-content
- after-content

デフォルト値：after-content

**--responseauthsource** *response\_auth\_source*

message-layer センダー認証（ユーザー名、パスワードなど）または内容認証（デジタル署名など）を応答メッセージに適用する要件を定義するauth-source 属性を指定します。

型：String

指定できる値を次に示します。

- sender
- content

デフォルト値：なし

**--responseauthrecipient** *response\_auth\_recipient*

応答メッセージのレシーバーがセンサーをmessage-layer 認証（XML 暗号による場合など）する要件を定義するauth-recipient 属性を指定します。

型：String

指定できる値を次に示します。

- before-content

- after-content

デフォルト値：after-content

--isdefaultprovider={false|true}

--providertype で指定したタイプのレイヤーで、デフォルトプロバイダーにします。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

--property name=value[:name=value]...

初期化される時にプロバイダーに渡すプロバイダー特有のプロパティの値を指定します。このようにして渡される各プロパティには、プロバイダーがキーストア、署名、正規化、暗号アルゴリズムなどからキーを取得するために使用するキーエイリアスが含まれる場合があります。

指定形式は、*name=value* です。複数指定する場合は、コロン (:) で区切ります。同じプロパティ名を複数指定した場合は、最後に指定したプロパティの値が有効となります。

プロパティ名と *value* に指定できる値を次に示します。

**security.config=value**

メッセージセキュリティ構成ファイルのロケーションを指定します。domain-dir/config ディレクトリーにある構成ファイルを指定する場合、システムプロパティ `{com.sun.aas.instanceRoot}/config/` を使用します。例えば、`{com.sun.aas.instanceRoot}/config/wss-server-config-1.0.xml` があります。

型：String

デフォルト値：*domain-dir/config/ wss-serverconfig-1.0.xml*

**debug=value**

true の場合、デバッグメッセージのサーバログへの出力を有効にします。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

**dynamic.username.password=value**

true に設定の場合、プロバイダーに信号が送られ、各要求の `CallbackHandler` からユーザー名とパスワードが収集されます。false の場合、`wsse:UsernameToken(s)` に対するユーザー名とパスワードがモジュール初期化の間に一度収集されます。このプロパティは `ClientAuthModule` にだけ適用できます。

型：Boolean

指定できる値を次に示します。

- true
- false

デフォルト値：false

`encryption.key.alias=value`

プロバイダーが使用する暗号化のキーを指定します。キーはキーストアエイリアスで確認されます。

型：String

デフォルト値：s1as

`signature.key.alias=value`

プロバイダーが使用する署名のキーを指定します。キーはキーストアエイリアスで確認されます。

型：String

デフォルト値：s1as

#### *provider\_name*

`provider-config` 要素を参照するために使用するプロバイダー名を指定します。

型：String

指定できる値を次に示します。

- プロバイダー名

デフォルト値：なし

## 使用例

次の例では、`instance1` という名称のサービンスタンスに、メッセージセキュリティープロバイダーを作成します。

```
asadmin create-message-security-provider --target instance1
--classname com.sun.enterprise.security.jauth.ClientAuthModule --layer SOAP
--providertype client mySecurityProvider
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.25.2 delete-message-security-provider

メッセージセキュリティープロバイダーの削除

## 書式

```
asadmin [asadmin-options] delete-message-security-provider [--help]
  [--target target] --layer message_layer
  provider_name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`delete-message-security-provider` サブコマンドは、メッセージセキュリティープロバイダーを削除します。このサブコマンドを実行すると、指定したメッセージレイヤーの `provider-config` 子要素 (`domain.xml` の `message-security-config` 要素) が削除されます。`domain.xml` ファイルには、Java EE Server のパラメーターとプロパティーを指定します。このサブコマンドで指定したオプションは、`domain.xml` の `message-security-config` および `provider-config` 子要素内の属性に適用されます。メッセージレイヤー (`message-security-config` 要素) が存在しない場合は作成され、さらにその配下に `provider-config` 子要素が作成されます。

このコマンドはリモートモードだけサポートします。

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## ファイル

このサブコマンドを実行すると、指定したメッセージレイヤーの `provider-config` 子要素 (`domain.xml` の `message-security-config` 要素) が削除されます。`domain.xml` ファイルには、Java EE Server のパラメーターとプロパティーを指定します。このサブコマンドで指定したオプションは、`domain.xml` の `message-security-config` および `provider-config` 子要素内の属性に適用されます。

## 引数

`--help | -?`

このサブコマンドのヘルプテキストを表示します。

`--target target`

削除するメッセージセキュリティープロバイダーの対象を指定します。

型: String

指定できる値を次に示します。

- `server`

ドメイン管理サーバを対象とします。`server` はドメイン管理サーバの名称です。

- `domain`



ドメインを対象とします。

- *cluster\_name*

指定した名称のクラスターを対象とします。

- *instance\_name*

指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

**--layer** *message\_layer*

削除対象のプロバイダーがあるメッセージレイヤーを指定します。

型：String

指定できる値を次に示します。

- HttpServlet
- SOAP

デフォルト値：HttpServlet

*provider\_name*

provider-config 要素を参照するときに使用されるプロバイダー名を指定します。

型：String

指定できる値を次に示します。

- プロバイダー名

デフォルト値：なし

## 使用例

次の例では、instance1 という名称のサーバインスタンス上のクライアントから、メッセージセキュリティプロバイダーを削除します。

```
asadmin delete-message-security-provider --target instance1 --layer SOAP mySecurityProvider
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.25.3 list-message-security-providers

メッセージレイヤーのメッセージセキュリティプロバイダーの一覧表示

## 書式

```
asadmin [asadmin-options] list-message-security-providers [--help]
  [--layer message_layer]
  [target]
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

`list-message-security-providers` サブコマンドを使用すると、管理者は指定したメッセージレイヤー (`domain.xml` の `message-security-config` 要素) のすべてのメッセージセキュリティープロバイダー (`provider-config` 子要素) を一覧表示できます。

このサブコマンドはリモートモードだけサポートします。

## 実行権限

スーパーユーザー

## 前提条件

ドメイン管理サーバは、稼働状態である必要があります。

## 引数

`--help` | `-?`

このサブコマンドのヘルプテキストを表示します。

`--layer` *message\_layer*

プロバイダーを一覧表示するメッセージレイヤーを指定します。

型: String

指定できる値を次に示します。

- SOAP
- HttpServlet

デフォルト値: HttpServlet

*target*

一覧表示するメッセージセキュリティープロバイダーの対象を指定します。

型: String

指定できる値を次に示します。

- server

ドメイン管理サーバを対象とします。server はドメイン管理サーバの名称です。

- `domain`  
ドメインを対象とします。
- `cluster_name`  
指定した名称のクラスターを対象とします。
- `instance_name`  
指定した名称のサーバインスタンスを対象とします。

デフォルト値：server

## 使用例

次の例では、特定のメッセージレイヤーのメッセージセキュリティープロバイダーを一覧表示します。

```
asadmin list-message-security-providers --layer SOAP
```

## 戻り値

戻り値	説明
0	サブコマンドの実行に成功しました。
1	サブコマンドの実行中にエラーが発生しました。

## 2.26 イベント検知時コマンド実行で使用するコマンド

イベント検知時コマンド実行で使用するコマンドの書式、機能などについて説明します。

### 2.26.1 create-event-hook

イベント検知時コマンドの登録

#### 書式

```
asadmin [asadmin-options] create-event-hook --messageid=messageid  
--script=script-file-path  
[--monitoring-time-span=monitoring-time-span  
--monitoring-max-count=monitoring-max-count] event-hook-name
```

#### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

#### 機能

`create-event-hook` サブコマンドは、ドメイン管理サーバでイベントが発生したときに、自動で実行される「イベント検知時コマンド」を登録します。イベント検知時コマンドには、ユーザーが作成した任意のスクリプトファイルを指定します。

イベント検知時には、指定したスクリプトファイルの配置場所で、登録したコマンドを実行します。

イベント検知時コマンドには任意の名称を指定できますが、次の引数の指定が同じ値のイベント検知時コマンドがすでに登録されている場合は、未登録の名称でもエラーとします。

- `--messageid`

イベント検知時コマンドのタイムアウト時間は 600 秒です。イベント検知時コマンドが 600 秒以内に完了しない場合はタイムアウトが発生し、KDKD10153-E メッセージがドメイン管理サーバのメッセージログに出力されます。このメッセージが出力されてもイベント検知時コマンドの処理は続行されます。

このコマンドはリモートモードだけサポートします。

#### 実行権限

一般ユーザー

## 引数

`--messageid=messageid`

イベント検知時コマンドを実行させる契機となるメッセージ ID を指定します。

型：String

指定できる値を次に示します。

- システムで公開されているメッセージ ID

デフォルト値：なし（必須）

`--script=script-file-path`

イベント検知時コマンドとして登録するスクリプトファイルを指定します。ドメイン管理サーバ上での絶対パスで指定してください。

型：String

指定できる値を次に示します。

- 存在するファイルの絶対パス

デフォルト値：なし（必須）

`--monitoring-time-span=monitoring-time-span`

このオプションで指定した時間ごとにイベント検知時コマンドが実行される回数を監視します。

値は秒数で指定します。0 を指定した場合は、常にイベント検知時コマンドの実行を許可します（監視しません）。

イベント検知時コマンドを実行したことによってイベント検知対象のイベントが発生し、この繰り返しで無限にイベント検知時コマンドが実行される場合があります。このオプションと `--monitoring-max-count` オプションを同時に指定することで、イベント検知時コマンドが無限に繰り返されることを防げます。

型：Integer

指定できる値を次に示します。

- 0 ~ 2147483647

デフォルト値：0

`--monitoring-max-count=monitoring-max-count`

`--monitoring-time-span` オプションで指定した時間内に、同じ種類のイベントに対するイベント検知時コマンドを実行する回数の上限を指定します。0 を指定した場合は、常にイベント検知時コマンドの実行を許可します。

同じ種類のイベントとは、次に示す条件を満たすイベントのことです。

サーバに関するイベントの場合、次のすべてが同じイベント

- イベントの契機となったメッセージ ID
- イベントが発生したサーバのサーバ名

ノードに関するイベントの場合、次のすべてが同じイベント

- イベントの契機となったメッセージ ID
- イベントが発生したノード名

サーバやノードとは関係ないイベントの場合、次の内容が同じイベント

- イベントの契機となったメッセージ ID

型：Integer

指定できる値を次に示します。

- 0 ~ 256

デフォルト値：0

#### *event-hook-name*

イベント検知時コマンドの名称を指定します。

型：String

指定できる値を次に示します。

- ASCII 文字であること。
- 名称は次の文字で構成すること。
  - 英小文字<a~z>
  - 英大文字<A~Z>
  - 数字<0~9>
  - ハイフン<->
  - アンダースコア<\_>
- 先頭文字は英大文字または英小文字を使用すること。
- ドメインに存在する次の名称は指定できません。

ノード名

JavaEE サーバ名

Web サーバ名

PRF のサーバ名

クラスター名

サーバ間関連名

JavaEE サーバの構成名

- 次に示す名称はシステムで使用済みまたは予約語のため指定できません。

domain

server

default

server-config

default-config

default-webserver-config

default-prf-config

javaee

webserver

prf

cluster

redirect

prf-relation

HJES\_で始まる名称

構築済み Web サーバ名-config

構築済み PRF 名-config

デフォルト値：なし（必須）

## 使用例

サーバの起動が成功したときにイベント検知時コマンドを実行する場合

```
asadmin create-event-hook --messageid KDKD10083-I
--script /work/failure-script.sh Event01
```

サーバの停止が成功したときにイベント検知時コマンドを実行する場合（イベント検知時コマンドの実行監視オプションを指定する場合）

```
asadmin create-event-hook --messageid KDKD10094-I
--script /work/failure-script.sh --monitoring-time-span=180
--monitoring-max-count=3 Event02
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

## 注意事項

- 登録したイベント検知時コマンドのスクリプトは、イベント検知時コマンドの登録を解除するまで、削除しないでください。

## 2.26.2 delete-event-hook

イベント検知時コマンドの登録解除

## 書式

```
asadmin [asadmin-options] delete-event-hook event-hook-name
```

## 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

## 機能

delete-event-hook サブコマンドは、イベント検知時コマンドの登録を解除します。

このコマンドはリモートモードだけサポートします。

## 実行権限

一般ユーザー

## 前提条件

イベント検知時コマンドを登録済みであること。

## 引数

*event-hook-name*

登録解除するイベント検知時コマンドの名称を指定します。

型：String

指定できる値を次に示します。

- 存在するイベント検知時コマンドの名称

デフォルト値：なし（必須）

## 使用例

- Event01 のイベント検知時コマンドの登録を解除する場合

```
asadmin delete-event-hook Event01
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了



## 2.26.3 list-event-hooks

イベント検知時コマンドの一覧表示

### 書式

```
asadmin [asadmin-options] list-event-hooks [--long={false|true}]
```

### 格納先

*Application Server*インストールディレクトリー/*javaee/glassfish/bin*

### 機能

`list-event-hooks` サブコマンドは、登録されているイベント検知時コマンドを一覧表示します。

このコマンドはリモートモードだけサポートします。

### 実行権限

一般ユーザー

### 引数

`--long={false|true}`

ヘッダー情報を表示するかどうかを指定します。

型：Boolean

指定できる値を次に示します。

- `true`

すべてのイベント検知時コマンドの一覧およびヘッダー情報を表示します。一覧には、すべての項目を表示します。

- `false`

すべてのイベント検知時コマンドの一覧だけを表示します。ヘッダー情報は表示しません。一覧には、名前、メッセージ ID およびスクリプトのパスを表示します。

デフォルト値：`false`

### 出力形式

1 行目は出力情報のヘッダー部となり、固定文字列で出力形式を示します。

カラム間は、カラム内で最も長い文字列と次のカラムの先頭までを、2つの半角スペースで区切ります。

ヘッダーを除く各行は`create-event-hook` サブコマンドの登録順に表示します。

NAME	MESSAGEID	SCRIPT	MONITORING-TIME-SPAN	MONITORING-MAX-COUNT
名前	メッセージID	スクリプトのパス	時間間隔	最大許容実行回数

名前：イベント検知時コマンド名を表示します。

メッセージID：メッセージIDを表示します。

スクリプトのパス：実行するスクリプトの絶対パスを表示します。

時間間隔：最大許容実行回数を設定する時間間隔を表示します。

最大許容実行回数：最大許容実行回数を表示します。

## 使用例

ヘッダーを含め、登録されているすべてのイベント検知時コマンドを表示する場合

```
list-event-hooks --long=true
```

[出力例]

NAME	MESSAGEID	SCRIPT	MONITORING-TIME-SPAN	MONITORING-MAX-COUNT
Event01	KDKD10083-I	/work/failure-script.sh	180	3

ヘッダーを表示しない場合

```
list-event-hooks --long=false
```

[出力例]

```
Event01 KDKD10083-I /work/failure-script.sh
```

## 戻り値

戻り値	説明
0	正常終了
1	異常終了

# 3

## Web サーバで使用するコマンド

この章では、Web サーバで使用するコマンドの書式、機能などについて説明します。

## 3.1 Web サーバで使用するコマンドの一覧

Web サーバで使用するコマンドの一覧を次の表に示します。

### Web サーバの運用で使用するコマンド

コマンド名称	要約	概要説明
htpasswd	ユーザー名とパスワードのパスワードファイルへの登録およびパスワードの変更	パスワードファイルにユーザー名、パスワードを登録します。パスワードを変更したり、パスワードファイルに登録されたユーザー名を削除したりできます。
httpsd	Web サーバの起動	デフォルト以外の httpsd.conf ファイル名称を指定する場合またはサーバのルートディレクトリーや httpsd.conf ファイルを指定して起動する場合に使用します。
httpsdctl	Web サーバの起動、再起動および停止	Web サーバを起動、再起動および停止します。
hwstraceinfo	内部トレースの採取	アプリケーションプログラムの実行時やリクエスト受け取り時など、システムで発生した事象を採取します。
logresolve	アクセスログファイル内の IP アドレスのホスト名への変換	レコードの先頭が IP アドレスであるアクセスログファイル内の IP アドレスをホスト名に変換し、新規ログファイルに出力します。

### SSL による認証、暗号化で使用するコマンド

コマンド名称	要約	概要説明
hwscertutil cert -hash	ハッシュリンクの作成	証明書のハッシュリンクを作成します。
hwscertutil cert -outform	証明書の形式変換	証明書の形式を変換します。
hwscertutil cert -text	証明書の内容表示	証明書ファイルの内容を表示します。
hwscertutil req	証明書発行要求 (CSR) の内容表示	証明書発行要求 (CSR) の内容を表示します。
hwscertutil reqgen	証明書発行要求 (CSR) の作成	証明書発行要求 (CSR) を作成します。ここで作成した CSR ファイルを CA に提出して、署名済みの証明書を発行してもらいます。
hwskeygen	Web サーバの秘密鍵の作成	Web サーバの秘密鍵を作成します。作成した Web サーバの秘密鍵のファイルは、SSLCertificateKeyFile ディレクティブに指定します。
sslpaswd	パスワード付きサーバ秘密鍵のパスワードファイル作成	パスワードによって保護されているサーバ秘密鍵を使用する場合、サーバ起動時のパスワード入力を省略できます。

**!** 重要

- コマンドの引数には、各コマンドで指定できる値を指定してください。それ以外の値が指定された場合、動作は保証されません。
- コマンドが正常に終了しても、エラーが発生している場合があります。メッセージログやスタックトレースログを確認してください。

## 3.2 Web サーバの運用で使用するコマンド

Web サーバの運用で使用するコマンドの書式、機能などについて説明します。

### 3.2.1 htpasswd

ユーザー名とパスワードのパスワードファイルへの登録およびパスワードの変更

#### 書式

```
htpasswd [-b] [-c|-D] パスワードファイル名 ユーザー名 [パスワード]
```

#### 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

#### 機能

パスワードファイルにユーザー名、パスワードを登録します。パスワードを変更したり、パスワードファイルに登録されたユーザー名を削除したりできます。

#### 実行権限

スーパーユーザー

#### 引数

-b

パスワードをコマンドラインに指定する場合に指定します。

-c

新規にパスワードファイルを作成する場合に指定します。すでに作成しているパスワードファイルにユーザーを追加する場合や、パスワードを変更する場合には、指定する必要はありません。

-D

ユーザーの登録を削除する場合に指定します。指定したパスワードファイルに、指定したユーザーが登録されている場合に、パスワードファイルから該当するユーザーを削除します。

#### パスワードファイル名

パスワードを登録、変更または削除するパスワードファイルを指定します。

#### ユーザー名

パスワードを登録、変更または削除するユーザー名を指定します。

## パスワード

登録または変更するパスワードを指定します。-b オプションを指定したときだけ指定できます。

## 使用例

パスワードファイル名と、登録するユーザー名またはパスワードを変更するユーザー名を指定してhtpasswdを起動すると、そのユーザーのパスワードの入力が要求されます。入力確認を含め、2回パスワードを入力すると、パスワードファイルにそのユーザーのユーザー名と、パスワードが登録されます。

```
$ "Application Serverインストールディレクトリー/httpsd/sbin/htpasswd" .passwd userxx ...1.  
New password: ...2.  
Re-type new password: ...3.  
Adding password for user userxx ...4.  
$
```

1. userxx のパスワードの変更
2. 新パスワード入力
3. 新パスワード再入力
4. 新パスワードの登録終了

登録を削除する場合は、-D オプション、パスワードファイル名および削除するユーザー名を指定してhtpasswdを起動します。

```
$ "Application Serverインストールディレクトリー/httpsd/sbin/htpasswd" -D .passwd userxx ...1.  
Deleting password for user userxx ...2.  
$
```

1. userxx の登録削除
2. userxx の登録削除終了

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 注意事項

- ユーザー名の最大長とパスワードの最大長は 128 文字です。
- htpasswd コマンド実行時は、パスワードファイルの作成先と同じディレクトリーに、作業ファイルが一時的に作成されます。作業ファイル名は、パスワードファイル名.プロセスID です。この作業ファイルは、htpasswd コマンドの終了時に自動的に削除されます。ただし、実行中にキャンセルした場合な

ど、作業ファイルが削除されないことがあります。作業ファイルが残っている場合は、手動で削除してください。

## 3.2.2 httpsd

Web サーバの起動

### 書式

```
httpsd [[-d ディレクトリー] [-f ファイル名] [-R ディレクトリー]]|-v|-t]
        [-D HWS_OPTION_HWS2]
```

### 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

### 機能

Web サーバを起動します。

デフォルト以外のhttpsd.conf ファイル名称を指定する場合またはサーバのルートディレクトリーやhttpsd.conf ファイルを指定して起動する場合に、この方法を使用します。

また、暗号化した秘密鍵を使って SSL を利用する場合は、起動時に秘密鍵のパスワード入力要求があります。

### 実行権限

スーパーユーザーまたは一般ユーザー

### 前提条件

PRF デーモンが起動している必要があります。

### 環境変数

- LD\_LIBRARY\_PATH(Linux)  
*Application Server*インストールディレクトリー/common/PRF/lib を指定してください。
- LIBPATH(AIX)  
*Application Server*インストールディレクトリー/common/PRF/lib を指定してください。
- PRFSP00L  
PRF デーモンの実行環境ディレクトリーを設定します。
- TZ



タイムゾーンを設定します。日本標準時の場合はJST-9 を指定します。

## 引数

### -d ディレクトリー

ServerRoot ディレクティブが設定ファイルに指定されていない場合の、デフォルト値を指定できます。

### -f ファイル名

httpsd.conf ファイルを指定できます。絶対パスまたはServerRoot ディレクティブの指定値からの相対パスで指定します。

### -R ディレクトリー

DSO 実行ライブラリーが格納されているディレクトリーを絶対パスで指定します。

### -v

バージョン情報を表示します。このオプションを指定した場合は、Web サーバは起動しません。

### -t

設定ファイルの文法チェックをします。文法エラーがあると、画面にエラーメッセージを表示します。このオプションを指定した場合は、Web サーバは起動しません。

### -D HWS\_OPTION\_HWS2

Red Hat Enterprise Linux Server 6 (64-bit x86\_64)、または AIX V7.1 で、SSL 機能を使用する場合に指定します。その他のプラットフォームや SSL 機能を使用しない場合は、指定しないでください。

## 起動確認方法

Web サーバの起動を確認するには、制御プロセスを確認してください。プロセス名（実行プログラム名）は、httpsd です。

また、Web サーバを起動したあと、エラーログにAH00163 のメッセージが出力されていることを確認してください。

## 再起動方法

kill コマンドで Web サーバを再起動できます。

```
kill {-HUP | -USR1} `cat PidFileディレクティブ指定値`
```

### -HUP

httpsdctl コマンドのrestart に相当する再起動をします。

### -USR1

httpsdctl コマンドのgraceful に相当する再起動をします。

### PidFileディレクティブ指定値

PidFile ディレクティブで指定した値（ファイル名）を指定します。

## 終了方法

httpsd コマンドで Web サーバを起動した場合、次に示すコマンドを実行してプロセスを終了し、Web サーバを停止してください。

```
kill {-TERM|-USR2} `cat PidFileディレクティブ指定値`
```

### -TERM

httpsdctl コマンドのstop に相当する停止をします。

### -USR2

httpsdctl コマンドのgracefulstop に相当する停止をします。

---

## 関連項目

- 3.4 一般ユーザーによる運用
- 

## 3.2.3 httpsdctl

Web サーバの起動、再起動および停止

### 書式

```
httpsdctl {start|stop|restart|graceful|gracefulstop|configtest|help}
```

### 格納先

*Application Server*インストールディレクトリー/*httpsd/sbin*

### 機能

Web サーバを起動、再起動および停止します。

### 実行権限

スーパーユーザーまたは一般ユーザー

### 前提条件

PRF デーモンが起動している必要があります。

### 環境変数

- LD\_LIBRARY\_PATH(Linux)

*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。

- LIBPATH(AIX)

*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。

- PRFSP00L

PRF デーモンの実行環境ディレクトリーを設定します。

- TZ

タイムゾーンを設定します。日本標準時の場合はJST-9 を指定します。

## ファイル

httpsd.conf ファイル

## 引数

### start

Web サーバを起動します。暗号化した秘密鍵を使って、SSL を利用する場合は、起動時に、秘密鍵のパスワードの入力要求があります。

### stop

Web サーバを停止します。

### restart

Web サーバを再起動します。実行中のサーバプロセスは、ただちに停止します。すべてのサーバプロセス終了後に再起動します。再起動時には、*MaxRequestWorkers* ディレクティブ指定値の変更は反映されなくて、前回の値が引き継がれます。*Listen* ディレクティブ指定値および SSL 通信で使用する秘密鍵の設定 (*SSLCertificateKeyFile* ディレクティブ) を変更した場合は、いったん Web サーバを停止してから、起動し直してください。

### graceful

Web サーバを再起動します。実行中のサーバプロセスは、実行終了後に停止します。サーバプロセスは、随時、新しい設定ファイルに基づいて起動します。再起動時には、*MaxRequestWorkers* ディレクティブ指定値の変更は反映されなくて、前回の値が引き継がれます。*Listen* ディレクティブ指定値および SSL 通信で使用する秘密鍵の設定 (*SSLCertificateKeyFile* ディレクティブ) を変更した場合は、いったん Web サーバを停止してから、起動し直してください。

### gracefulstop

Web サーバを停止します。実行中のサーバプロセスは、実行終了後に停止します。実行が終了しない場合は、*HWSGracefulStopTimeout* ディレクティブに指定した待ち時間が経過すると終了します。

### configtest

設定ファイルの文法チェックをします。文法エラーがあると、画面にエラーメッセージを表示します。このオプションを指定した場合は、Web サーバは起動しません。

### help

httpsdctl のヘルプテキストを表示します。

## 使用例

Web サーバを起動します。暗号化した秘密鍵を使用している場合はパスワードを入力します。

```
/opt/hitachi/APServer/httpsd/sbin/httpsdctl start  
Enter PEM pass phrase:
```

## 起動確認方法

Web サーバの起動を確認するには、制御プロセスを確認してください。プロセス名（実行プログラム名）は、httpsd です。

また、Web サーバを起動したあと、エラーログにAH00163 のメッセージが出力されていることを確認してください。

## 注意事項

- httpsdctl stop およびgracefulstop による Web サーバ停止操作実行時に、Web Server の設定ファイルの定義が不正な場合、httpsdctl の実行はエラーとなり Web サーバは停止しません。
- httpsdctl restart およびgraceful による Web サーバ再起動実行時に、Web Server の設定ファイルの定義が不正な場合、httpsdctl の実行はエラーとなり、Web サーバは停止しないで再起動しません。
- httpsdctl コマンドによる Web Server の起動、再起動および停止操作を実行した場合、起動完了および停止完了を示すメッセージは出力されません。

---

## 関連項目

- [3.4 一般ユーザーによる運用](#)

## 3.2.4 hwstraceinfo

内部トレースの採取

## 書式

```
hwstraceinfo -i 共有メモリー識別子 {-l ファイル名|-r}
```

## 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

## 機能

アプリケーションプログラムの実行時やリクエスト受け取り時など、システムで発生した事象は内部トレースとして採取されています。内部トレースは、共有メモリーにいったん出力され、その後ディレクティブの指定やコマンドによって、ファイルに出力されます。

### トレース情報の採取

Web サーバの各種事象発生を契機に、内部トレースが共有メモリーに採取されます。共有メモリーのメモリー識別子は、HWSTraceIdFile ディレクティブに指定したファイルに格納されます。

### ファイルへの出力方法

共有メモリーに採取された内部トレースは、サーバプロセスの異常終了時またはhwstraceinfo コマンドの実行によって、ファイルに出力されます。サーバプロセスが異常終了した場合は、HWSTraceLogFile ディレクティブで指定したファイルに出力されます。

hwstraceinfo コマンドでは、共有メモリーのメモリー識別子、出力先のファイル名を指定します。

内部トレースの出力ファイルサイズは次のとおりです。

`ps -efl` コマンドの出力サイズ+`vmstat` コマンドの出力サイズ+`ipcs -a` コマンドの出力サイズ+ 7KB  
×MaxRequestWorkers 値

## 実行権限

スーパーユーザーまたは一般ユーザー

## 引数

### -i 共有メモリー識別子

HWSTraceIdFile ディレクティブで指定したファイルに出力されている共有メモリー識別子を指定します。

### -l ファイル名

-i で指定した共有メモリー識別子に該当するトレースを出力するファイルを指定します。

### -r

-i で指定した共有メモリー識別子に割り当てられている共有メモリーを解放します。

Web サーバが終了してもトレース用の共有メモリーは残ります。残った共有メモリーを解放するために、このオプションを使用します。

## 使用例

共有メモリー識別子1800\_1133780652\_0 に該当するトレースを、traceinfo.log ファイルに出力する例を示します。

```
hwstraceinfo -i 1800_1133780652_0 -l traceinfo.log
```

## 注意事項

Web サーバが終了しても、トレース情報を残すために共有メモリーは解放しません。また、Web サーバを再起動する場合は、共有メモリーが再利用されます。

Web サーバを停止したあとに起動した場合は、HWSTraceIdFile ディレクティブに指定したファイルの値を基に、いったん共有メモリーを解放して、再度確保します。ただし、次のような場合は、以前使用していた共有メモリーが解放できなくなりますので、注意してください。

- 同一ユーザーで再起動していない (User ディレクティブまたは Group ディレクティブの値が変更されている)
- HWSTraceIdFile ディレクティブの値を変更している
- HWSTraceIdFile ディレクティブで指定していたファイルが消去されている

共有メモリーを解放する場合は、-r を指定したhwstraceinfo コマンドを実行してください。

## 3.2.5 logresolve

アクセスログファイル内の IP アドレスのホスト名への変換

### 書式

```
logresolve [-s ファイル名] [-c] < アクセスログファイル名 > 新ログファイル名
```

### 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

### 機能

logresolve コマンドは、レコードの先頭が IP アドレスであるアクセスログファイル内の IP アドレスをホスト名に変換し、新規ログファイルに出力します。変換規則は、ホスト名のルックアップの逆引きによります。

### 実行権限

スーパーユーザー

### 引数

-s ファイル名

変換したときの情報を出力するファイルを指定します。

変換後のホスト名が、変換前の IP アドレスと一致するかどうかチェックする場合に指定します。

#### アクセスログファイル名

入力ログファイル名を指定します。入力したファイルの IP アドレスから、ホスト名のルックアップの逆引きをします。レコードの先頭は、必ず IP アドレスでなければなりません。ホスト名の検索に失敗した場合、新ログファイルには IP アドレスを出力します。

#### 新ログファイル名

IP アドレスをホスト名に変換したアクセスログを出力するファイル名を指定します。

## 使用例

logs¥access.log に格納しているアクセスログ内の IP アドレスをホスト名に変換します。

アクセスログファイル：logs¥access.log

新ログファイル：logs¥new\_access.log

```
logresolve < logs¥access.log > logs¥new_access.log
```

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 3.3 SSL による認証、暗号化で使用するコマンド

SSL による認証、暗号化で使用するコマンドの書式、機能などについて説明します。

### 3.3.1 hwscertutil cert -hash

ハッシュリンクの作成

#### 書式

```
hwscertutil cert -noout -hash -in CAの証明書ファイル
```

#### 格納先

*Application Server*インストールディレクトリー/*httpsd/sbin*

#### 機能

証明書の妥当性チェックのために、証明書を発行した CA の証明書を `SSLCACertificateFile` ディレクティブまたは `SSLCACertificatePath` ディレクティブで指定します。 `SSLCACertificatePath` ディレクティブには、証明書発行元の CA の証明書をポイントするハッシュ値を使用したシンボリックリンク（ハッシュリンク）を格納したディレクトリーを指定します。

ハッシュ値は `hwscertutil cert` コマンドで作成します。

`SSLCACertificatePath` ディレクティブを指定すると、Web サーバでの証明書の検索はハッシュ値を用いて効率良く実行できます。したがって、CA の証明書が多い場合は、`SSLCACertificateFile` ディレクティブよりも `SSLCACertificatePath` ディレクティブを推奨します。なお、ハッシュ値は 1 つの証明書に 1 つである必要があるため、ハッシュリンク作成時には、複数の証明書が混在したファイルは指定できません。

`SSLCACertificatePath` ディレクティブで指定するハッシュリンクディレクトリー内のシンボリックリンク生成時には、ハッシュ値に `.0` を付ける必要があります。また、`SSLCACertificatePath` ディレクティブで指定するディレクトリーは、`User`、`Group` ディレクティブで指定したユーザーでアクセスできるように、ディレクトリーに読み込み権限、実行権限を設定してください。

#### 実行権限

スーパーユーザー

#### 引数

`-in` *CAの証明書ファイル*

ハッシュリンク値を作成する CA の証明書ファイルを指定します。



## 使用例

ハッシュリンクのディレクトリーおよび CA の証明書が次に示すディレクトリー、ファイルの場合の例を示します。

/opt/hitachi/APServer/httpsd/conf/ssl/cacerts : ハッシュリンクディレクトリー

/opt/hitachi/APServer/httpsd/conf/ssl/cacert/cacert.pem : CA の証明書

```
cd /opt/hitachi/APServer/httpsd/conf/ssl/cacerts
ln -s /opt/hitachi/APServer/httpsd/conf/ssl/cacert/cacert.pem `hWSCertUtil cert
-noout -hash -in /opt/hitachi/APServer/httpsd/conf/ssl/cacert/cacert.pem`.0
```

これによって、/opt/hitachi/APServer/httpsd/conf/ssl/cacert/cacert.pem についてのハッシュリンク xxxxxxxx.0 が作成されます。

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 3.3.2 hWSCertUtil cert -outform

証明書の形式変換

### 書式

```
hWSCertUtil cert -inform 入力形式 -outform 出力形式
-in 入力ファイル -out 出力ファイル
```

### 格納先

Application Serverインストールディレクトリー/httpsd/sbin

### 機能

証明書の形式を変換します。必要に応じて使用します。

### 実行権限

スーパーユーザー

## 引数

### -inform 入力形式

入力形式を指定します。

指定できる値を次に示します。

- DER
- PEM

デフォルト値：PEM

### -outform 出力形式

出力形式を指定します。

指定できる値を次に示します。

- DER
- PEM

デフォルト値：PEM

### -in 入力ファイル

変換前の証明書ファイルを指定します。

### -out 出力ファイル

変換後の証明書ファイルを指定します。

## 3.3.3 hwscertutil cert -text

証明書の内容表示

### 書式

```
hwscertutil cert -in 証明書ファイル -text
```

### 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

### 機能

証明書ファイルの内容を表示します。

"-----BEGIN CERTIFICATE-----"から、"-----END CERTIFICATE-----"の証明書ファイルの内容を表示します。

## 実行権限

スーパーユーザー

## 引数

-in 証明書ファイル

表示する証明書ファイルを指定します。

## 使用例

```
hwscertutil cert -in httpsd.pem -text
```

httpsd.pem：表示する証明書ファイル

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 3.3.4 hwscertutil req

証明書発行要求（CSR）の内容を表示

## 書式

```
hwscertutil req -in CSRファイル -text
```

## 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

## 機能

証明書発行要求（CSR）の内容を表示します。

## 実行権限

スーパーユーザー

## 引数

`-in CSRファイル`

表示する CSR ファイルを指定します。

## 使用例

```
hwscertutil req -in httpsd.csr -text
```

httpsd.csr : 表示する CSR ファイル

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 3.3.5 hwscertutil reqgen

証明書発行要求 (CSR) の作成

## 書式

```
hwscertutil reqgen [-sign {MD5|SHA1|SHA224|SHA256|SHA384|SHA512}]  
-key 鍵ファイル -out CSRファイル
```

## 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

## 機能

hwscertutil reqgen コマンドを使用して、証明書発行要求 (CSR) を作成します。ここで作成した CSR ファイルを CA に提出して、署名済みの証明書を発行してもらいます。CSR の形式は PKCS#10 に準拠しています。

## 実行権限

スーパーユーザー

## 引数

`-sign {MD5|SHA1|SHA224|SHA256|SHA384|SHA512}`

CSR 作成時に使用する署名アルゴリズムを指定します。

指定できる値を次に示します。

- -MD5 : md5WithRSAEncryption を使用します。
- -SHA1 : sha1WithRSAEncryption を使用します。
- -SHA224 : sha224WithRSAEncryption を使用します。
- -SHA256 : sha256WithRSAEncryption を使用します。
- -SHA384 : sha384WithRSAEncryption を使用します。
- -SHA512 : sha512WithRSAEncryption を使用します。

デフォルト値 : SHA1

-key *鍵ファイル*

Web サーバの秘密鍵のファイルを指定します。

-out *CSRファイル*

作成した CSR を出力するファイルを指定します。

## 使用例

証明書発行要求 (CSR) を作成します。

```
hwscertutil reqgen -sign SHA1 -key httpsdkey.pem -out httpsd.csr
```

httpsdkey.pem : 鍵ファイル

httpsd.csr : CSR ファイル

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 3.3.6 hwskeygen

Web サーバの秘密鍵の作成

### 書式

```
hwskeygen -rand ファイル名[:ファイル名]... [-des|-des3] -out 鍵ファイル  
[-bits {512|1024|2048|4096}]
```

## 格納先

*Application Server*インストールディレクトリー/httpsd/sbin

## 機能

hwskeygen コマンドを使用して、Web サーバの秘密鍵を作成します。作成した Web サーバの秘密鍵のファイルは、SSLCertificateKeyFile ディレクティブに指定します。

## 実行権限

スーパーユーザー

## 引数

-rand *ファイル名[:ファイル名]...*

乱数生成に利用する任意のファイルを指定します。乱数生成用のファイルは、十分大きい適当なファイルを指定してください。

-des|-des3

秘密鍵を暗号化する場合、暗号種別を指定します。このオプションを指定すると、秘密鍵作成時にパスワードの入力要求があります。パスワードは 64 文字以内です。また、証明書発行要求 (CSR) の作成 (hwscertutil reqgen コマンド) 時および Web サーバ起動時にパスワードの入力要求があります。

なお、Web サーバ起動時のパスワードの入力は省略できます。パスワードの入力を省略するには、sslpaswd コマンドで作成したパスワードファイルを指定したSSLCertificateKeyPassword ディレクティブを、サーバ秘密鍵ファイルを指定したSSLCertificateKeyFile ディレクティブとともにhttpsd.conf に設定します。

-des を指定した場合、暗号種別として DES(Data Encryption Standard)が選択されます。-des3 を指定した場合、トリプル DES が選択されます。Web サーバと Web ブラウザー間の通信での暗号種別とは関係ありません。

-out *鍵ファイル*

Web サーバの秘密鍵を出力するファイルを指定します。

-bits {512|1024|2048|4096}

作成する Web サーバの秘密鍵のビット長を指定します。  
指定できる値を次に示します。

- 512
- 1024
- 2048
- 4096

デフォルト値：1024

## 使用例

Web サーバの秘密鍵 `httpsdkey.pem` を作成します。

```
hwskeygen -rand file1:file2:file3:file4:file5 -out httpsdkey.pem -bits 1024
```

`file1`、`file2`、`file3`、`file4`、`file5` : 任意のファイル

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 関連項目

- [3.3.5 hwscertutil reqgen](#)
- [3.3.7 sslpasswd](#)

## 3.3.7 sslpasswd

パスワード付きサーバ秘密鍵のパスワードファイル作成

## 書式

```
sslpasswd サーバ秘密鍵ファイル名 パスワードファイル名
```

## 格納先

*Application Server*インストールディレクトリー/`httpsd/sbin`

## 機能

パスワードによって保護されているサーバ秘密鍵を使用する場合、パスワードをあらかじめファイルに格納しておき、ディレクティブを設定することで、Web サーバ起動時のパスワード入力を省略できます。その手順を次に示します。

1. `hwskeygen` コマンドによって、パスワード付きのサーバ秘密鍵を作成します。
2. `sslpasswd` コマンドによって、パスワードファイルを作成します。
3. 作成したパスワードファイルを指定した `SSLCertificateKeyPassword` ディレクティブを、サーバ秘密鍵ファイルを指定した `SSLCertificateKeyFile` ディレクティブとともに `httpsd.conf` に設定します。
4. Web サーバを起動または再起動します。

パスワードファイルの内容の漏洩には注意する必要があります。サーバ秘密鍵の格納ディレクトリーに加え、パスワードファイルの格納ディレクトリーでも、ほかのユーザーからのアクセスを禁止するように、ディレクトリーパーミッションやファイルパーミッションの設定をしてください。

## 実行権限

スーパーユーザー

## 引数

サーバ秘密鍵ファイル名

パスワードによって保護されたサーバ秘密鍵を指定します。

パスワードファイル名

パスワードファイルを出力するファイル名を指定します。

## 使用例

```
sslpaswd httpsdkey.pem .keypasswd
```

## 戻り値

戻り値	説明
0	正常終了
0 以外	異常終了

## 注意事項

- パスワードファイル名として、既存のファイル名は指定できません。
- Windows 版のsslpaswd コマンドで作成したパスワードファイルは、UNIX 版では使用できません。
- UNIX 版のsslpaswd コマンドで作成したパスワードファイルは、Windows 版では使用できません。



## 3.4 一般ユーザーによる運用

スーパーユーザーと一般ユーザーの違い、一般ユーザーによる Web サーバを運用するための環境構築方法などについて説明します。

### Web サーバを運用するユーザー

Web サーバは、通常の運用方法として、スーパーユーザーによる運用を想定しています。

インストールした状態では、スーパーユーザーによる運用ができるように各種設定が施されています。

このことから、スーパーユーザー以外のユーザー（以下、一般ユーザーと呼びます）で運用する場合、Web サーバの設定ファイルや関連するディレクトリー・ファイルの各種設定内容の変更が必要になります。また、Web サーバの一部の機能については、一般ユーザーによる運用は制限事項になるものがあります。

ここでは、スーパーユーザーと一般ユーザーの違い、一般ユーザーによる Web サーバを運用するための環境構築方法などについて説明します。

### 各プロセスの権限

スーパーユーザーまたは一般ユーザーで運用した場合、Web サーバの各プロセスの権限を次に示します。

項番	プロセス	スーパーユーザーによる運用	一般ユーザーによる運用
1	制御プロセス	スーパーユーザー  User、Group ディレクティブで指定したユーザー、グループ	一般ユーザー
2	rotatelog、rotatelog2 プロセス		
3	サーバプロセス		
4	CGI プロセス		
5	gcache サーバ		

### UNIX でのスーパーユーザーと一般ユーザーの違い

UNIX では、スーパーユーザーは一般ユーザーと異なり、システムの管理者権限を持つユーザーになります。UNIX でのスーパーユーザーと一般ユーザーの権限の差異（一例）を次に示します。

項番	項目	スーパーユーザーによる運用	一般ユーザーによる運用
1	別のユーザーが起動したプロセスの停止	可	不可
2	well-known ポート（1023 番以下のポート）を開く	可	不可
3	明示的に読み取り/書き込み権限が与えられていないファイルへのアクセス	可	不可

一般ユーザーで Web サーバを運用する場合、Web サーバの制御プロセスの権限が一般ユーザー権限で動作するため、このときの挙動はスーパーユーザーで Web サーバを運用した場合と異なる場合があります。したがって、一般ユーザーで Web サーバを運用する場合は、スーパーユーザーとの権限の差異を意識しながら環境を構築する必要があります。

## リソースの所有者・グループの変更

Web サーバのコンテンツ、設定ファイル類、および Web サーバが動作する際にアクセスする各種ファイル・ディレクトリーについて、UNIX 上での所有者・グループを変更します。

最低限、インストールディレクトリー (*Application Server* インストールディレクトリー/*httpsd* ディレクトリー) 以下のリソースに対しては変更が必要です。

将来、リソースの所有者・グループを元に戻したい場合は、変更作業の前に現在のリソースに対して、所有者とグループを保存しておきます。

保存作業は、スーパーユーザーで実行します。保存例を以下に示します。

(例)

`/opt/hitachi/APServer/httpsd` ディレクトリー以下のリソースに対して、所有者とグループの一覧を作成する。

```
ls laR /opt/hitachi/APServer/httpsd
```

変更作業は、スーパーユーザーで実行します。変更例を以下に示します。

(例)

`/opt/hitachi/APServer/httpsd` ディレクトリー以下のリソースに対して、所有者(*hwsuser*)とグループ(*hwsgroup*)を変更する。

```
chown R hwsuser:hwsgroup /opt/hitachi/APServer/httpsd
```

## 制限事項

次に示すコマンドは、一般ユーザーによる運用に対応していません。スーパーユーザーで運用してください。

- `hwscertutil` コマンド
- `htpasswd` コマンド
- `hwskeygen` コマンド
- `logresolve` コマンド
- `sslpaswd` コマンド

一般ユーザーによる運用では次に示すディレクティブは指定できません。指定があっても無視します。

- Group ディレクティブ
- User ディレクティブ

一般ユーザーによる運用では、well-known ポート（1023 番以下のポート）を開くことができません。

次のディレクティブにポート番号を指定する際は注意してください。

- Listen ディレクティブ
- SSLCacheServerPort ディレクティブ

# 4

## Java VM で使用するコマンド

この章では、Java VM で使用するコマンドの書式、機能などについて説明します。

## 4.1 Java VM で使用するコマンドの一覧

Java VM で使用するコマンドの一覧を次の表に示します。

### システムの運用で使用するコマンド

コマンド名称	要約	概要説明
car_tar_gz	core アーカイブ機能	core ファイルと関連するライブラリーなどをまとめて、1つのアーカイブファイルに出力します。gzip コマンドで圧縮します。
car_tar_Z	core アーカイブ機能	core ファイルと関連するライブラリーなどをまとめて、1つのアーカイブファイルに出力します。compress コマンドで圧縮します。
eheapprof	Explicit ヒープ詳細情報付き拡張スレッドダンプの出力	指定したプロセス ID の java プロセスについて、Explicit ヒープ詳細情報を含んだ拡張スレッドダンプを出力します。
java_hras	Application Server と連携しない Java アプリケーションプログラムの実行	トラブルシュートに関する独自機能を有効にして Java VM を起動します。
javacore	core ファイルとスレッドダンプの取得	コマンド実行時の core ファイルとスレッドダンプを同時に取得します。
javagc	GC の強制発生	指定したプロセス ID の java プロセスに対して、任意のタイミングで FullGC を発生させます。
javatrace	トレース情報の収集	core ファイルからスタックトレース情報を取得します。
jheapprof	クラス別統計情報付き拡張スレッドダンプの出力	指定したプロセス ID の java プロセスについて、クラス別統計情報を含んだ拡張スレッドダンプを出力します。
jheapprofanalyzer	クラス別統計情報解析ファイルの CSV 出力	クラス別統計情報解析ファイルを CSV 形式で出力します。

#### ❗ 重要

- コマンドの引数には、各コマンドで指定できる値を指定してください。それ以外の値が指定された場合、動作は保証されません。
- コマンドが正常に終了しても、エラーが発生している場合があります。メッセージログやスタックトレースログを確認してください。

## 4.2 システムの運用で使用するコマンド

---

システムの運用で使用するコマンドの書式、機能などについて説明します。

### 4.2.1 car\_tar\_gz

core アーカイブ機能

#### 書式

```
car_tar_gz [-i 実行ファイル名] [-f] [-s|-S] [coreファイル名] >  
          出力ファイル名
```

#### 格納先

*Application Server*インストールディレクトリー/jdk/jre/bin/

#### 機能

core ファイルと関連するライブラリーなどをまとめて、1つのアーカイブファイルに出力します。car\_tar\_gz コマンドは、gzip コマンドを使用して圧縮します。

#### 引数

-i 実行ファイル名

実行ファイル名を指定します。

-f

パス名付きでライブラリーなどのファイルを取得します。

-s

出力するアーカイブファイルのサイズ（単位：バイト）を報告します。

この場合、アーカイブファイルは出力しません。

-S

アーカイブファイルを作成するための、シェルスクリプトを出力します。

このオプションを指定する場合、出力ファイル名には、シェルスクリプト名を指定してください。出力されたシェルスクリプトを実行すると、アーカイブファイルが出力されます。

coreファイル名

core ファイル名を指定します。

デフォルト値：core

## 出力ファイル名

アーカイブファイル、またはシェルスクリプトのファイル名を指定します。

## 入力例

アーカイブファイルを作成する場合

```
car_tar_gz -f core.8326 > corefile.tar.gz
```

アーカイブファイルのサイズを調べる場合

```
car_tar_gz -s core.8326
```

シェルスクリプトからアーカイブファイルを作成する場合

1. シェルスクリプトを作成します。

```
car_tar_gz -S core.8326 > collect_cores.csh
```

2. シェルスクリプトに実行権を付与します。

```
chmod +x ./collect_cores.csh
```

3. シェルスクリプトを実行して、アーカイブファイルを作成します。

```
./collect_cores.csh > corefile.tar.gz
```

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。

## 出力メッセージ

次のメッセージを出力した場合、正常なアーカイブファイルは出力されません。

項番	エラーメッセージ	説明
1	usage: car_tar_gz [ -i executable-file ] [ -f ] [ -s   -S ] [ core-file ]	コマンドの引数が不正です。
2	car_tar_gz: inner error!	car_tar_gz コマンドの内部エラーです。
3	car_tar_gz: cannot create temporary name	一時的に使用するファイルが作成できません。カレントディレクトリーにある、ファイルcar_exec?? (??は00~99)を削除するか、移動してください。
4	car_tar_gz: ~ : file not found	ファイルが見つかりません。core ファイルおよび-i オプションで指定した実行ファイルを確認してください。
5	car_tar_gz: ~ : not supported platform	サポートしていない OS です。

## 注意事項

- AIX では使用できません。snapcore コマンドを使用してください。
- カレントディレクトリーに書き込み権限が必要です。
- 同じディレクトリー下で、同じ core ファイルまたは作成時間が同じ core ファイルに対して、car\_tar\_Z コマンドおよびcar\_tar\_gz コマンドは同時に実行できません。
- システムにtar コマンドおよびgzip コマンドがインストールされている必要があります。
- car\_tar\_gz コマンド実行中に強制終了した場合、カレントディレクトリーにcar\_tar\_gz コマンドが作成した一時的なファイルやディレクトリーが残ります。  
一時的に作成されるファイル：car\_exec?? (??は、00～99)  
一時的に作成されるディレクトリー：carYYMMDDhhmm/ (YYMMDDhhmm は、core ファイルの作成年月日時分)

## 4.2.2 car\_tar\_Z

core アーカイブ機能

### 書式

```
car_tar_Z [-i 実行ファイル名] [-f] [-s|-S] [coreファイル名] >  
          出力ファイル名
```

### 格納先

Application Serverインストールディレクトリー/jdk/jre/bin/

### 機能

core ファイルと関連するライブラリーなどをまとめて、1つのアーカイブファイルに出力します。car\_tar\_Z コマンドは、compress コマンドを使用して圧縮します。

### 引数

-i 実行ファイル名

実行ファイル名を指定します。

-f

パス名付きでライブラリーなどのファイルを取得します。

-s

出力するアーカイブファイルのサイズ（単位：バイト）を報告します。

この場合、アーカイブファイルは出力しません。



-S

アーカイブファイルを作成するための、シェルスクリプトを出力します。

このオプションを指定する場合、*出力ファイル名*には、シェルスクリプト名を指定してください。出力されたシェルスクリプトを実行すると、アーカイブファイルが出力されます。

#### coreファイル名

core ファイル名を指定します。

デフォルト値：core

#### 出力ファイル名

アーカイブファイル、またはシェルスクリプトのファイル名を指定します。

## 入力例

アーカイブファイルを作成する場合

```
car_tar_Z -f core.8326 > corefile.tar.Z
```

アーカイブファイルのサイズを調べる場合

```
car_tar_Z -s core.8326
```

シェルスクリプトからアーカイブファイルを作成する場合

1. シェルスクリプトを作成します。

```
car_tar_Z -S core.8326 > collect_cores.csh
```

2. シェルスクリプトに実行権を付与します。

```
chmod +x ./collect_cores.csh
```

3. シェルスクリプトを実行して、アーカイブファイルを作成します。

```
./collect_cores.csh > corefile.tar.Z
```

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。

## 出力メッセージ

次のメッセージを出力した場合、正常なアーカイブファイルは出力されません。

項番	エラーメッセージ	説明
1	usage: car_tar_Z [ -i executable-file ] [ -f ] [ -s   -S ] [ core-file ]	コマンドの引数が不正です。

項番	エラーメッセージ	説明
2	car_tar_Z: inner error!	car_tar_Z コマンドの内部エラーです。
3	car_tar_Z: cannot create temporary name	一時的に使用するファイルが作成できません。カレントディレクトリーにある、ファイルcar_exec?? (??は00~99) を削除するか、移動してください。
4	car_tar_Z: ~ : file not found	ファイルが見つかりません。core ファイルおよび-i オプションで指定した実行ファイルを確認してください。
5	car_tar_Z: ~ : not supported platform	サポートしていない OS です。

## 注意事項

- AIX では使用できません。snapcore コマンドを使用してください。
- カレントディレクトリーに書き込み権限が必要です。
- 同じディレクトリー下で、同じ core ファイルまたは作成時間が同じ core ファイルに対して、car\_tar\_Z コマンドおよびcar\_tar\_gz コマンドは同時に実行できません。
- システムにtar コマンドおよびcompress コマンドがインストールされている必要があります。
- car\_tar\_Z コマンド実行中に強制終了した場合、カレントディレクトリーにcar\_tar\_Z コマンドが作成した一時的なファイルやディレクトリーが残ります。  
一時的に作成されるファイル：car\_exec?? (??は、00~99)  
一時的に作成されるディレクトリー：carYYMMDDhhmm/ (YYMMDDhhmm は、core ファイルの作成年月日時分)

## 4.2.3 eheapprof

Explicit ヒープ詳細情報付き拡張スレッドダンプの出力

### 書式

```
eheapprof [-i|-f] [-force] [-freeratio] -p プロセスID
```

### 格納先

Application Serverインストールディレクトリー/jdk/jre/bin/

### 機能

引数に指定したプロセス ID の java プロセスについて、Explicit ヒープ詳細情報を含んだ拡張スレッドダンプを出力します。また、Explicit メモリーブロック内のオブジェクトの統計情報、および Explicit メモリーブロックの解放率情報も拡張スレッドダンプに出力できます。

## 実行権限

引数のプロセス ID で指定する java プロセスと同一のユーザーで実行してください。

## 引数

-i

Explicit ヒープ詳細情報付き拡張スレッドダンプの出力処理の実行を確認するメッセージが表示されます。表示されたメッセージに対して *y* または *n* を入力します。このとき、*y* を入力すると、Explicit ヒープ詳細情報を含んだ拡張スレッドダンプが出力されます。*n* を入力すると、何も出力しないで処理を終了します。省略した場合、*-f* オプションが指定されないかぎり、このオプションは有効です。

-f

*-i* オプションを無効にします。省略した場合、*-i* オプションが有効になります。

-force

java プロセスが作成する `/tmp/hsperfdata_ユーザー名/プロセスID` ファイルの有無の確認をしないで、*-p* オプションで指定したプロセス ID の java プロセスに対して拡張スレッドダンプの出力を要求します。

-freeratio

Explicit メモリーブロックのオブジェクト解放率情報の出力を有効にして、拡張スレッドダンプを出力します。

このオプションを指定して `eheapprof` コマンドを実行すると、Java VM によって次の処理が実行され、オブジェクト解放率情報が取得されます。

- FullGC
- Explicit メモリーブロックの解放処理

これらの処理によって、アプリケーションの実行が数秒間止まるおそれがあるため、Explicit メモリーブロックのオブジェクト解放率情報は、システム開発時や業務停止時間中に出力することをお勧めします。

-p *プロセスID*

Explicit ヒープ詳細情報を含んだ拡張スレッドダンプを出力する java プログラムのプロセス ID を指定します。

型：Integer

指定できる値を次に示します。

- 0 ~ 4294967295

## 出力形式

「Explicit メモリーブロックのオブジェクト解放率情報」が無効な場合

```
Explicit Heap Status
```

```
max EH_MAX, total EH_TOTAL, used EH_USED, garbage EH_GARB (EH_PER1
used/max, EH_PER2 used/total, EH_PER3 garbage/used), EM_NUMS spaces exist
```

Explicit Memories(EM\_MGR\_PTR)

```
"EM_NAME" eid=EID(EM_PTR)/EM_TYPE, total EM_TOTAL, used EM_USED,
garbage EM_GARB (EM_PER1 used/total, EM_PER2 garbage/used, FL_BLOCKS
blocks) EM_STAT
```

deployed objects

Size	Instances	Class
ISIZE	INUM	CNAME
...		
AISIZE	AINUM	total

「Explicit メモリーブロックのオブジェクト解放率情報」が有効な場合

Explicit Heap Status

```
max EH_MAX, total EH_TOTAL, used EH_USED, garbage EH_GARB (EH_PER1
used/max, EH_PER2 used/total, EH_PER3 garbage/used), EM_NUMS spaces exist
```

Explicit Memories(EM\_MGR\_PTR)

```
"EM_NAME" eid=EID(EM_PTR)/EM_TYPE, total EM_TOTAL, used EM_USED,
garbage EM_GARB (EM_PER1 used/total, EM_PER2 garbage/used, FL_BLOCKS
blocks) EM_STAT
```

deployed objects

Size	Instances	FreeRatio	Class
ISIZE	INUM	FRATIO	CNAME
...			
AISIZE	AINUM	total	

出力形式で示した各項目について説明します。

分類	出力項目	出力内容	意味
Explicit ヒープ情報	EH_MAX	<const>K	Explicit ヒープの最大サイズが出力されます。単位はキロバイトです。
	EH_TOTAL	<const>K	確保済み Explicit ヒープサイズが出力されます。単位はキロバイトです。
	EH_USED	<const>K	利用済み Explicit ヒープサイズが出力されます。単位はキロバイトです。
	EH_GARB	<const>K	Explicit ヒープの内部状態が出力されます。
	EH_PER1	<decimal> %	Explicit ヒープ利用率 (EH_USED/EH_MAX) が%表記で出力されます。
	EH_PER2	<decimal> %	Explicit ヒープ利用率 (EH_USED/EH_TOTAL) が%表記で出力されます。
	EH_PER3	<decimal> %	Explicit ヒープの内部状態が出力されます。
	EM_NUMS	<const>	有効な Explicit メモリーブロックの数が出力されます。

分類	出力項目	出力内容	意味
	<i>EM_MGR_PTR</i>	<ptr>	Explicit ヒープ制御のための内部情報があるメモリアドレスが出力されます。障害調査などに利用します。
Explicit メモリーブロック情報	<i>EM_NAME</i>	<letters>	Explicit メモリーブロックの名称が出力されます。 Explicit メモリーブロックの名称に多バイト文字が含まれている場合、出力内容は不定です（通常は文字化けして出力されます）。 Explicit メモリーブロックの初期化とほぼ同時に出力された場合や、Java VM が内部で生成した Explicit メモリーブロックの場合は、"NULL"が出力されることがあります。
	<i>EID</i>	<const>	Explicit メモリーブロックの ID が出力されます。
	<i>EM_PTR</i>	<ptr>	Explicit メモリーブロック内部構造があるメモリアドレスが出力されます。障害調査などに利用します。
	<i>EM_TYPE</i>	R B A	Explicit の種別が出力されます。 R は、Application Server の内部で利用されている Explicit メモリーブロックを示します。 B は、アプリケーションが利用している Explicit メモリーブロックを示します。 A は、自動配置設定ファイルを使って指定した Explicit メモリーブロックを示します。
	<i>EM_TOTAL</i>	<const>K	Explicit メモリーブロックのメモリー確保済みサイズが出力されます。単位はキロバイトです。
	<i>EM_USED</i>	<const>K	Explicit メモリーブロックの利用済みサイズが出力されます。単位はキロバイトです。
	<i>EM_GARB</i>	<const>K	Explicit メモリーブロックの内部状態が出力されます。単位はキロバイトです。
	<i>EM_PER1</i>	<decimal> %	Explicit メモリーブロック利用率 ( $EM\_USED/EM\_TOTAL$ ) が%表記で出力されます。
	<i>EM_PER2</i>	<decimal> %	Explicit メモリーブロックの内部状態が出力されます。
		<i>FL_BLOCK_S</i>	<const>
	<i>EM_STAT</i>	Enable  Disable	Explicit メモリーブロックのサブ状態が出力されます。
オブジェクト統計情報※1	<i>ISIZE</i>	<const>	あるクラスをインスタンス化したオブジェクトの Explicit メモリーブロック内のサイズが出力されます。
	<i>INUM</i>	<const>	あるクラスをインスタンス化したオブジェクトの Explicit メモリーブロック内の個数が出力されます。
	<i>CNAME</i>	<letters>	<i>ISIZE</i> および <i>INUM</i> が示すクラスの完全クラス名が出力されます。
	<i>AISIZE</i>	<const>	Explicit メモリーブロック内の全オブジェクトの合計サイズが出力されます。
	<i>AINUM</i>	<const>	Explicit メモリーブロック内の全オブジェクトの個数が出力されます。

分類	出力項目	出力内容	意味
オブジェクト解放率情報※ 2	FRATIO	<decimal> %	Explicit メモリーブロックの自動解放処理で解放されたオブジェクトの割合（オブジェクト解放率）が%表記で出力されます。 オブジェクト解放率=（自動解放処理前のクラスのオブジェクト数-自動解放処理後のクラスのオブジェクト数）/自動解放処理前のクラスのオブジェクト数×100 なお、オブジェクト解放率情報出力時に、自動解放処理の対象とならなかった Explicit メモリーブロックには、「-」が出力されます。

(凡例)

- <const>K：正の自然数を表します。
- <decimal>K：正の整数（小数点第1位まで）を表します。
- <ptr>K：ポインター値を表します。
- <letters>K：任意の文字または任意の文字列を表します。

注※1

オブジェクト統計情報は、`eheapprof` コマンドを実行している場合に出力されます。なお、オブジェクト統計情報には、実際に作成したサイズ、個数よりも多く、`int` 型配列を示す"`[I`"が出力されることがあります。この場合、"`[I`"は、Explicit メモリーブロック内で使用されていないオブジェクトを示します。Explicit メモリーブロック内で使用されていないオブジェクトは、Java VM の内部処理で `int` 型配列化されます。

注※2

オブジェクト解放率情報は、`-freeratio` オプションを指定した `eheapprof` コマンドを実行している場合に出力されます。

## 出力例

「Explicit メモリーブロック内のオブジェクト統計情報」および「Explicit メモリーブロックのオブジェクト解放率情報」が無効な場合

```
Explicit Heap Status
-----
max 31415926K, total 213971K, used 205369K, garbage 1234K (1.1% used/max,
96.2% used/total, 0.0% garbage/used), 3 spaces exist

Explicit Memories(0x12345678)

"EJBMgrData" eid=1(0x02f25610)/R, total 154272K, used 150176K, garbage 1234K
(97.0% used/total, 1.2% garbage/used, 0 blocks) Enable

"VJBStored" eid=3(0x02f25910)/B, total 54272K, used 50176K, garbage 0K (90.9%
used/total, 0.0% garbage/used, 2 blocks) Enable

"ExplicitMemory-2" eid=2(0x02f25700)/R, total 5427K, used 5017K, garbage 0K
(92.1% used/total, 0.0% garbage/used, 0 blocks) Enable
```

「Explicit メモリーブロック内のオブジェクト統計情報」および「Explicit メモリーブロックのオブジェクト解放率情報」が有効な場合

#### Explicit Heap Status

max 31415926K, total 162816K, used 150528K, garbage 10004K (0.0% used/max, 91.1% used/total, 6.6% garbage/used), 3 spaces exist

#### Explicit Memories(0x12345678)

"EJBMgrData" eid=1(0x02f25610)/R, total 54272K, used 50176K, garbage 0K (91.2% used/total, 0.0% garbage/used, 0 blocks)

deployed objects

Size	Instances	FreeRatio	Class
35234568	10648	-	java.util.HashMap
5678900	10668	-	[Ljava.util.HashMap\$Entry;
4456788	7436	-	java.util.HashMap\$Entry
4321000	200	-	java.util.WeakHashMap
1234568	190	-	[Ljava.util.WeakHashMap\$Entry;
454400	4	-	java.util.WeakHashMap\$Entry
51380224	29146		total

"VJBStored" eid=3(0x02f25910)/B, total 54272K, used 50176K, garbage 10004K (90.7% used/total, 19.9% garbage/used, 5 blocks)

deployed objects

Size	Instances	FreeRatio	Class
35234568	10648	49	java.util.HashMap
5678900	10668	43	[Ljava.util.HashMap\$Entry;
4456788	7436	50	java.util.HashMap\$Entry
4321000	200	32	java.util.WeakHashMap
1234568	190	45	[Ljava.util.WeakHashMap\$Entry;
454400	4	22	java.util.WeakHashMap\$Entry
51380224	29146		total

"ExplicitMemory-2" eid=2(0x02f25700)/B, total 54272K, used 50176K, garbage 0K (91.1% used/total, 0.0% garbage/used, 0 blocks)

deployed objects

Size	Instances	FreeRatio	Class
35234568	10648	-	java.util.HashMap
5678900	10668	-	[Ljava.util.HashMap\$Entry;
4456788	7436	-	java.util.HashMap\$Entry
4321000	200	-	java.util.WeakHashMap
1234568	190	-	[Ljava.util.WeakHashMap\$Entry;
454400	4	-	java.util.WeakHashMap\$Entry
51380224	29146		total

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。
2	一定時間内に Explicit ヒープ詳細情報を含んだ拡張スレッドダンプ出力処理終了の応答がありませんでした。

## 出力メッセージ

次のエラーメッセージまたは警告メッセージが出力された場合、Explicit ヒープ詳細情報付き拡張スレッドダンプは出力されません。

項番	エラーメッセージ	説明
1	usage: eheapprof [-f -i] [-force] [-freeratio] -p process-id	eheapprof コマンドへの引数の指定が間違っています。
2	eheapprof: illegal option -- オプション	eheapprof コマンドに指定したオプションが不正です。
3	プロセスID: Now processing previous request, this request canceled	eheapprof コマンドの引数に指定したプロセス ID に該当するプロセスが、現在 Explicit ヒープ詳細情報を出力しています。
4	プロセスID: No such process	eheapprof コマンドの引数に指定したプロセス ID に該当するプロセスがありません。または、eheapprof コマンドで指定したプロセス ID に該当するプロセスが java プロセス以外でした。
5	プロセスID: Not owner	eheapprof コマンドの引数に指定したプロセス ID のプロセスのオーナーではありません。
6	eheapprof: can't create work file at /tmp , this request canceled	一時ファイル用ディレクトリーに参照・書き込み権限がない場合、Explicit ヒープ詳細情報付き拡張スレッドダンプを出力できません。Explicit ヒープ詳細情報付き拡張スレッドダンプの出力要求はキャンセルされます。
7	eheapprof: please delete 削除できなかったファイル名 in 削除できなかったファイルの絶対パス	eheapprof コマンドを終了したときに、内部ファイルを削除できませんでした。削除できなかったファイルの絶対パスにある、削除できなかったファイルを削除してください。
8	eheapprof: unexpected error occurred: エラー原因	eheapprof コマンド実行中に予期しないエラーが発生しました。エラー原因には、例えば下記のような表示がされます。 <ul style="list-style-type: none"><li>作業用メモリー確保に失敗した場合 malloc systemcall fail (errno=Y)</li><li>オブジェクトのクローズに失敗した場合 close systemcall fail (errno=Y)</li></ul>
9	プロセスID: Timeout occurred. Java process not responding	eheapprof コマンドの引数に指定したプロセス ID に該当するプロセスから、一定時間内に Explicit ヒープ詳細情報出力処理終了の応答がありませんでした。

## 注意事項

- eheapprof コマンドはプログラムの開発用ユーティリティとして提供されているものです。システムの運用では使用しないでください。
- 同じ java プロセスに対して、同時に eheapprof コマンドは実行できません。前回の eheapprof コマンドによる Explicit ヒープ詳細情報が拡張スレッドダンプに出力されたあとに実行してください。
- eheapprof コマンド実行時に、/tmp/hsperfdata\_ユーザー名/プロセスID ファイルを使用します。該当するファイルが存在しない場合、eheapprof コマンドによる Explicit ヒープ詳細情報付き拡張スレッド



ダンプは出力できません。ただし、`-force` オプションを指定した場合、`/tmp/hsperfdata_ユーザー名/プロセスID` ファイルの有無の確認をしないで、`-p` オプションで指定したプロセス ID の java プロセスに対して拡張スレッドダンプの出力を要求します。

## 4.2.4 java\_hras

Application Server と連携しない Java アプリケーションプログラムの実行

### 書式

```
java_hras [options] class [argument...]  
java_hras [options] -jar file.jar [argument...]
```

### 格納先

Application Serverインストールディレクトリー/jdk/jre/bin/

### 機能

このコマンドは、単体 Java プログラム実行機能を有効にした状態で Java VM を起動し、Java アプリケーションを実行します。単体 Java プログラム実行機能が有効な場合は、トラブルシュートに必要な独自機能が有効となります。

java\_hras コマンド実行時のオプション指定として、トラブルシュートに必要な独自機能のON/OFF をオプションで指定した場合は、その指定が優先されます。同じオプションを複数指定した場合は、最後に指定したオプションの値が有効になります。

単体 Java プログラム実行機能有効時に切り替わるオプションを次に示します。

項番	カテゴリ	オプション	Java VM デフォルト値	単体 Java プログラム実行機能有効時
1	Java VM ログ出力	HitachiVerboseGC	OFF	ON
2		HitachiOutputMilliTime	OFF	ON
3		HitachiOutOfMemoryStackTrace	OFF	ON
4		HitachiJavaClassLibTrace	OFF	ON
5	スレッドダンプ出力	HitachiLocalsInStackTrace	OFF	ON
6		HitachiLocalsSimpleFormat	OFF	ON
7		HitachiOutOfMemoryAbort	OFF	ON
8	メモリーダンプ	HitachiFullCore	OFF	ON

単体 Java プログラム実行機能が有効になると、単体 Java プログラム実行機能未使用時の java コマンド実行時と比べ、次に示す違いがあります。

- スタックトレース  
単体 Java プログラム実行機能使用時、Java プログラムのメインスレッドのスタックトレース情報は、単体 Java プログラム実行機能未使用時の Java プログラム実行時と異なります。
- java\_hras コマンドは-Xhras オプションを指定できません。

これら以外の機能は、単体 Java プログラム実行機能未使用時の java コマンドと同じです。

## 引数

指定できるオプションなどは、java コマンドと同じです。ただし、-Xhras オプションだけは指定できません。

### *options*

コマンド行オプションを指定します。

### *class*

呼び出されるクラスの名称を指定します。

### *-jar file.jar*

呼び出される JAR ファイルの名称を指定します。

### *argument*

main 関数に渡される引数を指定します。

## 出力例

### 出力例 1

スレッドダンプ出力によってメインスレッドのスタックトレース情報を取得した場合。

- 単体 Java プログラム実行機能：使用時

```
main #1 prio=5 os_prio=0 jid=<N/A> tid=0x00000000230a800 nid=0x1988 waiting on
condition [0x00000000282e000..0x00000000282fbc0]
  java.lang.Thread.State: TIMED_WAITING (sleeping)

stack=[0x000000002830000..0x000000002734000..0x000000002731000..0x000000002730000]
 [user cpu time=218ms, kernel cpu time=46ms] [blocked count=0, waited count=0]
  at java.lang.Thread.sleep(Native Method)
  at Hoge.main(Hoge.java:5)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
43)
  at java.lang.reflect.Method.invoke(Method.java:483)
  at JP.co.Hitachi.soft.jvm.tools.Java_hras.execJava(Java_hras.java:52)
  at JP.co.Hitachi.soft.jvm.tools.Java_hras.main(Java_hras.java:39)
```

- 単体 Java プログラム実行機能：未使用時

```
"main" #1 prio=5 os_prio=0 jid=<N/A> tid=0x00000000002b7800 nid=0x1fa0 waiting on
condition [0x000000000278f000..0x000000000278f9c0]
  java.lang.Thread.State: TIMED_WAITING (sleeping)

stack=[0x0000000002790000..0x0000000002694000..0x0000000002691000..0x0000000002690000]
 [user cpu time=171ms, kernel cpu time=46ms] [blocked count=0, waited count=0]
  at java.lang.Thread.sleep(Native Method)
  at Hoge.main(Hoge.java:5)
```

## 出力例 2

起動時に指定したメインクラスのmainメソッド実行中に、mainメソッド内でキャッチされない java.lang.Exception 例外が発生した場合。

- 単体 Java プログラム実行機能：使用時

```
Exception in thread "main" java.lang.reflect.InvocationTargetException
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at JP.co.Hitachi.soft.jvm.tools.Java_hras.execJava(Java_hras.java:51)
  at JP.co.Hitachi.soft.jvm.tools.Java_hras.main(Java_hras.java:38)
Caused by: java.lang.Exception: java.lang.Throwable
  at Test.main(Test.java:6)
  ... 6 more
Caused by: java.lang.Throwable
  ... 7 more
```

- 単体 Java プログラム実行機能：未使用時

```
Exception in thread "main" java.lang.Exception: java.lang.Throwable
  at Test.main(Test.java:6)
Caused by: java.lang.Throwable
  ... 1 more
```

## 戻り値

java コマンドと同じです。

## 注意事項

java コマンドを実行するには、次に示す方法があります。

- *Application Server* インストールディレクトリー/jdk/jre/bin/java\_hras コマンドを使用する。
- -Xhras オプションを指定した java コマンドを使用する。
- java コマンド実行時、環境変数 JAVA\_HRAS を設定する。

ほかの方法で実行した場合、コマンドの実行に失敗し、次のメッセージが出力されます。

Java command cannot be used except in some special cases. When starting a Java program, use the java\_hras command, etc.

Depending on the Java program executed, it might be necessary to purchase the other product.

javaw コマンドを実行するには、次に示す方法があります。

- -Xhras オプションを指定したjavaw コマンドを使用する。
- javaw コマンド実行時、環境変数JAVAW\_HRAS を設定する。

ほかの方法で実行した場合、コマンドの実行に失敗し、次のメッセージが出力されます。

Javaw command cannot be used except in some special cases. When starting a Java program, use the java\_hras command, etc.

Depending on the Java program executed, it might be necessary to purchase the other product.

## 4.2.5 javacore

core ファイルとスレッドダンプの取得

### 書式

```
javacore [-i|-f] [-force] -p プロセスID
```

### 格納先

*Application Server*インストールディレクトリー/jdk/jre/bin/

### 機能

コマンド実行時の core ファイルとスレッドダンプを同時に取得します。

### 実行権限

引数のプロセス ID で指定する java プロセスと同一のユーザーで実行してください。

### 引数

-i

core ファイルおよびスレッドダンプの出力処理の実行を確認するメッセージが表示されます。表示されたメッセージに対してyまたはnを入力します。このとき、yを入力するとスレッドダンプが出力さ

れます。nを入力すると、何も出力しないで処理を終了します。省略した場合、-f オプションが指定されないかぎり、このオプションは有効です。

-f

-i オプションを無効にします。省略した場合、-i オプションが有効になります。

-force

java プロセスが作成する/tmp/hsperfdata\_ユーザー名/プロセスID ファイルの有無の確認をしないで、-p オプションで指定したプロセス ID の java プロセスに対して core を出力させます。

-p プロセスID

プロセスID には、core ファイルとスレッドダンプを取得する java プログラムのプロセス ID を指定します。

型：Integer

指定できる値を次に示します。

- 0~4294967295

## 出力形式

ヘッダー

```
EEE MMM dd hh:mm:ss yyyy※
```

```
Full thread dump Java HotSpot(TM) Client VM (1.5.0_05-b05-CDK0850-ビルド日  
mixed mode)  
起動コマンドライン  
...
```

注※

EEE は曜日、MMM は月、dd は日を表します。また、hh は時間、mm は分、ss は秒、yyyy は年（西暦）を表します。

システム設定

```
System Properties
```

```
-----  
Java Home Dir   : JDK実行環境インストールディレクトリー  
Java DLL Dir    : JDKを構成するライブラリーのインストールディレクトリー  
Sys Classpath   : システムクラスパス  
User Args       :  
Javaコマンドオプション1  
Javaコマンドオプション2  
...
```

動作環境

```
Operating Environment
```

```
-----  
Host      : ホスト名:IPアドレス  
OS        : OSバージョン  
CPU       : CPU種別, 利用可能CPU数/システム全体のCPU数
```

```

Resource Limits -
RLIMIT_CPU      :プロセスで使用可能な秒数
RLIMIT_FSIZE   :最大ファイルサイズ(単位:バイト)
RLIMIT_DATA    :malloc可能なサイズ(単位:バイト)
RLIMIT_STACK   :スタックの最大サイズ(単位:バイト)
RLIMIT_CORE    :coreの最大サイズ(単位:バイト)
RLIMIT_RSS     :プロセスの常駐サイズ(単位:バイト)
RLIMIT_AS      :プロセストータルの利用可能メモリー(単位:バイト)
RLIMIT_NOFILE  :最大のファイルディスクリプタ値

```

## Java ヒープ情報

### SerialGC の場合

```

Heap Status
-----
def new generation  max max size, total capacity, used size (max usage% used/max, total
usage% used/total)
                    [bottom, commit addr, reserve addr)
  eden space capacity,  usage% used [bottom, top, reserve addr)
  from space capacity,  usage% used [bottom, top, reserve addr)
  to   space capacity,  usage% used [bottom, top, reserve addr)
tenured generation  max max size, total capacity, used size (max usage% used/max, total
usage% used/total)
                    [bottom, commit addr, reserve addr)
  the space capacity,  usage% used [bottom, top, used block, reserve addr)
Metaspace           max max size, capacity capacity words, committed committed size,
reserved reserve size, used size (max usage% used/max, total usage% used/committed)
  class space       max max size, capacity capacity words, committed committed size,
reserved reserve size, used size (max usage% used/max, total usage% used/committed)
                    [bottom, top, commit addr, reserve addr)

```

#### *max size*

最大の容量。単位はキロバイト。

Metaspace の場合、-XX:MaxMetaspaceSize が未指定のとき最大値は無制限のため、"unlimited"が出力されます。

#### *capacity*

現在の容量。単位はキロバイト。

#### *capacity words*

Metaspace のコミット済みのメモリー領域からフリー領域を除いた合計サイズ。単位はキロバイト。

#### *committed size*

Metaspace のコミット済みの合計メモリーサイズ。単位はキロバイト。

#### *reserve size*

予約済みのメモリーサイズ。単位はキロバイト。

#### *size*

使用中のメモリーサイズ。単位はキロバイト。

#### *max usage*

最大の容量に対する使用率。

Metaspace の場合、`-XX:MaxMetaspaceSize` が未指定のとき最大値は無制限のため、`"-%"`が出力されます。

*total usage*

現在の容量に対する使用率。

*bottom*

領域先頭アドレス。

*top*

使用中領域の先頭アドレス。

*commit addr*

コミット済み領域の末尾アドレス。

*reserve addr*

予約済み領域の末尾アドレス。

*usage*

使用率。

*used block*

次の空きブロックの先頭アドレス。

注

class space は、64bit 版の Java VM で圧縮オブジェクトポインター機能が有効であり、かつ Compressed Class Space を使用している場合にだけ出力されます。

Metaspace の値は、class space を含んだ Metaspace 全体の合計値です。

G1GC の場合

```
Heap Status
-----
garbage-first heap  total heap capacity, used heap size [heap bottom, heap commit
addr, heap reserve addr)
  region size region size, young region young (young size), survivor region survivors
(survivor size)
Metaspace          max max size, capacity capacity words, committed committed size,
reserved reserve size, used size (max usage% used/max, total usage% used/committed)
  class space      max max size, capacity capacity words, committed committed size,
reserved reserve size, used size (max usage% used/max, total usage% used/committed)
                    [bottom, top, commit addr, reserve addr)
```

出力項目の詳細フォーマット

出力項目	出力内容	意味
<i>heap capacity</i>	<const>K	Java ヒープ領域の領域サイズ。単位はキロバイト。 (a)の合計サイズ+(b)の合計サイズ+(c)の合計サイズ+(d)の合計サイズ
<i>heap size</i>	<const>K	Java ヒープ領域の使用サイズ。単位はキロバイト。 (a)の合計サイズ+(b)の合計サイズ+(c)の合計サイズ

出力項目	出力内容	意味
<i>heap bottom</i>	<ptr64>	Java ヒープ領域の先頭アドレス。
<i>heap commit addr</i>	<ptr64>	Java ヒープ領域のコミット済みの末尾アドレス。
<i>heap reserve addr</i>	<ptr64>	Java ヒープ領域の予約済みの末尾アドレス。
<i>region size</i>	<const>K	1 リージョンのサイズ。単位はキロバイト。
<i>young region</i>	<const>	New 領域の使用中のリージョン数。 (a)のリージョン数+(b)のリージョン数
<i>young size</i>	<const>K	New 領域の使用サイズ。単位はキロバイト。 ((a)のリージョン数+(b)のリージョン数)*region_size
<i>survivor region</i>	<const>	Survivor 領域の使用中のリージョン数。 (b)のリージョン数
<i>survivor size</i>	<const>K	Survivor 領域の使用サイズ。単位はキロバイト。 (b)のリージョン数*region_size

Metaspace、class space の出力項目は Serial GC の場合と同じです。「Serial GC の場合」の説明を参照してください。

(凡例)

- (a) : Eden 領域
- (b) : Survivor 領域
- (c) : Tenured 領域
- (d) : Free 領域

#### Java VM 内部メモリーマップ情報

##### JVM Internal Memory Map

メモリー確保関数:address = 開始アドレス - 終了アドレス (size:サイズ)

注

- メモリー確保関数: mmap()か malloc()のどちらかが出力されます。
- 開始アドレス: メモリー領域の開始アドレスが 16 進で出力されます。
- 終了アドレス: メモリー領域の終了アドレスが 16 進で出力されます。
- サイズ: 確保しているメモリー領域のサイズが出力されます (単位: バイト)。

#### Java VM 内部メモリーサイズ情報

##### JVM Internal Memory Status

Heap Size :確保しているメモリーサイズ※  
 Alloc Size :使用中のメモリーサイズ※  
 Free Size :未使用のメモリーサイズ※



注※

単位：バイト

アプリケーション環境

```
Application Environment
-----
Signal Handlers -※1
  SIGHUP          :シグナルハンドラー情報
  SIGINT          :シグナルハンドラー情報
  ...
  SIGSOUND       :シグナルハンドラー情報
  SIGSAK         :シグナルハンドラー情報

Signal Handlers -
シグナル種別： [シグナルハンドラーアドレス], sa_mask[0]=シグナルマスク, sa_flags=特殊フラグ
...

Environment Variables -※2
環境変数=値
...

Current Directory -
Application Serverインストールディレクトリー/CC/server/...
```

注※1

次の情報が表示されます。

- ・シグナルハンドラーがインストールされている場合は、そのアドレス。
- ・SIG\_DFL である場合は、default。
- ・SIG\_IGN である場合は、ignored。

注※2

次の情報が表示されます。

- ・シグナル種別には、/usr/include/sys/signal.h に定義されているシグナル名。
- ・シグナルハンドラーアドレスには、シグナルハンドラーのアドレスが 16 進数で出力されます。ライブラリー名+オフセットという形式で表示されることもあります。
- ・シグナルマスクには、sigaction() で取り出せる構造の sa\_mask フィールド値が 16 進数で出力されます。
- ・特殊フラグには、sigaction() で取り出せる構造の sa\_flags フィールド値が 16 進数で出力されます。

ライブラリー情報 (Linux の場合)

```
Loaded Libraries
-----
Dynamic libraries :
開始アドレス-終了アドレス コマンド
開始アドレス-終了アドレス ライブラリー
...
```

## ライブラリー情報 (AIX の場合)

```
Loaded Libraries
-----
コマンド
  text      :開始アドレス-終了アドレス (サイズ)
  data      :開始アドレス-終了アドレス (サイズ)
ライブラリー
  text      :開始アドレス-終了アドレス (サイズ)
  data      :開始アドレス-終了アドレス (サイズ)
...
```

## スレッド情報

```
-----
"スレッド名" daemon prio=優先度 jid=ハッシュ値 tid=スレッドID
nid=nativeID status [開始アドレス...終了アドレス]
java.lang.Thread.State: スレッドの現在のステータス※
stack=[スタック開始アドレス..YellowPageアドレス..RedPageアドレス..
スタック終了アドレス]
[user cpu time=ユーザー時間ms, kernel cpu time=カーネル時間ms]
[blocked count=ブロック回数, waited count=待機回数]
  at クラス名.メソッド名(メソッド情報)
...
```

### 注※

スレッドの現在のステータスの情報は、JDK 6 がベースの場合で、バージョンが 08-10 以降のときにだけ出力されます。

出力内容を説明します。

### スレッド名

Thread クラスのコンストラクターに指定されたスレッド名称が出力されます。

*daemon* : デーモンスレッドである場合に、"daemon"と出力されます。

*優先度* : Thread#setPriority で設定された優先度が出力されます。

*ハッシュ値* : System.identityHashCode() を呼び出して得られる値と同一の値が 8 桁の 16 進数で出力されます。

*スレッド ID* : スレッドオブジェクトのメモリー上のアドレス。

*nativeID* : OS レベルのスレッド ID。

*status* : スレッドの状態。

*runnable* : 実行中または実行可能なスレッド

*in Object.wait()、waiting for monitor entry または waiting on condition* : モニターロック待ちのスレッド

*sleeping* : 中断状態のスレッド

*開始アドレス* : Java フレームの最高位スタックアドレスが 16 進数で出力されます。

*終了アドレス* : JavaLock のある最高位スタックアドレスが 16 進数で出力されます。

*ユーザー時間* : スレッド開始からのユーザー時間がミリ秒単位で出力されます。

*カーネル時間* : スレッド開始からのカーネル時間がミリ秒単位で出力されます。

ブロック回数：スレッド開始から、処理がブロックされた回数が出力されます。

待機回数：スレッド開始から、処理が待ち状態になった回数が出力されます。

#### スレッドの現在のステータス

スレッドの現在のステータスを表すメッセージが出力されます。メッセージの内容は `java.lang.Thread.State` 列挙型に対応します。

#### スタック開始アドレス

スタック開始アドレスが 16 進数で出力されます。

#### YellowPage アドレス

スタック Yellow ガードページ先頭アドレスが 16 進数で出力されます。

#### RedPage アドレス

スタック Red ガードページ先頭アドレスが 16 進数で出力されます。

#### スタック終了アドレス

スタック終了アドレスが 16 進数で出力されます。

#### クラス名

クラス名が出力されます。

#### メソッド名

メソッド名が出力されます。

#### メソッド情報

次のメソッド情報が出力されます。

##### Native Method

ネイティブメソッドの場合に出力されます。

ファイル名：行番号

Java メソッドで行番号付きでコンパイルされている場合に出力されます。

##### Unknown Source

Java メソッドで行番号なしでコンパイルされている場合に出力されます。

### Java モニターダンプ

```
Java monitor
-----
ロックオブジェクト@ハッシュコード オーナー情報
  待機状態:待機スレッド数
  待機スレッド情報
```

出力内容を説明します。

#### ロックオブジェクト

ロック対象オブジェクトのクラス名が出力されます。

#### ハッシュコード

`Object.hashCode` で得られるハッシュコードが出力されます。

## オーナー情報

owner "スレッド名"スレッドID：オーナーがある場合に出力されます。

no owner：オーナーがない場合に出力されます。

## 待機状態

... waiting to enter：メソッド実行待ちの場合に出力されます。

... waiting to be notified：通知待ちの場合に出力されます。

## 待機スレッド数

スレッド数が出力されます。

## 待機スレッド情報

["スレッド名" スレッドID] の形式で出力されます。

## JNI グローバル参照数の情報

### JNI Information

JNI global references: JNIグローバル参照数

出力内容を説明します。

## JNI グローバル参照数

Java VM が保持しているグローバル参照の数が出力されます。

## 注

JNI グローバル参照は Java VM の内部でも再利用されるため、JNI がサポートする DeleteGlobalRef 関数を発行して JNI グローバル参照を削除しても、数値は減少しません。また、NewGlobalRef 関数を発行して JNI グローバル参照を新規作成しても、Java VM が再利用した JNI グローバル参照を割り当てた場合は数値は増加しません。

## Explicit ヒープ情報および Explicit メモリーブロック情報

### Explicit Heap Status

max *EH\_MAX*, total *EH\_TOTAL*, used *EH\_USED*, garbage *EH\_GARB* (*EH\_PER1* used/max, *EH\_PER2* used/total, *EH\_PER3* garbage/used), *EM\_NUMS* spaces exist

Explicit Memories(*EM\_MGR\_PTR*)

"*EM\_NAME*" eid=*EID(EM\_PTR)/EM\_TYPE*, total *EM\_TOTAL*, used *EM\_USED*, garbage *EM\_GARB* (*EM\_PER1* used/total, *EM\_PER2* garbage/used, *FL\_BLOCKS* blocks) *EM\_STAT*

...

注意事項を次に示します。

- Explicit ヒープ情報と Explicit メモリーブロック情報の間には、空行が 1 行あります。
- Explicit メモリーブロック情報の出力順序（どの Explicit メモリーブロックから出力するか）は未定義とします。
- "*EM\_NAME*"の前には、半角 2 文字分のスペースがあります。

- 最後の行には空行が 1 行あります。したがって、Explicit メモリーブロックごとの出力の間には空行が 1 行あります。

フッター

```
Full thread dump completed.   EEE MMM dd hh:mm:ss yyyy※
```

注※

EEE は曜日、MMM は月、dd は日を表します。また、hh は時間、mm は分、ss は秒、yyyy は年（西暦）を表します。

拡張スレッドダンプ情報との比較を、次に示します。

出力情報	標準スレッドダンプ	拡張スレッドダンプ
ヘッダー	×	○
システム設定	×	○
動作環境	×	○
メモリー情報（Windows の場合だけ）	×	○
Java ヒープ情報	×	○
Java VM 内部メモリーマップ情報	×	○
Java VM 内部メモリーサイズ情報	×	○
アプリケーション環境	×	○
ライブラリー情報	×	○
スレッド情報	○	○※1
Java モニターダンプ	×	○
フッター	×	○
スレッドダンプ出力先	標準出力	標準出力※2 Java VM ログファイル

（凡例）

- ：出力されます。
- ×

注※1

スタックの開始および終了のアドレス情報などが出力されます。

注※2

-XX:+HitachiThreadDumpToStdout オプションが指定された場合に出力されます。

## 入力例

1. `-f` オプションを省略して `javacore` コマンドを実行します。

```
javacore -p 8326
```

2. `core` ファイルおよびスレッドダンプの出力処理の実行を確認するメッセージが表示されます。

```
send SIGQUIT to 8326:?(y/n)
```

3. `core` ファイルおよびスレッドダンプを取得する場合は `y` を、取得しない場合は `n` を入力します。

```
send SIGQUIT to 8326:?(y/n)y
```

4. `core` ファイルおよびスレッドダンプを取得すると、実行中の `java` プログラムでは次のメッセージが出力されます。

```
Now generating core file (javacore8662.030806215140.core)...
done
(スレッドダンプを出力)
Writing Java core to javacore8662.030806215140.txt... OK
```

5. 実行中の `java` プログラムは、カレントディレクトリーに次のファイルを作成し、プログラムを続けます。

- `core` ファイル  
javacoreプロセスID.日時.core
- スレッドダンプ  
javacoreプロセスID.日時.txt

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。
2	一定時間内に <code>core</code> 生成処理終了の応答がありませんでした。

## 出力メッセージ

次のエラーメッセージまたは警告メッセージを出力した場合、`core` ファイルやスレッドダンプは取得されません。

項番	エラーメッセージ	説明
1	usage: javacore [-f -i] [-force] -p process-id	コマンドの引数が不正です。
2	javacore: can't create work file at /tmp, this request canceled	/tmp に参照および書き込み権限がありません。
3	javacore: illegal option --オプション	コマンドの引数に指定したオプションが不正です。
4	javacore: unexpected error occurred:エラー原因	コマンド実行中に予期しないエラーが発生しました。

項番	エラーメッセージ	説明
5	javacore: please delete 削除できなかったファイル名 in 削除できなかったファイルの絶対パス	コマンド終了時にjavacore コマンドの内部処理で作成したファイルを削除できませんでした。削除できなかったファイルの絶対パスにある、削除できなかったファイルを削除してください。
6	プロセスID: No such process	javacore コマンドの引数に指定したプロセスID に該当するプロセスがありません。または、javacore コマンドで指定したプロセスID に該当するプロセスがjava プロセス以外でした。
7	プロセスID: Not owner	実行ユーザーは、コマンドの引数に指定したプロセスID に該当するプロセスのオーナーではありません。
8	プロセスID: Now processing previous request, this request canceled	コマンドの引数に指定したプロセスID のプロセスが現在 core を生成中です。
9	プロセスID: Timeout occurred. Java process not responding.	コマンドの引数に指定したプロセスID に該当するプロセスから、一定時間内に core 出力処理終了の応答がありませんでした。

## 注意事項

- javacore コマンドは、SIGQUIT シグナルを指定されたプロセスに送信します。誤って java プログラム以外を指定すると、ほかのプログラムが停止することがあります。
- 同じ Java プロセスに対して、同時にjavacore コマンドは実行できません。前回のjavacore コマンドによる core 出力処理が終了したあとに実行してください。
- Java VM プロセスが、スレッドダンプ取得要求に反応しないで無応答状態になっている場合には、javacore コマンドによる core ファイルも取得できません。この場合は、kill -6 コマンドの実行によって、Java VM プロセスを強制停止させて core ファイルを取得してください。
- AIX (5L V5.1) の場合、coredump システムコールで core を生成するため、次の環境変数を設定する必要があります。環境変数が設定されていない場合、core ファイルが正常に取得できません。

csh の場合

```
setenv LDR_CNTRL USERREGS
```

sh または ksh の場合

```
export LDR_CNTRL=USERREGS
```

LDR\_CNTRL 環境変数に複数指定する場合、@が区切り文字になります。例えば、MAXDATA と USERREGS の 2 つを指定する場合は、次のようになります。

csh の場合

```
setenv LDR_CNTRL MAXDATA=0x40000000@USERREGS
```

sh または ksh の場合

```
export LDR_CNTRL=MAXDATA=0x40000000@USERREGS
```

- Linux の場合、gdb のgcore コマンドで core を生成します。gdb がインストールされていない場合には、次のエラーメッセージが標準出力に出力されます。

```
Error occurred in generating core file, gdb not found.
```

また、インストールされている gdb のバージョンが古い場合には、次のエラーメッセージが標準出力に出力されます。

Error occurred in generating core file, gdb version 5.2 or later needed.

- javacore コマンドは、実行時に、/tmp/hsperfdata\_ユーザー名/プロセスID ファイルを使用します。該当するファイルが存在しない場合は、javacore コマンドによる core ファイルの出力はできません。ただし、-force オプションを指定した場合、/tmp/hsperfdata\_ユーザー名/プロセスID ファイルの有無の確認をしないで、-p オプションで指定したプロセス ID の Java プロセスに対して core を出力させます。

## 4.2.6 javagc

GC の強制発生

### 書式

```
javagc [-i|-f] [-v] [-s] [-force] [-ehgc] -p プロセスID
```

### 格納先

*Application Server*インストールディレクトリー/jdk/jre/bin/

### 機能

メモリーリーク、システム障害およびアプリケーションのデバッグのために、プロセス ID がプロセス ID の java プロセスに対して、任意のタイミングで FullGC を発生させます。また、プロセス ID がプロセス ID の java プロセスに対して、任意のタイミングで FullGC と、Explicit メモリーブロックの解放を実行します。

java プロセスとの通信には SIGQUIT シグナルを使用します。コマンドを実行すると、コマンドの処理内容をユーザーに確認します。n (発生させない、実行しない、または送信しない) と回答した場合には、コマンドの処理は実行されません (戻り値が 1 となります)。この確認動作は、-f オプションを指定することで省略できます。

プロセス ID の java プロセスが通常の要因で発生する CopyGC や FullGC を実行中の場合は、その終了を待ってからコマンドを実行します。

### 実行権限

引数のプロセス ID で指定する java プロセスと同一のユーザーで実行してください。

### 引数

-i

次に示すコマンドの処理内容をユーザーに確認します。



- プロセス ID のプロセスに対して、Explicit メモリーブロックの解放を実行するかどうか
  - GC を発生させるためにSIGQUIT シグナルを送信させるかどうか
- このオプションより前に指定した、-f オプションを無効にします。

-f

次に示すコマンドの処理内容をユーザーに確認しません。

- Explicit メモリーブロックの解放を実行するかどうか
  - GC を発生させるためにSIGQUIT シグナルを送信させるかどうか
- このオプションより前に指定した、-i オプションを無効にします。

-v

-XX:+HitachiVerboseGC オプションの指定がなくても、Java VM ログファイル作成の規則に従って Java VM ログファイルを作成して、拡張 verbosegc 情報を出力します。

その際、次のオプション値も反映した内容の拡張 verbosegc 情報を出力します。

- -XX:+HitachiVerboseGCPrintDate
- -XX:+HitachiVerboseGCPrintCause
- -XX:+HitachiVerboseGCCpuTime
- -XX:+HitachiCommaVerboseGC

-s

標準出力に拡張 verbosegc 情報を出力します。

その際、次のオプション値も反映した内容の拡張 verbosegc 情報を出力します。

- -XX:+HitachiVerboseGCPrintDate
- -XX:+HitachiVerboseGCPrintCause
- -XX:+HitachiVerboseGCCpuTime
- -XX:+HitachiCommaVerboseGC

-force

java プロセスが作成する/tmp/hsperfdata\_ユーザー名/プロセスID ファイルの有無の確認をしないで、-p オプションで指定したプロセス ID の java プロセスに対して GC や Explicit メモリーブロックの解放を実行します。

-ehgc

プロセス ID がプロセス ID の java プロセスに対して、FullGC と、Explicit メモリーブロックの解放を実行します。

自動解放機能が有効な場合の解放処理で解放されなかった Explicit メモリーブロックを、明示的に解放できます。

まず、Java VM によって FullGC が実行され、拡張 verbosegc 情報に GC の要因として「EMJavaGC Command」が出力されます。そのあと、次に示す Explicit メモリーブロックが解放されます。

- 明示管理ヒープ機能の自動解放機能が有効な場合に、明示解放予約によって予約された Explicit メモリーブロック
- 明示管理ヒープ自動配置設定ファイル、または Java VM で生成された Explicit メモリーブロック
- 前回の解放処理で解放されなかった Explicit メモリーブロック

また、次の場合は、解放処理が実行されません。

- 最大数の制限を超えて Explicit メモリーブロックを解放しようとした場合  
現存する Explicit メモリーブロックの数が 1,048,575 個の場合が該当します。
- 明示管理ヒープ機能がオフになっている場合  
-XX:-HitachiUseExplicitMemory オプションが指定されている場合が該当します。

この場合、コンストラクターの実行は成功しますが、無効な Explicit メモリーブロック (ExplicitMemory インスタンス) として扱われます。

javagc コマンドによる Explicit メモリーブロックの解放処理では、FullGC が実行されます。このため、動作中のアプリケーションに対する処理には適していません。アプリケーションのアンデプロイ時や夜間など、アプリケーションが動作していない時間帯に実行することをお勧めします。

#### -p プロセスID

FullGC や、Explicit メモリーブロックの解放を実行したいプロセス ID を指定します。

型：Integer

指定できる値を次に示します。

- 0～4294967295

## 出力形式 (SerialGC の場合)

オプション"v"または"s"を指定した場合は Java VM ログを出力します。

```
[id] <date> (Skip Full:full_count, Copy:copy_count) [gc_kind gc_info, gc_time secs][Eden:
eden_info][Survivor: survivor_info][Tenured: tenured_info][Metaspace: Metaspace_info][class
space: class_space_info] [cause:cause_info] [User: user_cpu secs][Sys: system_cpu secs][IM:
jvm_alloc_size, mmap_total_size, malloc_total_size][TC: thread_count][DOE: doe_alloc_size,
called_count][CCI: cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

出力内容を次に説明します。

*id*

Java VM ログファイル識別子。

*date*

GC 開始日時。

-XX:-HitachiVerboseGCPrintDate オプションが指定された場合、出力されません。

*full\_count*

full GC 情報出力をスキップした回数。

-XX:HitachiVerboseGCIntervalTime オプションが指定された場合に出力されます。

#### *copy\_count*

copy GC 情報出力をスキップした回数。

-XX:HitachiVerboseGCIntervalTime オプションが指定された場合に出力されます。

#### *gc\_kind*

GC 種別。「FullGC」または「GC」が出力されます。

#### *gc\_info*

GC 情報。次の形式で出力されます。

GC 前の領域長 -> GC 後の領域長 (領域サイズ)

#### *gc\_time*

GC 経過時間。

#### *Eden*

エデンの種別。「DefNew::Eden」が出力されます。

#### *eden\_info*

Eden 情報。次の形式で出力されます。

GC 前の領域長 -> GC 後の領域長 (領域サイズ)

#### *Survivor*

Survivor の種別。「DefNew:: Survivor」が出力されます。

#### *survivor\_info*

Survivor 情報。次の形式で出力されます。

GC 前の領域長 -> GC 後の領域長 (領域サイズ)

#### *Tenured*

Tenured の種別。「Tenured」が出力されます。

#### *tenured\_info*

Tenured 情報。次の形式で出力されます。

GC 前の領域長 -> GC 後の領域長 (領域サイズ)

#### *Metaspace\_info*

Metaspace 領域のメモリー情報。次の形式で出力されます。単位はキロバイトです。

<const1>K(<const2>K, <const3>K)-><const4>K(<const5>K, <const6>K)]

- <const1>: GC 前の Metaspace 領域の使用サイズ
- <const2>: GC 前の Metaspace 領域の capacity サイズ
- <const3>: GC 前の Metaspace 領域の commit サイズ
- <const4>: GC 後の Metaspace 領域の使用サイズ
- <const5>: GC 後の Metaspace 領域の capacity サイズ

- <const6>: GC 後の Metaspace 領域の commit サイズ

#### *class\_space\_info*

class space 領域のメモリー情報。次の形式で出力されます。単位はキロバイトです。  
圧縮オブジェクトポインタ機能が無効の場合、出力されません。

<const1>K(<const2>K, <const3>K)-><const4>K(<const5>K, <const6>K)]

- <const1>: GC 前の class space 領域の使用サイズ
- <const2>: GC 前の class space 領域の capacity サイズ
- <const3>: GC 前の class space 領域の commit サイズ
- <const4>: GC 後の class space 領域の使用サイズ
- <const5>: GC 後の class space 領域の capacity サイズ
- <const6>: GC 後の class space 領域の commit サイズ

#### *cause\_info*

GC 要因内容。

-XX:-HitachiVerboseGCPrintCause オプションが指定された場合、出力されません。

#### *user\_cpu*

GC スレッドがユーザーモードで費やした CPU 時間。単位は秒です。

CPU 時間取得に失敗した場合、[User: unknown]のように、「unknown」と表示されます。

-XX:-HitachiVerboseGCCpuTime オプションが指定された場合、出力されません。

#### *system\_cpu*

GC スレッドがカーネルモードで費やした CPU 時間。単位は秒です。

CPU 時間取得に失敗した場合、[Sys: unknown]のように、「unknown」と表示されます。

-XX:-HitachiVerboseGCCpuTime オプションが指定された場合、出力されません。

#### *jvm\_alloc\_size*

Java VM 内部で管理している領域のうち、現在使用中の領域のサイズ (*mmap\_total\_size* と *malloc\_total\_size* の合計サイズのうち、現在使用中の領域のサイズ)。

-XX:-HitachiVerboseGCPrintJVMInternalMemory オプションが指定された場合、出力されません。

#### *mmap\_total\_size*

Java VM 内部で管理している領域のうち、mmap (Windows の場合は VirtualAlloc) で割り当てた C ヒープの総サイズ。

-XX:-HitachiVerboseGCPrintJVMInternalMemory オプションが指定された場合、出力されません。

#### *malloc\_total\_size*

Java VM 内部で管理している領域のうち、malloc で割り当てた C ヒープの総サイズ。

-XX:-HitachiVerboseGCPrintJVMInternalMemory オプションが指定された場合、出力されません。

#### *thread\_count*

Java スレッドの数。

-XX:-HitachiVerboseGCPrintThreadCount オプションが指定された場合、出力されません。

#### *doe\_alloc\_size*

java.io.File.deleteOnExit()を呼び出して確保した累積のヒープサイズ。

-XX:-HitachiVerboseGCPrintDeleteOnExit オプションが指定された場合、出力されません。

#### *called\_count*

java.io.File.deleteOnExit()の呼び出し回数。

-XX:-HitachiVerboseGCPrintDeleteOnExit オプションが指定された場合、出力されません。

#### *cc\_used\_size*

GC発生時のコードキャッシュ領域の使用サイズ。単位はキロバイトです。

-XX:-PrintCodeCacheInfo オプションが指定された場合、出力されません。

#### *cc\_max\_size*

コードキャッシュ領域の最大サイズ。単位はキロバイトです。

-XX:-PrintCodeCacheInfo オプションが指定された場合、出力されません。

#### *cc\_info*

保守情報。

-XX:-PrintCodeCacheInfo オプションが指定された場合、出力されません。

## 出力形式 (G1GC の場合)

オプション"v"または"s"を指定した場合は Java VM ログを出力します。

G1GC では、Java VM ログファイルに GC に関するログ (以降、VG1 ログ) と Concurrent Marking (以降、CM) に関するログ (以降、VCM ログ) を出力します。CM はアプリケーションと並行で処理するため、1 回の CM のログは複数行に分かれて出力されます。

VG1 ログ、VCM ログの出力フォーマットの詳細を次に示します。

#### VG1 ログ

```
[id]<date>[gc_kind gc_info, gc_time secs][Status:gc_status][G1GC::Eden: eden_info]
[G1GC::Survivor: survivor_info][G1GC::Tenured: tenured_info][G1GC::Humongous: humongous_info]
[G1GC::Free: free_info][Metaspace: Metaspace_info][class space: class_space_info]
[cause:cause_info][RegionSize: region_sizeK][Target: target_time secs][Predicted:
predicted_time secs][TargetTenured: target_sizeK][Reclaimable: reclaimable_info][User:
user_cpu secs][Sys: system_cpu secs][IM: jvm_alloc_sizeK, mmap_total_sizeK,
malloc_total_sizeK][TC: thread_count][DOE: doe_alloc_sizeK, called_count][CCI: cc_used_sizeK,
cc_max_sizeK, cc_infoK]
```

注 改行や、項目間のスペースはありません。

出力項目	出力内容	意味
<i>id</i>	VG1	Java VM ログファイル識別子。G1GC の GC ログでは「[VG1]」が出力されます。
<i>date</i>	<letters>	GC または CM が開始された日時。-XX:-HitachiVerboseGCPrintDate 指定時は出力されません。
<i>gc_kind</i>	Full GC  Mixed GC Young GC   Young GC(initial-mark) CM Remark CM Cleanup	GC または CM の種別。
<i>gc_info</i>	<const1>K/ <const2>K(<const3>K)- ><const4>K/ <const5>K(<const6>K)	Java ヒープ領域のメモリー情報。 <const1>:GC 前の Java ヒープ領域の使用サイズ。 <const2>:GC 前の Java ヒープ領域の使用サイズ (リージョン換算) ※1。 <const3>:GC 前の Java ヒープ領域の領域サイズ (リージョン換算)。 <const4>:GC 後の Java ヒープ領域の使用サイズ。 <const5>:GC 後の Java ヒープ領域の使用サイズ (リージョン換算)。 <const6>:GC 後の Java ヒープ領域の領域サイズ (リージョン換算)。 単位はキロバイト。
<i>gc_time</i>	<time>	GC によるアプリケーション停止時間。 単位は秒。
<i>gc_status</i>	-lto exhausted	GC の状態。 <ul style="list-style-type: none"> <li>To 領域あふれが発生した場合、「to exhausted」が出力されます。</li> <li>上記の状態以外の場合、「-」が出力されます。</li> </ul> 各 GC で出力される状態を以下に示します。 gc_kind が Young GC または Young GC(initial-mark) または Mixed GC の場合：「-」または「to exhausted」が出力されます。 gc_kind が上記以外の場合：「-」が出力されます。
<i>eden_info</i>	<const1>K(<const2>K)- ><const3>K(<const4>K)	Eden 領域のメモリー情報。 <const1>:GC 前の Eden 領域の使用サイズ (リージョン換算)。 <const2>:GC 前の Eden 領域が取り得る最大の領域サイズ (リージョン換算) ※2。 <const3>:GC 後の Eden 領域の使用サイズ (リージョン換算)。 <const4>:GC 後の Eden 領域が取り得る最大の領域サイズ (リージョン換算) ※2。 単位はキロバイト。
<i>survivor_info</i>	<const1>K-><const2>K	Survivor 領域のメモリー情報。 <const1>:GC 前の Survivor 領域の使用サイズ (リージョン換算)。 <const2>:GC 後の Survivor 領域の使用サイズ (リージョン換算)。 単位はキロバイト。
<i>tenured_info</i>	<const1>K-><const2>K	Tenured 領域のメモリー情報。 <const1>: GC 前の Tenured 領域の使用サイズ (リージョン換算)。 <const2>: GC 後の Tenured 領域の使用サイズ (リージョン換算)。

出力項目	出力内容	意味
		単位はキロバイト。
<i>humongous_info</i>	<const1>K-><const2>K	Humongous 領域のメモリー情報。 <const1>: GC 前の Humongous 領域の使用サイズ (リージョン換算)。 <const2>: GC 後の Humongous 領域の使用サイズ (リージョン換算)。 単位はキロバイト。
<i>free_info</i>	<const1>K-><const2>K	Free 領域のメモリー情報。 <const1>: GC 前の Free 領域の使用サイズ (リージョン換算)。 <const2>: GC 後の Free 領域の使用サイズ (リージョン換算)。 単位はキロバイト。
<i>Metaspace_info</i>	<const1>K(<const2>K, <const3>K)- ><const4>K(<const5>K, <const6>K)]	Metaspace 領域のメモリー情報。 <ul style="list-style-type: none"> <li>• &lt;const1&gt;: GC 前の Metaspace 領域の使用サイズ</li> <li>• &lt;const2&gt;: GC 前の Metaspace 領域の capacity サイズ</li> <li>• &lt;const3&gt;: GC 前の Metaspace 領域の commit サイズ</li> <li>• &lt;const4&gt;: GC 後の Metaspace 領域の使用サイズ</li> <li>• &lt;const5&gt;: GC 後の Metaspace 領域の capacity サイズ</li> <li>• &lt;const6&gt;: GC 後の Metaspace 領域の commit サイズ</li> </ul> 単位はキロバイト。
<i>class_space_info</i>	<const1>K(<const2>K, <const3>K)- ><const4>K(<const5>K, <const6>K)]	class space 領域のメモリー情報。 圧縮オブジェクトポインタ機能が無効の場合、出力されません。 <ul style="list-style-type: none"> <li>• &lt;const1&gt;: GC 前の class space 領域の使用サイズ</li> <li>• &lt;const2&gt;: GC 前の class space 領域の capacity サイズ</li> <li>• &lt;const3&gt;: GC 前の class space 領域の commit サイズ</li> <li>• &lt;const4&gt;: GC 後の class space 領域の使用サイズ</li> <li>• &lt;const5&gt;: GC 後の class space 領域の capacity サイズ</li> <li>• &lt;const6&gt;: GC 後の class space 領域の commit サイズ</li> </ul> 単位はキロバイト。
<i>cause_info</i>	<letters>	GC 要因内容。 -XX:-HitachiVerboseGCPrintCause 指定時は出力されません。
<i>region_size</i>	<const>	1 リージョンのサイズ。 単位はキロバイト。
<i>target_time</i>	<time>	GC によるアプリケーション停止時間の目標時間。 単位は秒。
<i>predicted_time</i>	<time>	Java VM が予測した GC によるアプリケーション停止時間。 単位は秒。 GC 種別が「Full GC」、「CM Remark」または「CM Cleanup」の場合、予測を行わないため、0 が出力されます。
<i>target_size</i>	<const>	Mixed GC で GC 対象となった Tenured 領域のサイズ。

出力項目	出力内容	意味
		単位はキロバイト。 GC 種別が「Mixed GC」以外の場合は 0 が出力されます。
<i>reclaimable_info</i>	<const1>K(<const2>.<digit> <digit>%)	予測回収サイズ情報。 <const1>:予測回収サイズ。 単位はキロバイト。 <const2>.<digit><digit>:予測回収率 (小数第 2 位まで) なお、予測回収サイズ情報は、CM 終了直後の Young GC または Mixed GC だけが出力されます。それ以外の場合、予測を行わないため、0 が出力されます。
<i>user_cpu</i>	<time>	全 GC スレッドがユーザーモードで費やした CPU 時間の合計。 単位は秒。 -XX:-HitachiVerboseGCCpuTime 指定時は出力されません。 CPU 時間取得に失敗した場合には「unknown」と表示します:[User: unknown]
<i>system_cpu</i>	<time>	全 GC スレッドがカーネルモードで費やした CPU 時間の合計。 単位は秒。 -XX:-HitachiVerboseGCCpuTime 指定時は出力されません。 CPU 時間取得に失敗した場合には「unknown」と表示します:[Sys: unknown]

注※1 リージョン換算は、領域のサイズを 1 リージョンのサイズで切り上げ、1 リージョンのサイズの倍数で表した値です。

注※2 Eden 領域が取り得る最大の領域サイズは、(New 領域の領域サイズ)-(Survivor 領域の使用サイズ)を指します。

#### VCM ログ

[*id*]<*date*> [*cm\_event*][User: *user\_cpu* secs][Sys: *sys\_cpu* secs]

\* 改行なし。\* 項目間のスペースなし。

出力項目	出力内容	意味
<i>id</i>	VCM	Java VM ログファイル識別子。G1GC の CM ログでは「[VCM]」が出力されます。
<i>date</i>	<letters>	CM が開始された日時。 -XX:-HitachiVerboseGCPrintDate 指定時は出力されません。
<i>cm_event</i>	Concurrent Root Region Scan Start Concurrent Root Region Scan End Concurrent Mark Start Concurrent Mark End Concurrent Mark Stop	CM の状態。 Concurrent Root Region Scan Start: Concurrent Root Region Scan が開始された。 Concurrent Root Region Scan End: Concurrent Root Region Scan が終了した。



出力項目	出力内容	意味
	Concurrent Cleanup Start  Concurrent Cleanup End	Concurrent Mark Start:Concurrent Mark が開始された。 Concurrent Mark End:Concurrent Mark が終了した。 Concurrent Mark Stop:Concurrent Mark を中止した。 Concurrent Cleanup Start : Concurrent Cleanup が開始された。 Concurrent Cleanup End : Concurrent Cleanup が終了した。
<i>usr_cpu</i>	<time>	全 CM スレッドがユーザーモードで費やした CPU 時間の合計。 単位は秒。 -XX:-HitachiVerboseGCCpuTime 指定時は出力されません。 CPU 時間取得に失敗した場合には「unknown」と表示します:[User: unknown] CM の状態が Start の場合、0 が出力されます。
<i>sys_cpu</i>	<time>	全 CM スレッドがカーネルモードで費やした CPU 時間の合計。 単位は秒。 -XX:-HitachiVerboseGCCpuTime 指定時は出力されます。 CPU 時間取得に失敗した場合には「unknown」と表示します:[Sys: unknown] CM の状態が Start の場合、0 が出力されます。

## 入力例

1. -i オプションを指定して javagc コマンドを実行します。  
javagc -i -v -p 8326
2. プロセス ID の確認メッセージが表示されます。  
send SIGQUIT to 8326:?(y/n)
3. SIGQUIT シグナルを送信させる場合は y を、送信させない場合は n を入力します。  
send SIGQUIT to 8326:?(y/n)y

## 出力例

### VG1 ログ

```
[VG1]<Thu Oct 02 10:38:56.193 2014>[Full GC 753K/2048K(8192K)->678K/1024K(8192K), 0.0097901
secs][Status:-][G1GC::Eden: 1024K(2048K)->0K(2048K)][G1GC::Survivor: 0K->0K][G1GC::Tenured:
1024K->1024K][G1GC::Humongous: 0K->0K][G1GC::Free: 6144K->7168K] [Metaspace: 3634K(4492K,
4492K)->3634K(4492K, 4492K)][class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][RegionSize: 1024K][Target: 0.2000000 secs][Predicted: 0.0000000 secs]
[TargetTenured: 0K][Reclaimable: 0K(0.00%)] [User: 0.0000000 secs][Sys: 0.0000000 secs][IM:
20459K, 21920K, 0K][TC: 35][DOE: 0K, 0][CCI: 1172K, 245760K, 2496K]
```

### VCM ログ

```
[VCM]<Wed Jul 24 11:45:20 2013>[Concurrent Root Region Scan Start][User: 0.0000000 secs][Sys: 0.0000000 secs] [VCM]<Wed Jul 24 11:45:20 2013>[Concurrent Root Region Scan End][User: 0.0126134 secs][Sys: 0.0146961 secs] [VCM]<Wed Jul 24 11:45:20 2013>[Concurrent Mark Start][User: 0.0000000 secs][Sys: 0.0000000 secs] [VCM]<Wed Jul 24 11:45:34 2013>[Concurrent Mark End][User: 0.0156250 secs][Sys: 0.2495800 secs]
```

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。
2	一定時間内に GC 処理終了の応答がありませんでした。

## 出力メッセージ

次のエラーメッセージまたは警告メッセージが出力された場合、Explicit ヒープ詳細情報付き拡張スレッドダンプは出力されません。

項番	エラーメッセージ	説明
1	javagc [-f -i][-v][-s] [-force] [-ehgc] -p process-id	javagc コマンドへの引数の指定が間違っています。
2	javagc: illegal option--オプション	javagc コマンドに指定したオプションが不正です。
3	プロセスID: No such process	javagc コマンドの引数に指定したプロセス ID に該当するプロセスがありません。または、javagc コマンドで指定したプロセス ID に該当するプロセスが java プロセス以外でした。
4	プロセスID: Not owner	実行ユーザーは、javagc コマンドの引数に指定したプロセス ID に該当するプロセスのオーナーではありません。
5	プロセスID: Now processing previous request, this request canceled	javagc コマンドの引数に指定したプロセス ID に該当するプロセスは、前回の javagc コマンドによる GC を実行中です。javagc コマンドによる GC 実行要求はキャンセルされます。
6	javagc: can't create work file at /tmp, this request canceled	/tmp に参照および書き込み権限がないため、GC 要求ファイルが作成できません。javagc コマンドによる GC 実行要求はキャンセルされます。
7	javagc: unexpected error occurred:エラー原因	javagc コマンド実行中に予期しないエラーが発生しました。エラー原因には、例えば下記のような表示がされます。 <ul style="list-style-type: none"> <li>作業用メモリー確保に失敗した場合 malloc syscall fail (errno=Y)</li> <li>オブジェクトのクローズに失敗した場合 close syscall fail (errno=Y)</li> </ul>
8	プロセスID: Timeout occurred. Java process not responding.	javagc コマンドの引数に指定したプロセス ID に該当するプロセスから、一定時間内に GC 処理終了の応答がありませんでした。

項番	エラーメッセージ	説明
9	javagc : please delete 削除できなかったファイル名 in 削除できなかったファイルの絶対パス	javagc コマンドを終了したときに、内部ファイルを削除できませんでした。削除できなかったファイルの絶対パスにある、削除できなかったファイルを削除してください。
10	プロセスID: Failed to retry GC. Java process is GC locked.	javagc コマンドの引数に指定したプロセス ID のプロセスが、GC 実行が抑止される状態のため、GC 処理を実行できませんでした。

## 注意事項

- 同じ java プロセスに対して、同時に javagc コマンドは実行できません。前回の javagc コマンドによる GC 処理が終了してから実行してください。前回の GC 処理が終了している場合は、Java VM ログファイルに出力される拡張 verbosegc 機能の GC の要因に "JavaGC Command" が出力されます。
- javagc コマンドは SIGQUIT シグナルを指定されたプロセス ID に送信します。誤って java プロセス以外のプロセス ID を指定すると、ほかのプログラムが停止することがあります。
- javagc コマンド実行時に、/tmp/hsperfdata\_ユーザー名/プロセスID ファイルを使用します。該当するファイルが存在しない場合は、javagc コマンドによる GC の要求はできません。ただし、-force オプションを指定した場合、/tmp/hsperfdata\_ユーザー名/プロセスID ファイルの有無の確認をしないで、-p オプションで指定したプロセス ID の java プロセスに対して GC を発生させます。

## 4.2.7 javatrace

トレース情報の収集

### 書式

```
javatrace coreファイル名 実行ファイル名 [出力ファイル名]
[-l ライブラリーファイル名 ...]
```

### 格納先

Application Serverインストールディレクトリー/jdk/jre/bin/

### 機能

core ファイルからスタックトレース情報を取得します。

このコマンドは、Java VM が異常終了して core ファイルを生成した場合、その異常終了の原因究明に必要な情報を取得するために実行するコマンドです。プロセスダウン時の詳細要因などが調査できます。

## 引数

### coreファイル名

core ファイル名を指定します。

### 実行ファイル名

core ファイルを生成した実行ファイル名を指定します。

### 出力ファイル名

出力ファイル名を指定します。

省略した場合は、カレントディレクトリーの"javatrace.log"に出力されます。

### -l ライブラリーファイル名

使用したライブラリーファイルを指定します。

共用ライブラリーが絶対パスで実行ファイルに取り込まれている場合は、自動的に読み込まれるため、指定する必要はありません。

## 入力例

Java VM が異常終了して core ファイルを生成した場合、次のメッセージが表示されます。このメッセージ内のjavatrace コマンドの文字列を実行します。

なお、このメッセージは、異常終了時に生成されるエラーレポートファイル (hs\_err\_pidプロセスID.log) にも出力されます。

```
      :
# You can get further information from javatrace.log file generated
# by using javatrace command.
# usage: javatrace core-file-name loadmodule-name [out-file-name]
# [-l(library-name)...]
# Please use javatrace command as follows and submit a bug report
# to Hitachi with javatrace.log file:
# [Application Serverインストールディレクトリー/jdk/jre/bin/javatrace core
# Application Serverインストールディレクトリー/jdk/jre/bin/java]
#
```

- core ファイル名がcore の場合

```
Application Serverインストールディレクトリー/jdk/jre/bin/javatrace core Application Server
インストールディレクトリー/jdk/jre/bin/java
```

OSによっては実際に出力される core ファイル名がcore.プロセスID になる場合があります。その場合は、実際に出力された core ファイル名をjavatrace の引数に指定してください。

- core ファイル名がcore.プロセスID の場合

```
Application Serverインストールディレクトリー/jdk/jre/bin/javatracecore.8326 Application
Serverインストールディレクトリー/jdk/jre/bin/java
```

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。

## 出力メッセージ

次のエラーメッセージまたは警告メッセージを出力した場合、スタックトレース情報は出力されません。

項番	エラーメッセージ	説明
1	usage : javatrace core-filename loadmodule-name [out-filename] [-llibrary-name...]	javatrace コマンドへの引数の指定が間違っています。
2	javatrace: Cannot open <i>file-name</i> :No such file or directory	出力ファイルで指定したファイル以外のファイル ( <i>file-name</i> ) が見つかりません。
3	javatrace: Cannot open <i>file-name</i> : Permission denied	ファイル ( <i>file-name</i> ) の読み込みが許可されていません。
4	javatrace : Cannot create <i>file-name</i> : Already exist	出力ファイルに指定したファイル ( <i>file-name</i> ) がすでに存在しています。
5	* unknown core type(XXXXXXX) ignored.	core ファイル名で指定したファイルは、core ファイルではありません。
6	* ERROR : <i>file-name</i> , unknown magic(0xXXXX)	実行ファイル名で指定したファイルは、実行ファイルではありません。
7	WARNING : core file may not match loadmodule (core file from ' <i>loadmodule-name</i> ')	実行ファイル名で指定した実行ファイルと、core ファイル名で指定した core ファイルから取り出した実行ファイル ( <i>loadmodule-name</i> ) が一致していません。
8	javatrace : Bad argument : <i>argument-name</i> is directory	指定したファイル ( <i>file-name</i> ) はディレクトリーです。
9	* ERROR : <i>library-name</i> open : No such file or directory	実行ファイルが取り込んだ共用ライブラリー ( <i>library-name</i> ) が見つかりません。共用ライブラリーが相対パスで取り込まれている場合には、-l オプションで共用ライブラリーを明示的に指定する必要があります。
10	javatrace : illegal option -- <i>x</i>	誤ったオプション文字 ( <i>x</i> ) が指定されています。
11	javatrace : Cannot create <i>file-name</i> : Permission denied	出力ファイルで指定したファイル ( <i>file-name</i> ) の出力先に書き込み権限がありません。

## 注意事項

- javatrace コマンドは core ファイルが生成されたマシンで実行してください。

- Java VM が異常終了した場合に標準出力とエラーレポートファイルに出力されるメッセージで、jvmon コマンドの第 2 引数が、次のように絶対パスのロードモジュール名になっていないことがあります。

```
      :
# You can get further information from jvmon.log file generated
# by using jvmon command.
# usage: jvmon core-file-name loadmodule-name [out-file-name]
# [-l(library-name)...]
# Please use jvmon command as follows and submit a bug report
# to Hitachi with jvmon.log file:
# [Application Serverインストールディレクトリー/jdk/jre/bin/jvmon core
# Application Serverインストールディレクトリー/jdk/jre/bin/java]
#
```

その場合は、jvmon コマンドの第 2 引数に指定するロードモジュール名を、絶対パスに変換してから実行してください。

## 4.2.8 jheapprof

クラス別統計情報付き拡張スレッドダンプの出力

### 書式

```
jheapprof [-i|-f] [-class クラス名] [-staticroot|-nostaticroot]
[-explicit|-noexplicit] [-fullgc|-copygc|-nogc]
[-garbage|-nogarbage] [-rootobjectinfo|-norootobjectinfo]
[-rootobjectinfost 値] [-force] -p プロセスID
```

### 格納先

Application Serverインストールディレクトリー/jdk/jre/bin/

### 機能

引数に指定したプロセス ID の java プロセスについて、クラス別統計情報を含んだ拡張スレッドダンプを出力します。

各クラスのインスタンスが持つメンバーの配下にあるすべてのインスタンスのサイズを、クラスごとに統計情報として拡張スレッドダンプに出力できます。クラス別統計情報を複数回出力すると、Java オブジェクトのサイズの変化などを確認できます。これらの情報は、メモリーリークの検出などに利用できます。

### 実行権限

引数のプロセス ID で指定する java プロセスと同一のユーザーで実行してください。

## 引数

**-i**

クラス別統計情報付き拡張スレッドダンプの出力処理の実行を確認するメッセージが表示されます。表示されたメッセージに対してyまたはnを入力します。このとき、yを入力すると、クラス別統計情報を含んだ拡張スレッドダンプが出力されます。nを入力すると、何も出力しないで処理を終了します。省略した場合、**-f** オプションが指定されないかぎり、このオプションは有効です。

**-f**

**-i** オプションを無効にします。省略した場合、**-i** オプションが有効になります。

**-class** クラス名

クラス名に指定したクラス（インスタンス）をメンバーに持つクラスの構造を、一覧にしてスレッドダンプ中に出力します。

**-staticroot**

static フィールドを基点とした参照関係情報出力機能を有効にし、static フィールドを基点とした参照関係情報を出力します。省略した場合、**-nostaticroot** オプションが有効になります。

このオプションの前提は、**-class** オプションです。**-class** オプションの指定がない場合、このオプションは無効になります。

なお、このオプションと**-nostaticroot** オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

**-nostaticroot**

static フィールドを基点とした参照関係情報出力機能を無効にします。

なお、このオプションと**-staticroot** オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

**-explicit**

インスタンス統計機能の統計対象に Explicit ヒープを含めます。省略した場合、**-noexplicit** オプションが指定されないかぎり、このオプションは有効です。なお、このオプションと**-noexplicit** オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

**-noexplicit**

インスタンス統計機能の統計対象に Explicit ヒープを含めません。省略した場合、**-explicit** オプションが有効になります。なお、このオプションと**-explicit** オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

**-fullgc**

統計する前に実行する GC に FullGC を設定します。省略した場合、**-copygc** オプションや**-nogc** オプションが指定されないかぎり、このオプションは有効です。なお、このオプションと**-copygc** オプションまたは**-nogc** オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-copygc`

統計する前に実行する GC に CopyGC を設定します。このオプションと `-nogc` オプションを省略した場合、`-fullgc` オプションが有効になります。

なお、このオプションと `-fullgc` オプションまたは `-nogc` オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-nogc`

統計する前に GC を実行しません。このオプションと `-copygc` オプションを省略した場合、`-fullgc` オプションが有効になります。なお、このオプションと `-fullgc` オプションまたは `-copygc` オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-garbage`

Tenured 領域内不要オブジェクト統計機能が有効になり、Tenured 領域内の不要なオブジェクトを統計対象としたクラス別統計情報を出力します。また、インスタンス統計機能と STATIC メンバー統計機能は無効になります。省略した場合、`-nogarbage` オプションが有効になります。統計前 GC 選択機能については、`-fullgc` オプション、`-copygc` オプションが無効になり、`-nogc` オプションは有効になります。このため、統計処理前に GC を実行しません。なお、このオプションと `-nogarbage` オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-nogarbage`

Tenured 領域内不要オブジェクト統計機能が無効になります。そのため、Tenured 領域内の不要なオブジェクトを統計対象としたクラス別統計情報は出力しません。省略した場合、`-garbage` オプションが指定されないかぎり、このオプションは有効です。なお、このオプションと `-garbage` オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-rootobjectinfo`

Tenured 増加要因の基点オブジェクトリスト出力機能が有効になり、Tenured 増加要因の基点オブジェクトリストを出力します。

このオプションは、`-garbage` オプションが有効であることが前提です。そのため、`-nogarbage` オプションを有効にすると、このオプションは無効になります。省略した場合、`-norootobjectinfo` オプションが指定されないかぎり、このオプションは有効です。なお、このオプションと `-norootobjectinfo` オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-norootobjectinfo`

Tenured 増加要因の基点オブジェクトリスト出力機能が無効になります。そのため、Tenured 増加要因の基点オブジェクトリストは出力しません。省略した場合、`-rootobjectinfo` オプションが有効になります。なお、このオプションと `-rootobjectinfo` オプションを同時に指定している場合、最後に指定しているオプションが有効になります。

#### `-rootobjectinfo` 値

Tenured 増加要因の基点オブジェクトリストの情報量を調節します。インスタンスサイズの合計が、指定した値以上のクラス情報だけが、Tenured 増加要因の基点オブジェクトリストに出力されます。

このオプションは、`-rootobjectinfo` オプションが有効であることが前提です。そのため、`-norootobjectinfo` オプションを有効にすると、このオプションは無効になります。値には0以上の自然



数を指定できます。自然数以外や、文字列を指定した場合は、引数の指定に誤りがあるという内容のエラーメッセージが出力されて終了します。

デフォルト値：0

#### -force

java プロセスが作成する/tmp/hsperfdata\_ユーザー名/プロセスID ファイルの有無の確認をしないで、-p オプションで指定したプロセス ID の java プロセスに対して拡張スレッドダンプの出力を要求します。

#### -p プロセスID

プロセス ID には、クラス別統計情報を出力する java プログラムのプロセス ID を指定します。

型：Integer

指定できる値を次に示します。

- 0~4294967295

## 出力形式

クラス別統計情報を拡張スレッドダンプに出力するには、jheapprof コマンドを利用します。クラス別統計情報を出力したい Java プロセスや、参照関係情報を出力したいクラスを指定して、jheapprof コマンドを実行します。

jheapprof コマンド実行時には、次の指定ができます。

- クラス別統計情報に Explicit ヒープの情報を出力するかどうかの指定
- クラス別統計情報を取得する前に GC を実行するかどうかの指定
- jheapprof コマンドの実行例を次に示します。

ここでは、プロセス ID が2463 の Java プロセスのクラス別統計情報を出力します。

1. -p オプションに、クラス別統計情報を出力したい Java プロセスのプロセス ID を指定して、jheapprof コマンドを実行します。

```
% jheapprof -p 2463
```

jheapprof コマンドで-f オプションを省略している場合、次の確認メッセージが表示されます。

プロセス ID を確認するメッセージが次の形式で表示されます。

```
send SIGQUIT to 2463: ? (y/n)
```

2. y を入力します。

クラス別統計情報付き拡張スレッドダンプが出力されます。実行中の java プログラムでは次のメッセージが出力されます。

```
Writing Java core to javacore2463.030806215140.txt... OK
```

実行中の java プログラムは、カレントディレクトリーにクラス別統計情報付き拡張スレッドダンプ (javacoreプロセスID.日時.txt) を作成し、プログラムを継続します。

- Explicit ヒープの情報をクラス別統計情報に出力する場合

次の条件を満たしている場合、Explicit ヒープの情報をクラス別統計情報に出力できます。

- Java VM 起動オプションに `-XX:+HitachiUseExplicitMemory` を指定している。
- アプリケーションの実装、または実行環境 (J2EE サーバ) の設定で Explicit ヒープが使用されている。

Explicit ヒープの情報をクラス別統計情報に出力する場合は、`jheapprof` コマンドに `-explicit` オプションを指定して実行します。

- GC の実行有無を指定する場合

クラス別統計情報を出力する前に、GC を実行するかどうかを選択できます。この機能を統計前の GC 選択機能といいます。クラス別統計情報を出力する前に、GC を実行するかどうかは、`jheapprof` コマンドに次のどれかのオプションを指定して実行します。

- `-fullgc`  
FullGC を実行してから、クラス別統計情報を出力します。
- `-copygc`  
CopyGC を実行してから、クラス別統計情報を出力します。
- `-nogc`  
GC を実行しないで、クラス別統計情報を出力します。

## 入力例

1. `-f` オプションを省略して `jheapprof` コマンドを実行します。

```
% jheapprof -p 2463
```

2. プロセス ID の確認メッセージが表示されます。

```
send SIGQUIT to 2463: ? (y/n)
```

3. SIGQUIT シグナルを送信させる場合は `y` を、送信させない場合は `n` を入力します。

```
send SIGQUIT to 2463: ? (y/n)y
```

4. クラス別統計情報付き拡張スレッドダンプを出力すると、実行中の java プログラムでは次のメッセージが出力されます。

```
Writing Java core to javacore2463.030806215140.txt... OK
```

5. 実行中の java プログラムは、カレントディレクトリーにクラス別統計情報付き拡張スレッドダンプ (`javacoreプロセスID.日時.txt`) を作成し、プログラムを継続します。

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。
2	一定時間内にクラス別統計情報出力処理終了の応答がありませんでした。

## 出力メッセージ

次のエラーメッセージまたは警告メッセージが出力された場合、クラス別統計情報付き拡張スレッドダンプは出力されません。

項番	エラーメッセージ	説明
1	usage: jheapprof [-f -i] [-class classname] [-staticroot -nostaticroot] [-explicit -noexplicit] [-fullgc -copygc -nogc] [-garbage -nogarbage] [-rootobjectinfo -norootobjectinfo] [-rootobjectinfo size] [-force] -p process-id	jheapprof コマンドへの引数の指定が間違っています。
2	jheapprof: illegal option -- オプション	jheapprof コマンドに指定したオプションが不正です。
3	プロセスID: Now processing previous request, this request canceled	jheapprof コマンドの引数に指定したプロセス ID に該当するプロセスが現在クラス別統計情報の出力中です。
4	プロセスID: No such process	jheapprof コマンドの引数に指定したプロセス ID に該当するプロセスがありません。または、jheapprof コマンドで指定したプロセス ID に該当するプロセスが java プロセス以外でした。
5	プロセスID: Not owner	jheapprof コマンドの引数に指定したプロセス ID のプロセスのオーナーではありません。
6	jheapprof: can't create work file at /tmp , this request canceled	一時ファイル用ディレクトリーに参照・書き込み権限がない場合、クラス別統計情報付き拡張スレッドダンプを出力できません。クラス別統計情報付き拡張スレッドダンプの出力要求はキャンセルされます。
7	jheapprof: please delete 削除できなかったファイル名 in 削除できなかったファイルの絶対パス	jheapprof コマンドを終了したときに、内部ファイルを削除できませんでした。削除できなかったファイルのフルパスにある、削除できなかったファイルを削除してください。
8	jheapprof: unexpected error occurred: エラー原因	jheapprof コマンド実行中に予期しないエラーが発生しました。エラー原因には、例えば下記のような表示がされます。 <ul style="list-style-type: none"><li>作業用メモリー確保に失敗した場合 malloc systemcall fail (errno=Y)</li><li>オブジェクトのクローズに失敗した場合 close systemcall fail (errno=Y)</li></ul>
9	プロセスID: Timeout occurred. Java process not responding	jheapprof コマンドの引数に指定したプロセス ID に該当するプロセスから、一定時間内にクラス別統計情報出力処理終了の応答がありませんでした。
10	プロセスID: Failed to retry GC. Java process is GC locked.	jheapprof コマンドの引数に指定したプロセス ID のプロセスが、GC 実行が抑止される状態のため、GC 処理を実行できませんでした。

## 注意事項

- jheapprof コマンドはプログラムの開発用ユーティリティとして提供しているものです。システムの運用では使用しないでください。
- 同じ java プロセスに対して、同時に jheapprof コマンドは実行できません。前回の jheapprof コマンドによるクラス別統計情報が拡張スレッドダンプに出力されたあとに実行してください。
- jheapprof コマンドは、SIGQUIT シグナルを指定されたプロセスに送信します。誤って java プログラム以外を指定すると、ほかのプログラムが停止することがあります。
- jheapprof コマンド実行時に、/tmp/hisperfdata\_ユーザー名/プロセスID ファイルを使用します。該当するファイルが存在しない場合、jheapprof コマンドによるクラス別統計情報出力はできません。
- Explicit ヒープをインスタンス統計情報の統計対象に含めるためには、-XX:+HitachiUseExplicitMemory を指定して、明示管理ヒープ機能を有効にしている java プロセスに対して、jheapprof コマンドで -explicit オプションを指定してください。なお、-XX:-HitachiUseExplicitMemory を指定して、明示管理ヒープ機能を無効にしている場合、-explicit を指定しても Explicit ヒープ内のインスタンスはインスタンス統計情報の統計対象になりません。

## 4.2.9 jheapprofanalyzer

クラス別統計情報解析ファイルの CSV 出力

### 書式

```
jheapprofanalyzer [-J オプション名] [ファイル名]
```

### 格納先

Application Serverインストールディレクトリー/jdk/jre/bin/

### 機能

クラス別統計情報の付いた複数の拡張スレッドダンプファイルを入力ファイルとして、クラスごとのインスタンスの合計サイズ、およびクラスごとのインスタンス数を時系列に出力します。出力するファイルは、CSV 形式です。

クラス別統計情報解析機能では、インスタンスの合計サイズが大きいインスタンスの情報を出力して、そのインスタンスのメモリー使用量だけを確認することもできます。インスタンスの合計サイズが大きいものだけを出力する場合には-DJP.co.Hitachi.soft.jvm.tools.jheapprofanalyzer.threshold にしきい値を指定し、jheapprofanalyzer コマンドに指定して実行します。

### ファイル

クラス別統計情報付き拡張スレッドダンプファイル(javacoreプロセスID.日時.txt)

## 引数

### -J オプション名

オプション名には、次のオプションを指定できます。また、次のオプション以外を指定した場合は、動作保証の対象外となります。

- `-Xms`  
メモリー割り当てプールの初期サイズをバイト数で指定します。
- `-Xmx`  
メモリー割り当てプールの最大サイズをバイト数で指定します。
- `-DJP.co.Hitachi.soft.jvm.tools.jheapprofanalyzer.threshold=num`  
*num*: インスタンス合計サイズのしきい値を設定します。範囲は $0 \sim 2^{63}-1$  (Long.MAX\_VALUE) です。  
インスタンス合計サイズが *num* 以上のクラスだけ出力します。  
デフォルト値: 1024

### ファイル名

クラス別統計情報付き拡張スレッドダンプファイルを指定できます。ファイル名称の規定は、特にありません。また、ファイルの指定は、順不同であり、数に制限はありません。

## 出力形式

ここでは、クラス別統計情報解析機能の入力ファイル、出力ファイル、および出力形式について説明します。

### 入力ファイル

クラス別統計情報解析機能で使用する入力ファイルは、クラス別統計情報を出力した拡張スレッドダンプファイルです。

### 出力ファイル

クラス別統計情報解析機能で出力するファイルは、クラスごとのインスタンスの合計サイズを出力したファイル、およびクラスごとのインスタンス数を出力したファイルの 2 種類です。出力ファイルはカレントディレクトリーに次のファイル名で作成されます。

出力ファイルの種類	出力ファイルのファイル名の例
インスタンス合計サイズファイル	JheapprofAnalyzer_size_nnn.csv
インスタンス数ファイル	JheapprofAnalyzer_num_nnn.csv

(凡例)

*nnn*: ファイルの分割番号が出力されます。分割番号の範囲は001~999です。

列が 201 列を超える場合は出力ファイルが分割されます。また、999 ファイルを超えた場合は、001 に戻りファイルは書き換えられます。

分割する列数は、201 列 (クラス名 1 列 + 値 200 列) を超えた場合とし、出力形式は分割したファイルも同じになります。

## 出力形式

クラス別統計情報解析機能で出力されるファイルの出力形式を次の表に示します。なお、インスタンス合計サイズ、およびインスタンス数が出力された CSV ファイルの出力形式は同じです。1 列目はクラス名、入力ファイル名は最大 200 個（列）です。

class name,	入力ファイル名,	入力ファイル名,	...	入力ファイル名
クラス名,	値-1-1,	値-1-2,	...	値-1-xxx
:	:	:	...	:
クラス名,	値-y-1,	値-y-2,	...	値-y-xxx

### (凡例)

入力ファイル名:処理対象に指定したクラス別統計情報

クラス名:入力ファイルに出力されていたクラス名

値:インスタンスの合計サイズ、またはインスタンス数

クラス名と値、および値と値の間はコンマで区切ります。また、行の最後は値（スペースも含む）で終了します。

クラス名の出力順はランダムです。値は入力ファイルの先頭行にある日付を基に、日付の古いものから横に並びます。同じ日付の入力ファイルがある場合はランダムに連続して横に並びます。

## 参考

クラス別統計情報解析機能を複数回実行すると、処理途中のクラスが消滅したり追加されたりする場合があります。また、該当するクラスがない場合の値には0が出力されます。クラス情報の例を使用して説明します。

1 回目のクラス別統計情報(A.txt)	2 回目のクラス別統計情報(B.txt)	3 回目のクラス別統計情報(C.txt)
ClassA 100 ClassB 100	ClassA 100 ClassB 30 ClassC 50 ClassD 0	ClassA 100  ClassC 50

上記のようなクラス情報の場合で、`-DJP.co.Hitachi.soft.jvm.tools.jheapprofalyzer.threshold` のしきい値を0にしたときの出力結果は次のようになります。

<pre>class name, A.txt, B.txt, C.txt ClassA, 100, 100, 100 ClassB, 100, 30, 0 ClassC, 0, 50, 50 ClassD, 0, 0, 0</pre>
---

インスタンス合計サイズの最大値は $0 \sim 2^{63}-1$ 、インスタンス数の最大値は $0 \sim 2^{31}-1$ です。1つの入力ファイルに同じクラス名がある場合は、インスタンスサイズの合計が加算されます。また、インスタンス数も加算されます。加算されたことによって、それぞれの最大値を超えた場合は、指定した最大値が出力されます。なお、1つのクラスについて、すべての入力ファイルで該当するクラスの情報がなく、またはしきい値未満の場合は、そのクラスの情報は出力されません。

## 入力例

```
jheapprofanalyzer -J-Xms1024m -J-Xmx1024m -J-  
DJP.co.Hitachi.soft.jvm.tools.jheapprofanalyzer.threshold=5000 javacore22356.080523161703.txt  
javacore22356.080523161711.txt
```

## 戻り値

戻り値	説明
0	正常終了しました。
1	異常終了しました。

## 出力メッセージ

次のエラーメッセージが出力された場合、クラス別統計情報解析ファイルは出力されません。また、次のエラーメッセージ以外が出力された場合は、デフォルトの例外処理となります。

項番	エラーメッセージ	説明	出力後の動作
1	usage: jheapprofanalyzer [options] file...where options include:-J- Xms<size> set initial Java heap size-J- Xmx<size> set maximum Java heap size-J- DJP.co.Hitachi.soft.jvm.tools.jheapprofa nalyzer.threshold=<num> set instance total size threshold	JheapprofAnalyzer クラスへの引数の指定が間違っ ています。	(a)
2	JheapprofAnalyzer: Illegal property valuenum. Default is assumed.	JP.co.Hitachi.soft.jvm.tools.jheapprofanalyzer.thr eshold の num に数字以外を指定しました。または num が範囲外となっています。	(b)
3	JheapprofAnalyzer: can't open input file ファイル名	ディレクトリーにファイル名がありません。または、 別の原因でファイルを開けません。	(c)
4	JheapprofAnalyzer: can't read input file ファイル名	ファイル名の読み込みに失敗しました。	(c)
5	JheapprofAnalyzer: Illegal input file formatファイル名	ファイル名は、クラス別統計情報付き拡張スレッドダ ンプファイルではありません。	(c)
6	JheapprofAnalyzer: can't open output fileファイル名	出力ファイルが開けません。エラーの原因として、次 の状態が考えられます。 <ul style="list-style-type: none"><li>出力ファイルがディレクトリーとなっています。</li><li>出力ファイルがありません。</li><li>別の原因で出力ファイルを開けません。</li></ul>	(a)
7	JheapprofAnalyzer: can't write output fileファイル名	ファイル名の書き込みに失敗しました。	(a)

(凡例)

(a):エラーとなり処理を終了します。

(b):デフォルトを仮定して処理を続行します。

(c):処理を続行し、指定したすべての入力ファイルのエラーチェックをします。

## 注意事項

クラス別統計情報解析機能では、日付を取得するときと、データを読み込むときにファイルを開きます。そのため、コマンド実行中に入力ファイルの更新および削除の操作をした場合の結果は保証されません。



# 5

## PRF で使用するコマンド

この章では、PRF（性能解析トレース）で使用するコマンドの書式、機能などについて説明します。

## 5.1 PRF で使用するコマンドの一覧

PRF（性能解析トレース）で使用するコマンドの一覧を次の表に示します。

### PRF デーモンの起動や停止で使用するコマンド

コマンド名称	要約	概要説明
<code>cprfgetpid</code>	PRF デーモンプロセス ID の取得	PRF デーモンのプロセス ID を取得します。
<code>cprfstart</code>	PRF デーモンの開始	PRF デーモン（ <code>cprfd</code> ）を起動します。
<code>cprfstop</code>	PRF デーモンの停止	PRF デーモン（ <code>cprfd</code> ）を停止します。

### トレース情報の出力で使用するコマンド

コマンド名称	要約	概要説明
<code>cprfed</code>	性能解析トレース情報の編集出力	性能解析トレース情報を編集出力します。
<code>cprfflush</code>	バッファ内容の強制ファイル出力	バッファ内のトレース情報を出力します。
<code>cprflevel</code>	PRF トレース取得レベルの表示と変更	PRF トレース取得レベルを表示または変更します。

#### 重要

- コマンドの引数には、各コマンドで指定できる値を指定してください。それ以外の値が指定された場合、動作は保証されません。
- コマンドが正常に終了しても、エラーが発生している場合があります。メッセージログやスタックトレースログを確認してください。

## 5.2 PRF デーモンの起動や停止で使用するコマンド

PRF デーモンの起動や停止で使用するコマンドの書式、機能などについて説明します。

### 5.2.1 cprfgetpid

PRF デーモンプロセス ID の取得

#### 書式

```
cprfgetpid [-PRFID PRF識別子] [-h]
```

#### 格納先

*Application Server*インストールディレクトリー/*common/PRF/bin/*

#### 機能

指定した PRF 識別子のプロセス ID を取得します。また、PRF デーモンの起動確認を表示します。

#### 実行権限

次の権限を持つユーザー

- PRF を取得するプロセスと同じユーザーまたはスーパーユーザー
- 同じ PRFSPOOL 環境変数を使用する環境内では、スーパーユーザーとほかのユーザーが混在して PRF コマンドを実行できません。常に同じユーザーで PRF コマンドを実行する必要があります。

#### 前提条件

- PRF デーモンと同じ PRFSPOOL 環境変数が設定されている必要があります。
- PRF デーモンが一度も起動していない、または正常終了したあとにこのコマンドを実行した場合は、エラーになります。

#### 環境変数

- PATH  
*Application Server*インストールディレクトリー/*common/PRF/bin*、および */bin* を追加してください。
- LD\_LIBRARY\_PATH(Linux)  
*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。
- LIBPATH(AIX)  
*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。

- PRFSP00L

PRF デーモンの実行環境ディレクトリーを設定します。

PRFSP00L 環境変数は、ドメイン管理サーバが JavaEE サーバ、クラスターを起動するとき、ドメイン管理サーバによって設定されます。PRFSP00L 環境変数は、JavaEE サーバを関連元とする PRF 関連がある場合、関連先の PRF 名を基に、次の形式で設定されます。

"JavaEEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/nodes/ノード名/PRF名"

- TZ(UNIX)

タイムゾーンを設定します。日本標準時の場合はJST-9 を指定します。

- PSALLOC(AIX)

メモリー確保時に必要なページングスペースをすぐに確保する設定にします。

early を設定してください。

- NODISCLAIM(AIX)

free()に対するコールの処理方法として、disclaim()の発行を抑止する設定にします。

true を設定してください。

- EXTSHM(AIX)

プロセス空間の共有メモリー領域数の制限をなくす設定にします。

ON を設定してください。

- LDR\_CNTRL(AIX)

カーネルの従来より大きいデータエリアを扱えるようにするための設定です。

MAXDATA=0x40000000 を指定してください。

## 引数

### -PRFID PRF識別子

PRF 識別子には、asadmin のcreate-prf サブコマンドで PRF を構築したときに付けた PRF サーバの名称を指定します。PRF 識別子は 1~31 文字の英数字、またはアンダースコア ( \_ ) で指定します。"TSC"や"tsc"、または"CTM"や"ctm"で始まる文字列は指定しないでください。

デフォルト値：PRF\_ID

### -h

コマンドの使用方法を表示します。

## 入力例・出力例

- 入力例

```
cprfgetpid
```

- 出力例

出力結果は、標準出力に出力されます。

```
1700
```

## 戻り値

戻り値	説明
0	正常終了しました。プロセスは起動していません。
1	プロセスは起動していません。
2	PRFSPOOL 環境変数が設定されていません。
上記以外	コマンド処理中にエラーが発生しました。出力されたメッセージに従って対策したあと、再度コマンドを実行してください。エラーメッセージは標準エラー出力、ログファイルに出力しません。

## 5.2.2 cprfstart

PRF デーモンの開始

### 書式

```
cprfstart [-h] [-PRFID PRF識別子] [-PrfTraceLevel  
PRFトレース取得レベル[, [PRFトレース取得レベル]...]]  
[-PrfTraceCount PRFトレースファイル数]  
[-PrfTraceFileSize PRFトレースファイルサイズ]  
[-PrfTraceBufferSize PRFトレースバッファサイズ]  
[-PrfRemakeBuffer] [-PrfNoBackUp [0|1]] [-PrfConsole {0|1}]  
[-PrfLogShiftTime PRFログシフト時刻]  
[-PrfLogFileSize PRFログサイズ] [-PrfLogFileCount PRFログ最大個数]
```

### 格納先

Application Serverインストールディレクトリー/common/PRF/bin/

### 機能

PRF デーモン (cprfd) を起動します。なお、PRF デーモンは、ほかのプロセスよりも先に起動してください。

PRF デーモンは開始後にカレントディレクトリーを移動します。移動先は、次のディレクトリーです。

PRFトレース出力ディレクトリー\$PRFSPOOL/utt/prf/PRF識別子/

### 実行権限

次の権限を持つユーザー

- PRF を取得するプロセスと同じユーザーまたはスーパーユーザー
- 同じ PRFSPOOL 環境変数を使用する環境内では、スーパーユーザーとほかのユーザーが混在して PRF コマンドを実行できません。常に同じユーザーで PRF コマンドを実行する必要があります。

## 前提条件

- PRFSP00L 環境変数を設定してください。設定していない場合、コマンドがエラーリターンします。
- OS インストール後のデフォルトの設定では、`/etc/hosts` ファイルに `127.0.0.1` の IP アドレスに対して自ホスト名が設定されます。PRF を使う場合には、ネットワークインターフェースに割り当てられた IP アドレスに対して自ホスト名が設定されている必要があります。そのため、`/etc/hosts` ファイルには、自ホスト名に対してネットワークインターフェースに割り当てられた適切な IP アドレスを設定してください。

## 環境変数

- PATH  
*Application Server* インストールディレクトリー `/common/PRF/bin`、および `/bin` を追加してください。
- LD\_LIBRARY\_PATH(Linux)  
*Application Server* インストールディレクトリー `/common/PRF/lib` を指定してください。
- LIBPATH(AIX)  
*Application Server* インストールディレクトリー `/common/PRF/lib` を指定してください。
- PRFSP00L  
PRF デーモンの実行環境ディレクトリーを設定します。  
PRFSP00L 環境変数は、ドメイン管理サーバが JavaEE サーバ、クラスターを起動するとき、ドメイン管理サーバによって設定されます。PRFSP00L 環境変数は、JavaEE サーバを関連元とする PRF 関連がある場合、関連先の PRF 名を基に、次の形式で設定されます。  
"JavaEEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/nodes/ノード名/PRF名"
- TZ(UNIX)  
タイムゾーンを設定します。日本標準時の場合は JST-9 を指定します。
- PSALLOC(AIX)  
メモリー確保時に必要なページングスペースをすぐに確保する設定にします。  
`early` を設定してください。
- NODISCLAIM(AIX)  
`free()` に対するコールの処理方法として、`disclaim()` の発行を抑止する設定にします。  
`true` を設定してください。
- EXTSHM(AIX)  
プロセス空間の共有メモリー領域数の制限をなくす設定にします。  
`ON` を設定してください。
- LDR\_CNTRL(AIX)  
カーネルの従来 of 区分化より大きいデータエリアを扱えるようにするための設定です。  
`MAXDATA=0x40000000` を指定してください。

## 引数

-h

コマンドの使用方法を表示します。

### -PRFID *PRF*識別子

PRF 識別子には、`asadmin` の `create-prf` サブコマンドで PRF を構築したときに付けた PRF サーバの名称を指定します。PRF 識別子は 1~31 文字の英数字、またはアンダースコア (`_`) で指定します。"TSC"や"tsc"、または"CTM"や"ctm"で始まる文字列は指定しないでください。

デフォルト値：PRF\_ID

### -PrfTraceLevel *PRF*トレース取得レベル [, [*PRF*トレース取得レベル]...]

-PrfTraceLevel を省略した場合は標準レベルで情報を出力します。

PRF トレース取得レベルを 4 バイトの 16 進数 (8 桁の値) で指定します。ここで 4 バイトの 16 進数の先頭に `0x` を付けることができますが、`0x` は無視されます。

各インデックス番号のレベル値をコンマ (,) 区切りで左から複数指定できます。レベル値の指定を省略する場合は、値を記述しないでください。また、あるインデックス番号以降の値をすべて指定しない場合は、コンマ (,) の記述を省略してください。インデックス番号 1 のトレース取得レベルを (1)、インデックス番号 2 の PRF トレース取得レベルを (2) とした場合、PRF トレース取得レベルは「(1) , (2) , ...」と指定します。

PRF トレース取得レベルの指定例を次に示します。

PRF トレース取得レベルの指定例	オプションの指定
インデックス番号 1 の PRF トレース取得レベルを指定する	-PrfTraceLevel 0x44445555
インデックス番号 1 とインデックス番号 2 の PRF トレース取得レベルを指定する	-PrfTraceLevel 0x44445555, 0x55554444
インデックス番号 2 の PRF トレース取得レベルを指定する	-PrfTraceLevel ,0x55554444

各機能レイヤーでの PRF トレース取得レベルの割り当てについて説明します。

PRF トレース取得レベルは、インデックスごとに 8 桁の 16 進数で指定します。それぞれの桁には、機能レイヤーが 2 つずつ割り当てられています。16 進数を 2 進数にした場合の上位 2 ビットに割り当てられている機能レイヤーを上位レイヤー、下位 2 ビットに割り当てられている機能レイヤーを下位レイヤーといいます。

桁ごとの上位レイヤー、下位レイヤーに割り当てられている機能レイヤーについて、次の表に示します。

インデックス	桁番号	上位レイヤー	下位レイヤー
インデックス 1	1 桁目	(なし)	Java EE Server
	2 桁目	(なし)	Java VM
	5 桁目	(なし)	uCosminexus TP1 Connector、TP1/Client/J

PRF トレース取得レベルを指定する場合は、桁ごとの上位レイヤー、下位レイヤーの PRF トレース取得レベルを「標準」「詳細」「保守」のどのレベルで取得するかを決め、それを 16 進数で指定します。

上位レイヤーと下位レイヤーの PRF トレース取得レベルの組み合わせと指定値の対応を、次の表に示します。

なお、保守レベルは、障害発生時などの保守情報を取得するためのレベルです。通常は指定しないでください。

表 5-1 詳細レベルまたは保守レベルのトレース情報を取得する場合の指定値 (cprfstart コマンド)

上位レイヤー	下位レイヤー	指定値
標準	標準	0
標準	詳細	1
標準	保守	2
詳細	標準	4
詳細	詳細	5
詳細	保守	6
保守	標準	8
保守	詳細	9
保守	保守	a

PRF トレース取得レベルの指定例を、次の表に示します。

インデックス	指定例	説明
インデックス 1	00000000	インデックス 1 の各機能レイヤーの標準レベルのトレース情報を取得します。
	10000000	Java EE Server だけ詳細レベルのトレース情報を取得して、ほかの機能レイヤーは標準レベルのトレース情報を取得します。
	11000000	Java EE Server および Java VM の詳細レベルのトレース情報を取得します。

#### -PrfTraceCount PRF トレースファイル数

PRF トレースファイルの面数を 3~256 の範囲で指定します。

正常停止中と前回強制停止後の再起動中に、PRF トレースファイルをバックアップします。PRF トレースファイルサイズの合計値が大きい場合、バックアップ処理で PRF デーモンの起動や停止に時間が掛かることがあります。そのため、ドメイン管理サーバでプロセス監視する場合、デフォルトの監視時間ではタイムアウトが発生する場合があります。タイムアウトが発生した場合には、監視時間を長くしてください。

デフォルト値：4

#### -PrfTraceFileSize PRF トレースファイルサイズ

PRF トレースファイルの 1 ファイル当たりの最大容量を 1024~1048576 (単位：キロバイト) の範囲で指定します。ファイル容量を 1 メガバイトとする場合、1024 と指定します。実際のファイルサイズは、この値より多少の増減があります。増減の中で増分の幅は、最大で「-PrfTraceBufferSize 指定値-32 キロバイト」です。



-PrfNoBackUp オプションに0を指定した場合、正常停止中と前回強制停止後の再起動中に、PRF トレースファイルをバックアップします。PRF トレースファイルサイズの合計値が大きい場合、バックアップ処理でPRF デーモンの起動や停止に時間がかかることがあります。そのため、-PrfNoBackUp オプションに0を指定してドメイン管理サーバでプロセス監視する場合、デフォルトの監視時間ではタイムアウトが発生する場合があります。タイムアウトが発生した場合には、監視時間を長くしてください。

デフォルト値：8192

デフォルト値の8192より小さい値を指定する場合は、-PrfTraceBufferSize もデフォルト値より小さい値を指定してください。

#### -PrfTraceBufferSize PRF トレースバッファサイズ

共用メモリーに確保するバッファサイズを512~102400（単位：キロバイト）の範囲で指定します。このオプションには、-PrfTraceFileSize で指定した値よりも大きい値は指定できません。

バッファ領域が不足するとKFCT26999-W メッセージが出力され、PRF トレースが欠落する場合があります。そのため、メッセージが出力されなくなるまでバッファ領域を拡張してください。

デフォルト値：8192

#### -PrfRemakeBuffer

共用メモリーを再作成して起動します。再作成できない場合にはエラーとなります。

#### -PrfNoBackUp {0|1}

PRF デーモンの起動時、および停止時に PRF トレースファイルをバックアップするかを指定します。値に0を指定した場合、PRF トレースファイルをバックアップします。PRF トレースファイルをコピーするため、PRF デーモンの起動、および停止に時間が掛かることがあります。障害発生後、PRF トレースがラップするまでの間に PRF トレースファイルの採取できない場合は、値に0を指定してください。なお、ドメイン管理サーバを使用している時は、障害発生後に PRF トレースファイルを含む snapshot ログを自動的に収集します。

値に1を指定した場合、PRF トレースファイルをバックアップしません。

PRF トレースファイルをバックアップするタイミング、バックアップの有無、および-PrfNoBackUp オプションの関係は次のようになります。

実行コマンド	前回開始した PRF デーモンの終了状態	PRF トレースファイルのバックアップ	
		-PrfNoBackUp 0	-PrfNoBackUp 1
cprfstart	正常終了	しない	しない
	強制終了または異常終了	する	しない
cprfstop	終了状態に依存しない	する	しない
cprfstop -Force	終了状態に依存しない	しない	しない

デフォルト値：1

#### -PrfConsole {0|1}

Windows のデスクトップからcprfstart コマンドを実行した場合、PRF デーモンの DOS プロンプトを表示するかどうかを指定します。このオプションは Windows 以外の OS では効果はありません。

指定できる値を次に示します。

- 1  
DOS プロンプトを表示します。
- 0  
DOS プロンプトを表示しません。

デフォルト値：0

#### -PrfLogShiftTime PRFログシフト時刻

PRF が出力するログファイルを時刻でシフトさせる場合のシフト時刻を指定します。ログのシフトは、ログファイルにログを出力するタイミングでシフトします。このため、ログの出力がない場合、ログはシフトされません。

指定できる値を次に示します。

- 000000～235959

デフォルト値：000000

#### -PrfLogFileSize PRFログサイズ

PRF が出力するログファイルサイズの上限をメガバイト単位で指定します。

指定できる値を次に示します。

- 1～100

デフォルト値：10

#### -PrfLogFileCount PRFログファイル最大個数

PRF が出力するログファイルの最大個数を指定します。

指定できる値を次に示します。

- 1～32

デフォルト値：8

## 戻り値

戻り値	説明
0	PRF デーモンが正常に起動しました。
1	PRF デーモンの起動に失敗しました。PRFSPPOOL 環境変数が設定されていません。
上記以外	コマンド処理中にエラーが発生しました。出力されたメッセージに従って対策したあと、再度コマンドを実行してください。エラーメッセージは標準エラー出力、ログファイルに出力します。

## 注意事項

- PRF デーモンを再起動する場合、`-PrfTraceCount` と `-PrfTraceFileSize` に前回と同じ値を指定して再起動してください。指定を省略して再起動した場合、PRF トレースファイル数と PRF トレースファイルサイズについては、前回の起動時の値は引き継がれないので、デフォルトの値になります。
- PRF デーモンが異常終了して再起動した場合、終了時のトレース取得レベルが引き継がれます。そのため、トレース取得レベルを変更したあとに異常終了した場合、再起動時に指定したトレース取得レベル（前回起動時と同じ値）と異なる値（変更後の値）が設定されます。次に運用例を示します。

1. PRF デーモン起動：`cprfstart -PrfTraceLevel 0x00000001`
2. トレース取得レベル変更（`0x00000001` → `0x0000000f`）
3. PRF デーモン異常終了
4. PRF デーモン再起動：`cprfstart -PrfTraceLevel 0x00000001`

この場合、再起動後のトレース取得レベルは、`0x0000000f` になります。

- PRF デーモンを再起動する場合、前回使用したバッファを再利用し、トレース取得レベルを引き継ぎます。また、バッファのサイズが変更になった場合は、前回のバッファを削除し、再作成します。ほかのプロセスがバッファにアクセスしている場合、この作成処理に失敗します。

`$PRFSPool/utt/prf/PRF識別子/spool/save/`

- PRF デーモンが異常終了した場合、PRF デーモンだけを再起動してください。
- PRF デーモンが異常終了してから再起動するまでの間に取得されたトレースは破棄されます。
- `-PrfTraceCount` または `-PrfTraceFileSize` に大きな値を指定すると、PRF デーモンが異常終了したあとの PRF デーモンの再起動に時間が掛かる場合があります。

## 5.2.3 cprfstop

PRF デーモンの停止

### 書式

```
cprfstop [-PRFID PRF識別子] [-Force] [-h]
```

### 格納先

*Application Server*インストールディレクトリー/`common/PRF/bin/`

### 機能

PRF デーモン (`cprfd`) を停止します。なお、PRF デーモンはほかのプロセスよりもあとに停止してください。

## 実行権限

次の権限を持つユーザー

- PRF を取得するプロセスと同じユーザーまたはスーパーユーザー
- 同じ PRFSPPOOL 環境変数を使用する環境内では、スーパーユーザーとほかのユーザーが混在して PRF コマンドを実行できません。常に同じユーザーで PRF コマンドを実行する必要があります。

## 前提条件

停止する PRF デーモンと同じ PRFSPPOOL 環境変数が設定されている必要があります。設定されていない場合、コマンドがエラーリターンします。

## 環境変数

- PATH  
*Application Server* インストールディレクトリー/*common/PRF/bin*、および */bin* を追加してください。
- LD\_LIBRARY\_PATH(Linux)  
*Application Server* インストールディレクトリー/*common/PRF/lib* を指定してください。
- LIBPATH(AIX)  
*Application Server* インストールディレクトリー/*common/PRF/lib* を指定してください。
- PRFSPPOOL  
PRF デーモンの実行環境ディレクトリーを設定します。  
PRFSPPOOL 環境変数は、ドメイン管理サーバが JavaEE サーバ、クラスターを起動するとき、ドメイン管理サーバによって設定されます。PRFSPPOOL 環境変数は、JavaEE サーバを関連元とする PRF 関連がある場合、関連先の PRF 名を基に、次の形式で設定されます。  
”*JavaEEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/nodes/ノード名/PRF名*”
- TZ(UNIX)  
タイムゾーンを設定します。日本標準時の場合は JST-9 を指定します。
- PSALLOC(AIX)  
メモリー確保時に必要なページングスペースをすぐに確保する設定にします。  
*early* を設定してください。
- NODISCLAIM(AIX)  
*free()* に対するコールの処理方法として、*disclaim()* の発行を抑止する設定にします。  
*true* を設定してください。
- EXTSHM(AIX)  
プロセス空間の共有メモリー領域数の制限をなくす設定にします。  
*ON* を設定してください。
- LDR\_CNTRL(AIX)

カーネルの従来の区分化より大きいデータエリアを扱えるようにするための設定です。  
MAXDATA=0x40000000 を指定してください。

## 引数

### -PRFID PRF識別子

PRF 識別子には、asadmin のcreate-prf サブコマンドで PRF を構築したときに付けた PRF サーバの名称を指定します。PRF 識別子は 1~31 文字の英数字、またはアンダースコア ( \_ ) で指定します。"TSC"や"tsc"、または"CTM"や"ctm"で始まる文字列は指定しないでください。

デフォルト値：PRF\_ID

### -Force

PRF デーモンを強制停止します。

### -h

コマンドの使用方法を表示します。

## 戻り値

戻り値	説明
0	PRF デーモンが正常に停止しました。
1	PRF デーモンが起動していません。PRFSPPOOL 環境変数が異なっている、または-PRFID の値が誤っているおそれがあります。
2	PRFSPPOOL 環境変数が設定されていません。
3	タイムアウトが発生しました。
上記以外	コマンド処理中にエラーが発生しました。出力されたメッセージに従って対策したあと、再度コマンドを実行してください。エラーメッセージは標準エラー出力、ログファイルに出力します。

## 注意事項

- コマンドがタイムアウトして、PRF デーモンが正常終了しなかった場合は、-Force オプションを指定して強制停止してください。
- cprfstart コマンドの-PrfTraceCount または-PrfTraceFileSize に大きな値を指定すると、PRF デーモンの正常停止に時間が掛かる場合があります。

## 5.3 トレース情報の出力で使用するコマンド

トレース情報の出力で使用するコマンドの書式、機能などについて説明します。

### 5.3.1 cprfed

性能解析トレース情報の編集出力

#### 書式

```
cprfed [-h][[-Dump|-CSV] [-Time 開始時刻,終了時刻]
        [-ProcessID プロセスID[,プロセスID...]]
        [-TraceFile PRFトレースファイル名[,PRFトレースファイル名...]]
        [-AllTraceFile] [-Compact] [-AscLen ASCII表示文字数]
```

#### 格納先

*Application Server*インストールディレクトリー/*common/PRF/bin/*

#### 機能

PRF トレース情報を入力して、性能解析トレース情報を編集出力します。

PRF デーモンが起動している場合は、先に `cprfflush` コマンドを実行してバッファ内のトレース情報を書き出してからこのコマンドを実行してください。

#### 実行権限

次の権限を持つユーザー

- PRF を取得するプロセスと同じユーザーまたはスーパーユーザー
- 同じ PRFSPPOOL 環境変数を使用する環境内では、スーパーユーザーとほかのユーザーが混在して PRF コマンドを実行できません。常に同じユーザーで PRF コマンドを実行する必要があります。

#### 環境変数

- PATH  
*Application Server*インストールディレクトリー/*common/PRF/bin*、および */bin* を追加してください。
- LD\_LIBRARY\_PATH(Linux)  
*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。
- LIBPATH(AIX)  
*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。
- PRFSPPOOL

PRF デーモンの実行環境ディレクトリーを設定します。

PRFSPPOOL 環境変数は、ドメイン管理サーバが JavaEE サーバ、クラスターを起動するとき、ドメイン管理サーバによって設定されます。PRFSPPOOL 環境変数は、JavaEE サーバを関連元とする PRF 関連がある場合、関連先の PRF 名を基に、次の形式で設定されます。

”JavaEEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/nodes/ノード名/PRF名”

- TZ (UNIX)

タイムゾーンを設定します。日本標準時の場合は JST-9 を指定します。

- PSALLOC (AIX)

メモリー確保時に必要なページングスペースをすぐに確保する設定にします。

early を設定してください。

- NODISCLAIM (AIX)

free() に対するコールの処理方法として、disclaim() の発行を抑止する設定にします。

true を設定してください。

- EXTSHM (AIX)

プロセス空間の共有メモリー領域数の制限をなくす設定にします。

ON を設定してください。

- LDR\_CNTRL (AIX)

カーネルの従来の区分化より大きいデータエリアを扱えるようにするための設定です。

MAXDATA=0x40000000 を指定してください。

## 引数

-h

コマンドの使用方法を表示します。

-Dump

詳細情報をダンプ形式で出力します。

-CSV

詳細情報を CSV 形式で出力します。

-Time 開始時刻, 終了時刻

指定された時刻内に取得されたトレース情報だけを出力します。

開始時刻、終了時刻は、hhmmss[MMdd[yyyy]]形式で指定します。MMdd、yyyy は省略できます。省略した場合は、コマンドを実行した月日年が指定されます。

時刻の文字列の形式：hhmmss[MMdd[yyyy]]

hh：時 (00~23)

mm：分 (00~59)

ss：秒 (00~59)

MM : 月 (01~12)

dd : 日 (01~31)

yyyy : 年 (1970~20xx)

2003年10月11日にコマンドを実行した場合の引数の指定例とその編集範囲の例を次に示します。

引数の指定	編集範囲
-Time 102345,	2003年10月11日10時23分45秒からトレースファイルの最後まで。
-Time ,102345	トレースファイルの最初から2003年10月11日10時23分45秒まで。
-Time 102345,102350	2003年10月11日10時23分45秒から2003年10月11日10時23分50秒まで。
-Time 1023451012,1023501013	2003年10月12日10時23分45秒から2003年10月13日10時23分50秒まで。
-Time 10234510122002,10235010132002	2002年10月12日10時23分45秒から2002年10月13日10時23分50秒まで。

#### -ProcessID プロセスID

指定されたプロセスIDの情報を編集出力します。プロセスIDは10進数で指定してください。

#### -TraceFile PRFトレースファイル名

PRFトレースファイルを指定します。トレースファイルは最大256個指定できます。このオプションは、-AllTraceFileオプションと同時に指定できません。

PRFトレースファイルは、PRFトレースファイル出力先ディレクトリーに自動的に作成され、ファイル名はprf\_???になります。???は、001からトレースファイルの面数を最大値とした整数です。

PRFトレースファイル出力先ディレクトリー

\$PRFSP00L/utt/prf/PRF識別子/dcopltrc/

また、PRFトレースファイル名を指定する場合は、絶対パスを指定するか、PRFトレースファイル出力先ディレクトリーに移動してから、ファイル名だけを指定します。

#### -AllTraceFile

カレントディレクトリー下のすべてのファイルを入力ファイルとして読み込みます。このオプションは、-TraceFileオプションと同時に指定できません。カレントディレクトリーに256を超える数のファイルがある場合は256ファイルだけ編集します。なお、cprfedコマンドの結果をカレントディレクトリーにリダイレクトする場合は、リダイレクト先のファイルも入力ファイルとして数えます。

#### -Compact

次の項目のトレース出力を長さ0の文字列にします。

Client AP IP/Client AP PID/Client AP CommNo./OPT/ASCII

-CSVオプションと組み合わせて使用する必要があります。-CSVオプションを指定しない場合、このオプションは無視されます。

#### -AscLen ASCII表示文字数

ダンプ情報のうちのASCII出力項目の最大出力文字数を指定します。省略した場合は長さ0の文字列にします。



ASCII 出力項目の内容は常にダブルクォーテーションで囲まれます。値 1 を指定した場合、ダブルクォーテーション分が加算され、3 文字出力されます。ただし、ASCII 出力項目に出力情報がない場合は長さ 0 の文字列になります。

-Compact オプションと組み合わせて使用する必要があります。-Compact オプションを指定しない場合、このオプションは無視され、ASCII 出力項目の内容はすべて表示されます。

指定できる値を次に示します。

- 0~512

デフォルト値：0

指定できるコマンドオプションの組み合わせの一覧を次に示します。

コマンドオプション	-h	-Dump	-CSV	-Time	-ProcessID	-TraceFile	-AllTraceFile	-Compact
-Dump	△-h	—	—	—	—	—	—	—
-CSV	△-h	△-CSV	—	—	—	—	—	—
-Time	△-h	○	○	—	—	—	—	—
-ProcessID	△-h	○	○	○	—	—	—	—
-TraceFile	△-h	○	○	○	○	—	—	—
-AllTraceFile	△-h	○	○	○	○	×	—	—
-Compact	△-h	△-Dump	○	○	○	○	○	—
-AscLen	△-h	△-Dump	○	○	○	○	○	○

(凡例)

- ：同時に指定できます。
- △：△のあとに続くオプション名のオプションが有効になります。
- ×
- ：該当しません。

## 戻り値

戻り値	説明
0	正常終了しました。
0 以外	コマンド処理中にエラーが発生しました。出力されたメッセージに従って対策したあと、再度コマンドを実行してください。エラーメッセージは標準エラー出力、ログファイルに出力します。

## 出力形式

出力形式 1 (ダンプ形式)

```

PRF: aaaaaa Process: bbbbbbbbbb Thread: ccccccccc(CC..CC) Trace: dddddddddd
ProcessName: vv..vv
Event: eeeee Time: ffff/gg/hh ii:jj:kk lll/mmm/nnn
Rc: oo..oo
ClientAP:pp..pp qq..qq - rr..rr RootAP: ss..ss tt..tt - uu..uu
INT: XX..XX OPR: YY..YY
Offset +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f 0123456789abcdef
AAAA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AAAAAAAAAAAAAAAAAA
AAAA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AAAAAAAAAAAAAAAAAA
:
AAAA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AAAAAAAAAAAAAAAAAA

```

## 出力形式 2 (CSV 形式)

```

PRF, Process, Thread(hashcode), Trace, ProcessName, Event, Date, Time, Time(msec/usec/
nsec), Rc, ClientAP IP, ClientAP PID, ClientAP CommNo., RootAP IP, RootAP PID, RootAP
CommNo., INT, OPR, OPT, ASCII
aaaaaa, bbbbbbbbbb, ccccccccc(CC..CC), dddddddddd, vv..vv, eeeee, ffff/gg/hh, ii:jj:kk, lll/mmm/
nnn, oo..oo, pp..pp, qq..qq, rr..rr, ss..ss, tt..tt, uu..uu, XX..XX, YY..YY, AA..AA, BB..BB

```

## (凡例)

記号	説明
aa...aa	レコード状態 (6 文字以内) <ul style="list-style-type: none"> <li>• Rec: レコードの状態は正常です。</li> <li>• ErrRec: レコードの状態は異常です。</li> </ul>
bb...bb	トレース情報を取得したプロセスのプロセス ID (10 進数で 10 桁以内)
cc...cc	トレース情報を取得したプロセス内スレッドのスレッド ID (16 進数で 18 桁以内)
dd...dd	該当スレッドでのトレース通番 (10 進数で 10 桁以内)
ee...ee	イベント ID (16 進数で 6 桁 (先頭の0x 含む))
ffff	トレースを取得した時刻 (年)
gg	トレースを取得した時刻 (月)
hh	トレースを取得した時刻 (日)
ii	トレースを取得した時刻 (時)
jj	トレースを取得した時刻 (分)
kk	トレースを取得した時刻 (秒)
lll	トレースを取得した時刻 (ミリ秒)
mmm	トレースを取得した時刻 (マイクロ秒)
nnn	トレースを取得した時刻 (ナノ秒)
oo...oo	戻り値 (16 進数で 10 桁 (先頭の0x 含む))
pp...pp	クライアントアプリケーションの IP アドレス (xxx.xxx.xxx.xxx の形式)

記号	説明
qq…qq	クライアントアプリケーションのプロセス ID (10 進数で 10 桁以内)
rr…rr	クライアントの通信番号 (18 桁 (先頭の0x 含む))
ss…ss	ルートアプリケーションの IP アドレス (xxx.xxx.xxx.xxx の形式)
tt…tt	ルートアプリケーションのプロセス ID (10 進数で 10 桁以内)
uu…uu	ルートの通信番号 (18 桁 (先頭の0x 含む))
vv…vv	プロセス名称 (32 文字以内)
XX…XX	インターフェース名 (33 文字以内) ※
YY…YY	オペレーション名 (33 文字以内) ※
AA…AA	各イベントで取得した情報をダンプ形式で出力します (最大 514 文字)。 ダンプ形式の情報を” ” で囲んで出力します。
BB…BB	各イベントで取得した情報を ASCII 文字で出力します (最大 514 文字)。 ダンプ形式の ASCII 文字は,” ” で囲んで出力します。 ASCII 文字で表示できないコード (制御コード) は「.」(ピリオド) で出力し、ASCII 文字で” (1 文字) は”” (2 文字) で出力します。
CC…CC	スレッドのハッシュ値 (16 進数で 18 桁以内)

#### 注※

インターフェース名、オペレーション名が 32 文字を超える場合は、次のように編集され 33 文字で出力されます。

「前 16 文字」 + 「\*」 + 「後ろ 16 文字」

「前 32 文字」 + 「\*」

「\*」 + 「後ろ 32 文字」

## 出力例

- `cprfed -TraceFile prf_001` を実行した場合

```
PRF: Rec Process: 26006      Thread: 0x15(0x12345678)      Trace: 1023
ProcessName: tscd
Event: 0x1002 Time: 2000/02/12 13:43:44 363/200/000
Rc: 0
ClientAP: 172.17.113.19 26303      - 000000000000003ff RootAP: 172.17.113.19
26006      - 000000000000003ff
INT : testint                      OPR: test5
```

- `cprfed -Dump -TraceFile prf_001` を実行した場合

```
PRF: Rec Process: 26006      Thread: 0x15(0x12345678)      Trace: 1023
ProcessName: tscd
Event: 0x1002 Time: 2000/02/12 13:43:44 363/200/000
Rc: 0
ClientAP: 172.17.113.19 26303      - 000000000000003ff RootAP: 172.17.113.19
26006      - 000000000000003ff
```

```

INT : testint                                OPR: test5

Offset  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f  0123456789abcdef
+0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
+0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
+0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
+0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

```

- `cprfed -CSV -TraceFile prf_001` を実行した場合

```

PRF, Process, Thread(hashcode), Trace, ProcessName, Event, Date, Time, Time(msec/usec/
nsec), Rc, ClientAP IP, ClientAP PID, ClientAP CommNo., RootAP IP, RootAP PID, RootAP
CommNo., INT, OPR, OPT, ASCII
Rec, 1800, 0x2e00(0x0e9576fd), 1, Client, 0x1009, 2003/10/12, 16:26:52, 734/000/000, ***,
10.209.15.65, 1800, 0x0000000000000001, 10.209.15.65, 1800, 0x0000000000000001, ABC, ***, abcdef
g, "31323334"

```

**注意事項**

- 性能解析トレースでは高負荷時に排他によるボトルネックが生じないように、PRF トレース出力時に排他制御をしていません。そのため、一部トレース情報が正しく出力されない場合があります。
- CSV 形式に編集したファイルに ASCII 文字以外を含む場合は、Unicode をサポートしたエディター、または Excel 2003 以降で参照してください。

**5.3.2 cprfflush**

バッファ内容の強制ファイル出力

**書式**

```

cprfflush [-PRFID PRF識別子] [-h]

```

**格納先**

*Application Server*インストールディレクトリー/common/PRF/bin/

**機能**

PRF デーモンに対して、バッファに格納されているトレース情報の出力を要求します。  
出力する範囲は、前回ファイル出力したトレースの次のトレースから、コマンドが入力された時点までです。

**実行権限**

次の権限を持つユーザー

- PRF を取得するプロセスと同じユーザーまたはスーパーユーザー

- 同じ PRFSPOOL 環境変数を使用する環境内では、スーパーユーザーとほかのユーザーが混在して PRF コマンドを実行できません。常に同じユーザーで PRF コマンドを実行する必要があります。

## 前提条件

- `cprfflush` コマンドの実行中は、`cprfflush` コマンドを実行できません。
- 要求先の PRF デーモンと同じ PRFSPOOL 環境変数が設定されている必要があります。設定されていない場合、コマンドがエラーリターンします。

## 環境変数

- PATH  
*Application Server* インストールディレクトリー/`common/PRF/bin`、および/`bin` を追加してください。
- LD\_LIBRARY\_PATH(Linux)  
*Application Server* インストールディレクトリー/`common/PRF/lib` を指定してください。
- LIBPATH(AIX)  
*Application Server* インストールディレクトリー/`common/PRF/lib` を指定してください。
- PRFSPOOL  
PRF デーモンの実行環境ディレクトリーを設定します。  
PRFSPOOL 環境変数は、ドメイン管理サーバが JavaEE サーバ、クラスターを起動するとき、ドメイン管理サーバによって設定されます。PRFSPOOL 環境変数は、JavaEE サーバを関連元とする PRF 関連がある場合、関連先の PRF 名を基に、次の形式で設定されます。  
”JavaEEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/`nodes/ノード名/PRF名`”
- TZ(UNIX)  
タイムゾーンを設定します。日本標準時の場合は JST-9 を指定します。
- PSALLOC(AIX)  
メモリー確保時に必要なページングスペースをすぐに確保する設定にします。  
`early` を設定してください。
- NODISCLAIM(AIX)  
`free()` に対するコールの処理方法として、`disclaim()` の発行を抑止する設定にします。  
`true` を設定してください。
- EXTSHM(AIX)  
プロセス空間の共有メモリー領域数の制限をなくす設定にします。  
`ON` を設定してください。
- LDR\_CNTRL(AIX)  
カーネルの従来の区分化より大きいデータエリアを扱えるようにするための設定です。  
`MAXDATA=0x40000000` を指定してください。

## 引数

### -PRFID *PRF*識別子

PRF 識別子には、`asadmin` の `create-prf` サブコマンドで PRF を構築したときに付けた PRF サーバの名称を指定します。PRF 識別子は 1~31 文字の英数字、またはアンダースコア ( `_` ) で指定します。"TSC"や"tsc"、または"CTM"や"ctm"で始まる文字列は指定しないでください。

デフォルト値：PRF\_ID

### -h

コマンドの使用方法を表示します。

## 戻り値

戻り値	説明
0	正常終了しました。
1	PRF デーモンが起動していません。PRFSPPOOL 環境変数が異なっている、または-PRFID の値が誤っているおそれがあります。
2	PRFSPPOOL 環境変数が設定されていません。
上記以外	コマンド処理中にエラーが発生しました。出力されたメッセージに従って対策したあと、再度コマンドを実行してください。エラーメッセージは標準エラー出力、ログファイルに出力します。

## 5.3.3 cprflevel

PRF トレース取得レベルの表示と変更

### 書式

```
cprflevel [-PRFID PRF識別子] [-h] [-PrfChangeLevel  
PRF トレース取得レベル[, [PRF トレース取得レベル]...]]  
[-PrfLevelIndex インデックス番号] [-PrfLevelAll]
```

### 格納先

*Application Server*インストールディレクトリー/`common/PRF/bin/`

### 機能

PRF トレース取得レベルを表示または変更します。

-PrfChangeLevel オプションを省略した場合は、指定した PRF 識別子に指定されている PRF トレース取得レベルを表示します。-PrfChangeLevel オプションを指定した場合は、指定した PRF 識別子に指定されている PRF トレース取得レベルを指定値に変更します。

-PrfLevelIndex オプションを指定した場合は、指定したインデックスの PRF トレース取得レベルの一覧が表示されます。-PrfLevelAll オプションを指定した場合は、すべてのインデックスに対応する PRF トレース取得レベルが表示されます。

## 実行権限

次の権限を持つユーザー

- PRF を取得するプロセスと同じユーザーまたはスーパーユーザー
- 同じ PRFSPPOOL 環境変数を使用する環境内では、スーパーユーザーとほかのユーザーが混在して PRF コマンドを実行できません。常に同じユーザーで PRF コマンドを実行する必要があります。

## 前提条件

- このコマンドは、PRF デーモンの起動中に実行してください。PRF デーモンが起動していないときにコマンドを実行した場合、エラーメッセージを出力します。
- 変更する PRF デーモンと同じ PRFSPPOOL 環境変数が設定されている必要があります。

## 環境変数

- PATH  
*Application Server*インストールディレクトリー/*common/PRF/bin*、および*/bin* を追加してください。
- LD\_LIBRARY\_PATH(Linux)  
*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。
- LIBPATH(AIX)  
*Application Server*インストールディレクトリー/*common/PRF/lib* を指定してください。
- PRFSPPOOL  
PRF デーモンの実行環境ディレクトリーを設定します。  
PRFSPPOOL 環境変数は、ドメイン管理サーバが JavaEE サーバ、クラスターを起動するとき、ドメイン管理サーバによって設定されます。PRFSPPOOL 環境変数は、JavaEE サーバを関連元とする PRF 関連がある場合、関連先の PRF 名を基に、次の形式で設定されます。  
"*JavaEEサーバ構築先ノードのJava EE Serverログ出力先ディレクトリー/nodes/ノード名/PRF名*"
- TZ(UNIX)  
タイムゾーンを設定します。日本標準時の場合は JST-9 を指定します。
- PSALLOC(AIX)  
メモリー確保時に必要なページングスペースをすぐに確保する設定にします。  
*early* を設定してください。
- NODISCLAIM(AIX)  
*free()* に対するコールの処理方法として、*disclaim()* の発行を抑制する設定にします。  
*true* を設定してください。

- EXTSHM(AIX)

プロセス空間の共有メモリー領域数の制限をなくす設定にします。

ON を設定してください。

- LDR\_CNTRL(AIX)

カーネルの従来の区分化より大きいデータエリアを扱えるようにするための設定です。

MAXDATA=0x40000000 を指定してください。

## 引数

### -PRFID PRF識別子

PRF 識別子には、asadmin のcreate-prf サブコマンドで PRF を構築したときに付けた PRF サーバの名称を指定します。PRF 識別子は 1~31 文字の英数字、またはアンダースコア ( \_ ) で指定します。"TSC"や"tsc"、または"CTM"や"ctm"で始まる文字列は指定しないでください。

デフォルト値：PRF\_ID

### -h

コマンドの使用方法を表示します。

### -PrfChangeLevel PRFトレース取得レベル [, [PRFトレース取得レベル]...]

PRF トレース取得レベルを 4 バイトの 16 進数 (8 桁の値) で指定します。ここで 4 バイトの 16 進数の先頭に 0x を付けることができますが、0x は無視されます。各インデックス番号に対応するレベル値をコンマ ( , ) 区切りで左から複数指定できます。該当するインデックスのレベル値を変更しない場合は、値を指定しないでください。また、あるインデックス番号以降の値をすべて指定しない場合は、コンマ ( , ) の記述を省略してください。インデックス番号 1 のトレース取得レベルを (1)、インデックス番号 2 の PRF トレース取得レベルを (2) とした場合、PRF トレース取得レベルは「(1), (2), ...」と指定します。

PRF トレース取得レベルの指定例を次に示します。

PRF トレース取得レベルの指定例	オプションの指定
インデックス番号 1 の PRF トレース取得レベルを指定する	-PrfChangeLevel 0x44445555
インデックス番号 1 とインデックス番号 2 の PRF トレース取得レベルを指定する	-PrfChangeLevel 0x44445555, 0x55554444
インデックス番号 2 の PRF トレース取得レベルを指定する	-PrfChangeLevel , 0x55554444

各機能レイヤーでの PRF トレース取得レベルの割り当てについて説明します。

PRF トレース取得レベルは、インデックスごとに 8 桁の 16 進数で指定します。それぞれの桁には、機能レイヤーが 2 つずつ割り当てられています。16 進数を 2 進数にした場合の上位 2 ビットに割り当てられている機能レイヤーを上位レイヤー、下位 2 ビットに割り当てられている機能レイヤーを下位レイヤーといいます。

桁ごとの上位レイヤー、下位レイヤーに割り当てられている機能レイヤーについて、次の表に示します。



インデックス	桁番号	上位レイヤー	下位レイヤー
インデックス 1	1 桁目	(なし)	Java EE Server
	2 桁目	(なし)	Java VM
	5 桁目	(なし)	uCosminexus TP1 Connector、TP1/Client/J

PRF トレース取得レベルを指定する場合は、桁ごとの上位レイヤー、下位レイヤーの PRF トレース取得レベルを「標準」「詳細」「保守」のどのレベルで取得するかを決め、それを 16 進数で指定します。上位レイヤーと下位レイヤーの PRF トレース取得レベルの組み合わせと指定値の対応を、次の表に示します。

なお、保守レベルは、障害発生時などの保守情報を取得するためのレベルです。通常は指定しないでください。

上位レイヤー	下位レイヤー	指定値
標準	標準	0
標準	詳細	1
標準	保守	2
詳細	標準	4
詳細	詳細	5
詳細	保守	6
保守	標準	8
保守	詳細	9
保守	保守	a

PRF トレース取得レベルの指定例を、次の表に示します。

インデックス	指定例	説明
インデックス 1	00000000	インデックス 1 の各機能レイヤーの標準レベルのトレース情報を取得します。
	10000000	Java EE Server だけ詳細レベルのトレース情報を取得して、ほかの機能レイヤーは標準レベルのトレース情報を取得します。
	11000000	Java EE Server および Java VM の詳細レベルのトレース情報を取得します。

#### -PrfLevelIndex インデックス番号

変更または参照する PRF トレース取得レベルのインデックス番号を 1~16 の整数で指定します。

デフォルト値：1

この引数と一緒に -PrfChangeLevel を指定した場合は、指定したインデックス番号の PRF トレース取得レベルのレベル値を -PrfChangeLevel で指定した値に変更します。

この引数と一緒に -PrfChangeLevel を指定しなかった場合は、指定したインデックス番号の PRF トレース取得レベルのレベル値が表示されます。この引数と一緒に -PrfChangeLevel を指定する場合は、-PrfChangeLevel には必ず 1 つの値だけを指定してください。

(例) インデックス番号 2 番目の値を、0x55554444 に変更する場合

```
cprflevel -PrfChangeLevel 0x55554444 -PrfLevelIndex 2
```

-PrfChangeLevel に複数の値を指定した場合は、オプションエラーとなります。

(例) -PrfChangeLevel に複数の値を指定した場合

```
cprflevel -PrfChangeLevel ,0x55554444,0x55554444 -PrfLevelIndex 2
```

-PrfLevelIndex に 2 を指定しているにもかかわらず、-PrfChangeLevel に複数の値を指定したためです。なお、-PrfLevelIndex の指定を省略した場合は、1 が設定されます。

#### -PrfLevelAll

すべてのレベル値を表示します。-PrfChangeLevel または -PrfLevelIndex と組み合わせた指定はできません。このオプションと -PrfLevelIndex の組み合わせを、次に示します。

cprflevel 現在のレベル値の表示方法		-PrfLevelIndex	
		指定	未指定
-PrfLevelAll	指定	オプションエラー	すべてのレベル値を表示
	未指定	指定されたインデックス番号のレベル値だけを表示	インデックス 1 のレベル値だけを表示

指定できるコマンドオプションの組み合わせの一覧を次に示します。

コマンドオプション	-h	-PRFID	-PrfChangeLevel	-PrfLevelIndex	-PrfLevelAll
-h	-	△-h	△-h	△-h	△-h
-PRFID	△-h	-	○	○	○
-PrfChangeLevel	△-h	○	-	○	×
-PrfLevelIndex	△-h	○	○	-	×
-PrfLevelAll	△-h	○	×	×	-

(凡例)

○：同時に指定できます。

△：△のあとに続くオプション名のオプションが有効になります。

×：同時に指定できません。

-：該当しません。

## 入力例・出力例

- 入力例 1

```
cprflevel
```

- 出力例 1

```
KFCT73415-I 2661 1: prf trace level is 0x12345678
```

- 入力例 2

```
cprflevel -PrfLevelAll
```

- 出力例 2

```
KFCT73418-I 2661 1: all prf trace levels are:  
[ 1]=0x12345678 [ 2]=0x00000000 [ 3]=0x00000000 [ 4]=0x00000000  
[ 5]=0x00000000 [ 6]=0x00000000 [ 7]=0x00000000 [ 8]=0x00000000  
[ 9]=0x00000000 [10]=0x00000000 [11]=0x00000000 [12]=0x00000000  
[13]=0x00000000 [14]=0x00000000 [15]=0x00000000 [16]=0x00000000
```

- 入力例 3

```
cprflevel -PrfLevelIndex 5
```

- 出力例 3

```
KFCT73419-I 2661 1: prf trace level of [5] is 0x00000000.
```

- 入力例 4

```
cprflevel -PrfChangeLevel 00000001
```

- 出力例 4

```
KFCT73416-I 2661 1: prf trace level was changed from 0x00000000 to 0x00000001.
```

- 入力例 5

```
cprflevel -PrfChangeLevel 0x00000055,0x55554444
```

- 出力例 5

```
KFCT73420-I 2661 1: prf trace level was changed:  
[ 1]=0x00000000 -> [ 1]=0x00000055  
[ 2]=0x00000000 -> [ 2]=0x55554444
```

- 入力例 6

```
cprflevel -PrfChangeLevel 0x44445555 -PrfLevelIndex 2
```

- 出力例 6

```
KFCT73420-I 2661 1: prf trace level was changed: [ 2]=0x00000000 -> [ 2]=0x44445555
```

## 戻り値

戻り値	説明
0	正常終了しました。
1	PRF デーモンが起動していません。PRFSPPOOL 環境変数が異なっている、または-PRFID の値が誤っているおそれがあります。
2	PRFSPPOOL 環境変数が設定されていません。
上記以外	コマンド処理中にエラーが発生しました。出力されたメッセージに従って対策したあと、再度コマンドを実行してください。エラーメッセージは標準エラー出力、ログファイルに出力します。

## 注意事項

- 現在の PRF トレース取得レベルを表示する場合、-PrfLevelIndex と -PrfLevelAll の組み合わせで、出力されるメッセージ ID は、「KFCT73415-I」、「KFCT73418-I」、「KFCT73419-I」のどれか 1 つとなっています。PRF トレース取得レベルを表示する場合の出力されるメッセージ ID の対応は、次のとおりです。

PRF トレース取得レベルを表示する場合の メッセージ ID		-PrfLevelAll	
		指定	未指定
-PrfLevelIndex	指定	—	KFCT73419-I
	未指定	KFCT73418-I	KFCT73415-I

- PRF トレース取得レベルを変更する場合、-PrfChangeLevel の指定方法で出力されるメッセージ ID は、「KFCT73416-I」または「KFCT73420-I」のどちらかになります。次の条件をすべて満たした場合、「KFCT73416-I」が出力されます。それ以外は、「KFCT73420-I」が出力されます。
  - PrfChangeLevel 0x44445555 のように値を 1 つだけ指定し、コンマ (,) を使用していない。
  - PrfLevelIndex を指定していない。
- このコマンドでレベル変更した場合、次のリクエストから変更が有効になります。

# 索引

## 数字

- 1 つ以上のライブラリー JAR ファイルの Java EE Server からの除去 409
- 1 つ以上のライブラリーのアーカイブを Java EE Server に追加 399
- 1 つ以上のロガーのロギング属性の設定 311

## A

- add-library 399
- add-resources 435
- apclient 31
- Application Server と連携しない Java アプリケーションプログラムの実行 649
- asadmin 35

## B

- backup-domain 82

## C

- car\_tar\_gz 638
- car\_tar\_Z 640
- change-admin-password 84
- change-master-password 265
- cleanup-prf 133
- cleanup-webserver 135
- collect-local-snapshot 301
- collect-snapshot 305
- copy-config 137
- core アーカイブ機能 638, 640
- core ファイルとスレッドダンプの取得 652
- cprfed 702
- cprflush 708
- cprfgetpid 691
- cprflevel 710
- cprfstart 693
- cprfstop 699
- create-admin-object 505

- create-application-ref 214
- create-auth-realm 267
- create-cluster 141
- create-connector-connection-pool 508
- create-connector-resource 514
- create-connector-security-map 517
- create-connector-work-security-map 519
- create-context-service 570
- create-custom-resource 470
- create-domain 85
- create-event-hook 604
- create-file-user 272
- create-iiop-listener 429
- create-instance 148
- create-jacc-provider 275
- create-javamail-resource 463
- create-jdbc-connection-pool 442
- create-jdbc-resource 455
- create-jms-host 481
- create-jms-resource 483
- create-jmsdest 488
- create-jndi-resource 472
- create-jvm-options 320
- create-managed-executor-service 572
- create-managed-scheduled-executor-service 576
- create-managed-thread-factory 580
- create-message-security-provider 595
- create-node-config 109
- create-node-ssh 111
- create-password-alias 278
- create-prf 152
- create-profiler 401
- create-protocol 413
- create-relation 203
- create-resource-adapter-config 521
- create-resource-ref 437

create-ssl 414  
create-system-properties 322  
create-threadpool 549  
create-webserver 156

## D

delete-admin-object 523  
delete-application-ref 216  
delete-auth-realm 279  
delete-cluster 159  
delete-config 160  
delete-connector-connection-pool 525  
delete-connector-resource 526  
delete-connector-security-map 527  
delete-connector-work-security-map 529  
delete-context-service 582  
delete-custom-resource 475  
delete-domain 94  
delete-event-hook 607  
delete-file-user 281  
delete-iiop-listener 431  
delete-instance 161  
delete-jacc-provider 282  
delete-javamail-resource 466  
delete-jdbc-connection-pool 457  
delete-jdbc-resource 459  
delete-jms-host 493  
delete-jms-resource 495  
delete-jmsdest 496  
delete-jndi-resource 476  
delete-jvm-options 324  
delete-managed-executor-service 584  
delete-managed-scheduled-executor-service 585  
delete-managed-thread-factory 587  
delete-message-security-provider 599  
delete-node-config 115  
delete-node-ssh 116  
delete-password-alias 284

delete-prf 163  
delete-profiler 403  
delete-protocol 419  
delete-relation 207  
delete-resource-adapter-config 530  
delete-resource-ref 439  
delete-ssl 421  
delete-system-property 326  
delete-threadpool 551  
delete-webserver 165  
deploy 218  
disable 225  
disable-secure-admin 285

## E

eheapprof 642  
enable 227  
enable-secure-admin 286  
Explicit ヒープ詳細情報付き拡張スレッドダンプの出力 642  
export-sync-bundle 166

## F

flush-connection-pool 531  
flush-jmsdest 498

## G

GC の強制発生 664  
get 328  
get-client-stubs 229  
get-health 169

## H

htpasswd 614  
httpsd 616  
httpsdctl 618  
hwscertutil cert -hash 624  
hwscertutil cert -outform 625  
hwscertutil cert -text 626

hwscertutil req 627  
hwscertutil reqgen 628  
hwskeygen 629  
hwstraceinfo 620

## I

IIOp リスナーの一覧表示 433  
IIOp リスナーの削除 431  
IIOp リスナーの追加 429  
import-sync-bundle 170  
imqcmd list dst 40  
ist-auth-realms 288

## J

JACC プロバイダーの一覧表示 293  
JACC プロバイダーの削除 282  
Java EE Server クラスターの削除 159  
Java EE Server クラスターの作成 141  
Java EE Server で利用できる暗号化方式群の一覧表示 295  
Java EE Server ドメインの作成 85  
Java EE Server ノードの一覧表示 117  
Java EE Server のクラスターの停止 191  
Java EE Server のバージョン情報の表示 411  
Java EE Server モジュールの一覧表示 236  
Java EE サーバで使用する asadmin コマンドのサブコマンド一覧 69  
Java EE サーバで使用するコマンドの一覧 30  
java\_hras 649  
javacore 652  
javagc 664  
JavaMail セッションリソースの一覧表示 468  
JavaMail セッションリソースの削除 466  
JavaMail セッションリソースの作成 463  
javatrace 675  
Java VM オプションの一覧表示 332  
Java VM オプションの削除 324  
Java VM オプションの作成 320  
Java VM で使用するコマンドの一覧 637

Java インターフェースクラスからの WSDL 生成、WSDL からサービスエンドポイントインターフェースやスタブなどの Java クラスの生成 48

JAX-RPC の Web サービスとしてデプロイできる WAR ファイルの生成 52

JAX-WS Web サービスで使用する JAX-WS ポータブルのアーティファクトの生成 54

jax-ws ポータブルのアーティファクトの生成 58

JDBC コネクションプールの一覧表示 460

JDBC コネクションプールの削除 457

JDBC コネクションプールの登録 442

JDBC またはコネクタコネクションプールの検証 541

JDBC リソースの一覧表示 461

JDBC リソースの作成 455

jheapprof 678

jheapprofanalyzer 684

JMS 送信先内のメッセージの削除 498

JMS の物理送信先の作成 488

JMS 物理送信先の一覧表示 503

JMS 物理送信先の削除 496

JMS ホストの一覧表示 500

JMS ホストの削除 493

JMS ホストの作成 481

JMS リソースの一覧表示 501

JMS リソースの削除 495

JMS リソースの作成 483

JNDI 名を持つ JDBC リソースの削除 459

JNDI リソースの一覧表示 479

JNDI リソースの削除 476

JNDI リソースの登録 472

jspc 41

JSP ファイルの事前コンパイル 41

## L

list 330  
list-admin-objects 533  
list-application-refs 230  
list-applications 232  
list-backups 95

- list-batch-job-executions 558
- list-batch-job-steps 561
- list-batch-jobs 563
- list-batch-runtime-configuration 566
- list-clusters 172
- list-commands 405
- list-configs 331
- list-connector-connection-pools 535
- list-connector-resources 536
- list-connector-security-maps 537
- list-connector-work-security-maps 539
- list-containers 235
- list-context-services 588
- list-custom-resources 478
- list-domains 97
- list-event-hooks 609
- list-file-groups 289
- list-file-users 291
- list-iiop-listeners 433
- list-instances 174
- list-jacc-providers 293
- list-javamail-resources 468
- list-jdbc-connection-pools 460
- list-jdbc-resources 461
- list-jms-hosts 500
- list-jms-resources 501
- list-jmsdest 503
- list-jndi-resources 479
- list-jvm-options 332
- list-libraries 406
- list-log-attributes 309
- list-managed-executor-services 590
- list-managed-scheduled-executor-services 591
- list-managed-thread-factories 592
- list-message-security-providers 601
- list-modules 236
- list-network-listeners 422
- list-nodes 117
- list-nodes-config 119

- list-nodes-ssh 120
- list-password-aliases 294
- list-prfs 177
- list-protocols 424
- list-relations 208
- list-resource-adapter-configs 540
- list-resource-refs 440
- list-sub-components 237
- list-supported-cipher-suites 295
- list-system-properties 334
- list-threadpools 553
- list-timers 239
- list-transports 425
- list-virtual-servers 426
- list-web-context-param 240
- list-web-env-entry 242
- list-webservers 180
- login 408
- logresolve 622

## M

- ManagedExecutorService の一覧表示 590
- ManagedExecutorService リソースの削除 584
- ManagedExecutorService リソースの作成 572
- ManagedScheduledExecutorService リソースの一覧表示 591
- ManagedScheduledExecutorService リソースの削除 585
- ManagedScheduledExecutorService リソースの作成 576
- ManagedThreadFactory リソースの一覧表示 592
- ManagedThreadFactory リソースの削除 587
- ManagedThreadFactory リソースの作成 580

## P

- ping-connection-pool 541
- ping-node-ssh 121
- PRF 環境のクリーンアップ 133
- PRF デーモンの開始 693
- PRF デーモンの停止 699



PRF デーモンプロセス ID の取得 691  
PRF で使用するコマンドの一覧 690  
PRF トレース取得レベルの表示と変更 710  
PRF の一覧表示 177  
PRF の起動 188  
PRF の構築 152  
PRF の削除 163  
PRF の停止 195

## R

recover-transactions 555  
redploy 244  
remove-library 409  
restart-domain 98  
restore-domain 100

## S

schemagen 46  
secure admin 機能の無効化 285  
secure admin 機能の有効化 286  
set 335  
set-batch-runtime-configuration 568  
set-log-attributes 311  
set-log-attributes コマンドで使用するパラメーター一覧 313  
set-web-context-param 250  
set-web-env-entry 253  
setup-ssh 123  
set サブコマンドおよび get サブコマンドで使用するパラメーター一覧 336  
show-component-status 256  
SSH が有効なノードの作成 111  
SSH キーのセットアップ 123  
SSH 経由の通信をサポートするドメイン内の Java EE Server ノードの一覧表示 120  
SSH 通信が有効なノードの削除 116  
SSH 通信によるノード使用のテスト 121  
SSH を使ったノード構成データの更新 128  
sslpaswd 631  
SSL 設定の削除 421

start-cluster 183  
start-domain 103  
start-instance 185  
start-prf 188  
start-servers 189  
start-webserver 190  
stop-cluster 191  
stop-domain 106  
stop-instance 193  
stop-prf 195  
stop-servers 196  
stop-webserver 197

## U

undeploy 258  
unset-web-context-param 260  
unset-web-env-entry 263  
update-connector-security-map 544  
update-connector-work-security-map 546  
update-file-user 296  
update-node-config 127  
update-node-ssh 128  
update-password-alias 298

## V

validate-multicast 199  
version 411

## W

Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの設定 250  
Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの設定の解除 260  
Web アプリケーションまたはモジュールの環境エンタリーの設定 253  
Web アプリケーションまたはモジュールの環境エンタリーの設定の解除 263  
Web サーバ環境のクリーンアップ 135  
Web サーバで使用するコマンドの一覧 612

Web サーバの一覧表示 180  
Web サーバの起動 190, 616  
Web サーバの起動、再起動および停止 618  
Web サーバの構築 156  
Web サーバの削除 165  
Web サーバの停止 197  
Web サーバの秘密鍵の作成 629  
wscompile 48  
wsdeploy 52  
wsgen 54  
wsimport 58

## X

xjc 63  
XML Schema から Java へバインディングするコンパイラー 63  
XML ファイルに指定されたリソースの追加 435

## あ

アクセスログファイル内の IP アドレスのホスト名への変換 622  
アプリケーションコンテナの一覧表示 235  
アプリケーション参照の削除 216  
アプリケーション参照の作成 214

## い

一般ユーザーによる運用 633  
イベント検知時コマンドの一覧表示 609  
イベント検知時コマンドの登録 604  
イベント検知時コマンドの登録解除 607

## か

カスタムリソースの削除 475  
カスタムリソースの作成 470  
カスタムリソースの取得 478  
仮想サーバの一覧表示 426  
管理された全オブジェクトの取得 533  
管理対象オブジェクトの削除 523  
管理対象オブジェクトの作成 505  
管理タスクの実行 35

管理ユーザーパスワードの変更 84

## き

既存のアプリケーション参照の一覧表示 230  
既存の名前付き構成をコピーして、別の構成を作成 137

## <

クライアントアプリケーションの起動 31  
クライアントスタブの JAR ファイルの取得 229  
クラスター化されたインスタンスまたはスタンドアロンサーバインスタンスの構成データのインポート 170  
クラスターの起動 183  
クラスターの状態に関する情報の取得 169  
クラスターまたはスタンドアロンサーバインスタンスの構成データのアーカイブファイルへのエクスポート 166  
クラス別統計情報解析ファイルの CSV 出力 684  
クラス別統計情報付き拡張スレッドダンプの出力 678

## け

現在のファイルユーザーの更新 296

## こ

構成属性値の一覧表示 330  
構成属性値の取得 328  
構成属性値の設定 335  
構成のほかの部分から参照できるシステムプロパティーの追加 322  
コネクションプールで確立されたコネクションの再初期化 531  
コネクションプールの追加 508  
コネクターコネクションプールに属するセキュリティーマップの一覧表示 537  
コネクターコネクションプールの一覧表示 535  
コネクターコネクションプールの削除 525  
コネクターコネクションプールのセキュリティーマップの削除 527  
コネクターコネクションプールのセキュリティーマップの作成 517

コネクタコネクションプールのセキュリティーマップの変更 544

コネクタリソースの一覧表示 536

コネクタリソースの削除 526

コネクタリソースの登録 514

コマンドの入力方法 25

コマンドリファレンスの見方 27

コンテキストサービスのリソースの一覧表示 588

コンテキストサービスのリソースの削除 582

コンテキストサービスのリソースの作成 570

コンポーネントのデプロイ 218

コンポーネントの無効化 225

コンポーネントの有効化 227

コンポーネントのリデプロイ 244

## さ

サーバインスタンスが所有する永続タイマーの一覧表示 239

サーバインスタンスの一覧表示 174

サーバインスタンスの起動 185

サーバインスタンスの削除 161

サーバインスタンスの作成 148

サーバ間関連を一覧表示 208

サーバ間関連を削除 207

サーバ間関連を作成 203

サーバの一括起動 189

サーバの一括停止 196

## し

システムプロパティ（ドメイン／構成／クラスター／サーバインスタンス）の削除 326

実行中のサーバインスタンスの停止 193

実行待ちトランザクションの手動での回復 555

使用できるコマンドの一覧表示 405

証明書の形式変換 625

証明書の内容表示 626

証明書発行要求 CSR の作成 628

証明書発行要求 CSR の内容表示 627

新規プロトコルの追加 413

## す

スキーマジェネレーターの起動 46

スレッドプールの一覧表示 553

スレッドプールの削除 551

スレッドプールの追加 549

## せ

性能解析トレース情報の編集出力 702

セキュア通信の構成 414

## て

デプロイされた Web アプリケーションまたはモジュールの ServletContext 初期化パラメーターの一覧表示 240

デプロイされた Web アプリケーションまたはモジュールの環境エントリーの一覧表示 242

デプロイされたモジュール内またはデプロイされたアプリケーションモジュール内の EJB またはサーブレットのコンポーネントの一覧表示 237

デプロイ済みアプリケーションの一覧表示 232

デプロイ済みのコンポーネントの削除 258

デプロイ済みのコンポーネントの状態の表示 256

## と

ドメイン、構成、クラスターまたはサーバインスタンスのシステムプロパティの一覧表示 334

ドメイン管理サーバ稼働時のシステム情報の収集 305

ドメイン管理サーバの起動 103

ドメイン管理サーバの構成中の、名前付き構成の一覧表示 331

ドメイン管理サーバの再起動 98

ドメイン管理サーバの停止 106

ドメイン管理サーバ非稼働時のシステム情報の収集 301

ドメインディレクトリー内のドメインの一覧表示 97

ドメイン内の既存クラスターの一覧表示 172

ドメインの削除 94

ドメインのバックアップ 82

ドメインバックアップの一覧表示 95

ドメインへのログイン 408

トランスポートの一覧表示 425

トレース情報の収集 675

## な

内部トレースの採取 620

名前付き既存構成の削除 160

名前付き認証レールの削除 279

名前付き認証レールの追加 267

## に

任意の認証モジュールで使用できる JACC プロバイダーの作成 275

認証レールの一覧表示 288

## ね

ネットワークリスナーの一覧表示 422

## の

ノード構成データの更新 127

## は

パスワードエイリアスの一覧表示 294

パスワードエイリアスの更新 298

パスワードエイリアスの削除 284

パスワードエイリアスの作成 278

パスワード付きサーバ秘密鍵のパスワードファイル作成 631

バックアップからのファイルの復元 100

ハッシュリンクの作成 624

バッチ実行時の構成の表示 566

バッチジョブ実行の一覧表示 558

バッチジョブの一覧表示 563

バッチジョブの実行ステップの一覧表示 561

バッチランタイムの設定 568

バッファ内容の強制ファイル出力 708

## ふ

ファイルグループの一覧表示 289

ファイルユーザーの一覧表示 291

ファイルユーザーの削除 281

ファイルユーザーの作成 272

物理格納先の一覧表示 40

プロトコルの一覧表示 424

プロトコルの削除 419

プロファイラー要素の削除 403

プロファイラー要素の作成 401

## ま

マスターパスワードの変更 265

マルチキャストトランスポートの検証 199

## め

メッセージセキュリティープロバイダーの削除 599

メッセージセキュリティープロバイダーの作成 595

メッセージレイヤーのメッセージセキュリティープロバイダーの一覧表示 601

## ゆ

ユーザー名とパスワードのパスワードファイルへの登録およびパスワードの変更 614

## ら

ライブラリーのアーカイブの一覧表示 406

## り

リソースアダプターアーカイブ構成情報の削除 530

リソースアダプターアーカイブ構成情報の作成 521

リソースアダプターアーカイブ構成の名称の一覧表示 540

リソースアダプターに属するワークセキュリティーマップの一覧表示 539

リソースアダプターのワークセキュリティーマップの削除 529

リソースアダプターのワークセキュリティーマップの作成 519

リソースアダプターのワークセキュリティーマップの変更 546

リソース参照の一覧表示 440

リソース参照の削除 439

リソースへの参照の作成 437

リモート通信が無効なノードの削除 115

リモート通信が有効化されていないノードの作成 109

リモート通信をサポートしないドメイン内の Java EE  
Server ノードの一覧表示 119

## ろ

ログ属性の一覧表示 309

---

 株式会社 日立製作所

〒 100-8280 東京都千代田区丸の内一丁目 6 番 6 号

---