

Hitachi Application Server V10
定義リファレンス (Windows®用)

3021-3-420-40

前書き

■ 対象製品

適用 OS : Windows Server 2008 R2、Windows Server 2012、Windows Server 2012 R2

P-2943-5KA4 Hitachi Application Server 10-11

適用 OS : Windows 7 x64、Windows 8 x64、Windows 8.1 x64、Windows 10 x64、Windows Server 2008 R2、Windows Server 2012、Windows Server 2012 R2

P-2943-5FA4 Hitachi Application Server for Developers 10-11

注 このマニュアルで「Windows x86 の場合」と表記している個所については、これら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。

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。

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。

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。

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[®]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> 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 3021-3-420-40

■ 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(3021-3-420-40)

Hitachi Application Server 10-11、Hitachi Application Server for Developers 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

このマニュアルでは Hitachi Application Server のシステム基盤の構築で使用する定義ファイル、Deployment Descriptor (DD)、プロパティ などの形式や定義方法について説明します。このマニュアルを利用することで、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ています。

■ 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
- アプリケーション開発者

前提知識をそれぞれ次に示します。

- システム構築者
 - 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
- アプリケーション開発者
 - Windows の操作に関する知識
 - Java EE 標準仕様に関する知識
 - Java によるプログラム開発に関する知識
 - Eclipse および WTP 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。

Hitachi 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』 (Windows [®] 用)	3021-3-414

表記	正式名称	資料番号
『GUI リファレンス』	『Hitachi Application Server V10 GUI リファレンス』 (Windows®用)	3021-3-416
『コマンドリファレンス』	『Hitachi Application Server V10 コマンドリファレンス』 (Windows®用)	3021-3-418
『定義リファレンス』	『Hitachi Application Server V10 定義リファレンス』 (Windows®用)	3021-3-420
『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®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®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

JP1 関連

表記	正式名称	資料番号
『JP1/Automatic Job Management System 導入ガイド』	『JP1 Version 10 JP1/Automatic Job Management System 3 導入ガイド』	3021-3-102
『JP1/Automatic Job Management System 設計ガイド』	『JP1 Version 10 JP1/Automatic Job Management System 3 設計ガイド (システ ム構築編)』	3021-3-103
	『JP1 Version 10 JP1/Automatic Job Management System 3 設計ガイド (業務設 計編)』	3021-3-104
『JP1/Automatic Job Management System 構築ガイド 1』	『JP1 Version 10 JP1/Automatic Job Management System 3 構築ガイド 1』	3021-3-105
『JP1/Automatic Job Management System 操作ガイド』	『JP1 Version 10 JP1/Automatic Job Management System 3 操作ガイド』	3021-3-109

■ 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
ADT		Hitachi Application Development Tools
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
Eclipse		Eclipse IDE for Java EE Developers
Excel		Microsoft® Excel
Firefox		Firefox®
HiRDB	HiRDB Version 9	HiRDB Server Version 9
	HiRDB/Single Server	HiRDB/Single Server Version 9
Hyper-V		Microsoft® Hyper-V®
HWS		Hitachi Web Server
Web Server		
Internet Explorer		Windows® Internet Explorer®
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
		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

表記		製品名と機能名	
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	
Virtage		Hitachi Virtage	
VMware ESX		VMware vSphere® ESX	
VMware Horizon View		VMware Horizon® (with View)	
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)
			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)
		Windows 7 x64	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 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 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 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 Pro 日本語版(64 ビット版)	

表記		製品名と機能名
		Windows® 8.1 Enterprise 日本語版(64 ビット版)
Windows 10	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		

Windows に関しては、32 ビット版の Windows を Windows x86、64 ビット版の Windows を Windows x64 と表記することがあります。

■ 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

英略語	英字での表記
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
DN	Distinguished Name
DNS	Domain Name System
DoS	Denial of Service
DSO	Dynamic Shared Object
DTD	Document Type Definition
DTP	Eclipse Data Tools Platform
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

英略語	英字での表記
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
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

英略語	英字での表記
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
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
JSP	JavaServer™ Faces
	JavaServer™ Faces Reference Implementation (RI) Version: 1.1_01 FCS
JSON-P	Java™ API for JSON Processing
JSP	JavaServer Pages™
	JSP™

英略語	英字での表記
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
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
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

英略語	英字での表記
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
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
WTP	Eclipse Web Tools Platform
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、1,024² バイト、1,024³ バイト、1,024⁴ バイト、1,024⁵ バイトです。

目次

前書き	2
変更内容	5
はじめに	6

1 定義リファレンスの見方 39

1.1	定義リファレンスの記述形式	40
1.2	定義リファレンスで使用している記号	43
1.3	定義リファレンスで使用している構文要素	44

2 Web サーバで使用する定義ファイルおよび定義項目 45

2.1	Web サーバで使用する定義ファイルおよび定義項目の一覧	46
2.2	httpsd.conf	59
2.3	httpsd.conf に設定する定義項目の詳細	72
2.3.1	<Directory>	72
2.3.2	<DirectoryMatch>	73
2.3.3	<Files>	73
2.3.4	<FilesMatch>	74
2.3.5	<IfModule>	74
2.3.6	<Limit>	74
2.3.7	<Location>	76
2.3.8	<LocationMatch>	76
2.3.9	<Proxy>	77
2.3.10	<VirtualHost>	78
2.3.11	AccessFileName	79
2.3.12	Action	79
2.3.13	AddAlt	80
2.3.14	AddAltByEncoding	81
2.3.15	AddAltByType	81
2.3.16	AddCharset	82
2.3.17	AddDefaultCharset	83
2.3.18	AddDescription	84
2.3.19	AddEncoding	85
2.3.20	AddHandler	85
2.3.21	AddIcon	86
2.3.22	AddIconByEncoding	88

2.3.23	AddIconByType	88
2.3.24	AddLanguage	89
2.3.25	AddType	90
2.3.26	Alias	91
2.3.27	AliasMatch	92
2.3.28	Allow	93
2.3.29	AllowOverride	94
2.3.30	AuthBasicAuthoritative	95
2.3.31	AuthGroupFile	96
2.3.32	AuthName	97
2.3.33	AuthType	97
2.3.34	AuthUserFile	98
2.3.35	BalancerMember	99
2.3.36	BrowserMatch	100
2.3.37	BrowserMatchNoCase	101
2.3.38	CacheNegotiatedDocs	101
2.3.39	CustomLog	102
2.3.40	DefaultIcon	106
2.3.41	DefaultLanguage	106
2.3.42	Deny	107
2.3.43	DirectoryIndex	108
2.3.44	DocumentRoot	109
2.3.45	ErrorDocument	109
2.3.46	ErrorLog	112
2.3.47	ExpiresActive	112
2.3.48	ExpiresByType	113
2.3.49	ExpiresDefault	114
2.3.50	ExtendedStatus	116
2.3.51	FileETag	116
2.3.52	ForceType	118
2.3.53	Header	119
2.3.54	HeaderName	120
2.3.55	HostnameLookups	121
2.3.56	HWSErrorDocumentMETACharset	122
2.3.57	HWSErrorLogClientAddr	123
2.3.58	HWSSuccessfulStopLog	123
2.3.59	HWSSuccessfulStopTimeout	124
2.3.60	HWSSuccessfulMenuCharset	125
2.3.61	HWSSuccessfulSSLVerbose	126

2.3.62	HWSLogTimeVerbose	126
2.3.63	HWSMaxQueueSize	127
2.3.64	HWSPrfId	128
2.3.65	HWSPrfSpool	128
2.3.66	HWSProxyPassReverseCookie	129
2.3.67	HWSRequestLog	129
2.3.68	HWSRequestLogType	130
2.3.69	HWSSetEnvIfIPV6	132
2.3.70	HWSSuppressModuleTrace	132
2.3.71	HWSTraceIdFile	134
2.3.72	HWSTraceLogFile	135
2.3.73	ImapBase	136
2.3.74	ImapDefault	136
2.3.75	ImapMenu	137
2.3.76	Include	138
2.3.77	IndexIgnore	139
2.3.78	IndexOptions	139
2.3.79	IndexOrderDefault	143
2.3.80	KeepAlive	144
2.3.81	KeepAliveTimeout	145
2.3.82	LanguagePriority	146
2.3.83	LimitRequestBody	147
2.3.84	LimitRequestFields	147
2.3.85	LimitRequestFieldSize	148
2.3.86	LimitRequestLine	149
2.3.87	Listen	150
2.3.88	ListenBacklog	150
2.3.89	LoadFile	151
2.3.90	LoadModule	151
2.3.91	LogFormat	152
2.3.92	LogLevel	153
2.3.93	MaxKeepAliveRequests	154
2.3.94	MultiviewsMatch	155
2.3.95	Options	156
2.3.96	Order	157
2.3.97	PassEnv	158
2.3.98	PidFile	159
2.3.99	ProxyErrorOverride	159
2.3.100	ProxyPass	161

2.3.101	ProxyPassReverse	162
2.3.102	ProxyPreserveHost	164
2.3.103	ProxyTimeout	165
2.3.104	ProxyVia	166
2.3.105	QOSCookieDomain	167
2.3.106	QOSCookieExpires	167
2.3.107	QOSCookieName	169
2.3.108	QOSCookieSecure	171
2.3.109	QOSCookieServers	171
2.3.110	QOSRedirect	172
2.3.111	QOSRejectionServers	173
2.3.112	QOSResponse	174
2.3.113	ReadmeName	175
2.3.114	Redirect	175
2.3.115	RedirectMatch	177
2.3.116	RequestHeader	178
2.3.117	RequestReadTimeout	180
2.3.118	Require	181
2.3.119	Satisfy	182
2.3.120	Script	183
2.3.121	ScriptAlias	183
2.3.122	ScriptAliasMatch	184
2.3.123	ScriptInterpreterSource	185
2.3.124	ScriptLog	186
2.3.125	ScriptLogBuffer	186
2.3.126	ScriptLogLength	187
2.3.127	SendBufferSize	188
2.3.128	ServerAdmin	189
2.3.129	ServerAlias	189
2.3.130	ServerName	189
2.3.131	ServerPath	190
2.3.132	ServerRoot	191
2.3.133	ServerSignature	191
2.3.134	ServerTokens	192
2.3.135	SetEnv	193
2.3.136	SetEnvIf	194
2.3.137	SetEnvIfNoCase	196
2.3.138	SetHandler	197
2.3.139	SSLBanCipher	198

2.3.140	SSLCACertificateFile	198
2.3.141	SSLCertificateFile	199
2.3.142	SSLCertificateKeyFile	200
2.3.143	SSLCertificateKeyPassword	200
2.3.144	SSLCRLAuthoritative	201
2.3.145	SSLCRLDERPath	202
2.3.146	SSLCRLPEMPPath	202
2.3.147	SSLDenySSL	203
2.3.148	SSLDisable	203
2.3.149	SSLEnable	204
2.3.150	SSLExportCertChainDepth	204
2.3.151	SSLExportClientCertificates	205
2.3.152	SSLFakeBasicAuth	206
2.3.153	SSLProtocol	207
2.3.154	SSLRequireCipher	208
2.3.155	SSLRequiredCiphers	208
2.3.156	SSLRequireSSL	209
2.3.157	SSLSessionCacheSize	210
2.3.158	SSLSessionCacheTimeout	211
2.3.159	SSLVerifyClient	212
2.3.160	SSLVerifyDepth	212
2.3.161	ThreadsPerChild	214
2.3.162	Timeout	214
2.3.163	TraceEnable	215
2.3.164	TransferLog	216
2.3.165	TypesConfig	217
2.3.166	UnsetEnv	218
2.3.167	UseCanonicalName	219
2.3.168	UserDir	220
2.4	mime.types	222
2.5	アクセス制御ファイル	223
2.6	Web サーバで使用する定義項目 (ユーティリティー) の詳細	228
2.6.1	rotatelogs	228
2.6.2	rotatelogs2	230
3	Java EE サーバで使用する Deployment Descriptor (DD)	233
3.1	Java EE サーバで使用する DD の一覧	234
3.2	glassfish-acc.xml	235
3.3	glassfish-acc.xml に設定する要素の詳細	237

- 3.3.1 /client-container 237
- 3.3.2 /client-container/target-server 237
- 3.3.3 /client-container/target-server/description 238
- 3.3.4 /client-container/target-server/security 239
- 3.3.5 /client-container/auth-realm 239
- 3.3.6 /client-container/auth-realm/property 241
- 3.3.7 /client-container/message-security-config 242
- 3.3.8 /client-container/message-security-config/provider-config 242
- 3.3.9 /client-container/message-security-config/provider-config/request-policy 244
- 3.3.10 /client-container/message-security-config/provider-config/response-policy 244
- 3.3.11 /client-container/message-security-config/provider-config/property 245
- 3.3.12 /client-container/property 246
- 3.4 glassfish-application.xml 247
- 3.5 glassfish-application.xml に設定する要素の詳細 249
 - 3.5.1 /glassfish-application 249
 - 3.5.2 /glassfish-application/web 249
 - 3.5.3 /glassfish-application/web/web-uri 249
 - 3.5.4 /glassfish-application/web/context-root 250
 - 3.5.5 /glassfish-application/security-role-mapping 250
 - 3.5.6 /glassfish-application/security-role-mapping/role-name 251
 - 3.5.7 /glassfish-application/security-role-mapping/principal-name 251
 - 3.5.8 /glassfish-application/security-role-mapping/group-name 252
 - 3.5.9 /glassfish-application/realm 253
 - 3.5.10 /glassfish-application/ejb-ref 254
 - 3.5.11 /glassfish-application/ejb-ref/ejb-ref-name 254
 - 3.5.12 /glassfish-application/ejb-ref/jndi-name 255
 - 3.5.13 /glassfish-application/resource-ref 255
 - 3.5.14 /glassfish-application/resource-ref/res-ref-name 256
 - 3.5.15 /glassfish-application/resource-ref/jndi-name 256
 - 3.5.16 /glassfish-application/resource-ref/default-resource-principal 257
 - 3.5.17 /glassfish-application/resource-ref/default-resource-principal/name 257
 - 3.5.18 /glassfish-application/resource-ref/default-resource-principal/password 258
 - 3.5.19 /glassfish-application/resource-env-ref 258
 - 3.5.20 /glassfish-application/resource-env-ref/resource-env-ref-name 259
 - 3.5.21 /glassfish-application/resource-env-ref/jndi-name 259
 - 3.5.22 /glassfish-application/service-ref 260
 - 3.5.23 /glassfish-application/service-ref/service-ref-name 260
 - 3.5.24 /glassfish-application/service-ref/port-info 261
 - 3.5.25 /glassfish-application/service-ref/port-info/service-endpoint-interface 261

- 3.5.26 /glassfish-application/service-ref/port-info/wsdl-port 262
- 3.5.27 /glassfish-application/service-ref/port-info/wsdl-port/namespaceURI 262
- 3.5.28 /glassfish-application/service-ref/port-info/wsdl-port/localpart 262
- 3.5.29 /glassfish-application/service-ref/port-info/stub-property 263
- 3.5.30 /glassfish-application/service-ref/service-impl-class 264
- 3.5.31 /glassfish-application/service-ref/service-qname 264
- 3.5.32 /glassfish-application/service-ref/service-qname/namespaceURI 264
- 3.5.33 /glassfish-application/service-ref/service-qname/localpart 265
- 3.5.34 /glassfish-application/message-destination-ref 265
- 3.5.35 /glassfish-application/message-destination-ref/message-destination-ref-name 266
- 3.5.36 /glassfish-application/message-destination-ref/jndi-name 266
- 3.5.37 /glassfish-application/message-destination 267
- 3.5.38 /glassfish-application/message-destination/message-destination-name 267
- 3.5.39 /glassfish-application/message-destination/jndi-name 268
- 3.5.40 /glassfish-application/archive-name 268
- 3.5.41 /glassfish-application/version-identifier 269
- 3.6 glassfish-application-client.xml 271
- 3.7 glassfish-application-client.xml に設定する要素の詳細 273
- 3.7.1 /glassfish-application-client 273
- 3.7.2 /glassfish-application-client/ejb-ref 273
- 3.7.3 /glassfish-application-client/ejb-ref/ejb-ref-name 273
- 3.7.4 /glassfish-application-client/ejb-ref/jndi-name 274
- 3.7.5 /glassfish-application-client/resource-ref 274
- 3.7.6 /glassfish-application-client/resource-ref/res-ref-name 275
- 3.7.7 /glassfish-application-client/resource-ref/jndi-name 275
- 3.7.8 /glassfish-application-client/resource-ref/default-resource-principal 276
- 3.7.9 /glassfish-application-client/resource-ref/default-resource-principal/name 276
- 3.7.10 /glassfish-application-client/resource-ref/default-resource-principal/password 277
- 3.7.11 /glassfish-application-client/resource-env-ref 277
- 3.7.12 /glassfish-application-client/resource-env-ref/resource-env-ref-name 278
- 3.7.13 /glassfish-application-client/resource-env-ref/jndi-name 278
- 3.7.14 /glassfish-application-client/service-ref 279
- 3.7.15 /glassfish-application-client/service-ref/service-ref-name 279
- 3.7.16 /glassfish-application-client/service-ref/port-info 280
- 3.7.17 /glassfish-application-client/service-ref/port-info/service-endpoint-interface 280
- 3.7.18 /glassfish-application-client/service-ref/port-info/wsdl-port 281
- 3.7.19 /glassfish-application-client/service-ref/port-info/wsdl-port/namespaceURI 281
- 3.7.20 /glassfish-application-client/service-ref/port-info/wsdl-port/localpart 282
- 3.7.21 /glassfish-application-client/service-ref/port-info/stub-property 282

- 3.7.22 /glassfish-application-client/service-ref/port-info/call-property 283
- 3.7.23 /glassfish-application-client/service-ref/port-info/message-security-binding 284
- 3.7.24 /glassfish-application-client/service-ref/port-info/message-security-binding/
message-security 284
- 3.7.25 /glassfish-application-client/service-ref/port-info/message-security-binding/
message-security/message 285
- 3.7.26 /glassfish-application-client/service-ref/port-info/message-security-binding/
message-security/message/operation-name 285
- 3.7.27 /glassfish-application-client/service-ref/port-info/message-security-binding/
message-security/request-protection 286
- 3.7.28 /glassfish-application-client/service-ref/port-info/message-security-binding/
message-security/response-protection 286
- 3.7.29 /glassfish-application-client/service-ref/call-property 287
- 3.7.30 /glassfish-application-client/service-ref/wsdl-override 288
- 3.7.31 /glassfish-application-client/service-ref/service-impl-class 288
- 3.7.32 /glassfish-application-client/service-ref/service-qname 289
- 3.7.33 /glassfish-application-client/service-ref/service-qname/namespaceURI 289
- 3.7.34 /glassfish-application-client/service-ref/service-qname/localpart 290
- 3.7.35 /glassfish-application-client/message-destination-ref 290
- 3.7.36 /glassfish-application-client/message-destination-ref/message-destination-ref-name 291
- 3.7.37 /glassfish-application-client/message-destination-ref/jndi-name 291
- 3.7.38 /glassfish-application-client/message-destination 292
- 3.7.39 /glassfish-application-client/message-destination/message-destination-name 292
- 3.7.40 /glassfish-application-client/message-destination/jndi-name 292
- 3.7.41 /glassfish-application-client/version-identifier 293
- 3.8 glassfish-ejb-jar.xml 295
- 3.9 glassfish-ejb-jar.xml に設定する要素の詳細 298
 - 3.9.1 /glassfish-ejb-jar 298
 - 3.9.2 /glassfish-ejb-jar/security-role-mapping 298
 - 3.9.3 /glassfish-ejb-jar/security-role-mapping/role-name 298
 - 3.9.4 /glassfish-ejb-jar/security-role-mapping/principal-name 299
 - 3.9.5 /glassfish-ejb-jar/security-role-mapping/group-name 300
 - 3.9.6 /glassfish-ejb-jar/enterprise-beans 301
 - 3.9.7 /glassfish-ejb-jar/enterprise-beans/name 301
 - 3.9.8 /glassfish-ejb-jar/enterprise-beans/ejb 301
 - 3.9.9 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-name 302
 - 3.9.10 /glassfish-ejb-jar/enterprise-beans/ejb/jndi-name 303
 - 3.9.11 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-ref 303
 - 3.9.12 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-ref/ejb-ref-name 303
 - 3.9.13 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-ref/jndi-name 304

- 3.9.14 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref 304
- 3.9.15 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/res-ref-name 305
- 3.9.16 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/jndi-name 306
- 3.9.17 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/default-resource-principal 306
- 3.9.18 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/default-resource-principal/
name 306
- 3.9.19 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/default-resource-principal/
password 307
- 3.9.20 /glassfish-ejb-jar/enterprise-beans/ejb/resource-env-ref 308
- 3.9.21 /glassfish-ejb-jar/enterprise-beans/ejb/resource-env-ref/resource-env-ref-name 308
- 3.9.22 /glassfish-ejb-jar/enterprise-beans/ejb/resource-env-ref/jndi-name 308
- 3.9.23 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref 309
- 3.9.24 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-ref-name 309
- 3.9.25 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info 310
- 3.9.26 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/service-endpoint-
interface 310
- 3.9.27 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/wsdl-port 311
- 3.9.28 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/wsdl-port/
namespaceURI 311
- 3.9.29 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/wsdl-port/localpart 312
- 3.9.30 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/stub-property 312
- 3.9.31 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/call-property 313
- 3.9.32 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding 314
- 3.9.33 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security 314
- 3.9.34 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/message 315
- 3.9.35 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/message/java-method 315
- 3.9.36 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/message/java-method/method-name 316
- 3.9.37 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/message/java-method/method-params 316
- 3.9.38 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/message/java-method/method-params/method-param 317
- 3.9.39 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/message/operation-name 317
- 3.9.40 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/request-protection 318
- 3.9.41 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-
binding/message-security/response-protection 319
- 3.9.42 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/call-property 319

- 3.9.43 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/wsdl-override 320
- 3.9.44 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-impl-class 321
- 3.9.45 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-qname 321
- 3.9.46 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-qname/namespaceURI 322
- 3.9.47 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-qname/localpart 322
- 3.9.48 /glassfish-ejb-jar/enterprise-beans/ejb/message-destination-ref 323
- 3.9.49 /glassfish-ejb-jar/enterprise-beans/ejb/message-destination-ref/message-destination-ref-name 323
- 3.9.50 /glassfish-ejb-jar/enterprise-beans/ejb/message-destination-ref/jndi-name 324
- 3.9.51 /glassfish-ejb-jar/enterprise-beans/ejb/principal 324
- 3.9.52 /glassfish-ejb-jar/enterprise-beans/ejb/principal/name 324
- 3.9.53 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory 325
- 3.9.54 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/jndi-name 325
- 3.9.55 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/default-resource-principal 326
- 3.9.56 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/default-resource-principal/name 326
- 3.9.57 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/default-resource-principal/password 327
- 3.9.58 /glassfish-ejb-jar/enterprise-beans/ejb/jms-durable-subscription-name 327
- 3.9.59 /glassfish-ejb-jar/enterprise-beans/ejb/jms-max-messages-load 328
- 3.9.60 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config 328
- 3.9.61 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config 329
- 3.9.62 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/integrity 329
- 3.9.63 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/confidentiality 330
- 3.9.64 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/establish-trust-in-target 330
- 3.9.65 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/establish-trust-in-client 331
- 3.9.66 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context 331
- 3.9.67 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context/auth-method 332
- 3.9.68 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context/realm 332
- 3.9.69 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context/required 333
- 3.9.70 /glassfish-ejb-jar/enterprise-beans/ejb/commit-option 334
- 3.9.71 /glassfish-ejb-jar/enterprise-beans/ejb/cmt-timeout-in-seconds 334
- 3.9.72 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool 335
- 3.9.73 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/steady-pool-size 335
- 3.9.74 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/resize-quantity 336
- 3.9.75 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/max-pool-size 337
- 3.9.76 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/pool-idle-timeout-in-seconds 337

- 3.9.77 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter 338
- 3.9.78 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/resource-adapter-mid 338
- 3.9.79 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config 339
- 3.9.80 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/description 339
- 3.9.81 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/activation-config-property 339
- 3.9.82 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/activation-config-property/activation-config-property-name 340
- 3.9.83 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/activation-config-property/activation-config-property-value 340
- 3.9.84 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint 341
- 3.9.85 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/port-component-name 341
- 3.9.86 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/endpoint-address-uri 342
- 3.9.87 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/login-config 342
- 3.9.88 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/login-config/auth-method 343
- 3.9.89 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/login-config/realm 344
- 3.9.90 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding 345
- 3.9.91 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security 346
- 3.9.92 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message 346
- 3.9.93 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method 346
- 3.9.94 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method/method-name 347
- 3.9.95 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method/method-params 347
- 3.9.96 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method/method-params/method-param 348
- 3.9.97 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/operation-name 348
- 3.9.98 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/request-protection 349
- 3.9.99 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/response-protection 349
- 3.9.100 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/transport-guarantee 350
- 3.9.101 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/service-qname 351
- 3.9.102 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/service-qname/namespaceURI 351

- 3.9.103 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/service-qname/localpart 352
- 3.9.104 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/tie-class 352
- 3.9.105 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/servlet-impl-class 353
- 3.9.106 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/debugging-enabled 353
- 3.9.107 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/property 354
- 3.9.108 /glassfish-ejb-jar/enterprise-beans/message-destination 355
- 3.9.109 /glassfish-ejb-jar/enterprise-beans/message-destination/message-destination-name 355
- 3.9.110 /glassfish-ejb-jar/enterprise-beans/message-destination/jndi-name 355
- 3.9.111 /glassfish-ejb-jar/enterprise-beans/webservice-description 356
- 3.9.112 /glassfish-ejb-jar/enterprise-beans/webservice-description/webservice-description-name 356
- 3.9.113 /glassfish-ejb-jar/enterprise-beans/webservice-description/wsdl-publish-location 357
- 3.9.114 /glassfish-ejb-jar/version-identifier 358
- 3.10 glassfish-resources.xml 359
- 3.11 glassfish-resources.xml に設定する要素の詳細 363
 - 3.11.1 /resources 363
 - 3.11.2 /resources/custom-resource 363
 - 3.11.3 /resources/custom-resource/description 364
 - 3.11.4 /resources/custom-resource/property 364
 - 3.11.5 /resources/external-jndi-resource 365
 - 3.11.6 /resources/external-jndi-resource/description 366
 - 3.11.7 /resources/external-jndi-resource/property 366
 - 3.11.8 /resources/external-jndi-resource/property/description 367
 - 3.11.9 /resources/jdbc-resource 368
 - 3.11.10 /resources/jdbc-resource/description 368
 - 3.11.11 /resources/jdbc-resource/property 369
 - 3.11.12 /resources/jdbc-resource/property/description 369
 - 3.11.13 /resources/mail-resource 370
 - 3.11.14 /resources/mail-resource/description 371
 - 3.11.15 /resources/mail-resource/property 372
 - 3.11.16 /resources/mail-resource/property/description 372
 - 3.11.17 /resources/admin-object-resource 373
 - 3.11.18 /resources/admin-object-resource/description 374
 - 3.11.19 /resources/admin-object-resource/property 374
 - 3.11.20 /resources/admin-object-resource/property/description 375
 - 3.11.21 /resources/connector-resource 375
 - 3.11.22 /resources/connector-resource/description 376
 - 3.11.23 /resources/connector-resource/property 376
 - 3.11.24 /resources/connector-resource/property/description 377

- 3.11.25 /resources/resource-adapter-config 377
- 3.11.26 /resources/resource-adapter-config/property 378
- 3.11.27 /resources/resource-adapter-config/property/description 379
- 3.11.28 /resources/jdbc-connection-pool 379
- 3.11.29 /resources/jdbc-connection-pool/description 385
- 3.11.30 /resources/jdbc-connection-pool/property 385
- 3.11.31 /resources/jdbc-connection-pool/property/description 387
- 3.11.32 /resources/connector-connection-pool 388
- 3.11.33 /resources/connector-connection-pool/description 391
- 3.11.34 /resources/connector-connection-pool/security-map 391
- 3.11.35 /resources/connector-connection-pool/security-map/principal 392
- 3.11.36 /resources/connector-connection-pool/security-map/principal/name 392
- 3.11.37 /resources/connector-connection-pool/security-map/user-group 393
- 3.11.38 /resources/connector-connection-pool/security-map/backend-principal 393
- 3.11.39 /resources/connector-connection-pool/property 394
- 3.11.40 /resources/connector-connection-pool/property/description 397
- 3.11.41 /resources/work-security-map 397
- 3.11.42 /resources/work-security-map/description 398
- 3.11.43 /resources/work-security-map/principal-map 399
- 3.11.44 /resources/work-security-map/group-map 399
- 3.12 glassfish-web.xml 401
- 3.13 glassfish-web.xml に設定する要素の詳細 404
- 3.13.1 /glassfish-web-app 404
- 3.13.2 /glassfish-web-app/context-root 405
- 3.13.3 /glassfish-web-app/security-role-mapping 405
- 3.13.4 /glassfish-web-app/security-role-mapping/role-name 406
- 3.13.5 /glassfish-web-app/security-role-mapping/principal-name 406
- 3.13.6 /glassfish-web-app/security-role-mapping/group-name 407
- 3.13.7 /glassfish-web-app/servlet 407
- 3.13.8 /glassfish-web-app/servlet/servlet-name 408
- 3.13.9 /glassfish-web-app/servlet/principal-name 408
- 3.13.10 /glassfish-web-app/servlet/webservice-endpoint 409
- 3.13.11 /glassfish-web-app/servlet/webservice-endpoint/port-component-name 410
- 3.13.12 /glassfish-web-app/servlet/webservice-endpoint/endpoint-address-uri 410
- 3.13.13 /glassfish-web-app/servlet/webservice-endpoint/login-config 411
- 3.13.14 /glassfish-web-app/servlet/webservice-endpoint/login-config/auth-method 411
- 3.13.15 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding 412
- 3.13.16 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security 413

- 3.13.17 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/message 414
- 3.13.18 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/message/java-method 414
- 3.13.19 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/message/java-method/method-name 414
- 3.13.20 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/message/java-method/method-params 415
- 3.13.21 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/message/java-method/method-params/method-param 415
- 3.13.22 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/message/operation-name 416
- 3.13.23 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/request-protection 416
- 3.13.24 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/
message-security/response-protection 417
- 3.13.25 /glassfish-web-app/servlet/webservice-endpoint/transport-guarantee 418
- 3.13.26 /glassfish-web-app/servlet/webservice-endpoint/service-qname 419
- 3.13.27 /glassfish-web-app/servlet/webservice-endpoint/service-qname/namespaceURI 419
- 3.13.28 /glassfish-web-app/servlet/webservice-endpoint/service-qname/localpart 420
- 3.13.29 /glassfish-web-app/servlet/webservice-endpoint/tie-class 420
- 3.13.30 /glassfish-web-app/servlet/webservice-endpoint/servlet-impl-class 421
- 3.13.31 /glassfish-web-app/servlet/webservice-endpoint/debugging-enabled 421
- 3.13.32 /glassfish-web-app/idempotent-url-pattern 422
- 3.13.33 /glassfish-web-app/session-config 423
- 3.13.34 /glassfish-web-app/session-config/session-manager 424
- 3.13.35 /glassfish-web-app/session-config/session-manager/manager-properties 424
- 3.13.36 /glassfish-web-app/session-config/session-manager/manager-properties/property 424
- 3.13.37 /glassfish-web-app/session-config/session-manager/manager-properties/property/
description 425
- 3.13.38 /glassfish-web-app/session-config/session-properties 426
- 3.13.39 /glassfish-web-app/session-config/session-properties/property 426
- 3.13.40 /glassfish-web-app/session-config/session-properties/property/description 428
- 3.13.41 /glassfish-web-app/session-config/cookie-properties 428
- 3.13.42 /glassfish-web-app/session-config/cookie-properties/property 429
- 3.13.43 /glassfish-web-app/session-config/cookie-properties/property/description 431
- 3.13.44 /glassfish-web-app/ejb-ref 432
- 3.13.45 /glassfish-web-app/ejb-ref/ejb-ref-name 432
- 3.13.46 /glassfish-web-app/ejb-ref/jndi-name 432
- 3.13.47 /glassfish-web-app/resource-ref 433
- 3.13.48 /glassfish-web-app/resource-ref/res-ref-name 433
- 3.13.49 /glassfish-web-app/resource-ref/jndi-name 434

- 3.13.50 /glassfish-web-app/resource-ref/default-resource-principal 434
- 3.13.51 /glassfish-web-app/resource-ref/default-resource-principal/name 435
- 3.13.52 /glassfish-web-app/resource-ref/default-resource-principal/password 435
- 3.13.53 /glassfish-web-app/resource-env-ref 436
- 3.13.54 /glassfish-web-app/resource-env-ref/resource-env-ref-name 436
- 3.13.55 /glassfish-web-app/resource-env-ref/jndi-name 436
- 3.13.56 /glassfish-web-app/service-ref 437
- 3.13.57 /glassfish-web-app/service-ref/service-ref-name 437
- 3.13.58 /glassfish-web-app/service-ref/port-info 438
- 3.13.59 /glassfish-web-app/service-ref/port-info/service-endpoint-interface 438
- 3.13.60 /glassfish-web-app/service-ref/port-info/wsdl-port 439
- 3.13.61 /glassfish-web-app/service-ref/port-info/wsdl-port/namespaceURI 439
- 3.13.62 /glassfish-web-app/service-ref/port-info/wsdl-port/localpart 440
- 3.13.63 /glassfish-web-app/service-ref/port-info/stub-property 440
- 3.13.64 /glassfish-web-app/service-ref/port-info/call-property 441
- 3.13.65 /glassfish-web-app/service-ref/port-info/message-security-binding 441
- 3.13.66 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security 442
- 3.13.67 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message 443
- 3.13.68 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method 443
- 3.13.69 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method/method-name 443
- 3.13.70 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method/method-params 444
- 3.13.71 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method/method-params/method-param 444
- 3.13.72 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/operation-name 445
- 3.13.73 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/request-protection 445
- 3.13.74 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/response-protection 446
- 3.13.75 /glassfish-web-app/service-ref/call-property 447
- 3.13.76 /glassfish-web-app/service-ref/wsdl-override 447
- 3.13.77 /glassfish-web-app/service-ref/service-impl-class 448
- 3.13.78 /glassfish-web-app/service-ref/service-qname 448
- 3.13.79 /glassfish-web-app/service-ref/service-qname/namespaceURI 449
- 3.13.80 /glassfish-web-app/service-ref/service-qname/localpart 449
- 3.13.81 /glassfish-web-app/message-destination-ref 450
- 3.13.82 /glassfish-web-app/message-destination-ref/message-destination-ref-name 450

- 3.13.83 /glassfish-web-app/message-destination-ref/jndi-name 451
- 3.13.84 /glassfish-web-app/class-loader 451
- 3.13.85 /glassfish-web-app/class-loader/property 452
- 3.13.86 /glassfish-web-app/class-loader/property/description 453
- 3.13.87 /glassfish-web-app/jsp-config 454
- 3.13.88 /glassfish-web-app/jsp-config/property 454
- 3.13.89 /glassfish-web-app/jsp-config/property/description 458
- 3.13.90 /glassfish-web-app/property 459
- 3.13.91 /glassfish-web-app/property/description 461
- 3.13.92 /glassfish-web-app/valve 462
- 3.13.93 /glassfish-web-app/valve/description 463
- 3.13.94 /glassfish-web-app/valve/property 463
- 3.13.95 /glassfish-web-app/valve/property/description 464
- 3.13.96 /glassfish-web-app/message-destination 464
- 3.13.97 /glassfish-web-app/message-destination/message-destination-name 465
- 3.13.98 /glassfish-web-app/message-destination/jndi-name 465
- 3.13.99 /glassfish-web-app/webservice-description 466
- 3.13.100 /glassfish-web-app/webservice-description/webservice-description-name 466
- 3.13.101 /glassfish-web-app/webservice-description/wsd-publish-location 466
- 3.13.102 /glassfish-web-app/version-identifier 467
- 3.14 hitachi-application.xml 469
- 3.15 hitachi-application.xml に設定する要素の詳細 470
 - 3.15.1 /hitachi-application 470
- 3.16 hitachi-application-client.xml 471
- 3.17 hitachi-application-client.xml に設定する要素の詳細 472
 - 3.17.1 /hitachi-application-client 472
- 3.18 hitachi-ejb-jar.xml 473
- 3.19 hitachi-ejb-jar.xml に設定する要素の詳細 474
 - 3.19.1 /hitachi-ejb-jar 474
- 3.20 hitachi-ra.xml 475
- 3.21 hitachi-ra.xml に設定する要素の詳細 476
 - 3.21.1 /hitachi-ra 476
- 3.22 hitachi-web.xml 477
- 3.23 hitachi-web.xml に設定する要素の詳細 478
 - 3.23.1 /hitachi-web 478
- 3.24 標準 DD で設定できる独自のパラメーターの詳細 479
 - 3.24.1 jersey.config.server.provider.classnames 479

- 4 Java EE Server で使用するシステムプロパティ 480**
 - 4.1 Java EE Server で使用するシステムプロパティの一覧 481
 - 4.2 ドメイン管理サーバおよび Java EE サーバで使用するシステムプロパティの詳細 484
 - 4.2.1 ASADMIN_LISTENER_PORT 484
 - 4.2.2 com.hitachi.software.javaee.util.prf.PRFID 484
 - 4.2.3 com.sun.appserv.transaction.nofdsync 485
 - 4.2.4 com.sun.corba.ee.ORBVAAMHost 486
 - 4.2.5 com.sun.corba.ee.ORBVAAPort 487
 - 4.2.6 com.sun.enterprise.in-progress-local-transaction.completion-mode 487
 - 4.2.7 com.sun.jts.dblogging.deletequery 488
 - 4.2.8 com.sun.jts.dblogging.insertquery 489
 - 4.2.9 com.sun.jts.dblogging.selectquery 490
 - 4.2.10 com.sun.jts.dblogging.selectservernamequery 490
 - 4.2.11 com.sun.jts.dblogging.use.nontx.connection.for.add 491
 - 4.2.12 com.sun.xml.ws.spi.db.BindingContextFactory 492
 - 4.2.13 com.sun.xml.ws.transport.http.client.HttpTransportPipe.dump 493
 - 4.2.14 com.sun.xml.ws.transport.http.HttpAdapter.dump 494
 - 4.2.15 eclipselink.security.usedoprivileged 495
 - 4.2.16 EXTERNAL-ADDR 496
 - 4.2.17 GMS-BIND-INTERFACE-ADDRESS-*cluster-name* 496
 - 4.2.18 GMS_LISTENER_PORT-*cluster-name* 497
 - 4.2.19 HTTP_LISTENER_PORT 498
 - 4.2.20 HTTP_SSL_LISTENER_PORT 499
 - 4.2.21 IIOP_LISTENER_PORT 499
 - 4.2.22 IIOP_SSL_LISTENER_PORT 500
 - 4.2.23 IIOP_SSL_MUTUALAUTH_PORT 501
 - 4.2.24 java.security.auth.login.config 502
 - 4.2.25 java.security.policy 502
 - 4.2.26 JAVA_DEBUGGER_PORT 503
 - 4.2.27 JMS_PROVIDER_PORT 504
 - 4.2.28 JMX_SYSTEM_CONNECTOR_PORT 505
 - 4.2.29 sun.rmi.dgc.client.gclInterval 505
 - 4.2.30 sun.rmi.dgc.server.gclInterval 506
 - 4.3 アプリケーションクライアントで使用するシステムプロパティの詳細 508
 - 4.3.1 org.glassfish.appclient.acc.maxLoginRetries 508
- 5 Java EE Server で使用する定義ファイルおよびプロパティファイル 509**
 - 5.1 Java EE Server で使用する定義ファイルおよびプロパティファイルの一覧 510
 - 5.2 asenv.bat 515

- 5.3 asenv.bat に設定する環境変数の詳細 516
 - 5.3.1 AS_ADMIN_READTIMEOUT 516
 - 5.3.2 HJES_ASADMIN_CONSOLE_ENABLED 517
 - 5.3.3 HJES_ASADMIN_CONSOLE_FILE_NUMBER 517
 - 5.3.4 HJES_ASADMIN_CONSOLE_ROTATION_SIZE 518
 - 5.3.5 HJES_ASADMIN_CONSOLE_ROTATION_TIME 519
 - 5.3.6 HJES_ASADMIN_JVM_OPTIONS 520
 - 5.3.7 HJES_ASADMIN_LOG_DIVISION 521
 - 5.3.8 HJES_ASADMIN_MAINTENANCE_ENABLED 521
 - 5.3.9 HJES_ASADMIN_MAINTENANCE_FILE_NUMBER 522
 - 5.3.10 HJES_ASADMIN_MAINTENANCE_LOG_LEVEL 523
 - 5.3.11 HJES_ASADMIN_MAINTENANCE_ROTATION_SIZE 524
 - 5.3.12 HJES_ASADMIN_MAINTENANCE_ROTATION_TIME 524
 - 5.3.13 HJES_ASADMIN_MESSAGE_ENABLED 525
 - 5.3.14 HJES_ASADMIN_MESSAGE_FILE_NUMBER 526
 - 5.3.15 HJES_ASADMIN_MESSAGE_LOG_LEVEL 527
 - 5.3.16 HJES_ASADMIN_MESSAGE_ROTATION_SIZE 527
 - 5.3.17 HJES_ASADMIN_MESSAGE_ROTATION_TIME 528
 - 5.3.18 HJES_ASADMIN_STACKTRACE_ENABLED 529
 - 5.3.19 HJES_ASADMIN_STACKTRACE_FILE_NUMBER 530
 - 5.3.20 HJES_ASADMIN_STACKTRACE_ROTATION_SIZE 531
 - 5.3.21 HJES_ASADMIN_STACKTRACE_ROTATION_TIME 531
 - 5.3.22 HJES_LOGSDIR 532
- 5.4 config.properties 534
- 5.5 config.properties に設定するプロパティの詳細 535
 - 5.5.1 imq.authentication.client.response.timeout 535
 - 5.5.2 imq.autocreate.destination.limitBehavior 535
 - 5.5.3 imq.autocreate.destination.maxBytesPerMsg 536
 - 5.5.4 imq.autocreate.destination.maxNumMsgs 537
 - 5.5.5 imq.autocreate.destination.maxTotalMsgBytes 538
 - 5.5.6 imq.autocreate.destination.useDMQ 539
 - 5.5.7 imq.autocreate.queue 539
 - 5.5.8 imq.autocreate.queue.maxNumActiveConsumers 540
 - 5.5.9 imq.autocreate.queue.maxNumBackupConsumers 540
 - 5.5.10 imq.autocreate.topic 541
 - 5.5.11 imq.autocreate.topic.consumerFlowLimit 541
 - 5.5.12 imq.destination.DMQ.truncateBody 542
 - 5.5.13 imq.hostname 542
 - 5.5.14 imq.persist.file.destination.message.filepool.limit 543

5.5.15	imq.persist.file.message.cleanup	543
5.5.16	imq.persist.file.message.filepool.cleanratio	544
5.5.17	imq.persist.file.message.max_record_size	544
5.5.18	imq.persist.file.sync.enabled	545
5.5.19	imq.service.activelist	546
5.5.20	imq.system.max_count	547
5.5.21	imq.system.max_size	547
5.5.22	imq.txn.reapInterval	548
5.5.23	imq.txt.reapLimit	548
5.5.24	java.util.logging.FileHandler.count	549
5.5.25	java.util.logging.FileHandler.level	549
5.5.26	java.util.logging.FileHandler.limit	550
5.6	das.properties	551
5.7	das.properties に設定するプロパティの詳細	552
5.7.1	agent.das.host	552

6 Java EE 標準 API で設定できる Java EE Server 独自のプロパティ 553

6.1	Java EE 標準 API で設定できる Java EE Server 独自のプロパティの一覧	554
6.2	JavaMail で SMTP サーバに接続するためのプロパティの詳細	556
6.2.1	mail.smtp.allow8bitmime	556
6.2.2	mail.smtp.auth	556
6.2.3	mail.smtp.connectiontimeout	557
6.2.4	mail.smtp.ehlo	558
6.2.5	mail.smtp.localhost	558
6.2.6	mail.smtp.sendpartial	559
6.2.7	mail.smtp.ssl.protocols	560
6.2.8	mail.smtp.timeout	561
6.2.9	mail.smtp.writetimeout	561
6.3	JavaMail で IMAP サーバに接続するためのプロパティの詳細	563
6.3.1	mail.imap.connectiontimeout	563
6.3.2	mail.imap.fetchsize	563
6.3.3	mail.imap.peek	564
6.3.4	mail.imap.ssl.protocols	565
6.3.5	mail.imap.statuscachetimeout	565
6.3.6	mail.imap.timeout	566
6.3.7	mail.imap.writetimeout	567

7 JavaVM で使用するオプションとプロパティ 568

7.1	JavaVM で使用するオプションとプロパティの一覧	569
-----	----------------------------	-----

- 7.2 一覧表示で使用する JavaVM オプション 579
 - 7.2.1 -XX:+Hitachi 579
- 7.3 拡張スレッドダンプ機能で使用する JavaVM オプション 582
 - 7.3.1 -XX:[+|-]HitachiThreadDump 582
 - 7.3.2 -XX:[+|-]HitachiThreadDumpToStdout 599
 - 7.3.3 -XX:[+|-]HitachiThreadDumpWithBlockCount 600
 - 7.3.4 -XX:[+|-]HitachiThreadDumpWithCpuTime 601
 - 7.3.5 -XX:[+|-]HitachiThreadDumpWithHashCode 603
- 7.4 JavaVM ログファイルで使用する JavaVM オプション 605
 - 7.4.1 -XX:HitachiJavaLog 605
 - 7.4.2 -XX:HitachiJavaLogFileSize 606
 - 7.4.3 -XX:[+|-]HitachiJavaLogNoMoreOutput 607
 - 7.4.4 -XX:HitachiJavaLogNumberOfFile 608
 - 7.4.5 -XX:[+|-]JavaLogAsynchronous 609
 - 7.4.6 -XX:[+|-]JavaLogHeaderOutput 610
 - 7.4.7 -XX:JavaLogRotationTime 612
 - 7.4.8 -XX:JavaLogRotationType 613
 - 7.4.9 -XX:[+|-]StandardLogToHitachiJavaLog 614
- 7.5 詳細時間出力で使用する JavaVM オプション 616
 - 7.5.1 -XX:[+|-]HitachiOutputMilliTime 616
- 7.6 拡張 verbosegc 機能で使用する JavaVM オプション 618
 - 7.6.1 -XX:[+|-]HitachiCommaVerboseGC 618
 - 7.6.2 -XX:[+|-]HitachiVerboseGC 626
 - 7.6.3 -XX:[+|-]HitachiVerboseGCCpuTime 635
 - 7.6.4 -XX:HitachiVerboseGCIntervalTime 636
 - 7.6.5 -XX:[+|-]HitachiVerboseGCPrintCause 638
 - 7.6.6 -XX:[+|-]HitachiVerboseGCPrintDate 640
 - 7.6.7 -XX:[+|-]HitachiVerboseGCPrintDeleteOnExit 642
 - 7.6.8 -XX:[+|-]HitachiVerboseGCPrintJVMInternalMemory 646
 - 7.6.9 -XX:[+|-]HitachiVerboseGCPrintTenuringDistribution 648
 - 7.6.10 -XX:[+|-]HitachiVerboseGCPrintThreadCount 650
- 7.7 コードキャッシュ領域情報出力機能で使用する JavaVM オプション 653
 - 7.7.1 -XX:CodeCacheInfoPrintRatio 653
 - 7.7.2 -XX:[+|-]PrintCodeCacheFullMessage 654
 - 7.7.3 -XX:[+|-]PrintCodeCacheInfo 656
- 7.8 OutOfMemoryError 発生時の拡張機能で使用する JavaVM オプション 660
 - 7.8.1 -XX:[+|-]HitachiOutOfMemoryAbort 660
 - 7.8.2 -XX:[+|-]HitachiOutOfMemoryAbortThreadDump 661
 - 7.8.3 -XX:[+|-]HitachiOutOfMemoryAbortThreadDumpWithJHeapProf 662

- 7.8.4 -XX:[+|-]HitachiOutOfMemoryCause 663
- 7.8.5 -XX:[+|-]HitachiOutOfMemorySize 664
- 7.8.6 -XX:[+|-]HitachiOutOfMemoryStackTrace 665
- 7.8.7 -XX:HitachiOutOfMemoryStackTraceLineSize 667
- 7.9 クラスライブラリートレース機能で使用する JavaVM オプション 669
- 7.9.1 -XX:[+|-]HitachiJavaClassLibTrace 669
- 7.9.2 -XX:HitachiJavaClassLibTraceLineSize 670
- 7.10 ローカル変数情報出力機能で使用する JavaVM オプション 672
- 7.10.1 -XX:HitachiCallToString 672
- 7.10.2 -XX:[+|-]HitachiLocalsInStackTrace 674
- 7.10.3 -XX:[+|-]HitachiLocalsInThrowable 676
- 7.10.4 -XX:[+|-]HitachiLocalsSimpleFormat 680
- 7.10.5 -XX:[+|-]HitachiTrueTypeInLocals 682
- 7.11 明示管理ヒープ機能で使用する JavaVM オプション 685
- 7.11.1 -XX:ExplicitMemoryAutoReclaimTargetBlockRatio 685
- 7.11.2 -XX:ExplicitMemoryExcludeClassListFile 686
- 7.11.3 -XX:ExplicitMemoryFullGCPolicy 687
- 7.11.4 -XX:ExplicitMemoryNotExcludeClassListFile 688
- 7.11.5 -XX:[+|-]ExplicitMemoryUseExcludeClass 690
- 7.11.6 -XX:[+|-]HitachiAutoExplicitMemory 691
- 7.11.7 -XX:HitachiAutoExplicitMemoryFile 692
- 7.11.8 -XX:HitachiExplicitHeapMaxSize 694
- 7.11.9 -XX:[+|-]HitachiExplicitMemoryAutoReclaim 695
- 7.11.10 -XX:[+|-]HitachiExplicitMemoryAutoRefReclaim 695
- 7.11.11 -XX:[+|-]HitachiExplicitMemoryCompatibleToV8 697
- 7.11.12 -XX:HitachiExplicitMemoryJavaLog 698
- 7.11.13 -XX:HitachiExplicitMemoryJavaLogFileSize 699
- 7.11.14 -XX:HitachiExplicitMemoryJavaLogNumberOfFile 700
- 7.11.15 -XX:HitachiExplicitMemoryLogLevel 701
- 7.11.16 -XX:[+|-]HitachiExplicitMemoryMoveToTenuredFirst 702
- 7.11.17 -XX:[+|-]HitachiUseExplicitMemory 703
- 7.12 リソースの上限値指定で使用する JavaVM オプション 705
- 7.12.1 -XX:HitachiJITCompileMaxMemorySize 705
- 7.12.2 -XX:HitachiThreadLimit 707
- 7.13 JIT コンパイラ稼働継続機能で使用する JavaVM オプション 709
- 7.13.1 -XX:[+|-]JITCompilerContinuation 709
- 7.14 Java アプリケーションプログラムの実行で使用する JavaVM オプション 711
- 7.14.1 -Xhras 711
- 7.15 圧縮オブジェクトポインター機能で使用する JavaVM オプション 713

- 7.15.1 -XX:[+|-]UseCompressedOops 713
- 7.16 ファイナライズ滞留解消機能で使用する JavaVM プロパティ 715
- 7.16.1 JP.co.Hitachi.soft.jvm.autofinalizer 715
- 7.17 ユーザー拡張性能解析トレースで使用する JavaVM プロパティ 716
- 7.17.1 jvm.userprf.Enable 716
- 7.17.2 jvm.userprf.ExtendedSetting 717
- 7.17.3 jvm.userprf.File 718
- 7.17.4 jvm.userprf.Limit 719
- 7.17.5 jvm.userprf.LineNumber 720
- 7.17.6 jvm.userprf.LogLevel 721
- 7.17.7 jvm.userprf.ThrowableName 722
- 7.17.8 jvm.userprf.ThrowableNameEditMethod 723
- 7.17.9 jvm.userprf.Trace 725
- 7.18 Application Server で使用する Java HotSpot VM のオプション 727
- 7.18.1 -agentlib:hndlwrap2 727
- 7.18.2 -agentlib:libname 727
- 7.18.3 -XX:CompressedClassSpaceSize 728
- 7.18.4 -XX:ConcGCThreads 729
- 7.18.5 -XX:MaxGCPauseMillis 730
- 7.18.6 -XX:MaxMetaspaceSize 731
- 7.18.7 -XX:MaxTenuringThreshold 733
- 7.18.8 -XX:MetaspaceSize 733
- 7.18.9 -XX:NewRatio 735
- 7.18.10 -XX:ParallelGCThreads 736
- 7.18.11 -XX:ReservedCodeCacheSize 737
- 7.18.12 -XX:SurvivorRatio 738
- 7.18.13 -XX:TargetSurvivorRatio 739
- 7.18.14 -XX:[+|-]UseG1GC 740
- 7.18.15 -XX:[+|-]UseSerialGC 741
- 7.18.16 -verbose:information-type 742
- 7.18.17 -Xloggc:file 742
- 7.18.18 -Xms 743
- 7.18.19 -Xmx 744
- 7.18.20 -Xprof 745
- 7.18.21 -Xss 745
- 7.19 JavaVM で使用する環境変数 747
- 7.19.1 JAVA_HRAS 747
- 7.19.2 JAVACOREDIR 748
- 7.19.3 JAVAW_HRAS 749

8 JavaVM で使用する定義ファイル 751

8.1 JavaVM で使用する定義ファイルの一覧 752

8.2 exmemexcludeclass.cfg 753

8.3 exmemnotexcludeclass.cfg 756

8.4 userprf.cfg 759

索引 768

1

定義リファレンスの見方

定義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。

1.1 定義リファレンスの記述形式

定義リファレンスの記述形式について説明しています。

記述で使用する項目

定義リファレンスでは、次に示す項目を使用して記述しています。それぞれの説明では、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。

説明

機能について説明しています。

スキーマ

DD ファイルの schemaLocation 属性、または DOCTYPE 宣言について説明しています。

書式

ファイルや項目の書き方について説明しています。

格納先

ファイルの格納先について説明しています。

要素の階層

DD ファイルで指定できる要素の階層について説明しています。

パラメーターの設定対象となるファイル

パラメーターを設定できるファイルについて説明しています。

パラメーターの設定対象となる要素

パラメーターを設定できる要素について説明しています。

指定できる値

指定できる値の型や範囲について説明しています。

初期値

あらかじめシステムで設定されている値について説明しています。

デフォルト値

定義項目や値を設定しない場合に、システムで設定される値について説明しています。

依存関係のある要素または属性

DD ファイルで、ほかの要素や属性と依存関係にあるかどうかについて説明しています。

記述できる場所

ディレクトティブの記述できる場所について説明しています。記述できる場所には、次のものがあります。

- httpsd.conf
VirtualHost ブロック、Directory ブロック以外のhttpsd.conf
- <VirtualHost>

httpsd.conf のVirtualHost ブロック

- <Directory>

httpsd.conf のDirectory ブロック、Location ブロック、Files ブロック

- .htaccess

AccessFileName ディレクティブで指定したアクセス制御ファイル

- <Location>

httpsd.conf のLocation ブロック

また、ディレクティブは次に示す順に参照されます。

- 1.VirtualHost ブロック、Directory ブロック以外のhttpsd.conf

- 2.httpsd.conf のVirtualHost ブロック

- 3.httpsd.conf のDirectory ブロック

- 4.アクセス制御ファイル

- 5.httpsd.conf のFiles ブロック

- 6.httpsd.conf のLocation ブロック

Directory ブロックのAllowOverride ディレクティブの定義（上書き許可レベル）によって、アクセス制御ファイルで定義しているディレクティブを有効または無効にできます。

上書き許可

AllowOverride ディレクティブで上書きを許可する場合の許可レベルについて説明しています。許可レベルには、AuthConfig、FileInfo、Indexes、Limit、Options、All、None があります。各ディレクティブの説明で.htaccess が指定でき、かつ上書き許可レベルの記述がない場合には、許可レベルはAllになります。

出現パターン

繰り返して指定できるかどうかについて説明しています。

指定できる子要素

要素で指定できる子要素について説明しています。各子要素では、指定できる値、デフォルト値、依存関係、出現パターンについて説明しています。

指定できる属性

要素で指定できる属性について説明しています。各属性では、指定できる値、デフォルト値、依存関係、出現パターンについて説明しています。

property 要素の name 属性および value 属性について

property 要素の name 属性および value 属性は、次の形式で指定します。

```
<property name="プロパティ名" value=プロパティの値>
```

プロパティ名およびプロパティの値は、「指定できるプロパティ」の説明、または Java EE の標準仕様を参照してください。

なお、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な値として設定されます。

指定できるプロパティ

要素の属性で指定できるプロパティについて説明しています。各プロパティでは、指定できる値、デフォルト値、依存関係、出現パターンについて説明しています。

記述例または出力例

ファイルや項目の記述例、出力例について説明しています。

注意事項

ファイルや項目を定義する際の注意事項を説明しています。

フォルダーとパスの表記

このマニュアルでは、OS 共通の内容の場合、「フォルダー」を「ディレクトリー」と表記しています。また、「¥」を「/」と表記しています。「ディレクトリー」を「フォルダー」に、「/」を「¥」に置き換えてお読みください。

1.2 定義リファレンスで使用している記号

定義リファレンスで使用している記号の意味について説明しています。

定義リファレンスでは、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。
...	記述が省略されていることを示します。また、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 1) ABC... ABC の後ろに記述があり、その記述が省略されていることを示します。 (例 2) A, B, B, ... 「A と B の後ろに B を必要な個数分繰り返して指定できる」ことを示します。
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) プロパティ 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) ファイル名 ファイル名を指定します。

1.3 定義リファレンスで使用している構文要素

定義リファレンスで使用している構文要素の意味について説明しています。

定義リファレンスでは、次の表に示す構文要素を使用して記述しています。

構文要素	定義
数字	0~9
英数字	A~Z、a~z、0~9

注

すべて半角文字を使用してください。

2

Web サーバで使用する定義ファイルおよび定義項目

Web サーバの動作環境を設定するための定義ファイルおよび定義項目を説明しています。

2.1 Web サーバで使用する定義ファイルおよび定義項目の一覧

Web サーバの動作環境を設定するための定義ファイルおよび定義項目の概要を説明しています。

Web サーバで使用する定義ファイル

定義ファイル	概要	詳細説明
2.2 httpsd.conf	Web サーバの動作環境を設定するファイル	Web サーバの動作環境を各種ディレクティブで定義します。システム管理者が管理します。
2.4 mime.types	コンテンツのファイル拡張子とコンテンツタイプの関連づけを定義するファイル	コンテンツのファイル拡張子とコンテンツタイプ (MIME タイプ) の関連づけを定義します。システム管理者が管理します。
2.5 アクセス制御ファイル	アクセス制御を定義するアクセス制御ファイル	アクセス制御を定義するファイルです。アクセス制御ファイル (.htaccess) を特定のディレクトリ下に作成すれば、そのディレクトリに対するアクセス権を設定できます。

Web サーバで使用する定義項目 (ディレクティブ)

定義項目 (ディレクティブ)	概要	詳細説明
2.3.1 <Directory>	特定のディレクトリに対するディレクティブの定義	特定のディレクトリに対してディレクティブを定義する場合に指定します。
2.3.2 <DirectoryMatch>	正規表現で記述した条件を満たすディレクトリに対するディレクティブの定義	正規表現で記述した条件を満たすディレクトリに対してディレクティブを定義する場合に指定します。
2.3.3 <Files>	特定のファイルに対するディレクティブの定義	特定のファイルに対してディレクティブを定義する場合に指定します。
2.3.4 <FilesMatch>	正規表現で記述した条件を満たすファイルに対するディレクティブの定義	正規表現で記述した条件を満たすファイルに対してディレクティブを定義する場合に指定します。
2.3.5 <IfModule>	モジュールの存在有無に対するディレクティブの定義	指定したモジュールが組み込まれているとき、ブロック内で指定したディレクティブが有効になります。
2.3.6 <Limit>	特定の HTTP プロトコルメソッドに対するディレクティブの定義	特定の HTTP プロトコルメソッドだけに有効なアクセス制御のディレクティブを定義する場合に指定します。
2.3.7 <Location>	特定の URL に対するディレクティブの定義	特定の URL で示す場所へのリクエストについて、ディレクティブを定義する場合に指定します。
2.3.8 <LocationMatch>	正規表現で記述した条件を満たす URL へのリクエストに対するディレクティブの定義	正規表現で記述した条件を満たす URL へのリクエストについてディレクティブを定義する場合に指定します。
2.3.9 <Proxy>	ロードバランサー構成での振り分け先の定義	リバースプロキシを mod_proxy_balancer によるロードバランサー構成で使用する場合に、BalancerMember ディレクティブを定義するために指定します。

定義項目 (ディレクティブ)	概要	詳細説明
2.3.10 <VirtualHost>	特定のホスト名または IP アドレスに対するディレクティブの定義	ホスト名または IP アドレス[:ポート番号]で示すホストへのリクエストについてディレクティブを定義する場合に指定します。
2.3.11 AccessFileName	アクセス制御ファイルのファイル名の定義	アクセス制御のディレクティブを定義しているファイル (アクセス制御ファイル) のファイル名を定義します。
2.3.12 Action	特定の MIME タイプまたはハンドラーで指定したコンテンツに対する CGI スクリプトの実行	MIME タイプまたはハンドラーで指定したコンテンツが Web ブラウザーからリクエストされたとき、実行させるスクリプトを CGI スクリプト名で指定します。
2.3.13 AddAlt	ディレクトリーインデックス表示時に拡張子に関連づけてアイコンの代わりに表示する文字列	ディレクトリーインデックス表示時に、拡張子に指定したファイルと関連づけて文字列を表示する場合に指定します。
2.3.14 AddAltByEncoding	ディレクトリーインデックス表示時に MIME エンコーディングに関連づけてアイコンの代わりに表示する文字列	ディレクトリーインデックス表示時に、アイコンが表示できない環境で、MIME エンコーディング (x-compress など) と関連づけて文字列を表示する場合に指定します。
2.3.15 AddAltByType	ディレクトリーインデックス表示時に MIME タイプに関連づけてアイコンの代わりに表示する文字列	ディレクトリーインデックス表示時に、アイコンが表示できない環境で、MIME タイプ (text/html など) と関連づけて文字列を表示する場合に指定します。
2.3.16 AddCharset	ファイル拡張子に対する文字セットの指定	ファイル拡張子に対する文字セットを指定します。
2.3.17 AddDefaultCharset	ファイル拡張子に対する文字セットのデフォルト値の指定	ファイル拡張子に対する文字セットのデフォルト値を指定します。
2.3.18 AddDescription	ディレクトリーインデックス整形表示時にファイルに対して表示する説明	ディレクトリーインデックス整形表示時に、ファイル名で指定したファイル拡張子、ワイルドカード表記ファイル名またはパス情報なしの完全なファイル名に対して、説明文として文字列を表示する場合に指定します。
2.3.19 AddEncoding	Web サーバ内の圧縮データを Web ブラウザーに表示させるときに必要な拡張子と圧縮形式の関連づけの指定	Web サーバ内の圧縮データを Web ブラウザーに表示させるときに必要な拡張子と圧縮形式の関連づけを指定します。
2.3.20 AddHandler	ハンドラーで処理するファイル拡張子の定義	ハンドラーで処理するファイル拡張子を対応づける場合に定義します。
2.3.21 AddIcon	ディレクトリーインデックスの整形表示時のアイコンを拡張子と対応づけて表示	拡張子などにディレクトリーインデックスのアイコンを対応づけて表示する場合に指定します。
2.3.22 AddIconByEncoding	ディレクトリーインデックスの整形表示時のアイコンを MIME エンコーディングと対応づけて表示	ディレクトリーインデックスの整形表示時のアイコンを MIME エンコーディングと対応づけて表示する場合に指定します。
2.3.23 AddIconByType	ディレクトリーインデックスの整形表示時のアイコンを MIME タイプと対応づけて表示	ディレクトリーインデックスの整形表示時のアイコンを MIME タイプと対応づけて表示する場合に指定します。

定義項目 (ディレクティブ)	概要	詳細説明
2.3.24 AddLanguage	ドキュメントで使用する言語	ドキュメントで使用する言語を指定します。
2.3.25 AddType	TypesConfig ディレクティブで指定したファイルに未定義のコンテンツの拡張子と MIME タイプの関連づけ	TypesConfig ディレクティブで指定したファイルに未定義のコンテンツの拡張子と MIME タイプを関連づけたい場合に指定します。
2.3.26 Alias	URL のファイルシステムのパスへの置き換え	Web ブラウザーからリクエストされた特定の URL を別名に置き換える場合に指定します。
2.3.27 AliasMatch	正規表現を使用して URL のファイルシステムのパスへの置き換え	Web ブラウザーからリクエストされた URL を別名に置き換える場合に指定します
2.3.28 Allow	Web サーバへアクセスできるクライアントの制限	Web サーバへアクセスできるクライアントを制限する場合に指定します。
2.3.29 AllowOverride	AccessFileName ディレクティブで指定したファイルでのアクセス情報定義の上書き許可	AccessFileName ディレクティブで指定したファイルでアクセス情報定義の上書きを許可するかどうかを設定します。
2.3.30 AuthBasicAuthoritative	ユーザー認証する場合の制御方法	ユーザー認証をする場合の制御方法を指定します。
2.3.31 AuthGroupFile	認証するグループのリストを格納しているファイル名の指定	グループでユーザー認証をする場合、認証するグループのリストを格納しているファイル名を指定します
2.3.32 AuthName	ユーザー認証する場合の <i>realm</i> 名	ユーザー認証する場合の <i>realm</i> 名 (Web ブラウザーのユーザー認証画面に表示される) を指定します。
2.3.33 AuthType	ユーザー認証する場合の認証制御のタイプ	ユーザー認証する場合の認証制御のタイプを指定します。
2.3.34 AuthUserFile	認証するユーザー名とパスワードのリストを格納しているファイル名の指定	ユーザー名でユーザー認証をする場合、認証するユーザー名とパスワードのリストを格納しているファイル名を指定します。
2.3.35 BalancerMember	リバースプロキシをロードバランサー構成で使用する場合の転送先の URL の指定	リバースプロキシを <code>mod_proxy_balancer</code> によるロードバランサー構成で使用する場合に転送先の URL を定義します。
2.3.36 BrowserMatch	Web ブラウザーごとの環境変数の設定	Web ブラウザーごとに環境変数を設定する場合に指定します。
2.3.37 BrowserMatchNoCase	Web ブラウザーごとの環境変数の設定	Web ブラウザーごとに環境変数を設定する場合に指定します。
2.3.38 CacheNegotiatedDocs	コンテンツネゴシエーションをするリクエストでクライアント側のキャッシュを有効にするかどうかの指定	コンテンツネゴシエーションをするリクエストで、クライアント側のキャッシュを有効にするかどうかを指定します。

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.39 CustomLog	任意のフォーマットのアクセスログ出力	任意のフォーマットのログをファイルに出力させる場合に指定します。
2.3.40 DefaultIcon	ディレクトリーインデックスで特定のアイコンが設定されていない場合に表示するアイコン	ディレクトリーインデックスで表示するアイコンを指定します。
2.3.41 DefaultLanguage	ドキュメントで使用するデフォルトの言語	ドキュメントで使用するデフォルトの言語を指定します。
2.3.42 Deny	Web サーバへアクセスできるクライアントの制限	Web サーバへアクセスできるクライアントを制限する場合に指定します。
2.3.43 DirectoryIndex	デフォルトとしてクライアントに送信するコンテンツのファイル	Web ブラウザーからのリクエストが特定のコンテンツを指定していない場合に、デフォルトとしてクライアントに送信するコンテンツのファイル名を指定します。
2.3.44 DocumentRoot	コンテンツを格納するドキュメントルートディレクトリーの指定	コンテンツを格納するドキュメントルートディレクトリーを絶対パスで指定します。
2.3.45 ErrorDocument	エラーが発生したときに Web ブラウザーへ表示するメッセージのカスタマイズ	エラーが発生したときに、Web ブラウザーへ表示するメッセージをカスタマイズする場合に指定します。
2.3.46 ErrorLog	エラーログの出力	エラーログを出力するファイル名を指定します。出力するログの内容は、LogLevel ディレクティブで選択できます。
2.3.47 ExpiresActive	有効期限機能に関連するレスポンスヘッダーを追加するかどうかの指定	レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加するかどうかを指定します。
2.3.48 ExpiresByType	MIME タイプのドキュメントに対する有効期限の指定	レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加する場合に、指定する MIME タイプのドキュメントに対する有効期限を指定します。
2.3.49 ExpiresDefault	有効期限のデフォルトの指定	レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加する場合に、デフォルトの有効期限を指定します。
2.3.50 ExtendedStatus	ステータス情報表示時に各リクエストの拡張ステータス情報を表示するかどうかの指定	server-status ハンドラーによるステータス表示形式で、それぞれのリクエストの拡張ステータス情報を表示するかどうかを指定します。
2.3.51 FileETag	ETag レスポンスヘッダーフィールドを作成するために使用されるファイルの属性値	ETag レスポンスヘッダーフィールドを作成するために使用されるファイル属性値を指定します。
2.3.52 ForceType	特定のディレクトリー下のすべてのコンテンツに対して使用する MIME タイプ	<Directory>ブロックまたはアクセス制御ファイルに定義し、特定のディレクトリー下のすべてのコンテンツに対して使用する MIME タイプを指定します。
2.3.53 Header	200 番台のステータスコード応答時のレスポンスヘッダーのカスタマイズ	200 番台のステータスコード応答時のレスポンスヘッダーをカスタマイズする場合に指定します。リバースプロキシとして使用する場合、バックエンドの Web サーバが返すステータスコードの値に関係なく、レスポンスヘッダーをカスタマイズします。

2. Web サーバで使用する定義ファイルおよび定義項目

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.54 HeaderName	ディレクトリーインデックス表示時のヘッダーに付けるコメントを記述したファイルの指定	ディレクトリーインデックス表示時のヘッダーに付けるコメントを記述したファイルのファイル名 (パス情報なし) を指定します。
2.3.55 HostnameLookups	ホスト名のルックアップの逆引きをするかどうかの指定	CGI のREMOTE_HOST 環境変数の IP アドレスおよびログファイルに出力するクライアントの IP アドレスをホスト名に変換するために、ホスト名のルックアップの逆引きをするかどうかを指定します。
2.3.56 HWSErrorDocument METACharset	エラーが発生したときに Web ブラウザーへ表示するメッセージに対応した文字セット	エラーが発生したときに Web ブラウザーへ表示するメッセージ (以降、エラードキュメントと呼びます) についての文字セットを設定します。
2.3.57 HWSErrorLogClient Addr	エラーログに出力するクライアントアドレスを X-Forwarded-For ヘッダー値に変更	バックエンドサーバで、エラーログに出力するメッセージテキストの "[client クライアントアドレス:ポート番号]" を "[X-Forwarded-For X-Forwarded-For ヘッダー値]" に変更するよう設定します。
2.3.58 HWSGracefulStopLog	計画停止時に強制停止させたリクエスト情報をエラーログファイルに出力するかどうかの指定	計画停止時に、強制停止待ち時間を経過したあとに強制停止させたリクエスト情報を、エラーログファイルに出力するかどうかを指定します。
2.3.59 HWSGracefulStopTimeout	計画停止時に実行中のリクエストを強制的に終了させるまでの待ち時間	計画停止時に、実行中のリクエストを直ちに終了するまでの強制停止待ち時間を秒単位で指定します。
2.3.60 HWSImapMenuCharset	メニュー表示に対する文字セット	メニュー表示に対する文字セットを指定します。
2.3.61 HWSLogSSLVerbose	SSL ハンドシェイク処理中にログに出力されるエラーの詳細情報を表示するかどうかの指定	クライアントとサーバ間の SSL ハンドシェイク処理中に、ログに出力されるエラーのうち info レベルおよび error レベルのエラーについて、詳細情報を表示するかどうかを指定します。SSL を有効にする場合には、このディレクトティブを 0n に設定することを推奨します。
2.3.62 HWSLogTimeVerbose	ログに出力する時間をミリ秒まで表示するかどうかの指定	エラーログとリクエストログの時刻、アクセスログのアクセス時刻、リクエスト処理に掛かった時間 (%T)、およびリクエスト処理を開始した時刻 (%t) をミリ秒まで表示するかどうかを指定します。
2.3.63 HWSMaxQueueSize	クライアントからのリクエストについての最大待ちリクエスト数	クライアントからのリクエストについての最大待ちリクエスト数を指定します。
2.3.64 HWSPrfId	PRF デーモン起動時に PRF 識別子に指定した文字列の指定	PRF デーモン起動時に PRF 識別子に指定した文字列を指定します。
2.3.65 HWSPrfSpool	PRF デーモンを使用するときに必要な PRFSPPOOL 環境変数に設定するディレクトリー名の指定	PRF デーモンを使用するときに必要な PRFSPPOOL 環境変数に設定するディレクトリー名を指定します。

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.66 HWSProxyPassReverseCookie	リバースプロキシ使用時にバックエンドサーバから受信した Set-Cookie ヘッダーの変換	リバースプロキシを使用する場合、リバースプロキシはバックエンドサーバから受信した Set-Cookie ヘッダーを変換します。これは、Web ブラウザーが Set-Cookie ヘッダーを受信したあとに、リバースプロキシを経由するバックエンドサーバへのリクエストに対して、Cookie を送信させるために必要になります。
2.3.67 HWSRequestLog	リクエストログの出力	リクエストログを出力するファイル名を指定します。
2.3.68 HWSRequestLogType	リクエストログに出力するトレース種別	HWSRequestLog ディレクティブで設定するリクエストログに出力するトレース種別を指定します。
2.3.69 HWSSetEnvIfIPv6	クライアントまたはサーバの IPv6 アドレスによる環境変数の設定	クライアントまたはサーバの IPv6 アドレスを基に環境変数を定義します。
2.3.70 HWS SuppressModuleTrace	モジュールトレースの出力の抑止	モジュールトレースの出力を抑止するモジュールファイル名および抑止する関数種別を指定します。
2.3.71 HWSTraceIdFile	トレース採取のための共有メモリー ID を格納するファイル名の指定	トレース採取のための共有メモリー ID を格納するファイル名を指定します。
2.3.72 HWSTraceLogFile	サーバプロセスが異常終了した場合に共有メモリーに採取されたトレースを出力するファイル名の指定	サーバプロセスが異常終了した場合に共有メモリーに採取されたトレースを出力するファイル名を指定します。
2.3.73 ImapBase	イメージマップファイルの base 行のデフォルトの指定	イメージマップファイルの base 行のデフォルトを指定します。
2.3.74 ImapDefault	イメージマップファイルの default 行のデフォルトの指定	イメージマップファイルの default 行のデフォルトを指定します。
2.3.75 ImapMenu	メニュー表示についての指定	イメージマップファイルの指定値にmap を与えた場合またはイメージマップ画像の(0,0)座標をマウスでポイントした場合のメニュー表示を指定します。座標指定のない形でイメージマップファイルがリクエストされた場合の動作もこの設定に従います。
2.3.76 Include	設定ファイルとして使用するファイルの指定	ファイル名で指定したファイルを設定ファイルとして利用できるようにします。
2.3.77 IndexIgnore	ディレクトリーインデックス表示時に Web ブラウザーに表示させないファイル名の指定	ディレクトリーインデックス表示時に、Web ブラウザーに表示させないファイル名を指定します。正規表現でも指定できます。
2.3.78 IndexOptions	ディレクトリーインデックスの整形表示機能のオプション	ディレクトリーインデックスの整形表示機能のオプション設定をします。
2.3.79 IndexOrderDefault	ディレクトリーインデックス表示でのデフォルトのファイルの並び順	ディレクトリーインデックス表示での、ファイルの並び順のデフォルトを指定します。

2. Web サーバで使用する定義ファイルおよび定義項目

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.80 KeepAlive	KeepAlive 接続を有効にするかどうかの指定	KeepAlive 接続を有効にするかどうかを指定します。
2.3.81 KeepAliveTimeout	KeepAlive 接続時のリクエスト待ち時間	KeepAlive 接続時の要求待ち時間を秒単位で指定します。
2.3.82 LanguagePriority	クライアントが使用言語の優先度を示さなかった場合の優先順位の指定	使用言語を優先順位の高い順に指定します。
2.3.83 LimitRequestBody	サーバが受信するオブジェクトボディのサイズの上限	HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合のオブジェクトボディ (データ) のサイズの上限を指定します。
2.3.84 LimitRequestFields	サーバが受信する HTTP ヘッダー数の上限	HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合の HTTP ヘッダー数の上限を指定します。
2.3.85 LimitRequestFieldsize	サーバが受信する 1 つの HTTP ヘッダーのサイズの上限	HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合、1 つの HTTP ヘッダーの、サイズの上限を指定します。
2.3.86 LimitRequestLine	サーバが受信するリクエストラインの長さの上限	HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合のリクエストライン (メソッド、問い合わせ文字列などを含む URI、HTTP バージョン) の長さの上限を指定します。
2.3.87 Listen	リクエストを受け付ける IP アドレスおよびポート番号	リクエストを受け付ける IP アドレスおよびポート番号を指定します。
2.3.88 ListenBacklog	クライアントからの接続要求の最大の待ち行列数	クライアントからの接続要求の最大の待ち行列数を指定します。
2.3.89 LoadFile	DSO によって組み込むモジュールが参照するコードがあるオブジェクトファイルまたはライブラリー	DSO によって組み込むモジュールが参照するコードがあるオブジェクトファイルまたはライブラリーを指定します。
2.3.90 LoadModule	Web サーバに動的に組み込むモジュール	Web サーバに動的に組み込むモジュールを指定します。
2.3.91 LogFormat	ログのフォーマットにラベル名を定義	ログのフォーマットにラベル名を定義します。ここで定義したラベル名を CustomLog ディレクティブで指定できます。
2.3.92 LogLevel	エラーログに出力するエラーのレベル	エラーログに出力するエラーのレベルを指定します。指定したレベルの上位レベルのログを出力します。
2.3.93 MaxKeepAliveRequests	KeepAlive 連続接続回数の上限	KeepAlive 連続接続回数の上限を指定します。
2.3.94 MultiviewsMatch	コンテンツネゴシエーションの対象となる拡張子の種類	コンテンツネゴシエーションの対象となる拡張子の種類を指定します。
2.3.95 Options	ユーザーに利用させる機能の制限	ユーザーが利用できる機能を制限する場合に指定します。

2. Web サーバで使用する定義ファイルおよび定義項目

定義項目 (ディレクティブ)	概要	詳細説明
2.3.96 Order	Allow ディレクティブとDeny ディレクティブの指定の評価の順序	Allow ディレクティブとDeny ディレクティブの指定の評価の順序を指定します。
2.3.97 PassEnv	CGI スクリプトに渡す任意の環境変数	CGI スクリプトに渡す任意の環境変数を指定できます。
2.3.98 PidFile	制御プロセス ID を格納するファイル	制御プロセス ID を格納するファイル名を指定します。
2.3.99 ProxyErrorOverride	バックエンドサーバからのレスポンスがエラーのステータスコードであった場合にレスポンスを上書きするかどうかの指定	バックエンドサーバからのステータスコードが 400 番台または 500 番台の場合、レスポンスヘッダーとレスポンスボディをオーバーライドします。その結果、リバースプロキシはバックエンドサーバからのレスポンスではなく、自身が生成したレスポンスをクライアントに返します。
2.3.100 ProxyPass	バックエンドサーバに転送するリクエストと転送するアドレス	リバースプロキシを使用する場合、Web ブラウザーからのリクエストとそれを転送するアドレスを指定します。
2.3.101 ProxyPassReverse	バックエンドサーバから受信した Location ヘッダーの URL の変更	リバースプロキシを使用する場合、バックエンドサーバからのリダイレクトレスポンスの Location ヘッダーで示す URL を変更します。
2.3.102 ProxyPreserveHost	クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送するかどうかの指定	リバースプロキシを使用する場合、クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送するかどうかを指定します。
2.3.103 ProxyTimeout	バックエンドサーバとの送受信に関する待ち時間の指定	バックエンドサーバとの送受信に関する待ち時間を秒単位で指定します。
2.3.104 ProxyVia	Via ヘッダーの使用の制御	このディレクティブはプロキシで Via ヘッダーの使用を制御する場合に指定します。
2.3.105 QOSCookieDomain	流量制限機能に使用する Cookie が有効とされるドメイン	流量制限機能に使用する Cookie が有効とされるドメインを指定します。この値は、HWS 作成モードで使用され、ユーザー作成モードでは使用されません。
2.3.106 QOSCookieExpires	流量制限機能に使用する Cookie の有効時間	流量制限機能に使用する Cookie の有効時間を秒単位で指定します。このディレクティブで指定した値は、HWS 作成モードで使用され、ユーザー作成モードでは使用されません。
2.3.107 QOSCookieName	流量制限機能に使用する Cookie 名	流量制限機能に使用する Cookie 名を指定します。
2.3.108 QOSCookieSecure	SSL によるアクセス時だけに Cookie を送信させるかどうかの指定	クライアントに対し、SSL によるアクセス時だけに Cookie を送信させるよう設定します。この値は、HWS 作成モードで使用され、ユーザー作成モードでは使用されません。
2.3.109 QOSCookieServers	Cookie を送信してきたリクエストだけを処理するときのサーバスレッド数	リクエスト待ち状態のサーバスレッド数が減少した場合に、Cookie を送信してきたリクエストだけを処理するときの、サーバスレッド数を指定します。
2.3.110 QOSRedirect	流量制限機能によって拒否したリクエストのリダイレクト	流量制限機能によって処理が拒否された場合に、クライアントからのリクエストを指定されたパスにリダイレクトさせるときに指定します。

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.111 QOSRejectionServers	受信したすべてのリクエストを拒否するときのサーバプロセス数	リクエスト待ち状態のサーバスレッド数が減少し、受信したすべてのリクエストを拒否するようになるときの、サーバスレッド数を指定します。
2.3.112 QOSResponse	流量制限機能によって処理が拒否された場合にレスポンスとして返すコンテンツ	流量制限機能によって処理が拒否された場合に、ステータスコード 503 とともに返送するコンテンツを指定します。
2.3.113 ReadmeName	ディレクトリーインデックス表示時の Readme として付けるコメントを記述したファイルの指定	ディレクトリーインデックス表示時の Readme として付けるコメントを記述したファイルのファイル名 (パス情報なし) を指定します。
2.3.114 Redirect	クライアントからのリクエストをリダイレクトする場合の指定	旧パスに対するクライアントからのリクエストを、新パスに再リクエスト (リダイレクト) する場合に指定します。
2.3.115 RedirectMatch	正規表現で記述した条件を満たすクライアントからのリクエストをリダイレクトする場合の指定	正規表現で記述した条件を満たすパスに対するクライアントからのリクエストを、新パスに再リクエスト (リダイレクト) する場合に指定します。
2.3.116 RequestHeader	リクエストヘッダーのカスタマイズ	クライアントから受信したヘッダー値をカスタマイズする場合に指定します。
2.3.117 RequestReadTimeout	リクエスト受信開始からリクエストヘッダー受信完了まで、およびリクエストボディの受信開始から完了までのタイムアウト時間の指定	リクエスト受信開始からリクエストヘッダー受信完了まで、およびリクエストボディの受信開始から完了までのタイムアウト時間を秒単位で指定します。
2.3.118 Require	アクセス制限の定義	AuthName ディレクティブ、AuthType ディレクティブ、AuthUserFile ディレクティブ (または AuthGroupFile ディレクティブ) と一緒に指定し、アクセス制限を定義します。
2.3.119 Satisfy	コンテンツへのアクセスがユーザー認証とホスト名または IP アドレスの両方によって制限されている場合のアクセス条件	コンテンツへのアクセスが、ユーザー認証 (AuthUserFile、Require ディレクティブなどを指定) とホスト名または IP アドレス (Allow from、Deny from ディレクティブなどを指定) の両方によって制限されている場合にその関係を設定します。
2.3.120 Script	特定のメソッドに対する CGI スクリプトの実行	指定されたメソッドによるリクエストがあった場合に CGI スクリプト名で示すスクリプトを実行します。
2.3.121 ScriptAlias	CGI プログラムを実行する URL と CGI プログラムが配置されているディレクトリー名の指定	Web ブラウザーから URL で指定された CGI プログラム実行のリクエストに対して、実行する CGI プログラムのあるディレクトリー名を指定します。
2.3.122 ScriptAliasMatch	CGI プログラムを実行する URL に対する正規表現と CGI プログラムが配置されているディレクトリー名の指定	Web ブラウザーから指定された CGI プログラム実行要求の URL が正規表現で記述した条件を満たす場合、指定した新パスの CGI プログラムを実行します。
2.3.123 ScriptInterpreterSource	CGI スクリプトの実行に使用されるインタープリターの指定	CGI スクリプトの実行に使用されるインタープリターを定義します。
2.3.124 ScriptLog	CGI スクリプトのエラーログ出力先	CGI スクリプトのエラーログ出力先のファイルを指定します。

2. Web サーバで使用する定義ファイルおよび定義項目

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.125 ScriptLogBuffer	リクエストのボディー部のログを採取する場合の最大値	PUT、POST メソッドによるリクエストのボディー部のログを採取する場合の最大値をバイト単位で指定します。ScriptLog ディレクティブでエラーログ出力先のファイルを指定した場合だけ、この指定は有効になります。
2.3.126 ScriptLogLength	CGI スクリプトのエラーログファイルの最大サイズ	CGI スクリプトのエラーログファイルの最大サイズをバイト単位で指定します。ScriptLog ディレクティブでエラーログ出力先のファイルを指定した場合だけ指定が有効になります。
2.3.127 SendBufferSize	Web サーバの TCP 送信バッファサイズ	Web サーバの TCP 送信バッファサイズをバイト数で指定します。
2.3.128 ServerAdmin	サーバ管理者の E-Mail アドレス	サーバ管理者の E-Mail アドレスを指定します。ServerSignature ディレクティブで E-Mail を指定する場合は、必ず指定してください。
2.3.129 ServerAlias	サーバ名に基づくバーチャルホストで使用するホスト名の別名	サーバ名に基づくバーチャルホストで使用するホスト名 (ServerName) の別名を指定します。IPv6 アドレスに対応したホスト名も指定できます。
2.3.130 ServerName	Web サーバのサーバ名およびポート番号	Web サーバのサーバ名およびポート番号を指定します。
2.3.131 ServerPath	サーバ名に基づくバーチャルホストで Host ヘッダーの代わりにパス名を利用して各ホストに接続する場合の指定	サーバ名に基づくバーチャルホストで、Host ヘッダーの代わりにパス名を利用して各ホストに接続する場合に指定します。
2.3.132 ServerRoot	サーバのルートディレクトリーの指定	サーバのルートディレクトリーを絶対パスで指定します。
2.3.133 ServerSignature	Web サーバが作成するエラーメッセージなどのコンテンツのフッターに署名するかどうかの指定	Web サーバが作成するエラーメッセージなどのコンテンツのフッターに署名するかどうかを指定します。
2.3.134 ServerTokens	Server ヘッダーのフォーマット	HTTP レスポンスヘッダーの Server ヘッダーのフォーマットを設定します。
2.3.135 SetEnv	CGI スクリプトに渡す環境変数	CGI スクリプトに任意の環境変数を渡す場合に設定する環境変数の値を指定します。このディレクティブを複数指定する場合、同じ環境変数に異なる値は指定できません。
2.3.136 SetEnvIf	クライアントからのリクエストを基に環境変数の定義	クライアントからのリクエスト値が正規表現で表した条件を満たす場合、指定した環境変数を設定します。
2.3.137 SetEnvIfNoCase	クライアントからのリクエストを基に環境変数の定義	クライアントからのリクエスト値が正規表現で表した条件を満たす場合、指定した環境変数を設定します。
2.3.138 SetHandler	ある特定のハンドラーに関連づける場合の指定	指定した<Directory>またはアクセス制御ファイルの範囲すべてのリクエストをハンドラー名で指定したハンドラーに関連づける場合、指定します。

定義項目 (ディレクティブ)	概要	詳細説明
2.3.139 SSLBanCipher	指定した暗号種別でのアクセスを拒否してステータスコード 403 Forbidden を応答する場合の指定	指定した暗号種別でのアクセスを拒否し、クライアントにステータスコード 403 Forbidden を応答します。
2.3.140 SSLCACertificateFile	CA の証明書のファイルの指定	SSL でサーバ認証およびクライアント認証する場合、CA (認証局) の公開鍵 (PEM 形式) のファイル名を指定します。複数の証明書ファイルを連結させて、1 つのファイルに複数の証明書が混在できます。
2.3.141 SSLCertificateFile	SSL で認証する場合の Web サーバの証明書のファイルの指定	SSL で認証する場合、Web サーバの証明書 (PEM 形式) のファイル名を指定します。
2.3.142 SSLCertificateKeyFile	SSL で認証する場合の Web サーバの秘密鍵のファイルの指定	SSL で認証する場合、Web サーバの秘密鍵のファイル名を指定します。
2.3.143 SSLCertificateKeyPassword	パスワード保護されているサーバ秘密鍵のパスワードを格納するファイルの指定	パスワード保護をされているサーバ秘密鍵のパスワードを格納しておくファイルのパス名を指定します。
2.3.144 SSLCRLAuthoritative	CRL の次回発行日を経過した場合の動作の指定	SSL クライアント認証時に使用する CRL の次回発行日の扱いについて指定します。
2.3.145 SSLCRLDERPath	DER 形式の CRL を格納するディレクトリーの指定	DER 形式の CRL を格納するディレクトリーを絶対パスで指定します。指定したディレクトリーに必要な CRL を格納して、Web サーバを起動または再起動すると、SSL でのクライアント認証時に CRL を適用できます。SSLCRLPEMPath ディレクティブで指定したディレクトリー内の CRL も含め、同じ CA から発行された CRL を複数格納している場合、発行日が最新の CRL を適用します。
2.3.146 SSLCRLPEMPath	PEM 形式の CRL を格納するディレクトリーの指定	PEM 形式の CRL を格納するディレクトリーを絶対パスで指定します。指定したディレクトリーに必要な CRL を格納して、Web サーバを起動または再起動すると、SSL でのクライアント認証時に CRL を適用できます。SSLCRLDERPath ディレクティブで指定したディレクトリー内の CRL も含め、同じ CA から発行された CRL を複数格納している場合、発行日が最新の CRL を適用します。
2.3.147 SSLDenySSL	SSL によるアクセスの禁止	SSL によるアクセスを禁止する場合に指定します。このディレクティブが指定されている場合、SSLEnable ディレクティブで SSL を有効にしても https によるアクセスがステータスコード 403 Forbidden で拒否されます。SSLRequireSSL ディレクティブの逆の動作をします。
2.3.148 SSLDisable	SSL を無効にする場合の指定	SSL を無効にします。デフォルト値はSSLEnable ディレクティブ (SSL を有効) です。バーチャルホストで特定のホストに対して SSL を無効にする場合などに指定します。
2.3.149 SSLEnable	SSL を有効にする場合の指定	SSL を有効にします。SSLDisable ディレクティブを指定しないかぎり、デフォルトで有効になります。

2. Web サーバで使用する定義ファイルおよび定義項目

定義項目 (ディレクトティブ)	概要	詳細説明
2.3.150 SSLExportCertChainDepth	環境変数SSL_CLIENT_CERT_CHAIN_nにクライアントの証明書を発行したCAからルートCAまでの証明書を設定するときの指定	SSL クライアント認証をする場合、環境変数SSL_CLIENT_CERT_CHAIN_nにクライアントの証明書を発行したCAからルートCAまでの証明書を設定するとき指定します。
2.3.151 SSLExportClientCertificates	環境変数SSL_CLIENT_CERTにクライアント証明書を設定するときの指定	SSL クライアント認証をする場合、環境変数SSL_CLIENT_CERTにクライアント証明書を設定するとき指定します。環境変数SSL_CLIENT_CERTにはDER形式の証明書をBase64エンコーディングした値を設定します。
2.3.152 SSLFakeBasicAuth	クライアント証明書を利用したBasic認証	SSL クライアント認証の機能とあわせて、WebブラウザでユーザーIDとパスワードを入力することなく、クライアント証明書の提示だけでBasic認証をできるようにします。AuthUserFileディレクティブで指定するファイルにはX509クライアント証明書のSubjectとパスワードを記述します。
2.3.153 SSLProtocol	使用するSSLプロトコルのバージョンの指定	使用するSSLプロトコルのバージョンを指定します。
2.3.154 SSLRequireCipher	指定した暗号種別以外でのアクセスを拒否してステータスコード403Forbiddenを応答する場合の指定	指定した暗号種別以外でのアクセスを拒否し、クライアントにステータスコード403Forbiddenを応答します。
2.3.155 SSLRequiredCiphers	SSL通信で使用できる暗号種別の指定	SSL通信で使用できる暗号種別を指定します。ディレクティブに指定した暗号種別とクライアントが使用できる暗号種別との間で一致するものがあれば、SSL通信が確立されHTTPリクエストを受信します。一致するものがない場合は、SSL通信は確立されないでHTTPリクエストを受信しません。
2.3.156 SSLRequireSSL	SSL以外によるアクセスの禁止	SSL以外によるアクセスを禁止する場合に指定します。このディレクティブが指定されている場合、SSLDisableディレクティブでSSLを無効にしてもhttpによるアクセスがステータスコード403Forbiddenで拒否されます。異なるディレクティブの記述場所で、不用意にSSLを無効にしコンテンツを公開してしまうことを防止します。
2.3.157 SSLSessionCacheSize	セッションキャッシュサイズの上限值	セッションキャッシュサイズの上限值をバイト単位で指定します。0を指定した場合、セッションキャッシュは実施されません。
2.3.158 SSLSessionCacheTimeout	SSLセッションの有効時間	Webサーバ内またはSSLセッション管理キャッシュサーバgcache内で保持されるセッションIDなどのデータの有効時間を秒単位で指定します。
2.3.159 SSLVerifyClient	クライアント認証時の証明書に関する設定	クライアント認証時の証明書に関する設定を指定します。
2.3.160 SSLVerifyDepth	証明書のチェーンを何段階までたどるかの指定	証明書のチェーンを何段階までたどるかを指定します。
2.3.161 ThreadsPerChild	サーバとして起動するスレッド数	サーバとして起動するスレッド数を指定します。指定したスレッド数はサーバの最大同時接続数を示します。

定義項目 (ディレクティブ)	概要	詳細説明
2.3.162 Timeout	Web サーバの送受信の待ち時間	Web サーバの送受信の待ち時間を秒単位で指定します。
2.3.163 TraceEnable	TRACE メソッドによるリクエストを拒否するかどうかの指定	TRACE メソッドによるリクエストを拒否するかどうかを指定します。
2.3.164 TransferLog	ラベル名を指定しないLogFormat ディレクティブのフォーマットでのアクセスログ出力	<p>ログを格納するファイルまたはログを出力するプログラムを指定します。ログのフォーマットはラベル名を指定しないLogFormat ディレクティブで指定できます。</p> <p>LogFormat ディレクティブでログのフォーマットを指定する場合は、IPv6 アドレスや IPv6 アドレスに対応したホスト名も出力できます。</p> <p>LogFormat ディレクティブでフォーマットを指定しない場合は、標準のログフォーマットで出力します。</p>
2.3.165 TypesConfig	ファイル拡張子とコンテンツタイプ の関係を定義する設定ファイルの指定	ファイル拡張子とコンテンツタイプ (MIME タイプ) の関係を定義する設定ファイルを指定します。
2.3.166 UnsetEnv	CGI スクリプトに渡す環境変数から SetEnv ディレクティブまたは PassEnv ディレクティブで指定した環境変数を削除する場合の指定	CGI スクリプトに渡す環境変数から、SetEnv ディレクティブまたはPassEnv ディレクティブで指定した環境変数を削除する場合に指定します。
2.3.167 UseCanonicalName	サーバの正式な名前の生成方法の指定	サーバの正式な名前の生成方法を指定します。サーバの正式な名前は、自サーバを参照する URL や環境変数のSERVER_NAME とSERVER_PORT に設定されます。なお、On、Off、dns すべての場合で、IPv6 アドレスに対応しています。
2.3.168 UserDir	Web ブラウザーからの/ <code>ユーザー名</code> /へのリクエストに対して公開するサーバ上の場所の指定	Web ブラウザーからの/ <code>ユーザー名</code> /へのリクエストに対して公開するサーバ上の場所をディレクトリー名で指定します。

Web サーバで使用する定義項目 (ユーティリティー)

定義項目 (ユーティリティー)	概要	詳細説明
2.6.1 rotatelog	時間によるログの分割	アクセスログやエラーログを一定時間単位に分割して、複数のファイルに出力できます。
2.6.2 rotatelog2	ファイルサイズによるログの分割	アクセスログやエラーログをログファイルサイズで分割して、複数のファイルにラップアラウンドして出力できます。

2.2 httpsd.conf

Web サーバの動作環境を設定するファイルです。

説明

Web サーバの動作環境を各種ディレクティブで定義します。システム管理者が管理します。

ファイルに指定できる定義およびディレクティブを次に示します。

設定内容	ディレクティブ名	複数指定
httpsd.conf ファイル内のブロックの定義	<Directory>	○
	<DirectoryMatch>	○
	<Files>	○
	<FilesMatch>	○
	<IfModule>	○
	<Limit>	○
	<Location>	○
	<LocationMatch>	○
	<Proxy>	○
	<VirtualHost>	○
サーバの基本的な定義	ServerName※1、※2	×
	ServerAdmin	×
	ServerRoot	×
	ServerSignature	×
	Listen	○
	LoadModule	○
	LoadFile	○
	Include	○
	ExtendedStatus	×
	ServerTokens	×
	FileETag	○
コンテンツを管理するための定義	UserDir	○
	DocumentRoot	×
	ErrorDocument	○

設定内容	ディレクティブ名	複数指定
Web ブラウザーからのリクエストについての定義 (Alias)	Alias	○
	AliasMatch	○
	Redirect	○
	RedirectMatch	○
MIME タイプについての定義	TypesConfig	×
	AddCharset	○
	AddDefaultCharset	×
	AddType	○
	ForceType	×
	HTTPErrorDocument	×
コンテンツネゴシエーションについての定義	LanguagePriority	○
	AddEncoding	○
	AddLanguage	○
	DefaultLanguage	×
	CacheNegotiatedDocs	×
	MultiviewsMatch	×
ハンドラーについての定義	AddHandler	○
	SetHandler	×
Web サーバの性能についての定義	Timeout	×
	RequestReadTimeout	×
	ListenBacklog	×
	ThreadsPerChild	×
	HTTPMaxQueueSize	×
	SendBufferSize	×
KeepAlive の定義	KeepAlive	×
	MaxKeepAliveRequests	×
	KeepAliveTimeout	×
リクエストを制限する定義	LimitRequestBody	×
	LimitRequestFields	×
	LimitRequestFieldSize	×
	LimitRequestLine	×

設定内容	ディレクティブ名	複数指定
CGI、環境変数の定義	ScriptAlias	○
	ScriptAliasMatch	○
	UseCanonicalName	×
	BrowserMatch	○
	BrowserMatchNoCase	○
	PassEnv	○
	SetEnv	○
	UnsetEnv	○
	SetEnvIf	○
	SetEnvIfNoCase	○
	Action	○
	Script	○
	ScriptInterpreterSource	×
	HWSetEnvIfIPv6	○
ディレクトリーインデックスの表示内容の定義	DirectoryIndex	×
	AddIconByEncoding	○
	AddIconByType	○
	AddIcon	○
	DefaultIcon	×
	ReadmeName	×
	HeaderName	×
	IndexIgnore	○
	IndexOrderDefault	×
	AddAltByEncoding	○
	AddAltByType	○
	AddAlt	○
	AddDescription	○
	IndexOptions	○
Web サーバへのアクセスを制御する定義	AccessFileName	×
	AllowOverride	×
	AuthName	×

設定内容	ディレクティブ名	複数指定
	AuthType	×
	AuthGroupFile	×
	AuthUserFile	×
	AuthBasicAuthoritative	×
	Require	○
	Options	×
	Order	×
	Allow from	○
	Deny from	○
	Satisfy	×
	TraceEnable	×
SSL による暗号化および認証の定義	SSLRequireSSL	×
	SSLEnable	×
	SSLDisable ^{※1}	×
	SSLCertificateFile ^{※2}	×
	SSLCertificateKeyFile ^{※2}	×
	SSLCACertificateFile	×
	SSLVerifyClient	×
	SSLVerifyDepth	×
	SSLRequiredCiphers	×
	SSLRequireCipher	○
	SSLBanCipher	○
	SSLDenySSL	×
	SSLFakeBasicAuth	×
	SSLSessionCacheTimeout	×
	SSLSessionCacheSize	×
	SSLCRLAuthoritative	×
	SSLCRLDERPath	×
	SSLCRLPEMPath	×
	SSLExportCertChainDepth	×
SSLExportClientCertificates	×	

設定内容	ディレクティブ名	複数指定
	SSLCertificateKeyPassword	×
	SSLProtocol	×
Web サーバを運用形態に合わせて複数ホストに見せる定義	ServerAlias	○
	ServerPath	×
イメージマップファイルについての定義	ImapDefault	×
	ImapBase	×
	ImapMenu	×
	HWSImapMenuCharset	×
採取するログの定義	HostnameLookups	×
	ErrorLog	×
	LogLevel	×
	LogFormat	○
	CustomLog	○
	TransferLog	○
	PidFile	×
	ScriptLog	×
	ScriptLogBuffer	×
	ScriptLogLength	×
	HWSLogSSLVerbose	×
	HWSLogTimeVerbose	×
	HWSPrfId	×
	HWSPrfSpool	×
	HWSRequestLog	×
	HWSRequestLogType	×
	HWSSuppressModuleTrace	○
HWSErrorLogClientAddr	×	
採取するトレースの定義	HWSTraceIdFile	×
	HWSTraceLogFile	×
リバースプロキシについての定義	ProxyPass	○
	ProxyPassReverse	○
	ProxyVia	×

設定内容	ディレクティブ名	複数指定
	ProxyErrorOverride	×
	ProxyPreserveHost	×
	ProxyTimeout	×
	HWSProxyPassReverseCookie	○
	BalancerMember	○
流量制限機能についての定義	QOSCookieDomain	×
	QOSCookieExpires	×
	QOSCookieName	○
	QOSCookieSecure	×
	QOSCookieServers	×
	QOSRedirect	○
	QOSRejectionServers	×
	QOSResponse	×
ヘッダーカスタマイズ機能についての定義	Header	○
	RequestHeader	○
有効期限設定機能についての定義	ExpiresActive	×
	ExpiresByType	○
	ExpiresDefault	×
計画停止についての定義	HWSGracefulStopLog	×
	HWSGracefulStopTimeout	×

(凡例)

- ：指定できる。
- ×：指定できない。

注※1

Web サーバを起動するために、最低限設定が必要なディレクティブです (SSL を利用しない場合)。

注※2

Web サーバを起動するために、最低限設定が必要なディレクティブです (SSL を利用する場合)。

ディレクティブによっては、記述できる場所が制限されているものがあります。各ディレクティブの記述できる場所については、各ディレクティブの「記述できる場所」に示します。また、ディレクティブの上書きを許可する場合は、AllowOverride ディレクティブで許可レベルを定義する必要があります。各ディレクティブの上書き許可のレベルは、各ディレクティブの「上書きレベル」に示します。

各ディレクティブの「記述できる場所」および「上書き許可」に記載される内容を次に示します。

「記述できる場所」で示される内容

各ディレクティブの記述できる場所は、次のような形式で記載しています。

記述できる場所	説明
httpsd.conf	VirtualHost ブロック、Directory ブロック以外のhttpsd.conf ファイル
<VirtualHost>	httpsd.conf ファイルのVirtualHost ブロック
<Directory>	httpsd.conf ファイルのDirectory ブロック、Location ブロック、Files ブロック
.htaccess	AccessFileName ディレクティブで指定したアクセス制御ファイル
<Location>	httpsd.conf ファイルのLocation ブロック

また、ディレクティブは次に示す順に参照されます。

- 1.VirtualHost ブロック、Directory ブロック以外のhttpsd.conf ファイル
- 2.httpsd.conf ファイルのVirtualHost ブロック
- 3.httpsd.conf ファイルのDirectory ブロック
- 4.アクセス制御ファイル
- 5.httpsd.conf ファイルのFiles ブロック
- 6.httpsd.conf ファイルのLocation ブロック

Directory ブロックのAllowOverride ディレクティブの定義（上書き許可レベル）によって、アクセス制御ファイルで定義しているディレクティブを有効または無効にできます。

「上書き許可」で示される内容

AllowOverride ディレクティブで上書きを許可する場合に、定義する許可レベルを記載しています。許可レベルの内容は次のとおりです。

- AuthConfig レベル
サーバへのアクセス制御関連のディレクティブの上書きを許可します。サーバへのアクセス制御関連のディレクティブには、AuthGroupFile、AuthName、AuthType、AuthUserFile、Require ディレクティブなどが該当します。
- FileInfo レベル
コンテンツ管理、MIME タイプ、暗号化などファイル情報関連のディレクティブの上書きを許可します。ファイル情報関連のディレクティブには、AddType、AddEncoding、AddLanguage ディレクティブなどが該当します。
- Indexes レベル
ディレクトリーインデックス関連のディレクティブの上書きを許可します。ディレクトリーインデックス関連のディレクティブには、FancyIndexing、AddIcon、AddDescription ディレクティブなどが該当します。
- Limit レベル

ホスト名または IP アドレスを用いたアクセス制御のディレクティブの上書きを許可します。ホスト名または IP アドレスを用いたアクセス制御のディレクティブには、Allow from、Deny from、Order ディレクティブが該当します。

- Options レベル
Options ディレクティブの上書きを許可します。
- All レベル
すべての上書きを許可します。
- None レベル
すべての上書きを禁止します。

なお、各ディレクティブの説明で.htaccess が指定でき、かつ上書き許可レベルの記述がない場合には、許可レベルはAllになります。

書式

ディレクティブの記述規則を次に示します。

正規表現

ディレクティブの指定に使用できる正規表現を次に示します。

記号	機能	使用例	使用例の意味
.	任意の 1 文字。	a...c	a のあとに任意の 3 文字と c が続く。abcdc は適合する。
*	直前の 1 文字の 0 個以上の繰り返し。	ab*cd*	ac、abbbbc、abbbbcd は適合する。
+	直前の 1 文字の 1 個以上の繰り返し。	ab*c+	abbbc は適合する。abbb は適合しない。
?	直前の 1 文字があるかないか。	abbbc?	abbbc、abbb は適合する。
	選択肢の区切り。	a bc d	a、bc または d。
¥	直後の特殊文字 (. ^\$*+? ¥[](){}) の 1 文字。ただし、¥を表す場合は¥¥¥。	¥. ¥¥¥	. と適合する。 1 文字の ¥ と適合する。
^	行の先頭に適合する。	^ab	abcde は適合する。
\$	行の末尾に適合する。	abc\$	aaabc は適合する。
{m}	直前の正規表現の m 個の繰り返し。	a{5}	aaaaa が適合する。
{m,}	直前の正規表現の m 個以上の繰り返し。	a{3,}	aaa、aaaa は適合する。aa は適合しない。
{m,n}	直前の正規表現の m 個以上 n 個以下の繰り返し。	a{3,5}	aaa、aaaa、aaaaa が適合する。aa、aaaaaa は適合しない。
[文字列]	文字列にある任意の 1 文字。*	[abc]*または [a-c]*	aaa、bbb、ccc、cba、aab は適合する。
[^文字列]	文字列にない任意の 1 文字。	[^0-9]	数字以外の 1 字が適合する。

記号	機能	使用例	使用例の意味
(文字列)	文字列をグループ化する。	(ab)+	ababab が適合する。ababb は適合しない。
		aa(xx yy)bb	aaxxbb、aayybb が適合する。

注※

次の3文字は、[文字列]で特殊な意味を持ちます。

^: [の次に指定して、文字列に含まれないものを示すために用います。]

]: 文字列の最後を示すために用います。

-: 範囲を指定するために用います。

また、これら特殊文字の前の¥は省略されます。

[文字列]で特殊な意味を持つ文字を通常の文字として指定するには、次のようにします。なお、次の4つ以外の特殊文字は、通常の文字として扱われます。

^: 文字列の先頭以外で指定します。(例) [ab^yz]

]: 文字列の先頭に指定します。(例) []abxy]

-: 最後に指定します。(例) [abxy-]

¥: ¥¥¥と指定します。(例) [¥¥¥abxy]

ディレクティブに指定するパス情報

ディレクトリー名、ファイル名またはパス名を指定するディレクティブの場合、ディレクティブの種類によって、指定できるパス情報が異なります。

パスの種類には、次のものがあります。各ディレクティブのパス情報は、各ディレクティブで説明します。

- 絶対パスしか指定できません (ドライブ名を付けた指定も絶対パスに含まれます)。
- ServerRoot ディレクティブの指定値からの相対パスで指定できます (ただし、ServerRoot ディレクティブの指定が先に必要)。

また、パス情報にネットワーク上のディレクトリーやファイルを指定することはできません。ネットワークを使用したファイルシステム上のディレクトリーやファイルを指定することもできません。

コメント行

設定ファイル中、行の最初に#を付けると、コメント行になります。ただし、ディレクティブを指定したあとに#から始まる文字列を記述しても、#以降をコメントとして扱いません。

コメント行を指定する場合の記述例を次に示します。

- 正しい例

```
#Deny from all
```

#行はコメント行として扱われます。

- 誤った例

```
Deny from all #comment
```

#comment はディレクティブ指定値として扱われます。コメントとしては扱われません。

格納先

Application Serverのインストールディレクトリー¥httpsd¥conf¥httpsd.conf

記述例

記述例のインストール先は、C:/Hitachi/APServer/httpsd ディレクトリーです。

```
#####  
##  
## httpsd.conf - Hitachi Web Server configuration file  
##  
## All Rights Reserved. Copyright (C) 2000, 2014, Hitachi, Ltd.  
#####  
  
Listen 80  
HWSMaxQueueSize 8192  
ThreadsPerChild 50  
Timeout 60  
KeepAlive On  
MaxKeepAliveRequests 100  
KeepAliveTimeout 3  
HostnameLookups Off  
  
ServerRoot "C:/Hitachi/APServer/httpsd"  
  
ServerName www.example.com  
DocumentRoot "C:/Hitachi/APServer/httpsd/htdocs"  
#ScriptAlias /cgi-bin/ "C:/Hitachi/APServer/httpsd/cgi-bin/"  
DirectoryIndex index.html  
UseCanonicalName Off  
ServerSignature Off  
ServerTokens ProductOnly  
TraceEnable Off  
  
LogLevel info  
ErrorLog "|¥"¥"C:/Hitachi/APServer/httpsd/sbin/rotatelog2.exe¥"  
    ¥"C:/Hitachi/APServer/httpsd/logs/error¥" 8192 5¥"  
#ErrorLog logs/error.log  
HWSRequestLog "|¥"¥"C:/Hitachi/APServer/httpsd/sbin/rotatelog2.exe¥"  
    ¥"C:/Hitachi/APServer/httpsd/logs/hwsrequest¥" 86400 -fnum 8 -diff 540¥"  
#HWSRequestLog logs/hwsrequest.log  
LogFormat "%h %l %u %t ¥"¥r¥" %>s %b ¥"%{Referer}i¥" ¥"%{User-Agent}i¥" %I %O"  
    combinedio  
LogFormat "%h %l %u %t ¥"¥r¥" %>s %b ¥"%{Referer}i¥" ¥"%{User-Agent}i¥"  
    combined  
LogFormat "%h %l %u %t ¥"¥r¥" %>s %b" common  
LogFormat "%{Referer}i -> %U" referer  
LogFormat "%{User-agent}i" agent  
LogFormat "%h %l %u %t ¥"¥r¥" %>s %b %{hws_thread_id}P %{hws_ap_root}n  
    %I %O %X %D ¥"%{Referer}i¥" ¥"%{User-Agent}i¥" hws_trace  
LogFormat "%h %l %u %t ¥"¥r¥" %>s %b %T %{hws_thread_id}P %{hws_ap_root}n"  
    hws_std  
HWSLogTimeVerbose On
```

```

CustomLog "|¥¥"C:/Hitachi/APServer/httpsd/sbin/rotatelog.exe¥¥"
¥¥"C:/Hitachi/APServer/httpsd/logs/access¥¥" 86400 -fnum 8 -diff 540¥¥" hws_std
#CustomLog logs/access.log hws_std
PidFile logs/httpd.pid
HWSTraceIdFile logs/hws.trcid
HWSTraceLogFile logs/hws.trclog

SSLDisable
#SSLEnable
#SSLCertificateFile "C:/Hitachi/APServer/httpsd/conf/ssl/server/httpsd.pem"
#SSLCertificateKeyFile
"C:/Hitachi/APServer/httpsd/conf/ssl/server/httpsdkey.pem"
#SSLCertificateKeyPassword
"C:/Hitachi/APServer/httpsd/conf/ssl/server/.keypasswd"
#SSLCACertificateFile "C:/Hitachi/APServer/httpsd/conf/ssl/cacert/anycert.pem"
#SSLProtocol SSLv3 TLSv1
#SSLRequiredCiphers RC4-MD5:RC4-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA
#SSLVerifyClient 0
#SSLVerifyDepth 3
#SSLSessionCacheTimeout 300
#HWSLogSSLVerbose On

TypesConfig conf/mime.types
AddEncoding x-compress .Z
AddEncoding x-gzip .gz .tgz
AddLanguage ca .ca
AddLanguage cs .cz .cs
AddLanguage da .dk
AddLanguage de .de
AddLanguage el .el
AddLanguage en .en
AddLanguage eo .eo
AddLanguage es .es
AddLanguage et .et
AddLanguage fr .fr
AddLanguage he .he
AddLanguage hr .hr
AddLanguage it .it
AddLanguage ja .ja
AddLanguage ko .ko
AddLanguage ltz .ltz
AddLanguage nl .nl
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pl .po
AddLanguage pt .pt
AddLanguage pt-BR .pt-br
AddLanguage ru .ru
AddLanguage sv .sv
AddLanguage tr .tr
AddLanguage zh-CN .zh-cn
AddLanguage zh-TW .zh-tw
#LanguagePriority ja en ca cs da de el eo es et fr he hr it ko ltz nl nn no pl
pt pt-BR ru sv tr zh-CN zh-TW

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4¥.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4¥.0" force-response-1.0

```

```

BrowserMatch "Java/1¥.0" force-response-1.0
BrowserMatch "JDK/1¥.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
  redirect-carefully
BrowserMatch "MS FrontPage" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[01234]" redirect-carefully
BrowserMatch "^gnome-vfs/1.0" redirect-carefully
BrowserMatch "^XML Spy" redirect-carefully
BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully
BrowserMatch "Konqueror/4" redirect-carefully

```

```

Alias /icons/ "C:/Hitachi/APServer/httpsd/icons/"
IndexOptions FancyIndexing
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrm .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
DefaultIcon /icons/unknown.gif
ReadmeName README.html
HeaderName HEADER.html
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

```

```

<Directory />
  Options None
  AllowOverride None
#  <Limit PUT DELETE>
#      Order deny,allow
#      Deny from all
#  </Limit>
#  SSLRequireSSL
</Directory>

<Directory "C:/Hitachi/APServer/httpsd/htdocs">
  Options None
  AllowOverride None
</Directory>

```

```
<FilesMatch "^.*(ht|key)">
    Order allow,deny
    Deny from all
</FilesMatch>

#<Location /server-status>
#     SetHandler server-status
#     Order deny,allow
#     Deny from all
#     Allow from example.com
#</Location>

#Include "C:/Hitachi/APServer/httpsd/conf/reverse_proxy.conf"
#Include "C:/Hitachi/APServer/httpsd/conf/proxy_balancer.conf"
```

注意事項

ディレクティブに IPv6 アドレスを記述する場合は、「**[IPv6アドレス]**」のように IPv6 アドレスを [] で囲んで指定してください。また、ディレクティブに IPv6 アドレスとポート番号を併記する場合は、「**[IPv6アドレス]:ポート番号**」のように IPv6 アドレスを [] で囲み、「:」の後ろにポート番号を指定します。

ただし、次のディレクティブに IPv6 アドレスを記述する場合は、IPv6 アドレスを [] で囲まないで指定してください。

- Allow from ディレクティブ
- Deny from ディレクティブ
- HWSSetEnvIfIPv6 ディレクティブ

IPv6 アドレスを指定する場合は、グローバルユニキャストアドレスを指定してください。

2.3 httpsd.conf に設定する定義項目の詳細

httpsd.conf に設定する定義項目について説明しています。

2.3.1 <Directory>

特定のディレクトリーに対してディレクティブを定義する場合に指定します。

説明

特定のディレクトリーに対してディレクティブを定義する場合に指定します。ディレクトリー名にディレクトリー名を指定し、そのディレクトリーとサブディレクトリーだけに有効なディレクティブを定義するブロックを指定できます。

ディレクトリー名は、絶対パスで指定してください。

書式

```
<Directory ディレクトリー名> ディレクティブ [ディレクティブ ...] </Directory>
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
<Directory /> ...1.
  Options None ...2.
  AllowOverride None ...3.
</Directory> ...4.

<Directory "Application Serverのインストールディレクトリー/httpsd/htdocs"> ...5.
  Options Indexes ...6.
  AllowOverride None ...7.
  Order allow,deny ...8.
  Allow from all ...9.
</Directory> ...10.
```

1. ルートディレクトリーの定義
2. 機能はすべて無効
3. すべての上書き禁止
4. 定義終わり
5. Application Serverのインストールディレクトリー/httpsd/htdocs ディレクトリーの定義
6. ディレクトリーインデックス表示可

7. すべての上書き禁止
8. Allow ディレクティブの指定をDeny ディレクティブの指定より先に評価
9. すべてのホストからのアクセスを許可
10. 定義終わり

2.3.2 <DirectoryMatch>

正規表現で記述した条件を満たすディレクトリーに対してディレクティブを定義する場合に指定します。

説明

正規表現で記述した条件を満たすディレクトリーに対してディレクティブを定義する場合に指定します。ディレクトリー名を正規表現で指定し、そのディレクトリーとサブディレクトリーだけに有効なディレクティブを定義するブロックを指定できます。

正規表現のディレクトリー名は、絶対パスで指定してください。

書式

```
<DirectoryMatch 正規表現> . . . </DirectoryMatch>
```

記述できる場所

httpsd.conf、<VirtualHost>

2.3.3 <Files>

特定のファイルに対してディレクティブを定義する場合に指定します。

説明

特定のファイルに対してディレクティブを定義する場合に指定します。ファイル名にファイル名を指定し、そのファイルだけに有効なディレクティブを定義するブロックを指定できます。

書式

```
<Files ファイル名> ディレクティブ [ディレクティブ ...] </Files>
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

2.3.4 <FilesMatch>

正規表現で記述した条件を満たすファイルに対してディレクティブを定義する場合に指定します。

説明

正規表現で記述した条件を満たすファイルに対してディレクティブを定義する場合に指定します。ファイル名を正規表現で指定し、そのファイルだけに有効なディレクティブを定義するブロックを指定できます。

書式

```
<FilesMatch 正規表現> ディレクティブ [ディレクティブ ...] </FilesMatch>
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

2.3.5 <IfModule>

モジュールの存在有無に対してディレクティブを定義する場合に指定します。

説明

指定したモジュールが組み込まれているとき、ブロック内で指定したディレクティブが有効になります。モジュール名の前に!を付けた場合は、指定したモジュールが組み込まれていないとき、ブロック内で指定したディレクティブが有効になります。ブロック内で指定可能なディレクティブに制限はありません。

モジュール名にLoadModule ディレクティブで組み込まれるモジュールを指定する場合、<IfModule>ディレクティブよりも前に、LoadModule ディレクティブを指定する必要があります。

書式

```
<IfModule !モジュール名> ディレクティブ [ディレクティブ ...] </IfModule>
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

2.3.6 <Limit>

特定の HTTP プロトコルメソッドに対してディレクティブを定義する場合に指定します。

説明

特定の HTTP プロトコルメソッドだけに有効なアクセス制御のディレクティブを定義する場合に指定します。メソッド名は複数指定できます。

指定できるメソッド名

GET、POST、PUT、DELETE、CONNECT、OPTIONS
(HEAD は GET に含まれています)

ブロック内に指定できるディレクティブ

- Allow from
- Deny from
- AuthName
- AuthType
- AuthUserFile
- AuthGroupFile
- Order
- Require
- Satisfy

書式

```
<Limit メソッド名 [メソッド名 ...]> ディレクティブ [ディレクティブ ...] </Limit>
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

記述例

```
<Directory />  
  <Limit PUT DELETE> ...1.  
    Order deny,allow ...2.  
    Deny from all ...3.  
    Allow from .your_domain.com ...4.  
  </Limit> ...5.  
</Directory>
```

1. PUT および DELETE メソッドに対する定義
2. Deny ディレクティブの指定を、Allow ディレクティブの指定よりも先に評価
3. すべてのホストからの PUT および DELETE メソッドによるアクセスは不可
4. your_domain.com からの PUT および DELETE メソッドによるアクセスを許可

5. 定義終わり

2.3.7 <Location>

特定の URL に対してディレクティブを定義する場合に指定します。

説明

特定の URL で示す場所へのリクエストについて、ディレクティブを定義する場合に指定します。ただし、URL に、?以降（問い合わせ文字列）は指定できません。

書式

```
<Location URL> ディレクティブ [ディレクティブ ...] </Location>
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
<Location /server-status>          ...1.
    SetHandler server-status        ...2.
    Order deny,allow                ...3.
    Deny from all                   ...4.
    Allow from .your_domain.com     ...5.
</Location>                        ...6.
```

1. URL /server-status の定義
2. このディレクトリーのリクエストは server-status ハンドラーに関連づける
3. Deny ディレクティブの指定を Allow ディレクティブの指定よりも先に評価
4. すべてのホストからのアクセスは不可
5. your_domain.com からのアクセスを許可
6. 定義終わり

2.3.8 <LocationMatch>

正規表現で記述した条件を満たす URL へのリクエストに対してディレクティブを定義する場合に指定します。

説明

正規表現で記述した条件を満たす URL へのリクエストについてディレクティブを定義する場合に指定します。ただし、URL に、?以降（問い合わせ文字列）は指定できません。

書式

```
<LocationMatch 正規表現> ディレクティブ [ディレクティブ ...] </LocationMatch>
```

記述できる場所

httpsd.conf、<VirtualHost>

2.3.9 <Proxy>

ロードバランサー構成での振り分け先を定義する場合に指定します。

説明

リバースプロキシをmod_proxy_balancer によるロードバランサー構成で使用する場合に、BalancerMember ディレクティブを定義するために指定します。

URL には、ProxyPass ディレクティブで指定したbalancer://で始まる URL と同じ値を指定してください。

❗ 重要

このディレクティブを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so
- mod_proxy_balancer モジュール
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
- mod_lbmethod_byrequests モジュール
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
- mod_slotmem_shm モジュール
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so

書式

```
<Proxy URL> ディレクティブ [ディレクティブ ...] </Proxy>
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
ProxyPass /examples/ balancer://cluster1/  
<Proxy balancer://cluster1/>  
    BalancerMember http://IPアドレスA/examples/  
    BalancerMember http://IPアドレスB/examples/  
</Proxy>
```

2.3.10 <VirtualHost>

特定のホスト名または IP アドレスに対してディレクティブを定義する場合に指定します。

説明

ホスト名または *IPアドレス[:ポート番号]* で示すホストへのリクエストについてディレクティブを定義する場合に指定します。

なお、IPv6 アドレスに対応したホスト名も指定できます。IP アドレスに IPv6 アドレスを指定する場合は、IPv6 アドレスを [] で囲んでください。

書式

```
<VirtualHost {ホスト名|IPアドレス[:ポート番号]}  
  [{ホスト名|IPアドレス[:ポート番号]} ... ]>  
  ディレクティブ [ディレクティブ ...] </VirtualHost>
```

記述できる場所

httpsd.conf

記述例

```
<VirtualHost 172.17.40.30:80>  
  :  
</VirtualHost>  
<VirtualHost [2001::123:4567:89ab:cdef]:80>  
  :  
</VirtualHost>
```

2.3.11 AccessFileName

アクセス制御のディレクティブを定義しているファイル（アクセス制御ファイル）のファイル名を定義します。

説明

アクセス制御のディレクティブを定義しているファイル（アクセス制御ファイル）のファイル名を定義します。AllowOverride ディレクティブで許可されていれば、コンテンツリクエスト時に毎回このファイルを参照しアクセス制限がチェックされます。

書式

```
AccessFileName ファイル名 [ファイル名 ...]
```

デフォルト値

定義項目の省略

.htaccess

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
AccessFileName .htaccess
```

アクセス制御ファイルのファイル名は、.htaccess です。

2.3.12 Action

特定の MIME タイプまたはハンドラーで指定したコンテンツに対して CGI スクリプトを実行する場合に指定します。

説明

MIME タイプまたはハンドラーで指定したコンテンツが Web ブラウザーからリクエストされたとき、実行させるスクリプトを CGI スクリプト名で指定します。CGI スクリプト名は、URL で指定します。このディレクティブを複数指定する場合、同じ MIME タイプに異なる CGI スクリプトは指定できません。

書式

```
Action {MIMEタイプ|ハンドラー} CGIスクリプト名
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
Action image/gif /cgi-bin/images.cgi
```

2.3.13 AddAlt

ディレクトリーインデックス表示時に、拡張子に関連づけてアイコンの代わりに表示する文字列を指定します。

説明

ディレクトリーインデックス表示時に、*拡張子*に指定したファイルと関連づけて文字列を表示する場合に指定します。1つの文字列に対して複数の拡張子が指定できます。テキストモードの Web ブラウザーのようにアイコン表示ができない環境で、ファイルの属性を表示する場合などに利用できます。

*拡張子*に指定できるものを次に示します。

- ファイル拡張子
- ワイルドカード表記のファイル拡張子またはファイル名
- ファイル名

このディレクティブを複数指定する場合、同じ拡張子に異なる文字列は指定できません。

書式

```
AddAlt 文字列 拡張子 [拡張子 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
AddAlt "HTML" htm html
```

拡張子がhtmまたはhtmlのファイルの場合、文字列"HTML"を表示します。

2.3.14 AddAltByEncoding

ディレクトリーインデックス表示時に、MIME エンコーディングに関連づけてアイコンの代わりに表示する文字列を指定します。

説明

ディレクトリーインデックス表示時に、アイコンが表示できない環境で、*MIME* エンコーディング (x-compress など) と関連づけて文字列を表示する場合に指定します。1つの文字列に対して複数の*MIME* エンコーディングが指定できます。このディレクティブを複数指定する場合、同じMIMEタイプに異なる文字列は指定できません。

書式

```
AddAltByEncoding 文字列 MIMEエンコーディング [MIMEエンコーディング ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
AddAltByEncoding "gzip" x-gzip
```

2.3.15 AddAltByType

ディレクトリーインデックス表示時に、MIME タイプに関連づけてアイコンの代わりに表示する文字列を指定します。

説明

ディレクトリーインデックス表示時に、アイコンが表示できない環境で、*MIME* タイプ (text/html など) と関連づけて文字列を表示する場合に指定します。1つの文字列に対して複数の *MIME* タイプが指定できます。このディレクティブを複数指定する場合、同じ *MIME* タイプに異なる文字列は指定できません。

書式

```
AddAltByType 文字列 MIMEタイプ [MIMEタイプ ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
AddAltByType "plain text" text/plain
```

2.3.16 AddCharset

ファイル拡張子に対する文字セットを指定します。

説明

ファイル拡張子に対する文字セットを指定します。文字セットは Content-Type ヘッダーに charset= の値として設定されます。クライアントに対して文字セットを明示する場合に使用します。このディレクティブを複数指定する場合、同じ拡張子に異なる文字列は指定できません。指定する拡張子は、AddType ディレクティブまたは TypesConfig ディレクティブで指定したファイルで、MIME タイプの関連づけが必要です。

書式

```
AddCharset 文字セット 拡張子 [拡張子 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
AddCharset EUC-JP .euc
AddCharset ISO-2022-JP .jis
AddCharset SHIFT_JIS .sjis
```

2.3.17 AddDefaultCharset

ファイル拡張子に対する文字セットのデフォルト値を指定します。

説明

ファイル拡張子に対する文字セットのデフォルト値を指定します。AddCharset ディレクティブの設定に対するデフォルト値となります。Content-Type がtext/plain、text/html の場合のデフォルトとして設定されます。

書式

```
AddDefaultCharset [On|Off|文字セット]
```

指定できる値

On

デフォルト文字セットとして ISO-8859-1 を設定します。

Off

文字セットを設定しません。

文字セット

指定した文字セットをデフォルト文字セットとします。

デフォルト値

定義項目の省略

Off

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
AddDefaultCharset ISO-2022-JP
```

2.3.18 AddDescription

ディレクトリーインデックスの整形表示時に、ファイルに対して表示する説明を指定します。

説明

ディレクトリーインデックス整形表示時に、ファイル名で指定したファイル拡張子、ワイルドカード表記ファイル名またはパス情報なしの完全なファイル名に対して、説明文として文字列を表示する場合に指定します。なお、ファイル名にスラッシュで終わる文字列を指定した場合、*が付けられワイルドカード指定と同様に見なされます。

ファイル名に指定できるものを次に示します。

- ファイル拡張子
- ワイルドカード表記のファイル名
- ファイル名

このディレクティブを複数指定する場合、同じファイル名に異なる文字列は指定できません。

書式

```
AddDescription "文字列" ファイル名 [ファイル名 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
AddDescription "The planet Mars" /web/pics/mars.gif
```

2.3.19 AddEncoding

Web サーバ内の圧縮データを Web ブラウザーに表示させるときに必要な拡張子と圧縮形式の関連づけを指定します。

説明

Web サーバ内の圧縮データを Web ブラウザーに表示させるときに必要な拡張子と圧縮形式の関連づけを指定します。Web ブラウザーに圧縮ファイルの展開の情報として Content-Encoding ヘッダーを Web サーバから送信する場合に設定します。このヘッダーを利用した運用は、Web ブラウザー側の実装に依存します。このディレクティブを複数指定する場合、同じ拡張子に異なる圧縮形式は指定できません。

書式

```
AddEncoding 圧縮形式 拡張子
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
AddEncoding x-compress Z      ...1.  
AddEncoding x-gzip gz        ...2.
```

1. 拡張子がZ のファイルの圧縮形式は x-compress
2. 拡張子がgz のファイルの圧縮形式は x-gzip

2.3.20 AddHandler

ハンドラーで処理するファイル拡張子を対応づける場合に定義します。

説明

ハンドラーで処理するファイル拡張子を対応づける場合に定義します。

指定できるハンドラー名を次に示します。このディレクティブを複数指定する場合、同じ拡張子に異なるハンドラー名は指定できません。

cgi-script

CGI スクリプトの実行

imap-file

イメージマップ処理

server-status

ステータスの表示

書式

```
AddHandler ハンドラー名 拡張子 [拡張子 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
AddHandler cgi-script .cgi      ...1.  
AddHandler imap-file map        ...2.
```

1. 拡張子.cgi はcgi-script ハンドラー
2. 拡張子map はimap-file ハンドラー

2.3.21 AddIcon

ディレクトリーインデックスの整形表示時のアイコンを拡張子と対応づけて表示する場合に指定します。

説明

拡張子などにディレクトリーインデックスのアイコンを対応づけて表示する場合に指定します。文字列には画像表示ができない Web ブラウザーの場合に表示する文字を指定します。URL にアイコンの画像ファイルの URL を指定します。自ホスト内の画像ファイルを指定する場合、URL の「http://IPアドレス」は省略できます。なお、URL の「http://IPアドレス」を省略しないで IPv6 アドレスを指定する場合は、IPv6 アドレスを[]で囲んでください。

拡張子として指定できるものを次に示します。

- ファイル拡張子

- ワイルドカード表記のファイル拡張子またはファイル名
- ファイル名

拡張子として`^^DIRECTORY^^`を記述すると、ディレクトリーに対するアイコンを設定できます。また、`^^BLANKICON^^`と指定すると、ディレクトリーインデックスを表示した場合の、表示内容のヘッダーのインデントを合わせるためのアイコンを設定できます。

このディレクティブを複数指定する場合、同じ拡張子に異なる文字列や URL は指定できません。

書式

```
AddIcon {(文字列、URL)|URL} 拡張子 [拡張子 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

拡張子が .tar の場合のアイコン定義

```
AddIcon /icons/tar.gif .tar
```

拡張子が .html、.shtml、.htm、.pdf の場合のアイコン定義

```
AddIcon /icons/layout.gif .html .shtml .htm .pdf
```

拡張子が .txt の場合のアイコン定義

```
AddIcon /icons/text.gif .txt
```

親ディレクトリーのアイコン定義

```
AddIcon /icons/back.gif ..
```

README ファイルのアイコン定義

```
AddIcon /icons/hand.right.gif README
```

ディレクトリーの場合のアイコン定義

```
AddIcon /icons/folder.gif ^^DIRECTORY^^
```

ディレクトリーインデックスのヘッダーのインデントアイコン定義

```
AddIcon /icons/blank.gif ^^BLANKICON^^
```

```
AddIcon http://[2001::123:4567:89ab:cdef]/icons/text.gif .txt
```

2.3.22 AddIconByEncoding

ディレクトリーインデックスの整形表示時のアイコンを MIME エンコーディングと対応づけて表示する場合に指定します。

説明

ディレクトリーインデックスの整形表示時のアイコンを *MIME* エンコーディングと対応づけて表示する場合に指定します。文字列には画像表示ができない Web ブラウザーの場合に表示する文字を指定します。*URL* にアイコンの画像ファイルの *URL* を指定します。自ホスト内の画像ファイルを指定する場合、*URL* の「*http://IPアドレス*」は省略できます。なお、*URL* の「*http://IPアドレス*」を省略しないで IPv6 アドレスを指定する場合は、IPv6 アドレスを [] で囲んでください。

このディレクティブを複数指定する場合、同じ MIME タイプに異なる文字列や *URL* は指定できません。

書式

```
AddIconByEncoding {(文字列,URL)|URL} MIMEエンコーディング [MIMEエンコーディング ...]
```

記述できる場所

httpd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

MIME エンコーディング *x-compress* および *x-gzip* の場合のアイコン定義

```
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
```

2.3.23 AddIconByType

ディレクトリーインデックスの整形表示時のアイコンを MIME タイプと対応づけて表示する場合に指定します。

説明

ディレクトリーインデックスの整形表示時のアイコンを MIME タイプと対応づけて表示する場合に指定します。画像表示ができない Web ブラウザーの場合に表示する文字は文字列で指定できます。また、*URL* で表示するアイコンの画像ファイル名の場所を指定できます。なお、*URL* の「*http://IPアドレス*」を省略しないで IPv6 アドレスを指定する場合は、IPv6 アドレスを [] で囲んでください。

このディレクティブを複数指定する場合、同じ MIME タイプに異なるファイル名は指定できません。

書式

```
AddIconByType {(文字列, URL)|URL} MIMEタイプ [MIMEタイプ ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

MIME タイプ *text/** の場合のアイコン定義

```
AddIconByType (TXT, /icons/text.gif) text/*
```

MIME タイプ *image/** の場合のアイコン定義

```
AddIconByType (IMG, /icons/image2.gif) image/*
```

MIME タイプ *audio/** の場合のアイコン定義

```
AddIconByType (SND, /icons/sound2.gif) audio/*
```

MIME タイプ *video/** の場合のアイコン定義

```
AddIconByType (VID, /icons/movie.gif) video/*
```

2.3.24 AddLanguage

ドキュメントで使用する言語を指定します。

説明

ドキュメントで使用する言語を指定します。言語コードは Content-Language レスポンスヘッダーに設定されます。このディレクティブを指定すると、Web ブラウザーの言語設定で言語コードの優先順位

(Accept-Language ヘッダー) がリクエストに設定されている場合、Web サーバから送信するコンテンツを選択するコンテンツネゴシエーションができます。言語コードは Web ブラウザーが送信するヘッダー情報に依存します。基本的には、ISO639 に定義されている言語コードに従って指定します。なお、コンテンツネゴシエーションを有効にするためには、Options ディレクティブで MultiViews オプションを設定しなければなりません。このディレクティブを複数指定する場合、同じ拡張子に異なる言語コードは指定できません。

書式

```
AddLanguage 言語コード 拡張子
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

AddLanguage ja .ja	日本語
AddLanguage en .en	英語
AddLanguage fr .fr	フランス語
AddLanguage de .de	ドイツ語
AddLanguage da .da	デンマーク語
AddLanguage el .el	ギリシャ語
AddLanguage it .it	イタリア語

2.3.25 AddType

TypesConfig ディレクティブで指定したファイルに未定義のコンテンツの拡張子と MIME タイプを関連づけたい場合に指定します。

説明

TypesConfig ディレクティブで指定したファイルに未定義のコンテンツの拡張子と MIME タイプを関連づけたい場合に指定します。このディレクティブを複数指定する場合、同じ拡張子に異なる MIME タイプは指定できません。

書式

```
AddType MIMEタイプ 拡張子 [拡張子 ...]
```

記述できる場所

httpd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
AddType text/html .shtml
```

MIME タイプ `text/html` と拡張子 `.shtml` を関連づけます。

2.3.26 Alias

ファイルシステムのパスへ URL を置き換える場合に指定します。

説明

Web ブラウザーからリクエストされた特定の URL を別名に置き換える場合に指定します。ただし、URL には、?以降（問い合わせ文字列）を指定できません。URL で指定されたディレクトリーを、ディレクトリー名で指定したディレクトリーに置き換えて、Web ブラウザーに表示します。

次のディレクティブ指定値と重複する URL は指定できません。

- *ProxyPass* のパス名

例えば、次のような指定はできません。

```
Alias /aaa/bbb/ C:/alias/  
ProxyPass /aaa/ http://aaa.example.com/
```

ディレクトリー名は、絶対パスで指定してください。

書式

```
Alias URL ディレクトリー名
```

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
Alias /icons/ "Application Serverのインストールディレクトリー/httpsd/icons/"
```

/icons/をApplication Serverのインストールディレクトリー/httpsd/icons/に置き換えます。

2.3.27 AliasMatch

正規表現を使用してファイルシステムのパスへ URL を置き換える場合に指定します。

説明

Web ブラウザーからリクエストされた URL を別名に置き換える場合に指定します。ただし、URL には、?以降（問い合わせ文字列）を指定できません。

正規表現で記述した条件を満たす URL が Web ブラウザーからリクエストされた場合、指定した新パスのコンテンツを Web ブラウザーに表示します。正規表現で括弧 () を使用してグループ化している場合、その i 番目のグループの表現にマッチした文字列を、新パスで \$i を使用して参照できます。i には 1 から 9 までの数字を指定します。

次のディレクティブ指定値と重複する正規表現は指定できません。

- ProxyPass のパス名

例えば、次のような指定はできません。

```
AliasMatch ^/aaa/bbb/(.*) C:/alias/$1  
ProxyPass /aaa/ http://aaa.example.com/
```

新パスは、絶対パスで指定してください。また、新パスの文字として、'\$'または'&'を含める場合は、その文字の前に'¥'を付加してください。なお、\$i を指定する際には、'\$'の前に'¥'を付加する必要はありません。

書式

```
AliasMatch 正規表現 新パス
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
AliasMatch ^/html/(.*) "C:/htdocs/html/$1"
```

"/html/"で始まるリクエストのとき、/html/部分をC:/htdocs/html/に置き換えます。例えば、/html/index.html へのアクセスの場合、C:/htdocs/html/index.html に置き換えます。

2.3.28 Allow

Web サーバへアクセスできるクライアントを制限する場合に指定します。

説明

Web サーバへアクセスできるクライアントを制限する場合に指定します。ホストにはアクセスを許可するホストのドメイン名、IP アドレス、サブネット、ネットマスクを指定できます。すべてのホストからのアクセスを許可する場合はallを指定します。

また、ホストには、IPv6 アドレスに関するドメイン名、アドレスおよびプリフィクス長も指定できます。IPv6 アドレスを指定する場合は、IPv6 アドレスを[]で囲まないでください。プリフィクス長は、「IPv6 アドレス/プリフィクス長」の形式で指定します。プリフィクス長は 10 進数で指定してください。

env=環境変数を指定すると、サーバへのアクセスを環境変数で制御できます。BrowserMatch、BrowserMatchNoCase、SetEnvIf、SetEnvIfNoCase ディレクティブとあわせて使用すれば、HTTP リクエストヘッダーフィールドに基づいてアクセスを制限できます。

Allow ディレクティブ（アクセス許可）とDeny ディレクティブ（アクセス制限）は、Order ディレクティブで評価の順序を設定できます。

ホスト	意味
ドメイン名	ドメイン名で指定したホストからのアクセスを許可する。
IP アドレス	IP アドレスで指定したホストからのアクセスを許可する。
サブネット	サブネット（IP アドレスの最初の 1 から 3 バイト）で指定したホストからのアクセスを許可する。
ネットマスク	ネットマスク表記（例:10.1.0.0/255.255.0.0）で指定したホストからのアクセスを許可する。 10.1.0.0/16 形式で表記した場合、10.1.0.0/255.255.0.0 と同じ意味である。

書式

```
Allow from {ホスト|all|env=環境変数} [{ホスト|env=環境変数} ...]
```

記述できる場所

<Directory>、.htaccess

上書き許可

Limit レベル

記述例

(例 1)

```
SetEnvIf User-Agent Mozilla.* access_ok
<Directory /docroot>
  Order deny,allow
  Deny from all
  Allow from env=access_ok
</Directory>
```

この場合、User-Agent の文字列がMozilla を含むブラウザからのリクエストだけがアクセスを許可されて、ほかのアクセスは拒否されます。

(例 2)

ホストに IPv6 アドレスを指定する場合は、次のように指定します。

```
allow from 2001::123:4567:89ab:cdef
```

また、プリフィクス長を指定するとき、次の指定はどれも同じ意味となります。

```
allow from 2001:0:0:89ab::/64
allow from 2001:0:0:89AB::/64
allow from 2001::89ab:0:0:0:0/64
allow from 2001:0000:0000:89ab:0000:0000:0000:0000/64
```

2.3.29 AllowOverride

AccessFileName ディレクティブで指定したファイルでアクセス情報定義の上書きを許可するかどうかを設定します。

説明

AccessFileName ディレクティブで指定したファイルでアクセス情報定義の上書きを許可するかどうかを設定します。各指示子によって制御できるディレクティブは、各ディレクティブの上書き許可の記述を参照してください。

書式

```
AllowOverride 指示子 [指示子 ...]
```

指定できる値

AuthConfig

サーバへのアクセス制御関連のディレクティブの上書きを許可します。サーバへのアクセス制御関連のディレクティブには、AuthGroupFile、AuthName、AuthType、AuthUserFile、Require ディレクティブなどが該当します。

FileInfo

コンテンツ管理、MIME タイプ、暗号化などファイル情報関連のディレクティブの上書きを許可します。ファイル情報関連のディレクティブには、AddType、AddEncoding、AddLanguage ディレクティブなどが該当します。

Indexes

ディレクトリーインデックス関連のディレクティブの上書きを許可します。ディレクトリーインデックス関連のディレクティブには、FancyIndexing、AddIcon、AddDescription ディレクティブなどが該当します。

Limit

ホスト名または IP アドレスを用いたアクセス制御のディレクティブの上書きを許可します。ホスト名または IP アドレスを用いたアクセス制御のディレクティブには、Allow from、Deny from、Order ディレクティブが該当します。

Options

Options ディレクティブの上書きを許可します。

All

すべての上書きを許可します。

None

すべての上書きを禁止します。

デフォルト値

定義項目の省略

None

記述できる場所

<Directory>

2.3.30 AuthBasicAuthoritative

ユーザー認証をする場合の制御方法を指定します。

説明

ユーザー認証をする場合の制御方法を指定します。

書式

```
AuthBasicAuthoritative {0n|0ff}
```

指定できる値

0n

AuthUserFile、AuthGroupFile、Require ディレクティブの設定によるユーザー認証をします。ユーザー未登録またはパスワード不整合の場合は 401 エラーステータスを Web ブラウザーに表示します。

0ff

AuthUserFile、AuthGroupFile、Require ディレクティブの設定によるユーザー認証をします。そのとき、パスワード不整合の場合は 401 エラーステータスを Web ブラウザーに表示します。さらに、ユーザー未登録の場合には他製品のモジュール（機能）でユーザー認証をします。

デフォルト値

定義項目の省略

0n

記述できる場所

<Directory>、.htaccess

上書き許可

AuthConfig レベル

2.3.31 AuthGroupFile

グループでユーザー認証をする場合、認証するグループのリストを格納しているファイル名を指定します。

説明

グループでユーザー認証をする場合、認証するグループのリストを格納しているファイル名を指定します。ファイル名には、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が指定できます。

グループファイルはテキストエディターを使用して次に示すようなフォーマットで作成してください。

```
グループ名 : ユーザー名 [ ユーザー名 ... ]
```


任意のグループ名に、ユーザー認証のためのパスワードファイルに登録しているユーザー名を定義します。1行につき1グループで指定します。グループファイルには複数グループを定義できます。同じグループ名の行を複数指定した場合には、同じグループ名に登録されているすべてのユーザー名を含んだ1つのグループが定義されます。

書式

```
AuthGroupFile ファイル名
```

記述できる場所

<Directory>、.htaccess

上書き許可

AuthConfig レベル

2.3.32 AuthName

ユーザー認証する場合の realm 名（Web ブラウザーのユーザー認証画面に表示される）を指定します。

説明

ユーザー認証する場合の *realm* 名（Web ブラウザーのユーザー認証画面に表示される）を指定します。このディレクティブを指定する場合はAuthType、Require、AuthUserFile（またはAuthGroupFile）ディレクティブを必ず指定しなければなりません。ただし、ディレクトリーサービスを利用したユーザー認証を行う場合は、AuthUserFile（またはAuthGroupFile）ディレクティブの指定は必要ありません。

書式

```
AuthName realm名
```

記述できる場所

<Directory>、.htaccess

上書き許可

AuthConfig レベル

2.3.33 AuthType

ユーザー認証する場合の認証制御のタイプを指定します。

説明

ユーザー認証する場合の認証制御のタイプを指定します。認証タイプ名として"Basic"が指定できます。このディレクティブを指定する場合はAuthName、Require、AuthUserFile（またはAuthGroupFile）ディレクティブを必ず指定しなければなりません。ただし、ディレクトリーサービスを利用したユーザー認証を行う場合は、AuthUserFile（またはAuthGroupFile）ディレクティブの指定は必要ありません。

書式

AuthType 認証タイプ名

指定できる値

Basic

Base64 コード変換をします。

記述できる場所

<Directory>、.htaccess

上書き許可

AuthConfig レベル

2.3.34 AuthUserFile

ユーザー名でユーザー認証をする場合、認証するユーザー名とパスワードのリストを格納しているファイル名を指定します。

説明

ユーザー名でユーザー認証をする場合、認証するユーザー名とパスワードのリストを格納しているファイル名を指定します。

ファイル名には、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が指定できます。

書式

AuthUserFile ファイル名

記述できる場所

<Directory>、.htaccess

上書き許可

AuthConfig レベル

2.3.35 BalancerMember

リバースプロキシをロードバランサー構成で使用する場合に転送先の URL を指定します。

説明

リバースプロキシをmod_proxy_balancer によるロードバランサー構成で使用する場合に転送先の URL を定義します。

書式

```
BalancerMember URL [キー=値 [キー=値 ...]]
```

指定できる値

URL

転送先の URL を「http://IPアドレス」または「ホスト名[:ポート番号]/」を含む形で指定します。

URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

キー

指定できるキーを次に示します。

キー	意味
loadfactor	リクエストの負荷分散値を指定します。それぞれの転送先に振り分ける比率を、1 から100 の整数で指定します。 キーが省略された場合のデフォルト値は1 です。
route	振り分け先の識別に使用する任意の 64 バイト未満の文字列を指定します。 このキーを指定する場合は、ProxyPass ディレクティブにstickysession キーの指定も必要です。
timeout	次のバックエンドサーバとの送受信時の待ち時間を指定します。 <ul style="list-style-type: none">バックエンドサーバへのリクエスト送信中にデータを送信できなくなった場合の待ち時間バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間 指定できる値は1-65535 の範囲です。キーが省略された場合のデフォルト値は、ProxyTimeout ディレクティブ指定値です。
connectiontimeout	バックエンドサーバとの接続時の待ち時間を指定します。 指定できる値は1-65535 の範囲です。キーが省略された場合のデフォルト値は、timeout キー値です。

記述できる場所

<Proxy>

2.3.36 BrowserMatch

Web ブラウザーごとに環境変数を設定する場合に指定します。

説明

Web ブラウザーごとに環境変数を設定する場合に指定します。設定する値のデフォルト値は1です。環境変数の前に!が付いたときは、その環境変数の設定を解除します。ブラウザー名は正規表現で指定でき、大文字、小文字を区別します。

書式

```
BrowserMatch "ブラウザー名" 環境変数[=値] [環境変数[=値] ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

記述例

```
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4?.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4?.0" force-response-1.0
BrowserMatch "Java/1?.0" force-response-1.0
BrowserMatch "JDK/1?.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
  redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully
```

例で示した環境変数の意味を次に示します。

環境変数	内容
nokeepalive	KeepAlive 接続を無効にします。Via ヘッダーがリクエストに付加されている場合は、KeepAlive 接続を無効にできません。
downgrade-1.0	HTTP/1.1 以上のリクエストを、HTTP/1.0 のリクエストとして扱います。
force-response-1.0	HTTP/1.0 のリクエストに対して、常に HTTP/1.0 のレスポンスを応答します。
redirect-carefully	ディレクトリーへのアクセスで URL の最後に/を付加していなく、かつそれが GET メソッド以外を使用していたとき、クライアントにリダイレクトを要求しません。

2.3.37 BrowserMatchNoCase

Web ブラウザーごとに環境変数を設定する場合に指定します。

説明

Web ブラウザーごとに環境変数を設定する場合に指定します。設定する値のデフォルト値は1です。環境変数の前に!が付いたときは、その環境変数の設定を解除します。ブラウザー名は正規表現で指定でき、大文字、小文字を区別しません。

書式

```
BrowserMatchNoCase "ブラウザー名" 環境変数[=値] [環境変数[=値] ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

2.3.38 CacheNegotiatedDocs

コンテンツネゴシエーションをするリクエストで、クライアント側のキャッシュを有効にするかどうかを指定します。

説明

コンテンツネゴシエーションをするリクエストで、クライアント側のキャッシュを有効にするかどうかを指定します。ディレクティブの引数を省略した場合は、Onを指定した場合と同様の動作をします。ディレクティブを設定しない場合は、Offを指定した場合と同様の動作をします。このディレクティブの指定は、HTTP/1.1のリクエストに対しては無効です。

書式

```
CacheNegotiatedDocs [{On|Off}]
```

指定できる値

On

キャッシュされるようになります。

Off

Expires ヘッダーが付けられてキャッシュされなくなります。

デフォルト値

定義項目の省略

Off

値の省略

On

記述できる場所

httpd.conf

2.3.39 CustomLog

任意のフォーマットのログをファイルに出力させる場合に指定します。

説明

任意のフォーマットのログをファイルに出力させる場合に指定します。フォーマットはLogFormat ディレクティブで指定するフォーマットと同様です。

このディレクティブを複数指定する場合、同じファイル名は複数指定できません。

書式

```
CustomLog {ファイル名|パイプ} {"フォーマット"|ラベル名} [env=[!]環境変数]
```

指定できる値

ファイル名

ログの出力先ファイル名を指定します。ファイル名には、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が指定できます。

パイプ

標準入力からログ情報を受け取るプログラムを”|プログラム名”の形式で指定します。Web サーバはログ情報に含める改行コードをCRLFにして渡します。

パイプで指定されたプログラムは、制御プロセスと Web サーバプロセス用にログ情報を受け取るそれぞれ別のプロセスとして生成されます。これをパイププロセスと呼びます。サービスとして Web サーバを起動する場合には次の点に注意してください。

- 制御プロセスのログ情報取得不可

サービスとして Web サーバを起動した場合には、制御プロセスからのログ情報を受け取るための標準入力は NUL デバイスに関連づけられているため、制御プロセス用のパイププロセスは、制御プロセスからのログ情報を受け取ることはできません。制御プロセスからのログ情報とは、Web

サーバ起動、停止時のエラーログ情報であり、これらの情報は採取できないことになります。Webサーバ起動後のエラーログ、アクセスログの情報は Web サーバプロセスからのログ情報となりますので、Web サーバプロセス用のパイププロセスで受け取れます。

- プログラム作成時の留意点

制御プロセス用のパイププロセスは、NUL デバイスからのデータ読み込み処理で、read()のバッファが小さいと入力データ待ち状態が解除されないことがあります。read()のバッファを十分大きい値を取るなどしてパイププロセスが入力データ待ち状態にならないようにしてください。

- プログラムに引数を指定する場合の注意

プログラム、引数に空白を含む場合には、`¥`で囲んでください。

プログラム、引数を`¥`で囲む場合には、全体も`¥`で囲んでください。

(例)

```
CustomLog "|
¥"¥"Application Serverのインストールディレクトリー/httpsd/sbin/rotatelog.exe¥"
¥"Application Serverのインストールディレクトリー/httpsd/logs/access¥" 86400 -diff
540¥""
common
```

- プログラムの引数指定に誤りがある場合の注意

プログラムの引数指定に誤りがある場合、プログラムの起動に失敗しますが、Webサーバは起動します。この場合はログが出力されません。プログラムを指定した場合は、ログファイルが作成され、意図した分割が実施されることを確認してください。

フォーマット

ログフォーマットを指定します。指定できるフォーマット名を次に示します。

表 2-1 フォーマット一覧

フォーマット	意味
<code>%A</code> *1	Web サーバの IP アドレス。
<code>%a</code> *1	クライアントの IP アドレス。
<code>%B</code>	送信バイト数 (HTTP ヘッダーおよび chunked エンコーディングによって追加されたデータを除く)。
<code>%b</code>	送信バイト数 (HTTP ヘッダーおよび chunked エンコーディングによって追加されたデータを除く)。ただし、 <code>0</code> の場合は- (ハイフン)。
<code>%{cookie_name}C</code>	Cookie ヘッダー値に含まれる Cookie 名 <code>cookie_name</code> の値。Cookie ヘッダー値に複数の <code>cookie_name</code> が見つかった場合、すべての値を出力する。
<code>%D</code>	リクエスト処理時間をマイクロ秒で表示。
<code>%{env_name}e</code>	<code>env_name</code> に指定した環境変数の値。
<code>%f</code>	クライアントが要求したディレクトリーまたはファイル名。
<code>%H</code>	リクエストプロトコル(HTTP/1.0 など)。
<code>%h</code> *2	クライアントのホスト名。

フォーマット	意味
%I	リクエストとヘッダーを含む、全受信バイト数。
%{header_name}i	<i>header_name</i> に指定したリクエストヘッダーの値。
%l	クライアントの識別情報 (IdentityCheck ディレクティブが0n、かつクライアント上で identd が動作している場合)。
%m	リクエストメソッド(GET、POST など)。
%{note_name}n	<i>note_name</i> に指定した Web サーバ内モジュールの注記の値。 <ul style="list-style-type: none"> • hws_ap_root : ルートアプリケーション情報 • hws_ap_client : クライアントアプリケーション情報
%O	ヘッダーを含む、全送信バイト数。
%{header_name}o	<i>header_name</i> に指定したレスポンスヘッダーの値。
%P	HTTP 通信のリクエストを処理するプロセス ID。
%{hws_thread_id}P	HTTP 通信のリクエストを処理するスレッド ID。
%p	ServerName ディレクティブまたは<VirtualHost>の指定に従ったサーバのポート番号。
%q	問い合わせ文字列。
%r	HTTP 通信のリクエストの先頭行。
%s	ステータス (内部リダイレクトされた場合はオリジナルを示す)。
%T	リクエスト処理に掛かった時間 (秒)。HWSLogTimeVerbose ディレクティブで0n を指定すると、ミリ秒単位まで表示。
%t	リクエスト処理を開始した時刻。HWSLogTimeVerbose ディレクティブで0n を指定すると、ミリ秒単位まで表示。
%{format}t	リクエスト処理を開始した時刻。strftime() で定義されているフォーマットを <i>format</i> に指定する。
%U	URL。
%u	クライアントのユーザー名 (ユーザー認証をした場合)。
%V※2	UseCanonicalName ディレクティブの指定に従い、ServerName ディレクティブ指定値、サーバ名または IP アドレス。
%v	サーバ名。
%X	レスポンス完了時の接続ステータス。 <ul style="list-style-type: none"> • + : レスポンス送信後も接続を維持する。 • - : レスポンス送信後に接続を切断する。 • × : レスポンス完了前に接続を切断する。
%>s	最終ステータス。

注

フォーマットで示す{ }は選択を意味するものではありません。{ }内の太字はログを採取する変数名を、細字は文字列そのままを記述します。

注※1

フォーマットに%A または%a を指定した場合、IPv6 アドレスも出力できます。

注※2

フォーマットに%h または%V を指定した場合、IPv6 アドレスに対応したホスト名または IPv6 アドレスも出力できます。

表 2-2 SSL 関連のログフォーマット一覧

フォーマット	意味
%{version}c	SSL のバージョン。
%{cipher}c	現在の通信で使用している暗号種別。
%{clientcert}c	SSL クライアント証明書のsubject のDistinguished Name。

フォーマットの%の後ろにステータスコードを記述できます。

(例) エラーステータスコード 400 および 501 の場合、User-Agent リクエストヘッダー値のログを採取する。

```
%400,501{User-Agent}i
```

(例) エラーステータスコード 200、304 および 302 の 3 種類以外の場合、Referer リクエストヘッダー値のログを採取する。

```
%!200,304,302{Referer}i
```

また、env=は、指定した環境変数の設定によって、ログの採取を分ける場合に指定します。

(例) gif へのアクセスはgif.log に、gif 以外へのアクセスはnongif.log にログを採取する。

```
SetEnvIf Request-URI ¥.gif$ gif-image  
CustomLog logs/gif.log common env=gif-image  
CustomLog logs/nongif.log common env=!gif-image
```

ラベル名

LogFormat ディレクティブで定義したラベル名を指定します。

env=環境変数

指定した環境変数が設定されている場合に、ログを採取します。

env=!環境変数

指定した環境変数が設定されていない場合に、ログを採取します。

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
CustomLog logs/access.log common  
CustomLog logs/ssl.log "%t %{version}c %{cipher}c %{clientcert}c"
```

2.3.40 DefaultIcon

ディレクトリーインデックスで特定のアイコンが設定されていない場合に表示するアイコンを指定します。

説明

ディレクトリーインデックスで表示するアイコンを指定します。AddIcon、AddIconByType および AddIconByEncoding ディレクティブのどれにも該当しない場合に表示するアイコンの URL を指定します。なお、URL の「http://IPアドレス」を省略しないで IPv6 アドレスを指定する場合は、IPv6 アドレスを [] で囲んでください。

書式

```
DefaultIcon URL
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
DefaultIcon /icons/unknown.gif
```

2.3.41 DefaultLanguage

ドキュメントで使用するデフォルトの言語を指定します。

説明

ドキュメントで使用するデフォルトの言語を指定します。指定した言語コードはContent-Language レスポンスヘッダーに設定されます。AddLanguage ディレクティブの設定に対するデフォルト値となります。デフォルト値が設定されていない場合、Content-Language レスポンスヘッダーは送信しません。

書式

DefaultLanguage 言語コード

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.42 Deny

Web サーバへアクセスできるクライアントを制限する場合に指定します。

説明

Web サーバへアクセスできるクライアントを制限する場合に指定します。ホストにはアクセスを禁止するホストのドメイン名、IP アドレス、サブネット、ネットマスクを指定できます。すべてのホストからアクセスを禁止する場合は、all を指定します。

また、ホストには、IPv6 アドレスに関するドメイン名、アドレスおよびプリフィクス長も指定できます。IPv6 アドレスを指定する場合は、IPv6 アドレスを[]で囲まないでください。プリフィクス長は、「IPv6 アドレス/プリフィクス長」の形式で指定します。プリフィクス長は 10 進数で指定してください。

env=環境変数を指定すると、サーバへのアクセスを環境変数で制御できます。BrowserMatch、BrowserMatchNoCase、SetEnvIf、SetEnvIfNoCase ディレクティブとあわせて使用すれば、HTTP リクエストヘッダーフィールドに基づいてアクセスを制限できます。

Allow ディレクティブ（アクセス許可）とDeny ディレクティブ（アクセス制限）は、Order ディレクティブで評価の順序を設定できます。

ホスト	意味
ドメイン名	ドメイン名で示すホストからのアクセスを禁止する。
IP アドレス	IP アドレスで示すホストからのアクセスを禁止する。
サブネット	サブネット（IP アドレスの最初の 1 から 3 バイト）で指定したホストからのアクセスを禁止する。
ネットマスク	ネットマスク表記（例：10.1.0.0/255.255.0.0）で指定したホストからのアクセスを禁止する。10.1.0.0/16 形式で表記した場合10.1.0.0/255.255.0.0 と同じ意味である。

書式

```
Deny from {ホスト|all|env=環境変数} [{ホスト|env=環境変数} ...]
```

記述できる場所

<Directory>、.htaccess

上書き許可

Limit レベル

2.3.43 DirectoryIndex

Web ブラウザーからのリクエストが特定のコンテンツを指定していない場合に、デフォルトとしてクライアントに送信するコンテンツのファイルを指定します。

説明

Web ブラウザーからのリクエストが特定のコンテンツを指定していない場合に、デフォルトとしてクライアントに送信するコンテンツのファイル名を指定します。ファイル名を複数指定した場合は、先に指定したファイル名を優先して送信します。

ここで指定したファイルがリクエストされたディレクトリーにない場合、Options ディレクティブの指定によって Web ブラウザーの表示が変わります。

Indexes が有効の場合

Web ブラウザーに Web サーバで作成したディレクトリーのインデックスを表示します。

Indexes が無効の場合

ステータスコード 403 Forbidden を応答します。

書式

```
DirectoryIndex ファイル名 [ファイル名 ...]
```

デフォルト値

定義項目の省略

index.html

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
DirectoryIndex index.html
```

ファイル名の指定がないリクエストの場合、ディレクトリーに `index.html` があれば表示させます。

2.3.44 DocumentRoot

コンテンツを格納するドキュメントルートディレクトリーを指定します。

説明

コンテンツを格納するドキュメントルートディレクトリーを絶対パスで指定します。ディレクトリー名の終端にはスラッシュ (/) を記述しないでください。

ディレクトリー名は、絶対パスで指定してください。

書式

```
DocumentRoot ディレクトリー名
```

デフォルト値

定義項目の省略

ServerRoot ディレクティブのデフォルト値 /htdocs

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
DocumentRoot Application Serverのインストールディレクトリー/httpsd/htdocs
```

2.3.45 ErrorDocument

エラーが発生したときに、Web ブラウザーへ表示するメッセージをカスタマイズする場合に指定します。

説明

エラーが発生したときに、Web ブラウザーへ表示するメッセージをカスタマイズする場合に指定します。

テキスト

文字列を””で囲み指定します。

ローカル URL

先頭に/を記述して、自サイト内のコンテンツを指定します。

フル URL

「http://」または「https://」で始まる URL を記述し、他サイトのコンテンツを指定します。

このディレクティブに指定できるエラーステータス番号と、テキスト、ローカル URL、フル URL の指定可否について、次に示します。

エラーステータス番号 (意味)	テキスト	ローカル URL	フル URL
400 (Bad Request)	○	×	×
401 (Authorization Required)	○	○	×
403 (Forbidden)	○	○	○
404 (Not Found)	○	○	○
405 (Method Not Allowed)	○	○	○
406 (Not Acceptable)	○	○	○
408 (Request Time-out)	×	×	×
410 (Gone)	○	○	○
411 (Length Required)	○	×	×
412 (Precondition Failed)	○	○	○
413 (Request Entity Too Large)	○	○	○
414 (Request-URI Too Large)	○	×	×
416 (Requested Range Not Satisfiable)	○	○	○
417 (Expectation Failed)	○	×	×
500 (Internal Server Error)	○	○	○
501 (Method Not Implemented)	○	○	○
502 (Bad Gateway)	○	○	×
503 (Service Temporarily Unavailable)	○*	○*	○*
506 (Variant Also Negotiates)	○	○	○

(凡例)

- ：指定できる。
- ×：指定できない。

注※

流量制限機能が返すメッセージをカスタマイズする場合は、QOSResponse ディレクティブまたは QOSRedirect ディレクティブを使用してください。

このディレクティブ指定時には、次の点に留意してください。

- このディレクティブを複数指定する場合、同じエラー番号に異なる指定はできません。
- CGI プログラム内で設定されたエラーステータスに対しては、メッセージをカスタマイズできません。
- ローカル URL、フル URL の指定先でエラーとなる場合は、カスタマイズできません。
- ローカル URL の指定先でコンテンツネゴシエーションが発生する場合は、エラーとなりカスタマイズできないことがあります。
- LoadModule ディレクティブによって動的に接続したモジュール内で設定されたエラーステータスに対しても、そのモジュールの実装方法によってメッセージをカスタマイズできない場合があります。
- フル URL の指定時には、ステータスコード 302 Found および Location ヘッダーに新パスを設定した応答を返します。通常、ステータスコード 302 を受けた Web ブラウザーは、Location ヘッダーに指定されたアドレスに対して自動的にリダイレクトします。
- フル URL の指定時には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。IPv6 アドレスを指定する場合は、IPv6 アドレスを [] で囲んでください。

書式

```
ErrorDocument エラーステータス番号 {テキスト|ローカルURL|フルURL}
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
ErrorDocument 500 "Server Error."  
ErrorDocument 404 /missing.html  
ErrorDocument 403 http://some.other_server.com/subscription_info.html  
ErrorDocument 404 http://[2001::123:4567:89ab:cdef]/missing.html
```

2.3.46 ErrorLog

エラーログの出力先を指定します。

説明

エラーログを出力するファイル名を指定します。出力するログの内容は、LogLevel ディレクティブで選択できます。

ファイル名には、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が指定できます。

ファイル名

エラーログを格納するファイル名を指定します。ServerRoot ディレクティブ指定値からの相対パスで指定できます。

パイプ

標準入力からエラーログ情報を受け取るプログラムを”|プログラム名”の形式で指定します。

書式

```
ErrorLog {ファイル名|パイプ}
```

デフォルト値

定義項目の省略

logs/error.log

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
ErrorLog logs/error.log
```

2.3.47 ExpiresActive

レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加するかどうかを指定します。

説明

レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加するかどうかを指定します。

❗ 重要

- このディレクティブを使用するにはmod_expires モジュールの組み込みが必要です。
LoadModule expires_module modules/mod_expires.so
- ExpiresDefault ディレクティブまたはExpiresByType ディレクティブを指定していない場合は、ExpiresActive ディレクティブにOn が指定されていても、レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーは追加されません。

書式

```
ExpiresActive {On|Off}
```

指定できる値

On

Expires ヘッダーおよび Cache-Control ヘッダーを追加します。

Off

Expires ヘッダーおよび Cache-Control ヘッダーを追加しません。

デフォルト値

定義項目の省略

Off

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

2.3.48 ExpiresByType

レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加する場合の、MIME タイプのドキュメントに対する有効期限を指定します。

説明

レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加する場合に、指定する MIME タイプのドキュメントに対する有効期限を指定します。このディレクティブはExpiresActive ディレクティブ

で0nを指定している場合に有効になります。ExpiresDefault ディレクティブで設定されたデフォルトの有効期限は、この設定によって MIME タイプ別に上書きされます。

基準時刻をAまたはMで指定し、基準時刻から有効期限までの時間を秒単位で指定します。AまたはMと、時間との間に空白は入りません。

A

クライアントがアクセスした時刻を基準時刻とします。

M

ファイルを最後に修正した時刻を基準時刻とします。

❗ 重要

- このディレクティブを使用するにはmod_expires モジュールの組み込みが必要です。
LoadModule expires_module modules/mod_expires.so
- グリニッジ標準時 (GMT) の 2038 年 1 月 19 日 3 時 14 分 7 秒を超えないように、有効期限を設定してください。

書式

```
ExpiresByType MIMEタイプ {A|M}時間
```

指定できる値

0~2147483647

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
ExpiresByType text/html A604800
```

2.3.49 ExpiresDefault

レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加する場合に、デフォルトの有効期限を指定します。

説明

レスポンスに Expires ヘッダーおよび Cache-Control ヘッダーを追加する場合に、デフォルトの有効期限を指定します。このディレクティブは ExpiresActive ディレクティブで 0n を指定している場合に有効になります。この設定は ExpiresByType ディレクティブによって *MIME* タイプごとに上書きされます。

基準時刻を A または M で指定し、基準時刻から有効期限までの時間を秒単位で指定します。A または M と、時間との間に空白は入りません。

A

クライアントがアクセスした時刻を基準時刻とします。

M

ファイルを最後に修正した時刻を基準時刻とします。

❗ 重要

- このディレクティブを使用するには `mod_expires` モジュールの組み込みが必要です。
`LoadModule expires_module modules/mod_expires.so`
- グリニッジ標準時 (GMT) の 2038 年 1 月 19 日 3 時 14 分 7 秒を超えないように、有効期限を設定してください。

書式

```
ExpiresDefault {A|M}時間
```

指定できる値

0~2147483647

記述できる場所

`httpsd.conf`、`<VirtualHost>`、`<Directory>`、`.htaccess`

上書き許可

Indexes レベル

記述例

```
ExpiresDefault A604800
```

2.3.50 ExtendedStatus

ステータス情報表示で、それぞれのリクエストの拡張ステータス情報を表示するかどうかを指定します。

説明

server-status ハンドラーによるステータス表示形式で、それぞれのリクエストの拡張ステータス情報を表示するかどうかを指定します。

書式

```
ExtendedStatus {0n|0ff}
```

指定できる値

0n

拡張ステータス情報を表示します。この場合、クライアントの IP アドレスが IPv6 アドレスでも表示します。ただし、最大表示数は 31 バイトです。

0ff

拡張ステータス情報を表示しません。

デフォルト値

定義項目の省略

0n

記述できる場所

httpsd.conf

2.3.51 FileETag

ETag レスポンスヘッダーフィールドを作成するために使用されるファイルの属性値を指定します。

説明

ETag レスポンスヘッダーフィールドを作成するために使用されるファイル属性値を指定します。このディレクティブが指定されていない場合、ETag レスポンスヘッダーフィールドにはファイルの最終更新時刻およびバイト数が設定されます。

オプションに+-を指定しない場合は、オプションで指定した属性値が使用されます。

オプションに+-を指定する場合は、FileETag ディレクティブによって設定された属性値を変更できます。

+

設定されている属性値にオプションで指定した属性値が追加されます。

-

設定されている属性値からオプションで指定した属性値が削除されます。

指定できるオプションの一覧を次に示します。

オプション	意味
Inode	ファイルに割り振られた一意な ID が含まれます。
Mtime	ファイルの最終更新時刻が含まれます。
Size	ファイルのバイト数が含まれます。
All	Inode、Mtime、Size のオプションがすべて有効になります。
None	Etag ヘッダーが付きません。

❗ 重要

- FileETag ディレクティブのInode オプションを有効にした場合、負荷分散をしている Web サーバ環境などで、同一のコンテンツを要求するごとに、異なる IDEtag がヘッダーに含まれることがあります。このため、同一コンテンツでありながらその Etag ヘッダーの内容が異なり、ブラウザやプロキシでのキャッシングにとって不都合となることがあります。この場合、FileETag ディレクティブによって、Inode オプションを無効にするように指定することで回避できます。
- +-を使用しないでこのディレクティブを複数指定すると、最後に指定したディレクティブだけが有効になります。
- -を付加した属性値だけを指定した場合は、All オプションを指定した場合と同じ動作になります。
- All オプションとNone オプションには、+-を指定できません。
- オプションに'-Inode -Mtime -Size'と指定した場合は、このディレクティブを指定していない場合と同じ状態になります。ETag レスポンスヘッダーフィールドにはファイルの inode 番号、最終更新時刻およびバイト数が設定されます。

書式

```
FileETag [{+|-}]オプション [[{+|-}]オプション ...]
```

デフォルト値

定義項目の省略

MTime Size

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

(例 1)

```
FileETag Inode Mtime Size
FileETag -Inode
```

この指定では、ファイルの最終更新時刻およびバイト数が属性値として使用されます。

(例 2)

```
FileETag Inode Mtime
FileETag Size
```

この指定では、ファイルのバイト数が属性値として使用されます。

(例 3)

```
FileETag All
FileETag -Inode -Mtime -Size
```

この指定では、ファイルの一意な ID、最終更新時刻およびバイト数が属性値として使用されます。

2.3.52 ForceType

特定のディレクトリー下のすべてのコンテンツに対して使用する MIME タイプを指定します。

説明

<Directory>ブロックまたはアクセス制御ファイルに定義し、特定のディレクトリー下のすべてのコンテンツに対して使用する MIME タイプを指定します。none を指定すると、それまでの ForceType ディレクティブの指定が無効になります。

書式

```
ForceType MIMEタイプ
```

記述できる場所

<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.53 Header

200 番台のステータスコード応答時のレスポンスヘッダーをカスタマイズする場合に指定します。

説明

200 番台のステータスコード応答時のレスポンスヘッダーをカスタマイズする場合に指定します。リバースプロキシとして使用する場合、バックエンドの Web サーバが返すステータスコードの値に関係なく、レスポンスヘッダーをカスタマイズします。

なお、レスポンスヘッダーをカスタマイズするには `mod_headers` モジュールの組み込みが必要です。

```
LoadModule headers_module modules/mod_headers.so
```

書式

```
Header {{set|append|add}ヘッダー ヘッダー値[env=[!]環境変数]|unset ヘッダー}
```

指定できる値

set

ヘッダーを設定します。ヘッダーがある場合は、指定したヘッダー値に書き換えます。

append

存在するヘッダーにヘッダー値を追加します。存在するヘッダー値との間は、コンマで区切られます。ヘッダーがない場合は、ヘッダーを設定します。

add

ヘッダーがあっても、別の行にヘッダーを設定します。同じヘッダーを複数行設定する場合に使用します。

unset

指定したヘッダーがある場合、そのヘッダーをすべて削除します。

env=環境変数

指定した環境変数が設定されている場合に、Header ディレクティブで指定した内容を実行します。

env=!環境変数

指定した環境変数が設定されていない場合に、Header ディレクティブで指定した内容を実行します。

ヘッダー値

文字だけから成る文字列、フォーマット指示子を含む文字列または両方から成る文字列を指定できます。

ヘッダー値に空白がある場合は、”（ダブルクォーテーション）で囲む必要があります。
フォーマット指示子を次に示します。

フォーマット	意味
%t	リクエストを受け取った時刻を、1970年1月1日0時0分0秒（GMT：Greenwich Mean Time）から経過した時間で表示する。単位はマイクロ秒。先頭には"t="が付けられる。
%D	リクエスト処理に掛かった時間を表示する。単位はマイクロ秒。先頭には"D="が付けられる。
%{env_name}e	環境変数 env_name の値。

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
Header set Cache-Control no-cache
```

2.3.54 HeaderName

ディレクトリーインデックス表示時のヘッダーに付けるコメントを記述したファイルを指定します。

説明

ディレクトリーインデックス表示時のヘッダーに付けるコメントを記述したファイルのファイル名（パス情報なし）を指定します。HTML またはプレーンテキストで記述できます。ただし、AddType ディレクティブまたはTypesConfig ディレクティブで指定したファイルで、MIME タイプが正しく定義されている必要があります。プレーンテキストでコメントを作成した場合、ディレクトリーインデックス表示のHTMLには<PRE>タグが追加されます。

書式

```
HeaderName ファイル名
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
HeaderName HEADER.html
```

各ディレクトリ下のHEADER.htmlの内容をヘッダーに付けます。

2.3.55 HostnameLookups

ホスト名のルックアップの逆引きをするかどうかを指定します。

説明

CGI のREMOTE_HOST 環境変数の IP アドレスおよびログファイルに出力するクライアントの IP アドレスをホスト名に変換するために、ホスト名のルックアップの逆引きをするかどうかを指定します。なお、逆引きを使用する場合、レスポンスが遅くなります。

このディレクティブは、IPv6 アドレスにも対応しています。

書式

```
HostnameLookups {On|Off|double}
```

指定できる値

On

IP アドレスをホスト名に変換します。

Off

IP アドレスをホスト名に変換しません。

double

IP アドレスをホスト名に変換します。その後、再変換し、IP アドレスが正しいかどうかを確認します。

デフォルト値

定義項目の省略

Off

記述できる場所

httpd.conf、<VirtualHost>、<Directory>

記述例

```
HostnameLookups Off
```

IP アドレスをホスト名に変換しません。

2.3.56 HWSErrorDocumentMETACHarset

エラーが発生したときに Web ブラウザーへ表示するメッセージに対応した文字セットを指定します。

説明

エラーが発生したときに Web ブラウザーへ表示するメッセージ（以降、エラードキュメントと呼びます）についての文字セットを設定します。文字セットは、エラードキュメント中に META タグで charset= の値として設定されます。ErrorDocument ディレクティブで、カスタマイズされたエラードキュメントは、このディレクティブの META タグによる文字セットの設定対象とはなりません。

書式

```
HWSErrorDocumentMETACHarset {0n|0ff|文字セット}
```

指定できる値

0n

文字セット ISO-8859-1 を設定します。

0ff

文字セットを設定しません。

文字セット

指定した文字セットを設定します。

デフォルト値

定義項目の省略

0n

記述できる場所

httpsd.conf

記述例

```
HWSErrorDocumentMETACHarset ISO-2022-JP
```

2.3.57 HWSErrorLogClientAddr

エラーログに出力するクライアントアドレスを X-Forwarded-For ヘッダー値に変更する場合に指定します。

説明

バックエンドサーバで、エラーログに出力するメッセージテキストの "[client クライアントアドレス:ポート番号]" を "[X-Forwarded-For X-Forwarded-For ヘッダー値]" に変更するよう設定します。

ロードバランサーやプロキシサーバを介してバックエンドサーバがリクエストを受信すると、バックエンドサーバがエラーログに出力する「クライアント IP アドレス:ポート番号」がリクエストを送信したクライアントの IP アドレスではなく、ロードバランサーやプロキシサーバの IP アドレスになる場合があります。ただし、ロードバランサーやプロキシサーバの中には接続元のクライアント IP アドレスを X-Forwarded-For ヘッダーに付加する場合がありますため、出力内容を X-Forwarded-For X-Forwarded-For ヘッダーの値へ変更することで、接続元のクライアント IP アドレスが出力されるようにします。

注意事項

X-Forwarded-For ヘッダーを受信する前にエラーが発生した場合や、一部のメッセージでは変更できません。

書式

```
HWSErrorLogClientAddr X-Forwarded-For
```

指定できる値

X-Forwarded-For

エラーログに出力する "[client クライアントアドレス:ポート番号]" を "[X-Forwarded-For X-Forwarded-Forヘッダー値]" に変更します。

記述できる場所

httpsd.conf

2.3.58 HWSGracefulStopLog

計画停止時に強制停止させたリクエスト情報をエラーログファイルに出力するかどうかを指定します。

説明

計画停止時に、強制停止待ち時間を経過したあとに強制停止させたリクエスト情報を、エラーログファイルに出力するかどうかを指定します。

書式

```
HWSGracefulStopLog {0n|0ff}
```

指定できる値

0n

強制停止させたリクエスト情報をエラーログファイルに出力します。

0ff

強制停止させたリクエスト情報をエラーログファイルに出力しません。

デフォルト値

定義項目の省略

0n

記述できる場所

httpsd.conf

記述例

```
HWSGracefulStopLog 0n
```

2.3.59 HWSGracefulStopTimeout

計画停止時に実行中のリクエストを強制的に終了させるまでの待ち時間を指定します。

説明

計画停止時に、実行中のリクエストを直ちに終了するまでの強制停止待ち時間を秒単位で指定します。なお、0を指定すると、強制停止待ち時間の上限は設定されません。

書式

```
HWSGracefulStopTimeout 強制停止時間
```

指定できる値

0-3600

デフォルト値

定義項目の省略

300

記述できる場所

httpsd.conf

記述例

```
HWSGracefulStopTimeout 600
```

2.3.60 HWSImapMenuCharset

メニュー表示に対する文字セットを指定します。

説明

次の場合のメニュー表示に対する文字セットを指定します。

- イメージマップファイルの指定値にmap を指定した場合
- イメージマップ画像の座標(0,0)をマウスでポイントした場合
- 座標指定のない形でイメージマップファイルがリクエストされた場合

文字セットは、レスポンスの Content-Type ヘッダーでcharset=の値として設定されます。

書式

```
HWSImapMenuCharset 文字セット
```

デフォルト値

定義項目の省略

ISO-8859-1

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
HWSImapMenuCharset SHIFT_JIS
```

2.3.61 HWSLogSSLVerbose

SSL ハンドシェイク処理中にログに出力されるエラーの詳細情報を表示するかどうかを指定します。

説明

クライアントとサーバ間の SSL ハンドシェイク処理中に、ログに出力されるエラーのうち info レベルおよび error レベルのエラーについて、詳細情報を表示するかどうかを指定します。SSL を有効にする場合には、このディレクティブを 0n に設定することを推奨します。

書式

```
HWSLogSSLVerbose {0n|0ff}
```

指定できる値

0n

詳細情報を表示します。

0ff

詳細情報を表示しません。

デフォルト値

定義項目の省略

0ff

記述できる場所

httpsd.conf

2.3.62 HWSLogTimeVerbose

ログに出力する時間をミリ秒まで表示するかどうかを指定します。

説明

エラーログ*とリクエストログの時刻、アクセスログのアクセス時刻、リクエスト処理に掛かった時間 (%T)、およびリクエスト処理を開始した時刻 (%t) をミリ秒まで表示するかどうかを指定します。

なお、起動および再起動中に出力されるメッセージでは、このディレクティブで0nを指定してもミリ秒まで表示されない場合があります。

注※

ErrorLog ディレクティブで指定するエラーログが対象になります。ScriptLog ディレクティブで指定する CGI スクリプトのエラーログは対象になりません。

書式

```
HWSLogTimeVerbose {0n|0ff}
```

指定できる値

0n

時刻および時間をミリ秒まで表示します。

0ff

時刻および時間を秒まで表示します。

デフォルト値

定義項目の省略

0ff

記述できる場所

httpsd.conf

2.3.63 HWSMaxQueueSize

クライアントからのリクエストについての最大待ちリクエスト数を指定します。

説明

クライアントからのリクエストについての最大待ちリクエスト数を指定します。0を指定した場合は、無制限となります。このディレクティブで指定したリクエストキューサイズを超えたクライアントからのリクエストは、サーバ側で切断されます。

書式

```
HWSMaxQueueSize リクエストキューサイズ
```

指定できる値

0～2147483647

デフォルト値

定義項目の省略

8192

記述できる場所

httpsd.conf

2.3.64 HWSPrfId

PRF デーモン起動時に PRF 識別子に指定した文字列を指定します。

説明

PRF デーモン起動時に PRF 識別子に指定した文字列を指定します。

書式

HWSPrfId 文字列

デフォルト値

定義項目の省略

PRF_ID

記述できる場所

httpsd.conf

2.3.65 HWSPrfSpool

PRF デーモンを使用するときに必要なPRFSP00L 環境変数に設定するディレクトリー名を指定します。

説明

PRF デーモンを使用するときに必要なPRFSP00L 環境変数に設定するディレクトリー名を指定します。

書式

```
HWSPrfSpool ディレクトリー名
```

記述できる場所

httpsd.conf

2.3.66 HWSProxyPassReverseCookie

リバースプロキシ使用時に、バックエンドサーバから受信した Set-Cookie ヘッダーを変換する場合に指定します。

説明

リバースプロキシを使用する場合、リバースプロキシはバックエンドサーバから受信した Set-Cookie ヘッダーを変換します。これは、Web ブラウザーが Set-Cookie ヘッダーを受信したあとに、リバースプロキシを経由するバックエンドサーバへのリクエストに対して、Cookie を送信させるために必要になります。

パス名

ProxyPass ディレクティブと同じパス名を指定します。

なお、リバースプロキシを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
HWSProxyPassReverseCookie パス名
```

記述できる場所

httpsd.conf、<VirtualHost>

2.3.67 HWSRequestLog

リクエストログの出力先を指定します。

説明

リクエストログを出力するファイル名を指定します。リクエストログとは、モジュールトレース、リクエストトレース、I/O フィルタートレースおよびプロキシートレースの総称です。出力するリクエストログの種別は、HWSRequestLogType ディレクティブで選択できます。

❗ 重要

- このディレクティブを省略した場合のモジュールトレース出力先は、ErrorLog ディレクティブで指定したファイルになります。モジュールトレースの採取レベルは、LogLevel ディレクティブで指定してください。
- リクエストトレース、I/O フィルタートレースおよびプロキシートレースの出力先を、ErrorLog ディレクティブで指定したファイルにすることはできません。

書式

```
HWSRequestLog {ファイル名|パイプ}
```

指定できる値

ファイル名

リクエストログを出力するファイル名を指定します。ファイル名には、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を指定できます。

パイプ

標準入力からリクエストログ情報を受け取るプログラムを「|プログラム名」の形式で指定します。

記述できる場所

httpsd.conf

2.3.68 HWSRequestLogType

リクエストログに出力するトレース種別を指定します。

説明

HWSRequestLog ディレクティブで設定するリクエストログに出力するトレース種別を指定します。トレース種別を次に示します。

書式

```
HWSRequestLogType トレース種別 [トレース種別 ...]
```

指定できる値

指定できるトレース種別を次に示します。

module-debug

内部モジュールに対するモジュールトレースおよびmodule-info 相当のトレースを出力します。ただし、出力量が多いため、デバッグ目的以外では指定しないでください。

module-info

外部モジュールと CGI プログラム実行時のモジュールトレースを出力します。

request

リクエスト処理開始時およびリクエスト処理完了時にトレースを出力します。また、KeepAlive 接続の場合は、次のリクエストライン受信完了時にもトレースを出力します。これらのトレースをリクエストトレースと呼びます。

filter

モジュールが実装している入出力フィルター関数の実行契機を示す I/O フィルタートレースを出力します。ただし、出力量が多いため、デバッグ目的以外では指定しないでください。

proxy

プロキシ機能使用時にプロキシに関するトレースを出力します。

send-wait

一時的に送信処理ができないときの待機開始時と待機終了時にトレースを出力します。ただし、出力量が多いため、送信性能の調査目的以外では指定しないでください。

none

リクエストログを採取しません。

❗ 重要

指定したトレース種別にnone が含まれている場合、リクエストログを一切採取しません。

デフォルト値

定義項目の省略

module-info request proxy

記述できる場所

httpsd.conf

2.3.69 HWSSetEnvIfIPv6

クライアントまたはサーバのIPv6アドレスによる環境変数を定義する場合に指定します。

説明

クライアントまたはサーバのIPv6アドレスを基に環境変数を定義します。リクエスト値がIPv6アドレスで表した条件を満たす場合、指定した環境変数を設定します。設定する値のデフォルト値は1です。環境変数の前に「!」が付いたときは、その環境変数の設定を解除します。

リクエスト値として、次に示す値を指定できます。

リクエスト値	意味
Remote_Addr	クライアントのIPv6アドレス
Server_Addr	リクエストを受信したサーバのIPv6アドレス

IPv6アドレスは、[]で囲まないで指定してください。なお、IPv6アドレスの後ろに、10進数でプリフィクス長も指定できます。プリフィクス長は、「IPv6アドレス/プリフィクス長」の形式で指定します。

書式

```
HWSSetEnvIfIPv6 リクエスト値 IPv6アドレス 環境変数[=値][環境変数[=値]...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

クライアントのIPv6アドレスが2001:0:0:1230から始まる場合、環境変数IPV6_CLIENTを設定します。

```
HWSSetEnvIfIPv6 Remote_Addr 2001:0:0:1230::/64 IPV6_CLIENT
```

2.3.70 HWSSuppressModuleTrace

モジュールトレースの出力を抑止する場合に指定します。

説明

モジュールトレースの出力を抑止するモジュールファイル名および抑止する関数種別を指定します。

all

指定したモジュールが出力するモジュールトレースをすべて抑止します。

hook

指定したモジュールが出力するモジュールトレースのうち、`handler` 関数以外のモジュールトレースを抑止します。

handler

指定したモジュールが出力するモジュールトレースのうち、`handler` 関数のモジュールトレースを抑止します。

モジュールファイル名には、エラーログまたはリクエストログに出力されるモジュールファイル名称を指定します。次の例のモジュールトレースを抑止する場合は、モジュールファイル名に"`mod_example.c`"を指定します。

(例)

```
[Mon Dec 18 14:57:14 2006] [info] hws : module --> (mod_example.c[12])(1896)
[Mon Dec 18 14:57:14 2006] [info] hws : module <-- (mod_example.c[12])(1896)(-1)
```

Web サーバが標準提供している外部モジュールとモジュールファイル名の対応を次に示します。

表 2-3 Web サーバが標準提供している外部モジュールとモジュールファイル名の対応

モジュール名	モジュールファイル名
<code>mod_expires.so</code>	<code>mod_expires.c</code>
<code>mod_headers.so</code>	<code>mod_headers.c</code>
<code>mod_hws_qos.so</code>	<code>mod_hws_qos.c</code>
<code>mod_lbmethod_byrequests.so</code>	モジュールトレースは出力されません。
<code>mod_proxy.so</code>	<code>mod_proxy.c</code>
<code>mod_proxy_balancer.so</code>	<code>mod_proxy_balancer.c</code>
<code>mod_proxy_http.so</code>	モジュールトレースは出力されません。
<code>mod_reqtimeout.so</code>	<code>mod_reqtimeout.c</code>
<code>mod_slotmem_shm.so</code>	モジュールトレースは出力されません。

Web サーバが標準提供している外部モジュール以外を使用する場合は、そのモジュールのトレースが出力されることがあります。また、次のどちらかの場合は、内部モジュールに対するトレースも出力されます。

- `LogLevel` ディレクティブに `debug` を設定している。
- `HWSRequestLogType` ディレクティブに `module-debug` を設定している。

なお、このディレクティブは、複数指定できます。同じモジュールファイル名を指定した場合は、あとに指定したものが有効となります。

❗ 重要

CGI プログラム実行時のモジュールトレースは抑止できません。

書式

```
HWSSuppressModuleTrace モジュールファイル名 [all|hook|handler]
```

記述できる場所

httpd.conf

記述例

(例 1)

```
HWSSuppressModuleTrace mod_proxy.c
```

この指定では、プロキシモジュール内のすべての関数に対するモジュールトレースを抑止します。

(例 2)

```
HWSSuppressModuleTrace mod_proxy.c hook
```

この指定では、プロキシモジュール内の handler 以外の関数に対するモジュールトレースを抑止します。

2.3.71 HWSTraceIdFile

トレース採取のための共有メモリー ID を格納するファイル名を指定します。

説明

トレース採取のための共有メモリー ID を格納するファイル名を指定します。ファイル名には、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が指定できます。

このファイルは複数の Web サーバでは共有できません。同一 ServerRoot ディレクティブ指定で複数の Web サーバを起動する場合は、このディレクティブで異なるファイル名を指定する必要があります。

書式

```
HWSTraceIdFile ファイル名
```

デフォルト値

定義項目の省略

logs/hws.trcid

記述できる場所

httpsd.conf

2.3.72 HWSTraceLogFile

サーバプロセスが異常終了した場合に共有メモリーに採取されたトレースを出力するファイル名を指定します。

説明

サーバプロセスが異常終了した場合に共有メモリーに採取されたトレースを出力するファイル名を指定します。ファイル名には、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が指定できます。

トレースは複数のファイルにラップアラウンドして出力します。

最大2ファイル出力します。出力するファイルは「指定したファイル名.01」または「指定したファイル名.02」のファイル名になります。Webサーバの起動時には、「指定したファイル名.01」がカレントの出力ファイル名となります。カレントの出力ファイル名が「指定したファイル名.01」のときにトレースをファイルに出力すると、次のカレントのファイル名は「指定したファイル名.02」になります。なお、カレントの出力ファイル名が「指定したファイル名.02」のときにトレースをファイルに出力すると、次のカレントのファイル名は「指定したファイル名.01」になります。

書式

HWSTraceLogFile ファイル名

デフォルト値

定義項目の省略

logs/hws.trclog

記述できる場所

httpsd.conf

2.3.73 ImapBase

イメージマップファイルの base 行のデフォルトを指定します。

説明

イメージマップファイルの base 行のデフォルトを指定します。

書式

```
ImapBase {map|referer|URL}
```

指定できる値

map

マップファイルの場所

referer

ドキュメントの場所（イメージマップを表示した HTML ファイルの場所）

URL

指定した URL

URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

デフォルト値

定義項目の省略

map

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

2.3.74 ImapDefault

イメージマップファイルの default 行のデフォルトを指定します。

説明

イメージマップファイルの default 行のデフォルトを指定します。

書式

```
ImapDefault {error|nocontent|map|referer|URL}
```

指定できる値

error

標準のエラーメッセージを表示します（ステータスコード 500 Server Error を応答します）。

nocontent

リクエストを無視します（ステータスコード 204 No Content を応答します）。

map

マップファイル中の URL をメニュー表示します。

referer

ステータスコード 302 Found を応答します。

URL

指定した URL のコンテンツを表示します。

URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

デフォルト値

定義項目の省略

nocontent

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

2.3.75 ImapMenu

メニュー表示を指定します。

説明

イメージマップファイルの指定値にmapを与えた場合またはイメージマップ画像の(0,0)座標をマウスでポイントした場合のメニュー表示を指定します。座標指定のない形でイメージマップファイルがリクエストされた場合の動作もこの設定に従います。

書式

```
ImapMenu {none|formatted|semiformatted|unformatted}
```

指定できる値

none

メニューは生成しません。このときの動作は、マップファイル中の default 行の指定に従います。

formatted

ヘッダーおよびリンク一覧を表示します。マップファイル中のコメントは無視されます。

semiformatted

リンク一覧を表示します。マップファイル中のコメントも表示します。

unformatted

マップファイル中に HTML を記述することで、メニューの形式を自由に設定できます。

デフォルト値

定義項目の省略

formatted

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

2.3.76 Include

設定ファイルとして使用するファイルを指定します。

説明

ファイル名で指定したファイルを設定ファイルとして利用できるようにします。

ファイル名には、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が指定できます。このディレクティブを複数指定する場合、マージされた内容が使用されます。ファイル内に同じディレクティブがある場合、後ろに指定した方で上書きされます。

書式

```
Include ファイル名
```

記述できる場所

httpsd.conf

2.3.77 IndexIgnore

ディレクトリーインデックス表示時に、Web ブラウザーに表示させないファイル名を指定します。

説明

ディレクトリーインデックス表示時に、Web ブラウザーに表示させないファイル名を指定します。正規表現でも指定できます。

書式

```
IndexIgnore ファイル名 [ファイル名 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

2.3.78 IndexOptions

ディレクトリーインデックスの整形表示機能のオプションを指定します。

説明

ディレクトリーインデックスの整形表示機能のオプション設定をします。オプションの前に+を指定するかまたは+-を省略するとそのオプションが有効になります。デフォルトではすべてのオプションが無効です。指定できるオプションの一覧を次に示します。

表 2-4 オプション一覧

オプション	意味
Charset=文字セット	インデックス表示するページの文字セットを指定します。HeaderName ディレクティブまたはReadmeName ディレクティブで指定したファイルで使用している文字セットが、デフォルトの文字セット (UTF-8) と異なる場合は、このオプションで HeaderName ディレクティブまたはReadmeName ディレクティブで指定したファイルと同じ文字セットを指定してください。このオプションでは、=文字セットを省略できません。また、-Charset 指定時も+Charset 指定時と同様の動作をします。オプションが省略された場合のデフォルト値は、UTF-8 です。
DescriptionWidth[={文字数 *}]	ファイル説明文エリアの幅を文字数(1文字=1バイト)で指定します。*を指定した場合はAddDescription ディレクティブで指定したファイル説明文の最大長に合わせて表示します。このオプションを省略した場合のファイル説明文エリアの幅は、23バイト(ただし、SuppressSize 指定時+7、SuppressLastModified 指定時+19)です。-DescriptionWidth 指定時は={文字数 *}を省略できます。この場合の表示幅は23バイトです。 オプションが省略された場合のデフォルト値は、23、30、42 または49 です。
FancyIndexing	ディレクトリーインデックスの整形表示機能を有効にします。
FoldersFirst	ファイルよりディレクトリーを先にインデックス表示する場合に指定します。ただし、FancyIndexing が有効な場合だけです。
IconsAreLinks	ディレクトリーインデックス整形表示時のアイコンをファイルに対するアンカーにします。
IconHeight[={ピクセル数}]	ディレクトリーインデックス整形表示時のアイコンの高さをピクセル数で指定します。IconWidth オプションと一緒に指定します。インデックスを表示する HTML の IMG タグの HEIGHT 属性になります。 指定できる値は>0 です。 オプションが省略された場合のデフォルト値は22 です。
IconWidth[={ピクセル数}]	ディレクトリーインデックス整形表示時のアイコンの幅をピクセルで指定します。IconHeight オプションと一緒に指定します。インデックスを表示する HTML の IMG タグの WIDTH 属性になります。 指定できる値は>0 です。 オプションが省略された場合のデフォルト値は20 です。
IgnoreCase	ディレクトリーインデックス整形表示時に、ファイル名およびディレクトリー名の 大文字と小文字の区別をしないで並べ替えます。
NameWidth[={文字数 *}]	ファイル名およびディレクトリー名エリアの幅を文字数 (1文字=1バイト) で指定します。*を指定した場合はファイル名およびディレクトリー名の最大長に合わせて表示します。 ={文字数 *}を省略する場合は必ず-NameWidth と指定してください。 オプションが省略された場合のデフォルト値は23 です。
ScanHTMLTitles	AddDescription ディレクティブの指定がない場合に、HTML ファイル中の<TITLE> タグを検索し、説明文として表示します。
SuppressColumnSorting	ファイル名、ディレクトリー名、最終更新日時、ファイルサイズおよびファイルの説明文の各カラムでインデックスを並べ替える機能を抑止します。

オプション	意味
SuppressDescription	ファイルの説明文を表示しません。
SuppressHTMLPreamble	HeaderName ディレクティブが指定されている場合、HeaderName ディレクティブで指定されたファイルの内容と、自動生成される HTML ヘッダー部 (<HTML>や<TITLE>など) が共に出力されます。このオプションは、HeaderName ディレクティブで指定されたファイルが HTML で記述されている場合、自動生成される HTML ヘッダー部の出力を抑制します。
SuppressLastModified	最終更新日時を表示させません。
SuppressSize	ファイルサイズを表示させません。
TrackModified	ディレクトリー表示のためのレスポンスの HTTP レスポンスヘッダーに、Last-Modified 値とEtag 値を設定します。このオプションを指定すると、クライアントは HEAD リクエストでディレクトリーはファイル構成の変更を確認できるため、クライアントのキャッシュ機能を有効に活用できます。このオプションは OS とファイルシステムが stat() をサポートしている場合だけ有効です。

書式

```
IndexOptions [+|-]オプション [[+|-]オプション ...]
```

注意事項

- このディレクティブを複数指定する場合、同じファイル名に異なる文字列は指定できません。
- IconHeight、IconWidth、NameWidth で=値を指定する場合、-の指定はできません。
- 設定されたオプションは、httpsd.conf、<VirtualHost>、<Directory>、.htaccess の順で、また、上位ディレクトリーから下位ディレクトリーへ継承します。継承したオプションを最終的にマージして、インデックス整形表示形式を決定します。
- httpsd.conf で+-を付けてオプションを指定しても無効になります。ただし、httpsd.conf、<VirtualHost>、<Directory>、.htaccess の順で、また、下位ディレクトリーに継承されます。継承されたオプション指定はマージ処理で有効になります。参照順位が下位の指定場所でオプションの指定がある場合または次に示すディレクティブのどれかの指定がある場合、マージ処理が実行されます。
 - AddAlt
 - AddAltByEncoding
 - AddAltByType
 - AddDescription
 - AddIcon
 - AddIconByEncoding
 - AddIconByType
 - DefaultIcon

- HeaderName
- ReadmeName

(例)

httpsd.conf ファイルにIndexOptions +FancyIndexing +IconsAreLinks を指定した場合、下位の指定場所でインデックス関係のディレクティブ指定がなければFancyIndexing、IconsAreLinks は無効になります。

httpsd.conf ファイルにIndexOptions +FancyIndexing +IconsAreLinks、かつ下位ディレクトリーのアクセス制御ファイルに、AddDescription "テキストファイル" *.txtを指定した場合、FancyIndexing、IconsAreLinks は有効になります。

- +-指定のないCharset、IconHeight、IconWidth、NameWidth ディレクティブを指定すると、その指定場所内でそのオプションが指定されている位置より前に指定されている+-付のオプション (Charset、IconHeight、IconWidth、NameWidth を除いて) は無効になります。

(例)

```
IndexOptions FancyIndexing -IconsAreLinks IconHeight IconWidth
```

この場合、FancyIndexing、IconHeight、IconWidth ディレクティブが有効になります。IconsAreLinks の-指定は継承されません。

- 指定場所間で同じディレクトリーのインデックスを対象にオプション指定した場合のマージ処理は、参照順位がより後方の指定場所で+-のないオプションを指定すると、先に指定したオプションは無効になります。ただし、IconHeight、IconWidth、NameWidth は無効になりません。

(例 1)

- httpsd.conf ファイルの指定

```
IndexOptions +FancyIndexing +IconsAreLinks
```

- アクセス制御ファイルの指定

```
IndexOptions FancyIndexing SuppressLastModified
```

これらを指定した場合、IconsAreLinks は無効になります。FancyIndexing、SuppressLastModified は有効になります。

(例 2)

- httpsd.conf ファイルの指定

```
IndexOptions SuppressColumnSorting +FancyIndexing +IconsAreLinks
```

- アクセス制御ファイルの指定

```
IndexOptions FancyIndexing SuppressLastModified
```

これらを指定した場合、SuppressColumnSorting、IconsAreLinks は無効になります。また、FancyIndexing、SuppressLastModified は有効になります。

- 指定場所間で、同じディレクトリーのインデックスを対象にオプション指定した場合のマージ処理は、同じオプションに対して+と-の両方を指定すると-指定が有効になります。

(例)

- httpsd.conf ファイルの指定

```
IndexOptions +FancyIndexing -IconsAreLinks
```

- アクセス制御ファイルの指定

```
IndexOptions +IconsAreLinks
```

これらを指定した場合、IconsAreLinks は無効になります。

- 同じ指定場所で+-を指定しないオプションを指定すると、Charset、IconHeight、IconWidth、NameWidth ディレクティブ以外の+-で指定したオプションは無効になります。

(例 1)

- httpsd.conf ファイルの指定

```
IndexOptions +IconsAreLinks FancyIndexing +SuppressLastModified
```

この場合、IconsAreLinks は無効になります。

(例 2)

- <VirtualHost>ブロック、<Directory>ブロックまたはアクセス制御ファイルの指定

```
IndexOptions +IconsAreLinks FancyIndexing +SuppressLastModified
```

この場合、IconsAreLinks、SuppressLastModified は無効になります。

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

2.3.79 IndexOrderDefault

ディレクトリーインデックス表示での、デフォルトのファイルの並び順を指定します。

説明

ディレクトリーインデックス表示での、ファイルの並び順のデフォルトを指定します。

書式

```
IndexOrderDefault {Ascending|Descending} {Name|Date|Size|Description}
```

指定できる値

Ascending

昇順

Descending

降順

Name

ファイル名で並べます。

Date

ファイル更新日付で並べます。

Size

ファイルサイズで並べます。

Description

AddDescription ディレクティブで指定した説明文で並べます。

デフォルト値

定義項目の省略

Ascending Name

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

2.3.80 KeepAlive

KeepAlive 接続を有効にするかどうかを指定します。

説明

KeepAlive 接続を有効にするかどうかを指定します。実際に KeepAlive が実行されるのはクライアント側も KeepAlive に対応している場合だけです。KeepAlive はサーバプロセスとクライアントとの接続が持続されるので、連続したリクエストのレスポンスが良くなります。反面、サーバプロセスが特定のクライアント専用になるので、Web サーバ全体としてサービス能力が低下することもあります。

KeepAliveTimeout、MaxKeepAliveRequests ディレクティブを使用して調整する必要があります。

KeepAlive ディレクティブがOn の場合は、レスポンスの送信が完了し、KeepAliveTimeout ディレクティブで指定した時間を経過した後に、Web サーバから接続切断要求を行います。また、KeepAlive ディレクティブがOff の場合は、レスポンスの送信が完了した直後に、Web サーバから接続切断要求を行います。

この接続切断要求に対するクライアントの応答によっては、2 秒間サーバスレッドが専有されることがあります。

書式

```
KeepAlive {On|Off}
```

指定できる値

On

持続型接続 (KeepAlive) を有効にします。

Off

持続型接続 (KeepAlive) を無効にします。

デフォルト値

定義項目の省略

On

記述できる場所

httpsd.conf

記述例

```
KeepAlive On
```

2.3.81 KeepAliveTimeout

KeepAlive 接続時のリクエストの待ち時間を指定します。

説明

KeepAlive 接続時の要求待ち時間を秒単位で指定します。この時間以上経過しても、クライアントから次のリクエストが来ない場合、接続を切断します。KeepAlive はサーバプロセスが特定のクライアントに占有されます。ある Web ページから次の Web ページへ移る場合に必要とする標準的な時間以上

は、タイムアウトにして接続を切断し、サーバプロセスをほかのリクエストの処理に当てるようにします。時間に0を指定した場合は、KeepAlive 接続が無効になります。

書式

```
KeepAliveTimeout 時間
```

指定できる値

0-65535

デフォルト値

定義項目の省略

5

記述できる場所

httpsd.conf

記述例

```
KeepAliveTimeout 3
```

KeepAlive 接続時の要求待ち時間は 3 秒

2.3.82 LanguagePriority

クライアントが使用言語の優先度を示さなかった場合の優先順位を指定します。

説明

使用言語を優先順位の高い順に指定します。コンテンツネゴシエーションで、Web ブラウザーからのリクエストに言語コードの優先順位 (Accept-Language ヘッダー) が含まれていない場合に、ここで指定した優先順位が使用されます。

書式

```
LanguagePriority 言語コード [言語コード ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
LanguagePriority ja en fr de
```

優先順位は日本語、英語、フランス語、ドイツ語の順

2.3.83 LimitRequestBody

サーバが受信するオブジェクトボディーのサイズの上限を指定します。

説明

HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合のオブジェクトボディー（データ）のサイズの上限を指定します。Web ブラウザーから<FORM METHOD=POST ACTION=...>によるリクエストを送る場合などにオブジェクトボディーが用いられます。上限値を設定しない場合は、0を指定してください。

書式

```
LimitRequestBody リクエストボディーサイズ
```

指定できる値

0~2147483647

デフォルト値

定義項目の省略

0

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

2.3.84 LimitRequestFields

サーバが受信する HTTP ヘッダー数の上限を指定します。

説明

HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合の HTTP ヘッダー数の上限を指定します。リクエストの HTTP ヘッダー数は、Web ブラウザーやリクエストを中継するプロキシなどの仕様で変わります。上限値を設定しない場合は、0 を指定してください。

このディレクティブ指定値を、バーチャルホストに対するリクエストにも適用するには、<VirtualHost>ブロックよりも前に指定する必要があります。

書式

```
LimitRequestFields ヘッダー数
```

指定できる値

0–32767

デフォルト値

定義項目の省略

100

記述できる場所

httpd.conf

2.3.85 LimitRequestFieldsize

サーバが受信する 1 つの HTTP ヘッダーのサイズの上限を指定します。

説明

HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合、1 つの HTTP ヘッダーの、サイズの上限を指定します。リクエストヘッダーのサイズは Web ブラウザーやリクエストを中継するプロキシなどの仕様で変わります。

このディレクティブ指定値を、バーチャルホストに対するリクエストにも適用するには、<VirtualHost>ブロックよりも前に指定する必要があります。

書式

```
LimitRequestFieldsize ヘッダーサイズ
```

指定できる値

0–8190

デフォルト値

定義項目の省略

8190

記述できる場所

httpsd.conf

2.3.86 LimitRequestLine

サーバが受信するリクエストラインの長さの上限を指定します。

説明

HTTP 通信によって、Web ブラウザーが送信してくるリクエストをサーバが受信する場合のリクエストライン（メソッド、問い合わせ文字列などを含む URI、HTTP バージョン）の長さの上限を指定します。Web ブラウザーから<FORM METHOD=GET ACTION...>によるリクエストを送る場合などに問い合わせ文字列としてリクエストラインが用いられます。なお、リクエストラインとして Web ブラウザーから何バイト送れるかは、Web ブラウザーやリクエストを中継するプロキシなどの仕様で変わります。

このディレクティブ指定値を、バーチャルホストに対するリクエストにも適用するには、<VirtualHost>ブロックよりも前に指定する必要があります。

書式

LimitRequestLine リクエストライン長

指定できる値

0–8190

デフォルト値

定義項目の省略

8190

記述できる場所

httpsd.conf

2.3.87 Listen

リクエストを受け付ける IP アドレスおよびポート番号を指定します。

説明

リクエストを受け付ける IP アドレスおよびポート番号を指定します。このディレクティブの指定は必須です。

IP アドレスには IPv6 アドレスも指定できます。IPv6 アドレスを指定する場合は、IPv6 アドレスを [] で囲ってください。ただし、IP アドレスを省略してポート番号だけを指定した場合は、IPv4 アドレスを使用したリクエストだけを受け付けます。このため、IPv6 アドレスを使用する場合は、必ず Listen ディレクティブに IPv6 アドレスを指定してください。

❗ 重要

Listen ディレクティブの IP アドレスを変更してサーバを再起動する場合、サーバをいったん停止後、起動してください。コマンドなどで再起動を選択すると、サーバの起動に失敗する場合があります。

書式

```
Listen [IPアドレス:]ポート番号
```

記述できる場所

httpsd.conf

記述例

```
Listen 80
Listen [2001::123:4567:89ab:cdef]:8080
Listen [::]:80
```

2.3.88 ListenBacklog

クライアントからの接続要求の最大の待ち行列数を指定します。

説明

クライアントからの接続要求の最大の待ち行列数を指定します。この指定値はシステムコール listen() のバックログ数として設定されます。ただし、指定値の制限値や、実際の待ち行列数の最大値については OS によって異なるため、詳細は各 OS の listen() についてのマニュアルや、各 OS の TCP/IP 実装の詳細を説明しているドキュメントを参照してください。

書式

ListenBacklog バックログ数

指定できる値

1-2147483647

デフォルト値

定義項目の省略

511

記述できる場所

httpsd.conf

2.3.89 LoadFile

DSO によって組み込むモジュールが参照するコードがあるオブジェクトファイルまたはライブラリーを指定します。

説明

DSO によって組み込むモジュールが参照するコードがあるオブジェクトファイルまたはライブラリーを指定します。ファイル名には、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が指定できます。

LoadModule ディレクティブでこのファイルを参照するモジュールを指定する場合、それらがhttpsd.conf で使用される前に、このディレクティブを指定する必要があります。

書式

LoadFile ファイル名 [ファイル名 ...]

記述できる場所

httpsd.conf

2.3.90 LoadModule

Web サーバに動的に組み込むモジュールを指定します。

説明

Web サーバに動的に組み込むモジュールを指定します。ライブラリーファイル名には、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が指定できます。

書式

```
LoadModule module 構造体名 ライブラリーファイル名
```

記述できる場所

httpsd.conf

記述例

```
LoadModule hws01_module libexec/mod_hws01.so  
LoadModule hws02_module libexec/mod_hws02.so
```

モジュール hws01_module とモジュール hws02_module を組み込みます。

2.3.91 LogFormat

ログのフォーマットにラベル名を定義する場合に指定します。

説明

ログのフォーマットにラベル名を定義します。ここで定義したラベル名を CustomLog ディレクティブで指定できます。

なお、フォーマットに %A または %a を指定した場合、IPv6 アドレスも出力できます。また、フォーマットに %h または %V を指定した場合、IPv6 アドレスに対応したホスト名または IPv6 アドレスも出力できます。

ラベル名を付けない場合は、このディレクティブを複数指定できません。

書式

```
LogFormat "フォーマット" [ラベル名]
```

デフォルト値

定義項目の省略

```
"%h %l %u %t ¥"¥r¥" %>s %b"
```


記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
LogFormat "%h %l %u %t %r" "%s %b" "%{Referer}i" "%{User-Agent}i"
combined
LogFormat "%h %l %u %t %r" "%s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

2.3.92 LogLevel

エラーログに出力するエラーのレベルを指定します。

説明

エラーログに出力するエラーのレベルを指定します。指定したレベルの上位レベルのログを出力します。ただし、notice レベルのログはこの指定に関係なく出力されます。また、Web サーバ起動時など、レベル指定の解析終了前に出力されるメッセージは、この指定に関係なく出力される場合があります。

エラーレベルの上位順は、「emerg」、「alert」、「crit」、「error」、「warn」、「notice」、「info」、「debug」です。

書式

```
LogLevel {debug|info|notice|warn|error|crit|alert|emerg}
```

指定できる値

emerg

緊急メッセージ

alert

即時処理要求メッセージ

crit

致命的な状態のメッセージ

error

一般的エラーメッセージ

warn

警告レベルメッセージ

notice

標準的だが重要なメッセージ

info

インフォメーションメッセージ、外部モジュールと CGI プログラム実行時のモジュールトレース※

debug

デバッグレベルメッセージ、内部モジュールトレースおよび info 相当のモジュールトレース※

注※

モジュールトレースは、エラーログではなくリクエストログに出力するよう設定できます。

デフォルト値

定義項目の省略

warn

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
LogLevel info
```

2.3.93 MaxKeepAliveRequests

KeepAlive 連続接続回数の上限を指定します。

説明

KeepAlive 連続接続回数の上限を指定します。上限値を設定しない場合は0を指定します。KeepAlive はサーバプロセスが特定のクライアントに占有されるので、ほかのクライアントにもサービスの機会を与えるために上限を設けます。

書式

```
MaxKeepAliveRequests 接続数
```

指定できる値

0～2147483647

デフォルト値

定義項目の省略

100

記述できる場所

httpsd.conf

記述例

```
MaxKeepAliveRequests 100
```

2.3.94 MultiviewsMatch

コンテンツネゴシエーションの対象となる拡張子の種類を指定します。

説明

コンテンツネゴシエーションの対象となる拡張子の種類を指定します。

書式

```
MultiviewsMatch {NegotiatedOnly|Handlers}
```

指定できる値

NegotiatedOnly

拡張子が文字セット、圧縮形式、言語コード、MIME タイプと関連づけられたものだけをコンテンツネゴシエーションの対象にします。

Handlers

NegotiatedOnly を指定した場合の対象に加え、ハンドラーと関連づけられた拡張子についてもコンテンツネゴシエーションの対象にします。

デフォルト値

定義項目の省略

NegotiatedOnly

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
MultiviewsMatch Handlers
```

2.3.95 Options

ユーザーに利用させる機能を制限する場合に指定します。

説明

ユーザーが利用できる機能を制限する場合に指定します。

+

オプションで指定した機能の利用を許可します。

-

オプションで指定した機能の利用を禁止します。

オプション	機能
All	MultiViews、SymLinksIfOwnerMatch を除くすべてのオプションが有効です。
ExecCGI	CGI スクリプトの実行を許可します。
Indexes	URL にディレクトリーが指定されたとき、DirectoryIndex ディレクティブで指定したファイル（デフォルトはindex.html）がない場合、ディレクトリーのインデックスを表示します。
MultiViews	Content-negotiated Multiviews をサポートします。
None	すべてのオプションで指定できる機能を無効にします。

+-を使用しないでこのディレクティブを複数指定すると、最後に指定したディレクティブだけが有効になります。

(例 1)

```
Options All
Options ExecCGI
```

このようにオプションに+-を指定しないディレクティブを 2 行指定した場合、ユーザーは CGI スクリプトの実行機能だけが利用できます。ディレクトリーインデックスなどの機能は利用できません。

(例 2)

httpsd.conf ファイルの指定

```
Options All
```

アクセス制御ファイルの指定

```
Options ExecCGI
```

httpsd.conf ファイルのあとにアクセス制御ファイルが参照されるので、アクセス制御ファイルがあるディレクトリーでは CGI スクリプトの実行機能だけが利用できます。

(例 3)

```
Options Indexes ExecCGI
```

このように 1 行に+-を指定しないオプションを指定した場合は、指定した機能の両方を利用できます。

書式

```
Options {+|-}オプション [{+|-}オプション ...]
```

デフォルト値

定義項目の省略

None

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Options レベル

2.3.96 Order

Allow ディレクティブとDeny ディレクティブの指定の評価の順序を指定します。

説明

Allow ディレクティブとDeny ディレクティブの指定の評価の順序を指定します。指示子に指定できるものを次に示します。先に評価されたものは、あとに評価されるものに上書きされます。

指示子	意味
deny, allow	Deny ディレクティブの指定を、Allow ディレクティブの指定より先に評価
allow, deny	Allow ディレクティブの指定を、Deny ディレクティブの指定より先に評価

指示子	意味
mutual-failure	Allow ディレクティブに指定され、Deny ディレクティブに指定されていないホストだけアクセスを許可

書式

Order 指示子

デフォルト値

定義項目の省略

deny, allow

記述できる場所

<Directory>、.htaccess

上書き許可

Limit レベル

2.3.97 PassEnv

CGI スクリプトに渡す任意の環境変数を指定します。

説明

CGI スクリプトに渡す任意の環境変数を指定できます。

書式

PassEnv 環境変数 [環境変数 ...]

記述できる場所

httpd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

PassEnv TMP

2.3.98 PidFile

制御プロセス ID を格納するファイルを指定します。

説明

制御プロセス ID を格納するファイル名を指定します。ファイル名には、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が指定できます。

❗ 重要

PidFile ディレクティブ指定値を変更する場合は、Web サーバを停止してから変更してください。また、Web サーバのプロセス ID 取得コマンドのFILENAME 変数にも同じ値を指定してください。Web サーバのプロセス ID 取得コマンドのデフォルトの格納先を次に示します。Web サーバのプロセス ID 取得コマンドは、getpid-webserver で始まるファイルです。

*Application Server*のインストールディレクトリー/`javaee/glassfish/domains/ドメイン名/server_templates/webserver/bin`

書式

PidFile ファイル名

デフォルト値

定義項目の省略

logs/httpd.pid

記述できる場所

httpsd.conf

記述例

PidFile logs/httpd.pid

2.3.99 ProxyErrorOverride

バックエンドサーバからのレスポンスがエラーのステータスコードであった場合にレスポンスを上書きするかどうかを指定します。

説明

バックエンドサーバからのステータスコードが 400 番台または 500 番台の場合、レスポンスヘッダーとレスポンスボディをオーバーライドします。その結果、リバースプロキシはバックエンドサーバからのレスポンスではなく、自身が生成したレスポンスをクライアントに返します。

なお、リバースプロキシを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
ProxyErrorOverride {0n|0ff}
```

指定できる値

0n

バックエンドサーバからのステータスコードが 400 番台または 500 番台の場合、レスポンスヘッダーとレスポンスボディをオーバーライドします。

0ff

レスポンスヘッダーとレスポンスボディをオーバーライドしません。

デフォルト値

定義項目の省略

0ff

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
ProxyErrorOverride On
```

バックエンドサーバからのステータスコードが 400 番台または 500 番台の場合、リバースプロキシが生成したレスポンスをクライアントに返します。

2.3.100 ProxyPass

バックエンドサーバに転送するリクエストと転送するアドレスを指定します。

説明

リバースプロキシを使用する場合、Web ブラウザーからのリクエストとそれを転送するアドレスを指定します。

なお、リバースプロキシを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
ProxyPass パス名 {URL|!} [キー=値 [キー=値 ...]]
```

指定できる値

パス名

Web ブラウザーからリバースプロキシへのリクエストをスラッシュ (/) から始まる URL で指定します。

URL

転送先となるバックエンドサーバの URL を次の文字列を含むどちらかの形で指定します。

- "http://IPアドレス[:ポート番号]/"
- "http://ホスト名[:ポート番号]/"

URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

mod_proxy_balancer によるロードバランサー構成を使用する場合には、"balancer://クラスター名/"を指定します。クラスター名とは、<Proxy>ディレクティブと対応づけするための任意の文字列です。クラスター名には、/を含めないでください。mod_proxy_balancer によるロードバランサー構成では、<Proxy>ディレクティブとBalancerMember ディレクティブも同時に指定する必要があります。

!

パス名に指定したリクエストをリバースプロキシへのリクエストとしない場合に指定します。

キー

URL がhttp://で始まる場合、balancer://で始まる場合で指定できるキーが異なります。

それぞれの場合に指定できるキーを次に示します。

- URL がhttp://で始まる場合に指定できるキー

キー	値	内容
timeout	1~65535 (単位:秒)	次のバックエンドサーバとの送受信時の待ち時間を指定します。 <ul style="list-style-type: none"> バックエンドサーバへのリクエスト送信中にデータを送信できなくなった場合の待ち時間 バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間 バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間 キーが省略された場合のデフォルト値は、ProxyTimeout ディレクティブ指定値です。
connectiontimeout	1~65535 (単位:秒)	バックエンドサーバとの接続時の待ち時間を指定します。 キーが省略された場合のデフォルト値は、timeout キー値です。

- URL がbalancer://で始まる場合に指定できるキー

キー	値	内容
lbmethod	byrequests	ロードバランサーの振り分けアルゴリズムを指定します。 byrequests: リクエスト数による振り分けです。 キーの指定がない場合、byrequests による振り分けをします。
stickysession	64 バイト未満の文字列	振り分け先を決める際に使用する Cookie 名を指定します。 このキーを指定する場合は、同時にBalancerMember ディレクティブのroute キーも指定が必要です。 キーが省略された場合のデフォルト値はありません。

注意事項

!を指定する場合は、URL を指定するProxyPass ディレクティブより先に指定してください。

(例)

```
ProxyPass /abc/def/ !
ProxyPass /abc/ http://backend.example.com/
```

記述できる場所

httpsd.conf、<VirtualHost>

2.3.101 ProxyPassReverse

バックエンドサーバから受信した Location ヘッダーの URL を変更する場合に指定します。

説明

リバースプロキシを使用する場合、バックエンドサーバからのリダイレクトレスポンスの Location ヘッダーで示す URL を変更します。Web ブラウザーからのリダイレクトによるリクエストをリバースプロキシを通すリクエストにするために Location ヘッダーをこのディレクティブの指定値に変更します。

なお、リバースプロキシを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
ProxyPassReverse パス名 URL
```

指定できる値

パス名

リダイレクトのリクエスト先であるリバースプロキシのパス名を、スラッシュ (/) から始まる URL で指定します。

URL

変更対象となる Location ヘッダー中のバックエンドサーバの URL を”http://IPアドレスまたはホスト名[:ポート番号]/”を含む形で指定します。

URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。IPv6 アドレスにはさまざまな表記方法がありますので、指定値に注意してください。IPv6 アドレスの表記が指定値と一致しない場合、ディレクティブが有効になりません。IPv6 アドレスを指定する場合は、バックエンドサーバからの応答の Location ヘッダー値に含まれる IPv6 アドレスの表記を確認してください。

mod_proxy_balancer によるロードバランサー構成で使用する場合、ProxyPass ディレクティブの URL (balancer://クラスター名/) を指定できます。これを指定した場合、対応するBalancerMember ディレクティブに指定している URL が実際の変更対象となる URL になります。通常、BalancerMember ディレクティブは複数指定しているので、そのすべての URL が変更対象となります。

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
ProxyPass /examples/ balancer://cluster1/  
ProxyPassReverse /examples/ balancer://cluster1/  
<Proxy balancer://cluster1/>  
    BalancerMember http://IPアドレスA/examples/
```

```
BalancerMember http://IPアドレスB/examples/  
</Proxy>
```

バックエンドサーバからの Location ヘッダーに、*http://IPアドレスA/examples/*、または*http://IPアドレスB/examples/*が含まれていた場合に変更対象になります。

2.3.102 ProxyPreserveHost

クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送するかどうかを指定します。

説明

リバースプロキシを使用する場合、クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送するかどうかを指定します。

なお、リバースプロキシを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
ProxyPreserveHost {0n|0ff}
```

指定できる値

0n

クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送します。

0ff

クライアントから受信した Host ヘッダーの値をProxyPass ディレクティブの指定値に従って変更して、バックエンドサーバに転送します。

デフォルト値

定義項目の省略

0ff

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
ProxyPreserveHost On
```

クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送します。

2.3.103 ProxyTimeout

バックエンドサーバとの送受信に関する待ち時間を指定します。

説明

次の待ち時間を秒単位で指定します。

- リバースプロキシを使用している場合の、バックエンドサーバへのリクエスト送信中にデータを送信できなくなった場合の待ち時間
- リバースプロキシを使用している場合の、バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間
- リバースプロキシを使用している場合の、バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間

なお、リバースプロキシを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
ProxyTimeout 時間
```

指定できる値

1-65535

デフォルト値

定義項目の省略

Timeout ディレクティブの値

記述できる場所

httpsd.conf、<VirtualHost>

2.3.104 ProxyVia

Via ヘッダーの使用を制御する場合に指定します。

説明

このディレクティブはプロキシーで Via ヘッダーの使用を制御する場合に指定します。

なお、リバースプロキシーを使用するには、次のモジュールの組み込みが必要です。

- mod_proxy モジュール
LoadModule proxy_module modules/mod_proxy.so
- mod_proxy_http モジュール
LoadModule proxy_http_module modules/mod_proxy_http.so

書式

```
ProxyVia {on|off|full|block}
```

指定できる値

on

Via ヘッダーに自ホストの情報を追加します。すでにある情報は変更しません。

off

Via ヘッダーに自ホストの情報を追加しません。すでにある情報は変更しません。

full

コメントとして自ホストのバージョンを付けた情報を Via ヘッダーに追加します。すでにある情報は変更しません。

block

Via ヘッダーに自ホストの情報を追加しません。リクエスト中の Via ヘッダーは削除します。

デフォルト値

定義項目の省略

Off

記述できる場所

httpsd.conf、<VirtualHost>

2.3.105 QOSCookieDomain

流量制限機能に使用する Cookie が有効とされるドメインを指定します。

説明

流量制限機能に使用する Cookie が有効とされるドメインを指定します。この値は、HWS 作成モードで使用され、ユーザー作成モードでは使用されません。

複数のホストを設定している場合、このディレクティブを設定することでドメイン部分の共通するホスト間で Cookie を使用できるようになります。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

QOSCookieDomain <i>ドメイン名</i>

指定できる値

ドメイン名

流量制限機能に使用する Cookie が有効とされるドメインを指定します。

ドメイン名には、少なくとも"."が2つ含まれていなければなりません。

なお、IPv6 アドレスに対応したドメイン名も指定できます。

(例)

a.example.com と b.example.com の2つのホストを設定している場合、このディレクティブで.example.com と指定すると、2つのホストのどちらにアクセスしても優先度処理が行われます。

記述できる場所

httpsd.conf、<VirtualHost>

2.3.106 QOSCookieExpires

流量制限機能に使用する Cookie の有効時間を指定します。

説明

流量制限機能に使用する Cookie の有効時間を秒単位で指定します。このディレクティブで指定した値は、HWS 作成モードで使用され、ユーザー作成モードでは使用されません。

なお、流量制限機能を使用するには、`mod_hws_qos` モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

メモ

HWS 作成モードは Web サーバが作成する Cookie を用いて、またユーザー作成モードは Web サーバ以外で作成された Cookie を用いて、セッション管理を実施します。HWS 作成モードとユーザー作成モードは、`QOSCookieName` ディレクティブで指定します。

書式

```
QOSCookieExpires 値
```

指定できる値

0-86400

注意事項

- `QOSCookieName` ディレクティブを特定のブロックに指定した場合、上位に指定されている `QOSCookieName` ディレクティブは継承しません。

(例)

```
QOSCookieName Cookie1 hws
<Location /loc1>
  QOSCookieName Cookie2 user
</Location>
```

この場合、`"/loc1"` から始まるリクエストでは、Cookie 名 `Cookie2` の指定が有効になります。`"/loc1"` 以外から始まるリクエストでは、Cookie 名 `Cookie1` の指定が有効になります。

- `QOSCookieName` ディレクティブを複数指定する場合は、Cookie 名を重複させないでください。重複している場合は、起動エラーになります。

(例)

```
QOSCookieName Cookie1 hws
QOSCookieName Cookie1 user
```

この場合、Cookie 名が重複しているため起動エラーになります。

- HWS 作成モードの `QOSCookieName` ディレクティブを複数指定した場合は、後ろに指定した方が有効になります。

(例)

```
QOSCookieName Cookie1 hws
QOSCookieName Cookie2 hws
```

この場合、Cookie 名 Cookie1 の指定は無効になり、Cookie2 の指定が有効になります。

デフォルト値

定義項目の省略

300

記述できる場所

httpsd.conf、<VirtualHost>、<Location>

2.3.107 QOSCookieName

流量制限機能に使用する Cookie 名を指定します。

説明

流量制限機能に使用する Cookie 名を指定します。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

```
QOSCookieName Cookie名 [{hws|user}]
```

指定できる値

Cookie 名

流量制限機能に使用する Cookie 名を指定します。*Cookie* 名にセミコロン、コンマ、スペースは使用できません。

ホスト間および URL 間でそれぞれ異なる Cookie を利用したセッション管理を行う場合は、別の Cookie 名を指定する必要があります。

hws

Web サーバが作成する Cookie を用いて、セッション管理を実施します。これを HWS 作成モードと呼びます。

user

Web サーバ以外の外部モジュールなどで作成された Cookie を用いて、セッション管理を実施します。これをユーザー作成モードと呼びます。

注意事項

- QOSErrorName ディレクティブを特定のブロックに指定した場合、上位に指定されているQOSErrorName ディレクティブは継承しません。

(例)

```
QOSErrorName Cookie1 hws
<Location /loc1>
  QOSErrorName Cookie2 user
</Location>
```

この場合、"/loc1"から始まるリクエストでは、Cookie 名Cookie2 の指定が有効になります。"/loc1"以外から始まるリクエストでは、Cookie 名Cookie1 の指定が有効になります。

- QOSErrorName ディレクティブを複数指定する場合は、Cookie 名を重複させないでください。重複している場合は、起動エラーになります。

(例)

```
QOSErrorName Cookie1 hws
QOSErrorName Cookie1 user
```

この場合、Cookie 名が重複しているため起動エラーになります。

- HWS 作成モードのQOSErrorName ディレクティブを複数指定した場合は、後ろに指定した方が有効になります。

(例)

```
QOSErrorName Cookie1 hws
QOSErrorName Cookie2 hws
```

この場合、Cookie 名Cookie1 の指定は無効になり、Cookie2 の指定が有効になります。

デフォルト値

定義項目の省略

hws

記述できる場所

httpsd.conf、<VirtualHost>、<Location>

2.3.108 QOSCookieSecure

SSL によるアクセス時だけに Cookie を送信させるかどうかを指定します。

説明

クライアントに対し、SSL によるアクセス時だけに Cookie を送信させるよう設定します。この値は、HWS 作成モードで使用され、ユーザー作成モードでは使用されません。Cookie の確認は SSL の暗号処理の終了後であることに注意してください。

(例)

SSL が有効であるホストと無効であるホストを設定している場合、このディレクティブを設定すると、SSL が有効なホストへのアクセスだけ Cookie が送信されます。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

```
QOSCookieSecure {on|off}
```

指定できる値

on

SSL によるアクセス時だけ、クライアントに Cookie を送信させるよう設定します。

off

SSL 以外によるアクセス時にも、クライアントに Cookie を送信させるよう設定します。

デフォルト値

定義項目の省略

Off

記述できる場所

httpsd.conf、<VirtualHost>、<Location>

2.3.109 QOSCookieServers

Cookie を送信してきたリクエストだけを処理するときのサーバスレッド数を指定します。

説明

リクエスト待ち状態のサーバスレッド数が減少した場合に、Cookie を送信してきたリクエストだけを処理するときの、サーバスレッド数を指定します。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

QOSCookieServers <i>値</i>

指定できる値

0-ThreadsPerChild ディレクティブ指定値

デフォルト値

定義項目の省略

10

記述できる場所

httpsd.conf、<VirtualHost>、<Location>

2.3.110 QOSRedirect

流量制限機能によって拒否したリクエストをリダイレクトさせるときに指定します。

説明

流量制限機能によって処理が拒否された場合に、クライアントからのリクエストを指定されたパスにリダイレクトさせるときに指定します。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

QOSRedirect <i>旧パス</i> <i>新パス</i>

指定できる値

旧パス

スラッシュから始まるリクエスト URL のパスを指定します。ただし、旧パスには、?以降（問い合わせ文字列）を指定できません。

旧パスでリクエストを受けた場合、ステータスコード 302 と Location ヘッダーに新パスを設定したレスポンスを返します。レスポンスをカスタマイズすることはできません。

新パス

”プロトコル名://ホスト名[:ポート番号]”を含む URL のパスを指定します。また、URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

記述できる場所

httpsd.conf、<VirtualHost>、<Location>

2.3.111 QOSRejectionServers

受信したすべてのリクエストを拒否するときのサーバスレッド数を指定します。

説明

リクエスト待ち状態のサーバスレッド数が減少し、受信したすべてのリクエストを拒否するようになるときの、サーバスレッド数を指定します。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

QOSRejectionServers 値

指定できる値

0-ThreadsPerChild ディレクティブ指定値

デフォルト値

定義項目の省略

1

記述できる場所

httpsd.conf、<VirtualHost>、<Location>

2.3.112 QOSResponse

流量制限機能によって処理が拒否された場合にレスポンスとして返すコンテンツを指定します。

説明

流量制限機能によって処理が拒否された場合に、ステータスコード 503 とともに返送するコンテンツを指定します。コンテンツはサーバプロセス内にキャッシュされるため、変更する場合にはサーバの再起動が必要です。

なお、流量制限機能を使用するには、mod_hws_qos モジュールの組み込みが必要です。

```
LoadModule hws_qos modules/mod_hws_qos.so
```

書式

```
QOSResponse {file [MIMEタイプ] ファイル名 |message テキスト}
```

指定できる値

file

指定したファイルを、指定した *MIME* タイプで返送します。*MIME* タイプを省略したときはtext/html が設定されます。また、ファイル名には、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が指定できます。

message

指定したテキストを返送します。テキストは先頭に”を記述して文字列を指定します。*MIME* タイプにはtext/html が設定されます。

記述できる場所

httpsd.conf、<VirtualHost>、<Location>*Location*

記述例

```
QOSResponse file "text/html; charset=ISO-8859-1"htdocs/busy.html  
QOSResponse message "Server busy."
```

2.3.113 ReadmeName

ディレクトリーインデックス表示時の Readme として付けるコメントを記述したファイルを指定します。

説明

ディレクトリーインデックス表示時の Readme として付けるコメントを記述したファイルのファイル名 (パス情報なし) を指定します。HTML またはプレーンテキストで記述できます。ただし、AddType ディレクティブまたはTypesConfig ディレクティブで指定したファイルで、MIME タイプが正しく定義されている必要があります。プレーンテキストでコメントを作成した場合、ディレクトリーインデックス表示の HTML には<PRE>タグが追加されます。

書式

```
ReadmeName ファイル名
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

Indexes レベル

記述例

```
ReadmeName README.html
```

ディレクトリー下のREADME.html ファイルの内容を表示します。

2.3.114 Redirect

クライアントからのリクエストをリダイレクトする場合に指定します。

説明

旧パスに対するクライアントからのリクエストを、新パスに再リクエスト (リダイレクト) する場合に指定します。

旧パスでリクエストを受けた場合、指定したステータスコードと Location ヘッダーに新パスを設定した応答を返します。通常、300 番台のステータスコードを受けた Web ブラウザーは、自動的に Location ヘッダーに指定されたアドレスに対してリダイレクトします。

Redirect ディレクティブでは、特定のファイルへのリクエストを特定のファイルへリダイレクトするか、特定のディレクトリー下の、任意のパスへのリクエストを特定のディレクトリー下の、同名パスへリダイ

レクトする指定ができます。特定のディレクトリー下の、任意のパスへのリクエストを、特定のファイルヘリダイレクトしたい場合はRedirectMatch ディレクティブを使用してください。

書式

```
Redirect [{permanent|temp|seeother|gone|ステータスコード}] /旧パス 新パス
```

指定できる値

permanent

ステータスコード 301 Moved Permanently を応答する場合に指定します。

temp

ステータスコード 302 Found を応答する場合に指定します。

seeother

ステータスコード 303 See Other を応答する場合に指定します。

gone

ステータスコード 410 Gone を応答する場合に指定します。新パスは指定できません。

ステータスコード

指定したステータスコードを応答する場合に指定します。

ただし、300 番台以外を指定する場合、新パスは指定できません。

/旧パス

スラッシュから始まるリクエスト URL のパスを指定します。ただし、/旧パスには、?以降（問い合わせ文字列）を指定できません。

次のディレクティブ指定値と重複する/旧パスは指定できません。

- ProxyPass のパス名

例えば、次のような指定はできません。

```
Redirect temp /aaa/bbb/ http://aaa.example.com/  
ProxyPass /aaa/ http://aaa.example.com/
```

新パス

”プロトコル名://ホスト名[:ポート番号]”を含む URL のパスを指定します。また、新パスに指定する URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
Redirect temp /index.html http://ホスト名:port番号/default.html
```

/index.html に対するリクエストを、ステータスコード 302 で"http://ホスト名:port番号/default.html"にリダイレクトします。

2.3.115 RedirectMatch

正規表現で記述した条件を満たすクライアントからのリクエストをリダイレクトする場合に指定します。

説明

正規表現で記述した条件を満たすパスに対するクライアントからのリクエストを、新パスに再リクエスト (リダイレクト) する場合に指定します。

正規表現で括弧 () を使用してグループ化している場合、その *i* 番目のグループの表現にマッチした文字列を、新パスで $\$i$ を使用して参照できます。*i* には 1 から 9 までの数字を指定します。正規表現で記述した条件を満たすパスへのリクエストを受信した場合に、指定したステータスコードと、新パスを設定した Location ヘッダーを応答します。通常、300 番台のステータスコードを受けた Web ブラウザーは、自動的に Location ヘッダーに指定されたアドレスに対して再リクエスト (リダイレクト) します。

書式

```
RedirectMatch [{permanent|temp|seeother|gone|ステータスコード}] 正規表現 新パス
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

指定できる値

permanent

ステータスコード 301 Moved Permanently を応答する場合に指定します。

temp

ステータスコード 302 Found を応答する場合に指定します。

seeother

ステータスコード 303 See Other を応答する場合に指定します。

gone

ステータスコード 410 Gone を応答する場合に指定します。新パスは指定できません。

ステータスコード

指定したステータスコードを応答する場合に指定します。

ただし、300 番台以外を指定する場合、**新パス**は指定できません。

正規表現

スラッシュから始まるリクエスト URL の旧パスを正規表現で指定します。ただし、旧パスには、?以降（問い合わせ文字列）を指定できません。

次のディレクティブ指定値と重複する旧パスは指定できません。

- ProxyPass のパス名

例えば、次のような指定はできません。

```
RedirectMatch ^/aaa/bbb/(.*) http://aaa.example.com/$1
ProxyPass /aaa/ http://aaa.example.com/
```

新パス

”プロトコル名://ホスト名[:ポート番号]”を含む URL のパスを指定します。また、新パスに指定する URL には、IPv6 アドレスまたは IPv6 アドレスに対応したホスト名も指定できます。

上書き許可

FileInfo レベル

記述例

(例 1)

```
RedirectMatch ^/other/ http://www.example.com/
```

/other/で始まるすべてのリクエストを、ステータスコード 302 で”http://www.example.com/”にリダイレクトします。

(例 2)

```
RedirectMatch permanent ^/old/(.*) http://www.example.com/new/$1
```

/old/ファイル名”に対するリクエストを、ステータスコード 301 で”http://www.example.com/new/ファイル名”にリダイレクトします。

2.3.116 RequestHeader

リクエストヘッダーをカスタマイズする場合に指定します。

説明

クライアントから受信したヘッダー値をカスタマイズする場合に指定します。

カスタマイズ機能を使用するためにはモジュールの組み込みが必要です。

なお、リクエストヘッダーをカスタマイズするにはmod_headers モジュールの組み込みが必要です。

```
LoadModule headers_module modules/mod_headers.so
```

書式

```
RequestHeader {{set|append|add}ヘッダー ヘッダー値[env=[!]環境変数]|unset ヘッダー}
```

指定できる値

set

ヘッダーを設定します。ヘッダーがある場合は、指定したヘッダー値に書き換えます。

append

存在するヘッダーにヘッダー値を追加します。存在するヘッダー値との間は、コンマで区切られます。ヘッダーがない場合は、ヘッダーを設定します。

add

ヘッダーがあっても、別の行にヘッダーを設定します。同じヘッダーを複数行設定する場合に使用します。

unset

指定したヘッダーがある場合、そのヘッダーをすべて削除します。

env=環境変数

指定した環境変数が設定されている場合に、RequestHeader ディレクティブで指定した内容を実行します。

env=!環境変数

指定した環境変数が設定されていない場合に、RequestHeader ディレクティブで指定した内容を実行します。

ヘッダー値

文字だけから成る文字列、フォーマット指示子を含む文字列または両方から成る文字列を指定できます。ヘッダー値に空白がある場合は、”（ダブルクォーテーション）で囲む必要があります。フォーマット指示子を次に示します。

フォーマット指示子	意味
%t	リクエストを受け取った時刻を、1970年1月1日0時0分0秒（GMT: Greenwich Mean Time）から経過した時間で表示する。単位はマイクロ秒。先頭には"t="が付けられる。
%D	リクエスト処理に掛かった時間を表示する。単位はマイクロ秒。先頭には"D="が付けられる。
%{env_name}e	環境変数 env_name の値。

記述できる場所

httpd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
RequestHeader set Host www.example.com
```

2.3.117 RequestReadTimeout

リクエスト受信開始からリクエストヘッダー受信完了まで、およびリクエストボディの受信開始から完了までのタイムアウト時間を指定します。

説明

リクエスト受信開始からリクエストヘッダー受信完了まで、およびリクエストボディの受信開始から完了までのタイムアウト時間を秒単位で指定します。

データの送信が遅いリクエストに対して、サーバのリソースを長時間専有されることを避ける場合に有効です。

なお、タイムアウトを設定するにはmod_reqtimeout モジュールの組み込みが必要です。

```
LoadModule reqtimeout_module modules/mod_reqtimeout.so
```

また、モジュールトレースの出力の抑止を同時に設定することを推奨します。

```
HWSSuppressModuleTrace mod_reqtimeout.c
```

書式

```
RequestReadTimeout タイプ=時間 [タイプ=時間]
```

指定できる値

タイプ

header とbody を指定できます。

- header

リクエスト受信開始からリクエストヘッダー受信完了までの経過時間を監視する場合に指定します。

- body

リクエストボディの受信開始から完了までの経過時間を監視する場合に指定します。

時間

リクエスト受信開始からリクエストヘッダー受信完了まで、およびリクエストボディの受信開始から完了までのタイムアウト時間を秒単位で指定します。

指定できる値は、0~2147483647 です。0 を指定するとタイムアウトを設定しません。

デフォルト値

定義項目の省略

20

記述できる場所

httpsd.conf、<VirtualHost>

記述例

(例 1)

```
RequestReadTimeout header=10 body=30
```

リクエスト受信開始からリクエストヘッダー受信完了までのタイムアウト時間を 10 秒、リクエストボディの受信開始から完了までのタイムアウト時間を 30 秒に設定します。

(例 2)

```
RequestReadTimeout header=0 body=0
```

タイムアウトを設定しません。

2.3.118 Require

アクセス制限を定義する場合に指定します。

説明

AuthName ディレクティブ、AuthType ディレクティブ、AuthUserFile ディレクティブ（またはAuthGroupFile ディレクティブ）と一緒に指定し、アクセス制限を定義します。

書式

```
Require {user ユーザー名 [ユーザー名 ...]  
|group グループ名 [グループ名 ...]|valid-user}
```

指定できる値

user

AuthUserFile ディレクティブで指定したパスワードファイルに登録されているユーザーのうち、ユーザー名で指定したユーザーだけアクセスできます。

group

AuthGroupFile ディレクティブで指定したグループファイルに登録されているグループ名で指定したグループに属するユーザーだけがアクセスできます。

valid-user

AuthUserFile ディレクティブで指定したパスワードファイルに登録されているすべてのユーザーがアクセスできます。

記述できる場所

<Directory>、.htaccess

上書き許可

AuthConfig レベル

2.3.119 Satisfy

コンテンツへのアクセスがユーザー認証とホスト名または IP アドレスの両方によって制限されている場合にその関係を指定します。

説明

コンテンツへのアクセスが、ユーザー認証 (AuthUserFile、Require ディレクティブなどを指定) とホスト名または IP アドレス (Allow from、Deny from ディレクティブなどを指定) の両方によって制限されている場合にその関係を設定します。

書式

```
Satisfy {any|all}
```

指定できる値

any

そのどちらかの条件を満たしていれば、コンテンツへのアクセスを許可します。

all

そのどちらの条件も満たさなければ、コンテンツへのアクセスを禁止します。

デフォルト値

定義項目の省略

all

記述できる場所

<Directory>、.htaccess

2.3.120 Script

特定のメソッドに対する CGI スクリプトを実行する場合に指定します。

説明

指定されたメソッドによるリクエストがあった場合に *CGI* スクリプト名で示すスクリプトを実行します。

指定できるメソッド

GET、POST、PUT、DELETE

メソッドは大文字、小文字を区別します。

ただし、GET メソッドの場合、スクリプトは問い合わせ引数があるときだけ（例えば、/foo.html?bar）呼ばれます。

書式

```
Script メソッド CGIスクリプト名
```

記述できる場所

httpd.conf、<VirtualHost>、<Directory>

記述例

```
Script POST /cgi-bin/search
```

2.3.121 ScriptAlias

CGI プログラムを実行する URL と CGI プログラムが配置されているディレクトリー名を指定します。

説明

Web ブラウザーから URL で指定された CGI プログラム実行のリクエストに対して、実行する CGI プログラムのあるディレクトリー名を指定します。

書式

```
ScriptAlias URL ディレクトリー名
```

指定できる値

URL

CGI プログラムを実行する URL を指定します。

次のディレクティブ指定値と重複する URL は指定できません。

- ProxyPass のパス名

例えば、次のような指定はできません。

```
ScriptAlias /aaa/bbb/ C:/alias/  
ProxyPass /aaa/ http://aaa.example.com/
```

ディレクトリー名

CGI プログラムが配置されているディレクトリー名を指定します。

ディレクトリー名は、絶対パスで指定してください。

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
ScriptAlias /cgi-bin/ "Application Serverのインストールディレクトリー/httpsd/cgi-bin/"
```

2.3.122 ScriptAliasMatch

CGI プログラムを実行する URL に対する正規表現と CGI プログラムが配置されているディレクトリー名を指定します。

説明

Web ブラウザーから指定された CGI プログラム実行要求の URL が正規表現で記述した条件を満たす場合、指定した新パスの CGI プログラムを実行します。正規表現で括弧 () を使用してグループ化している場合、その *i* 番目のグループの表現にマッチした文字列を、新パスで \$*i* を使用して参照できます。*i* には 1 から 9 までの数字を指定します。

新パスは、絶対パスで指定してください。また、新パスの文字として、\$または&を含める場合は、その文字の前に¥を付加してください。なお、\$i を指定する際には、\$の前に¥を付加する必要はありません。

次のディレクティブ指定値と重複する正規表現は指定できません。

- ProxyPass のパス名

例えば、次のような指定はできません。

```
ScriptAliasMatch ^/aaa/bbb/(.*) C:/alias/$1
ProxyPass /aaa/ http://aaa.example.com/
```

書式

```
ScriptAliasMatch 正規表現 新パス
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
ScriptAliasMatch ^/cgi-bin/(.*)
"Application Serverのインストールディレクトリー/httpsd/cgi-bin/$1"
```

2.3.123 ScriptInterpreterSource

CGI スクリプトの実行に使用されるインタープリターを指定します。

説明

CGI スクリプトの実行に使用されるインタープリターを定義します。

書式

```
ScriptInterpreterSource {registry|script}
```

指定できる値

registry

レジストリーが検索され、拡張子に関連づけられているプログラムがインタープリターとして使用されます。

script

スクリプト内の#!行で指定されたインタープリターが使用されます。

デフォルト値

定義項目の省略

script

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.124 ScriptLog

CGI スクリプトのエラーログ出力先を指定します。

説明

CGI スクリプトのエラーログ出力先のファイルを指定します。

書式

ScriptLog ファイル名

指定できる値

ファイル名

CGI スクリプトのエラーログの出力先を指定します。

ファイル名には、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が指定できます。

記述できる場所

httpsd.conf

2.3.125 ScriptLogBuffer

リクエストのボディ部のログを採取する場合の最大値を指定します。

説明

PUT、POST メソッドによるリクエストのボディ部のログを採取する場合の最大値をバイト単位で指定します。ScriptLog ディレクティブでエラーログ出力先のファイルを指定した場合だけ、この指定は有効になります。

このディレクティブでの指定値分の領域が、リクエスト処理中に確保されます。そのため、大きい値を指定すると、メモリー確保失敗となって、Web サーバが終了する場合があります。デフォルト値または必要最小限の値を指定することを推奨します。

書式

```
ScriptLogBuffer バッファ数
```

指定できる値

0~2147483647

デフォルト値

定義項目の省略

1024

記述できる場所

httpsd.conf

2.3.126 ScriptLogLength

CGI スクリプトのエラーログファイルの最大サイズを指定します。

説明

CGI スクリプトのエラーログファイルの最大サイズをバイト単位で指定します。ScriptLog ディレクティブでエラーログ出力先のファイルを指定した場合だけ指定が有効になります。

書式

```
ScriptLogLength ファイルサイズ
```

指定できる値

0~2147483647

デフォルト値

定義項目の省略

10385760

記述できる場所

httpsd.conf

2.3.127 SendBufferSize

Web サーバの TCP 送信バッファサイズをバイト数で指定します。

説明

Web サーバの TCP 送信バッファサイズをバイト数で指定します。0 を指定した場合、OS のデフォルト値が使用されます。

高速のネットワーク環境では、OS のデフォルト値よりも大きな値を設定することで、レスポンス送信の性能が向上する場合があります。

書式

```
SendBufferSize 送信バッファサイズ
```

指定できる値

0、512–2147483647

デフォルト値

定義項目の省略

0

記述できる場所

httpsd.conf

記述例

```
SendBufferSize 131072
```

2.3.128 ServerAdmin

サーバ管理者の E-Mail アドレスを指定します。

説明

サーバ管理者の E-Mail アドレスを指定します。ServerSignature ディレクティブで E-Mail を指定する場合は、必ず指定してください。

書式

```
ServerAdmin E-Mailアドレス
```

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
ServerAdmin www-admin@server.example.com
```

2.3.129 ServerAlias

サーバ名に基づくバーチャルホストで使用するホスト名の別名を指定します。

説明

サーバ名に基づくバーチャルホストで使用するホスト名 (ServerName) の別名を指定します。IPv6 アドレスに対応したホスト名も指定できます。

書式

```
ServerAlias ホスト名 [ホスト名 ...]
```

記述できる場所

<VirtualHost>

2.3.130 ServerName

Web サーバのサーバ名およびポート番号を指定します。

説明

Web サーバのサーバ名およびポート番号を指定します。

UseCanonicalName ディレクティブ指定値に従い、イメージマップの利用または末尾をスラッシュ (/) で閉じないディレクトリ指定のリクエストなど、Web サーバでリダイレクトが指示された場合のリダイレクト先として Location ヘッダーに設定されクライアントに返信されるため、クライアントからアクセスできるサーバ名を指定しなければなりません。このディレクティブの指定は必須です。

書式

```
ServerName サーバ名[:ポート番号]
```

指定できる値

サーバ名

FQDN（完全修飾ドメイン名）または IP アドレスで指定します。

サーバ名には、IPv6 アドレスまたは IPv6 アドレスに対応した FQDN も指定できます。

IPv6 アドレスを指定し、かつポート番号を指定する場合は、IPv6 アドレスを [] で囲んでください。

ポート番号

Web サーバのポート番号を指定します。

ポート番号を省略した場合は、デフォルトのポート番号（SSL が無効の場合は 80、SSL が有効の場合は 443）が設定されます。

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
ServerName www.example.com
ServerName 2001::123:4567:89ab:cdef
ServerName [2001::123:4567:89ab:cdef]
ServerName [2001::123:4567:89ab:cdef]:8080
```

2.3.131 ServerPath

サーバ名に基づくバーチャルホストで、Host ヘッダーの代わりにパス名を利用して各ホストに接続する場合に指定します。

説明

サーバ名に基づくバーチャルホストで、Host ヘッダーの代わりにパス名を利用して各ホストに接続する場合に指定します。

書式

```
ServerPath パス名
```

記述できる場所

<VirtualHost>

2.3.132 ServerRoot

サーバのルートディレクトリーを指定します。

説明

サーバのルートディレクトリーを絶対パスで指定します。

書式

```
ServerRoot ディレクトリー名
```

デフォルト値

定義項目の省略

*Application Server*のインストールディレクトリー¥httpsd

記述できる場所

httpsd.conf

記述例

```
ServerRoot "Application Serverのインストールディレクトリー/httpsd"
```

2.3.133 ServerSignature

Web サーバが作成するエラーメッセージなどのコンテンツのフッターに署名するかどうかを指定します。

説明

Web サーバが作成するエラーメッセージなどのコンテンツのフッターに署名するかどうかを指定します。

書式

```
ServerSignature {On|Off|Email}
```

指定できる値

On

ServerTokens ディレクティブに従った文字列（「Hitachi Web Server」やバージョン番号など）およびUseCanonicalName ディレクティブ指定値に従ったサーバ名とポート番号を表示します。

```
Hitachi Web Server 10-00 at www.example.com Port 80
```

なお、On を指定した場合、ServerName ディレクティブに指定した IPv6 アドレスまたは IPv6 アドレスに対応したホスト名を表示できます。

Off

コンテンツのフッターに署名を表示しません。

Email

On を指定した場合の表示に加えServerAdmin ディレクティブの指定値をmailto タグで追加します。

デフォルト値

定義項目の省略

Off

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

記述例

```
ServerSignature On
```

2.3.134 ServerTokens

Server ヘッダーのフォーマットを指定します。

説明

HTTP レスポンスヘッダーの Server ヘッダーのフォーマットを設定します。

メモ

OS 種別には、Unix、Win32 または Win64 が設定されます。Server ヘッダーの値がどのように利用されるかはクライアントの仕様によります。

書式

```
ServerTokens {Minimal|OS|Full|ProductOnly}
```

指定できる値

指定できる Server ヘッダーの値を次に示します。

Minimal

Hitachi Web Server バージョン番号

OS

Hitachi Web Server バージョン番号 (OS種別)

Full

Hitachi Web Server バージョン番号 (OS種別) 付加PPで設定された情報

ProductOnly

Hitachi Web Server

デフォルト値

定義項目の省略

Full

記述できる場所

httpsd.conf

記述例

```
ServerTokens Full
```

2.3.135 SetEnv

CGI スクリプトに渡す環境変数の値を指定します。

説明

CGI スクリプトに任意の環境変数を渡す場合に設定する環境変数の値を指定します。このディレクティブを複数指定する場合、同じ環境変数に異なる値は指定できません。

書式

```
SetEnv 環境変数 値
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
SetEnv MY_ENV myenv
```

2.3.136 SetEnvIf

クライアントからのリクエストを基に環境変数を定義する場合に指定します。

説明

クライアントからのリクエスト値が正規表現で表した条件を満たす場合、指定した環境変数を設定します。

書式

```
SetEnvIf リクエスト値 正規表現 環境変数[=値] [環境変数[=値] ...]
```

指定できる値

リクエスト値

HTTP リクエストヘッダーか次の表に示す値を指定できます。先に指定された環境変数をリクエスト値として指定することで環境変数の検査ができます。ただし、この場合の環境変数は、HTTP リクエストヘッダーにも次の表に示す指定値にも一致していない必要があります。

リクエスト値	意味
Remote_Addr	クライアントの IP アドレス
Remote_Host	クライアントのホスト名 (リクエストに設定されている場合だけ)

リクエスト値	意味
Request_Protocol	リクエストのプロトコル (HTTP/1.1 など)
Request_Method	リクエストのメソッド名 (GET、POST、HEAD など)
Request_URI	リクエストの URI
Server_Addr	リクエストを受信したサーバの IP アドレス

このディレクティブを複数指定する場合、同じリクエスト値は複数指定できません。

なお、リクエスト値にRemote_Host を指定した場合、正規表現には IPv6 アドレスに対応したホスト名も指定できます。また、IPv6 を使用した接続に対しては、Remote_Addr とServer_Addr のリクエスト値は使用できません。Remote_Addr とServer_Addr を使用したい場合は、HWSSetEnvIfIPv6 ディレクティブで設定してください。

正規表現

条件を指定します。

環境変数

クライアントからのリクエスト値が正規表現で表した条件を満たす場合に設定する環境変数を指定します。

環境変数の前に!を付けると、その環境変数の設定を解除します。

値

環境変数の値を指定します。

設定する値のデフォルト値は1 です。

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

(例 1)

```
SetEnvIf User-Agent "Mozilla.*" SETENVIF_USER_AGENT=Mozilla
```

(例 2)

```
SetEnvIf Request_URI "¥.(gif)|(jpg)$" request_is_image
```

(例 3)

IPv4 を使用した接続のうち、特定のクライアントに対して環境変数を設定する場合は、次のように指定します。

```
Listen 123.123.123.123:80
Listen [2001::123:4567:89ab:cdef]:80
<VirtualHost 123.123.123.123:80>
    SetEnvIf Remote_Addr ^234¥.234¥.234¥.234$ IPV4_CLIENT
</VirtualHost>
```

2.3.137 SetEnvIfNoCase

クライアントからのリクエストを基に環境変数を定義する場合に指定します。

説明

クライアントからのリクエスト値が正規表現で表した条件を満たす場合、指定した環境変数を設定します。

ただし、このディレクティブでは、正規表現の大文字、小文字の区別をしません。また、このディレクティブを複数指定する場合、同じリクエスト値は複数指定できません。

書式

```
SetEnvIfNoCase リクエスト値 正規表現 環境変数[=値] [環境変数[=値] ...]
```

指定できる値

リクエスト値

HTTP リクエストヘッダーか次の表に示す値を指定できます。先に指定された環境変数をリクエスト値として指定することで環境変数の検査ができます。ただし、この場合の環境変数は、HTTP リクエストヘッダーにも次の表に示す指定値にも一致していない必要があります。

リクエスト値	意味
Remote_Addr	クライアントの IP アドレス
Remote_Host	クライアントのホスト名 (リクエストに設定されている場合だけ)
Request_Protocol	リクエストのプロトコル (HTTP/1.1 など)
Request_Method	リクエストのメソッド名 (GET、POST、HEAD など)
Request_URI	リクエストの URI
Server_Addr	リクエストを受信したサーバの IP アドレス

このディレクティブを複数指定する場合、同じリクエスト値は複数指定できません。

なお、リクエスト値にRemote_Hostを指定した場合、正規表現にはIPv6アドレスに対応したホスト名も指定できます。また、IPv6を使用した接続に対しては、Remote_AddrとServer_Addrのリクエスト値は使用できません。Remote_AddrとServer_Addrを使用したい場合は、HWSSetEnvIfIPv6ディレクティブで設定してください。

正規表現

条件を指定します。

ただし、このディレクティブでは、正規表現の大文字、小文字の区別をしません。

環境変数

クライアントからのリクエスト値が正規表現で表した条件を満たす場合に設定する環境変数を指定します。

環境変数の前に!を付けると、その環境変数の設定を解除します。

値

環境変数の値を指定します。

設定する値のデフォルト値は1です。

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.138 SetHandler

ある特定のハンドラーに関連づける場合に指定します。

説明

指定した<Directory>またはアクセス制御ファイルの範囲すべてのリクエストをハンドラー名で指定したハンドラーに関連づける場合、指定します。ハンドラー名としてnoneを指定すると、それまでのSetHandlerディレクティブの設定が無効になります。

書式

SetHandler ハンドラー名

記述できる場所

<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.139 SSLBanCipher

指定した暗号種別でのアクセスを拒否し、ステータスコード 403 Forbidden を応答する場合に指定します。

説明

指定した暗号種別でのアクセスを拒否し、クライアントにステータスコード 403 Forbidden を応答します。

暗号種別を次に示します。

暗号種別	鍵交換方式	認証方式	対称鍵暗号方式	暗号鍵サイズ (bit)	メッセージ認証アルゴリズム
RC4-MD5	RSA	RSA	RC4	128	MD5
RC4-SHA	RSA	RSA	RC4	128	SHA
AES128-SHA	RSA	RSA	AES	128	SHA
AES128-SHA256	RSA	RSA	AES	128	SHA256
DES-CBC3-SHA	RSA	RSA	DES	168	SHA
AES256-SHA	RSA	RSA	AES	256	SHA
AES256-SHA256	RSA	RSA	AES	256	SHA256

書式

```
SSLBanCipher 暗号種別 [暗号種別 ...]
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.140 SSLCACertificateFile

CA の証明書のファイルを指定します。

説明

SSL でサーバ認証およびクライアント認証する場合、CA（認証局）の公開鍵（PEM 形式）のファイル名を指定します。複数の証明書ファイルを連結させて、1 つのファイルに複数の証明書が混在できます。

ファイル名は、絶対パスで指定してください。

- サーバ認証時の利用
チェーンした CA で発行されたサーバ証明書を使用して運用する場合、チェーン CA の証明書を設定します。
- クライアント認証時の利用
クライアント証明書を発行した CA の証明書を設定します。チェーンされたクライアント証明書の場合、チェーン CA の証明書も設定します。

書式

```
SSLCACertificateFile ファイル名
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLCACertificateFile "Application Serverのインストールディレクトリー  
/httpsd/conf/ssl/cacert/anycert.pem"
```

2.3.141 SSLCertificateFile

SSL で認証する場合、Web サーバの証明書のファイルを指定します。

説明

SSL で認証する場合、Web サーバの証明書（PEM 形式）のファイル名を指定します。

ファイル名は、絶対パスで指定してください。

書式

```
SSLCertificateFile ファイル名
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLCertificateFile "Application Serverのインストールディレクトリー  
/httpsd/conf/ssl/server/httpsd.pem"
```

2.3.142 SSLCertificateKeyFile

SSL で認証する場合、Web サーバの秘密鍵のファイルを指定します。

説明

SSL で認証する場合、Web サーバの秘密鍵のファイル名を指定します。

ファイル名は、絶対パスで指定してください。

書式

```
SSLCertificateKeyFile ファイル名
```

デフォルト値

定義項目の省略

SSLCertificateFile ディレクティブ指定値

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLCertificateKeyFile "Application Serverのインストールディレクトリー  
/httpsd/conf/ssl/server/httpsdkey.pem"
```

2.3.143 SSLCertificateKeyPassword

パスワード保護されているサーバ秘密鍵のパスワードを格納するファイルを指定します。

説明

パスワード保護をされているサーバ秘密鍵のパスワードを格納しておくファイルのパス名を指定します。パスワード格納ファイルは、`sslpasswd` コマンドによって作成します。パス名は絶対パスまたはServerRoot ディレクティブからの相対パスで指定します。

書式

```
SSLCertificateKeyPassword パス名
```


記述できる場所

httpsd.conf、<VirtualHost>

2.3.144 SSLCRLAuthoritative

CRL の次回発行日を経過した場合の動作を指定します。

説明

SSL クライアント認証時に使用する CRL の次回発行日の扱いについて指定します。

書式

```
SSLCRLAuthoritative {On|Off}
```

指定できる値

On

SSL クライアント認証時、クライアント証明書に対応する CRL の次回発行日を過ぎていた場合、Web サーバは認証に失敗したとして、クライアントとの接続を拒否します。CRL の正しい運用が必要です。

Off

CRL の次回発行日を無視します。次回発行日を過ぎていても CRL は有効であると扱うため、CRL に登録されていなければ、クライアントは接続できます。セキュリティレベルは下がりますが、CRL を正しく運用しなかったときでも、最低限のセキュリティを維持して、サービスを継続できます。

デフォルト値

定義項目の省略

On

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLCRLAuthoritative On
```

CRL の次回発行日を過ぎている場合、その CRL を発行した CA が発行した証明書を持つクライアントのアクセスはすべて拒否します。

2.3.145 SSLCRLDERPath

DER 形式の CRL を格納するディレクトリーを指定します。

説明

DER 形式の CRL を格納するディレクトリーを絶対パスで指定します。指定したディレクトリーに必要な CRL を格納して、Web サーバを起動または再起動すると、SSL でのクライアント認証時に CRL を適用できます。SSLCRLPEMPath ディレクティブで指定したディレクトリー内の CRL も含め、同じ CA から発行された CRL を複数格納している場合、発行日が最新の CRL を適用します。ディレクトリー内に DER 形式の CRL 以外のファイルがある場合、Web サーバは起動しません。

書式

```
SSLCRLDERPath パス名
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLCRLDERPath "Application Serverのインストールディレクトリー/httpsd/conf/ssl/crl/DER"
```

DER 形式の CRL ファイルを格納しているディレクトリーを指定します。

2.3.146 SSLCRLPEMPath

PEM 形式の CRL を格納するディレクトリーを指定します。

説明

PEM 形式の CRL を格納するディレクトリーを絶対パスで指定します。指定したディレクトリーに必要な CRL を格納して、Web サーバを起動または再起動すると、SSL でのクライアント認証時に CRL を適用できます。SSLCRLDERPath ディレクティブで指定したディレクトリー内の CRL も含め、同じ CA から発行された CRL を複数格納している場合、発行日が最新の CRL を適用します。ディレクトリー内に PEM 形式の CRL 以外のファイルがある場合、Web サーバは起動しません。

書式

```
SSLCRLPEMPath パス名
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLCRLPEMPath "Application Serverのインストールディレクトリー/httpsd/conf/ssl/crl/PEM"
```

PEM 形式の CRL ファイルを格納しているディレクトリーを指定します。

2.3.147 SSLDenySSL

SSL によるアクセスを禁止する場合に指定します。

説明

SSL によるアクセスを禁止する場合に指定します。このディレクティブが指定されている場合、SSLEnable ディレクティブで SSL を有効にしても https によるアクセスがステータスコード403 Forbidden で拒否されます。SSLRequireSSL ディレクティブの逆の動作をします。

書式

```
SSLDenySSL
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.148 SSLDisable

SSL を無効にする場合に指定します。

説明

SSL を無効にします。デフォルト値はSSLEnable ディレクティブ (SSL を有効) です。バーチャルホストで特定のホストに対して SSL を無効にする場合などに指定します。

書式

```
SSLDisable
```

記述できる場所

httpsd.conf、<VirtualHost>

2.3.149 SSLEnable

SSL を有効にする場合に指定します。

説明

SSL を有効にします。SSLDisable ディレクティブを指定しないかぎり、デフォルトで有効になります。

書式

```
SSLEnable
```

記述できる場所

httpsd.conf、<VirtualHost>

2.3.150 SSLExportCertChainDepth

環境変数 SSL_CLIENT_CERT_CHAIN_n にクライアントの証明書を発行した CA からルート CA までの証明書を設定するときに指定します。

説明

SSL クライアント認証をする場合、環境変数 SSL_CLIENT_CERT_CHAIN_n にクライアントの証明書を発行した CA からルート CA までの証明書を設定するときに指定します。指定した値が n の最大値になります。このディレクティブは SSLExportClientCertificates ディレクティブを指定している場合だけ有効になります。指定された数の CA 証明書が gcache サーバへキャッシュされるため、CGI または Servlet で必要な数だけをこのディレクティブに指定することでキャッシュを有効に利用できます。ただし、メモリの制限でキャッシュされた一部の証明書が削除されて取得できなかった場合は、取得できたものだけを環境変数に設定します。

書式

```
SSLExportCertChainDepth 値
```

指定できる値

0

環境変数は設定しません。

1~9

クライアントの証明書に近い方から順に番号が割り当てられ、環境変数を設定します。環境変数には DER 形式の証明書を Base64 エンコーディングした値を設定します。1 つの証明書を Base64 エンコーディングした場合のバイト数は約 1KB です。

デフォルト値

定義項目の省略

0

記述できる場所

httpsd.conf、<VirtualHost>

記述例

「ルート CA - 下位 CA - クライアント証明書」という証明書チェーンの場合
環境変数と証明書の対応は次のようになります。

環境変数	証明書
SSL_CLIENT_CERT	クライアント証明書
SSL_CLIENT_CERT_CHAIN_1	下位 CA の証明書
SSL_CLIENT_CERT_CHAIN_2	ルート CA の証明書

この環境変数と証明書チェーンをすべて取得するには、次のようにディレクティブを指定します。

```
SSLEExportClientCertificates  
SSLEExportCertChainDepth 2
```

SSLEExportCertChainDepth には、2 以上の値を指定します。

2.3.151 SSLEExportClientCertificates

環境変数 SSL_CLIENT_CERT にクライアント証明書を設定するときに指定します。

説明

SSL クライアント認証をする場合、環境変数SSL_CLIENT_CERT にクライアント証明書を設定するときに指定します。環境変数SSL_CLIENT_CERT には DER 形式の証明書を Base64 エンコーディングした値を設定します。

書式

```
SSLExportClientCertificates
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

2.3.152 SSLFakeBasicAuth

クライアント証明書を利用した Basic 認証をする場合に指定します。

説明

SSL クライアント認証の機能とあわせて、Web ブラウザーでユーザー ID とパスワードを入力することなく、クライアント証明書の提示だけで Basic 認証をできるようにします。AuthUserFile ディレクティブで指定するファイルには X509 クライアント証明書の Subject とパスワードを記述します。パスワードは、次に示す値で常に固定とします（"password" を暗号化したもの）。

```
"{SHA}W6ph5Mm5Pz8GgiULbPgZG37mj9g="
```

書式

```
SSLFakeBasicAuth
```

記述できる場所

httpsd.conf、<VirtualHost>

記述例

hwscertutil コマンドで表示する証明書の Subject フィールドの値

```
Subject: EMAIL=username@userhost, CN=username, OU=Software, O=HITACHI,  
L=Yokohama-shi, ST=Kanagawa, C=JP
```

この場合、AuthUserFile ディレクティブで指定するファイルは次のように指定します。

```
/C=JP/ST=Kanagawa/L=Yokohama-shi/O=HITACHI/OU=Software/CN=username  
/Email=username@userhost:{SHA}W6ph5Mm5Pz8GgiULbPgZG37mj9g=
```

LogFormat ディレクティブのu 指定ではSubject がロギングされます。

認証に失敗した場合、ステータスコード 401 Authorization Required を応答します。

2.3.153 SSLProtocol

使用する SSL プロトコルのバージョンを指定します。

説明

使用する SSL プロトコルのバージョンを指定します。

プロトコル名として設定できる値は次のとおりです。

SSLv3

SSL プロトコルバージョン 3 を使用する。

TLSv1

TLS プロトコルバージョン 1.0 を使用する。

TLSv11

TLS プロトコルバージョン 1.1 を使用する。

TLSv12

TLS プロトコルバージョン 1.2 を使用する。

All

上記すべてのプロトコルを使用する。

書式

```
SSLProtocol プロトコル名 [プロトコル名 ...]
```

デフォルト値

定義項目の省略

All

記述できる場所

httpd.conf、<VirtualHost>

2.3.154 SSLRequireCipher

指定した暗号種別以外でのアクセスを拒否し、ステータスコード 403 Forbidden を応答する場合に指定します。

説明

指定した暗号種別以外でのアクセスを拒否し、クライアントにステータスコード 403 Forbidden を応答します。

暗号種別を次に示します。

暗号種別	鍵交換方式	認証方式	対称鍵暗号方式	暗号鍵サイズ (bit)	メッセージ認証アルゴリズム
RC4-MD5	RSA	RSA	RC4	128	MD5
RC4-SHA	RSA	RSA	RC4	128	SHA
AES128-SHA	RSA	RSA	AES	128	SHA
AES128-SHA256	RSA	RSA	AES	128	SHA256
DES-CBC3-SHA	RSA	RSA	DES	168	SHA
AES256-SHA	RSA	RSA	AES	256	SHA
AES256-SHA256	RSA	RSA	AES	256	SHA256

書式

```
SSLRequireCipher 暗号種別 [暗号種別 ...]
```

記述できる場所

httpd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

2.3.155 SSLRequiredCiphers

SSL 通信で使用できる暗号種別を指定します。

説明

SSL 通信で使用できる暗号種別を指定します。ディレクティブに指定した暗号種別とクライアントが使用できる暗号種別との間で一致するものがあれば、SSL 通信が確立され HTTP リクエストを受信します。一致するものがない場合は、SSL 通信は確立されずに HTTP リクエストを受信しません。

暗号種別を次に示します。

暗号種別	鍵交換方式	認証方式	対称鍵暗号方式	暗号鍵サイズ (bit)	メッセージ認証アルゴリズム
RC4-MD5	RSA	RSA	RC4	128	MD5
RC4-SHA	RSA	RSA	RC4	128	SHA
AES128-SHA	RSA	RSA	AES	128	SHA
AES128-SHA256	RSA	RSA	AES	128	SHA256
DES-CBC3-SHA	RSA	RSA	DES	168	SHA
AES256-SHA	RSA	RSA	AES	256	SHA
AES256-SHA256	RSA	RSA	AES	256	SHA256

書式

```
SSLRequiredCiphers 暗号種別 [:暗号種別 ...]
```

記述できる場所

httpd.conf、<VirtualHost>

記述例

```
SSLRequiredCiphers RC4-MD5:RC4-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA
```

2.3.156 SSLRequireSSL

SSL 以外のアクセスを禁止する場合に指定します。

説明

SSL 以外によるアクセスを禁止する場合に指定します。このディレクティブが指定されている場合、SSLDisable ディレクティブで SSL を無効にしても http によるアクセスがステータスコード 403 Forbidden で拒否されます。異なるディレクティブの記述場所で、不用意に SSL を無効にしコンテンツを公開してしまうことを防止します。

書式

```
SSLRequireSSL
```

記述できる場所

httpsd.conf、<VirtualHost>、<Directory>、.htaccess

上書き許可

FileInfo レベル

記述例

```
<VirtualHost 172.17.40.10:443>  
  SSLDisable  
  ...  
  <Directory /secure/dir>  
    SSLRequireSSL  
  ...  
</Directory>  
</VirtualHost>
```

172.17.40.10 ホストの 443 ポートに対する http アクセスは、/secure/dir ディレクトリーへのアクセスを除いてできます。/secure/dir ディレクトリーへの http アクセスは、ステータスコード 403 Forbidden を応答します。

2.3.157 SSLSessionCacheSize

セッションキャッシュサイズの上限値を指定します。

説明

セッションキャッシュサイズの上限値をバイト単位で指定します。0 を指定した場合、セッションキャッシュは実施されません。

max を指定した場合、上限を設定しません。1SSL セッション当たり、サーバ認証だけする場合約 200 バイト、クライアント認証もする場合約 1 キロバイト使用します。

書式

```
SSLSessionCacheSize {サイズ|max}
```

指定できる値

0~2147483647

デフォルト値

定義項目の省略

16777216

記述できる場所

httpsd.conf

記述例

```
SSLSessionCacheSize 1024
```

2.3.158 SSLSessionCacheTimeout

SSL セッションの有効時間を指定します。

説明

Web サーバ内または SSL セッション管理キャッシュサーバ gcache 内で保持されるセッション ID などのデータの有効時間を秒単位で指定します。

重要

グリニッジ標準時 (GMT) の 2038 年 1 月 19 日 3 時 14 分 7 秒を超えないように、有効時間を設定してください。

書式

```
SSLSessionCacheTimeout 値
```

指定できる値

1-2147483647

デフォルト値

定義項目の省略

3600

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLSessionCacheTimeout 3600
```

2.3.159 SSLVerifyClient

クライアント認証時の証明書に関する設定をする場合に指定します。

説明

クライアント認証時の証明書に関する設定を指定します。

書式

```
SSLVerifyClient {0|1|2}
```

指定できる値

0

証明書の要求をしません。

1

クライアントは証明書を提示できます（運用テスト用）。

2

クライアントは証明書を提示しなければなりません。

デフォルト値

定義項目の省略

0

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLVerifyClient 2
```

2.3.160 SSLVerifyDepth

証明書のチェーンを何段階までたどるかを指定します。

説明

証明書のチェーンを何段階までたどるかを指定します。

クライアント認証に使用する CA 証明書のチェーンについて、認証チェックをする段階数を指定します。チェーンされた CA をどこまで信用するかを制限するために使用します。自己署名の証明書は認証しないため、**段階数**は 2 以上を指定します。例を次に示します。

(例)

条件 1

CA1 は、root CA に署名されている。

条件 2

証明書 1 は、root CA に署名されている。

条件 3

証明書 2 は、CA1 に署名されている。

SSLVerifyDepth の指定

この場合、証明書 1、証明書 2 とも認証チェックをするためには、SSLVerifyDepth ディレクティブに 3 以上を指定します。また、証明書 1 は認証チェックし、証明書 2 は認証チェックをしないようにするには SSLVerifyDepth ディレクティブに 2 を指定します。

書式

```
SSLVerifyDepth 段階数
```

指定できる値

0-10

デフォルト値

定義項目の省略

0

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
SSLVerifyDepth 10
```

2.3.161 ThreadsPerChild

サーバとして起動するスレッド数を指定します。

説明

サーバとして起動するスレッド数を指定します。指定したスレッド数はサーバの最大同時接続数を示します。

書式

```
ThreadsPerChild スレッド数
```

指定できる値

1-1024

デフォルト値

定義項目の省略

64

記述できる場所

httpsd.conf

記述例

```
ThreadsPerChild 50
```

2.3.162 Timeout

Web サーバの送受信の待ち時間を指定します。

説明

次の待ち時間を秒単位で指定します。0 を指定した場合は、待ち時間が 0 秒になります。

- クライアントからのリクエスト受信（コネクション確立後、HTTP プロトコルの受信）中にデータを受信しなくなった場合の待ち時間
- クライアントへのレスポンス送信中にデータを送信できなくなった場合の待ち時間
- CGI プログラムへのリクエスト送信中にデータを送信できなくなった場合の待ち時間
- CGI プログラムへのリクエスト送信後からレスポンス受信までの待ち時間
- CGI プログラムからのレスポンス受信中にデータを受信しなくなった場合の待ち時間

- CGI プログラムからのレスポンス受信後、入出力用のパイプを閉じるまでの待ち時間
- リバースプロキシを使用している場合の、バックエンドサーバへのリクエスト送信中にデータを送信できなくなった場合の待ち時間
- リバースプロキシを使用している場合の、バックエンドサーバへのリクエスト送信後からレスポンス受信までの待ち時間
- リバースプロキシを使用している場合の、バックエンドサーバからのレスポンス受信中にデータを受信しなくなった場合の待ち時間

リバースプロキシ使用時のバックエンドサーバとの送受信の待ち時間は、ProxyTimeout ディレクティブの値が優先されます。

書式

```
Timeout 時間
```

指定できる値

0–65535

デフォルト値

定義項目の省略

60

記述できる場所

httpsd.conf

記述例

```
Timeout 300
```

2.3.163 TraceEnable

TRACE メソッドによるリクエストを拒否するかどうかを指定します。

説明

TRACE メソッドによるリクエストを拒否するかどうかを指定します。

書式

```
TraceEnable {0n|0ff|extended}
```

指定できる値

0n

TRACE メソッドによるリクエストを許可します。ただし、リクエストボディが付加されている場合は、413 Request Entity Too Large を応答します。

0ff

TRACE メソッドによるリクエストを拒否します。TRACE メソッドによるリクエストの場合は、ステータスコード 403 Forbidden を応答します。

extended

TRACE メソッドによるリクエストを許可します。リクエストボディが付加されていても許可します。ただし、リバースプロキシ以外のリクエストボディサイズの上限は 64KB です。

デフォルト値

定義項目の省略

0n

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
TraceEnable 0ff
```

2.3.164 TransferLog

ラベル名を指定しない LogFormat ディレクティブのフォーマットでのアクセスログを出力する場合に指定します。

説明

ログを格納するファイルまたはログを出力するプログラムを指定します。ログのフォーマットはラベル名を指定しない LogFormat ディレクティブで指定できます。

LogFormat ディレクティブでログのフォーマットを指定する場合は、IPv6 アドレスや IPv6 アドレスに対応したホスト名も出力できます。

LogFormat ディレクティブでフォーマットを指定しない場合は、標準のログフォーマットで出力します。

書式

```
TransferLog {ファイル名|パイプ}
```

指定できる値

ファイル名

ログを格納するファイル名を指定します。ファイル名には、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が指定できます。

パイプ

標準入力からログ情報を受け取るプログラムを"|プログラム名"のフォーマットで指定します。

記述できる場所

httpsd.conf、<VirtualHost>

記述例

```
TransferLog "|  
¥"¥"Application Serverのインストールディレクトリー/httpsd/sbin/rotatelogs.exe¥"  
¥"Application Serverのインストールディレクトリー/httpsd/logs/access¥" 86400¥"
```

rotatelogs プログラムを使用してログを 24 時間ごとに分割して採取します。

2.3.165 TypesConfig

ファイル拡張子とコンテンツタイプ (MIME タイプ) の関係を定義する設定ファイルを指定します。

説明

ファイル拡張子とコンテンツタイプ (MIME タイプ) の関係を定義する設定ファイルを指定します。ファイル名には、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が指定できます。

設定ファイル内の指定形式は、MIME タイプ ファイル拡張子 [ファイル拡張子 ...] です。MIME タイプだけ指定している行は無視します。また、行の最初に#を付けると、コメント行になります。

書式

```
TypesConfig ファイル名
```

デフォルト値

定義項目の省略

```
conf/mime.types
```

記述できる場所

```
httpsd.conf
```

記述例

```
TypesConfig conf/mime.types
```

MIME タイプの設定ファイルはmime.types

2.3.166 UnsetEnv

CGI スクリプトに渡す環境変数から、SetEnv ディレクティブまたは PassEnv ディレクティブで指定した環境変数を削除する場合に指定します。

説明

CGI スクリプトに渡す環境変数から、SetEnv ディレクティブまたは PassEnv ディレクティブで指定した環境変数を削除する場合に指定します。

書式

```
UnsetEnv 環境変数 [環境変数 ...]
```

記述できる場所

```
httpsd.conf、<VirtualHost>、<Directory>、.htaccess
```

上書き許可

FileInfo レベル

記述例

```
UnsetEnv MY_ENV
```

2.3.167 UseCanonicalName

サーバの正式な名前の生成方法を指定します。

説明

サーバの正式な名前の生成方法を指定します。サーバの正式な名前は、自サーバを参照する URL や環境変数の `SERVER_NAME` と `SERVER_PORT` に設定されます。なお、`On`、`Off`、`dns` すべての場合で、IPv6 アドレスに対応しています。

書式

```
UseCanonicalName {On|Off|dns}
```

指定できる値

On

サーバの正式な名前は、`ServerName` ディレクティブ指定値から作成され、自サーバを参照する URL や環境変数に設定されます。`VirtualHost` 指定時に IP アドレスを使用する場合は、`VirtualHost` ブロック内で `ServerName` を指定してください。ブロック内で `ServerName` を指定していない場合は、IP アドレスからホスト名を取得します。

Off

サーバの正式な名前は、`Host` ヘッダーによってクライアントから与えられたホスト名称とポート番号から作成され、自サーバを参照する URL や環境変数に設定されます。ただし、`Host` ヘッダーが与えられない場合は、`ServerName` ディレクティブ値と、実際のコネクションに使用されているポート番号から作成されます。

dns

`Host` ヘッダーを持たない古いクライアントのためのオプションです。このオプション指定時には、サーバの正式な名前は、クライアントから与えられたサーバの IP アドレスから逆引きしたホスト名称および実際にコネクションに使用されているポート番号から作成され、自サーバを参照する URL や環境変数に設定されます。

デフォルト値

定義項目の省略

`Off`

記述できる場所

`httpsd.conf`、`<VirtualHost>`、`<Directory>`

2.3.168 UserDir

Web ブラウザーからの/`~ユーザー名`/へのリクエストに対して公開するサーバ上の場所を指定します。

説明

Web ブラウザーからの/`~ユーザー名`/へのリクエストに対して公開するサーバ上の場所をディレクトリー名で指定します。`disabled` を指定すると、Web コンテンツを公開しないユーザーを指定できます。

ディレクトリー名は、絶対パスで指定します。

ディレクトリー名

- 絶対パスで指定した場合
ユーザーディレクトリーの場所を指定します。/`~ユーザー名`/へのリクエストがあった場合、"`ディレクトリー名/ユーザー名`"にアクセスします。

`disabled`

Web ブラウザーからの/`~ユーザー名`/へのリクエストに対して、Web コンテンツを公開しないユーザーを指定します。指定されたユーザー名でのリクエストに対しては、アクセスするディレクトリー名を変換しません。ユーザー名の指定がない場合は、すべてのユーザーについて`disabled` を指定したことになります。

❗ 重要

- 複数のUserDir ディレクティブでディレクトリー名を指定した場合、あとから指定したものに上書きされます。
- `disabled` に指定するユーザー名は、複数のUserDir ディレクティブを用いて指定できます。

書式

```
UserDir {ディレクトリー名|disabled [ユーザー名 [ユーザー名 ...]]}
```

デフォルト値

定義項目の省略

`disabled`

記述できる場所

`httpsd.conf`、`<VirtualHost>`

記述例

(例 1)

```
UserDir public_html
```

ユーザー user1 のホームディレクトリーを/home/user1 とすると、リクエストhttp://ホスト名[:ポート番号]/~user1/index.html で、/home/user1/public_html/index.html にアクセスします。

(例 2)

```
UserDir /home  
UserDir disabled user3  
UserDir disabled user4 user5
```

リクエストhttp://ホスト名[:ポート番号]/~user1/index.html で、/home/user1/index.html にアクセスします。ただし、user3 はdisabled が指定されているため、http://ホスト名[:ポート番号]/~user3/index.html というリクエストで/home/user3/index.html にアクセスできません。user4、user5 についても user3 と同様です。

(例 3)

```
UserDir disabled
```

すべてのユーザーに対してdisabled を指定します。

2.4 mime.types

コンテンツのファイル拡張子とコンテンツタイプの関連づけを定義するファイルです。

説明

コンテンツのファイル拡張子とコンテンツタイプ (MIME タイプ) の関連づけを定義します。システム管理者が管理します。

書式

MIMEタイプ ファイル拡張子 [ファイル拡張子 ...]

格納先

*Application Server*のインストールディレクトリー¥httpsd¥conf¥mime.types

2.5 アクセス制御ファイル

アクセス制御を定義するアクセス制御ファイルです。

説明

アクセス制御ファイル (.htaccess) を特定のディレクトリ下に作成すれば、そのディレクトリに対するアクセス権を設定できます。

アクセス制御ファイルの名称は、AccessFileName ディレクティブで指定します。デフォルトは.htaccess です。

アクセス制御ファイルによるアクセス制御は、Web サーバを再起動することなく、有効になります。ただし、正しく機能させるためには、httpsd.conf のAllowOverride ディレクティブを適切な上書き許可レベルに設定する必要があります。

アクセス制御ファイルにパスワードファイルを指定すると、ユーザーがそのディレクトリにアクセスする場合にユーザー名およびパスワードの入力を要求します。

アクセス制御ファイル (.htaccess) とパスワードファイル (.htpasswd) は 1 対 1 である必要はありません。異なるアクセス制御ファイルのAuthUserFile ディレクティブに同じパスワードファイルを指定できます。

書式

ディレクティブの記述規則を次に示します。

正規表現

ディレクティブの指定に使用できる正規表現を次に示します。

記号	機能	使用例	使用例の意味
.	任意の 1 文字。	a...c	a のあとに任意の 3 文字と c が続く。abcdc は適合する。
*	直前の 1 文字の 0 個以上の繰り返し。	ab*cd*	ac、abbbbc、abbbbcd は適合する。
+	直前の 1 文字の 1 個以上の繰り返し。	ab*c+	abbbc は適合する。abbb は適合しない。
?	直前の 1 文字があるかないか。	abbc?	abbbc、abbb は適合する。
	選択肢の区切り。	a bc d	a、bc または d。
¥	直後の特殊文字 (. ^\$*+? ¥[](){}) の 1 文字。ただし、¥を表す場合は¥¥¥。	¥. ¥¥¥	. と適合する。 1 文字の ¥ と適合する。
^	行の先頭に適合する。	^ab	abcde は適合する。
\$	行の末尾に適合する。	abc\$	aaabc は適合する。
{m}	直前の正規表現の m 個の繰り返し。	a{5}	aaaaa が適合する。

記号	機能	使用例	使用例の意味
{m,}	直前の正規表現の m 個以上の繰り返し。	a{3,}	aaa、aaaa は適合する。aa は適合しない。
{m,n}	直前の正規表現の m 個以上 n 個以下の繰り返し。	a{3,5}	aaa、aaaa、aaaaa が適合する。aa、aaaaaa は適合しない。
[文字列]	文字列にある任意の 1 文字。*	[abc]*または [a-c]*	aaa、bbb、ccc、cba、aab は適合する。
[^文字列]	文字列にない任意の 1 文字。	[^0-9]	数字以外の 1 字が適合する。
(文字列)	文字列をグループ化する。	(ab)+	ababab が適合する。ababb は適合しない。
		aa(xx yy)bb	aaxxbb、aayybb が適合する。

注※

次の 3 文字は、[文字列]で特殊な意味を持ちます。

^：[の次に指定して、文字列に含まれないものを示すために用います。

]：文字列の最後を示すために用います。

-：範囲を指定するために用います。

また、これら特殊文字の前の¥は省略されます。

[文字列]で特殊な意味を持つ文字を通常の文字として指定するには、次のようにします。なお、次の 4 つ以外の特殊文字は、通常の文字として扱われます。

^：文字列の先頭以外で指定します。(例) [ab^yz]

]：文字列の先頭に指定します。(例) [^]abxy]

-：最後に指定します。(例) [abxy-]

¥：¥¥¥と指定します。(例) [¥¥¥abxy]

ディレクティブに指定するパス情報

ディレクトリー名、ファイル名またはパス名を指定するディレクティブの場合、ディレクティブの種類によって、指定できるパス情報が異なります。

パスの種類には、次のものがあります。各ディレクティブのパス情報は、各ディレクティブで説明します。

- 絶対パスしか指定できません（ドライブ名を付けた指定も絶対パスに含まれます）。
- ServerRoot ディレクティブの指定値からの相対パスで指定できます（ただし、ServerRoot ディレクティブの指定が先に必要）。

また、パス情報にネットワーク上のディレクトリーやファイルを指定することはできません。ネットワークを使用したファイルシステム上のディレクトリーやファイルを指定することもできません。

コメント行

設定ファイル中、行の最初に#を付けると、コメント行になります。ただし、ディレクティブを指定したあとに#から始まる文字列を記述しても、#以降をコメントとして扱いません。

コメント行を指定する場合の記述例を次に示します。

- 正しい例

```
#Deny from all
```

#行はコメント行として扱われます。

- 誤った例

```
Deny from all    #comment
```

#comment はディレクティブ指定値として扱われます。コメントとしては扱われません。

IPv6 アドレスを指定するときの注意

ディレクティブに IPv6 アドレスを記述する場合は、「*[IPv6アドレス]*」のように IPv6 アドレスを[]で囲んで指定してください。また、ディレクティブに IPv6 アドレスとポート番号を併記する場合は、「*[IPv6アドレス]:ポート番号*」のように IPv6 アドレスを[]で囲み、「:」の後ろにポート番号を指定します。

ただし、次のディレクティブに IPv6 アドレスを記述する場合は、IPv6 アドレスを[]で囲まないで指定してください。

- Allow from ディレクティブ
- Deny from ディレクティブ
- HWSSetEnvIfIPv6 ディレクティブ

IPv6 アドレスを指定する場合は、グローバルユニキャストアドレスを指定してください。

格納先

```
アクセス制御するディレクトリー
```

記述例

次のようなディレクトリー構成で、各ディレクトリーに対してアクセス制御ファイルにアクセス権を設定します。

```
[user001のpublic_html]
|
|--[auth]---+.htaccess
|             +-index.html
|
|--[test1]---+.htaccess
|             +-htpasswd (user001/test1)
|             +-index.html
|             |--[test11]---+.htaccess
|                 +-htpasswd (user001/test11)
|                 +-index.html
|             |--[test12]---+index.html
|                 |--[test121]---+.htaccess
|                     +-index.html
```

```
+-[test2]-+-.htaccess
+-.htpasswd (user001/test21,user002/test22,user003/test23)
+-.groupfile(mygroup: user001 user002)
+-index.html
```

(例 1) auth ディレクトリー下のアクセス権の定義 (auth/.htaccess ファイル)

IP アドレスが172.18.102.11 および172.16.202.4 のサーバからのアクセスを拒否します。

```
Order deny,allow          ...1.
Deny from 172.18.102.11 172.16.202.4 ...2.
```

1. アクセス拒否の定義を先に評価
2. アクセス拒否の定義

(例 2) test1 ディレクトリー下のアクセス権の定義 (test1/.htaccess ファイル)

ユーザー名=user001、パスワード=test1 を入力した場合だけ、test1/index.html およびtest1/test12/index.html へのアクセスを許可します。

```
AuthUserFile C:/user001/public_html/test1/.htpasswd ...1.
AuthName "test1 Directory" ...2.
AuthType Basic
<Limit GET POST> ...3.
    Require user user001 ...4.
</Limit>
```

1. パスワードファイルに登録しているユーザー名 (user001) およびパスワード (test1) の定義
2. realm 名の定義
3. メソッドに対する定義
4. ユーザー名 : user001 のアクセスを許可

(例 3) test1/test11 ディレクトリー下のアクセス権の定義 (test1/test11/.htaccess ファイル)

ユーザー名=user001、パスワード=test11 を入力した場合だけ、test1/test11/index.html へのアクセスを許可します。

```
AuthUserFile C:/user001/public_html/test1/test11/.htpasswd ...1.
AuthName "test11 Directory" ...2.
AuthType Basic
<Limit GET POST> ...3.
    Require user user001 ...4.
</Limit>
```

1. パスワードファイルに登録しているユーザー名 (user001) およびパスワード (test11) の定義
2. realm 名の定義
3. メソッドに対する定義
4. ユーザー名 : user001 のアクセスを許可

(例 4) test1/test12/test121 ディレクトリー下のアクセス権の定義 (test1/test12/test121/.htaccess ファイル)

ユーザー名=user001、パスワード=test1 を入力し、Web ブラウザーが MSIE の場合だけ、test1/test12/test121/index.html へのアクセスを許可します。

```
Order deny,allow    ...1.
Allow from env=MSIE ...2.
Deny from all       ...3.
```

1. アクセス拒否の定義を先に評価
2. Web ブラウザーが MSIE の場合、アクセスを許可
3. すべてのホストからのアクセスを拒否

ただし、httpsd.conf に次のディレクティブを定義しているものとします。

```
SetEnvIf User-Agent ".*MSIE.*" MSIE
```

(例 5) test2 ディレクトリー下のアクセス権の定義 (test2/.htaccess ファイル)

mygroup グループのユーザー名、パスワードを入力した場合だけ、test2/index.html へのアクセスを許可します。

```
AuthUserFile C:/user001/public_html/test2/.htpasswd    ...1.
AuthGroupFile C:/user001/public_html/test2/.groupfile  ...2.
AuthName "test2 Directory"                             ...3.
AuthType Basic
<Limit GET POST>                                       ...4.
    Require group mygroup                               ...5.
</Limit>
```

1. パスワードファイルに登録している次の情報の定義
ユーザー名：user001、パスワード：test21
ユーザー名：user002、パスワード：test22
ユーザー名：user003、パスワード：test23
2. グループファイルに登録しているグループ名 (mygroup) およびmygroup に登録しているユーザー名 (user001、user002、user003) の定義
3. realm 名の定義
4. メソッドに対する定義
5. グループ名「mygroup」のアクセスを許可

2.6 Web サーバで使用する定義項目（ユーティリティー）の詳細

Web サーバの動作環境を設定するための各種ユーティリティーについて説明しています。

2.6.1 rotatelog

ログを時間単位に分割して、複数のファイルに出力します。

説明

アクセスログやエラーログを一定時間単位（例えば、24 時間ごと）に分割して、複数のファイルに出力できます。rotatelog プログラムは次のディレクティブに指定できます。

- CustomLog ディレクティブ
- ErrorLog ディレクティブ
- HWSRequestLog ディレクティブ
- TransferLog ディレクティブ

書式

```
rotatelog 分割ログファイルのプレフィックス ログ分割時間間隔  
[-fnum ファイル数] [-diff GMTに対する時差]
```

分割ログファイルのプレフィックス

分割ログファイルのプレフィックスを絶対パスで指定します。

「プレフィックス.nnnnnnnnn」 というファイルに、ログを採取します。

nnnnnnnnnn：ログ採取開始時刻を表します。ログ採取時刻とは次の式で示す値です。

$((1970 \text{ 年 } 1 \text{ 月 } 1 \text{ 日 の } 0 \text{ 時 } 0 \text{ 分 } 0 \text{ 秒 (GMT) を 起 点 と し た 、 ロ グ を 出 力 す る 時 間 の 通 算 秒 数 } \div \text{ ロ グ 分 割 時 間 間 隔 }) \text{ の 小 数 点 以 下 を 切 り 捨 て た 値 } \times \text{ ロ グ 分 割 時 間 間 隔}$

ログ分割時間間隔

1 つのログファイルを採取する時間間隔を秒単位に指定します。指定した時間が経過するごとに、新規ファイルにログを採取します。

指定できる値

1～31536000

-fnum ファイル数

分割したログファイルのファイル数を指定します。分割したファイル数がここで指定した数を超えた場合、最も古いファイルから削除します。このオプションを省略した場合、ファイルは削除しません。

指定できる値

1～256

-diff GMTに対する時差

ログファイルを分割する基準となる時間を、GMT に対する時差として分単位で指定します。指定しないまたは 0 を指定すると、1970 年 1 月 1 日 0 時 0 分 0 秒(GMT)が基準時間となります。GMT に対するローカルタイムの差が n 時間である場合に、ローカルタイムの m 時 0 分 0 秒を基準にするときには、(n-m)×60 を指定します。JST の 0 時 0 分 0 秒を基準にする場合には、(+9-0)×60 で 540 を指定します。

指定できる値

-1439~1439

格納先

*Application Server*インストールディレクトリー¥httpsd¥sbin

記述例

ディレクティブに、"|プログラム名"の形式で指定して使用します。ログファイルを定期的に別ファイルに分割して採取します。

24 時間ごとに、アクセスログを分割して*Application Server*インストールディレクトリー¥httpsd¥logs ¥access.nnnnnnnnnn ファイルに採取します。分割時間を日本時間に設定し、日本時間の毎 0 時にログファイルを分割する場合の指定を次に示します。

```
TransferLog "|¥¥"Application Serverインストールディレクトリー/httpsd/sbin/rotatelog.exe¥"  
¥"Application Serverインストールディレクトリー/httpsd/logs/access¥" 86400 -diff 540¥"
```

- ログファイル名：*Application Server*インストールディレクトリー¥httpsd¥logs¥access.nnnnnnnnnn
- ログ分割時間間隔：86400 秒 (= 24 時間)

注意事項

- ログファイルは、そのファイルを開いているプロセスがある間は削除できません。このため、-fnum オプションで指定した値より多いファイルが残ることがあります。例えば、制御プロセスがログを出力したファイルは、制御プロセスが終了するまで削除されません。
- 起動時に引数誤りなどがあった場合、Web サーバは起動しますがrotatelog プログラムは起動しません。その際、rotatelog プログラムではイベントログに、次に示す属性を持ったメッセージを出力します。
 - 種類：エラー
 - ソース：HitachiWebServer
 - 分類：なし
 - イベント：3299
 - 説明：rotatelog.exe: (メッセージ)

このメッセージが出力されていた場合、メッセージ内容に従って引数を見直したあとに、サーバを再起動してください。なお、同じメッセージが複数回出力されることがあります。

- `-fnum` オプションの指定でのログファイルの制御では、Web サーバの再起動時にディレクトリー名またはログファイルのプレフィックスを変更すると、以前に採取したログファイルが削除されません。この場合は運用に応じて削除してください。
- Web サーバを起動または再起動してから、指定したログ分割間隔時間が経過した場合、分割したログファイルのプレフィックスに一致するファイルの数が `-fnum` オプションの指定値を超えると、作成時間の古いログファイルから削除されます。
- 分割ログファイルのプレフィックスは絶対パスで指定してください。
- `rotatelog` プログラムの引数指定に誤りがある場合、`rotatelog` プログラムの起動に失敗しますが、Web サーバは起動します。この場合はログは出力されません。`rotatelog` プログラムを指定した場合は、ログファイルが作成され、意図した分割が実施されることを確認してください。

2.6.2 rotatelog2

ログをファイルサイズで分割して、複数のファイルに出力します。

説明

アクセスログやエラーログをログファイルサイズで分割して、複数のファイルにラップアラウンドして出力できます。`rotatelog2` プログラムは次のディレクティブに指定できます。

- `CustomLog` ディレクティブ
- `ErrorLog` ディレクティブ
- `HWSRequestLog` ディレクティブ
- `TransferLog` ディレクティブ

書式

```
rotatelog2 ログファイルプレフィックス名 ログファイルサイズ  
          ログファイル個数
```

ログファイルプレフィックス名

出力するログファイルのプレフィックス名を絶対パスで指定します。

出力するログファイルは、プレフィックス.nnn のファイル名となります。

.nnn は、.001 からログファイル個数で指定した値までです。

ログファイル個数を nnn 面とすると、nnn 面のうち、Web サーバ起動時の更新時刻が最新のものが、カレントのログファイルとなります。ログファイルは、ファイル名称のプレフィックスに拡張子.001~.nnn を付けて区別します。カレントのログファイルの拡張子が.mmm であった場合、カレントのログファイ

ルがいっぱいになると、続きは、.mmm+1 のログファイルをクリアして出力されます。.mmm が .nnn と一致した場合、続きは .001 に出力されます。

プレフィックス .index のインデックス番号格納用ファイルが作成されます。このファイルは .nnn 管理用ファイルであり、rotatelogs2 プログラムの起動時に作成され、停止時に削除されます。なお、起動エラーの一部などで削除されないことがあります。以降の Web サーバの動作に影響はありません。

ログファイルサイズ

ログファイルの最大サイズ（単位：KB）を指定します。

ログを出力するタイミングで、最大サイズを超えていると、次のログファイルをクリアして続きが出力されます。

指定できる値

1～2097151

ログファイル個数

出力するログファイルの最大数を指定します。

最大サイズを超えて次のログファイルに移る場合、それまで処理していたログファイルの拡張子が、最大個数と同じとき、再度 .001 のファイルから使用します。

指定できる値

1～256

格納先

*Application Server*インストールディレクトリー¥httpsd¥sbin

記述例

ディレクティブに、"|プログラム名"の形式で指定して使用します。

(例) 4,096KB ごとにエラーログを最大 5 個採取する場合

```
ErrorLog "|¥¥Application Serverインストールディレクトリー/httpsd/sbin/rotatelogs2.exe¥"  
¥Application Serverインストールディレクトリー/httpsd/logs/errorlog¥" 4096 5¥"
```

errorlog.001～errorlog.005 の順番にログが出力されます。errorlog.005 が 4,096KB を超えると errorlog.001 をクリアして続きが出力されます。Web サーバ起動時に、すでにこれらのログファイルがある場合には、更新時刻の最も新しいログファイルが、出力対象のログファイルとなります。このログファイルのサイズがすでに 4,096KB を超えている場合には、次のログファイルをクリアして続きが出力されます。4,096KB を超えない場合は、このファイルの続きに出力されます。

注意事項

- インデックス番号格納用ファイルは、rotatelogs2 プログラムが動作中の間は、編集や削除をしないでください。編集するとログが正しく出力されないことがあります。

- Web サーバの起動時にインデックス番号格納用ファイルプレフィックス `.index` と同名のファイルが存在した場合、ファイルは上書きされます。
- 起動時に引数誤りなどがあった場合、Web サーバは起動しますが `rotatelog2` プログラムは起動しません。その際、`rotatelog2` プログラムではイベントログに、次に示す属性を持ったメッセージを出力します。
 - 種類：エラー
 - ソース：HitachiWebServer
 - 分類：なし
 - イベント：3299
 - 説明：rotatelog2.exe: (メッセージ)

このメッセージが出力されていた場合、メッセージ内容に従って引数を見直したあとに、サーバを再起動してください。なお、同じメッセージが複数回出力されることがあります。

- ログファイルプレフィックス名は、絶対パスで指定してください。
- Web サーバ起動時の出力ログファイルは、更新日時が最新のものを対象とするため、誤ってファイルを更新した場合は、正しいファイルへの出力ができなくなります。
- ログファイルサイズには、同一秒内に複数のファイルが指定サイズを超えるような、小さいサイズを指定しないでください。このようなサイズを指定した場合には、最も新しいログファイル以外が出力対象となり、正しくローテーションされなくなることがあります。
- 設定ファイル内に、同一のログファイルプレフィックス名を、複数個所で指定しないでください。複数個所で指定した場合には、最も新しいログファイル以外が出力対象となり、正しくローテーションされなくなることがあります。
- `rotatelog2` プログラムの引数指定に誤りがある場合、`rotatelog2` プログラムの起動に失敗しますが、Web サーバは起動します。この場合はログは出力されません。`rotatelog2` プログラムを指定した場合は、ログファイルが作成され、意図した分割が実施されることを確認してください。

3

Java EE サーバで使用する Deployment Descriptor (DD)

Java EE サーバの機能をアプリケーションやリソースに設定するためのファイルや、構成情報を定義するためのファイルを説明しています。

3.1 Java EE サーバで使用する DD の一覧

Java EE サーバで使用するすべての DD の概要を説明しています。

ファイル	概要	詳細説明
3.2 glassfish-acc.xml	アプリケーションクライアントコンテナの設定をするための Java EE RI の DD	アプリケーションクライアントコンテナに関する、Java EE RI 固有の設定をするための DD です。
3.4 glassfish-application.xml	Java EE アプリケーションの設定をするための Java EE RI の DD	Java EE アプリケーションに関する、Java EE RI 固有の設定をするための DD です。
3.6 glassfish-application-client.xml	アプリケーションクライアントの設定をするための Java EE RI の DD	アプリケーションクライアントに関する、Java EE RI 固有の設定をするための DD です。
3.8 glassfish-ejb-jar.xml	EJB-JAR の設定をするための Java EE RI の DD	EJB-JAR に関する、Java EE RI 固有の設定をするための DD です。
3.10 glassfish-resources.xml	リソースの設定をするための Java EE RI の DD	リソースに関する、Java EE RI 固有の設定をするための DD です。
3.12 glassfish-web.xml	Web アプリケーション (WAR) の設定をするための Java EE RI の DD	Web アプリケーション (WAR) に関する、Java EE RI 固有の設定をするための DD です。
3.14 hitachi-application.xml	Java EE アプリケーションの設定をするための拡張 DD	Java EE アプリケーションに関する、Java EE Server 独自の設定をするための DD です。
3.16 hitachi-application-client.xml	アプリケーションクライアントの設定をするための拡張 DD	アプリケーションクライアントに関する、Java EE Server 独自の設定をするための DD です。
3.18 hitachi-ejb-jar.xml	EJB-JAR の設定をするための拡張 DD	EJB-JAR に関する、Java EE Server 独自の設定をするための DD です。
3.20 hitachi-ra.xml	リソースの設定をするための拡張 DD	リソースに関する、Java EE Server 独自の設定をするための DD です。
3.22 hitachi-web.xml	Web アプリケーション (WAR) の設定をするための拡張 DD	Web アプリケーション (WAR) に関する、Java EE Server 独自の設定をするための DD です。

3.2 glassfish-acc.xml

アプリケーションクライアントコンテナの設定をするための Java EE RI の DD です。

説明

アプリケーションクライアントコンテナに関する、Java EE RI 固有の設定をするための DD です。

glassfish-acc.xml のルート要素は `client-container` です。 `client-container` 要素は、「要素の階層」で示すように、子要素と属性を含みます（それらの子要素と属性の配下にも、さらに、子要素と属性を含むことがあります）。

スキーマ

```
<!DOCTYPE client-container PUBLIC
"-//GlassFish.org//
DTD GlassFish Application Server 3.1 Application Client Container//EN"
"http://glassfish.org/dtds/glassfish-application-client-container_1_3.dtd">
```

格納先

DD ファイルは次の場所に格納されます。

`domain-dir/config` ディレクトリー

`domain-dir` : ドメインの設定ファイルが格納されているディレクトリーを表します。

要素の階層

```
client-container
. target-server
.. description
.. security
. auth-realm
.. property
. message-security-config
.. provider-config
... request-policy
... response-policy
... property
. property
```

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE client-container PUBLIC
"-//GlassFish.org//
DTD GlassFish Application Server 3.1 Application Client Container//EN"
"http://glassfish.org/dtds/glassfish-application-client-container_1_3.dtd">
<client-container>
```

```
<target-server name="vm006248" address="localhost" port="3700"/>
<message-security-config auth-layer="SOAP">
  <provider-config class-name=
"com.sun.xml.wss.provider.ClientSecurityAuthModule"
provider-id="XWS_ClientProvider" provider-type="client">
    <request-policy auth-source="content"/>
    <response-policy auth-source="content"/>
  </provider-config>
</message-security-config>
</client-container>
```

3.3 glassfish-acc.xml に設定する要素の詳細

glassfish-acc.xml に設定する要素について説明しています。

3.3.1 /client-container

アプリケーションクライアントコンテナに Java EE Server 固有の設定を定義します。

説明

アプリケーションクライアントコンテナに Java EE Server 固有の設定を定義します。client-container はルート要素です。そのため、glassfish-acc.xml ファイルには、client-container 要素は 1 つだけ存在します。

出現パターン

1 回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
send-password	true の場合、クライアント認証資格がサーバに送信される必要があることを指定します。クライアント認証資格がないと、保護された EJB コンポーネントへのすべてのアクセスは例外となります。	型：Boolean {true false}	true	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.2 /client-container/target-server

対象のサーバインスタンスの IIOP リスナーとロードバランスで使用される IIOP エンドポイントを指定します。

説明

対象のサーバインスタンスの IIOP リスナーとロードバランスで使用される IIOP エンドポイントを指定します。アプリケーションクライアントがデプロイされたサーバインスタンスがクラスターに参加する場

合、Java EE Server は、クラスター内で現在アクティブなすべての IIOP エンドポイントを自動的に検出します。

ただし、エンドポイントの 1 つで障害が発生した場合に備えて、クライアントはブートストラップ目的で少なくとも 2 つのエンドポイントを指定しておく必要があります。

リスナーまたはエンドポイントは、**ホスト : ポート** の形式で表されます。ホストは IP アドレスまたはホスト名、**ポート** はポート番号を表します。

出現パターン

1 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	クライアントコンテナによってアクセスされるサーバインスタンスの名前を指定します。	型 : String	—	—	1 回
address	このクライアントがアタッチするサーバのホスト名または IP アドレス (DNS で解決可能) を指定します。	型 : String	—	—	1 回
port	このクライアントがアタッチするサーバのネーミングサービスのポート番号を指定します。 新しいサーバインスタンスには 3700 以外のポート番号を割り当ててください。 Administration Console でポート番号を変更できます。	型 : Integer 0~65535	—	—	1 回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.3 /client-container/target-server/description

target-server 要素以下に含まれる要素の説明を記述します。

説明

target-server 要素以下に含まれる要素の説明を記述します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.3.4 /client-container/target-server/security

対象のサーバインスタンスとの IIOP/SSL 通信のセキュリティの設定を指定します。

説明

対象のサーバインスタンスとの IIOP/SSL 通信のセキュリティの設定を指定します。

出現パターン

0回、または1回

3.3.5 /client-container/auth-realm

JAAS 認証レルムのオプションを設定します。

説明

JAAS 認証レルムのオプションを設定します。JAAS は アプリケーションクライアントコンテナ上で使用できます。

認証レルムはプロバイダー固有のプロパティを要求します。プロバイダー固有のプロパティは特定の実装が何を必要とするかによって変わります。

依存関係のある要素または属性

auth-realm 要素の使用は、auth-realm 要素の name 属性の値によって決まります。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	<p>このレルムの名称を定義します。 サポートされているレルムは次のとおりです。</p> <p>file ファイルにユーザー情報を格納します。これは Application Server を最初にインストールしたときのデフォルトレルムです。</p> <p>ldap LDAP ディレクトリーにユーザー情報を格納します。</p> <p>jdbc データベースにユーザー情報を格納します。JDBC レルムでは、サーバはデータベースからユーザークレデンシャルを取得します。Java EE Server は、データベース情報と設定ファイル内の有効な JDBC レルムオプションを使用します。ダイジェスト認証の場合、JDBC レルムは jdbcDigestRealm で JAAS コンテキストとして作成される必要があります。</p> <p>certificate Java EE Server セキュリティーコンテキストにユーザー識別を設定します。そして、暗号化されたクライアント証明書から得られるユーザー情報を、ユーザー識別に追加します。</p>	型：String {file ldap jdbc certificate}	なし	—	1 回
class-name	このレルムを実装する Java クラスを定義します。	型：String {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.ent	なし	—	1 回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
		erprise.security.auth.realm.ldap.LDAPRealm Any customrealm .}			

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.6 /client-container/auth-realm/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型: String	なし	—	1回
value	プロパティの値を指定します。	型: String	なし	—	1回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.7 /client-container/message-security-config

メッセージセキュリティープロバイダーのための設定をします。

説明

メッセージセキュリティープロバイダーのための設定をします。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-layer	認証が行われるメッセージレイヤーを指定します。値は SOAP である必要があります。	型：String	—	—	1回
default-provider	特定サーバのメッセージセキュリティープロバイダーに、バインドされていないアプリケーションのために利用される、メッセージセキュリティープロバイダーを指定します。	型：String	—	—	0回、または1回
default-client-provider	特定のクライアントプロバイダーにバインドされていない任意のアプリケーションのために呼び出される、クライアントプロバイダーを指定します。	型：String	—	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.8 /client-container/message-security-config/provider-config

メッセージセキュリティープロバイダーのための設定をします。

説明

メッセージセキュリティープロバイダーのための設定をします。

request-policy と response-policy 子要素はオプションですが、 provider-config 要素はこれらを指定しないと動作しません。

プロバイダー固有のプロパティを設定する場合、property 子要素を使用してください。

プロパティの値は、initialize メソッドが呼び出された際にプロバイダーに渡されます。

出現パターン

1 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
provider-id	プロバイダー ID を指定します。	型：String	—	—	1 回
provider-type	プロバイダーが、クライアント、サーバまたはクライアントサーバのどの認証プロバイダーなのかを指定します。	型：String {client server client-server}	—	—	1 回
class-name	プロバイダーの Java 実装クラスを指定します。 有効なプロバイダーは次のとおりです。 <ul style="list-style-type: none">クライアント認証のプロバイダーは 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。サーバ認証のプロバイダーは 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。クライアントサーバ認証プロバイダーは両方のインターフェースを実装する必要があります。	型：String 次のインターフェースのどちらかまたは両方を実装する任意のクラス（サブクラス）名 <ul style="list-style-type: none">com.sun.enterprise.security.jauth.ClientAuthModulecom.sun.enterprise.security.jauth.ServerAuthModule	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.9 /client-container/message-security-config/provider-config/request-policy

認証プロバイダーのリクエスト処理の認証ポリシー要件を定義します。

説明

認証プロバイダーのリクエスト処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	sender (ユーザー名とパスワード) または content (デジタル署名) どちらかの、要求される認証のタイプを指定します。	型: String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型: String {before-content after-content}	—	—	1回

(凡例)

—: 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.10 /client-container/message-security-config/provider-config/response-policy

認証プロバイダーのレスポンス処理の認証ポリシー要件を定義します。

説明

認証プロバイダーのレスポンス処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	sender (ユーザー名とパスワード) または content (デジタル署名) どちらかの、要求される認証のタイプを指定します。	型: String {content sender}	—	—	1 回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型: String {before-content after-content}	—	—	1 回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.11 /client-container/message-security-config/provider-config/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型: String	—	—	1 回
value	プロパティの値を指定します。	型: String	—	—	1 回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.3.12 /client-container/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1回
value	プロパティの値を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.4 glassfish-application.xml

Java EE アプリケーションの設定をするための Java EE RI の DD です。

説明

Java EE アプリケーションに関する、Java EE RI 固有の設定をするための DD です。

glassfish-application.xml のルート要素は glassfish-application です。glassfish-application 要素は、「要素の階層」で示すように、子要素と属性を含みます（それらの子要素と属性の配下にも、さらに、子要素と属性を含むことがあります）。

スキーマ

```
<!DOCTYPE glassfish-application PUBLIC "-//GlassFish.org//DTD
GlassFish Application Server 3.1 Java EE Application 6.0//EN"
"http://glassfish.org/dtds/glassfish-application_6_0-1.dtd">
```

格納先

DD ファイルは次の場所に格納されます。

application_ear_name/META-INF ディレクトリー

application_ear_name : アプリケーションのアーカイブ名を表します。

要素の階層

```
glassfish-application
. web
.. web-uri
.. context-root
. security-role-mapping
.. role-name
.. principal-name
.. group-name
. realm
. ejb-ref
.. ejb-ref-name
.. jndi-name
. resource-ref
.. res-ref-name
.. jndi-name
.. default-resource-principal
... name
... password
. resource-env-ref
.. resource-env-ref-name
.. jndi-name
. service-ref
.. service-ref-name
```

```
.. port-info
... service-endpoint-interface
... wsdl-port
.... namespaceURI
.... localpart
... stub-property
.. service-impl-class
.. service-qname
... namespaceURI
... localpart
. message-destination-ref
.. message-destination-ref-name
.. jndi-name
. message-destination
.. message-destination-name
.. jndi-name
. archive-name
. version-identifier
```

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-application PUBLIC "-//GlassFish.org//DTD
GlassFish Application Server 3.1 Java EE Application 6.0//EN"
"http://glassfish.org/dtds/glassfish-application_6_0-1.dtd">
<glassfish-application>
  <unique-id>67488732739338240</unique-id>
  <web>
    <web-uri>hello.war</web-uri>
    <context-root>/hello</context-root>
  </web>
</glassfish-application>
```


3.5 glassfish-application.xml に設定する要素の詳細

glassfish-application.xml に設定する要素について説明しています。

3.5.1 /glassfish-application

アプリケーションに Java EE Server 固有の設定を定義します。

説明

アプリケーションに Java EE Server 固有の設定を定義します。glassfish-application はルート要素です。そのため、glassfish-application.xml ファイルには、glassfish-application 要素は 1 つだけ存在します。

出現パターン

1 回

3.5.2 /glassfish-application/web

アプリケーションの Web tier の設定をします。

説明

アプリケーションの Web tier の設定をします。

出現パターン

0 回以上

3.5.3 /glassfish-application/web/web-uri

アプリケーションの Web URI を含みます。

説明

アプリケーションの Web URI を含みます。これは application.xml ファイルの対応する要素に一致する必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.4 /glassfish-application/web/context-root

アプリケーションまたは WAR ファイルとしてパッケージされた Web アプリケーションの Web コンテキストルートを含みます。

説明

アプリケーションまたは WAR ファイルとしてパッケージされた Web アプリケーションの Web コンテキストルートを含みます。application.xml の対応する要素を上書きします。

ロードバランスを設定する場合、Web モジュールコンテキストルートはサービインスタンス内で一意である必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

*Application.xml*に指定された値

出現パターン

1 回

3.5.5 /glassfish-application/security-role-mapping

対応する Java EE XML ファイル内のロールをユーザーまたはグループにマッピングします。

説明

ロールを、現在アクティブなレルム内のユーザーまたはグループにマッピングします。

ロールマッピング要素は、EJB JAR DD (glassfish-ejb-jar.xml) 中のrole-name 要素で指定されたように、ロールを環境固有のユーザーまたはグループにマッピングします。

ロールマッピング要素がロールをユーザーにマッピングする場合、そのユーザーは現在のレルムに存在し、かつ、認証方法を使用してサーバにログインする必要があります。

ロールマッピング要素がロールをグループにマッピングする場合、レルムはグループをサポートし、かつ、そのグループは現在のレルムに存在している必要があります。現在のレルムに存在しているグループであるには、レルム内に、そのグループに属するユーザーが少なくとも 1 人いる必要があります。

出現パターン

0 回以上

3.5.6 /glassfish-application/security-role-mapping/role-name

対応する Java EE RI の DD ファイルの、security-role 要素内の role-name を含みます。

説明

対応する Java EE RI の DD ファイルの、security-role 要素内の role-name を含みます。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.7 /glassfish-application/security-role-mapping/principal-name

現在のレルム内のプリンシパル (ユーザー) 名を含みます。

説明

現在のレルム内のプリンシパル (ユーザー) 名を含みます。Enterprise Bean では、プリンシパルには run-as ロールが指定されている必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

依存関係のある要素または属性

<security-role-mapping><group-name>

出現パターン

group-name が存在しない場合は 1 回以上、それ以外は 0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
class-name	このプリンシパルに対応する、カスタムプリンシパル実装クラスを指定します。	型：String	com.sun.enterprise.deployment.PrincipalImpl	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.5.8 /glassfish-application/security-role-mapping/group-name

現在のレルム内のグループ名を含みます。

説明

現在のレルム内のグループ名を含みます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

依存関係のある要素または属性

<security-role-mapping><principal-name>

出現パターン

principal-name が存在しない場合は 1 回以上、それ以外は 0 回以上

3.5.9 /glassfish-application/realm

認証レルムを指定します。

説明

このアプリケーションと関連づけられているすべての認証要求を処理するのに使用されるレルムの名前を指定します。

この要素が指定されない、または設定されたレルムの名前と一致しない場合、デフォルトのレルムが使用されます。

サポートされているレルムは次のとおりです。

file

ファイルにユーザー情報を格納します。これは Application Server を最初にインストールした時のデフォルトレルムです。

ldap

LDAP ディレクトリーにユーザー情報を格納します。

jdbc

データベースにユーザー情報を格納します。

JDBC レルムでは、サーバはデータベースからユーザークレデンシャルを取得します。Java EE Server は、データベース情報と設定ファイル内の有効な JDBC レルムオプションを使用します。ダイジェスト認証の場合、JDBC レルムは jdbcDigestRealm で JAAS コンテキストとして作成される必要があります。

certificate

Java EE Server セキュリティーコンテキストにユーザー識別を設定します。そして、暗号化されたクライアント証明書から得られるユーザー情報を、ユーザー識別に追加します。

指定できる値

型：String

{file | ldap | jdbc | certificate}

デフォルト値

定義項目の省略

file

出現パターン

0回、または1回

3.5.10 /glassfish-application/ejb-ref

JNDI 完全修飾名を、対応する Java EE XML ファイル内の `ejb-ref` エントリーにマッピングします。

説明

対応する Java EE RI の DD ファイル `ejb-ref` エントリー内の `ejb-ref-name` を、リソースの JNDI 完全修飾名にマッピングします。

`ejb-ref` 要素は EJB ホームへの参照の宣言に使用されます。

Session Bean または Entity Bean に適用します。

出現パターン

0回以上

3.5.11 /glassfish-application/ejb-ref/ejb-ref-name

対応する Java EE RI の DD ファイル `ejb-ref` エントリー 内の `ejb-ref-name` を指定します。

説明

対応する Java EE RI の DD ファイル `ejb-ref` エントリー 内の `ejb-ref-name` を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.12 /glassfish-application/ejb-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、JNDI 完全修飾名の値は EJBHome オブジェクトのグローバル JNDI 名を指定します。これは Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、JNDI 完全修飾名は、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名になります。または、この情報は `mdb-resource-adapter` 要素の `activation-config` 子要素内に指定されます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.13 /glassfish-application/resource-ref

JNDI 完全修飾名を、対応する Java EE XML ファイル内の `resource-ref` 要素にマッピングします。

説明

対応する Java EE RI の DD ファイル `resource-ref` 要素内の `res-ref-name` を、リソースの JNDI 完全修飾名にマッピングします。

出現パターン

0 回以上

3.5.14 /glassfish-application/resource-ref/res-ref-name

`resource-ref` 要素内の `res-ref-name` を指定します。

説明

対応する Java EE RI の DD ファイル `resource-ref` 要素内の `res-ref-name` を指定します。`res-ref-name` 要素は、リソースマネージャー接続ファクトリー参照の名称を指定します。

名称は、Enterprise Bean 内で一意である必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.15 /glassfish-application/resource-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、JNDI 完全修飾名の値は EJBHome オブジェクトのグローバル JNDI 名を指定します。これは Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、JNDI 完全修飾名は、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名になります。または、この情報は `mdb-resource-adapter` 要素の `activation-config` 子要素内に指定されます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.16 /glassfish-application/resource-ref/default-resource-principal

デフォルトのリソースプリンシパル（ユーザー）を指定します。

説明

デフォルトのリソースプリンシパル（ユーザー）を指定します。この要素が JMS 接続ファクトリーのリソースと一緒に使用される場合、名称とパスワードの子要素は Message Queue ブローカーユーザーリポジトリ内の有効なエントリーである必要があります。

出現パターン

0 回、または 1 回

3.5.17 /glassfish-application/resource-ref/default-resource-principal/name

リソースマネージャーへサインオンするのに使用される、デフォルトのリソースプリンシパル名を指定します。

説明

リソースマネージャーへサインオンするのに使用される、デフォルトのリソースプリンシパル名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.18 /glassfish-application/resource-ref/default-resource-principal/password

デフォルトのリソースプリンシパルのパスワードを指定します。

説明

デフォルトのリソースプリンシパルのパスワードを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.19 /glassfish-application/resource-env-ref

JNDI 完全修飾名を、対応する Java EE XML ファイル内の `resource-env-ref` にマッピングします。

説明

対応する Java EE RI の DD ファイル `resource-env-ref` 要素内の `res-ref-name` を、リソースの JNDI 完全修飾名にマッピングします。

出現パターン

0 回以上

3.5.20 /glassfish-application/resource-env-ref/resource-env-ref-name

対応する Java EE RI の DD ファイル `resource-env-ref` 要素内の `res-ref-name` の値を指定します。

説明

対応する Java EE RI の DD ファイル `resource-env-ref` 要素内の `res-ref-name` の値を指定します。

指定できる値

型: String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.21 /glassfish-application/resource-env-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、JNDI 完全修飾名の値は EJBHome オブジェクトのグローバル JNDI 名を指定します。これは Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、JNDI 完全修飾名は、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名になります。または、この情報は `mdb-resource-adapter` 要素の `activation-config` 子要素内に指定されます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.22 /glassfish-application/service-ref

Web サービス参照のランタイム設定を指定します。

説明

Web サービス参照のランタイム設定を指定します。ランタイム情報は次の場合にだけ必要です。

- コンテナ管理対象ポートを解決するためのポートを定義する場合
- スタブオブジェクトに、デフォルトの Stub プロパティ設定、またはデフォルトの Call プロパティ設定を定義する場合
- 標準 DD の service-ref と関連づけられている URL の代わりに使用される、最終的な WSDL ドキュメントの URL を定義する場合

出現パターン

0 回以上

3.5.23 /glassfish-application/service-ref/service-ref-name

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.24 /glassfish-application/service-ref/port-info

Web サービス参照内のポートの情報を指定します。

説明

Web サービス参照内のポートの情報を指定します。service-endpoint-interface もしくは wsdl-port 、または両方が指定される必要があります。両方が指定された場合、wsdl-port は、コンテナがコンテナ管理対象ポートの選択で選ぶポートを指定します。

出現パターン

0 回以上

3.5.25 /glassfish-application/service-ref/port-info/service-endpoint-interface

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.5.26 /glassfish-application/service-ref/port-info/wsdl-port

WSDL ポートを指定します。

説明

WSDL ポートを指定します。

出現パターン

0回、または1回

3.5.27 /glassfish-application/service-ref/port-info/wsdl-port/ namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.5.28 /glassfish-application/service-ref/port-info/wsdl-port/ localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.29 /glassfish-application/service-ref/port-info/stub-property

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC スタブ実装がサポートする任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回
value	エンティティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.5.30 /glassfish-application/service-ref/service-impl-class

生成されたサービス実装クラスの名称を指定します。

説明

生成されたサービス実装クラスの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.5.31 /glassfish-application/service-ref/service-qname

参照されている WSDL service 要素を指定します。

説明

参照されている WSDL service 要素を指定します。

出現パターン

0回、または1回

3.5.32 /glassfish-application/service-ref/service-qname/ namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.33 /glassfish-application/service-ref/service-qname/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.34 /glassfish-application/message-destination-ref

メッセージの物理的な送信先の名称を指定します。

説明

メッセージの物理的な参照を、キュー、トピック、またはほかの物理的な送信先の JNDI 名に、直接バインドします。

対応する Java EE RI の DD ファイルのメッセージ送信先参照が、論理的なメッセージ送信先へのメッセージ送信先リンクを指定していない場合にだけ `message-destination-ref` を使用します。

出現パターン

0 回以上

3.5.35 /glassfish-application/message-destination-ref/message-destination-ref-name

対応する Java EE RI の DD ファイル内に定義された物理的なメッセージ送信先の名称を指定します。

説明

対応する Java EE RI の DD ファイル内に定義された物理的なメッセージ送信先の名称を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.36 /glassfish-application/message-destination-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、JNDI 完全修飾名の値は EJBHome オブジェクトのグローバル JNDI 名を指定します。これは Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、JNDI 完全修飾名は、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.37 /glassfish-application/message-destination

メッセージの論理的な送信先の名称を指定します。

説明

アプリケーション内に定義された論理的なメッセージ送信先の名称を指定します。メッセージ送信先名は、対応する Java EE RI の DD ファイル内の対応するメッセージ送信先名と一致します。

対応する Java EE RI の DD ファイルのメッセージ送信先参照が、論理的なメッセージ送信先へのメッセージ送信先リンクを指定している場合に使用します。

出現パターン

0 回以上

3.5.38 /glassfish-application/message-destination/message-destination-name

対応する Java EE RI の DD ファイル内に定義された論理的なメッセージ送信先の名称を指定します。

説明

対応する Java EE RI の DD ファイル内に定義された論理的なメッセージ送信先の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.39 /glassfish-application/message-destination/jndi-name

関連づけられているエンティティの jndi-name を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、JNDI 完全修飾名の値は EJBHome オブジェクトのグローバル JNDI 名を指定します。これは Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、JNDI 完全修飾名は、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名になります。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.5.40 /glassfish-application/archive-name

アーカイブファイルの名称を指定します。

説明

アーカイブファイルの名称を指定します。archive-name 要素の値は、display-name が application.xml ファイルに存在しない場合、デフォルトのアプリケーション名を得るのに使用されます。

デフォルトのアプリケーション名は、archive-name の値からファイルの拡張子を除いたものです。例えば、archive-name が appl.ear の場合、デフォルトのアプリケーション名は「appl」になります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.5.41 /glassfish-application/version-identifier

アプリケーションまたはモジュールのバージョン情報を含みます。

説明

アプリケーションまたはモジュールのバージョン情報を含みます。バージョン識別子は、モジュール名またはアプリケーション名の接尾辞です。コロン (:) で名前から区切ります。文字か数字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。

次の例では、アプリケーションの有効なバージョン識別子を示します。

- アプリケーション名:1
- アプリケーション名:BETA-2e
- アプリケーション名:3.8
- アプリケーション名:patch39875

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.6 glassfish-application-client.xml

アプリケーションクライアントの設定をするための Java EE RI の DD です。

説明

アプリケーションクライアントに関する、Java EE RI 固有の設定をするための DD です。

glassfish-application-client.xml のルート要素は glassfish-application-client です。glassfish-application-client 要素は、「要素の階層」で示すように、子要素と属性を含みます（それらの子要素と属性の配下にも、さらに、子要素と属性を含むことがあります）。

スキーマ

```
<!DOCTYPE glassfish-application-client PUBLIC "-//GlassFish.org
//DTD GlassFish Application Server 3.1 Java EE Application Client 6.0//EN"
"http://glassfish.org/dtds/glassfish-application-client_6_0-1.dtd">
```

格納先

DD ファイルは次の場所に格納されます。

client_jar_name/META-INF ディレクトリー

client_jar_name : クライアントアプリケーションのアーカイブ名を表します。

要素の階層

```
glassfish-application-client
. .ejb-ref
.. .ejb-ref-name
.. .jndi-name
.  .resource-ref
.. .res-ref-name
.. .jndi-name
.. .default-resource-principal
... .name
... .password
.  .resource-env-ref
.. .resource-env-ref-name
.. .jndi-name
.  .service-ref
.. .service-ref-name
.. .port-info
... .service-endpoint-interface
... .wsdl-port
.... .namespaceURI
.... .localpart
... .stub-property
... .call-property
... .message-security-binding
```

```
.... message-security
..... message
..... operation-name
..... request-protection
..... response-protection
.. call-property
.. wsdl-override
.. service-impl-class
.. service-qname
... namespaceURI
... localpart
. message-destination-ref
.. message-destination-ref-name
.. jndi-name
. message-destination
.. message-destination-name
.. jndi-name
. version-identifier
```

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-application-client PUBLIC ""-//GlassFish.org
//DTD GlassFish Application Server 3.1 Application Client 6.0//EN""
"http://glassfish.org/dtds/glassfish-application-client_6_0-1.dtd">
<glassfish-application-client>
  <message-destination-ref>
    <message-destination-ref-name>ClientQueue
  </message-destination-ref-name>
    <jndi-name>jms/security_mdb_OutQueue</jndi-name>
  </message-destination-ref>
</glassfish-application-client>
```


3.7 glassfish-application-client.xml に設定する要素の詳細

glassfish-application-client.xml に設定する要素について説明しています。

3.7.1 /glassfish-application-client

アプリケーションクライアントに Java EE Server 固有の設定を定義します。

説明

アプリケーションクライアントに Java EE Server 固有の設定を定義します。glassfish-application-client はルート要素です。そのため glassfish-application-client.xml ファイルには、glassfish-application-client 要素は 1 つだけ存在することになります。

出現パターン

1 回

3.7.2 /glassfish-application-client/ejb-ref

JNDI 完全修飾名を、対応する Java EE XML ファイル内の ejb-ref 名にマッピングします。

説明

対応する Java EE RI の DD ファイル ejb-ref エントリー内の ejb-ref-name を、リソースの JNDI 完全修飾名にマッピングします。

ejb-ref 要素は EJB ホームへの参照の宣言に使用されます。ejb-ref 要素は Session Bean または Entity Bean に適用します。

出現パターン

0 回以上

3.7.3 /glassfish-application-client/ejb-ref/ejb-ref-name

対応する Java EE RI の DD ファイル ejb-ref エントリー内の ejb-ref-name を指定します。

説明

対応する Java EE RI の DD ファイル ejb-ref エントリー内の ejb-ref-name を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.4 /glassfish-application-client/ejb-ref/jndi-name

リソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。Entity Bean と Session Bean の場合、EJBHome オブジェクトのグローバル JNDI 名を指定します。

Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.5 /glassfish-application-client/resource-ref

JNDI 完全修飾名を、対応する Java EE XML ファイル内の resource-ref 要素 にマッピングします。

説明

対応する Java EE RI の DD ファイル `resource-ref` 要素内の `res-ref-name` を、リソースの JNDI 完全修飾名にマッピングします。

出現パターン

0 回以上

3.7.6 /glassfish-application-client/resource-ref/res-ref-name

`resource-ref` 要素内の `res-ref-name` を指定します。

説明

対応する Java EE RI の DD ファイル `resource-ref` 要素内の `res-ref-name` を指定します。`res-ref-name` 要素は、リソースマネージャー接続ファクトリー参照の名称を指定します。

名称は、Enterprise Bean 内で一意である必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.7 /glassfish-application-client/resource-ref/jndi-name

リソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

JMS Message-driven Bean の場合、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.8 /glassfish-application-client/resource-ref/default-resource-principal

リソースの、デフォルトのプリンシパル（ユーザー）を指定します。

説明

リソースの、デフォルトのプリンシパル（ユーザー）を指定します。

この要素が JMS 接続ファクトリーのリソースと一緒に使用される場合、名称とパスワードの子要素は Message Queue ブローカーユーザーリポジトリ内で、有効なエントリーである必要があります。

出現パターン

0 回、または 1 回

3.7.9 /glassfish-application-client/resource-ref/default-resource-principal/name

リソースマネージャーへサインオンするために使用される、デフォルトのリソースプリンシパル名を指定します。

説明

リソースマネージャーへサインオンするために使用される、デフォルトのリソースプリンシパル名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.10 /glassfish-application-client/resource-ref/default-resource-principal/password

デフォルトのリソースプリンシパルのパスワードを指定します。

説明

デフォルトのリソースプリンシパルのパスワードを指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.11 /glassfish-application-client/resource-env-ref

JNDI 完全修飾名を、対応する Java EE XML ファイル内の resource-env-ref にマッピングします。

説明

対応する Java EE RI の DD ファイル resource-env-ref 要素内の res-ref-name を、リソースの JNDI 完全修飾名にマッピングします。

出現パターン

0 回以上

3.7.12 /glassfish-application-client/resource-env-ref/resource-env-ref-name

対応する Java EE RI の DD ファイル `resource-env-ref` 要素内の `res-ref-name` を指定します。

説明

対応する Java EE RI の DD ファイル `resource-env-ref` 要素内の `res-ref-name` を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.13 /glassfish-application-client/resource-env-ref/jndi-name

リソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。Entity Bean と Session Bean の場合、EJBHome オブジェクトのグローバル JNDI 名を指定します。

JNDI 完全修飾名は、Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、JNDI 完全修飾名は、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名を指定します。また、この情報は `mdb-resource-adapter` 要素の子要素である、`activation-config` 要素に指定することもできます。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.14 /glassfish-application-client/service-ref

Web サービス参照のランタイム設定を指定します。

説明

Web サービス参照のランタイム設定を指定します。

ランタイム情報は次の場合にだけ必要です。

- コンテナ管理対象ポートを解決するためのポートを定義する場合
- スタブオブジェクトに、デフォルトの Stub プロパティ設定、またはデフォルトの Call プロパティ設定を定義する場合
- 標準 DD の service-ref と関連づけられている URL の代わりに使用される、最終的な WSDL ドキュメントの URL を定義する場合

出現パターン

0 回以上

3.7.15 /glassfish-application-client/service-ref/service-ref-name

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.16 /glassfish-application-client/service-ref/port-info

Web サービス参照内のポートの情報を指定します。

説明

Web サービス参照内のポートの情報を指定します。

`service-endpoint-interface` もしくは `wsdl-port`、または両方が指定される必要があります。両方が指定された場合、`wsdl-port` は、コンテナがコンテナ管理対象ポートの選択で選択可能なポートを指定します。

`service-ref` 内で、同じ `wsdl-port` 値を 2 つ以上使用することはできません。

`service-endpoint-interface` 値 がコンテナ管理対象ポートの選択を使用する場合、同じ `service-ref` 内でほかの `port-info` 要素には使用できません。

出現パターン

0 回以上

3.7.17 /glassfish-application-client/service-ref/port-info/service-endpoint-interface

`java:comp/env` で始まる Web サービス参照名を指定します。

説明

`java:comp/env` で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.7.18 /glassfish-application-client/service-ref/port-info/wsdl-port

WSDL ポートを指定します。

説明

WSDL ポートを指定します。

出現パターン

0回、または1回

3.7.19 /glassfish-application-client/service-ref/port-info/wsdl-port/ namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.7.20 /glassfish-application-client/service-ref/port-info/wsdl-port/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.7.21 /glassfish-application-client/service-ref/port-info/stub-property

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC スタブ実装がサポートする任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
value	エンティティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.7.22 /glassfish-application-client/service-ref/port-info/call-property

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC Call 実装でサポートされる任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回
value	エンティティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.7.23 /glassfish-application-client/service-ref/port-info/message-security-binding

カスタム認証プロバイダーのバインディングを指定します。

説明

次の方法のどちらか、または両方を使用して、親要素のport-info に対して、カスタム認証プロバイダーのバインディングを指定します。

- 特定のプロバイダーにバインドする。
- プロバイダーによって課せられるメッセージのセキュリティー要件を指定する。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-layer	認証が行われるメッセージレイヤーを指定します。	型：String SOAP	なし	—	1回
provider-id	アプリケーション固有のメッセージのセキュリティー要件を満たすために使用される認証プロバイダーを指定します。この属性を指定しない場合、デフォルトのプロバイダーがメッセージレイヤーに定義されているとき、定義されたプロバイダーが使用されます。さらにデフォルトのプロバイダーも定義されていない場合、message-security-binding で定義された認証要件は実行されません。	型：String	なし	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.7.24 /glassfish-application-client/service-ref/port-info/message-security-binding/message-security

メッセージセキュリティー要件を指定します。

説明

メッセージセキュリティー要件を指定します。

2階層上の要素が port-info のため、メッセージセキュリティー要件は参照されたサービスのポートに付属します。

出現パターン

0回以上

3.7.25 /glassfish-application-client/service-ref/port-info/message-security-binding/message-security/message

メッセージセキュリティー要件が適用されるメソッドまたは操作を指定します。

説明

メッセージセキュリティー要件が適用されるメソッドまたは操作を指定します。

出現パターン

1回以上

3.7.26 /glassfish-application-client/service-ref/port-info/message-security-binding/message-security/message/operation-name

Web サービスの操作のオペレーション名を指定します。

説明

Web サービスの操作のオペレーション名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.7.27 /glassfish-application-client/service-ref/port-info/message-security-binding/message-security/request-protection

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

説明

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	sender (ユーザー名とパスワード) または content (デジタル署名) どちらかの、要求される認証のタイプを指定します。	型: String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型: String {before-content after-content}	—	—	1回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.7.28 /glassfish-application-client/service-ref/port-info/message-security-binding/message-security/response-protection

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

説明

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	sender (ユーザー名とパスワード) または content (デジタル署名) どちらかの、要求される認証のタイプを指定します。	型: String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型: String {before-content after-content}	—	—	1回

(凡例)

—: 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.7.29 /glassfish-application-client/service-ref/call-property

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC Call 実装でサポートされる任意のプロパティを選択できます。

出現パターン

0回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1回
value	エンティティの値を指定します。	型：String	なし	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.7.30 /glassfish-application-client/service-ref/wsdl-override

最終的な WSDL ドキュメントを示す有効な URL を指定します。

説明

最終的な WSDL ドキュメントを示す有効な URL を指定します。指定しない場合、標準 DD 内の service-ref と関連づけられている WSDL ドキュメントが使用されます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.7.31 /glassfish-application-client/service-ref/service-impl-class

生成されたサービス実装クラスの名称を指定します。

説明

生成されたサービス実装クラスの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.7.32 /glassfish-application-client/service-ref/service-qname

参照されている WSDL service 要素を指定します。

説明

参照されている WSDL service 要素を指定します。

出現パターン

0回、または1回

3.7.33 /glassfish-application-client/service-ref/service-qname/ namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.34 /glassfish-application-client/service-ref/service-qname/ localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.35 /glassfish-application-client/message-destination-ref

物理的なメッセージ送信先の名称を指定します。

説明

メッセージ送信先の参照を、キュー、トピック、またはほかの物理的送信先の JNDI 名に直接バインドします。

対応する Java EE RI の DD ファイル内のメッセージ送信先の参照が、論理メッセージ送信先への `message-destination-link` を指定しない場合だけ使用します。

出現パターン

0 回以上

3.7.36 /glassfish-application-client/message-destination-ref/ message-destination-ref-name

対応する Java EE RI の DD ファイル内に定義された、物理的なメッセージ送信先の名称を指定します。

説明

対応する Java EE RI の DD ファイル内に定義された、物理的なメッセージ送信先の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.37 /glassfish-application-client/message-destination-ref/jndi- name

関連づけられているエンティティの jndi-name を指定します。

説明

JMS Message-driven Bean の場合、Message-driven Bean が JMS メッセージを消費する、JMS リソースの JNDI 名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.38 /glassfish-application-client/message-destination

論理メッセージ送信先の名称を指定します。

説明

アプリケーション内で定義されている 論理メッセージ送信先の名称を指定します。

この要素の子要素であるmessage-destination-name は、対応する Java EE RI の DD ファイル内の、message-destination-name 要素に合わせて記述してください。

対応する Java EE RI の DD ファイル内のメッセージ送信先の参照で、論理メッセージ送信先への message-destination-link 要素 を指定している場合に利用してください。

出現パターン

0 回以上

3.7.39 /glassfish-application-client/message-destination/message-destination-name

対応する Java EE RI の DD ファイル内に定義された論理的なメッセージ送信先の名称を指定します。

説明

対応する Java EE RI の DD ファイル内に定義された論理的なメッセージ送信先の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.40 /glassfish-application-client/message-destination/jndi-name

関連づけられているエンティティの jndi-name を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、EJBHome オブジェクトのグローバル JNDI 名を指定します。この指定は、Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

JMS Message-driven Bean の場合、Message-driven Bean が JMS メッセージを消費する JMS リソースの JNDI 名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.7.41 /glassfish-application-client/version-identifier

アプリケーションまたはモジュールのバージョン情報を含みます。

説明

アプリケーションまたはモジュールのバージョン情報を含みます。バージョン識別子は、モジュールまたはアプリケーションの接尾辞です。コロン (:) で区切って名前に続けます。アルファベットまたは数字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 (_)、ダッシュ (-) およびピリオド (.) です。

アプリケーションの有効なバージョン識別子の例を次に示します。

- アプリケーション名:1
- アプリケーション名:BETA-2e
- アプリケーション名:3.8
- アプリケーション名:patch39875

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.8 glassfish-ejb-jar.xml

EJB-JAR の設定をするための Java EE RI の DD です。

説明

EJB-JAR に関する、Java EE RI 固有の設定をするための DD です。

glassfish-ejb-jar.xml のルート要素はglassfish-ejb-jar です。glassfish-ejb-jar 要素は、「要素の階層」で示すように、子要素と属性を含みます（それらの子要素と属性の配下にも、さらに、子要素と属性を含むことがあります）。

スキーマ

```
<!DOCTYPE glassfish-ejb-jar PUBLIC
"-//GlassFish.org//DTD GlassFish Application Server 3.1 EJB 3.1//EN"
"http://glassfish.org/dtds/glassfish-ejb-jar_3_1-1.dtd">
```

格納先

DD ファイルは次の場所に格納されます。

ejb_jar_name/META-INF ディレクトリー

ejb_jar_name : EJB アプリケーションのアーカイブ名を表します。

要素の階層

```
glassfish-ejb-jar
. security-role-mapping
. . role-name
. . principal-name
. . group-name
. enterprise-beans
. . name
. . ejb
. . . ejb-name
. . . jndi-name
. . . ejb-ref
. . . . ejb-ref-name
. . . . jndi-name
. . . resource-ref
. . . . res-ref-name
. . . . jndi-name
. . . . default-resource-principal
. . . . . name
. . . . . password
. . . resource-env-ref
. . . . resource-env-ref-name
. . . . jndi-name
. . . service-ref
```

- service-ref-name
- port-info
- service-endpoint-interface
- wsdl-port
- namespaceURI
- localpart
- stub-property
- call-property
- message-security-binding
- message-security
- message
- java-method
- method-name
- method-params
- method-param
- operation-name
- request-protection
- response-protection
- call-property
- wsdl-override
- service-impl-class
- service-qname
- namespaceURI
- localpart
- message-destination-ref
- message-destination-ref-name
- jndi-name
- principal
- name
- mdb-connection-factory
- jndi-name
- default-resource-principal
- name
- password
- jms-durable-subscription-name
- jms-max-messages-load
- ior-security-config
- transport-config
- integrity
- confidentiality
- establish-trust-in-target
- establish-trust-in-client
- as-context
- auth-method
- realm
- required
- commit-option
- cmt-timeout-in-seconds
- bean-pool
- steady-pool-size
- resize-quantity
- max-pool-size
- pool-idle-timeout-in-seconds
- mdb-resource-adapter
- resource-adapter-mid
- activation-config
- description
- activation-config-property


```

. . . . . activation-config-property-name
. . . . . activation-config-property-value
. . . webservice-endpoint
. . . . port-component-name
. . . . endpoint-address-uri
. . . . login-config
. . . . . auth-method
. . . . . realm
. . . . message-security-binding
. . . . . message-security
. . . . . message
. . . . . . java-method
. . . . . . . method-name
. . . . . . . method-params
. . . . . . . method-param
. . . . . . operation-name
. . . . . request-protection
. . . . . response-protection
. . . . transport-guarantee
. . . . service-qname
. . . . namespaceURI
. . . . . localpart
. . . . tie-class
. . . . servlet-impl-class
. . . . debugging-enabled
. . . . property
. . message-destination
. . . message-destination-name
. . . jndi-name
. . webservice-description
. . . webservice-description-name
. . . wsdl-publish-location
. version-identifier

```

記述例

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-ejb-jar PUBLIC
"-//GlassFish.org//DTD GlassFish Application Server 3.1 EJB 3.1//EN"
"http://glassfish.org/dtds/glassfish-ejb-jar_3_1-1.dtd">
<glassfish-ejb-jar>
  <ejb>
    <ejb-name>CustomerEJB</ejb-name>
    <jndi-name>customer</jndi-name>
    <bean-pool>
      <steady-pool-size>10</steady-pool-size>
    </bean-pool>
  </ejb>
</glassfish-ejb-jar>

```

3.9 glassfish-ejb-jar.xml に設定する要素の詳細

glassfish-ejb-jar.xml に設定する要素について説明しています。

3.9.1 /glassfish-ejb-jar

EJB-JAR ファイルに、Java EE Server 固有の設定を定義します。

説明

EJB-JAR ファイルに、Java EE Server 固有の設定を定義します。これはルート要素です。glassfish-ejb-jar.xml ファイルには、glassfish-ejb-jar 要素は 1 つだけ存在します。

出現パターン

1 回

3.9.2 /glassfish-ejb-jar/security-role-mapping

標準 DD で指定したロールを、環境固有のユーザーまたはグループにマッピングするための要素です。

説明

ロールを、現在アクティブなレルム内のユーザーまたはグループにマッピングします。

ロールマッピング要素は、EJB JAR role-name エントリーで指定されたように、ロールを環境固有のユーザーまたはグループにマッピングします。ユーザーにマッピングする場合、そのユーザーは現在のレルムに存在する実在のユーザーで、現在の認証方法を使用してサーバにログインできる必要があります。グループにマッピングする場合、レルムはグループをサポートする必要があり、そのグループは現在のレルムに存在する実在のグループである必要があります。

この設定を利用するためには、指定したレルム内の、指定したグループに属するユーザーが最低 1 人必要です。

出現パターン

0 回以上

3.9.3 /glassfish-ejb-jar/security-role-mapping/role-name

標準 DD で指定したロール名を指定する要素です。

説明

標準 DD で指定したロール名を指定する要素です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.4 /glassfish-ejb-jar/security-role-mapping/principal-name

現在のレルムのプリンシパル（ユーザー）名を指定します。

説明

現在のレルムのプリンシパル（ユーザー）名を指定します。プリンシパル（ユーザ名）は、EJB-JAR で、run-as で指定されたロールである必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

依存関係のある要素または属性

group-name

出現パターン

group-name を使用しない場合は 1 回以上、それ以外は 0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
class-name	このプリンシパルに対応する、カスタムプリンシパル実装クラスを指定します。	型：String java.security.Principal インターフェースを実装するクラス名（サブクラス名）	org.glassfish.security.common.PrincipalImpl	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.5 /glassfish-ejb-jar/security-role-mapping/group-name

現在のレルムのグループ名を指定します。

説明

ロールにグループをマッピングするとき、レルムのグループ名を指定する要素です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

依存関係のある要素または属性

principal-name

出現パターン

principal-name を使用しない場合は1回以上、それ以外は0回以上

3.9.6 /glassfish-ejb-jar/enterprise-beans

EJB-JAR の実行時の設定をするための要素です。

説明

EJB-JAR ファイルの実行用の設定を指定するための要素です。

出現パターン

1 回

3.9.7 /glassfish-ejb-jar/enterprise-beans/name

enterprise-beans 要素の名称を指定します。

説明

enterprise-beans 要素の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0 回、または 1 回

3.9.8 /glassfish-ejb-jar/enterprise-beans/ejb

EJB-JAR に含まれる Enterprise Bean のための、実行時の設定を指定するための要素です。

説明

この EJB-JAR に含まれる Enterprise Bean の、実行時の設定をするための要素です。子要素として指定できる要素は、個々の Enterprise Bean 種別に、次のようになります。

- すべてのタイプの Bean

ejb-name、ejb-ref、resource-ref、resource-env-ref、ior-security-config、jndi-name、message-destination-ref、service-ref

- Stateless Session Bean
bean-pool、webservice-endpoint
- Stateful Session Bean
webservice-endpoint
- Entity Bean
commit-option、bean-pool
- Message-driven Bean
mdb-resource-adapter、mdb-connection-factory、jms-durable-subscription-name、jms-max-messages-load、bean-pool

出現パターン

0 回以上

3.9.9 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-name

ejb-jar.xml で定義した Enterprise Bean の `ejb-name` 要素の値を指定するための要素です。

説明

ejb-jar.xml で指定する名称は、対応する ejb-jar.xml の `ejb-name` で指定した名称と一致する必要があります。この `ejb-name` は同一 EJB-JAR でユニーク（一意）でなければなりません。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.10 /glassfish-ejb-jar/enterprise-beans/ejb/jndi-name

JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の場合、この値は EJBHome オブジェクトのグローバル JNDI 名を指定します。これは Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0 回以上

3.9.11 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-ref

標準 DD の `ejb-ref` 要素に対して EJB のホームオブジェクトの JNDI 完全修飾名のマッピングを定義する要素です。

説明

標準 DD で定義した、`ejb-ref` 要素の `ejb-ref-name` 子要素で指定した名称を、JNDI 完全修飾名とマッピングします。`ejb-ref` 要素は、EJB のホームオブジェクトへの参照の宣言に使用されます。この設定は、Session Bean または Entity Bean で利用可能です。

出現パターン

0 回以上

3.9.12 /glassfish-ejb-jar/enterprise-beans/ejb/ejb-ref/ejb-ref-name

標準 DD の `ejb-ref` 要素の `ejb-ref-name` を指定するための要素です

説明

標準 DD で定義した `ejb-ref` 要素の `ejb-ref-name` を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.13 /glassfish-`ejb-jar`/enterprise-beans/`ejb`/`ejb-ref`/`jndi-name`

`ejb-ref` を定義する Enterprise Bean の JNDI 完全修飾名を指定する要素です。

説明

`ejb-ref` で参照を定義する Enterprise Bean の JNDI 完全修飾名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.14 /glassfish-`ejb-jar`/enterprise-beans/`ejb`/`resource-ref`

標準 DD の `resource-ref` 要素に対する参照のマッピングを定義する要素です。

説明

標準 DD で定義した、`resource-ref` 要素の `res-ref-name` 子要素で指定した名前を、JNDI 完全修飾名とマッピングします。

注意事項

JMS 接続ファクトリーから獲得したコネクションは、現在リリースされている Java EE Server では共有できません。`ejb-jar.xml` ファイル `resource-ref` 要素の `res-sharing-scope` 要素は JMS 接続ファクトリーでは無視されます。`resource-ref` が Java EE Server の Message Queue に JMS 接続ファクトリーを指定する場合、`default-resource-principal` (ユーザー名/パスワード) が Message Queue のユーザーリポジトリに存在している必要があります。

出現パターン

0 回以上

3.9.15 /glassfish-`ejb-jar`/enterprise-beans/`ejb`/resource-ref/`res-ref-name`

標準 DD で定義した `resource-ref` の `res-ref-name` を指定する要素です。

説明

`glassfish-ejb-jar.xml` の `res-ref-name` で指定する名称は、標準 DD で定義した `resource-ref` の `res-ref-name` で指定した名称と一致する必要があります。また、この `res-ref-name` は、同一 EJB-JAR でユニーク (一意) でなければなりません。

指定できる値

型: `String`

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.16 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/jndi-name

resource-ref を定義するリソース JNDI 完全修飾名を指定する要素です。

説明

resource-ref で参照を定義するリソースの JNDI 完全修飾名を指定するための要素です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.17 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/default-resource-principal

リソースで、デフォルトで利用するプリンシパル（ユーザー）を指定します。

説明

デフォルトのリソースプリンシパル（ユーザー）を指定します。

この要素が JMS 接続ファクトリーのリソースと一緒に使用される場合、名称とパスワードの子要素は Message Queue ブローカーユーザーリポジトリ内の有効なエントリーである必要があります。

出現パターン

0 回、または 1 回

3.9.18 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/default-resource-principal/name

リソースマネージャーへサインオンするために使用される、デフォルトのユーザー名を指定します。

説明

リソースマネージャーへサインオンするために使用される、デフォルトのユーザー名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.19 /glassfish-ejb-jar/enterprise-beans/ejb/resource-ref/default-resource-principal/password

リソースマネージャーへサインオンするために使用される、デフォルトのユーザ名に対応したパスワードを指定します。

説明

リソースマネージャーへサインオンするために使用される、デフォルトのユーザ名に対応したパスワードを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.20 /glassfish-ejb-jar/enterprise-beans/ejb/resource-env-ref

標準 DD の resource-env-ref 要素に対する参照のマッピングを定義する要素です。

説明

標準 DD で定義した、resource-env-ref 要素の res-ref-name 子要素で指定した名前を、JNDI 完全修飾名とマッピングします。

出現パターン

0 回以上

3.9.21 /glassfish-ejb-jar/enterprise-beans/ejb/resource-env-ref/ resource-env-ref-name

標準 DD で定義した resource-env-ref の res-ref-name を指定する要素です。

説明

glassfish-ejb-jar.xml の res-ref-name で指定する名称は、標準 DD で定義した res-ref-name で指定した名称と一致する必要があります。また、この res-ref-name は、同一 EJB-JAR でユニーク（一意）でなければなりません。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.22 /glassfish-ejb-jar/enterprise-beans/ejb/resource-env-ref/ jndi-name

resource-env-ref を定義するリソース JNDI 完全修飾名を指定する要素です

説明

resource-env-ref で参照を定義するリソースの JNDI 完全修飾名を指定するための要素です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.23 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref

Web サービスの参照のための、実行時設定をする要素です。

説明

実行時設定は、次の場合にだけ必要です。

- コンテナによって管理されるポートの参照を解決する場合。
- スタブオブジェクトに、デフォルトの Stub の設定、またはデフォルトの Call の設定を定義する場合。
- 標準 DD で、service-ref 要素を利用して関連づけられた定義の代わりに、WSDL ドキュメントの URL を定義する場合。

出現パターン

0 回以上

3.9.24 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-ref-name

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.25 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info

Web サービス参照内のポートの情報を指定します。

説明

service-endpoint-interface または wsdl-port、もしくは両方を指定する必要があります。両方が指定された場合、wsdl-port は、コンテナーがコンテナー管理対象ポートの選択で選ぶポートを指定します。同じ wsdl-port 値は、同じ service-ref 内の 1 つの port-info 要素でしか使用できません。service-endpoint-interface がコンテナー管理対象ポートの選択を使用する場合、その値は同じ service-ref 内の 1 つの port-info 要素でしか使用できません。

出現パターン

0 回以上

3.9.26 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ service-endpoint-interface

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.27 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ wsdl-port

WSDL ポートを指定します。

説明

WSDL ポートを指定します。

出現パターン

0回、または1回

3.9.28 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ wsdl-port/namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.9.29 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/wsdl-port/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.9.30 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/stub-property

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC スタブ実装がサポートする任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
value	関連するネームエンティティの値を指定します。	型：String	なし	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.31 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/call-property

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC Call 実装でサポートされる任意のプロパティを選択できます。

出現パターン

0回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1回
value	エンティティの値を指定します。	型：String	なし	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.32 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-binding

カスタム認証プロバイダーのバインディングを定義します。

説明

Web サービスの参照に対する、Web Service Security の設定を行うための要素です。port-info 要素でバインドするカスタム認証プロバイダーを、次のどれか、または両方の方法を用いて指定します。

- 特定のカスタム認証プロバイダーを指定する。
- カスタム認証プロバイダーによって提供されるメッセージセキュリティ要件を指定する。

出現パターン

0 回、または 1 回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-layer	認証が行われるメッセージレイヤーを指定します。	型：String SOAP	—	—	1 回
provider-id	アプリケーション固有のメッセージのセキュリティ要件を満たすために使用される認証プロバイダーを指定します。この属性を指定しない場合、デフォルトのプロバイダーがメッセージレイヤーに定義されているとき、定義されたプロバイダーが使用されます。さらにデフォルトのプロバイダーも定義されていない場合、message-security-binding で定義された認証要件は実行されません。	型：String	—	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.33 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-binding/message-security

メッセージセキュリティ要件を指定します。

説明

メッセージセキュリティ要件を指定します。

指定されたメッセージセキュリティ要件は、参照されるサービスのポートに対して適用します。

出現パターン

0 回以上

3.9.34 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ message-security-binding/message-security/message

メッセージセキュリティ要件が適用されるメソッドまたは操作を指定します。

説明

メッセージセキュリティ要件が適用されるメソッドまたは操作を指定します。

出現パターン

1 回以上

3.9.35 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ message-security-binding/message-security/message/java- method

メッセージセキュリティ要件が適用されるメソッドまたは操作を指定します。

説明

メッセージセキュリティ要件を適用するメソッドを指定します。

出現パターン

0 回、または 1 回

3.9.36 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-binding/message-security/message/java-method/method-name

メソッド名を指定します。

説明

メソッド名を指定、またはアスタリスク (*) によってすべてのメソッドを指定します。メソッドがオーバーロードされる場合、すべてのメソッドを同じ名称で指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.37 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-binding/message-security/message/java-method/method-params

メソッド引数を指定します。

説明

メソッドの引数を指定します。

出現パターン

0 回、または 1 回

3.9.38 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ message-security-binding/message-security/message/java- method/method-params/method-param

メソッドの引数の完全修飾名を指定します。

説明

メソッドの引数の完全修飾名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回以上

3.9.39 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/ message-security-binding/message-security/message/ operation-name

Web サービスの操作のオペレーション名を指定します。

説明

Web サービスの操作のオペレーション名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.40 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-binding/message-security/request-protection

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

説明

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	リクエスト処理の認証ポリシー要件によって要求される認証タイプを指定します。ユーザー名とパスワードを利用する場合はsender、デジタル署名を利用する場合にはcontentを指定します。	型：String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型：String {before-content after-content}	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.41 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/port-info/message-security-binding/message-security/response-protection

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

説明

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	レスポンス処理の認証ポリシー要件によって要求される認証タイプを指定します。ユーザー名とパスワードを利用する場合はsender、デジタル署名を利用する場合にはcontentを指定します。	型：String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型：String {before-content after-content}	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.42 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/call-property

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定できる JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC Call 実装がサポートする任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回
value	エンティティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.43 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/wsdl-override

最終的な WSDL ドキュメントを示す有効な URL を指定します。

説明

最終的な WSDL ドキュメントを示す有効な URL を指定します。有効な URL を指定しない場合、標準 DD の service-ref で指定された WSDL ドキュメントが使用されます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.44 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-impl-class

サービス実装クラスの名称を指定します。

説明

サービス実装クラスの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.45 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-qname

Enterprise Bean から参照される WSDL サービスを指定します。

説明

Enterprise Bean から参照される WSDL サービスを指定します。

出現パターン

0回、または1回

3.9.46 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-qname/namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.47 /glassfish-ejb-jar/enterprise-beans/ejb/service-ref/service-qname/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.48 /glassfish-ejb-jar/enterprise-beans/ejb/message-destination-ref

物理的なメッセージ送信先の参照を定義するための要素です。

説明

この要素を利用して、メッセージ送信先の Queue、Topic、または他の物理的送信先の JNDI 名に、メッセージの送信先の参照をバインドします。

標準 DD で、メッセージ送信先の参照として、論理的な message-destination への message-destination-link を指定しない場合にだけ使用します。

出現パターン

0 回以上

3.9.49 /glassfish-ejb-jar/enterprise-beans/ejb/message-destination-ref/message-destination-ref-name

標準 DD で指定された message-destination-ref-name を指定する要素です。

説明

標準 DD で指定された message-destination-ref-name を指定する要素です。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.50 /glassfish-ejb-jar/enterprise-beans/ejb/message-destination-ref/jndi-name

参照に対して関連づける、物理的なメッセージ送信先の JNDI 名を指定する要素です。

説明

参照に対して関連づける、物理的なメッセージ送信先の JNDI 名を指定する要素です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.51 /glassfish-ejb-jar/enterprise-beans/ejb/principal

run-as ロールが指定された Enterprise Bean 内のプリンシパル（ユーザー）名を指定します。

説明

プラットフォームにユーザー名を指定します。

出現パターン

0 回、または 1 回

3.9.52 /glassfish-ejb-jar/enterprise-beans/ejb/principal/name

ユーザーの名称を指定します。

説明

ユーザーの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.53 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory

Message-driven Bean に関連づける ConnectionFactory を指定します。

説明

Message-driven Bean に関連づける ConnectionFactory を指定します。キューまたはトピックは、Message driven Bean に関連づける JMS のメッセージ受信タイプと一致している必要があります。

出現パターン

0 回、または 1 回

3.9.54 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/jndi-name

Message-driven Bean に関連づける ConnectionFactory の参照を取得するための JNDI 完全修飾名を指定します。

説明

Message-driven Bean に関連づける ConnectionFactory の参照を取得するための JNDI 完全修飾名を指定します。ここで指定するのは、Message-driven Bean に、JMS メッセージを配信する JMS リソースの JNDI 名となります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.9.55 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/default-resource-principal

リソースマネージャーへのデフォルトのサインオン（名称とパスワード）を指定します。

説明

リソースの、デフォルトのプリンシパル（ユーザー）を指定します。

この要素が JMS コネクションファクトリーのリソースと一緒に使用される場合、ユーザー名称とパスワードの子要素は Message Queue ブローカーユーザーリポジトリ内の有効なエントリーである必要があります。

出現パターン

0回、または1回

3.9.56 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/default-resource-principal/name

リソースマネージャーへサインオンするために使用される、デフォルトのリソースプリンシパル名を指定します。

説明

リソースマネージャーへサインオンするために使用される、デフォルトのリソースプリンシパル名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.57 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-connection-factory/default-resource-principal/password

デフォルトのリソースプリンシパルのパスワードを指定します。

説明

デフォルトのリソースプリンシパルのパスワードを指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.58 /glassfish-ejb-jar/enterprise-beans/ejb/jms-durable-subscription-name

Message-driven Bean に継続的に関連づけられるサブスクリプションを指定します。

説明

Message-driven Bean に継続的に関連づけられるサブスクリプションを指定します。この要素は、デスティネーションタイプが JMS の Topic で、サブスクリプションの継続性の定義が Durable の場合にだけ指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.59 /glassfish-ejb-jar/enterprise-beans/ejb/jms-max-messages-load

Message-driven Bean による処理で、一度に JMS セッションに読み込む最大メッセージ数を指定します。

説明

Message-driven Bean による処理で、一度に JMS セッションに読み込む最大メッセージ数を指定します。

指定できる値

型：Integer

1～2147483647

デフォルト値

定義項目の省略

1

出現パターン

0回、または1回

3.9.60 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config

IOR のセキュリティー情報を指定します。

説明

IOR のセキュリティー情報を指定します。

出現パターン

0回、または1回

3.9.61 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config

トランスポートのセキュリティー情報を指定します。

説明

トランスポートのセキュリティー情報を指定します。

出現パターン

0回、または1回

3.9.62 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/integrity

ターゲットが、整合性が保護されたメッセージをサポートするかどうかを指定します。

説明

ターゲットが、整合性が保護されたメッセージをサポートするかどうかを指定します。

指定できる値

型：String

{NONE | SUPPORTED | REQUIRED}

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.9.63 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/confidentiality

ターゲットが、プライバシーが保護されたメッセージをサポートするかどうかを指定します。

説明

ターゲットが、プライバシーが保護されたメッセージをサポートするかどうかを指定します。

指定できる値

型：String

{NONE | SUPPORTED | REQUIRED}

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.64 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/establish-trust-in-target

ターゲットが、クライアントに対して認証できるかどうか指定します。

説明

ターゲットが、クライアントに対して認証できるかどうか指定します。

指定できる値

型：String

{NONE | SUPPORTED | REQUIRED}

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.65 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/transport-config/establish-trust-in-client

ターゲットがクライアントを認証できるかどうか指定します。

説明

ターゲットがクライアントを認証できるかどうか指定します。

指定できる値

型：String

{NONE | SUPPORTED | REQUIRED}

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.66 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context

クライアントの認証に使用される認証の方法を指定します。

説明

クライアントの認証に使用される認証の方法を指定します。

出現パターン

0 回、または 1 回

3.9.67 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context/auth-method

認証方式を指定します。

説明

認証方式を指定します。値は、*USERNAME_PASSWORD* の形式で指定します。

指定できる値

型：String

USERNAME_PASSWORD

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.68 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context/realm

ユーザーが認証される、レルムを指定します。

説明

このアプリケーションと関連づけられているすべての認証要求を処理するレルムの名称を指定します。この要素が指定されない、または設定されたレルムの名称と一致しない場合、デフォルトのレルムが使用されます。

サポートされているレルムは次のとおりです。

file

ファイルにユーザー情報を格納します。Application Server を最初にインストールしたときの、デフォルトのレルムです。

ldap

LDAP ディレクトリーにユーザー情報を格納します。

jdbc

データベースにユーザー情報を格納します。

JDBC レalmでは、サーバはデータベースからユーザークレデンシャルを取得します。Java EE Server は、データベース情報と設定ファイル内の有効な JDBC レalmオプションを使用します。ダイジェスト認証の場合、JDBC レalmは jdbcDigestRealm で JAAS コンテキストとして作成される必要があります。

certificate

Java EE Server セキュリティーコンテキストにユーザー識別を設定します。そして、暗号化されたクライアント証明書から得られるユーザー情報を、ユーザー識別に追加します。

指定できる値

型：String

{file | ldap | jdbc | certificate}

デフォルト値

定義項目の省略

file

出現パターン

1 回

3.9.69 /glassfish-ejb-jar/enterprise-beans/ejb/ior-security-config/as-context/required

auth-method 要素で指定された認証方式がクライアント認証に使用される必要があるかどうかを指定します。

説明

auth-method 要素で指定された認証方式がクライアント認証に使用される必要があるかどうかを指定します。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

false

出現パターン

1 回

3.9.70 /glassfish-ejb-jar/enterprise-beans/ejb/commit-option

トランザクション完了時に使用されるコミットオプションを指定します。

説明

トランザクション完了時に使用されるコミットオプションを指定します。Java EE Server での有効な値は B または C です。Entity Bean に適用されます。

指定できる値

型 : String

{B | C}

デフォルト値

定義項目の省略

B

出現パターン

0 回、または 1 回

3.9.71 /glassfish-ejb-jar/enterprise-beans/ejb/cmt-timeout-in-seconds

トランザクションタイムアウト時間を指定します。

説明

各 Bean でのトランザクションタイムアウト時間の設定を上書きします。0 を指定すると、トランザクションサービスにおけるデフォルトのトランザクションタイムアウト時間が適用されます。正数を指定した場

合、新たに CMT のトランザクションを開始した Bean の全てのメソッドで、この値が利用されます。この Bean が、クライアントトランザクションに参加する場合、この値は利用されません

指定できる値

型：Integer

0~2147483647

デフォルト値

定義項目の省略

0

出現パターン

0回、または1回

3.9.72 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool

EJB のインスタンスプールの設定を行う要素です。

説明

EJB のインスタンスプールの設定を行う要素です。Stateless Session Bean または Entity Bean 、Message-driven Bean で使用されます。

出現パターン

0回、または1回

3.9.73 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/steady-pool-size

EJB のインスタンスプールが保持する、Bean の初期値と最少数を指定します。

説明

EJB のインスタンスプールが保持する、Bean の初期値と最少数を指定します。Stateless Session Bean と Message-driven Bean に適用します。

指定できる値

型：Integer

0~2147483647

デフォルト値

定義項目の省略

asadmin ユーティリティーコマンドのset サブコマンドで使用する、`configs.config.Java EEサーバの構成名.ejb-container.steady-pool-size` パラメーターで設定されている値

出現パターン

0回、または1回

3.9.74 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/resize-quantity

EJB のインスタンスプールに対して、同時に作成または削除される Bean のインスタンス数を指定します。

説明

次の Bean インスタンスの数を指定します。

プールが `steady-pool-size` より少ない数の Bean のインスタンスを保持するとき、リクエストを受けると、指定した数の Bean インスタンスが作成されます。

`pool-idle-timeout-in-seconds` を経過し、利用されていない Bean インスタンスをクリーンアップスレッドが削除するとき、指定した数の Bean インスタンスが削除されます。

指定できる値

型：Integer

0~2147483647

デフォルト値

定義項目の省略

16

出現パターン

0回、または1回

3.9.75 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/max-pool-size

EJB のインスタンスプールが保持する、Bean の最大数を指定します。

説明

EJB のインスタンスプールが保持する、Bean の最大数を指定します。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

asadmin ユーティリティーコマンドのset サブコマンドで使用する、configs.config.Java EEサーバの構成名ejb-container.max-pool-size パラメーターで設定されている値

出現パターン

0回、または1回

3.9.76 /glassfish-ejb-jar/enterprise-beans/ejb/bean-pool/pool-idle-timeout-in-seconds

EJB のインスタンスプールで、アイドル状態で保持される最大時間を指定します。

説明

EJB のインスタンスプールで、アイドル状態で保持される最大時間を指定します。この要素で指定した時間を経過すると、EJB インスタンスプール内の Bean のインスタンスは、パッシベートまたは削除の候補になります。0 を指定した場合、Bean インスタンスは経過時間によるパッシベートまたは削除の候補にはなりません。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

600

出現パターン

0回、または1回

3.9.77 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter

Message-driven Bean の実行時の設定をします。

説明

Message-driven Bean の実行時の設定をします。

出現パターン

0回、または1回

3.9.78 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/resource-adapter-mid

リソースアダプターモジュール ID を指定します。

説明

Message-driven Bean にメッセージを配信するリソースアダプターのモジュール ID を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.79 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config

Message-driven Bean にメッセージを配信するリソースアダプターに渡す activation-config を指定します。

説明

Message-driven Bean にメッセージを配信するリソースアダプターに渡す activation-config を指定します。例えば、物理的な JMS 送信先の名称といった情報も含まれます。ejb-jar.xml ファイル内の activation-config 要素を上書きすることもできます。

出現パターン

1 回以上

3.9.80 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/description

activation-config 要素の説明を記述します。

説明

activation-config 要素の説明を記述します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

0 回、または 1 回

3.9.81 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/activation-config-property

activation-config で設定するプロパティを指定します。

説明

activation-config で設定するプロパティを指定します。

出現パターン

1 回以上

3.9.82 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/activation-config-property/activation-config-property-name

activation-config で設定するプロパティ名称を指定します。

説明

activation-config で設定するプロパティ名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.83 /glassfish-ejb-jar/enterprise-beans/ejb/mdb-resource-adapter/activation-config/activation-config-property/activation-config-property-value

activation-config で設定するプロパティ値を指定します。

説明

activation-config で設定するプロパティ値を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.84 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint

Web サービスエンドポイントの情報を指定します。

説明

Web サービスエンドポイントの情報を指定します。

出現パターン

0 回以上

3.9.85 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/port-component-name

EJB モジュール内のポートコンポーネントに一意の名称を指定します。

説明

EJB モジュール内のポートコンポーネントに一意の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.86 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/endpoint-address-uri

自動生成されたエンドポイントのアドレスを指定します。

説明

EJB エンドポイントの要求要素です。完全修飾エンドポイントアドレスにするために、Web サーバルートと組み合わせた相対パスを Web サービスエンドポイントに指定します。これらのすべての場合に、この値は固定のパターンとする必要があります（「*」は使用できません）。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0 回、または 1 回

3.9.87 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/login-config

EJB Web サービスエンドポイントに認証のための設定をします。

説明

EJB Web サービスエンドポイントに認証のための設定をします。サーブレット Web サービスエンドポイントの場合は、指定は不要です。サーブレットのセキュリティーの設定はweb.xml に含まれます。

出現パターン

0 回、または 1 回

3.9.88 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/login-config/auth-method

認証方式を指定します。

説明

認証方式を指定します。

親要素が `login-config` のため、Web サービスエンドポイントに認証方式を指定します。承認制約によって保護された Web リソースにアクセスするには、設定された認証方式を使用して認証される必要があります。

使用できる認証タイプは次のとおりです。

BASIC

サーバに組み込まれたログインダイアログボックスを使用します。通信プロトコルは HTTP (SSL はオプション) です。SSL を使用しないかぎりユーザークレデンシャルの暗号化はありません。SSL のような何らかの外部のセキュアシステムと一緒に使用しないかぎり、この認証タイプは安全なユーザー認証方法ではないと考えられています。

FORM

アプリケーションが独自のログインページとエラーページを提供します。通信プロトコルは HTTP (SSL はオプション) です。SSL を使用しないと、ユーザークレデンシャルは暗号化されません。

CLIENT-CERT

公開鍵を使用して、サーバがクライアントを認証します。通信プロトコルは HTTPS (HTTP over SSL) です。ユーザークレデンシャルの暗号化は SSL です。

DIGEST

ユーザー名とパスワードに基づいて、サーバがユーザーを認証します。BASIC 認証とは違い、パスワードは決してネットワークを介して送信されません。オプションで、SSL と HTTP Digest を一緒に使用できます。

指定できる値

型 : String

{BASIC | DIGEST | FORM | CLIENT-CERT}

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.89 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/login-config/realm

ユーザーが認証される、レルムを指定します。

説明

このアプリケーションと関連づけられているすべての認証要求を処理するために使用される、レルムの名称を指定します。この要素が指定されない、または設定されたレルムの名称と一致しない場合、デフォルトのレルムが使用されます。

サポートされるレルムは次のとおりです。

file

ファイルにユーザー情報を格納します。Application Server を最初にインストールしたときのデフォルトレルムです。

ldap

LDAP ディレクトリーにユーザー情報を格納します。

jdbc

データベースにユーザー情報を格納します。

JDBC レルムでは、サーバはデータベースからユーザークレデンシャルを取得します。Java EE Server は、データベース情報と設定ファイル内の有効な JDBC レルムオプションを使用します。ダイジェスト認証の場合、JDBC レルムは jdbcDigestRealm で JAAS コンテキストとして作成される必要があります。

certificate

Java EE Server セキュリティーコンテキストにユーザー識別を設定します。そして、暗号化されたクライアント証明書から得られるユーザー情報を、ユーザー識別に追加します。

指定できる値

型 : String

{file | ldap | jdbc | certificate}

デフォルト値

定義項目の省略

file

出現パターン

0回、または1回

3.9.90 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding

カスタム認証プロバイダーのバインディングを指定します。

説明

次の方法のどちらか、または両方を使用して、親要素のwebservice-endpoint に対して、カスタム認証プロバイダーのバインディングを指定します。

- 特定のプロバイダーにバインドする
- プロバイダーによって課せられるメッセージのセキュリティー要件を指定する

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-layer	認証が行われるメッセージレイヤーを指定します。	型：String SOAP	—	—	1回
provider-id	アプリケーション固有のメッセージのセキュリティー要件を満たすために使用される認証プロバイダーを指定します。この属性を指定しない場合、デフォルトのプロバイダーがメッセージレイヤーに定義されているとき、定義されたプロバイダーが使用されます。さらにデフォルトのプロバイダーも定義されていない場合、message-security-binding で定義された認証要件は実行されません。	型：String	—	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.91 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security

メッセージセキュリティー要件を指定します。

説明

メッセージセキュリティー要件を指定します。

先祖要素は `webservice-endpoint` のため、これらの要件はエンドポイントの要求と応答メッセージに関係します。

出現パターン

0 回以上

3.9.92 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message

メッセージセキュリティー要件が適用されるメソッドまたは操作を指定します。

説明

メッセージセキュリティー要件が適用されるメソッドまたは操作を指定します。

出現パターン

1 回以上

3.9.93 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method

メッセージセキュリティー要件が適用されるメソッドまたは操作を指定します。

説明

メッセージセキュリティー要件が適用されるメソッドまたは操作を指定します。

出現パターン

0回、または1回

3.9.94 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method/method-name

メソッド名を指定します。

説明

メソッド名を指定、またはアスタリスク (*) によってすべてのメソッドを指定します。メソッドがオーバーロードされる場合、すべてのメソッドを同じ名称で指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.9.95 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method/method-params

メソッドのパラメーターの完全修飾 Java 型名を指定します。

説明

メソッドのパラメーターの完全修飾 Java 型名を指定します。

出現パターン

0回、または1回

3.9.96 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/java-method/method-params/method-param

メソッドのパラメーターの完全修飾 Java 型名を指定します。

説明

メソッドのパラメーターの完全修飾 Java 型名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0 回以上

3.9.97 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/message/operation-name

Web サービスの操作のオペレーション名を指定します。

説明

Web サービスの操作のオペレーション名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.98 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/request-protection

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

説明

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	sender (ユーザー名とパスワード) または content (デジタル署名) どちらかの、要求される認証のタイプを指定します。	型: String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型: String {before-content after-content}	—	—	1回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.99 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/message-security-binding/message-security/response-protection

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

説明

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	sender (ユーザー名とパスワード) または content (デジタル署名) どちらかの、要求される認証のタイプを指定します。	型: String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型: String {before-content after-content}	—	—	1回

(凡例)

—: 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.100 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/transport-guarantee

クライアントとサーバ間の通信が NONE、INTEGRAL、または CONFIDENTIAL であることを指定します。

説明

クライアントとサーバ間の通信に NONE、INTEGRAL、または CONFIDENTIAL を指定します。

NONE

アプリケーションはトランスポート保障を必要としません。

INTEGRAL

アプリケーションは、通信データがクライアントとサーバ間で、伝送中に変更を加えられないような方法でデータ通信することを要求します。

CONFIDENTIAL

アプリケーションは、ほかのエンティティが送信内容を見ることができない方法で、データ通信することを要求します。

INTEGRAL、CONFIDENTIAL を指定した場合は、SSL の使用を要求されます。

指定できる値

型：String

{NONE | INTEGRAL | CONFIDENTIAL}

デフォルト値

定義項目の省略

NONE

出現パターン

0回、または1回

3.9.101 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/service-qname

参照されている WSDL service 要素を指定します。

説明

参照されている WSDL service 要素を指定します。

出現パターン

0回、または1回

3.9.102 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/service-qname/namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.103 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/service-qname/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.104 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/tie-class

ポートコンポーネントに、自動生成される実装クラスの名称を指定します。

説明

ポートコンポーネントに、自動生成される実装クラスの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.105 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/servlet-impl-class

自動生成されたサーブレット実装クラスの名称を指定します。

説明

自動生成されたサーブレット実装クラスの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.9.106 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/debugging-enabled

Web サービスエンドポイントでデバッグを有効にするかどうかを指定します。

説明

Web サービスエンドポイントでデバッグを有効にするかどうかを指定します。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

true

依存関係のある要素または属性

false

出現パターン

0回、または1回

3.9.107 /glassfish-ejb-jar/enterprise-beans/ejb/webservice-endpoint/property

webservice-endpoint 要素のプロパティを設定します。

説明

webservice-endpoint 要素のプロパティを設定します。

出現パターン

0回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	なし	—	1回
value	プロパティの値を指定します。	型：String	なし	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.9.108 /glassfish-ejb-jar/enterprise-beans/message-destination

論理的なメッセージの送信先の名称を指定します。

説明

論理的なメッセージの送信先の名称を指定します。

message-destination-name では、標準 DD で指定した message-destination-name を指定する必要があります。

この要素は、メッセージ送信先の参照として、message-destination-link を指定する場合に使用します。

出現パターン

0 回以上

3.9.109 /glassfish-ejb-jar/enterprise-beans/message-destination/ message-destination-name

標準 DD で指定した message-destination-name を指定します。

説明

標準 DD で指定した message-destination-name を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.110 /glassfish-ejb-jar/enterprise-beans/message-destination/ jndi-name

論理的なメッセージ送信先の JNDI 名を指定します。

説明

論理的なメッセージ送信先の JNDI 名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.111 /glassfish-ejb-jar/enterprise-beans/webservice-description

Web サービスの名称と、オプションの WSDL の配置先を指定します。

説明

Web サービスの名称と、オプションの WSDL の配置先を指定します。

出現パターン

0 回以上

3.9.112 /glassfish-ejb-jar/enterprise-beans/webservice-description/webservice-description-name

EJB モジュール内の Web サービスに一意の名称を指定します。

説明

EJB モジュール内の Web サービスに一意の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.9.113 /glassfish-ejb-jar/enterprise-beans/webservice-description/wsd-publish-location

デプロイメント時に Web サービスの WSDL がパブリッシュされるディレクトリーの URL を指定します。

説明

デプロイ時に Web サービスの WSDL が配置されるディレクトリーの URL を指定します。必要なファイルはこのディレクトリーにパブリッシュされ、その場所が、モジュール固有の WSDL ディレクトリー (META-INF/wsd) に相対となるよう維持されます。

例

webservices.xml ファイルの wsd-file 要素が次の参照を含む ejb.jar ファイルがあるとします。

```
META-INF/wsd/a/Foo.wsdl
```

glassfish-ejb.jar ファイルが次の要素を含むとします。

```
<wsd-publish-location>file:/home/user1/publish</wsd-publish-location>
```

最終的な WSDL は /home/user1/publish/a/Foo.wsdl. に保存されます。

指定できる値

型: String

デフォルト値

定義項目の省略

なし

出現パターン

0 回、または 1 回

3.9.114 /glassfish-ejb-jar/version-identifier

EJB モジュールのバージョン情報を定義します。

説明

EJB モジュールのバージョン情報を定義します。

この要素で指定するバージョン識別子は、EJB モジュールの suffix となり、コロン (:) で区切って名前に続けます。バージョン識別子はアルファベットまたは数字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 (_)、ダッシュ (-) およびピリオド (.) です。foo という EJB モジュールに対して、有効なバージョン識別子を指定した例を以下に示します。

foo:1

foo:BETA-2e

foo:3.8

foo:patch39875

指定できる値

型: String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.10 glassfish-resources.xml

リソースの設定をするための Java EE RI の DD です。

説明

リソースに関する、Java EE RI 固有の設定をするための DD です。

glassfish-resources.xml のルート要素はresources です。resources 要素は、「要素の階層」で示すように、子要素と属性を含みます（それらの子要素と属性の配下にも、さらに、子要素と属性を含むことがあります）。

JDBC コネクションプールを作成する場合、接続するデータベースに応じて次の属性を指定します。

- HiRDB に接続する場合に指定する属性

属性	値	備考	
/resources/jdbc-connection-pool 要素のdatasource-classname 属性	JP.co.Hitachi.soft.HiRDB.JDBC.Pr dbDataSource	※1	
	JP.co.Hitachi.soft.HiRDB.JDBC.Pr dbXADataSource	※2	
/resources/jdbc-connection-pool 要素のres-type 属性	javax.sql.DataSource	※1	
	javax.sql.XADataSource	※2	
/resources/jdbc-connection-pool 要素のconnection-validation-method 属性	custom-validation	※3	
/resources/jdbc-connection-pool 要素のvalidation-classname 属性	com.hitachi.software.javaee.jca. connectionvalidation.HiRDBConnec tionValidation	※3	
/resources/jdbc-connection-pool/property 要素	name 属性の値がdescription の value 属性	ポート番号	※1
		環境変数グループ識別子	※2
	name 属性の値がDBHostName のvalue 属性	HiRDBのホスト名	—
	name 属性の値がXALocalCommitMode のvalue 属性	true	※2、※4
	name 属性の値がXAOpenString の value 属性	環境変数グループ識別子+HiRDB.ini ファイルのパス	※2
	name 属性の値がuser のvalue 属性	ユーザー名	—
name 属性の値がpassword のvalue 属性	パスワード	—	

- Oracle に接続する場合に指定する属性

属性	値	備考	
/resources/jdbc-connection-pool 要素のdatasource-classname 属性	oracle.jdbc.pool.OracleDataSource	※1	
	oracle.jdbc.xa.client.OracleXADataSource	※2	
/resources/jdbc-connection-pool 要素のres-type 属性	javax.sql.DataSource	※1	
	javax.sql.XADataSource	※2	
/resources/jdbc-connection-pool 要素のconnection-validation-method 属性	custom-validation	※3	
/resources/jdbc-connection-pool 要素のvalidation-classname 属性	org.glassfish.api.jdbc.validation.OracleConnectionValidation	※3	
/resources/jdbc-connection-pool/property 要素	name 属性の値がdatabaseName のvalue 属性	Oracle SID	※5
	name 属性の値がserverName のvalue 属性	Oracleのホスト名称、またはIPアドレス	※5
	name 属性の値がportNumber のvalue 属性	ポート番号	※5
	name 属性の値がuser のvalue 属性	ユーザー名	—
	name 属性の値がpassword のvalue 属性	パスワード	—
	name 属性の値がdriverType のvalue 属性	thin	※5
	name 属性の値がurl のvalue 属性	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	※5

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

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。

スキーマ

```
<!DOCTYPE resources PUBLIC
"-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions
//EN" "http://glassfish.org/dtds/glassfish-resources_1_5.dtd">
```

格納先

DD ファイルは次の場所に格納されます。

- *application_ear_name*/META-INF ディレクトリー
application_ear_name : アプリケーションのアーカイブ名を表します。
- *war_name*/WEB-INF ディレクトリー
war_name : Web アプリケーションのアーカイブ名を表します。
- *ejb_jar_name*/META-INF ディレクトリー
ejb_jar_name : EJB アプリケーションのアーカイブ名を表します。
- *client_jar_name*/META-INF ディレクトリー
client_jar_name : クライアントアプリケーションのアーカイブ名を表します。
- *connector_rar_name*/META-INF ディレクトリー
connector_rar_name : リソースアダプターアーカイブのアーカイブ名を表します。

要素の階層

```
resources
. custom-resource
. . description
. . property
. external-jndi-resource
. . description
. . property
. . . description
. jdbc-resource
. . description
. . property
. . . description
. mail-resource
. . description
. . property
. . . description
. admin-object-resource
. . description
. . property
. . . description
```

- . connector-resource
 - . . description
 - . . property
 - . . . description
- . resource-adapter-config
 - . . property
 - . . . description
- . jdbc-connection-pool
 - . . description
 - . . property
 - . . . description
- . connector-connection-pool
 - . . description
 - . . security-map
 - . . . principal
 - name
 - user-group
 - backend-principal
 - . . property
 - . . . description
- . work-security-map
 - . . description
 - . . principal-map
 - . . group-map

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC
"-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions
//EN" "http://glassfish.org/dtds/glassfish-resources_1_5.dtd">
<resources>
  <jdbc-connection-pool>
    <property name="serverName" value="localhost"/>
    <property name="portNumber" value="1527"/>
    <property name="databaseName" value="forest"/>
    <property name="User" value="forest"/>
    <property name="Password" value="forest"/>
    <property name="URL" value="jdbc:oracle://localhost:1527/forest"/>
  </jdbc-connection-pool>
  <jdbc-resource enabled="true" jndi-name="jdbc/forest"
pool-name="oracle_net_forest_forestPool"/>
  <connector-connection-pool associate-with-thread="false"
connection-creation-retry-attempts="0" />
</resources>
```

3.11 glassfish-resources.xml に設定する要素の詳細

glassfish-resources.xml に設定する要素について説明しています。

3.11.1 /resources

エンタープライズ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定義します。

説明

エンタープライズアプリケーション、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、またはアプリケーションクライアントのモジュールの、アプリケーションスコープのリソースを定義します。これはルート要素です。glassfish-resources.xml ファイルには、リソース要素は 1 つだけ存在します。

出現パターン

1 回

3.11.2 /resources/custom-resource

カスタムリソースを定義します。

説明

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するカスタムリソースを定義します。このオブジェクトファクトリーは、javax.naming.spi.ObjectFactory インターフェースを実装します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
jndi-name	リソースの JNDI 名を指定します。	型 : String	—	—	1 回
res-type	リソースタイプの完全修飾名を指定します。	型 : String	—	—	1 回
factory-class	javax.naming.spi.ObjectFactory を実装した、ファクトリークラスの完全修飾名を指定します。	型 : String javax.naming.spi.ObjectFactory	—	—	1 回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
		インターフェースを実装するファクトリークラス			

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.3 /resources/custom-resource/description

custom-resource 要素の説明を指定します。

説明

custom-resource 要素の説明を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.4 /resources/custom-resource/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1 回
value	プロパティの値を指定します。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.5 /resources/external-jndi-resource

外部の JNDI リポジトリに存在するリソースを定義します。

説明

外部の JNDI リポジトリに存在するリソースを定義します。例えば、一般的な Java オブジェクトを LDAP サーバに格納できます。外部の JNDI ファクトリーは、`javax.naming.spi.InitialContextFactory` インターフェースを実装する必要があります。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
jndi-name	リソースの JNDI 名を指定します。	型：String	—	—	1 回
jndi-lookup-name	リソースの JNDI ルックアップ名を指定します。	型：String	—	—	1 回
res-type	リソースの完全修飾タイプを指定します。	型：String	—	—	1 回
factory-class	<code>javax.naming.spi.InitialContextFactory</code> を実装したファクトリークラスの完全修飾名を指定します。	型：String <code>javax.naming.spi.InitialContextFa</code>	—	—	1 回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
		ctory インターフェースを実装するファクトリークラス			
enabled	この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 {true false}	true	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.6 /resources/external-jndi-resource/description

external-jndi-resource 要素の説明を指定します。

説明

external-jndi-resource 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.7 /resources/external-jndi-resource/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1回
value	プロパティの値を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.8 /resources/external-jndi-resource/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.9 /resources/jdbc-resource

JDBC リソースを定義します。

説明

JDBC リソースを定義します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
jndi-name	リソースの JNDI 名を指定します。	型：String	—	—	1 回
pool-name	関連づけられている JDBC コネクションプールの名称を指定します。	型：String	—	—	1 回
enabled	この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 {true false}	true	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.10 /resources/jdbc-resource/description

jdbc-resource 要素の説明を指定します。

説明

jdbc-resource 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.11 /resources/jdbc-resource/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1回
value	プロパティの値を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.12 /resources/jdbc-resource/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.13 /resources/mail-resource

JavaMail リソースを定義します。

説明

JavaMail (javax.mail.Session) リソースを定義します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
jndi-name	リソースの JNDI 名を指定します。	型：String	—	—	1回
store-protocol	メッセージの受信プロトコルを指定します。	型：String {imap pop3 imaps pop3s}	imap	—	0回、または 1回
store-protocol-class	受信プロトコルのサービスプロバイダー実装クラスを指定します。	型：String {com.sun.mail.imap.IMAPStore com.sun.mail.pop3.POP3Store com.sun.mail.imap.IMAPSSLStore com.sun.mail.pop3.POP3SSLStore}	com.sun.mail.imap.IMAPStore	—	0回、または 1回
transport-protocol	メッセージの送信プロトコルを指定します。	型：String	smtp	—	0回、または 1回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
		{smtp smtps}			
transport-protocol-class	送信プロトコルのサービスプロバイダー実装クラスを指定します。	型：String {com.sun.mail.smtp.SMTPTransport com.sun.mail.smtp.SMTPSSLTransport}	com.sun.mail.smtp.SMTPTransport	—	0回、または1回
host	メールサーバのホスト名を指定します。	型：String	—	—	1回
user	メールサーバのユーザー名を指定します。	型：String	—	—	1回
from	メッセージの送信者を示すための電子メールアドレスを指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.14 /resources/mail-resource/description

mail-resource 要素の説明を指定します。

説明

mail-resource 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.15 /resources/mail-resource/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

mail-resource 要素に対してプロパティを設定して、JavaMail Session オブジェクトでこれらのプロパティをあとで取得できます。すべてのプロパティ名は接頭文字「mail-」で始まる必要があります。Java EE サーバは、プロパティ名の-（ハイフン）を.（ピリオド）に変更し、プロパティを MailConfiguration オブジェクト、および JavaMail Session オブジェクトへ保存します。プロパティの名称が「mail-」で始まらない場合、プロパティは無視されます。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1 回
value	プロパティの値を指定します。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.16 /resources/mail-resource/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.17 /resources/admin-object-resource

インバウンドのリソースアダプターの管理対象オブジェクトを定義します。

説明

インバウンドのリソースアダプターの管理対象オブジェクトを定義します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
jndi-name	リソースの JNDI 名を指定します。	型：String	—	—	1回
res-type	リソースの完全修飾タイプを指定します。	型：String	—	—	1回
class-name	同一res-typeの実装クラスが複数存在する場合に、対応する実装クラスの名称を指定します。	型：String	—	—	1回
res-adapter	インバウンドのリソースアダプターの名称を指定します。	型：String	—	—	1回
enabled	この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 {true false}	true	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.18 /resources/admin-object-resource/description

admin-object-resource 要素の説明を指定します。

説明

admin-object-resource 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.19 /resources/admin-object-resource/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1回
value	プロパティの値を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.20 /resources/admin-object-resource/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.21 /resources/connector-resource

コネクタリソースを定義します。

説明

リソースアダプターの特定のコネクション定義でのコネクションファクトリーを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
jndi-name	リソースの JNDI 名を指定します。	型：String	—	—	1回
pool-name	関連づけられてい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	—	—	1回
enabled	この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 {true false}	true	—	0回、または1回

(凡例)

－：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.22 /resources/connector-resource/description

connector-resource 要素の説明を指定します。

説明

connector-resource 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.23 /resources/connector-resource/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1回
value	プロパティの値を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.24 /resources/connector-resource/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.25 /resources/resource-adapter-config

リソースアダプターアーカイブ構成情報を定義します。

説明

リソースアダプターアーカイブ構成情報を定義します。子要素のproperty 要素を使用して、リソースアダプターアーカイブ構成情報を設定します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
resource-adapter-name	デプロイされた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を指定します。リソースアダプターがアプリケーションに組み込まれている場合、名称はアプリケーション名#リソースアダプター名になります。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.26 /resources/resource-adapter-config/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	なし	—	1 回
value	プロパティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.27 /resources/resource-adapter-config/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.28 /resources/jdbc-connection-pool

JDBC コネクションプールに関する設定を定義します。

説明

JDBC コネクションプールに関する設定を定義します。

接続するデータベースに応じて指定するオプションについては、『コマンドリファレンス』の『create-jdbc-connection-pool』を参照してください。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	コネクションプールの名称を指定します。 jdbc-resource 要素のpool-name 属性は、この名称を参照します。	型：String	—	—	1回
datasource- classname	JDBC ドライバーのデータソースのクラス名を指定します。このクラスは、	型：String	—	—	0回、または1回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	javax.sql.DataSource、 javax.sql.XADataSource、 javax.sql.ConnectionPoolDataSource、またはこれらを組み合わせて実装する必要があります。				
res-type	データソースクラスが実装するインターフェースを指定します。 java.sql.Driver を実装する JDBC ドライバー、および java.sql.Driver を使用するアプリケーションの組み合わせで構成される場合、この属性に java.sql.Driver を設定します。また、データソースクラスが2つ以上のインターフェースを実装する場合、または driver-classname が指定された場合は、あいまいさを避けるために、この属性を指定する必要があります。この属性が正当な値を持っていて、データソースクラスが示されたインターフェースを実装していない場合、エラーが発生します。	型：String {javax.sql.DataSource javax.sql.XADataSource javax.sql.ConnectionPoolDataSource java.sql.Driver}	—	—	0回、または1回
driver-classname	データベース製品によって提供される JDBC ドライバーの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する必要があります。	型：String java.sql.Driver インターフェースを実装するクラス (サブクラス) 名	—	—	0回、または1回
ping	コネクションプール生成時にデータベースコネクションの確認を行うことで、誤りのある属性値を特定し警告するかどうかを指定します。	型：Boolean {true false}	false	—	0回、または1回
steady-pool-size	コネクションプールの最小コネクション数を指定します。	型：Integer 0～ 2147483647	24	—	0回、または1回
max-pool-size	コネクションプールの最大コネクション数を指定します。	型：Integer 1～ 2147483647	24	—	0回、または1回
max-wait-time-in-millis	コネクション取得を待つ時間 (ミリ秒) を指定します。0の場合、リソースが使用できるようになるまで、またはエラーが発生するまで、呼び出し側は無期限にブロックされます。	型：Integer 0～ 2147483647	60000	—	0回、または1回

3. Java EE サーバで使用する Deployment Descriptor (DD)

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
pool-resize-quantity	<p>コネクションプールで保持しているアイドル状態のコネクションを削除する数を指定します。</p> <p>これは、idle-timeout-in-seconds の間隔で定期的に行われます。アイドル状態のコネクションとは、idle-timeout-in-seconds の間使用されていないコネクションを指します。プールサイズがsteady-pool-size に達すると、コネクションの削除は停止します。</p>	型：Integer 1～ 2147483647	2	—	0回、または1回
idle-timeout-in-seconds	<p>コネクションプール内でアイドル状態のコネクションを維持できる最長時間を秒単位で指定します。この時間が過ぎると、コネクションプールはこのコネクションを破棄します。</p> <p>この属性で指定する時間は、データベースサーバのタイムアウト値よりも短くすることで、アプリケーション内で使用できないコネクションが保持されるのを防止する必要があります。0を指定した場合、無効になります。</p>	型：Integer 0～ 2147483647	300	—	0回、または1回
transaction-isolation-level	<p>プールされたデータベースコネクションの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。</p> <p>read-uncommitted ダーティリード、ノンリピータブルリード、およびファントムリードが発生する可能性があります。</p> <p>read-committed ダーティリードは阻止されますが、ノンリピータブルリード、およびファントムリードは発生する可能性があります。</p> <p>repeatable-read ダーティリード、およびノンリピータブルリードは阻止されますが、ファントムリードが発生する可能性があります。</p> <p>serializable ダーティリード、ノンリピータブルリード、およびファントムリードが阻止されます。</p>	型：String {read-uncommitted read-committed repeatable-read serializable}	JDBC ドライバーでのデフォルトのアイソレーションレベル	—	0回、または1回
is-isolation-level-guaranteed	<p>transaction-isolation-level が明示的に設定された場合にだけ有効になります。true の場合、プールから取得したコネクションはすべて、任意のアイソレーションレベルを持つことが保障されます。これによって、性能に影響を受けるおそれのある JDBC ドライバーがいくつかあります。ホスト型アプリケーション</p>	型：Boolean {true false}	true	—	0回、または1回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	ンが、アイソレーションレベルが変更されたコネクションを戻さないと確信する場合にだけ、この属性を false に設定します。				
is-connection-validation-required	アプリケーションでコネクションを取得する前に、コネクション障害検知を実施するかどうかを指定します。障害検知でコネクションに問題がある場合、コネクションは破棄され、再生成されます。	型：Boolean {true false}	false	—	0 回、または 1 回
connection-validation-method	コネクション障害検知の方法を指定します。 auto-commit Connection#setAutoCommit (Connection#getAutoCommit()) を実行します。 meta-data Connection#getMetaData() を実行します。 table validation-table-name で指定したテーブルに対して、SELECT 文を実行します。 custom-validation validation-classname で指定したユーザー定義のクラスを使用します。	型：String {auto-commit meta-data table custom-validation}	table	—	0 回、または 1 回
validation-table-name	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。	型：String	—	—	0 回、または 1 回
validation-classname	カスタムの検証を提供するクラスを指定します。このパラメーターは、connection-validation-method に custom-validation が設定されている場合、必須です。Java EE Server にアクセスできるクラス名を指定する必要があります。	型：String org.glassfish.api.jdbc.ConnectionValidation インターフェースを実装するクラス (サブクラス) 名	—	—	0 回、または 1 回
non-transactional-connections	true に設定すると、JDBC コネクションプールに作成される非トランザクションコネクションを許可します。これらのコネクションは、トランザクションマネージャーとのトランザクションに自動的に参加しません。	型：Boolean {true false}	false	—	0 回、または 1 回
validate-atmost-once-	コネクション障害検知をする間隔を秒単位で指定します。コネクション障害検知の実施回数を最小限にします。値「0」は、Java EE	型：Integer	0	—	0 回、または 1 回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
period-in-seconds	Server が接続による接続障害検知の実施回数を最小に抑えようとしなことを示します。すなわち、値「0」はこの属性を無効にします。	0～ 214748364 7			
connection-leak-timeout-in-seconds	アプリケーションによる潜在的な接続リークを検出します。指定した期間内にアプリケーションがプールに戻さない接続は、潜在的なリークと判断され、スタックトレースがログに記録されます。値「0」は、リーク検出を無効にします。0以外の値は、リーク追跡を有効にします。この属性は、アプリケーションからの潜在的な接続リークを回避するためにconnection-leak-reclaimと共に使用します。	型：Integer 0～ 214748364 7	0	—	0回、または 1回
connection-leak-reclaim	true に設定すると、connection-leak-timeout-in-seconds の発生後に、接続を再要求します。	型：Boolean {true false}	false	—	0回、または 1回
connection-creation-retry-attempts	新しい接続の作成に失敗した場合に行われる試行の回数を指定します。	型：Integer 0～ 214748364 7	0	—	0回、または 1回
connection-creation-retry-interval-in-seconds	connection-creation-retry-attempts の値が0より大きい場合に、接続の作成を試行する間隔を秒単位で指定します。	型：Integer 0～ 214748364 7	10	—	0回、または 1回
statement-timeout-in-seconds	ステートメントのクエリタイムアウトプロパティを設定して、異常に長い時間実行されているクエリを終了させる機能を有効にします。-1 を指定するとこの機能は無効になります。アプリケーションが実行した JDBC クエリが異常に長い時間実行されている場合、タイムアウトをステートメントに明示的に設定しないかぎり、JDBC クエリは応答なし状態のままになるおそれがあります。この属性は、すべてのクエリが指定した時間内に完了しない場合、自動的にタイムアウトすることを保証します。ステートメントが作成されると、この属性の指定値に従い queryTimeout が設定されます。これは、基本的な JDBC ドライバーが Statement、PreparedStatement、CallableStatement、および ResultSet の queryTimeout をサポートする場合にだけ機能します。	型：Integer -1～ 214748364 7	-1	—	0回、または 1回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
lazy-connection-enlistment	true に設定すると、コネクションは、実際に利用されるまでトランザクションに参加しません。false に設定すると、コネクションが取得された時点で、トランザクションに参加します。	型：Boolean {true false}	false	—	0 回、または 1 回
lazy-connection-association	true に設定すると、物理コネクションは、使用されるまで論理コネクションに関連づけられません。false に設定すると、物理コネクションは、使用される前でも論理コネクションに関連づけられます。	型：Boolean {true false}	false	—	0 回、または 1 回
match-connections	プールから取得したコネクションが、特定の資格を持つかコネクションマッチングを行います。true に設定すると、コネクションマッチングを有効にします。コネクションが同じ種類の場合、false に設定できます。複数のユーザークレデンシャルを持つアプリケーションがコネクションプールを使用する場合、match-connections を true に設定する必要があります。 コネクションプールはリクエストの資格とプール内のコネクションを照合し、一致したコネクションを戻します。新規のリクエストが異なる資格を持つ場合、一致しない未使用のコネクションを自動的に破棄し、新しいコネクションを提供して新規のリクエストを満たします。 この属性は、アプリケーションが 1 つの資格だけしか使用せず、プールが同じ種類のコネクションを持っているとわかっている場合、true にする必要はありません。	型：Boolean {true false}	false	—	0 回、または 1 回
max-connection-usage-count	コネクションがプールに再利用される回数を指定します。指定した回数だけ再利用されると、コネクションはクローズされます。値「0」は、この機能を無効にします。コネクションを再利用できる最大回数を制限することで、アプリケーションがステートメントを閉じない場合にステートメントリークを回避できます。	型：Integer 0～ 2147483647	0	—	0 回、または 1 回
statement-cache-size	LRU キャッシュ方式を使用してキャッシュするステートメント数を指定します。値「0」はステートメントのキャッシュを無効にします。	型：Integer 0～ 2147483647	0	—	0 回、または 1 回
pooling	コネクションをプールするかどうかを指定します。	型：Boolean {true false}	true	—	0 回、または 1 回

3. Java EE サーバで使用する Deployment Descriptor (DD)

(凡例)

－：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.29 /resources/jdbc-connection-pool/description

jdbc-connection-pool 要素の説明を指定します。

説明

jdbc-connection-pool 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.30 /resources/jdbc-connection-pool/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

この要素には、指定のプロパティ以外に、JDBC ドライバーがサポートする DataSource のプロパティを指定できます。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。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
dynamic-reconfiguration-wait-timeout-in-seconds	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：Integer 0～ 2147483647	—	—	1回
user	データベースに接続するためのユーザー名を指定します。	型：String	—	—	1回
password	データベースに接続するためのパスワードを指定します。	型：String	—	—	1回
databaseName	コネクションプールのデータベースを指定します。	型：String	—	—	1回
serverName	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	—	—	1回
portNumber	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 1～65535	—	—	1回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
networkProtocol	通信プロトコルを指定します。	型：String	—	—	1回
roleName	初期 SQL ロール名を指定します。	型：String	—	—	1回
datasourceName	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	—	—	1回
description	説明テキストを指定します。	型：String	—	—	1回
url	コネクションプールの URL を指定します。このプロパティは標準プロパティではありませんが、共通で使用されます。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.31 /resources/jdbc-connection-pool/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.32 /resources/connector-connection-pool

コネクタコネクションプールを定義します。

説明

コネクタコネクションプールに関する設定を定義します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	コネクションプールの名称を指定します。 connector-resource 要素のpool-name 属性は、この名称を参照します。	型：String	—	—	1 回
resource-adapter-name	デプロイされた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の名称を指定します。デプロイ中に名称が指定されていない場合、.rar ファイルの名称が使用されます。リソースアダプターがアプリケーションに組み込まれている場合、名称はアプリケーション名#リソースアダプター名になります。	型：String	—	—	1 回
connection-definition-name	ra.xml ファイルのリソースアダプターの connection-definition 要素を特定する一意の名称を指定します。この名称は通常、connection-definition 要素の connectionfactory-interface になります。	型：String	—	—	1 回
steady-pool-size	コネクションプールの最小コネクション数を指定します。	型：Integer 0～ 2147483647	24	—	0 回、または 1 回
max-pool-size	コネクションプールの最大コネクション数を指定します。	型：Integer 1～ 2147483647	24	—	0 回、または 1 回
max-wait-time-in-millis	コネクション取得を待つ時間（ミリ秒）を指定します。0 の場合、リソースが使用できるようになるまで、またはエラーが発生するまで、呼び出し側は無期限にブロックされます。	型：Integer 0～ 2147483647	60000	—	0 回、または 1 回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
pool- resize- quantity	<p>コネクションプールで保持しているアイドル状態のコネクションを削除する数を指定します。</p> <p>これは、idle-timeout-in-seconds の間隔で定期的に行われます。アイドル状態のコネクションとは、idle-timeout-in-seconds の間使用されていないコネクションを指します。プールサイズがsteady-pool-size に達すると、コネクションの削除は停止します。</p>	型：Integer 1～ 214748364 7	2	—	0 回、または 1 回
idle- timeout-in- seconds	<p>コネクションプール内でアイドル状態のコネクションを維持できる最長時間を秒単位で指定します。この時間が過ぎると、コネクションプールはこのコネクションを破棄します。</p>	型：Integer 1～ 214748364 7	300	—	0 回、または 1 回
transaction- support	<p>このコネクションプールのトランザクションサポートを指定します。リソースアダプターで定義されたトランザクションサポートを下位互換で上書きし、リソースアダプター以下のトランザクションレベルをサポートするようにします。</p> <p>XATransaction XA トランザクションをサポートします。</p> <p>LocalTransaction ローカルトランザクションだけをサポートします。</p> <p>NoTransaction トランザクションをサポートしません。</p>	型：String {XATransaction LocalTransaction NoTransaction}	—	—	0 回、または 1 回
is- connection- validation- required	<p>アプリケーションでコネクションを取得する前に、コネクション障害検知を実施するかどうかを指定します。障害検知でコネクションに問題がある場合、コネクションは破棄され、再生成されます。</p> <p>一定間隔のコネクション障害検知については、指定値に依存しないで常に有効になります。</p>	型：Boolean {true false}	false	—	0 回、または 1 回
validate- atmost- once- period-in- seconds	<p>最大で 1 回、コネクション障害検知をする間隔を秒単位で指定します。検証コール数を最小限にします。値「0」を指定すると、検証コール数が無制限になります。</p>	型：Integer 0～ 214748364 7	0	—	1 回
connection- leak- timeout-in- seconds	<p>アプリケーションによる潜在的なコネクションリークを検出します。指定した期間内にアプリケーションがプールに戻さないコネクションは、潜在的なリークと判断され、スタックトレースがログに記録されます。値「0」は、</p>	型：Integer 0～ 214748364 7	0	—	1 回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	リーク検出を無効にします。0以外の値は、リーク追跡を有効にします。				
connection-leak-reclaim	trueに設定した場合、プールは、connection-leak-timeout-in-secondsの発生後に接続が再取得されます。	型: Boolean {true false}	false	—	1回
connection-creation-retry-attempts	新しい接続の作成の試行回数を指定します。	型: Integer 0~ 2147483647	0	—	1回
connection-creation-retry-interval-in-seconds	connection-creation-retry-attemptsの値が0より大きい場合に、接続の作成を試行する間隔を秒単位で指定します。	型: Integer 0~ 2147483647	10	—	1回
lazy-connection-enlistment	trueに設定すると、接続は、実際に利用されるまでトランザクションに参加しません。falseに設定すると、接続が取得された時点で、トランザクションに参加します。	型: Boolean {true false}	false	—	1回
lazy-connection-association	trueに設定した場合、物理接続は、使用されるまで論理接続に関連づけられません。falseに設定すると、物理接続は、使用される前でも論理接続に関連づけられます。	型: Boolean {true false}	false	—	1回
match-connections	trueに設定した場合、接続マッチングを有効にします。接続が同じ種類の場合、falseに設定できます。	型: Boolean {true false}	true	—	1回
max-connection-usage-count	接続がプールに再利用される回数を指定します。接続は、指定された回数利用されると、クローズされます。値「0」は、この機能を無効にします。	型: Integer 0~ 2147483647	0	—	1回
ping	接続プール生成時にデータベース接続の確認を行うことで、誤りのある属性値を特定し警告するかどうかを指定します。	型: Boolean {true false}	false	—	0回、または1回
pooling	falseに設定した場合、接続プーリングを無効にします。	型: Boolean {true false}	true	—	0回、または1回

(凡例)

—: 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.33 /resources/connector-connection-pool/description

connector-connection-pool 要素の説明を指定します。

説明

connector-connection-pool 要素の説明を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.34 /resources/connector-connection-pool/security-map

サーブレットまたは EJB の認証中に受信したプリンシパルを、EIS でのプリンシパルにマッピングします。

説明

サーブレットまたは EJB の認証中に受信したプリンシパルを、EIS でのプリンシパルにマッピングします。このマッピングは任意です。複数の Java EE Server のプリンシパルを同じ backend principal にマップできます。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	セキュリティーマッピングの名称を 指定します。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.35 /resources/connector-connection-pool/security-map/ principal

サーバインスタンスでのプリンシパルを定義します。

説明

サーバインスタンスでのプリンシパルを定義します。

出現パターン

1 回以上

3.11.36 /resources/connector-connection-pool/security-map/ principal/name

サーバインスタンスでのユーザー名を指定します。

説明

サーバインスタンスでのユーザー名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.11.37 /resources/connector-connection-pool/security-map/user-group

指定したプリンシパルが属するグループを指定します。

説明

指定したプリンシパルが属するグループを指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回以上

3.11.38 /resources/connector-connection-pool/security-map/backend-principal

EIS が必要とするユーザー名およびパスワードを指定します。

説明

エンタープライズ情報システム (EIS) が必要とするユーザー名およびパスワードを指定します。

出現パターン

1 回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
user-name	EIS が必要とするユーザー名を指定します。	型：String	—	—	1 回
password	パスワードがある場合、EIS が必要とするパスワードを指定します。	型：String	—	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.39 /resources/connector-connection-pool/property

プロパティの名称と値を指定します。

説明

プロパティの名称と値を指定します。

この要素を使用することで、ManagedConnectionFactory クラスが保持している設定値を JavaBeans 仕様に基づいて上書きできます。

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 要素とこの要素に同じプロパティを設定した場合、この要素の設定が優先されます。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1 回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
AddressList	MQ ブローカーのホストとポートの組み合わせのリストを指定します。 javax.jms.TopicConnectionFactory または javax.jms.QueueConnectionFactory 型の JMS リソース用です。	型：String	—	—	1 回
ClientId	TopicConnectionFactory クラスの createTopicConnection メソッドを使用して作成されるコネクションに関連づけられる JMS クライアントの識別子を指定します。 javax.jms.TopicConnectionFactory 型の JMS リソースについては、持続性のあるサブスクリプションの名称は一意で、クライアント識別子の範囲内だけで有効です。持続性のあるサブスクリプターを作成または再活性化するには、コネクションが有効なクライアント識別子を持っている必要があります。JMS の仕様書情報では、クライアント識別子は一意で、クライアント識別子は一度に 1 つのアクティブなコネクションでだけ使用できます。	型：String	—	—	1 回
UserName	MQ ブローカーに接続するためのユーザー名を指定します。 javax.jms.TopicConnectionFactory または javax.jms.QueueConnectionFactory 型の JMS リソース用です。	型：String	guest	—	1 回
Password	MQ ブローカーに接続するためのパスワードを指定します。 javax.jms.TopicConnectionFactory または javax.jms.QueueConnectionFactory 型の JMS リソース用です。	型：String	guest	—	1 回
ReconnectAttempts	imqAddressList 内の各アドレスへのコネクション（または再コネクション）の試行回数を指定します。指定された試行回数に達すると、クライアントランタイムがリスト内の次のアドレスに移動します。値「-1」は、再コネクションの試行回数が無制限であることを	型：Integer -1 ~ 2147483647	6	—	1 回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	示します（クライアントランタイムは、成功するまで、最初のアドレスに接続しようとしています）。				
ReconnectInterval	再コネクションを試行する間隔をミリ秒単位で指定します。この間隔は、imqAddressList内の各アドレス、およびリスト内の連続するアドレスへのコネクションの試行に適用されます。間隔が短過ぎる場合、ブローカーが回復するとき間を確保できません。間隔が長過ぎる場合は、再コネクション時に許容できない遅延が発生することがあります。	型：Integer 1～ 2147483647	30000	—	1回
ReconnectEnabled	true の場合、コネクションが失われたとき、クライアントランタイムがMQブローカー（またはimqAddressList内のアドレス）への再コネクションを試行します。	型：Boolean {true false}	false	—	1回
AddressListBehavior	コネクションの試行をimqAddressList属性のアドレスの順序（priority）、またはランダムな順序（random）で行うように指定します。 多くのクライアントが同じコネクションファクトリーを使用してコネクションを試行する場合、これらすべてのクライアントが同じアドレスに接続しないよう、ランダムな順序を使用してください。 指定できる値は次のとおりです。 PRIORITY parallel-broker の再コネクションを有効にするには、imqAddressListBehavior属性をPRIORITYに設定します。通常、このタイプの再コネクションで指定できるブローカーは2つ以下です。この場合、メッセージは1つのブローカーに送信され、すべてのクライアントが最初のブローカーから2番目のブローカーにフェイルオーバーします。 RANDOM クラスタ化したブローカーの再コネクションを有効にするには、imqAddressListBehavior属性をRANDOMに設定します。この場合、クライアントランタイムはリスト内のランダムなアドレスに接続しようとするため、クライアントコネクションがブローカークラスター全体に均等に分散されます。	型：String {priority random}	priority	—	1回
AddressListIterations	クライアントランタイムが、コネクションを確立（または再確立）するため、	型：Integer	-1	—	1回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	imqAddressList を巡回する回数を指定します。値「-1」は、試行回数が無制限であることを示します。	-1 ~ 214748364 7			

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.40 /resources/connector-connection-pool/property/description

property 要素の説明を指定します。

説明

property 要素の説明を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.41 /resources/work-security-map

ワークセキュリティーマップを定義します。

説明

Work インスタンスに関連づけられたプリンシパルと、Java EE Server のサーバインスタンスでのプリンシパルのマッピングを定義します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	ワークセキュリティーマップの一意の名前を指定します。	型：String	—	—	1 回
resource-adapter-name	デプロイされた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の名称を指定します。 リソースアダプターがアプリケーションに埋め込まれている場合には、 アプリケーション名/リソースアダプター名 となります。	型：String	—	—	1 回
enabled	実行時にこのリソースが有効になるかどうかを決定します。	型：Boolean {true false}	true	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.42 /resources/work-security-map/description

work-security-map 要素の説明を指定します。

説明

work-security-map 要素の説明を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.11.43 /resources/work-security-map/principal-map

EIS でのプリンシパルを、Java EE Server のサービンスタンスでのプリンシパルにマッピングします。

説明

EIS でのプリンシパルを、Java EE Server のサービンスタンスでのプリンシパルにマッピングします。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
eis-principal	EIS でのプリンシパルを指定します。	型：String	—	—	1回
mapped-principal	Java EE Server ドメインのサービンスタンスでのプリンシパルを指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.11.44 /resources/work-security-map/group-map

EIS グループを Java EE Server のサービンスタンスでのグループにマップします。

説明

EIS グループを Java EE Server のサービンスタンスでのグループにマップします。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
eis-group	EIS でのグループを指定します。	型：String	—	—	1 回
mapped-group	Java EE Server のサービンスタンスでのグループを指定します。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.12 glassfish-web.xml

Web アプリケーション (WAR) の設定をするための Java EE RI の DD です。

説明

Web アプリケーション (WAR) に関する、Java EE RI 固有の設定をするための DD です。

glassfish-web.xml のルート要素はglassfish-web-app です。glassfish-web-app 要素は、「要素の階層」で示すように、子要素と属性を含みます (それらの子要素と属性の配下にも、さらに、子要素と属性を含むことがあります)。

スキーマ

```
<!DOCTYPE glassfish-web-app PUBLIC
"-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN"
"http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
```

格納先

DD ファイルは次の場所に格納されます。

war_name/WEB-INF ディレクトリー

war_name : Web アプリケーションのアーカイブ名を表します。

要素の階層

```
glassfish-web-app
. context-root
. security-role-mapping
. . role-name
. . principal-name
. . group-name
. servlet
. . servlet-name
. . principal-name
. . webservice-endpoint
. . . port-component-name
. . . endpoint-address-uri
. . . login-config
. . . . auth-method
. . . message-security-binding
. . . . message-security
. . . . . message
. . . . . . java-method
. . . . . . . method-name
. . . . . . . method-params
. . . . . . . . method-param
. . . . . . . . operation-name
. . . . . request-protection
```

- response-protection
- . . . transport-guarantee
- . . . service-qname
- namespaceURI
- localpart
- . . . tie-class
- . . . servlet-impl-class
- . . . debugging-enabled
- . idempotent-url-pattern
- . session-config
- . . session-manager
- . . . manager-properties
- property
- description
- . . session-properties
- . . . property
- description
- . . cookie-properties
- . . . property
- description
- . ejb-ref
- . . ejb-ref-name
- . . jndi-name
- . resource-ref
- . . res-ref-name
- . . jndi-name
- . . default-resource-principal
- . . . name
- . . . password
- . resource-env-ref
- . . resource-env-ref-name
- . . jndi-name
- . service-ref
- . . service-ref-name
- . . port-info
- . . . service-endpoint-interface
- . . . wsdl-port
- namespaceURI
- localpart
- . . . stub-property
- . . . call-property
- . . . message-security-binding
- message-security
- message
- java-method
- method-name
- method-params
- method-param
- operation-name
- request-protection
- response-protection
- . . call-property
- . . wsdl-override
- . . service-impl-class
- . . service-qname
- . . . namespaceURI
- . . . localpart
- . message-destination-ref

```
. . message-destination-ref-name
. . jndi-name
. class-loader
. . property
. . . description
. jsp-config
. . property
. . . description
. property
. . description
. valve
. . description
. . property
. . . description
. message-destination
. . message-destination-name
. . jndi-name
. webservice-description
. . webservice-description-name
. . wsdl-publish-location
. version-identifier
```

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-web-app PUBLIC
"-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN"
"http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
<glassfish-web-app>
  <session-config>
    <session-manager/>
  </session-config>
  <resource-ref>
    <res-ref-name>mail/Session</res-ref-name>
    <jndi-name>mail/Session</jndi-name>
  </resource-ref>
  <jsp-config/>
</glassfish-web-app>
```

3.13 glassfish-web.xml に設定する要素の詳細

glassfish-web.xml に設定する要素について説明しています。

3.13.1 /glassfish-web-app

Web モジュールに、Java EE Server 固有の設定を定義します。

説明

Web モジュールに、Java EE Server 固有の設定を定義します。これはルート要素です。したがって、glassfish-web.xml ファイルには、glassfish-web-app 要素は 1 つだけ存在することになります。

出現パターン

1 回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
httpServlet -security- provider	<p>この属性は、使用アプリケーションのメッセージセキュリティプロバイダーをバインドするのに使用されます。</p> <p>属性の値を、メッセージセキュリティプロバイダーに割り当てられたプロバイダー名に設定して、アプリケーションに対応する DD ファイルのhttpServlet-security-provider 属性を定義します。</p> <p>例えば、メッセージセキュリティプロバイダーを作成するときに MySAM を使用すると、エントリーはhttpServlet-security-provider="MySAM"になります。</p>	型 : String	—	—	0 回、または 1 回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.2 /glassfish-web-app/context-root

アプリケーションまたは WAR ファイルとしてパッケージされた Web アプリケーションの Web コンテキストルートを含みます。

説明

WAR ファイルとしてパッケージされたアプリケーションまたは Web アプリケーションの Web コンテキストルートを指定します。application.xml ファイル内の対応する要素を上書きします。

ロードバランスをセットアップ中の場合、Web モジュールコンテキストルートはサーバインスタンス内で一意である必要があります。

指定できる値

型: String

デフォルト値

定義項目の省略

<拡張子.war を除いた WAR ファイル名>

出現パターン

0回、または1回

3.13.3 /glassfish-web-app/security-role-mapping

現在アクティブなレルムのユーザーまたはグループにロールをマップします。

説明

現在アクティブなレルムのユーザーまたはグループにロールをマップします。ロールマッピング要素は、環境固有のユーザーまたはグループにロールをマップします。

ユーザーにマップする場合、現在のレルムに存在し、現在の認証方式を使用してサーバにログインできる実在のユーザーである必要があります。

グループにマップする場合、レルムはグループをサポートし、グループは現在のレルムに存在する実在のグループである必要があります。有用性のため、そのレルムの1人以上のユーザーがそのグループに所属している必要があります。

出現パターン

0回以上

3.13.4 /glassfish-web-app/security-role-mapping/role-name

対応する Java EE RI の DD ファイルの security-role 要素の role-name を指定します。

説明

対応する Java EE RI の DD ファイルの security-role 要素の role-name を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.5 /glassfish-web-app/security-role-mapping/principal-name

現在のレルムのプリンシパル (ユーザー) 名を指定します。

説明

現在のレルムのプリンシパル (ユーザー) 名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

group-name が ない場合は 1 回以上、その他の場合は 0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
class-name	このプリンシパルに対応する、プリンシパル実装クラスを指定します。	型：String java.security.Principal インターフェースを実装するクラス (サブクラス) 名	com.sun.enterprise.deployment.PrincipalImpl	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.6 /glassfish-web-app/security-role-mapping/group-name

現在のレルムにグループ名を指定します。

説明

現在のレルムにグループ名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

principal-name がない場合は 1 回以上、その他の場合は 0 回以上

3.13.7 /glassfish-web-app/servlet

web.xml に定義された run-as ロールに使用されるサーブレットのプリンシパル名を指定します。

説明

web.xml に定義された run-as ロールに使用されるサーブレットのプリンシパル名を指定します。

出現パターン

0 回以上

3.13.8 /glassfish-web-app/servlet/servlet-name

web.xml の servlet-name に適合するサーブレット名を指定します。

説明

web.xml の servlet-name に適合するサーブレット名を指定します。この名称は web.xml ファイルに存在している必要があります。

指定できる値

型 : String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.13.9 /glassfish-web-app/servlet/principal-name

現在のレルムのプリンシパル (ユーザー) 名を指定します。

説明

現在のレルムのプリンシパル (ユーザー) 名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

group-name がない場合は 1 回以上、その他の場合は 0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
class-name	このプリンシパルに対応する、プリンシパル実装クラスを指定します。	型：String java.security.Principal インターフェースを実装するクラス (サブクラス)名	com.sun.enterprise.deployment.PrincipalImpl	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.10 /glassfish-web-app/servlet/webservice-endpoint

Web サービスエンドポイントの情報を指定します。

説明

Web サービスエンドポイントについての情報を指定します。

出現パターン

0 回以上

3.13.11 /glassfish-web-app/servlet/webservice-endpoint/port-component-name

Web モジュール内にあるポートコンポーネントの一意の名称を指定します。

説明

Web モジュール内にあるポートコンポーネントの一意の名称を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.12 /glassfish-web-app/servlet/webservice-endpoint/endpoint-address-uri

自動生成されたエンドポイントのアドレスを指定します。

説明

Web サーバのルートを組み合わせた相対パスを指定して、Web サービスのエンドポイントの完全修飾エンドポイントアドレスを生成します。

これは、サーブレットのエンドポイントの任意の要素です。

サーブレットのエンドポイントでは、この値は、Web アプリケーションのコンテキストルートの相対値になります。コンテキストルート部分は、同じ Web サーバにデプロイされている Web アプリケーションのコンテキストルートと競合しないものとします。

この値は常に、固定パターンでなければなりません (*は指定できません)。

Web サービスのエンドポイントが、エンドポイントを 1 つだけ実装しているサーブレットで、URL パターンを 1 つしか持たない場合、Web コンテナは値を web.xml ファイルから取得するため、この値を設定する必要はありません。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.13 /glassfish-web-app/servlet/webservice-endpoint/login-config

EJB Web サービスエンドポイントの認証のための設定をします。

説明

EJB Web サービスエンドポイントの認証のための設定をします。サーブレット Web サービスエンドポイントには必要ありません。サーブレットのセキュリティー設定は、web.xml ファイルに含まれます。

出現パターン

0回、または1回

3.13.14 /glassfish-web-app/servlet/webservice-endpoint/login-config/auth-method

認証方式を指定します。

説明

Web サービスのエンドポイントの認証機構を指定します。承認制約によって保護された Web リソースにアクセスするには、設定された認証方式を使用して認証される必要があります。

使用できる認証タイプは次のとおりです。

BASIC

サーバに組み込まれたログインダイアログボックスを使用します。通信プロトコルは HTTP (SSL はオプション) です。SSL を使用しないかぎりユーザークレデンシャルは暗号化されません。SSL のよう

な何らかの外部のセキュアシステムと一緒に使用しないかぎり、ユーザーの認証方式として安全とはいえません。

FORM

アプリケーションが独自のログインページとエラーページを提供します。通信プロトコルは HTTP (SSL はオプション) です。SSL を使用しないと、ユーザークレデンシャルは暗号化されません。

CLIENT-CERT

サーバは、公開鍵証明書を使用して、クライアントを認証します。通信プロトコルは HTTPS (HTTP over SSL) です。ユーザークレデンシャルの暗号化は SSL です。

DIGEST

ユーザー名とパスワードに基づいて、サーバがユーザーを認証します。BASIC 認証とは違い、パスワードは決してネットワークを介して送信されません。オプションで、SSL と HTTP Digest を一緒に使用できます。

指定できる値

型: String

{BASIC | DIGEST | FORM | CLIENT-CERT}

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.15 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding

カスタム認証プロバイダーのバインディングを指定します。

説明

次の方法のどちらか、または両方を使用して、親要素の `webservice-endpoint` に対して、カスタム認証プロバイダーのバインディングを指定します。

- 特定のプロバイダーにバインドする。
- プロバイダーによって課せられるメッセージのセキュリティー要件を指定する。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-layer	認証が行われるメッセージレイヤーを指定します。	型：String SOAP	—	—	1回
provider-id	アプリケーション固有のメッセージのセキュリティ要件を満たすために使用される認証プロバイダーを指定します。この属性を指定しない場合、デフォルトのプロバイダーがメッセージレイヤーに定義されているとき、定義されたプロバイダーが使用されます。さらにデフォルトのプロバイダーも定義されていない場合、message-security-binding で定義された認証要件は実行されません。	型：String	—	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.16 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security

メッセージセキュリティ要件を指定します。

説明

メッセージセキュリティ要件を指定します。親の親要素がwebservice-endpointであるため、これらのメッセージセキュリティ要件は、このエンドポイントの要求および応答電文に関連しています。

出現パターン

0回以上

3.13.17 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/message

メッセージセキュリティー要件を適用するメソッドまたは操作を指定します。

説明

メッセージセキュリティー要件を適用するメソッドまたは操作を指定します。

出現パターン

1 回以上

3.13.18 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/message/java-method

メッセージセキュリティー要件を適用するメソッドまたは操作を指定します。

説明

メッセージセキュリティー要件を適用するメソッドまたは操作を指定します。

出現パターン

0 回、または 1 回

3.13.19 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/message/java-method/method-name

メソッド名を指定します。

説明

メソッド名を指定、またはアスタリスク (*) によってすべてのメソッドを指定します。メソッドがオーバーロードされる場合、すべてのメソッドを同じ名称で指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.20 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/message/java-method/method-params

メソッドのパラメーターの完全修飾 Java 型名を指定します。

説明

メソッドのパラメーターの完全修飾 Java 型名を指定します。

出現パターン

0 回、または 1 回

3.13.21 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/message/java-method/method-params/method-param

1 つのメソッドのパラメーターの完全修飾 Java 型名を指定します。

説明

1 つのメソッドのパラメーターの完全修飾 Java 型名を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0 回以上

3.13.22 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/message/operation-name

Web サービスの操作のオペレーション名を指定します。

説明

Web サービスの操作のオペレーション名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

0 回、または 1 回

3.13.23 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/request-protection

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

説明

アプリケーションのリクエスト処理の認証ポリシー要件を定義します。

出現パターン

0 回、または 1 回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	必須の認証タイプを指定します。これは、sender（ユーザー名およびパスワード）認証タイプ、またはcontent（デジタル署名）認証タイプのどちらかになります。	型：String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型：String {before-content after-content}	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.24 /glassfish-web-app/servlet/webservice-endpoint/message-security-binding/message-security/response-protection

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

説明

アプリケーションのレスポンス処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	必須の認証タイプを指定します。これは、sender（ユーザー名およびパスワード）認証タイプ、またはcontent（デジタル署名）認証タイプのどちらかになります。	型：String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型：String {before-content	—	—	1回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
		after-content}			

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.25 /glassfish-web-app/servlet/webservice-endpoint/transport-guarantee

クライアントとサーバ間の通信がNONE、INTEGRAL、またはCONFIDENTIALであることを指定します。

説明

クライアントとサーバ間の通信にNONE、INTEGRAL、またはCONFIDENTIALを指定します。

NONE

アプリケーションはトランスポート保障を必要としません。

INTEGRAL

アプリケーションは、通信データがクライアントとサーバ間で、伝送中に変更を加えられないような方法でデータ通信をすることを要求します。

CONFIDENTIAL

アプリケーションは、ほかのエンティティが送信内容を見ることができない方法で、データ通信することを要求します。

INTEGRAL、CONFIDENTIALを指定した場合は、SSLの使用を要求されます。

指定できる値

型 : String

{NONE | INTEGRAL | CONFIDENTIAL}

デフォルト値

定義項目の省略

NONE

出現パターン

0回、または1回

3.13.26 /glassfish-web-app/servlet/webservice-endpoint/service-qname

参照する WSDL サービス要素を指定します。

説明

参照する WSDL サービス要素を指定します。

出現パターン

0回、または1回

3.13.27 /glassfish-web-app/servlet/webservice-endpoint/service-qname/namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.13.28 /glassfish-web-app/servlet/webservice-endpoint/service-qname/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカル部分を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.13.29 /glassfish-web-app/servlet/webservice-endpoint/tie-class

ポートコンポーネント用の tie 実装クラスの自動生成された名称を指定します。

説明

ポートコンポーネント用の tie 実装クラスの自動生成された名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

0 回、または 1 回

3.13.30 /glassfish-web-app/servlet/webservice-endpoint/servlet-impl-class

生成されたサーブレット実装クラスの自動生成された名称を指定します。

説明

生成されたサーブレット実装クラスの自動生成された名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

0回、または1回

3.13.31 /glassfish-web-app/servlet/webservice-endpoint/debugging-enabled

デバッグ用のサーブレットを Web サービスエンドポイントに対して有効にするかどうかを指定します。

説明

デバッグ用のサーブレットを Web サービスエンドポイントに対して有効にするかどうかを指定します。

指定できる値

型：String

{true | false}

デフォルト値

定義項目の省略
true

出現パターン

0回、または1回

3.13.32 /glassfish-web-app/idempotent-url-pattern

べき等なリクエストの URL パターンを指定します。

説明

べき等なリクエストの URL パターンを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
url-pattern	URL パターンは、サーブレットの仕様書情報で指定されているマッピングに従うものとしてします。 指定できるurl-patternの値は、サーブレットの仕様書情報のとおりです。	型：String 指定できるパターンは次のとおりです。 1.「/」で始まり、「/*」で終わる文字列は、パスのマッピングに使用されます。 2.「*。」で始まる文字列は、拡張子のマッピングとして使用されます。 3.「/」だけを含む文字列は、アプリケーションの「デ	—	—	1回

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
		フォルト」のサーブレットを示します。この場合、サーブレットのパスは、リクエストの URI からコンテンツパスを除いたものになります。パス情報は null です。 4. その他の文字列はすべて、完全に一致にだけ使用されます。			
no-of-retries	ロードバランサーのべき等なリクエストの試行回数を指定します。値「-1」は、試行回数が無制限であることを示します。	型：Integer -1～2147483647	-1	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.33 /glassfish-web-app/session-config

セッションの設定をします。

説明

セッションマネージャー、セッション Cookie、およびその他のセッションの設定をします。個々の Web モジュールの Web コンテナ設定を上書きします。

出現パターン

0回、または1回

3.13.34 /glassfish-web-app/session-config/session-manager

セッションの設定をします。

説明

セッションの設定をします。

出現パターン

0回、または1回

3.13.35 /glassfish-web-app/session-config/session-manager/ manager-properties

セッションマネージャーのプロパティを指定します。

説明

セッションマネージャーのプロパティを指定します。

出現パターン

0回、または1回

3.13.36 /glassfish-web-app/session-config/session-manager/ manager-properties/property

名称と値を持つプロパティを指定します。

説明

名前と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値※	依存関係	出現パターン
reapIntervalSeconds	期限切れのセッションをチェックする周期を秒単位で指定します。また、maxSessionsの値を超えた場合には、この周期でセッションが非活性化されます。	型： Integer 1～ 2147483647	60	—	0回、または1回
maxSessions	キャッシュ内に格納できるセッションの最大数を指定します。キャッシュ内に格納できるセッション数を制限しない場合は、-1に設定します。最大数を超えて上限値を超えて新しいセッションを作成しようとすると、IllegalStateExceptionがスローされます。	型： Integer -1～ 2147483647	-1	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.37 /glassfish-web-app/session-config/session-manager/manager-properties/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.38 /glassfish-web-app/session-config/session-properties

セッションのプロパティを指定します。

説明

セッションのプロパティを指定します。

出現パターン

0回、または1回

3.13.39 /glassfish-web-app/session-config/session-properties/ property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
timeoutSeconds	Web モジュールで作成されたすべてのセッションについて、非アクティブな状態が継続する最長時間のデフォルト値を秒単位で指定します。0 またはそれ以下の値に設定した場合、この Web モジュールのセッションは期限切れになることはありません。web.xml ファイルに session-timeout 要素が指定されている場合、session-timeout 値は timeoutSeconds 値をオーバーライドします。session-timeout も timeoutSeconds も指定されていない場合、timeoutSeconds のデフォルト値が使用されます。 web.xml の session-timeout 要素は分単位で指定しますが、glassfish-web.xml の timeoutSeconds は秒単位で指定します。	型：Integer -2147483648～ 2147483647	asadmin ユーティリティコマンドの set サブコマンドで使用する、configs.config.Java EE サーバの構成名.web-container.session-config.session-properties.timeout-in-seconds パラメーターで設定されている値	—	0回、または1回
enableCookies	true に設定すると、セッションの追跡に Cookie を使用します。	型：Boolean {true false}	true	—	0回、または1回
enableURLRewriting	ブラウザが Cookie を受け入れない場合に、URL の書き換えを通したセッションの追跡を提供します。サーブレットや JSP で	型：Boolean {true false}	true	—	0回、または1回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	encodeURL または encodeRedirectURL の呼び出しを使用する必要があります。				

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.40 /glassfish-web-app/session-config/session-properties/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略
なし

出現パターン

0回、または1回

3.13.41 /glassfish-web-app/session-config/cookie-properties

セッション Cookie のプロパティを指定します。

説明

セッション Cookie のプロパティを指定します。web.xml ファイルに Cookie の設定が宣言的に定義されている場合、ここで定義した Cookie のプロパティが優先されます。

javax.servlet.SessionCookieConfig メソッドを使用して、Cookie の設定がプログラムで定義されている場合、その Cookie の設定がここで定義した Cookie のプロパティより優先されます。

出現パターン

0 回、または 1 回

3.13.42 /glassfish-web-app/session-config/cookie-properties/ property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1 回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
cookieName	Cookie の名称を指定します。	型：String	JSESSIONID	—	0 回、または 1 回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	<p>セッション ID を示す HTTP Cookie 名を変更する場合、Cookie 名に <code>csfcfc</code> は指定できません。</p> <p>HTTP Cookie 名に使用できる文字には次の条件があります。セッションの不正な動作の原因になることがあるため、条件に合った文字を使用してください。</p> <ul style="list-style-type: none"> • ASCII 文字を使用してください。ただし、次の文字は使用できません。 #、(、)、<、>、@、,、;、:、¥、/、”、[、]、?、=、{、} • 空白、または制御文字は使用できません。 • 文字列の先頭に \$ は使用できません。 <p>Web アプリケーションで HTTP Cookie を使用する場合、次の名称は指定できません。また、大文字と小文字は区別されません。</p> <ul style="list-style-type: none"> • HTTP セッションのセッション ID で使用する HTTP Cookie の名称。デフォルトは <code>JSESSIONID</code>。 • Java EE サーバ内部で使用する HTTP Cookie の名称 <code>csfcfc</code>。 				
<code>cookiePath</code>	<p>リクエストのパス名に、パス名 <code>@name=cookiePath</code> が含まれている場合、Cookie を送信します。/ (スラッシュ) に設定すると、ブラウザは Java EE Server が提供するすべての URL に対して Cookie を送信します。パスをより狭い範囲のマッピングに設定し、ブラウザが Cookie を送信するリクエスト URL を制限できます。</p>	型: String	コンテキストパス <i>Web モジュールのインストール先</i>	—	0 回、または 1 回
<code>cookieMaxAgeSeconds</code>	<p>Cookie の有効期限を秒単位で設定します。有効期限が設定されていない場合、Cookie の有効期限は切れません。</p>	型: Integer 1 ~ 2147483647	—	—	0 回、または 1 回
<code>cookieDomain</code>	<p>Cookie が有効となるドメインを指定します。</p>	型: String	—	—	0 回、または 1 回
<code>cookieComment</code>	<p><code>cookie</code> ファイル内でセッション追跡 Cookie を識別するコメントを指定します。</p>	型: String	—	—	0 回、または 1 回
<code>cookieSecure</code>	<p>その Web アプリケーションに関連するすべての <code>JSESSIONID</code> Cookie に Secure 属性を付与するかどうかを設定します。指定できる値は次のとおりです。</p> <p><code>true</code> Cookie に Secure 属性を付与します。</p>	型: String {true false dynamic}	dynamic	—	0 回、または 1 回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	<p>false Cookie に Secure 属性を付与しません。</p> <p>dynamic Java EE サーバが受けるリクエストが HTTP の場合は、Cookie に Secure 属性を付与しません。HTTPS の場合は、Cookie に Secure 属性を付与します。ただし、Java EE サーバが受け付けるリクエストは HTTP だけのため、常に false と同様の動作となります。</p>				
cookieHttpOnly	Cookie に HTTP だけがアクセスできるかどうかを指定します。	型 : Boolean {true false}	—	—	0 回、または 1 回

(凡例)

— : 該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.43 /glassfish-web-app/session-config/cookie-properties/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0 回、または 1 回

3.13.44 /glassfish-web-app/ejb-ref

対応する Java EE RI の DD ファイルの `ejb-ref` に、JNDI 完全修飾名を割り当てます。

説明

対応する Java EE RI の DD ファイルの `ejb-ref` エントリー内の `ejb-ref-name` を、リソースの JNDI 完全修飾名に割り当てます。 `ejb-ref` 要素は、EJB のホームへの参照を宣言するのに使用されます。

出現パターン

0 回以上

3.13.45 /glassfish-web-app/ejb-ref/ejb-ref-name

対応する Java EE RI の DD ファイルの `ejb-ref` エントリー内の `ejb-ref-name` を指定します。

説明

対応する Java EE RI の DD ファイルの `ejb-ref` エントリー内の `ejb-ref-name` を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.13.46 /glassfish-web-app/ejb-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。 Entity Bean および Session Bean に対して、この値は EJBHome オブジェクトのグローバル JNDI 名を指定します。 Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.47 /glassfish-web-app/resource-ref

対応する Java EE XML ファイルの `resource-ref` に、JNDI 完全修飾名を割り当てます。

説明

対応する Java EE XML ファイルの `resource-ref` に、JNDI 完全修飾名を割り当てます。

出現パターン

0 回以上

3.13.48 /glassfish-web-app/resource-ref/res-ref-name

リソースマネージャー接続ファクトリー参照の名称を指定します。

説明

リソースマネージャー接続ファクトリー参照の名称を指定します。

対応する Java EE RI の DD ファイルの `resource-ref` エントリー内の `res-ref-name` を指定してください。

リソースマネージャー接続ファクトリー参照の名称は、Enterprise Bean 内で一意の名称である必要があります。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.49 /glassfish-web-app/resource-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の両方に対して、この値は EJBHome オブジェクトのグローバル JNDI 名を指定します。Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.50 /glassfish-web-app/resource-ref/default-resource-principal

デフォルトのリソースプリンシパル（ユーザー）を指定します。

説明

デフォルトのリソースプリンシパル（ユーザー）を指定します。

出現パターン

0回、または1回

3.13.51 /glassfish-web-app/resource-ref/default-resource-principal/ name

デフォルトのプリンシパル名を指定します。

説明

リソースマネージャーにサインオンするのに使用されるデフォルトのリソースプリンシパル名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.13.52 /glassfish-web-app/resource-ref/default-resource-principal/ password

デフォルトのリソースプリンシパルのパスワードを指定します。

説明

デフォルトのリソースプリンシパルのパスワードを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.53 /glassfish-web-app/resource-env-ref

対応する Java EE XML ファイルの `resource-env-ref` に、JNDI 完全修飾名を割り当てます。

説明

対応する Java EE XML ファイルの `resource-env-ref` に、JNDI 完全修飾名を割り当てます。

出現パターン

0 回以上

3.13.54 /glassfish-web-app/resource-env-ref/resource-env-ref-name

対応する Java EE RI の DD ファイルの `resource-env-ref` エントリー内の `res-ref-name` を指定します。

説明

対応する Java EE RI の DD ファイルの `resource-env-ref` エントリー内の `res-ref-name` を指定します。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.55 /glassfish-web-app/resource-env-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の両方に対して、この値は EJBHome オブジェクトのグローバル JNDI 名を指定します。Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.56 /glassfish-web-app/service-ref

Web サービス参照のランタイム設定を指定します。

説明

Web サービス参照のランタイム設定を指定します。ランタイム情報は、次の条件の場合にだけ必要です。

- コンテナ管理対象ポートを解決するために使用するポートを定義する場合
- スタブオブジェクトに対してデフォルトの Stub プロパティ設定、またはデフォルトの Call プロパティ設定を定義する場合
- 標準 DD の service-ref と関連づけられている WSDL ドキュメントの代わりに使用される最終的な WSDL ドキュメントの URL を定義する場合

出現パターン

0 回以上

3.13.57 /glassfish-web-app/service-ref/service-ref-name

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.58 /glassfish-web-app/service-ref/port-info

Web サービス参照内のポートの情報を指定します。

説明

Web サービス参照内のポートの情報を指定します。

出現パターン

0 回以上

3.13.59 /glassfish-web-app/service-ref/port-info/service-endpoint-interface

java:comp/env で始まる Web サービス参照名を指定します。

説明

java:comp/env で始まる Web サービス参照名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.60 /glassfish-web-app/service-ref/port-info/wsdl-port

WSDL ポートを指定します。

説明

WSDL ポートを指定します。

出現パターン

0回、または1回

3.13.61 /glassfish-web-app/service-ref/port-info/wsdl-port/ namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.13.62 /glassfish-web-app/service-ref/port-info/wsdl-port/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.13.63 /glassfish-web-app/service-ref/port-info/stub-property

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC Stub 実装がサポートする任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回
value	関連するネームエンティティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.64 /glassfish-web-app/service-ref/port-info/call-property

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。プロパティ名は、JAX-RPC Call 実装でサポートされる任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	—	1 回
value	エンティティの値を指定します。	型：String	なし	—	1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.65 /glassfish-web-app/service-ref/port-info/message-security-binding

カスタム認証プロバイダーのバインディングを指定します。

説明

次の方法のどちらか、または両方を使用して、親要素のport-info に対して、カスタム認証プロバイダーのバインディングを指定します。

- 特定のプロバイダーにバインドする。
- プロバイダーによって課せられるメッセージのセキュリティー要件を指定する。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値※	依存関係	出現パターン
auth-layer	認証が行われるメッセージレイヤーを指定します。	型：String SOAP	なし	—	1回
provider-id	アプリケーション固有のメッセージのセキュリティー要件を満たすために使用される認証プロバイダーを指定します。この属性を指定しない場合、デフォルトのプロバイダーがメッセージレイヤーに定義されているとき、定義されたプロバイダーが使用されます。さらにデフォルトのプロバイダーも定義されていない場合、message-security-binding で定義された認証要件は実行されません。	型：String	なし	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.66 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security

メッセージのセキュリティー要件を指定します。

説明

メッセージのセキュリティー要件を指定します。

- 孫要素がport-info なので、これらの要件は参照されるサービスのポートに関係します。

出現パターン

0 回以上

3.13.67 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message

メッセージのセキュリティー要件を適用するメソッドまたは操作を指定します。

説明

メッセージのセキュリティー要件を適用するメソッドまたは操作を指定します。

出現パターン

1 回以上

3.13.68 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method

メッセージのセキュリティー要件を適用するメソッドまたは操作を指定します。

説明

メッセージのセキュリティー要件を適用するメソッドまたは操作を指定します。

出現パターン

0 回、または 1 回

3.13.69 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method/method-name

メソッド名を指定します。

説明

メソッド名を指定、またはアスタリスク (*) によってすべてのメソッドを指定します。メソッドがオーバーロードされる場合、すべてのメソッドを同じ名称で指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.70 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method/method-params

method パラメーターの完全修飾 Java 型名を指定します。

説明

method パラメーターの完全修飾 Java 型名を指定します。

出現パターン

0 回、または 1 回

3.13.71 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/java-method/method-params/method-param

method パラメーターの完全修飾 Java 型名を指定します。

説明

method パラメーターの完全修飾 Java 型名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回以上

3.13.72 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/message/operation-name

Web サービスの操作のオペレーション名を指定します。

説明

Web サービスの操作のオペレーション名を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.73 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/request-protection

認証リクエスト処理の認証ポリシー要件を定義します。

説明

認証リクエスト処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	必要な認証のタイプを、sender（ユーザー名とパスワード）またはcontent（デジタル署名）で指定します。	型：String {content sender}	—	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型：String {before-content after-content}	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.74 /glassfish-web-app/service-ref/port-info/message-security-binding/message-security/response-protection

認証レスポンス処理の認証ポリシー要件を定義します。

説明

認証レスポンス処理の認証ポリシー要件を定義します。

出現パターン

0回、または1回

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
auth-source	必要な認証のタイプを、sender（ユーザー名とパスワード）またはcontent（デジタル署名）で指定します。	型：String {content sender}	なし	—	1回
auth-recipient	受信者認証が、コンテンツ認証の前後どちらで起こるのかを指定します。	型：String {before-content after-content}	なし	—	1回

(凡例)

－：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.75 /glassfish-web-app/service-ref/call-property

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。

説明

Web サービスクライアントに返却される前に javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値を指定します。JAX-RPC Call 実装でサポートされる任意のプロパティを選択できます。

出現パターン

0 回以上

指定できる子要素

要素名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	エンティティの名称を指定します。	型：String	なし	－	1 回
value	エンティティの値を指定します。	型：String	なし	－	1 回

(凡例)

－：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.76 /glassfish-web-app/service-ref/wsdl-override

最終的な WSDL ドキュメントを示す有効な URL を指定します。

説明

指定しない場合、標準 DD の service-ref と関連づけられている WSDL ドキュメントが使用されます。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.77 /glassfish-web-app/service-ref/service-impl-class

生成されたサービス実装クラスの名称を指定します。

説明

生成されたサービス実装クラスの名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.78 /glassfish-web-app/service-ref/service-qname

参照する WSDL サービス要素を指定します。

説明

参照する WSDL サービス要素を指定します。

出現パターン

0回、または1回

3.13.79 /glassfish-web-app/service-ref/service-qname/ namespaceURI

名前空間 URI を指定します。

説明

名前空間 URI を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1回

3.13.80 /glassfish-web-app/service-ref/service-qname/localpart

QNAME のローカルパートを指定します。

説明

QNAME のローカルパートを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.81 /glassfish-web-app/message-destination-ref

メッセージの物理的な送信先の名称を指定します。

説明

メッセージの物理的な参照を、キュー、トピック、またはほかの物理的な送信先の JNDI 名に、直接バインドします。

対応する Java EE RI の DD ファイルのメッセージ送信先参照が、論理的なメッセージ送信先へのメッセージ送信先リンクを指定していない場合にだけ `message-destination-ref` を使用します。

出現パターン

0 回以上

3.13.82 /glassfish-web-app/message-destination-ref/message-destination-ref-name

対応する Java EE RI の DD ファイル内に定義されたメッセージの物理的な送信先の名称を指定します。

説明

対応する Java EE RI の DD ファイル内に定義されたメッセージの物理的な送信先の名称を指定します。

指定できる値

型: String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.83 /glassfish-web-app/message-destination-ref/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean と Session Bean の両方に対して、この値は EJBHome オブジェクトのグローバル JNDI 名を指定します。Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.84 /glassfish-web-app/class-loader

クラスローダーの設定情報を指定します。

説明

Web モジュールのクラスローダーを設定します。

出現パターン

0 回、または 1 回

指定できる属性

属性名	説明	指定できる値	デフォルト値*	依存関係	出現パターン
extra-class-path	この Web モジュールに対して追加のクラスパスを、コロン区切りリストまたはセミコロン区切りリストで指定します。絶対パスまたは	型 : String	なし	—	0 回、または 1 回

属性名	説明	指定できる値	デフォルト値*	依存関係	出現パターン
	Web モジュールのルートからの相対パスを指定できます。				
delegate	<p>true の場合、Web モジュールは標準のクラスローダー委譲モデルに従い、ローカルのクラスローダーを調べる前に、その親クラスローダーに委譲します。EJB コンポーネントにアクセスする Web モジュールの場合、または Web サービスクライアントやエンドポイントとして動作する Web モジュールの場合、この値を true に設定する必要があります。</p> <p>false の場合、Web モジュールはサーブレット仕様で指定されている委譲モデルに従い、親クラスローダーを調べる前に、そのクラスローダーを調べます。ほかのモジュールと相互作用しない Web モジュールの場合にだけ、この値を false に設定すると安全です。</p> <p>java.* や javax.* を含む多くのパッケージでは、委譲の設定に関係なく、クラスまたはリソースの検索、および解決は常に親クラスローダーに委譲されます。この設定は、アプリケーションによって Java ランタイムのコアクラスがオーバーライドされたり、Java EE プラットフォームの一部である仕様の API バージョンが変更されたりすることを防ぎます。</p>	型：Boolean {true false}	true	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.85 /glassfish-web-app/class-loader/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0 回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
ignoreHiddenJarFiles	true の場合、WEB-INF/lib ディレクトリー内のピリオド (.) で始まるのすべての JAR ファイルおよび ZIP ファイルを、クラスローダーが無視することを指定します。	型：Boolean {true false}	false	—	0回、または1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.86 /glassfish-web-app/class-loader/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.87 /glassfish-web-app/jsp-config

JSP の設定情報を指定します。

説明

JSP の設定情報を指定します。

出現パターン

0回、または1回

3.13.88 /glassfish-web-app/jsp-config/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
checkInterval	jsp-config のdevelopment プロパティが false に設定されていて、checkInterval の時間が 0 より大きい場合、バックグラウンドでのコンパイルが有効になります。 checkInterval は、JSP ファイルに変更があるかどうかを確認するための検証の秒単位の間隔です。変更があった場合、JSP ファイルを再コンパイルする必要があります。	型：Integer 0～ 2147483647	0	—	0 回、または 1 回
classdebuginfo	自動生成される Java サブレットを、デバッグオプション (javac コマンドの-g オプション) を有効にしてコンパイルするかどうかを指定します。	型：Boolean {true false}	true	—	0 回、または 1 回
classpath	生成されたサブレットをコンパイルするときに使用するクラスパスを指定します。	型：String	現在の Web アプリケーションに基づくクラスパス	—	0 回、または 1 回
compilerSourceVM	生成されたサブレットのソースの互換性を提供する JDK リリースを指定します。javac の-source release オプションと同じです。	型：String • 1.5 または 5 • 1.6 または 6 • 1.7 または 7 • 1.8 または 8	1.5	—	0 回、または 1 回
compilerTargetVM	サブレットクラスファイルを生成する対象の Java プラットフォーム (JVM ソフトウェア) バージョンに対して仮想マシンを指定します。javac の-target release オプションと同じです。	型：String • 1.5 または 5 • 1.6 または 6 • 1.7 または 7 • 1.8 または 8	1.5	—	0 回、または 1 回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
defaultBufferNone	true の場合、page ディレクティブのbuffer 属性のデフォルトはnone です。	型：Boolean {true false}	false	—	0 回、または 1 回
development	true に設定した場合、JSP ファイルに変更があるかどうかのチェックを可能にする、開発モードを有効にします。modification TestInterval プロパティを使用して、JSP をチェックする頻度を指定します。	型：Boolean {true false}	true	—	0 回、または 1 回
dumpSmap	true に設定した場合、JSR 45 デバッグの SMAP 情報をファイルにダンプします。suppressSmap の値がtrue の場合、この値を false に設定します。	型：Boolean {true false}	false	—	0 回、または 1 回
enablePooling	true に設定した場合、タグハンドラーのプーリングが有効になります。	型：Boolean {true false}	true	—	0 回、または 1 回
enableTldValidation	true に設定した場合、基礎となるスキーマまたは DTD ファイルに対して、Web アプリケーションが参照するすべての TLD ファイルを検証します。	型：Boolean {true false}	false	—	0 回、または 1 回
errorOnUseBeanInvalidClassAttribute	true に設定した場合、useBean アクションの class 属性値が bean クラスとして有効でない場合にエラーを発行します。	型：Boolean {true false}	false	—	0 回、または 1 回
fork	Application Server が動作しているのとは別の JavaVM を使用して、Ant が JSP ファイルをコンパイルするかどうかを指定します。	型：Boolean {true false}	true	—	0 回、または 1 回
genStrAsByteArray	true に設定すると、ページがバッファリングされていない場合に、テキスト文字列をバイト列（ページエンコーディングによってエンコードしたバイト列）として生成します。	型：Boolean {true false}	true	—	0 回、または 1 回
genStrAsCharArray	true に設定した場合、テキスト文字列を char 配列として生成します。場合によってはパフォーマンスが向上します。	型：Boolean {true false}	false	—	0 回、または 1 回
httpMethods	JspServlet がサポートしている HTTP メソッドをコンマ区切りのリストで指定します。	型：String	アスタリスク (*) (すべてのメソッドを指定する場合)	—	0 回、または 1 回
ieClassId	Internet Explorer の Java プラグイン COM クラス ID を指定します。<jsp:plugin>タグによって使用されます。	型：String	clsid: 8AD9C840-04 4E-11D1- B3E9-00805F 499D93	—	0 回、または 1 回

3. Java EE サーバで使用する Deployment Descriptor (DD)

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
ignoreJspFragmentErrors	true に設定した場合、静的にインクルードされた JSP セグメントに関する JSP の事前コンパイルエラーを無視するように、コンパイラに指示します。最上位の JSP ファイルではないが、推奨された.jspf ではなく、.jsp または.jspx 拡張子を使用している JSP セグメントが対象です。	型：Boolean {true false}	false	—	0 回、または 1 回
initialCapacity	JSP ファイルを対応するサーブレットに変換する HashMap の初期容量を指定します。	型：Integer 0～ 2147483647	32	—	0 回、または 1 回
javaEncoding	生成された Java サーブレットのエンコーディングを指定します。 このエンコーディングは、サーブレットのコンパイルにも使用される Java コンパイラに渡されます。Web コンテナは UTF8 の使用を試みます。失敗した場合、Java のエンコーディング値の使用を試みます。	型：String	UTF8	—	0 回、または 1 回
keepgenerated	true に設定した場合、生成された Java ファイルを保持します。false に設定した場合、Java ファイルを削除します。	型：Boolean {true false}	true	—	0 回、または 1 回
mappedfile	true に設定した場合、デバッグを簡単にするために、入力行当たり 1 つの print 文で静的コンテンツを生成します。	型：Boolean {true false}	true	—	0 回、または 1 回
modificationTestInterval	JSP に変更があるかどうかをチェックする頻度を秒単位で指定します。値「0」を指定すると、アクセスするたびに JSP がチェックされます。development が true に設定されている場合にだけ使用されます。	型：Integer 0～ 2147483647	0	—	0 回、または 1 回
reload-interval	JSP ファイルに変更があるかどうかをチェックする頻度を秒単位で指定します。この値を 0 に設定すると、リクエストのたびに JSP ファイルの更新の有無がチェックされます。この値を-1 に設定すると、JSP の更新と再コンパイルのチェックが無効になります。	型：Integer -1～ 2147483647	0	—	0 回、または 1 回
saveBytecode	true に設定した場合、生成されたバイトコードは、.class ファイルに保存されます。このオプションは、Java compiler API の JSR 199 (Java 6 で利用でき、デフォルトとして使用されている) が javac コンパイルに使用されている場合にだけ有効です。	型：Boolean {true false}	jspc の場合は true、 その他の場合は false	—	0 回、または 1 回
scratchdir	生成されたすべてのコードを格納するために作成される作業ディレクトリを指定します。	型：String	Application Server のイ	—	0 回、または 1 回

3. Java EE サーバで使用する Deployment Descriptor (DD)

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
			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/ generated/ jsp/[ア プ リ ケ ー シ ョ ン 名]モ ジュール名		
suppressSnap	true に設定した場合、JSR 45 デバッグの SMAP 情報の生成が抑制されます。	型：Boolean {true false}	false	—	0 回、または 1 回
trimSpaces	true に設定した場合、アクションやディレクトタイプの間のテンプレートテキストの空白を削除します。	型：Boolean {true false}	false	—	0 回、または 1 回
usePrecompiled	true に設定した場合、アクセスされた JSP ファイルはコンパイルされません。代わりに、事前コンパイルされたサーブレットクラスが使用されます。JSP ファイルが事前コンパイルされていて、対応するサーブレットクラスが Web アプリケーションのWEB-INF/lib またはWEB-INF/classes ディレクトリーにバンドルされていると見なされます。	型：Boolean {true false}	false	—	0 回、または 1 回
xpoweredBy	true に設定した場合、生成されたサーブレットによって、X-Powered-By レスポンスヘッダーが追加されます。	型：Boolean {true false}	true	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.89 /glassfish-web-app/jsp-config/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.90 /glassfish-web-app/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	「指定できるプロパティ」のプロパティ名を指定します。	型：String	—	—	1回
value	「指定できるプロパティ」のプロパティの値（指定できる値）を指定します。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

指定できるプロパティ

「指定できる属性」に指定できるプロパティを次に示します。

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
allowLinking	<p>Web アプリケーション内のシンボリックリンクで表されるリソースをサーバ提供するかどうかを指定します。</p> <p>true に設定すると、この Web アプリケーション内のリソース（シンボリックリンク）が提供されます。このプロパティを仮想サーバに定義することもできます。このプロパティを定義していない仮想サーバ上の Web アプリケーションは、仮想サーバの値を使用します。</p> <p>Windows システムでこのプロパティを true に設定すると、JSP ソースコードが開示されます。</p>	<p>型：Boolean</p> <p>{true false}</p>	false	—	0 回、または 1 回
alternatedocroot_n	<p>代替ドキュメントルート（docroot）を指定します。ここでは、<i>n</i> は正の整数です。代替 docroot によって、Web アプリケーションは特定のリソースに対するそれぞれの docroot 以外からのリクエストに対応できます。これは、それらのリクエストが Web アプリケーションの代替 docroot の URI パターンの 1 つ（または複数）に一致するかどうかに基づきます。</p> <p>リクエストが代替 docroot の URI パターンと一致する場合、リクエストの URI（Web アプリケーションのコンテキストルートをマイナスしたもの）を代替 docroot の物理的な位置（ディレクトリー）に付加することによって、リクエストは代替 docroot にマップされます。リクエストが複数の URI パターンに一致する場合、代替 docroot は次の優先順位で決定されます。</p> <ul style="list-style-type: none"> • 完全一致 • 最長パス一致 • 拡張一致 	<p>型：String</p>	—	—	0 回、または 1 回
crossContextAllowed	<p>Web アプリケーションが ServletContext.getContext() メソッドを使用してほかの Web アプリケーションのコンテキストにアクセスできるようにするには、true を指定します。</p>	<p>型：Boolean</p> <p>{true false}</p>	true	—	0 回、または 1 回
relativeRedirectAllowed	<p>この Web アプリケーションが HttpServletResponse.sendRedirect() を使用して相対 URL をクライアントに送信できるよ</p>	<p>型：Boolean</p> <p>{true false}</p>	false	—	0 回、または 1 回

プロパティ名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
	うするには、true を指定し、Web コンテナが相対 URL を完全な URL に変換しないように指示します。				
reuseSessionID	この Web アプリケーション用に生成されたセッションがリクエストで指定されたセッション ID を使用する場合、true を指定します。	型：Boolean {true false}	false	—	0 回、または 1 回
securePagesWithPragma	SSL を使用した Web アプリケーションファイルのダウンロードが Internet Explorer で正しく機能するようにするには、false を指定します。 特定の仮想サーバ上のすべての Web アプリケーションに対して、このプロパティを設定します。	型：Boolean {true false}	true	—	0 回、または 1 回
singleThreadedServletPoolSize	Web アプリケーション内の各 SingleThreadModel サブレットに割り当てられるサブレットインスタンスの最大数を指定します。	型：Integer 1 ~ 2147483647	5	—	0 回、または 1 回
tempdir	Web モジュールが使用する一時ディレクトリを指定します。この値は、javax.servlet.context.tempdir コンテキスト属性の値を構築するために使用されます。コンパイルされた JSP ファイルもこのディレクトリに配置されます。	型：String	<domain-dir/generated/app-name> または <domain-dir/generated/module-name>	—	0 回、または 1 回
useResponseCTForHeaders	レスポンスヘッダーがデフォルト (UTF-8) ではなくレスポンスの charset を使用してエンコードされる場合、true を指定します。	型：Boolean {true false}	false	—	0 回、または 1 回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.91 /glassfish-web-app/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.92 /glassfish-web-app/valve

カスタムバルブを指定します。

説明

この Web アプリケーションにカスタムバルブを指定します。特定の仮想サーバ上のすべての Web アプリケーションに対して、バルブを定義できます。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	バルブに一意の名称を指定します。	型：String	—	—	1回
class-name	バルブの完全修飾クラス名を指定します。バルブクラスは、org.glassfish.web.valve.GlassFishValve インターフェースを実装する必要があります。ただし、org.apache.catalina.Valve インターフェースを実装している場合は指定できません。	型：String	—	—	1回

(凡例)

—：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.93 /glassfish-web-app/valve/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

0回、または1回

3.13.94 /glassfish-web-app/valve/property

名称と値を持つプロパティを指定します。

説明

名称と値を持つプロパティを指定します。

出現パターン

0回以上

指定できる属性

属性名	説明	指定できる値	デフォルト値 ※	依存関係	出現パターン
name	プロパティの名称を指定します。	型：String	—	—	1回
value	プロパティの値を指定します。	型：String	—	—	1回

(凡例)

－：該当しません。

注※

定義項目の省略時のデフォルト値です。

3.13.95 /glassfish-web-app/valve/property/description

プロパティを説明する任意のテキストを指定します。

説明

プロパティを説明する任意のテキストを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.13.96 /glassfish-web-app/message-destination

メッセージの論理的な送信先の名称を指定します。

説明

アプリケーション内に定義された論理的なメッセージ送信先の名称を指定します。メッセージ送信先名は、対応する Java EE RI の DD ファイル内の対応するメッセージ送信先名と一致します。

対応する Java EE RI の DD ファイルのメッセージ送信先参照が、論理的なメッセージ送信先へのメッセージ送信先リンクを指定している場合に使用します。

出現パターン

0回以上

3.13.97 /glassfish-web-app/message-destination/message-destination-name

対応する Java EE RI の DD ファイル内に定義されたメッセージの論理的な送信先の名称を指定します。

説明

対応する Java EE RI の DD ファイル内に定義されたメッセージの論理的な送信先の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.13.98 /glassfish-web-app/message-destination/jndi-name

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

説明

URL リソースまたはリソースの JNDI 完全修飾名を指定します。

Entity Bean および Session Bean に対して、この値は EJB Home オブジェクトのグローバル JNDI 名を指定します。Entity Bean または Session Bean がリモートインターフェースを実装している場合にだけ必要です。

指定できる値

型：String

デフォルト値

定義項目の省略
なし

出現パターン

1 回

3.13.99 /glassfish-web-app/webservice-description

Web サービスの名称と任意のパブリッシュする場所を指定します。

説明

Web サービスの名称と任意のパブリッシュする場所を指定します。

出現パターン

0 回以上

3.13.100 /glassfish-web-app/webservice-description/webservice-description-name

Web 内の Web サービスに名称を指定します。

説明

Web 内で Web サービスに一意の名称を指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

出現パターン

1 回

3.13.101 /glassfish-web-app/webservice-description/wsdl-publish-location

デプロイメント中に Web サービスの WSDL をパブリッシュするディレクトリーの URL を指定します。

説明

配備中に Web サービスの WSDL をパブリッシュするディレクトリーの URL を指定します。モジュール固有の WSDL ディレクトリー (META-INF/wsdl または WEB-INF/wsdl) に対するファイルの相対位置を保持しながら、必要なファイルがこのディレクトリーにパブリッシュされます。

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0 回、または 1 回

3.13.102 /glassfish-web-app/version-identifier

Web アプリケーションのバージョン情報を指定します。

説明

アプリケーションまたはモジュールのバージョン情報を指定します。バージョン識別子は、モジュール名またはアプリケーション名の接尾辞です。コロン (:) で名前から区切ります。文字か数字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。

次の例では、アプリケーションの有効なバージョン識別子を示します。

- アプリケーション名:1
- アプリケーション名:BETA-2e
- アプリケーション名:3.8
- アプリケーション名:patch39875

指定できる値

型 : String

デフォルト値

定義項目の省略

なし

出現パターン

0回、または1回

3.14 hitachi-application.xml

Java EE アプリケーションの設定をするための拡張 DD です。

説明

Java EE アプリケーションに関する、Java EE Server 独自の設定をするための DD です。

スキーマ

```
<!DOCTYPE hitachi-application PUBLIC
'--//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Java EE Application//EN'
'hitachi-application_10_00_00.dtd'>
```

格納先

EAR のルート /META-INF/hitachi-application.xml

要素の階層

```
hitachi-application
```

記述例

```
<!DOCTYPE hitachi-application PUBLIC
'--//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Java EE Application//EN'
'hitachi-application_10_00_00.dtd'>
<hitachi-application>
</hitachi-application>
```

注意事項

ここに記載した要素以外に、Java EE RI の DD の要素 (glassfish-application.xml の子要素) を記述できます。

3.15 hitachi-application.xml に設定する要素の詳細

hitachi-application.xml に設定する要素について説明しています。

3.15.1 /hitachi-application

ルート要素です。

説明

拡張 DD (アプリケーション) のルート要素です。

出現パターン

1 回

3.16 hitachi-application-client.xml

アプリケーションクライアントの設定をするための拡張 DD です。

説明

アプリケーションクライアントに関する、Java EE Server 独自の設定をするための DD です。

スキーマ

```
<!DOCTYPE hitachi-application-client PUBLIC
' -//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Java EE Application Client//EN'
' hitachi-application-client_10_00_00.dtd' >
```

格納先

アプリケーションクライアントJARのルート/META-INF/hitachi-application-client.xml

要素の階層

```
hitachi-application-client
```

記述例

```
<!DOCTYPE hitachi-application-client PUBLIC
' -//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Java EE Application Client//EN'
' hitachi-application-client_10_00_00.dtd' >
<hitachi-application-client>
</hitachi-application-client>
```

注意事項

ここに記載した要素以外に、Java EE RI の DD の要素 (glassfish-application-client.xml の子要素) を記述できます。

3.17 hitachi-application-client.xml に設定する要素の詳細

hitachi-application-client.xml に設定する要素について説明しています。

3.17.1 /hitachi-application-client

ルート要素です。

説明

拡張 DD (アプリケーションクライアント) のルート要素です。

出現パターン

1 回

3.18 hitachi-ejb-jar.xml

EJB-JAR の設定をするための拡張 DD です。

説明

EJB-JAR に関する、Java EE Server 独自の設定をするための DD です。

スキーマ

```
<!DOCTYPE hitachi-ejb-jar PUBLIC
' -//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 EJB//EN'
' hitachi-ejb-jar_10_00_00.dtd' >
```

格納先

EJB-JAR のルート /META-INF/hitachi-ejb-jar.xml

WAR のルート /WEB-INF/hitachi-ejb-jar.xml

要素の階層

```
hitachi-ejb-jar
```

記述例

```
<!DOCTYPE hitachi-ejb-jar PUBLIC
' -//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 EJB//EN'
' hitachi-ejb-jar_10_00_00.dtd' >
<hitachi-ejb-jar>
</hitachi-ejb-jar>
```

注意事項

ここに記載した要素以外に、Java EE RI の DD の要素 (glassfish-ejb-jar.xml の子要素) を記述できません。

3.19 hitachi-ejb-jar.xml に設定する要素の詳細

hitachi-ejb-jar.xml に設定する要素について説明しています。

3.19.1 /hitachi-ejb-jar

ルート要素です。

説明

拡張 DD (EJB-JAR) のルート要素です。

出現パターン

1 回

3.20 hitachi-ra.xml

リソースの設定をするための拡張 DD です。

説明

リソースに関する、Java EE Server 独自の設定をするための DD です。

スキーマ

```
<!DOCTYPE hitachi-ra PUBLIC
' -//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Connector//EN'
' hitachi-ra_10_00_00.dtd' >
```

格納先

*RAR*のルート /META-INF/hitachi-ra.xml

要素の階層

```
hitachi-ra
```

記述例

```
<!DOCTYPE hitachi-ra PUBLIC
' -//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Connector//EN'
' hitachi-ra_10_00_00.dtd' >
<hitachi-ra>
</hitachi-ra>
```

注意事項

ここに記載した要素以外に、Java EE RI の DD の要素 (sun-ra.xml の子要素) を記述できます。

3.21 hitachi-ra.xml に設定する要素の詳細

hitachi-ra.xml に設定する要素について説明しています。

3.21.1 /hitachi-ra

ルート要素です。

説明

拡張 DD (リソースアダプター) のルート要素です。

出現パターン

1 回

3.22 hitachi-web.xml

Web アプリケーション (WAR) の設定をするための拡張 DD です。

説明

Web アプリケーション (WAR) に関する、Java EE Server 独自の設定をするための DD です。

スキーマ

```
<!DOCTYPE hitachi-web PUBLIC
'--//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Servlet//EN'
'hitachi-web_10_00_00.dtd'>
```

格納先

WAR のルート /WEB-INF/hitachi-web.xml

要素の階層

```
hitachi-web
```

記述例

```
<!DOCTYPE hitachi-web PUBLIC
'--//Hitachi, Ltd.//DTD Hitachi Java EE Server 10.0.0 Servlet//EN'
'hitachi-web_10_00_00.dtd'>
<hitachi-web>
</hitachi-web>
```

注意事項

ここに記載した要素以外に、Java EE RI の DD の要素 (glassfish-web.xml の子要素) を記述できます。

3.23 hitachi-web.xml に設定する要素の詳細

hitachi-web.xml に設定する要素について説明しています。

3.23.1 /hitachi-web

ルート要素です。

説明

拡張 DD (WAR) のルート要素です。

出現パターン

1 回

3.24 標準 DD で設定できる独自のパラメーターの詳細

標準 DD で設定できる独自のパラメーターについて説明しています。

3.24.1 jersey.config.server.provider.classnames

JAX-RS のサーブレットに対するリソースとプロバイダーを定義します。

説明

JAX-RS のサーブレットに対するリソースやプロバイダーを指定します。このパラメーターは、`init-param` 要素に設定します。

このパラメーターの設定方法を次に示します。

```
<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
<init-param>
  <param-name>jersey.config.server.provider.classnames</param-name>
  <param-value>指定値</param-value>
</init-param>
```

*指定値*には、1 つ以上のクラスパスを指定します。クラスパスを複数指定する場合は、区切り文字を使用します。区切り文字には、半角スペース、`[,]` `[/]` `[\n]` が使用できます。

JAX-RS のサーバ側で、電文のヘッダーをログに出力したい場合は、*指定値*に `org.glassfish.jersey.filter.LoggingFilter` を指定します。

パラメーターの設定対象となるファイル

`web.xml`

パラメーターの設定対象となる要素

`/web-app/servlet/init-param`

指定できる値

`org.glassfish.jersey.filter.LoggingFilter`

4

Java EE Server で使用するシステムプロパティ

Java EE Server の動作環境を設定するためのプロパティを説明しています。

4.1 Java EE Server で使用するシステムプロパティの一覧

Java EE Server で使用するすべてのシステムプロパティの概要を説明しています。

ドメイン管理サーバおよび Java EE Server で使用するシステムプロパティ

システムプロパティ	概要	詳細説明
4.2.1 ASADMIN_LISTENER_PORT	ドメイン管理サーバがインスタンスを管理するためにインスタンスへ接続する HTTP ポートまたは HTTPS ポートのポート番号の指定	ドメイン管理サーバがインスタンスを管理するためにインスタンスへ接続する HTTP ポートまたは HTTPS ポートのポート番号を指定します。有効な値は 1~65535 です。
4.2.2 com.hitachi.software.javaee.util.prf.PRFD	接続先の PRF デーモンの PRF 識別子	アプリケーションクライアントの性能解析トレース出力で使用する、PRF デーモンの PRF 識別子を指定します。
4.2.3 com.sun.appserv.transaction.nofdsync	トランザクションのステータスファイルの同期を無効に設定	JavaVM オプションにこのシステムプロパティを設定して、トランザクションのステータスファイルの同期を無効にします。
4.2.4 com.sun.corba.ee.OBVAHost	パブリックアドレスのホストの設定	パブリックアドレスのホストを設定します。
4.2.5 com.sun.corba.ee.OBVAAPort	パブリックアドレスのポートの指定	パブリックアドレスのポートを設定します。
4.2.6 com.sun.enterprise.in-progress-local-transaction.completion-mode	Java EE Server がシャットダウン時に非 XA トランザクションをコミットするかロールバックするかの指定	サーバインスタンスのシャットダウン時に、すべてのデータベースコネクションが切断されます。OracleJDBC では、ドライバをベースにしたデータベースコネクションが非 XA トランザクションの最中に切断されると、すべての保留中の変更がコミットされます。ほかのデータベースでは、接続が切断された場合、通常、保留中の変更はコミットされないでロールバックされます。
4.2.7 com.sun.jts.dblogging.deletequery	データベースに保存されたトランザクションサービスログを削除する際に使用する SQL の定義	トランザクションマネージャーが、データベースに保存されたトランザクションサービスログを削除する際に使用する SQL を定義します。
4.2.8 com.sun.jts.dblogging.insertquery	トランザクションサービスログを保存する際に使用する SQL の定義	トランザクションマネージャーが、トランザクションサービスログを保存する際に使用する SQL を定義します。
4.2.9 com.sun.jts.dblogging.selectquery	データベースに保存されたトランザクションサービスログを取得する際に使用する SQL の定義	トランザクションマネージャーが、データベースに保存されたトランザクションサービスログを取得する際に使用する SQL を定義します。
4.2.10 com.sun.jts.dblogging.selectservernamequery	データベースに保存されたトランザクションサービスログのサーバ名を取得する際に使用する SQL の定義	トランザクションマネージャーが、データベースに保存されたトランザクションサービスログのサーバ名を取得する際に使用する SQL を定義します。

システムプロパティ	概要	詳細説明
4.2.11 com.sun.jts.dblogging.use.nontx.connection.for.add	トランザクションを利用しないコネクションを使用したデータベースへのトランザクションサービスログの登録	トランザクションを利用しないコネクションを使用して、データベースにトランザクションサービスログを登録します。
4.2.12 com.sun.xml.ws.spi.d.b.BindingContextFactory	JAX-WS の製品内部処理で利用する JAXB のバインディングプロバイダーの指定	JAX-WS の製品内部処理で利用する JAXB のバインディングプロバイダーを指定します。
4.2.13 com.sun.xml.ws.transport.http.client.HttpTransportPipe.dump	クライアント側での JAX-WS のリクエストおよびレスポンスを標準出力する機能の設定	このシステムプロパティに true を設定することで、クライアント側で、JAX-WS のリクエストおよびレスポンスを標準出力する機能を有効にします。
4.2.14 com.sun.xml.ws.transport.http.HttpAdapter.dump	サーバ側での JAX-WS のリクエストおよびレスポンスを標準出力する機能の設定	このシステムプロパティに true を設定することで、サーバ側で、JAX-WS のリクエストおよびレスポンスを標準出力する機能を有効にします。
4.2.15 eclipselink.security.usedoprivileged	セキュリティーマネージャーを有効にする環境で、JPA 利用するための設定	セキュリティーマネージャーを有効にする環境で、 <code>javax.persistence.Persistence.createEntityManagerFactory</code> メソッドを利用するアプリケーションを使用する場合、このプロパティを true に設定する必要があります。
4.2.16 EXTERNAL-ADDR	クラスターインスタンスの作成時の External IP アドレスの設定	クラスターインスタンスの作成時に External IP アドレスを設定します。
4.2.17 GMS-BIND-INTERFACE-ADDRESS-cluster-name	GMS-BIND-INTERFACE-ADDRESS-cluster-name システムプロパティの指定	インスタンスがグループ管理サービスによって管理されていて、マルチホームホストを表すノードに存在する場合、GMS-BIND-INTERFACE-ADDRESS-cluster-name システムプロパティを指定します。
4.2.18 GMS_LISTENER_PORT-cluster-name	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番号の定義	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番号を定義します。
4.2.19 HTTP_LISTENER_PORT	HTTP リクエストをリッスンするために使用するポート番号の指定	HTTP リクエストをリッスンするために使用するポート番号を指定します。
4.2.20 HTTP_SSL_LISTENER_PORT	HTTPS リクエストをリッスンするポートのポート番号の指定	HTTPS リクエスト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。
4.2.21 IIOP_LISTENER_PORT	IIOP 接続で使われるポートのポート番号の指定	IIOP 接続で使われるポートのポート番号を指定します。
4.2.22 IIOP_SSL_LISTENER_PORT	セキュア IIOP 接続で使われるポートのポート番号の指定	セキュア IIOP 接続で使われるポートのポート番号を指定します。

4. Java EE Server で使用するシステムプロパティ

システムプロパティ	概要	詳細説明
4.2.23 IIOP_SSL_MUTUAL_AUTH_PORT	クライアント認証を伴うセキュア IIOP 接続で使われるポート番号の指定	クライアント認証を伴うセキュア IIOP 接続で使われるポートのポート番号を指定します。
4.2.24 java.security.auth.login.config	ログインモジュールの使用順序と使用条件の指定	ログインモジュールが2つ以上使われる場合、使われる順序とその使用条件を指定します。これは、Java SE が提供するシステムプロパティです。
4.2.25 java.security.policy	server.policy ファイルのロケーションの指定	server.policy ファイルのロケーションを指定します。これは、Java SE が提供するシステムプロパティです。
4.2.26 JAVA_DEBUGGER_PORT	Java Platform Debugger Architecture (JPDA) デバッガーとの接続に使用されるポート番号の指定	Java Platform Debugger Architecture (JPDA) デバッガーとの接続に使用されるポートのポート番号を指定します。
4.2.27 JMS_PROVIDER_PORT	JMS プロバイダーのポート番号の指定	JMS プロバイダーのポート番号を指定します。
4.2.28 JMX_SYSTEM_CONNECTOR_PORT	JMX コネクタがリッスンするポート番号の指定	JMX コネクタがリッスンするポート番号を指定します。
4.2.29 sun.rmi.dgc.client.gcInterval	定期的発生する GC の最大間隔の指定	リモートオブジェクトの参照に対して定期的 GC が発生することがあります。このシステムプロパティでは、定期的発生する GC の最大間隔を指定します。単位はミリ秒です。sun.rmi.dgc.server.gcInterval もあわせて指定する必要があります。
4.2.30 sun.rmi.dgc.server.gcInterval	定期的発生する GC の最大間隔の指定	リモートオブジェクトの参照に対して定期的 GC が発生することがあります。このシステムプロパティでは、定期的発生する GC の最大間隔を指定します。単位はミリ秒です。sun.rmi.dgc.client.gcInterval もあわせて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で使用するシステムプロパティ

システムプロパティ	概要	詳細説明
4.3.1 org.glassfish.appclient.acc.maxLoginRetries	許容されるログインのリトライ回数の決定	許容されるログインのリトライ回数を決定します。

4.2 ドメイン管理サーバおよび Java EE サーバで使用するシステムプロパティの詳細

ドメイン管理サーバおよび Java EE サーバの動作環境を設定するためのプロパティを説明しています。

4.2.1 ASADMIN_LISTENER_PORT

ドメイン管理サーバがインスタンスを管理するためにインスタンスへ接続する HTTP ポートまたは HTTPS ポートのポート番号を指定します。

説明

ドメイン管理サーバがインスタンスを管理するためにインスタンスへ接続する HTTP ポートまたは HTTPS ポートのポート番号を指定します。有効な値は 1~65535 です。

指定できる値

型：Integer

1~65535

初期値

24848

デフォルト値

定義項目の省略

4848

記述例

```
asadmin> set servers.server.pmdsainst.system-property
.ASADMIN_LISTENER_PORT.value=24859
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.2 com.hitachi.software.javaee.util.prf.PRFID

接続先の PRF デーモンの PRF 識別子を指定します。

説明

アプリケーションクライアントの性能解析トレース出力で使用する、PRF デーモンの PRF 識別子を指定します。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略

PRF_ID

値の省略

PRF_ID

記述例

```
appclient -Dcom.hitachi.software.javaee.util.prf.PRFID=MyPrfId Test
```

4.2.3 com.sun.appserv.transaction.nofdsync

トランザクションのステータスファイルの同期を無効にします。

説明

JavaVM オプションにこのシステムプロパティを設定して、トランザクションのステータスファイルの同期を無効にします。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略

なし

記述例

```
asadmin> create-jvm-options --target instance1  
-Dcom.sun.appserv.transaction.nofdsync
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.4 com.sun.corba.ee.ORBVAAMHost

パブリックアドレスのホストを設定します。

説明

パブリックアドレスのホストを設定します。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略

なし

記述例

```
asadmin> create-jvm-options --target instance1  
-Dcom.sun.corba.ee.ORBVAAMHost=192.155.35.4
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.5 com.sun.corba.ee.ORBVAAPort

パブリックアドレスのポートを設定します。

説明

パブリックアドレスのポートを設定します。

指定できる値

型 : Integer

1~65535

初期値

指定なし

デフォルト値

定義項目の省略

なし

記述例

```
asadmin> create-jvm-options --target instance1  
-Dcom.sun.corba.ee.ORBVAAPort=9090
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.6 com.sun.enterprise.in-progress-local-transaction.completion-mode

Java EE Server が、シャットダウン時に、非 XA トランザクション (JTA トランザクションなど) をコミットするかロールバックするかを指定します。

説明

サーバインスタンスのシャットダウン時に、すべてのデータベースコネクションが切断されます。Oracle JDBC ドライバーを使用したデータベースコネクションが非 XA トランザクションの最中に切断されると、すべての保留中の変更がコミットされます。ほかのデータベースでは、接続が切断された場合、通常、保留中の変更はコミットされないでロールバックされます。

Java EE Server が、シャットダウン時に、非 XA トランザクションをコミットするかロールバックするかを明示的に指定するには、`create-jvm-options` サブコマンドを使用して、`com.sun.enterprise.in-progress-local-transaction.completion-mode` JVM オプションをコミットまたはロールバックに設定します。

指定できる値

型 : String

{commit | rollback}

初期値

rollback

デフォルト値

定義項目の省略

commit も rollback もしません。

記述例

```
asadmin> create-jvm-options --target instance1  
-Dcom.sun.enterprise.in-progress-local-transaction.completion-mode=rollback
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.7 com.sun.jts.dblogging.deletequery

データベースに保存されたトランザクションサービスを削除する際に使用する SQL を定義します。

説明

トランザクションマネージャーが、データベースに保存されたトランザクションサービスを削除する際に使用する SQL を定義します。

指定できる値

型 : String

初期値

指定なし

デフォルト値

定義項目の省略

```
delete from txn_log_table where localtid = ? and servername = ?
```

記述例

```
asadmin> create-jvm-options --target instance1  
'-Dcom.sun.jts.dblogging.deletequery=delete from txn_log_table where localtid =  
? and servername = ?'
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.8 com.sun.jts.dblogging.insertquery

トランザクションサービスログを保存する際に使用する SQL を定義します。

説明

トランザクションマネージャーが、トランザクションサービスログを保存する際に使用する SQL を定義します。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略

```
insert into txn_log_table values ( ?, ?, ?, ?)
```

記述例

```
asadmin> create-jvm-options --target instance1  
'-Dcom.sun.jts.dblogging.insertquery=insert into txn_log_table values  
( ?, ?, ?, ?)'
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.9 com.sun.jts.dblogging.selectquery

データベースに保存されたトランザクションサービスログを取得する際に使用する SQL を定義します。

説明

トランザクションマネージャーが、データベースに保存されたトランザクションサービスログを取得する際に使用する SQL を定義します。

指定できる値

型: String

初期値

指定なし

デフォルト値

定義項目の省略

```
select * from txn_log_table where servername = ?
```

記述例

```
asadmin> create-jvm-options --target instance1  
'-Dcom.sun.jts.dblogging.selectquery=  
select * from txn_log_table where servername = ?'
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.10 com.sun.jts.dblogging.selectservernamequery

データベースに保存されたトランザクションサービスログのサーバ名を取得する際に使用する SQL を定義します。

説明

トランザクションマネージャーが、データベースに保存されたトランザクションサービスログのサーバ名を取得する際に使用する SQL を定義します。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略

```
select distinct servername from txn_log_table where instancename = ?
```

記述例

```
asadmin> create-jvm-options --target instance1  
'-Dcom.sun.jts.dblogging.selectservernamequery=  
select distinct servername from txn_log_table where instancename = ?'
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.11 com.sun.jts.dblogging.use.nontx.connection.for.add

トランザクションを利用しないコネクションを使用して、データベースにトランザクションサービスログを登録します。

説明

トランザクションを利用しないコネクションを使用して、データベースにトランザクションサービスログを登録します。

true

データベースへのトランザクションサービスログの登録に、トランザクションを利用しないコネクションを利用します。次の条件をすべて満たす場合に、このプロパティを true にする必要があります。

- トランザクションサービスログをデータベースに保存したい場合
- 既存の transactional resource を使いたい場合
- INSERT だけでは、non-transactional connection を使いたい場合

false

データベースへのトランザクションサービスログの登録に、トランザクションを利用しないコネクションを利用しません。

指定できる値

型：Boolean

{true | false}

初期値

指定なし

デフォルト値

定義項目の省略

false

記述例

```
asadmin> create-jvm-options --target instance1  
-Dcom.sun.jts.dblogging.use.nontx.connection.for.add=true
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.12 com.sun.xml.ws.spi.db.BindingContextFactory

JAX-WS の製品内部処理で利用する JAXB のバインディングプロバイダーを指定します。

説明

JAX-WS の製品内部処理で利用する JAXB のバインディングプロバイダーを指定します。

com.sun.xml.ws.db.glassfish.JAXBRIContextFactory

JAXB リファレンス実装を指定します。これがデフォルトになります。

com.sun.xml.ws.db.toplink.JAXBContextFactory

EclipseLink MOxy JAXB binding を指定します。

指定できる値

型：String

```
{com.sun.xml.ws.db.glassfish.JAXBRIContextFactory |  
com.sun.xml.ws.db.toplink.JAXBContextFactory}
```

初期値

指定なし

デフォルト値

定義項目の省略

```
com.sun.xml.ws.db.glassfish.JAXBRIContextFactory
```

記述例

```
asadmin> create-jvm-options --target instance1  
-Dcom.sun.xml.ws.spi.db.BindingContextFactory=  
com.sun.xml.ws.db.toplink.JAXBContextFactory
```

注意事項

Java EE サーバで使用するシステムプロパティです。

このシステムプロパティで指定した JAXB のデータバインディングプロバイダーは、JAX-WS に関する製品の処理だけで適用されます。

4.2.13

com.sun.xml.ws.transport.http.client.HttpTransportPipe.dump

このシステムプロパティに `true` を設定することで、クライアント側で、JAX-WS のリクエストおよびレスポンスを標準出力する機能を有効にします。

説明

このシステムプロパティに `true` を設定することで、クライアント側で、JAX-WS のリクエストおよびレスポンスを標準出力する機能を有効にします。

指定できる値

型：Boolean

```
{true | false}
```

初期値

指定なし

デフォルト値

定義項目の省略

false

記述例

```
asadmin> create-system-properties --target instance1  
com.sun.xml.ws.transport.http.HttpTransportPipe.dump=true
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.14 com.sun.xml.ws.transport.http.HttpAdapter.dump

このシステムプロパティにtrueを設定することで、サーバ側で、JAX-WSのリクエストおよびレスポンスを標準出力する機能を有効にします。

説明

このシステムプロパティにtrueを設定することで、サーバ側で、JAX-WSのリクエストおよびレスポンスを標準出力する機能を有効にします。

指定できる値

型：Boolean

{true | false}

初期値

指定なし

デフォルト値

定義項目の省略

false

記述例

```
asadmin> create-system-properties --target instance1  
com.sun.xml.ws.transport.http.HttpAdapter.dump=true
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.15 eclipselink.security.usedoprivileged

セキュリティーマネージャーを有効にする環境で、JPA 利用するための設定をします。

説明

セキュリティーマネージャーを有効にする環境で、`javax.persistence.Persistence`の `createEntityManagerFactory` メソッドを利用するアプリケーションを使用する場合、このプロパティを `true` に設定する必要があります。

指定できる値

型：Boolean

{true | false}

初期値

指定なし

デフォルト値

定義項目の省略

なし

記述例

```
asadmin> create-jvm-options --target instance1  
-Declipselink.security.usedoprivileged=true
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.16 EXTERNAL-ADDR

クラスターインスタンスの作成時に External IP アドレスとグループ管理サービス IP アドレスを設定します。

説明

クラスターインスタンスの作成時に External IP アドレスとグループ管理サービス IP アドレスを設定します。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略
なし

記述例

```
asadmin> create-system-properties --target instance1 EXTERNAL-ADDR=192.155.35.4
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.17 GMS-BIND-INTERFACE-ADDRESS-*cluster-name*

インスタンスがグループ管理サービスによって管理されていて、マルチホームホストを表すノードに存在する場合、GMS-BIND-INTERFACE-ADDRESS-*cluster-name* システムプロパティを指定します。

説明

インスタンスがグループ管理サービスによって管理されていて、マルチホームホストを表すノードに存在する場合、GMS-BIND-INTERFACE-ADDRESS-*cluster-name* システムプロパティを指定します。*cluster-name* には、クラスター名を指定します。

指定できる値

型：String

初期値

指定なし

デフォルト値

定義項目の省略

`${GMS-BIND-INTERFACE-ADDRESS-cluster-name}`

記述例

```
asadmin> create-system-properties --target instance1  
GMS-BIND-INTERFACE-ADDRESS-c1=10.12.152.20
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.18 GMS_LISTENER_PORT-*cluster-name*

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番号を定義します。

説明

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番号を定義します。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。*cluster-name* には、クラスター名を指定します。

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。この場合、グループ管理サービスは、`GMS_TCPSTARTPORT` プロパティと `GMS_TCPENDPORT` プロパティで定義された範囲から未使用のポートを選択します。デフォルトではこの範囲は 9090～9200 です。ほとんどの場合、デフォルトの動作で十分です。

指定できる値

型：Integer

9090～9200

初期値

指定なし

デフォルト値

定義項目の省略

9090

記述例

```
asadmin> create-system-properties --target instance1  
GMS_LISTENER_PORT-tcpcluster=9090
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.19 HTTP_LISTENER_PORT

HTTP リクエストをリッスンするために使用するポート番号を指定します。

説明

HTTP リクエストをリッスンするために使用するポート番号を指定します。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

指定なし

デフォルト値

定義項目の省略

28080

記述例

```
asadmin> create-instance --node localhost-domain1  
--systemproperties HTTP_LISTENER_PORT=58294  
pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.20 HTTP_SSL_LISTENER_PORT

HTTPS リクエスト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。

説明

HTTPS リクエスト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。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

28181

デフォルト値

定義項目の省略

1043

記述例

```
asadmin> create-instance --node localhost-domain1  
--systemproperties HTTP_SSL_LISTENER_PORT=58297 pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.21 IIOP_LISTENER_PORT

IIOP 接続で使われるポートのポート番号を指定します。

説明

IIOP 接続で使われるポートのポート番号を指定します。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

23700

デフォルト値

定義項目の省略

3700

記述例

```
asadmin> create-instance --node localhost-domain1  
--systemproperties IIOP_LISTENER_PORT=58300 pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.22 IIOP_SSL_LISTENER_PORT

セキュア IIOP 接続で使われるポートのポート番号を指定します。

説明

セキュア IIOP 接続で使われるポートのポート番号を指定します。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

23820

デフォルト値

定義項目の省略

1060

記述例

```
asadmin> create-instance --node localhost-domain1
--systemproperties IIOP_SSL_LISTENER_PORT=58303 pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.23 IIOP_SSL_MUTUALAUTH_PORT

クライアント認証を伴うセキュア IIOP 接続で使われるポートのポート番号を指定します。

説明

クライアント認証を伴うセキュア IIOP 接続で使われるポートのポート番号を指定します。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

23920

デフォルト値

定義項目の省略

1061

記述例

```
asadmin> create-instance --node localhost-domain1
--systemproperties IIOP_SSL_MUTUALAUTH_PORT=58306 pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.24 java.security.auth.login.config

ログインモジュールが2つ以上使われる場合、順序とその使用条件を指定します。

説明

ログインモジュールが2つ以上使われる場合、使われる順序とその使用条件を指定します。これは、Java SE が提供するシステムプロパティです。

指定できる値

型：String

初期値

`${com.sun.aas.instanceRoot}/config/login.conf`

デフォルト値

定義項目の省略

なし

記述例

```
asadmin> create-jvm-options --target instance1  
-Djava.security.auth.login.config=${com.sun.aas.instanceRoot}/config/login.conf
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.25 java.security.policy

server.policy ファイルのロケーションを指定します。

説明

server.policy ファイルのロケーションを指定します。これは、Java SE が提供するシステムプロパティです。

このシステムプロパティの値を変更する場合は、次の点に注意してください。

- ドメインにあるすべてのサーバインスタンスに server.policy を適用する場合
このシステムプロパティの値を変更する必要はありません。
- サーバインスタンス単体で server.policy を適用する場合

次に示す server.policy ファイルを作業用フォルダにコピーした上で、そのファイルパスを指定してください。

```
Application Serverのインストールディレクトリー/javaee/glassfish/domains/ドメイン名/config/  
server.policy
```

なお、server.policy ファイルにセキュリティポリシーを追加する場合は、server.policy ファイルに設定を追記する形で編集し、既存の設定項目は削除しないでください。

指定できる値

型：String

初期値

```
${com.sun.aas.instanceRoot}/config/server.policy
```

デフォルト値

定義項目の省略

なし

記述例

```
asadmin> create-jvm-options --target instance1 -Djava.security.policy=  
${com.sun.aas.instanceRoot}/config/pmd-config/server.policy
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.26 JAVA_DEBUGGER_PORT

Java Platform Debugger Architecture (JPDA) デバッガーとの接続に使用されるポートのポート番号を指定します。

説明

Java Platform Debugger Architecture (JPDA) デバッガーとの接続に使用されるポートのポート番号を指定します。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

29009

デフォルト値

定義項目の省略

29009

記述例

```
asadmin> create-instance --node localhost-domain1  
--systemproperties JAVA_DEBUGGER_PORT=58294 pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.27 JMS_PROVIDER_PORT

JMS プロバイダーのポート番号を指定します。

説明

JMS プロバイダーのポート番号を指定します。有効な値は 1～65535 です。

指定できる値

型：Integer

1～65535

初期値

27676

デフォルト値

定義項目の省略

27676

記述例

```
asadmin> set configs.config.clusterpresets  
-config.system-property.JMS_PROVIDER_PORT.value=27678
```


注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.28 JMX_SYSTEM_CONNECTOR_PORT

JMX コネクタがリッスンするポート番号を指定します。

説明

JMX コネクタがリッスンするポート番号を指定します。有効な値は 1~65535 です。

指定できる値

型：Integer

1~65535

初期値

28686

デフォルト値

定義項目の省略

8686

記述例

```
asadmin> create-instance --node localhost-domain1  
--systemproperties JMX_SYSTEM_CONNECTOR_PORT=58309 pmdcpinst
```

注意事項

Java EE サーバで使用するシステムプロパティです。

4.2.29 sun.rmi.dgc.client.gcInterval

リモートオブジェクトの参照に対して定期的に GC が発生することがあります。このシステムプロパティでは、定期的に発生する GC の最大間隔を指定します。単位はミリ秒です。sun.rmi.dgc.server.gcInterval もあわせて指定する必要があります。sun.rmi.dgc.server.gcInterval を同時に指定しない場合、FullGC が意図しないタイミングで発生するおそれがあります。

説明

リモートオブジェクトの参照に対して定期的に GC が発生することがあります。このシステムプロパティーでは、定期的に発生する GC の最大間隔を指定します。単位はミリ秒です。sun.rmi.dgc.server.gcInterval もあわせて指定する必要があります。sun.rmi.dgc.server.gcInterval を同時に指定しない場合、FullGC が意図しないタイミングで発生するおそれがあります。

サービンスタンスでのsun.rmi.dgc.server.gcInterval と、sun.rmi.dgc.client.gcInterval の初期値は、1 日 (86400000 ミリ秒) です。これらのシステムプロパティーで FullGC の発生間隔を広げても GC 発生回数が削減されない場合、Java ヒープが不足していることがあります。この場合は、Java ヒープのチューニングを実施することで改善することがあります。

指定できる値

型：Integer

1~9223372036854775806

初期値

86400000

デフォルト値

定義項目の省略

3600000

記述例

```
asadmin> create-system-properties --target instance1  
sun.rmi.dgc.client.gcInterval=43200000
```

注意事項

Java EE サーバで使用するシステムプロパティーです。

4.2.30 sun.rmi.dgc.server.gcInterval

リモートオブジェクトの参照に対して定期的に GC が発生することがあります。このシステムプロパティーでは、定期的に発生する GC の最大間隔を指定します。単位はミリ秒です。sun.rmi.dgc.client.gcInterval もあわせて指定する必要があります。sun.rmi.dgc.client.gcInterval を同時に指定しない場合、FullGC が意図しないタイミングで発生するおそれがあります。

説明

リモートオブジェクトの参照に対して定期的に GC が発生することがあります。このシステムプロパティーでは、定期的に発生する GC の最大間隔を指定します。単位はミリ秒です。sun.rmi.dgc.client.gcInterval もあわせて指定する必要があります。sun.rmi.dgc.client.gcInterval を同時に指定しない場合、FullGC が意図しないタイミングで発生するおそれがあります。

サーバインスタンスでのsun.rmi.dgc.server.gcInterval と、sun.rmi.dgc.client.gcInterval の初期値は、1 日 (86400000 ミリ秒) です。これらのシステムプロパティーで FullGC の発生間隔を広げても GC 発生回数が削減されない場合、Java ヒープが不足していることがあります。この場合は、Java ヒープのチューニングを実施することで改善することがあります。

指定できる値

型：Integer

1~9223372036854775806

初期値

86400000

デフォルト値

定義項目の省略

3600000

記述例

```
asadmin> create-system-properties --target instance1  
sun.rmi.dgc.server.gcInterval=43200000
```

注意事項

Java EE サーバで使用するシステムプロパティーです。

4.3 アプリケーションクライアントで使用するシステムプロパティの詳細

クライアントプロセスの動作環境を設定するためのプロパティを説明しています。

4.3.1 org.glassfish.appclient.acc.maxLoginRetries

アプリケーションクライアントコンテナーがインジェクションを行う際に許容されるログインのリトライ数を決定します。

説明

許容されるログインのリトライ回数を決定します。

クライアントのメインクラスに注釈（例えば @Resource）がある場合、アプリケーションクライアントコンテナーがインジェクションを行う際、リトライループが発生します。

指定できる値

型：Integer

0~2147483647

初期値

指定なし

デフォルト値

定義項目の省略

3

記述例

```
asadmin> create-jvm-options -Dorg.glassfish.appclient.acc.maxLoginRetries=5
```

注意事項

アプリケーションクライアントで使用するシステムプロパティです。

5

Java EE Server で使用する定義ファイルおよびプロパティファイル

Java EE Server の接続情報や構成定義を設定するためのファイルを説明しています。

5.1 Java EE Server で使用する定義ファイルおよびプロパティファイルの一覧

Java EE Server で使用するすべての定義ファイルおよびプロパティファイルの概要を説明しています。

Java EE Server で使用するすべての定義ファイルおよびプロパティファイル

ファイル	概要	詳細説明
5.2 asenv.bat	Java EE Server の環境変数定義ファイル	Java EE Server 固有の環境変数を保存するファイルです。
5.4 config.properties	ブローカーの構成に関するプロパティの設定	config.properties ファイルは、ブローカーの構成に関するプロパティ設定ファイルです。
5.6 das.properties	ノードがドメイン管理サーバに接続するための接続情報を含むファイル	ノードがドメイン管理サーバに接続するための接続情報を含むファイルです。

asenv.bat に設定する環境変数

環境変数	概要	詳細説明
5.3.1 AS_ADMIN_READTIMEOUT	asadmin プロセスおよびドメイン管理サーバに設定する HTTP リードタイムアウト値の指定	asadmin プロセスとドメイン管理サーバ間の通信、およびドメイン管理サーバとサーバインスタンス間の通信に適用する、HTTP リードタイムアウト値 (単位: ミリ秒) を指定します。
5.3.2 HJES_ASADMIN_CONSOLE_ENABLED	asadmin ユーティリティコマンドのプロセス起動時ログの有効または無効の指定	asadmin ユーティリティコマンドのプロセス起動時ログを出力するかどうかを指定します。
5.3.3 HJES_ASADMIN_CONSOLE_FILE_NUMBER	asadmin ユーティリティコマンドのプロセス起動時ログファイルの個数	asadmin ユーティリティコマンドのプロセス起動時ログの Java EE Server 稼働中に保持するログファイルの個数を指定します。
5.3.4 HJES_ASADMIN_CONSOLE_ROTATION_SIZE	asadmin ユーティリティコマンドのプロセス起動時ログファイルのサイズ	asadmin ユーティリティコマンドのプロセス起動時ログのログ出力先ファイルを切り替えるサイズ (単位: バイト) を指定します。
5.3.5 HJES_ASADMIN_CONSOLE_ROTATION_TIME	asadmin ユーティリティコマンドのプロセス起動時ログファイルの切り替え時刻の指定	asadmin ユーティリティコマンドのプロセス起動時ログのログ出力先ファイルを切り替える時刻を"HHMMSS"形式で指定します。
5.3.6 HJES_ASADMIN_JVM_OPTIONS	asadmin ユーティリティコマンドのプロセスに適用する JavaVM オプションの指定	asadmin ユーティリティコマンドのプロセスに適用する JavaVM オプションを指定します。複数の JavaVM オプションを指定する場合は、半角空白で区切って、オプション全体を""で囲んでください。
5.3.7 HJES_ASADMIN_LOG_DIVISION	asadmin ユーティリティコマンドのログ出力先ディレクトリーの分割の有効または無効の指定	asadmin ユーティリティコマンドのログを、OS のユーザーアカウントごとにディレクトリーを分けて出力するかどうかを指定します。

環境変数	概要	詳細説明
5.3.8 HJES_ASADMIN_MAINTENANCE_ENABLED	asadmin ユーティリティーコマンドの保守ログの有効または無効の指定	asadmin ユーティリティーコマンドの保守ログを出力するかどうかを指定します。
5.3.9 HJES_ASADMIN_MAINTENANCE_FILE_NUMBER	asadmin ユーティリティーコマンドの保守ログファイルの面数	asadmin ユーティリティーコマンドの保守ログの Java EE Server 稼働中に保持するログファイルの個数を指定します。
5.3.10 HJES_ASADMIN_MAINTENANCE_LOG_LEVEL	asadmin ユーティリティーコマンドの保守ログのレベル	asadmin ユーティリティーコマンドの保守ログのログに出力される内容のレベルを指定します。
5.3.11 HJES_ASADMIN_MAINTENANCE_ROTATION_SIZE	asadmin ユーティリティーコマンドの保守ログファイルのサイズ	asadmin ユーティリティーコマンドの保守ログのログ出力先ファイルを切り替えるサイズ (単位: バイト) を指定します。
5.3.12 HJES_ASADMIN_MAINTENANCE_ROTATION_TIME	asadmin ユーティリティーコマンドの保守ログファイルの切り替え時刻	asadmin ユーティリティーコマンドの保守ログのログ出力先ファイルを切り替える時刻を "HHMMSS" 形式で指定します。
5.3.13 HJES_ASADMIN_MESSAGE_ENABLED	asadmin ユーティリティーコマンドのメッセージログの有効または無効の指定	asadmin ユーティリティーコマンドのメッセージログを出力するかどうかを指定します。
5.3.14 HJES_ASADMIN_MESSAGE_FILE_NUMBER	asadmin ユーティリティーコマンドのメッセージログファイルの面数	asadmin ユーティリティーコマンドのメッセージログの Java EE Server 稼働中に保持するログファイルの個数を指定します。
5.3.15 HJES_ASADMIN_MESSAGE_LOG_LEVEL	asadmin ユーティリティーコマンドのメッセージログのレベル	asadmin ユーティリティーコマンドのメッセージログのログに出力される内容のレベルを指定します。
5.3.16 HJES_ASADMIN_MESSAGE_ROTATION_SIZE	asadmin ユーティリティーコマンドのメッセージログファイルのサイズ	asadmin ユーティリティーコマンドのメッセージログのログ出力先ファイルを切り替えるサイズ (単位: バイト) を指定します。
5.3.17 HJES_ASADMIN_MESSAGE_ROTATION_TIME	asadmin ユーティリティーコマンドのメッセージログファイルの切り替え時刻	asadmin ユーティリティーコマンドのメッセージログのログ出力先ファイルを切り替える時刻を "HHMMSS" 形式で指定します。
5.3.18 HJES_ASADMIN_STACKTRACE_ENABLED	asadmin ユーティリティーコマンドのスタックトレースログの有効または無効の指定	asadmin ユーティリティーコマンドのスタックトレースログを出力するかどうかを指定します。

環境変数	概要	詳細説明
5.3.19 HJES_ASADMIN_ST ACKTRACE_FILE_N UMBER	asadmin ユーティリティーコマンドのスタックトレースログファイルの面数	asadmin ユーティリティーコマンドのスタックトレースログの Java EE Server 稼働中に保持するログファイルの個数を指定します。
5.3.20 HJES_ASADMIN_ST ACKTRACE_ROTATION_SIZE	asadmin ユーティリティーコマンドのスタックトレースログファイルのサイズ	asadmin ユーティリティーコマンドのスタックトレースログのログ出力先ファイルを切り替えるサイズ (単位: バイト) を指定します。
5.3.21 HJES_ASADMIN_ST ACKTRACE_ROTATION_TIME	asadmin ユーティリティーコマンドのスタックトレースログファイルの切り替え時刻	asadmin ユーティリティーコマンドのスタックトレースログのログ出力先ファイルを切り替える時刻を "HHMMSS" 形式で指定します。
5.3.22 HJES_LOGSDIR	Java EE Server のログファイルの出力先ディレクトリー	Java EE Server のログファイルの出力先ディレクトリーを変更する場合に指定します。

config.properties に設定するプロパティー

プロパティー	概要	詳細説明
5.5.1 imq.authentication.c lient.response.timeo ut	認証リクエストに対するクライアントからの応答を待機する時間の秒単位の指定	認証リクエストに対するクライアントからの応答を待機する時間を秒単位で指定します。この値は <code>imq.authentication.client.response.timeout</code> プロパティーで定義されています。
5.5.2 imq.autocreate.desti nation.limitBehavior	メモリー制限のしきい値に達した場合のブローカー動作の指定	メモリー制限のしきい値に達した場合のブローカー動作を指定します。
5.5.3 imq.autocreate.desti nation.maxBytesPer Msg	単一メッセージの最大サイズのバイト単位の指定	単一メッセージの最大サイズをバイト単位で指定します。
5.5.4 imq.autocreate.desti nation.maxNumMsgs	未配信メッセージの最大数の指定	未配信メッセージの最大数を指定します。
5.5.5 imq.autocreate.desti nation.maxTotalMsg Bytes	未配信メッセージの最大の総メモリー量のバイト単位の指定	未配信メッセージの最大の総メモリー量をバイト単位で指定します。
5.5.6 imq.autocreate.desti nation.useDMQ	デッドメッセージキューへのデッドメッセージの送信の有効または無効の指定	デッドメッセージキューへのデッドメッセージの送信を有効または無効にします。
5.5.7 imq.autocreate.queu e	キュー送信先の自動作成の有効または無効の指定	キュー送信先の自動作成を有効または無効にします。

プロパティ	概要	詳細説明
5.5.8 imq.autocreate.queue.maxNumActiveConsumers	アクティブなメッセージコンシューマーの最大数の指定	キュー送信先からのロードバランス配信での、アクティブなメッセージコンシューマーの最大数を指定します。
5.5.9 imq.autocreate.queue.maxNumBackupConsumers	バックアップメッセージコンシューマーの最大数の指定	キュー送信先からのロードバランス配信での、バックアップメッセージコンシューマーの最大数を指定します。
5.5.10 imq.autocreate.topic	トピック送信先の自動作成の有効または無効の指定	トピック送信先の自動作成を有効または無効にします。
5.5.11 imq.autocreate.topic.consumerFlowLimit	トピックコンシューマーに送信できる最大のメッセージ数の指定	単一のバッチで、トピックコンシューマーに送信できる最大のメッセージ数を指定します。
5.5.12 imq.destination.DMQ.truncateBody	メッセージをデッドメッセージキューに保存する前のメッセージボディの消去の有効または無効の指定	メッセージをデッドメッセージキューに保存する前のメッセージボディの消去を、有効または無効にします。
5.5.13 imq.hostname	デフォルトのホスト名または IP アドレスの指定	すべての接続サービスの、デフォルトのホスト名または IP アドレスを指定します。
5.5.14 imq.persist.file.destination.message.filepool.limit	送信先ファイルプールが保持するファイルの最大数の指定	送信先ファイルプールが保持するファイルの最大数を指定します。
5.5.15 imq.persist.file.message.cleanup	ファイルプール内のファイルの消去の有効または無効の指定	シャットダウン時の、ファイルプール内のファイルの消去を有効または無効にします。
5.5.16 imq.persist.file.message.filepool.cleanratio	ファイルプールにクリーン（空）の状態 で保持するファイルの割合の指定	ファイルプールにクリーン（空）の状態 で保持するファイルの割合を指定します。
5.5.17 imq.persist.file.message.max_record_size	メッセージ退避ファイルに追加するメッセージの最大サイズの指定	メッセージ退避ファイルに追加するメッセージの最大サイズを指定します。
5.5.18 imq.persist.file.sync.enabled	メモリー内の状態を物理的なストレージデバイスとの同期の有効または無効の指定	メモリーの状態を物理的なストレージデバイスと同期させることを有効または無効にします。
5.5.19 imq.service.activelist	ブローカーの起動時に活性化する接続サービスのリストの指定	ブローカーの起動時に活性化する接続サービスのリストをコマンドで区切って指定します。
5.5.20 imq.system.max_count	ブローカーが保持するメッセージの最大数の指定	ブローカーが保持するメッセージの最大数を指定します。

プロパティ	概要	詳細説明
5.5.21 imq.system.max_size	ブローカーが保持するメッセージの最大合計サイズの指定	ブローカーが保持するメッセージの最大合計サイズを指定します。
5.5.22 imq.txn.reapInterval	リーパースレッドのリープまたはクリーンアップする周期の秒単位の指定	リーパースレッドが、コミットされたトランザクションをリープまたはクリーンアップするための周期を秒単位で指定します。
5.5.23 imq.txt.reapLimit	最後にコミットされたトランザクションの最大数の指定	コミットされたトランザクションのリープまたはクリーンアップ時に維持される、最後にコミットされたトランザクションの最大数を指定します。
5.5.24 java.util.logging.FileHandler.count	出力ファイル数の指定	出力ファイルの数を指定します。
5.5.25 java.util.logging.FileHandler.level	ハンドラーのデフォルトレベルの指定	ハンドラーのデフォルトレベルを指定します。
5.5.26 java.util.logging.FileHandler.limit	1つのログファイルに書き込める上限の指定	1つのログファイルに書き込める上限を指定します。

das.properties に設定するプロパティ

プロパティ	概要	詳細説明
5.7.1 agent.das.host	ドメイン管理サーバのホスト名または IP アドレスの指定	プロパティ設定対象のサーバインスタンスを管理する、ドメイン管理サーバのホスト名、または IP アドレス指定します。

5.2 asenv.bat

Java EE Server の環境変数定義ファイルです。

説明

Java EE Server 固有の環境変数を保存するファイルです。

このファイルで設定した環境変数は、OS の環境変数（システム環境変数およびユーザー環境変数）よりも優先的に設定されます。

なお、リモートホスト上にノードがある場合、ドメイン管理サーバがノードを操作するためにリモートホスト上で `asadmin` ユーティリティーコマンドを実行することがあります。このリモートホスト上の `asadmin` プロセスに環境変数を適用する場合は、リモートホスト上にある `asenv.bat` に環境変数を設定する必要があります。

書式

Windows バッチファイル

格納先

Java EE Server のインストールディレクトリー `¥glassfish¥config¥asenv.bat`

記述例

```
set HJES_ASADMIN_MESSAGE_ENABLED=true
set HJES_ASADMIN_MESSAGE_ROTATION_SIZE=8388608
set HJES_ASADMIN_MESSAGE_ROTATION_TIME=121530
set HJES_ASADMIN_MESSAGE_FILE_NUMBER=16
```

5.3 asenv.bat に設定する環境変数の詳細

asenv.bat に設定する環境変数について説明しています。

5.3.1 AS_ADMIN_READTIMEOUT

asadmin プロセスおよびドメイン管理サーバに設定する HTTP リードタイムアウト値を指定します。

説明

asadmin プロセスとドメイン管理サーバ間の通信、およびドメイン管理サーバとサーバインスタンス間の通信に適用する、HTTP リードタイムアウト値（単位：ミリ秒）を指定します。

HTTP リードタイムアウト値に0を指定した場合、コマンド処理時間が長期化すると、そのコマンド処理がasadmin ユーティリティーコマンドのサブコマンド、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。このため、HTTP リードタイムアウト値に0（タイムアウトの無効）を指定することは推奨しません。

指定できる値

型：Integer

0～2147483647 の整数値

0 を指定すると、HTTP リードタイムアウトは設定されません（0 は非推奨の値です）。

デフォルト値

定義項目の省略

900000

値の省略

900000

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
AS_ADMIN_READTIMEOUT=900000
```

5.3.2 HJES_ASADMIN_CONSOLE_ENABLED

asadmin ユーティリティーコマンドのプロセス起動時ログの有効または無効を指定します。

説明

asadmin ユーティリティーコマンドのプロセス起動時ログを出力するかどうかを指定します。

指定できる値

型 : boolean(ignore)

{true | false}

デフォルト値

定義項目の省略

true

値の省略

true

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_CONSOLE_ENABLED=true
```

5.3.3 HJES_ASADMIN_CONSOLE_FILE_NUMBER

asadmin ユーティリティーコマンドのプロセス起動時ログファイルの面数を指定します。

説明

asadmin ユーティリティーコマンドのプロセス起動時ログの Java EE Server 稼働中に保持するログファイルの個数を指定します。

指定できる値

型 : Integer

1～64 の整数値

デフォルト値

定義項目の省略

8

値の省略

8

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_CONSOLE_FILE_NUMBER=16
```

5.3.4 HJES_ASADMIN_CONSOLE_ROTATION_SIZE

asadmin ユーティリティーコマンドのプロセス起動時ログファイルのサイズ（単位：バイト）を指定します。

説明

asadmin ユーティリティーコマンドのプロセス起動時ログのログ出力先ファイルを切り替えるサイズ（単位：バイト）を指定します。

指定できる値

型：Integer

4096～16777216 の整数値

デフォルト値

定義項目の省略

16777216

値の省略

16777216

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_CONSOLE_ROTATION_SIZE=8388608
```

5.3.5 HJES_ASADMIN_CONSOLE_ROTATION_TIME

asadmin ユーティリティーコマンドのプロセス起動時ログファイルの切り替え時刻を指定します。

説明

asadmin ユーティリティーコマンドのプロセス起動時ログのログ出力先ファイルを切り替える時刻を "HHMMSS"形式で指定します。

指定できる値

型 : String

"HHMMSS"形式で 000000~235959 の範囲

デフォルト値

定義項目の省略

000000

値の省略

000000

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_CONSOLE_ROTATION_TIME=121530
```

5.3.6 HJES_ASADMIN_JVM_OPTIONS

asadmin ユーティリティーコマンドのプロセスに適用する JavaVM オプションを指定します。

説明

asadmin ユーティリティーコマンドのプロセスに適用する JavaVM オプションを指定します。複数の JavaVM オプションを指定する場合は、半角空白で区切って、オプション全体を””で囲んでください。

指定できる値

型：String

デフォルト値

定義項目の省略

なし

値の省略

なし

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

- JavaVM オプションを単独で指定する場合

```
HJES_ASADMIN_JVM_OPTIONS=-Xmx512m
```

- JavaVM オプションを複数指定する場合

```
HJES_ASADMIN_JVM_OPTIONS="-Xmx512m -XX:+HitachiThreadDumpToStdout"
```


5.3.7 HJES_ASADMIN_LOG_DIVISION

asadmin ユーティリティーコマンドのログ出力先ディレクトリーを分割するかどうかを指定します。

説明

asadmin ユーティリティーコマンドのログを、OS のユーザーアカウントごとにディレクトリーを分けて出力するかどうかを指定します。

指定できる値

型：boolean(ignore)

{true | false}

デフォルト値

定義項目の省略

false

値の省略

false

上書き許可

上書きできます。

出現パターン

0 回、または 1 回

記述例

```
HJES_ASADMIN_LOG_DIVISION=true
```

5.3.8 HJES_ASADMIN_MAINTENANCE_ENABLED

asadmin ユーティリティーコマンドの保守ログの有効または無効を指定します。

説明

asadmin ユーティリティーコマンドの保守ログを出力するかどうかを指定します。

指定できる値

型：boolean(ignore)

{true | false}

デフォルト値

定義項目の省略

true

値の省略

true

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MAINTENANCE_ENABLED=true
```

5.3.9 HJES_ASADMIN_MAINTENANCE_FILE_NUMBER

asadmin ユーティリティーコマンドの保守ログファイルの面数を指定します。

説明

asadmin ユーティリティーコマンドの保守ログの Java EE Server 稼働中に保持するログファイルの個数を指定します。

指定できる値

型: Integer

1~64 の整数値

デフォルト値

定義項目の省略

8

値の省略

8

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MAINTENANCE_FILE_NUMBER=16
```

5.3.10 HJES_ASADMIN_MAINTENANCE_LOG_LEVEL

asadmin ユーティリティーコマンドの保守ログのレベルを指定します。

説明

asadmin ユーティリティーコマンドの保守ログのログに出力される内容のレベルを指定します。

指定できる値

型：Integer

1～2

デフォルト値

定義項目の省略

1

値の省略

1

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MAINTENANCE_LOG_LEVEL=2
```

5.3.11 HJES_ASADMIN_MAINTENANCE_ROTATION_SIZE

asadmin ユーティリティーコマンドの保守ログファイルのサイズ（単位：バイト）を指定します。

説明

asadmin ユーティリティーコマンドの保守ログのログ出力先ファイルを切り替えるサイズ（単位：バイト）を指定します。

指定できる値

型：Integer

4096～16777216 の整数値

デフォルト値

定義項目の省略

16777216

値の省略

16777216

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MAINTENANCE_ROTATION_SIZE=8388608
```

5.3.12 HJES_ASADMIN_MAINTENANCE_ROTATION_TIME

asadmin ユーティリティーコマンドの保守ログファイルの切り替え時刻を指定します。

説明

asadmin ユーティリティーコマンドの保守ログのログ出力先ファイルを切り替える時刻を"HHMMSS"形式で指定します。

指定できる値

型：String

"HHMMSS"形式で 000000~235959 の範囲

デフォルト値

定義項目の省略

000000

値の省略

000000

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MAINTENANCE_ROTATION_TIME=121530
```

5.3.13 HJES_ASADMIN_MESSAGE_ENABLED

asadmin ユーティリティーコマンドのメッセージログの有効または無効を指定します。

説明

asadmin ユーティリティーコマンドのメッセージログを出力するかどうかを指定します。

指定できる値

型：boolean(ignore)

{true | false}

デフォルト値

定義項目の省略

true

値の省略

true

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MESSAGE_ENABLED=true
```

5.3.14 HJES_ASADMIN_MESSAGE_FILE_NUMBER

asadmin ユーティリティーコマンドのメッセージログファイルの面数を指定します。

説明

asadmin ユーティリティーコマンドのメッセージログの Java EE Server 稼働中に保持するログファイルの個数を指定します。

指定できる値

型：Integer

1～64 の整数値

デフォルト値

定義項目の省略

8

値の省略

8

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MESSAGE_FILE_NUMBER=16
```

5.3.15 HJES_ASADMIN_MESSAGE_LOG_LEVEL

asadmin ユーティリティーコマンドのメッセージログのレベルを指定します。

説明

asadmin ユーティリティーコマンドのメッセージログのログに出力される内容のレベルを指定します。

指定できる値

型：Integer

1～3

デフォルト値

定義項目の省略

2

値の省略

2

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MESSAGE_LOG_LEVEL=3
```

5.3.16 HJES_ASADMIN_MESSAGE_ROTATION_SIZE

asadmin ユーティリティーコマンドのメッセージログファイルのサイズ（単位：バイト）を指定します。

説明

asadmin ユーティリティーコマンドのメッセージログのログ出力先ファイルを切り替えるサイズ（単位：バイト）を指定します。

指定できる値

型：Integer

4096～16777216 の整数値

デフォルト値

定義項目の省略

16777216

値の省略

16777216

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MESSAGE_ROTATION_SIZE=8388608
```

5.3.17 HJES_ASADMIN_MESSAGE_ROTATION_TIME

asadmin ユーティリティーコマンドのメッセージログファイルの切り替え時刻を指定します。

説明

asadmin ユーティリティーコマンドのメッセージログのログ出力先ファイルを切り替える時刻を"HHMMSS"形式で指定します。

指定できる値

型：String

"HHMMSS"形式で 000000～235959 の範囲

デフォルト値

定義項目の省略

000000

値の省略

000000

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_MESSAGE_ROTATION_TIME=121530
```

5.3.18 HJES_ASADMIN_STACKTRACE_ENABLED

asadmin ユーティリティーコマンドのスタックトレースログの有効または無効を指定します。

説明

asadmin ユーティリティーコマンドのスタックトレースログを出力するかどうかを指定します。

指定できる値

型：boolean(ignore)

{true | false}

デフォルト値

定義項目の省略

true

値の省略

true

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_STACKTRACE_ENABLED=true
```

5.3.19 HJES_ASADMIN_STACKTRACE_FILE_NUMBER

asadmin ユーティリティーコマンドのスタックトレースログファイルの面数を指定します。

説明

asadmin ユーティリティーコマンドのスタックトレースログの Java EE Server 稼働中に保持するログファイルの個数を指定します。

指定できる値

型：Integer

1～64 の整数値

デフォルト値

定義項目の省略

8

値の省略

8

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_STACKTRACE_FILE_NUMBER=16
```

5.3.20 HJES_ASADMIN_STACKTRACE_ROTATION_SIZE

asadmin ユーティリティーコマンドのスタックトレースログファイルのサイズ（単位：バイト）を指定します。

説明

asadmin ユーティリティーコマンドのスタックトレースログのログ出力先ファイルを切り替えるサイズ（単位：バイト）を指定します。

指定できる値

型：Integer

4096～16777216 の整数値

デフォルト値

定義項目の省略

16777216

値の省略

16777216

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_STACKTRACE_ROTATION_SIZE=8388608
```

5.3.21 HJES_ASADMIN_STACKTRACE_ROTATION_TIME

asadmin ユーティリティーコマンドのスタックトレースログファイルの切り替え時刻を指定します。

説明

asadmin ユーティリティーコマンドのスタックトレースログのログ出力先ファイルを切り替える時刻を"HHMMSS"形式で指定します。

指定できる値

型：String

"HHMMSS"形式で 000000～235959 の範囲

デフォルト値

定義項目の省略

000000

値の省略

000000

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_ASADMIN_STACKTRACE_ROTATION_TIME=121530
```

5.3.22 HJES_LOGSDIR

Java EE Server のログファイルの出力先ディレクトリーを指定します。

説明

Java EE Server のログファイルの出力先ディレクトリーを変更する場合に指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

*Java EE Server*のインストールディレクトリー/logs

値の省略

*Java EE Server*のインストールディレクトリー/logs

上書き許可

上書きできます。

出現パターン

0回、または1回

記述例

```
HJES_LOGSDIR=/home/user/logs
```

5.4 config.properties

config.properties ファイルは、ブローカーの構成に関するプロパティ設定ファイルです。

説明

config.properties ファイルは、ブローカーの構成に関するプロパティ設定ファイルです。

ブローカーは、インスタンス設定ファイルを起動時にだけ読み込みます。

ブローカーの設定への変更を適用するには、ブローカーをシャットダウンしてから再起動する必要があります。

config.properties ファイル（または任意の設定ファイル）のプロパティ定義で使用される構文は次のとおりです。

```
propertyName=value [ [,value1] ... ]
```

格納先

プロパティ設定ファイルは次の場所に格納されます。

IMQ_VARHOME/instances/*instanceName*/props/config.properties

IMQ_VARHOME

環境変数です。

記述例

```
imq.system.max_count=50000
```

5.5 config.properties に設定するプロパティの詳細

config.properties に設定するプロパティについて説明しています。

5.5.1 imq.authentication.client.response.timeout

認証リクエストに対するクライアントからの応答を待機する時間を秒単位で指定します。

説明

認証リクエストに対するクライアントからの応答を待機する時間を秒単位で指定します。この値は `imq.authentication.client.response.timeout` プロパティで定義されています。

指定できる値

型：Integer

0~2147483647

デフォルト値

定義項目の省略

180

5.5.2 imq.autocreate.destination.limitBehavior

メモリー制限のしきい値に達した場合のブローカー動作を指定します。

説明

メモリー制限のしきい値に達した場合のブローカー動作を指定します。

有効な値は次のとおりです。

FLOW_CONTROL

プロデューサーの速度を落とします。

REMOVE_OLDEST

最も古いメッセージを消去します。

REMOVE_LOW_PRIORITY

最も優先度の低いメッセージをその有効期間に従って消去します。メッセージの作成・送信を行っているクライアントには通知されません。

REJECT_NEWEST

最も新しいメッセージを拒否します。メッセージが持続する場合だけ、例外としてメッセージの作成・送信を行っているクライアントに通知します。

FLOW_CONTROL が指定されても、メッセージ数は `imq.system.max_count` を超過することがあります。この場合ブローカーは、送信先が満杯で新しいメッセージが拒否されることを示すために、`ResourceAllocationException` を生成します。

値が `REMOVE_OLDEST` または `REMOVE_LOW_PRIORITY` であり、さらに `imq.autocreate.destination.useDMQ` プロパティーが `true` の場合、超過し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。

指定できる値

型: String

{FLOW_CONTROL | REMOVE_OLDEST | REMOVE_LOW_PRIORITY | REJECT_NEWEST}

デフォルト値

定義項目の省略

REJECT_NEWEST

5.5.3 imq.autocreate.destination.maxBytesPerMsg

単一メッセージの最大サイズをバイト単位で指定します。

説明

単一メッセージの最大サイズをバイト単位で指定します。

値は、次の接尾辞を使用して、バイト単位、キロバイト単位、メガバイト単位で表現できます。

b

バイト

k

キロバイト (1,024 バイト)

m

メガバイト (1,024 × 1,024 = 1,048,576 バイト)

接尾辞のない値はバイト単位になります。値「0」と「-1」は、メッセージサイズが無制限であることを示します。

指定できる値

型：String

デフォルト値

定義項目の省略

10k

記述例

```
imq.autocreate.destination.maxBytesPerMsg=<例>
```

例：

- 1600：1600バイト
- 1600b：1600バイト
- 16k：16キロバイト (= 16,384 バイト)
- 16m：16メガバイト (= 16,777,216 バイト)
- -1：無制限

5.5.4 imq.autocreate.destination.maxNumMsgs

未配信メッセージの最大数を指定します。

説明

未配信メッセージの最大数を指定します。

指定できる値

型：Integer

1～2147483647

デフォルト値

定義項目の省略

100000

注意事項

フロー制御が有効な場合 (imq.autocreate.destination.limitBehavior = FLOW_CONTROL)、ブローカーがメッセージ受信のフローを停止するのに間に合わないで、指定されたメッセージ数の制限を超過する場合があります。その場合、imq.autocreate.destination.maxNumMsgs に指定した値は、厳しく制限するものではなく、単なるヒントとして使われます。

しかし、未配信メッセージ数が `imq.system.max_count` を超えた場合、ブローカーは、送信先が満杯で新しいメッセージが拒否されることを示すために、`ResourceAllocationException` を生成します。

5.5.5 `imq.autocreate.destination.maxTotalMsgBytes`

未配信メッセージの最大の総メモリー量をバイト単位で指定します。

説明

未配信メッセージの最大の総メモリー量をバイト単位で指定します。

構文は `imq.autocreate.destination.maxBytesPerMsg` に同じです。値は、次の接尾辞を使用して、バイト単位、キロバイト単位、メガバイト単位で表現できます。

b

バイト

k

キロバイト (1,024 バイト)

m

メガバイト (1,024 × 1,024 = 1,048,576 バイト)

接尾辞のない値はバイト単位になります。値「0」と「-1」は、メッセージサイズが無制限であることを示します。

指定できる値

型: String

デフォルト値

定義項目の省略

10m

記述例

```
imq.autocreate.destination.maxTotalMsgBytes=<例>
```

例:

- 1600 : 1600バイト
- 1600b : 1600バイト
- 16k : 16キロバイト (= 16,384 バイト)
- 16m : 16メガバイト (= 16,777,216 バイト)
- -1 : 無制限

5.5.6 imq.autocreate.destination.useDMQ

デッドメッセージキューへのデッドメッセージの送信を有効または無効にします。

説明

デッドメッセージキューへのデッドメッセージの送信を有効または無効にします。false を指定した場合、デッドメッセージは廃棄されます。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

true

5.5.7 imq.autocreate.queue

キュー送信先の自動作成を有効または無効にします。

説明

キュー送信先の自動作成を有効または無効にします。

true を指定した場合、メッセージコンシューマーまたはメッセージプロデューサーが存在しない送信先にアクセスしようとする、ブローカーは自動的に物理送信先を作成します。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

true

注意事項

キュー送信先だけに使用できます。

5.5.8 imq.autocreate.queue.maxNumActiveConsumers

キュー送信先からのロードバランス配信での、アクティブなメッセージコンシューマーの最大数を指定します。

説明

キュー送信先からのロードバランス配信での、アクティブなメッセージコンシューマーの最大数を指定します。

値「0」と「-1」は、コンシューマー数が無制限であることを示します。

指定できる値

型：Integer

-1～2147483647

デフォルト値

定義項目の省略

-1

注意事項

キュー送信先だけに使用できます。

5.5.9 imq.autocreate.queue.maxNumBackupConsumers

キュー送信先からのロードバランス配信での、バックアップメッセージコンシューマーの最大数を指定します。

説明

キュー送信先からのロードバランス配信での、バックアップメッセージコンシューマーの最大数を指定します。

値「-1」は、コンシューマー数が無制限であることを示します。

指定できる値

型：Integer

-1～2147483647

デフォルト値

定義項目の省略

0

注意事項

キュー送信先だけに使用できます。

5.5.10 imq.autocreate.topic

トピック送信先の自動作成を有効または無効にします。

説明

トピック送信先の自動作成を有効または無効にします。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

true

注意事項

トピック送信先だけに使用できます。

5.5.11 imq.autocreate.topic.consumerFlowLimit

単一のバッチで、トピックコンシューマーに送信できる最大のメッセージ数を指定します。

説明

単一のバッチで、トピックコンシューマーに送信できる最大のメッセージ数を指定します。

値「0」または「-1」は、メッセージ数が無制限であることを示します。

指定できる値

型：Integer

-1 ~ 2147483647

デフォルト値

定義項目の省略

1000

注意事項

Java EE Server クラスタで、JMS リソースアダプター (jmsra) がメッセージを消費するために使われている場合、このシステムプロパティは使用できません。

トピック送信先だけに使用できます。

5.5.12 imq.destination.DMQ.truncateBody

メッセージをデッドメッセージキューに保存する前のメッセージボディーの消去を、有効または無効にします。

説明

メッセージをデッドメッセージキューに保存する前のメッセージボディーの消去を、有効または無効にします。

指定された値が true の場合、メッセージヘッダーとプロパティデータだけが保存されます。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

false

5.5.13 imq.hostname

すべての接続サービスの、デフォルトのホスト名または IP アドレスを指定します。

説明

すべての接続サービスの、デフォルトのホスト名または IP アドレスを指定します。

指定できる値

型：String

デフォルト値

定義項目の省略

すべての使用可能な IP アドレス

5.5.14 imq.persist.file.destination.message.filepool.limit

送信先ファイルプールが保持するファイルの最大数を指定します。

説明

送信先ファイルプールが保持するファイルの最大数を指定します。

この制限を超えるファイルは削除されます。ブローカーは、この制限に基づき追加するファイルを作成したり、ファイルを削除したりします。

制限値が高いほど、ブローカーは持続データを速く処理できます。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

100

5.5.15 imq.persist.file.message.cleanup

シャットダウン時の、ファイルプール内のファイルの消去を有効または無効にします。

説明

シャットダウン時の、ファイルプール内のファイルの消去を有効または無効にします。

このプロパティの値に `true` を指定すると、ファイルストア用のディスクスペースを節約できます。ただし、ブローカーのシャットダウンが遅くなります。

指定できる値

型：Boolean

{`true` | `false`}

デフォルト値

定義項目の省略

`false`

5.5.16 imq.persist.file.message.filepool.cleanratio

ファイルプールにクリーン（空） の状態で保持するファイルの割合を指定します。

説明

ファイルプールにクリーン（空） の状態で保持するファイルの割合を指定します。

この値が高いほど、ファイルプールに必要なディスクスペースが少なくなりますが、操作時にファイルを消去するために多くのオーバーヘッドが必要になります。

指定できる値

型：Integer

0~100

デフォルト値

定義項目の省略

`0`

5.5.17 imq.persist.file.message.max_record_size

メッセージ退避ファイルに追加するメッセージの最大サイズを指定します。

説明

メッセージ退避ファイルに追加するメッセージの最大サイズを指定します。

このサイズを超えるメッセージは、別の専用ファイルに格納されます。

値は、次の接尾辞を使用して、バイト単位、キロバイト単位、メガバイト単位で表現できます。

b

バイト

k

キロバイト (1,024 バイト)

m

メガバイト (1,024 × 1,024 = 1,048,576 バイト)

接尾辞のない値はバイト単位になります。

指定できる値

型 : String

デフォルト値

定義項目の省略

1m

記述例

```
imq.persist.file.message.max_record_size=<例>
```

例 :

- 1600 : 1600バイト
- 1600b : 1600バイト
- 16k : 16キロバイト (= 16,384 バイト)
- 16m : 16メガバイト (= 16,777,216 バイト)

5.5.18 imq.persist.file.sync.enabled

メモリー内の状態を物理的なストレージデバイスと同期させることを有効または無効にします。

説明

メモリーの状態を物理的なストレージデバイスと同期させることを有効または無効にします。

このプロパティーの値に `true` に指定すると、システムダウンによるデータの損失はなくなりますが、パフォーマンスが下がります。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

false

5.5.19 imq.service.activelist

ブローカーの起動時に活性化する接続サービスのリストをコンマで区切って指定します。

説明

ブローカーの起動時に活性化する接続サービスのリストをコンマで区切って指定します。

利用可能な接続サービスは次のとおりです。

- jms
- ssljms
- httpjms
- httpsjms
- admin
- ssladmin

ブローカーの `imq.service.activelist` プロパティを設定すると、これらの接続サービスの一部または全部を実行するように設定できます。このプロパティの値は、ブローカーが起動するときに活性化する接続サービスのリストです。このプロパティを明示的に指定しない場合、デフォルトでは、jms と admin サービスが活性化されます。

指定できる値

型：String

[[jms] | [ssljms] | [httpjms] | [httpsjms] | [admin] | [ssladmin]]

デフォルト値

定義項目の省略

jms, admin

5.5.20 imq.system.max_count

ブローカーが保持するメッセージの最大数を指定します。

説明

ブローカーが保持するメッセージの最大数を指定します。

値「0」と「-1」は、メッセージ数が無制限であることを示します。

指定できる値

型：Integer

-1～2147483647

デフォルト値

定義項目の省略

-1

5.5.21 imq.system.max_size

ブローカーが保持するメッセージの最大合計サイズを指定します。

説明

ブローカーが保持するメッセージの最大合計サイズを指定します。

値は、次の接尾辞を使用して、バイト単位、キロバイト単位、メガバイト単位で表現できます。

b

バイト

k

キロバイト (1,024 バイト)

m

メガバイト (1,024 × 1,024 = 1,048,576 バイト)

値「0」と「-1」は、メッセージ数が無制限であることを示します。

指定できる値

型：String

デフォルト値

定義項目の省略

-1

5.5.22 imq.txn.reapInterval

リーパースレッドが、コミットされたトランザクションをリープまたはクリーンアップする周期を秒単位で指定します。

説明

リーパースレッドが、コミットされたトランザクションをリープまたはクリーンアップするための周期を秒単位で指定します。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

900

5.5.23 imq.txt.reapLimit

コミットされたトランザクションのリープまたはクリーンアップ時に維持される、最後にコミットされたトランザクションの最大数を指定します。

説明

コミットされたトランザクションのリープまたはクリーンアップ時に維持される、最後にコミットされたトランザクションの最大数を指定します。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

0

5.5.24 java.util.logging.FileHandler.count

出力ファイルの数を指定します。

説明

出力ファイルの数を指定します。

1つのファイルが所定の上限サイズに達すると、そのファイルは閉じられ、新しいファイルが開きます。

古いファイルには新たに名称がつけられます。その際、元のファイル名に「0」、「1」、「2」などといった値が付け加えられます。デフォルトでは、IOライブラリーでバッファリングが有効になっていますが、それぞれのログレコードは完了時にフラッシュされます。

指定できる値

型：Integer

1～2147483647

デフォルト値

定義項目の省略

8

5.5.25 java.util.logging.FileHandler.level

ハンドラーのデフォルトレベルを指定します。

説明

ハンドラーのデフォルトレベルを指定します。

ロギング Level オブジェクトは順序づけされており、順序づけされた整数を使って指定できます。所定のレベルでロギングを有効にすると、すべての上位のレベルでもロギングが有効になります。

値をOFFに指定すると、ロギングを無効にできます。

指定できる値

型：String

{SEVERE(最高値) | WARNING | INFO | CONFIG | FINE | FINER | FINEST | ALL | OFF}

デフォルト値

定義項目の省略

ALL

5.5.26 java.util.logging.FileHandler.limit

1つのログファイルに書き込める上限を指定します（単位はバイト）。

説明

1つのログファイルに書き込める上限を指定します（単位はバイト）。この値を0にすると、無制限になります。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

16777216

5.6 das.properties

ノードがドメイン管理サーバに接続するための接続情報を含むファイルです。

説明

ノードがドメイン管理サーバに接続するための接続情報を含むファイルです。

格納先

プロパティは次の場所に格納されます。

*Application Server*のインストールディレクトリー/`javaee/glassfish/nodes/node-name/agent/config/`
`das.properties`

5.7 das.properties に設定するプロパティの詳細

das.properties に設定するプロパティについて説明しています。

5.7.1 agent.das.host

ドメイン管理サーバのホスト名、または IP アドレスを指定します。

説明

プロパティ設定対象のサーバインスタンスを管理する、ドメイン管理サーバのホスト名、または IP アドレス指定します。

指定できる値

型 : String

ドメイン管理サーバのホスト名、または IP アドレス

初期値

localhost

デフォルト値

定義項目の省略

localhost

6

Java EE 標準 API で設定できる Java EE Server 独自のプロパティ

Java EE 標準 API で設定できる Java EE Server 独自のプロパティを説明しています。

6.1 Java EE 標準 API で設定できる Java EE Server 独自のプロパティの一覧

Java EE 標準 API で設定できる Java EE Server 独自のプロパティの概要について説明しています。

JavaMail で SMTP サーバに接続するためのプロパティ

プロパティ	概要	詳細説明
6.2.1 mail.smtp.allow8bitmime	サーバ側での 8BITMIME 拡張サポートの指定	このプロパティに <code>true</code> を指定し、サーバが 8BITMIME 拡張をサポートしている場合に、 <code>quoted-printable</code> エンコーディング、または <code>Base64</code> エンコーディングを使用するメッセージのテキスト部分は、8-bit エンコーディングへ変換されます。
6.2.2 mail.smtp.auth	AUTH コマンドを使用したユーザー認証の指定	<code>mail.smtp.auth</code> に <code>true</code> を指定すると、AUTH コマンドを使用してユーザー認証をします。
6.2.3 mail.smtp.connectiontimeout	ソケットのコネクションタイムアウト値の指定	<code>mail.smtp.connectiontimeout</code> では、ソケットのコネクションタイムアウト値をミリ秒単位で指定します。
6.2.4 mail.smtp.ehlo	EHLO コマンドによるサインオンの指定	<code>mail.smtp.ehlo</code> に <code>false</code> を指定すると、EHLO コマンドによるサインオンを行いません。
6.2.5 mail.smtp.localhost	SMTP HELO コマンドまたは EHLO コマンドで使用するローカルホストの名前の指定	<code>mail.smtp.localhost</code> では、SMTP HELO コマンドまたは EHLO コマンドで使用する、ローカルホストの名前を指定します。
6.2.6 mail.smtp.sendpartial	有効なアドレスへのメール送信の指定	<code>true</code> を指定した場合、メッセージに有効なアドレスと無効なアドレスが混在しているときは、メッセージが送信され、 <code>SendFailedException</code> 例外で部分障害が通知されます。
6.2.7 mail.smtp.ssl.protocols	SSL 接続で使用できる SSL プロトコルの指定	<code>mail.smtp.ssl.protocols</code> では、SSL 接続で使用できる SSL プロトコルを指定します。
6.2.8 mail.smtp.timeout	ソケットの読み込みタイムアウト値の指定	<code>mail.smtp.timeout</code> では、ソケットの読み込みタイムアウト値をミリ秒単位で指定します。
6.2.9 mail.smtp.writetimeout	ソケットの書き込みタイムアウト値の指定	<code>mail.smtp.writetimeout</code> では、ソケットの書き込みタイムアウト値をミリ秒単位で指定します。

JavaMail で IMAP サーバに接続するためのプロパティ

プロパティ	概要	詳細説明
6.3.1 mail.imap.connectiontimeout	ソケットのコネクションタイムアウト値の指定	<code>mail.imap.connectiontimeout</code> では、ソケットのコネクションタイムアウト値をミリ秒単位で指定します。
6.3.2 mail.imap.fetchsize	部分フェッチサイズの指定	<code>mail.imap.fetchsize</code> では、部分フェッチサイズをバイト単位で指定します。

プロパティ	概要	詳細説明
6.3.3 mail.imap.peek	ボディー部分のフェッチ時にメッセージへの SEEN フラグ設定の防止	mail.imap.peek に true を指定すると、ボディー部分をフェッチするときに IMAP PEEK オプションを使用して、メッセージに SEEN フラグが設定されるのを防ぎます。
6.3.4 mail.imap.ssl.protocols	SSL 接続で使用できる SSL プロトコルの指定	mail.imap.ssl.protocols では、SSL 接続で使用できる SSL プロトコルを指定します。
6.3.5 mail.imap.statuscachetimeout	STATUS コマンドの応答をキャッシュするタイムアウト値の指定	mail.imap.statuscachetimeout では、STATUS コマンドの応答をキャッシュするタイムアウト値をミリ秒単位で指定します。
6.3.6 mail.imap.timeout	ソケットの読み込みタイムアウト値の指定	mail.imap.timeout では、ソケットの読み込みタイムアウト値をミリ秒単位で指定します。
6.3.7 mail.imap.writetimeout	ソケットの書き込みタイムアウト値の指定	mail.imap.writetimeout では、ソケットの書き込みタイムアウト値をミリ秒単位で指定します。

6.2 JavaMail で SMTP サーバに接続するためのプロパティの詳細

JavaMail で SMTP サーバに接続するためのプロパティを説明しています。

6.2.1 mail.smtp.allow8bitmime

mail.smtp.allow8bitmime に true を指定すると、サーバでは 8BITMIME 拡張 (8 ビット転送) がサポートされます。

説明

このプロパティに true を指定し、サーバが 8BITMIME 拡張をサポートしている場合に、quoted-printable エンコーディング、または Base64 エンコーディングを使用するメッセージのテキスト部分は、8-bit エンコーディングへ変換されます。ただし、変換対象のメッセージのテキスト部分は、8-bit テキストに関する RFC 2045 の規定に準拠している必要があります。デフォルト値は false です。

指定できる値

型 : Boolean

{true | false}

デフォルト値

定義項目の省略

false

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.allow8bitmime", "true");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.2 mail.smtp.auth

mail.smtp.auth に true を指定すると、AUTH コマンドを使用してユーザー認証をします。デフォルト値は false です。

説明

mail.smtp.auth に true を指定すると、AUTH コマンドを使用してユーザー認証をします。デフォルト値は false です。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

false

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.auth", "true");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.3 mail.smtp.connectiontimeout

mail.smtp.connectiontimeout では、ソケットのコネクションタイムアウト値をミリ秒単位で指定します。

説明

mail.smtp.connectiontimeout では、ソケットのコネクションタイムアウト値をミリ秒単位で指定します。このタイムアウト値は、java.net.Socket クラスで実装されます。デフォルト値は-1 です。タイムアウト値に-1 もしくは0 を指定、またはこのプロパティを省略した場合、タイムアウトは設定されません。

指定できる値

型：Integer

-1 ~ 2147483647

デフォルト値

定義項目の省略

-1 (タイムアウトは設定されません)

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.connectiontimeout", "30");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.4 mail.smtp.ehlo

mail.smtp.ehlo に false を指定すると、EHLO コマンドによるサインオンを行いません。

説明

mail.smtp.ehlo に false を指定すると、EHLO コマンドによるサインオンを行いません。デフォルト値は true です。通常、EHLO コマンドでサインオンに失敗した場合、代替に HELO コマンドによってサインオンを行います。このプロパティは、EHLO コマンドをまったく失敗しないサーバ、または EHLO コマンドが正しく実装されないサーバに対してだけ使用します。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

true

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.ehlo", "false");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.5 mail.smtp.localhost

mail.smtp.localhost では、SMTP HELO コマンドまたは EHLO コマンドで使用する、ローカルホストの名前を指定します。

説明

mail.smtp.localhost では、SMTP HELO コマンドまたはEHLO コマンドで使用する、ローカルホストの名前を指定します。デフォルト値は、InetAddress.getLocalHost().getCanonicalHostName()メソッドの戻り値です。JDK およびネーミングサービスが正しく設定されている場合、ローカルホストの名前を指定する必要はありません。

指定できる値

型：String

デフォルト値

定義項目の省略

java.net.InetAddress.getLocalHost().getCanonicalHostName()メソッドの戻り値

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.localhost", "localhost");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.6 mail.smtp.sendpartial

mail.smtp.sendpartial にtrue を指定した場合、メッセージに無効なアドレスが含まれているときは、有効なアドレスにだけメールが送信されます。このとき、すべての無効なアドレスには、部分障害として SendFailedException 例外が通知されます。false を指定した場合は、どのアドレスにもメッセージは送信されません。

説明

true を指定した場合、メッセージに有効なアドレスと無効なアドレスが混在しているときは、メッセージが送信され、SendFailedException 例外で部分障害が通知されます。false を指定した場合、メッセージに無効なアドレスが存在するときは、どの受信者（アドレス）にもメッセージは送信されません。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

false

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.sendpartial", "true");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.7 mail.smtp.ssl.protocols

mail.smtp.ssl.protocols では、SSL 接続で使用できる SSL プロトコルを指定します。

説明

mail.smtp.ssl.protocols では、SSL 接続で使用できる SSL プロトコルを指定します。プロパティ値には、javax.net.ssl.SSLSocket.setEnabledProtocols メソッドに設定できるトークンの一覧を空白で区切って指定します。

指定できる値

型 : String

指定できる値については、JDK のドキュメントを参照してください。

デフォルト値

定義項目の省略

TLSv1.1 および TLSv1.2

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.ssl.protocols", "TLSv1 TLSv1.1");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```


6.2.8 mail.smtp.timeout

mail.smtp.timeout では、ソケットの読み込みタイムアウト値をミリ秒単位で指定します。

説明

mail.smtp.timeout では、ソケットの読み込みタイムアウト値をミリ秒単位で指定します。このタイムアウト値は、java.net.Socket クラスで実装されます。デフォルト値は-1 です。タイムアウト値に-1 もしくは0 を指定、またはこのプロパティを省略した場合、タイムアウトは設定されません。

指定できる値

型：Integer

-1 ~ 2147483647

デフォルト値

定義項目の省略

-1 (タイムアウトは設定されません)

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.timeout", "30");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.2.9 mail.smtp.writetimeout

mail.smtp.writetimeout では、ソケットの書き込みタイムアウト値をミリ秒単位で指定します。

説明

mail.smtp.writetimeout では、ソケットの書き込みタイムアウト値をミリ秒単位で指定します。このタイムアウト値は、接続ごとに java.util.concurrent.ScheduledExecutorService クラスで実装されます。java.util.concurrent.ScheduledExecutorService クラスでは、タイムアウト値を超えた場合にソケットを閉じるためのスレッドをスケジュールします。mail.smtp.writetimeout 使用時のオーバーヘッドは、1 接続あたり 1 スレッドです。デフォルト値は-1 です。タイムアウト値に-1 もしくは0 を指定、またはこのプロパティを省略した場合、タイムアウトは設定されません。

指定できる値

型：Integer

-1～2147483647

デフォルト値

定義項目の省略

-1 (タイムアウトは設定されません)

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.smtp.writetimeout", "30");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.3 JavaMail で IMAP サーバに接続するためのプロパティの詳細

JavaMail で IMAP サーバに接続するためのプロパティを説明しています。

6.3.1 mail.imap.connectiontimeout

mail.imap.connectiontimeout では、ソケットの接続タイムアウト値をミリ秒単位で指定します。

説明

mail.imap.connectiontimeout では、ソケットの接続タイムアウト値をミリ秒単位で指定します。このタイムアウト値は、java.net.Socket クラスで実装されます。デフォルト値は-1 です。タイムアウト値に-1 もしくは0 を指定、またはこのプロパティを省略した場合、タイムアウトは設定されません。

指定できる値

型：Integer

-1 ~ 2147483647

デフォルト値

定義項目の省略

-1 (タイムアウトは設定されません)

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.connectiontimeout", "30");
Session session = Session.getDefaultInstance(properties);
Store store = session.getStore("imap");
store.connect("user", "password");
messages = folder.getMessages();
```

6.3.2 mail.imap.fetchsize

mail.imap.fetchsize では、部分フェッチサイズをバイト単位で指定します。デフォルト値は 16 キロバイトです。

説明

mail.imap.fetchsize では、部分フェッチサイズをバイト単位で指定します。デフォルト値は 16 キロバイトです。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

16384

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.fetchsize", "1000");
Session session = Session.getDefaultInstance(properties);
Store store = session.getStore("imap");
store.connect("user", "password");
messages = folder.getMessages();
```

6.3.3 mail.imap.peek

mail.imap.peek にtrue を指定すると、ボディー部分をフェッチするときにIMAP PEEK オプションを使用して、メッセージに SEEN フラグが設定されるのを防ぎます。

説明

mail.imap.peek にtrue を指定すると、ボディー部分をフェッチするときにIMAP PEEK オプションを使用して、メッセージに SEEN フラグが設定されるのを防ぎます。デフォルト値はfalse です。このオプションは、IMAPMessage クラスの setPeek メソッドを使用して、メッセージごとにオーバーライドできます。

指定できる値

型：Boolean

{true | false}

デフォルト値

定義項目の省略

false

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.peek", "true");
```

```
Session session = Session.getDefaultInstance(properties);
Store store = session.getStore("imap");
store.connect("user", "password");
messages = folder.getMessages();
```

6.3.4 mail.imap.ssl.protocols

mail.imap.ssl.protocols では、SSL 接続で使用できる SSL プロトコルを指定します。

説明

mail.imap.ssl.protocols では、SSL 接続で使用できる SSL プロトコルを指定します。プロパティ値には、javax.net.ssl.SSLSocket.setEnabledProtocols メソッドに設定できるトークンの一覧を空白で区切って指定します。

指定できる値

型：String

指定できる値については、JDK のドキュメントを参照してください。

デフォルト値

定義項目の省略

TLsv1.1 およびTLsv1.2

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.ssl.protocols", "TLsv1 TLsv1.1");
Session session = Session.getDefaultInstance(properties);
Message message = new MimeMessage(session);
message.setSubject(sentDate);
Transport.send(message);
```

6.3.5 mail.imap.statuscachetimeout

mail.imap.statuscachetimeout では、STATUS コマンドの応答をキャッシュするタイムアウト値をミリ秒単位で指定します。

説明

mail.imap.statuscachetimeout では、STATUS コマンドの応答をキャッシュするタイムアウト値をミリ秒単位で指定します。デフォルト値は1000（1 秒）です。この値が0の場合は、キャッシュしません。

指定できる値

型：Integer

0～2147483647

デフォルト値

定義項目の省略

1000

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.statuscachetimeout", "30");
Session session = Session.getDefaultInstance(properties);
Store store = session.getStore("imap");
store.connect("user", "password");
messages = folder.getMessages();
```

6.3.6 mail.imap.timeout

mail.imap.timeout では、ソケットの読み込みタイムアウト値をミリ秒単位で指定します。

説明

mail.imap.timeout では、ソケットの読み込みタイムアウト値をミリ秒単位で指定します。このタイムアウト値は、java.net.Socket クラスで実装されます。デフォルト値は-1 です。タイムアウト値に-1 もしくは0 を指定、またはこのプロパティを省略した場合、タイムアウトは設定されません。

指定できる値

型：Integer

-1～2147483647

デフォルト値

定義項目の省略

-1 (タイムアウトは設定されません)

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.timeout", "30");
Session session = Session.getDefaultInstance(properties);
```

```
Store store = session.getStore("imap");
store.connect("user", "password");
messages = folder.getMessages();
```

6.3.7 mail.imap.writetimeout

mail.imap.writetimeout では、ソケットの書き込みタイムアウト値をミリ秒単位で指定します。

説明

mail.imap.writetimeout では、ソケットの書き込みタイムアウト値をミリ秒単位で指定します。このタイムアウト値は、接続ごとに java.util.concurrent.ScheduledExecutorService クラスで実装されます。java.util.concurrent.ScheduledExecutorService クラスでは、タイムアウト値を超えた場合にソケットを閉じるためのスレッドをスケジュールします。mail.imap.writetimeout 使用時のオーバーヘッドは、1 接続当たり 1 スレッドです。デフォルト値は-1 です。タイムアウト値に-1 もしくは0 を指定、またはこのプロパティを省略した場合、タイムアウトは設定されません。

指定できる値

型：Integer

-1～2147483647

デフォルト値

定義項目の省略

-1 (タイムアウトは設定されません)

記述例

```
Properties properties = new Properties();
properties.setProperty("mail.imap.writetimeout", "30");
Session session = Session.getDefaultInstance(properties);
Store store = session.getStore("imap");
store.connect("user", "password");
messages = folder.getMessages();
```

7

JavaVM で使用するオプションとプロパティ

JavaVM をチューニングするためのオプションとプロパティを説明しています。

7.1 JavaVM で使用するオプションとプロパティの一覧

JavaVM で使用するすべてのオプションとプロパティの概要を説明しています。また、JavaVM で使用する環境変数の概要も説明しています。

注意事項

- ここで説明している JavaVM のオプションおよびプロパティのデフォルト値は、サーバのデフォルト値とは異なる場合があります。
- 各サーバのデフォルト値については、マニュアル『ユーザーズガイド』の『create-jvm-options サブコマンドを使用して JavaVM オプションを変更する』に関する説明を参照してください。

一覧表示で使用する JavaVM オプション

オプション	概要	詳細説明
7.2.1 -XX:+Hitachi	JavaVM 拡張オプションの一覧の表示	JavaVM 拡張オプションを一覧表示します。

拡張スレッドダンプ機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.3.1 -XX: [+ -]HitachiThreadD ump	拡張スレッドダンプ情報の出力	拡張スレッドダンプ情報を出力するかどうかを指定します。
7.3.2 -XX: [+ -]HitachiThreadD umpToStdout	拡張スレッドダンプの標準出力への出力	拡張スレッドダンプを標準出力へ出力するかどうかを指定します。
7.3.3 -XX: [+ -]HitachiThreadD umpWithBlockCoun t	拡張スレッドダンプのスレッド情報へのスレッドが処理をブロックした回数と待ち状態になった回数の出力	拡張スレッドダンプのスレッド情報に、スレッドが処理をブロックした回数と待ち状態になった回数を出力するかどうかを指定します。
7.3.4 -XX: [+ -]HitachiThreadD umpWithCpuTime	拡張スレッドダンプのスレッド情報へのユーザー CPU 時間とカーネル CPU 時間の出力	拡張スレッドダンプのスレッド情報に、ユーザー CPU 時間とカーネル CPU 時間を出力するかどうかを指定します。
7.3.5 -XX: [+ -]HitachiThreadD umpWithHashCode	拡張スレッドダンプのスレッド情報へのハッシュコードの出力	拡張スレッドダンプのスレッド情報にハッシュコードを出力するかどうかを指定します。

JavaVM ログファイルで使用する JavaVM オプション

オプション	概要	詳細説明
7.4.1 - XX:HitachiJavaLog	JavaVM ログファイルのプリフィクスおよびログファイルの出力先ディレクトリーの指定	JavaVM ログファイルのプリフィクスおよびログファイルの出力先ディレクトリーを指定します。
7.4.2 - XX:HitachiJavaLogFile Size	JavaVM ログファイルの 1 ファイル当たりの最大ファイルサイズの指定	ログファイルの単純増加を防ぐため、1 ファイルの最大ファイルサイズを指定します。最大ファイルサイズを超えた場合は、そのファイルへは出力しません。
7.4.3 -XX: [+ -]HitachiJavaLog NoMoreOutput	ログファイル作成時に入出力エラーが発生した場合のログ情報の出力方法の指定	ログファイル作成時に入出力エラーが発生した場合に、ログ情報の出力方法について指定します。なお、どちらを指定した場合も、JavaVM の処理は継続されます。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。
7.4.4 - XX:HitachiJavaLogN umberOfFile	JavaVM ログファイルの作成する最大ファイル数の指定	ログファイルの単純増加を防ぐため、作成する最大ファイル数を指定します。
7.4.5 -XX: [+ -]JavaLogAsynchr onous	ログファイルの非同期出力機能の有効または無効の設定	ログファイルの非同期出力機能の有効、無効を指定します。
7.4.6 -XX: [+ -]JavaLogHeader Output	JavaVM ログファイルオープン時のヘッダー情報の出力	JavaVM ログファイルのオープン時にヘッダー情報を出力するかどうかを指定します。ヘッダー情報は、1 行目に JavaVM のバージョン情報、2 行目に起動コマンドライン情報で構成されます。-XX:HitachiJavaLogFile Size オプションで 1 ファイルの最大ファイルサイズを指定する場合は、出力されるヘッダー情報のサイズを考慮して指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。
7.4.7 - XX:JavaLogRotation Time	JavaVM ログファイルのローテーション時刻の指定	JavaVM ログファイルのローテーション方式がシフト方式の場合に、出力先ファイルを切り替える時刻をこのオプションで指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。このオプションに指定した時刻になったタイミングで、JavaVM ログファイルを切り替えます。
7.4.8 - XX:JavaLogRotation Type	JavaVM ログファイルのローテーション方式の指定	JavaVM ログファイルのローテーション方式を指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。
7.4.9 -XX: [+ -]StandardLogTo HitachiJavaLog	JavaVM の標準出力と標準エラー出力の内容の JavaVM ログファイルへの出力	JavaVM からの標準出力と標準エラーへの出力結果を JavaVM ログファイルに出力します。障害発生時のトラブルシュートのためにこの機能を使用します。通常の運用では使用しません。

詳細時間出力で使用する JavaVM オプション

オプション	概要	詳細説明
7.5.1 -XX: [+/-]HitachiOutputMilliTime	JavaVM ログファイルに出力する日時のミリ秒までの出力	ミリ秒までの時間を出力するかどうかを指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。

拡張 verbosegc 機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.6.1 -XX: [+/-]HitachiCommaVerboseGC	拡張 verbosegc 情報の CSV 形式での出力	拡張 verbosegc 情報を CSV 形式で出力するかどうかを指定します。
7.6.2 -XX: [+/-]HitachiVerboseGC	GC が発生した場合に拡張 verbosegc 情報の JavaVM ログファイルへの出力	GC が発生した時の拡張 verbosegc 情報を出力するかどうかを指定します。
7.6.3 -XX: [+/-]HitachiVerboseGCCpuTime	拡張 verbosegc 情報へのユーザー CPU 時間とカーネル CPU 時間の出力	拡張 verbosegc 情報に CPU 利用時間を出力するかどうかを指定します。
7.6.4 -XX: HitachiVerboseGCIntervalTime	拡張 verbosegc 情報を秒単位で出力する時間の間隔の指定	拡張 verbosegc 情報を出力する時間（秒）の間隔を指定します。
7.6.5 -XX: [+/-]HitachiVerboseGCPrintCause	GC の要因内容の出力	GC の要因内容を出力するかどうかを指定します。
7.6.6 -XX: [+/-]HitachiVerboseGCPrintDate	拡張 verbosegc 情報への GC 開始日時の出力	拡張 verbosegc 情報に GC 開始日時を出力するかどうかを指定します。
7.6.7 -XX: [+/-]HitachiVerboseGCPrintDeleteOnExit	java.io.File.deleteOnExit()の呼び出しで JavaVM が確保した累積のヒープサイズとメソッドの呼び出し回数の JavaVM ログファイルへの出力	java.io.File.deleteOnExit()を呼び出したことによって JavaVM が確保した累積のヒープサイズとメソッドの呼び出し回数を、JavaVM ログファイルに出力するかどうかを指定します。
7.6.8 -XX: [+/-]HitachiVerboseGCPrintJVMInternalMemory	JavaVM 内部で管理しているヒープ情報の JavaVM ログファイルへの出力	JavaVM 内部で管理しているヒープ情報を JavaVM ログファイルに出力するかどうかを指定します。
7.6.9 -XX: [+/-]HitachiVerboseGCPrintTenuringDistribution	Survivor 領域の年齢分布の JavaVM ログファイルへの出力	-XX:+PrintTenuringDistribution 指定時に出力される Survivor 領域の年齢分布を JavaVM ログファイルへ出力するかどうかを指定します。
7.6.10 -XX: [+/-]HitachiVerbose	Java スレッドの数の JavaVM ログファイルへの出力	Java スレッドの数を監視するために、Java スレッドの数を JavaVM ログファイルに出力するかどうかを指定します。

オプション	概要	詳細説明
GCPrintThreadCount		

コードキャッシュ領域情報出力機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.7.1 -XX:CodeCacheInfoPrintRatio	コードキャッシュ領域の使用率のしきい値の指定	コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージを出力する契機となる、コードキャッシュ領域の使用率を指定します。
7.7.2 -XX:[+/-]PrintCodeCacheFullMessage	Java メソッドの JIT コンパイル対象時にコードキャッシュ領域が枯渇したときのメッセージの JavaVM ログファイルへの出力	Java メソッドが JIT コンパイルの対象になった場合、コードキャッシュ領域が枯渇していたときに、メッセージを JavaVM ログファイルに出力するかどうかを指定します。
7.7.3 -XX:[+/-]PrintCodeCacheInfo	コードキャッシュ領域の使用量の出力としきい値に達したことを知らせるメッセージの JavaVM ログファイルへの出力の指定	コードキャッシュ領域の使用量を出力するかどうかを指定します。また、使用量がしきい値に達したことを知らせるメッセージを JavaVM ログファイルに出力するかどうかを指定します。

OutOfMemoryError 発生時の拡張機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.8.1 -XX:[+/-]HitachiOutOfMemoryAbort	OutOfMemoryError 発生時に強制終了する機能の有効または無効の設定	OutOfMemoryError が発生した場合、標準出力にメッセージを出力して強制終了します。
7.8.2 -XX:[+/-]HitachiOutOfMemoryAbortThreadDump	OutOfMemoryError 発生時のスレッドダンプの出力	OutOfMemoryError 発生時にスレッドダンプを出力します。ただし、J2SE クラスライブラリーで C ヒープ不足の場合は、スレッドダンプの出力による再度の C ヒープ不足発生を避けるため、スレッドダンプは出力しません。
7.8.3 -XX:[+/-]HitachiOutOfMemoryAbortThreadDumpWithJHeapProf	OutOfMemoryError 発生時に出力するスレッドダンプログファイルへのクラス別統計情報の出力	-XX:+HitachiOutOfMemoryAbortThreadDump で出力するスレッドダンプログファイルにクラス別統計情報を出力するかどうかを指定します。 なお、G1GC を使用している場合、この機能は使用できません。G1GC 使用時にこの機能を有効にするオプションを指定した場合、指定内容は反映されなくて、この機能は無効となります。
7.8.4 -XX:[+/-]HitachiOutOfMemoryCause	OutOfMemoryError 発生時の発生要因種別の出力	OutOfMemoryError 発生時の発生要因種別を出力するかどうかを指定します。
7.8.5 -XX:[+/-]HitachiOutOfMemorySize	OutOfMemoryError 発生時に要求したメモリのサイズの出力	OutOfMemoryError 発生時に要求したメモリのサイズを出力します。

オプション	概要	詳細説明
7.8.6 -XX: [+ -]HitachiOutOfMemoryStackTrace	OutOfMemoryError 発生時の例外情報とスタックトレースの JavaVM ログファイルへの出力	OutOfMemoryError 発生時に、例外情報とスタックトレースを JavaVM ログファイルに出力するかどうかを指定します。
7.8.7 -XX:HitachiOutOfMemoryStackTraceLineSize	OutOfMemoryError 発生時に出力するスタックトレース 1 行の文字数のバイト数での指定	OutOfMemoryError 発生時に、出力するスタックトレース 1 行の文字数をバイト数で指定します。

クラスライブラリトレース機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.9.1 -XX: [+ -]HitachiJavaClassLibTrace	クラスライブラリーのスタックトレースの出力	システム全体に影響を与えるメソッドが使用された場合、これらの API のスタックトレースを、JavaVM ログファイルへ出力します。
7.9.2 -XX:HitachiJavaClassLibTraceLineSize	クラスライブラリーのスタックトレースの 1 行の文字数のバイト数での指定	-XX:+HitachiJavaClassLibTrace オプションを指定した場合に出力される、クラスライブラリーのスタックトレースの 1 行の文字数をバイト数で指定します。

ローカル変数情報出力機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.10.1 -XX:HitachiCallToString	ローカル変数オブジェクトに対して取得した String オブジェクトの文字列の出力	適用範囲 に該当するクラスのローカル変数オブジェクトに対して取得した String オブジェクトの文字列を、変数値を表現する文字列として出力します。
7.10.2 -XX: [+ -]HitachiLocalsInStackTrace	スレッドダンプ出力時のスタックトレースへの各メソッドのローカル変数情報の追加出力	スレッドダンプ出力時のスタックトレースに、各メソッドのローカル変数情報を追加して出力します。
7.10.3 -XX: [+ -]HitachiLocalsInThrowable	スタックトレース中のメソッドのローカル変数情報の出力	java.lang.Throwable.fillInStackTrace メソッド実行時に、スタックトレース中のメソッドのローカル変数情報を収集します。
7.10.4 -XX: [+ -]HitachiLocalsSimpleFormat	ローカル変数情報の出力フォーマットの 1 変数 1 行で出力する簡易出力フォーマットへの変更	ローカル変数情報の出力フォーマットを、1 変数 1 行で出力する簡易出力フォーマットに変更します。
7.10.5 -XX: [+ -]HitachiTrueTypeInLocals	ローカル変数情報への実際のオブジェクト型名の出力	ローカル変数情報収集時に、クラスまたは配列型のローカル変数について、実際に代入されているオブジェクトの型名を出力します。

明示管理ヒープ機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.11.1 - XX:ExplicitMemoryAutoReclaimTargetBlockRatio	明示管理ヒープ機能の自動解放処理で設定している解放率の上限値の指定	明示管理ヒープ機能の自動解放処理で設定している解放率の上限値を指定します。解放率とは、自動解放処理で選択した量に対する、実際に自動解放処理で解放した量の割合を指します。
7.11.2 - XX:ExplicitMemoryExcludeClassListFile	明示管理ヒープ機能適用除外設定ファイルのファイルパスの指定	明示管理ヒープ機能適用除外クラス指定機能で利用する、明示管理ヒープ機能適用除外設定ファイルのファイルパスを指定します。このファイルに記述されているクラスには、明示管理ヒープ機能が適用されません。
7.11.3 - XX:ExplicitMemoryFullGCPolicy	FullGC 発生時の参照関係に基づくオブジェクトの Java ヒープから Explicit メモリーブロックへの移動の制御	FullGC 発生時に、参照関係に基づくオブジェクトの Java ヒープから Explicit メモリーブロックへの移動を制御するかを指定します。このオプションを使用すると、FullGC 発生時に Explicit ヒープへ移動していたオブジェクトを、Tenured 領域へ移動できます。
7.11.4 - XX:ExplicitMemoryNotExcludeClassListFile	明示管理ヒープ機能適用除外無効設定ファイルのファイルパスの指定	明示管理ヒープ機能適用除外クラス指定機能で利用する、明示管理ヒープ機能適用除外無効設定ファイルのファイルパスを指定します。明示管理ヒープ機能適用除外設定ファイルに記述されていても、このファイルに記述されているクラスには、明示管理ヒープ機能適用除外クラス指定機能が適用されません。
7.11.5 -XX: [+ -]ExplicitMemoryUseExcludeClass	明示管理ヒープ機能適用除外クラス指定機能の有効または無効の設定	明示管理ヒープ機能適用除外クラス指定機能の有効、無効を指定します。
7.11.6 -XX: [+ -]HitachiAutoExplicitMemory	Explicit メモリーブロックの自動配置機能の有効または無効の設定	Explicit メモリーブロックの自動配置機能の有効、無効を指定します。
7.11.7 - XX:HitachiAutoExplicitMemoryFile	Explicit メモリーブロックの自動配置機能で利用する自動配置設定ファイルのパスの指定	Explicit メモリーブロックの自動配置機能で利用する自動配置設定ファイルのパスを指定します。
7.11.8 - XX:HitachiExplicitHeapMaxSize	Explicit ヒープ全体の最大サイズの指定	Explicit ヒープ全体の最大サイズを指定します。
7.11.9 -XX: [+ -]HitachiExplicitMemoryAutoReclaim	明示管理ヒープ機能の自動解放機能の有効または無効の設定	明示管理ヒープ機能の自動解放機能を利用するかどうか指定します。
7.11.10 -XX: [+ -]HitachiExplicitMemoryAutoRefReclaim	HTTP セッションで利用する Explicit ヒープの省メモリー化機能の有効または無効の設定	HTTP セッションで利用する Explicit ヒープの省メモリー化機能の有効、無効を指定します。
7.11.11 -XX: [+ -]HitachiExplicitMemoryCompatibleToV8	Explicit メモリーブロックの確保方法を 08-00 と同様にするかどうかの指定	Explicit メモリーブロックを確保する方法を、08-00 と同様にするかどうか指定します。08-50 以降の新機能を利用しないで、08-00 で動作するアプリケーションをそのまま 08-50 で動作させる場合、このオプションを有効にします。

オプション	概要	詳細説明
7.11.12 - XX:HitachiExplicitMemoryJavaLog	明示管理ヒープ機能のログファイルのプリフィクスおよびログファイルの出力先ディレクトリーの指定	明示管理ヒープ機能のログファイルのプリフィクスおよびログファイルの出力先ディレクトリーを指定します。
7.11.13 - XX:HitachiExplicitMemoryJavaLogFileSiz e	明示管理ヒープ機能のログファイルの1ファイル当たりの最大ファイルサイズのバイト単位での指定	1ファイルの最大ファイルサイズをバイト単位で設定します。
7.11.14 - XX:HitachiExplicitMemoryJavaLogNumb erOfFile	明示管理ヒープ機能のログファイルを作成する最大ファイル数の指定	ログファイルの単純増加を防ぐため、作成する最大ファイル数を指定します。
7.11.15 - XX:HitachiExplicitMemoryLogLevel	明示管理ヒープ機能のイベントログのレベルの指定	明示管理ヒープ機能のイベントログのレベルを指定します。
7.11.16 -XX: [+ -]HitachiExplicitMemoryMoveToTenur edFirst	Explicitメモリーブロックの解放処理でのオブジェクトの移動先のTenured領域への変更	Explicitメモリーブロックの解放処理でのオブジェクトの移動先を指定します。
7.11.17 -XX: [+ -]HitachiUseExplicitMemory	明示管理ヒープ機能の有効または無効の設定	明示管理ヒープ機能の有効、無効を指定します。

リソースの上限値指定で使用する JavaVM オプション

オプション	概要	詳細説明
7.12.1 - XX:HitachiJITCompileMaxMemorySize	JIT コンパイル時に確保する C ヒープのメモリーサイズの上限値の指定	JIT コンパイル時に確保する C ヒープのメモリーサイズの上限値を指定します。
7.12.2 - XX:HitachiThreadLimit	スレッド数の上限値の指定	スレッド数の上限値を指定します。

JIT コンパイラー稼働継続機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.13.1 -XX: [+ -]JITCompilerContinuation	JIT コンパイラー稼働継続機能の有効または無効の設定	JIT コンパイラー稼働継続機能を有効にするかどうかを指定します。

Java アプリケーションプログラムの実行で使用する JavaVM オプション

オプション	概要	詳細説明
7.14.1 -Xhras	Java アプリケーションプログラムの実行	java コマンド、または javaw コマンド実行時にこのオプションを設定すると、Java アプリケーションプログラムが実行できます。

圧縮オブジェクトポインター機能で使用する JavaVM オプション

オプション	概要	詳細説明
7.15.1 -XX: [+ -]UseCompressed Oops	圧縮オブジェクトポインター機能の有効または無効の設定	圧縮オブジェクトポインター機能の有効、無効を指定します。

ファイナライズ滞留解消機能で使用する JavaVM プロパティー

プロパティー	概要	詳細説明
7.16.1 JP.co.Hitachi.soft.jv m.autofinalizer	ファイナライズ滞留解消機能の有効または無効の設定	ファイナライズ滞留解消機能の有効、無効を指定します。

ユーザー拡張性能解析トレースで使用する JavaVM プロパティー

プロパティー	概要	詳細説明
7.17.1 jvm.userprf.Enable	ユーザー拡張性能解析トレースの有効または無効の設定	ユーザー拡張性能解析トレースを有効にするかどうかを指定します。
7.17.2 jvm.userprf.Extende dSetting	ユーザー拡張性能解析トレース設定ファイルの記述形式の指定	ユーザー拡張性能解析トレース設定ファイルの記述形式に、次の方式を有効にするかどうかを指定します。
7.17.3 jvm.userprf.File	ユーザー拡張性能解析トレース設定ファイルのファイルパスの指定	ユーザー拡張性能解析トレース設定ファイルのファイルパスを指定します。
7.17.4 jvm.userprf.Limit	ユーザー拡張性能解析トレースの対象となるメソッド数の上限値の指定	ユーザー拡張性能解析トレースの対象となるメソッド数の上限値を指定します。
7.17.5 jvm.userprf.LineNum ber	メソッドの最後に実行した行の行番号のトレース情報への出力	メソッドが正常に終了した場合、メソッドの最後に実行した行の行番号をトレース情報に出力するかどうかを指定します。
7.17.6 jvm.userprf.LogLevel	ユーザー拡張性能解析トレースのトレース出力レベルの指定	ユーザー拡張性能解析トレースのトレース情報のオプション情報 (OPT) に出力する、トレース対象のメソッド名の出力レベルを指定します。
7.17.7 jvm.userprf.Throwab leName	例外またはエラーのクラス名のトレース情報への出力	メソッドが異常終了した場合、例外またはエラーのクラス名をトレース情報のオペレーション情報 (OPR) に出力するかどうかを指定します。

プロパティ	概要	詳細説明
7.17.8 jvm.userprf.ThrowableNameEditMethod	例外またはエラーのクラス名が文字数制限 32 文字を超えた場合の名前の編集方法の指定	例外またはエラーのクラス名が 32 文字を超える場合のクラス名の編集方法を指定します。例外またはエラーのクラス名が 32 文字以内であれば、このプロパティの指定に関係なく、すべての文字がトレース情報に出力されます。
7.17.9 jvm.userprf.Trace	ユーザー拡張性能解析トレース設定ファイルで指定したクラスファイルの書き換えの成功ログの出力	ユーザー拡張性能解析トレース設定ファイルで指定したクラスファイルの書き換えの成功ログを出力するかどうかを指定します。

Application Server で使用する Java HotSpot VM のオプション

オプション	概要	詳細説明
7.18.1 - agentlib:hndlwrap2	Windows ログオフ時の JavaVM の実行の継続	JDK のバージョンが 6 以降の環境で Windows をログオフしたとき、JavaVM の実行を継続するかどうかを設定します。
7.18.2 - agentlib:libname	ネイティブエージェントライブラリーのロード	ネイティブエージェントライブラリー <i>libname</i> をロードします。
7.18.3 - XX:CompressedClassSpaceSize	Compressed Class Space の最大値の指定	Compressed Class Space の最大値を指定します。
7.18.4 - XX:ConcGCThreads	Concurrent Marking の処理を並列実行するスレッド数の指定	G1GC が有効な場合に、Concurrent Marking の処理を並列実行するスレッドの数を指定します。
7.18.5 - XX:MaxGCPauseMillis	G1GC の GC によるアプリケーション停止時間の目標時間の指定	G1GC の GC によるアプリケーション停止時間の目標時間をミリ秒 (ms) 単位で指定します。
7.18.6 - XX:MaxMetaspaceSize	Metaspace 領域の最大値の指定	Metaspace 領域の最大値を指定します。
7.18.7 - XX:MaxTenuringThreshold	CopyGC 実行時に From 空間と To 空間で Java オブジェクトを入れ替える回数のしきい値の指定	CopyGC 実行時に、From 空間と To 空間で Java オブジェクトを入れ替える回数のしきい値を指定します。指定した回数を超えて入れ替え対象になった Java オブジェクトは、Tenured 領域に移動されます。
7.18.8 - XX:MetaspaceSize	Metaspace 領域に起因する FullGC の基準値の指定	Metaspace 領域に起因する FullGC の基準値を指定します。
7.18.9 - XX:NewRatio	New 領域に対する Tenured 領域の割合の指定	New 領域に対する Tenured 領域の割合を指定します。
7.18.10 - XX:ParallelGCThreads	G1GC を並列実行するスレッドの数の指定	G1GC を並列実行するスレッドの数を指定します。
7.18.11 - XX:ReservedCodeCacheSize	コードキャッシュ領域の最大サイズの指定	コードキャッシュ領域の最大サイズを指定します。

オプション	概要	詳細説明
7.18.12 -XX:SurvivorRatio	Survivor 領域の From 空間と To 空間に対する Eden 領域の割合の指定	New::Survivor 領域の From 空間と To 空間に対する New::Eden 領域の割合を指定します。
7.18.13 -XX:TargetSurvivorRatio	GC 実行後の Survivor 領域内で Java オブジェクトが占める割合の目標値の指定	GC 実行後の New::Survivor 領域内で Java オブジェクトが占める割合の目標値 (0~100 (単位: %)) を指定します。
7.18.14 -XX:[+ -]UseG1GC	G1GC の実行	G1GC を実行するかどうかを指定します。 このオプションは、ほかの GC 方式を選択するオプション (UseSerialGC、HitachiUseExplicitMemory) と同時に指定できません。
7.18.15 -XX:[+ -]UseSerialGC	SerialGC の実行	SerialGC を実行するかどうかを指定します。
7.18.16 -verbose:information-type	<i>information-type</i> に指定した情報の出力	<i>information-type</i> に指定した情報を出力します。
7.18.17 -Xloggc:file	GC イベントが発生するたびに <i>file</i> へのログの出力	-verbose:gc と同様に GC イベントが発生するたびに報告しますが、そのデータを <i>file</i> に記録します。-verbose:gc を指定したときに報告される情報のほかに、報告される各イベントの先頭に、最初の GC イベントからの経過時間 (秒単位) が付け加えられます。
7.18.18 -Xms	Java ヒープの初期サイズの指定	Java ヒープの初期サイズを指定します。
7.18.19 -Xmx	Java ヒープの最大サイズの指定	Java ヒープの最大サイズを指定します。
7.18.20 -Xprof	実行中のプログラムのプロファイルを生成してプロファイリングデータの標準出力への出力	このオプションを指定した場合、実行中のプログラムのプロファイルを生成し、プロファイリングデータを標準出力に出力します。このオプションは、プログラム開発用のユーティリティとして提供されています。本番稼働システムでの使用を目的としたものではありません。
7.18.21 -Xss	1 スタック領域の最大サイズの指定	1 スタック領域の最大サイズを指定します。

JavaVM で使用する環境変数

環境変数	概要	詳細説明
7.19.1 JAVA_HRAS	java コマンドを実行可能にするための環境変数	java コマンド実行時にこの環境変数を設定すると、Application Server と連携しない Java アプリケーションプログラムが実行できます。
7.19.2 JAVACOREDIR	スレッドダンプファイルの出力先ディレクトリーの指定	スレッドダンプファイルを出力する場合の、出力先ディレクトリーをバッチファイルで指定します。
7.19.3 JAWAW_HRAS	javaw コマンドを実行可能にするための環境変数	javaw コマンド実行時にこの環境変数を設定すると、Application Server と連携しない Java アプリケーションプログラムが実行できます。

7.2 一覧表示で使用する JavaVM オプション

JavaVM オプションの一覧を表示するオプションを説明しています。

7.2.1 -XX:+Hitachi

JavaVM 拡張オプションの一覧を表示します。

説明

JavaVM 拡張オプションを一覧表示します。

このオプションは、Java プログラムを実行しません。また、Java EE サーバの起動オプションに指定した場合、Java EE サーバは起動されません。

書式

```
-XX:+Hitachi
```

指定できる値

型 : String

-XX:+Hitachi

デフォルト値

定義項目の省略

なし

出力例

```
-XX:[+|-]HitachiVerboseGC
    print GC log in detail
-XX:[+|-]HitachiCommaVerboseGC
    output GC log separated by ", "
-XX:HitachiVerboseGCIntervalTime=<time>
    set interval for printing GC log
-XX:[+|-]HitachiVerboseGCPrintCause
    add GC cause to GC log
-XX:[+|-]HitachiVerboseGCPrintDate
    add date to GC log
-XX:[+|-]HitachiVerboseGCCpuTime
    add CPU time spent by GC to GC log
-XX:[+|-]HitachiVerboseGCPrintTenuringDistribution
    add tenuring age info. to GC log
-XX:[+|-]HitachiVerboseGCPrintJVMInternalMemory
```

```

    add JVM Internal Memory to GC log
-XX:[+|-]HitachiVerboseGCPrintThreadCount
    add thread count to GC log
-XX:[+|-]HitachiVerboseGCPrintDeleteOnExit
    add memory size and called count by java.io.File#deleteOnExit to GC log
-XX:[+|-]JavaLogAsynchronous
    enable output asynchronous javalog
-XX:HitachiJITCompileMaxMemorySize=<size>
    set max memory size of JIT compiler
-XX:[+|-]JITCompilerContinuation
    continue JIT compilation when failure occurs
-XX:HitachiThreadLimit=<num>
    set maximum number of threads
-XX:[+|-]HitachiOutOfMemoryStackTrace
    print stack trace when OutOfMemoryError occurs
-XX:[+|-]HitachiOutOfMemoryCause
    print cause of OutOfMemoryError
-XX:[+|-]HitachiOutOfMemorySize
    print requested allocation size when OutOfMemoryError occurs
-XX:HitachiOutOfMemoryStackTraceLineSize=<size>
    set stack trace line size
-XX:[+|-]HitachiOutOfMemoryAbort
    abort when OutOfMemoryError occurs
-XX:[+|-]HitachiOutOfMemoryAbortThreadDump
    print thread dump before abort at OOM
-XX:[+|-]HitachiOutOfMemoryAbortThreadDumpWithJHeapProf
    print Java heap profiling data before abort at OOM
-XX:[+|-]HitachiJavaClassLibTrace
    print java class library trace when specific API called
-XX:HitachiJavaClassLibTraceLineSize=<size>
    set java class library trace line size
-XX:[+|-]HitachiOutputMilliTime
    use milliseconds precision timer for javalog
-XX:[+|-]HitachiLocalsInThrowable
    output locals info. in Throwable#printStackTrace
-XX:[+|-]HitachiLocalsInStackTrace
    output locals info. in thread dump
-XX:[+|-]HitachiLocalsSimpleFormat
    change output format of locals info. to simpler one
-XX:[+|-]HitachiTrueTypeInLocals
    add true type name to locals info
-XX:HitachiCallToString=<range>
    specify range of locals targets to which apply toString method call
-XX:[+|-]HitachiThreadDump
    print expand thread dump
-XX:[+|-]HitachiThreadDumpToStdout
    print expand thread dump to stdout
-XX:[+|-]HitachiThreadDumpWithHashCode
    add hashCode value for thread object to expand thread dump
-XX:[+|-]HitachiThreadDumpWithCpuTime
    add CPU time spent by thread to expand thread dump
-XX:[+|-]HitachiThreadDumpWithBlockCount
    add blocked count and waited count of thread to expand thread dump
-XX:HitachiJavaLog:[<file>]
    set log filename
-XX:JavaLogRotationType:[WRAP|SHIFT]
    specify type of javalog rotation
-XX:JavaLogRotationTime:<time>

```

```

    specify time of javalog rotation
-XX:[+|-]JavaLogHeaderOutput
    output header information in javalog
-XX:HitachiJavaLogFileSize=<size>
    set maximum size of log file
-XX:HitachiJavaLogNumberOfFile=<value>
    set maximum number of log file
-XX:[+|-]HitachiJavaLogNoMoreOutput
    cease to output log when file output error occurs
-XX:[+|-]HitachiUseExplicitMemory
    use explicit memory
-XX:HitachiExplicitHeapMaxSize=<size>
    set maximum size of explicit heap
-XX:[+|-]HitachiExplicitMemoryAutoReclaim
    use explicit memory auto reclaim
-XX:[+|-]HitachiAutoExplicitMemory
    use explicit memory auto creation
-XX:HitachiAutoExplicitMemoryFile:[<file>]
    set explicit memory auto creation file
-XX:[+|-]HitachiExplicitMemoryCompatibleToV8
    maintain compatibility with version 08-00 specification of explicit
memory instead of new features
-XX:[+|-]HitachiExplicitMemoryMoveToTenuredFirst
    move Java objects to tenured area before new area in reclaim
-XX:[+|-]HitachiExplicitMemoryAutoRefReclaim
    enable auto reclaiming of reference explicit memory
-XX:HitachiExplicitMemoryLogLevel:[none|normal|verbose|debug]
    set explicit memory log level
-XX:HitachiExplicitMemoryJavaLog:[<file>]
    set explicit memory log filename
-XX:HitachiExplicitMemoryJavaLogFileSize=<size>
    set maximum size of explicit memory log file
-XX:HitachiExplicitMemoryJavaLogNumberOfFile=<value>
    set maximum number of explicit memory log file
-XX:ExplicitMemoryFullGCPolicy=<value>
    set full GC policy when using explicit memory
-XX:[+|-]ExplicitMemoryUseExcludeClass
    use explicit memory exclude class
-XX:ExplicitMemoryExcludeClassListFile:[<file>]
    set explicit memory exclude class list file
-XX:ExplicitMemoryNotExcludeClassListFile:[<file>]
    set explicit memory not exclude class list file
-XX:ExplicitMemoryAutoReclaimTargetBlockRatio=<value>
    set ratio of auto reclaim target block
-XX:[+|-]PrintCodeCacheInfo
    print code cache information
-XX:CodeCacheInfoPrintRatio=<value>
    set tuning parameter of code cache monitoring threshold
-XX:[+|-]PrintCodeCacheFullMessage
    print message when code cache is full
-Xhras
    use utility for stand-alone programming in executing java command
-XX:[+|-]StandardLogToHitachiJavaLog
    save JVM output to javalog

```

7.3 拡張スレッドダンプ機能で使用する JavaVM オプション

拡張スレッドダンプの出力を設定するためのオプションを説明しています。

7.3.1 -XX:[+|-]HitachiThreadDump

拡張スレッドダンプ情報を出力します。

説明

拡張スレッドダンプ情報を出力するかどうかを指定します。

スレッドダンプは、標準出力、および次に示すファイルに出力されます。

スレッドダンプ出力先

デフォルトでは、JavaVM 実行時のカレントディレクトリーです。環境変数JAVACOREDIR を指定することで、出力先を変更できます。

スレッドダンプ出力ファイル名

javacoreプロセス番号.YYMMDDhhmmss[※].txt

注※

YY：年（西暦で下2桁）、MM：月（2桁）、DD：日（2桁）

hh：時間（24時間表記）、mm：分（2桁）、ss：秒（2桁）

スレッドダンプ情報の構成を、次に示します。

表 7-1 スレッドダンプ情報の構成

出力情報	内容
ヘッダー	スレッドダンプを開始した日付および時刻、JavaVM バージョン情報、起動コマンドラインを出力します。
システム設定	次の情報を出力します。 <ul style="list-style-type: none">• JDK の実行環境のインストールディレクトリー• 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• システムクラスパス• Java コマンドオプション
動作環境	次の情報を出力します。 <ul style="list-style-type: none">• ホスト名• OS バージョン• CPU 情報
メモリー情報	次の情報を出力します。 <ul style="list-style-type: none">• メモリー使用状況

出力情報	内容
	<ul style="list-style-type: none"> 物理メモリー使用量 仮想メモリー使用量 ページング使用量
Java ヒープ情報	Java ヒープの各世代のメモリー使用状況を出力します。
JavaVM 内部メモリーマップ情報	JavaVM 自身の確保しているメモリーの領域情報を出力します。
JavaVM 内部メモリーサイズ情報	JavaVM 自身の確保しているメモリーのサイズ情報を出力します。
アプリケーション情報	次の情報を出力します。 <ul style="list-style-type: none"> シグナルハンドラー 環境変数 カレントディレクトリー情報
ライブラリー情報	ローディングされているライブラリーの情報を出力します。
スレッド情報 <スレッド1> : <スレッドn>	スレッドごとにスレッド情報を出力します。現存する全スレッドのスタックトレースを出力します。
Java モニターダンプ	Java モニターオブジェクトの一覧を表示します。スレッド間の排他待ちの状況を確認できます。
JNI グローバル参照数の情報	JavaVM が保持している JNI のグローバル参照の数を出力します。 JNI グローバル参照は、次の場合に作成されます。 <ul style="list-style-type: none"> JavaVM が自身の起動や実行のために必要な場合。 JNI がサポートする NewGlobalRef 関数を発行した場合。
Explicit ヒープ情報	Explicit ヒープ情報では、Explicit ヒープ全体での利用状況を出力します。Explicit ヒープ利用の有無に関係なく、-XX:+HitachiUseExplicitMemory が有効の場合は必ず出力されます。
Explicit メモリーブロック情報	Explicit メモリーブロック情報では、次の情報を出力します。ただし、Explicit メモリーブロックが 1 つも存在しない場合は何も出力されません。 <ul style="list-style-type: none"> Explicit メモリーブロックごとの利用状況 オブジェクト統計情報 (eheapprof コマンド実行時に出力) オブジェクト解放率情報 (-freeratio オプションを指定したeheapprof コマンド実行時に出力)
フッター	スレッドダンプを終了した日付および時刻を出力します。

各スレッドダンプ情報の出力形式を、次に示します。

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実行環境の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

メモリー情報

Memory Status

Memory in use : メモリー使用状況(単位:パーセント)
Physical memory : 物理メモリー使用量※ free
Virtual memory : 仮想メモリー使用量※ free
Paging file : ページング使用量※ free

注※

利用可能サイズ/総サイズ の形式で表示されます (単位:バイト)。

Java ヒープ情報 (SerialGC の場合)

Heap Status

def new generation max 最大の容量, total 現在の容量, ¥
used 使用中のメモリーサイズ ¥
(最大の容量に対する使用率% used/max, 現在の容量に対する使用率% used/total)
[領域先頭アドレス, コミット済み領域の末尾アドレス, ¥
予約済み領域の末尾アドレス)
eden space 現在の容量, 使用率% used [領域先頭アドレス, ¥
使用中領域の先頭アドレス, 予約済み領域の末尾アドレス)
from space 現在の容量, 使用率% used [領域先頭アドレス, ¥
使用中領域の先頭アドレス, 予約済み領域の末尾アドレス)
to space 現在の容量, 使用率% used [領域先頭アドレス, ¥
使用中領域の先頭アドレス, 予約済み領域の末尾アドレス)
tenured generation max 最大の容量, total 現在の容量, ¥
used 使用中のメモリーサイズ ¥
(最大の容量に対する使用率% used/max, 現在の容量に対する使用率% used/total)


```

        [領域先頭アドレス, コミット済み領域の末尾アドレス, ¥
予約済み領域の末尾アドレス)
    the space 現在の容量, 使用率% used [領域先頭アドレス, ¥
使用中領域の先頭アドレス, 次の空きブロックの先頭アドレス, ¥
予約済み領域の末尾アドレス)
Metaspace      max 最大の容量, capacity ¥
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, ¥
committed Metaspaceのコミット済みの合計メモリーサイズ, ¥
reserved 予約済みのメモリーサイズ, used 使用中のメモリーサイズ ¥
(最大の容量に対する使用率% used/max, 現在の容量に対する使用率% used/committed)
    class space  max 最大の容量, capacity ¥
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, ¥
committed Metaspaceのコミット済みの合計メモリーサイズ, ¥
reserved 予約済みのメモリーサイズ, used 使用中のメモリーサイズ ¥
(最大の容量に対する使用率% used/max, 現在の容量に対する使用率% used/committed)
        [領域の先頭アドレス, 使用中領域の先頭アドレス, ¥
コミット済み領域の末尾アドレス, 予約済み領域の末尾アドレス)

```

注

- 容量およびメモリーサイズの単位はキロバイトです。
- -XX:MaxMetaspaceSizeオプションを指定していない場合、Metaspace 領域の最大値は無制限です。Metaspace 領域の場合、最大の容量には「unlimited」、最大の容量に対する使用率には「-%」が出力されます。
- class space は、64 ビット版の Java VM で圧縮オブジェクトポインター機能を使用し、かつ Compressed Class Space を使用している場合にだけ出力されます。
- Metaspace 領域の値は、Compressed Class Space を含む Metaspace 領域全体の合計値です。

Java ヒープ情報 (G1GC の場合)

```

Heap Status
-----
garbage-first heap  total Javaヒープ領域の容量, ¥
used Javaヒープ領域の使用量 [Javaヒープ領域の先頭アドレス, ¥
Javaヒープ領域のコミット済みの末尾アドレス, ¥
Javaヒープ領域の予約済みの末尾アドレス)
    region size 1リージョンのサイズ, New領域の使用中のリージョン数 young ¥
(New領域の使用量), Survivor領域の使用中のリージョン数 survivors ¥
(Survivor領域の使用量)
Metaspace      max 最大の容量, capacity ¥
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, ¥
committed Metaspaceのコミット済みの合計メモリーサイズ, ¥
reserved 予約済みのメモリーサイズ, used 使用中のメモリーサイズ ¥
(最大の容量に対する使用率% used/max, 現在の容量に対する使用率% used/committed)
    class space  max 最大の容量, capacity ¥
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, ¥
committed Metaspaceのコミット済みの合計メモリーサイズ, ¥
reserved 予約済みのメモリーサイズ, used 使用中のメモリーサイズ ¥
(最大の容量に対する使用率% used/max, 現在の容量に対する使用率% used/committed)
        [領域の先頭アドレス, 使用中領域の先頭アドレス, ¥
コミット済み領域の末尾アドレス, 予約済み領域の末尾アドレス)

```

注

容量の単位はキロバイトです。

JavaVM 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)。

JavaVM 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

アプリケーション情報

Application Environment

Signal Handlers -

SIGINT :シグナルハンドラー情報

SIGILL :シグナルハンドラー情報

...

SIGBREAK :シグナルハンドラー情報

SIGABRT :シグナルハンドラー情報

Environment Variables -

環境変数=値

...

Current Directory -

C:¥Program Files¥Hitachi¥APServer¥CC¥...

ライブラリー情報

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

...

スレッド情報

Stack Trace

```

”スレッド名” #スレッド識別子 daemon prio=優先度 os_prio=OS優先度 ¥
jid=ハッシュ値 tid=スレッドID nid=native ID 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)
...

```

出力内容を説明します。

出力項目	出力内容
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。
スレッド識別子	Java スレッドが作成されたときに生成される一意の数字が出力されます。 この値は、java.lang.Thread.getId()で取得できる値と同じです。
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。
OS 優先度	OS レベルの優先度が出力されます。優先順位が設定できない OS の場合は、0 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が出力されます。
nativeID	OS レベルのスレッド ID が出力されます。
status	スレッドの状態が出力されます。出力される内容を次に示します。 <ul style="list-style-type: none"> 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。
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。
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。

出力項目	出力内容
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。
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。 <ul style="list-style-type: none"> 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。

出力項目	出力内容
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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> ... 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。

出力項目	出力内容
JNI グローバル参照数	JavaVM 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 JavaVM 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、JavaVM 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
  deployed objects
    _____
    ISIZE      INUM      FRATIO CNAME
    .....
    AISIZE     AINUM total
...

```

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 文字分の空白があります。
- deployed objects の前には、半角 4 文字分の空白があります。
- ISIZE は、_____Size_ の"e"に行末をそろえて記載しています。
- INUM は、__Instances の最後"s"に行末をそろえて記載しています。
- FRATIO は、__FreeRatio_ の"o"に行末をそろえて記載しています。
- CNAME は、__Class_____の最初から 2 つ目の"_"に行頭をそろえて記載しています。
- 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。

表 7-2 標準スレッドダンプと拡張スレッドダンプの出力情報の比較

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	×	○
Java ヒープ情報	○	○
JavaVM 内部メモリーマップ情報	×	○
JavaVM 内部メモリーサイズ情報	×	○
アプリケーション情報	×	○
ライブラリー情報	×	○
スレッド情報	○	○※1
Java モニターダンプ	×	○
JNI グローバル参照数の情報	○	○
Explicit ヒープ情報	×	○※2
Explicit メモリーブロック情報	×	○※2
クラス別統計情報	×	○※3
フッター	×	○
スレッドダンプ出力先	標準出力	標準出力※4 JavaVM ログファイル

(凡例)

- ：出力します。
- ×：出力しません。

注※1

スタックの開始および終了のアドレス情報などを出力します。

注※2

-XX:+HitachiUseExplicitMemory オプションが有効の場合に出力します。

注※3

スレッドダンプの出力契機がjheapprof コマンドの場合に出力します。

注※4

-XX:+HitachiThreadDumpToStdout オプションが指定された場合に出力します。

書式

```
-XX:[+|-]HitachiThreadDump
```

指定できる値

型 : String

-XX:+HitachiThreadDump

-Xrs オプションが指定されていない場合、スレッドダンプ出力時に拡張スレッドダンプ情報を出力します。

-XX:-HitachiThreadDump

スレッドダンプ出力時に標準のスレッドダンプ情報を出力します。

デフォルト値

定義項目の省略

-XX:+HitachiThreadDump

出力例

```
Thu Nov 06 13:56:20 2014

Full thread dump Java HotSpot(TM) 64-Bit Server VM
(25.20-b23-HJDK1010-20141105 mixed mode)
E:\home\Binary\Image\jdk\bin\java.exe -Xhras -Xms256m -Xmx512m
-XX:+UseCompressedOops -XX:MaxMetaspaceSize=128m Test

System Properties
-----
Java Home Dir   : E:\home\Binary\Image\jdk\jre
Java DLL Dir    : E:\home\Binary\Image\jdk\jre\bin
Sys Classpath   : E:\home\Binary\Image\jdk\jre\lib\resources.jar;
E:\home\Binary\Image\jdk\jre\lib\rt.jar;
E:\home\Binary\Image\jdk\jre\lib\sunrsasign.jar;
E:\home\Binary\Image\jdk\jre\lib\jsse.jar;
E:\home\Binary\Image\jdk\jre\lib\jce.jar;
E:\home\Binary\Image\jdk\jre\lib\charsets.jar;
E:\home\Binary\Image\jdk\jre\lib\jfr.jar;E:\home\Binary\Image\jdk\jre\classes
User Args       :
-Djava.class.path=.
-XX:+HitachiVerboseGC
-XX:+HitachiOutputMilliTime
```



```
-XX:+HitachiOutOfMemoryStackTrace
-XX:+HitachiJavaClassLibTrace
-XX:+HitachiLocalsInStackTrace
-XX:+HitachiLocalsSimpleFormat
-XX:+HitachiOutOfMemoryAbort
-XX:-HitachiThreadDumpToStdout
-Xms256m
-Xmx512m
-XX:+UseCompressedOops
-XX:MaxMetaspaceSize=128m
-Dsun.java.command=Test
-Dsun.java.launcher=SUN_STANDARD
```

Operating Environment

```
-----
Host      : WINDOWS64PC:10.255.255.255
OS       : Microsoft Windows Server 2008 R2 , 64 bit 6.1.7601 Service Pack 1
CPU      : amd64 8664 (Level 6, Rev. 0x2d07), 8/8 active
```

Memory Status

```
-----
Memory in use   : 72%
Physical memory : 2387939328 / 8589402112 free
Virtual memory  : 8794114576384 / 8796092891136 free
Paging file     : 11622756352 / 17292402688 free
```

Heap Status

```
-----
def new generation   max 157248K, total 78656K, used 6995K (4.4% used/max,
8.9% used/total)
[0x00000000e0000000, 0x00000000e5550000,
0x00000000eaaa0000)
eden space 69952K, 10% used [0x00000000e0000000, 0x00000000e06d4e18,
0x00000000e4450000)
from space 8704K, 0% used [0x00000000e4450000, 0x00000000e4450000,
0x00000000e4cd0000)
to space 8704K, 0% used [0x00000000e4cd0000, 0x00000000e4cd0000,
0x00000000e5550000)
tenured generation  max 349568K, total 174784K, used 0K (0.0% used/max,
0.0% used/total)
[0x00000000eaaa0000, 0x00000000f5550000,
0x0000000100000000)
the space 174784K, 0% used [0x00000000eaaa0000, 0x00000000eaaa0000,
0x00000000eaaa0200, 0x00000000f5550000)
Metaspace           max 131072K, capacity 4492K, committed 4864K,
reserved 1056768K, used 3738K, (2.9% used/max, 76.9% used/committed)
class space         max 1048576K, capacity 388K, committed 512K,
reserved 1048576K, used 378K, (0.0% used/max, 74.0% used/committed)
[0x0000000011c90000, 0x0000000011cf1000, 0x0000000011d10000,
0x0000000051c90000)
```

JVM Internal Memory Map

```
-----
mmap() : address = 0x000000001bb0000 - 0x000000001bf0000
(size:262144)
mmap() : address = 0x000000001c00000 - 0x000000001c40000
(size:262144)
```



```
mmap() : address = 0x00000000024e0000 - 0x0000000002560000
(size:524288)
mmap() : address = 0x00000000052490000 - 0x000000000525d0000
(size:1310720)
mmap() : address = 0x000000000525f0000 - 0x000000000526b0000
(size:786432)
mmap() : address = 0x00000000052800000 - 0x00000000052840000
(size:262144)
mmap() : address = 0x00000000052990000 - 0x000000000529d0000
(size:262144)
mmap() : address = 0x00000000052db0000 - 0x00000000052e30000
(size:524288)
```

JVM Internal Memory Status

```
-----
Heap Size : 4358144
Alloc Size : 3820000
Free Size : 538144
```

Application Environment

Signal Handlers -

```
SIGINT : 0x0000000006ab67470
SIGILL : default
SIGFPE : default
SIGSEGV : default
SIGTERM : 0x0000000006ab67470
SIGBREAK : default
SIGABRT : default
```

Environment Variables -

```
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\%user%\AppData\Roaming
CLIENTNAME=WIN-WINDOWS64PC
CommonProgramFiles=C:\Program Files\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=WINDOWS64PC
ComSpec=C:\Windows\system32\cmd.exe
COSMINEXUS_HNTRLIB_HOME=C:\Program Files\Hitachi\HNTRLib2
COSMINEXUS_HOME=C:\Program Files\Hitachi\Cosminexus
DXSDK_DIR=C:\Program Files (x86)\Microsoft DirectX 9.0 SDK (Summer 2004)
FP_NO_HOST_CHECK=NO
HOMEDRIVE=C:
HOMEPATH=%user%
LOCALAPPDATA=C:\Users\%user%\AppData\Local
LOGONSERVER=\\WINDOVS64PC
NUMBER_OF_PROCESSORS=8
OS=Windows_NT
Path=C:\cygwin\bin;C:\Windows\system32;C:\Windows
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PRFSPPOOL=C:\Program Files\Hitachi\Cosminexus\PRF\spool
PROCESSOR_ARCHITECTURE=AMD64
PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 45 Stepping 7, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=2d07
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files
```

```

ProgramFiles(x86)=C:\Program Files (x86)
ProgramW6432=C:\Program Files
PROMPT=$P$G
PSModulePath=C:\Windows\system32\WindowsPowerShell\v1.0\Modules\
PUBLIC=C:\Users\Public
REPLAY_TEST=1
SESSIONNAME=RDP-Tcp#4
SystemDrive=C:
SystemRoot=C:\Windows
TEMP=C:\Users\guest\AppData\Local\Temp\16
TMP=C:\Users\guest\AppData\Local\Temp\16
USERDOMAIN=WINDOWS64PC
USERNAME=guest
USERPROFILE=C:\Users\guest
VS100COMNTOOLS=c:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\
Tools\
WINDBG=C:\Program Files\Debugging Tools for Windows (x64)\windbg.exe
windir=C:\Windows
windows_tracing_flags=3
windows_tracing_logfile=C:\BVTBin\Tests\installpackage\csi logfile.log

```

```

Current Directory -
      E:\home\guest

```

Loaded Libraries

Dynamic libraries:

```

0x000000013f250000 - 0x000000013f270000      E:\home\Binary\Image\jdk\bin\
java.exe
0x0000000077890000 - 0x0000000077a39000      C:\Windows\SYSTEM32\ntdll.dll
0x0000000077670000 - 0x000000007778f000      C:\Windows\system32\
kernel32.dll
0x000007fef980000 - 0x000007fef9ec000      C:\Windows\system32\
KERNELBASE.dll
0x000007fefddb0000 - 0x000007fefde8b000      C:\Windows\system32\ADVAPI32.dll
0x000007feff360000 - 0x000007feff3ff000      C:\Windows\system32\msvcrt.dll
0x000007feff840000 - 0x000007feff85f000      C:\Windows\SYSTEM32\sechost.dll
0x000007feff0c0000 - 0x000007feff1ed000      C:\Windows\system32\RPCRT4.dll
0x0000000077790000 - 0x000000007778a000      C:\Windows\system32\USER32.dll
0x000007fefefc0000 - 0x000007feff027000      C:\Windows\system32\GDI32.dll
0x000007feff030000 - 0x000007feff03e000      C:\Windows\system32\LPK.dll
0x000007feff210000 - 0x000007feff2d9000      C:\Windows\system32\USP10.dll
0x000007fefc1f0000 - 0x000007fefc3e4000      C:\Windows\WinSxS\amd64_
microsoft.windows.common-controls_6595b64144ccf1df_6.0.7601.17514_none_
fa396087175ac9ac\COMCTL32.dll
0x000007feff2e0000 - 0x000007feff351000      C:\Windows\system32\SHLWAPI.dll
0x0000000006e1e0000 - 0x0000000006e2b1000      E:\home\Binary\Image\jdk\bin\
MSVCR100.dll
0x000007fefdd80000 - 0x000007fefddae000      C:\Windows\system32\IMM32.DLL
0x000007feff9f0000 - 0x000007feffaf9000      C:\Windows\system32\MSCTF.dll
0x0000000006dfe0000 - 0x0000000006e0b1000      E:\home\Binary\Image\jdk\jre\
bin\msvcr100.dll
0x0000000006a8c0000 - 0x0000000006b13f000      E:\home\Binary\Image\jdk\jre\
bin\server\jvm.dll
0x000007fef8ff0000 - 0x000007fef8ff9000      C:\Windows\system32\WSOCK32.dll
0x000007fefdba0000 - 0x000007fefdbed000      C:\Windows\system32\WS2_32.dll
0x000007fefde90000 - 0x000007fefde98000      C:\Windows\system32\NSI.dll
0x000007fefafa870000 - 0x000007fefafa8ab000      C:\Windows\system32\WINMM.dll

```

```

0x0000000077a50000 - 0x0000000077a57000 C:\Windows\system32\PSAPI.DLL
0x0000007fef6bc0000 - 0x0000007fef6bcf000 E:\home\Binary\Image\jdk\jre\
bin\verify.dll
0x0000007fef6a20000 - 0x0000007fef6a4b000 E:\home\Binary\Image\jdk\jre\
bin\java.dll
0x0000007fef6850000 - 0x0000007fef6866000 E:\home\Binary\Image\jdk\jre\
bin\zip.dll
0x0000007fefc400000 - 0x0000007fefc415000 C:\Windows\system32\NLAapi.dll
0x0000007fef9080000 - 0x0000007fef9095000 C:\Windows\system32\napinsp.dll
0x0000007fefcfc0000 - 0x0000007fefd015000 C:\Windows\System32\mswsock.dll
0x0000007fefce40000 - 0x0000007fefce9b000 C:\Windows\system32\DNSAPI.dll
0x0000007fef9070000 - 0x0000007fef907b000 C:\Windows\System32\winrnr.dll
0x0000007fefbb30000 - 0x0000007fefbb57000 C:\Windows\system32\IPHLPAPI.DLL
0x0000007fefbb20000 - 0x0000007fefbb2b000 C:\Windows\system32\WINNSI.DLL
0x0000007fefb8b0000 - 0x0000007fefb903000 C:\Windows\System32\fwpuclnt.dll
0x0000007fefaf190000 - 0x0000007fefaf198000 C:\Windows\system32\rasadhlp.dll
0x0000007fefef230000 - 0x0000007fefefb8000 C:\Windows\system32\SHELL32.dll
0x0000007fefdf40000 - 0x0000007fefef143000 C:\Windows\system32\ole32.dll
0x0000007fefdf790000 - 0x0000007fefdf79f000 C:\Windows\system32\profapi.dll

```

Stack Trace

```

-----
"JavaLogRotationTimer" #12 daemon prio=5 os_prio=0 jid=<N/A>
tid=0x0000000052622000 nid=0x2bb78 in Object.wait()
[0x00000000540df000..0x00000000540df520]
  java.lang.Thread.State: TIMED_WAITING (on object monitor)
  stack=
[0x00000000540e0000..0x0000000053fe4000..0x0000000053fe1000..0x0000000053fe0000]
  [user cpu time=0ms, kernel cpu time=0ms] [blocked count=1, waited count=2]
  at java.lang.Object.wait(Native Method)
  - waiting on <0x00000000e0087358> (a java.util.TaskQueue)
  locals:
    (java.lang.Object) this = <0x00000000e0087358>
    (long) [arg1] = 36223964
  at java.util.TimerThread.mainLoop(Timer.java:552)
  - locked <0x00000000e0087358> (a java.util.TaskQueue)
  locals:
    (java.util.TimerThread) this = <0x00000000e00876d0>
    (java.util.TimerTask) task = <0x00000000e00ecc18>
    (boolean) taskFired = false
    (long) currentTime = 1415249776949
    (long) executionTime = 1415286000913
  at java.util.TimerThread.run(Timer.java:505)
  locals:
    (java.util.TimerThread) this = <0x00000000e00876d0>

"Service Thread" #11 daemon prio=9 os_prio=0 jid=<N/A> tid=0x000000005261b000
nid=0x1d04c runnable [0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x0000000053ba0000..0x0000000053aa4000..0x0000000053aa1000..0x0000000053aa0000]
  [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"C1 CompilerThread3" #10 daemon prio=9 os_prio=2 jid=<N/A>
tid=0x000000005258e800 nid=0x3ad64 waiting on condition
[0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=

```

```

[0x0000000053a30000..0x0000000053934000..0x0000000053931000..0x0000000053930000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"C2 CompilerThread2" #9 daemon prio=9 os_prio=2 jid=<N/A>
tid=0x0000000052581800 nid=0x1eaa8 waiting on condition
[0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x00000000538d0000..0x00000000537d4000..0x00000000537d1000..0x00000000537d0000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"C2 CompilerThread1" #8 daemon prio=9 os_prio=2 jid=<N/A>
tid=0x0000000052575800 nid=0x3cf68 waiting on condition
[0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x0000000053710000..0x0000000053614000..0x0000000053611000..0x0000000053610000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"C2 CompilerThread0" #7 daemon prio=9 os_prio=2 jid=<N/A>
tid=0x000000005256e800 nid=0x3f304 waiting on condition
[0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x00000000534b0000..0x00000000533b4000..0x00000000533b1000..0x00000000533b0000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"Attach Listener" #6 daemon prio=5 os_prio=2 jid=<N/A> tid=0x000000005256c000
nid=0xb93c runnable [0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x00000000532c0000..0x00000000531c4000..0x00000000531c1000..0x00000000531c0000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"Signal Dispatcher" #5 daemon prio=9 os_prio=2 jid=<N/A> tid=0x0000000052569800
nid=0x1b6ac waiting on condition [0x0000000000000000..0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x0000000052990000..0x0000000052894000..0x0000000052891000..0x0000000052890000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]

"FinalizerWatcherThread" #4 daemon prio=5 os_prio=0 jid=<N/A>
tid=0x000000005253f000 nid=0x6f70 waiting on condition
[0x00000000530ef000..0x00000000530ef530]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
  stack=
[0x00000000530f0000..0x0000000052ff4000..0x0000000052ff1000..0x0000000052ff0000]
 [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=0]
  at java.lang.Thread.sleep(Native Method)
  locals:
    (long) [arg1] = 60000
  at java.lang.ref.Finalizer$FinalizerWatcherThread.run
(Finalizer.java:300)
  locals:
    (java.lang.ref.Finalizer$FinalizerWatcherThread)
this = <0x00000000e0063700>
    (int) interval = 60000
    (java.lang.String) prop = (null)

```

```

    (java.lang.Object) oldObj = (null)
    (java.lang.Object) newObj = (null)
    (long) length = 0

"Finalizer" #3 daemon prio=8 os_prio=1 jid=<N/A> tid=0x000000002559800
nid=0xd50c in Object.wait() [0x0000000052f2e000..0x0000000052f2f240]
  java.lang.Thread.State: WAITING (on object monitor)
  stack=
[0x0000000052f30000..0x0000000052e34000..0x0000000052e31000..0x0000000052e30000]
  [user cpu time=0ms, kernel cpu time=0ms] [blocked count=3, waited count=2]
  at java.lang.Object.wait(Native Method)
  - waiting on <0x00000000e0006460> (a java.lang.ref.ReferenceQueue$Lock)
  locals:
    (java.lang.Object) this = <0x00000000e0006460>
    (long) [arg1] = 0
  at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:143)
  - locked <0x00000000e0006460> (a java.lang.ref.ReferenceQueue$Lock)
  locals:
    (java.lang.ref.ReferenceQueue) this = <0x00000000e0006440>
    (long) timeout [arg1] = 0
    (java.lang.ref.Reference) r = (null)
  at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:159)
  locals:
    (java.lang.ref.ReferenceQueue) this = <0x00000000e0006440>
  at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:222)
  locals:
    (java.lang.ref.Finalizer$FinalizerThread) this = <0x00000000e0006510>
    (sun.misc.JavaLangAccess) jla = <0x00000000e0063218>

"Reference Handler" #2 daemon prio=10 os_prio=2 jid=<N/A>
tid=0x000000002551000 nid=0xd6b8 in Object.wait()
[0x0000000052daf000..0x0000000052daf3d0]
  java.lang.Thread.State: WAITING (on object monitor)
  stack=
[0x0000000052db0000..0x0000000052cb4000..0x0000000052cb1000..0x0000000052cb0000]
  [user cpu time=0ms, kernel cpu time=0ms] [blocked count=0, waited count=1]
  at java.lang.Object.wait(Native Method)
  - waiting on <0x00000000e0005ed0> (a java.lang.ref.Reference$Lock)
  locals:
    (java.lang.Object) this = <0x00000000e0005ed0>
    (long) [arg1] = 0
  at java.lang.Object.wait(Object.java:502)
  locals:
    (java.lang.Object) this = <0x00000000e0005ed0>
  at java.lang.ref.Reference$ReferenceHandler.run(Reference.java:157)
  - locked <0x00000000e0005ed0> (a java.lang.ref.Reference$Lock)
  locals:
    (java.lang.ref.Reference$ReferenceHandler)
  this = <0x00000000e0005f50>

"main" #1 prio=5 os_prio=0 jid=<N/A> tid=0x000000001bba800
nid=0x25244 waiting on condition [0x00000000281e000..0x00000000281fa00]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
  stack=
[0x000000002820000..0x000000002724000..0x000000002721000..0x000000002720000]
  [user cpu time=140ms, kernel cpu time=31ms] [blocked count=0, waited count=0]
  at java.lang.Thread.sleep(Native Method)
  locals:

```

```

    (long) [arg1] = 5000
at Test.main(Test.java:5)
  locals:
    (java.lang.String[]) args [arg1] = <0x00000000e00f00f0>
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  locals:
    (java.lang.reflect.Method) [arg1] = <0x00000000e00f0170>
    (java.lang.Object) [arg2] = (null)
    (java.lang.Object[]) [arg3] = <0x00000000e00f01c8>
at sun.reflect.NativeMethodAccessorImpl.invoke
(NativeMethodAccessorImpl.java:62)
  locals:
    (sun.reflect.NativeMethodAccessorImpl) this = <0x00000000e00f0250>
    (java.lang.Object) obj [arg1] = (null)
    (java.lang.Object[]) args [arg2] = <0x00000000e00f01c8>
at sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:43)
  locals:
    (sun.reflect.DelegatingMethodAccessorImpl)
this = <0x00000000e00f02d8>
    (java.lang.Object) obj [arg1] = (null)
    (java.lang.Object[]) args [arg2] = <0x00000000e00f01c8>
at java.lang.reflect.Method.invoke(Method.java:483)
  locals:
    (java.lang.reflect.Method) this = <0x00000000e00f0170>
    (java.lang.Object) obj [arg1] = (null)
    (java.lang.Object[]) args [arg2] = <0x00000000e00f01c8>
    (sun.reflect.MethodAccessor) ma = <0x00000000e00f02d8>
at JP.co.Hitachi.soft.jvm.tools.Java_hras.execJava(Java_hras.java:53)
  locals:
    (JP.co.Hitachi.soft.jvm.tools.Java_hras)
this = <0x00000000e00f00d8>
    (java.lang.ClassLoader) loader = <0x00000000e0080f58>
    (java.lang.Class) klazz = <0x00000000e00ef170>
    (java.lang.reflect.Method) m = <0x00000000e00f0170>
at JP.co.Hitachi.soft.jvm.tools.Java_hras.main(Java_hras.java:40)
  locals:
    (java.lang.String[]) args [arg1] = <0x00000000e00efd68>
    (java.lang.String) className [arg2] = <0x00000000e00efd38>

```

"VM Thread" os_prio=2 tid=0x0000000002549800 nid=0xdfc runnable

"VM Periodic Task Thread" os_prio=2 tid=0x0000000005261f000
nid=0x240a4 waiting on condition

Java monitor

```

-----
java.util.TaskQueue@0x00000000e0087358 no owner
  waiting to be notified: 1
    "JavaLogRotationTimer" 0x00000000052622000
java.lang.ref.ReferenceQueue$Lock@0x00000000e0006460 no owner
  waiting to be notified: 1
    "Finalizer" 0x0000000002559800
java.lang.ref.Reference$Lock@0x00000000e0005ed0 no owner
  waiting to be notified: 1
    "Reference Handler" 0x0000000002551000

```

注意事項

- 環境変数JAVACOREDIR で指定したディレクトリーへの出力に失敗した場合、カレントディレクトリーに出力されます。
- カレントディレクトリーへの出力に失敗した場合、標準エラー出力に出力されます。なお、この場合、スレッドダンプは標準出力に出力されません。
- 次に示すオプションの[+|-]指定が「-」の場合、スレッド情報の一部が出力されないのをご注意ください。

オプション名称	出力されない情報
-XX:[+ -]HitachiThreadDumpWithHashCode	ハッシュ値
-XX:[+ -]HitachiThreadDumpWithCpuTime	ユーザー時間、カーネル時間
-XX:[+ -]HitachiThreadDumpWithBlockCount	ブロック回数、待機回数

7.3.2 -XX:[+|-]HitachiThreadDumpToStdout

拡張スレッドダンプを標準出力へ出力します。

説明

拡張スレッドダンプを標準出力へ出力するかどうかを指定します。

このオプションの指定に関係なく、次のメッセージは出力されます。また、拡張スレッドダンプはJavaVMログファイルへ出力されます。

Writing Java core to ファイル名※... OK

注※

絶対パス

前提オプション

- -XX:+HitachiThreadDump

書式

```
-XX:[+|-]HitachiThreadDumpToStdout
```


指定できる値

型：String

-XX:+HitachiThreadDumpToStdout

拡張スレッドダンプを標準出力およびスレッドダンプ出力ファイルに出力します。

-XX:-HitachiThreadDumpToStdout

拡張スレッドダンプを標準出力に出力しません。スレッドダンプ出力ファイルだけに出力します。

デフォルト値

定義項目の省略

-XX:+HitachiThreadDumpToStdout

注意事項

- 環境変数JAVACOREDIR で指定したディレクトリーへの出力に失敗した場合、カレントディレクトリーに出力されます。
- カレントディレクトリーへの出力に失敗した場合、標準エラー出力に出力されます。なお、この場合、スレッドダンプは標準出力に出力されません。
- 次に示すオプションの[+|-]指定が「-」の場合、スレッド情報の一部が出力されないのをご注意ください。

オプション名称	出力されない情報
-XX:[+ -]HitachiThreadDumpWithHashCode	ハッシュ値
-XX:[+ -]HitachiThreadDumpWithCpuTime	ユーザー時間、カーネル時間
-XX:[+ -]HitachiThreadDumpWithBlockCount	ブロック回数、待機回数

7.3.3 -XX:[+|-]HitachiThreadDumpWithBlockCount

拡張スレッドダンプのスレッド情報に、スレッドが処理をブロックした回数と待ち状態になった回数を出力します。

説明

拡張スレッドダンプのスレッド情報に、スレッドが処理をブロックした回数と待ち状態になった回数を出力するかどうかを指定します。

なお、ハッシュコードは、Java プログラムを実行しているスレッドに対して出力されます。JavaVM の内部動作スレッドに対しては出力されません。

前提オプション

- XX:+HitachiThreadDump

書式

```
-XX:[+|-]HitachiThreadDumpWithBlockCount
```

指定できる値

型：String

-XX:+HitachiThreadDumpWithBlockCount

拡張スレッドダンプのスレッド情報に、スレッドが処理をブロックした回数と待ち状態になった回数を出力します。

-XX:-HitachiThreadDumpWithBlockCount

拡張スレッドダンプのスレッド情報に、スレッドが処理をブロックした回数と待ち状態になった回数を出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiThreadDumpWithBlockCount

注意事項

- 環境変数JAVACOREDIRで指定したディレクトリーへの出力に失敗した場合、カレントディレクトリーに出力されます。
- カレントディレクトリーへの出力に失敗した場合、標準エラー出力に出力されます。なお、この場合、スレッドダンプは標準出力に出力されません。
- 次に示すオプションの[+|-]指定が「-」の場合、スレッド情報の一部が出力されないのをご注意ください。

オプション名称	出力されない情報
-XX:[+ -]HitachiThreadDumpWithHashCode	ハッシュ値
-XX:[+ -]HitachiThreadDumpWithCpuTime	ユーザー時間、カーネル時間
-XX:[+ -]HitachiThreadDumpWithBlockCount	ブロック回数、待機回数

7.3.4 -XX:[+|-]HitachiThreadDumpWithCpuTime

拡張スレッドダンプのスレッド情報に、ユーザー CPU 時間とカーネル CPU 時間を出力します。

説明

拡張スレッドダンプのスレッド情報に、ユーザー CPU 時間とカーネル CPU 時間を出力するかどうかを指定します。

なお、ユーザー CPU 時間とカーネル CPU 時間は、Java プログラムを実行しているスレッドに対して出力されます。JavaVM の内部動作スレッドに対しては出力されません。

前提オプション

- `-XX:+HitachiThreadDump`

書式

```
-XX:[+|-]HitachiThreadDumpWithCpuTime
```

指定できる値

型：String

`-XX:+HitachiThreadDumpWithCpuTime`

拡張スレッドダンプのスレッド情報に、スレッド開始からのユーザー CPU 時間とカーネル CPU 時間を出力します。

`-XX:-HitachiThreadDumpWithCpuTime`

拡張スレッドダンプのスレッド情報に、スレッド開始からのユーザー CPU 時間とカーネル CPU 時間を出力しません。

デフォルト値

定義項目の省略

`-XX:+HitachiThreadDumpWithCpuTime`

注意事項

- 環境変数 `JAVACOREDIR` で指定したディレクトリへの出力に失敗した場合、カレントディレクトリに出力されます。
- カレントディレクトリへの出力に失敗した場合、標準エラー出力に出力されます。なお、この場合、スレッドダンプは標準出力に出力されません。
- 次に示すオプションの `[+|-]` 指定が `[-]` の場合、スレッド情報の一部が出力されないのでご注意ください。

オプション名称	出力されない情報
<code>-XX:[+ -]HitachiThreadDumpWithHashCode</code>	ハッシュ値
<code>-XX:[+ -]HitachiThreadDumpWithCpuTime</code>	ユーザー時間、カーネル時間
<code>-XX:[+ -]HitachiThreadDumpWithBlockCount</code>	ブロック回数、待機回数

7.3.5 -XX:[+|-]HitachiThreadDumpWithHashCode

拡張スレッドダンプのスレッド情報にハッシュコードを出力します。

説明

拡張スレッドダンプのスレッド情報にハッシュコードを出力するかどうかを指定します。

なお、ハッシュコードは、Java プログラムを実行しているスレッドに対して出力されます。JavaVM の内部動作スレッドに対しては出力されません。

前提オプション

- -XX:+HitachiThreadDump

書式

```
-XX:[+|-]HitachiThreadDumpWithHashCode
```

指定できる値

型：String

-XX:+HitachiThreadDumpWithHashCode

拡張スレッドダンプのスレッド情報にハッシュコードを出力します。

-XX:-HitachiThreadDumpWithHashCode

拡張スレッドダンプのスレッド情報にハッシュコードを出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiThreadDumpWithHashCode

注意事項

- 環境変数JAVACOREDIR で指定したディレクトリーへの出力に失敗した場合、カレントディレクトリーに出力されます。
- カレントディレクトリーへの出力に失敗した場合、標準エラー出力に出力されます。なお、この場合、スレッドダンプは標準出力に出力されません。
- 次に示すオプションの[+|-]指定が「-」の場合、スレッド情報の一部が出力されないのでご注意ください。

オプション名称	出力されない情報
-XX:[+ -]HitachiThreadDumpWithHashCode	ハッシュ値
-XX:[+ -]HitachiThreadDumpWithCpuTime	ユーザー時間、カーネル時間

オプション名称	出力されない情報
-XX:[+ -]HitachiThreadDumpWithBlockCount	ブロック回数、待機回数

7.4 JavaVM ログファイルで使用する JavaVM オプション

標準の JavaVM よりも多くのトラブルシューティング情報を出力するためのオプションを説明しています。

7.4.1 -XX:HitachiJavaLog

JavaVM ログファイルのプリフィクスおよびログファイルの出力先ディレクトリーを指定します。

説明

JavaVM ログファイルのプリフィクスおよびログファイルの出力先ディレクトリーを指定します。

同じカレントディレクトリーで複数の JavaVM プロセスを同時に実行した場合や、-XX:HitachiJavaLog: オプションで同じログ出力先を指定した複数の JavaVM プロセスを同時に実行した場合、JavaVM ログファイルに情報が正しく出力されません。複数の JavaVM プロセスを実行する場合は、-XX:HitachiJavaLog: オプションでプロセスごとにログ出力先が異なるようにパスを指定してください。

前提オプション

次のどれかを指定します。

- -XX:+HitachiVerboseGC
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutOfMemoryCause
- -XX:+HitachiOutOfMemorySize
- -XX:+HitachiJavaClassLibTrace
- -XX:HitachiJITCompileMaxMemorySize=*数値* (*数値*が 1 以上の場合)
- -XX:+JITCompilerContinuation
- -XX:+StandardLogToHitachiJavaLog

書式

```
-XX:HitachiJavaLog:文字列
```

指定できる値

文字列

型: String

プリフィクスおよびパスを指定します。このオプションを指定しない場合、文字列には、"javalog"が設定されます。プリフィクス以下のファイル名称は、JavaVM ログファイルのローテーション方式によって異なります。

JavaVM ログファイルのローテーション方式がラップアラウンド方式の場合、ログファイル名は、**文字列??**.log (??は 01~99 の通し番号) で生成されます。例えば、文字列に"Samp"を指定すると、ログファイル名はSamp01.log になります。

JavaVM ログファイルのローテーション方式がシフト方式の場合、出力対象となるカレントログファイルとバックアップファイルが生成されます。カレントログファイルは**文字列**.log、バックアップログファイルは**文字列??**.log (??は 01~98 の通し番号) で生成されます。例えば、文字列に"Samp"を指定すると、カレントログファイル名はSamp.log、カレントログファイルがバックアップされると、バックアップログファイル名はSamp01.log になります。

プリフィクスおよびパスには次の 3 種類の指定ができます。

プリフィクスを指定する場合

指定した**文字列**のファイル名でカレントディレクトリーに出力されます。

パスを指定する場合

文字列にディレクトリーを指定した場合、そのディレクトリーにデフォルトの"java.log"でファイルが作成されます。

パスとプリフィクスを同時に指定する場合

文字列にディレクトリーとプリフィクスを指定した場合、そのディレクトリーにファイルが作成されます。例えば、文字列に"d:¥temp¥Samp"を指定すると、d:¥temp ディレクトリーに、ラップアラウンド方式ではSamp01.log、シフト方式ではSamp.log が生成されます。

デフォルト値

定義項目の省略

-XX:HitachiJavaLog:java.log

7.4.2 -XX:HitachiJavaLogFileSize

JavaVM ログファイルの 1 ファイル当たりの最大ファイルサイズを指定します。

説明

ログファイルの単純増加を防ぐため、1 ファイルの最大ファイルサイズを指定します。最大ファイルサイズを超えた場合は、そのファイルへは出力しません。

前提オプション

次のどれかを指定します。

- -XX:+HitachiVerboseGC
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutOfMemoryCause
- -XX:+HitachiOutOfMemorySize

- -XX:+HitachiJavaClassLibTrace
- -XX:HitachiJITCompileMaxMemorySize=*数値* (*数値*が 1 以上の場合)
- -XX:+JITCompilerContinuation
- -XX:+StandardLogToHitachiJavaLog

書式

```
-XX:HitachiJavaLogFileSize=整数値
```

指定できる値

整数値

型：Integer

8192～2147483647 の範囲で整数値（単位：バイト）を指定します。範囲外の値が指定された場合は 8192 が設定されます。負の値を指定した場合はエラーとなります。

デフォルト値

定義項目の省略

ラップア라운드方式の場合

```
-XX:HitachiJavaLogFileSize=256k
```

シフト方式の場合

```
-XX:HitachiJavaLogFileSize=128M
```

7.4.3 -XX:[+|-]HitachiJavaLogNoMoreOutput

ログファイル作成時に入出力エラーが発生した場合の、ログ情報の出力方法を指定します。

説明

ログファイル作成時に入出力エラーが発生した場合に、ログ情報の出力方法について指定します。なお、どちらを指定した場合も、JavaVM の処理は継続されます。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。

前提オプション

次のどれかを指定します。

- -XX:+HitachiVerboseGC
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutOfMemoryCause

- -XX:+HitachiOutOfMemorySize
- -XX:+HitachiJavaClassLibTrace
- -XX:HitachiExplicitMemoryLogLevel:none 以外
- -XX:HitachiJITCompileMaxMemorySize=数値 (数値が 1 以上の場合)
- -XX:+JITCompilerContinuation
- -XX:+StandardLogToHitachiJavaLog

書式

```
-XX:[+|-]HitachiJavaLogNoMoreOutput
```

指定できる値

型: String

-XX:+HitachiJavaLogNoMoreOutput

ログファイル出力時にファイル入出力エラーが発生した場合、次のメッセージを標準エラー出力に出力し、ログ情報の出力を停止します。

```
Java logfile output failed.(errno=エラーとなった入出力関数名:エラー番号)
```

-XX:-HitachiJavaLogNoMoreOutput

ログファイル出力時にファイル入出力エラーが発生した場合、次のメッセージを標準エラー出力に出力し、ログ情報の出力先を標準エラー出力に変更して出力を継続します。

```
Java logfile output failed.(errno=エラーとなった入出力関数名:エラー番号)
Changing output to stderr
```

デフォルト値

定義項目の省略

-XX:+HitachiJavaLogNoMoreOutput

7.4.4 -XX:HitachiJavaLogNumberOfFile

JavaVM ログファイルの作成する最大ファイル数を指定します。

説明

ログファイルの単純増加を防ぐため、作成する最大ファイル数を指定します。

ローテーション方式がラップアラウンド方式で最大ファイル数を超えた場合は、再度最初に作成したファイルへ出力します。

ローテーション方式がシフト方式で最大ファイル数を超えた場合は、いちばん古いバックアップファイルを削除します。

前提オプション

次のどれかを指定します。

- `-XX:+HitachiVerboseGC`
- `-XX:+HitachiOutOfMemoryStackTrace`
- `-XX:+HitachiOutOfMemoryCause`
- `-XX:+HitachiOutOfMemorySize`
- `-XX:+HitachiJavaClassLibTrace`
- `-XX:HitachiJITCompileMaxMemorySize=数値` (数値が 1 以上の場合)
- `-XX:+JITCompilerContinuation`
- `-XX:+StandardLogToHitachiJavaLog`

書式

```
-XX:HitachiJavaLogNumberOfFile=整数値
```

指定できる値

整数値

型: Integer

1~99 の範囲で指定します。100 以上の値が指定された場合は99 が、0 が指定された場合は1 が設定されます。負の値を指定した場合はエラーとなります。

デフォルト値

定義項目の省略

ラップアラウンド方式の場合

```
-XX:HitachiJavaLogNumberOfFile=4
```

シフト方式の場合

```
-XX:HitachiJavaLogNumberOfFile=8
```

7.4.5 -XX:[+|-]JavaLogAsynchronous

ログファイルの非同期出力機能の有効または無効を設定します。

説明

ログファイルの非同期出力機能の有効、無効を指定します。

ログファイルの非同期出力機能使用時、`-XX:HitachiOutOfMemoryStackTraceLineSize` オプション、または `-XX:HitachiJavaClassLibTraceLineSize` オプションに、4096 より大きな値を指定した場合は、出力するスタックトレース 1 行の文字数に 4096 バイトが指定されたものとして動作します。指定したバイト数が確保できない場合は警告メッセージが出力され、スタックトレースは出力されません。また、1 行の文字数が指定した文字数を超えた場合、「at」以降の文字列の前半部分を削除して、指定された文字数分出力します。

書式

```
-XX:[+|-]JavaLogAsynchronous
```

指定できる値

型 : String

`-XX:+JavaLogAsynchronous`

ログファイルの非同期出力機能を有効にします。

`-XX:-JavaLogAsynchronous`

ログファイルの非同期出力機能を無効にします。

デフォルト値

定義項目の省略

`-XX:-JavaLogAsynchronous`

7.4.6 -XX:[+|-]JavaLogHeaderOutput

JavaVM ログファイルのオープン時にヘッダー情報を出力します。

説明

JavaVM ログファイルのオープン時にヘッダー情報を出力するかどうかを指定します。ヘッダー情報は、1 行目に JavaVM のバージョン情報、2 行目に起動コマンドライン情報で構成されます。-

`XX:HitachiJavaLogFileSize` オプションで 1 ファイルの最大ファイルサイズを指定する場合は、出力されるヘッダー情報のサイズを考慮して指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。

ローテーション方式がシフト方式の場合、JavaVM の起動時に JavaVM ログファイルを追記モードでオープンします。そのため、JavaVM の開始と停止を繰り返すと、JavaVM の実行ごとに出力されたログ情報

が、1つのJavaVM ログファイルに書き込まれます。このオプションが有効な場合は、ヘッダー情報でJavaVM が起動したタイミングを確認でき、JavaVM の実行ごとに出力されたログ情報を切り分けられますが、このオプションが無効な場合は、JavaVM の実行ごとに出力されたログ情報の切り分けが難しくなります。そのため、このオプションは有効にすることをお勧めします。

前提オプション

次のどれかを指定します。

- -XX:+HitachiVerboseGC
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutOfMemoryCause
- -XX:+HitachiOutOfMemorySize
- -XX:+HitachiJavaClassLibTrace
- -XX:HitachiExplicitMemoryLogLevel:none 以外
- -XX:HitachiJITCompileMaxMemorySize=**数値** (数値が 1 以上の場合)
- -XX:+JITCompilerContinuation
- -XX:+StandardLogToHitachiJavaLog

書式

```
-XX:[+|-]JavaLogHeaderOutput
```

指定できる値

型 : String

-XX:+JavaLogHeaderOutput

JavaVM ログファイルのオープン時にヘッダー情報を出力します。

-XX:-JavaLogHeaderOutput

JavaVM ログファイルのオープン時にヘッダー情報を出力しません。

デフォルト値

値の省略

-XX:+JavaLogHeaderOutput

出力例

```
[JVM]<Fri May 30 19:27:24 2014>Java HotSpot(TM) 64-Bit Server VM  
(24.51-b03-HJDK1000-20140529) for windows-amd64 JRE  
(1.7.0_51-b13-HJDK1000-20140529), built on May 29 2014 16:04:41 by "Java"  
with unknown MS VC++:1600
```

```
[JVM]Command : C:\Hitachi\APServer\jdk\bin\java.exe
-Djava.class.path=. -XX:+HitachiVerboseGC HelloWorld
```

7.4.7 -XX:JavaLogRotationTime

JavaVM ログファイルのローテーション時刻を指定します。

説明

JavaVM ログファイルのローテーション方式がシフト方式の場合に、出力先ファイルを切り替える時刻をこのオプションで指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。このオプションに指定した時刻になったタイミングで、JavaVM ログファイルを切り替えます。このオプションの指定がない場合、または範囲外の値が指定された場合は、デフォルトの00時00分00秒になったタイミングでログファイルを切り替えます。ラップアラウンド方式(-XX:JavaLogRotationType:WRAP)の場合、時刻による出力先ファイルの切り替えは動作しません。

OSのリソースの負荷や、GC実行中でスレッドが停止状態の場合など、状況によっては、出力先ファイルを切り替えるタイミングが、指定した時刻よりも数ミリ秒～数秒遅れることがあります。切り替え時刻が遅れた場合でも、次回は指定した時刻になったタイミングで切り替えるため、切り替えタイミングの遅れが蓄積されていくことはありません。

また、バックアップログとカレントログファイルに出力するログ情報の出力時間についても、切り替えタイミングの遅れによって誤差が出ます。例えば、このオプションに000000(00時00分00秒)を指定した場合、2ファイルにわたって00時00分00秒前後のログ情報が出力されることがあります。ログファイルの非同期出力機能を使用した場合、ログ出力専用のスレッドで非同期にログファイルへ出力するため、この切り替え時間とログ情報の出力時間の誤差が出るおそれがあります。

出力先ファイルの切り替えは、毎日指定した時刻に実行されますが、指定した時刻が現時刻よりも前であった場合は、翌日の同時刻にJavaVM ログファイルを切り替えます。例えば、このオプションに144500(14時45分00秒)を指定して、2014/2/12 15:00:00にjavaを起動した場合、2014/2/13 14:45:00にJavaVM ログファイルを切り替えます。

前提オプション

次のオプションを指定します。

- -XX:JavaLogRotationType:SHIFT

書式

```
-XX:JavaLogRotationTime:適用範囲
```

指定できる値

適用範囲

型：Integer

時刻を指定します。時刻は 000000～235959 の HHMMSS 形式で指定します。コロンなどの区切り文字は入れないで、数字 6 文字で指定してください。範囲外の不正な時刻値が指定された場合は、デフォルト値が設定されます。

デフォルト値

定義項目の省略

-XX:JavaLogRotationTime:000000

値の省略

-XX:JavaLogRotationTime:000000

7.4.8 -XX:JavaLogRotationType

JavaVM ログファイルのローテーション方式を指定します。

説明

JavaVM ログファイルのローテーション方式を指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。

前提オプション

次のどれかを指定します。

- -XX:+HitachiVerboseGC
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutOfMemoryCause
- -XX:+HitachiOutOfMemorySize
- -XX:+HitachiJavaClassLibTrace
- -XX:HitachiJITCompileMaxMemorySize=*数値* (*数値*が 1 以上の場合)
- -XX:+JITCompilerContinuation
- -XX:+StandardLogToHitachiJavaLog

書式

```
-XX:JavaLogRotationType:文字列
```

指定できる値

文字列

型：String

WRAP

JavaVM ログファイルのローテーション方式にラップアラウンド方式を設定します。

SHIFT

JavaVM ログファイルのローテーション方式にシフト方式を設定します。

このオプションを指定しない場合、または「WRAP」、「SHIFT」以外の値を指定した場合、デフォルトのシフト方式が設定されます。

デフォルト値

定義項目の省略

-XX:JavaLogRotationType:SHIFT

値の省略

-XX:JavaLogRotationType:SHIFT

7.4.9 -XX:[+|-]StandardLogToHitachiJavaLog

JavaVM の標準出力と標準エラー出力の内容を JavaVM ログファイルに出力します。

説明

JavaVM からの標準出力と標準エラーへの出力結果を JavaVM ログファイルに出力します。障害発生時のトラブルシューティングのためにこの機能を使用します。通常の運用では使用しません。

この機能を有効にした場合、JavaVM の標準出力と標準エラー出力の内容を JavaVM ログファイルにも併せて出力します。

出力形式

<i>[id]<date>stdout</i>	
出力項目	出力内容
<i>id</i>	識別子です。標準出力の場合は[ST0]、標準エラー出力の場合は[STE]を出力します。
<i>date</i>	ログ出力開始時間です。-XX:+HitachiOutputMilliTime オプションが指定されている場合は、ミリ秒単位まで出力します。
<i>stdout</i>	標準出力または標準エラー出力のメッセージ内容です。

出力したメッセージの内容が次の場合、上記の出力形式と異なる形式で出力します。

- 出力したメッセージ文字列が 2,000 バイトを超えていて、かつ文字列の最後に改行（`\n`）がある場合、次の行に空行を出力します。
- 出力対象のメッセージ文字列の途中で改行がある場合、改行したあとのメッセージには識別子やログ出力開始時間を出力しません。

書式

```
-XX:[+|-]StandardLogToHitachiJavaLog
```

指定できる値

型：String

`-XX:+StandardLogToHitachiJavaLog`

JavaVM が標準出力や標準エラーに出力するメッセージを JavaVM ログファイルに出力します。

`-XX:-StandardLogToHitachiJavaLog`

JavaVM が標準出力や標準エラーに出力するメッセージを JavaVM ログファイルに出力しません。

オプションを複数指定した場合は、最後に指定した値が適用されます。

デフォルト値

定義項目の省略

`-XX:-StandardLogToHitachiJavaLog`

出力例

```
[ST0]<Fri Jun 19 11:31:40 2015>VM option '+StandardLogToHitachiJavaLog'  
[ST0]<Fri Jun 19 11:31:40 2015>VM option '+PrintVMOptions'  
[STE]<Fri Jun 19 11:31:40 2015>Invalid initial young generation size: -Xmn0k
```

7.5 詳細時間出力で使用する JavaVM オプション

JavaVM ログに出力する時間の形式を設定するためのオプションを説明しています。

7.5.1 -XX:[+|-]HitachiOutputMilliTime

JavaVM ログファイルに出力する日時を、ミリ秒まで出力します。

説明

ミリ秒までの時間を出力するかどうかを指定します。このオプションの設定は、JavaVM ログファイルと明示管理ヒープのログファイルの両方が対象となります。

前提オプション

次のどれかを指定します。

- -XX:+HitachiVerboseGC
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutOfMemoryCause
- -XX:+HitachiOutOfMemorySize
- -XX:+HitachiJavaClassLibTrace
- -XX:HitachiJITCompileMaxMemorySize=*数値* (*数値*が 1 以上の場合)
- -XX:HitachiExplicitMemoryLogLevel:none 以外
- -XX:+JITCompilerContinuation
- -XX:+PrintCodeCacheInfo
- -XX:+PrintCodeCacheFullMessage
- -XX:+StandardLogToHitachiJavaLog

書式

```
-XX:[+|-]HitachiOutputMilliTime
```

指定できる値

型: String

-XX:+HitachiOutputMilliTime

JavaVM ログファイルに出力する日時に、ミリ秒まで出力します。

-XX:-HitachiOutputMilliTime

JavaVM ログファイルに出力する日時に、秒まで出力します。

デフォルト値

定義項目の省略

-XX:-HitachiOutputMilliTime

出力例

- 拡張 verbosegc 情報の出力

```
[VGC]<Fri Feb 06 14:09:04.661 2015>[GC 5339K->3061K(8256K), 0.0019990 secs]
[DefNew::Eden: 2688K->0K(2688K)][DefNew::Survivor: 0K->64K(64K)]
[Tenured: 2651K->2997K(5504K)]
[Metaspace: 9753K(10808K, 10880K)->9753K(10808K, 10880K)]
[class space: 3612K(3708K, 3712K)->3612K(3708K, 3712K)]
[cause:ObjAllocFail][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 8685K, 14240K, 0K][TC: 10][DOE: 0K, 0][CCI: 791K, 49152K, 2496K]
```

- OutOfMemoryError 発生時の出力

```
[OOM][Thread: 0x00957820]<Wed Mar 17 00:47:00.662 2004>
[java.lang.OutOfMemoryError:(C Heap):340]
```

- クラスライブラリトレースの出力

```
[CLT][Thread: 0x00286348]<Wed Mar 17 00:47:00.662 2004>
[CLT][Thread: 0x00286348] at java.lang.Shutdown.halt0(Native Method)
[CLT][Thread: 0x00286348] at java.lang.Shutdown.halt(Shutdown.java:145)
```

- JIT コンパイラー稼働継続機能 (JIT コンパイル失敗情報)

```
[JCC][Thread: 0x05432c00]<Thu Nov 15 17:10:40.347 2012>
[Method: chosa_cmp.func(Ljava/lang/String;)V][Fail: 3][JITCT: 1]
```

- JavaVM の標準出力の JavaVM ログファイルへの出力機能

```
[ST0]<Mon Jun 22 12:09:45.676 2015>VM option '+HitachiOutputMilliTime'
[ST0]<Mon Jun 22 12:09:45.676 2015>VM option '+PrintVMOptions'
[ST0]<Mon Jun 22 12:09:45.677 2015>VM option '+StandardLogToHitachiJavaLog'
[STE]<Mon Jun 22 12:09:45.677 2015>Invalid initial young generation size: -Xmn0k
```

7.6 拡張 verbosegc 機能で使用する JavaVM オプション

GC 発生時に出力する拡張 verbosegc 情報の出力を設定するためのオプションを説明しています。

7.6.1 -XX:[+|-]HitachiCommaVerboseGC

拡張 verbosegc 情報を CSV 形式で出力します。

説明

拡張 verbosegc 情報を CSV 形式で出力するかどうかを指定します。

前提オプション

- -XX:+HitachiVerboseGC

出力形式 (SerialGC の場合)

-XX:-HitachiVerboseGCIntervalTime オプション指定時の出力形式

```
id,date,full_count,copy_count,gc_kind,gc_info,gc_time,eden_info,
survivor_info,tenured_info,metaspace_info,classspace_info,cause_info,
user_cpu,system_cpu,jvm_alloc_size,mmap_total_size,malloc_total_size,
thread_count,doe_alloc_size,called_count,cc_used_size,cc_max_size,cc_info
```

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	JavaVM ログファイル識別子。
<i>date</i>	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。
<i>full_count</i>	FullGC 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。
<i>copy_count</i>	CopyGC 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。
<i>gc_kind</i>	GC 種別。FullGC または GC が出力されます。
<i>gc_info</i>	GC 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前の領域長,GC後の領域長,領域サイズ</i>
<i>gc_time</i>	GC 経過時間。単位は秒です。
<i>eden_info</i>	Eden 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前の領域長,GC後の領域長,領域サイズ</i>
<i>survivor_info</i>	Survivor 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前の領域長,GC後の領域長,領域サイズ</i>

出力項目	出力内容
<i>tenured_info</i>	Tenured 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前の領域長,GC後の領域長,領域サイズ</i>
<i>metaspace_info</i>	Metaspace 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前のMetaspace領域の使用サイズ,GC前のMetaspace領域のcapacityサイズ,GC前のMetaspace領域のcommitサイズ,GC後のMetaspace領域の使用サイズ,GC後のMetaspace領域のcapacityサイズ,GC後のMetaspace領域のcommitサイズ</i>
<i>classspace_info</i>	Compressed Class Space 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前のCompressed Class Spaceの使用サイズ,GC前のCompressed Class Spaceのcapacityサイズ,GC前のCompressed Class Spaceのcommitサイズ,GC後のCompressed Class Spaceの使用サイズ,GC後のCompressed Class Spaceのcapacityサイズ,GC後のCompressed Class Spaceのcommitサイズ</i> 圧縮オブジェクトポインター機能を使用していない場合、それぞれのサイズには0が出力されます。
<i>cause_info</i>	GC 要因番号。 -XX:-HitachiVerboseGCPrintCause オプションが指定された場合、出力されません。
<i>user_cpu</i>	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 時間の取得に失敗した場合は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。
<i>system_cpu</i>	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 時間の取得に失敗した場合は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。
<i>jvm_alloc_size</i>	JavaVM 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)。単位はキロバイトです。 -XX:-HitachiVerboseGCPrintJVMInternalMemory オプションが指定された場合、出力されません。
<i>mmap_total_size</i>	JavaVM 内部で管理している領域のうち、VirtualAlloc で割り当てた C ヒープの総サイズ。単位はキロバイトです。 -XX:-HitachiVerboseGCPrintJVMInternalMemory オプションが指定された場合、出力されません。
<i>malloc_total_size</i>	JavaVM 内部で管理している領域のうち、malloc で割り当てた C ヒープ総サイズ。単位はキロバイトです。 -XX:-HitachiVerboseGCPrintJVMInternalMemory オプションが指定された場合、出力されません。
<i>thread_count</i>	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。
<i>doe_alloc_size</i>	java.io.File.deleteOnExit() を呼び出して確保した累積のヒープサイズ。単位はキロバイトです。 -XX:-HitachiVerboseGCPrintDeleteOnExit オプションが指定された場合、出力されません。

出力項目	出力内容
<i>called_count</i>	java.io.File.deleteOnExit()の呼び出し回数。 -XX:-HitachiVerboseGCPrintDeleteOnExit が指定された場合、出力されません。
<i>cc_used_size</i>	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。
<i>cc_max_size</i>	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。
<i>cc_info</i>	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。

-XX:+HitachiVerboseGCPrintTenuringDistribution オプション指定時の出力形式

id, date, size, value, max_value, total_age1, total_age2, total_agen

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	PTD (JavaVM ログファイル識別子)。
<i>date</i>	GC 開始日時。
<i>size</i>	GC 後の Survivor 領域内オブジェクト目標サイズ。
<i>value</i>	次の CopyGC で Tenured 領域に昇格する Java オブジェクトの年齢のしきい値。 この値は、-XX:MaxTenuringThreshold= <i>value</i> オプションと、Survivor 領域のメモリーサイズ、および、-XX:TargetSurvivorRatio= <i>value</i> オプションに設定した値を基に、CopyGC ごとに動的に設定されます。 <i>value</i> 値以上の年齢の Java オブジェクトが、次の CopyGC で Tenured 領域に昇格します。
<i>max_value</i>	CopyGC で Tenured 領域に昇格する Java オブジェクトの年齢のしきい値 (<i>value</i> 値)の最大値 (MaxTenuringThreshold オプションの指定値)。 <i>value</i> 値は、CopyGC ごとに動的に設定されますが、 <i>max_value</i> 値を超えることはありません。 また、年齢が <i>max_value</i> 値以上の Java オブジェクトは、次の CopyGC で、必ず、Tenured 領域に昇格します。
<i>total_age1</i>	1 歳のオブジェクトのバイト数の合計。
<i>total_age2</i>	1 歳から 2 歳までのオブジェクトのバイト数の合計。
<i>total_agen</i>	1 歳から n 歳までのオブジェクトのバイト数の合計。 n が <i>max_value</i> に近ければ寿命の長いオブジェクトが存在するということになります。

出力形式 (G1GC の場合)

JavaVM ログファイルには、GC に関するログと Concurrent Marking に関するログが出力されます。Concurrent Marking は、アプリケーションと並行で処理を実行するため、1 回の Concurrent Marking に関するログは複数行に分かれて出力されます。

GC に関するログの出力形式

```
id,date,gc_kind,gc_info,gc_time,gc_status,eden_info,survivor_info,tenured_info,
humongous_info,free_info,metaspace_info,classspace_info,cause_info,
region_size,target_time,predicted_time,target_size,reclaimable_info,user_cpu,
system_cpu,jvm_alloc_size,mmap_total_size,malloc_total_size,thread_count,
doe_alloc_size,called_count,cc_used_size,cc_max_size,cc_info
```

Concurrent Marking に関するログの出力形式

```
id,date,cm_event,user_cpu,sys_spu
```

出力内容を次に説明します。

分類	出力項目	出力内容
GC に関するログ	<i>id</i>	VG1 (JavaVM ログファイル識別子)。
	<i>date</i>	GC の 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。
	<i>gc_kind</i>	GC または Concurrent Marking の種別。"FullGC"、"Mixed GC"、"Young GC"、"Young GC(initial-mark)"、"CM Remark"、"CM Cleanup"が出力されます。
	<i>gc_info</i>	Java ヒープ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。 <i>GC前のJavaヒープ領域の使用サイズ,GC前のJavaヒープ領域の使用サイズ※,GC前のJavaヒープ領域の領域サイズ※,GC後のJavaヒープ領域の使用サイズ,GC後のJavaヒープ領域の使用サイズ※,GC後のJavaヒープ領域の領域サイズ※</i>
	<i>gc_time</i>	GC によるアプリケーション停止時間。単位は秒です。
	<i>gc_status</i>	GC の状態。 <ul style="list-style-type: none"> To 空間あふれが発生した場合、"to exhausted"が出力されます。 上記以外の場合、 "-"が出力されます。 GC の種別ごとに、出力される GC の状態を次に示します。 <ul style="list-style-type: none"> <i>gc_kind</i> が"Young GC"、"Young GC(initial-mark)"または"Mixed GC"の場合 "-"または"to exhausted"が出力されます。 <i>gc_kind</i> が上記以外の場合 "-"が出力されます。
	<i>eden_info</i>	Eden 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。 <i>GC前のEden領域の使用サイズ※,GC前のEden領域が取得できる最大の領域サイズ※,GC後のEden領域の使用サイズ※,GC後のEden領域が取得できる最大の領域サイズ※</i> Eden 領域が取得できる最大の領域サイズは、「 <i>New領域の領域サイズ-Survivor領域の使用サイズ</i> 」を指します。
	<i>survivor_info</i>	Survivor 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。 <i>GC前のSurvivor領域の使用サイズ※,GC後のSurvivor領域の使用サイズ※</i>
<i>tenured_info</i>	Tenured 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。 <i>GC前のTenured領域の使用サイズ※,GC後のTenured領域の使用サイズ※</i>	

分類	出力項目	出力内容
	<i>humongous_info</i>	Humongous 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。 <i>GC前のHumongous領域の使用サイズ</i> ※, <i>GC後のHumongous領域の使用サイズ</i> ※
	<i>free_info</i>	Free 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。 <i>GC前のFree領域の使用サイズ</i> ※, <i>GC後のFree領域の使用サイズ</i> ※
	<i>metaspace_info</i>	Metaspace 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前のMetaspace領域の使用サイズ</i> , <i>GC前のMetaspace領域のcapacityサイズ</i> , <i>GC前のMetaspace領域のcommitサイズ</i> , <i>GC後のMetaspace領域の使用サイズ</i> , <i>GC後のMetaspace領域のcapacityサイズ</i> , <i>GC後のMetaspace領域のcommitサイズ</i>
	<i>classspace_info</i>	Compressed Class Space 情報。次の形式で出力されます。単位はキロバイトです。 <i>GC前のCompressed Class Spaceの使用サイズ</i> , <i>GC前のCompressed Class Spaceのcapacityサイズ</i> , <i>GC前のCompressed Class Spaceのcommitサイズ</i> , <i>GC後のCompressed Class Spaceの使用サイズ</i> , <i>GC後のCompressed Class Spaceのcapacityサイズ</i> , <i>GC後のCompressed Class Spaceのcommitサイズ</i> 圧縮オブジェクトポインター機能を使用していない場合、それぞれのサイズには 0 が出力されます。
	<i>cause_info</i>	GC 要因番号。 -XX:-HitachiVerboseGCPrintCause オプションが指定された場合、出力されません。
	<i>region_size</i>	1 リージョンのサイズ。単位はキロバイトです。
	<i>target_time</i>	GC によるアプリケーション停止時間の目標時間。単位は秒です。 目標時間は、-XX:+ MaxGCPauseMillis オプションで指定できます。
	<i>predicted_time</i>	JavaVM が予測した GC によるアプリケーション停止時間。単位は秒です。 GC の種別が"Full GC"、"CM Remark"または"CM Cleanup"の場合、JavaVM は予測しないため、0 が出力されます。
	<i>target_size</i>	Mixed GC で GC 対象となった Tenured 領域のサイズ。単位はキロバイトです。 GC の種別が"Mixed GC"以外の場合、0 が出力されます。
	<i>reclaimable_info</i>	予測回収サイズ情報。次の形式で出力されます。 <i>予測回収サイズ</i> , <i>予測回収率</i> 予測回収サイズの単位はキロバイトです。予測回収率は小数点以下第 2 位までが出力されます。 なお、予測回収サイズ情報は、Concurrent Marking 終了直後の Young GC または Mixed GC の場合に出力されます。ほかの場合、JavaVM は予測しないため、0 が出力されます。
	<i>user_cpu</i>	すべての 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。

分類	出力項目	出力内容
	<i>system_cpu</i>	すべての 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。
	<i>jvm_alloc_size</i>	JavaVM 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 (<i>mmap_total_size</i> と <i>malloc_total_size</i> 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。
	<i>mmap_total_size</i>	JavaVM 内部で管理している領域のうち、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。
	<i>malloc_total_size</i>	JavaVM 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。
	<i>thread_count</i>	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。
	<i>doe_alloc_size</i>	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。
	<i>called_count</i>	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。
	<i>cc_used_size</i>	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。
	<i>cc_max_size</i>	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。
	<i>cc_info</i>	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。
Concurrent Marking に関するログ	<i>id</i>	VCM (JavaVM ログファイル識別子)。
	<i>date</i>	Concurrent Marking 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。
	<i>cm_event</i>	Concurrent Marking の状態。 <ul style="list-style-type: none"> Concurrent Root Region Scan Start Concurrent Root Region Scan が開始されました。

分類	出力項目	出力内容
		<ul style="list-style-type: none"> • Concurrent Root Region Scan End Concurrent Root Region Scan が終了しました。 • Concurrent Mark Start Concurrent Mark が開始されました。 • Concurrent Mark End Concurrent Mark が終了しました。 • Concurrent Mark Stop Concurrent Mark を中止しました。 • Concurrent Cleanup Start Concurrent Cleanup が開始されました。 • Concurrent Cleanup End Concurrent Cleanup が終了しました。
	<i>user_cpu</i>	<p>すべての GC スレッドがユーザーモードで費やした CPU 時間の合計。単位は秒です。</p> <p>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。</p> <p>-XX:-HitachiVerboseGCCpuTime オプションが指定された場合、出力されません。</p> <p>Concurrent Marking の状態が"Concurrent Mark Start"の場合、0 が出力されません。</p>
	<i>sys_cpu</i>	<p>すべての GC スレッドがカーネルモードで費やした CPU 時間の合計。単位は秒です。</p> <p>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。</p> <p>-XX:-HitachiVerboseGCCpuTime オプションが指定された場合、出力されません。</p> <p>Concurrent Marking の状態が"Concurrent Mark Start"の場合、0 が出力されません。</p>

注※

サイズは、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となります。

cause_info に出力される GC 要因番号を次に示します。

表 7-3 GC 要因一覧

要因番号	要因内容	説明	GC の種類	
			G1GC 以外	G1GC
0	ObjAllocFail	オブジェクトの領域確保失敗によって GC が発生しました。	○	○
1	System.gc	java.lang.System.gc メソッド呼び出しによって GC が発生しました。	○	○
3	DelayedGC	JNI や JVMTI によって保留されていた GC が起動されました。	○	○
4	JavaGCCommand	javagc コマンドによって GC が発生しました。	○	○

要因番号	要因内容	説明	GCの種類	
			G1GC以外	G1GC
6	JHeapProfCommand	jheapprof コマンドによって GC が発生しました。	○	×
8	EMReclaimFail	Explicit メモリーの明示解放による Java ヒープへのオブジェクト移動が発生して Java ヒープがあふれました。	○	×
9	EMMigrateFail	Explicit メモリーの自動解放による Java ヒープへのオブジェクト移動が発生して Java ヒープがあふれました。	○	×
10	JVMTIForceGC	JVMTI 関数 ForceGarbageCollection()によって GC が発生しました。	○	○
11	PromotionFail	CopyGC の昇格失敗によって GC が発生しました。	○	×
12	EMJavaGCCommand	javagc コマンドによる Explicit メモリーブロックの解放によって GC が発生しました。	○	○
13	EHeapProfCommand	eheapprof コマンドによって GC が発生しました。	○	○
14	G1HumAllocFail	G1GC 使用時、Humongous 用の領域確保のために GC が発生しました。	×	○
15	G1EvacuationPause	G1GC 使用時にエバキュエーションが発生しました。	×	○
16	Concurrent Marking	GC は発生していないが、アプリケーションを停止して実行する Concurrent Marking 処理をログに出力しました。	×	○
17	EvacuationFail	G1GC 使用時にエバキュエーションの失敗による GC が発生しました。	×	○
18	MetaspaceAllocFail	Metaspace の領域確保失敗によって GC が発生しました。	○	○
19	LastMetaspaceGC	Metaspace の OutOfMemory を出す前の最後の GC が発生しました。	○	○

(凡例)

- ：出力します。
- ×：出力しません。

書式

```
-XX:[+|-]HitachiCommaVerboseGC
```

指定できる値

型：String

-XX:+HitachiCommaVerboseGC

拡張 verbosegc 情報の出力を、CSV ファイルで取得できるようにコンマ形式で出力します。

拡張 verbosegc 情報に出力される括弧（丸括弧()、角括弧[]、山括弧<>）およびコロン（:）をすべて削除し、コンマ（,）で区切った数値または文字列を出力します。

-XX:-HitachiCommaVerboseGC

拡張 verbosegc 情報を通常形式で出力します。

デフォルト値

定義項目の省略

-XX:-HitachiCommaVerboseGC

出力例

出力例 1

SerialGC の場合に-XX:HitachiVerboseGCIntervalTime オプションが指定されているとき

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 2

SerialGC の場合に-XX:+HitachiVerboseGCPrintTenuringDistribution オプションが指定されているとき

```
PTD, Wed May 28 11:45:23 2008, 5467547, 30, 31, 1357527, 1539661
```

出力例 3

G1GC の場合

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

7.6.2 -XX:[+|-]HitachiVerboseGC

GC が発生した場合、拡張 verbosegc 情報を JavaVM ログファイルに出力します。

説明

GC が発生した時の拡張 verbosegc 情報を出力するかどうかを指定します。

出力形式 (SerialGC の場合)

```
[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]
[GCI: cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	VGC (JavaVM ログファイル識別子)。
<i>date</i>	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。
<i>full_count</i>	FullGC 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。
<i>copy_count</i>	CopyGC 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。
<i>gc_kind</i>	GC 種別。"FullGC"、"GC"が出力されます。
<i>gc_info</i>	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)
<i>gc_time</i>	GC 経過時間。
<i>Eden</i>	Eden の種別。"DefNew::Eden"が出力されます。
<i>eden_info</i>	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)
<i>Survivor</i>	Survivor の種別。"DefNew::Survivor"が出力されます。
<i>survivor_info</i>	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)
<i>Tenured</i>	Tenured の種別。"Tenured"が出力されます。
<i>tenured_info</i>	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)
<i>Metaspace_info</i>	Metaspace 領域のメモリ情報。次の形式で出力されます。 GC前のMetaspace領域の使用サイズK(GC前のMetaspace領域のcapacityサイズK, GC前のMetaspace領域のcommitサイズK)->GC後のMetaspace領域の使用サイズK(GC後のMetaspace領域のcapacityサイズK, GC後のMetaspace領域のcommitサイズK)
<i>classspace_info</i>	Compressed Class Space のメモリ情報。次の形式で出力されます。 GC前のCompressed Class Spaceの使用サイズK(GC前のCompressed Class SpaceのcapacityサイズK, GC前のCompressed Class SpaceのcommitサイズK)->GC後のCompressed Class Spaceの使用サイズK(GC後のCompressed Class SpaceのcapacityサイズK, GC後のCompressed Class SpaceのcommitサイズK)

出力項目	出力内容
	圧縮オブジェクトポインター機能を使用していない場合、出力されません。
<i>cause_info</i>	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。
<i>user_cpu</i>	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。
<i>system_cpu</i>	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。
<i>jvm_alloc_size</i>	JavaVM 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 (<i>mmap_total_size</i> と <i>malloc_total_size</i> 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。
<i>mmap_total_size</i>	JavaVM 内部で管理している領域のうち、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。
<i>malloc_total_size</i>	JavaVM 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。
<i>thread_count</i>	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。
<i>doe_alloc_size</i>	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。
<i>called_count</i>	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。
<i>cc_used_size</i>	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。
<i>cc_max_size</i>	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。
<i>cc_info</i>	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)

JavaVM ログファイルには、GC に関するログと Concurrent Marking に関するログが出力されます。Concurrent Marking は、アプリケーションと並行で処理を実行するため、1 回の Concurrent Marking に関するログは複数行に分かれて出力されます。

GC に関するログの出力形式

```
[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][CGI: cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

Concurrent Marking に関するログの出力形式

```
[id]<date>[cm_event][User: user_cpu secs][Sys: sys_cpu secs]
```

出力内容を次に説明します。

分類	出力項目	出力内容
GC に関するログ	<i>id</i>	VG1 (JavaVM ログファイル識別子)。
	<i>date</i>	GC または Concurrent Marking の 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。
	<i>gc_kind</i>	GC または Concurrent Marking の種別。"FullGC"、"Mixed GC"、"Young GC"、"Young GC(initial-mark)"、"CM Remark"、"CM Cleanup"が出力されます。
	<i>gc_info</i>	Java ヒープ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。 <i>GC前のJavaヒープ領域の使用サイズK/GC前のJavaヒープ領域の使用サイズ**K(GC前のJavaヒープ領域の領域サイズ**K)->GC後のJavaヒープ領域の使用サイズK/GC後のJavaヒープ領域の使用サイズ**K(GC後のJavaヒープ領域の領域サイズ**K)</i>
	<i>gc_time</i>	GC によるアプリケーション停止時間。単位は秒です。
	<i>gc_status</i>	GC の状態。 <ul style="list-style-type: none"> To 空間あふれが発生した場合、"to exhausted"が出力されます。 上記以外の場合、 "-"が出力されます。 GC の種別ごとに、出力される GC の状態を次に示します。 <ul style="list-style-type: none"> <i>gc_kind</i> が"Young GC"、"Young GC(initial-mark)"または"Mixed GC"の場合 "-"または"to exhausted"が出力されます。 <i>gc_kind</i> が上記以外の場合 "-"が出力されます。
	<i>eden_info</i>	Eden 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。 <i>GC前のEden領域の使用サイズ**K(GC前のEden領域が取得できる最大の領域サイズ**K)->GC後のEden領域の使用サイズ**K(GC後のEden領域が取得できる最大の領域サイズ**K)</i> Eden 領域が取得でき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。
<i>survivor_info</i>	Survivor 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。	

分類	出力項目	出力内容
		<i>GC前のSurvivor領域の使用サイズ</i> *K-> <i>GC後のSurvivor領域の使用サイズ</i> *K
	<i>tenured_info</i>	Tenured 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。 <i>GC前のTenured領域の使用サイズ</i> *K-> <i>GC後のTenured領域の使用サイズ</i> *K
	<i>humongous_info</i>	Humongous 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。 <i>GC前のHumongous領域の使用サイズ</i> *K-> <i>GC後のHumongous領域の使用サイズ</i> *K
	<i>free_info</i>	Free 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。 <i>GC前のFree領域の使用サイズ</i> *K-> <i>GC後のFree領域の使用サイズ</i> *K
	<i>Metaspace_info</i>	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。 <i>GC前のMetaspace領域の使用サイズ</i> K(<i>GC前のMetaspace領域のcapacityサイズ</i> K, <i>GC前のMetaspace領域のcommitサイズ</i> K)-> <i>GC後のMetaspace領域の使用サイズ</i> K(<i>GC後のMetaspace領域のcapacityサイズ</i> K, <i>GC後のMetaspace領域のcommit</i> <i>サイズ</i> K)
	<i>classspace_info</i>	Compressed Class Space 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。 <i>GC前のCompressed Class Spaceの使用サイズ</i> K(<i>GC前のCompressed Class Space</i> <i>のcapacityサイズ</i> K, <i>GC前のCompressed Class Spaceのcommitサイズ</i> K)-> <i>GC後の</i> <i>Compressed Class Spaceの使用サイズ</i> K(<i>GC後のCompressed Class Spaceの</i> <i>capacityサイズ</i> K, <i>GC後のCompressed Class Spaceのcommitサイズ</i> K) 圧縮オブジェクトポインター機能を使用していない場合、出力されません。
	<i>cause_info</i>	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。
	<i>region_size</i>	1 リージョンのサイズ。単位はキロバイトです。
	<i>target_time</i>	GC によるアプリケーション停止時間の目標時間。単位は秒です。 目標時間は、-XX:+MaxGCPauseMillis オプションで指定できます。
	<i>predicted_time</i>	JavaVM が予測した GC によるアプリケーション停止時間。単位は秒です。 GC または Concurrent Marking の種別が"Full GC"、"CM Remark"または"CM Cleanup"の場合、JavaVM は予測しないため、0 が出力されます。
	<i>target_size</i>	Mixed GC で GC 対象となった Tenured 領域のサイズ。単位はキロバイトです。 GC または Concurrent Marking の種別が"Mixed GC"以外の場合、0 が出力され ます。
	<i>reclaimable_info</i>	予測回収サイズ情報。次の形式で出力されます。 <i>予測回収サイズ</i> K(<i>予測回収率</i> %) 予測回収サイズの単位はキロバイトです。予測回収率は小数点以下第 2 位までが 出力されます。 なお、予測回収サイズ情報は、Concurrent Marking 終了直後の Young GC ま たは Mixed GC の場合に出力されます。ほかの場合、JavaVM は予測しないた め、0 が出力されます。

分類	出力項目	出力内容
	<i>user_cpu</i>	すべての 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。
	<i>system_cpu</i>	すべての 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。
	<i>jvm_alloc_size</i>	JavaVM 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 (<i>mmap_total_size</i> と <i>malloc_total_size</i> 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。
	<i>mmap_total_size</i>	JavaVM 内部で管理している領域のうち、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。
	<i>malloc_total_size</i>	JavaVM 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。
	<i>thread_count</i>	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。
	<i>doe_alloc_size</i>	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。
	<i>called_count</i>	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。
	<i>cc_used_size</i>	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。
	<i>cc_max_size</i>	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。
	<i>cc_info</i>	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。
Concurrent Marking	<i>id</i>	VCM (JavaVM ログファイル識別子)。
	<i>date</i>	Concurrent Marking 開始日時。

分類	出力項目	出力内容
に関する ログ		-XX:-HitachiVerboseGCPrintDate オプションが指定された場合、出力されません。
	<i>cm_event</i>	Concurrent Marking の状態。 <ul style="list-style-type: none"> Concurrent Root Region Scan Start Concurrent Root Region Scan が開始されました。 Concurrent Root Region Scan End Concurrent Root Region Scan が終了しました。 Concurrent Mark Start Concurrent Mark が開始されました。 Concurrent Mark End Concurrent Mark が終了しました。 Concurrent Mark Stop Concurrent Mark を中止しました。 Concurrent Cleanup Start Concurrent Cleanup が開始されました。 Concurrent Cleanup End Concurrent Cleanup が終了しました。
	<i>user_cpu</i>	すべての 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。 Concurrent Marking の状態が"Concurrent Mark Start"の場合、0 が出力されません。
	<i>sys_cpu</i>	すべての 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。 Concurrent Marking の状態が"Concurrent Mark Start"の場合、0 が出力されません。

注※

サイズは、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となります。

cause_info に出力される GC 要因内容を次に示します。

表 7-4 GC 要因一覧

要因 番号	要因内容	説明	GC の種類	
			G1GC 以外	G1GC
0	ObjAllocFail	オブジェクトの領域確保失敗によって GC が発生しました。	○	○

要因番号	要因内容	説明	GCの種類	
			G1GC以外	G1GC
1	System.gc	java.lang.System.gc メソッド呼び出しによって GC が発生しました。	○	○
3	DelayedGC	JNI や JVMTI によって保留されていた GC が起動されました。	○	○
4	JavaGCCommand	javagc コマンドによって GC が発生しました。	○	○
6	JHeapProfCommand	jheapprof コマンドによって GC が発生しました。	○	×
8	EMReclaimFail	Explicit メモリーの明示解放による Java ヒープへのオブジェクト移動が発生して Java ヒープがあふれました。	○	×
9	EMMigrateFail	Explicit メモリーの自動解放による Java ヒープへのオブジェクト移動が発生して Java ヒープがあふれました。	○	×
10	JVMTIForceGC	JVMTI 関数 ForceGarbageCollection() によって GC が発生しました。	○	○
11	PromotionFail	CopyGC の昇格失敗によって GC が発生しました。	○	×
12	EMJavaGCCommand	javagc コマンドによる Explicit メモリーブロックの解放によって GC が発生しました。	○	○
13	EHeapProfCommand	eheapprof コマンドによって GC が発生しました。	○	○
14	G1HumAllocFail	G1GC 使用時、Humongous 用の領域確保のために GC が発生しました。	×	○
15	G1EvacuationPause	G1GC 使用時にエバキュエーションが発生しました。	×	○
16	Concurrent Marking	GC は発生していないが、アプリケーションを停止して実行する Concurrent Marking 処理をログに出力しました。	×	○
17	EvacuationFail	G1GC 使用時にエバキュエーションの失敗による GC が発生しました。	×	○
18	MetaspaceAllocFail	Metaspace の領域確保失敗によって GC が発生しました。	○	○
19	LastMetaspaceGC	Metaspace の OutOfMemory を出す前の最後の GC が発生しました。	○	○

(凡例)

○：出力します。

×：出力しません。

書式

```
-XX:[+|-]HitachiVerboseGC
```

指定できる値

型：String

-XX:+HitachiVerboseGC

GCが発生した場合、拡張 verbosegc 情報を JavaVM ログファイルに出力します。

GCの内部領域である Eden、Survivor、Tenured、Metaspace 種別の情報を拡張 verbosegc 情報として出力します。

-XX:-HitachiVerboseGC

GCが発生した場合、拡張 verbosegc 情報を JavaVM ログファイルに出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiVerboseGC

出力例

出力例 1

SerialGC の場合に-XX:HitachiVerboseGCIntervalTime オプションが指定されているとき

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2

G1GC の場合

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

7.6.3 -XX:[+|-]HitachiVerboseGCCpuTime

拡張 verbosegc 情報にユーザー CPU 時間とカーネル CPU 時間を出力します。

説明

拡張 verbosegc 情報に CPU 利用時間を出力するかどうかを指定します。

前提オプション

- -XX:+HitachiVerboseGC

書式

```
-XX:[+|-]HitachiVerboseGCCpuTime
```

指定できる値

型: String

-XX:+HitachiVerboseGCCpuTime

拡張 verbosegc 情報に、GC の開始から終了までで、GC 実行スレッドのユーザーモードおよびカーネルモードに費やされたプロセッサ時間を出力します。

-XX:-HitachiVerboseGCCpuTime

拡張 verbosegc 情報に、GC の開始から終了までで、GC 実行スレッドのユーザーモードおよびカーネルモードに費やされたプロセッサ時間を出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiVerboseGCCpuTime

出力例

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

7.6.4 -XX:HitachiVerboseGCIntervalTime

拡張 verbosegc 情報を出力する時間 (秒) の間隔を指定します。

説明

拡張 verbosegc 情報を出力する時間 (秒) の間隔を指定します。

オプションを指定した場合

拡張 verbosegc 情報を GC ごとに出力するのではなく、指定された時間を超えた次の GC を出力します。このとき、前回の出力から今回の出力までの間に発生した、次に示す GC の回数も出力します。これらは必ず出力されます。

表 7-5 回数を出力する GC の一覧

文字列	意味
Full	FullGC をスキップした回数
Copy	CopyGC をスキップした回数

オプションを指定しない場合

0 秒がデフォルトで設定され、GC 発生ごとに拡張 verbosegc 情報を出力します。

前提オプション

- -XX:+HitachiVerboseGC

書式

```
-XX:HitachiVerboseGCIntervalTime=整数値
```

指定できる値

整数値

型: Integer

0~2147483647 の範囲の整数値 (単位: 秒) を指定します。範囲外の値が指定された場合は 0 が設定されます。負の値を指定した場合はエラーとなります。

デフォルト値

定義項目の省略

```
-XX:HitachiVerboseGCIntervalTime=0
```

出力例

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

7.6.5 -XX:[+|-]HitachiVerboseGCPrintCause

GC の要因内容を出力します。

説明

GC の要因内容を出力するかどうかを指定します。

前提オプション

- -XX:+HitachiVerboseGC

-XX:+HitachiCommaVerboseGC オプションが指定されている場合は、次に示す要因番号が出力されます。

表 7-6 GC 要因一覧

要因 番号	要因内容	説明	GC の種類	
			G1GC 以外	G1GC
0	ObjAllocFail	オブジェクトの領域確保失敗によって GC が発生しました。	○	○
1	System.gc	java.lang.System.gc メソッド呼び出しによって GC が発生しました。	○	○
3	DelayedGC	JNI や JVMTI によって保留されていた GC が起動されました。	○	○
4	JavaGCCommand	javagc コマンドによって GC が発生しました。	○	○
6	JHeapProfCommand	jheapprof コマンドによって GC が発生しました。	○	×
8	EMReclaimFail	Explicit メモリーの明示解放による Java ヒープへのオブジェクト移動が発生して Java ヒープがあふれました。	○	×
9	EMMigrateFail	Explicit メモリーの自動解放による Java ヒープへのオブジェクト移動が発生して Java ヒープがあふれました。	○	×
10	JVMTIForceGC	JVMTI 関数 ForceGarbageCollection()によって GC が発生しました。	○	○
11	PromotionFail	CopyGC の昇格失敗によって GC が発生しました。	○	×
12	EMJavaGCCommand	javagc コマンドによる Explicit メモリーブロックの解放によって GC が発生しました。	○	○
13	EHeapProfCommand	eheapprof コマンドによって GC が発生しました。	○	○
14	G1HumAllocFail	G1GC 使用時、Humongous 用の領域確保のために GC が発生しました。	×	○

要因番号	要因内容	説明	GCの種類	
			G1GC以外	G1GC
15	G1EvacuationPause	G1GC 使用時にエバキュエーションが発生しました。	×	○
16	Concurrent Marking	GC は発生していないが、アプリケーションを停止して実行する Concurrent Marking 処理をログに出力しました。	×	○
17	EvacuationFail	G1GC 使用時にエバキュエーションの失敗による GC が発生しました。	×	○
18	MetaspaceAllocFail	Metaspace の領域確保失敗によって GC が発生しました。	○	○
19	LastMetaspaceGC	Metaspace の OutOfMemory を出す前の最後の GC が発生しました。	○	○

(凡例)

○：出力します。

×：出力しません。

書式

```
-XX:[+|-]HitachiVerboseGCPrintCause
```

指定できる値

型：String

-XX:+HitachiVerboseGCPrintCause

GC の要因内容を、拡張 verbosegc 情報の行末に出力します。

-XX:-HitachiVerboseGCPrintCause

拡張 verbosegc 情報を通常形式で出力します。

デフォルト値

定義項目の省略

-XX:+HitachiVerboseGCPrintCause

出力例

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
```

```
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

7.6.6 -XX:[+|-]HitachiVerboseGCPrintDate

拡張 verbosegc 情報に GC 開始日時を出力します。

説明

拡張 verbosegc 情報に GC 開始日時を出力するかどうかを指定します。

前提オプション

- -XX:+HitachiVerboseGC

書式

```
-XX:[+|-]HitachiVerboseGCPrintDate
```

指定できる値

型: String

-XX:+HitachiVerboseGCPrintDate

拡張 verbosegc 情報の各出力行の先頭に GC 開始日時を出力します。

-XX:-HitachiVerboseGCPrintDate

拡張 verbosegc 情報の各出力行の先頭に GC 開始日時を出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiVerboseGCPrintDate

出力例

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

7.6.7 -XX:[+|-]HitachiVerboseGCPrintDeleteOnExit

java.io.File.deleteOnExit() を呼び出したことによって JavaVM が確保した累積のヒープサイズとメソッドの呼び出し回数を、JavaVM ログファイルへ出力します。

説明

java.io.File.deleteOnExit() を呼び出したことによって JavaVM が確保した累積のヒープサイズとメソッドの呼び出し回数を、JavaVM ログファイルに出力するかどうかを指定します。

java.io.File.deleteOnExit() は、呼び出すたびに指定されたファイルのパス情報をヒープに確保しますが、確保した領域はプロセスの終了まで解放しないため、メモリーの圧迫につながるおそれがあります。-XX:+HitachiVerboseGCPrintDeleteOnExit オプションを指定すると、JavaVM が java.io.File.deleteOnExit() を呼び出して確保したヒープサイズをログに出力して監視できるようになります。また、java.io.File.deleteOnExit() の呼び出し状況を把握するための補助的な情報として、メソッドの呼び出し回数も同時に出力できます。

出力した情報は、障害発生時、java.io.File.deleteOnExit()の呼び出しによって確保されたヒープサイズを把握してメモリー不足の原因を調査するために役立てられます。また、運用開始前の開発やテスト段階で、java.io.File.deleteOnExit()の呼び出しによって確保したヒープサイズの増加の推移を確認して、運用時にメモリーを圧迫する予兆がないかを事前確認するためにも利用できます。

なお、エラーが発生した場合は、エラーメッセージが出力されます。

前提オプション

- -XX:+HitachiVerboseGC

出力形式

JavaVM ログファイルの出力形式を次に示します。

```
[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]
```

このオプションによって出力される項目と出力内容を次に示します。

出力項目	出力内容
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。
called_count	java.io.File.deleteOnExit()の呼び出し回数。

エラーが発生した場合のエラーメッセージの出力形式を次に示します。

```
[DOE]<date>Error occurred during processing of java.io.File.deleteOnExit's
heap size output function. (<保守情報>)
[DOE]java.io.File.deleteOnExit's heap size output function stopped.
```

エラーメッセージの出力内容を次に示します。

出力項目	出力内容
DOE	java.io.File.deleteOnExit()のヒープサイズ出力機能でエラーが発生したことを示す識別子。
date	エラーが発生した日時。

書式

```
-XX:[+|-]HitachiVerboseGCPrintDeleteOnExit
```

指定できる値

型: String

-XX:+HitachiVerboseGCPrintDeleteOnExit

java.io.File.deleteOnExit() を呼び出して確保した累積のヒープサイズと、メソッドの呼び出し回数を出力します。

-XX:-HitachiVerboseGCPrintDeleteOnExit

java.io.File.deleteOnExit() を呼び出して確保した累積のヒープサイズと、メソッドの呼び出し回数を出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiVerboseGCPrintDeleteOnExit

出力例

JavaVM ログファイルの出力例を次に示します。

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

エラーメッセージの出力例を次に示します。

出力例

```
[DOE]<Wed Jan 27 13:03:36 2010> Error occurred during processing of
java.io.File.deleteOnExit's heap size output function.
(FindClass:java.lang.String)
[DOE]java.io.File.deleteOnExit's heap size output function stopped.
```

注意事項

- 次の場合は、java.io.File.deleteOnExit()を呼び出しても累積のヒープサイズ、メソッドの呼び出し回数がカウントされません。
 - java.io.File.deleteOnExit()を呼び出した場合に SecurityException 例外が発生したとき（この例外はセキュリティーマネージャーの SecurityManager.checkDelete()がファイルへの削除アクセスを許可しない場合に発生します。この場合、メソッドの入り口で例外が挙がり、ヒープは確保されません）。
 - Application Server のバッチアプリケーション実行基盤で作成されたアプリケーションから呼び出した場合。
 - 同一のパス名文字列で作成した File インスタンスを使用して java.io.File.deleteOnExit()を呼び出したとき。
- この機能が出力するヒープサイズを確認する際には、次の点に注意してください。
 - java.io.File.deleteOnExit()が確保するヒープの種類は Java ヒープです。
 - ヒープサイズはキロバイト単位で出力され、1 キロバイト未満は切り捨てられます。java.io.File.deleteOnExit()の 1 回の呼び出しで確保するヒープサイズは、ファイルパスの長さに応じて数十バイトから 100 バイト程度であるため、呼び出しごとにヒープサイズの出力結果が増加しないことがあります。この場合、メソッドの呼び出し回数からメソッドの実行を確認できます。

7.6.8 -XX:[+|-]HitachiVerboseGCPrintJVMinernalMemory

JavaVM 内部で管理しているヒープ情報を JavaVM ログファイルへ出力します。

説明

JavaVM 内部で管理しているヒープ情報を JavaVM ログファイルに出力するかどうかを指定します。

C ヒープ領域のうち、次の 2 種類の方法で取得したヒープ領域は、JavaVM 内部で管理する方式で管理されています。

- mmap で取得した C ヒープ領域
- malloc で取得した C ヒープ領域

-XX:+HitachiVerboseGCPrintJVMinernalMemory オプションを有効にした場合、mmap で取得した C ヒープの総サイズ (*mmap_total_size*) と、malloc で取得した C ヒープの総サイズ (*malloc_total_size*) を出力できます。また、これらの割り当て済みの領域のうち、使用中の領域のサイズの合計値 (*jvm_alloc_size*) も出力できます。

前提オプション

- -XX:+HitachiVerboseGC

出力形式

```
[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]
```

このオプションによって出力される項目と出力内容を次に示します。

出力項目	出力内容
<i>jvm_alloc_size</i>	JavaVM 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 (<i>mmap_total_size</i> と <i>malloc_total_size</i> の合計サイズのうち、現在使用中の領域のサイズ)。
<i>mmap_total_size</i>	JavaVM 内部で管理している領域のうち、VirtualAlloc で割り当てた C ヒープの総サイズ。
<i>malloc_total_size</i>	JavaVM 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。

書式

```
-XX:[+|-]HitachiVerboseGCPrintJVMinernalMemory
```

指定できる値

型: String

-XX:+HitachiVerboseGCPrintJVMInternalMemory

JavaVM 内部で管理しているヒープ情報を JavaVM ログファイルへ出力します。

-XX:-HitachiVerboseGCPrintJVMInternalMemory

JavaVM 内部で管理しているヒープ情報を JavaVM ログファイルへ出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiVerboseGCPrintJVMInternalMemory

出力例

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```


出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,  
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,  
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,  
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000  
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

注意事項

このオプションを有効にする場合は、拡張 VerboseGC 機能のほかに独自のメモリー管理機能が有効になっている必要があります。

7.6.9 -XX:[+|-]HitachiVerboseGCPrintTenuringDistribution

Survivor 領域の年齢分布を JavaVM ログファイルへ出力します。

説明

-XX:+PrintTenuringDistribution 指定時に出力される Survivor 領域の年齢分布を JavaVM ログファイルへ出力するかどうかを指定します。

前提オプション

- -XX:+HitachiVerboseGC

関連オプション

- -XX:+PrintTenuringDistribution
- -XX:+HitachiVerboseGCPrintDate
- -XX:+HitachiCommaVerboseGC

出力形式

```
[id]<date>[Desired survivor:size bytes][New threshold:value]  
[MaxTenuringThreshold: max_value][age1:total_age1][age2:total_age2]....  
[agen:total_agen]
```

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	PTD (JavaVM ログファイル識別子)。
<i>date</i>	GC 開始日時。

出力項目	出力内容
<i>size</i>	GC 後の Survivor 領域内オブジェクト目標サイズ。
<i>value</i>	次の CopyGC で Tenured 領域に昇格する Java オブジェクトの年齢のしきい値。 この値は、 <code>-XX:MaxTenuringThreshold=value</code> オプションと、Survivor 領域のメモリーサイズ、および、 <code>-XX:TargetSurvivorRatio=value</code> オプションに設定した値を基に、CopyGC ごとに動的に設定されます。 <i>value</i> 値以上の年齢の Java オブジェクトが、次の CopyGC で Tenured 領域に昇格します。
<i>max_value</i>	CopyGC で Tenured 領域に昇格する Java オブジェクトの年齢のしきい値 (<i>value</i> 値) の最大値 (<code>MaxTenuringThreshold</code> オプションの指定値)。 <i>value</i> 値は、CopyGC ごとに動的に設定されますが、 <i>max_value</i> 値を超えることはありません。 また、年齢が <i>max_value</i> 値以上の Java オブジェクトは、次の CopyGC で、必ず、Tenured 領域に昇格します。
<i>total_age1</i>	1 歳のオブジェクトのバイト数の合計。
<i>total_age2</i>	1 歳から 2 歳までのオブジェクトのバイト数の合計。
<i>total_agen</i>	1 歳から n 歳までのオブジェクトのバイト数の合計。 n が <i>max_value</i> に近ければ寿命の長いオブジェクトが存在するというようになります。

書式

```
-XX:[+|-]HitachiVerboseGCPrintTenuringDistribution
```

指定できる値

型：String

`-XX:+HitachiVerboseGCPrintTenuringDistribution`

`-XX:+PrintTenuringDistribution` 指定時に出力される Survivor 領域の年齢分布を JavaVM ログファイルへ出力します。

`-XX:-HitachiVerboseGCPrintTenuringDistribution`

`-XX:+PrintTenuringDistribution` 指定時に出力される Survivor 領域の年齢分布を JavaVM ログファイルへ出力しません。

デフォルト値

定義項目の省略

`-XX:-HitachiVerboseGCPrintTenuringDistribution`

出力例

```
[PTD]<Wed Jan 28 17:47:10 2009>[Desired survivor:32768 bytes][New threshold:30]
[MaxTenuringThreshold:30][age1:6872][age2:9632][age3:25632]
```

注意事項

このオプションを指定した場合、CopyGC 発生時のログの出力量が、指定しない場合と比較して倍以上に増えます。

そのため、このオプションはチューニングの時にだけ使用してください。

7.6.10 -XX:[+|-]HitachiVerboseGCPrintThreadCount

Java スレッドの数を JavaVM ログファイルへ出力します。

説明

Java スレッドの数を監視するために、Java スレッドの数を JavaVM ログファイルに出力するかどうかを指定します。

スレッドは、スタックに使用するためのメモリーを C ヒープとして個々で確保します。このため、スレッド数が多くなると、C ヒープの確保量もスレッド数に比例して多くなります。-XX:
+HitachiVerboseGCPrintThreadCount オプションを指定することで、Java スレッドの数を監視できるようになり、C ヒープの確保量の把握ができるようになります。

なお、スレッド数が OS で設定するスレッド数の上限値に達した場合、それ以上スレッドを作成できないことがあります。

前提オプション

- -XX:+HitachiVerboseGC

出力形式

```
[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]
```

このオプションによって出力される項目と出力内容を次に示します。

出力項目	出力内容
<i>thread_count</i>	Java スレッドの数。

書式

```
-XX:[+|-]HitachiVerboseGCPrintThreadCount
```

指定できる値

型 : String

-XX:+HitachiVerboseGCPrintThreadCount

Java スレッドの数を出力します。

-XX:-HitachiVerboseGCPrintThreadCount

Java スレッドの数を出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiVerboseGCPrintThreadCount

出力例

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

7.7 コードキャッシュ領域情報出力機能で使用する JavaVM オプション

コードキャッシュ領域情報の出力を設定するためのオプションを説明しています。

7.7.1 -XX:CodeCacheInfoPrintRatio

コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージを出力する契機となる、コードキャッシュ領域の使用率を指定します。

説明

コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージを出力する契機となる、コードキャッシュ領域の使用率を指定します。

ここで指定した使用率を基に、次の計算式でしきい値が計算されます。

コードキャッシュ領域の最大サイズ×-XX:CodeCacheInfoPrintRatio オプションの値÷100

コードキャッシュ領域は、JIT コンパイルによって生成された JIT コンパイルコードが格納される領域です。JavaVM は、呼び出し回数やループ回数が多い Java メソッドを JIT コンパイルして実行することで、処理の高速化を行います。

コードキャッシュ領域の最大サイズは、ReservedCodeCacheSize オプションに指定します。

ReservedCodeCacheSize オプションには、デフォルト値以上の値を指定してください。

また、コードキャッシュ領域が枯渇していた場合、または枯渇するおそれがある場合は、コードキャッシュ領域の拡張を検討してください。JIT コンパイルコードのサイズは計算で見積もることができません。そのため、Java アプリケーション実行環境でコードキャッシュ領域の使用量を実測し、システムが使用するコードキャッシュ領域（32 ビット版で最大 500 キロバイト、64 ビット版で最大 2 メガバイト）の使用量を考慮した上で、コードキャッシュ領域の最大サイズを見積もってください。

前提オプション

- -XX:+PrintCodeCacheInfo

書式

```
-XX:CodeCacheInfoPrintRatio=整数値
```

指定できる値

整数値

型：Integer

0～100 の範囲で整数値（単位：%）を指定します。範囲外の値が指定された場合は 80 が設定されます。

デフォルト値

定義項目の省略

`-XX:CodeCacheInfoPrintRatio=80`

注意事項

- コードキャッシュ領域の使用量がしきい値に達した状態で推移している場合、Java メソッドが JIT コンパイルされてもメッセージは出力されません。
一方、コードキャッシュ領域の使用量がしきい値未満まで下がったあと、Java メソッドが JIT コンパイルされたことによって再びコードキャッシュ領域の使用量がしきい値に達した場合には、メッセージが出力されます。
- システムが使用するコードキャッシュ領域は 32 ビット版で最大 500 キロバイト、64 ビット版で最大 2 メガバイトです。そのため、システムのコードキャッシュ領域の使用量によっては、コードキャッシュ領域を最大サイズまで使用していない場合でも、コードキャッシュ領域が枯渇することがあります。また、コードキャッシュ領域の使用量のしきい値に大きな値を指定している場合、メッセージを出力する前にコードキャッシュ領域が枯渇する場合があります。コードキャッシュ領域が枯渇する前にメッセージを出力したいときは、「コードキャッシュ領域の最大サイズーしきい値」の値が 4 メガバイト以上になるように、`-XX:CodeCacheInfoPrintRatio` (コードキャッシュ領域使用率指定オプション) の値を指定してください。

7.7.2 -XX:[+|-]PrintCodeCacheFullMessage

Java メソッドが JIT コンパイルの対象になった場合、コードキャッシュ領域が枯渇していたときに、メッセージを JavaVM ログファイルへ出力します。

説明

Java メソッドが JIT コンパイルの対象になった場合、コードキャッシュ領域が枯渇していたときに、メッセージを JavaVM ログファイルに出力するかどうかを指定します。コードキャッシュ領域は、JIT コンパイルによって生成された JIT コンパイルコードが格納される領域です。JavaVM は、呼び出し回数やループ回数が多い Java メソッドを JIT コンパイルして実行することで、処理の高速化を行います。

前提オプション

- `-XX:+HitachiVerboseGC`

出力形式

メッセージの出力形式を次に示します。

```
[cc_id]<cc_date>CodeCache is full. Compiler has been disabled.  
[cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

メッセージの出力内容を次に示します。

出力項目	出力内容
<i>cc_id</i>	CCI (JavaVM ログファイル識別子)。
<i>cc_date</i>	Java メソッドが JIT コンパイルの対象になった日時。
<i>cc_used_size</i>	Java メソッドが JIT コンパイルの対象になったとき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。
<i>cc_max_size</i>	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。
<i>cc_info</i>	保守情報。

書式

```
-XX:[+|-]PrintCodeCacheFullMessage
```

指定できる値

型: String

-XX:+PrintCodeCacheFullMessage

Java メソッドが JIT コンパイルの対象になった場合、コードキャッシュ領域が枯渇していたときにメッセージを出力します。メッセージは一度だけ出力されます。

-XX:-PrintCodeCacheFullMessage

Java メソッドが JIT コンパイルの対象になった場合、コードキャッシュ領域が枯渇してもメッセージを出力しません。

デフォルト値

定義項目の省略

-XX:+PrintCodeCacheFullMessage

出力例

メッセージの出力例を次に示します。

```
[CCI]<Wed Dec 26 14:38:29 2012>CodeCache is full. Compiler has been disabled.
[49151K, 49152K, 49152K]
```

注意事項

システムが使用するコードキャッシュ領域は 32 ビット版で最大 500 キロバイト、64 ビット版で最大 2 メガバイトです。そのため、システムのコードキャッシュ領域の使用量によっては、コードキャッシュ領域を最大サイズまで使用していない場合でも、コードキャッシュ領域が枯渇することがあります。

7.7.3 -XX:[+|-]PrintCodeCacheInfo

コードキャッシュ領域の使用量を出力するかどうかを指定します。また、使用量がしきい値に達したことを知らせるメッセージを JavaVM ログファイルに出力するかどうかを指定します。

説明

コードキャッシュ領域の使用量を出力するかどうかを指定します。また、使用量がしきい値に達したことを知らせるメッセージを JavaVM ログファイルに出力するかどうかを指定します。コードキャッシュ領域は、JIT コンパイルによって生成された JIT コンパイルコードが格納される領域です。JavaVM は、呼び出し回数やループ回数が多い Java メソッドを JIT コンパイルして実行することで、処理の高速化を行います。

このオプションを有効にすると、GC 発生時、コードキャッシュ領域の使用量が拡張 verbosegc 情報に出力されます。また、コードキャッシュ領域の使用量がしきい値に達したとき、メッセージが出力されます。

しきい値は「コードキャッシュ領域の最大サイズ×-XX:CodeCacheInfoPrintRatio オプションの値÷100」です。コードキャッシュ領域の最大サイズは、-XX:ReservedCodeCacheSize オプションに指定します。

-XX:-HitachiVerboseGC オプションを指定している場合でも、javagc コマンドの-v オプションや-s オプションによって出力する拡張 verbosegc 情報には、コードキャッシュ領域の使用量が出力されます。

前提オプション

- -XX:+HitachiVerboseGC

関連オプション

- -XX:+HitachiCommaVerboseGC
- -XX:CodeCacheInfoPrintRatio

出力形式

コードキャッシュ領域の使用量の出力形式を次に示します。

```
[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]
```

このオプションによって出力される項目と出力内容を次に示します。

出力項目	出力内容
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。
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。
cc_info	保守情報。

コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージの出力形式を次に示します。

```
[cc_id]<cc_date>CodeCache usage has exceeded the threshold.  
[cc_used_sizeK, cc_max_sizeK, cc_infoK]
```

メッセージの出力内容を次に示します。

出力項目	出力内容
cc_id	CCI (JavaVM ログファイル識別子)。
cc_date	JIT コンパイルを実行した日時。
cc_used_size	JIT コンパイル後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。
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。
cc_info	保守情報。

書式

```
-XX:[+|-]PrintCodeCacheInfo
```

指定できる値

型 : String

-XX:+PrintCodeCacheInfo

コードキャッシュ領域の使用量を出力します。

また、コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージを出力します。

-XX:-PrintCodeCacheInfo

コードキャッシュ領域の使用量を出力しません。

また、コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージを出力しません。

デフォルト値

定義項目の省略

-XX:+PrintCodeCacheInfo

出力例

コードキャッシュ領域の使用量の出力例を次に示します。

出力例 1 (-XX:+HitachiVerboseGC オプションの SerialGC の場合)

```
[VGC]<Thu Oct 02 10:38:53.658 2014>(Skip Full:1, Copy:0)  
[Full GC 770K->682K(8064K), 0.0050003 secs][DefNew::Eden: 88K->0K(2304K)]  
[DefNew::Survivor: 0K->0K(256K)][Tenured: 681K->682K(5504K)]  
[Metaspace: 3634K(4492K, 4492K)->3634K(4492K, 4492K)]
```

```
[class space: 356K(388K, 388K)->356K(388K, 388K)]
[cause:System.gc][User: 0.0000000 secs][Sys: 0.0000000 secs]
[IM: 11944K, 12448K, 0K][TC: 22][DOE: 0K, 0][CCI: 1173K, 245760K, 2496K]
```

出力例 2 (-XX:+HitachiVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
[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]
```

- Concurrent Marking に関するログ

```
[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]
```

出力例 3 (-XX:+HitachiCommaVerboseGC オプションの SerialGC の場合)

```
VGC, Thu Oct 02 10:38:52.442 2014, 1, 0, 0, Full GC, 770, 682, 8064, 0.0040002, 88, 0, 2304,
0, 0, 256, 681, 682, 5504, 3634, 3634, 4492, 3634, 3634, 4492, 356, 356, 388, 356, 356, 388, 1, 0.
0000000, 0.0000000, 11913, 12448, 0, 22, 0, 0, 1173, 245760, 2496
```

出力例 4 (-XX:+HitachiCommaVerboseGC オプションの G1GC の場合)

- GC に関するログ

```
VG1, Thu Oct 02 10:38:54.920 2014, Full GC, 753, 2048, 8192, 678, 1024, 8192, 0.0064767,
-, 1024, 2048, 0, 2048, 0, 0, 1024, 1024, 0, 0, 6144, 7168, 3634, 3634, 4492, 3634, 3634, 4492,
356, 356, 388, 356, 356, 388, 1, 1024, 0.2000000, 0.0000000, 0, 0, 0.00, 0.0000000,
0.0000000, 20459, 21920, 0, 35, 0, 0, 1171, 245760, 2496
```

- Concurrent Marking に関するログ

```
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark Start, 0.0000000, 0.0000000
VCM, Fri Jul 26 21:35:50 2013, Concurrent Mark End, 0.0124532, 0.0245698
```

コードキャッシュ領域の使用量がしきい値に達したことを知らせるメッセージの出力例を次に示します。

出力例

```
[CCI]<Wed Dec 26 14:27:53 2012>CodeCache usage has exceeded the threshold.
[39358K, 49152K, 39360K]
```

注意事項

- コードキャッシュ領域の使用量がしきい値に達した状態で推移している場合、Java メソッドが JIT コンパイルされてもメッセージは出力されません。
一方、コードキャッシュ領域の使用量がしきい値未満まで下がったあと、Java メソッドが JIT コンパイルされたことによって再びコードキャッシュ領域の使用量がしきい値に達した場合には、メッセージが出力されます。
- システムが使用するコードキャッシュ領域は 32 ビット版で最大 500 キロバイト、64 ビット版で最大 2 メガバイトです。そのため、システムのコードキャッシュ領域の使用量によっては、コードキャッシュ領域を最大サイズまで使用していない場合でも、コードキャッシュ領域が枯渇することがあります。また、コードキャッシュ領域の使用量のしきい値に大きな値を指定している場合、メッセージを出力する前にコードキャッシュ領域が枯渇する場合があります。コードキャッシュ領域が枯渇する前にメッセージを出力したいときは、「コードキャッシュ領域の最大サイズーしきい値」の値が 4 メガバイト以上になるように、`-XX:CodeCacheInfoPrintRatio` (コードキャッシュ領域使用率指定オプション) の値を指定してください。

7.8 OutOfMemoryError 発生時の拡張機能で使用する JavaVM オプション

OutOfMemoryError 発生時の出力情報や動作に関する設定をするためのオプションを説明しています。

7.8.1 -XX:[+|-]HitachiOutOfMemoryAbort

OutOfMemoryError 発生時に強制終了する機能の有効または無効を設定します。

説明

次の原因で OutOfMemoryError が発生した場合、標準出力にメッセージを出力して強制終了します。

- Java ヒープ不足の場合
- Metaspace 不足の場合
- Compressed Class Space 不足の場合
- J2SE クラスライブラリー中での C ヒープ不足の場合

なお、JavaVM 処理中に C ヒープ不足となった場合には、このオプションの指定に関係なく、強制停止します。

強制終了時の JavaVM の終了コード

1

書式

```
-XX:[+|-]HitachiOutOfMemoryAbort
```

指定できる値

型：String

-XX:+HitachiOutOfMemoryAbort

OutOfMemoryError 発生時に強制終了します。

-XX:-HitachiOutOfMemoryAbort

OutOfMemoryError 発生時に強制終了しません。

デフォルト値

定義項目の省略

-XX:-HitachiOutOfMemoryAbort

出力例

```
java.lang.OutOfMemoryError occurred.  
JavaVM aborted because of specified -XX:+HitachiOutOfMemoryAbort options.
```

注意事項

- `-XX:+HitachiOutOfMemoryStackTrace` オプションが指定されている場合は、JavaVM ログファイルにスタックトレースを出力したあとに終了します。
- このオプションが指定されている場合、`java.io.File.deleteOnExit` メソッドや `java.lang.Runtime.addShutdownHook` メソッドで登録している、JavaVM 終了時の処理は実行されずに強制終了します。

7.8.2 -XX:[+|-]HitachiOutOfMemoryAbortThreadDump

OutOfMemoryError 発生時にスレッドダンプを出力します。

説明

OutOfMemoryError 発生時にスレッドダンプを出力します。ただし、J2SE クラスライブラリーで C ヒープ不足の場合は、スレッドダンプの出力による再度の C ヒープ不足発生を避けるため、スレッドダンプは出力しません。

スレッドダンプの出力先は環境変数 `JAVACOREDIR`、または `-XX:+HitachiThreadDumpToStdout` オプションで指定します。

前提オプション

- `-XX:+HitachiOutOfMemoryAbort`
- `-XX:+HitachiThreadDump`

書式

```
-XX:[+|-]HitachiOutOfMemoryAbortThreadDump
```

指定できる値

型：String

`-XX:+HitachiOutOfMemoryAbortThreadDump`

OutOfMemoryError 発生時にスレッドダンプを出力します。

`-XX:+HitachiOutOfMemoryAbort` オプションが指定されている場合に、このオプションは指定できます。

-XX:-HitachiOutOfMemoryAbortThreadDump

OutOfMemoryError 発生時にスレッドダンプを出力しません。

デフォルト値

定義項目の省略

-XX:+HitachiOutOfMemoryAbortThreadDump

7.8.3 -XX:

[+|-]HitachiOutOfMemoryAbortThreadDumpWithJHeapProf

OutOfMemoryError 発生時に出力するスレッドダンプログファイルに、クラス別統計情報を出力します。

説明

-XX:+HitachiOutOfMemoryAbortThreadDump で出力するスレッドダンプログファイルにクラス別統計情報を出力するかどうかを指定します。

なお、G1GC を使用している場合、この機能は使用できません。G1GC 使用時にこの機能を有効にするオプションを指定した場合、指定内容は反映されなくて、この機能は無効となります。

前提オプション

- -XX:+HitachiOutOfMemoryAbort
- -XX:+HitachiOutOfMemoryAbortThreadDump
- -XX:+HitachiThreadDump

書式

```
-XX:[+|-]HitachiOutOfMemoryAbortThreadDumpWithJHeapProf
```

指定できる値

型: String

-XX:+HitachiOutOfMemoryAbortThreadDumpWithJHeapProf

-XX:+HitachiOutOfMemoryAbortThreadDump で出力するスレッドダンプログファイルにクラス別統計情報を出力します。

-XX:-HitachiOutOfMemoryAbortThreadDumpWithJHeapProf

-XX:+HitachiOutOfMemoryAbortThreadDump で出力するスレッドダンプログファイルにクラス別統計情報を出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiOutOfMemoryAbortThreadDumpWithJHeapProf

出力例

```
Java Heap Profile
-----
      Size  Instances  Class
-----
      856560      5 [Ljava.io.ObjectStreamField;
      113488      2 java.lang.ThreadGroup
      113280      2 [Ljava.lang.Thread;
      108264      2 java.io.PrintStream
...
      64          1 classA
...
      32          1 classB
...
      16          1 classD
...
      16          1 classC
...
```

7.8.4 -XX:[+|-]HitachiOutOfMemoryCause

OutOfMemoryError 発生時の発生要因種別を出力します。

説明

OutOfMemoryError 発生時の発生要因種別を出力するかどうかを指定します。

出力される要因種別を次に示します。

表 7-7 例外発生要因種別一覧

項番	要因メッセージ	説明
1	C Heap	C ヒープ確保時の例外
2	Java Heap	Java ヒープ確保時の例外
3	Meta Space	Metaspace 確保時の例外
4	Compressed Class Space	Compressed Class Space 確保時の例外
5	Unknown	上記例外を特定できない場合
6	Thread Limit	スレッド数の上限値設定機能 (-XX:HitachiThreadLimit オプション) を指定した場合に、作成したスレッド数が指定した上限値を超えたときの例外

書式

```
-XX:[+|-]HitachiOutOfMemoryCause
```

指定できる値

型 : String

-XX:+HitachiOutOfMemoryCause

OutOfMemoryError 発生時に、例外発生要因種別を JavaVM ログファイルに出力します。

-XX:+HitachiOutOfMemoryStackTrace オプションが指定されている場合は、このオプションも設定されます。

-XX:-HitachiOutOfMemoryCause

OutOfMemoryError 発生時に、例外発生要因種別を JavaVM ログファイルに出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiOutOfMemoryCause

出力例

```
[OOM][Thread: 0x00062fd0]<Tue Dec 2 16:42:39 2003>[java.lang.OutOfMemoryError  
:(C Heap) : unable to create thread : 340 threads exist]
```

7.8.5 -XX:[+|-]HitachiOutOfMemorySize

OutOfMemoryError 発生時に要求したメモリーのサイズを出力します。

説明

OutOfMemoryError 発生時に要求したメモリーのサイズを出力します。

書式

```
-XX:[+|-]HitachiOutOfMemorySize
```

指定できる値

型 : String

-XX:+HitachiOutOfMemorySize

OutOfMemoryError 発生時に、要求したメモリーサイズをバイト単位で出力します。

-XX:+HitachiOutOfMemoryStackTrace オプションが指定されている場合は、このオプションも設定されます。

-XX:-HitachiOutOfMemorySize

OutOfMemoryError 発生時に、要求したメモリーサイズを出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiOutOfMemorySize

出力例

```
[OOM][Thread: 0x00062fd0]<Tue Dec 2 16:42:39 2003>
[java.lang.OutOfMemoryError : requested 1024 bytes. (Java Heap) :
20 threads exist]
```

注意事項

次の場合、要求したメモリーサイズが取り出せません。

- Java のメモリー確保機能を使用しなくて、明示的に OutOfMemoryError をスローした (J2SE クラスライブラリーがスローするものを含まず) 場合。

(例)

```
throw new OutOfMemoryError();
```

- クラスロード時の verifier が OutOfMemoryError を発生させた場合。

```
[OOM][Thread: 0x00062fd0]<Tue Dec 2 16:42:39 2003>
[java.lang.OutOfMemoryError : requested size unknown. (Unknown) :
10 threads exist]
```

7.8.6 -XX:[+|-]HitachiOutOfMemoryStackTrace

OutOfMemoryError 発生時に、例外情報とスタックトレースを JavaVM ログファイルへ出力します。

説明

OutOfMemoryError 発生時に、例外情報とスタックトレースを JavaVM ログファイルに出力するかどうかを指定します。

出力形式

```
[id] [Thread:thread_id]<date>[java.lang.OutOfMemoryError :
requested size bytes (cause) : reason : thread_count threads exist]
[id] [Thread:thread_id] stack_trace
```

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	00M (JavaVM ログファイル識別子)。
<i>thread_id</i>	スレッド ID (スレッドダンプに出力されている tid)。
<i>date</i>	OutOfMemory 例外発生日時。
<i>size</i>	<p>確保しようとしたメモリのサイズ (単位: バイト) が出力されます。-XX:-HitachiOutOfMemorySize オプションが指定されている場合、出力されません。</p> <p>なお、次の場合は要求したメモリサイズが取り出せません。これらの場合は、サイズとして「unknown」が出力されます。</p> <ul style="list-style-type: none"> Java のメモリ確保機能を使用しないで、明示的に OutOfMemoryError をスローした場合 標準クラスライブラリーによってスローされるものも含まれます。例えば、「throw new OutOfMemoryError();」などの処理によってスローされた場合、メモリサイズは出力できません。 クラスロード時の verifier によって OutOfMemoryError が発生した場合
<i>cause</i>	例外発生要因種別*。ただし、-XX:-HitachiOutOfMemoryCause オプションが指定されている場合は出力されません。
<i>reason</i>	例外発生理由。スレッドの作成に失敗した場合に出力されます。
<i>thread_count</i>	OutOfMemoryError 発生時のスレッド数。作成に失敗したスレッド数も含まれます。
<i>stack_trace</i>	スタックトレース

注※

例外発生要因種別を次に示します。

表 7-8 例外発生要因種別一覧

項番	要因メッセージ	説明
1	C Heap	C ヒープ確保時の例外
2	Java Heap	Java ヒープ確保時の例外
3	Meta Space	Metaspace 確保時の例外
4	Compressed Class Space	Compressed Class Space 確保時の例外
5	Unknown	上記例外を特定できない場合
6	Thread Limit	スレッド数の上限値設定機能 (-XX:HitachiThreadLimit オプション) を指定した場合に、作成したスレッド数が指定した上限値を超えたときの例外

書式

```
-XX:[+|-]HitachiOutOfMemoryStackTrace
```

指定できる値

型 : String

-XX:+HitachiOutOfMemoryStackTrace

OutOfMemoryError 発生時に、例外情報とスタックトレースを JavaVM ログファイルに出力します。スタックトレースは 1 スタックごとにバッファに格納し、コード変換したあとに出力します。スタックトレースの出力は、OutOfMemoryError がスローされるたびに行われるため、OutOfMemoryError をキャッチして再スローした場合には複数回出力されます。なお、スレッド作成時に OutOfMemoryError となった場合は、スタックトレースは出力されません。

-XX:-HitachiOutOfMemoryStackTrace

OutOfMemoryError 発生時に、スタックトレースを JavaVM ログファイルに出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiOutOfMemoryStackTrace

出力例

```
[OOM][Thread: 0x00062fd0] <Wed Mar 17 00:41:17 2004>
[java.lang.OutOfMemoryError :requested 400000 bytes. (C Heap):
unable to create thread : 1500 threads exist]
[OOM][Thread: 0x00062fd0] at java.lang.Thread.start(Native Method)
[OOM][Thread: 0x00062fd0] at sub1.<init>(Thread0012.java:22)
[OOM][Thread: 0x00062fd0] at Thread0012.test01(Thread0012.java:73)
[OOM][Thread: 0x00062fd0] at Thread0012.main(Thread0012.java:57)
```

注意事項

JavaVM 自身で作成するスレッドが、メモリー不足によってスレッドの作成に失敗した場合は、例外情報だけが出力されます。スタックトレースは出力されません。

7.8.7 -XX:HitachiOutOfMemoryStackTraceLineSize

OutOfMemoryError 発生時に、出力するスタックトレース 1 行の文字数をバイト数で指定します。

説明

OutOfMemoryError 発生時に、出力するスタックトレース 1 行の文字数をバイト数で指定します。オプションの指定がない場合は、1024 バイトがデフォルトで設定されます。指定したバイト数が確保できない場合は警告メッセージが出力され、スタックトレースは出力されません。また、1 行の文字数が指定した文字数を超えた場合、「at」以降の文字列の前半部分を削除して、指定された文字数分出力します。

なお、`-XX:+JavaLogAsynchronous` オプションを指定している場合、このオプションには4096 が設定されます。

前提オプション

- `-XX:+HitachiOutOfMemoryStackTrace`

書式

```
-XX:HitachiOutOfMemoryStackTraceLineSize=整数値
```

指定できる値

整数値

型：Integer

1024～2147483647 の範囲で整数値（単位：バイト）を指定します。範囲外の値が指定された場合は1024 が設定されます。負の値を指定した場合はエラーとなります。

デフォルト値

定義項目の省略

```
-XX:HitachiOutOfMemoryStackTraceLineSize=1024
```

7.9 クラスライブラリトレース機能で使用する JavaVM オプション

クラスライブラリのスタックトレースの出力情報に関する設定をするためのオプションを説明しています。

7.9.1 -XX:[+|-]HitachiJavaClassLibTrace

クラスライブラリのスタックトレースを出力します。

説明

次に示すシステム全体に影響を与えるメソッドが使用された場合、これらの API のスタックトレースを、JavaVM ログファイルへ出力します。

- java.lang.System.gc
- java.lang.System.exit
- java.lang.System.runFinalizersOnExit
- java.lang.Runtime.exit
- java.lang.Runtime.halt
- java.lang.Runtime.runFinalizersOnExit

出力形式

```
[id] [Thread:thread_id]<date>  
[id] [Thread:thread_id] stack_trace
```

出力項目	出力内容
<i>id</i>	CLT (JavaVM ログファイル識別子)。
<i>thread_id</i>	スレッド ID (スレッドダンプに出力されている tid)。
<i>date</i>	クラスライブラリ使用日時。
<i>stack_trace</i>	スタックトレース。

書式

```
-XX:[+|-]HitachiJavaClassLibTrace
```

指定できる値

型 : String

-XX:+HitachiJavaClassLibTrace

クラスライブラリーのスタックトレースを出力します。

-XX:-HitachiJavaClassLibTrace

クラスライブラリーのスタックトレースを出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiJavaClassLibTrace

出力例

出力例 1

```
[CLT][Thread: 0x00286348]<Thu Oct 21 14:55:50 2004>
[CLT][Thread: 0x00286348] at java.lang.Shutdown.halt0(Native Method)
[CLT][Thread: 0x00286348] at java.lang.Shutdown.halt(Shutdown.java:145)
[CLT][Thread: 0x00286348] at java.lang.Shutdown.exit(Shutdown.java:222)
[CLT][Thread: 0x00286348] at java.lang.Runtime.exit(Runtime.java:90)
[CLT][Thread: 0x00286348] at java.lang.System.exit(System.java:715)
[CLT][Thread: 0x00286348] at J250048_02.main(J250048_02.java:4)
```

出力例 2

```
[CLT][Thread: 0x009c4000]<Tue Oct 09 15:36:18 2012>
[CLT][Thread: 0x009c4000] at java.lang.Runtime.outputJavaClassLibTrace
(Native Method)
[CLT][Thread: 0x009c4000] at java.lang.Runtime.runFinalizersOnExit
(Runtime.java:378)
[CLT][Thread: 0x009c4000] at java.lang.System.runFinalizersOnExit
(System.java:978)
[CLT][Thread: 0x009c4000] at Program.main(Program.java:8)
```

7.9.2 -XX:HitachiJavaClassLibTraceLineSize

クラスライブラリーのスタックトレースの 1 行の文字数をバイト数で指定します。

説明

-XX:+HitachiJavaClassLibTrace オプションを指定した場合に出力される、クラスライブラリーのスタックトレースの 1 行の文字数をバイト数で指定します。

なお、-XX:+JavaLogAsynchronous オプションを指定している場合、このオプションには4096 が設定されません。

前提オプション

- -XX:+HitachiJavaClassLibTrace

書式

```
-XX:HitachiJavaClassLibTraceLineSize=整数値
```

指定できる値

整数値

型：Integer

1024～2147483647 の範囲で整数値（単位：バイト）を指定します。範囲外の値が指定された場合は1024が設定されます。

デフォルト値

定義項目の省略

```
-XX:HitachiJavaClassLibTraceLineSize=1024
```

出力例

```
[CLT][Thread: 0x00286c58]<Thu Oct 21 14:56:24 2004>  
[CLT][Thread: 0x00286c58] at java.lang.Runtime.gc(Native Method)  
[CLT][Thread: 0x00286c58] at java.lang.System.gc(System.java:737)  
[CLT][Thread: 0x00286c58] at mple.func_012345678～省略～xyz(Sample.java:9)  
[CLT][Thread: 0x00286c58] at Sample.main(Sample.java:5)
```

1行の文字数が指定したバイト数を超える場合、「at」以降の文字列の前半部分が削除されます。出力例の場合、4行目のSampleの前半が削除されてmpleとなります。

7.10 ローカル変数情報出力機能で使用する JavaVM オプション

ローカル変数情報の出力を設定するためのオプションを説明しています。

7.10.1 -XX:HitachiCallToString

該当するクラスのローカル変数オブジェクトに対して取得した String オブジェクトの文字列を、変数値を表現する文字列として出力します。

説明

*適用範囲*に該当するクラスのローカル変数オブジェクトに対して取得した String オブジェクトの文字列を、変数値を表現する文字列として出力します。

なお、ローカル変数に格納されているオブジェクトがない場合、または JavaVM 内部のオブジェクトの場合は、出力されません。

前提オプション

- -XX:+HitachiLocalsInThrowable

書式

```
-XX:HitachiCallToString=適用範囲
```

指定できる値

適用範囲

型 : String

minimal を指定します。

java.lang パッケージ内の次に示すクラスが対象になります。

- String
- StringBuffer
- Boolean
- Byte
- Character
- Short
- Integer
- Long
- Float

- Double

なお、空文字 ("") を指定した場合も、minimal と同じです。

デフォルト値

定義項目の省略

```
-XX:HitachiCallToString=minimal
```

出力例

次の Java プログラム例を使用した出力例（簡易出力フォーマット）を、次に示します。

Java プログラム例

```
class Example3 {
    public static void main(String[] args) {
        Example3 e3 = new Example3();
        e3.method(); // 4行目
    }

    void method() {
        String l1 = "local 1";
        StringBuffer l2 = new StringBuffer(l1);
        l2.append(" + local 2");
        Boolean l3 = new Boolean(false);
        Character l4 = new Character('X');
        Long l5 = new Long(Long.MIN_VALUE); // long型の最小値
        Object l6 = new Thread();
        Object[] l7 = new Thread[10];

        try {
            <例外発生!> // 18行目
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public String toString() {
        return "I am an Example3 instance.";
    }
}
```

出力例 1

-XX:HitachiCallToString=minimal の場合

```
at Example3.method(Example3.java:18)
locals:
  (Example3) this = <0xaa07db58>
  (java.lang.String) l1 = <0xae173a28> "local 1"
  (java.lang.StringBuffer) l2 = <0xaa07dca0> "local 1 + local 2"
  (java.lang.Boolean) l3 = <0xaa07de18> "false"
  (java.lang.Character) l4 = <0xaa07df68> "X"
  (java.lang.Long) l5 = <0xaa07e078> "-9223372036854775808"
  (java.lang.Object) l6 = <0xaa07e1a8>
```

```
(java.lang.Object[]) l7 = <0xaa07e298>
at Example3.main(Example3.java:4)
locals:
...
```

注意事項

ローカル変数情報を完全に収集するためには、`javac` で class ファイルを生成する際に、`-g` オプションまたは `-g:vars` オプションを付加して class ファイル内にローカル変数情報を埋め込んでおく必要があります。`-g` オプションまたは `-g:vars` オプションを付加しないで作成された class ファイルについては、収集可能な範囲でローカル変数情報が出力されます。

7.10.2 -XX:[+|-]HitachiLocalsInStackTrace

スレッドダンプ出力時のスタックトレースに、各メソッドのローカル変数情報を追加して出力します。

説明

スレッドダンプ出力時のスタックトレースに、各メソッドのローカル変数情報を追加して出力します。

書式

```
-XX:[+|-]HitachiLocalsInStackTrace
```

指定できる値

型: String

`-XX:+HitachiLocalsInStackTrace`

スレッドダンプ出力時のスタックトレースに、ローカル変数情報を出力します。

`-XX:-HitachiLocalsInStackTrace`

スレッドダンプ出力時のスタックトレースに、ローカル変数情報を出力しません。

デフォルト値

定義項目の省略

`-XX:-HitachiLocalsInStackTrace`

出力例

次の Java プログラム例を使用した出力例を、次に示します。

Java プログラム例

```
class Example2 {
    public static void main(String[] args) {
        Example2 e2 = new Example2();
        e2.method(); // 4行目
    }

    synchronized void method() {
        int l1 = 1;
        float l2 = 2.0f;
        String l3 = "local 3";
        Character l4 = new Character('X');
        Object l5 = new Thread();
        Object[] l6 = new Thread[10];

        <ここでスレッドダンプ出力!> // 15行目
    }
}
```

出力例

-XX:+HitachiLocalsSimpleFormat オプションおよび-XX:+HitachiTrueTypeInLocals オプションが指定されている場合

```
"main" #1 prio=5 os_prio=0 jid=<N/A> tid=0x000000000297800 nid=0x11500
runnable [0x0000000002a8f000..0x0000000002a8f6c0]
  java.lang.Thread.State: RUNNABLE
  stack=
[0x0000000002a90000..0x0000000002994000..0x0000000002991000..0x0000000002990000]
 [user cpu time=3790ms, kernel cpu time=78ms] [blocked count=0, waited count=0]
 at Example2.method(Unknown Source)
- locked <0x00000000120d03a8> (a Example2)
  locals:
    (Example2) this = <0x00000000120d03a8>
    (int) l1 = 1
    (float) l2 = 2
    (java.lang.String) l3 = <0x00000000120d03b8>
    (java.lang.Character) l4 = <0x00000000120d0600>
    (java.lang.Object) l5 = <0x00000000120d07f8>
    (java.lang.Object[]) l6 = <0x00000000120d0bc8>
 at Example2.main(Unknown Source)
  locals:
    (java.lang.String[]) args [arg1] = <0x00000000120d0390>
    (Example2) e2 = <0x00000000120d03a8>
```

注意事項

- ローカル変数情報を完全に収集するためには、javac で class ファイルを生成する際に、-g オプションまたは-g:vars オプションを付加して class ファイル内にローカル変数情報を埋め込んでおく必要があります。-g オプションまたは-g:vars オプションを付加しないで作成された class ファイルについては、収集可能な範囲でローカル変数情報が出力されます。

- 一般にスタックトレース情報の収集を行うスレッドと収集対象のスレッドは一致しません。このため、情報収集を行うためには、対象スレッドを停止させる必要があります、toString メソッドを呼び出すことはできません。このため、-XX:HitachiCallToString オプションの指定は無効になります。
- 制御構造が複雑で行数が多いメソッドのローカル変数情報を出力する場合、解析に時間が掛かるため、拡張スレッドダンプの出力やスレッドスタックトレースの取得に時間が掛かることがあります。

7.10.3 -XX:[+|-]HitachiLocalsInThrowable

スタックトレース中のメソッドのローカル変数情報を出力します。

説明

java.lang.Throwable.fillInStackTrace メソッド実行時に、スタックトレース中のメソッドのローカル変数情報を収集します。

出力形式

```

locals:
  name: name
  type: type
  value: value
...

```

- 1 行目に「locals:」という見出しが出力されます。
- 2 行目以降は、収集できた各ローカル変数の情報が 1 行ずつ出力されます。

出力されるローカル変数の情報を次に示します。

1. 変数名
2. 型名（基本型名、クラス名または配列型名）
3. 変数値を表現する文字列

なお、各ローカル変数の出力内容は、空行で区切られています。

出力項目	出力内容
<i>name</i>	ローカル変数名。 メソッドに渡される引数の場合は、変数名に続いて[<i>arg***</i>] (***)は引数番号) が表示されます。
<i>type</i>	ローカル変数の型名（基本型名、クラス名または配列型名）。
<i>value</i>	ローカル変数の値を表現する文字列。 <ul style="list-style-type: none"> • 基本型 値をそのまま文字列化したもの • クラスまたは配列型 変数値がnull の場合：(null)

出力項目	出力内容
	<p>それ以外の場合：オブジェクトの存在するアドレス</p> <p>値表現の最大文字列長は 64 文字です。これを超える場合は 64 文字目までを出力したあと、「...」という文字列が出力されます。クラスまたは配列型の場合、次の追加オプションを指定することでより詳細な表現が追加できます。</p> <ul style="list-style-type: none"> • -XX:+HitachiLocalsSimpleFormat • -XX:+HitachiTrueTypeInLocals • -XX:HitachiCallToString

書式

```
-XX:[+|-]HitachiLocalsInThrowable
```

指定できる値

型：String

-XX:+HitachiLocalsInThrowable

スタックトレース中のメソッドのローカル変数情報を出力します。

ただし、java.lang.StackOverflowError 発生時には、このオプションは無視されます。

-XX:-HitachiLocalsInThrowable

スタックトレース中のメソッドのローカル変数情報を出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiLocalsInThrowable

出力例

次の Java プログラム例を使用した出力例を、次に示します。

Java プログラム例

```
class Example1 {
    public static void main(String[] args) {
        Example1 e1 = new Example1();
        Object obj = new Object();
        e1.method(1, 'Q', obj); // 5行目
    }

    void method(int l1, char l2, Object l3) {
        float l4 = 4.0f;
        boolean l5 = true;
        double l6 = Double.MAX_VALUE; // double型の最大値
        Object[] l7 = new Object[10];
    }
}
```

```

    try {
        <例外発生!> // 15行目
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

出力例 1

すべてのローカル変数情報が出力される場合

```
at Example1.method(Example1.java:15)
```

```
locals:
```

```
name: this
type: Example1
value: <0x922f42d0>
```

```
name: l1 [arg1]
type: int
value: 1
```

```
name: l2 [arg2]
type: char
value: 'Q'
```

```
name: l3 [arg3]
type: java.lang.Object
value: <0xaf112f08>
```

```
name: l4
type: float
value: 4
```

```
name: l5
type: boolean
value: true
```

```
name: l6
type: double
value: 1.79769E+308
```

```
name: l7
type: java.lang.Object[]
value: <0x922f42d8>
```

```
at Example1.main(Example1.java:5)
```

```
locals:
```

```
...
```

出力例 2

ローカル変数情報が存在しない場合

- `-g` オプションまたは `-g:vars` オプションを付加しないで class ファイルを生成した場合

- -g オプションまたは-g:vars オプションを付加して生成した class ファイルの、native メソッドの場合

```

at Example1.method(Example1.java:15)
  locals:
    name: this
    type: Example1
    value: <0x922f42d0>

    name: [arg1]
    type: int
    value: 1

    name: [arg2]
    type; char
    value: 'Q'

    name: [arg3]
    type: java.lang.Object
    value: <0xaf112f08>

at Example1.main(Example1.java:5)
  locals:
...

```

注意事項

- ローカル変数情報を完全に収集するためには、javac で class ファイルを生成する際に、-g オプションまたは-g:vars オプションを付加して class ファイル内にローカル変数情報を埋め込んでおく必要があります。-g オプションまたは-g:vars オプションを付加しないで作成された class ファイルについては、収集可能な範囲でローカル変数情報が出力されます。
- -g オプションまたは-g:vars オプションを付加して生成された class ファイルでも、native メソッドの場合はローカル変数情報が存在しません。
- JIT コンパイラーがメソッドを JIT コンパイルする際、最適化の一環として、不要と判断したローカル変数を除去することがあります。

(例) int not_used = 12345 などの宣言および初期化以降未使用のローカル変数

この場合、例外発生時点のローカル変数情報には、次の値が出力されます。

型名	出力情報
boolean 型	false
char 型	'¥0'
byte 型	0
short 型	
int 型	
long 型	
float 型	

型名	出力情報
double 型	
クラス型 配列型	(null)

- 制御構造が複雑で行数が多いメソッドのローカル変数情報を出力する場合、解析に時間が掛かるため、例外発生時の例外オブジェクト生成処理に時間が掛かることがあります。
- java.lang.Thread クラスの getStackTrace メソッドを使用して取得した、カレントスレッドのスタックトレースにローカル変数情報を出力するためには、例外発生時のスタックトレースにローカル変数情報を出力する、例外発生時のローカル変数情報収集オプション (-XX:+HitachiLocalsInThrowable) が必要になります。

7.10.4 -XX:[+|-]HitachiLocalsSimpleFormat

ローカル変数情報の出力フォーマットを、1 変数 1 行で出力する簡易出力フォーマットに変更します。

説明

ローカル変数情報の出力フォーマットを、1 変数 1 行で出力する簡易出力フォーマットに変更します。

前提オプション

- -XX:+HitachiLocalsInThrowable
- -XX:+HitachiLocalsInStackTrace

出力形式

```

locals:
  (type) name = value
  (type) name = value
...

```

出力内容を次に示します。

出力項目	出力内容
<i>name</i>	ローカル変数名。 メソッドに渡される引数の場合は、変数名に続いて[arg***] (**は引数番号) が表示されます。
<i>type</i>	ローカル変数の型名 (基本型名、クラス名または配列型名)。
<i>value</i>	ローカル変数の値を表現する文字列。 <ul style="list-style-type: none"> • 基本型 値をそのまま文字列化したもの • クラスまたは配列型

出力項目	出力内容
	変数値がnullの場合：(null) それ以外の場合：オブジェクトの存在するアドレス 値表現の最大文字列長は64文字です。これを超える場合は64文字目までを出力したあと、「...」という文字列が出力されます。クラスまたは配列型の場合、次の追加オプションを指定することでより詳細な表現が追加できます。 <ul style="list-style-type: none"> • -XX:+HitachiLocalsSimpleFormat • -XX:+HitachiTrueTypeInLocals • -XX:HitachiCallToString

書式

```
-XX:[+|-]HitachiLocalsSimpleFormat
```

指定できる値

型：String

-XX:+HitachiLocalsSimpleFormat

ローカル変数情報出力を、簡易フォーマットで出力します。

-XX:-HitachiLocalsSimpleFormat

ローカル変数情報出力を、通常フォーマットで出力します。

デフォルト値

定義項目の省略

-XX:-HitachiLocalsSimpleFormat

出力例

次のJavaプログラム例を使用した出力例を、次に示します。

Javaプログラム例

```
class Example1 {
    public static void main(String[] args) {
        Example1 e1 = new Example1();
        Object obj = new Object();
        e1.method(1, 'Q', obj); // 5行目
    }

    void method(int l1, char l2, Object l3) {
        float l4 = 4.0f;
        boolean l5 = true;
        double l6 = Double.MAX_VALUE; // double型の最大値
        Object[] l7 = new Object[10];

        try {
```

```

        <例外発生!> // 15行目
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

出力例

```

at Example1.method(Example1.java:15)
  locals:
    (Example1) this = <0x922f42d0>
    (int) l1 [arg1] = 1
    (char) l2 [arg2] = 'Q'
    (java.lang.Object) l3 [arg3] = <0xaf112f08>
    (float) l4 = 4
    (boolean) l5 = true
    (double) l6 = 1.79769E+308
    (java.lang.Object[]) l7 = <0x922f42d8>
at Example1.main(Example1.java:5)
  locals:
...

```

7.10.5 -XX:[+|-]HitachiTrueTypeInLocals

ローカル変数情報に、実際のオブジェクト型名を出力します。

説明

ローカル変数情報収集時に、クラスまたは配列型のローカル変数について、実際に代入されているオブジェクトの型名を出力します。変数値を表現する文字列の後ろの丸括弧内に表示されます。

なお、ローカル変数に格納されているクラスまたは配列型のオブジェクトが JavaVM 内部のオブジェクトの場合は、"internal type"と出力されます。

前提オプション

- -XX:+HitachiLocalsInThrowable
- -XX:+HitachiLocalsInStackTrace

書式

```
-XX:[+|-]HitachiTrueTypeInLocals
```

指定できる値

型 : String

-XX:+HitachiTrueTypeInLocals

ローカル変数情報に、実際のオブジェクト型名を出力します。

-XX:-HitachiTrueTypeInLocals

ローカル変数情報に、実際のオブジェクト型名を出力しません。

デフォルト値

定義項目の省略

-XX:-HitachiTrueTypeInLocals

出力例

次の Java プログラム例を使用した出力例を、次に示します。

Java プログラム例

```
class Example3 {
    public static void main(String[] args) {
        Example3 e3 = new Example3();
        e3.method(); // 4行目
    }

    void method() {
        String l1 = "local 1";
        StringBuffer l2 = new StringBuffer(l1);
        l2.append(" + local 2");
        Boolean l3 = new Boolean(false);
        Character l4 = new Character('X');
        Long l5 = new Long(Long.MIN_VALUE); // long型の最小値
        Object l6 = new Thread();
        Object[] l7 = new Thread[10];

        try {
            <例外発生!> // 18行目
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public String toString() {
        return "I am an Example3 instance.";
    }
}
```

出力例

-XX:+HitachiLocalsSimpleFormat オプションおよび-XX:HitachiCallToString=full が指定されている場合

```
at Example3.method(Example3.java:18)
  locals:
    (Example3) this = <0xaa07db58> "I am an Example3 instance." (Example3)
    (java.lang.String) l1 = <0xae173a28> "local 1" (java.lang.String)
```

```
(java.lang.StringBuffer) l2 = <0xaa07dca0> "local 1 + local 2"  
(java.lang.StringBuffer)  
(java.lang.Boolean) l3 = <0xaa07de18> "false" (java.lang.Boolean)  
(java.lang.Character) l4 = <0xaa07df68> "X" (java.lang.Character)  
(java.lang.Long) l5 = <0xaa07e078> "-9223372036854775808" (java.lang.Long)  
(java.lang.Object) l6 = <0xaa07e1a8> "Thread[Thread-0, 5, main]"  
(java.lang.Thread)  
(java.lang.Object[]) l7 = <0xaa07e298> "[Ljava.lang.Thread;@26e431"  
(java.lang.Thread[])  
at Example3.main(Example3.java:4)  
  locals:  
...
```

7.11 明示管理ヒープ機能で使用する JavaVM オプション

FullGC の発生を抑止する機能（明示管理ヒープ機能）を使用するためのオプションを説明しています。

7.11.1 -XX:ExplicitMemoryAutoReclaimTargetBlockRatio

明示管理ヒープ機能の自動解放処理で設定している解放率の上限値を指定します。

説明

このオプションは、Explicit ヒープのチューニング、および自動配置機能による Explicit ヒープに配置するオブジェクトの選定を実施したあと、自動解放処理を繰り返し実行しても、Explicit ヒープ領域の利用済みサイズが単調増加を続ける場合に使用するオプションです。Explicit メモリーブロックの自動解放処理では、自動解放処理対象の Explicit メモリーブロックを選択します。しかし、オブジェクトの参照関係によっては、1つの Explicit メモリーブロックを選択しただけではオブジェクトが削除できないうで、複数の Explicit メモリーブロックを選択しなければ削除できない場合があります。このような場合に、このオプションを使用して、自動解放処理の対象とする選択量を増やします。

明示管理ヒープ機能の自動解放処理で設定している解放率の上限値を指定します。解放率とは、自動解放処理で選択した量に対する、実際に自動解放処理で解放した量の割合を指します。

明示管理ヒープ機能では、自動解放処理の対象とする選択量を決定するために、過去の自動解放処理の解放率を使用しています。この解放率によって、Explicit ヒープへの流入量に対して自動解放処理の対象とする選択量を調節し、Explicit ヒープが単調増加しないようにしています。ただし、自動解放処理の対象とする選択量を増やすと、自動解放処理時間が長くなります。自動解放処理実行中は、FullGC と同様にアプリケーションの実行が停止します。そのため、自動解放処理時間が長くなり過ぎないように、自動解放処理の対象とする選択量に上限を設けています。

0 を指定すると、このオプションを指定しない場合と同じ上限値となります。100 を指定すると、過去の自動解放処理で設定している解放率の上限値を最大に引き上げます。ただし、自動解放処理の動作上、100 を指定しても、Explicit ヒープ内のすべてのオブジェクトが自動解放処理の対象となるわけではありません。

前提オプション

- -XX:+HitachiUseExplicitMemory
- -XX:+HitachiExplicitMemoryAutoReclaim

書式

```
-XX:ExplicitMemoryAutoReclaimTargetBlockRatio=整数値
```

指定できる値

整数値

型：Integer

0~100の整数（単位：%）で指定します。

デフォルト値

定義項目の省略

`-XX:ExplicitMemoryAutoReclaimTargetBlockRatio=0`

値の省略

次のメッセージを標準エラーに出力して、JavaVMをリターンコード1で終了します。

```
Improperly specified VM option 'ExplicitMemoryAutoReclaimTargetBlockRatio='  
Error: Could not create the Java Virtual Machine.  
Error: A fatal exception has occurred. Program will exit.
```

7.11.2 -XX:ExplicitMemoryExcludeClassListFile

明示管理ヒープ機能適用除外設定ファイルのファイルパスを指定します。

説明

明示管理ヒープ機能適用除外クラス指定機能で利用する、明示管理ヒープ機能適用除外設定ファイルのファイルパスを指定します。このファイルに記述されているクラスには、明示管理ヒープ機能が適用されません。

JavaVM起動時に、このオプションに指定したファイルパスのファイルを明示管理ヒープ機能適用除外設定ファイルとして読み込みます。なお、このオプションを省略、または空文字を指定した場合は、デフォルトのファイルパスのファイルを明示管理ヒープ機能適用除外設定ファイルとして読み込みます。

前提オプション

- `-XX:+HitachiUseExplicitMemory`
- `-XX:+ExplicitMemoryUseExcludeClass`

関連オプション

- `-XX:ExplicitMemoryFullGCPolicy`
- `-XX:+HitachiAutoExplicitMemory`
- `-XX:ExplicitMemoryNotExcludeClassListFile`

書式

```
-XX:ExplicitMemoryExcludeClassListFile:文字列
```

指定できる値

文字列

型：String

JavaVM プロセスのカレントディレクトリーからの相対パス、または絶対パスを指定します。パスはディレクトリー名を含めて指定できます。

なお、指定できる明示管理ヒープ機能適用除外設定ファイルは 1 つだけです。また、ファイルの文字エンコーディングには、ASCII コードを使用してください。

デフォルト値

定義項目の省略

`-XX:ExplicitMemoryExcludeClassListFile:<空文字>`

<空文字>の場合に読み込まれるファイルパスを次に示します。

`JDKのインストールディレクトリー¥usrconf¥exmemexcludeclass.cfg`

注意事項

- 指定されたファイルがオープンできない、または読み込めない場合は、明示管理ヒープ機能のイベントログにワーニングメッセージを出力し、デフォルトパスにある設定ファイルを読み込みます。デフォルトパスにあるファイルをオープンできない、または読み込めない場合は、明示管理ヒープ機能適用除外クラス指定機能は有効になりません。
- 指定されたファイルの記述フォーマットに誤りがある場合は、明示管理ヒープ機能のイベントログにワーニングメッセージを出力して、次の行から構文解析処理を継続します。
- 明示管理ヒープ機能適用除外設定ファイルと、明示管理ヒープ機能適用除外無効設定ファイルの両方に、同じのクラス名が記述されていた場合は、明示管理ヒープ機能適用除外無効設定ファイルの指定が優先されます。この場合、そのクラスに対する明示管理ヒープ機能適用除外クラス指定機能は、無効になります。

7.11.3 -XX:ExplicitMemoryFullGCPolicy

FullGC 発生時に、参照関係に基づくオブジェクトの Java ヒープから Explicit メモリーブロックへの移動を制御します。

説明

FullGC 発生時に、参照関係に基づくオブジェクトの Java ヒープから Explicit メモリーブロックへの移動を制御するかどうかを指定します。このオプションを使用すると、FullGC 発生時に Explicit ヒープへ移動していたオブジェクトを、Tenured 領域へ移動できます。

前提オプション

- `-XX:+HitachiUseExplicitMemory`

関連オプション

- `-XX:+HitachiAutoExplicitMemory`
- `-XX:+ExplicitMemoryUseExcludeClass`

書式

```
-XX:ExplicitMemoryFullGCPolicy=数値
```

指定できる値

数値

型：Integer

0 または 1 を指定します。範囲外の値が指定された場合は 0 が設定されます。

0 を指定すると、FullGC 発生時、Explicit メモリーブロック内にあるオブジェクトから参照されている Java ヒープ内のオブジェクトは、Explicit メモリーブロックへ移動します。

1 を指定すると、FullGC 発生時、Explicit メモリーブロック内にあるオブジェクトから参照されている Java ヒープ内のオブジェクトは、Explicit メモリーブロックへ移動しません。New 領域にあるオブジェクトは Tenured 領域へ移動します。

デフォルト値

定義項目の省略

```
-XX:ExplicitMemoryFullGCPolicy=0
```

7.11.4 -XX:ExplicitMemoryNotExcludeClassListFile

明示管理ヒープ機能適用除外無効設定ファイルのファイルパスを指定します。

説明

明示管理ヒープ機能適用除外クラス指定機能で利用する、明示管理ヒープ機能適用除外無効設定ファイルのファイルパスを指定します。明示管理ヒープ機能適用除外設定ファイルに記述されていても、このファイルに記述されているクラスには、明示管理ヒープ機能適用除外クラス指定機能が適用されません。

JavaVM 起動時に、このオプションに指定したファイルパスのファイルを明示管理ヒープ機能適用除外無効設定ファイルとして読み込みます。なお、このオプションを省略、または空文字を指定した場合は、デフォルトのファイルパスのファイルを明示管理ヒープ機能適用除外無効設定ファイルとして読み込みます。

前提オプション

- `-XX:+HitachiUseExplicitMemory`
- `-XX:+ExplicitMemoryUseExcludeClass`

関連オプション

- `-XX:ExplicitMemoryFullGCPolicy`
- `-XX:HitachiAutoExplicitMemory`
- `-XX:ExplicitMemoryExcludeClassListFile`

書式

```
-XX:ExplicitMemoryNotExcludeClassListFile:文字列
```

指定できる値

文字列

型: String

JavaVM プロセスのカレントディレクトリーからの相対パス、および絶対パスを指定します。パスはディレクトリー名を含めて指定できます。

なお、指定できる明示管理ヒープ機能適用除外設定ファイルは 1 つだけです。また、ファイルの文字エンコーディングには、ASCII コードを使用してください。

デフォルト値

定義項目の省略

`-XX:ExplicitMemoryNotExcludeClassListFile:<空文字>`

<空文字>の場合に読み込まれるファイルパスを次に示します。

`JDKのインストールディレクトリー¥usrconf¥exmemnotexcludeclass.cfg`

注意事項

- 指定されたファイルがオープンできない、または読み込めない場合は、明示管理ヒープ機能のイベントログにワーニングメッセージを出力し、デフォルトパスにある設定ファイルを読み込みます。デフォルトパスにあるファイルをオープンできない、または読み込めない場合は、明示管理ヒープ機能適用除外クラス指定機能は無効になりません。
- 指定されたファイルの記述フォーマットに誤りがある場合は、明示管理ヒープ機能のイベントログにワーニングメッセージを出力して、次の行から構文解析処理を継続します。
- 明示管理ヒープ機能適用除外設定ファイルと、明示管理ヒープ機能適用除外無効設定ファイルの両方に、同じのクラス名が記述されていた場合は、明示管理ヒープ機能適用除外無効設定ファイルの指定が優先されます。この場合、そのクラスに対する明示管理ヒープ機能適用除外クラス指定機能は、無効になります。

7.11.5 -XX:[+|-]ExplicitMemoryUseExcludeClass

明示管理ヒープ機能適用除外クラス指定機能の有効または無効を設定します。

説明

明示管理ヒープ機能適用除外クラス指定機能の有効、無効を指定します。明示管理ヒープ機能適用除外クラス指定機能を有効にすると、次のファイルに記述されているクラスには、明示管理ヒープ機能が適用されません。このため、ファイルに記述されているクラスのオブジェクトは、Explicit ヒープではなく、Tenured 領域に移動します。

- *JDK*のインストールディレクトリー¥jre¥lib¥explicitmemory¥sysexmemexcludeclass.cfg
- *JDK*のインストールディレクトリー¥usrconf¥exmemexcludeclass.cfg

前提オプション

- -XX:+HitachiUseExplicitMemory

関連オプション

- -XX:+HitachiAutoExplicitMemory
- -XX:ExplicitMemoryFullGCPolicy
- -XX:ExplicitMemoryExcludeClassListFile
- -XX:ExplicitMemoryNotExcludeClassListFile

書式

```
-XX:[+|-]ExplicitMemoryUseExcludeClass
```

指定できる値

型: String

-XX:+ExplicitMemoryUseExcludeClass

明示管理ヒープ機能適用除外クラス指定機能を有効にします。

-XX:-ExplicitMemoryUseExcludeClass

明示管理ヒープ機能適用除外クラス指定機能を無効にします。

デフォルト値

定義項目の省略

-XX:-ExplicitMemoryUseExcludeClass

注意事項

- 明示管理ヒープ機能適用除外クラス指定機能を有効にすると、Explicit メモリーブロックへのオブジェクト移動制御機能も有効になります。
- JavaVM 内のクラス (java.lang、java.util など) は、明示管理ヒープ機能適用除外クラスとして指定できません。

7.11.6 -XX:[+|-]HitachiAutoExplicitMemory

Explicit メモリーブロックの自動配置機能の有効または無効を設定します。

説明

Explicit メモリーブロックの自動配置機能の有効、無効を指定します。有効にした場合、32-bit 版、64-bit 版に関係なく、JavaVM 起動時にあらかじめ必要とする Explicit メモリーブロックを確保する必要があります。ただし、起動時にメモリ空間に連続領域が確保できない環境の場合、JavaVM が起動できないため、注意が必要です。

なお、このオプションを有効にした場合、Explicit メモリーブロックの最小サイズは 16 キロバイトになります。無効に指定した場合、Explicit メモリーブロックの最小サイズは 64 キロバイトになります。

前提オプション

- -XX:+HitachiUseExplicitMemory
- -XX:+HitachiExplicitMemoryAutoReclaim
- -XX:-HitachiExplicitMemoryCompatibleToV8

関連オプション

- -XX:HitachiAutoExplicitMemoryFile

書式

```
-XX:[+|-]HitachiAutoExplicitMemory
```

指定できる値

型 : String

-XX:+HitachiAutoExplicitMemory

明示管理ヒープ機能の自動配置機能を有効にします。

-XX:-HitachiAutoExplicitMemory

明示管理ヒープ機能の自動配置機能を無効にします。

デフォルト値

定義項目の省略

`-XX:-HitachiAutoExplicitMemory`

7.11.7 -XX:HitachiAutoExplicitMemoryFile

Explicit メモリーブロックの自動配置機能で利用する自動配置設定ファイルのパスを指定します。

説明

Explicit メモリーブロックの自動配置機能で利用する自動配置設定ファイルのパスを指定します。

前提オプション

- `-XX:+HitachiAutoExplicitMemory`

書式

```
-XX:HitachiAutoExplicitMemoryFile:文字列
```

指定できる値

文字列

型: String

JavaVM プロセスのカレントディレクトリーからの相対パス、および絶対パスを指定します。パスはディレクトリー名を含めて指定できます。

なお、指定できる自動配置設定ファイルは 1 つだけです。

1 文字以上の文字列を指定した場合、明示管理ヒープ自動配置設定のデフォルト設定およびこのオプションで指定したファイルの設定が有効になります。

デフォルト値

定義項目の省略

`-XX:HitachiAutoExplicitMemoryFile:<空文字>`

注意事項

- クラス名、およびメソッドの引数は、`java.lang` パッケージのクラスも含め、すべて完全修飾クラス名で記述してください。

(誤った例)

String

(正しい例)

```
java.lang.String
```

- ジェネリックス (総称) を用いたクラス名は記述できません。パラメーター化されていないクラス名 (raw 型) を記述してください。

(誤った例)

```
java.util.HashMap<java.lang.String, java.lang.Object>
```

(正しい例)

```
java.util.HashMap
```

- ネストしたクラスは、「.」ではなく「\$」で区切った名前を記述してください。

(誤った例)

```
java.util.AbstractMap.SimpleEntry
```

(正しい例)

```
java.util.AbstractMap$SimpleEntry
```

- コンストラクターは、クラス名と同じメソッド名、または<init>と記述してください。MyMain クラスのコンストラクターの場合は次のように記述してください。

(例)

```
MyMain.MyMain()または MyMain.<init>()
```

- クラス名と同じ名前のメソッドが存在する場合、コンストラクターを指定しているのか、メソッドを指定しているのか判断できません。そのため、コンストラクター、およびメソッドの両方を指定したものとして扱われます。

(例)

```
MyMain.MyMain(int) # MyMain クラスの int 引数を持つコンストラクターと # MyMain(int)メソッドの両方を生成点とする
```

- 静的初期化子は、<clinit>と記述してください。MyMain クラスの静的初期化子の場合、次のように記述します。

(例)

```
MyMain.<clinit>()
```

- フィールド宣言時の代入によるオブジェクトの生成を生成点に指定する場合、生成点にデフォルトコンストラクターを記述します。
- ユーザー指定オブジェクトの完全修飾クラス名に配列を指定することはできません。

(例)

```
java.lang.String[]
```

- 存在しないクラス名、メソッド名、およびバイトコードを持たないメソッド (native メソッドおよび abstract メソッド) を含む行が存在する場合、その行は存在しないものとして扱います。

- ユーザー指定オブジェクトのクラス名に J2SE の内部クラスを指定した場合、明示管理ヒープ機能が適切なクラス名に読み替えることがあります。例えば、`java.util.HashMap$Entry` を `java.util.HashMap` に読み替えます。

7.11.8 -XX:HitachiExplicitHeapMaxSize

Explicit ヒープ全体の最大サイズを指定します。

説明

Explicit ヒープ全体の最大サイズを指定します。

前提オプション

- `-XX:+HitachiUseExplicitMemory`

書式

```
-XX:HitachiExplicitHeapMaxSize=整数値
```

指定できる値

整数値

型：Integer

単位を使って、次に示す範囲で整数値（単位：バイト）を指定します。

32 ビット版の JavaVM の場合

1 ~ $2^{31}-1$ (2147483647)

64 ビット版の JavaVM の場合

1 ~ $2^{63}-1$ (9223372036854775807)

使用できる単位を次に示します。なお、大文字・小文字は区別されません。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

設定した値が有効な値の範囲に含まれていない場合、次のメッセージを標準出力に出力して JavaVM をリターンコード1 で終了します。

```
Invalid maximum explicit heap size: -XX:HitachiExplicitHeapMaxSize=[指定した値]  
Could not create the Java virtual machine.
```

デフォルト値

定義項目の省略

`-XX:HitachiExplicitHeapMaxSize=64m`

7.11.9 -XX:[+|-]HitachiExplicitMemoryAutoReclaim

明示管理ヒープ機能の自動解放機能の有効または無効を設定します。

説明

明示管理ヒープ機能の自動解放機能を利用するかどうか指定します。

前提オプション

- `-XX:-HitachiExplicitMemoryCompatibleToV8`

関連オプション

- `-XX:+HitachiAutoExplicitMemory`

書式

```
-XX:[+|-]HitachiExplicitMemoryAutoReclaim
```

指定できる値

型 : String

`-XX:+HitachiExplicitMemoryAutoReclaim`

明示管理ヒープ機能の自動解放機能を有効にします。

`-XX:-HitachiExplicitMemoryAutoReclaim`

明示管理ヒープ機能の自動解放機能を無効にします。

デフォルト値

定義項目の省略

`-XX:+HitachiExplicitMemoryAutoReclaim`

7.11.10 -XX:[+|-]HitachiExplicitMemoryAutoRefReclaim

HTTP セッションで利用する Explicit ヒープの省メモリー化機能の有効または無効を設定します。

説明

HTTP セッションで利用する Explicit ヒープの省メモリー化機能の有効、無効を指定します。

無効にした場合は、Application Server が作成した Explicit メモリーブロックの自動解放の自動予約をしません。

有効にした場合は、自動配置設定ファイルで作成された Explicit メモリーブロックに加えて、HTTP セッションオブジェクトを配置するために Application Server が作成した Explicit メモリーブロックに対して自動解放の自動予約をします。これによって、必要な Explicit ヒープサイズを減少させることができます。

ただし、有効にした場合、稼働情報の出力内容に注意が必要です。

表 7-9 稼働情報ファイルの出力内容が異なる項目

稼働情報の種類	項目 (ヘッダーファイルに出力される文字列)	省メモリー化機能を有効にした場合	省メモリー化機能を無効にした場合
HTTP セッションで取得した Explicit メモリーブロックの個数	HTTPSessionEMemoryBlockCount.HighWaterMark HTTPSessionEMemoryBlockCount.LowWaterMark HTTPSessionEMemoryBlockCount.Current	内部動作を反映した値が出力されるため、システムで有効な HTTP セッション数とは異なる値が出力されます。	システムで有効な HTTP セッション数が出力されます。
アプリケーションで利用する Explicit ヒープ領域のサイズ	ApplicationEHeapSize.HighWaterMark ApplicationEHeapSize.LowWaterMark	この機能が自動解放対象とした <i>Explicit</i> メモリーのサイズ + 自動配置機能で利用する <i>Explicit</i> メモリーのサイズの合計サイズとなります。	自動配置機能で利用される Explicit メモリーのサイズが出力されます。

前提オプション

- `-XX:+HitachiExplicitMemoryAutoReclaim`

書式

```
-XX:[+|-]HitachiExplicitMemoryAutoRefReclaim
```

指定できる値

型: String

`-XX:+HitachiExplicitMemoryAutoRefReclaim`

HTTP セッションで利用する Explicit ヒープの省メモリー化機能を有効にします。

`-XX:-HitachiExplicitMemoryAutoRefReclaim`

HTTP セッションで利用する Explicit ヒープの省メモリー化機能を無効にします。

デフォルト値

定義項目の省略

`-XX:-HitachiExplicitMemoryAutoRefReclaim`

7.11.11 -XX:[+|-]HitachiExplicitMemoryCompatibleToV8

Explicit メモリーブロックを確保する方法を、08-00 と同様にするかどうかを指定します。

説明

Explicit メモリーブロックを確保する方法を、08-00 と同様にするかどうか指定します。08-50 以降の新機能を利用しないで、08-00 で動作するアプリケーションをそのまま 08-50 で動作させる場合、このオプションを有効にします。

なお、このオプションを利用する場合、明示管理ヒープの自動解放機能は無効になります。

前提オプション

- `-XX:+HitachiUseExplicitMemory`

関連オプション

- `-XX:+HitachiExplicitMemoryAutoReclaim`
- `-XX:+HitachiAutoExplicitMemory`

書式

```
-XX:[+|-]HitachiExplicitMemoryCompatibleToV8
```

指定できる値

型：String

`-XX:+HitachiExplicitMemoryCompatibleToV8`

08-00 と同様の方法で Explicit メモリーブロックを確保します。

`-XX:-HitachiExplicitMemoryCompatibleToV8`

08-00 と異なる方法で Explicit メモリーブロックを確保します。

デフォルト値

定義項目の省略

`-XX:-HitachiExplicitMemoryCompatibleToV8`

7.11.12 -XX:HitachiExplicitMemoryJavaLog

明示管理ヒープ機能のログファイルのプリフィクスおよびログファイルの出力先ディレクトリーを指定します。

説明

明示管理ヒープ機能のログファイルのプリフィクスおよびログファイルの出力先ディレクトリーを指定します。

同じカレントディレクトリーで複数の JavaVM プロセスを同時に実行した場合や、-XX:HitachiExplicitMemoryJavaLog:オプションで同じログ出力先を設定した複数の JavaVM プロセスを同時に実行した場合、明示管理ヒープ機能のログファイルに情報が正しく出力されません。複数の JavaVM プロセスを実行する場合は、-XX:HitachiExplicitMemoryJavaLog:オプションでプロセスごとにログ出力先が異なるようにパスを設定してください。

関連オプション

- -XX:HitachiJavaLog

書式

```
-XX:HitachiExplicitMemoryJavaLog:文字列
```

指定できる値

文字列

型: String

プリフィクスおよびパスを指定します。このオプションを指定しない場合、文字列には、"ehjavaLog"が設定されます。プリフィクス以下のファイル名称は、明示管理ヒープ機能のログファイルのローテーション方式によって異なります。

明示管理ヒープ機能のログファイルのローテーション方式がラップアラウンド方式の場合、ログファイル名は、文字列???.log (??は 01~99 の通し番号) で生成されます。例えば、文字列に"Samp"を指定すると、ログファイル名はSamp01.log になります。

明示管理ヒープ機能のログファイルのローテーション方式がシフト方式の場合、出力対象となるカレントログファイルとバックアップファイルが生成されます。カレントログファイルは文字列.log、バックアップログファイルは文字列???.log (??は 01~98 の通し番号) で生成されます。例えば、文字列に"Samp"を指定すると、カレントログファイル名はSamp.log、カレントログファイルがバックアップされると、バックアップログファイル名はSamp01.log になります。

プリフィクスおよびパスには次の 3 種類の指定ができます。

プリフィクスを指定する場合

指定した文字列のファイル名でカレントディレクトリーに出力されます。

パスを指定する場合

文字列にディレクトリーを指定した場合、そのディレクトリーにデフォルトの"ehjava.log"でファイルが作成されます。

パスとプリフィクスを同時に指定する場合

文字列にディレクトリーとプリフィクスを指定した場合、そのディレクトリーにファイルが作成されます。例えば、文字列に"d:¥temp¥Samp"を指定すると、d:¥temp ディレクトリーに、ラップアラウンド方式ではSamp01.log、シフト方式ではSamp.log が生成されます。

デフォルト値

定義項目の省略

-XX:HitachiExplicitMemoryJavaLog:ehjava.log

7.11.13 -XX:HitachiExplicitMemoryJavaLogFileSize

明示管理ヒープ機能のログファイルの 1 ファイル当たりの最大ファイルサイズをバイト単位で指定します。

説明

1 ファイルの最大ファイルサイズをバイト単位で設定します。

関連オプション

- -XX:HitachiJavaLogFileSize

書式

-XX:HitachiExplicitMemoryJavaLogFileSize=*整数値*

指定できる値

整数値

型: Integer

8192~2147483647 の範囲で整数値 (単位: バイト) を指定します。

範囲外の値が指定された場合は8192 が設定されます。

デフォルト値

定義項目の省略

ラップアラウンド方式の場合

-XX:HitachiExplicitMemoryJavaLogFileSize=256k

シフト方式の場合

`-XX:HitachiExplicitMemoryJavaLogFileSize=128M`

注意事項

自然数以外の値が指定された場合は、定義していないオプションを指定した場合と同様の動作になります。

7.11.14 -XX:HitachiExplicitMemoryJavaLogNumberOfFile

明示管理ヒープ機能のログファイルを作成する最大ファイル数を指定します。

説明

ログファイルの単純増加を防ぐため、作成する最大ファイル数を指定します。

ローテーション方式がラップアラウンド方式で最大ファイル数を超えた場合は、再度最初に作成したファイルへ出力します。

ローテーション方式がシフト方式で最大ファイル数を超えた場合は、いちばん古いバックアップファイルを削除します。

関連オプション

- `-XX:HitachiJavaLogNumberOfFile`

書式

`-XX:HitachiExplicitMemoryJavaLogNumberOfFile=整数値`

指定できる値

整数値

型：Integer

1～99 の範囲で指定します。

ただし、100 以上の値が指定された場合は99、0 が指定された場合は1 となります。

デフォルト値

定義項目の省略

ラップアラウンド方式の場合

`-XX:HitachiExplicitMemoryJavaLogNumberOfFile=4`

シフト方式の場合

`-XX:HitachiExplicitMemoryJavaLogNumberOfFile=8`

注意事項

自然数以外の値が指定された場合は、定義していないオプションを指定した場合と同様の動作になります。

7.11.15 -XX:HitachiExplicitMemoryLogLevel

明示管理ヒープ機能のイベントログのレベルを指定します。

説明

明示管理ヒープ機能のイベントログのレベルを指定します。

前提オプション

- -XX:+HitachiUseExplicitMemory

書式

```
-XX:HitachiExplicitMemoryLogLevel:適用範囲
```

指定できる値

適用範囲

型: String

none

明示管理ヒープ機能のイベントログを出力しません。

normal

通常運用での指定を想定しています。Explicit ヒープの状態を定期的 (GC ごと) に出力します。また、Explicit ヒープの状態が大きく変化するイベント時に出力します。

verbose

normal の内容に加え、Explicit ヒープの状態が変化するイベント時に出力します。ログ量が normal よりも多くなります。

debug

verbose の内容に加え、一部のイベント発生時の詳細な情報を出力します。ログ量が大量になるため、パフォーマンスが著しく低下します。

デフォルト値

定義項目の省略

```
-XX:HitachiExplicitMemoryLogLevel:none
```

注意事項

none、normal、verbose、debug 以外の値が指定された場合、次のメッセージを標準出力に出力して JavaVM をリターンコード1 で終了します。

```
Invalid explicit heap log level: -XX:HitachiExplicitMemoryLogLevel:適用範囲
Could not create the Java virtual machine.
(log levels: none normal verbose debug)
```

7.11.16 -XX:[+|-]HitachiExplicitMemoryMoveToTenuredFirst

Explicit メモリーブロックの解放処理でのオブジェクトの移動先を、Tenured 領域に変更します。

説明

Explicit メモリーブロックの解放処理でのオブジェクトの移動先を指定します。

前提オプション

- -XX:HitachiUseExplicitMemory

書式

```
-XX:[+|-]HitachiExplicitMemoryMoveToTenuredFirst
```

指定できる値

型: String

-XX:+HitachiExplicitMemoryMoveToTenuredFirst

Explicit メモリーブロックの解放処理でのオブジェクトの移動先が、Tenured 領域になります。

-XX:-HitachiExplicitMemoryMoveToTenuredFirst

Explicit メモリーブロックの解放処理でのオブジェクトの移動先が、New 領域になります。

デフォルト値

定義項目の省略

-XX:+HitachiExplicitMemoryMoveToTenuredFirst

注意事項

このオプションが有効になっている場合、解放処理が実行されるときに被参照のオブジェクトは、Tenured 領域へ移動します。

参照元のオブジェクトが一時的に短命なオブジェクトの場合、このオプションを無効とし、Explicit メモリーブロック内の被参照のオブジェクトを New 領域に移動することで、次回の CopyGC の際に参照元オブジェクトと同時に回収できます。そうすることによって、このオプションが無効となっている場合に、Tenured 領域の利用済みサイズが上昇することを防げます。

ただし、次の点に注意が必要です。

- CopyGC 頻度の増加

Explicit メモリーブロック解放時に、被参照のオブジェクトを New 領域に移動するため、移動したオブジェクトが New 領域を消費します。その結果、次回の CopyGC までの間隔が短くなり、スループットの低下や、オブジェクトの加齢速度が上昇するおそれがあります。

- Tenured 領域に移動するオブジェクトが New 領域と明示管理ヒープ領域間で循環

複数の Explicit メモリーブロック内のオブジェクトから参照されるオブジェクト（例えばアプリケーション共通のデータ定義）が存在する場合、そのオブジェクトは長寿命オブジェクトとなります。そのため、このオプションが有効な場合に、オブジェクトが Explicit メモリーブロックへ移動したあと、その Explicit メモリーブロックの解放処理が実行される際に Tenured 領域へ移動し、オブジェクトの寿命（アプリケーションの終了時など）まで Tenured 領域に存在し続けることとなります。

一方、このオプションが有効な場合に、Explicit メモリーブロックの解放処理が実行される際に New 領域へ移動し、その後しきい値までの CopyGC が実行されると、別の Explicit メモリーブロックへ移動します。この繰り返しによって、オブジェクトが New 領域と明示管理ヒープ領域を循環し、CopyGC や Explicit メモリーブロックの解放処理のオーバーヘッドが増大するおそれがあります。

7.11.17 -XX:[+|-]HitachiUseExplicitMemory

明示管理ヒープ機能の有効または無効を設定します。

説明

明示管理ヒープ機能の有効、無効を指定します。なお、明示管理ヒープ機能を有効にする場合は、-Xms と -Xmx の指定値が同じであることを推奨します。

関連オプション

- -Xms
- -Xmx

書式

```
-XX:[+|-]HitachiUseExplicitMemory
```

指定できる値

型：String

`-XX:+HitachiUseExplicitMemory`

明示管理ヒープ機能を有効にします。

`-XX:-HitachiUseExplicitMemory`

明示管理ヒープ機能を無効にします。

デフォルト値

定義項目の省略

`-XX:-HitachiUseExplicitMemory`

注意事項

`-XX:+UseG1GC` オプションを指定している場合、このオプションは指定できません。これらのオプションと同時に指定した場合、メッセージが標準出力に出力されて、JavaVM がリターンコード1 で終了します。

7.12 リソースの上限値指定で使用する JavaVM オプション

リソースの上限値を設定するためのオプションを説明しています。

7.12.1 -XX:HitachiJITCompileMaxMemorySize

JIT コンパイル時に確保する C ヒープのメモリーサイズの上限值を指定します。

説明

JIT コンパイル時に確保する C ヒープのメモリーサイズの上限值を指定します。JIT コンパイルによって確保する C ヒープのメモリーが指定値を超えた場合、JavaVM ログファイルにログを出力するとともに、以降の JIT コンパイルを抑制します。JIT コンパイル処理の対象となっていた Java メソッドは、以降、インタープリターでだけ実行されるようになります。なお、JIT コンパイルが抑制された場合でも、JavaVM は強制終了しないで、処理を続行します。

0 を指定した場合、JIT コンパイル時に確保するメモリーは上限値で制限されません。

なお、JIT コンパイルは JavaVM の内部スレッド (JIT コンパイラースレッド) で実行されます。JIT コンパイラースレッドは 2 つあるため、このオプションで指定した上限値を 2 で割った値が、1 つの JIT コンパイラースレッドに対する上限値となります。

また、このオプションで上限値を設定した場合も、実際に使用できる C ヒープがそれよりも少ないときには、上限値に達する前に C ヒープ不足が発生することがあります。この場合、JavaVM は強制終了します。

前提オプション

- -server

出力形式

```
[id][Thread: thread_id]<date>["thread_name" exceeded max memory size.]  
[current_sizeK->new_sizeK/limit_size_per_threadK/limit_sizeK]  
[compile_target][byte_code_size]
```

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	JMS (JavaVM ログファイル識別子)。
<i>thread_id</i>	JIT コンパイルを抑制した JIT コンパイラースレッドのスレッド ID。
<i>date</i>	JIT コンパイルを抑制した日時。 -XX:+HitachiOutputMilliTime オプションが指定された場合、ミリ秒単位で出力されます。
<i>thread_name</i>	JIT コンパイルを抑制した JIT コンパイラースレッドのスレッド名。

出力項目	出力内容
<i>current_size</i>	JIT コンパイルを抑制した JIT コンパイラースレッドの、現在のメモリー確保サイズ (単位：キロバイト)。
<i>new_size</i>	JIT コンパイルを抑制した JIT コンパイラースレッドの、現在のメモリー確保サイズと追加で確保しようとしたサイズの合計値 (単位：キロバイト)。
<i>limit_size_per_thread</i>	1 つの JIT コンパイラースレッドの上限値 (単位：キロバイト)。
<i>limit_size</i>	JIT コンパイラースレッド全体の上限値 (単位：キロバイト)。
<i>compile_target</i>	JIT コンパイル処理の対象となっていた Java メソッド。
<i>byte_code_size</i>	JIT コンパイル処理の対象となっていた Java メソッドのバイトコードのサイズ (単位：バイト)。

書式

```
-XX:HitachiJITCompileMaxMemorySize=整数値
```

指定できる値

整数値

型：Integer

JIT コンパイル時に確保するメモリーの上限値を指定します。単位はバイトです。単位文字として、「k」(キロ)、「m」(メガ)、「g」(ギガ) も指定できます。指定できる範囲は次のとおりです。範囲外の値を指定した場合は、0 が指定されます。

32 ビット版の JavaVM の場合

0~ $2^{32}-1$ (4294967295)

64 ビット版の JavaVM の場合

0~ $2^{64}-1$ (18446744073709551615)

0 を指定した場合は、JIT コンパイル時に確保するメモリーは上限値で制限されません。JIT コンパイル中に C ヒープ不足が発生した場合、JavaVM は強制終了します。

デフォルト値

定義項目の省略

```
-XX:HitachiJITCompileMaxMemorySize=0
```

出力例

オプションとして「-XX:HitachiJITCompileMaxMemorySize=536870912」を指定して、1 つの JIT コンパイラースレッドに対して 262145 キロバイトの C ヒープを確保した場合に、上限値に達したときの出力例を示します。

```
[JMS][Thread: 0x03bf1150]<Wed Feb 24 14:33:58 2010>
["CompilerThread0" exceeded max memory size.][262143K->262145K/262144K/524288K]
[test1.func][213]
```

注意事項

- JIT コンパイル時に確保するメモリーが、このオプションで指定した上限値を超えて、JIT コンパイルが抑制されると、アプリケーションのスループットが低下します。
- 自然数以外の値が指定された場合は、定義していないオプションを指定した場合と同様の動作になります。

7.12.2 -XX:HitachiThreadLimit

スレッド数の上限値を指定します。

説明

スレッド数の上限値を指定します。スレッド数の上限値が指定値を超えた場合に、OutOfMemoryError 例外がスローされます。ただし、JavaVM が起動する前に上限値を超えた場合、例外はスローされません。また、スレッド数には、アプリケーションが JNI などを使用して JavaVM の管理外の範囲で作成したスレッド数は含みません。0 を指定した場合、上限値は設定されません。

なお、-XX:+HitachiOutOfMemoryAbort オプションと同時に指定した場合、このオプションの機能によって OutOfMemoryError 例外がスローされた場合、JavaVM の強制終了は実行されません。

また、-XX:+HitachiOutOfMemoryStackTrace オプションと同時に指定した場合、例外メッセージが JavaVM ログファイルに出力されます。

出力形式

```
Could not create "name" thread. Threadlimit Exceeded. num threads exist.
```

このメッセージは、java.lang.Throwable.getMessage() で取得できます。また、メッセージ全体が半角 128 文字以上になった場合は、メッセージの末尾が省略されます。

出力項目	出力内容
<i>name</i>	作成に失敗したスレッド名。
<i>num</i>	現時点でのスレッド数。

-XX:+HitachiOutOfMemoryStackTrace オプションが指定されている場合、JavaVM ログファイルに例外情報とスタックトレースを出力できます。例外発生要因種別は、"Thread Limit" となります。例外発生要因種別を次に示します。

表 7-10 例外発生要因種別一覧

項番	要因メッセージ	説明
1	C Heap	C ヒープ確保時の例外
2	Java Heap	Java ヒープ確保時の例外
3	Meta Space	Metaspace 確保時の例外
4	Compressed Class Space	Compressed Class Space 確保時の例外
5	Unknown	上記例外を特定できない場合
6	Thread Limit	スレッド数の上限値設定機能 (-XX:HitachiThreadLimit オプション) を指定した場合に、作成したスレッド数が指定した上限値を超えたときの例外

書式

```
-XX:HitachiThreadLimit=整数値
```

指定できる値

整数値

型：Integer

スレッド数の上限値を 0~2147483647 の範囲で整数値（単位：スレッド数）を指定します。範囲外の値が指定された場合は 0 が設定されます。負の値を指定した場合はエラーとなります。

デフォルト値

定義項目の省略

```
-XX:HitachiThreadLimit=0
```

出力例

```
Could not create "Thread-1" thread. Threadlimit Exceeded. 9 threads exist.
```

この例は、main スレッドによって java.lang.Thread.start が呼び出され、この機能によって例外をスローした場合に、java.lang.Throwable.getMessage() でメッセージを取り出したメッセージの例です。

注意事項

上限値に小さな値を設定すると、起動前に例外をスローします。Java EE サーバが使用するスレッド数より小さな値を設定しないでください。

7.13 JIT コンパイラ稼働継続機能で使用する JavaVM オプション

JIT コンパイル失敗時の JavaVM の動作を設定するためのオプションを説明しています。

7.13.1 -XX:[+|-]JITCompilerContinuation

JIT コンパイラ稼働継続機能の有効または無効を設定します。

説明

JIT コンパイラ稼働継続機能を有効にするかどうかを指定します。

JIT コンパイルがアプリケーションを構成するメソッドの論理矛盾で失敗しても、アプリケーションを正常に継続できるようにするためには、この機能を有効にすることをお勧めします。

JIT コンパイラ稼働継続機能を有効にした場合、JIT コンパイルがアプリケーションを構成するメソッドの論理矛盾で失敗しても、JavaVM は JavaVM ログファイルにこの機能のログを出力して、処理を続行します。この場合、JIT コンパイルに失敗したメソッドでは、以降のコンパイルはインタープリター方式で実行されるようになります。アプリケーションを構成しているメソッドのうち、JIT コンパイルに失敗したメソッド以外のメソッドは JIT コンパイルで実行されるため、アプリケーションは正常に継続されます。ただし、JIT コンパイルが 6 回以上失敗すると、JavaVM はエラーレポートファイル、およびメモリーダンプまたは core ダンプを出力して、強制終了します。

JIT コンパイラ稼働継続機能を無効にした場合、JIT コンパイルがアプリケーションを構成するメソッドの論理矛盾で失敗すると、JavaVM はエラーレポートファイル、およびメモリーダンプまたは core ダンプを出力して、強制終了します。

前提オプション

- -server
- -XX:+HitachiVerboseGC

出力形式

[*id*]

注

[*id*]以降には JIT コンパイラ稼働継続機能のログが出力されます。

出力内容を次に説明します。

出力項目	出力内容
<i>id</i>	JCC (JavaVM ログファイル識別子)。

書式

```
-XX:[+|-]JITCompilerContinuation
```

指定できる値

型：String

-XX:+JITCompilerContinuation

JIT コンパイラ稼働継続機能を有効にします。

-XX:-JITCompilerContinuation

JIT コンパイラ稼働継続機能を無効にします。

デフォルト値

定義項目の省略

-XX:+JITCompilerContinuation

出力例

```
[JCC][Thread: 0x05432c00]<Thu Nov 15 17:10:40 2012>[Method: jit_sample.func()V]
[Fail: 3][JITCT: 1]
[JCC][Thread: 0x05432c00][PC: 0x083aff9a][Lib: D:¥work¥jdk¥jre¥bin¥server¥
jvm.dll+0x3aff9a][VM: Java HotSpot(TM) Server VM
(20.8-b03-CDK0950-20121115 mixed mode windows-x86 )]
[JCC][Thread: 0x05432c00][EAX=0x00000000, EBX=0x00618128, ECX=0x00000000,
EDX=0x05485340]
[JCC][Thread: 0x05432c00][ESP=0x0566d3c0, EBP=0x0566d3c4, ESI=0x00618278,
EDI=0x00000000]
[JCC][Thread: 0x05432c00][EIP=0x083aff9a, EFLAGS=0x00010202]
[JCC][Thread: 0x05432c00][siginfo: read 0x00000000]
[JCC][Thread: 0x05432c00][Unlock: MethodCompileQueue_Lock]
[JCC][Thread: 0x05432c00][NewJITCT: 0x05438800][JITCT: 2]
[JCC][Thread: 0x05432c00][Free: "ResourceArea" 524288 bytes.]
[JCC][Thread: 0x05432c00][stop]
[JCC][Fail: 1][date: Thu Nov 15 10:10:40 2012][Method: jit_sample.func1
(Ljava/lang/String;)V][PC: 0x083ff00a]
[Lib: D:¥work¥jdk¥jre¥bin¥server¥jvm.dll+0x3ff00a]
[JCC][Fail: 2][date: Thu Nov 15 11:11:16 2012][Method: jit_sample.func2()V]
[PC: 0x083afe3a][Lib: D:¥work¥jdk¥jre¥bin¥server¥jvm.dll+0x3afe3a]
```

7.14 Java アプリケーションプログラムの実行で使用する JavaVM オプション

java コマンド、または javaw コマンドで、Java アプリケーションプログラムを実行する際に使用するオプションを説明しています。

7.14.1 -Xhras

java コマンド、および javaw コマンドに対して有効なオプションです。Java アプリケーションプログラムの実行を行います。

説明

❗ 重要

Application Server の実行環境（アプリケーション開発環境以外の環境）で、Application Server と関連しない Java プログラムを使用する場合に、このオプションを使用するときは、別途、Application Server - Optional License for Java を購入してください。

java コマンド、または javaw コマンド実行時にこのオプションを設定すると、Java アプリケーションプログラムが実行できます。実行時には、自動的に JavaVM の稼働情報を出力する、次のオプションが指定されます。

- -XX:+HitachiJavaClassLibTrace
- -XX:+HitachiLocalsInStackTrace
- -XX:+HitachiLocalsSimpleFormat
- -XX:+HitachiOutOfMemoryAbort
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutputMilliTime
- -XX:-HitachiThreadDumpToStdout
- -XX:+HitachiVerboseGC

なお、Java アプリケーションプログラムは、java コマンド、java_hras コマンド、または javaw コマンドで実行できます。それぞれの実行方法を次に示します。

java コマンドの実行方法

次のどちらかの方法で実行します。

- -Xhras オプションを指定して実行する
- 環境変数 JAVA_HRAS を設定して実行する

これら以外の方法で実行すると、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.
```

java_hras コマンドの実行方法

*Application Server*のインストールディレクトリー/jdk/jre/bin/下のjava_hras コマンドを実行します。

java_hras コマンドは、java コマンドと同等の機能を持つコマンドです。java_hras コマンドでは、java コマンドの-Xhras オプションや環境変数JAVA_HRAS は指定不要です。

javaw コマンドの実行方法

次のどちらかの方法で実行します。

- -Xhras オプションを指定して実行する
- 環境変数JAVAW_HRAS を設定して実行する

これら以外の方法で実行すると、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.
```

書式

```
-Xhras
```

デフォルト値

定義項目の省略

なし

注意事項

このオプション以外のオプションを指定する場合は、このオプションよりあとに指定してください。

7.15 圧縮オブジェクトポインター機能で使用する JavaVM オプション

Java オブジェクトのサイズを圧縮して管理することで、JavaVM 実行時の Java ヒープ領域および Explicit ヒープ領域の使用サイズを削減する機能を使用するためのオプションを説明しています。

7.15.1 -XX:[+|-]UseCompressedOops

圧縮オブジェクトポインター機能の有効または無効を設定します。

説明

圧縮オブジェクトポインター機能の有効、無効を指定します。圧縮オブジェクトポインター機能は、Java オブジェクトのサイズを圧縮して管理することで、JavaVM 実行時の Java ヒープ領域、および Explicit ヒープ領域の使用サイズを削減します。なお、Explicit ヒープ領域は、明示管理ヒープ機能が有効な場合にだけ該当します。

この機能を有効にするには、次の条件を満たしていることが前提となります。

- 64 ビット版の JavaVM を使用していること
- Java ヒープ領域および Explicit ヒープ領域の指定サイズの合計値が 32 ギガバイト未満であること

JavaVM 起動時、Java ヒープ領域および Explicit ヒープ領域の指定サイズの合計値が、32 ギガバイト以上の場合、JavaVM は次のメッセージを標準出力に出力し、圧縮オブジェクトポインター機能を無効とします。

```
Java HotSpot(TM) 64-Bit Server VM warning: Max heap size too large for
Compressed Oops
```

書式

```
-XX:[+|-]UseCompressedOops
```

指定できる値

型 : String

-XX:+UseCompressedOops

JavaVM は、Java ヒープ領域および Explicit ヒープ領域にある Java オブジェクトのサイズを圧縮して管理します。

-XX:-UseCompressedOops

JavaVM は、Java ヒープ領域および Explicit ヒープ領域にある Java オブジェクトのサイズを圧縮しません。

デフォルト値

定義項目の省略

-XX:-UseCompressedOops

7.16 ファイナライズ滞留解消機能で使用する JavaVM プロパティ

JavaVM 内でファイナライズ処理が滞留している状態を検知して、滞留しているファイナライズ処理を解消する機能を使用するためのプロパティを説明しています。

7.16.1 JP.co.Hitachi.soft.jvm.autofinalizer

ファイナライズ滞留解消機能の有効または無効を設定します。

説明

ファイナライズ滞留解消機能の有効、無効を指定します。

true、false 以外の値が設定された場合、デフォルト値になります。

書式

```
JP.co.Hitachi.soft.jvm.autofinalizer={true | false}
```

指定できる値

型：boolean

true

Java 起動時に、FinalizerThread を監視するファイナライズ処理監視スレッドを生成します。

false

FinalizerThread を監視するファイナライズ処理監視スレッドは生成しません。

デフォルト値

定義項目の省略

```
JP.co.Hitachi.soft.jvm.autofinalizer=true
```

出力例

ファイナライズ処理の滞留を検知して、ファイナライズ処理監視スレッドを新たに生成する場合

```
# FinalizerWatcherThread: Create: create secondary finalizer thread.  
[queue length = 128] <Mon May 26 18:00:36 JST 2008>
```

生成したファイナライズ処理監視スレッドが終了した場合

```
# FinalizerWatcherThread: Finish: secondary finalizer thread is finished.  
<Mon May 26 20:12:26 JST 2008>
```

7.17 ユーザー拡張性能解析トレースで使用する JavaVM プロパティ

ユーザー拡張性能解析トレースの動作や出力情報に関する設定をするためのプロパティを説明しています。

7.17.1 jvm.userprf.Enable

ユーザー拡張性能解析トレースの有効または無効を設定します。

説明

ユーザー拡張性能解析トレースを有効にするかどうかを指定します。

前提プロパティはありません。

書式

```
jvm.userprf.Enable={true | false}
```

指定できる値

型：Boolean

true

ユーザー拡張性能解析トレースを有効にします。

false

ユーザー拡張性能解析トレースを無効にします。

true、false 以外を指定した場合、デフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.Enable=false
```

値の省略

```
jvm.userprf.Enable=false
```

記述例

```
-Djvm.userprf.Enable=true
```

注意事項

このプロパティにtrueを指定した場合、ユーザー拡張性能解析トレースは自動的にユーザー拡張性能解析トレース設定ファイルを読み込みます。事前にjvm.userprf.Fileプロパティでユーザー拡張性能解析トレース設定ファイルを設定してください。

7.17.2 jvm.userprf.ExtendedSetting

ユーザー拡張性能解析トレース設定ファイルの記述形式を指定します。

説明

ユーザー拡張性能解析トレース設定ファイルの記述形式に、次の方式を有効にするかどうかを指定します。

- パッケージ名とクラス名を指定する方式
- パッケージ名だけを指定する方式

前提プロパティ

- jvm.userprf.Enable=true

書式

```
jvm.userprf.ExtendedSetting={true | false}
```

指定できる値

型：Boolean

true

ユーザー拡張性能解析トレース設定ファイルに、次の方式で記述できます。

- パッケージ名とクラス名を指定する方式
- パッケージ名だけを指定する方式

false

ユーザー拡張性能解析トレース設定ファイルに、次の方式は記述できません。

- パッケージ名とクラス名を指定する方式
- パッケージ名だけを指定する方式

true、false 以外を指定した場合、デフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.ExtendedSetting=false
```

値の省略

```
jvm.userprf.ExtendedSetting=false
```

記述例

```
-Djvm.userprf.ExtendedSetting=true
```

注意事項

このプロパティに `false` を指定して、`true` で有効になる方式でユーザー拡張性能解析トレース設定ファイルに記述した場合、ユーザー拡張性能解析トレース設定ファイルのフォーマットエラーになります。

7.17.3 jvm.userprf.File

ユーザー拡張性能解析トレース設定ファイルのファイルパスを指定します。

説明

ユーザー拡張性能解析トレース設定ファイルのファイルパスを指定します。

ユーザー拡張性能トレースは、このプロパティに指定したファイルパスのファイルユーザー拡張性能解析トレース設定ファイルとして読み込みます。なお、このプロパティを指定していない場合、デフォルトのファイルパスのファイルユーザー拡張性能解析トレース設定ファイルとして読み込みます。

ユーザー拡張性能解析トレース設定ファイルに記述されたメソッドが、ユーザー拡張性能解析トレースのトレース対象となります。

前提プロパティ

- `jvm.userprf.Enable=true`

書式

```
jvm.userprf.File=文字列
```

指定できる値

文字列

型: `String`

- ユーザー拡張性能解析トレース設定ファイルのファイルパスを指定します。
- 指定できるファイルパスは1つだけです。
- カレントディレクトリーからの相対パスおよび絶対パスがパス文字列として有効です。

デフォルト値

定義項目の省略

JDKのインストールディレクトリー¥usrconf¥userprf.cfg

値の省略

ファイルが存在しないワーニングメッセージをJavaVM ログファイルに出力します。

記述例

```
-Djvm.userprf.File=/test/setting.txt
```

注意事項

- ユーザー拡張性能解析トレース設定ファイルを読み込むときの文字エンコーディングは、JDKの動作プラットフォームのデフォルトエンコーディングです。文字エンコーディングがデフォルトエンコーディング以外の場合、デフォルトエンコーディングで記述されているものとして処理されます。
- サーバごとにユーザー拡張性能解析トレースを異なる設定にする場合は、サーバごとにユーザー拡張性能解析トレース設定ファイルを用意して、このプロパティーでサーバごとのファイルパスを指定してください。
- 指定されたファイルパスのファイルが読み込めない場合や読み込んだ設定が有効でない場合、ユーザー拡張性能解析トレース設定ファイルの読み込み時のログがJavaVM ログファイルに出力されます。

7.17.4 jvm.userprf.Limit

ユーザー拡張性能解析トレースの対象となるメソッド数の上限値を指定します。

説明

ユーザー拡張性能解析トレースの対象となるメソッド数の上限値を指定します。

前提プロパティー

- `jvm.userprf.Enable=true`

書式

```
jvm.userprf.Limit=整数値
```

指定できる値

整数値

型：Integer

0～2147483647 までの整数値で指定します。それ以外を指定した場合はデフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.Limit=2147483647
```

値の省略

```
jvm.userprf.Limit=2147483647
```

記述例

```
-Djvm.userprf.Limit=100
```

注意事項

ユーザー拡張性能解析トレース設定ファイルで指定したメソッドのうち、このプロパティで指定した値のメソッド数までがトレースの対象になります。指定したメソッドがこのプロパティで指定した値をはじめて超えたとき、ログメッセージが JavaVM ログファイルに出力されます。

7.17.5 jvm.userprf.LineNumber

メソッドの最後に実行した行の行番号をトレース情報へ出力します。

説明

メソッドが正常に終了した場合、メソッドの最後に実行した行の行番号をトレース情報に出力するかどうかを指定します。

前提プロパティ

- `jvm.userprf.Enable=true`

書式

```
jvm.userprf.LineNumber={true | false}
```

指定できる値

型：Boolean

true

トレース情報のオペレーション情報 (OPR) に、メソッドの最後に実行した行の行番号を出力します。

false

メソッドの最後に実行した行の行番号をトレース情報に出力しません。

true、false 以外を指定した場合、デフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.LineNumber=false
```

値の省略

```
jvm.userprf.LineNumber=false
```

記述例

```
-Djvm.userprf.LineNumber=true
```

注意事項

javac コマンドの-g オプションで行番号のデバッグ情報を生成しない場合は、このプロパティにtrueを指定しても行番号は出力されません。

7.17.6 jvm.userprf.LogLevel

ユーザー拡張性能解析トレースのトレース出力レベルを指定します。

説明

ユーザー拡張性能解析トレースのトレース情報のオプション情報 (OPT) に出力する、トレース対象のメソッド名の出力レベルを指定します。

前提プロパティ

- jvm.userprf.Enable=true

書式

```
jvm.userprf.LogLevel={class | package | method | signature}
```

指定できる値

型: String

class

クラス名

package

完全修飾クラス名

method

完全修飾クラス名+メソッド名

signature

完全修飾クラス名+メソッド名+メソッドの引数の型

なお、`jvm.userprf.ThrowableName` プロパティに `true` を指定している場合、このプロパティの指定によってメソッドの異常出口で、次のようにオペレーション情報 (OPR) が出力されます。

- class を指定した場合：例外またはエラーのクラス名
- package、method または signature を指定した場合：例外またはエラーの完全修飾クラス名

デフォルト値

定義項目の省略

```
jvm.userprf.LogLevel=class
```

値の省略

```
jvm.userprf.LogLevel=class
```

記述例

パッケージ名「com.sample」、クラス名「ClassA」、およびメソッド名「methodA」のトレース情報を、完全修飾クラス名+メソッド名でトレース情報のオプション情報 (OPT) に出力する場合の記述例を次に示します。

```
-Djvm.userprf.LogLevel=method
```

トレース情報には次のよう出力されます。

```
com.sample.ClassA.methodA
```

7.17.7 jvm.userprf.ThrowableName

例外またはエラーのクラス名をトレース情報へ出力します。

説明

メソッドが異常終了した場合、例外またはエラーのクラス名をトレース情報のオペレーション情報 (OPR) に出力するかどうかを指定します。

前提プロパティ

- `jvm.userprf.Enable=true`

書式

```
jvm.userprf.ThrowableName={true | false}
```

指定できる値

型: Boolean

true

例外またはエラーのクラス名をトレース情報に出力します。

false

例外またはエラーのクラス名をトレース情報に出力しません。

true、false 以外を指定した場合、デフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.ThrowableName=false
```

値の省略

```
jvm.userprf.ThrowableName=false
```

記述例

```
-Djvm.userprf.ThrowableName=true
```

注意事項

このプロパティにtrueを指定した場合、`jvm.userprf.ThrowableNameEditMethod` プロパティに指定した編集方法でクラス名が出力されます。

7.17.8 jvm.userprf.ThrowableNameEditMethod

例外またはエラーのクラス名が文字数制限 32 文字を超えた場合の名前の編集方法を指定します。

説明

例外またはエラーのクラス名が 32 文字を超える場合のクラス名の編集方法を指定します。例外またはエラーのクラス名が 32 文字以内であれば、このプロパティの指定に関係なく、すべての文字がトレース情報に出力されます。

前提プロパティ

- `jvm.userprf.Enable=true`
- `jvm.userprf.ThrowableName=true`

書式

```
jvm.userprf.ThrowableNameEditMethod={FRONT_CUT | BACK_CUT | CENTER_CUT}
```

指定できる値

型：String

FRONT_CUT

例外またはエラーのクラス名の後方 32 文字が出力されます。なお、文字の先頭には省略を示す「*」が付与されます。

BACK_CUT

例外またはエラーのクラス名の前方 32 文字が出力されます。なお、文字の末尾には省略を示す「*」が付与されます。

CENTER_CUT

例外またはエラーのクラス名の前方 16 文字と後方 16 文字が出力されます。なお、文字の前方 16 文字と後方 16 文字の間に、省略を示す「*」が付与されます。

上記以外の文字列を指定した場合、デフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.ThrowableNameEditMethod=FRONT_CUT
```

値の省略

```
jvm.userprf.ThrowableNameEditMethod=FRONT_CUT
```

記述例

パッケージ名が「com.sample」、クラス名が「Class0123456789012345678901234567890」の例外で終了するとき、例外のクラス名または完全修飾クラス名の前方 32 文字をトレース情報のオペレーション情報（OPR）に出力する場合の記述例を次に示します。

```
-Djvm.userprf.ThrowableNameEditMethod=BACK_CUT
```

jvm.userprf.LogLevel プロパティの指定によって、トレース情報のオペレーション情報 (OPR) には次のように出力されます。

class を指定した場合

```
Class012345678901234567890123456*
```

package、method または signature を指定した場合

```
com.sample.Class0123456789012345*
```

7.17.9 jvm.userprf.Trace

ユーザー拡張性能解析トレース設定ファイルで指定したクラスファイルの書き換えの成功ログを出力します。

説明

ユーザー拡張性能解析トレース設定ファイルで指定したクラスファイルの書き換えの成功ログを出力するかどうかを指定します。

前提プロパティ

- jvm.userprf.Enable=true

書式

```
jvm.userprf.Trace={true | false}
```

指定できる値

型: Boolean

true

ユーザー拡張性能解析トレースの対象クラスの書き換えの成功ログを出力します。

false

ユーザー拡張性能解析トレースの対象クラスの書き換えの成功ログを出力しません。

true、false 以外を指定した場合、デフォルト値となります。

デフォルト値

定義項目の省略

```
jvm.userprf.Trace=false
```

値の省略

```
jvm.userprf.Trace=false
```

記述例

```
-Djvm.userprf.Trace=true
```

注意事項

ユーザー拡張性能解析トレースの対象クラスを書き換えに失敗した場合は、このプロパティの指定に関係なく、ログメッセージが JavaVM ログファイルに出力されます。

7.18 Application Server で使用する Java HotSpot VM のオプション

Java HotSpot VM に関する設定をするためのオプションを説明しています。

7.18.1 -agentlib:hndlwrap2

JDK のバージョンが 6 以降の環境で Windows をログオフしたとき、JavaVM の実行を継続します。

説明

JDK のバージョンが 6 以降の環境で Windows をログオフしたとき、JavaVM の実行を継続するかどうか設定します。

このオプションは、ほかの JVMTI プログラムと同時に使用できません。使用した場合は保証しません。使用した場合の動作は、JVMTI の動作に依存します。

デフォルト値

定義項目の省略

なし

7.18.2 -agentlib:libname

ネイティブエージェントライブラリーをロードします。

説明

ネイティブエージェントライブラリー *libname* をロードします。

書式

```
-agentlib:libname[=options]
```

指定できる値

入力には制限されません。

デフォルト値

定義項目の省略

なし

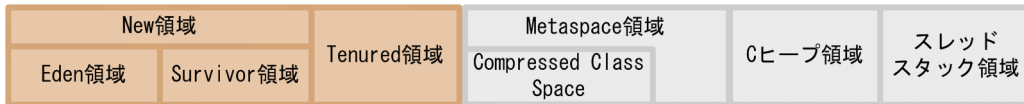
7.18.3 -XX:CompressedClassSpaceSize

Compressed Class Space の最大値を指定します。

説明

Compressed Class Space の最大値を指定します。

圧縮オブジェクトポインター機能が有効な場合、Compressed Class Space が Metaspace 領域内に作成されます。



この場合、それぞれの領域には、次の情報が配置されます。

- Compressed Class Space
Java ヒープ内のオブジェクトから参照されるクラス情報
- Metaspace 領域 (Compressed Class Space 以外)
上記以外のメソッド情報など

前提オプション

- -XX:+UseCompressedOops

書式

```
-XX:CompressedClassSpaceSize=size
```

指定できる値

size

型 : Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]

指定できる範囲は 1024k ~ 3g の整数値です。

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x64 [server]の場合

`-XX:CompressedClassSpaceSize=1G`

(凡例)

[server] : Java HotSpot Server VM が使用されることを示します。

注意事項

このオプションは、Windows x86 ではサポートしていません。

7.18.4 -XX:ConcGCThreads

Concurrent Marking の処理を並列実行するスレッド数を指定します。

説明

G1GC が有効な場合に、Concurrent Marking の処理を並列実行するスレッドの数を指定します。

JavaVM 起動時に、このオプションに指定した数の Concurrent Marking 処理用スレッドが作成されて、処理が実行されます。このオプションを省略した場合は、次の式で算出された数の Concurrent Marking 処理用スレッドが作成されます。

$\text{Max}((\text{ParallelGCThreads}^{\ast}+2)/4,1)$

注

小数点以下は切り捨てです。Max(A,B)は、A と B のうちから大きい方の値を選択することを示します。

注※

ParallelGCThreads は、`-XX:ParallelGCThreads` に指定した値です。

このオプションを指定する場合は、次の点に注意してください。

- 1 を指定するとスレッド 1 つで並列実行することになるため、実行性能が下がります。
- 実行環境で作成できるスレッド数を超える値を指定した場合は、JavaVM 起動時に Concurrent Marking 処理用スレッドの作成に失敗するため、JavaVM を起動できません。
- Concurrent Marking 処理用スレッドのスレッド数は、GC スレッド数を超えて作成できません。GC スレッド数を超える値を指定した場合は、次のメッセージが標準出力に出力され、JavaVM の起動に失敗します。

```
Error occurred during initialization of VM
Can't have more ConcGCThreads than ParallelGCThreads.
```

- 有効な範囲外の値を指定した場合、次のメッセージが標準出力に出力され、JavaVM がリターンコード1 で終了します。

```
Invalid number of concurrent marking threads: -XX:ConcGCThreads=[指定値]
```

前提オプション

- -XX:+UseG1GC

関連オプション

- -XX:ParallelGCThreads

書式

```
-XX:ConcGCThreads=整数値
```

指定できる値

整数値

型：Integer

0～1000 の整数で指定します。

デフォルト値

定義項目の省略

$\text{Max}((\text{ParallelGCThreads}+2)/4,1)$

値の省略

$\text{Max}((\text{ParallelGCThreads}+2)/4,1)$

7.18.5 -XX:MaxGCPauseMillis

G1GC の GC によるアプリケーション停止時間の目標時間をミリ秒単位で指定します。

説明

G1GC の GC によるアプリケーション停止時間の目標時間をミリ秒 (ms) 単位で指定します。

書式

```
-XX:MaxGCPauseMillis=整数値
```

指定できる値

型：Integer

1～4294967295 の整数で指定します。

デフォルト値

定義項目の省略

`-XX:MaxGCPauseMillis=200`

7.18.6 -XX:MaxMetaspaceSize

Metaspace 領域の最大値を指定します。

説明

Metaspace 領域の最大値を指定します。

Metaspace 領域はロードされたクラス情報が格納される領域です。

Metaspace 領域のサイズは、アプリケーションで必要なクラス情報のサイズから見積もることができません。見積もった値は、`-XX:MetaspaceSize` オプション、および `-XX:MaxMetaspaceSize` オプションに指定します。Metaspace 領域のサイズを正しく指定することで、Metaspace 領域の `OutOfMemoryError` の発生を防ぐことができます。また、`-XX:MetaspaceSize` オプションと `-XX:MaxMetaspaceSize` オプションに同じ値を指定することで、Metaspace 領域に起因する FullGC の発生を抑止できます。

Metaspace 領域のメモリー使用量は、アプリケーションが実際に使用しているサイズです。このため、`-XX:MetaspaceSize` オプション、および `-XX:MaxMetaspaceSize` オプションに指定した値のメモリーは使用されません。この特性から、Metaspace 領域は次のようなチューニングもできます。

- Metaspace 領域の `OutOfMemoryError` の発生リスクを下げる

Metaspace 領域の使用量が見積もり値以上に増加すると、Metaspace 領域の `OutOfMemoryError` が発生することがあります。この発生リスクを下げるため、次のオプションには、見積もり時に余裕を持った値を指定してください。

- `-XX:MaxMetaspaceSize`
- `-XX:CompressedClassSpaceSize`

Metaspace 領域の使用量がこれらのオプションの指定値を超えるまで、Metaspace 領域の `OutOfMemoryError` は発生しません。

- Metaspace 領域に起因する FullGC の発生リスクを下げる

Metaspace 領域の使用量が見積もり値以上に増加すると、Metaspace 領域に起因する FullGC が発生することがあります。この発生リスクを下げるため、次のオプションには、見積もり時に余裕を持った値を指定してください。

- `-XX:MetaspaceSize`
- `-XX:MaxMetaspaceSize`

- `-XX:CompressedClassSpaceSize`

Metaspace 領域の使用量がこれらのオプションの指定値を超えるまで、Metaspace 領域に起因する FullGC は発生しません。

書式

```
-XX:MaxMetaspaceSize=size
```

指定できる値

size

型：Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

指定できる範囲は次のとおりです。

32 ビット版の Java VM の場合

256K \sim 2³²-1 の整数値

64 ビット版の Java VM の場合

256K \sim 2⁶⁴-1 の整数値

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x86 [client] の場合

`-XX:MaxMetaspaceSize=232-1`

Windows x86 [server] の場合

`-XX:MaxMetaspaceSize=232-1`

Windows x64 [server] の場合

`-XX:MaxMetaspaceSize=264-1`

(凡例)

[client] : Java HotSpot Client VM が使用されることを示します。

[server] : Java HotSpot Server VM が使用されることを示します。

7.18.7 -XX:MaxTenuringThreshold

CopyGC 実行時に、From 空間と To 空間で Java オブジェクトを入れ替える回数のしきい値を指定します。

説明

CopyGC 実行時に、From 空間と To 空間で Java オブジェクトを入れ替える回数のしきい値を指定します。指定した回数を超えて入れ替え対象になった Java オブジェクトは、Tenured 領域に移動されます。

このオプションの有効範囲は、0~デフォルト値です。範囲外の値を指定した場合、しきい値を超えた場合に Tenured 領域へ移動する機能は無効になります。

書式

```
-XX:MaxTenuringThreshold=value
```

指定できる値

value

型：Integer

自然数の値を指定します。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

```
-XX:MaxTenuringThreshold=14
```

Windows x86 [server]の場合

```
-XX:MaxTenuringThreshold=14
```

Windows x64 [server]の場合

```
-XX:MaxTenuringThreshold=14
```

(凡例)

[client]：Java HotSpot Client VM が使用されることを示します。

[server]：Java HotSpot Server VM が使用されることを示します。

7.18.8 -XX:MetaspaceSize

Metaspace 領域に起因する FullGC の基準値を指定します。

説明

Metaspace 領域に起因する FullGC の基準値を指定します。

Metaspace 領域はロードされたクラス情報が格納される領域です。

Metaspace 領域のサイズは、アプリケーションで必要なクラス情報のサイズから見積もることができます。見積もった値は、`-XX:MetaspaceSize` オプション、および `-XX:MaxMetaspaceSize` オプションに指定します。Metaspace 領域のサイズを正しく指定することで、Metaspace 領域の `OutOfMemoryError` の発生を防ぐことができます。また、`-XX:MetaspaceSize` オプションと `-XX:MaxMetaspaceSize` オプションに同じ値を指定することで、Metaspace 領域に起因する FullGC の発生を抑止できます。

Metaspace 領域のメモリー使用量は、アプリケーションが実際に使用しているサイズです。このため、`-XX:MetaspaceSize` オプション、および `-XX:MaxMetaspaceSize` オプションに指定した値のメモリーは使用されません。この特性から、Metaspace 領域は次のようなチューニングもできます。

- Metaspace 領域の `OutOfMemoryError` の発生リスクを下げる
Metaspace 領域の使用量が見積もり値以上に増加すると、Metaspace 領域の `OutOfMemoryError` が発生することがあります。この発生リスクを下げるため、次のオプションには、見積もり時に余裕を持った値を指定してください。

- `-XX:MaxMetaspaceSize`
- `-XX:CompressedClassSpaceSize`

Metaspace 領域の使用量がこれらのオプションの指定値を超えるまで、Metaspace 領域の `OutOfMemoryError` は発生しません。

- Metaspace 領域に起因する FullGC の発生リスクを下げる
Metaspace 領域の使用量が見積もり値以上に増加すると、Metaspace 領域に起因する FullGC が発生することがあります。この発生リスクを下げるため、次のオプションには、見積もり時に余裕を持った値を指定してください。

- `-XX:MetaspaceSize`
- `-XX:MaxMetaspaceSize`
- `-XX:CompressedClassSpaceSize`

Metaspace 領域の使用量がこれらのオプションの指定値を超えるまで、Metaspace 領域に起因する FullGC は発生しません。

書式

```
-XX:MetaspaceSize=size
```

指定できる値

size

型：Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

指定できる範囲は次のとおりです。

32 ビット版の Java VM の場合

256K \sim 2³²-1 の整数値

64 ビット版の Java VM の場合

256K \sim 2⁶⁴-1 の整数値

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x86 [client] の場合

-XX:MetaspaceSize=12M

Windows x86 [server] の場合

-XX:MetaspaceSize=16M

Windows x64 [server] の場合

-XX:MetaspaceSize=16M

(凡例)

[client] : Java HotSpot Client VM が使用されることを示します。

[server] : Java HotSpot Server VM が使用されることを示します。

7.18.9 -XX:NewRatio

New 領域に対する Tenured 領域の割合を指定します。

説明

New 領域に対する Tenured 領域の割合を指定します。

value が 2 の場合は、New 領域と Tenured 領域の割合が、1:2 になります。

[New 領域の使用サイズ \geq Tenured 領域の空き領域サイズ] になると FullGC が発生します。このオプションに 1 を指定すると、FullGC が多発するので注意してください。

書式

```
-XX:NewRatio=value
```

指定できる値

value

型：Integer

自然数の値を指定します。

デフォルト値

定義項目の省略

Windows x86 [client] の場合

```
-XX:NewRatio=2
```

Windows x86 [server] の場合

```
-XX:NewRatio=2
```

Windows x64 [server] の場合

```
-XX:NewRatio=2
```

(凡例)

[client]：Java HotSpot Client VM が使用されることを示します。

[server]：Java HotSpot Server VM が使用されることを示します。

7.18.10 -XX:ParallelGCThreads

G1GC を並列実行するスレッドの数を指定します。

説明

G1GC を並列実行するスレッドの数を指定します。-XX:+UseG1GC を指定したときに有効になります。なお、このオプションは、デフォルト値を使用することを推奨します。変更する場合は、実際に JavaVM を動作させて測定した値を基に、最適な値を算出してから変更してください。

JavaVM 起動時に、このオプションに指定した数の GC スレッドが作成され、処理が実行されます。

このオプションに 0 を指定した場合は、デフォルト値が適用されます。

指定時には、次の点に注意してください。

- 1 を指定するとスレッド 1 つで並列実行することになるため、実行性能が下がります。
- 実行環境で作成できるスレッド数を超える値を指定した場合は、JavaVM 起動時に GC スレッドの作成に失敗するため、JavaVM を起動できません。
- 有効な範囲外の値を指定した場合、メッセージが標準出力に出力され、JavaVM がリターンコード 1 で終了します。
- G1GC の場合、このオプションの値を大きくすると、Evacuation 処理用のスレッド数が増えます。このため、Evacuation 処理に割り当てられるリソースが増加し、スループットが低下することがあります。このオプションの値を変更した場合、性能の要件を満たしているかどうかを確認してください。

書式

```
-XX:ParallelGCThreads=value
```

指定できる値

value

型：Integer

自然数の値を指定します。

デフォルト値

定義項目の省略

実行環境の論理 CPU 数が 8 以下の場合

`-XX:ParallelGCThreads=CPU数`

実行環境の論理 CPU 数が 9 以上の場合

`-XX:ParallelGCThreads=(8+(CPU数-8)×(5÷8))※`

注※

小数点以下は切り捨ての値になります。

例えば、4CPU の場合は、「ParallelGCThreads = 4」でデフォルト値は 4 になります。また、72CPU の場合は、「ParallelGCThreads = 8 + (72 - 8) × (5 ÷ 8) = 48」でデフォルト値は 48 になります。

7.18.11 -XX:ReservedCodeCacheSize

コードキャッシュ領域の最大サイズを指定します。

説明

コードキャッシュ領域の最大サイズを指定します。

書式

```
-XX:ReservedCodeCacheSize=整数値
```

指定できる値

整数値

型：Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

```
-XX:ReservedCodeCacheSize=32M
```

Windows x86 [server]の場合

```
-XX:ReservedCodeCacheSize=48M
```

Windows x64 [server]の場合

```
-XX:ReservedCodeCacheSize=48M
```

(凡例)

[client]：Java HotSpot Client VM が使用されることを示します。

[server]：Java HotSpot Server VM が使用されることを示します。

7.18.12 -XX:SurvivorRatio

Survivor 領域の From 空間と To 空間に対する Eden 領域の割合を指定します。

説明

New::Survivor 領域の From 空間と To 空間に対する New::Eden 領域の割合を指定します。

value に8を設定した場合は、New::Eden 領域、From 空間、To 空間の割合が、8:1:1 になります。

書式

```
-XX:SurvivorRatio=value
```

指定できる値

value

型：Integer

自然数の値を指定します。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

```
-XX:SurvivorRatio=8
```

Windows x86 [server]の場合

```
-XX:SurvivorRatio=8
```

Windows x64 [server]の場合

```
-XX:SurvivorRatio=8
```

(凡例)

[client]：Java HotSpot Client VM が使用されることを示します。

[server]：Java HotSpot Server VM が使用されることを示します。

7.18.13 -XX:TargetSurvivorRatio

GC 実行後の Survivor 領域内で Java オブジェクトが占める割合の目標値を指定します。

説明

GC 実行後の New::Survivor 領域内で Java オブジェクトが占める割合の目標値 (0~100 (単位：%)) を指定します。

書式

```
-XX:TargetSurvivorRatio=value
```

指定できる値

value

型：Integer

自然数の値を指定します。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

`-XX:TargetSurvivorRatio=50`

Windows x86 [server]の場合

`-XX:TargetSurvivorRatio=50`

Windows x64 [server]の場合

`-XX:TargetSurvivorRatio=50`

(凡例)

[client] : Java HotSpot Client VM が使用されることを示します。

[server] : Java HotSpot Server VM が使用されることを示します。

7.18.14 -XX:[+|-]UseG1GC

G1GC を実行します。

説明

G1GC を実行するかどうかを指定します。

このオプションは、ほかの GC 方式を選択するオプション (UseSerialGC、HitachiUseExplicitMemory) と同時に指定できません。

書式

```
-XX:[+|-]UseG1GC
```

指定できる値

型 : String

`-XX:+UseG1GC`

G1GC を実行します。

`-XX:-UseG1GC`

G1GC を実行しません。

デフォルト値

定義項目の省略

-XX:-UseG1GC

7.18.15 -XX:[+|-]UseSerialGC

SerialGC を実行します。

説明

SerialGC を実行するかどうかを指定します。

書式

```
-XX:[+|-]UseSerialGC
```

指定できる値

型 : String

-XX:+UseSerialGC

SerialGC を実行します。

-XX:-UseSerialGC

SerialGC を実行しません。

デフォルト値

定義項目の省略

-XX:+UseSerialGC

注意事項

- Application Server の GC は、SerialGC (UseSerialGC) または G1GC (UseG1GC) を選択できます。これらの GC を同時に指定しないようにしてください。同時に指定した場合は、Java プロセスが起動できません。
- -XX:+UseSerialGC と -XX:+UseG1GC の両方が指定された場合は、エラーメッセージを標準エラー出力に出力して、コマンドの起動に失敗します。JNI 経由の場合、JNI_CreateJavaVM が JNI_ERR(-1) を返します。

7.18.16 -verbose:information-type

information-type に指定した情報を出力します。

説明

information-type に指定した情報を出力します。

書式

```
-verbose:information-type
```

指定できる値

information-type

型: String

class

クラスがロードされるたびにクラスに関する情報を出力します。

gc

GC イベントが発生するたびに報告します。

jni

ネイティブメソッドの使用およびそのほかの Java Native Interface (JNI) アクティビティに関する情報を報告します。

デフォルト値

定義項目の省略

なし

7.18.17 -Xloggc:file

GC イベントが発生するたびに、*file* にログを出力します。

説明

-verbose:gc と同様に GC イベントが発生するたびに報告しますが、そのデータを *file* に記録します。-verbose:gc を指定したときに報告される情報のほかに、報告される各イベントの先頭に、最初の GC イベントからの経過時間（秒単位）が付け加えられます。

書式

```
-Xloggc:file
```

指定できる値

入力は制限されません。

デフォルト値

定義項目の省略

なし

7.18.18 -Xms

Java ヒープの初期サイズを指定します。

説明

Java ヒープの初期サイズを指定します。

書式

```
-Xmssize
```

指定できる値

size

型：Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

-Xms5120K

Windows x86 [server]の場合

-Xms5120K

Windows x64 [server]の場合

-Xms7.8m

(凡例)

[client] : Java HotSpot Client VM が使用されることを示します。

[server] : Java HotSpot Server VM が使用されることを示します。

7.18.19 -Xmx

Java ヒープの最大サイズを指定します。

説明

Java ヒープの最大サイズを指定します。

書式

```
-Xmxsize
```

指定できる値

size

型 : Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

-Xmx64M

Windows x86 [server]の場合

-Xmx64M

Windows x64 [server]の場合

-Xmx83M

(凡例)

[client] : Java HotSpot Client VM が使用されることを示します。

[server] : Java HotSpot Server VM が使用されることを示します。

7.18.20 -Xprof

実行中のプログラムのプロファイルを生成し、プロファイリングデータを標準出力へ出力します。

説明

このオプションを指定した場合、実行中のプログラムのプロファイルを生成し、プロファイリングデータを標準出力に出力します。このオプションは、プログラム開発用のユーティリティとして提供されています。本番稼働システムでの使用を目的としたものではありません。

デフォルト値

定義項目の省略

なし

7.18.21 -Xss

1 スタック領域の最大サイズを指定します。

説明

1 スタック領域の最大サイズを指定します。

書式

`-Xsssize`

指定できる値

size

型：Integer

自然数の値を次に示す単位を使って指定します。

- キロ [k]
- メガ [m]
- ギガ [g]
- テラ [t]

なお、大文字・小文字は区別されません。

デフォルト値

定義項目の省略

Windows x86 [client]の場合

-Xss320K

Windows x86 [server]の場合

-Xss320K

Windows x64 [server]の場合

-Xss1M

(凡例)

[client]：Java HotSpot Client VM が使用されることを示します。

[server]：Java HotSpot Server VM が使用されることを示します。

7.19 JavaVM で使用する環境変数

システムで使用できる JavaVM の環境変数を説明しています。

7.19.1 JAVA_HRAS

java コマンドを実行可能にするための環境変数です。

説明

Application Server の実行環境（アプリケーション開発環境以外の環境）で、Application Server と関連しない Java プログラムを使用する場合に、この環境変数を使用するときは、別途、Application Server - Optional License for Java を購入してください。

java コマンド実行時にこの環境変数を設定すると、Application Server と連携しない Java アプリケーションプログラムが実行できます。実行時には、自動的に JavaVM の稼働情報を出力する、次のオプションが指定されます。

- -XX:+HitachiJavaClassLibTrace
- -XX:+HitachiLocalsInStackTrace
- -XX:+HitachiLocalsSimpleFormat
- -XX:+HitachiOutOfMemoryAbort
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutputMilliTime
- -XX:-HitachiThreadDumpToStdout
- -XX:+HitachiVerboseGC

この環境変数を設定しない場合、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.
```

なお、Application Server と連携しない Java アプリケーションプログラムは、java コマンド、java_hras コマンド、または javaw コマンドで実行できます。それぞれの実行方法を次に示します。

java コマンドの実行方法

次のどちらかの方法で実行します。

- -Xhras オプションを指定して実行する
- 環境変数 JAVA_HRAS を設定して実行する

java_hras コマンドの実行方法

*Application Server*のインストールディレクトリー/jdk/jre/bin/下のjava_hras コマンドを実行します。

java_hras コマンドは、java コマンドと同等の機能を持つコマンドです。java_hras コマンドでは、java コマンドの-Xhras オプションや環境変数JAVA_HRAS は指定不要です。

javaw コマンドの実行方法

次のどちらかの方法で実行します。

- -Xhras オプションを指定して実行する
- 環境変数JAVAW_HRAS を設定して実行する

指定できる値

入力は制限されません。

デフォルト値

定義項目の省略

なし

値の省略

空文字も有効になります。

7.19.2 JAVACOREDİR

スレッドダンプファイルの出力先ディレクトリーを指定します。

説明

スレッドダンプファイルを出力する場合の、出力先ディレクトリーをバッチファイルで指定します。

前提オプション

- -XX:+HitachiThreadDump

書式

```
set JAVACOREDİR=スレッドダンプファイルの出力先ディレクトリー
```

指定できる値

スレッドダンプファイルの出力先ディレクトリー

型：String

デフォルトでは、スレッドダンプファイルはカレントディレクトリーに出力されます。

カレントディレクトリーを次に示します。

*Application Server*の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*

デフォルト値

定義項目の省略

なし

値の省略

カレントディレクトリー

記述例

スレッドダンプをC:¥home¥user¥threaddump に出力します。

```
set JAVACOREDIR=C:¥home¥user¥threaddump
```

注意事項

- JAVACOREDIR で指定したディレクトリーへのスレッドダンプファイルの出力に失敗した場合は、カレントディレクトリーにスレッドダンプファイルを出力します。
- カレントディレクトリーへの出力も失敗した場合は、標準エラー出力だけにスレッドダンプを出力しません。この場合は、標準出力にはスレッドダンプは出力されません。

7.19.3 JAVAW_HRAS

javaw コマンドを実行可能にするための環境変数です。

説明

Application Server の実行環境（アプリケーション開発環境以外の環境）で、Application Server と関連しない Java プログラムを使用する場合に、この環境変数を使用するときは、別途、Application Server - Optional License for Java を購入してください。

javaw コマンド実行時にこの環境変数を設定すると、Application Server と連携しない Java アプリケーションプログラムが実行できます。実行時には、自動的に JavaVM の稼働情報を出力する、次のオプションが指定されます。

- -XX:+HitachiJavaClassLibTrace
- -XX:+HitachiLocalsInStackTrace
- -XX:+HitachiLocalsSimpleFormat

- -XX:+HitachiOutOfMemoryAbort
- -XX:+HitachiOutOfMemoryStackTrace
- -XX:+HitachiOutputMilliTime
- -XX:-HitachiThreadDumpToStdout
- -XX:+HitachiVerboseGC

この環境変数を設定しない場合、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.
```

なお、Application Server と連携しない Java アプリケーションプログラムは、java コマンド、java_hras コマンド、またはjavaw コマンドで実行できます。それぞれの実行方法を次に示します。

java コマンドの実行方法

次のどちらかの方法で実行します。

- -Xhras オプションを指定して実行する
- 環境変数JAVA_HRAS を設定して実行する

java_hras コマンドの実行方法

*Application Server*のインストールディレクトリー/jdk/jre/bin/下のjava_hras コマンドを実行します。

java_hras コマンドは、java コマンドと同等の機能を持つコマンドです。java_hras コマンドでは、java コマンドの-Xhras オプションや環境変数JAVA_HRAS は指定不要です。

javaw コマンドの実行方法

次のどちらかの方法で実行します。

- -Xhras オプションを指定して実行する
- 環境変数JAVAW_HRAS を設定して実行する

指定できる値

入力は制限されません。

デフォルト値

定義項目の省略

なし

値の省略

空文字も有効になります。

8

JavaVM で使用する定義ファイル

明示管理ヒープ機能や、ユーザー拡張性能解析トレースの設定で使用する定義ファイルを説明しています。

8.1 JavaVM で使用する定義ファイルの一覧

明示管理ヒープ機能や、ユーザー拡張性能解析トレースの設定で使用する定義ファイルの概要を説明しています。

JavaVM で使用する定義ファイル

定義ファイル	概要	詳細説明
8.2 exmemexcludeclass.cfg	Explicit ヒープへ移動させたくないオブジェクトのクラスを指定する設定ファイル（明示管理ヒープ機能適用除外設定ファイル）	Explicit ヒープへ移動させたくないオブジェクトのクラスを指定します。この設定ファイルに指定したクラスのオブジェクトは、GC が発生しても、Explicit ヒープへ移動しません。昇格するタイミングで Tenured 領域へ移動します。
8.3 exmemnotexcludeclass.cfg	明示管理ヒープ機能適用除外設定ファイルに指定したクラスのうち適用除外設定を無効にしたいクラスを指定する設定ファイル（明示管理ヒープ機能適用除外無効設定ファイル）	明示管理ヒープ機能適用除外設定ファイルに指定したクラスのうち、適用除外設定を無効にしたいクラスを指定します。このファイルに指定したクラスのオブジェクトは、GC が発生すると、Explicit ヒープへ移動します。
8.4 userprf.cfg	ユーザー拡張性能解析トレースで使用するユーザー拡張性能解析トレース設定ファイル	ユーザー拡張性能解析トレースのトレース対象メソッドの情報は、ユーザー拡張性能解析トレース設定ファイルを使って設定します。

8.2 exmemexcludeclass.cfg

Explicit ヒープへ移動させたくないオブジェクトのクラスを指定する設定ファイルのことです。この設定ファイルを明示管理ヒープ機能適用除外設定ファイルといいます。

説明

Explicit ヒープへ移動させたくないオブジェクトのクラスを指定します。この設定ファイルに指定したクラスのオブジェクトは、GC が発生しても、Explicit ヒープへ移動しません。昇格するタイミングで Tenured 領域へ移動します。

明示管理ヒープ機能適用除外設定ファイルには、システムで提供しているファイルがあります。明示管理ヒープ機能適用除外クラス指定機能を有効にすると、システムで提供している設定ファイル (sysexmemexcludeclass.cfg) が使用されます。システムで提供している設定ファイルのほかに、明示管理ヒープ機能の適用対象から除外するクラスを追加したい場合は、格納先に示すファイルパスにある設定ファイルを更新するか、または新たな設定ファイルを作成してください。なお、新たに設定ファイルを作成した場合は、`-XX:ExplicitMemoryExcludeClassListFile` オプションにファイルパスを指定してください。

システムで提供している設定ファイルのファイルパスを次に示します。

JDKのインストールディレクトリ-%jre%lib%explicitmemory%sysexmemexcludeclass.cfg

この設定ファイルに指定したクラスに対して適用除外設定を無効にする制御は、明示管理ヒープ機能適用除外無効設定ファイルで指定できます。明示管理ヒープ機能適用除外無効設定ファイルの指定は、この設定ファイルの指定よりも優先されます。

パッケージ「com.sample」を例に、各設定ファイルの指定と明示管理ヒープ機能の適用範囲について説明します。パッケージ「com.sample」には、ClassA と ClassB の 2 つのクラスがあります。各設定ファイルを次のように指定します。

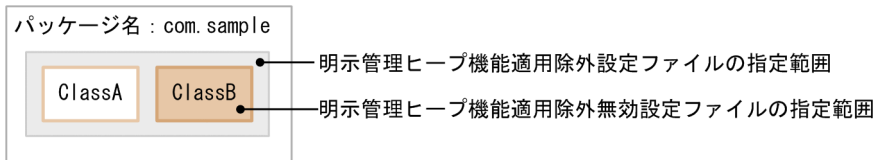
明示管理ヒープ機能適用除外設定ファイルの指定例

```
com.sample.*
```

明示管理ヒープ機能適用除外無効設定ファイルの指定例

```
com.sample.ClassB
```

明示管理ヒープ機能適用除外設定ファイルの指定には、ClassA と ClassB の両方が含まれています。しかし、明示管理ヒープ機能適用除外無効設定ファイルの指定が優先されるため、次の図のように、明示管理ヒープ機能の適用が除外されるのは ClassA だけとなり、ClassB には明示管理ヒープ機能が適用されます。



(凡例)

- : 明示管理ヒープ機能が適用されないクラス
- : 明示管理ヒープ機能が適用されるクラス

書式

配列型以外の場合

```
指定したクラスの完全修飾クラス名#コメント
...
指定したクラスの完全修飾クラス名
```

配列型の場合

```
配列の次元数分の "[ ]" * 指定したクラスの完全修飾クラス名 ;
```

注※

多次元配列のときは、「[]」を次元数分続けて指定します。3次元配列の場合は「[[[]」となります。

(例) aaa.bbb.Myclass クラスの1次元配列の場合

```
[Laaa.bbb.Myclass;
```

この設定ファイルの記述規則を次に示します。

- クラス名は1行に1つずつ記述します。
- 1行に記述できる文字数は1,024文字までです。この文字数は空文字やコメントを含みます。1行に1,025文字以上記述すると、パースに失敗してワーニングメッセージを出力し、その行を無視して読み込み処理を続けます。
- クラス名は、「パッケージ名.*」と指定すると省略できます。Java言語のimport宣言の「*」とは異なり、サブパッケージのクラスも対象となります。
- 行末は、改行文字（「`\n`」もしくは「`0x0A`」）または復帰文字（「`\r`」もしくは「`0x0D`」）が1文字以上続いたものとなります。
- 空白文字は、半角スペース文字（「`0x20`」）またはタブ文字（「`\t`」もしくは「`0x09`」）となります。なお、設定ファイルに空白文字を記述した場合は無視されます。
- コメントは、「#」で始まり、「#」から行末までの間の文字すべてをコメントとします。

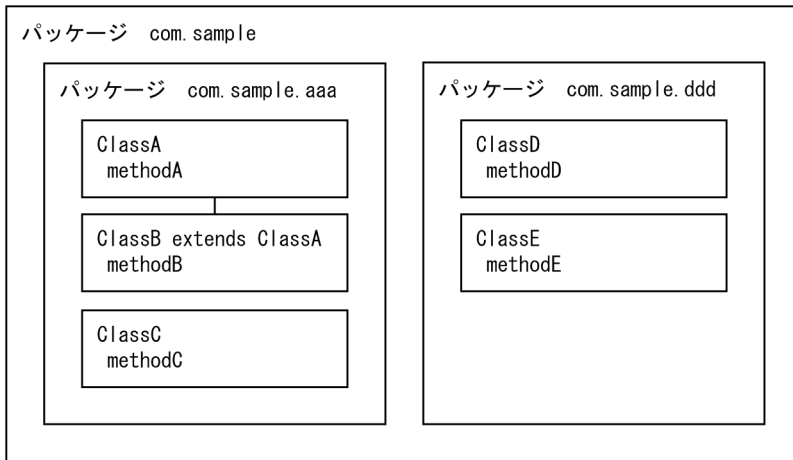
格納先

JDKのインストールディレクトリー`\usrconf\%exmemexcludeclass.cfg`

記述例

ここで説明する記述例は、パッケージ名が「com.sample」で、次の図に示すクラス構造とします。

図 8-1 クラス構造の例



完全修飾クラス名で指定する場合

完全修飾クラス名で指定する場合の明示管理ヒープ機能適用除外設定ファイルの記述例を次に示します。

```
com.sample.aaa.ClassA
com.sample.aaa.ClassC
com.sample.ddd.ClassD
```

この例では、ClassA クラス、ClassC クラス、および ClassD クラスのオブジェクトが Tenured 領域へ移動します。

クラス名を省略して指定する場合

クラス名を省略して指定する場合の明示管理ヒープ機能適用除外設定ファイル、および明示管理ヒープ機能適用除外無効設定ファイルの記述例を次に示します。

明示管理ヒープ機能適用除外設定ファイルの記述例

```
com.sample.*
```

明示管理ヒープ機能適用除外無効設定ファイルの記述例

```
com.sample.aaa.ClassB
com.sample.ddd.ClassE
```

この例では、明示管理ヒープ機能適用除外設定ファイルの記述から、同一パッケージ内のクラスだけでなく、サブパッケージに存在するクラスも含めすべてのクラスが Tenured 領域への移動対象となります。しかし、明示管理ヒープ機能適用除外無効設定ファイルの記述から、ClassB クラスと ClassE クラスのオブジェクトが Explicit メモリーブロックへの移動対象となります。このため、ClassA クラス、ClassC クラス、および ClassD クラスのオブジェクトが Tenured 領域へ移動します。

完全修飾クラス名で指定するか、またはクラス名を省略して指定するかは、設定ファイルの記述量が少ない方で指定することをお勧めします。記述例はどちらも同じ制御となります。この場合は、クラス名を省略して指定する方が望ましい記述です。

8.3 exmemnotexcludeclass.cfg

明示管理ヒープ機能適用除外設定ファイルに指定したクラスのうち、適用除外設定を無効にしたいクラスを指定する設定ファイルのことです。この設定ファイルを明示管理ヒープ機能適用除外無効設定ファイルといいます。

説明

明示管理ヒープ機能適用除外設定ファイルに指定したクラスのうち、適用除外設定を無効にしたいクラスを指定します。このファイルに指定したクラスのオブジェクトは、GCが発生すると、Explicit ヒープへ移動します。

明示管理ヒープ機能の適用対象から除外されているクラスを無効にしたい場合は、格納先に示すファイルパスにある設定ファイルを更新するか、または新たな設定ファイルを作成してください。システムで提供している明示管理ヒープ機能適用除外設定ファイル (sysexmemexcludeclass.cfg) に指定されているクラスも設定を無効にできます。なお、新たに設定ファイルを作成した場合は、`-XX:ExplicitMemoryNotExcludeClassListFile` オプションにファイルパスを指定してください。

この設定ファイルの指定は、明示管理ヒープ機能適用除外設定ファイルの指定よりも優先されます。

パッケージ「com.sample」を例に、各設定ファイルの指定と明示管理ヒープ機能の適用範囲について説明します。パッケージ「com.sample」には、ClassA と ClassB の2つのクラスがあります。各設定ファイルを次のように指定します。

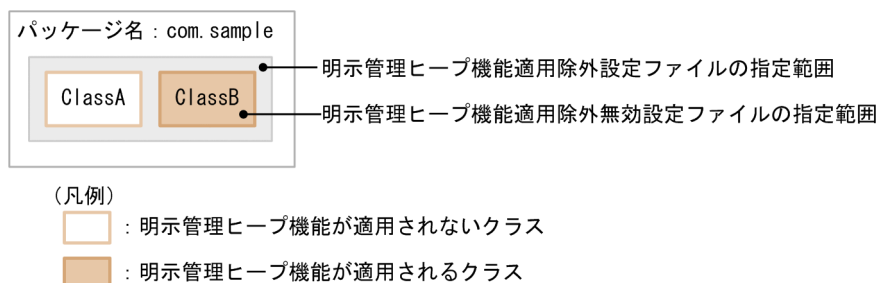
明示管理ヒープ機能適用除外設定ファイルの指定例

```
com.sample.*
```

明示管理ヒープ機能適用除外無効設定ファイルの指定例

```
com.sample.ClassB
```

明示管理ヒープ機能適用除外設定ファイルの指定には、ClassA と ClassB の両方が含まれています。しかし、明示管理ヒープ機能適用除外無効設定ファイルの指定が優先されるため、次の図のように、明示管理ヒープ機能の適用が除外されるのは ClassA だけとなり、ClassB には明示管理ヒープ機能が適用されます。



書式

配列型以外の場合

```
指定したクラスの完全修飾クラス名#コメント  
...  
指定したクラスの完全修飾クラス名
```

配列型の場合

```
配列の次元数分の"[ "※指定したクラスの完全修飾クラス名;
```

注※

多次元配列のときは、「[」を次元数分続けて指定します。3次元配列の場合は「[[[」となります。

(例) aaa.bbb.Myclass クラスの1次元配列の場合

```
[Laaa.bbb.Myclass;
```

この設定ファイルの記述規則を次に示します。

- クラス名は1行に1つずつ記述します。
- 1行に記述できる文字数は1,024文字までです。この文字数は空文字やコメントを含みます。1行に1,025文字以上記述すると、パースに失敗してワーニングメッセージを出力し、その行を無視して読み込み処理を続けます。
- クラス名は、「パッケージ名.*」と指定すると省略できます。Java言語のimport宣言の「*」とは異なり、サブパッケージのクラスも対象となります。
- 行末は、改行文字（「¥n」もしくは「0x0A」）または復帰文字（「¥r」もしくは「0x0D」）が1文字以上続いたものとなります。
- 空白文字は、半角スペース文字（「0x20」）またはタブ文字（「¥t」もしくは「0x09」）となります。なお、設定ファイルに空白文字を記述した場合は無視されます。
- コメントは、「#」で始まり、「#」から行末までの間の文字すべてをコメントとします。

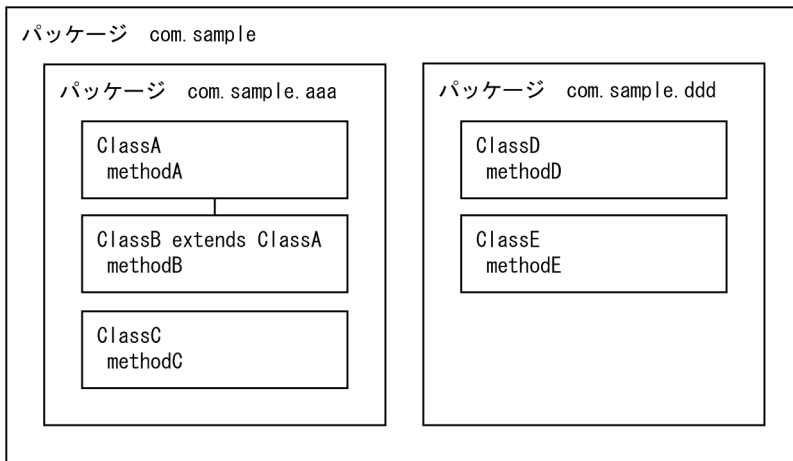
格納先

JDKのインストールディレクトリ¥usrconf¥exmemnotexcludelass.cfg

記述例

ここで説明する記述例は、パッケージ名が「com.sample」で、次の図に示すクラス構造とし、明示管理ヒープ機能適用除外設定ファイルでは、クラス名を省略して指定するとします。

図 8-2 クラス構造の例



明示管理ヒープ機能適用除外設定ファイルの記述例

```
com.sample.*
```

明示管理ヒープ機能適用除外無効設定ファイルの記述例

```
com.sample.aaa.ClassB
com.sample.ddd.ClassE
```

この例では、明示管理ヒープ機能適用除外設定ファイルの記述から、同一パッケージ内のクラスだけでなく、サブパッケージに存在するクラスも含めすべてのクラスが Tenured 領域への移動対象となります。しかし、明示管理ヒープ機能適用除外無効設定ファイルの記述から、ClassB クラスと ClassE クラスのオブジェクトが Explicit メモリーブロックへの移動対象となります。このため、ClassA クラス、ClassC クラス、および ClassD クラスのオブジェクトが Tenured 領域へ移動します。

注意事項

この設定ファイルは、明示管理ヒープ機能適用除外設定ファイルに指定されたクラスの中で明示管理ヒープ機能適用除外設定を無効にするクラスを指定する設定ファイルであり、Explicit メモリーブロックへ移動するクラスを指定する設定ファイルではありません。

8.4 userprf.cfg

ユーザー拡張性能解析トレースで使用するユーザー拡張性能解析トレース設定ファイルのことです。

説明

ユーザー拡張性能解析トレースのトレース対象メソッドの情報は、ユーザー拡張性能解析トレース設定ファイルを使って設定します。

書式

```
指定形式, 識別ID, サブクラスフラグ[, [イベントID][, [トレース取得レベル]]]  
[#コメント]
```

注

クラス名およびパッケージ名が一致するすべてのメソッドを対象とする場合、`jvm.userprf.ExtendedSetting` プロパティの値を `true` に設定してください。

ユーザー拡張性能解析トレース設定ファイルの記述項目を次に示します。

記述項目	内容
指定形式	<p>トレース対象メソッドを指定します。</p> <p>指定形式は次のどれかの形式で指定します。</p> <ul style="list-style-type: none">メソッド名と引数の型が一致するメソッドをトレースの対象とする場合 パッケージ名.クラス名.メソッド名(メソッドの引数の型名)メソッド名が一致するメソッドをトレースの対象とする場合 パッケージ名.クラス名.メソッド名(*)クラス名が一致するすべてのメソッドをトレースの対象とする場合 パッケージ名.クラス名パッケージ名が一致するすべてのメソッドをトレースの対象とする場合 パッケージ名.* <p>指定形式の項目について説明します。</p> <p>パッケージ名 トレース対象メソッドのクラスまたはインターフェースのパッケージ名を指定します。</p> <p>クラス名 トレース対象メソッドのクラス名またはインターフェース名を指定します。クラス名の代わりにインターフェース名を指定し、かつ、サブクラスフラグに <code>true</code> を指定した場合、そのインターフェースを実装しているメソッドがトレース対象となります。</p> <p>メソッド名 トレース対象メソッドの名前を指定します。</p> <p>メソッドの引数の型名 トレース対象メソッドの引数の型を完全修飾名で指定します。</p>
識別ID	トレース対象メソッドを識別するための文字列を指定します。

記述項目	内容
	<p>使用できる文字は ASCII 文字で 0x21 (!) から 0x7e (~) までです。ただし、0x22 (")、0x23 (#)、および 0x2c (,) は指定できません。</p> <p>識別 ID は PRF トレースファイルに出力されます。なお、PRF トレースファイルに出力される文字数は 32 文字までです。33 文字目以降は省略されて、33 文字目に*が出力されます。</p> <p>トレース対象メソッドが定義されているクラスにサブクラスが存在し、そのサブクラスでトレース対象メソッドがオーバーライドされていない場合、スーパークラスのトレース対象メソッドに対して指定された識別 ID が出力されます。</p>
サブクラスフラグ	<p>指定したメソッドのクラスまたはインターフェースと継承関係にあるクラスのメソッドをトレース対象に含めるかどうかを true または false で指定します。</p> <ul style="list-style-type: none"> • true を指定した場合、指定したメソッドと、それをオーバーライドしているメソッドがトレース対象となります。 • false を指定した場合、指定したメソッドだけがトレース対象となり、指定したメソッドをオーバーライドしているメソッドはトレース対象となりません。
イベント ID	<p>トレース情報を取得するポイント (トレース取得ポイント) を 16 進数 (0xae02~0xae7e および 0xc000~0xcffe) の値で指定します。デフォルト値は 0xae00 です。</p> <p>メソッドの入口で出力されるトレース情報にはこの値が出力され、メソッドの出口で出力されるトレース情報にはこの値+1 が出力されます。</p>
トレース取得レベル	<p>トレース対象メソッドのトレース取得レベルに、A、B、C、またはそれぞれの小文字のどれかを指定します。デフォルト値は A です。</p> <ul style="list-style-type: none"> • A：標準レベル • B：詳細レベル • C：保守レベル <p>トレース取得レベルは <code>cprflvel</code> コマンドの PRF トレース取得レベルと同じです。トレース対象メソッドはすべて Java VM レイヤのレベルを参照します。</p>
コメント	<p>コメントは「#」で始まり、「#」から行末までの間の文字すべてをコメントとします。</p>

ユーザー拡張性能解析トレース設定ファイルの記述規則を次に示します。

- 記述にはシングルバイト文字だけが使用できます。
- 空白文字は、半角スペース文字 (「0x20」) またはタブ文字 (「\t」) もしくは「0x09」) となります。なお、ユーザー拡張性能解析トレース設定ファイルの読み込みでは空白文字は無視されます。
- トレース対象メソッドは 1 行に 1 つずつ記述します。
- 1 行に指定できる文字数は 2,048 文字までです。この文字数には空白やコメントを含みます。
- 行末は改行文字 (「\n」) もしくは「0x0A」) または復帰文字 (「\r」) もしくは「0x0D」) が 1 文字以上続いたものとなります。
- ユーザー拡張性能解析トレース設定ファイルの記述に誤りがある場合、誤りの内容を示すメッセージが出力されます。また、各項目に無効な値を記述した場合、誤りの内容を示すメッセージが出力され、その行の設定は無効になります。

ユーザー拡張性能解析トレースでユーザー拡張性能解析トレース設定ファイルの読み込み時に出力されるログ情報を次の表に示します。

出力形式	出力内容	説明
[UPR※1]<DATE>Setting file not found.<file=FILEPATH>	<ul style="list-style-type: none"> • DATE※2 ユーザー拡張性能解析トレース設定ファイルの読み込みに失敗した日時。 • FILEPATH 読み込みに失敗したユーザー拡張性能解析トレース設定ファイルの絶対パス。 	デフォルト、またはjvm.userprf.Fileプロパティーで指定したユーザー拡張性能解析トレース設定ファイルが存在しません。
[UPR※1]<DATE>Failed to open setting file.<file=FILEPATH>	<ul style="list-style-type: none"> • DATE※2 ユーザー拡張性能解析トレース設定ファイルの読み込みに失敗した日時。 • FILEPATH 読み込みに失敗したユーザー拡張性能解析トレース設定ファイルの絶対パス。 	ユーザー拡張性能解析トレース設定ファイルのオープンまたは読み込みができません。
[UPR※1]<DATE>Failed to parse setting file.<file=FILEPATH><line=LINE>	<ul style="list-style-type: none"> • DATE※2 ユーザー拡張性能解析トレース設定ファイルの誤りを検出した日時。 • FILEPATH ユーザー拡張性能解析トレース設定ファイルの絶対パス。 • LINE ユーザー拡張性能解析トレース設定ファイルの行番号。 	ユーザー拡張性能解析トレース設定ファイルの内容に記述フォーマット上の誤りがあります。
[UPR※1]<DATE>Event ID is invalid value.<file=FILEPATH><eventID=EventID>	<ul style="list-style-type: none"> • DATE※2 ユーザー拡張性能解析トレース設定ファイルの誤りを検出した日時。 • FILEPATH ユーザー拡張性能解析トレース設定ファイルの絶対パス。 • EventID ユーザー拡張性能解析トレース設定ファイルに指定したイベント ID。 	ユーザー拡張性能解析トレース設定ファイルのイベント ID が有効な値の範囲外です。
[UPR※1]<DATE>No valid settings in setting file.<file=FILEPATH>	<ul style="list-style-type: none"> • DATE※2 ユーザー拡張性能解析トレース設定ファイルに有効な設定がないことを検出した日時。 • FILEPATH ユーザー拡張性能解析トレース設定ファイルの絶対パス。 	ユーザー拡張性能解析トレース設定ファイルに有効な設定がありません。
[UPR※1]<DATE>User Extended PRF started successfully.<file=FILEPATH>	<ul style="list-style-type: none"> • DATE※2 ユーザー拡張性能解析トレース設定ファイルの読み込みに成功した日時。 • FILEPATH 	ユーザー拡張性能解析トレース設定ファイルが正常に読み込まれ、ユーザー拡張性能解析トレースが有効になりました。

出力形式	出力内容	説明
	有効に読み込みが行われたユーザー拡張性能解析トレース設定ファイルの絶対パス。	

注※1

ユーザー拡張性能解析トレースが出力したログであることを示す識別子です。

注※2

拡張 verbosegc 情報と同じ形式で出力します。

- ユーザー拡張性能解析トレースでは、JavaVM 内のクラスや Application Server 内のクラスはトレース対象のメソッドに指定できません。次のパッケージが該当します。

- java 以下
- javax 以下
- com.hitachi 以下
- JP.co.Hitachi 以下

そのため、`jvm.userprf.ExtendedSetting` プロパティを指定して、パッケージ名指定をする場合は、アプリケーションのクラスだけが含まれるようにパッケージ名を記述してください。

- 次のメソッドはトレース対象のメソッドに指定できません。
 - 存在しないパッケージ名、クラス名、およびメソッド名
 - native メソッド
 - abstract メソッド
 - JavaVM 内のクラス、およびそのクラスのメソッド
(例) java および javax で始まるパッケージのクラス
 - `-Xbootclasspath` で指定したクラス、およびそのクラスのメソッド
 - 製品内のクラス
- 次のメソッドは次の記述でトレース対象のメソッドに指定できます。
 - コンストラクターは、クラス名と同じメソッド名、または `<init>` で指定できます。
(例) MyMain クラスのコンストラクターの場合、次のように指定します。
MyMain.MyMain() または MyMain.<init>()
 - コンストラクターではないクラス名と同じ名前のメソッドを指定した場合は、コンストラクターを指定しているのか、メソッドを指定しているのか判別できないためコンストラクターとメソッドがトレースの対象となります。
 - 引数が可変長であるメソッドを指定する場合は、可変長の引数を配列として記述してください。
(例) 引数が可変長であるメソッドの指定
正しい指定例：`com.sample.Test.method(java.lang.String[])`

誤った指定例：com.sample.Test.method(java.lang.String...)

- ネストクラスは、「.」ではなく「\$」で区切った名前を指定してください。

(例) ネストクラスの指定

正しい指定例：com.sample.Test\$NestClass

誤った指定例：com.sample.Test.NestClass

- ジェネリックスではなく、パラメーター化されていないクラス名 (raw 型) であれば指定できます。

(例) ジェネリックスではないクラスの指定

正しい指定例：com.sample.Test.method()

誤った指定例：com.sample.Test<java.lang.String,java.lang.Object>.method()

- 指定形式のクラス名にクラス名を指定した場合とインターフェース名を指定した場合にはトレース対象メソッドが異なります。

クラスまたはインターフェースの指定によるトレース対象メソッドを次の表に示します。

クラスまたはインターフェース	トレース対象メソッド	
	サブクラスフラグにfalseを指定した場合	サブクラスフラグにtrueを指定した場合
クラス	指定したクラスのメソッド	指定したクラスのメソッドと、そのメソッドをオーバーライドしているメソッド
インターフェース	なし	指定したインターフェースを直接実装しているクラス ^{※1} と間接実装しているクラス ^{※2} のメソッド、およびそのメソッドをオーバーライドしているメソッド

注※1

クラスの宣言時に implements を使用して指定したインターフェースを実装しているクラスです。

注※2

指定したインターフェースを直接実装しているクラスのサブクラス、または指定したインターフェースを継承しているインターフェースを直接実装しているクラスです。

ユーザー拡張性能解析トレース設定ファイルの記述形式の例を次に示します。

```
com.sample.Test.method(), TEST1, false, 0xae02, A
com.sample.Test.method(), TEST1, false, 0xae02
com.sample.Test.method(), TEST1, false, 0xae02,
com.sample.Test.method(), TEST1, false, , A
com.sample.Test.method(), TEST1, false
com.sample.Test.method(), TEST1, false,
com.sample.Test.method(), TEST1, false, ,
```

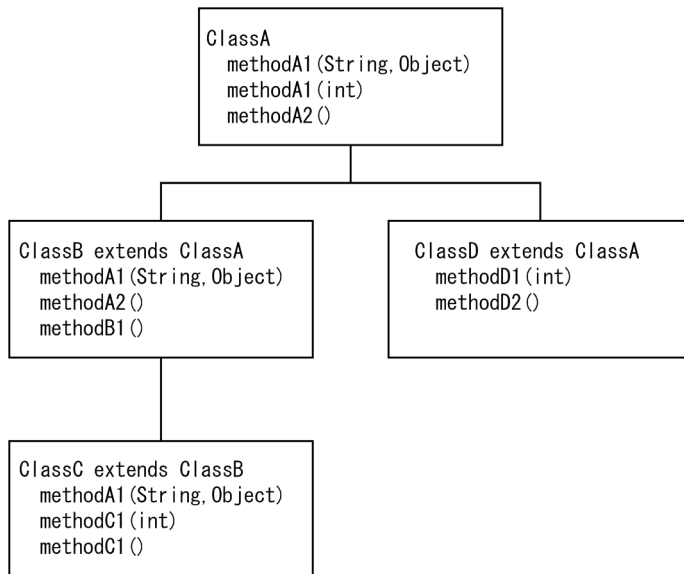
格納先

JDKのインストールディレクトリー¥usrconf¥userprf.cfg

記述例

ここで説明する記述例は、パッケージ名が「com.sample」で、次の図に示すクラス構造を持っているアプリケーションとします。

図 8-3 アプリケーションのクラス構造の例



メソッド名と引数の型が一致するメソッドをトレースの対象とする場合のユーザー拡張性能解析トレース設定ファイルの記述例を次に示します。

サブクラスフラグがfalseの場合

```
com.sample.ClassA.methodA1(java.lang.String, java.lang.Object), 1000, false
```

サブクラスフラグがfalseの場合、記述したメソッド名と引数の型が一致するメソッドがトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA クラスの methodA1(String, Object)

トレース対象メソッドに対してイベント ID の設定を省略しているため、トレース対象となったメソッドが呼び出されると、イベント ID としてメソッドの入口で 0xae00、メソッドの出口で 0xae01 のデフォルト値が出力されます。また、識別 ID に「1000」を設定しているため、識別 ID として「1000」が出力されます。

サブクラスフラグがtrueの場合

```
com.sample.ClassA.methodA2(), 2000, true
```

サブクラスフラグがtrueの場合、メソッド名と引数の型が記述と一致するメソッドに加えて、記述したメソッドをオーバーライドしているメソッドもトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA クラスの methodA2()

- ClassA クラスの methodA2() をオーバーライドしている ClassB クラスの methodA2()

トレース対象メソッドに対してイベント ID の設定を省略しているため、トレース対象となったメソッドが呼び出されると、イベント ID としてメソッドの入口で 0xae00、メソッドの出口で 0xae01 のデフォルト値が出力されます。また、識別 ID に「2000」を設定しているため、トレース対象のすべてのメソッドに識別 ID として「2000」が出力されます。

メソッド名が一致するメソッドをトレースの対象とする場合のユーザー拡張性能解析トレース設定ファイルの記述例を次に示します。

サブクラスフラグが false の場合

```
com.sample.ClassA.methodA1(*),methodA1,false,0xae30
```

サブクラスフラグが false の場合、記述したメソッド名と一致するメソッドのすべてがトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA クラスの methodA1(String,Object)
- ClassA クラスの methodA1(int)

トレース対象となったメソッドが呼び出されると、イベント ID としてメソッドの入口で 0xae30、メソッドの出口で 0xae31 が出力されます。また、トレース対象のすべてのメソッドに識別 ID として「methodA1」が出力されます。

サブクラスフラグが true の場合

```
com.sample.ClassA.methodA1(*),methodA1,true,0xae30
```

サブクラスフラグが true の場合、記述したメソッド名と一致するメソッドのすべてがトレース対象となります。また、記述したメソッドに加えて、記述したメソッドをオーバーライドしているメソッドもトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA クラスの methodA1(String,Object) と methodA1(int)
- ClassA クラスの methodA1(String,Object) をオーバーライドしている ClassB クラスの methodA1(String,Object)
- ClassB クラスの methodA1(String,Object) をオーバーライドしている ClassC クラスの methodA1(String,Object)

トレース対象となったメソッドが呼び出されると、イベント ID としてメソッドの入口で 0xae30、メソッドの出口で 0xae31 が出力されます。また、トレース対象のすべてのメソッドに識別 ID として「methodA1」が出力されます。

メソッドと引数を省略して、クラス名が一致するすべてのメソッドをトレースの対象とする場合のユーザー拡張性能解析トレース設定ファイルの記述例を次に示します。

サブクラスフラグがfalse の場合

```
com.sample.ClassA, TEST01, false
```

サブクラスフラグがfalse の場合、記述したクラス名 (ClassA クラス) のすべてのメソッドがトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA クラスの methodA1(String,Object)、methodA1(int)、および methodA2()

トレース対象となったメソッドが呼び出されると、イベント ID が省略されているため、メソッド入口で 0xae00、メソッド出口で 0xae01 が出力されます。また、これらすべてのメソッドに識別 ID として「TEST01」が出力されます。

サブクラスフラグがtrue の場合

```
com.sample.ClassB, TEST02, true
```

サブクラスフラグがtrue の場合、記述したクラス名 (ClassB クラス) のすべてのメソッドと、そのメソッドをオーバーライドしているメソッドのすべてがトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassB クラスの methodA1(String,Object)
- ClassB クラスの methodA2()
- ClassB クラスの methodB1()
- ClassB クラスの methodA1(String,Object)をオーバーライドしている ClassC クラスの methodA1(String,Object)

トレース対象となったメソッドが呼び出されると、イベント ID が省略されているため、メソッド入口で 0xae00、メソッド出口で 0xae01 が出力されます。また、これらすべてのメソッドに識別 ID として「TEST02」が出力されます。

クラス名とメソッド名と引数を省略して、パッケージ名が一致するすべてのクラスのすべてのメソッドをトレースの対象とする場合のユーザー拡張性能解析トレース設定ファイルの記述例を次に示します。

❗ 重要

この指定ではサブパッケージも対象に含まれます。トレース対象となった場合、対象となったメソッドが呼び出されるとトレースが出力されます。

com.sample パッケージにサブパッケージがある場合、そのサブパッケージのすべてのクラスのすべてのメソッドもトレース対象となります。

サブクラスフラグがfalse の場合

```
com.sample.*, 6000, false
```


サブクラスフラグがfalse の場合、記述したパッケージ (com.sample) にあるすべてのクラスのすべてのメソッドがトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA、ClassB、ClassC、ClassD のすべてのメソッド

トレース対象となったメソッドが呼び出されると、イベント ID が省略されているため、メソッド入口で 0xae00、メソッド出口で 0xae01 が出力されます。また、これらすべてのメソッドに識別 ID として「6000」が出力されます。

サブクラスフラグがtrue の場合

```
com.sample.*,6000,true
```

サブクラスフラグがtrue の場合、記述したパッケージ (com.sample) にあるすべてのクラスのすべてのメソッドと、そのメソッドをオーバーライドしているメソッドのすべてがトレース対象となります。

トレース対象のメソッドを次に示します。

- ClassA、ClassB、ClassC、ClassD のすべてのメソッドと、そのメソッドをオーバーライドしているすべてのメソッド

トレース対象となったメソッドが呼び出されると、イベント ID が省略されているため、メソッド入口で 0xae00、メソッド出口で 0xae01 が出力されます。また、これらすべてのメソッドに識別 ID として「6000」が出力されます。

注意事項

- ユーザー拡張性能解析トレース設定ファイルで、同じメソッドに対して複数行で異なるイベント ID または識別 ID を指定した場合、ユーザー拡張性能解析トレース設定ファイルの前方に記述された設定が優先されます。
- ユーザー拡張性能解析トレース設定ファイルで、複数のメソッドを対象とする記述をした場合、対象となるすべてのメソッドが同一イベント ID または識別 ID で出力されます。その場合、出力されるメソッド名が出力できる文字数を超えるとメソッドを特定できないことがあります。メソッドを 1 つに特定できるように記述してください。
- トレース対象にインターフェースを指定した場合、サブクラスフラグにtrue を指定してください。インターフェースにはトレース対象となるメソッドが存在しないため、サブクラスフラグにfalse を指定するとトレース情報は出力されません。
- 複数のメソッドをユーザー拡張性能解析トレースの対象とする指定形式は、アプリケーションの動作を把握したい場合などに限り指定してください。ユーザー拡張性能解析トレース設定ファイルの記述で、サブクラスフラグにtrue を指定した場合や、メソッド名が一致するメソッドをトレースの対象とする指定形式で指定した場合、ユーザーが意図しない多くのメソッドがユーザー拡張性能解析トレースの対象となり、性能劣化要因の特定が困難になることがあります。性能劣化要因の特定のためには、ユーザー拡張性能解析トレース設定ファイルのサブクラスフラグにfalse を指定する、またはメソッド名と引数の型が一致するメソッドをトレースの対象とする指定形式で指定するなど、ユーザー拡張性能解析トレースの対象を限定することを推奨します。

索引

数字

- 1 スタック領域の最大サイズの指定 745
- 1つのメソッドのパラメーターの完全修飾 Java 型名の指定 415
- 1つのログファイルに書き込める上限の指定 550
- 200番台のステータスコード応答時のレスポンスヘッダーのカスタマイズ 119

A

- AccessFileName ディレクティブで指定したファイルでのアクセス情報定義の上書き許可 94
- activation-config で設定するプロパティー値の指定 340
- activation-config で設定するプロパティーの指定 339
- activation-config で設定するプロパティー名称の指定 340
- activation-config 要素の説明の記述 339
- admin-object-resource 要素の説明の指定 374
- Allow ディレクティブと Deny ディレクティブの指定の評価の順序 157
- asadmin プロセスおよびドメイン管理サーバに設定する HTTP リードタイムアウト値の指定 516
- asadmin ユーティリティーコマンドのスタックトレースログの有効または無効の指定 529
- asadmin ユーティリティーコマンドのスタックトレースログファイルの切り替え時刻 531
- asadmin ユーティリティーコマンドのスタックトレースログファイルのサイズ 531
- asadmin ユーティリティーコマンドのスタックトレースログファイルの面数 530
- asadmin ユーティリティーコマンドのプロセス起動時ログの有効または無効の指定 517
- asadmin ユーティリティーコマンドのプロセス起動時ログファイルの切り替え時刻の指定 519
- asadmin ユーティリティーコマンドのプロセス起動時ログファイルのサイズ 518
- asadmin ユーティリティーコマンドのプロセス起動時ログファイルの面数 517

- asadmin ユーティリティーコマンドのプロセスに適用する JavaVM オプションの指定 520
- asadmin ユーティリティーコマンドの保守ログの有効または無効の指定 521
- asadmin ユーティリティーコマンドの保守ログのレベル 523
- asadmin ユーティリティーコマンドの保守ログファイルの切り替え時刻 524
- asadmin ユーティリティーコマンドの保守ログファイルのサイズ 524
- asadmin ユーティリティーコマンドの保守ログファイルの面数 522
- asadmin ユーティリティーコマンドのメッセージログの有効または無効の指定 525
- asadmin ユーティリティーコマンドのメッセージログのレベル 527
- asadmin ユーティリティーコマンドのメッセージログファイルの切り替え時刻 528
- asadmin ユーティリティーコマンドのメッセージログファイルのサイズ 527
- asadmin ユーティリティーコマンドのメッセージログファイルの面数 526
- asadmin ユーティリティーコマンドのログ出力先ディレクトリーの分割の有効または無効の指定 521
- auth-method 要素で指定された認証方式がクライアント認証に使用される必要があるかどうかの指定 333
- AUTH コマンドを使用したユーザー認証の指定 556

C

- CA の証明書のファイルの指定 198
- CGI スクリプトに渡す環境変数 193
- CGI スクリプトに渡す環境変数から SetEnv ディレクティブまたは PassEnv ディレクティブで指定した環境変数を削除する場合の指定 218
- CGI スクリプトに渡す任意の環境変数 158
- CGI スクリプトのエラーログ出力先 186
- CGI スクリプトのエラーログファイルの最大サイズ 187
- CGI スクリプトの実行に使用されるインタープリターの指定 185

CGI プログラムを実行する URL と CGI プログラムが配置されているディレクトリー名の指定 183

CGI プログラムを実行する URL に対する正規表現と CGI プログラムが配置されているディレクトリー名の指定 184

Compressed Class Space の最大値の指定 728

Concurrent Marking の処理を並列実行するスレッド数の指定 729

connector-connection-pool 要素の説明の指定 391

connector-resource 要素の説明の指定 376

Cookie を送信してきたリクエストだけを処理するときのサーバスレッド数 171

CopyGC 実行時に From 空間と To 空間で Java オブジェクトを入れ替える回数のしきい値の指定 733

CRL の次回発行日を経過した場合の動作の指定 201

custom-resource 要素の説明の指定 364

D

DER 形式の CRL を格納するディレクトリーの指定 202

DSO によって組み込むモジュールが参照するコードがあるオブジェクトファイルまたはライブラリー 151

E

EHLO コマンドによるサインオンの指定 558

EIS が必要とするユーザー名およびパスワードの指定 393

EIS グループの Java EE Server のサーバインスタンスでのグループへのマップ 399

EIS でのプリンシパルの Java EE Server のサーバインスタンスでのプリンシパルのマッピング 399

EJB Web サービスエンドポイントの認証のための設定 342, 411

ejb-jar.xml で定義した Enterprise Bean の ejb-name 要素の値の指定 302

EJB-JAR に含まれる Enterprise Bean のための実行時の設定を指定するための要素 301

EJB-JAR の実行時の設定をするための要素 301

EJB-JAR の設定をするための Java EE RI の DD 295

EJB-JAR の設定をするための拡張 DD 473

ejb-ref 要素の ejb-ref-name の指定 303

ejb-ref を定義する Enterprise Bean の JNDI 完全修飾名の指定 304

EJB のインスタンスプールが保持する、Bean の最大数の指定 337

EJB のインスタンスプールが保持する Bean の初期値と最小数の指定 335

EJB のインスタンスプールで、アイドル状態で保持される最大時間の指定 337

EJB のインスタンスプールに対して、同時に作成または削除される Bean のインスタンス数の指定 336

EJB のインスタンスプールの設定 335

EJB のホームオブジェクトの JNDI 完全修飾名のマッピングの定義 303

EJB モジュール内の Web サービスの名称の指定 356

EJB モジュール内のポートコンポーネントに一意の名称の指定 341

EJB モジュールのバージョン情報の定義 358

Enterprise Bean から参照されている WSDL サービスの指定 321

enterprise-beans 要素の名称の指定 301

ETag レスポンスヘッダーフィールドを作成するために使用されるファイルの属性値 116

Explicit ヒープ全体の最大サイズの指定 694

Explicit ヒープへ移動させたくないオブジェクトのクラスを指定する設定ファイル（明示管理ヒープ機能適用除外設定ファイル） 753

Explicit メモリーブロックの解放処理でのオブジェクトの移動先の Tenured 領域への変更 702

Explicit メモリーブロックの確保方法を 08-00 と同様にするかどうかの指定 697

Explicit メモリーブロックの自動配置機能で利用する自動配置設定ファイルのパスの指定 692

Explicit メモリーブロックの自動配置機能の有効または無効の設定 691

external-jndi-resource 要素の説明の指定 366

F

FullGC 発生時の参照関係に基づくオブジェクトの Java ヒープから Explicit メモリーブロックへの移動の制御 687

G

G1GC の GC によるアプリケーション停止時間の目標時間の指定 730

G1GC の実行 740

G1GC を並列実行するスレッドの数の指定 736

GC イベントが発生するたびに file へのログの出力 742

GC が発生した場合に拡張 verbosegc 情報の JavaVM ログファイルへの出力 626

GC 実行後の Survivor 領域内で Java オブジェクトが占める割合の目標値の指定 739

GC の要因内容の出力 638

GMS-BIND-INTERFACE-ADDRESS-cluster-name システムプロパティの指定 496

H

HTTPS リクエストをリッスンするポートのポート番号の指定 499

HTTP セッションで利用する Explicit ヒープの省メモリ化機能の有効または無効の設定 695

HTTP リクエストをリッスンするために使用するポート番号の指定 498

I

IIOIP 接続で使われるポートのポート番号の指定 499

information-type に指定した情報の出力 742

IOR のセキュリティー情報の指定 328

J

JAAS 認証レールのオプションの設定 239

Java EE Server がシャットダウン時に非 XA トランザクションをコミットするかロールバックするかの指定 487

Java EE Server 固有の設定の定義 298

Java EE Server で使用するすべてのシステムプロパティの概要 481

Java EE Server で使用するすべての定義ファイルの概要 510

Java EE Server で使用するすべてのプロパティファイルの概要 510

Java EE Server の環境変数定義ファイル 515

Java EE Server のログファイルの出力先ディレクトリー 532

Java EE アプリケーションの設定をするための Java EE RI の DD 247

Java EE アプリケーションの設定をするための拡張 DD 469

Java EE サーバで使用するすべての DD の概要 234

Java EE 標準 API で設定できる Java EE Server 独自のプロパティの概要 554

Java Platform Debugger Architecture (JPDA) デバッガーとの接続に使用されるポート番号の指定 503

java:comp/env で始まる Web サービス参照名の指定 260, 261, 279, 280, 309, 310, 437, 438

java.io.File.deleteOnExit() の呼び出しで JavaVM が確保した累積のヒープサイズとメソッドの呼び出し回数の JavaVM ログファイルへの出力 642

JavaMail リソースの定義 370

JavaVM 拡張オプションの一覧の表示 579

JavaVM で使用するオプションの概要 569

JavaVM で使用する環境変数の概要 569

JavaVM で使用する定義ファイルの概要 752

JavaVM で使用するプロパティの概要 569

JavaVM 内部で管理しているヒープ情報の JavaVM ログファイルへの出力 646

JavaVM の標準出力と標準エラー出力の内容の JavaVM ログファイルへの出力 614

JavaVM ログファイルオープン時のヘッダー情報の出力 610

JavaVM ログファイルに出力する日時のミリ秒までの出力 616

JavaVM ログファイルの 1 ファイル当たりの最大ファイルサイズの指定 606

JavaVM ログファイルの作成する最大ファイル数の指定 608

JavaVM ログファイルのプリフィクスおよびログファイルの出力先ディレクトリーの指定 605

JavaVM ログファイルのローテーション時刻の指定 612

JavaVM ログファイルのローテーション方式の指定 613

javaw コマンドを実行可能にするための環境変数 749

javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値の指定 283, 287, 441, 447

javax.xml.rpc.Call オブジェクトに設定される JAX-RPC プロパティの値の指定 313, 319

javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値の指定 263, 282, 440

javax.xml.rpc.Stub オブジェクトに設定される JAX-RPC プロパティの値の指定 312

Java アプリケーションプログラムの実行 711

java コマンドを実行可能にするための環境変数 747

Java スレッドの数の JavaVM ログファイルへの出力 650

Java ヒープの最大サイズの指定 744

Java ヒープの初期サイズの指定 743

Java メソッドの JIT コンパイル対象時にコードキャッシュ領域が枯渇したときのメッセージの JavaVM ログファイルへの出力 654

JAX-RS のサブレットに対するリソースとプロバイダーの定義 479

JAX-WS の製品内部処理で利用する JAXB のバインディングプロバイダーの指定 492

jdbc-connection-pool 要素の説明の指定 385

jdbc-resource 要素の説明の指定 368

JDBC コネクションプールに関する設定の定義 379

JDBC リソースの定義 368

JIT コンパイラ稼働継続機能の有効または無効の設定 709

JIT コンパイル時に確保する C ヒープのメモリーサイズの上限值の指定 705

JMS プロバイダーのポート番号の指定 504

JMX コネクタがリッスンするポート番号の指定 505

JNDI 完全修飾名の ejb-ref へのマッピング 254

JNDI 完全修飾名の ejb-ref 名へのマッピング 273

JNDI 完全修飾名の resource-env-ref へのマッピング 258, 277

JNDI 完全修飾名の resource-ref へのマッピング 255, 274

JNDI 完全修飾名の指定 303

JSP の設定情報の指定 454

K

KeepAlive 接続時のリクエスト待ち時間 145

KeepAlive 接続を有効にするかどうかの指定 144

KeepAlive 連続接続回数の上限 154

M

mail-resource 要素の説明の指定 371

Message-driven Bean に継続的に関連づけられるサブスクリプションの指定 327

Message-driven Bean が一度に JMS セッションに読み込む最大メッセージ数の指定 328

Message-driven Bean に関連づける ConnectionFactory の参照を取得するための JNDI 完全修飾名の指定 325

Message-driven Bean に関連づける ConnectionFactory の指定 325

Message-driven Bean にメッセージを配信するリソースアダプターに渡す activation-config の指定 339

Message-driven Bean の実行時の設定 338

Metaspace 領域に起因する FullGC の基準値の指定 733

Metaspace 領域の最大値の指定 731

method パラメーターの完全修飾 Java 型名の指定 444

MIME タイプのドキュメントに対する有効期限の指定 113

N

New 領域に対する Tenured 領域の割合の指定 735

O

OutOfMemoryError 発生時に強制終了する機能の有効または無効の設定 660

OutOfMemoryError 発生時に出力するスタックトレース 1 行の文字数のバイト数での指定 667

OutOfMemoryError 発生時に出力するスレッドダンプログファイルへのクラス別統計情報の出力 662

OutOfMemoryError 発生時に要求したメモリーのサイズの出力 664

OutOfMemoryError 発生時のスレッドダンプの出力
661

OutOfMemoryError 発生時の発生要因種別の出力
663

OutOfMemoryError 発生時の例外情報とスタック
トレースの JavaVM ログファイルへの出力 665

P

PEM 形式の CRL を格納するディレクトリーの指定
202

PRF デーモン起動時に PRF 識別子に指定した文字列
の指定 128

PRF デーモンを使用するときに必要な PRFSPOOL 環
境変数に設定するディレクトリー名の指定 128

property 要素の説明の指定 367, 369, 372, 375,
377, 379, 387, 397

Q

QNAME のローカルパートの指定 262, 265, 282,
290, 352, 420, 440, 449

QNAME のローカルパートの指定 312, 322

R

resource-env-ref 要素に対する参照のマッピングの
定義 308

resource-env-ref を定義するリソース JNDI 完全修飾
名の指定 308

resource-ref 要素内の res-ref-name の指定 256,
275

resource-ref 要素に対する参照のマッピングの定義
304

resource-ref を定義するリソース JNDI 完全修飾名の
指定 306

run-as ロールが指定された Enterprise Bean 内のプ
リンシパル (ユーザー) 名の指定 324

S

SerialGC の実行 741

server.policy ファイルのロケーションの指定 502

Server ヘッダーのフォーマット 192

SMTP HELO コマンドまたは EHLO コマンドで使用
するローカルホストの名前の指定 558

SSL 以外によるアクセスの禁止 209

SSL セッションの有効時間 211

SSL 接続で使用できる SSL プロトコルの指定 (IMAP
サーバ接続の場合) 565

SSL 接続で使用できる SSL プロトコルの指定 (SMTP
サーバ接続の場合) 560

SSL 通信で使用できる暗号種別の指定 208

SSL で認証する場合の Web サーバの証明書のファイ
ルの指定 199

SSL で認証する場合の Web サーバの秘密鍵のファイ
ルの指定 200

SSL によるアクセス時だけに Cookie を送信させるか
どうかの指定 171

SSL によるアクセスの禁止 203

SSL ハンドシェイク処理中にログに出力されるエラー
の詳細情報を表示するかどうかの指定 126

SSL を無効にする場合の指定 203

SSL を有効にする場合の指定 204

STATUS コマンドの応答をキャッシュするタイムアウ
ト値の指定 565

Survivor 領域の From 空間と To 空間に対する Eden
領域の割合の指定 738

Survivor 領域の年齢分布の JavaVM ログファイルへ
の出力 648

T

target-server 要素以下に含まれる要素の説明の記述
238

TRACE メソッドによるリクエストを拒否するかどう
かの指定 215

TypesConfig ディレクティブで指定したファイルに
未定義のコンテンツの拡張子と MIME タイプの関連
づけ 90

U

URL のファイルシステムのパスへの置き換え 91

URL リソースまたはリソースの JNDI 完全修飾名の
指定 255, 256, 259, 266, 432, 434, 436, 451, 465

V

Via ヘッダーの使用の制御 166

W

web.xml に定義された run-as ロールに使用されるサーブレットのプリンシパル名の指定 407

web.xml の servlet-name に適合するサーブレット名の指定 408

webservice-endpoint 要素のプロパティの設定 354

Web アプリケーション (WAR) の設定をするための Java EE RI の DD 401

Web アプリケーション (WAR) の設定をするための拡張 DD 477

Web アプリケーションのバージョン情報の指定 467

Web サーバが作成するエラーメッセージなどのコンテンツのフッターに署名するかどうかの指定 191

Web サーバ内の圧縮データを Web ブラウザーに表示させるときに必要な拡張子と圧縮形式の関連づけの指定 85

Web サーバへアクセスできるクライアントの制限 93, 107

Web サーバに動的に組み込むモジュール 151

Web サーバの TCP 送信バッファサイズ 188

Web サーバのサーバ名およびポート番号 189

Web サーバの送受信の待ち時間 214

Web サーバの動作環境を設定するための定義ファイルおよび定義項目の概要 46

Web サーバの動作環境を設定するファイル 59

Web サービスエンドポイントでデバッグを有効にするかどうかの指定 353

Web サービスエンドポイントの情報の指定 341, 409

Web サービス参照内のポート情報の指定 438

Web サービス参照内のポートの情報の指定 261, 280, 310

Web サービス参照のランタイム設定の指定 260, 279, 437

Web サービスの参照のための実行時設定 309

Web サービスの操作のオペレーション名の指定 285, 317, 348, 416, 445

Web サービスの名称とオプションの WSDL の配置先の指定 356

Web サービスの名称と任意のパブリッシュする場所の指定 466

Web 内の Web サービスの名称を指定 466

Web ブラウザーからの/~ユーザー名/へのリクエストに対して公開するサーバ上の場所の指定 220

Web ブラウザーごとの環境変数の設定 100, 101

Web モジュール内にあるポートコンポーネントの一意の名称の指定 410

Web モジュールへの Java EE Server 固有の設定の定義 404

Web アプリケーションの Web コンテキストルート 250, 405

Windows ログオフ時の JavaVM の実行の継続 727

work-security-map 要素の説明の指定 398

WSDL ポートの指定 262, 281, 311

WSDL ポートの指定 439

あ

アーカイブファイルの名称の指定 268

アクセス制御ファイルのファイル名の定義 79

アクセス制御を定義するアクセス制御ファイル 223

アクセス制限の定義 181

アクティブなメッセージコンシューマーの最大数の指定 540

アクティベーションコンフィグレーションプロパティの指定 339

圧縮オブジェクトポインター機能の有効または無効の設定 713

アプリケーションクライアントコンテナの設定をするための Java EE RI の DD 235

アプリケーションクライアントコンテナへの Java EE Server 固有の設定の定義 237

アプリケーションクライアントの設定をするための Java EE RI の DD 271

アプリケーションクライアントの設定をするための拡張 DD 471

アプリケーションクライアントへの Java EE Server 固有の設定の定義 273

アプリケーションスコープのリソースの定義 363

アプリケーションに Java EE Server 固有の設定の定義 249

アプリケーションの Web tier の設定 249

アプリケーションの Web URI 249

アプリケーションのリクエスト処理の認証ポリシー要件の定義 286, 318, 349, 416

アプリケーションのレスポンス処理の認証ポリシー要件の定義 286, 319, 349, 417

アプリケーションまたはモジュールのバージョン情報 269, 293

ある特定のハンドラーに関連づける場合の指定 197

い

イメージマップファイルの base 行のデフォルトの指定 136

イメージマップファイルの default 行のデフォルトの指定 136

インバウンドのリソースアダプターの管理対象オブジェクトの定義 373

え

エラーが発生したときに Web ブラウザーへ表示するメッセージに対応した文字セット 122

エラーが発生したときに Web ブラウザーへ表示するメッセージのカスタマイズ 109

エラーログに出力するエラーのレベル 153

エラーログに出力するクライアントアドレスを X-Forwarded-For ヘッダー値に変更 123

エラーログの出力 112

か

外部の JNDI リポジトリに存在するリソースの定義 365

拡張 verbosegc 情報の CSV 形式での出力 618

拡張 verbosegc 情報への GC 開始日時の出力 640

拡張 verbosegc 情報へのユーザー CPU 時間とカーネル CPU 時間の出力 635

拡張 verbosegc 情報を秒単位で出力する時間の間隔の指定 636

拡張スレッドダンプ情報の出力 582

拡張スレッドダンプのスレッド情報へのスレッドが処理をブロックした回数と待ち状態になった回数の出力 600

拡張スレッドダンプのスレッド情報へのハッシュコードの出力 603

拡張スレッドダンプのスレッド情報へのユーザー CPU 時間とカーネル CPU 時間の出力 601

拡張スレッドダンプの標準出力への出力 599

カスタム認証プロバイダーのバインディングの指定 284, 345, 412, 441

カスタム認証プロバイダーのバインディングの定義 314

カスタムバルブの指定 462

カスタムリソースの定義 363

環境変数 SSL_CLIENT_CERT_CHAIN_n にクライアントの証明書を発行した CA からルート CA までの証明書を設定するときの指定 204

環境変数 SSL_CLIENT_CERT にクライアント証明書を設定するときの指定 205

関連づけられているエンティティの jndi-name の指定 268, 291, 292

き

キュー送信先の自動作成の有効または無効の指定 539

許容されるログインのリトライ回数の決定 508

く

クライアントが使用言語の優先度を示さなかった場合の優先順位の指定 146

クライアントから受信した Host ヘッダーの値をそのままバックエンドサーバに転送するかどうかの指定 164

クライアントからの接続要求の最大の待ち行列数 150

クライアントからのリクエストについての最大待ちリクエスト数 127

クライアントからのリクエストを基に環境変数の定義 194, 196

クライアントからのリクエストをリダイレクトする場合の指定 175

クライアント側での JAX-WS のリクエストおよびレスポンスを標準出力する機能の設定 493

クライアント証明書を利用した Basic 認証 206

クライアントとサーバ間の通信の指定 350, 418

クライアント認証時の証明書に関する設定 212

クライアント認証を伴うセキュア IIOP 接続で使われるポート番号の指定 501

クライアントの認証に使用される認証の方法の指定
331

クライアントまたはサーバの IPv6 アドレスによる環境変数の設定 132

クラスターインスタンスの作成時の External IP アドレスとグループ管理サービス IP アドレスの設定 496

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番号の定義 497

クラスライブラリーのスタックトレースの 1 行の文字数のバイト数での指定 670

クラスライブラリーのスタックトレースの出力 669

クラスローダーの設定情報の指定 451

け

計画停止時に強制停止させたリクエスト情報をエラーログファイルに出力するかどうかの指定 123

計画停止時に実行中のリクエストを強制的に終了させるまでの待ち時間 124

現在アクティブなレルムのユーザーまたはグループへのロールのマップ 405

現在のレルム内のグループ名 252

現在のレルム内のプリンシパル (ユーザー) 名 251

現在のレルムにグループ名を指定 407

現在のレルムのグループ名の指定 300

現在のレルムのプリンシパル (ユーザー) 名の指定
299, 406, 408

こ

コードキャッシュ領域の最大サイズの指定 737

コードキャッシュ領域の使用率のしきい値の指定 653

コードキャッシュ領域の使用量の出力としきい値に達したことを知らせるメッセージの JavaVM ログファイルへの出力の指定 656

コネクターコネクションプールの定義 388

コネクターリソースの定義 375

コンテンツのファイル拡張子とコンテンツタイプの関連づけを定義するファイル 222

コンテンツへのアクセスがユーザー認証とホスト名または IP アドレスの両方によって制限されている場合のアクセス条件 182

コンテンツを格納するドキュメントルートディレクトリーの指定 109

コンテンツネゴシエーションをするリクエストでクライアント側のキャッシュを有効にするかどうかの指定
101

コンテンツネゴシエーションの対象となる拡張子の種類 155

さ

サーバインスタンスでのプリンシパルの定義 392

サーバインスタンスでのユーザー名の指定 392

サーバが受信する 1 つの HTTP ヘッダーのサイズの上限 148

サーバが受信する HTTP ヘッダー数の上限 147

サーバが受信するオブジェクトボディーのサイズの上限 147

サーバが受信するリクエストラインの長さの上限 149

サーバ側での 8BITMIME 拡張サポートの指定 556

サーバ側での JAX-WS のリクエストおよびレスポンスを標準出力する機能の設定 494

サーバ管理者の E-Mail アドレス 189

サーバとして起動するスレッド数 214

サーバの正式な名前の生成方法の指定 219

サーバのルートディレクトリーの指定 191

サーバプロセスが異常終了した場合に共有メモリーに採取されたトレースを出力するファイル名の指定 135

サーバ名に基づくバーチャルホストで Host ヘッダーの代わりにパス名を利用して各ホストに接続する場合の指定 190

サーバ名に基づくバーチャルホストで使用するホスト名の別名 189

サービス実装クラスの名称の指定 321

サブレットまたは EJB の認証中に受信したプリンシパルの EIS でのプリンシパルへのマッピング 391

最後にコミットされたトランザクションの最大数の指定 548

最終的な WSDL ドキュメントを示す有効な URL の指定 320

最終的な WSDL ドキュメントを示す有効な URL の指定 288, 447

サインオンするために使用されるデフォルトのユーザー名の指定 306

サインオンするために使用されるデフォルトのユーザー名に対応したパスワードの指定 307

参照されている WSDL service 要素の指定 264, 289, 351
参照する WSDL サービス要素の指定 419, 448
参照に対して関連づける、物理的なメッセージ送信先の JNDI 名の指定 324

し

時間によるログの分割 228
実行中のプログラムのプロファイルを生成してプロファイリングデータの標準出力への出力 745
指定した暗号種別以外でのアクセスを拒否してステータスコード 403 Forbidden を応答する場合の指定 208
指定した暗号種別でのアクセスを拒否してステータスコード 403 Forbidden を応答する場合の指定 198
自動生成されたエンドポイントのアドレスの指定 342, 410
自動生成されたサーブレット実装クラスの名称の指定 353
受信したすべてのリクエストを拒否するときのサーバスレッド数 173
出力ファイル数の指定 549
使用する SSL プロトコルのバージョンの指定 207
証明書のチェーンを何段階までたどるかの指定 212

す

スタックトレース中のメソッドのローカル変数情報の出力 676
ステータス情報表示時に各リクエストの拡張ステータス情報を表示するかどうかの指定 116
スレッド数の上限値の指定 707
スレッドダンプ出力時のスタックトレースへの各メソッドのローカル変数情報の追加出力 674
スレッドダンプファイルの出力先ディレクトリーの指定 748

せ

正規表現で記述した条件を満たす URL へのリクエストに対するディレクティブの定義 76
正規表現で記述した条件を満たすクライアントからのリクエストをリダイレクトする場合の指定 177

正規表現で記述した条件を満たすディレクトリーに対するディレクティブの定義 73
正規表現で記述した条件を満たすファイルに対するディレクティブの定義 74
正規表現を使用して URL のファイルシステムのパスへの置き換え 92
制御プロセス ID を格納するファイル 159
整合性が保護されたメッセージをターゲットがサポートするかどうかの指定 329
生成されたサービス実装クラスの名称の指定 264, 288, 448
生成されたサーブレット実装クラスの自動生成された名称の指定 421
セキュア IIOP 接続で使われるポートのポート番号の指定 500
セキュリティーマネージャーを有効にする環境で、JPA 利用するための設定 495
セッション Cookie のプロパティーの指定 428
セッションキャッシュサイズの上限值 210
セッションの設定 423, 424
セッションのプロパティーの指定 426
セッションマネージャーのプロパティーの指定 424
接続先の PRF デーモンの PRF 識別子 484
設定ファイルとして使用するファイルの指定 138

そ

送信先ファイルプールが保持するファイルの最大数の指定 543
ソケットの書き込みタイムアウト値の指定 (IMAP サーバ接続の場合) 567
ソケットの書き込みタイムアウト値の指定 (SMTP サーバ接続の場合) 561
ソケットのコネクションタイムアウト値の指定 (IMAP サーバ接続の場合) 563
ソケットのコネクションタイムアウト値の指定 (SMTP サーバ接続の場合) 557
ソケットの読み込みタイムアウト値の指定 (IMAP サーバ接続の場合) 566
ソケットの読み込みタイムアウト値の指定 (SMTP サーバ接続の場合) 561

た

- ターゲットがクライアントに対して認証できるかどうかの指定 330
- ターゲットがクライアントを認証できるかどうかの指定 331
- 対応する Java EE RI の DD で定義された物理的なメッセージ送信先の名称の指定 291
- 対応する Java EE RI の DD ファイル resource-env-ref 要素内の res-ref-name の指定 278
- 対応する Java EE RI の DD ファイルで定義された物理的なメッセージ送信先の名称の指定 266
- 対応する Java EE RI の DD ファイルで定義されたメッセージの物理的な送信先の名称の指定 450
- 対応する Java EE RI の DD ファイルで定義されたメッセージの論理的な送信先の名称の指定 465
- 対応する Java EE RI の DD ファイルで定義された論理的なメッセージ送信先の名称の指定 267, 292
- 対応する Java EE RI の DD ファイルの ejb-ref-name の指定 254, 273, 432
- 対応する Java EE RI の DD ファイルの ejb-ref への JNDI 完全修飾名の割り当て 432
- 対応する Java EE RI の DD ファイルの res-ref-name の値の指定 259
- 対応する Java EE RI の DD ファイルの res-ref-name の指定 436
- 対応する Java EE RI の DD ファイルの role-name 251
- 対応する Java EE RI の DD ファイルの role-name の指定 406
- 対応する Java EE XML ファイル内のロールのユーザーまたはグループへのマッピング 250
- 対応する Java EE XML ファイルの resource-ref への JNDI 完全修飾名の割り当て 433
- 対応する Java EE XML ファイルの resource-env-ref への JNDI 完全修飾名の割り当て 436
- 対象のサーバインスタンスとの IIOP/SSL 通信のセキュリティの設定の指定 239
- 対象のサーバインスタンスの IIOP リスナーとロードバランスで使用される IIOP エンドポイントの指定 237
- 単一メッセージの最大サイズのバイト単位の指定 536

て

- 定期的に発生する GC の最大間隔の指定 505, 506
- 定義リファレンスで使用している記号 43
- 定義リファレンスで使用している構文要素 44
- 定義リファレンスの記述形式 40
- ディレクトリーインデックスで特定のアイコンが設定されていない場合に表示するアイコン 106
- ディレクトリーインデックスの整形表示機能のオプション 139
- ディレクトリーインデックスの整形表示時にファイルに対して表示する説明 84
- ディレクトリーインデックスの整形表示時のアイコンを MIME エンコーディングと対応づけて表示 88
- ディレクトリーインデックスの整形表示時のアイコンを MIME タイプと対応づけて表示 88
- ディレクトリーインデックスの整形表示時のアイコンを拡張子と対応づけて表示 86
- ディレクトリーインデックス表示時に MIME エンコーディングに関連づけてアイコンの代わりに表示する文字列 81
- ディレクトリーインデックス表示時に MIME タイプに関連づけてアイコンの代わりに表示する文字列 81
- ディレクトリーインデックス表示時に Web ブラウザーに表示させないファイル名の指定 139
- ディレクトリーインデックス表示時に拡張子に関連づけてアイコンの代わりに表示する文字列 80
- ディレクトリーインデックス表示時の Readme として付けるコメントを記述したファイルの指定 175
- ディレクトリーインデックス表示時のヘッダーに付けるコメントを記述したファイルの指定 120
- ディレクトリーインデックス表示でのデフォルトのファイルの並び順 143
- データベースに保存されたトランザクションサービスログのサーバ名を取得する際に使用する SQL の定義 490
- データベースに保存されたトランザクションサービスログを削除する際に使用する SQL の定義 488
- データベースに保存されたトランザクションサービスログを取得する際に使用する SQL の定義 490
- デッドメッセージキューへのデッドメッセージの送信の有効または無効の指定 539

デバッグ用のサブレットを Web サービスエンドポイントに対して有効にするかどうかの指定 421

デフォルトで利用するプリンシパル (ユーザー) の指定 306

デフォルトとしてクライアントに送信するコンテンツのファイル 108

デフォルトのプリンシパル (ユーザー) の指定 276

デフォルトのプリンシパル名の指定 435

デフォルトのホスト名または IP アドレスの指定 542

デフォルトのリソースプリンシパル (ユーザー) の指定 257, 434

デフォルトのリソースプリンシパルのパスワードの指定 258, 277, 327, 435

デフォルトのリソースプリンシパル名の指定 276

デプロイメント中に Web サービスの WSDL をパブリッシュするディレクトリーの URL の指定 466

デプロイメント時に Web サービスの WSDL がパブリッシュされるディレクトリーの URL の指定 357

と

ドキュメントで使用する言語 89

ドキュメントで使用するデフォルトの言語 106

特定の HTTP プロトコルメソッドに対するディレクティブの定義 74

特定の MIME タイプまたはハンドラーで指定したコンテンツに対する CGI スクリプトの実行 79

特定の URL に対するディレクティブの定義 76

特定のディレクトリーに対するディレクティブの定義 72

特定のディレクトリー下のすべてのコンテンツに対して使用する MIME タイプ 118

特定のファイルに対するディレクティブの定義 73

特定のホスト名または IP アドレスに対するディレクティブの定義 78

特定のメソッドに対する CGI スクリプトの実行 183

トピックコンシューマーに送信できる最大のメッセージ数の指定 541

トピック送信先の自動作成の有効または無効の指定 541

ドメイン管理サーバがインスタンスを管理するためにインスタンスへ接続する HTTP ポートまたは HTTPS ポートのポート番号の指定 484

ドメイン管理サーバのホスト名または IP アドレスの指定 552

トランザクション完了時に使用されるコミットオプションの指定 334

トランザクションサービスログを保存する際に使用する SQL の定義 489

トランザクションタイムアウト時間の指定 334

トランザクションのステータスファイルの同期を無効に設定 485

トランザクションを利用しないコネクションを使用したデータベースへのトランザクションサービスログの登録 491

トランスポートのセキュリティ情報の指定 329

トレース採取のための共有メモリー ID を格納するファイル名の指定 134

な

名前空間 URI の指定 262, 264, 281, 289, 311, 322, 351, 419, 439, 449

に

任意のフォーマットのアクセスログ出力 102

認証するグループのリストを格納しているファイル名の指定 96

認証するユーザー名とパスワードのリストを格納しているファイル名の指定 98

認証プロバイダーのリクエスト処理の認証ポリシー要件の定義 244

認証プロバイダーのレスポンス処理の認証ポリシー要件の定義 244

認証方式の指定 332, 343, 411

認証リクエスト処理の認証ポリシー要件の定義 445

認証リクエストに対するクライアントからの応答を待機する時間の秒単位の指定 535

認証レスポンス処理の認証ポリシー要件の定義 446

認証レルムの指定 253

ね

ネイティブエージェントライブラリーのロード 727

の

ノードがドメイン管理サーバに接続するための接続情報を含むファイル 551

は

パスワード保護されているサーバ秘密鍵のパスワードを格納するファイルの指定 200

バックアップメッセージコンシューマーの最大数の指定 540

バックエンドサーバから受信した Location ヘッダーの URL の変更 162

バックエンドサーバからのレスポンスがエラーのステータスコードであった場合にレスポンスを上書きするかどうかの指定 159

バックエンドサーバとの送受信に関する待ち時間の指定 165

バックエンドサーバに転送するリクエストと転送するアドレス 161

パブリックアドレスのポートの指定 487

パブリックアドレスのホストの設定 486

ハンドラーで処理するファイル拡張子の定義 85

ハンドラーのデフォルトレベルの指定 549

ひ

標準 DD で指定された message-destination-ref-name の指定 323

標準 DD で指定した message-destination-name の指定 355

標準 DD で指定したロール名の指定 298

標準 DD で指定したロールを環境固有のユーザーまたはグループにマッピングするための要素 298

標準 DD で定義した resource-env-ref の res-ref-name の指定 308

標準 DD で定義した resource-ref の res-ref-name の指定 305

ふ

ファイナライズ滞留解消機能の有効または無効の設定 715

ファイル拡張子とコンテンツタイプの関係性を定義する設定ファイルの指定 217

ファイル拡張子に対する文字セットの指定 82

ファイル拡張子に対する文字セットのデフォルト値の指定 83

ファイルサイズによるログの分割 230

ファイルプール内のファイルの消去の有効または無効の指定 543

ファイルプールにクリーン (空) の状態で保持するファイルの割合の指定 544

物理的なメッセージ送信先の参照の定義 323

物理的なメッセージ送信先の名称の指定 290

部分フェッチサイズの指定 563

プライバシーが保護されたメッセージをターゲットがサポートするかどうかの指定 330

プリンシパルが属するグループの指定 393

ブローカーが保持するメッセージの最大合計サイズの指定 547

ブローカーが保持するメッセージの最大数の指定 547

ブローカーの起動時に活性化する接続サービスのリストの指定 546

ブローカーの構成に関するプロパティの設定 534

プロパティの名称と値の指定 364, 366, 369, 372, 374, 376, 378, 385, 394

プロパティを説明する任意のテキストの指定 425, 428, 431, 453, 458, 461, 463, 464

へ

べき等なリクエストの URL パターンの指定 422

ほ

ポートコンポーネントに自動生成される実装クラスの名前の指定 352

ポートコンポーネント用の tie 実装クラスの自動生成された名称の指定 420

ホスト名のルックアップの逆引きをするかどうかの指定 121

ボディー部分のフェッチ時にメッセージへの SEEN フラグ設定の防止 564

み

未配信メッセージの最大数の指定 537

未配信メッセージの最大の総メモリー量のバイト単位の指定 538

め

明示管理ヒープ機能適用除外クラス指定機能の有効または無効の設定 690

明示管理ヒープ機能適用除外設定ファイルに指定したクラスのうち適用除外設定を無効にしたいクラスを指定する設定ファイル（明示管理ヒープ機能適用除外無効設定ファイル） 756

明示管理ヒープ機能適用除外設定ファイルのファイルパスの指定 686

明示管理ヒープ機能適用除外無効設定ファイルのファイルパスの指定 688

明示管理ヒープ機能のイベントログのレベルの指定 701

明示管理ヒープ機能の自動解放機能の有効または無効の設定 695

明示管理ヒープ機能の自動解放処理で設定している解放率の上限値の指定 685

明示管理ヒープ機能の有効または無効の設定 703

明示管理ヒープ機能のログファイルの1ファイル当たりの最大ファイルサイズのバイト単位での指定 699

明示管理ヒープ機能のログファイルのプリフィクスおよびログファイルの出力先ディレクトリーの指定 698

明示管理ヒープ機能のログファイルを作成する最大ファイル数の指定 700

名称と値を持つプロパティーの指定 241, 245, 246, 424, 426, 429, 452, 454, 459, 463

メソッドの最後に実行した行の行番号のトレース情報への出力 720

メソッドのパラメーターの完全修飾 Java 型名の指定 347, 348, 415

メソッドの引数の完全修飾名の指定 317

メソッド名の指定 316, 347, 414, 443

メッセージセキュリティープロバイダーのための設定 242

メッセージセキュリティー要件が適用されるメソッドまたは操作の指定 285, 315, 346

メッセージセキュリティー要件の指定 284, 314, 346, 413

メッセージセキュリティー要件を適用するメソッドまたは操作の指定 414

メッセージ退避ファイルに追加するメッセージの最大サイズの指定 544

メッセージのセキュリティー要件の指定 442

メッセージのセキュリティー要件を適用するメソッドまたは操作の指定 443

メッセージの物理的な送信先の名称の指定 265, 450

メッセージの論理的な送信先の名称の指定 267, 464

メッセージをデッドメッセージキューに保存する前のメッセージボディの消去の有効または無効の指定 542

メニュー表示に対する文字セット 125

メニュー表示についての指定 137

メソッド引数の指定 316

メモリー制限のしきい値に達した場合のブローカー動作の指定 535

メモリー内の状態を物理的なストレージデバイスとの同期の有効または無効の指定 545

も

モジュールトレースの出力の抑止 132

モジュールの存在有無に対するディレクティブの定義 74

ゆ

有効期限機能に関連するレスポンスヘッダーを追加するかどうかの指定 112

有効期限のデフォルトの指定 114

有効なアドレスへのメール送信の指定 559

ユーザー拡張性能解析トレース設定ファイルで指定したクラスファイルの書き換えの成功ログの出力 725

ユーザー拡張性能解析トレース設定ファイルの記述形式の指定 717

ユーザー拡張性能解析トレース設定ファイルのファイルパスの指定 718

ユーザー拡張性能解析トレースで使用するユーザー拡張性能解析トレース設定ファイル 759

ユーザー拡張性能解析トレースの対象となるメソッド数の上限値の指定 719

ユーザー拡張性能解析トレースのトレース出力レベルの指定 721

ユーザー拡張性能解析トレースの有効または無効の設定 716

ユーザーが認証される、レルムの指定 344

ユーザーに利用させる機能の制限 156

ユーザー認証する場合の realm 名 97

ユーザー認証する場合の制御方法 95

ユーザー認証する場合の認証制御のタイプ 97

ユーザーの名称を指定 324

ら

ラベル名を指定しない LogFormat ディレクティブのフォーマットでのアクセスログ出力 216

り

リーパースレッドのリープまたはクリーンアップする周期の秒単位の指定 548

リクエスト受信開始からリクエストヘッダー受信完了までのタイムアウト時間の指定 180

リクエストのボディー部のログを採取する場合の最大値 186

リクエストヘッダーのカスタマイズ 178

リクエストボディーの受信開始から完了までのタイムアウト時間の指定 180

リクエストログに出力するトレース種別 130

リクエストログの出力 129

リクエストを受け付ける IP アドレスおよびポート番号 150

リソースアダプターアーカイブ構成情報の定義 377

リソースアダプターモジュール ID の指定 338

リソースの JNDI 完全修飾名の指定 274, 275, 278

リソースの設定をするための Java EE RI の DD 359

リソースの設定をするための拡張 DD 475

リソースマネージャー接続ファクトリー参照の名称の指定 433

リソースマネージャーへのサインオンに使用するデフォルトのリソースプリンシパル名の指定 257, 326

リソースマネージャーへのデフォルトのサインオン(名称とパスワード)の指定 326

リバースプロキシ使用時にバックエンドサーバから受信した Set-Cookie ヘッダーの変換 129

リバースプロキシをロードバランサー構成で使用する場合の転送先の URL の指定 99

流量制限機能に使用する Cookie が有効とされるドメイン 167

流量制限機能に使用する Cookie の有効時間 167

流量制限機能に使用する Cookie 名 169

流量制限機能によって拒否したリクエストのリダイレクト 172

流量制限機能によって処理が拒否された場合にレスポンスとして返すコンテンツ 174

る

ルート要素(hitachi-application.xml) 470

ルート要素(hitachi-application-client.xml) 472

ルート要素(hitachi-ejb-jar.xml) 474

ルート要素(hitachi-ra.xml) 476

ルート要素(hitachi-web.xml) 478

れ

例外またはエラーのクラス名が文字数制限 32 文字を超えた場合の名前の編集方法の指定 723

例外またはエラーのクラス名のトレース情報への出力 722

レルムの指定 332

ろ

ローカル変数オブジェクトに対して取得した String オブジェクトの文字列の出力 672

ローカル変数情報の出力フォーマットの 1 変数 1 行で出力する簡易出力フォーマットへの変更 680

ローカル変数情報への実際のオブジェクト型名の出力 682

ロードバランサー構成での振り分け先の定義 77

ログインモジュールの使用順序と使用条件の指定 502

ログに出力する時間をミリ秒まで表示するかどうかの指定 126

ログのフォーマットにラベル名を定義 152

ログファイル作成時に入出力エラーが発生した場合のログ情報の出力方法の指定 607

ログファイルの非同期出力機能の有効または無効の設定 609

論理的なメッセージ送信先の JNDI 名の指定 355

論理的なメッセージの送信先の名称を指定 355

論理メッセージ送信先の名称の指定 292

わ

ワークセキュリティーマップの定義 397

 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
