

インメモリデータグリッド

uCosminexus Elastic Application Data store  
ユーザーズガイド

解説・手引・文法・操作書

3020-3-V21-40

---

## 前書き

### ■ 対象製品

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

P-9W43-8A31 uCosminexus Elastic Application Data store 03-60

P-9W43-8B31 uCosminexus Elastic Application Data store Client for Java 03-60

P-9W43-8C31 uCosminexus Elastic Application Data store Client for C 03-60

●適用 OS : Windows Server 2008 x86, Windows Server 2008 x64, Windows Server 2008 R2, Windows Server 2012

P-2443-8B34 uCosminexus Elastic Application Data store Client for Java 03-60<sup>※</sup>

●適用 OS : Windows 7 x86, Windows 7 x64, Windows 8 x86, Windows 8 x64

P-2443-8B34 uCosminexus Elastic Application Data store Client for Java 03-60<sup>※</sup>

P-2443-8D34 uCosminexus Elastic Application Data store Developer 03-60<sup>※</sup>

注※ このプログラムプロダクトについては、出荷時期をご確認ください。

### ■ 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

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

BSAFE は、EMC 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。

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

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

Microsoft 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 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。

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。

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

RSA は、EMC 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 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。  
本製品は、米国 EMC コーポレーションの RSA BSAFE(R) ソフトウェアを搭載しています。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

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

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

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

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

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

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

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

This product includes software developed by the Java Apache Project for use in the Apache JServ servlet engine project (<http://java.apache.org/>).

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

This product includes software developed by Andy Clark.

**HITACHI**  
Inspire the Next

株式会社 日立製作所



本製品には、市販暗号プログラムとして「uCosminexus Primary Server Base」が含まれています。  
 当該製品の使用方法につきましては、以下のマニュアルを参照して下さい。

日立ソフトウェアマニュアル：オンラインマニュアル Cosminexus

[<http://www.hitachi.co.jp/Prod/comp/soft1/manual/common/cosmi/v9/index.html>]

その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

## ■ マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

表記		製品名
Windows	Windows 7 x86	Microsoft(R) Windows(R) 7 Professional 日本語版(32 ビット版)
		Microsoft(R) Windows(R) 7 Enterprise 日本語版(32 ビット版)
		Microsoft(R) Windows(R) 7 Ultimate 日本語版(32 ビット版)
	Windows 7 x64	Microsoft(R) Windows(R) 7 Professional 日本語版(64 ビット版)
		Microsoft(R) Windows(R) 7 Enterprise 日本語版(64 ビット版)
		Microsoft(R) Windows(R) 7 Ultimate 日本語版(64 ビット版)
	Windows 8 x86	Windows(R) 8 Pro 日本語版(32 ビット版)
		Windows(R) 8 Enterprise 日本語版(32 ビット版)
	Windows 8 x64	Windows(R) 8 Pro 日本語版(64 ビット版)
		Windows(R) 8 Enterprise 日本語版(64 ビット版)
	Windows Server 2008 x86	Microsoft(R) Windows Server(R) 2008 Standard 32-bit 日本語版
		Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit 日本語版
Windows Server 2008 x64	Microsoft(R) Windows Server(R) 2008 Standard 日本語版	
	Microsoft(R) Windows Server(R) 2008 Enterprise 日本語版	
Windows Server 2008 R2	Microsoft(R) Windows Server(R) 2008 R2 Standard 日本語版	
	Microsoft(R) Windows Server(R) 2008 R2 Enterprise 日本語版	
Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Standard 日本語版	
	Microsoft(R) Windows Server(R) 2012 Datacenter 日本語版	

## ■ 発行

2014年7月 3020-3-V21-40



■ 著作権

All Rights Reserved. Copyright (C) 2012, 2014, Hitachi, Ltd.

## 変更内容

### 変更内容 (3020-3-V21-40) uCosminexus Elastic Application Data store 03-60

追加・変更内容		変更箇所
キャッシュのデータをディスク上で管理できるようにしました。	<p>次の2種類のキャッシュを作成できるようにしました。</p> <ul style="list-style-type: none"> <li>• ディスクキャッシュ</li> <li>• 2Way キャッシュ</li> </ul>	1.3, 1.3.5, 2.3.1, 14.3.4(4)(c), 14.3.4(5), 14.3.9(1)
	<p>次のリソースを見積もる計算式および説明を変更しました。</p> <ul style="list-style-type: none"> <li>• Java ヒープサイズ</li> <li>• Explicit ヒープサイズ</li> <li>• 1EADs サーバプロセス当たりのファイルディスクリブタ数</li> </ul>	4.1.2, 4.1.3(1), 4.3.2(1), 9.1.1
	ディスク上にキャッシュを作成した場合に出力されるファイル (キャッシュファイル) の見積もりを追加しました。	4.2.1, 4.4
	EADs サーバおよび Developer のディレクトリ構成の説明を変更しました。	5.1.3(1), 16.2.2(1)
	定義ファイルにキャッシュ定義ファイルを追加しました。	5.2.2, 7.1, 7.7, 11.3.1(2), 11.3.2, 14.3.9(4)(a), 14.3.9(6), 14.3.10(4)(a), 14.3.14(4)(a), 14.3.15(4)(a), 14.3.16(4)(a), 14.3.18(4)(c), 23.1.1(5), 付録 B.5
	<p>ログファイルの種類に、キャッシュファイル操作ログファイルを追加しました。</p> <p>これに伴い、サーバ定義に次のパラメタを追加しました。</p> <ul style="list-style-type: none"> <li>• eads.logger.cache.rotationStyle</li> <li>• eads.logger.cache.filesize</li> <li>• eads.logger.cache.filenum</li> </ul>	7.4.1, 7.4.2, 7.4.5, 23.1.1(1), 付録 B.1
	統計情報に、キャッシュの統計情報を追加しました。	7.4.2, 11.10, 11.10.3
	<p>キャッシュデータファイルを最適化 (コンパクション) できるようにしました。これに伴い、eztool コマンドに compaction サブコマンドを追加しました。</p> <p>また、サーバ定義に次のパラメタを追加しました。</p> <ul style="list-style-type: none"> <li>• eads.statistics.compaction.effect.division</li> </ul> <p>また、コマンド定義に次のパラメタを追加しました。</p>	7.4.5, 9.4, 10.9, 12.2.2(2), 14.3, 14.3.1, 14.3.4(5), 14.3.24, 23.1.1(1), 23.1.1(4), 23.2

追加・変更内容	変更箇所
	<ul style="list-style-type: none"> <li>• eads.management.compaction.effect.threshold</li> <li>• eads.management.compaction.effect.division</li> </ul>
	サーバ定義に eads.cache.disk.getError.isolate.enable パラメタを追加しました。
	ディスク上のキャッシュを再開できるようにしました。これに伴い、eztool コマンドに resume サブコマンドを追加しました。 また、サーバ定義に次のパラメタを追加しました。 <ul style="list-style-type: none"> <li>• eads.management.resume.sendSize</li> <li>• eads.management.resume.sendInterval</li> </ul>
	サーバ定義の次のパラメタに、ディスク上のキャッシュの場合の説明を追加しました。 <ul style="list-style-type: none"> <li>• eads.restore.dataSender.sendSize</li> <li>• eads.restore.dataSender.sendInterval</li> </ul>
	ディスク上のデータを再配置できるようにしました。これに伴い、eztool コマンドに importecf サブコマンドを追加しました。
	コマンドの追加および仕様変更に伴い、コマンド定義の次のパラメタの説明を変更しました。 <ul style="list-style-type: none"> <li>• eads.management.long_running.read.timeout</li> <li>• eads.management.long_running.command.timeout</li> </ul>
	ディスク上にキャッシュを作成する手順を追加しました。
	ディスク上のキャッシュを終了する手順を追加しました。
	キャッシュファイルを削除できるようにしました。これに伴い、eztool コマンドに deleteecf サブコマンドを追加しました。
	永続データに関する情報の一覧を出力できるようにしました。これに伴い、eztool コマンドに listecf サブコマンドを追加しました。
	キャッシュと同時にキャッシュファイルを削除できるようにしました。これに伴い、eztool コマンドの

追加・変更内容	変更箇所
<p>deletecache サブコマンドに、--with_deleteecf オプションを追加しました。</p> <p>キャッシュのデータをディスク上で管理できるようになったことに伴い、ユーザ関数で利用できる次の API を追加しました。</p> <ul style="list-style-type: none"> <li>• ServerInfo インタフェースの次のメソッド <ul style="list-style-type: none"> <li>• getCacheType()</li> <li>• getCacheDataFileSpecifiedSize()</li> <li>• getCacheDataFileRemainingSize()</li> <li>• getCacheDataFileSpecifiedNumber()</li> <li>• getCacheDataFileUnusedNumber()</li> </ul> </li> <li>• 列挙型 CacheType</li> </ul> <p>また、ServerInfo インタフェースの使用例を追加しました。</p> <p>ユーザ関数で利用できる次の API に、value をメモリに格納しないキャッシュの説明を追加しました。</p> <ul style="list-style-type: none"> <li>• Group インタフェースの次のメソッド <ul style="list-style-type: none"> <li>• getValueUsageSize() (文字列指定)</li> <li>• getValueUsageSize() (Key インタフェース指定)</li> <li>• getValueUsageSize()</li> </ul> </li> </ul>	<p>20.2, 20.2.5(3), 20.2.5(7), 20.2.5(10), 20.2.5(11), 20.2.5(12), 20.2.5(13), 20.2.5(14), 20.2.13, 23.3.1(2)</p> <p>20.2.7(23)(d), 20.2.7(24)(d), 20.2.7(25)(c)</p>
<p>eztool コマンドの status サブコマンドで、次の内容が表示されるようにしました。</p> <ul style="list-style-type: none"> <li>• EADs サーバ ID</li> </ul>	<p>5.3.2(2), 10.2.1(2), 10.2.1(6), 10.3.1(2), 10.3.1(6), 10.4.1(2), 10.5.1(2), 10.7, 11.4.1(2), 11.4.1(6), 11.5.1(2), 11.5.1(6), 11.9.1(5), 14.3.4</p>
<p>サーバ定義の eads.statistics.enable パラメータを削除しました。これに伴い、サーバ定義の次のパラメータの説明を変更しました。</p> <ul style="list-style-type: none"> <li>• eads.statistics.interval</li> <li>• eads.statistics.backupnum</li> </ul> <p>また、次のパラメータを Developer で使用するサーバ定義に追加しました。</p> <ul style="list-style-type: none"> <li>• eads.statistics.interval</li> <li>• eads.statistics.backupnum</li> </ul>	<p>7.4.5, 23.1.1(1), 付録 B.1</p>
<p>起動設定の eads.java.external.heapsize パラメータの最小値、および見積もりの計算式を変更しました。これに伴い、サーバ定義の eads.consensus.external.heapsize パラメータの説明を変更しました。</p>	<p>9.1.3</p>
<p>eztool コマンドの listcache サブコマンドで、次の項目が表示されるようにしました。</p> <ul style="list-style-type: none"> <li>• CacheType</li> </ul>	<p>10.2.1(4), 10.3.1(4), 10.8, 11.4.1(4), 14.3.5</p>



追加・変更内容	変更箇所
<ul style="list-style-type: none"> <li>TotalMasterKeyCount</li> </ul>	
<p>eztool コマンドの listcache サブコマンドの -v または --verbose オプションを指定することで、コマンドの実行結果の詳細として、次の項目を表示できるようにしました。</p> <ul style="list-style-type: none"> <li>Server</li> <li>MasterKeyCount</li> </ul>	10.8, 14.3.5
<p>eztool コマンドの snapshot サブコマンドで収集するファイルを追加しました。</p>	14.3, 14.3.27, 16.6.1, 23.2
<p>eztool コマンドの export サブコマンドに -s または --single オプションを指定したときの説明を追加しました。</p>	14.3.11(4)(a)
<p>eztool コマンドの snapshot サブコマンドに -sd または --safedump オプションを指定した場合に、コマンドを実行できる条件の説明を変更しました。</p>	14.3.27(2)
<p>Developer 用 EADs クライアントの、サンプルプログラムの格納先のパスを変更しました。</p>	16.4.2
<p>Java 言語のクライアントライブラリが出力する次のエラーコードの説明を追加しました。 4000</p>	20.1.4(5)(c)
<p>ユーザファンクションから複数のキャッシュを操作できるようにしました。 これに伴い、ユーザファンクションで使用できる次の API を追加しました。</p> <ul style="list-style-type: none"> <li>FunctionContext インタフェースの getStore() (キャッシュ名指定)</li> <li>ServerInfo インタフェースの getCacheNames()</li> </ul>	20.2.2(3), 20.2.2(9), 20.2.5(3), 20.2.5(6)
<p>指定した key が格納済みかどうかを確認できるようにしました。 これに伴い、ユーザファンクションで使用できる次の API を追加しました。 Store インタフェースの containsKey()</p>	20.2.6(3), 20.2.6(7)
<p>ユーザファンクションが出力する次のエラーコードを追加しました。 1130 1140 また、次のエラーコードの説明を追加しました。 4000</p>	20.2.10(5)(c)
<p>C 言語のクライアントライブラリが出力する次のエラーコードの説明を追加しました。 4000</p>	22.2
<p>次の項目を EADs の最大値・最小値の一覧に追加しました。</p> <ul style="list-style-type: none"> <li>1 レンジ当たりのキャッシュデータファイルのサイズ</li> <li>1 レンジ当たりのキャッシュデータファイルのファイル数</li> <li>1 キャッシュデータファイルに格納できるデータ数</li> </ul> <p>次の項目を EADs の最大値・最小値の一覧から削除しました。</p> <ul style="list-style-type: none"> <li>1EADs サーバ当たりの、Explicit ヒープのサイズ</li> <li>ユーザファンクション数</li> <li>ユーザイベントハンドラ数</li> </ul>	23.5

追加・変更内容	変更箇所
<p>次の項目の最小値を 0 から 1 に変更しました。</p> <ul style="list-style-type: none"> <li>グループ数</li> <li>グループの階層数</li> </ul> <p>また、グループの階層数の最大値を「制限なし」から 511 に変更しました。</p>	
<p>次のメッセージを追加しました。</p> <p>KDEA01901-I~KDEA01903-I, KDEA07101-E, KDEA07104-E, KDEA07106-E, KDEA07107-E~KDEA07113-E, KDEA07115-E~KDEA07117-E, KDEA07120-E~KDEA07128-E, KDEA08059-I~KDEA08068-E, KDEA08070-E~KDEA08079-I, KDEA08414-W, KDEA08585-E~KDEA08589-E, KDEA08591-I~KDEA08598-I</p>	24.2, 24.6, 24.7
<p>次のメッセージを変更しました。</p> <p>KDEA00058-E, KDEA04692-I, KDEA04696-E, KDEA04805-E, KDEA04808-E, KDEA08026-W, KDEA08032-E, KDEA08530-E</p>	24.2, 24.5, 24.7

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

このマニュアルは、uCosminexus Elastic Application Data store の設計・構築・運用方法、および AP の開発方法について説明したものです。

### ■ 対象読者

uCosminexus Elastic Application Data store を導入したシステムを設計、構築、運用する方々、および AP の開発者の方々を対象にしています。

このマニュアルの記述は、次に示す知識があることを前提にしています。

- インメモリデータグリッドに関する基本的な知識
- Linux に関する基本的な知識
- Java 言語または C 言語を使用したプログラム開発に関する基本的な知識（AP の開発者の方）

### ■ このマニュアルでの表記

このマニュアルでは、製品名および Java 関連用語について次のように表記しています。

表記		製品名または Java 関連用語	
EADs	EADs サーバ	uCosminexus Elastic Application Data store	
	EADs クライアント	EADs クライアント (Java)	uCosminexus Elastic Application Data store Client for Java
	EADs クライアント (C)	uCosminexus Elastic Application Data store Client for C	
	Developer	uCosminexus Elastic Application Data store Developer	
JavaVM	Java Virtual Machine		

### ■ このマニュアルで使用する略語

このマニュアルで使用している英略語の一覧を次に示します。

英字での表記	英略語
Application Program	AP
Application Programming Interface	API

英字での表記	英略語
Key-Value store	KVS

## ■ このマニュアルで使用する記号

コマンドの説明で使用している記号を次に示します。

なお、この記号は説明のために使用している記号のため、コマンド中に記述しないでください。

記号	意味	例
[ ]	この記号で囲まれている項目は省略できます。	<pre>eztool status [-v --verbose]</pre> <p>このコマンドの例の場合、-v または--verbose を省略できることを意味しています。</p>

## ■ このマニュアルで使用する構文要素記号

構文要素記号	意味
<文字列>	任意の文字列を指定できます。半角文字を使用してください。

## ■ 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) はそれぞれ 1,024 バイト, 1,024<sup>2</sup> バイト, 1,024<sup>3</sup> バイト, 1,024<sup>4</sup> バイトです。

# 目次

前書き	2
変更内容	6
はじめに	11

## 第1編 解説

<b>1</b>	<b>uCosminexus Elastic Application Data store とは</b>	<b>26</b>
1.1	IT システムの特性に合わせて、大量データを効率的に処理するインメモリデータグリッド	27
1.2	EADs の概要	28
1.2.1	シンプルなデータ管理, シンプルなインタフェース	28
1.2.2	メモリ上でデータを管理	28
1.2.3	複数サーバで構成される大容量のメモリ領域	29
1.3	EADs の特長	30
1.3.1	高いスケーラビリティとスモールスタートへの対応	30
1.3.2	データの多重化によるシステムの高可用性と耐障害性の実現	30
1.3.3	高速データ処理によるレスポンス性能の向上	30
1.3.4	ユーザのニーズに応じた柔軟なデータ操作の実現	31
1.3.5	ディスク領域を使用したデータの永続化	31
1.4	EADs の適用ケース	32
1.4.1	ライトバッファとしての利用	32
1.4.2	リードキャッシュとしての利用	33
1.5	対象ユーザの作業項目とマニュアルの対応	34
<b>2</b>	<b>アーキテクチャ</b>	<b>35</b>
2.1	EADs の構成	36
2.1.1	システム構成	36
2.1.2	プロセス構成	38
2.2	EADs の通信処理の仕組み	39
2.2.1	通信に使用するプロトコル	39
2.2.2	通信で使用するバッファ	40
2.3	key と value を格納する領域	42
2.3.1	キャッシュの種類 (キャッシュタイプ)	42
2.4	データアクセス	44
2.4.1	データの格納 (put)	45
2.4.2	データの新規格納 (create)	45

2.4.3	データの更新 (update)	46
2.4.4	データの置換 (replace)	47
2.4.5	データの取得 (get)	49
2.4.6	データの削除 (remove)	50
2.4.7	EADs サーバの位置を意識しないデータアクセス	50
2.4.8	データアクセスの流れ	51
2.4.9	データアクセス時の排他制御	54
2.5	コンシステント・ハッシングによるデータの分散配置	56
2.5.1	データの分散配置の概要	56
2.5.2	データの分散配置の詳細	57
2.6	特定の EADs サーバへのデータ配置 (key のグルーピング)	58
2.6.1	key のグルーピング	58
2.6.2	key をグルーピングした場合のデータの分散配置	60
2.7	ユーザファンクションによる効率的なデータ処理	61
2.7.1	前提条件	61
2.7.2	ユーザファンクションの仕組み	61
2.8	データ操作を契機とするユーザ処理の実行 (ユーザイベントハンドラ)	64
2.8.1	ユーザイベントハンドラが動作するケース, 動作しないケース	64
2.8.2	ユーザイベントハンドラの仕組み	65
2.9	データの多重化	67
2.9.1	データの多重化の概要	67
2.9.2	データの多重化による, 障害時の処理の継続	67
2.10	クラスタ監視	69
2.10.1	ハートビートの送信によるクラスタ監視の概要	69
2.10.2	一定数の同意によって確定する EADs サーバダウン	70
2.11	EADs クライアントによるクラスタ構成情報の更新確認	73
2.12	クラスタおよび EADs サーバの状態遷移	74
2.12.1	クラスタの状態遷移	74
2.12.2	EADs サーバの状態遷移	75
2.13	スレッドプールおよびコネクションプールを使用したスループットの向上	78
2.13.1	スレッドプール	78
2.13.2	コネクションプール	79

## 第 2 編 設計・構築

### 3 システム設計・構築の流れ 80

3.1	システム設計・構築の流れ	81
3.1.1	必要なリソースを確認する	81
3.1.2	インストールおよびセットアップを行う	81

3.1.3	環境依存パラメタを設計する	81
3.1.4	チューニングパラメタを設計する	82
<b>4</b>	<b>必要なリソースを確認する</b>	<b>83</b>
4.1	メモリ使用量を見積もる	84
4.1.1	メモリ構成	84
4.1.2	Java ヒープサイズを見積もる	85
4.1.3	Explicit ヒープサイズを見積もる	89
4.1.4	EADs サーバのメモリ使用量を見積もる	91
4.2	ディスク使用量を見積もる	93
4.2.1	EADs サーバのディスク使用量を見積もる	93
4.2.2	EADs クライアントのディスク使用量を見積もる	97
4.3	スレッド数とファイルディスクリプタ数を見積もる	99
4.3.1	スレッド数を見積もる	99
4.3.2	ファイルディスクリプタ数を見積もる	99
4.4	キャッシュファイルの容量を見積もる	102
4.4.1	キャッシュデータファイルのサイズおよび面数を見積もる	102
4.4.2	キャッシュインデックスファイルのサイズを見積もる	107
<b>5</b>	<b>インストールおよびセットアップを行う (EADs サーバ)</b>	<b>108</b>
5.1	EADs サーバをインストールする	109
5.1.1	インストール前にすること	109
5.1.2	インストール手順	109
5.1.3	インストール後にすること	110
5.2	EADs サーバをセットアップする	113
5.2.1	運用ディレクトリを作成する	113
5.2.2	定義ファイルを編集する	114
5.2.3	AP を配置する	115
5.3	テストを行う	116
5.3.1	EADs サーバを開始する (キャッシュを新規に作成する)	116
5.3.2	コマンドを使用してテスト用のデータを操作する	116
5.3.3	EADs サーバを終了する	118
5.4	EADs サーバをアンセットアップする	119
5.5	EADs サーバをアンインストールする	120
<b>6</b>	<b>インストールおよびセットアップを行う (EADs クライアント)</b>	<b>121</b>
6.1	EADs クライアントをインストールする	122
6.1.1	インストール後にすること	122
6.2	EADs クライアントをセットアップする	125
6.2.1	AP を配置する	125

- 6.2.2 クライアント定義ファイルを編集する 125
- 6.3 EADs クライアントをアンインストールする 126
  
- 7 環境依存パラメタを設計する (EADs サーバ) 127**
  - 7.1 定義ファイルの種類 (EADs サーバ) 128
  - 7.2 定義ファイルの形式 129
  - 7.3 通信に依存するパラメタの設計 130
    - 7.3.1 IP アドレスまたはホスト名, およびポート番号の設定 130
    - 7.3.2 通信に依存するパラメタ 131
  - 7.4 ログファイルに依存するパラメタの設計 133
    - 7.4.1 ログファイルの種類 133
    - 7.4.2 ファイルの出力先の設定 133
    - 7.4.3 ファイルサイズおよびファイル数の設定 136
    - 7.4.4 統計情報ファイルのローテーションの設定 136
    - 7.4.5 ログファイルに依存するパラメタ 137
  - 7.5 クラスタ構成に依存するパラメタの設計 147
    - 7.5.1 EADs サーバの位置の設定 147
    - 7.5.2 データの多重度の設定 148
    - 7.5.3 クラスタ構成に依存するパラメタ 149
  - 7.6 バックアップファイルに依存するパラメタの設計 151
    - 7.6.1 ファイルの出力先の設定 151
    - 7.6.2 ストアデータファイルの世代数の設定 152
    - 7.6.3 バックアップファイルに依存するパラメタ 154
  - 7.7 キャッシュの運用に依存するパラメタの設計 156
    - 7.7.1 キャッシュタイプごとのパラメタの設定 156
    - 7.7.2 キャッシュファイルの種類と格納先の設定 156
    - 7.7.3 キャッシュファイルのサイズの設定 159
    - 7.7.4 キャッシュの運用に依存するパラメタ 159
  
- 8 環境依存パラメタを設計する (EADs クライアント) 166**
  - 8.1 定義ファイルの種類 (EADs クライアント) 167
  - 8.2 定義ファイルの形式 168
  - 8.3 通信に依存するパラメタの設計 169
    - 8.3.1 接続先 EADs サーバ, IP アドレスまたはホスト名, およびポート番号の設定 169
    - 8.3.2 通信に依存するパラメタ 169
  - 8.4 ログファイルに依存するパラメタの設計 172
    - 8.4.1 ログファイルの種類 172
    - 8.4.2 ファイルの出力先の設定 172
    - 8.4.3 ファイルサイズおよびファイル数の設定 173



8.4.4 ログファイルに依存するパラメタ 173

## 9 チューニングパラメタを設計する 176

9.1 メモリ・バッファに関連するパラメタの設計 177

9.1.1 メモリサイズの設定 177

9.1.2 バッファサイズの設定 178

9.1.3 メモリ・バッファに関連するパラメタ 179

9.2 スレッドプールおよびコネクションプールに関連するパラメタの設計 186

9.2.1 最大同時接続数の設定 186

9.2.2 最大同時実行スレッド数の設定 186

9.2.3 スレッドプールおよびコネクションプールに関連するパラメタ 187

9.3 タイムアウトに関連するパラメタの設計 191

9.3.1 通信を監視するタイマの設定 191

9.3.2 クラスタを監視するタイマの設定 195

9.3.3 タイムアウトに関連するパラメタ 205

9.4 コンパクションに関連するパラメタの設計 212

9.4.1 コンパクションの効果のしきい値の設定 212

9.4.2 コンパクションに関連するパラメタ 212

## 第3編 運用

## 10 通常運用 214

10.1 システム運用管理者が行うこと 215

10.2 EADs サーバを開始する（キャッシュを新規に作成する場合） 216

10.2.1 EADs サーバを開始する手順（メモリ上にキャッシュを作成する） 216

10.2.2 EADs サーバを開始する手順（ディスク上にキャッシュを作成する） 218

10.2.3 複数のキャッシュを使用する場合の注意点 222

10.3 EADs サーバを開始する（ファイルからデータを読み込んでキャッシュを作成する場合） 223

10.3.1 EADs サーバを開始する手順（メモリ上にキャッシュを作成する） 223

10.3.2 EADs サーバを開始する手順（ディスク上のキャッシュを再開する） 226

10.4 EADs サーバを終了する（メモリ上のデータを破棄する場合） 230

10.4.1 EADs サーバを終了する手順 230

10.5 EADs サーバを終了する（メモリ上のデータをファイルに書き出す場合） 232

10.5.1 EADs サーバを終了する手順 232

10.6 EADs サーバを終了する（ディスク上のキャッシュを終了する場合） 234

10.6.1 EADs サーバを終了する手順 234

10.7 クラスタおよび EADs サーバの状態を確認する 236

10.8 キャッシュの一覧を表示する 237

10.9 キャッシュデータファイルの使用量を削減する（キャッシュデータファイルをコンパクションする） 238

- 10.9.1 キャッシュデータファイルをコンパクションする手順 239
- 10.9.2 コンパクションを中断する手順 243
  
- 11 保守運用 244**
- 11.1 EADs サーバをクラスタに追加したい 245
  - 11.1.1 EADs サーバをクラスタに追加する手順 (メモリ上のキャッシュだけを使用する場合) 245
  - 11.1.2 EADs サーバをクラスタに追加する手順 (ディスク上のキャッシュを使用する場合) 246
- 11.2 EADs サーバをクラスタから削除したい 250
  - 11.2.1 EADs サーバをクラスタから削除する手順 (メモリ上のキャッシュだけを使用する場合) 250
  - 11.2.2 EADs サーバをクラスタから削除する手順 (ディスク上のキャッシュを使用する場合) 251
- 11.3 定義内容を変更したい 253
  - 11.3.1 定義内容を変更する手順 253
  - 11.3.2 定義内容を変更する場合の注意点 254
- 11.4 キャッシュを追加または削除したい 255
  - 11.4.1 メモリキャッシュを追加または削除する手順 255
  - 11.4.2 ディスクキャッシュおよび 2Way キャッシュを追加または削除する手順 257
- 11.5 バックアップを取得したい 263
  - 11.5.1 データのバックアップを取得する手順 263
- 11.6 ストアデータファイルを管理したい 266
  - 11.6.1 ストアデータファイルの確認および削除手順 266
- 11.7 データの格納先を確認したい 268
- 11.8 ユーザファンクションまたはユーザイベントハンドラが、各 EADs サーバに正しく配置されているかどうか、また、実行できるかどうか確認したい 269
  - 11.8.1 ユーザファンクションの実行可否を表示する 269
  - 11.8.2 ユーザイベントハンドラの実行可否を表示する 270
- 11.9 クラスタを稼働させたまま、EADs サーバの修正パッチを適用したい 272
  - 11.9.1 EADs サーバの修正パッチを適用する手順 272
- 11.10 統計情報を取得したい 274
  - 11.10.1 統計情報の格納先 274
  - 11.10.2 統計情報 (eads\_stats.csv) 274
  - 11.10.3 キャッシュの統計情報 (eads\_cache\_stats.csv) 279
  - 11.10.4 ユーザファンクションの統計情報 (eads\_function\_stats.csv) 282
  - 11.10.5 ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv) 285
- 11.11 キャッシュファイルを管理したい 288
  - 11.11.1 キャッシュファイルの確認および削除手順 288
  
- 12 障害運用 290**
- 12.1 障害に備えた運用 (障害監視と障害検知) 291
  - 12.1.1 メッセージの監視 291
  - 12.1.2 EADs サーバプロセスの監視 292

- 12.2 障害発生時のシステム運用管理者の作業 293
- 12.2.1 縮退状態が発生した場合 293
- 12.2.2 クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の場合 297
- 12.2.3 レスpons低下の報告があった場合 301
- 12.3 障害情報の取得方法 303
- 12.3.1 原因調査に必要な障害情報 303
- 12.3.2 統計情報の取得方法 304
- 12.3.3 スレッドダンプの取得方法 304
- 12.3.4 PRF トレースの取得方法 304

## 13 障害原因の調査 306

- 13.1 レスpons低下の原因調査 307
- 13.1.1 原因調査の流れ 307
- 13.1.2 原因の調査方法 308
- 13.2 PRF トレースの仕組みと出力される情報 315
- 13.2.1 PRF トレースの仕組み 315
- 13.2.2 PRF トレースに出力される情報 316

## 14 コマンドリファレンス 321

- 14.1 コマンドの格納先 322
- 14.2 EADs のコマンド 323
- 14.2.1 ezstart (EADs サーバの開始) 323
- 14.2.2 ezserver (EADs サーバのフォアグラウンドでの開始) 324
- 14.2.3 eztool (クラスタの運用操作) 326
- 14.3 eztool コマンドのサブコマンド 329
- 14.3.1 コマンド間の排他制御 330
- 14.3.2 close (クラスタの閉塞化) 332
- 14.3.3 open (クラスタの閉塞解除) 332
- 14.3.4 status (クラスタ状態の確認) 333
- 14.3.5 listcache (キャッシュの一覧表示) 340
- 14.3.6 listesd (ストアデータファイルの一覧表示) 343
- 14.3.7 getposition (データの格納先の表示) 347
- 14.3.8 unlock (排他解除) 350
- 14.3.9 createcache (キャッシュの作成) 351
- 14.3.10 deletetecache (キャッシュの削除) 352
- 14.3.11 export (データの書き出し) 353
- 14.3.12 import (データの読み込み) 356
- 14.3.13 deleteesd (ストアデータファイルの削除) 358
- 14.3.14 put (指定したデータの格納) 359

14.3.15	get (指定したデータの取得)	361
14.3.16	remove (指定したデータの削除)	363
14.3.17	listfunc (ユーザファンクションの実行可否の表示)	364
14.3.18	execfunc (ユーザファンクションの実行)	367
14.3.19	listevent (ユーザイベントハンドラの実行可否の表示)	369
14.3.20	listecf (永続データに関する情報の一覧表示)	373
14.3.21	resume (キャッシュの再開)	381
14.3.22	importecf (永続データの再配置)	383
14.3.23	deleteecf (キャッシュファイルの削除)	385
14.3.24	compaction (キャッシュデータファイルのコンパクション)	386
14.3.25	cprfed (PRF トレース情報の編集出力)	389
14.3.26	threaddump (スレッドダンプの出力)	392
14.3.27	snapshot (ログ, 設定情報, ハードウェア情報, およびネットワーク情報の収集)	393
14.3.28	stop (クラスタの終了)	396
14.3.29	forcestop (EADs サーバの強制終了)	397
14.3.30	isolate (EADs サーバの縮退)	398

## 第 4 編 AP 開発

### 15 AP 開発の流れ 400

15.1	AP 開発の流れ	401
15.1.1	開発環境を構築する	401
15.1.2	AP を作成する	402
15.1.3	AP をテストする	402
15.1.4	作成した AP を実行環境に移行する	402
15.2	AP を開発する前に知っておくべき前提条件	403
15.2.1	AP の作成言語と EADs クライアント	403
15.2.2	key, value, キャッシュ名, EADs クライアント名に指定できるデータ	404
15.2.3	予約パッケージ名および予約システムプロパティ名 (Java 言語の場合)	406

### 16 Developer を構築する 407

16.1	Developer に必要なリソースを確認する	408
16.2	Developer のインストールおよびセットアップを行う	409
16.2.1	インストール手順	409
16.2.2	インストール後にすること	409
16.3	Developer 用 EADs サーバをセットアップする	412
16.3.1	Developer 用 EADs サーバの運用ディレクトリを作成する	412
16.3.2	Developer 用 EADs サーバで使用する定義ファイルを確認する	412
16.4	Developer 用 EADs クライアントをセットアップする	413
16.4.1	Developer 用クライアント定義ファイルを編集する	413

- 16.4.2 サンプルプログラムを使用する 413
- 16.5 Developer をアンインストールする 414
- 16.6 Developer で実行できるコマンドおよびタスク 415
  - 16.6.1 EADs のコマンド 415
  - 16.6.2 Apache Ant のタスク 415
  - 16.6.3 Apache Ant のタスクの格納先 416
  - 16.6.4 Developer 用 EADs サーバを開始する 416
  - 16.6.5 Developer 用 EADs サーバの状態を確認する 418
  - 16.6.6 キャッシュを作成する 418
  - 16.6.7 キャッシュを削除する 418
  - 16.6.8 ソースプログラムをコンパイルする 419
  - 16.6.9 ユーザファンクションまたはユーザイベントハンドラをデプロイする 421
  - 16.6.10 Developer 用 EADs サーバを終了する 422

## **17 クライアント AP を作成する (Java 言語) 423**

- 17.1 ソースプログラムを作成する (Java 言語) 424
  - 17.1.1 キャッシュへのアクセスとデータ操作の流れ 424
- 17.2 クライアント AP 作成上の留意事項 (Java 言語) 428
  - 17.2.1 EADs クライアントの初期設定に関する留意事項 428
  - 17.2.2 キャッシュへのアクセス開始に関する留意事項 428
  - 17.2.3 データ操作に関する留意事項 428
  - 17.2.4 キャッシュへのアクセス終了に関する留意事項 429
  - 17.2.5 EADs クライアントの使用終了に関する留意事項 429
- 17.3 ソースプログラムをコンパイルする (Java 言語) 431

## **18 ユーザファンクションを作成する 432**

- 18.1 ユーザファンクションを作成する前に知っておくべき前提条件 433
  - 18.1.1 ユーザファンクションの作成言語 433
  - 18.1.2 ユーザファンクションの実行方法の種類 433
  - 18.1.3 EADs サーバが使用する Java クラスローダ 434
  - 18.1.4 ユーザファンクションの作成の流れ 435
- 18.2 ソースプログラムを作成する (ユーザファンクション) 436
  - 18.2.1 ユーザファンクションの流れ 436
- 18.3 ユーザファンクション作成上の留意事項 439
  - 18.3.1 jar ファイル名に関する留意事項 439
  - 18.3.2 パッケージ名に関する留意事項 439
  - 18.3.3 ユーザファンクションの実装に関する留意事項 439
  - 18.3.4 効率的に key の一覧を取得する方法 440
- 18.4 ファンクション定義ファイルを作成する (任意) 442

- 18.5 ソースプログラムをコンパイルする (ユーザファンクション) 444
- 18.5.1 ユーザファンクションをコンパイルする手順 444
- 18.6 ユーザファンクションをパッケージングする 445
- 18.6.1 ユーザファンクションをパッケージングする手順 445
- 18.7 ユーザファンクションをデプロイする 446
- 18.7.1 ユーザファンクションをデプロイする手順 446
- 18.8 ユーザファンクションを実行する 448
- 18.8.1 ユーザファンクションの呼び出し 448
- 18.8.2 ユーザログへの出力 448
- 18.8.3 ユーザファンクション運用上の留意事項 448
- 18.9 実行環境に配布する 450

## 19 ユーザイベントハンドラを作成する 451

- 19.1 ユーザイベントハンドラを作成する前に知っておくべき前提条件 452
- 19.1.1 ユーザイベントハンドラの作成言語 452
- 19.1.2 ユーザイベントハンドラが動作するケース, 動作しないケース 452
- 19.1.3 ユーザイベントハンドラが動作する条件の限定 453
- 19.1.4 ユーザイベントハンドラの優先度の設定 454
- 19.1.5 ユーザイベントハンドラの作成の流れ 455
- 19.2 ソースプログラムを作成する (ユーザイベントハンドラ) 456
- 19.2.1 ユーザイベントハンドラの流れ 456
- 19.3 ユーザイベントハンドラ作成上の留意事項 459
- 19.3.1 パッケージ名に関する留意事項 459
- 19.3.2 ユーザイベントハンドラの実装に関する留意事項 459
- 19.4 ソースプログラムをコンパイルする (ユーザイベントハンドラ) 460
- 19.4.1 ユーザイベントハンドラをコンパイルする手順 460
- 19.5 ユーザイベントハンドラをパッケージングする 461
- 19.5.1 ユーザイベントハンドラをパッケージングする手順 461
- 19.6 ユーザイベントハンドラをデプロイする 463
- 19.6.1 ユーザイベントハンドラをデプロイする手順 463
- 19.7 ユーザイベントハンドラを実行する 465
- 19.7.1 ユーザイベントハンドラの実行 465
- 19.7.2 ユーザログへの出力 465
- 19.7.3 ユーザイベントハンドラ運用上の留意事項 465
- 19.8 実行環境に配布する 466

## 20 API リファレンス (Java 言語) 467

- 20.1 Java 言語のクライアントライブラリが提供するクラス 468
- 20.1.1 Cache クラス 469

20.1.2	CacheManager クラス	479
20.1.3	Node クラス	491
20.1.4	CacheException クラス	493
20.1.5	InitializeException クラス	500
20.1.6	InternalClientException クラス	500
20.1.7	InternalServerException クラス	501
20.1.8	ServerCommunicationException クラス	501
20.1.9	UserOperationException クラス	502
20.2	ユーザ関クションで使用できる API	503
20.2.1	Function インタフェース	504
20.2.2	FunctionContext インタフェース	506
20.2.3	InitConfig インタフェース	511
20.2.4	ClientInfo インタフェース	513
20.2.5	ServerInfo インタフェース	514
20.2.6	Store インタフェース	522
20.2.7	Group インタフェース	525
20.2.8	Key インタフェース	540
20.2.9	UserLogger インタフェース	541
20.2.10	EADsStoreException クラス	544
20.2.11	InternalServerException クラス	546
20.2.12	UserOperationException クラス	547
20.2.13	列挙型 CacheType	547
20.3	ユーザイベントハンドラで使用できる API	549
20.3.1	EventHandler インタフェース	550
20.3.2	EventContext インタフェース	552
20.3.3	OperationFilter アノテーション	557
20.3.4	DataFilter アノテーション	558
20.3.5	Priority アノテーション	559
20.3.6	列挙型 OperationType	560
20.3.7	列挙型 MethodType	561
20.3.8	列挙型 PriorityLevel	561
<b>21</b>	<b>クライアント AP を作成する (C 言語)</b>	<b>563</b>
21.1	ソースプログラムを作成する (C 言語)	564
21.1.1	キャッシュへのアクセスとデータ操作の流れ	564
21.2	クライアント AP 作成上の留意事項 (C 言語)	569
21.2.1	EADs クライアントの初期設定に関する留意事項	569
21.2.2	キャッシュへのアクセス開始に関する留意事項	569
21.2.3	データ操作に関する留意事項	569

- 21.2.4 キャッシュへのアクセス終了に関する留意事項 570
- 21.2.5 EADs クライアントの終了に関する留意事項 570
- 21.3 ソースプログラムをコンパイルする (C 言語) 572

## 22 API リファレンス (C 言語) 573

- 22.1 C 言語のクライアントライブラリが提供する関数 574
  - 22.1.1 ead\_init\_client() (EADs クライアントの初期設定) 575
  - 22.1.2 ead\_init\_client\_n() (EADs クライアントの初期設定) 576
  - 22.1.3 ead\_start\_cache() (キャッシュへのアクセス開始) 577
  - 22.1.4 ead\_stop\_cache() (キャッシュへのアクセス終了) 578
  - 22.1.5 ead\_terminate\_client() (EADs クライアントの終了) 579
  - 22.1.6 ead\_put() (key と value の格納) 580
  - 22.1.7 ead\_put\_array\_value() (複数の value の連結格納) 581
  - 22.1.8 ead\_create() (key と value の新規格納) 582
  - 22.1.9 ead\_update() (value の更新) 583
  - 22.1.10 ead\_replace() (value の置換) 584
  - 22.1.11 ead\_get() (value の取得) 585
  - 22.1.12 ead\_remove() (value の削除) 586
  - 22.1.13 ead\_execute\_function() (グループ指定によるユーザ関数の実行) 586
  - 22.1.14 ead\_execute\_function\_rt() (グループおよび受信タイムアウト指定によるユーザ関数の実行) 588
  - 22.1.15 ead\_execute\_node\_function() (EADs サーバ指定によるユーザ関数の実行) 590
  - 22.1.16 ead\_execute\_node\_function\_rt() (EADs サーバおよび受信タイムアウト指定によるユーザ関数の実行) 592
  - 22.1.17 ead\_get\_nodelist() (接続先 EADs サーバの情報取得) 594
  - 22.1.18 ead\_get\_node() (指定した key が格納されている接続先 EADs サーバの情報取得) 595
  - 22.1.19 ead\_get\_slave\_nodelist() (データの複製先 EADs サーバの情報取得) 596
  - 22.1.20 ead\_value\_element 構造体 (value 情報) 597
  - 22.1.21 ead\_object 構造体 (ユーザ関数で使用するオブジェクト) 598
  - 22.1.22 ead\_nodelist 構造体 (EADs サーバ情報) 598
  - 22.1.23 ead\_node 構造体 (EADs サーバ指定のユーザ関数で使用するオブジェクト) 599
  - 22.1.24 ead\_address 構造体 (EADs サーバのアドレス情報) 600
- 22.2 クライアントライブラリのエラーコード (C 言語) 601

## 第5編 よく使用する一覧集・メッセージ

### 23 よく使用する一覧集 606

- 23.1 パラメーター一覧 607
  - 23.1.1 EADs サーバで使用する定義ファイル 607
  - 23.1.2 EADs クライアントで使用する定義ファイル 614



23.2	コマンド一覧	616
23.3	API一覧	618
23.3.1	Java 言語のライブラリが提供する API	618
23.3.2	C 言語のクライアントライブラリが提供する関数	621
23.4	クライアントライブラリが出力するエラーコード	623
23.4.1	Java 言語のクライアントライブラリが出力するエラーコード	623
23.4.2	C 言語のクライアントライブラリが出力するエラーコード	623
23.5	最小値・最大値一覧	624

## 24 メッセージ 625

24.1	メッセージの記述形式	626
24.2	KDEA00001~KDEA01999	628
24.3	KDEA02000~KDEA02999	652
24.4	KDEA03000~KDEA03999	655
24.5	KDEA04000~KDEA05999	658
24.6	KDEA06000~KDEA07999	698
24.7	KDEA08000~KDEA09999	711
24.8	KDEA10000~KDEA11999	747

## 付録 750

付録 A	サンプルプログラム	751
付録 A.1	ExportCsvFunction (データの出力)	751
付録 A.2	RemoveAllFunction (データの削除)	753
付録 B	Developer で使用する定義ファイルで設定される値	755
付録 B.1	サーバ定義ファイル	755
付録 B.2	クラスタ定義ファイル	757
付録 B.3	起動設定ファイル	757
付録 B.4	コマンド定義ファイル	758
付録 B.5	キャッシュ定義ファイル	759
付録 C	用語解説	760

## 索引 762

## 1

## uCosminexus Elastic Application Data store とは

uCosminexus Elastic Application Data store は、インメモリデータグリッドを構築するための製品です。この章では、uCosminexus Elastic Application Data store の概要、および特長について説明します。

## 1.1 IT システムの特性に合わせて、大量データを効率的に処理するインメモリデータグリッド

---

uCosminexus Elastic Application Data store (以降、EADs と表記します) は、インメモリデータグリッドを構築する、データ処理基盤製品です。インメモリデータグリッドとは、複数サーバのメモリ上に大量データを分散配置し、そのデータを効率的に処理するデータ処理基盤のことです。

IT 技術の進歩、特にインターネット技術の進歩と、インターネット上に展開されるサービスの拡充によって、社会インフラは急激に変化しています。また、携帯電話やスマートフォンといった高性能のモバイル機器の普及が、社会インフラの変化をさらに加速させています。

このような社会インフラの変化に伴って、IT システムが取り扱うデータ量が爆発的に増加しています。また、「アクセスのピーク時にも、高速なレスポンスを実現したい」、「アクセス数やデータ量の変動に合わせてサーバを追加したい」というように、IT システムに対する要件も多様化しています。

そのため、これまでの RDBMS (リレーショナルデータベース管理システム) だけがデータ管理を担う IT システムでは、多様化する要件に応えることが難しくなっています。

そこで、インメモリデータグリッドなど、従来の RDBMS とは異なる思想に基づくデータ処理基盤ソフトウェアが注目されるようになりました。いわゆる NoSQL (Not Only SQL) と言われる、RDBMS を補完するソフトウェアです。

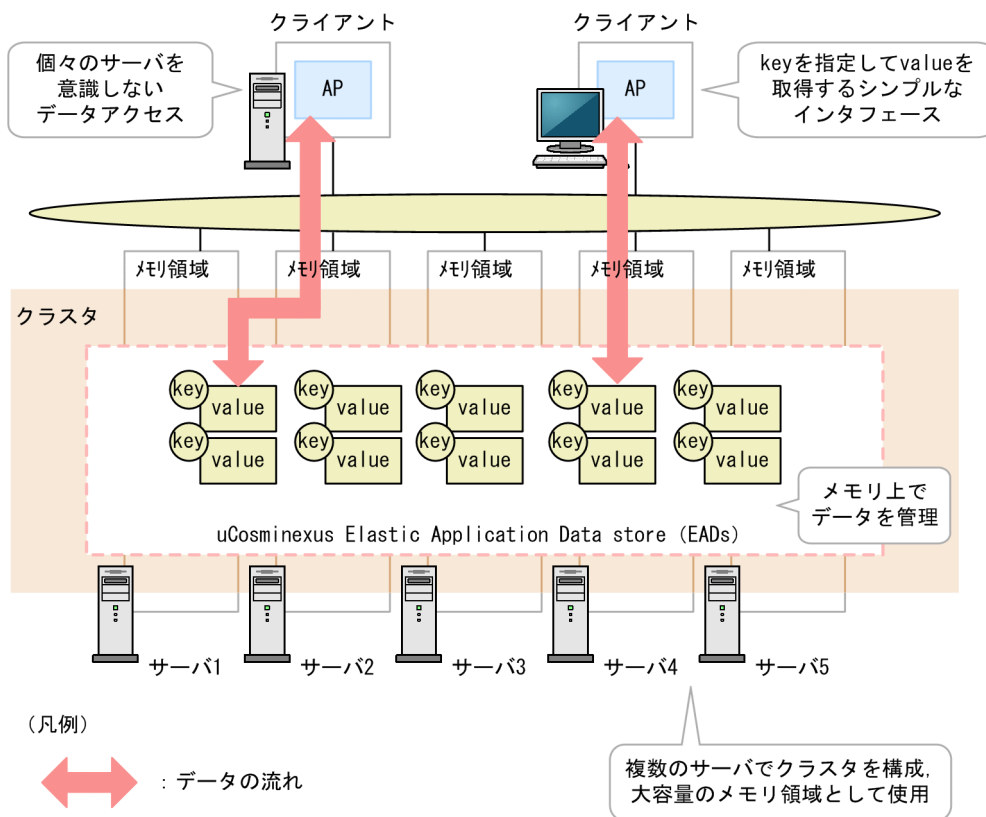
NoSQL の登場によって、RDBMS だけにすべてのデータ管理を任せるのではなく、RDBMS とそれを補完するソフトウェアを組み合わせることで、多様な要件に応えることができるようになってきました。

EADs は、高スケーラビリティと高信頼性を両立するインメモリデータグリッドを構築することで、RDBMS を補完するデータ処理基盤を提供します。

## 1.2 EADs の概要

EADs の概要について説明します。

図 1-1 EADs の概要



### 1.2.1 シンプルなデータ管理, シンプルなインタフェース

EADs では、インメモリデータグリッドの実装方式の 1 つとしてインメモリ型分散 KVS を採用しています。KVS では、データ (value) と、データを一意に識別するキー (key) というシンプルな形式でデータを管理します。key を指定して value を取得するというシンプルなインタフェースのため、AP の開発も容易です。

### 1.2.2 メモリ上でデータを管理

EADs では、メモリ領域でデータを管理します。これによって、ディスクアクセスによる処理のオーバーヘッドをなくしています。

### 1.2.3 複数サーバで構成される大容量のメモリ領域

EADs では、複数のサーバをグループとして管理します。このグループをクラスタと呼びます。EADs では、クラスタを構成する個々のサーバのメモリ領域を、1つのメモリ領域として使用します。これによって、個々のサーバのメモリ容量を超える大容量のデータを処理できます。

また、クライアントからは個々のサーバを意識しないでデータにアクセスできます。

## 1.3 EADs の特長

---

EADs には、次の特長があります。

- 高いスケーラビリティとスモールスタートへの対応
- データの多重化によるシステムの高可用性と耐障害性の実現
- 高速データ処理によるレスポンス性能の向上
- ユーザのニーズに応じた柔軟なデータ操作の実現
- ディスク領域を使用したデータの永続化

それぞれの特長について説明します。

### 1.3.1 高いスケーラビリティとスモールスタートへの対応

EADs では、サーバの追加や削除といったシステム構成の変更を容易に行うことができます。また、システム構成の変更時にデータを再配置することで、柔軟なスケールイン・スケールアウトが可能となります。これによって、高いスケーラビリティを実現します。

運用開始後のシステム構成の変更が容易なため、初期費用を抑えることができ（スモールスタート）、ビジネス規模の拡大に応じてシステムを拡張することができます。

### 1.3.2 データの多重化によるシステムの高可用性と耐障害性の実現

EADs では、複数のサーバに自動的にデータをコピーして、データを多重化できます。そのため、あるサーバに障害が発生しても、他サーバにコピーしたデータを使って処理を継続できます。

また、データを多重化することによって、メモリ上のデータが障害時に消失することを防ぐことができます。

### 1.3.3 高速データ処理によるレスポンス性能の向上

EADs は、処理するデータをすべてメモリ上に配置します。そのため、ディスクアクセスに掛かるオーバーヘッドをなくすことができ、高速にデータを処理できます。

また、key と value というシンプルなデータ構造のため、例えば SQL のような複雑な問い合わせはできませんが、データアクセスに掛かる時間は短縮されます。

これらによって、高速なレスポンス性能を実現しています。

### 1.3.4 ユーザのニーズに応じた柔軟なデータ操作の実現

EADs では、基本的なデータ操作のための API 以外にも、ユーザのニーズに応じた柔軟なデータ操作を実現するための API を提供しています。EADs が提供する API を利用することで、次のような運用ができます。

- データの集計や分析など、一連のデータ操作（ユーザ処理）を定義したプログラムを、サーバ上で実行できます。
- データの更新操作を契機に、ユーザ処理のプログラムを自動的に実行できます。

### 1.3.5 ディスク領域を使用したデータの永続化

EADs では基本的にメモリ領域でデータを管理しますが、データを永続化するために、メモリ領域とディスク領域を併用したり、ディスク領域だけでデータを管理したりすることもできます。

ただし、ディスク領域を使用する場合は、ディスクアクセスによる処理のオーバーヘッドが発生するため、メモリ領域でデータを管理する場合に比べると、レスポンス性能は低下します。

## 1.4 EADs の適用ケース

EADs を導入して、効果が期待できるケースと期待できないケースを次に示します。

### ■EADs を導入すると効果が期待できるケース

大量データの高速処理など、定型業務に向いています。

また、データの多重化によってシステムの高可用性と耐障害性を実現しているため、高い信頼性が要求される基幹系システムにも適用できます。

EADs をライトバッファとして利用する例を「1.4.1 ライトバッファとしての利用」に、リードキャッシュとして利用する例を「1.4.2 リードキャッシュとしての利用」に示します。

### ■EADs を導入しても効果が期待できないケース

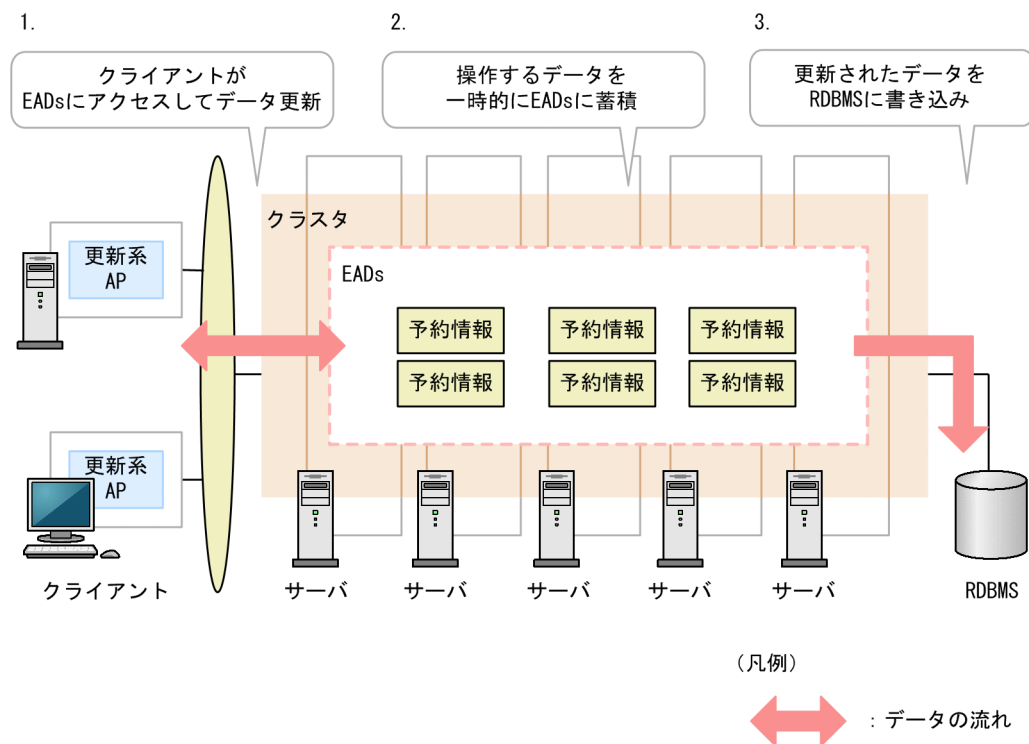
key と value というシンプルな形式でデータを管理するため、SQL のような複雑な問い合わせを必要とする非定型業務には向きません。

### 1.4.1 ライトバッファとしての利用

EADs は、業務システム内のライトバッファとして利用できます。ライトバッファとは、データ書き込み時のディスクアクセスに掛かるオーバーヘッドを回避するために、データを一時的に格納しておくメモリのことです。

予約システムやオンライントレードなど、多種多様な大量データを扱うシステムの更新系 AP で、EADs をライトバッファとして利用する例を次の図に示します。

図 1-2 ライトバッファとしての利用



1. uCosminexus Elastic Application Data store とは



1. クライアントから EADs にアクセスして、予約情報などのデータを更新します。
2. クライアントが操作するデータを EADs で一時的に蓄積します (ライトバッファ)。
3. 更新されたデータを RDBMS に書き込みます。

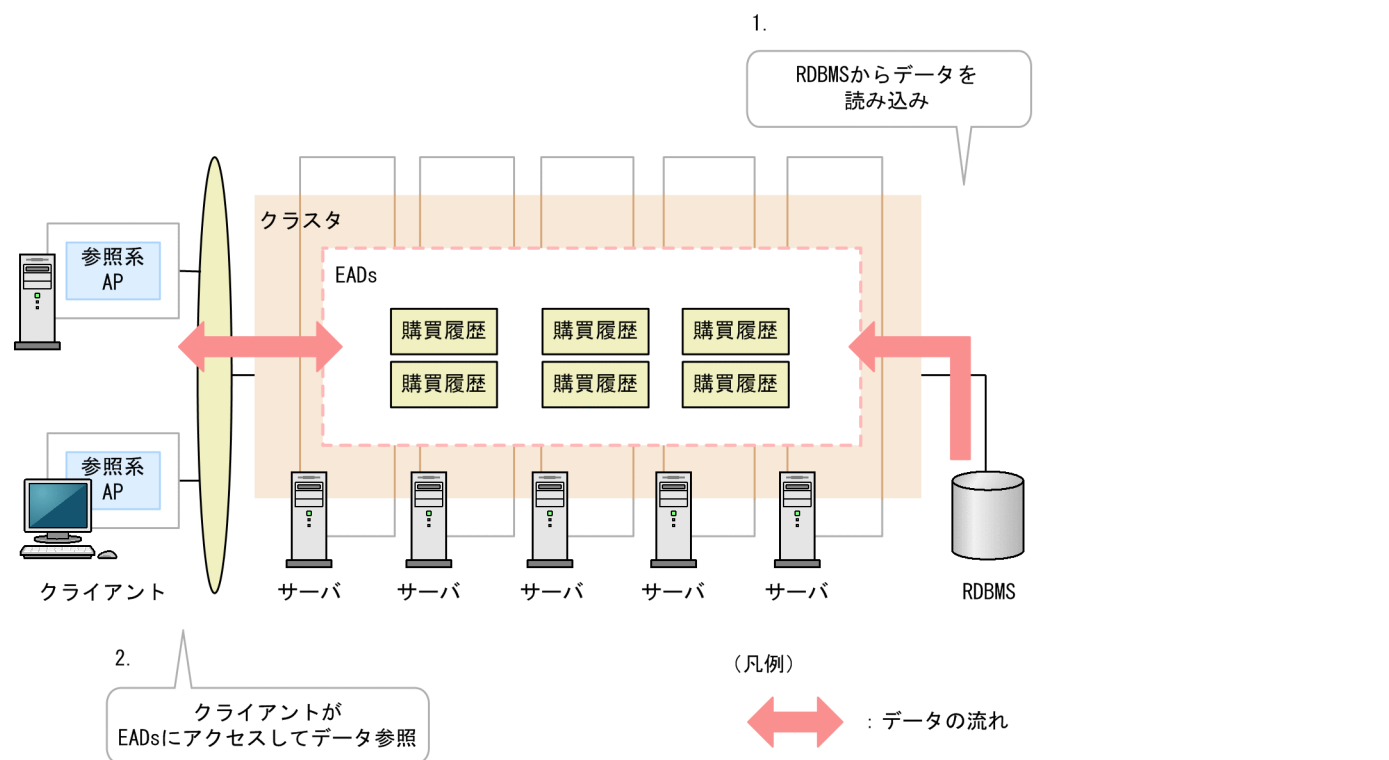
データベースにデータを直接書き込むのではなく、EADs をライトバッファとして利用することで、ディスクアクセスに掛かるオーバヘッドをなくすことができ、高速にデータを処理できます。

## 1.4.2 リードキャッシュとしての利用

EADs は、業務システム内のリードキャッシュとして利用できます。リードキャッシュとは、読み込みを高速化するために、データを一時的に格納しておくメモリのことです。

オンラインショッピングなど、会員情報や購買履歴を表示するような参照系 AP で、EADs をリードキャッシュとして利用する例を次の図に示します。

図 1-3 リードキャッシュとしての利用



1. 購買履歴など、必要なデータをあらかじめ RDBMS から EADs に読み込ませます (リードキャッシュ)。
2. クライアントは EADs にアクセスしてデータを参照します。

RDBMS から直接データを読み込むのではなく、EADs をリードキャッシュとして利用することで、ディスクアクセスに掛かるオーバヘッドをなくすことができ、高速にデータを処理できます。

## 1.5 対象ユーザの作業項目とマニュアルの対応

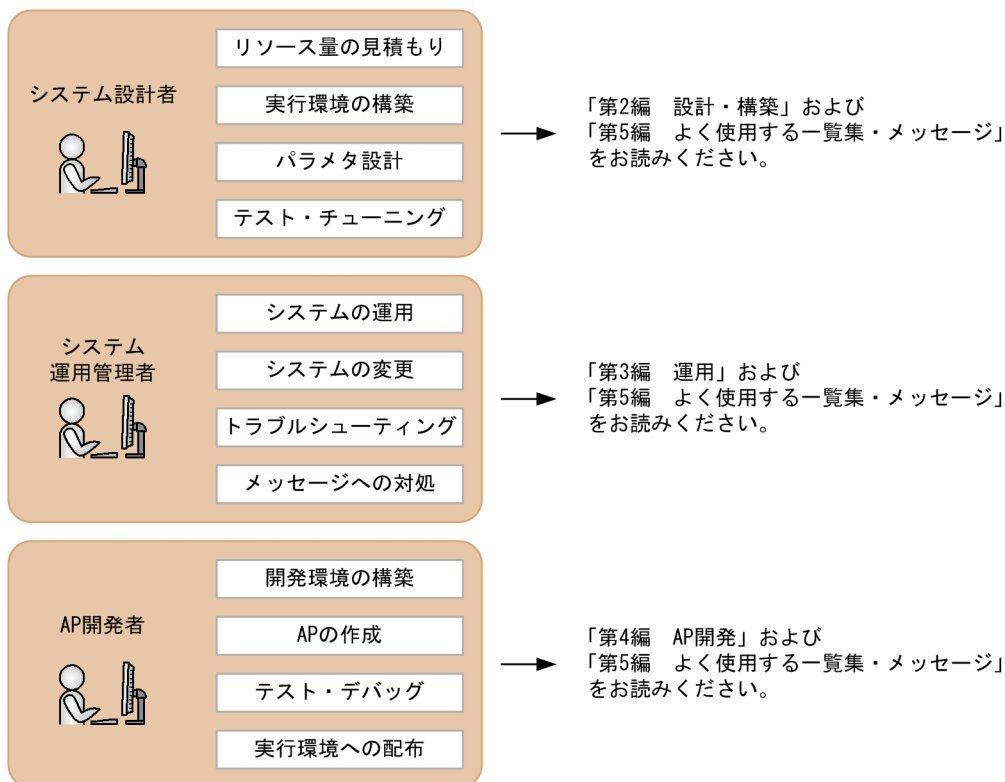
このマニュアルでは、次の対象読者を想定しています。

- システム設計者
- システム運用管理者
- AP 開発者

このマニュアルは次の 5 編から構成されます。

- 解説
- 設計・構築
- 運用
- AP 開発
- よく使用する一覧集・メッセージ

対象ユーザの作業項目とマニュアルの対応を次の図に示します。



# 2

## アーキテクチャ

この章では、EADs の構成、およびアーキテクチャについて説明します。

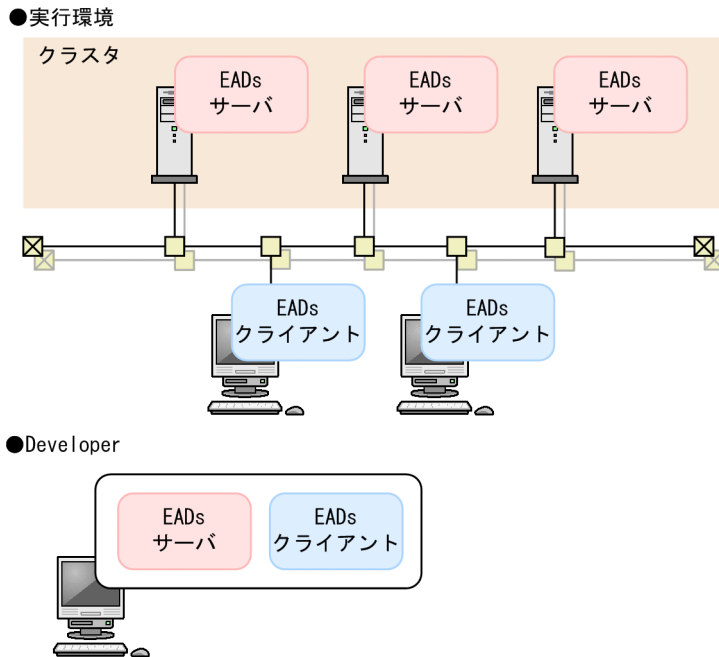
## 2.1 EADs の構成

EADs を導入したシステムの構成、およびプロセス構成について説明します。

### 2.1.1 システム構成

システム構成を次の図に示します。

図 2-1 システム構成



#### (1) 実行環境

分散 KVS にアクセスする AP や分散 KVS 自体を使用するための環境です。

信頼性の高いネットワークでの使用が前提となります。信頼性を高めるため、EADs サーバ間のネットワークは二重化することを推奨します。

##### (a) EADs サーバ

key と value で構成されるデータを管理するサーバプロセスです。

EADs サーバを構築するためのプログラムプロダクトを次に示します。

- uCosminexus Elastic Application Data store

## (b) クラスタ

通常、システムは複数の EADs サーバで構成されます。複数の EADs サーバの集合を、クラスタといいます。クラスタは、EADs クライアントからは 1 つの格納先として認識されます。

クラスタは同じセグメント内で同じマルチキャストアドレスおよびポート番号を設定した EADs サーバ群で構成されます。

### 注意事項

クラスタを構成する EADs サーバ数は、「データの多重度×2-1」以上であることが前提です。

データの多重度については、「[2.9 データの多重化](#)」を参照してください。

## (c) EADs クライアント

EADs が提供するクライアントライブラリを使用して、EADs サーバに接続するユーザプログラムです。

EADs クライアントには、AP を作成する言語 (Java 言語または C 言語) によって、次の 2 種類のプログラムプロダクトがあります。

- EADs クライアント (Java)  
uCosminexus Elastic Application Data store Client for Java
- EADs クライアント (C)  
uCosminexus Elastic Application Data store Client for C

### 注意事項

EADs クライアント・EADs サーバ間の通信で、IP アドレスまたはポート番号の変換を行うシステム構成では使用できません。

## (2) Developer

EADs では、Java 言語を使用した AP の開発をサポートするための製品として、Developer を提供しています。

Developer を構築するためのプログラムプロダクトを次に示します。

- uCosminexus Elastic Application Data store Developer

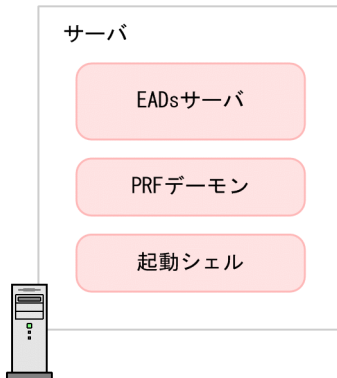
Developer の対象 OS は Windows だけです。

Developer は、1 マシン上に EADs サーバと EADs クライアントの機能を持ちます。1EADs サーバプロセスで構成されるため、クラスタを構成しません。

## 2.1.2 プロセス構成

EADs を導入したサーバは次の図に示すプロセスで構成されます。

図 2-2 プロセス構成



### (1) EADs サーバ

key と value で構成されるデータを管理するサーバプロセスです。

### (2) PRF デーモン

EADs サーバが共用メモリに出力した PRF トレース情報をファイルに出力するための I/O プロセスです。

PRF デーモンが出力する PRF トレース情報は、システムのボトルネックを検証したり、トラブルシューティングの効率向上を図ったりするために役立ちます。

PRF デーモンは、EADs サーバ 1 プロセス当たり 1 プロセス起動します。

なお、PRF トレース情報を出力しない場合、PRF デーモンは起動しません。

### (3) 起動シェル

EADs サーバと PRF デーモンを起動します。

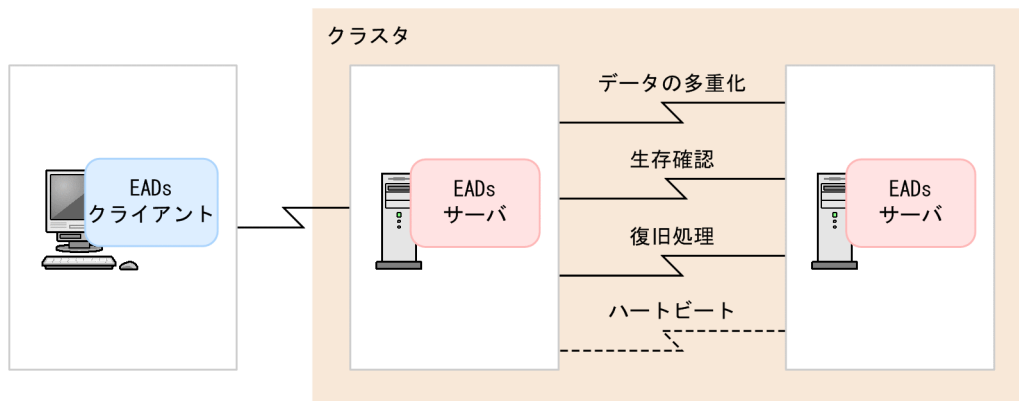
## 2.2 EADs の通信処理の仕組み

EADs の通信処理の仕組みについて説明します。

### 2.2.1 通信に使用するプロトコル

EADs クライアント・EADs サーバ間、および EADs サーバ間の通信に使用するプロトコルを次の図に示します。

図 2-3 通信に使用するプロトコル



(凡例)

— : TCPプロトコル

- - - : UDPプロトコル

#### (1) EADs クライアント・EADs サーバ間の通信

EADs クライアント・EADs サーバ間の通信には TCP プロトコルを使用します。

#### (2) EADs サーバ間の通信

EADs サーバ間の通信には、TCP プロトコルおよび UDP プロトコルを使用します。

次に示す EADs サーバ間の通信には、TCP プロトコルを使用します。

- データの多重化
- 生存確認※
- 復旧処理

また、次に示す EADs サーバ間の通信には、UDP プロトコルを使用します。

- ハートビート※

## 注※

クラスタ内では、ハートビートをマルチキャストで送信しています。

ハートビートとは、EADs サーバが正常に稼働していることを知らせるためにクラスタ内で配信するパケットのことです。

ハートビートを送信しない EADs サーバに対して、ダウンしていないかどうか、クラスタ内の他 EADs サーバが確認することを生存確認といいます。

詳細については、「[2.10 クラスタ監視](#)」を参照してください。

## 2.2.2 通信で使用するバッファ

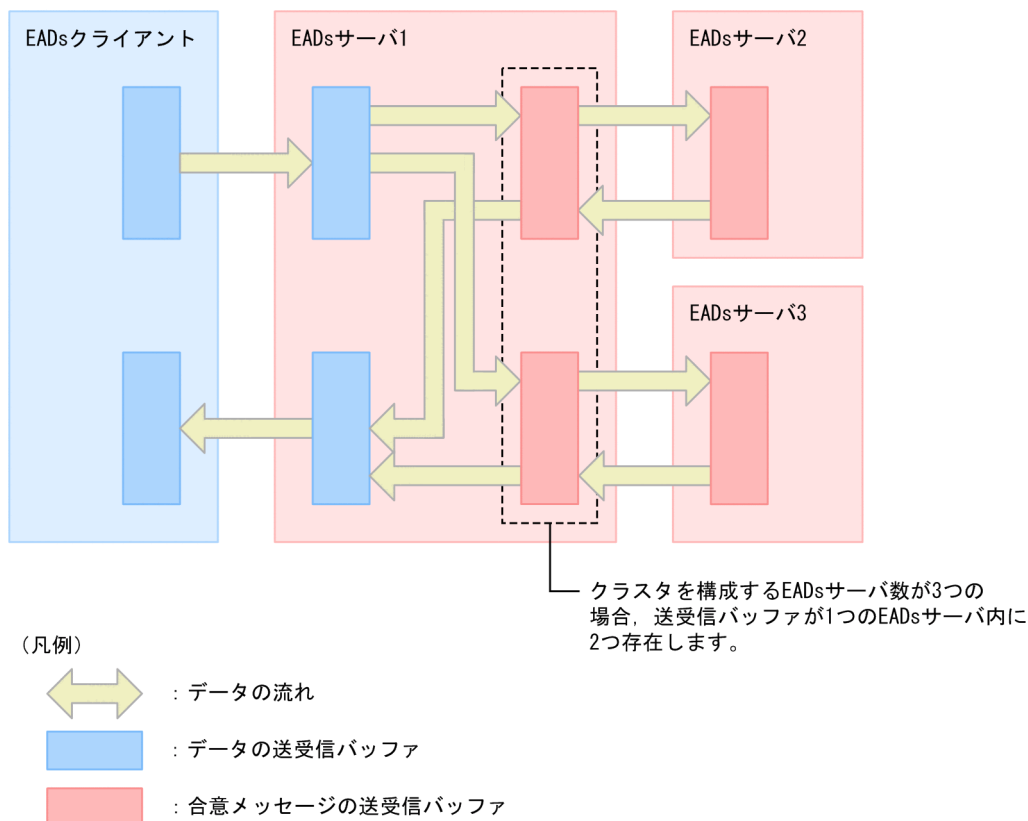
TCP プロトコルを使用する EADs クライアント・EADs サーバ間、および EADs サーバ間の通信では、次のバッファを使用します。

- データの送受信バッファ
- 合意メッセージの送受信バッファ

扱うデータ量に合わせて、それぞれのバッファサイズを調節することで、通信効率の向上を図ることができます。詳細については、「[9.1.2 バッファサイズの設定](#)」を参照してください。

通信で使用するバッファを次の図に示します。

図 2-4 通信で使用するバッファの概要





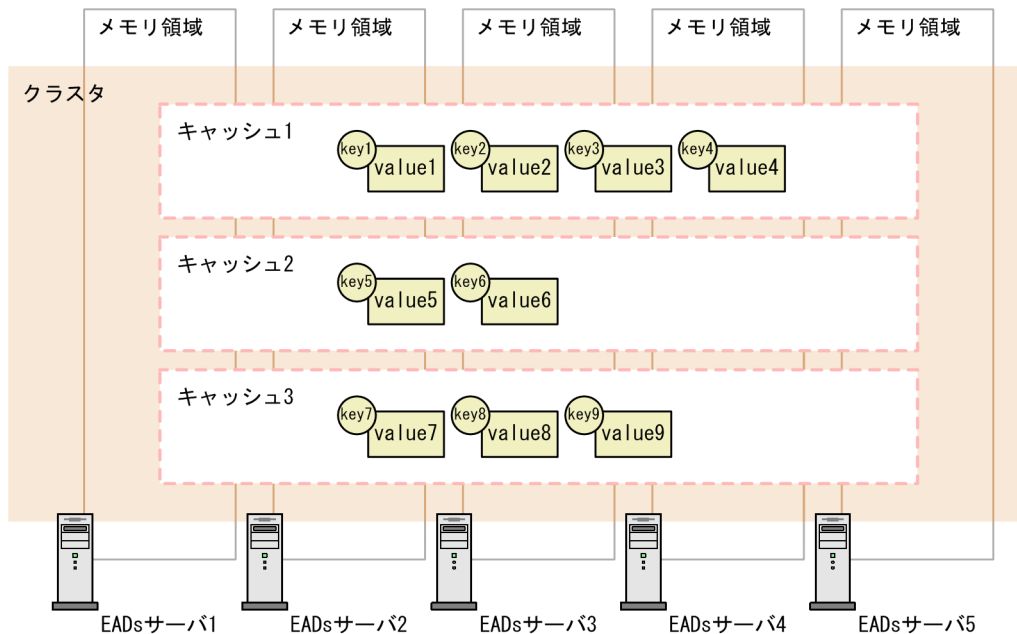
合意メッセージについては、「[2.4.8 データアクセスの流れ](#)」を参照してください。

## 2.3 key と value を格納する領域

key と value はキャッシュという領域に格納されます。

キャッシュは複数の EADs サーバにわたって作成される、論理的なデータ格納領域です。キャッシュを共有し、1つの論理的な KVS を構成する EADs サーバの集合をクラスタといいます。

図 2-5 キャッシュの概要



キャッシュの作成には、`eztool createcache` コマンドを使用します。キャッシュはクラスタ内に最大 64 個作成できます。

キャッシュの作成方法については、「[10.2 EADs サーバを開始する \(キャッシュを新規に作成する場合\)](#)」または「[11.4 キャッシュを追加または削除したい](#)」を参照してください。

### 2.3.1 キャッシュの種類 (キャッシュタイプ)

EADs で作成できるキャッシュの種類 (キャッシュタイプ) には、次の 3 種類があります。運用方法に合わせた種類のキャッシュを作成してください。

表 2-1 キャッシュの種類 (キャッシュタイプ) と特徴

項番	キャッシュの種類	データの格納先	特徴
1	メモリキャッシュ	メモリ領域	<ul style="list-style-type: none"><li>• 高速にデータを参照・更新できます。</li><li>• メモリ上でデータを管理するため、障害発生時にデータが消失するおそれがあります。</li></ul>
2	ディスクキャッシュ	ディスク領域	<ul style="list-style-type: none"><li>• ファイルを使用することで、データを永続化できます。</li></ul>

項番	キャッシュの種類	データの格納先	特徴
			<ul style="list-style-type: none"> <li>データの参照・更新では、ディスクアクセスによるオーバーヘッドが発生します。</li> <li>物理メモリ以上の大容量のデータを格納できます。</li> <li>データを多重化する場合、多重度以上の EADs サーバに障害が発生しても、データの復旧ができます。</li> </ul>
3	2Way キャッシュ	メモリ領域およびディスク領域	<ul style="list-style-type: none"> <li>高速にデータを参照できます。</li> <li>ファイルを使用することで、データを永続化できます。</li> <li>データの更新では、ディスクアクセスによるオーバーヘッドが発生します。</li> <li>データを多重化する場合、多重度以上の EADs サーバに障害が発生しても、データの復旧ができます。</li> </ul>

## 2.4 データアクセス

キャッシュのデータ操作には次の種類があります。

### データの更新操作

- put (データの格納)
- create (データの新規格納)
- update (データの更新)
- replace (データの置換)
- remove (データの削除)

### データの参照操作

- get (データの取得)

キャッシュのデータは API およびコマンドで操作できます。データ操作と API およびコマンドの対応を次の表に示します。

表 2-2 データ操作と API およびコマンドの対応

データ操作	API	コマンド
データの更新操作	<ul style="list-style-type: none"><li>• put</li><li>• create</li><li>• update</li><li>• replace</li><li>• remove</li></ul>	<ul style="list-style-type: none"><li>• <code>eztool put</code> コマンド</li><li>• <code>eztool remove</code> コマンド</li></ul>
データの参照操作	get	<code>eztool get</code> コマンド

### 注意事項

指定できるデータ型とサイズについて、API・コマンド間に完全な互換性はありません。

### データの更新操作の違い

データの更新操作を行う API (put, create, update, replace) および `eztool put` コマンドによって、実行できるデータ操作が異なります。API およびコマンドで実行できるデータ操作を次の表に示します。

API またはコマンド	新規のデータを格納する	既存のデータを更新する
<ul style="list-style-type: none"><li>• put</li><li>• <code>eztool put</code> コマンド</li></ul>	○	○
create	○	×
update	×	○

API またはコマンド	新規のデータを格納する	既存のデータを更新する
replace	×	○※

(凡例)

○：実行できます。

×：実行できません。

注※

指定した値と value が一致している場合だけ value を置換します。

## 2.4.1 データの格納 (put)

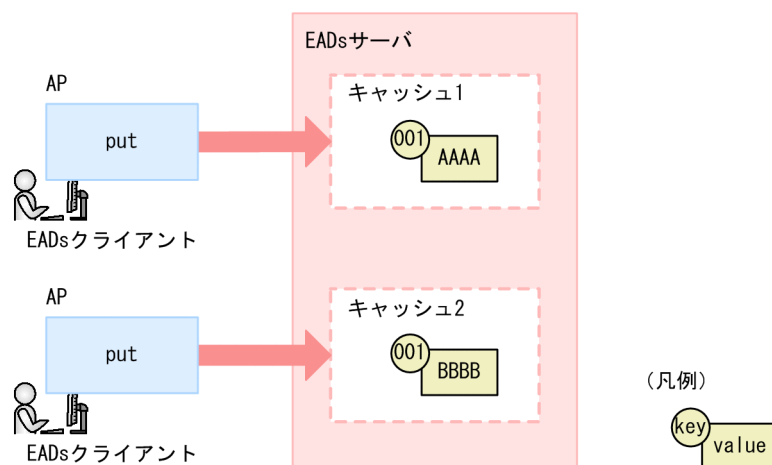
キャッシュにデータを格納します。

value に任意の key を関連づけ、key と value のペアで格納 (put) します。指定した key がすでにキャッシュに格納されている場合は、無条件に value を更新 (put) します。

key にはキャッシュ内でユニークな値を設定します。key が同じでも、格納先のキャッシュ名が異なれば、データを格納できます。

キャッシュ 1 およびキャッシュ 2 に、同じ key のデータを格納する例を次の図に示します。

図 2-6 データの格納 (put) の概要



## 2.4.2 データの新規格納 (create)

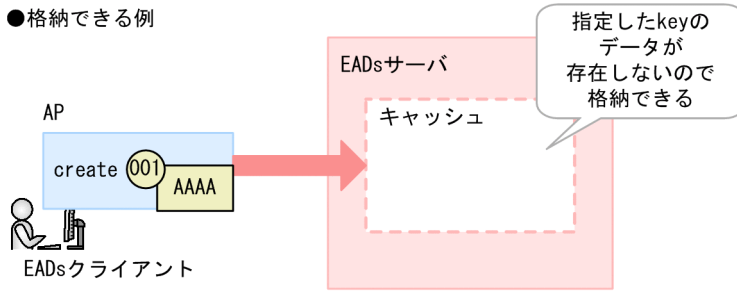
キャッシュに新規のデータを格納します。

新規に key を格納する場合だけ、value に任意の key を関連づけ、key と value のペアで格納 (create) します。

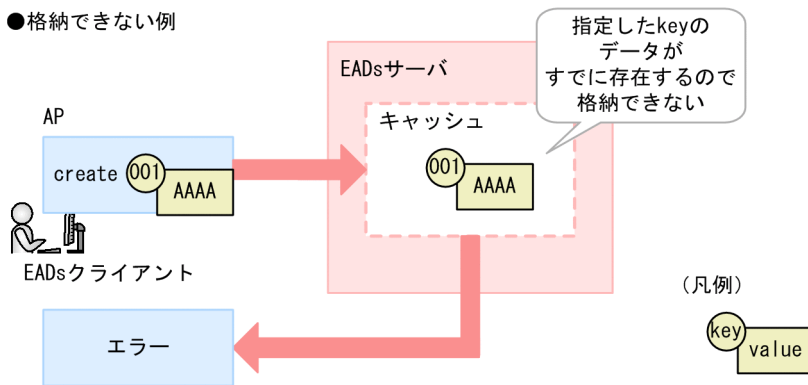
create は新規のデータの格納だけを行います。そのため、指定した key がすでにキャッシュに格納されている場合はエラーとなります。

## 図 2-7 データの新規格納 (create) の概要

### ●格納できる例



### ●格納できない例

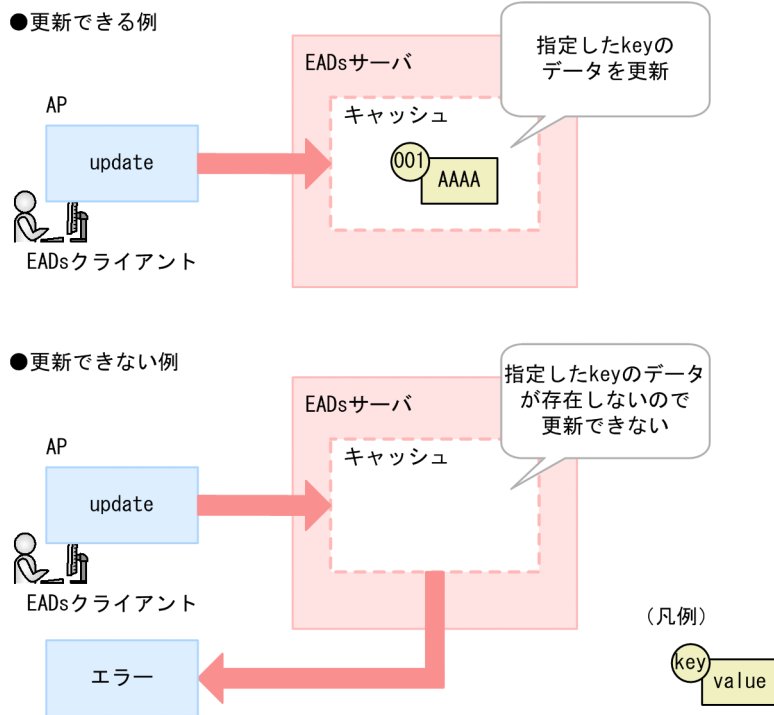


## 2.4.3 データの更新 (update)

すでにキャッシュに格納されているデータを更新します。

update は、指定した key がすでにキャッシュに格納されている場合だけ、value を key に関連づけて更新 (update) します。そのため、指定した key がキャッシュに格納されていない場合はエラーとなります。

図 2-8 データの更新 (update) の概要



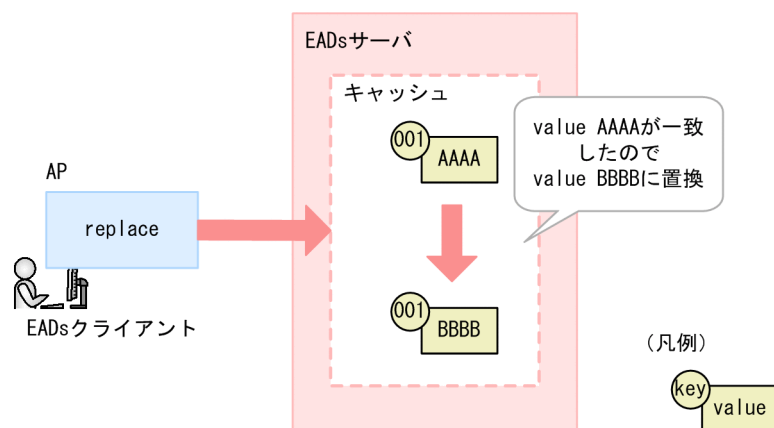
## 2.4.4 データの置換 (replace)

キャッシュのデータを置換します。

putが無条件にvalueを更新するのに対して、replaceは、指定した値とvalueが一致している場合だけvalueを置換します。

keyが001のvalueをBBBBに置換する例を次の図に示します。

図 2-9 データの置換 (replace) の概要



指定した値 (AAAA) とvalueが一致したため、valueをBBBBに置換します。

## 参考

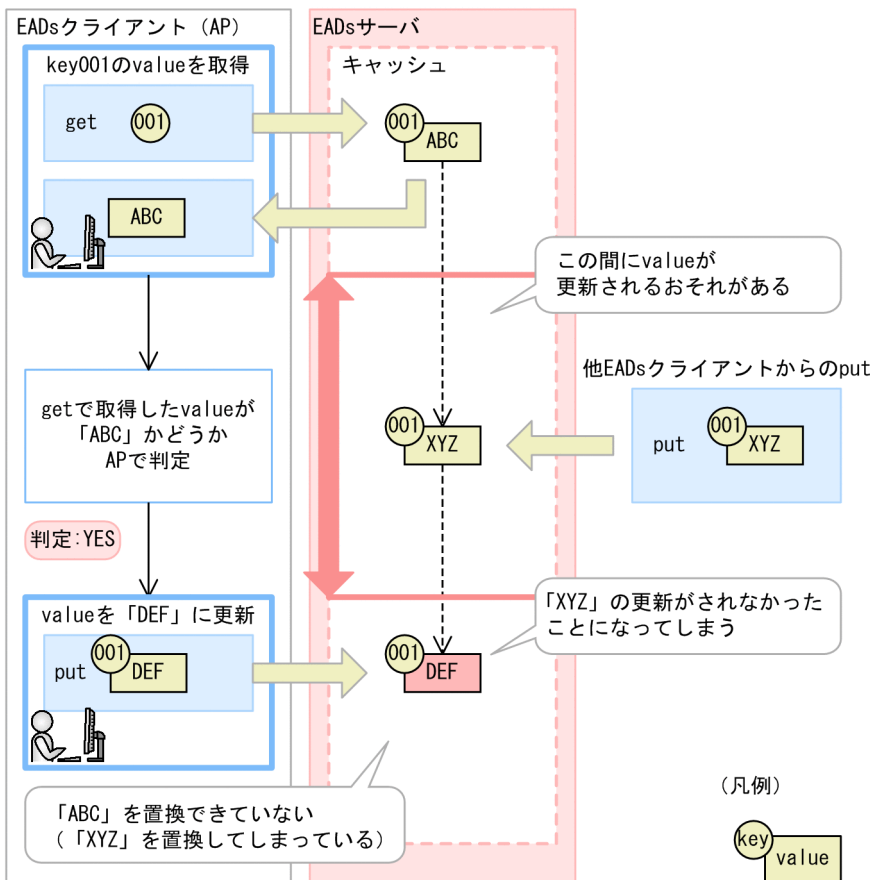
### put と replace の違い

例えば、格納されている value の内容を確認してから value を更新したい場合、get で value を取得し、AP で値を判定してから put を実行すると、get と put の処理の間に、ほかの AP によって value が更新されてしまうおそれがあります。replace を使用すれば、このようなケースを防ぐことができます。

「ABC」を「DEF」に置換する場合を例にして、put と replace の違いを次の図に示します。

### 図 2-10 put と replace の違い

#### ●getとputによる置換

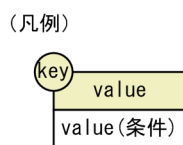
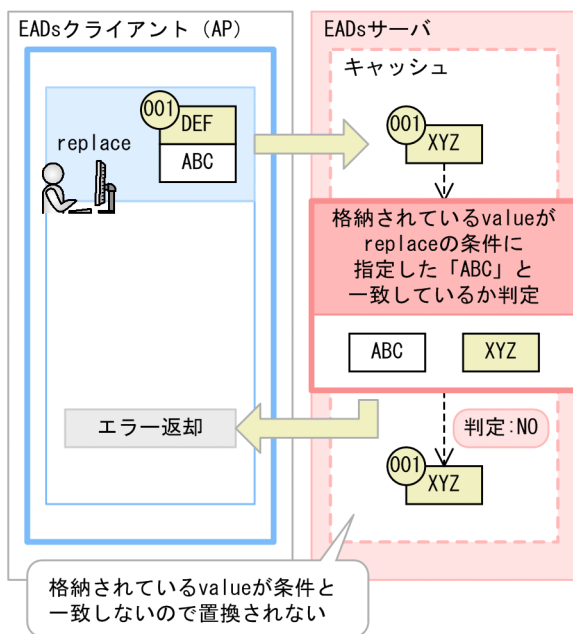
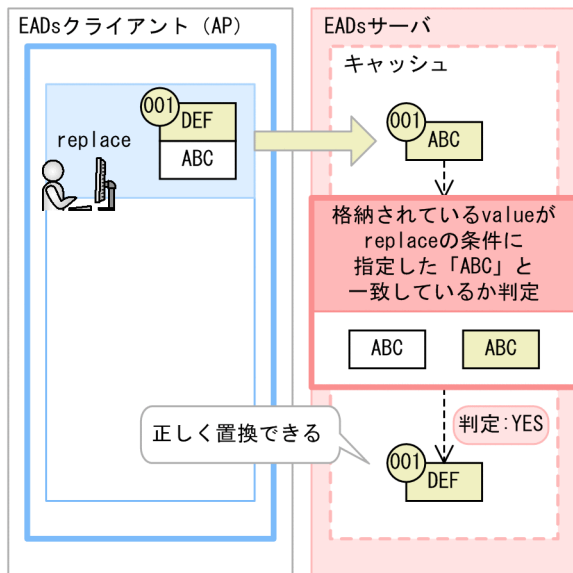


#### (説明)

getとputを使用して置換すると、この例のように途中で割り込みが入る可能性があるため、データの一貫性が確保できないおそれがあります。



●replaceによる置換



(説明)  
replaceを使用すると、データの一貫性を確保して、置換できます。

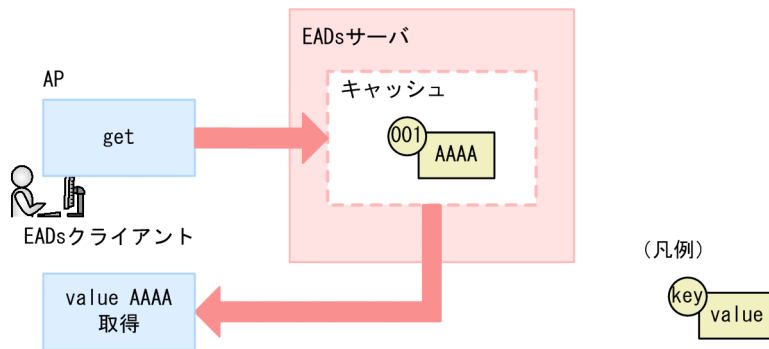
## 2.4.5 データの取得 (get)

キャッシュからデータを取得します。

取得したい value に関連づけられた key を指定して、value を取得 (get) します。

key が 001 の value を取得する例を次の図に示します。

図 2-11 データの取得 (get) の概要



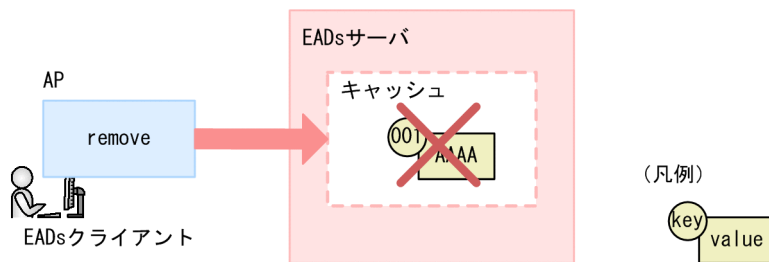
## 2.4.6 データの削除 (remove)

キャッシュからデータを削除します。

削除したい value に関連づけられた key を指定して、key と value を削除 (remove) します。

key が 001 のデータを削除する例を次の図に示します。

図 2-12 データの削除 (remove) の概要

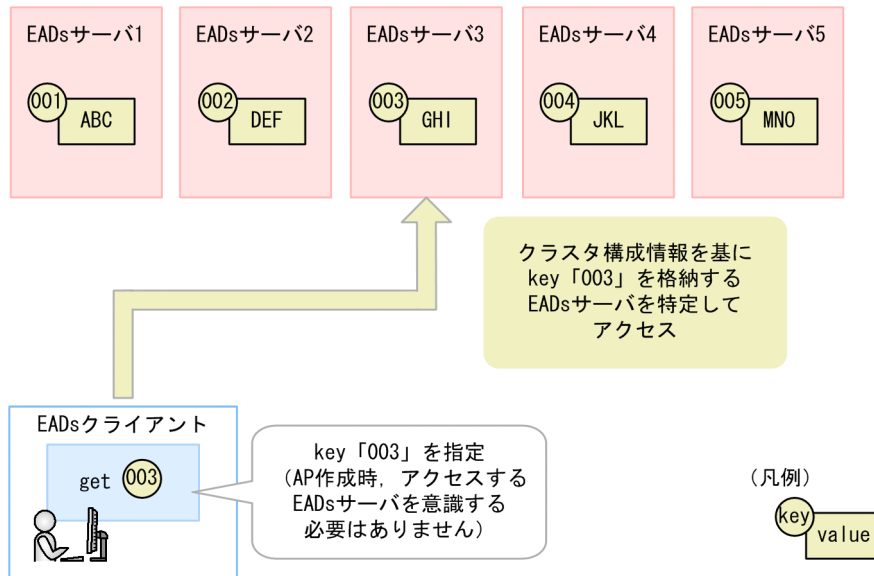


## 2.4.7 EADs サーバの位置を意識しないデータアクセス

EADs では、コンシステント・ハッシングというアルゴリズムを利用して、クラスタ内の EADs サーバにデータを分散配置します。

分散配置されているデータにアクセスする際、EADs クライアントは、EADs クライアントが保持している接続先の情報 (クラスタ構成情報) を基にデータの格納先 EADs サーバを特定します。このため、AP 作成時にデータが格納されている EADs サーバの物理位置を意識する必要はありません。

図 2-13 EADs サーバに対するアクセスの概要



なお、データが多重化されている場合は、データの格納先 EADs サーバにアクセスします。データの多重化については、「2.9 データの多重化」を参照してください。

クラスタ構成情報については、「2.10.1 ハートビートの送信によるクラスタ監視の概要」を参照してください。

データの分散配置の概要については、「2.5 コンシステント・ハッシングによるデータの分散配置」を参照してください。

## 2.4.8 データアクセスの流れ

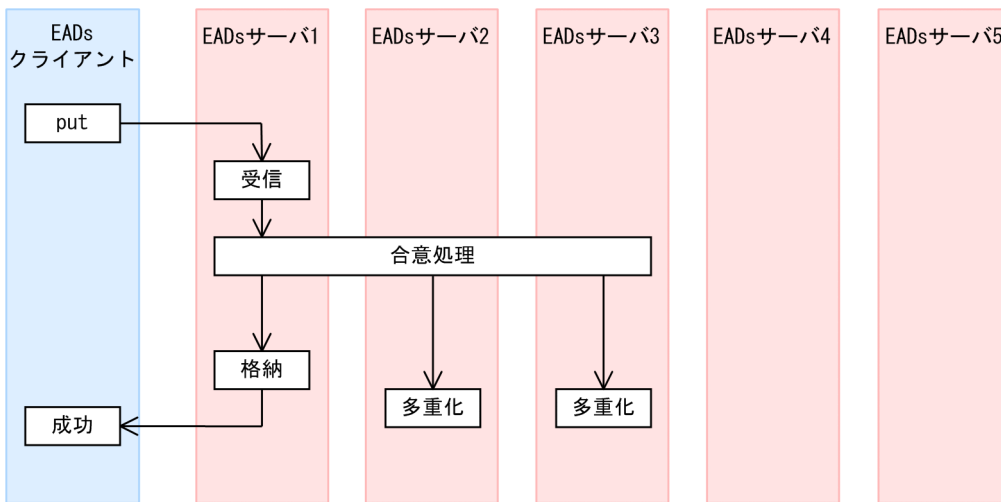
データアクセスの流れについて説明します。

### (1) データの更新操作

EADs クライアントは、EADs クライアントが保持しているクラスタ構成情報を基にデータの格納先 EADs サーバを特定し、データを更新します。

多重度を 3 に設定している場合の put 処理を例に、データアクセスの流れを次の図に示します。

図 2-14 データアクセスの流れ



EADs クライアントが put 処理のリクエストを送信します。

EADs サーバ 1 がリクエストを受信すると、データの複製先 EADs サーバに対して合意メッセージを送信し、put 処理を実行してよいか合意を取ります。

EADs サーバ 1 は、各 EADs サーバから合意メッセージを受信し、多重度と同数の合意が得られると処理を行います。これによって、データを多重化する際の、データの整合性を確保します。この例の場合、多重度が 3 なので、put 処理を実行するためには 3 つの EADs サーバの合意が必要です。

なお、一定時間内（デフォルトは 0.8 秒）に合意処理が完了しない場合はタイムアウトして、再度、合意処理を行います。

自 EADs サーバを含む 3 つの合意を得ると、データの格納および多重化を行います。この例の場合、多重度が 3 なので、EADs サーバ 2、および EADs サーバ 3 にデータがコピーされます。なお、各 EADs サーバ内の処理は非同期に行われます。

リクエストを受信した EADs サーバ 1 にデータが格納されると、EADs クライアントに処理結果が返されます。

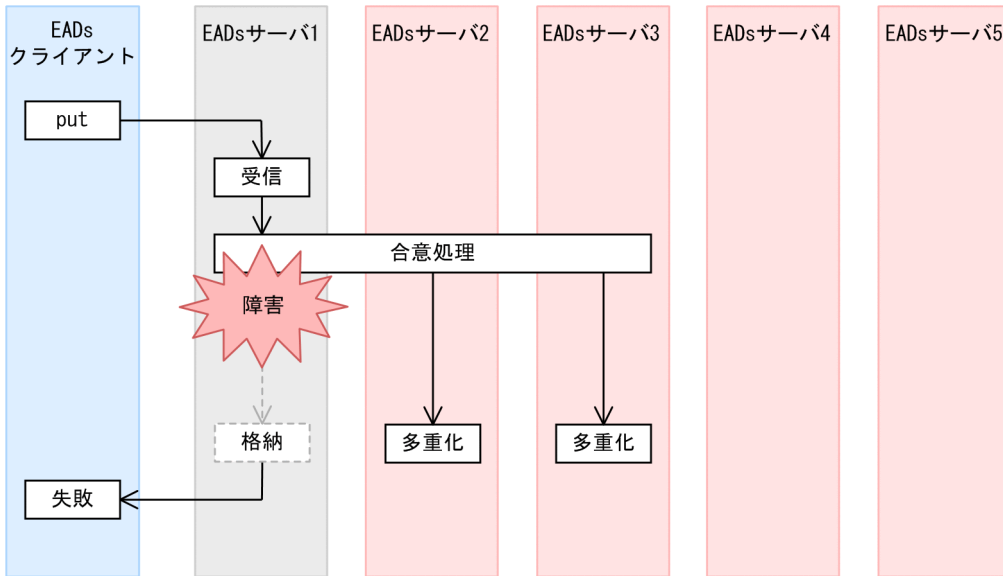
次に、データの格納や多重化に失敗した場合について説明します。

データの格納や多重化に失敗した場合は、次の要因が考えられます。

- 接続先 EADs サーバやネットワークに障害が発生している。
- key と value のうち、value 部分を格納する領域が不足している。
- 1EADs サーバに格納できるデータ数を超えている。

### (a) データの格納に失敗した場合

多重度を 3 に設定している場合の put 処理を例に、データの格納に失敗した場合の動作について説明します。



この例の場合、EADs サーバ 1 へのデータの格納に失敗し、EADs クライアントにエラーが返却されます。

データの多重化は実行されます。

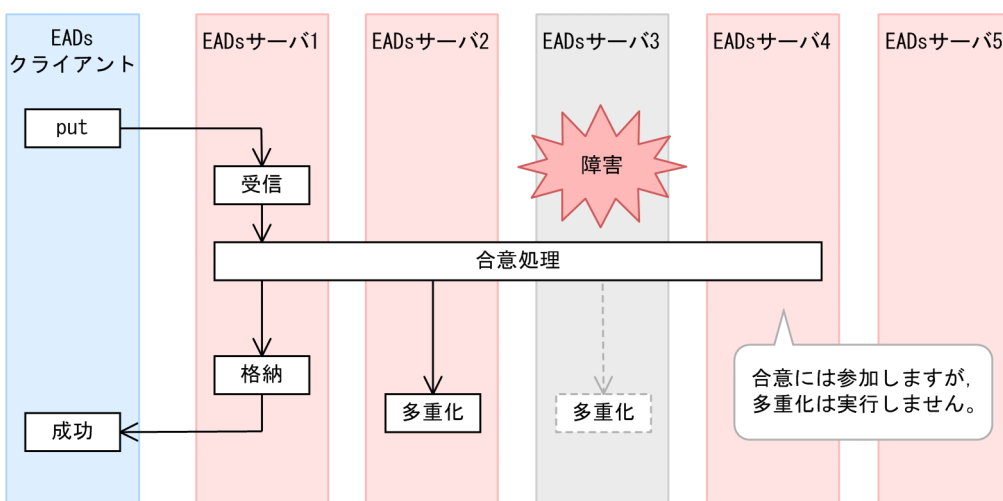
データの格納に失敗した EADs サーバは縮退状態※に遷移しますが、クラスタは継続します。この状態では、データの多重度が下がったままです。データの多重度を元に戻すには、縮退状態に遷移した EADs サーバの問題を取り除いてから、復旧を行います。

注※

縮退状態とは、EADs クライアントからのリクエストが受け付けられない状態です。

## (b) データの格納は成功したが、データの多重化に失敗した場合

多重度を 3 に設定している場合の put 処理を例に、データの格納は成功したが、データの多重化に失敗した場合の動作について説明します。



この例の場合、put 処理を実行するためには、3つの EADs サーバの合意が必要です。しかし、データのコピー先である EADs サーバ 3 からの応答がないため、別の EADs サーバ (EADs サーバ 4) の合意を得て、put 処理を実行します。ただし、EADs サーバ 4 は、データの多重化は実行しません。

データの多重化に失敗した EADs サーバは縮退状態に遷移しますが、クラスタは継続します。この状態では、データの多重度が下がったままです。データの多重度を元に戻すには、縮退状態に遷移した EADs サーバの問題を取り除いてから、復旧を行います。

## (2) データの参照操作

EADs クライアントは、EADs クライアントが保持しているクラスタ構成情報を基にデータの格納先 EADs サーバを特定し、データを参照します。

データの格納先 EADs サーバがダウンしている場合、データが多重化されていれば、データのコピー先 EADs サーバにアクセスしてデータを参照します。

### 2.4.9 データアクセス時の排他制御

複数の EADs クライアントやコマンドによって同一データが同時に操作され、データに矛盾が発生することを防ぐため、EADs サーバではデータの更新処理中に、レンジ単位に排他制御を行います。

レンジとは、あるキャッシュのデータを、EADs サーバの位置によって区切った、データの格納範囲のことです。詳細については、「[2.5.1 データの分散配置の概要](#)」を参照してください。

同一レンジのデータに対する複数アクセス時の同時実行可否を次の表に示します。

表 2-3 同一レンジのデータに対する複数アクセスの同時実行可否

実行中の処理	同時に実行する処理	
	データの更新操作※1	データの参照操作※2
データの更新操作※1	×	○
データの参照操作※2	○	○

(凡例)

- ：同時に実行できます。実行中の処理の完了を待たないで、処理を実行します。
- ×

注※1

データの更新操作とは、次の API またはコマンドのことです。

- API (put, create, update, replace, remove)
- `eztool put` コマンド
- `eztool remove` コマンド

API には、ユーザファンクション内で実行される API を含みます。

注※2

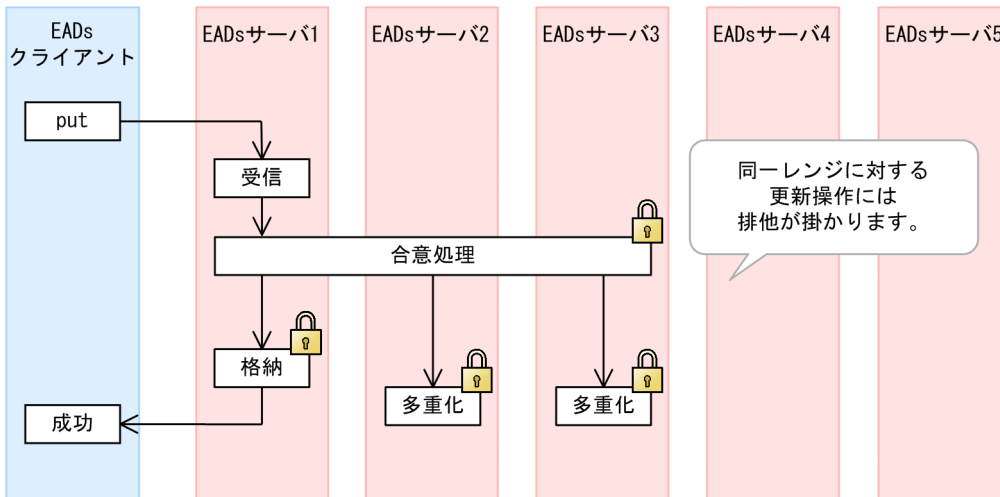
データの参照操作とは、次の API またはコマンドのことです。

- API (get)
- Group インタフェースの、次に示す Iterator 系メソッド
  - keyIterator()
  - descendingKeyIterator()
  - higherKeyIterator()
  - lowerDescendingKeyIterator()
- eztool get コマンド

API には、ユーザファンクション内で実行される API を含みます。

put 処理を例に排他範囲を次の図に示します。

図 2-15 排他範囲



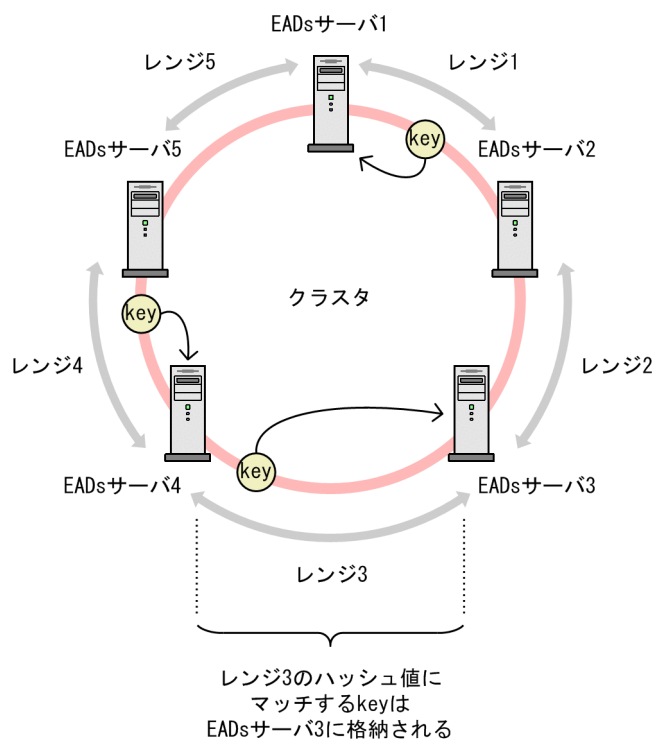
## 2.5 コンシステント・ハッシングによるデータの分散配置

EADs では、コンシステント・ハッシングというアルゴリズムを利用して、クラスタ内の EADs サーバにデータを分散配置します。

### 2.5.1 データの分散配置の概要

データの分散配置の概要について説明します。

図 2-16 コンシステント・ハッシングによるデータの分散配置



コンシステント・ハッシングでは、サーバや key が同一の円周上に配置されるものとして考えます。この円周上には、連番の整数が時計回りに昇順で割り振られています。

この図の場合、まず、EADs サーバ (1~5)、および格納するデータの key のハッシュ値を求めます。そのハッシュ値に従ってそれぞれを円周上に配置します。

各データの key は、配置された地点から反時計回りで最初に見つかる EADs サーバに保存されます。

また、あるキャッシュのコンシステント・ハッシングの円周上を、EADs サーバの位置によって区切ったハッシュ値の範囲をレンジといいます。レンジはレンジ ID で管理されます。この図の場合、レンジ 3 に配置された key は、EADs サーバ 3 に格納されます。

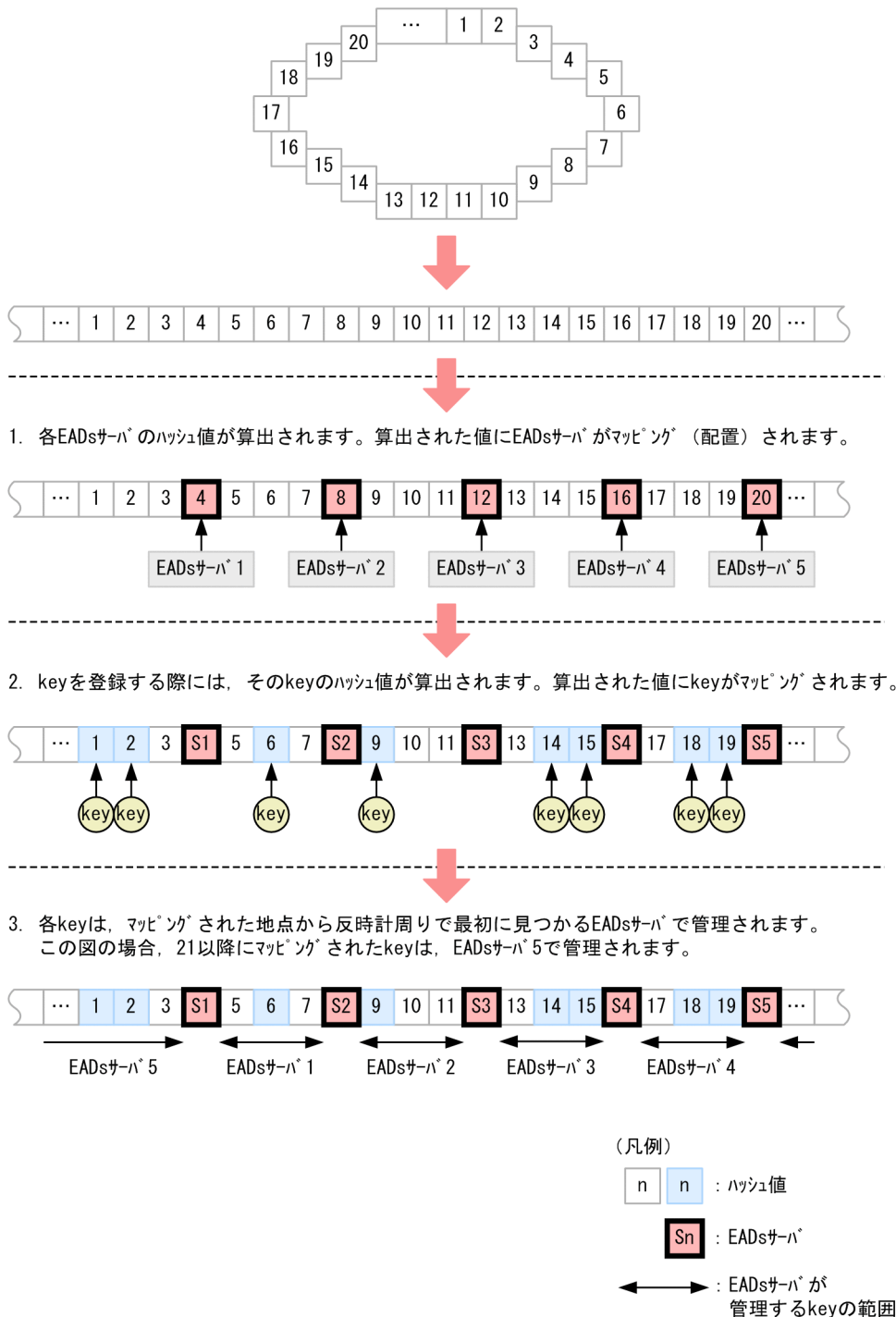
なお、この円は EADs サーバの物理配置とは関係ありません。



## 2.5.2 データの分散配置の詳細

データの分散配置の詳細について説明します。

コンシステント・ハッシングでは、サーバやkeyが円周上に配置されるものとして考えますが、便宜上、ここでは次の図のように線で表して説明します。



## 2.6 特定のEADsサーバへのデータ配置 (keyのグルーピング)

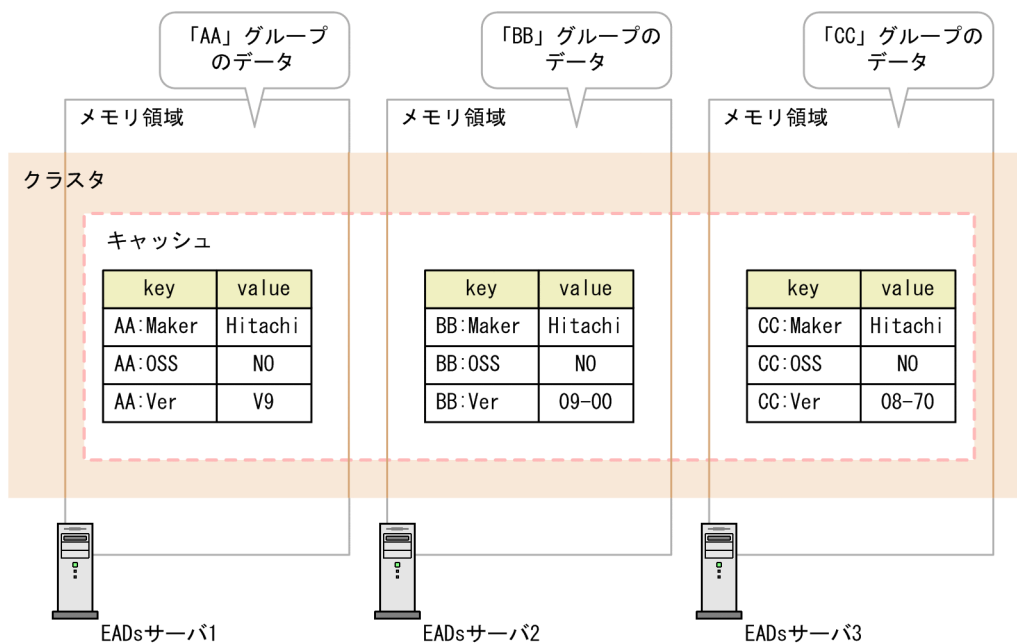
EADsでは、関連するデータを同じEADsサーバに配置するために、複数のkeyを1つのグループとしてまとめることができます。これをkeyのグルーピングといいます。

keyをグルーピングしない場合、データはクラスタ上のEADsサーバに分散配置されます。そのため、データの集計などでデータを操作する際には、クラスタ上に分散配置されたデータにアクセスする必要があります。このとき、各EADsサーバからデータを取得するために、毎回通信を行うことになります。

このように、処理対象のkeyが決まっているときは、そのkeyをグルーピングすることで、関連するデータを同じEADsサーバに配置できます。これによって、ユーザファンクションを利用した通信処理の削減ができるようになり、データ処理の効率化を図ることができます。

ユーザファンクションについては、「[2.7 ユーザファンクションによる効率的なデータ処理](#)」を参照してください。

図 2-17 特定のEADsサーバへのデータ配置 (keyのグルーピング)



### 2.6.1 keyのグルーピング

keyにグループ名を指定することで、keyをグルーピングできます。ここでは、グループ名と要素名、およびグループの階層化について説明します。

#### (1) グループ名と要素名

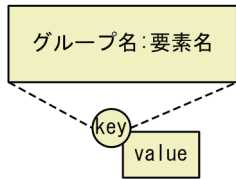
keyにグループ名を指定することで、keyをグルーピングできます。

keyは「グループ名:要素名」の形式で定義します。グループ名と要素名はコロン(:)で区切ります。

key をグルーピングする場合の、key の構成を次の図に示します。

## 図 2-18 key の構成

keyの構成



### グループ名

特定の key をまとめるグループの名称です。

### 要素名

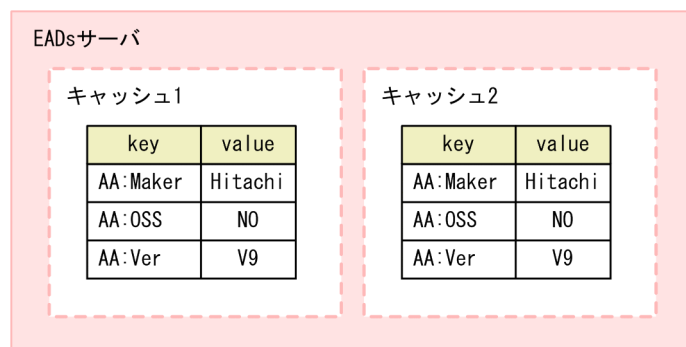
value を一意に識別するための名称です。

グルーピングしない場合、これが key になります。

## (2) グループ名と要素名の定義

key は、キャッシュ内ではユニークにする必要があります。そのため、グループ名と要素名の組み合わせが同じ key は同じキャッシュ内には定義できません。

次の図に示すように、グループ名と要素名の組み合わせが同じでも、格納先のキャッシュが異なれば、key として定義できます。



## (3) グループの階層化

「Group1:Group2:Group3:ElementA」のように、複数のグループ名を定義することで、グループを階層化できます。

グループを階層化することで、例えば、「Group1 のデータのうち、Group2 に属するデータの集計」といった、特定の階層に絞り込んだデータ操作もできます。

## 2.6.2 key をグルーピングした場合のデータの分散配置

グループ名が定義されたデータでは、グループ名のハッシュ値から格納先の EADs サーバが決まります。

グループが階層化されている場合は、先頭のグループ名のハッシュ値から格納先の EADs サーバが決まります。先頭のグループ名が同じデータは同じハッシュ値になるため、格納先の EADs サーバが同じになります。

例えば、次に示す key のデータは、同じ EADs サーバに格納されます。

- 「Group1:Group2:ElementA」
- 「Group1:Group3:ElementB」
- 「Group1:ElementC」

データの分散配置の概要については、「[2.5 コンシステント・ハッシングによるデータの分散配置](#)」を参照してください。

## 2.7 ユーザファンクションによる効率的なデータ処理

---

ユーザファンクションとは、データの集計や分析など、キャッシュ内の特定のデータに対して実行する一連のデータ操作（ユーザ処理）を定義したプログラムです。

ユーザが作成して、あらかじめ EADs サーバ上に配置しておき、EADs クライアントから呼び出すことで実行できます。

ユーザファンクションは、ユーザファンクションが実行された EADs サーバに格納されているデータにだけアクセスして処理を実行します。そのため、EADs クライアントを使用して処理するよりも、通信のオーバーヘッドが減り、効率的にデータを処理できます。

ユーザファンクションは、Java 言語で作成します。

### 2.7.1 前提条件

ユーザファンクションを使用する場合は、key をグルーピングすることが前提となります。あらかじめ key をグルーピングして、処理対象とするデータが特定の EADs サーバに配置されるようにしてください。

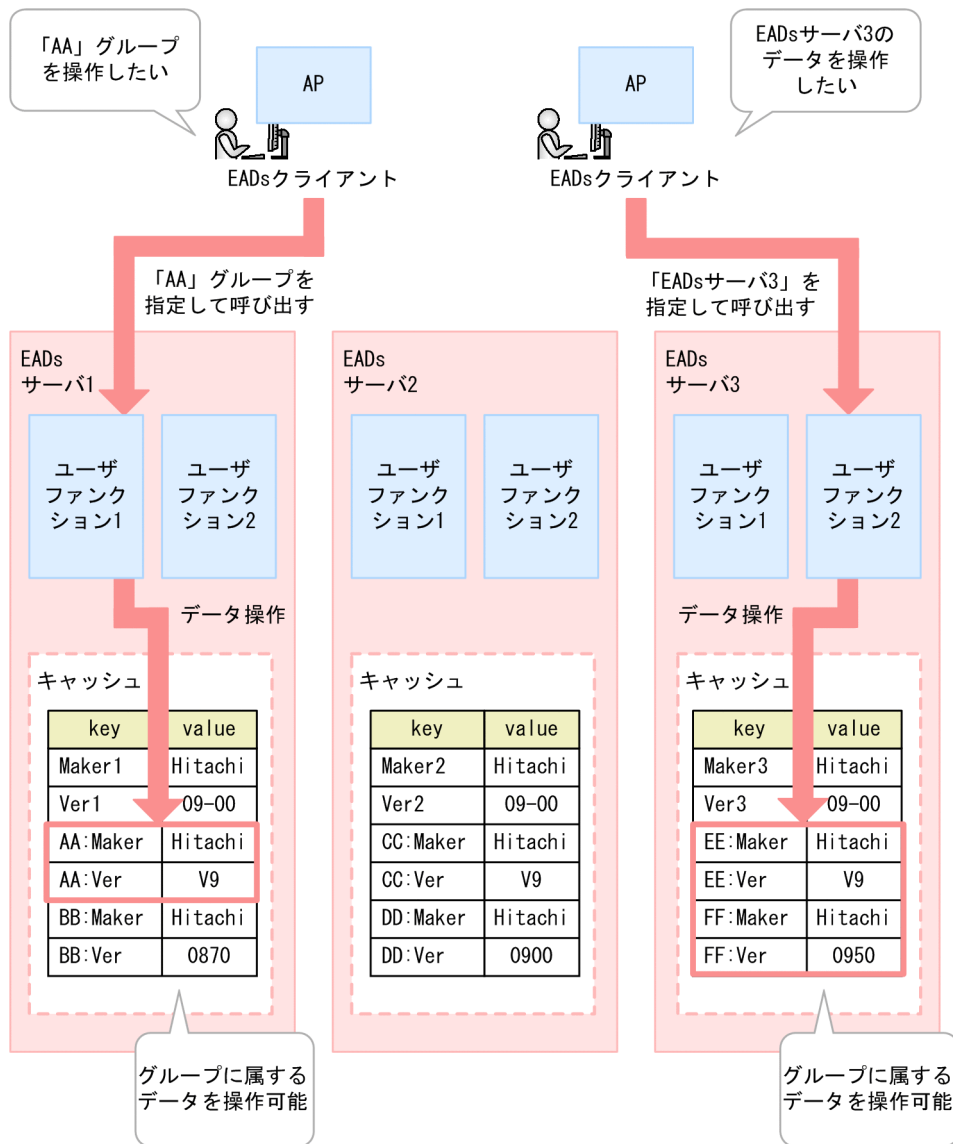
key のグルーピングについては、「[2.6 特定の EADs サーバへのデータ配置 \(key のグルーピング\)](#)」を参照してください。

### 2.7.2 ユーザファンクションの仕組み

ユーザファンクションには次の 2 種類の実行方法があります。

- グループ指定による実行方法
- EADs サーバ指定による実行方法

図 2-19 ユーザファンクションの仕組み



ユーザファンクションを使用する場合は、ユーザファンクションを作成し、クラスタ内のすべてのEADsサーバに配置します。

それぞれのユーザファンクションの仕組みについて説明します。

## (1) グループ指定による実行方法

EADsクライアントからグループ名、ユーザファンクション名を指定して、ユーザファンクションを呼び出します。指定されたグループ名のハッシュ値から、アクセスするEADsサーバを決定します。

グループ指定による実行方法では、指定したグループが属するEADsサーバに格納されていて、かつグループ化されている全データを操作できます。

## (2) EADs サーバ指定による実行方法

EADs クライアントから EADs サーバ、ユーザファンクション名を指定して、ユーザファンクションを呼び出します。指定された EADs サーバのアドレス情報 (IP アドレスとポート番号) を取得し、アクセスする EADs サーバを決定します。

EADs サーバ指定による実行方法では、指定した EADs サーバに格納されていて、かつグルーピングされている全データを操作できます。

## 2.8 データ操作を契機とするユーザ処理の実行（ユーザイベントハンドラ）

ユーザイベントハンドラとは、データの更新操作を契機に、自動的に動作するユーザ処理のプログラムです。

ユーザが作成して、あらかじめ EADs サーバ上に配置しておきます。

ユーザイベントハンドラがデータの更新操作と同期して動作することで、EADs の他クラスタと連携したり、DBMS などの他システムとデータの同期を図ったりすることができます。

ユーザイベントハンドラは、Java 言語で作成します。

### 2.8.1 ユーザイベントハンドラが動作するケース、動作しないケース

ユーザイベントハンドラが動作するケース、動作しないケースについて説明します。

#### (1) ユーザイベントハンドラが動作するケース

ユーザイベントハンドラで設定した条件を満たし、かつデータの更新操作を行った場合に、ユーザイベントハンドラが動作します。

データの更新操作とは次の API またはコマンドのことです。

- API (put, create, update, replace, remove) ※
- `eztool put` コマンド
- `eztool remove` コマンド

注※ ユーザファンクション内で実行される API を含みます。

#### (2) ユーザイベントハンドラが動作しないケース

次の場合はユーザイベントハンドラが動作しません。

- データの参照操作を行った場合  
データの参照操作とは次の API またはコマンドのことです。
  - API (get) ※
  - `eztool get` コマンド

注※ ユーザファンクション内で実行される API を含みます。

- データの更新操作で例外が発生した場合
- create, update, replace の処理で、データが更新されなかった場合

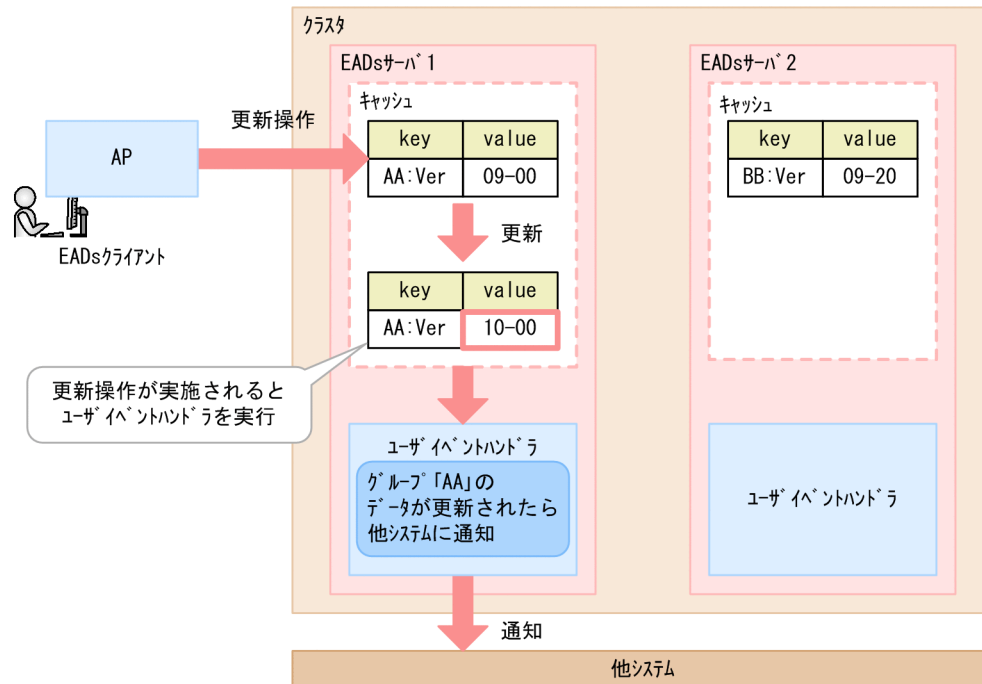


## 2.8.2 ユーザイベントハンドラの仕組み

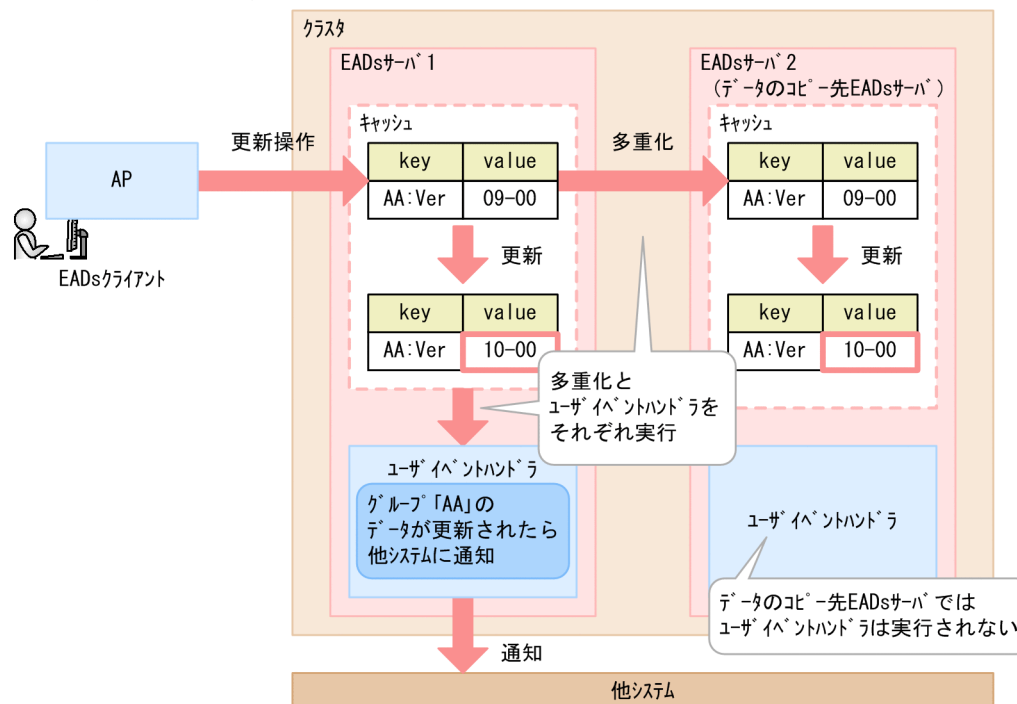
ユーザイベントハンドラの仕組みについて説明します。

図 2-20 ユーザイベントハンドラの仕組み

●データを多重化していない場合



●データを多重化している場合



ユーザイベントハンドラを使用する場合は、ユーザイベントハンドラを作成し、クラスタ内のすべてのEADsサーバに配置します。

ユーザイベントハンドラには、ユーザイベントハンドラが動作する条件（データの更新操作）と、条件を満たした場合の動作を設定します。

設定された条件を満たすデータの更新操作が行われると、ユーザイベントハンドラが実行されます。

データを多重化している場合、データの多重化とユーザイベントハンドラの処理がそれぞれ実行されます。なお、データのコピー先 EADs サーバでは、ユーザイベントハンドラを実行されません。

ユーザイベントハンドラはデータの内容を取得し、取得した情報を基に他クラスタや他システムに、データの更新操作が行われたことを通知します。他クラスタと連携する場合、ユーザイベントハンドラでクライアントライブラリを使用します。

1 回のデータの更新操作で、複数のユーザイベントハンドラが動作する場合、ユーザイベントハンドラをどの順序で実行するか、実行優先順序を指定できます。

なお、ユーザイベントハンドラでエラーが発生すると、データの更新操作は成功し、EADs クライアントにはエラーが返却されます。

## 注意事項

1 回のデータの更新操作で、複数のユーザイベントハンドラが動作する場合、ユーザイベントハンドラでエラーが発生すると、以降のユーザイベントハンドラは動作しません。

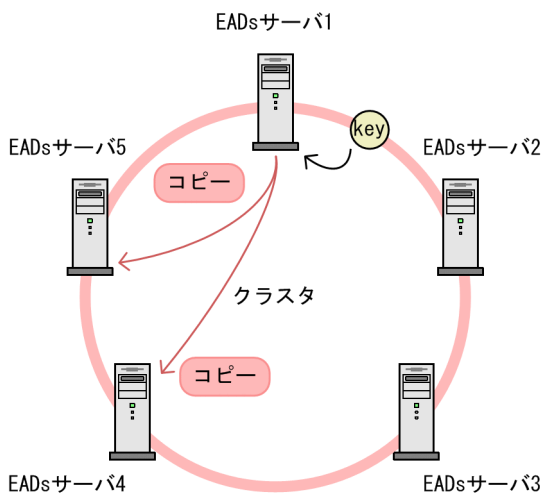
## 2.9 データの多重化

EADs では、複数の EADs サーバに自動的にデータをコピーして、データを多重化できます。そのため、ある EADs サーバに障害が発生しても、他 EADs サーバにコピーしたデータを使って処理を継続できます。また、データを多重化することによって、メモリ上のデータが障害時に消失することを防ぐことができます。

### 2.9.1 データの多重化の概要

データの多重化の概要について説明します。

図 2-21 データの多重化の概要



EADs では、指定された多重度に従って、反時計回りの方向にある複数の EADs サーバにデータをコピーします。

この図のように、多重度が3で、データの格納先が EADs サーバ 1 の場合、まず EADs サーバ 1 にデータが格納されます。それから、EADs サーバ 1 から反時計回りの方向にある EADs サーバ 5 および EADs サーバ 4 に、データが複製としてコピーされます。

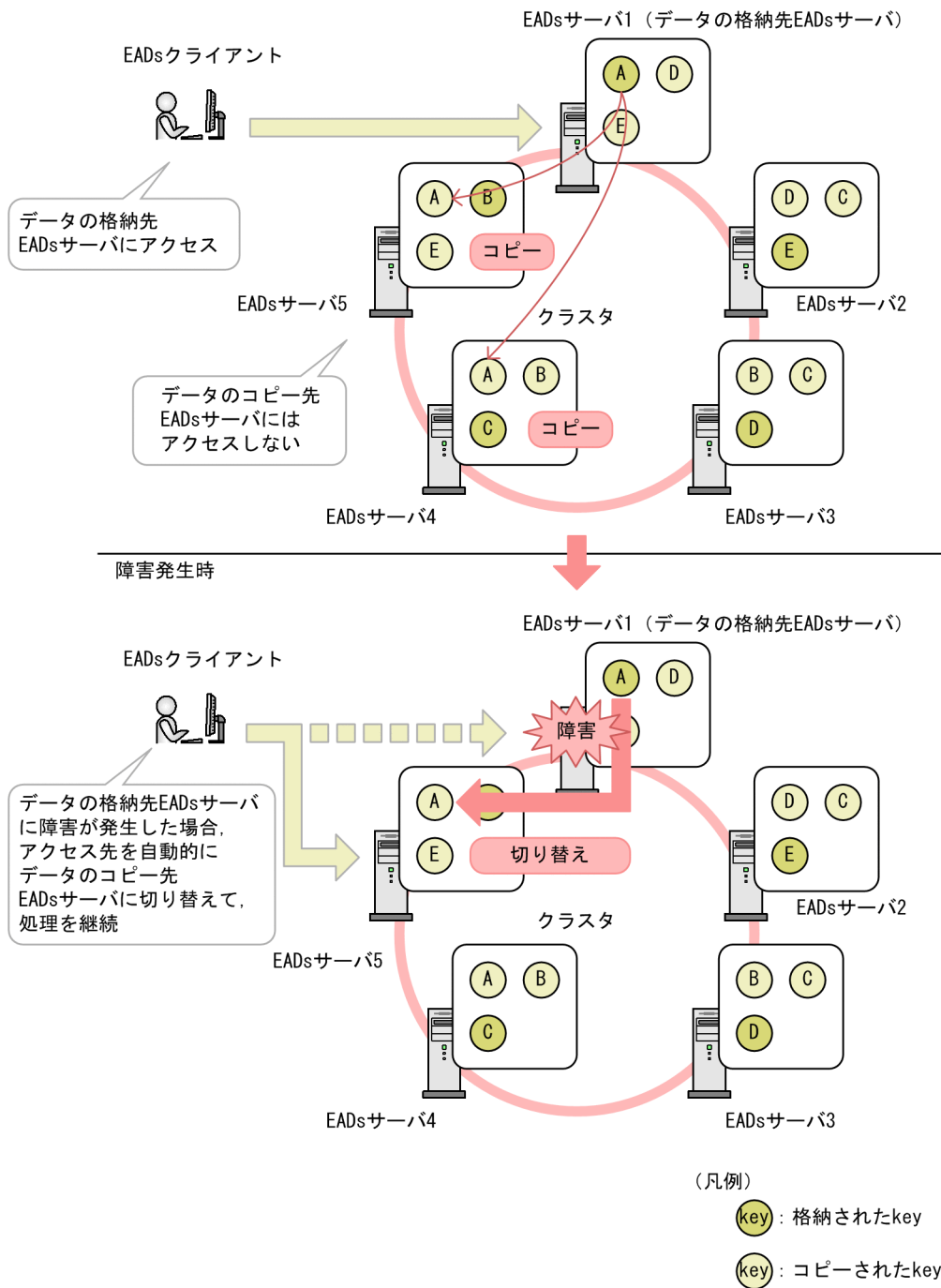
### 2.9.2 データの多重化による、障害時の処理の継続

データを多重化している場合、EADs クライアントは、データの格納先 EADs サーバにアクセスします。データの複製先 EADs サーバにはアクセスしません。

障害などが発生し、データの格納先 EADs サーバにアクセスできない場合、EADs クライアントでは、アクセス先を自動的にデータの複製先 EADs サーバに切り替えて処理を継続します。

EADs サーバ 1 をデータの格納先 EADs サーバとした場合のデータアクセスと、障害発生時のアクセス先の切り替えを次の図に示します。

図 2-22 データを多重化している場合のデータアクセスと障害発生時のアクセス先の切り替え



## 注意事項

クラスタを構成する EADS サーバ数は、「データの多重度×2-1」以上であることが前提です。

障害発生時にデータの多重度未満の数の EADS サーバがダウンしても、データの整合性は確保されます。

例えば、データの多重度が 3、EADS サーバ数が 5 つの場合、2 つの EADS サーバがダウンしても、データが欠落することはありません。

## 2.10 クラスタ監視

クラスタ内の EADs サーバは、互いにハートビート（クラスタ内で配信するパケット）を送信し合い、正常に稼働していることをクラスタ内に知らせています。ハートビートを送信しない EADs サーバに対しては、クラスタ内の EADs サーバが、その EADs サーバに対して生存確認を行います。これをクラスタ監視といいます。

クラスタ監視では、ハートビートの間隔などをユーザが指定することで、障害を検知するスピードをチューニングできます。

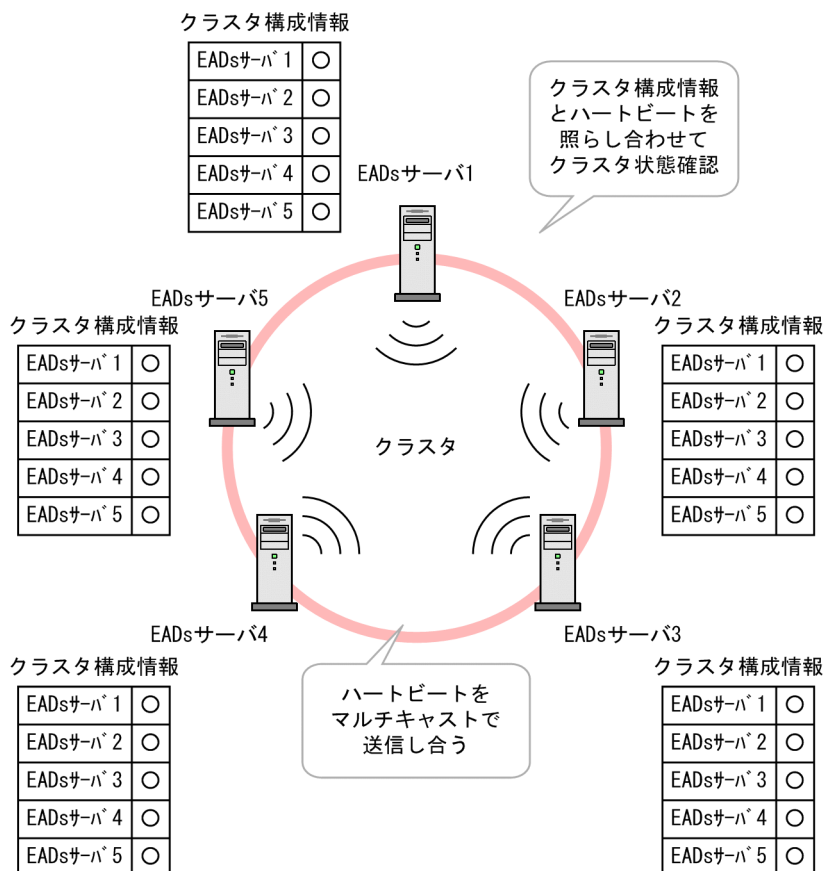
### 2.10.1 ハートビートの送信によるクラスタ監視の概要

クラスタ内の EADs サーバは、互いにハートビートをあらかじめ設定した間隔（デフォルトは 0.4 秒）で送信し合います。

クラスタ内の EADs サーバは、各自で管理するクラスタ構成情報と、マルチキャストで送信されたハートビートを照らし合わせて、クラスタの状態を確認します。

クラスタ構成情報とは、クラスタ内の EADs サーバの死活状態に関する情報のことです。

図 2-23 クラスタ監視の概要



## 2.10.2 一定数の同意によって確定する EADs サーバダウン

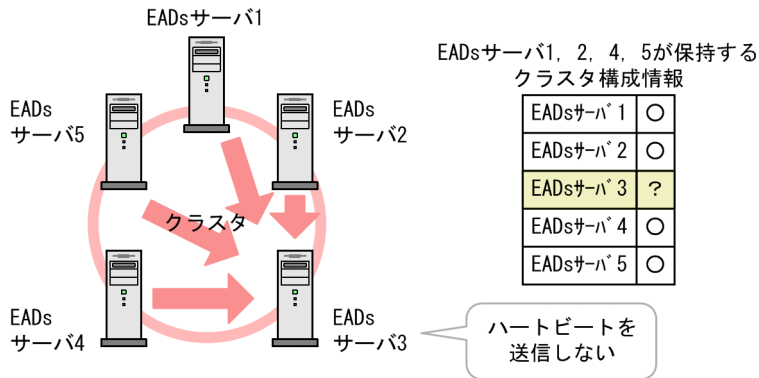
あらかじめ設定したタイムアウト時間（デフォルトは 2 秒）が経過してもハートビートを送信しない EADs サーバがある場合、クラスタ内の EADs サーバはその EADs サーバに対して生存確認を行います。生存確認がタイムアウトすると、一定数（デフォルトは 1）の同意によって EADs サーバダウンを判定します。ダウンが確定すると、その EADs サーバはクラスタから削除されます。

EADs サーバ間のハートビートの送信によって常時クラスタを監視しているため、早期に EADs サーバダウンを検知できます。

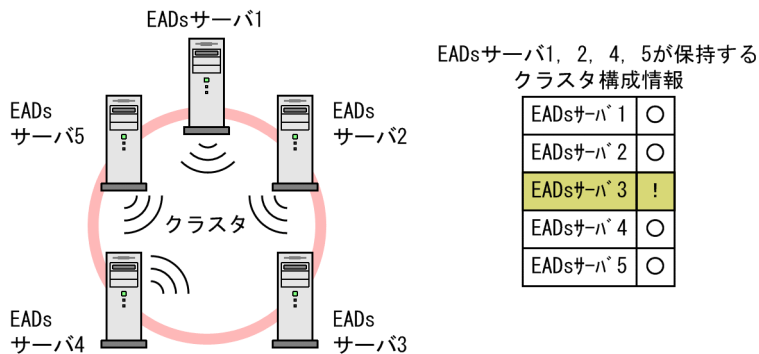
EADs サーバダウンを検知するまでの流れを、次の図に示します。

## 図 2-24 EADs サーバダウンを検知するまでの流れ

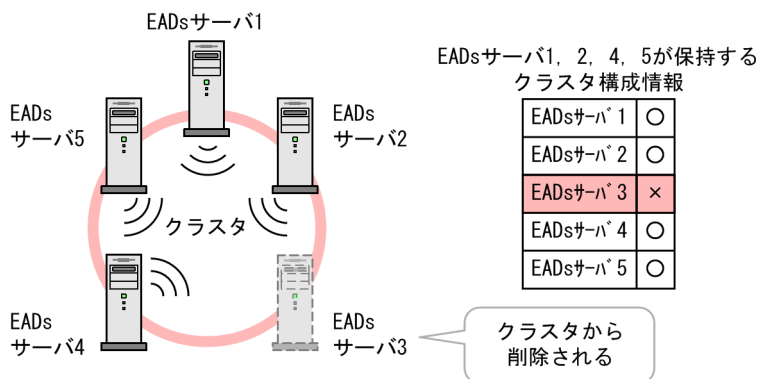
1. ハートビートを送信しないEADsサーバ3に対して生存確認を行う



2. 生存確認がタイムアウトすると「EADsサーバ3が危険状態である」という情報を付与してハートビートを送信



3. 一定数のEADsサーバがEADsサーバ3の状態が危険であることに同意すると、EADsサーバ3のダウンが確定  
クラスタ内で過半数の合意を得たら、EADsサーバ3はクラスタから削除される



1. EADsサーバ3からハートビートが送信されない場合、クラスタ内のEADsサーバはEADsサーバ3に対して生存確認を行います。
2. EADsサーバ3からハートビートが送信されないまま、生存確認がタイムアウトすると、クラスタ内のEADsサーバは、「EADsサーバ3が危険状態である」という情報を付与してハートビートを送信し合います。
3. 一定数（デフォルトは1）のEADsサーバがEADsサーバ3の状態が危険であることに同意すると、EADsサーバ3のダウンが確定します。

クラスタ内の EADs サーバに対して、EADs サーバ 3 をクラスタから削除してよいか合意を取ります。クラスタ内で過半数の合意を得たら、EADs サーバ 3 はクラスタから削除されます。削除された EADs サーバは強制的に縮退状態となります。縮退状態とは、EADs クライアントからのリクエストを受け付けない状態のことです。

## 注意事項

障害発生時にデータの多重度未満の数の EADs サーバがダウンしても、クラスタを継続できます。

例えば、データの多重度が 3、EADs サーバ数が 5 つの場合、2 つの EADs サーバがダウンしても、クラスタを継続して監視できます。



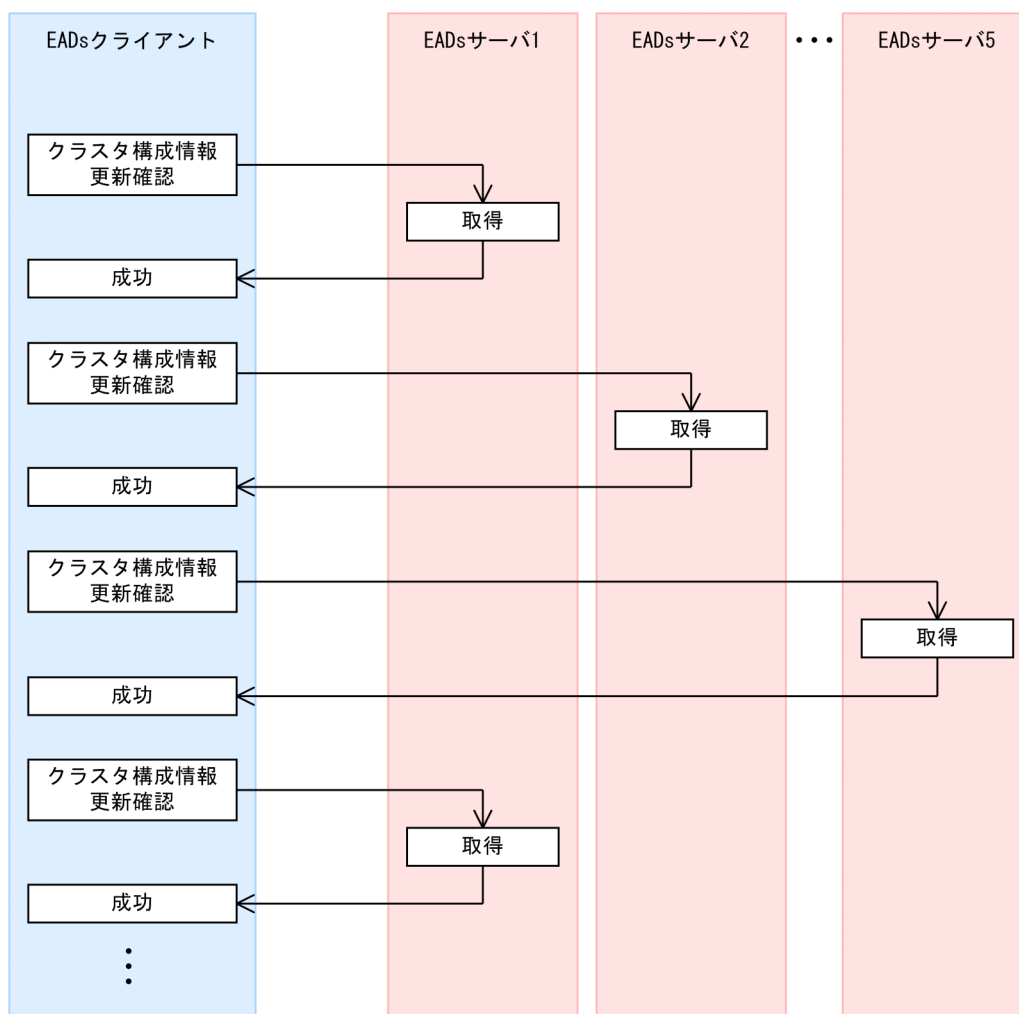
## 2.11 EADs クライアントによるクラスタ構成情報の更新確認

キャッシュのデータにアクセスする際、EADs クライアントは、EADs クライアントが保持しているクラスタ構成情報を基にデータの格納先 EADs サーバを特定します。このため、EADs クライアントからデータにアクセスする際に、データの格納先の EADs サーバを意識する必要はありません。

EADs クライアントの初期設定完了から EADs クライアントの終了までの間、EADs クライアントは、一定の間隔（デフォルトは 1 秒）でクラスタ内の 1 つ 1 つの EADs サーバからクラスタ構成情報を取得します。これによって、EADs クライアントが古いクラスタ構成情報を保持したままになることを防止します。

なお、EADs サーバからクラスタ構成情報を取得する際には、EADs クライアントが保持しているクラスタ構成情報を基に、EADs サーバの論理的な位置（ハッシュ値）の昇順にアクセスします。

図 2-25 クラスタ構成情報の更新確認の概要



タイムアウト時間を超えても応答がない場合はエラーとなります。エラーとなった場合でも、EADs クライアントは全 EADs サーバに対してクラスタ構成情報の更新確認を続けます。

## 2.12 クラスタおよび EADs サーバの状態遷移

クラスタや EADs サーバの状態によって、実行できる運用（API やコマンド）が異なります。

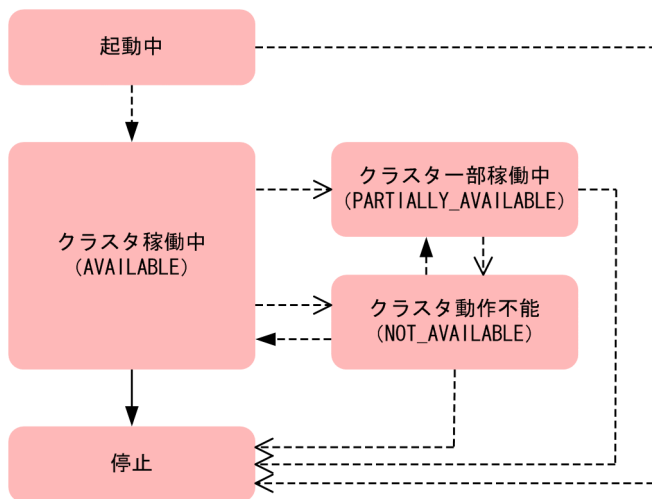
そのため、次の点に注意してください。

- AP 開発者はクラスタや EADs サーバの状態を意識して、AP を設計する必要があります。
- システム運用管理者は、クラスタおよび EADs サーバの状態を確認してからコマンドを実行する必要があります。クラスタおよび EADs サーバの状態は、`eztool status` コマンドで確認できます。

### 2.12.1 クラスタの状態遷移

クラスタの状態遷移を次の図に示します。

図 2-26 クラスタの状態遷移



(凡例)

- : クラスタの状態
- : コマンド実行による状態遷移
- - - -> : 状態遷移 (自動)
- - - -> : 障害による状態遷移 (自動)

図中の各状態の説明を、次の表に示します。

表 2-4 クラスタの状態

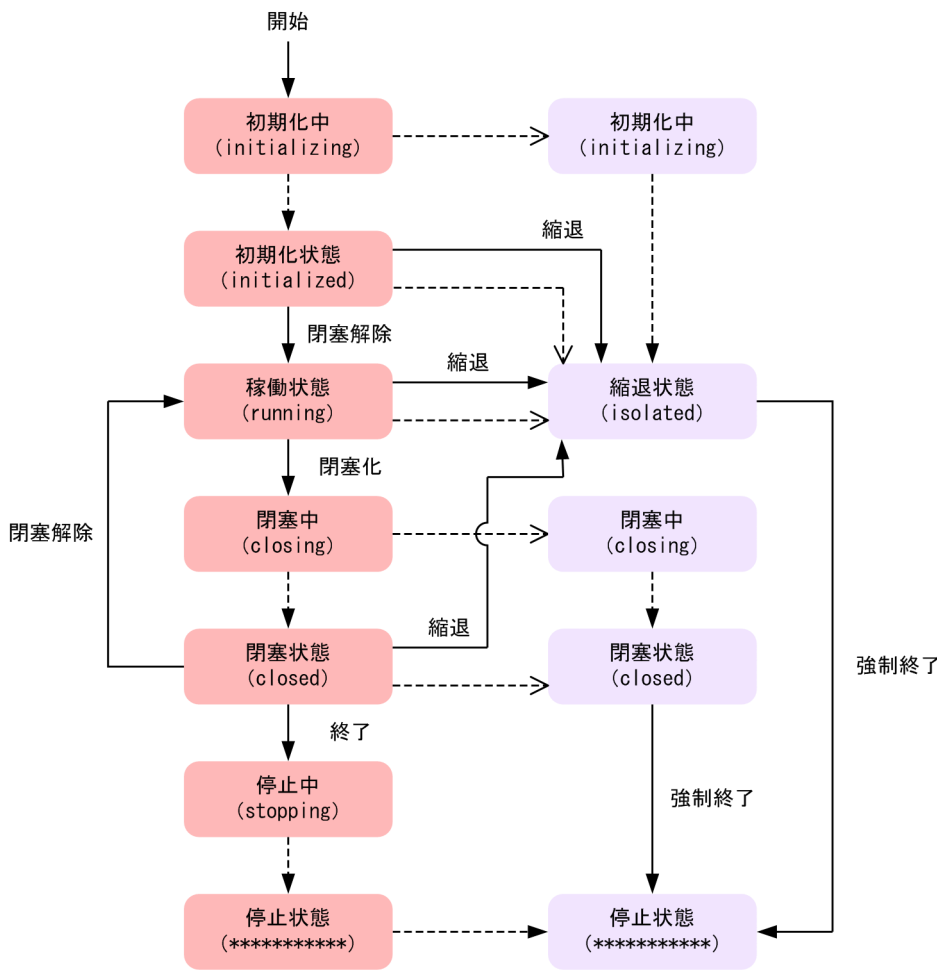
クラスタの状態		説明
状態名	eztool status コマンドでの表示内容	
クラスタ稼働中	AVAILABLE	クラスタが正常に稼働している状態です。

クラスタの状態		説明
状態名	eztool status コマンドでの表示内容	
クラスタ一部稼働中	PARTIALLY_AVAILABLE	<p>クラスタの一部が稼働している状態です。</p> <p>keyによっては、アクセスできない場合があります。</p> <p>クラスタを構成する EADs サーバ数が「データの多重度×2-1」以上の場合、多重度以上の数の EADs サーバがダウンすると、この状態に遷移する可能性があります。</p> <p>この状態に一度遷移すると、全 EADs サーバを停止しないかぎり、クラスタ稼働中 (AVAILABLE) の状態には遷移できません。</p>
クラスタ動作不能	NOT_AVAILABLE	<p>クラスタが稼働していない状態です。</p> <p>次のどちらかの場合、この状態に遷移します。</p> <ul style="list-style-type: none"> <li>• スプリットブレインが発生した場合</li> <li>• クラスタ内の半数以上の EADs サーバが同時にダウンした場合</li> </ul> <p>スプリットブレインから回復した場合は、クラスタ稼働中 (AVAILABLE) の状態またはクラスタ一部稼働中 (PARTIALLY_AVAILABLE) の状態に遷移できます。</p>

## 2.12.2 EADs サーバの状態遷移

EADs サーバの状態遷移を次の図に示します。

図 2-27 EADs サーバの状態遷移



(凡例)

- : EADsサーバの状態 (クラスタへの参加状況がonline)
- : EADsサーバの状態 (クラスタへの参加状況がoffline)
- > : コマンド実行による状態遷移
- > : 状態遷移 (自動)
- > : 障害による状態遷移 (自動)

図中の各状態の説明を、次の表に示します。

表 2-5 EADs サーバの状態

EADs サーバの状態		説明
状態名	eztool status コマンドでの表示内容	
初期化中	initializing	EADs サーバを初期化中の状態です。
初期化状態	initialized	EADs サーバの初期化が完了した状態です。閉塞状態と同じく、EADs クライアントからのリクエストは受け付けられない状態です。

EADs サーバの状態		説明
状態名	eztool status コマンドでの表示内容	
稼働状態	running	EADs クライアントからのリクエストが受け付けられる状態です。
閉塞中	closing	EADs クライアントからのリクエストは受け付けられませんが、まだ動作中のスレッドが残っている状態です。
閉塞状態	closed	EADs クライアントからのリクエストが受け付けられない状態です。
縮退状態	isolated	閉塞状態と同じく、EADs クライアントからのリクエストが受け付けられない状態です。 通信障害などによって、クラスタから削除された EADs サーバは、強制的に縮退状態となります。 縮退状態になった EADs サーバは、 <code>eztool isolate --stop</code> コマンドを実行して、個別に終了してください。
停止中	stopping	EADs サーバを停止処理中の状態です。
停止状態	*****	EADs サーバを停止した状態です。

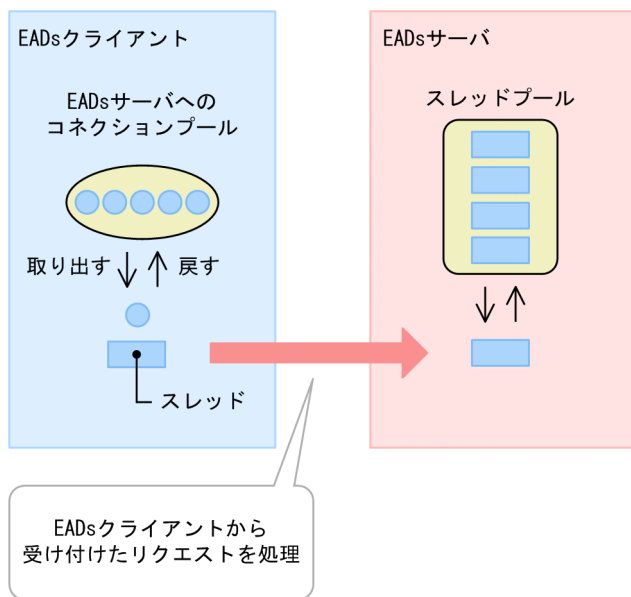
## 2.13 スレッドプールおよびコネクションプールを使用したスループットの向上

EADs クライアント・EADs サーバ間では、スレッドプールおよびコネクションプールを使用します。

あらかじめスレッドやコネクションをプールしておくことで、リクエスト集中時の EADs サーバ全体のレスポンス速度の低下を防ぐことができます。

EADs を使用するシステムの要件に応じて、スレッドプールおよびコネクションプールをチューニングしてください。

図 2-28 スレッドプールおよびコネクションプールの概要



### 2.13.1 スレッドプール

EADs サーバでは、EADs クライアントからのリクエストを処理するためのスレッドをあらかじめ生成してプールしておきます。これをスレッドプールと呼びます。

スレッドプールには、EADs サーバ起動時に最大同時接続数と同数のスレッドがプールされます。

EADs サーバでは、リクエストを受け付けるとスレッドプールからスレッドが割り当てられ、リクエストが処理されます。

スレッドプールに空いているスレッドがない場合、エラーとなって通信が切断されます。最大同時実行スレッド数を超えるリクエストを受け付けた場合は、処理待ちになります。

リクエストの処理が完了すると、スレッドは次のリクエストを待ちます。

スレッドプールを使用することで、何度もスレッドを生成・削除する必要がなくなるため、スレッド生成・削除に掛かるオーバーヘッドを軽減できます。

## 2.13.2 コネクションプール

EADs クライアントでは、一度確立されたコネクションを接続先ごとにプールして再利用します。これをコネクションプールと呼びます。

通信開始時、EADs クライアントは接続先のコネクションがすでに存在するかどうか確認します。コネクションがプールされている場合は、そのコネクションを使用して通信を開始します。

通信が終了すると、使用していたコネクションはコネクションプールに戻されます。

リクエストが集中してコネクションプールのコネクションがすべて使用されている場合、プールできるコネクションの最大個数分まで、新しくコネクションを確立します。

すでにプールできるコネクションの最大個数に達している場合は、コネクションがプールに戻されるまで処理待ちとなります。なお、処理待ちにしないでエラーにすることもできます。

コネクションプールを使用することで、何度もコネクションを確立する必要がなくなるため、通信時のオーバーヘッドを軽減できます。

# 3

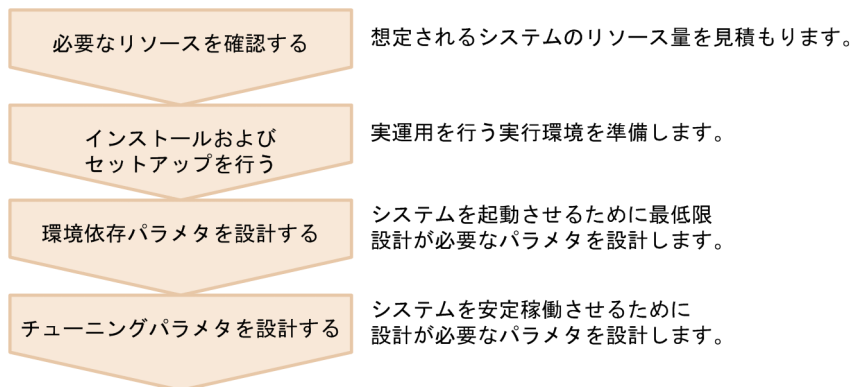
## システム設計・構築の流れ

この章では、EADsを導入したシステムの設計・構築の流れについて説明します。



## 3.1 システム設計・構築の流れ

システム設計・構築の流れを次の図に示します。



### 3.1.1 必要なリソースを確認する

システムを構築する前に、想定されるシステムのリソース量を見積もり、それを処理するのに十分なマシン台数や EADs サーバ数を決定します。

なお、クラスタを構成する EADs サーバ数は、「データの多重度×2-1」以上であることが前提です。

[4. 必要なリソースを確認する] を参照してください。

### 3.1.2 インストールおよびセットアップを行う

前提となるプログラムプロダクトをインストールし、EADs サーバおよび EADs クライアントをセットアップします。また、準備した実行環境で実運用に近い負荷を掛けてテストを行います。

[5. インストールおよびセットアップを行う (EADs サーバ)] または [6. インストールおよびセットアップを行う (EADs クライアント)] を参照してください。

### 3.1.3 環境依存パラメタを設計する

システムの環境依存パラメタを設計します。

環境依存パラメタとは、システムを構築する環境に依存し、システムを起動させるために最低限設計が必要なパラメタのことです。

[7. 環境依存パラメタを設計する (EADs サーバ)] または [8. 環境依存パラメタを設計する (EADs クライアント)] を参照してください。

### 3.1.4 チューニングパラメタを設計する

システムのチューニングパラメタを設計します。

チューニングパラメタとは、システムを安定稼働させるために設計が必要なパラメタのことです。

[9. チューニングパラメタを設計する] を参照してください。

# 4

## 必要なリソースを確認する

この章では、メモリ使用量やディスク使用量の見積もり方法について説明します。

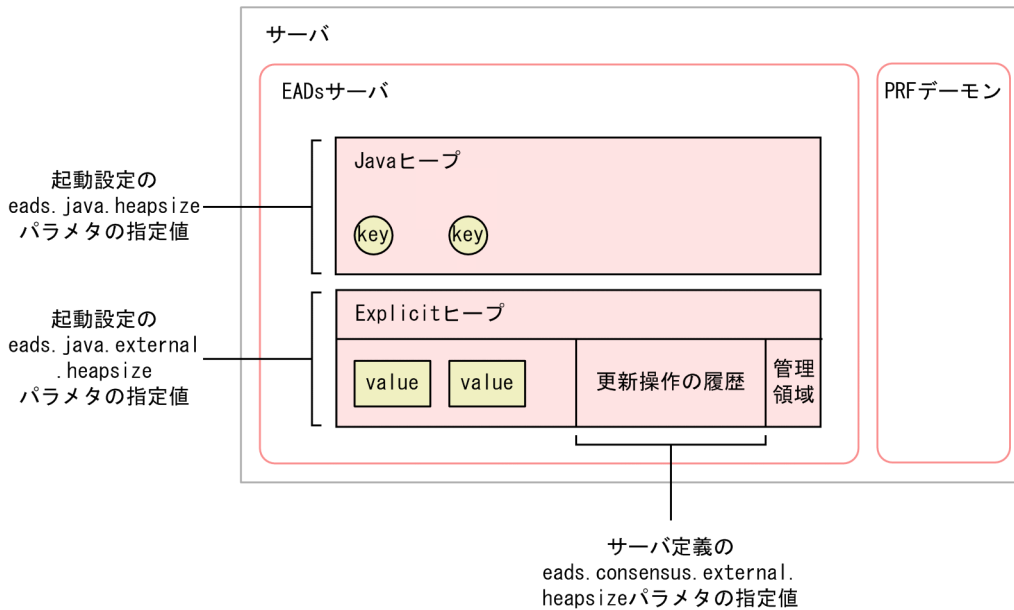
## 4.1 メモリ使用量を見積もる

EADs を使用するために必要なメモリ使用量の見積もりについて説明します。

### 4.1.1 メモリ構成

EADs サーバが使用するメモリの構成を次の図に示します。

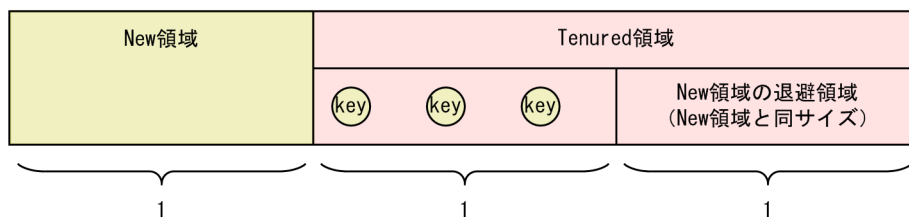
図 4-1 メモリ構成



#### (1) Java ヒープ

Java ヒープは大きく New 領域と Tenured 領域から構成され、Tenured 領域にデータの key 部分が格納されます。

Java ヒープ



格納される key が Tenured 領域の 1/2 を超えてしまうと、FullGC (フルガーベージコレクション) が発生します。そのため、Tenured 領域サイズを基に Java ヒープサイズを見積もる必要があります。Java ヒープサイズの見積もりについては、「[4.1.2 Java ヒープサイズを見積もる](#)」を参照してください。

## (2) Explicit ヒープ

Explicit ヒープは、key と value のうち、value 部分を格納するための領域と、更新操作の履歴が格納される領域から構成されます。

更新操作の履歴とは、API の種別、および key と value の情報のことです。

なお、Explicit ヒープサイズの 3%は管理領域として使用されます。

Explicit ヒープサイズの見積もりについては、「[4.1.3 Explicit ヒープサイズを見積もる](#)」を参照してください。

### 4.1.2 Java ヒープサイズを見積もる

Java ヒープサイズを見積もる計算式を次に示します。

#### (1) 1EADs サーバ当たりの Java ヒープサイズを見積もる

1EADs サーバ当たりの Tenured 領域サイズを 3 倍にした値が Java ヒープサイズとなります。

算出した値は、起動設定の `eads.java.heapsize` パラメータに指定します。デフォルトは 3 ギガバイトです。

#### 参考

ディスク上にキャッシュを作成する場合は、そのキャッシュが使用する 1EADs サーバ当たりのメモリ使用量の合計を、Java ヒープサイズに加算します。見積もりについては、「[4.1.2\(2\) ディスク上のキャッシュが使用する Java ヒープサイズを見積もる](#)」を参照してください。

```
Java ヒープサイズ (単位:メガバイト) =
  {(クラスタに格納できる key の最大サイズ (単位:バイト) + 750 バイト)
  × 1EADs サーバに格納するデータ件数
  + (クラスタに格納できる key の最大サイズ (単位:バイト) + 250 バイト)
  × グループ名の数※1
  + (EADs サーバへの最大同時接続数 × データの多重度)
  × ((データの送受信バッファのサイズ (単位:バイト) × 2 + 15 × 1,024)
  + (データの送受信バッファのサイズ (単位:バイト) × 2 + 3,072)) ※2
  + EADs サーバ数 × 409,600 バイト
  + 更新操作の履歴の管理で使用する Java ヒープ領域 (単位:バイト)
  + 縮退処理・復旧処理で使用する Java ヒープ領域 (単位:バイト)
  + ユーザイベントハンドラで使用する Java ヒープ領域 (単位:バイト)}
  × 3 ÷ 1,0242
```

#### 注※1

グループを階層化している場合は、先頭のグループ名の数です。

#### 注※2

この括弧内の計算式は 64 キロバイト単位で切り上げて見積もってください。

クラスタに格納できる key の最大サイズ (単位: バイト):

クラスタ定義の `eads.server.max_key_size` パラメタの指定値

EADs サーバへの最大同時接続数:

サーバ定義の `eads.server.max_connections` パラメタの指定値

データの多重度:

クラスタ定義の `eads.replication.factor` パラメタの指定値

データの送受信バッファのサイズ (単位: バイト):

サーバ定義の `eads.connection.buffer_size` パラメタの指定値

更新操作の履歴の管理で使用する Java ヒープ領域 (単位: バイト):

更新操作の履歴の管理で使用する Java ヒープ領域を見積もる計算式を次に示します。

更新操作の履歴の管理で使用する Java ヒープ領域 (単位: バイト) =  
3,584 × (更新操作の履歴が保持される最大数  
+ 合意処理で一度に合意できる処理の最大数)  
× (データの多重度 × 2 - 1) × キャッシュ数  
+ 16 × 合意メッセージの送信キューの長さ × (EADs サーバ数 - 1)

更新操作の履歴が保持される最大数:

更新操作の履歴が保持される最大数を見積もる計算式を次に示します。

更新操作の履歴が保持される最大数 =  
(ハートビートのタイムアウト時間 ÷ 1,000)  
× 1EADs サーバへのデータの更新操作のスループット (単位: 件/秒) ※

ハートビートのタイムアウト時間:

サーバ定義の `eads.cluster.heartbeat.timeout` パラメタの指定値

#### 注※

キャッシュごとにスループットが大きく異なる場合は、キャッシュごとに算出したスループットのうち、最大の値を代入してください。

合意処理で一度に合意できる処理の最大数:

クラスタ定義の `eads.cache.prepare.num` パラメタの指定値

データの多重度:

クラスタ定義の `eads.replication.factor` パラメタの指定値

合意メッセージの送信キューの長さ:

サーバ定義の `eads.consensus.message.send.queue_size` パラメタの指定値

縮退処理・復旧処理で使用する Java ヒープ領域 (単位: バイト):

縮退処理および復旧処理で使用する Java ヒープ領域の最大値をそれぞれ見積もり、どちらか大きい方の値を代入してください。

縮退処理で使用する Java ヒープ領域 (単位: バイト) =  
(合意処理で一度に合意できる処理の最大数 × 2  
× 更新操作の履歴の最大サイズ (単位: バイト)  
× (キャッシュ単位に行われる処理の最大同時実行スレッド数  
+ キャッシュ数) + (データの多重度 - 1) × キャッシュ数  
× 更新操作の履歴の補完処理で送信するデータサイズ (単位: バイト))  
× MIN (4 × (データの多重度 - 1), EADs サーバ数 - 1)

復旧処理で使用する Java ヒープ領域 (単位: バイト) =  
更新操作の履歴が格納される領域のサイズ (単位: メガバイト) × 1,024<sup>2</sup>  
÷ (データの多重度 × 2 - 1)  
+ 復旧処理で送信するデータサイズ (単位: バイト)

合意処理で一度に合意できる処理の最大数:

クラスタ定義の `eads.cache.prepare.num` パラメタの指定値

更新操作の履歴の最大サイズ (単位: バイト):

更新操作の履歴の最大サイズを見積もる計算式を次に示します。

更新操作の履歴の最大サイズ (単位: バイト) =  
クラスタに格納できる key の最大サイズ + value の最大サイズ × 2

クラスタに格納できる key の最大サイズ:

クラスタ定義の `eads.server.max_key_size` パラメタの指定値

value の最大サイズ:

`put`, `create`, `update`, `replace` 実行時に指定する value の最大サイズ

キャッシュ単位に行われる処理の最大同時実行スレッド数:

サーバ定義の `eads.cluster.controller.cache.max_execute_threads` パラメタの指定値

このパラメタに 0 を指定している場合、データの多重度 - 1 を代入します。

データの多重度:

クラスタ定義の `eads.replication.factor` パラメタの指定値

更新操作の履歴の補完処理で送信するデータサイズ (単位: バイト):

サーバ定義の `eads.consensus.fillgap.copy.dataSize` パラメタの指定値

MIN:

計算結果の最も小さい値を選ぶことを示しています。

(例) MIN (3 × 6, 4 + 7) の計算結果は 11 となります。

更新操作の履歴が格納される領域のサイズ:

サーバ定義の `eads.consensus.external.heapsize` パラメタの指定値

見積もりについては、「4.1.3(2) 更新操作の履歴が格納される領域のサイズ」を参照してください。

復旧処理で送信するデータサイズ：

復旧処理で送信するデータサイズをキャッシュごとに見積もり、その中で最大の値を代入してください。

- メモリキャッシュの場合  
サーバ定義の `eads.restore.dataSender.sendSize` パラメタの指定値
- ディスクキャッシュ、および 2Way キャッシュの場合  
キャッシュ定義の `eads.cache.restore.dataSender.sendSize` パラメタの指定値

ユーザイベントハンドラで使用する Java ヒープ領域（単位：バイト）：

ユーザイベントハンドラで使用する Java ヒープ領域を見積もる計算式を次に示します。ただし、ユーザイベントハンドラを使用しない場合は 0 としてください。

ユーザイベントハンドラで使用する Java ヒープ領域（単位：バイト） =  
(クラスタに格納できる key の最大サイズ（単位：バイト） + 100 バイト)  
× MIN（データ操作の最大同時実行スレッド数 + ユーザファンクション全体の最大同時実行スレッド数,  
EADs サーバへの最大同時接続数）

クラスタに格納できる key の最大サイズ（単位：バイト）：

クラスタ定義の `eads.server.max_key_size` パラメタの指定値

MIN：

計算結果の最も小さい値を選ぶことを示しています。

（例）MIN（3×6, 4 + 7）の計算結果は 11 となります。

データ操作の最大同時実行スレッド数：

サーバ定義の `eads.server.cache.max_execute_threads` パラメタの指定値

ユーザファンクション全体の最大同時実行スレッド数：

サーバ定義の `eads.server.function_container.max_execute_threads` パラメタの指定値

EADs サーバへの最大同時接続数：

サーバ定義の `eads.server.max_connections` パラメタの指定値

## (2) ディスク上のキャッシュが使用する Java ヒープサイズを見積もる

ディスクキャッシュ、および 2Way キャッシュを使用する場合は、各キャッシュが使用する Java ヒープサイズを求めます。

ディスクキャッシュ、および 2Way キャッシュ 1 つ当たりの Java ヒープサイズを見積もる計算式を次に示します。この計算式でキャッシュごとに見積もったあと、算出した値を合計します。その値を「4.1.2(1) 1EADs サーバ当たりの Java ヒープサイズを見積もる」で見積もった Java ヒープサイズに加算します。

- ディスクキャッシュの場合



Java ヒープサイズの見積もりの値に加算する値 (単位: キロバイト) =  
(1,600 + 8×1 レンジ当たりのキャッシュデータファイルの面数  
+ 0.8×格納するデータ件数) ×データの多重度

- 2Way キャッシュの場合※

Java ヒープサイズの見積もりの値に加算する値 (単位: キロバイト) =  
(1,600 + 8×1 レンジ当たりのキャッシュデータファイルの面数  
- 0.4×格納するデータ件数) ×データの多重度

注※

格納するデータ件数が多いときは、計算結果が負の値になることがあります。そのときは、負の値のまま代入してください。

1 レンジ当たりのキャッシュデータファイルの面数:

キャッシュ定義の `eads.cache.disk.filenum` パラメタの指定値

見積もりについては、「4.4.1 キャッシュデータファイルのサイズおよび面数を見積もる」を参照してください。

格納するデータ件数:

格納するデータ件数を見積もる計算式を次に示します。

格納するデータ件数 =  
クラスタ全体 (1 キャッシュ) に格納するデータ件数 ÷ クラスタを構成する EADs サーバ数

データの多重度:

クラスタ定義の `eads.replication.factor` パラメタの指定値

### 4.1.3 Explicit ヒープサイズを見積もる

Explicit ヒープサイズを見積もる計算式を次に示します。

key と value のうち、value 部分が格納される領域と、更新操作の履歴が格納される領域を足した値が Explicit ヒープサイズとなります。なお、Explicit ヒープサイズの 3% は管理領域として使用されます。

算出した値は、起動設定の `eads.java.external.heapsize` パラメタに指定します。

Explicit ヒープサイズ (単位: メガバイト) =  $(a \div 1,024^2 + b \div 1,024^2) \div 0.97$

#### 変数の説明

a: value 部分が格納される領域のサイズ

b: 更新操作の履歴が格納される領域のサイズ

各領域のサイズを見積もる計算式を次に示します。

## (1) value 部分が格納される領域のサイズ

key と value のうち、value 部分が格納される領域のサイズを見積もる計算式を次に示します。ただし、value を Explicit ヒープに格納しない（キャッシュタイプがディスクキャッシュのキャッシュだけを使用する）場合は、0 としてください。

$$\begin{aligned} & \text{value 部分が格納される領域のサイズ (単位: バイト) =} \\ & \quad (\text{メモリキャッシュおよび 2Way キャッシュに格納するデータ 1 件当たりの value サイズ}^{\ast 1} \text{ (単位: バイト)} \\ & \quad + 2 \text{ バイト}) \ast 2 \\ & \quad \times (\text{メモリキャッシュおよび 2Way キャッシュに格納するデータ件数の合計} \\ & \quad + \text{メモリキャッシュおよび 2Way キャッシュのキャッシュ数の合計} \\ & \quad \times \text{データの多重度} + 500) \end{aligned}$$

注※1

value のサイズについては、「15.2.2(2) value として指定できるデータ」を参照してください。

注※2

この括弧内の計算式は 16 バイト単位で切り上げて見積もってください。

データの多重度:

クラスタ定義の `eads.replication.factor` パラメタの指定値

## (2) 更新操作の履歴が格納される領域のサイズ

更新操作の履歴が格納される領域のサイズを見積もる計算式を次に示します。

算出した値は、メガバイト単位に切り上げてから、サーバ定義の `eads.consensus.external.heapsize` パラメタに指定します。

$$\begin{aligned} & \text{更新操作の履歴が格納される領域 (単位: バイト) =} \\ & \quad \{(\text{データの多重度} \times 2 - 1) \times \text{キャッシュ数} \times \text{更新操作の履歴の最大数} + 500\} \\ & \quad \times (\text{更新操作の履歴の最大サイズ (単位: バイト)} + 1,024) + 1,048,576 \end{aligned}$$

注 小数点以下の値は切り上げます。

データの多重度:

クラスタ定義の `eads.replication.factor` パラメタの指定値

更新操作の履歴の最大数:

更新操作の履歴の最大数を見積もる計算式を次に示します。

$$\begin{aligned} & \text{更新操作の履歴の最大数 =} \\ & \quad \text{更新操作の履歴が保持される最大数} + \text{合意処理で一度に合意できる処理の最大数} \end{aligned}$$

更新操作の履歴が保持される最大数:

更新操作の履歴が保持される最大数を見積もる計算式を次に示します。

更新操作の履歴が保持される最大数 =  
(ハートビートのタイムアウト時間 ÷ 1,000)  
× 1EADs サーバへのデータの更新操作のスループット (単位: 件/秒) ※

ハートビートのタイムアウト時間:

サーバ定義の `eads.cluster.heartbeat.timeout` パラメタの指定値

注※

キャッシュごとにスループットが大きく異なる場合は、キャッシュごとに算出したスループットのうち、最大の値を代入してください。

合意処理で一度に合意できる処理の最大数:

クラスタ定義の `eads.cache.prepare.num` パラメタの指定値

更新操作の履歴の最大サイズ (単位: バイト):

更新操作の履歴の最大サイズを見積もる計算式を次に示します。

更新操作の履歴の最大サイズ (単位: バイト) =  
クラスタに格納できる key の最大サイズ + value の最大サイズ × 2

クラスタに格納できる key の最大サイズ:

クラスタ定義の `eads.server.max_key_size` パラメタの指定値

value の最大サイズ:

put, create, update, replace 実行時に指定する value の最大サイズ

## 注意事項

- Explicit ヒープの見積もりが適切でない場合、Explicit ヒープの空き容量が少なくなると、value および更新操作の履歴のフラグメントが発生し、性能が低下するおそれがあります。また、Explicit ヒープの空き容量が不足して、value を格納できなくなると、EADs サーバが縮退します。
- Explicit ヒープの空き容量が不足して更新操作の履歴が格納できなくなると、EADs サーバは既存の更新操作の履歴を削除することで Explicit ヒープを確保しようとします。これによって、EADs サーバ間での更新操作の履歴の補完処理に失敗するおそれがあります。更新操作の履歴の補完処理については、「[9.3.2\(6\) 更新操作の履歴の補完処理](#)」を参照してください。

## 4.1.4 EADs サーバのメモリ使用量を見積もる

EADs サーバのメモリ使用量を見積もる計算式を次に示します。

EADs サーバのメモリ使用量 (単位: メガバイト) =  
Java ヒープサイズ + Explicit ヒープサイズ + 150  
+ (EADs サーバへの最大同時接続数 × データの多重度) × 52 ÷ 1,024

EADs サーバへの最大同時接続数：

サーバ定義の `eads.server.max_connections` パラメタの指定値

データの多重度：

クラスタ定義の `eads.replication.factor` パラメタの指定値

## 4.2 ディスク使用量を見積もる

EADs を使用するために必要なディスク使用量の見積もりについて説明します。

### 4.2.1 EADs サーバのディスク使用量を見積もる

EADs サーバのディスク使用量の見積もりについて説明します。

ディスクキャッシュ、および 2Way キャッシュを使用する場合は、ここで算出する次の値に、「[4.4 キャッシュファイルの容量を見積もる](#)」で見積もったディスク使用量を加算してください。

- 1EADs サーバ当たりのログファイルの容量
- ストアデータファイルの容量

#### (1) 1EADs サーバ当たりのログファイルの容量を算出する

1EADs サーバ当たりのログファイルの容量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{1EADs サーバ当たりのログファイルの容量 (単位: メガバイト) =} \\ & a + b + c \times 2 + d + e + f + 100 \text{ メガバイト} \end{aligned}$$

##### 変数の説明

- a: EADs サーバによって出力されるメッセージログファイルの容量
- b: コマンド実行時に出力されるメッセージログファイルの容量
- c: Java ログファイルの容量
- d: PRF トレースファイルの容量
- e: 統計情報ファイルの容量
- f: ユーザログファイルの容量

各ログファイルの容量を見積もる計算式を次に示します。

なお、EADs サーバが管理するログファイルについては、「[7.4.1 ログファイルの種類](#)」を参照してください。

#### (a) EADs サーバによって出力されるメッセージログファイルの容量

EADs サーバによって出力されるメッセージログファイルの容量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{EADs サーバによって出力されるメッセージログファイルの容量} \\ & \text{(単位: メガバイト) =} \\ & \text{メッセージログファイルのファイルサイズ (単位: バイト) } \div 1,024^2 \\ & \times \text{メッセージログファイルのファイル数} \end{aligned}$$

メッセージログファイルのファイルサイズ (単位: バイト):

サーバ定義の `eads.logger.message.filesize` パラメタの指定値

メッセージログファイルのファイル数:

サーバ定義の `eads.logger.message.filenum` パラメタの指定値

サーバ定義の `eads.logger.message.filesize` パラメタの指定値はバイト単位に指定するので、メガバイト単位に切り上げて計算します。

## (b) コマンド実行時に出力されるメッセージログファイルの容量

コマンド実行時に出力されるメッセージログファイルの容量を見積もる計算式を次に示します。

コマンド実行時に出力されるメッセージログファイルの容量

(単位: メガバイト) =

メッセージログファイルのファイルサイズ (単位: バイト)  $\div$  1,024<sup>2</sup>  
×メッセージログファイルのファイル数

メッセージログファイルのファイルサイズ (単位: バイト):

コマンド定義の `eads.logger.message.filesize` パラメタの指定値

メッセージログファイルのファイル数:

コマンド定義の `eads.logger.message.filenum` パラメタの指定値

コマンド定義の `eads.logger.message.filesize` パラメタの指定値はバイト単位に指定するので、メガバイト単位に切り上げて計算します。

## (c) Java ログファイルの容量

Java ログファイルの容量を見積もる計算式を次に示します。

Java ログファイルの容量 (単位: メガバイト) =

Java ログファイルのファイルサイズ (単位: メガバイト)  
×Java ログファイルのファイル数

Java ログファイルのファイルサイズ (単位: メガバイト):

起動設定の `eads.java.log.filesize` パラメタの指定値

Java ログファイルのファイル数:

起動設定の `eads.java.log.filenum` パラメタの指定値

## (d) PRF トレースファイルの容量

PRF トレースファイルの容量を見積もる計算式を次に示します。

PRF トレースファイルの容量 (単位: メガバイト) =

150 +

(PRF デーモンが共用メモリに確保するバッファのサイズ (単位: キロバイト)  
+ 20) ÷ 1,024 × 5  
+ PRF トレースファイルのファイルサイズ (単位: キロバイト) ÷ 1,024  
× PRF トレースファイルのファイル数

PRF デーモンが共用メモリに確保するバッファのサイズ (単位: キロバイト):

起動設定の `eads.prf.buffersize` パラメタの指定値

PRF トレースファイルのファイルサイズ (単位: キロバイト):

起動設定の `eads.prf.filesize` パラメタの指定値

PRF トレースファイルのファイル数:

起動設定の `eads.prf.fileenum` パラメタの指定値

起動設定の `eads.prf.buffersize` パラメタおよび `eads.prf.filesize` パラメタの指定値はキロバイト単位に指定するので、メガバイト単位に切り上げて計算します。

## (e) 統計情報ファイルの容量

統計情報ファイルの容量を見積もる計算式を次に示します。

統計情報ファイルの容量 (単位: メガバイト) =  
(統計情報ファイル (`eads_stats.csv`) の容量  
+ ユーザ関クションの統計情報ファイル (`eads_function_stats.csv`) の容量  
+ ユーザイベントハンドラの統計情報ファイル (`eads_event_stats.csv`) の容量)  
÷ 1,024<sup>2</sup>

各統計情報ファイルの容量を見積もる計算式を次に示します。

### ■ 統計情報ファイル (`eads_stats.csv`) の容量

統計情報ファイル (`eads_stats.csv`) の容量 (単位: バイト) =  
{1,024 + 1,024 × (86,400 ÷ 統計情報を出力する間隔)}  
× (統計情報の取得ファイル数 + 1)

### ■ ユーザ関クションの統計情報ファイル (`eads_function_stats.csv`) の容量

ユーザ関クションの統計情報ファイル (`eads_function_stats.csv`) の容量  
(単位: バイト) =  
{1,024 + (2,048 × ユーザ関クション数)  
× (86,400 ÷ 統計情報を出力する間隔)}  
× (統計情報の取得ファイル数 + 1)

### ■ ユーザイベントハンドラの統計情報ファイル (`eads_event_stats.csv`) の容量

ユーザイベントハンドラの統計情報ファイル (`eads_event_stats.csv`) の容量  
(単位: バイト) =

$$\{1,024 + (2,048 \times \text{ユーザイベントハンドラ数}) \\ \times (86,400 \div \text{統計情報を出力する間隔}) \\ \times (\text{統計情報の取得ファイル数} + 1)\}$$

統計情報を出力する間隔：

サーバ定義の `eads.statistics.interval` パラメタの指定値

統計情報の取得ファイル数：

サーバ定義の `eads.statistics.backupnum` パラメタの指定値

## (f) ユーザログファイルの容量

ユーザログファイルの容量を見積もる計算式を次に示します。

$$\text{ユーザログファイルの容量 (単位: メガバイト)} = \\ \text{ユーザログファイルのファイルサイズ (単位: バイト)} \div 1,024^2 \\ \times \text{ユーザログファイルのファイル数}$$

ユーザログファイルのファイルサイズ (単位: バイト)：

サーバ定義の `eads.logger.user.filesize` パラメタの指定値

ユーザログファイルのファイル数：

サーバ定義の `eads.logger.user.filenum` パラメタの指定値

サーバ定義の `eads.logger.user.filesize` パラメタの指定値はバイト単位に指定するので、メガバイト単位に切り上げて計算します。

## (2) ストアデータファイルの容量を算出する

ストアデータファイルの容量を見積もる計算式を次に示します。

$$\text{ストアデータファイルの容量 (単位: メガバイト)} = \\ (1 \text{ 件当たりの key サイズ}^{\ast 1} + 1 \text{ 件当たりの value サイズ}^{\ast 1} + 100) \\ \times \text{1EADs サーバに格納するデータ件数}^{\ast 2} \\ \times (\text{eztool export コマンド実行時に出力されるストアデータファイルの世代数の上限値} \\ + \text{eztool stop コマンド実行時に出力されるストアデータファイルの世代数の上限値} + 1) \\ \div 1,024^2$$

注※1

シリアライズ後のサイズです。

C 言語の場合は、EADs クライアントで指定した value のサイズに 10 バイトを付加したサイズとなります。

注※2

多重化によってコピーされたデータ件数も含まれます。



eztool export コマンド実行時に出力されるストアデータファイルの世代数の上限値：

クラスタ定義の eads.storeData.generation.count パラメタの指定値

eztool stop コマンド実行時に出力されるストアデータファイルの世代数の上限値：

クラスタ定義の eads.storeData.stop\_command.generation.count パラメタの指定値

## 注意事項

メモリ上にデータがない状態でストアデータファイルに書き出した場合でも、管理情報が格納され、最大4キロバイトを使用します。そのため、ストアデータファイルの使用量は0バイトにはなりません。

なお、ストアデータファイルについては、「7.4.1 ログファイルの種類」を参照してください。

### (3) 1EADs サーバ当たりのログファイルの容量とストアデータファイルの容量の合計値を算出する

「(1)1EADs サーバ当たりのログファイルの容量を算出する」および「(2)ストアデータファイルの容量を算出する」で算出した値を合計します。

ディスクキャッシュ、および2Way キャッシュを使用する場合は、「4.4 キャッシュファイルの容量を見積もる」で見積もったディスク使用量をさらに加算します。

## 4.2.2 EADs クライアントのディスク使用量を見積もる

EADs クライアントのディスク使用量の見積もりについて説明します。

### (1) 1EADs クライアント当たりのログファイルの容量を見積もる

1EADs クライアント当たりのログファイルの容量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{1EADs クライアント当たりのログファイルの容量 (単位: メガバイト)} = \\ & a + b + 70 \text{ メガバイト} \end{aligned}$$

#### 変数の説明

a : EADs クライアントによって出力されるメッセージログファイルの容量

b : EADs クライアントによって出力される通信トレースファイルの容量

なお、EADs クライアントが管理するログファイルについては、「8.4.1 ログファイルの種類」を参照してください。

## (a) EADs クライアントによって出力されるメッセージログファイルの容量

EADs クライアントによって出力されるメッセージログファイルの容量を見積もる計算式を次に示します。

EADs クライアントによって出力されるメッセージログファイルの容量  
(単位：メガバイト) =  
メッセージログファイルのファイルサイズ (単位：バイト)  $\div$  1,024<sup>2</sup>  
 $\times$  メッセージログファイルのファイル数

メッセージログファイルのファイルサイズ (単位：バイト)：

クライアント定義の `eads.logger.message.filesize` パラメタの指定値

メッセージログファイルのファイル数：

クライアント定義の `eads.logger.message.filenum` パラメタの指定値

クライアント定義の `eads.logger.message.filesize` パラメタの指定値はバイト単位に指定するので、メガバイト単位に切り上げて計算します。

## (b) EADs クライアントによって出力される通信トレースファイルの容量

EADs クライアントによって出力される通信トレースファイルの容量を見積もる計算式を次に示します。

EADs クライアントによって出力される通信トレースファイルの容量  
(単位：メガバイト) =  
通信トレースファイルのファイルサイズ (単位：バイト)  $\div$  1,024<sup>2</sup>  
 $\times$  通信トレースファイルのファイル数

通信トレースファイルのファイルサイズ (単位：バイト)：

クライアント定義の `eads.logger.commTrace.filesize` パラメタの指定値

通信トレースファイルのファイル数：

クライアント定義の `eads.logger.commTrace.filenum` パラメタの指定値

クライアント定義の `eads.logger.commTrace.filesize` パラメタの指定値はバイト単位に指定するので、メガバイト単位に切り上げて計算します。

## 4.3 スレッド数とファイルディスクリプタ数を見積もる

EADs を使用するために必要なスレッド数とファイルディスクリプタ数の見積もりについて説明します。

### 4.3.1 スレッド数を見積もる

#### (1) 1EADs サーバプロセス当たりのスレッド数

1EADs サーバプロセス当たりのスレッド数を見積もる計算式を次に示します。

```
1EADs サーバプロセス当たりのスレッド数 =  
キャッシュ数 × (データの多重度 × 2 - 1) × 3 + EADs サーバ数 × 5  
+ キャッシュ単位に行われる処理の最大同時実行スレッド数  
+ EADs サーバへの最大同時接続数 + 40
```

データの多重度：

クラスタ定義の `eads.replication.factor` パラメタの指定値

キャッシュ単位に行われる処理の最大同時実行スレッド数：

サーバ定義の `eads.cluster.controller.cache.max_execute_threads` パラメタの指定値

このパラメタに 0 を指定している場合、データの多重度 - 1 を代入します。

EADs サーバへの最大同時接続数：

サーバ定義の `eads.server.max_connections` パラメタの指定値

#### (2) 1 コマンドプロセス当たりのスレッド数

1 コマンドプロセス当たりのスレッド数を見積もる計算式を次に示します。

```
1 コマンドプロセス当たりのスレッド数 = EADs サーバプロセス数 × 2 + 50
```

### 4.3.2 ファイルディスクリプタ数を見積もる

#### (1) 1EADs サーバプロセス当たりのファイルディスクリプタ数

1EADs サーバプロセス当たりのファイルディスクリプタ数を見積もる計算式を次に示します。

```
1EADs サーバプロセス当たりのファイルディスクリプタ数 =  
EADs サーバ数 × 14 + EADs サーバへの最大同時接続数  
+ ユーザが作成するライブラリ数* + Java のライブラリ数  
+ 統計情報の取得ファイル数 + 35
```

+ディスク上のキャッシュが使用するファイルディスクリプタ数

#### 注※

ユーザが作成するユーザファンクションおよびユーザイベントハンドラの jar ファイル数と、ユーザファンクションおよびユーザイベントハンドラ共通で使用するライブラリ数の合計です。

EADs サーバへの最大同時接続数：

サーバ定義の `eads.server.max_connections` パラメタの指定値

統計情報の取得ファイル数：

サーバ定義の `eads.statistics.backupnum` パラメタの指定値

ディスク上のキャッシュが使用するファイルディスクリプタ数：

ディスクキャッシュ、および 2Way キャッシュを使用する場合は、次の計算式で見積もった値を加算します。

ディスク上のキャッシュが使用するファイルディスクリプタ数=  
5×データの多重度×ディスクを使用するキャッシュ数

データの多重度：

クラスタ定義の `eads.replication.factor` パラメタの指定値

ディスクを使用するキャッシュ数：

ディスクキャッシュ、および 2Way キャッシュのキャッシュ数の合計

`eztool import` コマンド実行時のファイルディスクリプタ数を見積もる計算式を次に示します。

`eztool import` コマンド実行時のファイルディスクリプタ数=  
EADs サーバへの最大同時接続数 × (データの多重度 - 1) × 2  
+ EADs サーバプロセス数 × 100 + ユーザが作成するライブラリ数※  
+ ユーザプログラムに必要なファイルディスクリプタ数

#### 注※

ユーザが作成するユーザファンクションおよびユーザイベントハンドラの jar ファイル数と、ユーザファンクションおよびユーザイベントハンドラ共通で使用するライブラリ数の合計です。

EADs サーバへの最大同時接続数：

サーバ定義の `eads.server.max_connections` パラメタの指定値

データの多重度：

クラスタ定義の `eads.replication.factor` パラメタの指定値

## (2) 1 コマンドプロセス当たりのファイルディスクリプタ数

1 コマンドプロセス当たりのファイルディスクリプタ数を見積もる計算式を次に示します。

1 コマンドプロセス当たりのファイルディスクリプタ数=

EADs サーバプロセス数×10 + 100

## 4.4 キャッシュファイルの容量を見積もる

この節の内容は、ディスクキャッシュ、および2Way キャッシュを使用する場合にお読みください。

ディスクキャッシュ、および2Way キャッシュを使用する場合は、「4.2.1 EADs サーバのディスク使用量を見積もる」で見積もったディスク使用量に、キャッシュごとに次のファイルの容量を見積もって加算します。

- キャッシュデータファイル
- キャッシュインデックスファイル

なお、キャッシュデータファイルおよびキャッシュインデックスファイルについては、「7.7.2 キャッシュファイルの種類と格納先の設定」を参照してください。

### 4.4.1 キャッシュデータファイルのサイズおよび面数を見積もる

ここでは、次の値の見積もりについて説明します。

- キャッシュデータファイルのサイズ
- キャッシュデータファイルの面数

キャッシュデータファイルのサイズおよび面数は次の手順で見積もります。

1. 媒体への書き込みブロックサイズを決定します。
2. キャッシュデータファイルに格納する、レコード1件分のサイズを見積もります。
3. キャッシュデータファイル1面のサイズと実際に格納できるデータ量を見積もります。
4. 2.を基に、システムで管理する有効データの最大量を見積もります。
5. コンパクションの実施間隔を決定します。
6. 2.と5.を基に、コンパクション中に更新されるデータを格納するための容量（更新データ用の容量）を見積もります。
7. 4.と6.から、キャッシュデータファイルに必要な容量を見積もります。
8. 7.から、キャッシュ定義の `eads.cache.disk.filesize` パラメタに指定する、1レンジ当たりのキャッシュデータファイルのサイズ、およびキャッシュ定義の `eads.cache.disk.filenum` パラメタに指定する、1レンジ当たりのキャッシュデータファイルの面数を見積もります。
9. システムの運用前に、コンパクションに掛かる時間などに問題がないことを確認します。

それぞれの説明を次に示します。

## (1) 書き込みブロックサイズを決定する

キャッシュデータファイルに対して、1回で書き込むデータサイズを、書き込みブロックサイズといいます。

書き込みブロックサイズは、キャッシュデータファイルの格納先の媒体に合わせて設定します。設定した値は、キャッシュ定義の `eads.cache.disk.blocksize` パラメタに指定します。

- 格納先の媒体が HDD の場合  
1 キロバイト (デフォルト)
- 格納先の媒体が SSD の場合  
使用する SSD のページサイズ (単位: キロバイト)

## (2) レコード 1 件分のサイズを見積もる

EADs サーバに格納する 1 件分のデータは、key, value, および制御情報を合わせて、レコードという単位で格納します。

レコード 1 件分のサイズを見積もる計算式を次に示します。

```
レコード 1 件分のサイズ (単位: キロバイト) =  
(EADs サーバに格納される key のサイズ (単位: バイト)  
+ EADs サーバに格納される value のサイズ (単位: バイト) + 36) * ÷ 1,024
```

### 注※

この括弧内の値は、「4.4.1(1) 書き込みブロックサイズを決定する」で決定した書き込みブロックサイズ (単位: キロバイト) (キャッシュ定義の `eads.cache.disk.blocksize` パラメタの指定値) の整数倍に切り上げて見積もってください。

EADs サーバに格納される key のサイズ:

EADs サーバに格納される key のサイズを見積もる計算式を次に示します。

```
EADs サーバに格納される key のサイズ (単位: バイト) =  
key の文字数 + 4
```

EADs サーバに格納される value のサイズ:

EADs サーバに格納される value のサイズを見積もる計算式は、AP を作成する言語によって異なります。計算式を次に示します。

- C 言語の場合、または Java 言語でバイト配列を使用する場合

```
EADs サーバに格納される value のサイズ (単位: バイト) =  
API で指定した value のバイト数 + 2
```

- Java 言語でバイト配列以外を使用する場合

```
EADs サーバに格納される value のサイズ (単位: バイト) =  
java.io.ObjectOutputStream クラスで直列化したあとの value のバイト数 + 2
```

## 参考

EADs サーバは、処理の効率化のために、key および value を独自の形式で格納しています。このため、EADs サーバに格納される key および value のサイズは、上記の計算式を使用して見積もる必要があります。

### (3) キャッシュデータファイル 1 面のサイズと格納できるデータ量を見積もる

キャッシュデータファイルの 1 面のサイズは、キャッシュ定義の `eads.cache.disk.filesize` パラメタに指定したサイズです。キャッシュデータファイルにはブロックサイズの 2 倍となる管理情報が含まれるため、実際のデータの格納領域のサイズは、次に示す計算式で算出されるサイズとなります。

$$\begin{aligned} & \text{キャッシュデータファイル 1 面のデータの格納領域 (単位: キロバイト)} = \\ & (\text{キャッシュデータファイル 1 面のサイズ (単位: メガバイト)} \times 1,024 \\ & - (\text{書き込みブロックサイズ (単位: キロバイト)} \times 2) \end{aligned}$$

キャッシュデータファイル 1 面のサイズ

キャッシュ定義の `eads.cache.disk.filesize` パラメタの指定値

書き込みブロックサイズ:

「[4.4.1\(1\) 書き込みブロックサイズを決定する](#)」で決定した値 (キャッシュ定義の `eads.cache.disk.blocksize` パラメタの指定値)

上記の式で算出したデータの格納領域のサイズと「[4.4.1\(2\) レコード 1 件分のサイズを見積もる](#)」で算出したレコードのサイズによって、キャッシュデータファイルに格納できるデータ量を求めることができます。

### (4) 有効データの最大量を見積もる

システムで管理する有効データの最大量を算出します。

ここでの有効データとは、更新や削除で無効となったデータを除く、`get` で取得できるデータのことです。

有効データの最大量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{有効データの最大量 (単位: キロバイト)} = \\ & 1 \text{ レンジに同時に格納される最大の key 数} \times \text{レコード 1 件分のサイズ (単位: キロバイト)} \end{aligned}$$

レコード 1 件分のサイズ:

「[4.4.1\(2\) レコード 1 件分のサイズを見積もる](#)」で算出した値



## 参考

更新や削除で無効となるデータについては、「[10.9 キャッシュデータファイルの使用量を削減する \(キャッシュデータファイルをコンパクションする\)](#)」を参照してください。

## (5) コンパクションの実施間隔を決定する

ディスクキャッシュ、および 2Way キャッシュを使用する場合は、キャッシュデータファイルのコンパクションが必要です。

EADs サーバに格納するデータの量に応じて、「[コンパクションの実施間隔 \(単位：時間\)](#)」を決定します。

コンパクションの処理については、「[10.9 キャッシュデータファイルの使用量を削減する \(キャッシュデータファイルをコンパクションする\)](#)」を参照してください。

## (6) 更新データ用の容量を見積もる

キャッシュデータファイルには常に追記でデータが格納されるため、データを更新すると、キャッシュデータファイル内のデータ量が増加します。そのため、キャッシュデータファイルには、「[コンパクションの実施間隔 \(単位：時間\)](#)」および「[1 回のコンパクションに掛かる時間 \(単位：時間\)](#)」の間に増加するデータ分の容量が必要となります。

1 回のコンパクションを実行している間に更新されるデータ (更新データ) の容量を見積もります。更新データ用の容量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{更新データ用の容量 (単位：キロバイト)} = \\ & \text{LEADs サーバの 1 時間当たりの更新回数} \times \text{レコード 1 件分のサイズ (単位：キロバイト)} \\ & \times (\text{コンパクションの実施間隔 (単位：時間)} + 1 \text{ 回のコンパクションに掛かる時間 (単位：時間)}) \end{aligned}$$

LEADs サーバの 1 時間当たりの更新回数：

put, create, update, および replace の実行回数

レコード 1 件分のサイズ：

「[4.4.1\(2\) レコード 1 件分のサイズを見積もる](#)」で算出した値

コンパクションの実施間隔：

「[4.4.1\(5\) コンパクションの実施間隔を決定する](#)」で決定した値

1 回のコンパクションに掛かる時間：

この時間はシステムによって異なります。システムの運用前にコンパクションにかかる時間を確認し、その値を代入してください。

## (7) キャッシュデータファイルに必要な容量を見積もる

キャッシュデータファイルに必要な容量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{キャッシュデータファイルに必要な容量 (単位: メガバイト)} = \\ & (\text{有効データの最大量 (単位: キロバイト)} \times 2 \\ & + \text{更新データ用の容量 (単位: キロバイト)}) \div 1,024 \end{aligned}$$

有効データの最大量:

「4.4.1(4) 有効データの最大量を見積もる」で算出した値

更新データ用の容量:

「4.4.1(6) 更新データ用の容量を見積もる」で算出した値

## (8) キャッシュデータファイルのサイズおよび面数を設計する

「4.4.1(7) キャッシュデータファイルに必要な容量を見積もる」で算出した、キャッシュデータファイルに必要な容量を基に、次の式を満たすように、1レンジ当たりのキャッシュデータファイルのサイズ、およびキャッシュデータファイルの面数を設計します。

$$\begin{aligned} & \text{キャッシュデータファイルに必要な容量 (単位: メガバイト)} \leq \\ & 1 \text{レンジ当たりのキャッシュデータファイルのサイズ (単位: メガバイト)} \times \text{キャッシュデータファイルの面数}^* \end{aligned}$$

注※

キャッシュデータファイルの容量の拡張が考えられる場合は、その分の面数も考慮して決定してください。

1レンジ当たりのキャッシュデータファイルのサイズは、キャッシュ定義の `eads.cache.disk.filesize` パラメタに指定します。

1レンジ当たりのキャッシュデータファイルの面数を見積もる計算式を次に示します。

$$\begin{aligned} & 1 \text{レンジ当たりのキャッシュデータファイルの面数} = \\ & \text{キャッシュデータファイルの面数} + 1 \end{aligned}$$

算出した値は、キャッシュ定義の `eads.cache.disk.fileenum` パラメタに指定します。

## (9) コンパクションの運用計画を確認する

システムの運用前に、コンパクションに掛かる時間、およびに確保される容量に問題がないことを確かめてください。

次の計算式を満たしていない場合、コンパクションがデータの更新に間に合わないため、容量不足によるエラーが発生するおそれがあります。

$$\text{更新データ用の容量} < 1 \text{回のコンパクションで確保される容量}$$

## 注意事項

上記の計算式を満たしていない原因がハードウェアの性能不足の場合、コンパクションの回数や実施間隔のチューニングだけでなく、ハードウェアの更新やホストの追加も考慮した対策を検討してください。

### 4.4.2 キャッシュインデックスファイルのサイズを見積もる

キャッシュインデックスファイル 1 ファイル分のサイズを見積もる計算式を次に示します。

$$\begin{aligned} \text{キャッシュインデックスファイルのサイズ (単位: バイト)} = \\ 48 + 4 \times \{ (1 \text{ レンジ当たりのキャッシュデータファイルのサイズ (単位: メガバイト)} \times 1,024 \\ \div \text{書き込みブロックサイズ (単位: キロバイト)}) - 2 \} \end{aligned}$$

1 レンジ当たりのキャッシュデータファイルのサイズ:

キャッシュ定義の `eads.cache.disk.filesize` パラメタの指定値

書き込みブロックサイズ:

「[4.4.1\(1\) 書き込みブロックサイズを決定する](#)」で決定した値 (キャッシュ定義の `eads.cache.disk.blocksize` パラメタの指定値)

# 5

## インストールおよびセットアップを行う (EADs サーバ)

この章では、EADs サーバの構築方法について説明します。

## 5.1 EADs サーバをインストールする

EADs サーバをインストールする手順を説明します。

インストールは root (スーパーユーザ) で行います。

### 5.1.1 インストール前にすること

EADs サーバをインストールする前に、root (スーパーユーザ) で次のことを行います。

#### (1) hosts ファイルを設定する

hosts ファイルで、IP アドレスとホスト名を対応づけます。

##### 注意事項

設定に誤りがあると、EADs が正しく動作しないおそれがあります。

### 5.1.2 インストール手順

##### 注意事項

すでに別バージョンの EADs サーバがインストールされている場合、上書きインストールされます (ただし、03-00 以降のバージョンに限ります)。また、EADs サーバプロセスが起動していないことを確認してからインストールしてください。

インストール手順について説明します。

1. インストールするマシンに root 権限でログインします。
2. インストール CD-ROM を CD-ROM ドライブにセットします。
3. OS の mount コマンドで、CD-ROM ファイルシステムをマウントします。

mount コマンドの実行例を次に示します。

下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。

```
mount -r -o mode=0544 /dev/cdrom /mnt/cdrom
```

4. 日立 PP インストーラの setup コマンドで、セットアッププログラムを起動します。

setup コマンドの実行例を次に示します。

下線部には、CD-ROM ファイルシステムのマウントディレクトリ名を指定します。

5. インストールおよびセットアップを行う (EADs サーバ)

```
/mnt/cdrom/X64LIN/setup /mnt/cdrom
```

5. 日立 PP インストーラのメインメニューで、[I] キーを押します。

プログラムの選択画面が表示されます。

6. プログラムの選択画面で、インストールしたいプログラムにカーソルを移動させ、[スペース] キーを押します。

インストールしたいプログラムを選択してください。選択したプログラムの左側には「@」が表示されます。

7. 選択したプログラムの左側に「@」が表示されていることを確認し、[I] キーを押します。

いちばん下の行に、次のメッセージが表示されます。

```
Install PP? (y: install, n: cancel)==>
```

8. インストールするかどうかを確認するメッセージが出力されたら、[Y] キーを押します。

インストールが開始されます。[N] キーを押すと、インストールが中止されてプログラムの選択画面に戻ります。

9. インストール終了を示すメッセージが出力されたら、[Q] キーを押します。

メインメニューに戻ります。

10. メインメニューで、[Q] キーを押します。

これで、インストールは完了です。

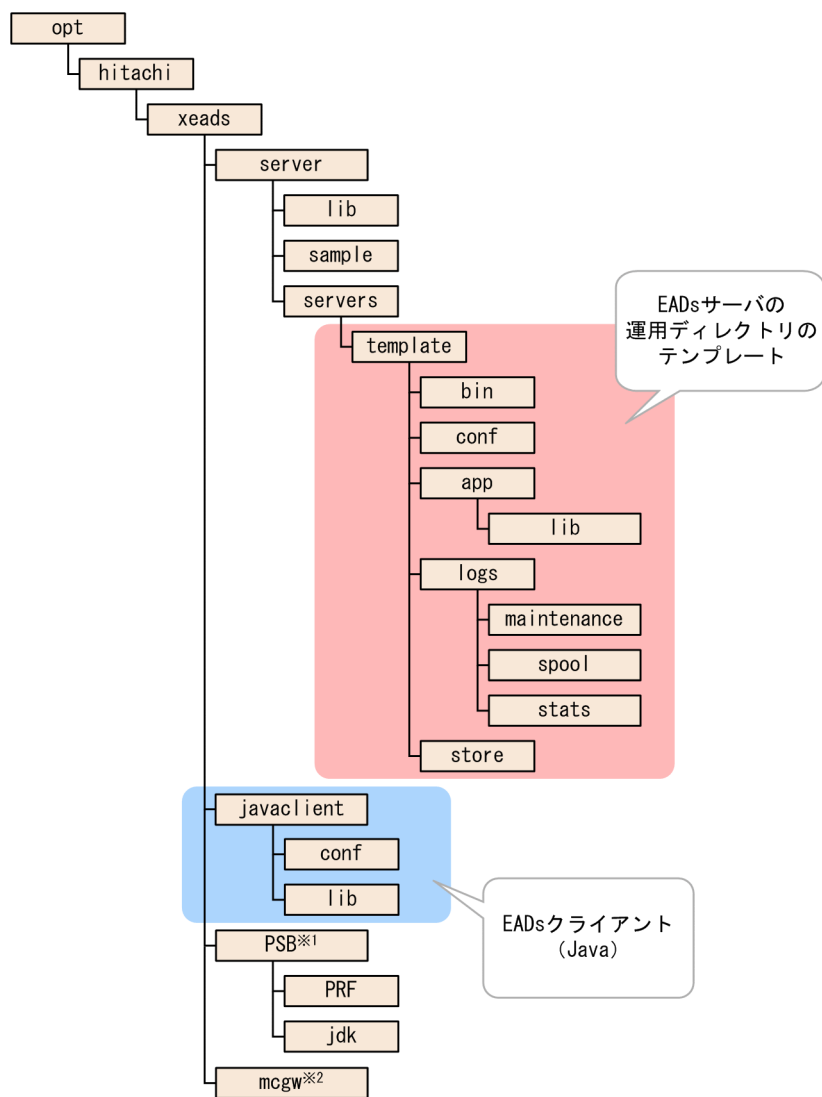
### 5.1.3 インストール後にすること

EADs サーバをインストールしたあと、root（スーパーユーザ）で次のことを行います。

#### (1) ディレクトリ構成を確認する

インストール後、EADs サーバのディレクトリ構成を確認します。

図 5-1 EADs サーバのディレクトリ構成



注※1  
PRFおよびjdk以外のディレクトリは、EADsでは使用しません。

注※2  
通常の運用では使用しません。

図に示した EADs サーバのディレクトリ構成について説明します。

ディレクトリ	説明
/opt/hitachi/xeads/server/	EADs サーバのインストールディレクトリです。
lib	EADs サーバを実行するためのライブラリファイルが格納されています。 ライブラリファイルは編集できません。
sample	<code>eztool execfunc</code> コマンドで実行できるユーザ関クションのサンプルプログラムが格納されています。
servers	EADs サーバの運用ディレクトリ※を配置します。
template	運用ディレクトリ※のテンプレートです。

## 5. インストールおよびセットアップを行う (EADs サーバ)

ディレクトリ	説明
bin	EADs サーバを運用操作するコマンドが格納されています。 コマンドは編集できません。
conf	次に示す定義ファイルが格納されています。 <ul style="list-style-type: none"> <li>• サーバ定義ファイル</li> <li>• クラスタ定義ファイル</li> <li>• 起動設定ファイル</li> <li>• コマンド定義ファイル</li> <li>• キャッシュ定義ファイル</li> </ul>
app	ファンクション定義ファイル、および jar ファイル（ユーザファンクションまたはユーザイベントハンドラ）の格納先です。
lib	ユーザファンクション、およびユーザイベントハンドラを実行するためのライブラリファイルの格納先です。
logs	ログファイルの出力先（デフォルト）です。
maintenance	保守ログファイル（システムが使用するファイル）の出力先（デフォルト）です。
spool	PRF トレースファイルの出力先（デフォルト）です。
stats	統計情報ファイルの出力先（デフォルト）です。
store	ストアデータファイル、キャッシュ情報ファイルおよびキャッシュインデックスファイルの出力先（デフォルト）です。
/opt/hitachi/xeads/javaclient/	EADs クライアント（Java）のインストールディレクトリです。
conf	クライアント定義ファイルが格納されています。
lib	ライブラリファイルが格納されています。
/opt/hitachi/xeads/PSB	EADs で使用する PRF および JDK の格納用ディレクトリです。
PRF	パフォーマンストレーサ（PRF デーモン）のインストールディレクトリです。
jdk	JDK のインストールディレクトリです。

#### 注※

運用ディレクトリとは、EADs サーバの運用で使用するディレクトリのことです。運用ディレクトリには、EADs サーバの稼働に必要な定義ファイルや稼働時に出力されるログなどが格納されます。



## 5.2 EADs サーバをセットアップする

インストール以降の作業は OS ユーザ（システム運用管理者）が行います。

### 5.2.1 運用ディレクトリを作成する

EADs サーバの運用で使用するディレクトリを運用ディレクトリといいます。運用ディレクトリには、EADs サーバの稼働に必要な定義ファイルや稼働時に出力されるログなどが格納されます。

EADs サーバを運用する OS ユーザ（システム運用管理者）でログインし、運用ディレクトリのテンプレートである `/opt/hitachi/xeads/server/servers/template` ディレクトリを、`/opt/hitachi/xeads/server/servers` ディレクトリ下に別名でコピーします。運用ディレクトリ名には、最大 32 文字の半角英数字（0～9, A～Z, a～z）、およびアンダースコア（`_`）が指定できます。

```
cp -rf /opt/hitachi/xeads/server/servers/template /opt/hitachi/xeads/server/servers/<任意の運用ディレクトリ名>
```

なお、このマニュアルで表記する「EADs サーバ名」とは、運用ディレクトリ名を指します。

EADs サーバごとに OS ユーザ（システム運用管理者）を割り当てることができます。運用ディレクトリのアクセス権は必要に応じて変更してください。

#### 注意事項

すでに存在する運用ディレクトリを使用する場合、次に示すファイルとディレクトリを別のディレクトリに移動するか、または削除してください。

- サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリ
- コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリ

また、すでに存在する運用ディレクトリを使用する場合、`<運用ディレクトリ>/bin` 下のコマンドは、`/opt/hitachi/xeads/server/servers/template/bin` ディレクトリ下のコマンドに、インストール時に上書きされます。

#### 参考

`/opt/hitachi/xeads/server/servers` ディレクトリ下に作成したシンボリックリンクを運用ディレクトリとして使用できます。

運用ディレクトリ直下に出力されるコアダンプを別のディスクに出力したい場合などに使用します。ただし、実体側の運用ディレクトリではコマンドを実行できないため、注意してください。

## 5.2.2 定義ファイルを編集する

「7. 環境依存パラメタを設計する (EADs サーバ)」で設計したパラメタの指定値を、<運用ディレクトリ>/conf 下の次の定義ファイルに設定します。

- サーバ定義ファイル (eads\_server.properties)
- クラスタ定義ファイル (eads\_cluster.properties)
- 起動設定ファイル (eads\_start.properties)
- コマンド定義ファイル (eads\_management.properties)
- キャッシュ定義ファイル (eads\_cache.<キャッシュ名>.properties) ※

注※ メモリキャッシュを作成する場合、キャッシュ定義ファイルの作成は任意です。

それぞれの定義ファイルに指定できるパラメタの一覧については、「23.1 パラメタ一覧」を参照してください。

チューニングパラメタについては、「9. チューニングパラメタを設計する」を参照して設計します。設計した値を各定義ファイルに設定してください。

ユーザファンクションを使用する場合、ファンクション定義ファイルを任意で作成します。ファンクション定義ファイルは、実行環境の EADs サーバが停止した状態で、ユーザファンクションと一緒に、<運用ディレクトリ>/app 下にコピーします。詳細については、「18.4 ファンクション定義ファイルを作成する (任意)」および「18.9 実行環境に配布する」を参照してください。

### 注意事項

- クラスタ定義ファイルの定義内容は、クラスタを構成する全 EADs サーバで同一になるようにしてください。同一でないとクラスタが構築できないため、EADs サーバは起動しません。
- キャッシュ定義ファイルの次のパラメタの定義内容は、クラスタを構成する全 EADs サーバで同じになるようにしてください。同じでないとキャッシュを作成できません。
  - eads.cache.type
  - eads.cache.disk.filesize
  - eads.cache.disk.filenum
  - eads.cache.disk.blocksize

## 5.2.3 AP を配置する

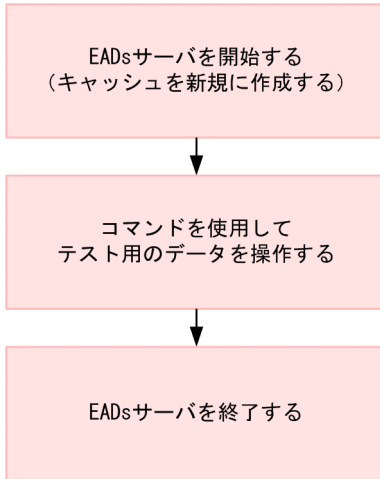
開発環境で作成した AP を，実行環境に配布します。

ユーザ機能の実行環境への配布については，「[18.9 実行環境に配布する](#)」を参照してください。ユーザイベントハンドラの実行環境への配布については，「[19.8 実行環境に配布する](#)」を参照してください。

## 5.3 テストを行う

構築後の実行環境が正常に動作するかどうかテストします。テストにはコマンドを使用します。テストの流れについて、次の図に示します。

図 5-2 テストの流れ



### 5.3.1 EADs サーバを開始する (キャッシュを新規に作成する)

EADs サーバを開始して、キャッシュを新規に作成します。

手順については、「10.2 EADs サーバを開始する (キャッシュを新規に作成する場合)」を参照してください。

### 5.3.2 コマンドを使用してテスト用のデータを操作する

キャッシュが作成できたら、コマンドを使用してテスト用のデータを操作します。

#### (1) テスト用のデータを格納する

`eztool put` コマンドを実行して、テスト用のデータを格納します。

```
eztool put <キャッシュ名> <key> <value>
```

#### コマンド実行例

```
$ eztool put cache1 key1 value1
KDEA08001-I The command will now start. (subcommand = put, parameter = [put, cache1, key1, value1])
KDEA08002-I The command will now end.
```

## (2) 正常に動作したかどうか確認する

eztool status -v コマンドを実行して、EADs サーバが保持する key の数を表示します。

```
eztool status -v
```

### コマンド実行例

```
$ eztool status -v
KDEA08001-I The command will now start. (subcommand = status, parameter = [status -v])

Cluster Health: AVAILABLE
TotalCount: 1
OnlineCount: 1
OfflineCount: 0
StandbyCount: 0

ID IP_Address ServerName ClientPort ServerPort ManagePort Position Cluster State Operation Lock KeyCount UsedCache Version
1 XX.XXX.XXX.168 server01 24600 24633 24620 1288490189 online running none unlock 1 0% ( 0 / 92 MB ) 03-60-00

KDEA08002-I The command will now end.
$
```

KeyCount 欄に、格納された key の数が表示されます。

## (3) 格納したデータを取得する

eztool get コマンドを実行して、格納した value を取得します。

```
eztool get <キャッシュ名> <key>
```

### コマンド実行例

```
$ eztool get cache1 key1
KDEA08001-I The command will now start. (subcommand = get, parameter = [get, cache1, key1])

Value: value1
ValueSize: 12

KDEA08002-I The command will now end.
```

## (4) 格納したデータを削除する

eztool remove コマンドを実行して、格納した value を削除します。

```
eztool remove <キャッシュ名> <key>
```

### コマンド実行例

```
$ eztool remove cache1 key1
KDEA08001-I The command will now start. (subcommand = remove, parameter = [remove, cache1, key1])
KDEA08002-I The command will now end.
```

### 5.3.3 EADs サーバを終了する

EADs サーバを終了します。

EADs サーバを終了すると、メモリ上のデータはすべて破棄されます。

手順については、「[10.4 EADs サーバを終了する \(メモリ上のデータを破棄する場合\)](#)」を参照してください。

## 5.4 EADs サーバをアンセットアップする

---

EADs サーバをアンセットアップする場合は、次に示す OS のコマンドを実行して、セットアップした運用ディレクトリを削除します。

```
rm -rf /opt/hitachi/xeads/server/servers/<運用ディレクトリ名>
```

これで、インストール時の状態に戻すことができます。

## 5.5 EADs サーバをアンインストールする

### 注意事項

EADs サーバプロセスが起動していないことを確認してからアンインストールしてください。

EADs サーバをアンインストールする手順を説明します。

なお、EADs サーバをアンインストールすると運用ディレクトリ以外は、すべて削除されます。そのため、製品のディレクトリ下にユーザ固有のファイルを格納しないでください。

1. EADs サーバをアンインストールするマシンに root 権限でログインします。
2. OS のコマンドで、セットアッププログラムを起動します。

```
/etc/hitachi_x64setup
```

3. 日立 PP インストーラのメインメニューで、[D] キーを押します。

現在インストールされているプログラムの一覧が表示されます。

4. アンインストールしたいプログラムにカーソルを移動させ、[スペース] キーを押します。

選択したプログラムの左側には「@」が表示されます。

5. [D] キーを押します。

アンインストールが開始されます。アンインストールが終了すると、次のメッセージが表示されます。

```
Delete procedure completed.
```

6. アンインストール終了を示すメッセージが出力されたら、[Q] キーを押します。

メインメニューに戻ります。

7. メインメニューで、[Q] キーを押します。

これで、アンインストール作業は完了です。



# 6

## インストールおよびセットアップを行う (EADs クライアント)

この章では、EADs クライアントの構築方法について説明します。

## 6.1 EADs クライアントをインストールする

EADs クライアントをインストールする手順を説明します。

インストールは root (スーパーユーザ) で行います。

使用する AP の言語によって、インストールする手順が異なります。

### Java 言語の場合

まず EADs サーバとして使用するマシンに、EADs サーバをインストールします。EADs クライアント (Java) は EADs サーバのインストールディレクトリに格納されています。

EADs サーバのインストール手順については、「[5.1.2 インストール手順](#)」を参照してください。

EADs サーバをインストール後、EADs クライアントをインストールする別のマシンに次のディレクトリをコピーします。

```
/opt/hitachi/xeads/javaclient
```

### C 言語の場合

次のプログラムプロダクトをインストールしてください。

- uCosminexus Elastic Application Data store Client for C

インストール手順については、「[5.1.2 インストール手順](#)」を参照してください。

### 注意事項

Developer がインストールされている環境に、EADs クライアントをインストールしないでください。

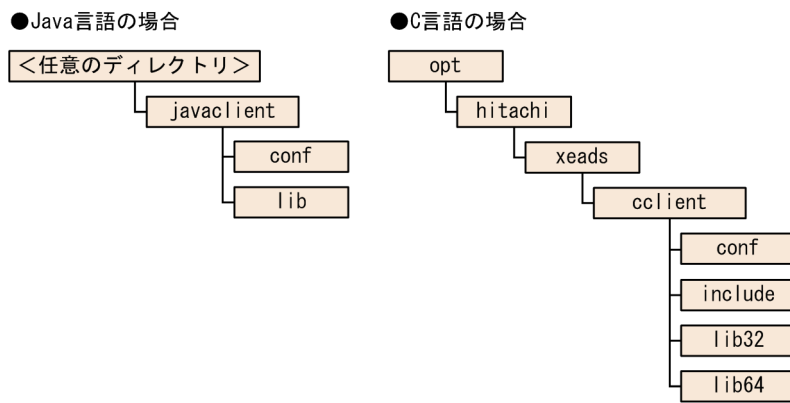
### 6.1.1 インストール後にすること

EADs クライアントをインストールしたあと、次のことを行います。

#### (1) ディレクトリ構成を確認する

インストール後、EADs クライアントのディレクトリ構成を確認します。

図 6-1 EADs クライアントのディレクトリ構成



図に示した EADs クライアントのディレクトリ構成について説明します。

ディレクトリ	説明
<任意のディレクトリ>/javaclient/	EADs クライアント (Java) のコピー先ディレクトリです。
conf	クライアント定義ファイルが格納されています。
lib	ライブラリファイルが格納されています。
/opt/hitachi/xeads/cclient/	EADs クライアント (C) のインストールディレクトリです。
conf	クライアント定義ファイルが格納されています。
include	ヘッダファイルが格納されています。
lib32	32 ビット版のライブラリファイルが格納されています。
lib64	64 ビット版のライブラリファイルが格納されています。

## (2) ライブラリを設定する

ライブラリを設定します。

### (a) Java 言語の場合

次に示すライブラリを、クラスパスに設定します。

- /opt/hitachi/xeads/javaclient/lib/eads-client.jar
- /opt/hitachi/xeads/javaclient/lib/eads-common.jar
- /opt/hitachi/xeads/javaclient/lib/hntrlib2-eads-j.jar

なお、J2EE サーバ (uCosminexus Application Server) 上で実行する場合は、次に示すライブラリを AP (WAR ファイル (WEB-INF/lib ディレクトリ)) に取り込む必要があります。J2EE サーバでは、Servlet または JSP から実行できます。

- /opt/hitachi/xeads/javaclient/lib/eads-client.jar

- /opt/hitachi/xeads/javaclient/lib/eads-common.jar
- /opt/hitachi/xeads/javaclient/lib/hntrlib2-eads-j.jar

## (b) C 言語の場合

次に示すライブラリを、環境変数 LD\_LIBRARY\_PATH に設定します。

- 32 ビット版の場合  
/opt/hitachi/xeads/cclient/lib32
- 64 ビット版の場合  
/opt/hitachi/xeads/cclient/lib64

## 6.2 EADs クライアントをセットアップする

AP の配置と、クライアント定義ファイルの編集方法について説明します。

### 6.2.1 AP を配置する

AP を任意のディレクトリに配置します。

#### 注意事項

J2EE サーバ (uCosminexus Application Server) 上で実行する場合は、デプロイします。この場合、セキュリティの設定を変更して、SecurityManager を解除する必要があります。

AP の作成方法については、「[第 4 編 AP 開発](#)」を参照してください。

### 6.2.2 クライアント定義ファイルを編集する

クライアント定義ファイルを編集する手順を説明します。

1. 次に示すディレクトリ下のクライアント定義ファイルを、AP で指定したディレクトリにコピーします (ファイル名も AP で指定したファイル名に合わせます)。

- Java 言語の場合  
/opt/hitachi/xeads/javaclient/conf/eads\_sample\_client.properties
- C 言語の場合  
/opt/hitachi/xeads/cclient/conf/eads\_sample\_client.properties

2. クライアント定義ファイルを編集します。

「[8. 環境依存パラメタを設計する \(EADs クライアント\)](#)」で設計した値をクライアント定義ファイルに設定します。

クライアント定義ファイルに指定できるパラメタの一覧については、「[23.1.2\(1\) クライアント定義ファイル](#)」を参照してください。

チューニングパラメタについては、「[9. チューニングパラメタを設計する](#)」を参照して設計します。設計した値をクライアント定義ファイルに設定してください。

## 6.3 EADs クライアントをアンインストールする

---

EADs クライアントをアンインストールする手順を次に示します。使用する AP の言語によって、アンインストールの手順が異なります。

### Java 言語の場合

EADs サーバの次のディレクトリをコピーしたディレクトリを削除してください。

```
/opt/hitachi/xeads/javaclient
```

### C 言語の場合

次のプログラムプロダクトをアンインストールしてください。

- uCosminexus Elastic Application Data store Client for C

アンインストールする手順については、「[5.5 EADs サーバをアンインストールする](#)」を参照してください。参照する際は、「EADs サーバ」を「EADs クライアント」に読み替えてください。

# 7

## 環境依存パラメタを設計する (EADs サーバ)

この章では、EADs サーバの、環境依存パラメタの設計指針について説明します。

## 7.1 定義ファイルの種類 (EADs サーバ)

EADs サーバで使用する定義ファイルの種類を次の表に示します。

なお、この章では、環境依存パラメタについてだけ説明しています。それぞれの定義ファイルに指定できるパラメタの一覧については、「[23.1 パラメタ一覧](#)」を参照してください。

表 7-1 定義ファイルの種類 (EADs サーバ)

項番	定義ファイル	ファイル名	説明
1	サーバ定義ファイル	eads_server.properties	EADs サーバの実行環境を定義します。
2	クラスタ定義ファイル	eads_cluster.properties	クラスタの構成およびデータの多重度を定義します。
3	起動設定ファイル	eads_start.properties	EADs サーバ、および PRF デーモンの起動設定を定義します。
4	コマンド定義ファイル	eads_management.properties	コマンド実行時の設定を定義します。
5	キャッシュ定義ファイル	eads_cache.<キャッシュ名>.properties	キャッシュの設定を定義します。キャッシュごとに作成します。メモリキャッシュの場合、作成は任意です。ディスクキャッシュまたは 2Way キャッシュの場合、必ず作成してください。
6	ファンクション定義ファイル	ファイル名は任意です。詳細については、「 <a href="#">18.4 ファンクション定義ファイルを作成する (任意)</a> 」を参照してください。	ユーザファンクション実行時の設定を定義します。作成は任意です。

### 注意事項

- クラスタ定義ファイルの定義内容は、クラスタを構成する全 EADs サーバで同じになるようにしてください。同じでないとクラスタが構築できないため、EADs サーバは起動しません。
- キャッシュ定義ファイルの次のパラメタの定義内容は、クラスタを構成する全 EADs サーバで同じになるようにしてください。同じでないとキャッシュを作成できません。
  - `eads.cache.type`
  - `eads.cache.disk.filesize`
  - `eads.cache.disk.filenum`
  - `eads.cache.disk.blocksize`
- その他の定義ファイルの定義内容については、同一にする必要はありませんが、同一にすることを推奨します (IP アドレスやポート番号は除きます)。



## 7.2 定義ファイルの形式

---

次のようにパラメタを指定します。

```
<パラメタ名>=<指定値>
```

### 指定方法

- 改行までが値になります。
- #で始まる行はコメントと見なされます。
- 空行は無視されます。
- 値の後ろには、コメントなどの文字列は追加できません。追加した場合、不正な値と解釈されます。
- 同じパラメタを指定した場合、後から記述したパラメタの指定が有効になります。例えば、次のように指定されている場合、`false` が有効になります。

```
eads.prf.enable=true  
eads.prf.enable=false
```

## 7.3 通信に依存するパラメタの設計

通信に依存するパラメタを設計します。

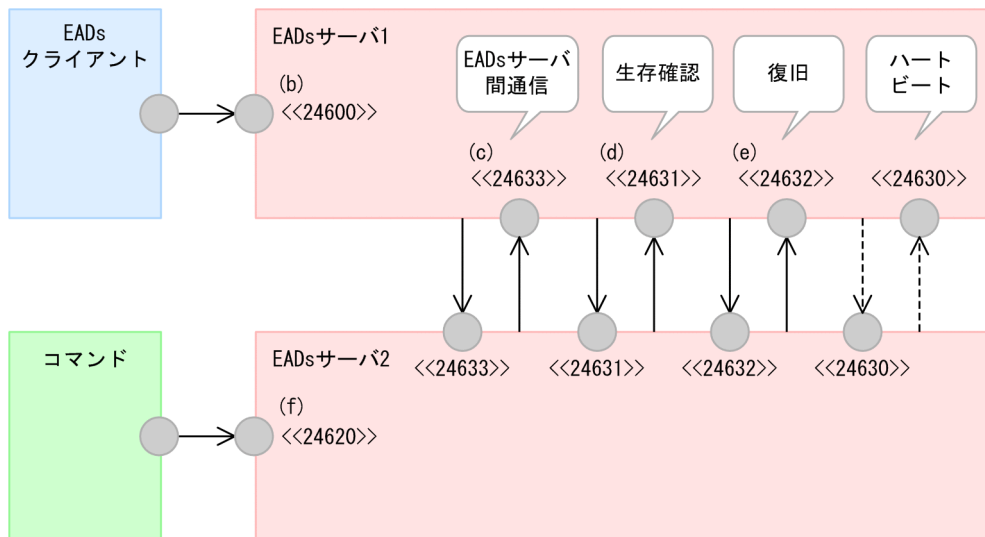
### 7.3.1 IP アドレスまたはホスト名、およびポート番号の設定

EADs サーバが EADs クライアントおよび他 EADs サーバとの通信で使用する IP アドレス (IPv4) またはホスト名、およびポート番号を指定します。

通信処理の仕組みについては、「[2.2 EADs の通信処理の仕組み](#)」を参照してください。

EADs クライアントおよび EADs サーバとの通信で使用する通信ポートを次の図に示します。

図 7-1 EADs クライアントおよび EADs サーバとの通信で使用する通信ポート



(凡例)

⇔ : TCP/IP通信

⇔ : UDP/IP通信

● : 通信ポート

<<XXXXX>> : ポート番号 (デフォルト)

図中のポート番号に付いているアルファベットは、「[7.3.2 通信に依存するパラメタ](#)」の説明と次のように対応しています。

(b) : 「[7.3.2\(1\)\(b\) eads.server.port](#)」

(c) : 「[7.3.2\(1\)\(c\) eads.consensus.message.port](#)」

(d) : 「[7.3.2\(1\)\(d\) eads.cluster.failureDetector.port](#)」

(e) : 「7.3.2(1)(e) eads.restore.port」

(f) : 「7.3.2(1)(f) eads.management.port」

EADs サーバ間のハートビートの送信に使用するポート番号については、「7.5.3 クラスタ構成に依存するパラメタ」の「7.5.3(1)(e) eads.cluster.heartbeat.port」を参照してください。

## 参考

ハートビートの送信はマルチキャスト通信を使用するため、ネットワークの設計時に注意が必要です。クラスタ定義の `eads.cluster.heartbeat.address` パラメタには、マルチキャストアドレスを指定する必要があります。

ポート番号を指定する場合は、ほかのアプリケーションとの重複を避けるため、OS が任意に割り当てるポート番号は使用しないでください。OS が任意に割り当てるポート番号は、OS の種別やバージョンによって異なります。詳細については、ご使用の OS のマニュアルを参照してください。

## 7.3.2 通信に依存するパラメタ

通信に依存するパラメタの一覧を次の表に示します。

なお、必ず指定するパラメタを太字で表記しています。

表 7-2 通信に依存するパラメタ一覧 (EADs サーバ)

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	サーバ定義ファイル	<b>eads.server.address</b>	IP アドレスまたはホスト名	なし
2		<code>eads.server.port</code>	ポート番号	24600
3		<code>eads.consensus.message.port</code>	ポート番号	24633
4		<code>eads.cluster.failureDetector.port</code>	ポート番号	24631
5		<code>eads.restore.port</code>	ポート番号	24632
6		<code>eads.management.port</code>	ポート番号	24620

### (1) サーバ定義ファイル

#### (a) **eads.server.address**

EADs サーバの IP アドレスまたはホスト名を指定します。

ホスト名で指定する場合は、ホスト名から IP アドレスが一意に識別できるホスト名を指定してください。

### **(b) eads.server.port**

EADs クライアントとの通信に使用する EADs サーバのポート番号を指定します。

### **(c) eads.consensus.message.port**

EADs サーバ間の通信に使用するポート番号を指定します。

### **(d) eads.cluster.failureDetector.port**

EADs サーバ間の生存確認に使用するポート番号を指定します。

### **(e) eads.restore.port**

EADs サーバの復旧に使用するポート番号を指定します。

### **(f) eads.management.port**

コマンドが使用するポート番号を指定します。

## 7.4 ログファイルに依存するパラメタの設計

ログファイルに依存するパラメタを設計します。

### 7.4.1 ログファイルの種類

EADs サーバが管理するログファイルの種類を次の表に示します。

表 7-3 ログファイルの種類 (EADs サーバ)

項番	ログファイル	説明
1	メッセージログファイル	動作確認やエラー監視用のメッセージログを出力するファイルです。 メッセージログは、EADs サーバまたはコマンドによって出力されます。
2	例外ログファイル	障害原因の調査で必要となるスタックトレースを出力するファイルです。
3	ユーザログファイル	ユーザファンクションおよびユーザイベントハンドラの実行情報を出力するファイルです。
4	ユーザ例外ログファイル	ユーザファンクションおよびユーザイベントハンドラで発生した例外のスタックトレースを出力するファイルです。
5	キャッシュファイル操作ログファイル	キャッシュデータファイルの使用状況、およびコンパクションの状況を確認、監視するための情報を出力するファイルです。
6	保守ログファイル	システムが使用するファイルです。 ユーザが設定するパラメタはありません。
7	分散保守ログファイル	システムが使用するファイルです。 ユーザが設定するパラメタはありません。
8	PRF トレースファイル	性能解析および問題解析用の PRF トレース情報を出力するファイルです。
9	統計情報ファイル	チューニングの見直し、性能測定、リソース見積もりなどに使用する統計情報を出力するファイルです。
10	Java ログファイル	JavaVM の GC (ガーベージコレクション) やメモリに関する情報を出力するファイルです。
11	スレッドダンプ	Java プロセス内で動作しているスレッドの情報を出力するファイルです。
12	起動ログファイル	<code>ezstart</code> コマンドによる EADs サーバ起動時のログを出力するファイルです。 ユーザが設定するパラメタはありません。

### 7.4.2 ファイルの出力先の設定

ログファイルの出力先は変更できます。ファイルの出力先を次の表に示します。

表 7-4 ファイルの出力先 (EADs サーバ)

項番	ログファイル種別	出力先	ファイル名	
1	EADs サーバによって出力されるメッセージログファイル (メッセージログの情報)	サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_server_message[n].log</code>
			Shift	<code>eads_server_message.log</code>
2	EADs サーバによって出力される例外ログファイル (例外ログ情報)	サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_server_exception[n].log</code>
			Shift	<code>eads_server_exception.log</code>
3	ユーザログファイル	サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_user_message[n].log</code>
			Shift	<code>eads_user_message.log</code>
4	ユーザ例外ログファイル	サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_user_exception[n].log</code>
			Shift	<code>eads_user_exception.log</code>
5	キャッシュファイル操作ログファイル	サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_server_cache[n].log</code>
			Shift	<code>eads_server_cache.log</code>
6	EADs サーバによって出力される保守ログファイル (保守ログ情報)	<サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ>/maintenance	<code>eads_server_maintenance[n].log</code>	
7	EADs サーバによって出力される分散保守ログファイル (保守ログ情報)	<サーバ定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ>/maintenance	<code>eads_dist_maintenance[n].log</code>	
8	コマンド実行時に出力されるメッセージログファイル (メッセージログの情報)	コマンド定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_management_message[n].log</code>
			Shift	<code>eads_management_message.log</code>
9	コマンド実行時に出力される例外ログファイル (例外ログ情報)	コマンド定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_management_exception[n].log</code>
			Shift	<code>eads_management_exception.log</code>
10	コマンド実行時に出力される保守ログファイル (保守ログ情報)	<コマンド定義の <code>eads.logger.dir</code> パラメータに指定したディレクトリ>/maintenance	<code>eads_management_maintenance[n].log</code>	
11	PRF トレースファイル	<サーバ定義の <code>eads.logger.dir</code> パラメ	<code>prf_[nnn]</code>	

項番	ログファイル種別	出力先	ファイル名
		タに指定したディレクトリ>/spool/utt/prf/REQ_PRF_ID/dcopltrc	
12	統計情報ファイル (EADs サーバ)	<サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/stats	<ul style="list-style-type: none"> <li>• eads_stats.csv</li> <li>• eads_stats_[n].csv<sup>*1</sup></li> </ul>
13	統計情報ファイル (キャッシュ)	<サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/stats	<ul style="list-style-type: none"> <li>• eads_cache_stats.csv</li> <li>• eads_cache_stats_[n].csv<sup>*1</sup></li> </ul>
14	統計情報ファイル (ユーザファンクション)	<サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/stats	<ul style="list-style-type: none"> <li>• eads_function_stats.csv</li> <li>• eads_function_stats_[n].csv<sup>*1</sup></li> </ul>
15	統計情報ファイル (ユーザイベントハンドラ)	<サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/stats	<ul style="list-style-type: none"> <li>• eads_event_stats.csv</li> <li>• eads_event_stats_[n].csv<sup>*1</sup></li> </ul>
16	統計情報ファイル (保守情報)	<サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/maintenance/stats	<ul style="list-style-type: none"> <li>• eads_maintenance_stats.csv</li> <li>• eads_maintenance_stats_[n].csv<sup>*1</sup></li> </ul>
17	Java ログファイル	サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ	<ul style="list-style-type: none"> <li>• javalog[nn].log</li> <li>• ehjavalog[nn].log<sup>*2</sup></li> </ul>
18	スレッドダンプ	サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ	javacore[PID].[YYMMDDhhmmss].txt
19	起動ログファイル	サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ	eads_start.log

(凡例)

[n], [nn], [nnn] : ファイル数の通番

[PID] : EADs サーバのプロセス ID

[YYMMDDhhmmss] : YY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

Wrap, Shift : ログのローテーション方式

ログのローテーション方式は、次のどちらかを選択できます。

- Wrap：ラップアラウンド方式  
ファイル名にファイル数の通番が付与されます。
- Shift：シフト方式  
ファイル名は固定です。  
ログファイルのバックアップファイル名にファイル数の通番が付与されます。

#### 注※1

統計情報ファイルは1日ごとにファイルをローテーションし、ファイル名には通番が付与されます。ローテーションする際、サーバ定義の `eads.statistics.backupnum` パラメタに指定した数の統計情報ファイルをバックアップファイルとして残します。

統計情報ファイルのローテーションについては、「[7.4.4 統計情報ファイルのローテーションの設定](#)」を参照してください。

#### 注※2

Explicit ヒープに関するログを出力するファイルです。

### 注意事項

ファイルの出力先に、ネットワークで接続された別マシンのディレクトリは指定できません。

## 7.4.3 ファイルサイズおよびファイル数の設定

ログファイルのファイルサイズおよびファイル数は、基本的にデフォルト値を設定してください。EADsサーバ構築後、必要に応じてパラメタの指定値を変更してください。

## 7.4.4 統計情報ファイルのローテーションの設定

統計情報は、統計情報が更新されるタイミングでファイルに出力されます。

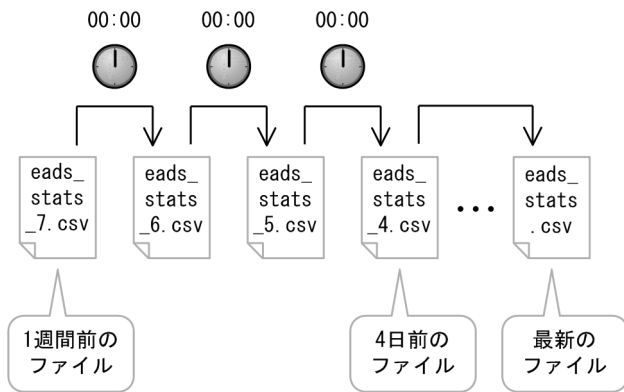
取得する統計情報ファイル数をサーバ定義の `eads.statistics.backupnum` パラメタに指定します。

1日分の統計情報が1ファイルに出力されます。統計情報ファイルは1日ごとにファイルをローテーションします。

固定の時刻（00時00分）を過ぎてから、最初に統計情報が更新されるタイミングで出力先のファイルが切り替わります。

なお、統計情報ファイルは、ローテーションするごとにファイル名が `eads_stats_[n].csv` にリネームされます。[n]はファイルの通番を表します。番号の小さいファイルほど新しく作成されたファイルとなり、`eads_stats.csv`（通番のないファイル）が最新のファイルになります。





(凡例)

→ : 出力先のファイルの切り替え

## 7.4.5 ログファイルに依存するパラメタ

ログファイルに依存するパラメタの一覧を次の表に示します。

表 7-5 ログファイルに依存するパラメタ一覧 (EADs サーバ)

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	サーバ定義ファイル	<code>eads.logger.dir</code>	パス名	<運用ディレクトリ>/logs
2		<code>eads.logger.message.rotationStyle</code>	<ul style="list-style-type: none"> <li>Wrap</li> <li>Shift</li> </ul>	Wrap
3		<code>eads.logger.message.filesize</code>	ファイルサイズ (4096 バイト ~ 2147483647 バイト)	1048576 バイト
4		<code>eads.logger.message.filenum</code>	ファイル数 (1 ~ 16)	2
5		<code>eads.logger.message.console.enable</code>	<ul style="list-style-type: none"> <li>true</li> <li>false</li> </ul>	false
6		<code>eads.logger.exception.rotationStyle</code>	<ul style="list-style-type: none"> <li>Wrap</li> <li>Shift</li> </ul>	Wrap
7		<code>eads.logger.exception.filesize</code>	ファイルサイズ (4096 バイト ~ 2147483647 バイト)	1048576 バイト
8		<code>eads.logger.exception.filenum</code>	ファイル数 (1 ~ 16)	2
9		<code>eads.logger.user.rotationStyle</code>	<ul style="list-style-type: none"> <li>Wrap</li> </ul>	Wrap

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
			<ul style="list-style-type: none"> <li>• Shift</li> </ul>	
10		eads.logger.user.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
11		eads.logger.user.filenum	ファイル数 (1～ 16)	2
12		eads.logger.userException.rotationStyle	<ul style="list-style-type: none"> <li>• Wrap</li> <li>• Shift</li> </ul>	Wrap
13		eads.logger.userException.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
14		eads.logger.userException.filenum	ファイル数 (1～ 16)	2
15		eads.logger.cache.rotationStyle	<ul style="list-style-type: none"> <li>• Wrap</li> <li>• Shift</li> </ul>	Wrap
16		eads.logger.cache.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
17		eads.logger.cache.filenum	ファイル数 (1～ 16)	2
18		eads.prf.enable	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	false
19		eads.statistics.interval	1～3600	60
20		eads.statistics.backupnum	1～366	7
21		eads.statistics.compaction.effect.division	1～10	5
22	起動設定ファイル	eads.prf.filesize	ファイルサイズ (1024 キロバイ ト～1048576 キ ロバイト)	65536 キロバイト
23		eads.prf.filenum	ファイル数 (3～ 256)	16
24		eads.prf.level	<ul style="list-style-type: none"> <li>• 0x00000000 (標準)</li> <li>• 0x40000000 (詳細)</li> </ul>	0x40000000

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
25		eads.java.log.filesize	ファイルサイズ (1 メガバイト～ 2097152 メガバ イト)	8 メガバイト
26		eads.java.log.filenum	ファイル数 (1～ 99)	4
27	コマンド定義ファ イル	eads.logger.dir	パス名	<運用ディレクトリ >/logs
28		eads.logger.message.rotationStyle	<ul style="list-style-type: none"> <li>• Wrap</li> <li>• Shift</li> </ul>	Wrap
29		eads.logger.message.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
30		eads.logger.message.filenum	ファイル数 (1～ 64)	2
31		eads.logger.exception.rotationStyle	<ul style="list-style-type: none"> <li>• Wrap</li> <li>• Shift</li> </ul>	Wrap
32		eads.logger.exception.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
33		eads.logger.exception.filenum	ファイル数 (1～ 16)	2

## (1) サーバ定義ファイル

### (a) eads.logger.dir

EADs サーバによって出力されるログファイルの出力先ディレクトリのパスを指定します。

指定されたパスが存在しない場合は作成します。

このパラメタに指定したディレクトリ下に出力されるログファイルについては、「[7.4.2 ファイルの出力先の設定](#)」を参照してください。

出力先ディレクトリのパスには、最大 96 文字の半角英数字 (0～9, A～Z, a～z), アンダースコア (\_), コロン (:), およびセパレータ (Linux の場合はスラッシュ (/), Windows の場合は円符号 (¥)) が指定できます。

このパラメタの指定値を変更した場合は、このパラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (b) eads.logger.message.rotationStyle

EADs サーバによって出力されるメッセージログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (c) eads.logger.message.filesize

EADs サーバによって出力されるメッセージログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (d) eads.logger.message.filenum

EADs サーバによって出力されるメッセージログのファイル数を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (e) eads.logger.message.console.enable

EADs サーバによって出力されるメッセージログの、標準出力への出力を有効にするか、無効にするかを指定します。

true :

標準出力への出力を有効にします。

false :

標準出力への出力を無効にします。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (f) eads.logger.exception.rotationStyle

EADs サーバによって出力される例外ログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

### (g) `eads.logger.exception.filesize`

EADs サーバによって出力される例外ログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

### (h) `eads.logger.exception.filenum`

EADs サーバによって出力される例外ログのファイル数を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

### (i) `eads.logger.user.rotationStyle`

ユーザログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

### (j) `eads.logger.user.filesize`

ユーザログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

### (k) `eads.logger.user.filenum`

ユーザログのファイル数を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (l) `eads.logger.userException.rotationStyle`

ユーザ例外ログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (m) `eads.logger.userException.filesize`

ユーザ例外ログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (n) `eads.logger.userException.filenum`

ユーザ例外ログのファイル数を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (o) `eads.logger.cache.rotationStyle`

キャッシュファイル操作ログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (p) `eads.logger.cache.filesize`

キャッシュファイル操作ログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (q) eads.logger.cache.filenum

キャッシュファイル操作ログのファイル数を指定します。

このパラメタの指定値を変更した場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (r) eads.prf.enable

PRF トレースの出力を有効にするか、無効にするかを指定します。

true :

PRF トレースの出力を有効にします。

false :

PRF トレースの出力を無効にします。

PRF トレースの出力を有効にすると、保守情報を取得できます。ただし、性能に影響を与えるおそれがあります。

PRF トレースの出力を無効にする場合、クラスタ内の全 EADs サーバの PRF トレースの出力を無効にしてください。クラスタ内で有効と無効が混在すると、正しい内容が出力されないことがあります。なお、無効にした場合、PRF デーモンは起動しません。

### ■ 注意事項

PRF トレースの出力を無効にした場合、障害発生時の原因調査ができないおそれがあります。

## (s) eads.statistics.interval

統計情報を出力する間隔（単位：秒）を指定します。

## (t) eads.statistics.backupnum

統計情報の取得ファイル数を指定します。

このパラメタの指定値を変更した場合は、次に示すファイルを別のディレクトリに移動するか、または削除してください。

- ・ <サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ>/stats 下の、拡張子が .mm の全ファイル
- ・ <サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ>/maintenance/stats 下の、拡張子が .mm の全ファイル

## (u) eads.statistics.compaction.effect.division

キャッシュの統計情報で使用する、コンパクションの効果のファイル数の分布を分割する個数を指定します。

例えば、このパラメタに 4 を指定した場合、分布は 4 つに分割されるため、コンパクションの効果が「0～25%」、「26～50%」、「51～75%」および「76～100%」であるファイル数がそれぞれ出力されます。

効果の値が整数にならない場合、小数点以下はすべて切り捨てられます。

## (2) 起動設定ファイル

### (a) eads.prf.filesize

PRF トレースのファイルサイズ（単位：キロバイト）を指定します。

サーバ定義の `eads.prf.enable` パラメタが `false` の場合、この指定は無効となります。

### (b) eads.prf.filenum

PRF トレースのファイル数を指定します。

サーバ定義の `eads.prf.enable` パラメタが `false` の場合、この指定は無効となります。

### (c) eads.prf.level

PRF トレースの出力レベルを指定します。

0x00000000 :

標準

0x40000000 :

詳細

サーバ定義の `eads.prf.enable` パラメタが `false` の場合、この指定は無効となります。

#### 注意事項

不正な値を指定すると PRF デーモンの起動でエラーとなるため、注意してください。

### (d) eads.java.log.filesize

Java ログのファイルサイズ（単位：メガバイト）を指定します。

### (e) eads.java.log.filenum

Java ログのファイル数を指定します。



### (3) コマンド定義ファイル

#### (a) eads.logger.dir

コマンド実行時に出力されるログファイルの出力先ディレクトリのパスを指定します。

指定されたパスが存在しない場合は作成します。

このパラメタに指定したディレクトリ下に出力されるログファイルについては、「[7.4.2 ファイルの出力先の設定](#)」を参照してください。

出力先ディレクトリのパスには、最大 96 文字の半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), コロン (:), およびセパレータ (Linux の場合はスラッシュ (/), Windows の場合は円符号 (¥)) が指定できます。

このパラメタの指定値を変更した場合は、このパラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

#### (b) eads.logger.message.rotationStyle

コマンド実行時に出力されるメッセージログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

#### (c) eads.logger.message.filesize

コマンド実行時に出力されるメッセージログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

#### (d) eads.logger.message.filenum

コマンド実行時に出力されるメッセージログのファイル数を指定します。

このパラメタの指定値を変更した場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (e) `eads.logger.exception.rotationStyle`

コマンド実行時に出力される例外ログのローテーション方式を指定します。

Wrap :

ラップアラウンド方式を使用します。

Shift :

シフト方式を使用します。

このパラメタの指定値を変更した場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (f) `eads.logger.exception.filesize`

コマンド実行時に出力される例外ログの 1 ファイル当たりのファイルサイズ (単位: バイト) を指定します。

このパラメタの指定値を変更した場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## (g) `eads.logger.exception.filenum`

コマンド実行時に出力される例外ログのファイル数を指定します。

このパラメタの指定値を変更した場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

## 7.5 クラスタ構成に依存するパラメタの設計

クラスタ構成に依存するパラメタを設計します。

### 7.5.1 EADs サーバの位置の設定

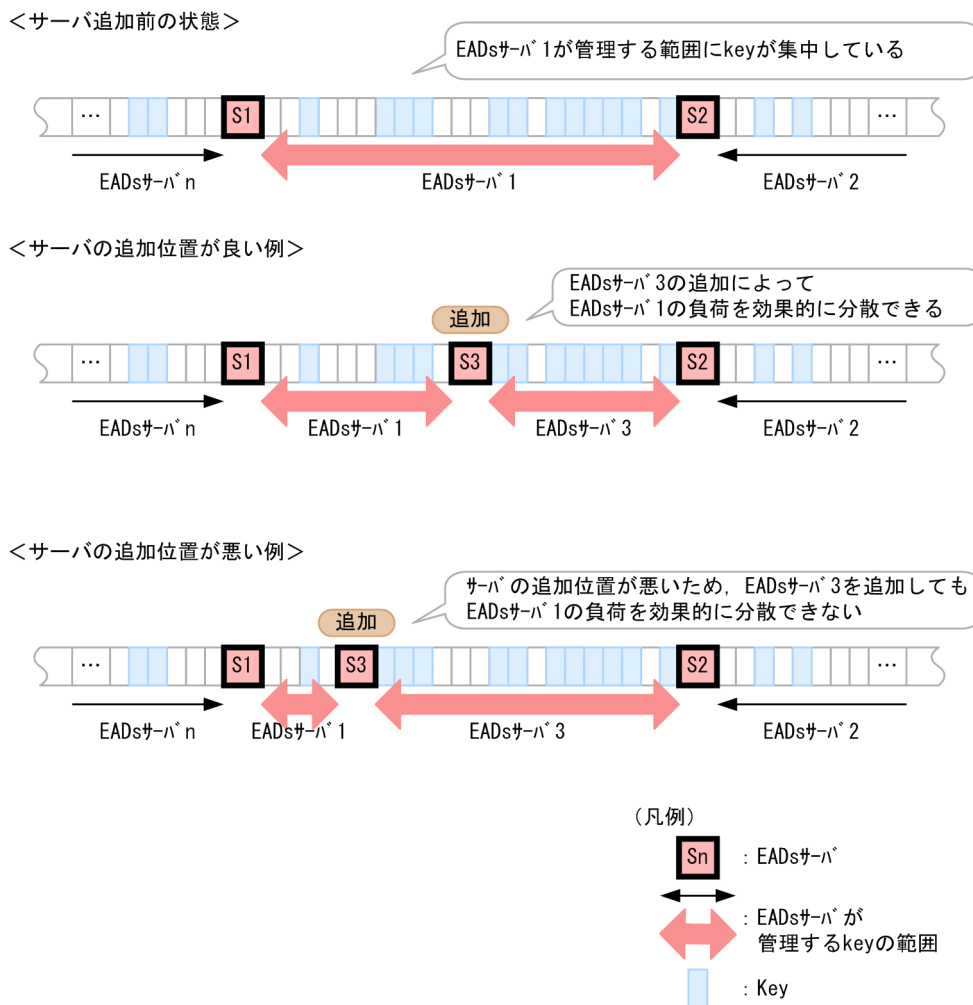
EADs サーバをクラスタ内のどこに配置するかを `eads.node.< EADs サーバ ID >.position` パラメタで設定します。EADs サーバの位置（ハッシュ値）を指定することで、負荷を分散させることができます。

位置を指定しなければ、EADs サーバはクラスタ上の論理的な位置に均等に分散配置されます。

key や業務の特性によって、データの配置（データ数、データ使用量、またはデータアクセス数）が偏ることがあります。そうすると、一部の EADs サーバに負荷が掛かり、レスポンス速度の低下やリソース不足などにつながります。

EADs サーバの位置を指定する場合は、統計情報（KvCount, UsedMemorySize, および RequestCount）を確認し、位置を分析します。

図 7-2 EADs サーバの位置の設定



データの分散配置については、「2.5 コンシステント・ハッシングによるデータの分散配置」を参照してください。

## 7.5.2 データの多重度の設定

`eads.replication.factor` パラメタでデータの多重度を設定します。

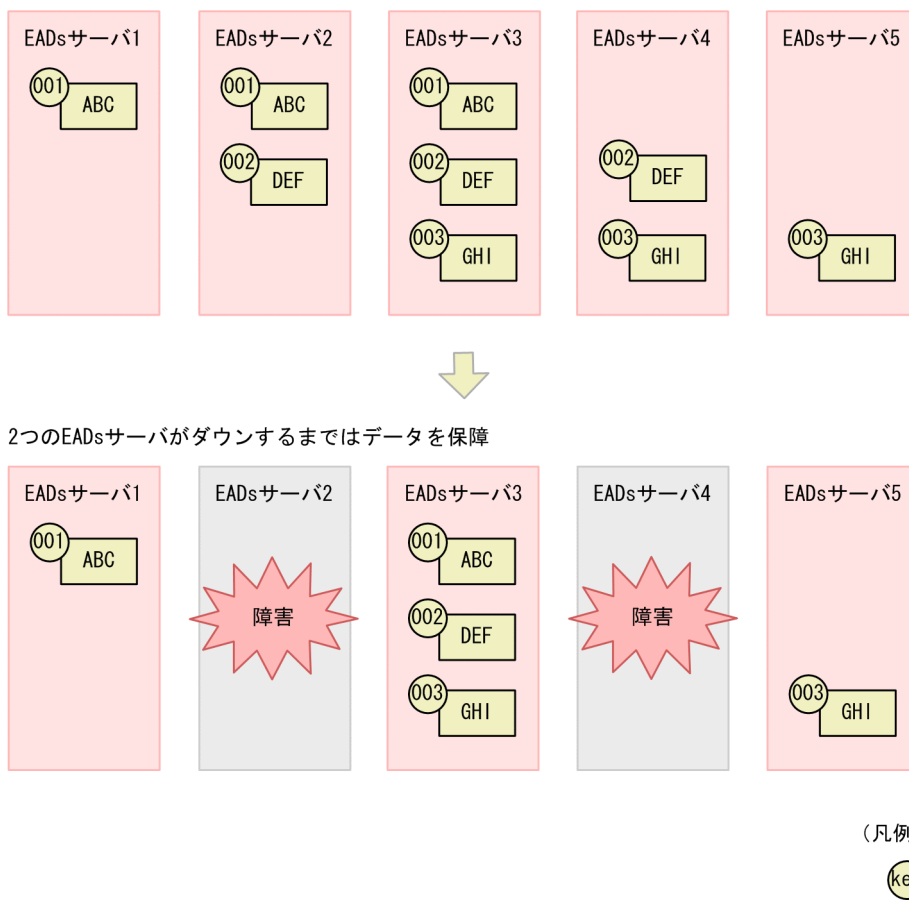
クラスタを構成する EADs サーバ数は、「データの多重度×2-1」以上であることが前提です。障害発生時にデータの多重度未満の数の EADs サーバがダウンしても、データの整合性は確保されます。そのため、多重度を上げると信頼性と耐障害性が向上します。

一方で、必要なメモリ所要量が増加したり、EADs サーバ間の通信などのオーバーヘッドが大きくなったりします。

多重度はこのようなメリット、デメリットを考慮して設定する必要があります。

多重度を 3 に設定した場合の例を次の図に示します。

図 7-3 データの多重度の設定



多重度を 3 に設定した場合、EADs サーバ数は最小で 5 つ必要です。5 つであれば、障害発生時に 2 つの EADs サーバがダウンしても、データが欠落することはありません。

データの多重化については、「2.9 データの多重化」を参照してください。

### 7.5.3 クラスタ構成に依存するパラメタ

クラスタ構成に依存するパラメタの一覧を次の表に示します。

なお、必ず指定するパラメタを太字で表記しています。

表 7-6 クラスタ構成に依存するパラメタ一覧

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	クラスタ定義ファイル	<b>eads.node.&lt; EADs サーバ ID &gt;.address</b>	IP アドレスまたはホスト名	なし
2		<b>eads.node.&lt; EADs サーバ ID &gt;.port</b>	ポート番号	なし
3		<b>eads.node.&lt; EADs サーバ ID &gt;.position</b>	-2147483648～2147483647	なし
4		<b>eads.cluster.heartbeat.address</b>	IP アドレス (マルチキャストアドレス)	239.255.2.1
5		<b>eads.cluster.heartbeat.port</b>	ポート番号	24630
6		<b>eads.replication.factor</b>	1～10	2

#### 注意事項

クラスタ定義ファイルの定義内容は、クラスタを構成する全 EADs サーバで同じになるようにしてください。同一でないとクラスタが構築できないため、EADs サーバは起動しません。

また、範囲外の値が設定されていた場合は、デフォルト値が設定されます。

## (1) クラスタ定義ファイル

### (a) **eads.node.< EADs サーバ ID >.address**

クラスタを構成する EADs サーバの IP アドレスまたはホスト名 (サーバ定義の `eads.server.address` パラメタの値) を指定します。

ホスト名で指定する場合は、ホスト名から IP アドレスが一意に識別できるホスト名を指定してください。

EADs サーバ ID はユーザが任意に割り振る番号 (1～96 の整数) です。EADs サーバ ID はクラスタ内の EADs サーバを一意に識別するために、ストアデータファイル名や PRF トレース情報に使用されます。連番でなくてもかまいませんが、クラスタ内ではユニークにする必要があります。

## (b) eads.node.< EADs サーバ ID >.port

クラスタを構成する EADs サーバのポート番号（サーバ定義の `eads.server.port` パラメタの値）を指定します。

EADs サーバ ID はユーザが任意に割り振る番号（1～96 の整数）です。EADs サーバ ID はクラスタ内の EADs サーバを一意に識別するために、ストアデータファイル名や PRF トレース情報に使用されます。連番でなくてもかまいませんが、クラスタ内ではユニークにする必要があります。

## (c) eads.node.< EADs サーバ ID >.position

EADs サーバの位置（ハッシュ値）を指定します。

EADs サーバ ID はユーザが任意に割り振る番号（1～96 の整数）です。EADs サーバ ID はクラスタ内の EADs サーバを一意に識別するために、ストアデータファイル名や PRF トレース情報に使用されます。連番でなくてもかまいませんが、クラスタ内ではユニークにする必要があります。

このパラメタは省略できます。省略した場合、EADs サーバがクラスタ上の論理的な位置に均等に分散配置されるように設定されます。

指定する場合は、クラスタを構成する全 EADs サーバの位置を指定してください。指定の有無が混在した場合はエラーとなり、EADs サーバの起動に失敗します。

## (d) eads.cluster.heartbeat.address

EADs サーバ間のハートビートの送信に使用する IP アドレス（マルチキャストアドレス）を指定します。

IP アドレスのエイリアス名でも指定できます。

## (e) eads.cluster.heartbeat.port

EADs サーバ間のハートビートの送信に使用するポート番号を指定します。

## (f) eads.replication.factor

データの多重度を指定します。

### 注意事項

クラスタを構成する EADs サーバ数は、「データの多重度×2-1」以上であることが前提です。

クラスタを構成する EADs サーバ数が、「データの多重度×2-1」より少なかった場合、エラーとなり EADs サーバの開始に失敗します。

## 7.6 バックアップファイルに依存するパラメタの設計

バックアップファイルに依存するパラメタを設計します。

メモリキャッシュに格納されているデータを出力するバックアップファイルを、ストアデータファイルといいます。

### 7.6.1 ファイルの出力先の設定

ストアデータファイルの出力先は変更できます。ファイルの出力先を次の表に示します。

表 7-7 ストアデータファイルの出力先

項番	ストアデータファイル種別	出力先	ファイル名
1	<code>eztool export</code> コマンド実行時に出力されるストアデータファイル	サーバ定義の <code>eads.storeData.dir</code> パラメタに指定したディレクトリ	<ul style="list-style-type: none"><li><code>eads_[xxx]_[EADs サーバ ID].esd</code></li><li><code>eads_single_[xxx]_[EADs サーバ ID].esd</code></li></ul>
2	<code>eztool stop</code> コマンド実行時に出力されるストアデータファイル	サーバ定義の <code>eads.storeData.dir</code> パラメタに指定したディレクトリ	<code>eads_stop_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd</code>

(凡例)

[xxx] : `eztool export` コマンド実行時に指定したストアデータファイルキー

[YYYYMMDDhhmmss] : YY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

注

ストアデータファイル名は、接頭辞、ストアデータファイルキー (任意の値)、接尾辞で構成されます。接頭辞は、ストアデータファイルを出力したコマンドによって異なります。

`eztool export` コマンド実行時に出力されるストアデータファイルの場合、ストアデータファイルキーを省略すると、次の表に示すように、コマンド実行日時がストアデータファイルキーとなります。

なお、`eztool stop` コマンド実行時に出力されるストアデータファイルの場合、ストアデータファイルキーは指定できません。コマンド実行日時がストアデータファイルキーとなります。

-s または --single オプション指定	ストアデータファイル名	世代管理の対象
なし	<code>eads_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd</code>	対象
あり	<code>eads_single_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd</code>	対象外

(凡例)

[YYYYMMDDhhmmss] : コマンド実行日時

YYYY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

## 注意事項

ファイルの出力先に、ネットワークで接続された別マシンのディレクトリは指定できません。

## 7.6.2 ストアデータファイルの世代数の設定

ストアデータファイルは `eztool export` コマンド、または `eztool stop` コマンド実行時に出力されます。

### (1) eztool export コマンド実行時に出力されるストアデータファイル

`eztool export` コマンドを実行してデータを書き出すとき、通常はストアデータファイルキーを省略します。省略すると、次に示すように、コマンド実行日時がストアデータファイルキーとなります。

```
eads_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd
```

(凡例)

[YYYYMMDDhhmmss]：コマンド実行日時

YYYY：年、MM：月、DD：日、hh：時 (00~23)、mm：分、ss：秒

ストアデータファイルキーには、自動的に接頭辞 (`eads_`) と接尾辞 (`_EADs サーバ ID.拡張子`) が付きます。拡張子は `[.esd]` です。

ストアデータファイルキーが上記の形式であれば、`eztool import` コマンドを実行してデータを読み込むとき、コマンド実行日時が最新のストアデータファイルが自動的に読み込まれます。

ストアデータファイルには、EADs サーバが保持する全データが出力されます。全データには、多重化によってコピーされたデータも含まれます。1つのデータに対してキャッシュ名、key、更新日付、および value が出力されます。

ストアデータファイルの出力先については、「[7.4.2 ファイルの出力先の設定](#)」を参照してください。

なお、出力先にすでに同じ名称のストアデータファイルが存在する場合は、上書きされないで `eztool export` コマンドがエラーとなります。

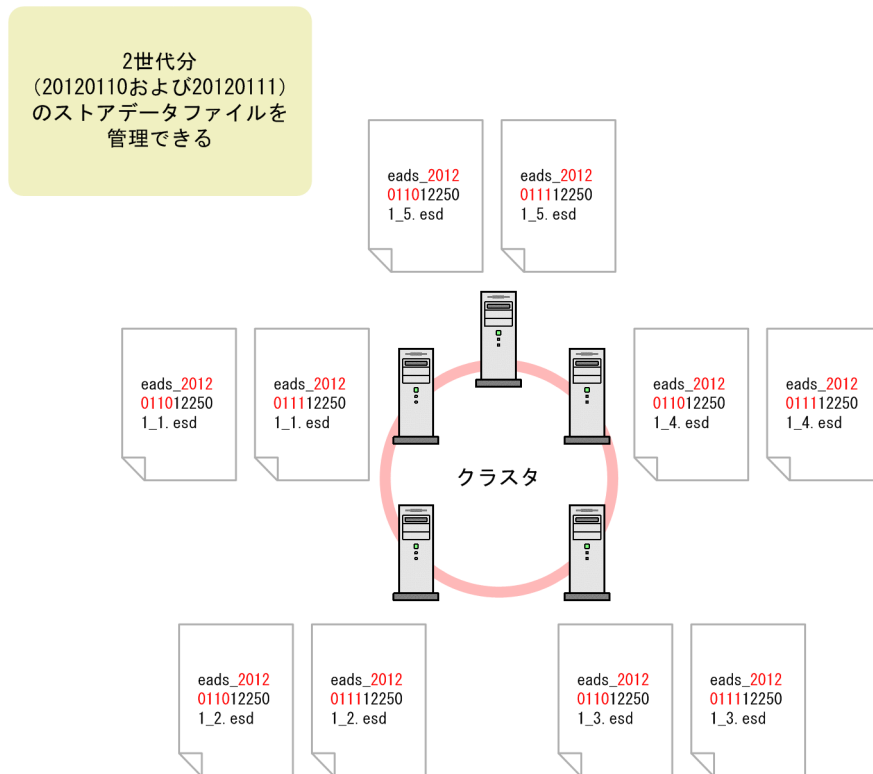
### (2) ストアデータファイルの世代管理

コマンド実行日時が同じストアデータファイルは1つのグループとして見なされ、クラスタ定義の `eads.storeData.generation.count` パラメタに指定した上限値まで、クラスタ全体で世代管理されます。これによって、ストアデータファイルの増加によるディスク容量の圧迫を防ぎます。

クラスタ全体で保持するストアデータファイルの世代数の上限値を2 (デフォルト) に設定した場合の例を次の図に示します。



図 7-4 ストアデータファイルの世代数の設定



この上限値を超えるストアデータファイルが出力されると、`eztool export` コマンドはエラーとなります。  
`eztool deleteesd` コマンドを実行して、不要なストアデータファイルを削除してください。

ストアデータファイルの確認および削除手順については、「11.6 ストアデータファイルを管理したい」を参照してください。

## 注意事項

ストアデータファイルが世代管理されないケースを次に示します。

- 世代数の上限値（クラスタ定義の `eads.storeData.generation.count` パラメタ）に 0 を指定した場合
- `eztool export` コマンドに `-s` または `--single` オプションを指定した場合

## (3) eztool stop コマンド実行時に出力されるストアデータファイル

`eztool stop` コマンドを実行して EADs サーバを終了するとき、データをファイルに書き出してから EADs サーバを終了します。

このとき、ストアデータファイルキーは指定できません。次に示すように、コマンド実行日時がストアデータファイルキーとなります。

また、ストアデータファイルキーには、自動的に接頭辞 (eads\_stop\_) と接尾辞 (\_EADs サーバ ID.拡張子) が付きます。拡張子は「.esd」です。

eads\_stop\_[YYYYMMDDhhmmss]\_[EADs サーバ ID].esd

(凡例)

[YYYYMMDDhhmmss]：コマンド実行日時

YYYY：年，MM：月，DD：日，hh：時 (00~23)，mm：分，ss：秒

`eztool stop` コマンド実行時に出力されるストアデータファイルの世代数の上限値を、クラスタ定義の `eads.storeData.stop_command.generation.count` パラメタに指定します。

この上限値を超えるストアデータファイルが出力されると、最新のストアデータファイルが出力されたあと、最も古いストアデータファイルが自動的に削除されます。

## ■ 注意事項

`eztool stop` コマンドに `--no_export` オプションを指定した場合は、EADs サーバ終了時に、データをファイルに出力しません。

## 7.6.3 バックアップファイルに依存するパラメタ

バックアップファイルに依存するパラメタの一覧を次の表に示します。

表 7-8 バックアップファイルに依存するパラメタ一覧

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	サーバ定義ファイル	<code>eads.storeData.dir</code>	パス名	<運用ディレクトリ>/store
2	クラスタ定義ファイル	<code>eads.storeData.generation.count</code>	0~32	2
3		<code>eads.storeData.stop_command.generation.count</code>	1~32	1

### (1) サーバ定義ファイル

#### (a) `eads.storeData.dir`

ストアデータファイルの出力先ディレクトリのパスを指定します。

## (2) クラスタ定義ファイル

### (a) `eads.storeData.generation.count`

`eztool export` コマンド実行時に出力されるストアデータファイルの世代数の上限値を指定します。

この上限値を超えるストアデータファイルが出力されると、`eztool export` コマンドはエラーとなります。

また、出力先にすでに同じ名称のストアデータファイルが存在する場合は、上書きされないで `eztool export` コマンドがエラーとなります。

このパラメタに 0 を指定した場合、ストアデータファイルは世代管理されません。

### (b) `eads.storeData.stop_command.generation.count`

`eztool stop` コマンド実行時に出力されるストアデータファイルの世代数の上限値を指定します。

この上限値を超えるストアデータファイルが出力されると、最新のストアデータファイルが出力されたあと、最も古いストアデータファイルが自動的に削除されます。

## 7.7 キャッシュの運用に依存するパラメタの設計

---

この節の内容は、ディスクキャッシュ、および2Wayキャッシュを使用する場合にお読みください。

キャッシュの運用に依存するパラメタを設計します。

### 7.7.1 キャッシュタイプごとのパラメタの設定

作成するキャッシュタイプによって、キャッシュ定義のパラメタの設定が異なります。

キャッシュタイプについては、「[2.3.1 キャッシュの種類 \(キャッシュタイプ\)](#)」を参照してください。

#### (1) メモリキャッシュを作成する場合

メモリキャッシュを作成する場合、キャッシュ定義ファイルの作成は任意です。

キャッシュ定義ファイルを作成する場合、`eads.cache.type` パラメタに `Memory` を指定します。

#### (2) ディスクキャッシュ、または2Wayキャッシュを作成する場合

ディスクキャッシュ、または2Wayキャッシュを作成する場合、キャッシュ定義ファイルを必ず作成します。

また、次のキャッシュ定義のパラメタは必ず指定します。

- `eads.cache.type`  
ディスクキャッシュを作成する場合は、`Disk` を指定します。  
2Wayキャッシュを作成する場合は、`2Way` を指定します。
- `eads.cache.disk.<n>.dir`  
「[7.7.2\(2\) キャッシュファイルの格納先の設定](#)」を参照してください。
- `eads.cache.disk.filesize`  
「[4.4.1 キャッシュデータファイルのサイズおよび面数を見積もる](#)」で見積もった値を指定します。
- `eads.cache.disk.filenum`  
「[4.4.1 キャッシュデータファイルのサイズおよび面数を見積もる](#)」で見積もった値を指定します。

### 7.7.2 キャッシュファイルの種類と格納先の設定

ディスクキャッシュ、および2Wayキャッシュを使用する場合に、キャッシュの情報を格納するファイルの格納先を指定します。

キャッシュの情報を格納するファイルを、キャッシュファイルといいます。

ディスクキャッシュ、および 2Way キャッシュを使用する場合は、キャッシュファイルの格納先を指定します。

## (1) キャッシュファイルの種類

キャッシュファイルには次の 3 種類があります。

### (a) キャッシュデータファイル

キャッシュに格納するデータ (key および value) を格納するファイルです。

### (b) キャッシュインデックスファイル

キャッシュデータファイルに含まれるデータのインデックスを格納するファイルです。

キャッシュデータファイルと同じ数のファイルが作成されます。

### (c) キャッシュ情報ファイル

キャッシュの設定情報を格納するファイルです。

キャッシュごとにファイルが 1 つ作成されます。

## (2) キャッシュファイルの格納先の設定

キャッシュファイルの格納先を指定します。

キャッシュファイルのうち、キャッシュデータファイルの格納先は必ず指定します。キャッシュファイルの格納先を次の表に示します。

表 7-9 キャッシュファイルの格納先とファイル名

項番	キャッシュファイル種別	格納先	ファイル名
1	キャッシュデータファイル	<キャッシュ定義の eads.cache.disk.<n>.dir パラメタで指定したディレクトリ>/<キャッシュ名>	eads_data_[EADs サーバ ID]_[キャッシュ名]_[レンジ ID]_[nnnnn].ecf
2	キャッシュインデックスファイル	<キャッシュ定義の eads.cache.disk.info.dir パラメタで指定したディレクトリ>/<キャッシュ名>	eads_index_[EADs サーバ ID]_[キャッシュ名]_[レンジ ID]_[nnnnn].ecf
3	キャッシュ情報ファイル	<キャッシュ定義の eads.cache.disk.info.dir パラメタで指定したディレクトリ>/<キャッシュ名>	eads_info_[EADs サーバ ID]_[キャッシュ名].ecf

(凡例)

[レンジ ID]：キャッシュ内のレンジを識別するための番号 (01~96 の整数) です。データのコピー元 EADs サーバのサーバ ID と一致します。

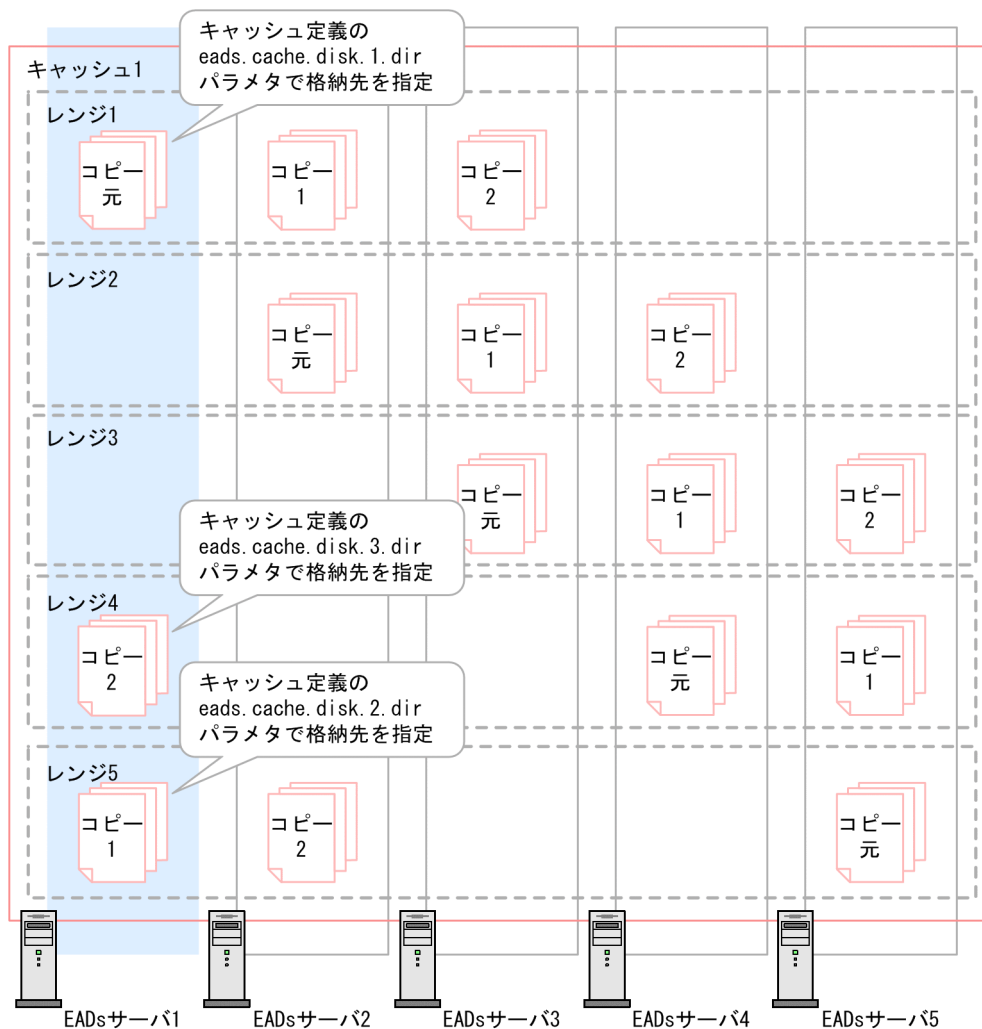
[nnnnn]：ファイル数の連番 (5桁の整数) です。

### (3) データを多重化している場合の、キャッシュデータファイルの格納先と EADs サーバの関係


キャッシュデータファイルの格納先は、`eads.cache.disk.<n>.dir` パラメタで、指定した多重度の数だけ指定する必要があります。例えば、データの多重度を3に設定している場合、nには1~3を指定します。

データの多重度を3に設定している場合の、キャッシュデータファイルの格納先と EADs サーバの関係について、EADs サーバ1を例にして説明します。

図 7-5 キャッシュデータファイルの格納先と EADs サーバの関係



(凡例)

 : キャッシュ1のキャッシュデータファイル

EADs ではデータが分散配置されるため、キャッシュデータファイルは各 EADs サーバに格納されます。

EADs サーバ 1 には、次のキャッシュデータファイルが格納されます。

- レンジ 1 の範囲のデータ（コピー元のデータ）を格納するキャッシュデータファイル
- レンジ 4 の範囲のデータのコピーを格納するキャッシュデータファイル
- レンジ 5 の範囲のデータのコピーを格納するキャッシュデータファイル

このキャッシュデータファイルの格納先を、それぞれキャッシュ定義の `eads.cache.disk.<n>.dir` パラメータで指定します。

`eads.cache.disk.<n>.dir` パラメータの指定例（キャッシュデータファイルを同じディレクトリで管理する場合）

```
eads.cache.disk.1.dir=/hdd/cache_server01 # コピー元のデータの格納先
eads.cache.disk.2.dir=/hdd/cache_server01 # レンジ5のデータ（コピー1）の格納先
eads.cache.disk.3.dir=/hdd/cache_server01 # レンジ4のデータ（コピー2）の格納先
```

`eads.cache.disk.<n>.dir` パラメータの指定例（キャッシュデータファイルを異なるディレクトリで管理する場合）

```
eads.cache.disk.1.dir=/hdd/cache_server01_range01 # コピー元のデータの格納先
eads.cache.disk.2.dir=/hdd/cache_server01_range05 # レンジ5のデータ（コピー1）の格納先
eads.cache.disk.3.dir=/hdd/cache_server01_range04 # レンジ4のデータ（コピー2）の格納先
```

### 7.7.3 キャッシュファイルのサイズの設定

キャッシュファイルのサイズの設定については、「[4.4 キャッシュファイルの容量を見積もる](#)」を参照してください。

### 7.7.4 キャッシュの運用に依存するパラメータ

キャッシュの運用に依存するパラメータの一覧を次の表に示します。

なお、必ず指定するパラメータを太字で表記しています。

表 7-10 キャッシュの運用に依存するパラメータ一覧

項番	定義ファイル	パラメータ名	指定する値	デフォルト値
1	サーバ定義ファイル	<code>eads.cache.disk.getError.isolate.enable</code>	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	true
2	キャッシュ定義ファイル	<code>eads.cache.type</code>	<ul style="list-style-type: none"><li>• Memory</li></ul>	なし

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
			<ul style="list-style-type: none"> <li>• Disk</li> <li>• 2Way</li> </ul>	
3		<code>eads.cache.disk.info.dir</code>	パス名	<運用ディレクトリ>/store
4		<code>eads.cache.disk.&lt;n&gt;.dir</code>	パス名	なし
5		<code>eads.cache.disk.filesize</code>	ファイルサイズ (16 メガバイト～ 128 メガバイト)	なし
6		<code>eads.cache.disk.filenum</code>	ファイル数 (8～ 32768)	なし
7		<code>eads.cache.disk.blocksize</code>	データサイズ <ul style="list-style-type: none"> <li>• 1 キロバイト</li> <li>• 2 キロバイト</li> <li>• 4 キロバイト</li> <li>• 8 キロバイト</li> <li>• 16 キロバイト</li> <li>• 32 キロバイト</li> <li>• 64 キロバイト</li> <li>• 128 キロバイト</li> </ul>	1 キロバイト
8		<code>eads.cache.restore.dataSender.sendInterval</code>	0～60000 (ミリ秒)	1000 ミリ秒
9		<code>eads.cache.restore.dataSender.sendSize</code>	10240 バイト～ 2147483647 バイト	102400 バイト

## ■ 注意事項

キャッシュ定義ファイルの次のパラメタの定義内容は、クラスタを構成する全 EADs サーバで同じになるようにしてください。同じでないとキャッシュを作成できません。

- `eads.cache.type`
- `eads.cache.disk.filesize`
- `eads.cache.disk.filenum`



## (1) サーバ定義ファイル

### (a) `eads.cache.disk.getError.isolate.enable`

ディスクキャッシュおよび 2Way キャッシュでディスク I/O エラーが発生した場合に、EADs サーバを縮退させるかどうかを指定します。

true :

ディスク I/O エラーが発生した場合に、EADs サーバを縮退させます。

false :

ディスク I/O エラーが発生しても、次に示すデータの整合性に影響を与えない API でのディスク I/O エラーの場合は、EADs サーバを縮退させません。

- API (`get`)
- Group インタフェースの次に示すメソッド
  - `getLastUpdateTime()`
  - `getValueUsageSize()`
- `eztool get` コマンド

API には、ユーザファンクション内で実行される API を含みます。

このパラメタに `false` を指定すると、一時的なディスク障害によるディスク I/O エラーが `get` 処理などで発生しても EADs サーバが縮退しません。このため、可用性の向上が期待できます。

しかし、`get` 処理だけが長時間続くような AP の場合は、恒久的なディスク障害が発生しても縮退処理が行われないため、`get` 処理のエラーが長時間にわたって発生してしまうおそれがあります。

なお、ディスクキャッシュおよび 2Way キャッシュを使用しない場合は、このパラメタの指定値は無視されます。

## (2) キャッシュ定義ファイル

### (a) `eads.cache.type`

キャッシュのキャッシュタイプを指定します。

Memory :

メモリキャッシュを使用します。

Disk :

ディスクキャッシュを使用します。

2Way :

2Way キャッシュを使用します。

## (b) eads.cache.disk.info.dir

キャッシュ情報ファイル、およびキャッシュインデックスファイルの格納先を指定します。

格納先ディレクトリのパスには、最大 200 文字の半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), コロン (:), およびセパレータ (Linux の場合はスラッシュ (/), Windows の場合は円符号 (¥)) が指定できます。

このパラメタに指定したディレクトリ下に格納されるファイルのファイル名については、「7.7.2(2) キャッシュファイルの格納先の設定」を参照してください。

### 注意事項

同一 EADs サーバ内で、このパラメタの指定値の絶対パスと `eads.cache.disk.<n>.dir` パラメタの指定値の絶対パスが同じにならないようにしてください。

## (c) eads.cache.disk.<n>.dir

キャッシュデータファイルの格納先を指定します。

<n>には、1~データの多重度 (`eads.replication.factor` パラメタの指定値) の整数を指定します。

データを多重化している場合は、データの多重度の数だけ、このパラメタを指定します。

格納先ディレクトリのパスには、最大 1,024 文字の半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), コロン (:), およびセパレータ (Linux の場合はスラッシュ (/), Windows の場合は円符号 (¥)) が指定できます。

このパラメタに指定したディレクトリ下に格納されるファイルのファイル名については、「7.7.2(2) キャッシュファイルの格納先の設定」を参照してください。

### 注意事項

- 同一 EADs サーバ内で、このパラメタの指定値の絶対パスと `eads.cache.disk.info.dir` パラメタの指定値の絶対パスが同じにならないようにしてください。
- `eads.cache.disk.info.dir` パラメタに指定したディレクトリ直下のディレクトリを指定する場合、キャッシュ名と同じ名前のディレクトリを指定しないでください。
- クラスタを構成する EADs サーバ間で、キャッシュデータファイルの格納先に SSD と HDD を混在させるなど、極端に速度差が発生するようなディスク構成にすることはできません。

- 1 台の物理マシン上で複数の EADs サーバを動作させる場合、EADs サーバごとに異なる格納先を指定してください。

#### (d) `eads.cache.disk.filesize`

1 レンジ当たりのキャッシュデータファイルのサイズ（単位：メガバイト）を指定します。

#### (e) `eads.cache.disk.filenum`

1 レンジ当たりのキャッシュデータファイルの面数を指定します。

#### (f) `eads.cache.disk.blocksize`

キャッシュデータファイルに 1 回で書き込むデータサイズ（単位：キロバイト）を指定します。

キャッシュデータファイルの格納先の媒体が HDD の場合は、このパラメタには 1 キロバイト（デフォルト）を指定してください。SSD の場合は、SSD のページサイズを指定してください。

#### (g) `eads.cache.restore.dataSender.sendInterval`

ディスクキャッシュ、および 2Way キャッシュの復旧処理でのデータ送信間隔（単位：ミリ秒）を指定します。

このパラメタの指定値とキャッシュ定義の `eads.cache.restore.dataSender.sendSize` パラメタの指定値は、次の式を満たすように設定してください。

$$\text{復旧処理のデータ送信に使用できる帯域 (単位: bps)} \geq \text{MAX (各キャッシュの復旧処理のデータ送信に使用できる帯域 (単位: bps))}$$

MAX :

計算結果の最も大きい値を選ぶことを示しています。

(例) MAX (3×6, 4 + 7) の計算結果は 18 となります。

各キャッシュの復旧処理のデータ送信に使用できる帯域（単位：bps）：

キャッシュごとに、次の式で求めます。

- メモリキャッシュの場合

$$\begin{aligned} &\text{復旧処理のデータ送信に使用できる帯域 (単位: bps)} \geq \\ &(\text{復旧処理で送信するデータサイズ (単位: バイト)} \times 8) \\ &\div \{(\text{復旧処理でのデータ送信間隔 (単位: ミリ秒)} \\ &+ \text{データ送信に掛かる時間 (単位: ミリ秒)}) \div 1,000\} \end{aligned}$$

復旧処理で送信するデータサイズ（単位：バイト）：

サーバ定義の `eads.restore.dataSender.sendSize` の指定値

復旧処理でのデータ送信間隔（単位：ミリ秒）：

サーバ定義の `eads.restore.dataSender.sendInterval` パラメタの指定値

データ送信に掛かる時間（単位：ミリ秒）：

サーバ定義の `eads.restore.dataSender.sendSize` パラメタに指定したサイズのデータ送信に掛かる時間（単位：ミリ秒）

- ディスクキャッシュおよび 2Way キャッシュの場合

復旧処理のデータ送信に使用できる帯域（単位：bps） $\geq$   
(復旧処理で送信するデータサイズ（単位：バイト） $\times 8$ )  
 $\div$  {(復旧処理でのデータ送信間隔（単位：ミリ秒）  
+ データ送信に掛かる時間（単位：ミリ秒）)  $\div 1,000$ }

復旧処理で送信するデータサイズ（単位：バイト）：

キャッシュ定義の `eads.cache.restore.dataSender.sendSize` の指定値

復旧処理でのデータ送信間隔（単位：ミリ秒）：

キャッシュ定義の `eads.cache.restore.dataSender.sendInterval` パラメタの指定値

データ送信に掛かる時間（単位：ミリ秒）：

キャッシュ定義の `eads.cache.restore.dataSender.sendSize` パラメタに指定したサイズのデータ送信に掛かる時間（単位：ミリ秒）

「9.3.2(5) クラスタの復旧処理」の図中の「復旧処理（データ送信）」に掛かる時間を示します。  
この時間は環境によって異なります。

復旧処理に掛かる時間を短くすると、復旧処理の通信負荷が高くなります。逆に、復旧処理の通信負荷を低くすると、復旧処理に掛かる時間が長くなります。

メモリキャッシュを使用する場合、このパラメタの指定は無効になり、サーバ定義の `eads.restore.dataSender.sendInterval` パラメタの指定値が有効となります。

復旧処理については、「9.3.2(5) クラスタの復旧処理」を参照してください。

## 注意事項

ディスクの書き込み性能を超えないように、このパラメタの指定値を決定してください。

## (h) `eads.cache.restore.dataSender.sendSize`

ディスクキャッシュ、および 2Way キャッシュの復旧処理で送信するデータサイズ（単位：バイト）を指定します。

復旧処理では、データの整合性を回復するために、稼働中の EADs サーバが復旧対象の EADs サーバにデータを送信します。`eads.cache.restore.dataSender.sendInterval` パラメタに指定した間隔で、このパラメタの指定値を超えるまで、10 キロバイト単位でデータを連続して送信します。

メモリキャッシュを使用する場合、このパラメタの指定は無効になり、サーバ定義の `eads.restore.dataSender.sendSize` パラメタの指定値が有効となります。

## ■ 注意事項

- 復旧処理では最低 1 個のデータを送信します。そのため、送信データのサイズが、このパラメタの指定値を超えることがあります。なお、このパラメタに EADs サーバに格納しているデータのサイズよりも小さい値を指定しても、送信量を制限することはできません。
- 復旧データの転送速度がディスクの書き込み速度を超えないように、このパラメタの指定値を決定してください。

# 8

## 環境依存パラメタを設計する (EADs クライアント)

この章では、EADs クライアントの、環境依存パラメタの設計指針について説明します。

## 8.1 定義ファイルの種類 (EADs クライアント)

EADs クライアントで使用する定義ファイルの種類を次の表に示します。

この章では、環境依存パラメタについてだけ説明しています。クライアント定義ファイルに指定できるパラメタの一覧については、「23.1.2(1) クライアント定義ファイル」を参照してください。

表 8-1 定義ファイルの種類 (EADs クライアント)

定義ファイル	ファイル名	説明
クライアント定義ファイル	任意*	EADs クライアントの実行環境を定義します。

### 注※

次に示すディレクトリ下のクライアント定義ファイルを、AP で指定したディレクトリにコピーして編集します (ファイル名も AP で指定したファイル名に合わせます)。

- Java 言語の場合  
/opt/hitachi/xeads/javaclient/conf/eads\_sample\_client.properties
- C 言語の場合  
/opt/hitachi/xeads/cclient/conf/eads\_sample\_client.properties

## 8.2 定義ファイルの形式

次のようにパラメタを指定します。

```
<パラメタ名>=<指定値>
```

### 指定方法

- 改行までが値になります。
- #で始まる行はコメントと見なされます。
- 空行は無視されます。
- 値の後ろには、コメントなどの文字列は追加できません。追加した場合、不正な値と解釈されます。
- 同じパラメタを指定した場合、後から記述したパラメタの指定が有効になります。例えば、次のように指定されている場合、false が有効になります。

```
eads.logger.message.console.enable=true  
eads.logger.message.console.enable=false
```

なお、AP で使用する言語が C 言語の場合は、次の点にも注意してください。

- 使用できる文字は ASCII コード「0x20」～「0x7E」の文字、改行文字（'\n'）、復帰文字（'\r'）、タブ（'\t'）です。
- 1 行に 1 つのパラメタを指定します。1 行は改行（'\n'または'\r\n'）または EOF（End Of File）までとなります。
- 区切り文字（パラメタ名と指定値の区切り）には、イコール（=）のほかに、半角空白、タブが使用できます。
- 区切り文字を 2 つ以上記述した場合、2 つ目以降の区切り文字は指定値の一部と解釈されます。

(例 1) a=b=c

a がパラメタ名、b=c が指定値と解釈されます。

(例 2) a b = c

a がパラメタ名、b = c が指定値と解釈されます。

- 指定値がない場合（パラメタ名と区切り文字だけの場合）、指定値は空文字列と解釈されます。
- 制御文字を含んでいたり、パラメタ名だけを記述したりした場合は無効となります。
- パラメタ名と指定値の前後には、半角空白またはタブが使用できますが、半角空白またはタブは削除されて解釈されます。

(例) △a△=△b△

注 △は半角空白またはタブを表します。

a がパラメタ名、b が指定値と解釈されます。



## 8.3 通信に依存するパラメタの設計

通信に依存するパラメタを設計します。

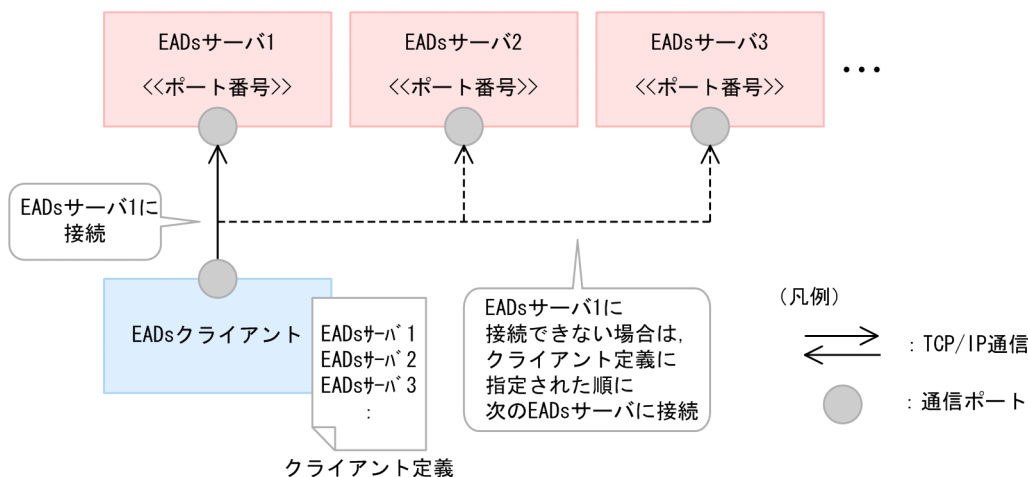
### 8.3.1 接続先 EADs サーバ, IP アドレスまたはホスト名, およびポート番号の設定

EADs サーバへの初回接続時, EADs クライアントはクライアント定義の `eads.client.node.list` パラメタで指定した EADs サーバからランダムに接続先を決定します (2 回目以降の接続では, EADs サーバから取得したクラスタ構成情報に従って接続先を選択します)。

クライアント定義に複数の EADs サーバを指定しておくことで, EADs サーバに接続できないときに, 別の EADs サーバに接続を試みることができます。

通信処理の仕組みについては「2.2 EADs の通信処理の仕組み」を参照してください。

図 8-1 通信に依存するパラメタの設計 (EADs クライアント)



### 8.3.2 通信に依存するパラメタ

通信に依存するパラメタの一覧を次の表に示します。

なお, 必ず指定するパラメタを太字で表記しています。

表 8-2 通信に依存するパラメタ一覧 (EADs クライアント)

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	クライアント定義ファイル	<b><code>eads.client.node.list</code></b>	接続先 EADs サーバを識別するため	なし

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
			の名称（任意の名称）	
2		<code>eads.client.&lt;接続先 EADs サーバ&gt;.address</code>	IP アドレスまたはホスト名	なし
3		<code>eads.client.&lt;接続先 EADs サーバ&gt;.port</code>	ポート番号	24600

## (1) クライアント定義ファイル

### (a) `eads.client.node.list`

EADs クライアントの初期設定時に接続する、接続先 EADs サーバを識別するための名称（任意の名称）を指定します。

名称には、半角英数字（0～9、A～Z、a～z）が指定できます。

複数指定する場合は、コンマ区切りで指定してください。複数指定する場合、EADs サーバへの初回接続時、EADs クライアントはこのパラメタに指定した EADs サーバからランダムに決定します。2 回目以降の接続では、EADs サーバから取得したクラスタ構成情報に従って接続先を選択します。

EADs サーバへの接続に失敗した場合は、接続に失敗した EADs サーバを基点として、指定された順に EADs サーバへの接続が試行されます。

### (b) `eads.client.<接続先 EADs サーバ>.address`

EADs クライアントの初期設定時に接続する、接続先 EADs サーバの IP アドレスまたはホスト名を指定します。

ホスト名で指定する場合は、ホスト名から IP アドレスが一意に識別できるホスト名を指定してください。

`eads.client.node.list` パラメタと対で指定します。接続先 EADs サーバには、`eads.client.node.list` パラメタに指定した EADs サーバを識別するための名称を指定してください。

### (c) `eads.client.<接続先 EADs サーバ>.port`

EADs クライアントの初期設定時に接続する、接続先 EADs サーバのポート番号を指定します。

`eads.client.node.list` パラメタと対で指定します。接続先 EADs サーバには、`eads.client.node.list` パラメタに指定した EADs サーバを識別するための名称を指定してください。

### (d) `eads.client.node.list` パラメタの指定例

`eads.client.node.list` パラメタの指定例を次に示します。

```
eads.client.node.list=sv1,sv2,sv3
eads.client.sv1.address=XXX.XXX.X.138
```

```
eads.client.sv1.port=24600  
eads.client.sv2.address=XXX.XXX.X.139  
eads.client.sv2.port=24601  
eads.client.sv3.address=XXX.XXX.X.140  
eads.client.sv3.port=24602
```

## 8.4 ログファイルに依存するパラメタの設計

ログファイルに依存するパラメタを設計します。

### 8.4.1 ログファイルの種類

EADs クライアントが管理するログファイルの種類を次の表に示します。

表 8-3 ログファイルの種類 (EADs クライアント)

項番	ログファイル	説明
1	メッセージログファイル	動作確認やエラー監視用のメッセージログを出力するファイルです。
2	通信トレースファイル	通信に関するトレース情報を出力するファイルです。
3	保守ログファイル	システムが使用するファイルです。 ユーザが設定するパラメタはありません。

### 8.4.2 ファイルの出力先の設定

ログファイルの出力先は変更できます。ファイルの出力先を次の表に示します。

表 8-4 ファイルの出力先の設定 (EADs クライアント)

ログファイル名	出力先	ファイル名
メッセージログファイル	< eads.logger.dir パラメタに指定したディレクトリ>/< EADs クライアント名 * >	eads_client_message[n].log
通信トレースファイル		eads_client_comm_trace[n].log
保守ログファイル	< eads.logger.dir パラメタに指定したディレクトリ>/< EADs クライアント名 * >/maintenance	eads_client_maintenance[n].log

(凡例)

[n]：ファイル数の通番です。

注※

クライアント API で指定した EADs クライアント名です。EADs クライアント名が空文字列の場合、EADs クライアント名のサブディレクトリを省略して出力します。

#### 注意事項

ファイルの出力先に、ネットワークで接続された別マシンのディレクトリは指定できません。

## 8.4.3 ファイルサイズおよびファイル数の設定

ログファイルのファイルサイズおよびファイル数は、基本的にデフォルト値を設定してください。EADs クライアント構築後、必要に応じてパラメタの指定値を変更してください。

## 8.4.4 ログファイルに依存するパラメタ

ログファイルに依存するパラメタの一覧を次の表に示します。

なお、必ず指定するパラメタを太字で表記しています。

表 8-5 ログファイルに依存するパラメタ一覧 (EADs クライアント)

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	クライアント定義 ファイル	<b>eads.logger.dir</b>	パス名	なし
2		eads.logger.message.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
3		eads.logger.message.filenum	ファイル数 (1～ 16)	2
4		eads.logger.message.console.enable	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	false
5		eads.logger.commTrace.enable	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	false
6		eads.logger.commTrace.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	4194304 バイト
7		eads.logger.commTrace.filenum	ファイル数 (1～ 16)	2
8		eads.logger.commTrace.commNo.prefix	プレフィックス (0～127)	0

### (1) クライアント定義ファイル

#### (a) eads.logger.dir

ログファイルの出力先ディレクトリのパスを指定します。

指定されたパスが存在しない場合は作成します。

このパラメタに指定したディレクトリ下に出力されるログファイルについては、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。

なお、複数プロセスで同じ出力先を同時に使用しないでください。

### (b) `eads.logger.message.filesize`

メッセージログの 1 ファイル当たりのファイルサイズ（単位：バイト）を指定します。

### (c) `eads.logger.message.filenum`

メッセージログのファイル数を指定します。

### (d) `eads.logger.message.console.enable`

メッセージログの標準出力への出力を有効にするか、無効にするかを指定します。

true :

標準出力への出力を有効にします。

false :

標準出力への出力を無効にします。

### (e) `eads.logger.commTrace.enable`

通信トレースの出力を有効にするか、無効にするかを指定します。

true :

通信トレースの出力を有効にします。

false :

通信トレースの出力を無効にします。

通信トレースの出力を有効にすると、保守情報を取得できます。ただし、性能に影響を与えるおそれがあります。

## ■ 注意事項

通信トレースの出力を無効にする場合、障害発生時の原因調査ができないおそれがあります。

例えば、EADs クライアント・EADs サーバ間で通信障害が発生した場合、通信時の情報がないため、EADs クライアントと EADs サーバのどちらに原因があるのか特定できません。

### (f) `eads.logger.commTrace.filesize`

通信トレースの 1 ファイル当たりのファイルサイズ（単位：バイト）を指定します。

`eads.logger.commTrace.enable` パラメタが `false` の場合、この指定は無効となります。

### (g) eads.logger.commTrace.filenum

通信トレースのファイル数を指定します。

`eads.logger.commTrace.enable` パラメタが `false` の場合、この指定は無効となります。

### (h) eads.logger.commTrace.commNo.prefix

PRF トレースに出力される RootAP 情報の通信番号 (RootAP CommNo.) に含めるプレフィックスを指定します。

1 プロセス内に複数のクラスローダを持ち、それぞれでライブラリを使用する場合、RootAP 情報が一意となるように指定します。

RootAP 情報の通信番号は 16 桁の 16 進数です。プレフィックスは 9 桁目から 10 桁目に該当します。

(例)

00000000XXYYYYYY

XX：プレフィックス

YYYYYY：000001 から FFFFFFFF までの値

`eads.logger.commTrace.enable` パラメタが `false` の場合、この指定は無効となります。

# 9

## チューニングパラメタを設計する

この章では、チューニングパラメタの設計指針について説明します。

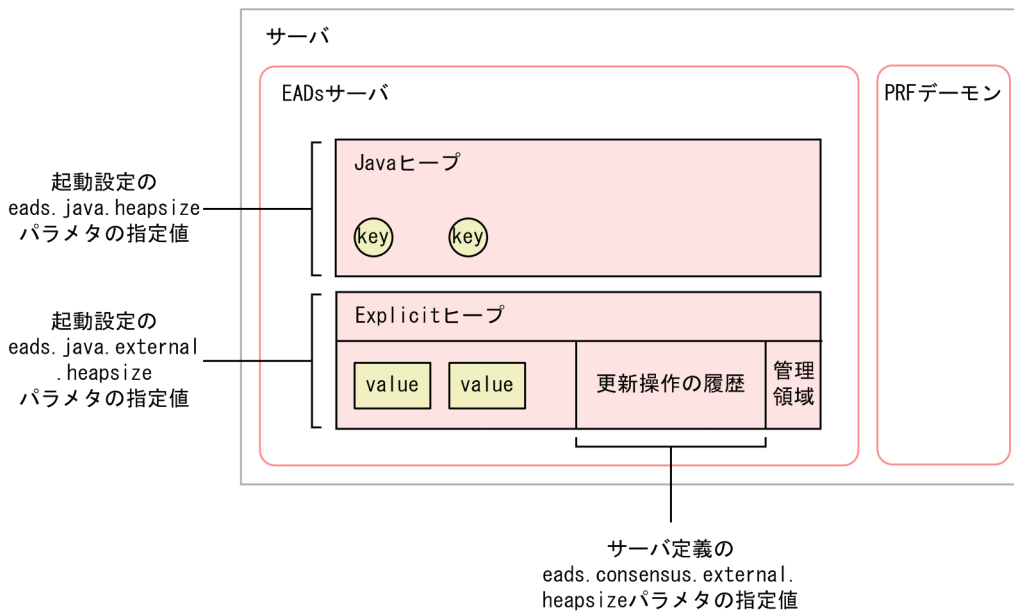


## 9.1 メモリ・バッファに関連するパラメタの設計

メモリ・バッファに関連するパラメタを設計します。

### 9.1.1 メモリサイズの設定

EADs サーバが使用するメモリの構成を次の図に示します。



Explicit ヒープは、次の領域から構成されます。

- 更新操作の履歴が格納される領域
- メモリキャッシュ、および 2Way キャッシュの場合に、key と value のうち、value 部分を格納するための領域

更新操作の履歴とは、API の種別、および key と value の情報のことです。更新操作の履歴はデータを多重化の際、データの整合性を確保するために使用されます。

合意処理中に障害が発生した場合、障害が発生した EADs サーバの代わりに、稼働している別の EADs サーバがこの履歴を基に合意処理に参加します。履歴は前の処理の合意および書き込み処理が完了するまで保持されます。

更新操作の履歴は、EADs サーバが不要と判断した時点で自動的に削除されます。

### 更新操作の履歴

書き込み順序	1	2	3	4	
	合意済み	未合意	合意済み	合意済み	...

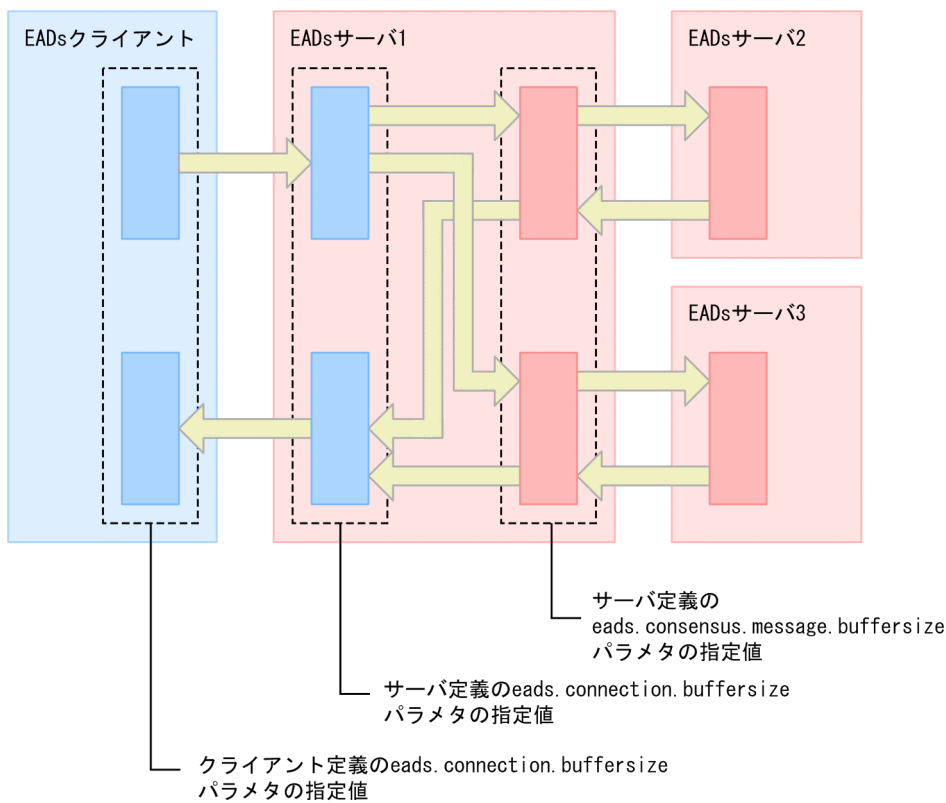
2番目の更新操作が合意され、書き込みが完了されるまで保持

### 考え方

「4.1.2 Java ヒープサイズを見積もる」, および「4.1.3 Explicit ヒープサイズを見積もる」を参照し、1EADs サーバ当たりのJava ヒープサイズと Explicit ヒープサイズを決定します。

## 9.1.2 バッファサイズの設定

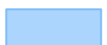
EADs サーバが通信で使用するバッファの概要を次の図に示します。



(凡例)



: データの流れ

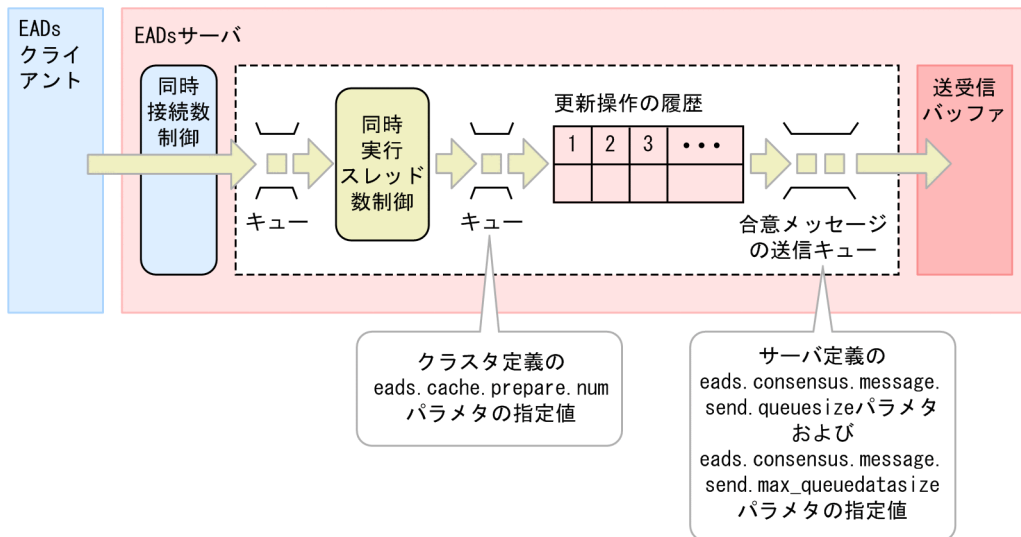


: データの送受信バッファ



: 合意メッセージの送受信バッファ

EADs クライアントから送信されたリクエストは、EADs サーバ内で次のように制御されます。



EADS クライアントから送信されたリクエスト（データの更新操作）は、履歴に保持され、合意メッセージの送信キューで待機します。合意メッセージの送信キューは、合意メッセージの送信先となる EADS サーバの数だけ確保されます。

また、合意メッセージの送信キューの長さを、サーバ定義の `eads.consensus.message.send.queue.size` パラメタで、送信キューに格納できるデータの最大サイズを、サーバ定義の `eads.consensus.message.send.max_queuedata.size` パラメタで設定できます。

なお、クラスタ定義の `eads.cache.prepare.num` パラメタで、一度に合意できる処理の最大数を設定することもできます。

最大同時接続数および最大同時実行スレッド数の設定については、「9.2 スレッドプールおよびコネクションプールに関連するパラメタの設計」を参照してください。

### 考え方

扱うデータ量に合わせてバッファサイズを調節することで、効率良くデータを送受信できます。

## 9.1.3 メモリ・バッファに関連するパラメタ

メモリ・バッファに関連するパラメタの一覧を次の表に示します。

表 9-1 メモリ・バッファに関連するパラメタ一覧

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	サーバ定義ファイル	<code>eads.connection.buffer.size</code>	バッファサイズ (1024 バイト～ 16777216 バイト)	4096 バイト
2		<code>eads.server.max.data.size.enable</code>	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	false

項番	定義ファイル	パラメタ名	指定する値	デフォルト値	
3		eads.server.max.datasize	1024 バイト～ 67108864 バイト	67108864 バイト	
4		eads.consensus.message.bufferSize	バッファサイズ (1024 バイト～ 16777216 バ イト)	131071 バイト	
5		eads.consensus.message.send.queueSize	キューの長さ (10000～ 1000000)	100000	
6		eads.consensus.message.send.max_queue datasize	1048576 バイ ト～2147483647 バイト	16777216 バイト	
7		eads.restore.connection.bufferSize	バッファサイズ (1024 バイト～ 16777216 バ イト)	131071 バイト	
8		eads.restore.dataSender.sendSize	10240 バイト～ 2147483647 バ イト	1048576 バイト	
9		eads.consensus.fillgap.copy.dataSize	1024 バイト～ 16777216 バイト	10240 バイト	
10		eads.consensus.external.heapSize	1 メガバイト～ 268435456 メガ バイト	450 メガバイト	
11		eads.management.resume.sendSize	0 バイト～ 2147483647 バ イト	1048576 バイト	
12		クラスタ定義ファ イル	eads.server.max_key_size	1 バイト～1024 バイト	1024 バイト
13			eads.cache.prepare.num	1～100	10
14	起動設定ファイル	eads.prf.bufferSize	バッファサイズ (512 キロバイト～ 102400 キロバ イト)	8192 キロバイト	
15		eads.java.heapSize	ヒープサイズ※1	3072 メガバイト	
16		eads.java.external.heapSize	ヒープサイズ (2 メガバイト～ 2147483647 メガ バイト)	1024 メガバイト	
17		eads.java.permanent.size	メモリサイズ※2	83 メガバイト	

## 9. チューニングパラメタを設計する

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
18	クライアント定義ファイル	<code>eads.connection.bufferSize</code>	バッファサイズ (1024 バイト ~ 16777216 バイト)	4096 バイト

#### 注※1

指定した値が、JavaVM のヒープサイズオプション (-Xmx および-Xms) に適用されます。

#### 注※2

指定した値が、JavaVM のメモリサイズオプション (-XX:PermSize および-XX:MaxPermSize) に適用されます。

## (1) サーバ定義ファイル

### (a) `eads.connection.bufferSize`

データの送受信バッファのサイズ (単位: バイト) を指定します。

### (b) `eads.server.max.datasize.enable`

EADs サーバが受け付けるデータの最大サイズのチェックを有効にするか、無効にするかを指定します。

true :

データの最大サイズのチェックを有効にします。

false :

データの最大サイズのチェックを無効にします。

### (c) `eads.server.max.datasize`

EADs サーバが受け付けるデータの最大サイズ (単位: バイト) を指定します。

`eads.server.max.datasize.enable` パラメタが false の場合、この指定は無効となります。

### (d) `eads.consensus.message.bufferSize`

合意メッセージの送受信バッファサイズ (単位: バイト) を指定します。

OS で設定された TCP のウィンドウサイズを指定することを推奨します。

なお、OS によって、このパラメタに指定した値と異なるサイズのバッファが使用されることがあります。

### (e) `eads.consensus.message.send.queueSize`

合意メッセージの送信キューの長さを指定します。

送信キューは、合意メッセージの送信先となる EADs サーバの数だけ確保されます。

次の計算式から見積もってください。見積もった結果、このパラメタの最小値よりも小さい場合は、最小値を設定してください。

$$2 \times a \times b + 2 \times a \times \text{キャッシュ数} + (c - 1) \times a \times \text{キャッシュ数}$$

- a: クラスタ定義の `eads.cache.prepare.num` パラメタの指定値
- b: サーバ定義の `eads.cluster.controller.cache.max_execute_threads` パラメタの指定値
- c: クラスタ定義の `eads.replication.factor` パラメタの指定値

### 注意事項

設定した送信キューの長さを超える量の合意メッセージが送信キューに入った場合、通信エラーになります。

## (f) `eads.consensus.message.send.max_queuedatasize`

合意メッセージの送信キューに格納できるデータの最大サイズ（単位：バイト）を指定します。

次の計算式から見積もってください。

$$2 \times a \times b \times (c + \text{キャッシュ数}) + d \times (e - 1) \times \text{キャッシュ数}$$

- a: クラスタ定義の `eads.cache.prepare.num` パラメタの指定値
- b: 更新操作の履歴の最大サイズ  
更新操作の履歴の最大サイズ（単位：バイト）を見積もる計算式を次に示します。  
クラスタ定義の `eads.server.max_key_size` パラメタの指定値 + value の最大サイズ<sup>※</sup> × 2  
注※ put, create, update, replace 実行時に指定する value の最大サイズです。
- c: サーバ定義の `eads.cluster.controller.cache.max_execute_threads` パラメタの指定値
- d: サーバ定義の `eads.consensus.fillgap.copy.dataSize` パラメタの指定値
- e: クラスタ定義の `eads.replication.factor` パラメタの指定値

### 注意事項

見積もった結果に対して非常に小さい値を設定した場合、送信キューあふれによるリトライが頻発し、処理が継続できなくなるおそれがあります。

## (g) `eads.restore.connection.buffersize`

復旧処理で使用するデータの送受信バッファサイズ（単位：バイト）を指定します。

OS で設定された TCP のウィンドウサイズを指定することを推奨します。

なお、OS によって、このパラメタに指定した値と異なるサイズのバッファが使用されることがあります。

## (h) `eads.restore.dataSender.sendSize`

復旧処理で送信するデータサイズ（単位：バイト）を指定します。

復旧処理では、データの整合性を回復するために、稼働中の EADs サーバが復旧対象の EADs サーバにデータを送信します。`eads.restore.dataSender.sendInterval` パラメタに指定した間隔で、このパラメタの指定値を超えるまで、10 キロバイト単位でデータを連続して送信します。

復旧処理については、「[9.3.2\(5\) クラスタの復旧処理](#)」を参照してください。

ディスクキャッシュ、および 2Way キャッシュを使用する場合、このパラメタの指定は無効になり、キャッシュ定義の `eads.cache.restore.dataSender.sendSize` パラメタの指定値が有効となります。

### 注意事項

復旧処理では最低 1 個のデータを送信します。そのため、このパラメタに EADs サーバに格納しているデータのサイズよりも小さい値を指定しても、送信量を制限することはできません。

## (i) `eads.consensus.fillgap.copy.dataSize`

EADs サーバに対して、更新操作の履歴の補完処理でデータをコピーする際、1 回当たりに送信するデータサイズ（単位：バイト）を指定します。

更新操作の履歴の補完処理については、「[9.3.2\(6\) 更新操作の履歴の補完処理](#)」を参照してください。

次の計算式から見積もってください。

更新操作の履歴の最大サイズ × 合意処理で一度に合意できる処理の最大数

更新操作の履歴の最大サイズ：

更新操作の履歴の最大サイズ（単位：バイト）を見積もる計算式を次に示します。

クラスタ定義の `eads.server.max_key_size` パラメタの指定値 + value の最大サイズ※ × 2

注※ put, create, update, replace 実行時に指定する value の最大サイズです。

合意処理で一度に合意できる処理の最大数：

クラスタ定義の `eads.cache.prepare.num` パラメタの指定値

### 注意事項

- メモリ使用量や CPU 使用率に影響するため、必要以上に大きな値を設定しないでください。
- このパラメタの指定値を超えるサイズのデータが送信されることがあります。
- データサイズを小さく設定しても、最低 1 履歴分は補完処理が行われます。

## (j) `eads.consensus.external.heapsize`

更新操作の履歴が格納される領域のサイズ（単位：メガバイト）を指定します。

このパラメタの指定値が、起動設定の `eads.java.external.heapsize` パラメタの指定値×0.97（メガバイト単位で小数点以下は切り捨て）より大きい場合、EADs サーバの起動に失敗するので注意してください。

## (k) `eads.management.resume.sendSize`

`eztool resume` コマンド実行時に、1 回で転送する差分のデータサイズ（単位：バイト）を指定します。

0 を指定すると、1 回の通信でデータ 1 つを送信します。

## (2) クラスタ定義ファイル

### (a) `eads.server.max_key_size`

クラスタに格納できる key の最大サイズ（単位：バイト）を指定します。

このパラメタに指定したサイズで key の格納領域を確保します。

key の最大サイズを制限することで、デフォルトのまま使用するよりも、key が格納される Java ヒープのサイズを小さく設計することができます。

#### 注意事項

- key の最大サイズを大きく設定したクラスタで出力されたストアデータファイルは、それよりも小さく設定したクラスタでは読み込めないことがあります。
- ディスクキャッシュ、および 2Way キャッシュを使用する場合、key の最大サイズを大きく設定したクラスタで出力されたキャッシュデータファイルとキャッシュインデックスファイルは、それよりも小さく設定したクラスタでは、データの再配置に使用できないことがあります。

### (b) `eads.cache.prepare.num`

合意処理で、一度に合意できる処理の最大数を指定します。

次に示す値のうち、小さい方の値を設定してください。ただし、EADs サーバ間で値が異なる場合は、クラスタ内で最大の値を設定してください。

- サーバ定義の `eads.server.max_connections` パラメタの指定値
- サーバ定義の `eads.server.cache.max_execute_threads` パラメタの指定値 + サーバ定義の `eads.server.function_container.max_execute_threads` パラメタの指定値



## 注意事項

メモリ使用量に影響するため、必要以上に大きな値を設定しないでください。

### (3) 起動設定ファイル

#### (a) `eads.prf.buffersize`

PRF デーモンが共用メモリに確保するバッファのサイズ（単位：キロバイト）を指定します。

起動設定の `eads.prf.filesize` パラメタの指定値よりも大きな値は指定できません。

#### (b) `eads.java.heapsize`

key が格納される Java ヒープのサイズ（単位：メガバイト）を指定します。

#### (c) `eads.java.external.heapsize`

value および更新操作の履歴が格納される Explicit ヒープのサイズ（単位：メガバイト）を指定します。

指定した Explicit ヒープサイズの 3%（メガバイト単位で小数点以下は切り上げ）は管理領域として使用されます。

サーバ定義の `eads.consensus.external.heapsize` パラメタの指定値が、このパラメタの指定値×0.97（メガバイト単位で小数点以下は切り捨て）より大きい場合、EADs サーバの起動に失敗するので注意してください。

#### (d) `eads.java.permanent.size`

Permanent 領域のサイズ（単位：メガバイト）を指定します。

Permanent 領域は、ロードされた EADs サーバ、ユーザファンクション、ユーザイベントハンドラの class などが格納される領域です。

### (4) クライアント定義ファイル

#### (a) `eads.connection.buffersize`

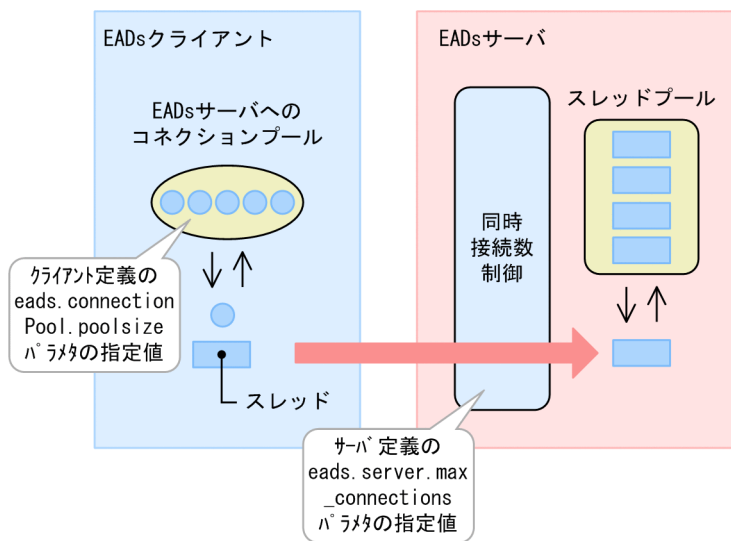
データの送受信バッファのサイズ（単位：バイト）を指定します。

## 9.2 スレッドプールおよびコネクションプールに関連するパラメタの設計

同時に実行されるリクエスト処理数に応じて、スレッドプールおよびコネクションプールに関連するパラメタを設計します。

### 9.2.1 最大同時接続数の設定

EADs では、スレッドプールおよびコネクションプールを使用することで、スレッドやコネクションの生成に掛かるオーバーヘッドを軽減し、スループットの向上を図ります。スレッドプールおよびコネクションプールについては、「2.13 スレッドプールおよびコネクションプールを使用したスループットの向上」を参照してください。



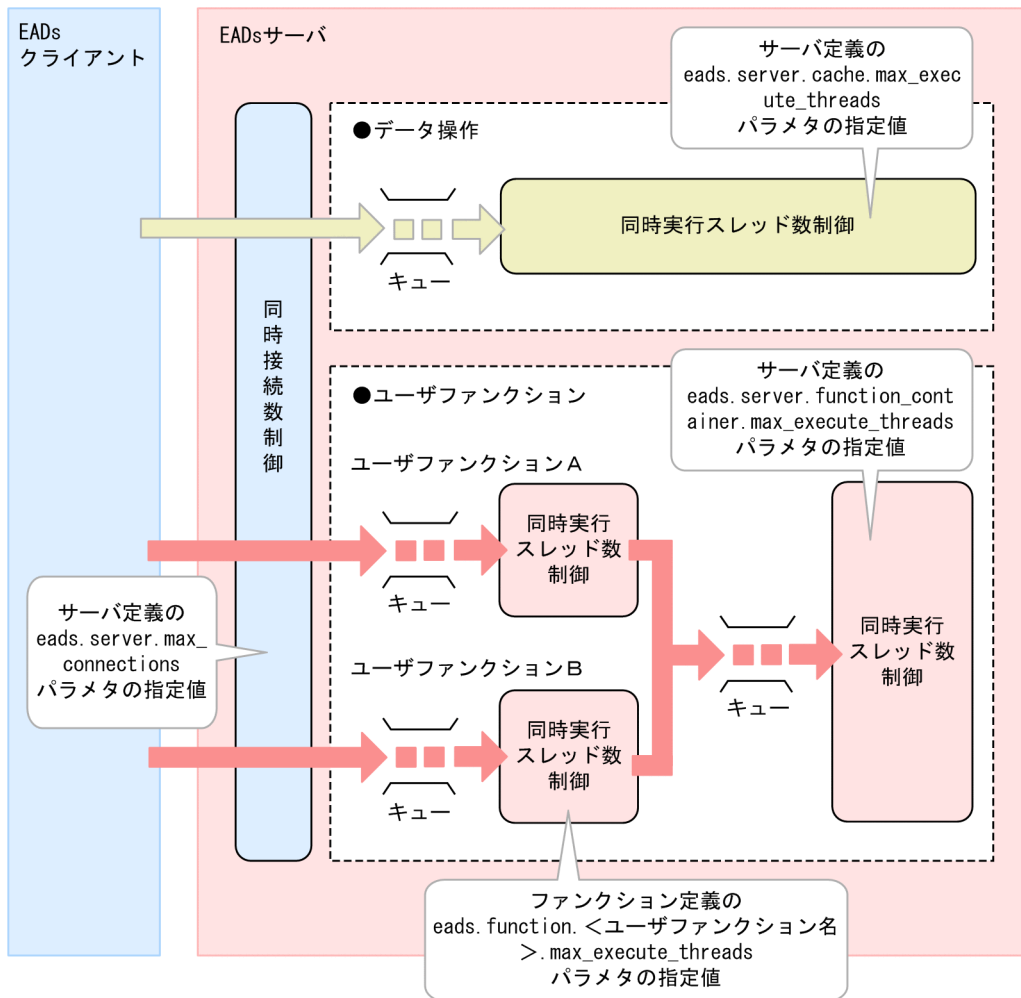
#### 考え方

最大同時接続数（サーバ定義の `eads.server.max_connections` パラメタの指定値）を調節することで、リソースの使用効率を改善できます。

### 9.2.2 最大同時実行スレッド数の設定

EADs サーバは、EADs クライアントからの同時接続数および同時実行スレッド数を制御します。

同時実行スレッド数については、EADs サーバに対するリクエストをデータ操作とユーザファンクションに分けて制御します。



- データ操作の同時実行スレッド数  
サーバ定義の `eads.server.cache.max_execute_threads` パラメタで指定します。  
EADS クライアント・EADS サーバ間の接続を切断するまでの間（サーバ定義の `eads.server.session.keepAlive.timeout` パラメタの指定値）、最大同時実行スレッド数を超えるリクエストはキューに待機します。
- ユーザ関数の同時実行スレッド数  
ユーザ関数ごとの同時実行スレッド数（ファンクション定義の `eads.function.<ユーザファンクション名>.max_execute_threads` パラメタ）を制御し、その上でユーザ関数全体の同時実行スレッド数（サーバ定義の `eads.server.function_container.max_execute_threads` パラメタ）を制御します。

## 考え方

最大同時実行スレッド数を調節することで、リソースの使用効率を改善できます。

## 9.2.3 スレッドプールおよびコネクションプールに関連するパラメタ

スレッドプールおよびコネクションプールに関連するパラメタの一覧を次の表に示します。

表 9-2 スレッドプールおよびコネクションプールに関連するパラメータ一覧

項番	定義ファイル	パラメータ名	指定する値	デフォルト値
1	サーバ定義ファイル	<code>eads.server.max_connections</code>	1～1024	10
2		<code>eads.server.cache.max_execute_threads</code>	1～ <code>eads.server.max_connections</code> パラメータの指定値	<code>eads.server.max_connections</code> パラメータの指定値
3		<code>eads.server.function_container.max_execute_threads</code>	1～ <code>eads.server.max_connections</code> パラメータの指定値	<code>eads.server.max_connections</code> パラメータの指定値
4		<code>eads.cluster.controller.cache.max_execute_threads</code>	0～1024	0
5	ファンクション定義ファイル	<code>eads.function.&lt;ユーザファンクション名&gt;.max_execute_threads</code>	0～ <code>eads.server.function_container.max_execute_threads</code> パラメータの指定値	0
6	クライアント定義ファイル	<code>eads.connectionPool.poolsize</code>	1～1024	10
7		<code>eads.connectionPool.exceedMaxSizeError.enable</code>	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	false

## (1) サーバ定義ファイル

### (a) `eads.server.max_connections`

EADs サーバへの最大同時接続数を指定します。

最大同時接続数を超える接続に対しては、エラーを返却し、通信を切断します。

接続する EADs クライアントのクライアント定義に設定した `eads.connectionPool.poolsize` パラメータの値の総和以上を設定してください。

なお、`eads.connectionPool.poolsize` パラメータに設定した値は Listen キューの backlog にも設定されます。設定値が OS の制限値を超えた場合は、OS の制限値が設定されます。

クライアント定義の `eads.connectionPool.poolsize` パラメータに設定する値を考慮して、OS の制限値を拡張してください。制限値を拡張する方法については、OS のマニュアルを参照してください。

### (b) `eads.server.cache.max_execute_threads`

データ操作の最大同時実行スレッド数を指定します。

### (c) `eads.server.function_container.max_execute_threads`

ユーザファンクション全体の最大同時実行スレッド数を指定します。

### (d) `eads.cluster.controller.cache.max_execute_threads`

キャッシュ単位に行われる処理の最大同時実行スレッド数を指定します。

EADs サーバの縮退処理、復旧処理、および排他制御は、キャッシュ単位に行われるため、キャッシュ数が増えるほど実行数も多くなります。

このパラメタの指定値を大きくすると、これらの処理の並列実行性を向上させ、EADs サーバの縮退処理や復旧処理に掛かる時間を軽減できます。ただし、指定値を大きくし過ぎると、リソースを多く消費し、ほかの処理に影響を与えるおそれがあるため、注意してください。

キャッシュが1つの場合は、基本的にデフォルト値を設定してください。

キャッシュが複数で、並列実行性を向上させたい場合は、(データの多重度-1) × キャッシュ数を設定してください。

#### ■ 注意事項

0 を指定した場合、データの多重度-1 の値が設定されます。ただし、データの多重度が1 の場合は、1 が設定されます。

## (2) ファンクション定義ファイル

### (a) `eads.function.<ユーザファンクション名>.max_execute_threads`

ユーザファンクション単位の最大同時実行スレッド数を指定します。

ユーザファンクション名には、完全修飾クラス名を指定します。

0 を指定した場合は無制限となります。

## (3) クライアント定義ファイル

### (a) `eads.connectionPool.poolsize`

同一接続先に対してプールしておく、接続の最大個数を指定します。

指定した個数まで、接続を確立できます。

クライアント API を同時に実行するスレッド数の値を設定してください。

## (b) `eads.connectionPool.exceedMaxSizeError.enable`

同一接続先に対してプールしておくコネクションがすでに最大個数に達していて、すべて使用中の場合に、EADs クライアントからのリクエストをエラーにするかどうかを指定します。

true :

クライアント API をエラーにします。

false :

クライアント API をエラーにしません。クライアント API は処理待ちとなります。

## 9.3 タイムアウトに関連するパラメタの設計

タイムアウトに関連するパラメタを設計します。

### 9.3.1 通信を監視するタイマの設定

TCP プロトコルを使用する EADs クライアント・EADs サーバ間の通信では、次に示す時間を監視することによって、通信障害を検知しています。

- ソケットによる接続開始から接続完了までの時間
- データの書き込み開始から書き込み完了までの時間
- データの読み込み開始から読み込み完了までの時間

データの読み込みについては、1つの電文を受信する際に複数回発生することがあります。

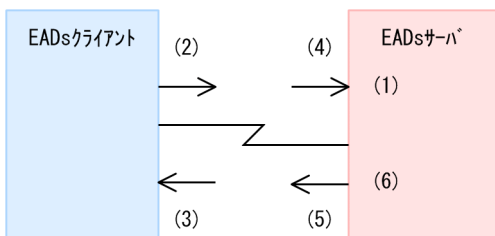
#### 考え方

監視時間を短くすることで通信障害を検知するスピードを早めたり、長くすることで頻繁にタイムアウトが発生するのを防いだりします。

#### (1) EADs クライアント・EADs サーバ間での通信タイムアウトの設定

EADs クライアント・EADs サーバ間の通信では、次の図に示す通信タイムアウトが設定できます。

図 9-1 EADs クライアント・EADs サーバ間での通信タイムアウトの設定



通信タイムアウトの設定で使用するパラメタを次の表に示します。表の項番は、図 9-1 の(1)～(6)に対応しています。

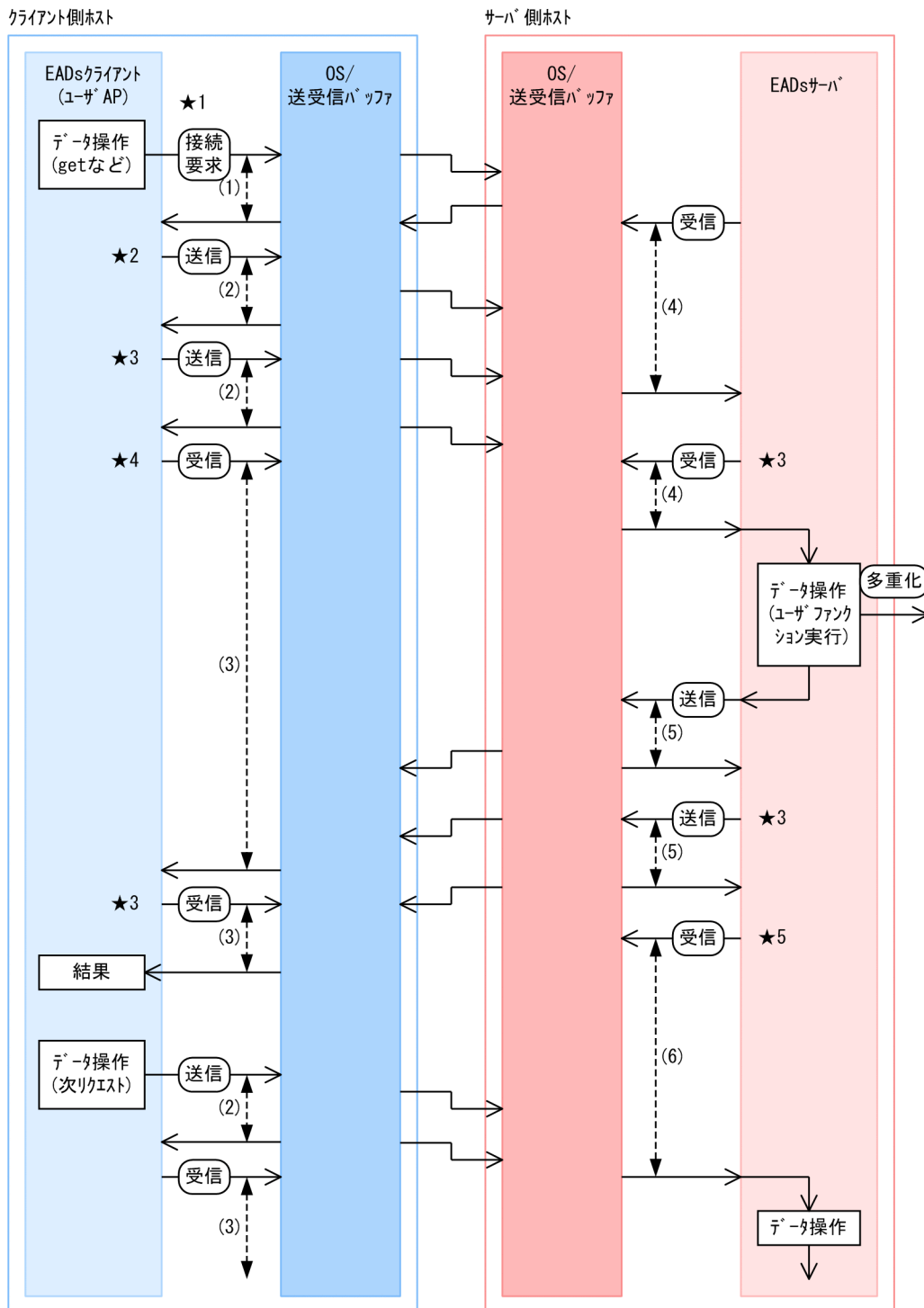
表 9-3 通信タイムアウトの設定で使用するパラメタ

項番	設定するタイムアウト	定義ファイル	パラメタ名
1	EADs サーバへの接続	クライアント定義ファイル	<code>eads.connection.send.timeout</code>
2	EADs サーバへのデータ送信		
3	EADs サーバからのデータ受信		<code>eads.connection.receive.timeout</code>
4	EADs クライアントからのデータ受信	サーバ定義ファイル	<code>eads.connection.timeout</code>

項番	設定するタイムアウト	定義ファイル	パラメタ名
5	EADs クライアントへのデータ送信		
6	通信のない常設接続の切断		<code>eads.server.session.keepAlive.timeout</code>

EADs クライアント・EADs サーバ間の通信で、タイムアウトが設定できる個所を次の図に示します。

図 9-2 EADs クライアント・EADs サーバ間の通信で、タイムアウトが設定できる個所





各タイムアウト（図中の(1)～(6)）に設定する値の考え方については、次の個所で説明します。

- 「9.3.1(2) 通信タイムアウトに設定する値の考え方」
- 「9.3.1(3) 常設コネクション切断までのタイムアウト値の設定」

また、ポイントとなる項目（★1～★5）については、「9.3.1(4) タイムアウトに設定する値を考慮する際のポイント」で説明します。

## (2) 通信タイムアウトに設定する値の考え方

通信タイムアウト（図 9-2 の(1)～(5)）は、設定したタイムアウト時間内に処理が完了することを期待するものではなく、次のような状態を検知するための機能です。

- 通信路の物理的な切断
- 通信先の障害などによる、通信先との通信路の切断

そのため、期待する時間内に応答がない場合でも、通信タイムアウトの機能で通信を切断することは推奨しません。

EADs サーバの処理は、EADs クライアントとの通信路の状態とは関係なく進みます。EADs クライアントへの応答時に通信路の切断を検知しても、データのロールバックは行いません。そのため、通信タイムアウトの機能で通信を切断してしまうと、EADs クライアントでは、データ操作の結果がわからなくなります。

処理の遅延やネットワークの遅延ではなく、通信先の障害や通信路の切断などを正しく検知できるように、通信タイムアウトの時間をチューニングすることを推奨します。

## (3) 常設コネクション切断までのタイムアウト値の設定

常設コネクション切断までのタイムアウト値の設定（図 9-2 の(6)）は、AP（ユーザプログラム）が動作する OS、またはホストがダウンした場合に EADs サーバが気づくことができない場合に、OS に設定された TCP の keep-alive のアイドル時間までコネクション、およびスレッドが占有されることを防ぐためのものです。

設定する際は次の 3 点を考慮して、ほかの機能に問題のない範囲で、かつむだな通信の切断やエラーを避けられるように、大きめの値を設定することを推奨します。

### • EADs クライアント側のコネクション数との関係

EADs クライアントは AP から一定の期間 EADs サーバへの操作がないときでも常設コネクションを保持します。常設コネクションはクラスタ構成情報の確認処理のために定期的に使用されます。1 つの EADs サーバに対する常設コネクション数は最大で、クライアント定義の `eads.connectionPool.poolsize` パラメタで指定した数となります。また、各 EADs サーバに対してクライアント定義の `eads.client.clusterInfo.update.interval` パラメタに指定した間隔で常設コネクションを使用します。

クラスタ構成情報の確認は 1 スレッドで実行しているため、OS が正常に動作していないホストのアドレスに対して、クラスタ構成情報の確認処理を行うと、最大でクライアント定義の `eads.connection.send.timeout` パラメタに設定した時間の待ちが発生し、その間、ほかの EADs サーバに対するクラスタ構成情報の確認処理で通信が滞ります。

上記の場合で、かつ EADs クライアントからのリクエストがないときに、常設コネクションの切断を防ぐために、サーバ定義の `eads.server.session.keepAlive.timeout` パラメタの値には、次の関係を保つことができる、十分な時間を設定する必要があります。

サーバ定義の `eads.server.session.keepAlive.timeout` パラメタの指定値

> (クライアント定義の `eads.connection.send.timeout` パラメタの指定値 × [許容する EADs サーバの障害数]  
+ クライアント定義の `eads.client.clusterInfo.update.interval` パラメタの指定値)  
× クライアント定義の `eads.connectionPool.poolsize` パラメタの指定値

なお、EADs サーバプロセスがダウンしても OS に問題がない場合、上記の `eads.connection.send.timeout` パラメタに指定した分の待ち時間は発生しません。そのため、許容する EADs サーバの障害数は、プロセス障害ではなく、ホスト障害などの許容範囲を考慮してください。

#### • EADs サーバ側の接続最大数との関係

サーバ定義の `eads.server.max_connections` パラメタを指定すると、EADs クライアントからの接続数が制限され、設定した値を超えた接続はエラーとなります。

必要最小限の値をこのパラメタに設定している場合、ホストや OS の異常によって EADs サーバ側のネットワークで検知できないネットワーク切断が発生すると、サーバ定義の `eads.server.session.keepAlive.timeout` パラメタに設定した時間まで EADs サーバの接続が保持されることによって、EADs クライアント側で障害回復後の再接続ができなくなるおそれがあります。この問題を解決するためには、次の式を満たすように、パラメタを設定する必要があります。

サーバ定義の `eads.server.session.keepAlive.timeout` パラメタの指定値

< [障害発生から EADs クライアントが再接続までに必要とする時間]

ただし、障害の内容によって対応に掛かる時間が大きく変わるため、メモリに余裕がある場合は、再接続に使用する数を `eads.server.max_connections` パラメタにあらかじめ加算し、`eads.server.session.keepAlive.timeout` パラメタには 3600 秒など余裕を持った値を設定することを推奨します。

## (4) タイムアウトに設定する値を考慮する際のポイント

タイムアウトに設定する値を考慮する際に、ポイントとなる項目について説明します。なお、このポイントは図 9-2 の★1～★5 に対応しています。

### ポイント 1 (★1)

EADs クライアントからの通信は、通常、常設コネクションを使用します。そのため、初回接続時や、クライアントライブラリを使用するスレッドが増加した場合にだけ接続処理が発生します。接続タイムアウトには送信タイムアウトと同じ値（クライアント定義の `eads.connection.send.timeout` パラメタの値）が使用されます。

## ポイント 2 (★2)

データ送信処理は送信側の送信バッファにデータを格納した時点で送信完了とみなされます。そのため、送信バッファに収まるサイズの送信処理では、受信側プロセスや通信路の状態に関係なく、データの送信処理が成功します（通信路に問題がある場合は、次の受信処理でエラーになります）。

送信バッファより大きいサイズのデータを送信する場合、FullGC など受信側プロセスの問題で送信エラーが発生することがあります。送信バッファのサイズを超えるような大きなデータを送信する場合は、受信側プロセスの FullGC などに掛かる時間を考慮する必要があります。

## ポイント 3 (★3)

送信処理、受信処理は、内部では複数回に分けて実行されます。その際、それぞれの処理でタイムアウト値が適用されます。そのため、タイムアウト値は応答が戻るまでの時間を保証するものではありません。

## ポイント 4 (★4)

EADs クライアントは、EADs クライアント側の送信処理が終わった時点から次の読み込み処理に移ります。このとき、送信したデータを EADs サーバが受信しているかどうかは考慮されません。

また、EADs クライアントの受信タイムアウトに含まれる時間には、EADs サーバ上のデータ操作やユーザ関数の実行時間が含まれます。処理に時間が掛かるユーザ関数が存在する場合、ユーザ関数の処理時間に合わせてクライアント定義を設定すると、通常の通信には適さない値となるおそれがあります。そのような場合には、クライアント定義でタイムアウト値を設定するのではなく、タイムアウト付きの API の使用を推奨します。

## ポイント 5 (★5)

EADs クライアントへのレスポンス送信を終えた EADs サーバは、次のリクエスト待ちとなります。この受信処理には、通常の通信タイムアウト（サーバ定義の `eads.connection.timeout` パラメタで指定した値）ではなく、常設コネクションのタイムアウト（サーバ定義の `eads.server.session.keepAlive.timeout` パラメタで指定した値）が適用されます。常設コネクションのタイムアウトの設定値については、「[9.3.1\(3\) 常設コネクション切断までのタイムアウト値の設定](#)」を参照してください。

## 9.3.2 クラスタを監視するタイマの設定

クラスタ内の EADs サーバは、互いにハートビートを送信し合い、正常に稼働していることをクラスタ内に知らせています。

また、コマンド実行時のクラスタ構成情報の更新確認や、コマンド実行開始から終了までに掛かる時間を監視することによって、通信障害を検知しています。

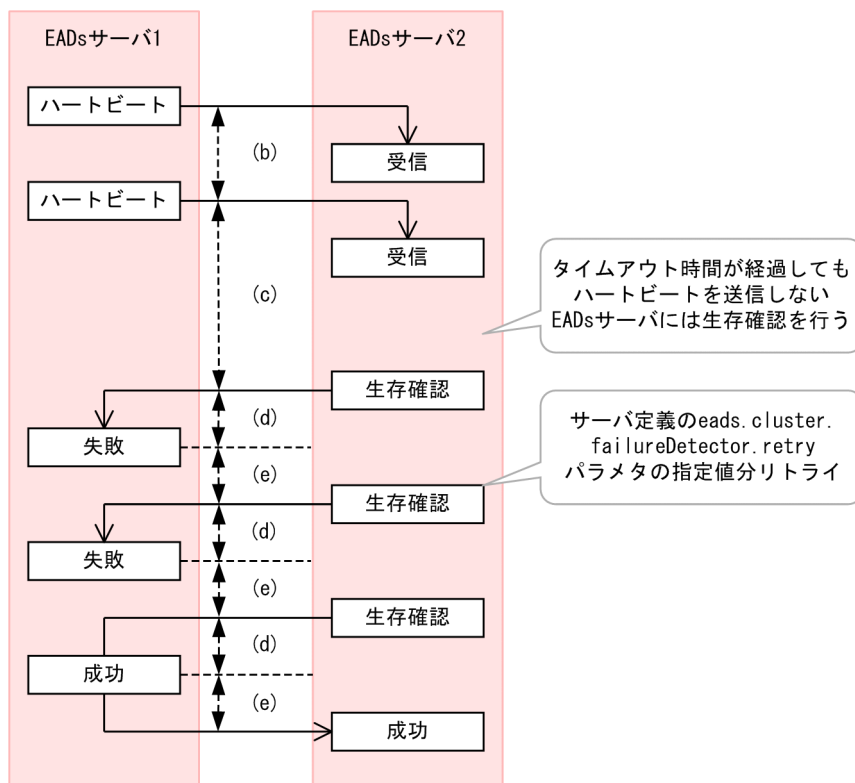
### 考え方

監視時間を短くすることで通信障害を検知するスピードを早めたり、長くすることで頻りにタイムアウトが発生するのを防いだりします。

## (1) ハートビートの送信, および生存確認

ハートビートの送信によるクラスタ監視については、「2.10 クラスタ監視」を参照してください。

ハートビートの送信, および生存確認のタイマを次の図に示します。



図中のアルファベットは、「9.3.3 タイムアウトに関連するパラメタ」の説明と次のように対応しています。

(b) : 「9.3.3(1)(b) `eads.cluster.heartbeat.interval`」

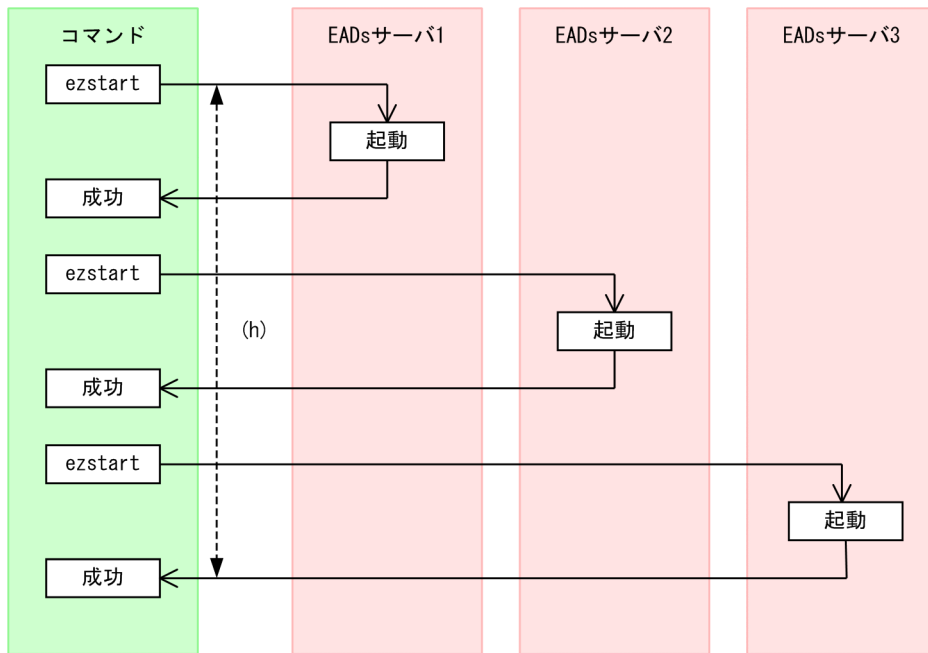
(c) : 「9.3.3(1)(c) `eads.cluster.heartbeat.timeout`」

(d) : 「9.3.3(1)(d) `eads.cluster.failureDetector.connect.timeout`」

(e) : 「9.3.3(1)(e) `eads.cluster.failureDetector.read.timeout`」

## (2) クラスタの開始

`ezstart` コマンドを実行してクラスタを開始する場合を例に, EADs サーバ開始時のタイマを次の図に示します。



図中のアルファベットは、「9.3.3 タイムアウトに関連するパラメタ」の説明と次のように対応しています。

(h) : 「9.3.3(1)(h) eads.cluster.boot.timeout」

なお、クラスタ内では、クラスタ定義で設定した EADs サーバ ID がいちばん小さい EADs サーバから起動します。以降、順番に起動した EADs サーバは、最初に起動した EADs サーバからのハートビートを受信し、クラスタに参加していきます。

最初に起動した EADs サーバは、ほかの EADs サーバから受信したハートビートを基に、クラスタ構成情報を更新します。更新したクラスタ構成情報は、クラスタ内で共有します。

### (a) クラスタ定義の内容がほかの EADs サーバと異なる場合

すでに起動している EADs サーバは、ハートビートにクラスタ定義の内容をハッシュ化した値を付与して送信します。

すでに起動している EADs サーバからハートビートを受信すると、そのハッシュ値をチェックして、ハッシュ値が異なれば、起動に失敗します。

### (b) クラスタ起動中に、すでにクラスタに参加している EADs サーバがダウンした場合

クラスタ起動中に、すでにクラスタに参加している EADs サーバがダウンした場合、ダウンした EADs サーバは縮退状態に遷移します。ほかの EADs サーバも起動処理を中断し、起動に失敗します。

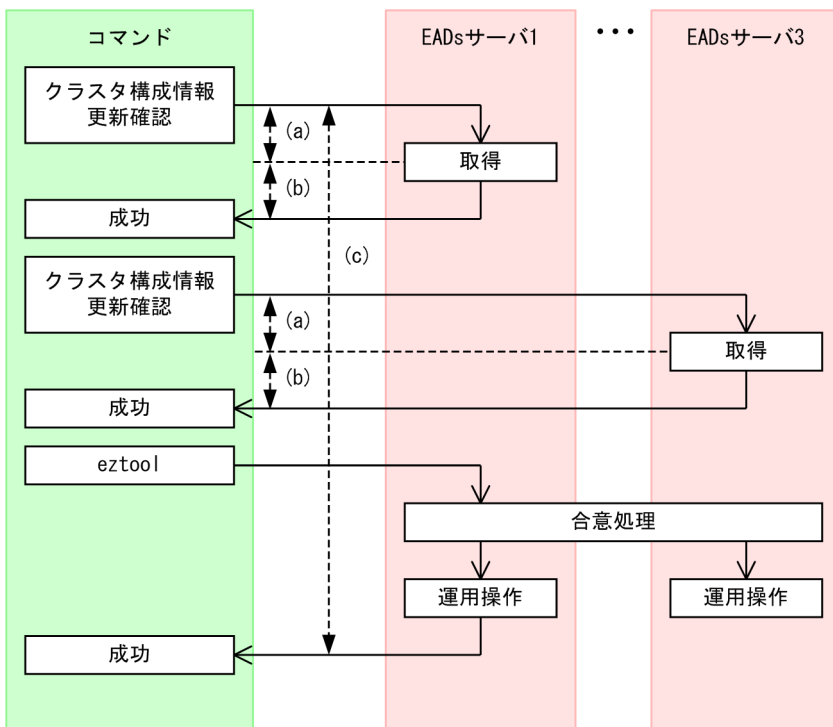
クラスタ内の半数以上の EADs サーバが同時にダウンした場合、タイムアウトします。

### (c) クラスタ起動中に、まだクラスタに参加していない EADs サーバがダウンした場合

クラスタ起動中に、まだクラスタに参加していない EADs サーバがダウンした場合、全 EADs サーバの起動処理が完了しないため、すでに起動している EADs サーバはタイムアウトします。

### (3) クラスタの運用操作

eztool コマンドを実行して、クラスタを運用操作する際のタイムを次の図に示します。



図中のアルファベットは、「9.3.3 タイムアウトに関連するパラメタ」の説明と次のように対応しています。

(a) : 「9.3.3(2)(a) eads.management.connect.timeout」

(b) : 「9.3.3(2)(b) eads.management.read.timeout」 ※1

(c) : 「9.3.3(2)(c) eads.management.command.timeout」 ※2

注※1

次に示すコマンドを実行する場合は、コマンド定義の `eads.management.long_running.read.timeout` パラメタの値が適用されます。

- eztool export
- eztool import
- eztool createcache
- eztool deletcache
- eztool execfunc
- eztool importecf
- eztool resume
- eztool compaction

- eztool stop

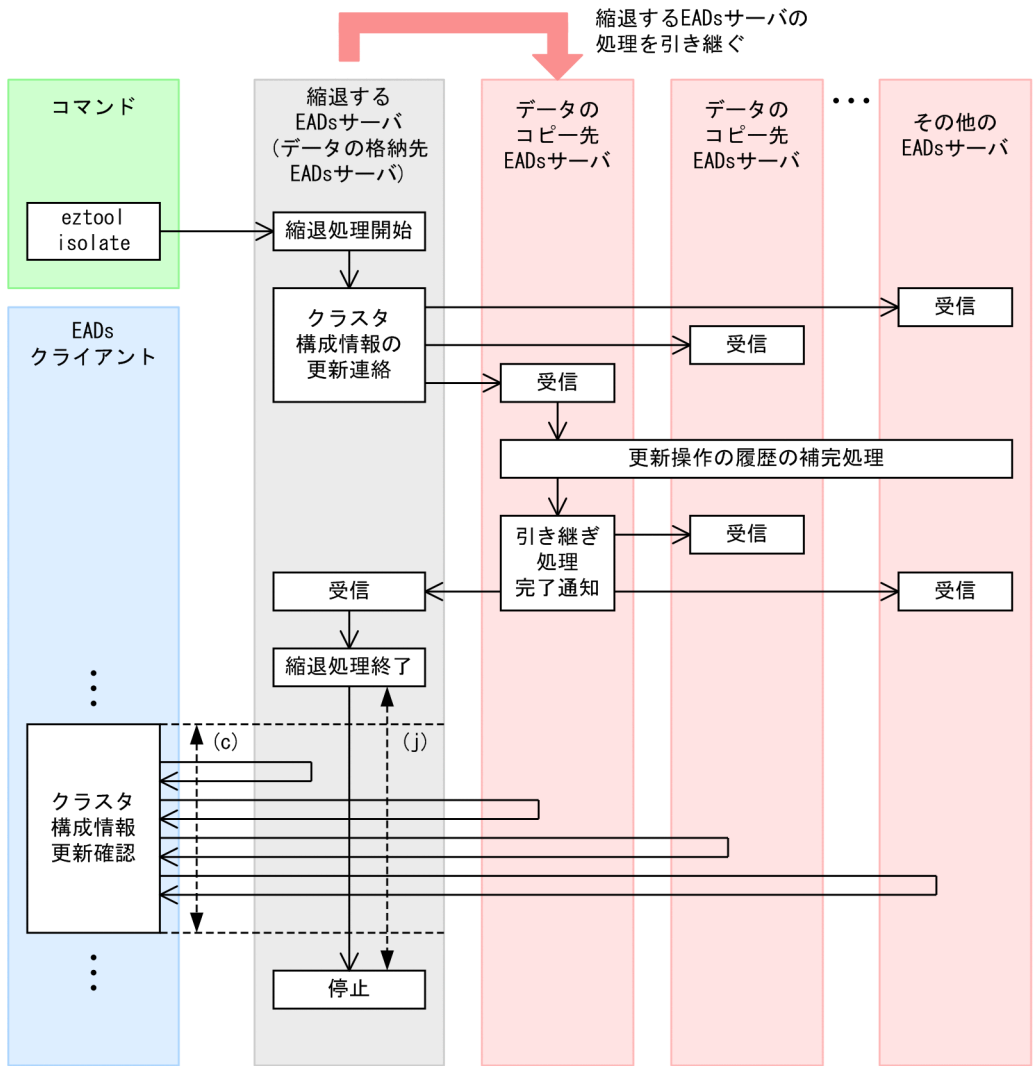
#### 注※2

次に示すコマンドを実行する場合は、コマンド定義の `eads.management.long_running.command.timeout` パラメタの値が適用されます。

- eztool export
- eztool import
- eztool createcache
- eztool deletetecache
- eztool execfunc
- eztool importecf
- eztool resume
- eztool compaction
- eztool stop

## (4) EADs サーバの縮退処理

EADs サーバの縮退処理の流れとタイマの関係を次の図に示します。



図中のアルファベットは、「9.3.3 タイムアウトに関連するパラメタ」の説明と次のように対応しています。

(c) : 「9.3.3(3)(c) eads.client.clusterInfo.update.interval」

(j) : 「9.3.3(1)(j) eads.gracefulstop.wait\_time」

eztool isolate コマンドを実行して EADs サーバを縮退させる場合、サーバ定義の eads.gracefulstop.wait\_time パラメタで、EADs サーバの縮退処理が完了してから停止するまでの時間を指定できます。クライアント定義の eads.client.clusterInfo.update.interval パラメタの指定値よりもこの値を大きくすることで、EADs クライアントのクラスタ構成情報の更新完了後に、EADs サーバを停止できます。

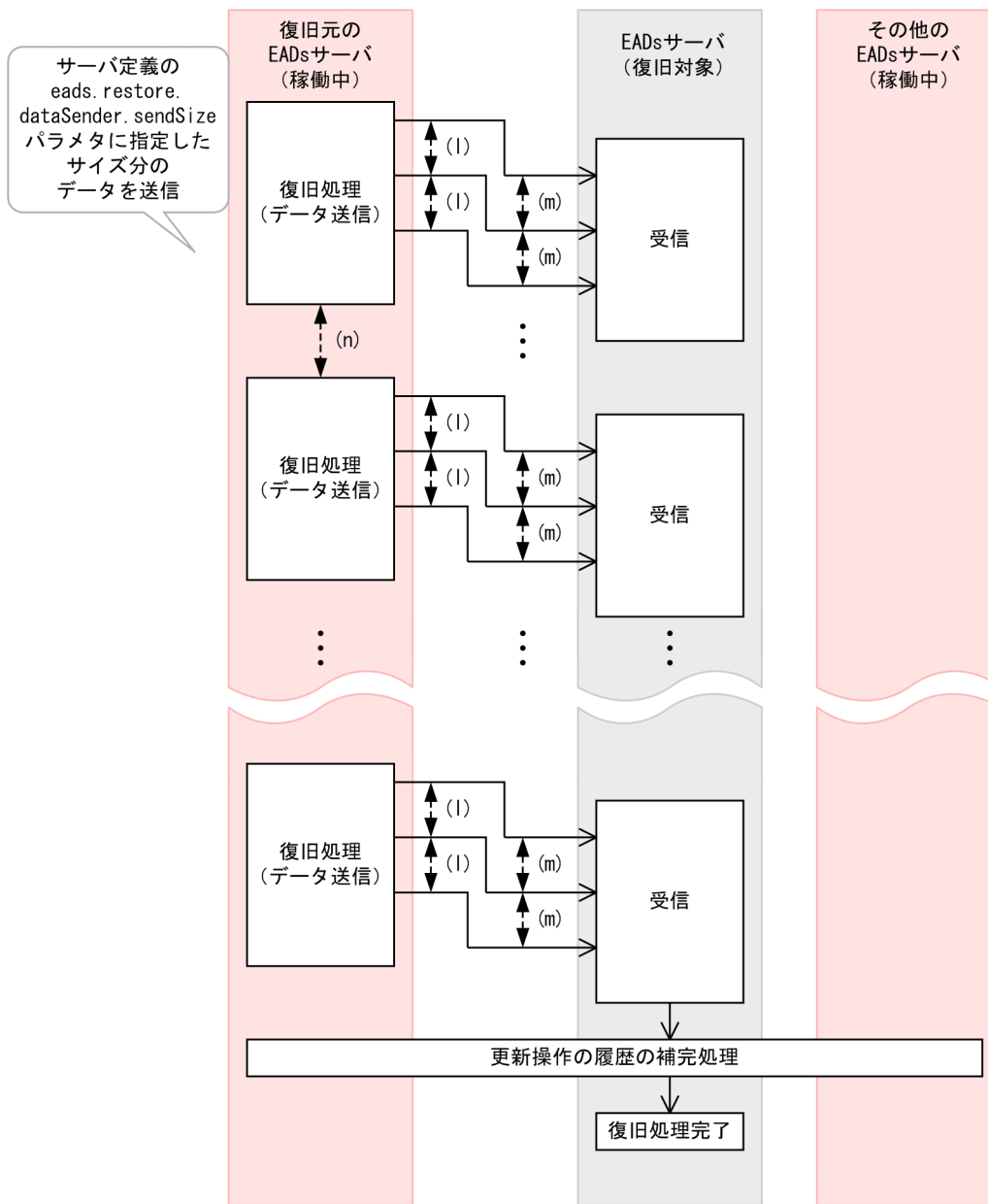
なお、クラスタ監視によって EADs サーバが縮退する場合、eads.gracefulstop.wait\_time パラメタの指定は無効となります。

更新操作の履歴の補完処理については、「9.3.2(6) 更新操作の履歴の補完処理」を参照してください。



## (5) クラスターの復旧処理

クラスタを復旧する際のタイマを次の図に示します。



図中のアルファベットは、「9.3.3 タイムアウトに関連するパラメタ」の説明と次のように対応しています。

(l) : 「9.3.3(1)(l) eads.restore.dataSender.send.timeout」

(m) : 「9.3.3(1)(m) eads.restore.dataReceiver.read.timeout」

(n) : 「9.3.3(1)(n) eads.restore.dataSender.sendInterval」

サーバ定義の `eads.restore.dataSender.sendSize` パラメタに指定したサイズを超えるまで、10 キロバイト単位でデータを連続して送信します。例えば、25 キロバイトを指定した場合は、30 キロバイトまでデータを送信します。

復旧処理では、データの整合性を回復するために、稼働中の EADs サーバが復旧対象の EADs サーバにデータを送信します。

そのため、次の点に留意してください。

- EADs サーバが復旧するまでに、少なくともデータ取得分の時間が掛かります。
- データ送信元の EADs サーバでは、データを送信する分、オンライン性能に影響があります。

## 参考

ディスクキャッシュ、および 2Way キャッシュを復旧する場合は、復旧処理で送信するデータサイズをキャッシュ定義の `eads.cache.restore.dataSender.sendSize` パラメタで指定します。また、復旧処理でのデータ送信間隔を `eads.cache.restore.dataSender.sendInterval` パラメタで指定します。

データの更新中でも、データの整合性を回復した状態で、縮退した EADs サーバをクラスタに復帰させることができます。縮退状態が発生した場合の復旧までの流れについては、「[12.2.1 縮退状態が発生した場合](#)」を参照してください。

更新操作の履歴の補完処理については、「[9.3.2\(6\) 更新操作の履歴の補完処理](#)」を参照してください。

## (6) 更新操作の履歴の補完処理

EADs サーバの縮退処理、復旧処理、および排他制御では、EADs サーバ間で更新操作の履歴を確認します。EADs サーバ間で更新操作の履歴に差異がある場合は、更新操作の履歴の補完処理を行います。これによって、データの書き込み順序の整合性を確保します。

更新操作の履歴の補完処理は、次の 2 つの処理から成ります。

- 他 EADs サーバの更新操作の履歴の補完処理
- 自 EADs サーバの更新操作の履歴の補完処理

なお、この場合の自 EADs サーバとは次の EADs サーバを指します。

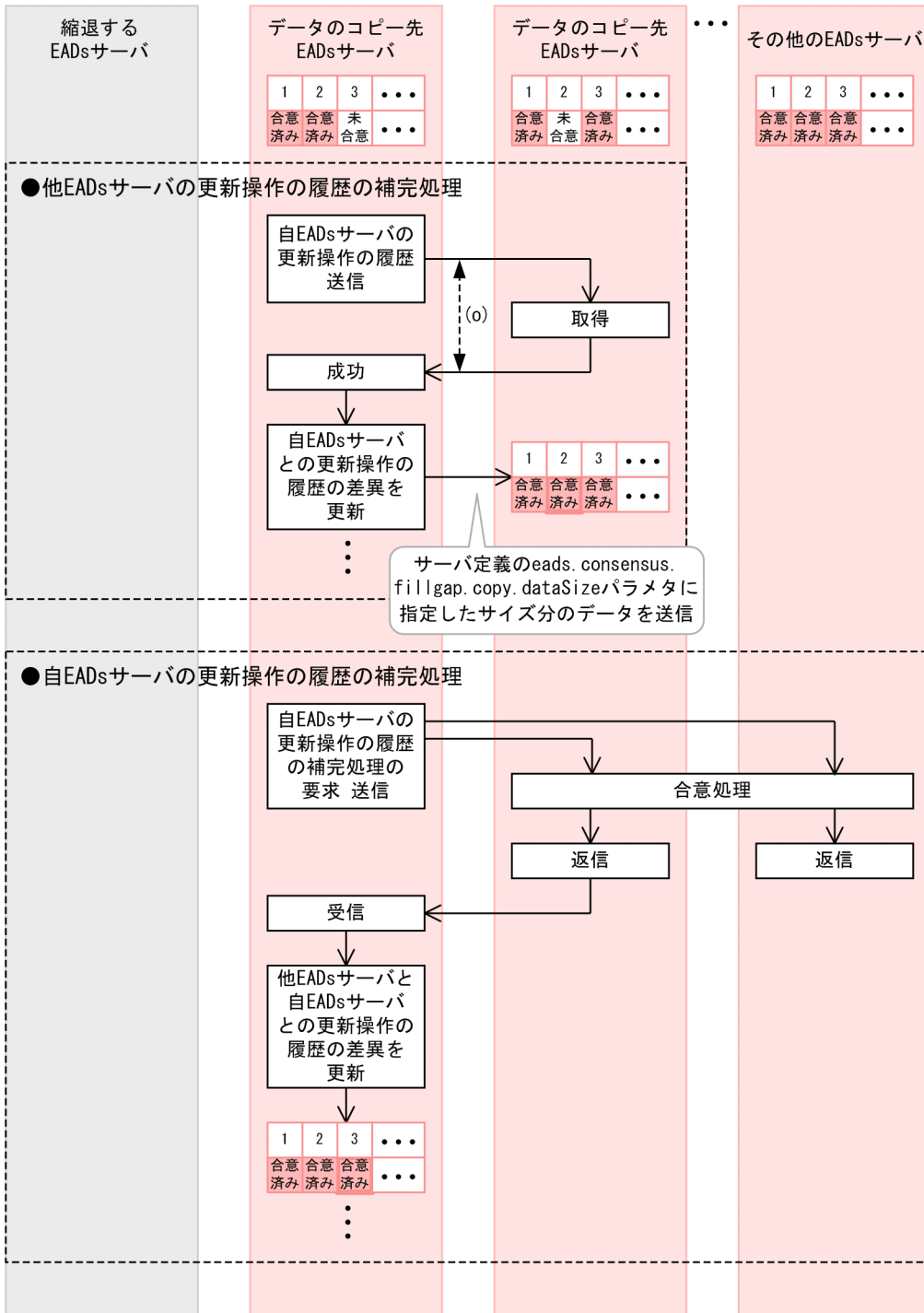
縮退処理の場合：

縮退する EADs サーバの処理を引き継ぐ EADs サーバ（データのコピー先 EADs サーバ）

復旧処理の場合：

復旧対象の EADs サーバ

更新操作の履歴の補完処理の流れとタイマの関係を次の図に示します。



(凡例)

1	2	3	...	: 更新操作の履歴
合意 済み	合意 済み	未 合意	...	

図中のアルファベットは、「9.3.3 タイムアウトに関連するパラメタ」の説明と次のように対応しています。

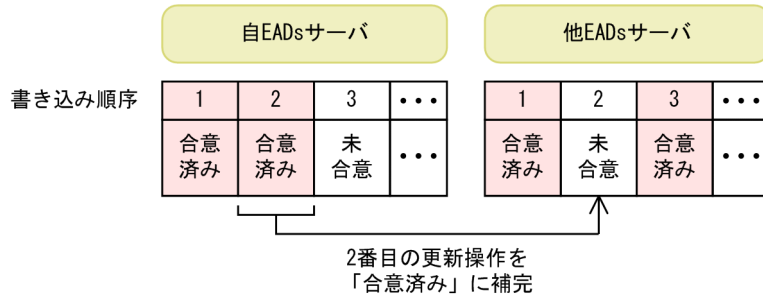
(o) : 「9.3.3(1)(o) eads.consensus.fillgap.copy.timeout」

### 他 EADs サーバの更新操作の履歴の補完処理

1. 自 EADs サーバの更新操作の履歴を各 EADs サーバに送信します。

- 自 EADs サーバと他 EADs サーバの更新操作の履歴に差異がある場合、他 EADs サーバに対して、自 EADs サーバの更新操作の履歴を送信します。このとき、サーバ定義の `eads.consensus.fillgap.copy.dataSize` パラメタに指定したサイズのデータを送信します。
- 他 EADs サーバは、自 EADs サーバから送信された更新操作の履歴を基に更新操作の履歴を補完します。

他 EADs サーバの更新操作の履歴は次のように補完されます。

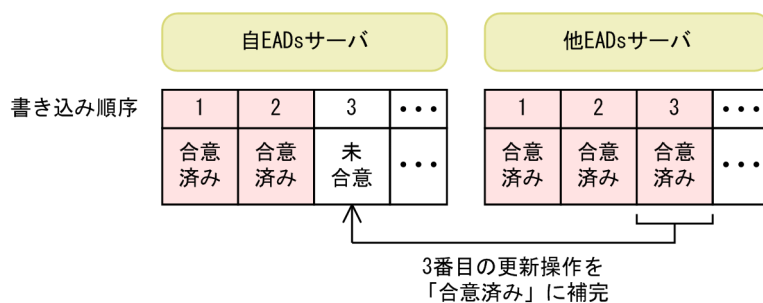


他 EADs サーバの更新操作の履歴の補完処理は、データのコピー先 EADs サーバに対して、更新操作の履歴の差異の数だけ実行されます。

### 自 EADs サーバの更新操作の履歴の補完処理

- 自 EADs サーバの更新操作の履歴に差異があるかを確認するため、各 EADs サーバに更新操作の履歴の補完処理の要求を送信します。
- 更新操作の履歴の補完処理の要求に対して合意処理が行われます。  
サーバ定義の `eads.cache.consensus.timeout` パラメタに指定した時間内に合意処理が完了しない場合はタイムアウトして、再度、合意処理を行います。合意できるまで無限に繰り返します。
- 合意処理によって、他 EADs サーバから自 EADs サーバに更新操作の履歴が送信されます。
- 自 EADs サーバは、各 EADs サーバから送信された更新操作の履歴を基に更新操作の履歴を補完します。

自 EADs サーバの更新操作の履歴は次のように補完されます。



自 EADs サーバの更新操作の履歴の補完処理は、自 EADs サーバの更新操作の履歴の差異の数だけ実行されます。

1 回の復旧処理、または縮退処理に対して、更新操作の履歴の補完処理が 1 回だけとは限りません。最大で  $(\text{データの多重度} - 1) \times (\text{キャッシュ数})$  回、実行されます。

更新操作の履歴の補完処理の同時実行スレッド数は、サーバ定義の `eads.cluster.controller.cache.max_execute_threads` でパラメタ指定できます。

### 9.3.3 タイムアウトに関連するパラメタ

タイムアウトに関連するパラメタの一覧を次の表に示します。

表 9-4 タイムアウトに関連するパラメタに関するパラメタ一覧

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
1	サーバ定義ファイル	<code>eads.connection.timeout</code>	100~3600000 (ミリ秒)	3000
2		<code>eads.cluster.heartbeat.interval</code>	10~60000 (ミリ秒)	400
3		<code>eads.cluster.heartbeat.timeout</code>	10~86400000 (ミリ秒)	2000
4		<code>eads.cluster.failureDetector.connect.timeout</code>	1~60000 (ミリ秒)	500
5		<code>eads.cluster.failureDetector.read.timeout</code>	1~60000 (ミリ秒)	500
6		<code>eads.cluster.failureDetector.retry</code>	0~100	0
7		<code>eads.cluster.assertive.threshold</code>	1~49	1
8		<code>eads.cluster.boot.timeout</code>	1~86400 (秒)	60
9		<code>eads.server.session.keepAlive.timeout</code>	0~3600 (秒)	3600
10		<code>eads.gracefulstop.wait_time</code>	0~60000 (ミリ秒)	3000
11		<code>eads.cache.consensus.timeout</code>	10~60000 (ミリ秒)	800
12		<code>eads.restore.dataSender.send.timeout</code>	100~3600000 (ミリ秒)	10000
13		<code>eads.restore.dataReceiver.read.timeout</code>	100~3600000 (ミリ秒)	1000
14		<code>eads.restore.dataSender.sendInterval</code>	0~60000 (ミリ秒)	1000
15		<code>eads.consensus.fillgap.copy.timeout</code>	100~3600000 (ミリ秒)	2000

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
16		<code>eads.management.resume.sendInterval</code>	0~2147483647 (ミリ秒)	0
17	コマンド定義ファイル	<code>eads.management.connect.timeout</code>	0~2147483647 (ミリ秒)	3000
18		<code>eads.management.read.timeout</code>	0~2147483647 (ミリ秒)	10000
19		<code>eads.management.command.timeout</code>	0~2147483647 (秒)	60
20		<code>eads.management.long_running.read.timeout</code>	0~2147483647 (ミリ秒)	600000
21		<code>eads.management.long_running.command.timeout</code>	0~2147483647 (秒)	600
22	クライアント定義ファイル	<code>eads.connection.send.timeout</code>	100~3600000 (ミリ秒)	1000
23		<code>eads.connection.receive.timeout</code>	100~3600000 (ミリ秒)	1000
24		<code>eads.client.clusterInfo.update.interval</code>	10~60000 (ミリ秒)	1000

## (1) サーバ定義ファイル

### (a) `eads.connection.timeout`

クラスタ構成情報の更新確認やデータ送受信のタイムアウト時間（単位：ミリ秒）を指定します。

### (b) `eads.cluster.heartbeat.interval`

ハートビートの送信間隔（単位：ミリ秒）を指定します。

### (c) `eads.cluster.heartbeat.timeout`

ハートビートのタイムアウト時間（単位：ミリ秒）を指定します。

### (d) `eads.cluster.failureDetector.connect.timeout`

生存確認の接続タイムアウト時間（単位：ミリ秒）を指定します。

### (e) `eads.cluster.failureDetector.read.timeout`

生存確認の受信タイムアウト時間（単位：ミリ秒）を指定します。

## (f) `eads.cluster.failureDetector.retry`

生存確認タイムアウト時のリトライ回数を指定します。

## (g) `eads.cluster.assertive.threshold`

EADs サーバダウンに同意する EADs サーバ数を指定します。

(クラスタを構成する EADs サーバ数 + 1) ÷ 2 を切り上げた値以下となるように設定してください。例えば、クラスタを構成する EADs サーバ数が 5 つの場合は、このパラメタに 3 以下の値を指定してください。

## (h) `eads.cluster.boot.timeout`

クラスタを構成する全 EADs サーバが開始するまでの最大待ち時間（単位：秒）を指定します。

## (i) `eads.server.session.keepAlive.timeout`

EADs クライアント・EADs サーバ間の接続を切断するまでの時間（単位：秒）を指定します。

リクエスト待ちの状態（通信が行われない状態）が、指定した時間以上続いた場合、接続が切断されます。

0 を指定した場合、タイムアウトなし（無限）となります。

## (j) `eads.gracefulstop.wait_time`

`eztool isolate` コマンドを実行した EADs サーバの縮退処理が完了してから、停止するまでの時間（単位：ミリ秒）を指定します。

EADs クライアントからクラスタ構成情報の更新確認を受信した EADs サーバが縮退する際、EADs クライアントのクラスタ構成情報の更新が完了する前に接続がクローズしてしまうと、通信エラーになります。このパラメタの指定値を調整することで、EADs クライアントのクラスタ構成情報の更新完了後に、EADs サーバを停止できます。

このパラメタの指定値には、クライアント定義の `eads.client.clusterInfo.update.interval` パラメタの指定値よりも大きく、かつコマンド定義の `eads.management.read.timeout` パラメタの指定値より小さい値を設定してください。

## (k) `eads.cache.consensus.timeout`

合意処理のタイムアウト時間（単位：ミリ秒）を指定します。

指定した時間内に合意処理ができない場合は、タイムアウトして再度合意処理をやり直します。

## (l) `eads.restore.dataSender.send.timeout`

復旧処理でのデータ送信タイムアウト時間（単位：ミリ秒）を指定します。

扱うデータサイズに合わせて指定してください。

## (m) eads.restore.dataReceiver.read.timeout

復旧処理でのデータ受信タイムアウト時間（単位：ミリ秒）を指定します。

扱うデータサイズに合わせて指定してください。

## (n) eads.restore.dataSender.sendInterval

復旧処理でのデータ送信間隔（単位：ミリ秒）を指定します。

このパラメタの指定値とサーバ定義の `eads.restore.dataSender.sendSize` パラメタの指定値は、次の式を満たすように設定してください。

$$\begin{aligned} & \text{復旧処理のデータ送信に使用できる帯域 (単位: bps)} \geq \\ & \quad \{ \text{復旧処理で送信するデータサイズ (単位: バイト)} \times 8 \} \\ & \quad \div \{ (\text{復旧処理でのデータ送信間隔 (単位: ミリ秒)} \\ & \quad + \text{データ送信に掛かる時間 (単位: ミリ秒)}) \div 1,000 \} \end{aligned}$$

復旧処理で送信するデータサイズ（単位：バイト）：

サーバ定義の `eads.restore.dataSender.sendSize` パラメタの指定値

復旧処理でのデータ送信間隔（単位：ミリ秒）：

サーバ定義の `eads.restore.dataSender.sendInterval` パラメタの指定値

データ送信に掛かる時間（単位：ミリ秒）：

サーバ定義の `eads.restore.dataSender.sendSize` パラメタに指定したサイズのデータ送信に掛かる時間（単位：ミリ秒）

「9.3.2(5) クラスタの復旧処理」の図中の「復旧処理（データ送信）」に掛かる時間を示します。この時間は環境によって異なります。

復旧処理に掛かる時間を短くすると、復旧処理の通信負荷が高くなります。逆に、復旧処理の通信負荷を低くすると、復旧処理に掛かる時間が長くなります。

ディスクキャッシュ、および 2Way キャッシュを使用する場合、このパラメタの指定は無効になり、キャッシュ定義の `eads.cache.restore.dataSender.sendInterval` パラメタの指定値が有効となります。

## (o) eads.consensus.fillgap.copy.timeout

更新操作の履歴の補完処理で、更新操作の履歴を送信する際の送信タイムアウト時間（単位：ミリ秒）を指定します。

次のどちらか大きい方の値を処理できる時間を設定してください。

- データのコピー先 EADs サーバに対して、1 回当たりに送信するデータサイズ（サーバ定義の `eads.consensus.fillgap.copy.dataSize` パラメタの値）
- key の最大サイズ + value の最大サイズ × 2



更新操作の履歴の補完処理については、「9.3.2(6) 更新操作の履歴の補完処理」を参照してください。

## (p) `eads.management.resume.sendInterval`

`eztool resume` コマンド実行時の差分転送間隔（単位：ミリ秒）を指定します。

## (2) コマンド定義ファイル

### (a) `eads.management.connect.timeout`

コマンド実行時のクラスタ構成情報の更新確認の接続タイムアウト時間（単位：ミリ秒）を指定します。

このパラメタに 0 を指定した場合、タイムアウトしません。

### (b) `eads.management.read.timeout`

コマンド実行時のクラスタ構成情報の更新確認の受信タイムアウト時間（単位：ミリ秒）を指定します。

このパラメタに 0 を指定した場合、タイムアウトしません。

次のコマンドを実行する場合は、このパラメタの値を、クライアント定義の `eads.connection.receive.timeout` パラメタに適用します。

- `eztool put`
- `eztool get`
- `eztool remove`

ただし、次の場合はこのパラメタの値を適用しません。

- このパラメタに 0 を指定した場合、または、クライアント定義の `eads.connection.receive.timeout` パラメタの最大値以上の値を指定した場合  
`eads.connection.receive.timeout` パラメタの最大値を適用します。
- クライアント定義の `eads.connection.receive.timeout` パラメタの最小値以下の値を指定した場合  
`eads.connection.receive.timeout` パラメタの最小値を適用します。

### (c) `eads.management.command.timeout`

コマンド実行開始からのタイムアウト時間（単位：秒）を指定します。

このパラメタに 0 を指定した場合、タイムアウトしません。

### (d) `eads.management.long_running.read.timeout`

次に示すコマンド実行時のクラスタ構成情報の更新確認の受信タイムアウト時間（単位：ミリ秒）を指定します。

- `eztool export`

- `eztool import`
- `eztool createcache`
- `eztool deletetecache`
- `eztool execfunc`
- `eztool importecf`
- `eztool resume`
- `eztool compaction`
- `eztool stop`

このパラメタに 0 を指定した場合、タイムアウトしません。

`eztool execfunc` コマンドを実行する場合は、このパラメタの値を、クライアント定義の `eads.connection.receive.timeout` パラメタに適用します。ただし、次の場合はこのパラメタの値を適用しません。

- このパラメタに 0 を指定した場合、または、クライアント定義の `eads.connection.receive.timeout` パラメタの最大値以上の値を指定した場合  
`eads.connection.receive.timeout` パラメタの最大値を適用します。
- クライアント定義の `eads.connection.receive.timeout` パラメタの最小値以下の値を指定した場合  
`eads.connection.receive.timeout` パラメタの最小値を適用します。

### (e) `eads.management.long_running.command.timeout`

次に示すコマンドの実行開始からのタイムアウト時間（単位：秒）を指定します。

- `eztool export`
- `eztool import`
- `eztool createcache`
- `eztool deletetecache`
- `eztool execfunc`
- `eztool importecf`
- `eztool resume`
- `eztool compaction`
- `eztool stop`

このパラメタに 0 を指定した場合、タイムアウトしません。

### (3) クライアント定義ファイル

#### (a) `eads.connection.send.timeout`

クラスタ構成情報の更新確認やデータ送信のタイムアウト時間（単位：ミリ秒）を指定します。

#### (b) `eads.connection.receive.timeout`

クラスタ構成情報の更新確認やデータ受信のタイムアウト時間（単位：ミリ秒）を指定します。

次の時間を考慮してください。

- EADs サーバに配置しているユーザファンクションおよびユーザイベントハンドラの実行時間
- 縮退処理および復旧処理に掛かる時間

縮退処理および復旧処理に掛かる時間については、「[9.3.2\(4\) EADs サーバの縮退処理](#)」と「[9.3.2\(5\) クラスタの復旧処理](#)」を参考にして、十分に大きな値を指定してください。

#### (c) `eads.client.clusterInfo.update.interval`

EADs クライアントがクラスタ内の一つ一つの EADs サーバに対してクラスタ構成情報の更新確認を行う間隔（単位：ミリ秒）を指定します。

クラスタを構成する EADs サーバ数を考慮してください。

クラスタを構成する EADs サーバ数が増えると、このパラメタに指定した時間内に通信する回数が増えるため、EADs クライアントへの負荷が増加します。

## 9.4 コンパクションに関連するパラメタの設計

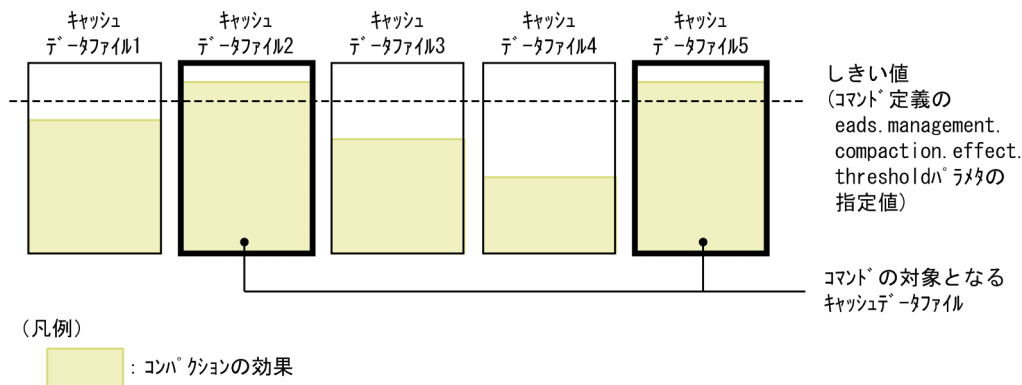
この節の内容は、ディスクキャッシュ、および2Way キャッシュを使用する場合にお読みください。

キャッシュデータファイルのコンパクションに関連するパラメタを設計します。コンパクションについては、「10.9 キャッシュデータファイルの使用量を削減する（キャッシュデータファイルをコンパクションする）」を参照してください。

### 9.4.1 コンパクションの効果のしきい値の設定

キャッシュデータファイルに対する、コンパクションの効果のしきい値を設定できます。コンパクションの効果のしきい値の概要を次の図に示します。

図 9-3 コンパクションの効果のしきい値の概要



#### 考え方

キャッシュデータファイルごとに、コンパクションの効果は異なります。eztool compaction コマンドを実行する際、しきい値を設定することで、例えば、一定以上のコンパクションの効果があるキャッシュデータファイルに絞って、コンパクションを実行する、という運用ができます。

次のコマンドで--threshold オプションを省略した際に設定される、コンパクションの効果のしきい値を、コマンド定義の eads.management.compaction.effect.threshold パラメタで設定できます。

- eztool listecf コマンド
- eztool compaction コマンド

この図の例では、キャッシュデータファイル 2 と 5 がコンパクションの効果がしきい値以上のファイルとなります。このとき、-threshold オプションを省略して eztool compaction コマンドを実行すると、キャッシュデータファイル 2 と 5 がコンパクションの対象となります。

### 9.4.2 コンパクションに関連するパラメタ

コンパクションに関連するパラメタの一覧を次の表に示します。

表 9-5 コンパクションに関連するパラメーター一覧

項番	定義ファイル	パラメータ名	指定する値	デフォルト値
1	コマンド定義ファイル	<code>eads.management.compaction.effect.threshold</code>	1～100 (%)	50
2		<code>eads.management.compaction.effect.division</code>	1～10	5

## (1) コマンド定義ファイル

### (a) `eads.management.compaction.effect.threshold`

次に示すコマンドで使用する、コンパクションの効果のしきい値（単位：%）を指定します。

- `eztool listecf`
- `eztool compaction`

### (b) `eads.management.compaction.effect.division`

`eztool listecf` コマンドで使用する、コンパクションの効果のファイル数の分布を分割する個数を指定します。

例えば、このパラメータに 4 を指定した場合、分布は 4 つに分割されるため、コンパクションの効果が「0～25%」、「26～50%」、「51～75%」および「76～100%」であるファイル数がそれぞれ出力されます。

効果の値が整数にならない場合、小数点以下はすべて切り捨てられます。

# 10

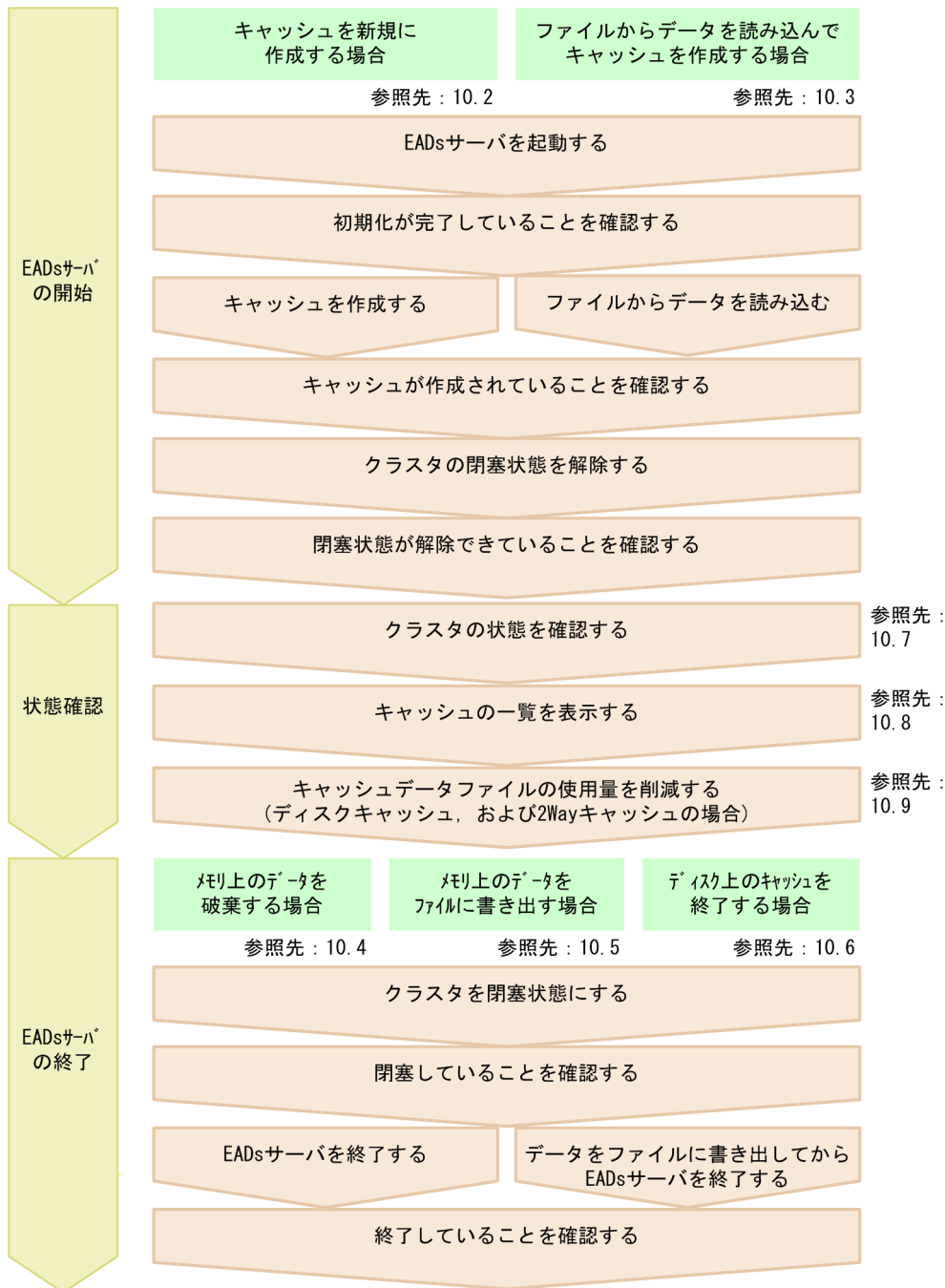
## 通常運用

この章では、システム運用管理者の主な作業項目と、EADs の通常運用の方法について説明します。

## 10.1 システム運用管理者が行うこと

通常運用の流れを次の図に示します。

図 10-1 通常運用の流れ



システムのメンテナンスが必要な際は、保守運用を行います。保守運用については、「11. 保守運用」を参照してください。

システムに障害が発生した際は、障害運用を行います。障害運用については、「12. 障害運用」を参照してください。

## 10.2 EADs サーバを開始する (キャッシュを新規に作成する場合)

EADs サーバを起動したあとに、キャッシュを新規に作成します。

### 10.2.1 EADs サーバを開始する手順 (メモリ上にキャッシュを作成する)

EADs サーバを起動したあとに、メモリキャッシュを新規に作成する手順について説明します。

#### (1) EADs サーバを起動する

EADs サーバを起動するホストにログインして `ezstart` コマンドを実行し、EADs サーバを起動します。このコマンドは EADs サーバごとに実行する必要があります。

```
ezstart
```

#### (2) 初期化が完了していることを確認する

EADs サーバの起動に成功すると、初期化状態に遷移します。

`eztool status` コマンドを実行して、全 EADs サーバの初期化が完了していることを確認します。

```
eztool status
```

#### コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State      Operation
1   XX.XXX.XXX.168  24600      1288490189  online   initialized none
2   XX.XXX.XXX.168  24700      429496730   online   initialized none
3   XX.XXX.XXX.168  24800      -429496729  online   initialized none
4   XX.XXX.XXX.168  24900      -1288490188 online   initialized none
5   XX.XXX.XXX.168  25000      -2147483648 online   initialized none
-----
KDEA08002-I          The command will now end.
$
```

初期化が完了している場合、State 欄に `initialized` が表示されます。

#### (3) キャッシュを作成する

`eztool createcache` コマンドを実行して、key と value を格納するキャッシュを作成します。



```
eztool createcache <キャッシュ名>
```

## コマンド実行例

```
$ eztool createcache cache1
KDEA08001-I The command will now start. (subcommand = createcache, parameter = [createcache, cache1])
KDEA08002-I The command will now end.
$
```

## (4) キャッシュが作成されていることを確認する

`eztool listcache` コマンドを実行して、キャッシュの一覧を表示します。

```
eztool listcache
```

## コマンド実行例

```
$ eztool listcache
KDEA08001-I The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName CacheType TotalMasterKeyCount
cache1 Memory 0
cache2 Memory 0
cache3 Memory 0
-----
KDEA08002-I The command will now end.
$
```

CacheName 欄に表示されるキャッシュ名、および CacheType 欄に表示されるキャッシュタイプから、キャッシュが作成されていることを確認してください。

## (5) クラスタの閉塞状態を解除する

初期化状態は閉塞状態と同じく、EADs クライアントからのリクエストが受け付けられない状態です。

`eztool open` コマンドを実行して、クラスタの閉塞状態を解除します。

```
eztool open
```

## コマンド実行例

```
$ eztool open
KDEA08001-I The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する

eztool status コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

### コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   running none
 2  XX.XXX.XXX.168  24700      429496730  online   running none
 3  XX.XXX.XXX.168  24800      -429496729 online   running none
 4  XX.XXX.XXX.168  24900      -1288490188 online   running none
 5  XX.XXX.XXX.168  25000      -2147483648 online   running none
-----
KDEA08002-I          The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 10.2.2 EADs サーバを開始する手順（ディスク上にキャッシュを作成する）

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

EADs サーバを起動したあとに、ディスクキャッシュまたは 2Way キャッシュを新規に作成する手順について説明します。

### (1) キャッシュ定義ファイルを作成する

キャッシュタイプやキャッシュの格納先などを指定するための、キャッシュ定義ファイルを作成します。

キャッシュ定義ファイルのファイル名は eads\_cache.<キャッシュ名>.properties です。

キャッシュ定義のパラメタについては、「[7.7 キャッシュの運用に依存するパラメタの設計](#)」を参照してください。

#### 注意事項

キャッシュ定義ファイルの次のパラメタの定義内容は、クラスタを構成する全 EADs サーバで同じになるようにしてください。同じでないとキャッシュを作成できません。

- eads.cache.type
- eads.cache.disk.filesize
- eads.cache.disk.filenum
- eads.cache.disk.blocksize

EADs サーバ 5 台、データの多重度 3 でディスクキャッシュを作成する場合の、キャッシュ定義ファイルの例を次に示します。

```
eads.cache.type=Disk
eads.cache.disk.info.dir=store
eads.cache.disk.1.dir=/hdd/cache_server01_range01
eads.cache.disk.2.dir=/hdd/cache_server01_range05
eads.cache.disk.3.dir=/hdd/cache_server01_range04
eads.cache.disk.filesize=128
eads.cache.disk.filenum=8
eads.cache.disk.blocksize=1
```

## (2) EADs サーバを起動する

EADs サーバを起動するホストにログインして `ezstart` コマンドを実行し、EADs サーバを起動します。このコマンドは EADs サーバごとに実行する必要があります。

```
ezstart
```

## (3) 初期化が完了していることを確認する

EADs サーバの起動に成功すると、初期化状態に遷移します。

`eztool status` コマンドを実行して、全 EADs サーバの初期化が完了していることを確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State      Operation
 1  XX.XXX.XXX.168  24600      1288490189 online    initialized none
 2  XX.XXX.XXX.168  24700      429496730  online    initialized none
 3  XX.XXX.XXX.168  24800      -429496729 online    initialized none
 4  XX.XXX.XXX.168  24900      -1288490188 online    initialized none
 5  XX.XXX.XXX.168  25000      -2147483648 online    initialized none
-----
KDEA08002-I          The command will now end.
$
```

初期化が完了している場合、State 欄に initialized が表示されます。

## (4) キャッシュを作成する

eztool createcache コマンドを実行して、キャッシュを作成します。

```
eztool createcache <キャッシュ名>
```

### コマンド実行例

```
$ eztool createcache cache1
KDEA08001-I          The command will now start. (subcommand = createcache, parameter = [createcache, cache1])
KDEA08002-I          The command will now end.
$
```

キャッシュを作成すると、次のディレクトリにキャッシュファイルが作成されます。

- キャッシュデータファイル  
 <キャッシュ定義の eads.cache.disk.<n>.dir パラメタで指定したディレクトリ>/<キャッシュ名>
- キャッシュインデックスファイルおよびキャッシュ情報ファイル  
 <キャッシュ定義の eads.cache.disk.info.dir パラメタで指定したディレクトリ>/<キャッシュ名>  
 (デフォルトは<運用ディレクトリ>/store/<キャッシュ名>)

## (5) キャッシュが作成されていることを確認する

eztool listcache コマンドを実行して、キャッシュの一覧を表示します。

```
eztool listcache
```

### コマンド実行例

```
$ eztool listcache
KDEA08001-I      The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount
cache1     Disk       0
cache2     Disk       0
cache3     Disk       0
-----
KDEA08002-I      The command will now end.
$
```

CacheName 欄に表示されるキャッシュ名、および CacheType 欄に表示されるキャッシュタイプから、キャッシュが作成されていることを確認してください。

## (6) クラスタの閉塞状態を解除する

初期化状態は閉塞状態と同じく、EADs クライアントからのリクエストが受け付けられない状態です。[eztool open](#) コマンドを実行して、クラスタの閉塞状態を解除します。

```
eztool open
```

コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (7) 閉塞状態が解除できていることを確認する

[eztool status](#) コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online  running none
 2  XX.XXX.XXX.168  24700      429496730  online  running none
 3  XX.XXX.XXX.168  24800      -429496729 online  running none
 4  XX.XXX.XXX.168  24900      -1288490188 online  running none
 5  XX.XXX.XXX.168  25000      -2147483648 online  running none
-----

KDEA08002-I          The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

### 10.2.3 複数のキャッシュを使用する場合の注意点

複数のキャッシュを使用する場合、次に示す点に注意してください。

- キャッシュ単位にできる操作が限られているため、キャッシュ単位でのきめ細かな運用ができません。
- キャッシュに関連した各種リソースが増加します。
- 1つのプロセスが使用する Java ヒープが大きくなるため、GC による性能の問題が発生しやすくなります。

使用するキャッシュ数が多い場合は、次に示す方法でキャッシュ数を減らすことができます。

- 1つのクラスタ内でキャッシュを分けるのではなく、クラスタ自体を分ける。
- 異なるデータで同じ key 名を使うときに、複数のキャッシュを使用するのではなく、複数のグループを使用する。

## 10.3 EADs サーバを開始する（ファイルからデータを読み込んでキャッシュを作成する場合）

EADs サーバを起動したあとに、前回稼働時にストアデータファイルに書き出したデータを、再度メモリ上に読み込ませます。または、EADs サーバを起動したあとに、キャッシュファイルにアクセスしてキャッシュを再開します。

### 10.3.1 EADs サーバを開始する手順（メモリ上にキャッシュを作成する）

EADs サーバを起動したあとに、前回稼働時にストアデータファイルに書き出したデータを、再度メモリキャッシュに読み込ませる手順について説明します。

#### (1) EADs サーバを起動する

EADs サーバを起動するホストにログインして `ezstart` コマンドを実行し、EADs サーバを起動します。このコマンドは EADs サーバごとに実行する必要があります。

```
ezstart
```

#### (2) 初期化が完了していることを確認する

EADs サーバの起動に成功すると、初期化状態に遷移します。

`eztool status` コマンドを実行して、全 EADs サーバの初期化が完了していることを確認します。

```
eztool status
```

#### コマンド実行例

```
$ eztool status
KDEAO8001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State      Operation
1   XX.XXX.XXX.168  24600      1288490189 online   initialized none
2   XX.XXX.XXX.168  24700      429496730  online   initialized none
3   XX.XXX.XXX.168  24800      -429496729 online   initialized none
4   XX.XXX.XXX.168  24900      -1288490188 online   initialized none
5   XX.XXX.XXX.168  25000      -2147483648 online   initialized none
-----
KDEAO8002-I          The command will now end.
$
```

初期化が完了している場合、State 欄に initialized が表示されます。

### (3) データをメモリ上に読み込ませる

`eztool import` コマンドを実行して、前回稼働時にストアデータファイルに書き出した最新のデータを、メモリ上に読み込ませます。

```
eztool import
```

#### コマンド実行例

```
$ eztool import
KDEA08001-I      The command will now start. (subcommand = import, parameter = [import])
KDEA08054-I      The store data file was imported. (store data file key = stop_20130418100014)
KDEA08002-I      The command will now end.
$
```

この例のように、ストアデータファイルキーを省略すると、`eztool listesd` コマンドで、latest に表示される最新のストアデータファイルキーのストアデータファイルを自動的に読み込みます。

省略しない場合は、任意のストアデータファイルのストアデータファイルキーを指定します。

#### 注意事項

前回稼働時にメモリ容量が満杯に近い状態で `eztool export` コマンドを実行した場合、次回開始時に `eztool import` コマンドを実行すると、メモリ容量不足のためにデータの読み込みに失敗することがあります。

例えば、データを多重化する際にコピー先のメモリ容量があふれ、その状態で `eztool export` コマンドを実行した場合などが該当します。

このような場合は、起動設定の `eads.java.external.heapsize` パラメタの指定値を大きくしてから、`eztool import` コマンドを再実行してください。

### (4) キャッシュが作成されていることを確認する

`eztool listcache` コマンドを実行して、キャッシュの一覧を表示します。

```
eztool listcache
```

#### コマンド実行例



```
$ eztool listcache
KDEA08001-I      The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount
cache1     Memory     0
cache2     Memory     0
cache3     Memory     0
-----
KDEA08002-I      The command will now end.
$
```

CacheName 欄に表示されるキャッシュ名、および CacheType 欄に表示されるキャッシュタイプから、キャッシュが作成されていることを確認してください。

## (5) クラスタの閉塞状態を解除する

初期化状態は閉塞状態と同じく、EADs クライアントからのリクエストが受け付けられない状態です。`eztool open` コマンドを実行して、クラスタの閉塞状態を解除します。

```
eztool open
```

コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する

`eztool status` コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online  running none
 2  XX.XXX.XXX.168  24700      429496730  online  running none
 3  XX.XXX.XXX.168  24800      -429496729 online  running none
 4  XX.XXX.XXX.168  24900      -1288490188 online  running none
 5  XX.XXX.XXX.168  25000      -2147483648 online  running none
-----
KDEA08002-I          The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 10.3.2 EADs サーバを開始する手順（ディスク上のキャッシュを再開する）

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

EADs サーバを起動したあとに、前回終了時の状態のキャッシュファイルにアクセスして、ディスクキャッシュ、および 2Way キャッシュを再開する手順について説明します。

### (1) EADs サーバを起動する

EADs サーバを起動するホストにログインして `ezstart` コマンドを実行し、EADs サーバを起動します。このコマンドは EADs サーバごとに実行する必要があります。

```
ezstart
```

### (2) 初期化が完了していることを確認する

EADs サーバの起動に成功すると、初期化状態に遷移します。

`eztool status` コマンドを実行して、全 EADs サーバの初期化が完了していることを確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State      Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   initialized none
 2  XX.XXX.XXX.168  24700      429496730  online   initialized none
 3  XX.XXX.XXX.168  24800      -429496729 online   initialized none
 4  XX.XXX.XXX.168  24900      -1288490188 online   initialized none
 5  XX.XXX.XXX.168  25000      -2147483648 online   initialized none

-----

KDEA08002-I          The command will now end.
$
```

初期化が完了している場合、State 欄に initialized が表示されます。

### (3) キャッシュファイルにアクセスしてキャッシュを再開する

eztool resume コマンドを実行し、キャッシュファイルにアクセスしてキャッシュを再開します。

```
eztool resume
```

#### コマンド実行例

```
$ eztool resume
KDEA08001-I          The command will now start. (subcommand = resume, parameter = [resume])
KDEA08073-I          The resumption of cache started. (cache name = cache1)
KDEA08074-I          The resumption of cache finished. (cache name = cache1)
KDEA08073-I          The resumption of cache started. (cache name = cache3)
KDEA08074-I          The resumption of cache finished. (cache name = cache3)
KDEA08073-I          The resumption of cache started. (cache name = cache2)
KDEA08074-I          The resumption of cache finished. (cache name = cache2)
KDEA08002-I          The command will now end.
$
```

2Way キャッシュの場合は、キャッシュの再開のタイミングで、キャッシュデータファイルの内容をメモリ上に読み込みます。

### (4) キャッシュが再開されていることを確認する

eztool listcache コマンドを実行して、キャッシュの一覧を表示します。

```
eztool listcache
```

#### コマンド実行例

```
$ eztool listcache
KDEA08001-I      The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount
cache1     Disk       0
cache2     Disk       0
cache3     Disk       0
-----
KDEA08002-I      The command will now end.
$
```

CacheName 欄に表示されるキャッシュ名、および CacheType 欄に表示されるキャッシュタイプから、キャッシュが再開されていることを確認してください。

## (5) クラスタの閉塞状態を解除する

初期化状態は閉塞状態と同じく、EADs クライアントからのリクエストが受け付けられない状態です。`eztool open` コマンドを実行して、クラスタの閉塞状態を解除します。

```
eztool open
```

コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する

`eztool status` コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   running none
 2  XX.XXX.XXX.168  24700      429496730  online   running none
 3  XX.XXX.XXX.168  24800      -429496729 online   running none
 4  XX.XXX.XXX.168  24900      -1288490188 online   running none
 5  XX.XXX.XXX.168  25000      -2147483648 online   running none
-----

KDEA08002-I          The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 10.4 EADs サーバを終了する（メモリ上のデータを破棄する場合）

メモリキャッシュのデータをストアデータファイルに書き出さないで EADs サーバを終了します。

`eztool stop` コマンド実行時、`--no_export` オプションを指定すると、メモリキャッシュのデータはすべて破棄されます。

次回開始時にメモリキャッシュのデータを引き継ぎたい場合は、ストアデータファイルにメモリキャッシュのデータを書き出してから EADs サーバを終了します。手順については、「[10.5 EADs サーバを終了する（メモリ上のデータをファイルに書き出す場合）](#)」を参照してください。

### 10.4.1 EADs サーバを終了する手順

メモリキャッシュのデータをストアデータファイルに書き出さないで EADs サーバを終了する手順について説明します。

#### (1) クラスタを閉塞状態にする

`eztool close` コマンドを実行して、クラスタを閉塞状態にします。

```
eztool close
```

コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

#### (2) 閉塞していることを確認する

EADs サーバを閉塞後、`eztool status` コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   closed none
 2  XX.XXX.XXX.168  24700      429496730  online   closed none
 3  XX.XXX.XXX.168  24800      -429496729 online   closed none
 4  XX.XXX.XXX.168  24900      -1288490188 online   closed none
 5  XX.XXX.XXX.168  25000      -2147483648 online   closed none
-----

KDEA08002-I          The command will now end.
$
```

閉塞している場合、State 欄に closed が表示されます。

### (3) EADs サーバを終了する

`eztool stop --no_export` コマンドを実行して、メモリキャッシュのデータをストアデータファイルに書き出さずに、クラスタ内の全 EADs サーバを終了します。

```
eztool stop --no_export
```

コマンド実行例

```
$ eztool stop --no_export
KDEA08001-I          The command will now start. (subcommand = stop, parameter = [stop, --no_export])
KDEA08002-I          The command will now end.
$
```

### (4) 終了していることを確認する

EADs サーバのメッセージログを確認します。

```
0405 2012/01/12 14:09:09.974    EADs          05F934AD 2BF14CEB KDEA00002-I
The server will now shut down. (server name = testserver)
```

## 10.5 EADs サーバを終了する（メモリ上のデータをファイルに書き出す場合）

次回開始時にデータを再構築できるように、ストアデータファイルにメモリキャッシュのデータを書き出してから EADs サーバを終了します。

### 10.5.1 EADs サーバを終了する手順

次回開始時にデータを再構築できるように、ストアデータファイルにメモリキャッシュのデータを書き出してから EADs サーバを終了する手順について説明します。

#### (1) クラスタを閉塞状態にする

`eztool close` コマンドを実行して、クラスタを閉塞状態にします。

```
eztool close
```

コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

#### (2) 閉塞していることを確認する

EADs サーバを閉塞後、`eztool status` コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   closed none
 2  XX.XXX.XXX.168  24700      429496730  online   closed none
 3  XX.XXX.XXX.168  24800      -429496729 online   closed none
 4  XX.XXX.XXX.168  24900      -1288490188 online   closed none
 5  XX.XXX.XXX.168  25000      -2147483648 online   closed none
-----
KDEA08002-I      The command will now end.
$
```



閉塞している場合、State 欄に closed が表示されます。

### (3) EADs サーバを終了する

`eztool stop` コマンドを実行して、ストアデータファイルに最新のデータを書き出してから、クラスタ内の全 EADs サーバを終了します。

```
eztool stop
```

#### コマンド実行例

```
$ eztool stop
KDEA08001-I      The command will now start. (subcommand = stop, parameter = [stop])
KDEA08055-I      The store data file was exported. (store data file key = stop_20120112140900)
KDEA08002-I      The command will now end.
$
```

### (4) 終了していることを確認する

EADs サーバのメッセージログを確認します。

```
0405 2012/01/12 14:09:09.974    EADs          05F934AD 2BF14CEB KDEA00002-I
The server will now shut down. (server name = testserver)
```

## 10.6 EADs サーバを終了する (ディスク上のキャッシュを終了する場合)

この節の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

ディスクキャッシュ、および 2Way キャッシュを使用する EADs サーバを終了します。

### 10.6.1 EADs サーバを終了する手順

ディスクキャッシュ、および 2Way キャッシュを終了する手順について説明します。

#### (1) クラスタを閉塞状態にする

`eztool close` コマンドを実行して、クラスタを閉塞状態にします。

```
eztool close
```

コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

#### (2) 閉塞していることを確認する

EADs サーバを閉塞後、`eztool status` コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1  XX.XXX.XXX.168  24600      1288490189  online  closed none
2  XX.XXX.XXX.168  24700      429496730   online  closed none
3  XX.XXX.XXX.168  24800      -429496729  online  closed none
4  XX.XXX.XXX.168  24900      -1288490188 online  closed none
5  XX.XXX.XXX.168  25000      -2147483648 online  closed none
-----
KDEA08002-I      The command will now end.
$
```

閉塞している場合、State 欄に closed が表示されます。

### (3) EADs サーバを終了する

`eztool stop` コマンドを実行して、ディスクキャッシュ、および 2Way キャッシュを終了してから、クラスタ内の全 EADs サーバを終了します。

```
eztool stop
```

#### コマンド実行例

```
$ eztool stop
KDEA08001-I      The command will now start. (subcommand = stop, parameter = [stop])
KDEA08079-I      Exporting was not executed because no memory cache exist on the server.
KDEA08002-I      The command will now end.
$
```

なお、EADs サーバにメモリキャッシュが存在しない場合、`eztool stop` コマンド実行時にストアデータファイルは作成されません。

### (4) 終了していることを確認する

EADs サーバのメッセージログを確認します。

```
0405 2012/01/12 14:09:09.974      EADs      05F934AD 2BF14CEB KDEA00002-I
The server will now shut down. (server name = testserver)
```

## 10.7 クラスタおよび EADs サーバの状態を確認する

`eztool status` コマンドを実行して、クラスタおよび EADs サーバの状態を確認します。

クラスタおよび EADs サーバの状態については、「[2.12 クラスタおよび EADs サーバの状態遷移](#)」を参照してください。

```
eztool status [-v|--verbose]
```

### コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online  running none
 2  XX.XXX.XXX.168  24700      429496730  online  running none
 3  XX.XXX.XXX.168  24800      -429496729 online  running none
 4  XX.XXX.XXX.168  24900      -1288490188 online  running none
 5  XX.XXX.XXX.168  25000      -2147483648 online  running none
-----

KDEA08002-I          The command will now end.
$
```

また、`-v` または `--verbose` オプションを指定すると、詳細を表示できます。

```
$ eztool status -v
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status, -v])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ServerName  ClientPort  ServerPort  ManagePort  Position    Cluster  State  Operation  Lock  KeyCount  UsedCache  Version
 1  XX.XXX.XXX.168  server01   24600      24633      24620      1288490189 online  running  none    unlock  0 0% ( 0 / 92 MB ) 03-60-00
 2  XX.XXX.XXX.168  server02   24700      24733      24720      429496730  online  running  none    unlock  0 0% ( 0 / 92 MB ) 03-60-00
 3  XX.XXX.XXX.168  server03   24800      24833      24820      -429496729 online  running  none    unlock  0 0% ( 0 / 92 MB ) 03-60-00
 4  XX.XXX.XXX.168  server04   24900      24933      24920      -1288490188 online  running  none    unlock  0 0% ( 0 / 92 MB ) 03-60-00
 5  XX.XXX.XXX.168  server05   25000      25033      25020      -2147483648 online  running  none    unlock  0 0% ( 0 / 92 MB ) 03-60-00
-----

KDEA08002-I          The command will now end.
$
```

## 10.8 キャッシュの一覧を表示する

`eztool listcache` コマンドを実行して、キャッシュの一覧を表示します。

```
eztool listcache [-v|--verbose]
```

### コマンド実行例

```
$ eztool listcache
KDEA08001-I      The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount
cache1     Memory     0
cache2     Memory     0
cache3     Memory     0
-----
KDEA08002-I      The command will now end.
$
```

また、`-v` または `--verbose` オプションを指定すると、詳細を表示できます。

```
$ eztool listcache -v
KDEA08001-I      The command will now start. (subcommand = listcache, parameter = [listcache, -v])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount  Server                               MasterKeyCount
cache1     Memory     0  XX.XXX.XXX.168:24600  0
           0  XX.XXX.XXX.168:24700  0
           0  XX.XXX.XXX.168:24800  0
           0  XX.XXX.XXX.168:24900  0
           0  XX.XXX.XXX.168:25000  0
cache2     Memory     0  XX.XXX.XXX.168:24600  0
           0  XX.XXX.XXX.168:24700  0
           0  XX.XXX.XXX.168:24800  0
           0  XX.XXX.XXX.168:24900  0
           0  XX.XXX.XXX.168:25000  0
cache3     Memory     0  XX.XXX.XXX.168:24600  0
           0  XX.XXX.XXX.168:24700  0
           0  XX.XXX.XXX.168:24800  0
           0  XX.XXX.XXX.168:24900  0
           0  XX.XXX.XXX.168:25000  0
-----
KDEA08002-I      The command will now end.
$
```

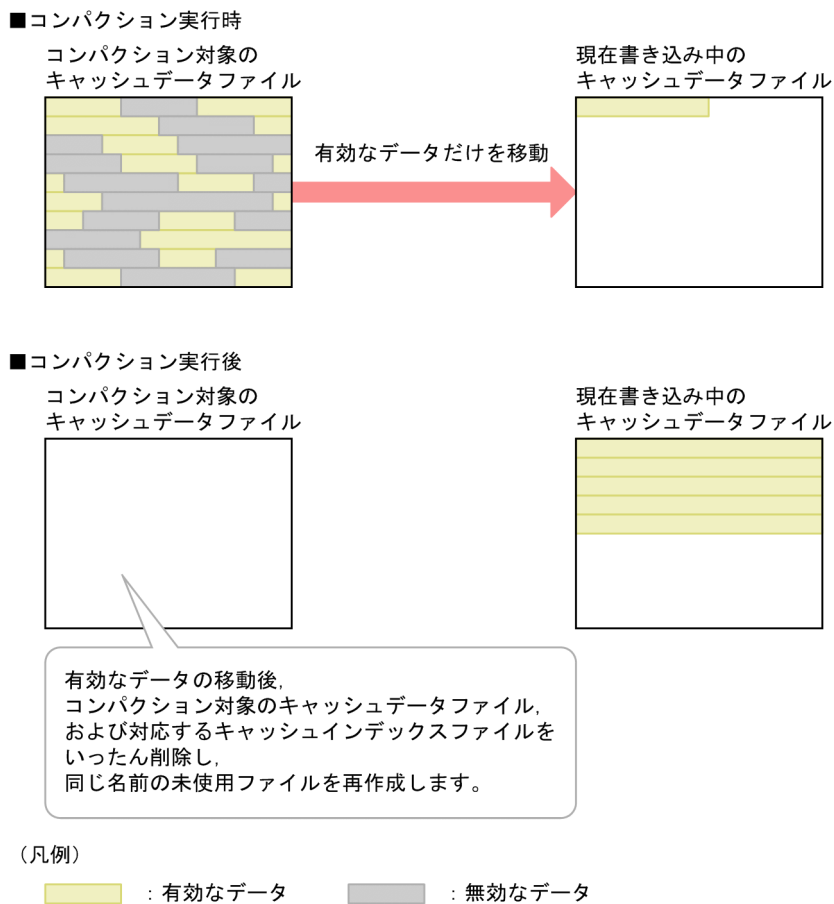
## 10.9 キャッシュデータファイルの使用量を削減する（キャッシュデータファイルをコンパクションする）

この節の内容は、ディスクキャッシュ、および2Wayキャッシュを使用する場合にお読みください。

ディスクキャッシュ、および2Wayキャッシュでは、追記方式でキャッシュデータファイルにデータを格納します。このため、データの更新や削除をすると、ファイル内に無効なデータができます。この無効なデータを削除して、キャッシュデータファイルの使用量を削減することを、コンパクションといいます。

コンパクションの処理を次の図に示します。

図 10-2 コンパクションの処理



なお、コンパクションを実行しても、キャッシュ定義の `eads.cache.disk.filesize` に指定したファイルサイズは変更されません。

### 参考

EADs では、キャッシュデータファイルの更新処理方式として追記方式を採用しています。追記方式では、データを削除した場合、削除対象のデータは無効になりますが、ディスクからは削除され

ません。また、データを更新した場合、更新対象のデータが無効になり、更新後のデータが新たなデータとして追加されます。更新前の無効になったデータは、ディスクからは削除されません。

定期的にコンパクションをすることで、無効なデータを整理し、効率良くディスクを使用することができます。

## 10.9.1 キャッシュデータファイルをコンパクションする手順

キャッシュデータファイルをコンパクションする手順について説明します。

### (1) コンパクションの効果を確認する

`eztool listecf -s` コマンドを実行して、コンパクションの効果を確認します。

```
eztool listecf -s
```

#### コマンド実行例

```
$ eztool listecf -s
KDEA08001-I      The command will now start. (subcommand = listecf, parameter = [listecf, -s])

FC: FileCount
CE: CompactionEffect

Cache  Range  UnusedFC  MaxCE  FilterCE (50%)
-----
cache1  1       6         0      0
cache1  4       6         0      0
cache1  5       6         0      0

cache2  1       6         0      0
cache2  4       6         0      0
cache2  5       6         0      0

-----

KDEA08002-I      The command will now end.
$
```

- UnusedFC 欄に、未使用のキャッシュデータファイルの数が表示されます。データを格納 (put) した順とは無関係に、ランダムにデータの格納 (put) や削除 (remove) をするような運用をする場合、この項目を監視してください。
- MaxCE 欄に、レンジ内のキャッシュデータファイルが持つコンパクションの効果の最大値が表示されます。データを格納 (put) した順に削除 (remove) するような運用をする場合、この項目が 100%、または 100%に近くなったらコンパクションを実行してください。

### (2) クラスタを閉塞状態にする (任意)

必要に応じて、`eztool close` コマンドを実行し、クラスタを閉塞状態にします。

## 参考

EADs サーバが稼働状態（running）の場合もコンパクションは実行できますが、性能に影響が出るおそれがあります。

```
eztool close
```

### コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

## (3) 閉塞していることを確認する（任意）

「10.9.1(2) クラスタを閉塞状態にする（任意）」でクラスタを閉塞状態にした場合は、`eztool status` コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

### コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1   XX.XXX.XXX.168  24600      1288490189 online   closed none
2   XX.XXX.XXX.168  24700      429496730  online   closed none
3   XX.XXX.XXX.168  24800      -429496729 online   closed none
4   XX.XXX.XXX.168  24900      -1288490188 online   closed none
5   XX.XXX.XXX.168  25000      -2147483648 online   closed none
-----
KDEA08002-I      The command will now end.
$
```

閉塞している場合、State 欄に closed が表示されます。

## (4) キャッシュデータファイルをコンパクションする

`eztool compaction` コマンドを実行して、コンパクションを実行します。

```
eztool compaction
```



## コマンド実行例

```
$ eztool compaction
KDEA08001-I The command will now start. (subcommand = compaction, parameter = [compaction])
KDEA08002-I The command will now end.
$
```

### 注意事項

コンパクションは長時間実行される場合が多いため、必要に応じて、タイムアウト時間の設計をしてください。

### (a) データを格納した順に削除するような運用をする場合

データを格納 (put) した順に削除 (remove) するような運用をする場合、データを削除するごとに、キャッシュデータファイル中に無効なデータが増えていきます。このような運用の場合には、無効なデータだけになったキャッシュデータファイルを対象にコンパクションを実行します。--threshold オプションでしきい値を 100% にして `eztool compaction` コマンドを実行してください。

### 参考

データを格納 (put) した順と削除 (remove) する順が完全に一致しないような運用の場合は、しきい値を 100% より小さい値に設定してください。

### (b) データを格納した順とは無関係に、ランダムにデータの格納や削除をするような運用をする場合

データを格納 (put) した順とは無関係に、ランダムにデータの格納 (put) や削除 (remove) をするような運用をする場合、キャッシュデータファイルごとにファイルの使用率が異なります。そのため、データの追加や削除に備えて未使用ファイルを確認しておく必要があります。このような運用の場合には、未使用ファイル数を基準にして、定期的にコンパクションを実行します。確認しておく必要のある未使用ファイル数を --unused\_fc オプションに指定して、`eztool compaction` コマンドを実行してください。

コンパクションを実行しても、予定の未使用ファイル数まで到達しない場合には、--threshold オプションでしきい値を下げて再実行してください。

### (c) 早急にキャッシュデータファイルの空き容量を確保したい場合

例えば未使用ファイル数が非常に少なく、早急にキャッシュデータファイルの空き容量を確保したいような場合は、--threshold オプションで指定するしきい値を小さくしてコンパクションを実行します。--cache オプション、--range オプションで該当するキャッシュやレンジを優先して、`eztool compaction` コマンドを実行してください。

なお、このとき、すでにコンパクションを実行中の場合は、実行中のコンパクションを中断してから、再度 `eztool compaction` コマンドを実行します。コンパクションを中断する手順については、「[10.9.2 コンパクションを中断する手順](#)」を参照してください。

## ■ 参考

キャッシュデータファイルの空き容量が非常に少ないような状態の場合は、可能であれば、クラスタを閉塞状態にしてからコンパクションを実行することを推奨します。

## (5) クラスタの閉塞状態を解除する (任意)

「[10.9.1\(2\) クラスタを閉塞状態にする \(任意\)](#)」でクラスタを閉塞状態にした場合は、`eztool open` コマンドを実行し、クラスタの閉塞状態を解除します。

```
eztool open
```

### コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する (任意)

「[10.9.1\(2\) クラスタを閉塞状態にする \(任意\)](#)」でクラスタを閉塞状態にした場合は、`eztool status` コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

### コマンド実行例

```

$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   running none
 2  XX.XXX.XXX.168  24700      429496730  online   running none
 3  XX.XXX.XXX.168  24800      -429496729 online   running none
 4  XX.XXX.XXX.168  24900      -1288490188 online   running none
 5  XX.XXX.XXX.168  25000      -2147483648 online   running none
-----

KDEA08002-I          The command will now end.
$

```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 10.9.2 コンパクションを中断する手順

コンパクションを実行しているときに、コンパクションを中断する手順について説明します。

### (1) コンパクションを中断する

コンパクションを実行しているときに、`eztool compaction --break` コマンドを実行し、コンパクションを中断します。

```
eztool compaction --break
```

#### コマンド実行例

```

$ eztool compaction --break
KDEA08001-I          The command will now start. (subcommand = compaction, parameter = [compaction, --break])
KDEA08062-I          The compaction of cache data files was stopped.
KDEA08002-I          The command will now end.
$

```

コンパクションを実行中のキャッシュデータファイルの処理が完了したあとで、コンパクションが中断されます。

# 11

## 保守運用

この章では、EADs の保守運用の方法について説明します。

## 11.1 EADs サーバをクラスタに追加したい

---

EADs サーバをクラスタに追加します。

### 11.1.1 EADs サーバをクラスタに追加する手順（メモリ上のキャッシュだけを使用する場合）

メモリキャッシュだけを使用する場合に、EADs サーバをクラスタに追加する手順について説明します。

#### (1) クラスタ内の全 EADs サーバを終了する（ファイルにデータを書き出す）

ファイルにデータを書き出してからクラスタ内の全 EADs サーバを終了します。

手順については、「[10.5 EADs サーバを終了する（メモリ上のデータをファイルに書き出す場合）](#)」を参照してください。

#### (2) 追加する EADs サーバのインストールおよびセットアップを行う

追加する EADs サーバのインストールおよびセットアップを行います。

インストールおよびセットアップの手順については、「[5. インストールおよびセットアップを行う（EADs サーバ）](#)」を参照してください。

#### (3) クラスタ定義を変更する

クラスタを構成する EADs サーバが増えるため、クラスタ定義を変更する必要があります。

定義内容を変更する手順については、「[11.3 定義内容を変更したい](#)」を参照してください。

#### (4) クラスタ内の全 EADs サーバを開始する（ファイルからデータを読み込む）

クラスタ内の全 EADs サーバを起動したあとに、前回稼働時にファイルに書き出したデータを、再度メモリ上に読み込ませます。

手順については、「[10.3 EADs サーバを開始する（ファイルからデータを読み込んでキャッシュを作成する場合）](#)」を参照してください。

## 11.1.2 EADs サーバをクラスタに追加する手順（ディスク上のキャッシュを使用する場合）

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

ディスクキャッシュまたは 2Way キャッシュを使用する場合に、EADs サーバをクラスタに追加する手順について説明します。

### (1) クラスタ内の全 EADs サーバを終了する

クラスタ内の全 EADs サーバを終了します。

手順については、「[10.6 EADs サーバを終了する（ディスク上のキャッシュを終了する場合）](#)」を参照してください。

### (2) 全 EADs サーバのキャッシュファイルを移動する

全 EADs サーバのキャッシュファイルを、任意のディレクトリに移動します。

EADs サーバごとに、次のディレクトリ下のすべてのファイルを移動してください（次のディレクトリ下のすべてのファイルをコピーしたあとで、コピー元のファイルを削除してください）。

- <キャッシュ定義の `eads.cache.disk.<n>.dir` パラメタで指定したディレクトリ>/<キャッシュ名>
- <キャッシュ定義の `eads.cache.disk.info.dir` パラメタで指定したディレクトリ>/<キャッシュ名>（デフォルトは<運用ディレクトリ>/store/<キャッシュ名>）

移動先のディレクトリのパスは、全 EADs サーバで同一になるようにしてください。

### (3) 追加する EADs サーバのインストールおよびセットアップを行う

追加する EADs サーバのインストールおよびセットアップを行います。

インストールおよびセットアップの手順については、「[5. インストールおよびセットアップを行う（EADs サーバ）](#)」を参照してください。

### (4) クラスタ定義を変更する

クラスタを構成する EADs サーバが増えるため、クラスタ定義を変更する必要があります。

定義内容を変更する手順については、「[11.3 定義内容を変更したい](#)」を参照してください。

### (5) クラスタ内の全 EADs サーバを起動する

EADs サーバを起動するホストにログインして `ezstart` コマンドを実行し、EADs サーバを起動します。このコマンドは EADs サーバごとに実行する必要があります。

```
ezstart
```

## (6) 初期化が完了していることを確認する

EADs サーバの起動に成功すると、初期化状態に遷移します。

eztool status コマンドを実行して、全 EADs サーバの初期化が完了していることを確認します。

```
eztool status
```

### コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State      Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   initialized none
 2  XX.XXX.XXX.168  24700      429496730  online   initialized none
 3  XX.XXX.XXX.168  24800      -429496729 online   initialized none
 4  XX.XXX.XXX.168  24900      -1288490188 online   initialized none
 5  XX.XXX.XXX.168  25000      -2147483648 online   initialized none
-----
KDEA08002-I          The command will now end.
$
```

初期化が完了している場合、State 欄に initialized が表示されます。

## (7) データを再配置する

eztool importecf コマンドを実行して、データを再配置します。<キャッシュデータファイルとキャッシュインデックスファイルの格納先パス名>には、「11.1.2(2) 全 EADs サーバのキャッシュファイルを移動する」でキャッシュデータファイルとキャッシュインデックスファイルを移動したディレクトリを指定します。

### 注意事項

<キャッシュデータファイルとキャッシュインデックスファイルの格納先パス名>には、次のディレクトリは指定できません。

- <キャッシュ定義の eads.cache.disk.<n>.dir パラメタで指定したディレクトリ>/<キャッシュ名>
- <キャッシュ定義の eads.cache.disk.info.dir パラメタで指定したディレクトリ>/<キャッシュ名> (デフォルトは<運用ディレクトリ>/store/<キャッシュ名>)

```
eztool importecf <キャッシュデータファイルとキャッシュインデックスファイルの格納先パス名>
```

## コマンド実行例

```
$ eztool importecf temp
KDEA08001-I      The command will now start. (subcommand = importecf, parameter = [importecf, temp])
KDEA08002-I      The command will now end.
$
```

## (8) クラスタの閉塞状態を解除する

初期化状態は閉塞状態と同じく、EADs クライアントからのリクエストが受け付けられない状態です。  
`eztool open` コマンドを実行して、クラスタの閉塞状態を解除します。

```
eztool open
```

## コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (9) 閉塞状態が解除できていることを確認する

`eztool status` コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

## コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   running none
 2  XX.XXX.XXX.168  24700      429496730  online   running none
 3  XX.XXX.XXX.168  24800      -429496729 online   running none
 4  XX.XXX.XXX.168  24900      -1288490188 online   running none
 5  XX.XXX.XXX.168  25000      -2147483648 online   running none
-----
KDEA08002-I      The command will now end.
$
```



閉塞状態が解除できている場合、State 欄に running が表示されます。

## 11.2 EADs サーバをクラスタから削除したい

---

EADs サーバをクラスタから削除します。

### 11.2.1 EADs サーバをクラスタから削除する手順（メモリ上のキャッシュだけを使用する場合）

メモリキャッシュだけを使用する場合に、EADs サーバをクラスタから削除する手順について説明します。

#### (1) クラスタ内の全 EADs サーバを終了する（ファイルにデータを書き出す）

ファイルにデータを書き出してからクラスタ内の全 EADs サーバを終了します。

手順については、「[10.5 EADs サーバを終了する（メモリ上のデータをファイルに書き出す場合）](#)」を参照してください。

#### (2) 削除する EADs サーバのストアデータファイルを退避する

削除する EADs サーバのストアデータファイルを任意のディレクトリに退避します。

削除する EADs サーバのストアデータファイルを、削除しない EADs サーバのストアデータファイル格納先にコピーします。

#### (3) クラスタ定義を変更する

クラスタを構成する EADs サーバが減るため、クラスタ定義を変更する必要があります。

定義内容を変更する手順については、「[11.3 定義内容を変更したい](#)」を参照してください。

#### (4) クラスタ内の全 EADs サーバを開始する（ファイルからデータを読み込む）

クラスタ内の全 EADs サーバを起動したあとに、前回稼働時にファイルに書き出したデータを、再度メモリ上に読み込ませます。

手順については、「[10.3 EADs サーバを開始する（ファイルからデータを読み込んでキャッシュを作成する場合）](#)」を参照してください。

## 11.2.2 EADs サーバをクラスタから削除する手順（ディスク上のキャッシュを使用する場合）

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

ディスクキャッシュまたは 2Way キャッシュを使用する場合に、EADs サーバをクラスタから削除する手順について説明します。

### (1) クラスタ内の全 EADs サーバを終了する

クラスタ内の全 EADs サーバを終了します。

手順については、「[10.6 EADs サーバを終了する（ディスク上のキャッシュを終了する場合）](#)」を参照してください。

### (2) EADs サーバのキャッシュファイルを移動する

削除する EADs サーバ以外の EADs サーバのキャッシュファイルを、任意のディレクトリに移動します。

EADs サーバごとに、次のディレクトリ下のすべてのファイルを移動してください（次のディレクトリ下のすべてのファイルをコピーしたあとで、コピー元のファイルを削除してください）。

- ・ <キャッシュ定義の `eads.cache.disk.<n>.dir` パラメタで指定したディレクトリ>/<キャッシュ名>
- ・ <キャッシュ定義の `eads.cache.disk.info.dir` パラメタで指定したディレクトリ>/<キャッシュ名>  
（デフォルトは<運用ディレクトリ>/store/<キャッシュ名>）

移動先のディレクトリのパスは、全 EADs サーバで同一になるようにしてください。

### (3) 削除する EADs サーバのキャッシュファイルを退避する

削除する EADs サーバのキャッシュファイルを、ほかのどれか 1 つの EADs サーバの、「[11.2.2\(2\) EADs サーバのキャッシュファイルを移動する](#)」でキャッシュファイルを移動したディレクトリに退避します。

削除する EADs サーバの次のディレクトリ下のすべてのファイルを移動してください。

- ・ <キャッシュ定義の `eads.cache.disk.<n>.dir` パラメタで指定したディレクトリ>/<キャッシュ名>
- ・ <キャッシュ定義の `eads.cache.disk.info.dir` パラメタで指定したディレクトリ>/<キャッシュ名>  
（デフォルトは<運用ディレクトリ>/store/<キャッシュ名>）

### (4) クラスタ定義を変更する

クラスタを構成する EADs サーバが減るため、クラスタ定義を変更する必要があります。

定義内容を変更する手順については、「[11.3 定義内容を変更したい](#)」を参照してください。

## (5) クラスタ内の全 EADs サーバを開始する (データを再配置する)

クラスタ内の全 EADs サーバを起動したあとに、データを再配置します。

手順については、「[11.1.2\(5\) クラスタ内の全 EADs サーバを起動する](#)」～「[11.1.2\(9\) 閉塞状態が解除できていることを確認する](#)」を参照してください。

## 11.3 定義内容を変更したい

定義内容を変更します。

### 11.3.1 定義内容を変更する手順

定義内容を変更する手順について説明します。

#### (1) EADs サーバを終了する（ファイルにデータを書き出す）

ファイルにデータを書き出してから EADs サーバを終了します。

手順については、「[10.5 EADs サーバを終了する（メモリ上のデータをファイルに書き出す場合）](#)」を参照してください。

#### 注意事項

コマンド定義を変更する場合は、EADs サーバを終了する必要はありません。

#### (2) 定義内容を変更する

定義内容を変更します。

「`eads.logger.`」で始まるサーバ定義のパラメタを変更する場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

クラスタ定義を変更する場合は、クラスタを構成する全 EADs サーバで定義内容が同じになるようにしてください。

キャッシュ定義を変更する場合は、クラスタを構成する全 EADs サーバで次のパラメタの定義内容が同じになるようにしてください

- `eads.cache.type`
- `eads.cache.disk.filesize`
- `eads.cache.disk.fileenum`
- `eads.cache.disk.blocksize`

「`eads.logger.`」で始まるコマンド定義のパラメタを変更する場合は、コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリを別のディレクトリに移動するか、または削除してください。

### (3) EADs サーバを開始する (ファイルからデータを読み込む)

EADs サーバを起動したあとに、前回稼働時にファイルに書き出したデータを、再度メモリ上に読み込ませませす。手順については、「[10.3 EADs サーバを開始する \(ファイルからデータを読み込んでキャッシュを作成する場合\)](#)」を参照してください。

#### 11.3.2 定義内容を変更する場合の注意点

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

ディスクキャッシュ、および 2Way キャッシュを使用する場合、次に示すパラメータを変更すると、キャッシュの再開ができなくなります。次に示すパラメータを変更したい場合は、データを再配置してください。データの再配置については、「[11.1.2 EADs サーバをクラスタに追加する手順 \(ディスク上のキャッシュを使用する場合\)](#)」を参照してください。

- クラスタ定義の次のパラメータ
  - `eads.node.< EADs サーバ ID >.position` パラメータ
  - `eads.replication.factor` パラメータ
  - `eads.server.max_key_size` パラメータ

ディスクキャッシュ、および 2Way キャッシュを使用する場合、次に示すパラメータは変更しないでください。変更するとキャッシュの再開ができなくなります。

- キャッシュ定義の次のパラメータ
  - `eads.cache.type` パラメータ
  - `eads.cache.disk.filesize` パラメータ

注 キャッシュ定義の `eads.cache.disk.fileenum` パラメータについては、指定値を増やすことはできますが、減らすことはできません。

## 11.4 キャッシュを追加または削除したい

キャッシュを追加または削除します。

### 11.4.1 メモリキャッシュを追加または削除する手順

メモリキャッシュを追加または削除する手順について説明します。

#### (1) クラスタを閉塞状態にする

`eztool close` コマンドを実行し、クラスタを閉塞状態にします。

```
eztool close
```

#### コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

#### (2) 閉塞していることを確認する

EADs サーバを閉塞後、`eztool status` コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

#### コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1   XX.XXX.XXX.168  24600      1288490189 online   closed none
2   XX.XXX.XXX.168  24700      429496730  online   closed none
3   XX.XXX.XXX.168  24800      -429496729 online   closed none
4   XX.XXX.XXX.168  24900      -1288490188 online   closed none
5   XX.XXX.XXX.168  25000      -2147483648 online   closed none
-----
KDEA08002-I      The command will now end.
$
```

閉塞している場合、State 欄に closed が表示されます。

### (3) キャッシュを追加または削除する

#### (a) キャッシュを追加する

`eztool createcache` コマンドを実行し、キャッシュを追加します。

```
eztool createcache <キャッシュ名>
```

#### コマンド実行例

```
$ eztool createcache cache1
KDEA08001-I      The command will now start. (subcommand = createcache, parameter = [createcache, cache1])
KDEA08002-I      The command will now end.
$
```

#### (b) キャッシュを削除する

`eztool deletetecache` コマンドを実行し、キャッシュを削除します。

```
eztool deletetecache <キャッシュ名>
```

#### コマンド実行例

```
$ eztool deletetecache cache1
KDEA08001-I      The command will now start. (subcommand = deletetecache, parameter = [deletetecache, cache1])
KDEA08002-I      The command will now end.
$
```

### (4) キャッシュが追加または削除されていることを確認する

`eztool listcache` コマンドを実行し、キャッシュの一覧を表示します。

キャッシュが追加または削除されていることを確認してください。

```
eztool listcache
```

#### コマンド実行例

```
$ eztool listcache
KDEA08001-I      The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount
cache1     Memory     0
cache2     Memory     0
cache3     Memory     0
-----
KDEA08002-I      The command will now end.
$
```



## (5) クラスタの閉塞状態を解除する

eztool open コマンドを実行し、クラスタの閉塞状態を解除します。

```
eztool open
```

コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する

eztool status コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   running none
 2  XX.XXX.XXX.168  24700      429496730  online   running none
 3  XX.XXX.XXX.168  24800      -429496729 online   running none
 4  XX.XXX.XXX.168  24900      -1288490188 online   running none
 5  XX.XXX.XXX.168  25000      -2147483648 online   running none
-----
KDEA08002-I      The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 11.4.2 ディスクキャッシュおよび 2Way キャッシュを追加または削除する手順

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

ディスクキャッシュ、および 2Way キャッシュを追加または削除する手順について説明します。

## (1) クラスタを閉塞状態にする

eztool close コマンドを実行し、クラスタを閉塞状態にします。

```
eztool close
```

### コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

## (2) 閉塞していることを確認する

EADs サーバを閉塞後、eztool status コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

### コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1   XX.XXX.XXX.168  24600      1288490189 online  closed none
2   XX.XXX.XXX.168  24700      429496730  online  closed none
3   XX.XXX.XXX.168  24800      -429496729 online  closed none
4   XX.XXX.XXX.168  24900      -1288490188 online  closed none
5   XX.XXX.XXX.168  25000      -2147483648 online  closed none
-----
KDEA08002-I      The command will now end.
$
```

閉塞している場合、State 欄に closed が表示されます。

## (3) キャッシュを追加または削除する

### (a) キャッシュを追加する

追加するキャッシュのキャッシュ定義ファイルを作成します。手順については、「[10.2.2\(1\) キャッシュ定義ファイルを作成する](#)」を参照してください。

そのあとで、eztool createcache コマンドを実行し、キャッシュを追加します。

```
eztool createcache <キャッシュ名>
```

## コマンド実行例

```
$ eztool createcache cache1
KDEA08001-I The command will now start. (subcommand = createcache, parameter = [createcache, cache1])
KDEA08002-I The command will now end.
$
```

キャッシュを作成すると、次のディレクトリにキャッシュファイルが作成されます。

- キャッシュデータファイル  
 <キャッシュ定義の `eads.cache.disk.<n>.dir` パラメタで指定したディレクトリ>/<キャッシュ名>
- キャッシュインデックスファイルおよびキャッシュ情報ファイル  
 <キャッシュ定義の `eads.cache.disk.info.dir` パラメタで指定したディレクトリ>/<キャッシュ名>  
 (デフォルトは<運用ディレクトリ>/store/<キャッシュ名>)

## (b) キャッシュを削除する

次のどちらかの方法で、キャッシュおよびキャッシュファイルを削除します。

- `eztool deletetecache --with_deleteecf` コマンドを実行する
- `eztool deletetecache` コマンドを実行したあとに `eztool deleteecf` コマンドを実行する

`eztool deletetecache --with_deleteecf` コマンドを実行する場合

```
eztool deletetecache <キャッシュ名> --with_deleteecf
```

## コマンド実行例

```
$ eztool deletetecache cache1 --with_deleteecf
KDEA08001-I The command will now start. (subcommand = deletetecache, parameter = [deletetecache, cache1,
--with_deleteecf])
KDEA08064-I The cache files were deleted.
KDEA08002-I The command will now end.
$
```

`eztool deletetecache` コマンドを実行したあとに `eztool deleteecf` コマンドを実行する場合

```
eztool deletetecache <キャッシュ名>
```

```
eztool deleteecf <キャッシュ名>
```

## コマンド実行例

```
$ eztool deletetecache cache1
KDEA08001-I The command will now start. (subcommand = deletetecache, parameter = [deletetecache, cache1])
KDEA08002-I The command will now end.
$ eztool deletetecf cache1
KDEA08001-I The command will now start. (subcommand = deletetecf, parameter = [deletetecf, cache1])
KDEA08064-I The cache files were deleted.
KDEA08002-I The command will now end.
$
```

## (4) キャッシュが追加または削除されていることを確認する

キャッシュを追加した場合は、`eztool listcache` コマンドを実行し、キャッシュの一覧を表示します。キャッシュが追加されていることを確認してください。

```
eztool listcache
```

### コマンド実行例

```
$ eztool listcache
KDEA08001-I The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName CacheType TotalMasterKeyCount
cache1 Disk 0
cache2 Disk 0
cache3 Disk 0
-----
KDEA08002-I The command will now end.
$
```

キャッシュを削除した場合は、`eztool listecf -v` コマンドを実行し、永続データに関する情報の一覧を表示します。

```
eztool listecf -v
```

### コマンド実行例

```

$ eztool listecf -v
KDEA08001-1 The command will now start. (subcommand = listecf, parameter = [[listecf, -v]])

CP: CacheProperties
CI: CacheInfoFile
CD: CacheDataFiles
FC: FileCount
CE: CompactionEffect
Ac: Accord
Ex: Exist

Cache  ExCache  AcCP  AcCI  Range  Server  ExCP  ExCI  ExCD  UnusedFC  MaxCE  FilterCE(50%)  CE(0-20%)  (21-40%)  (41-60%)  (61-80%)  (81-100%)
cache1  false  true  true  1  XX.XXX.XXX.168:24600  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24700  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24800  true  false  false  6  0  0  8  0  0  0  0
  2  XX.XXX.XXX.168:24700  true  false  false  5  100  1  7  0  0  0  1
   XX.XXX.XXX.168:24800  true  false  false  5  100  1  7  0  0  0  1
   XX.XXX.XXX.168:24900  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24900  true  false  false  6  0  0  8  0  0  0  0
  3  XX.XXX.XXX.168:24900  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24600  true  false  false  6  0  0  8  0  0  0  0
  4  XX.XXX.XXX.168:25000  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24600  true  false  false  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24700  true  false  false  6  0  0  8  0  0  0  0
cache2  true  true  true  1  XX.XXX.XXX.168:24600  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24700  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24800  true  true  true  6  0  0  8  0  0  0  0
  2  XX.XXX.XXX.168:24700  true  true  true  5  100  1  7  0  0  0  1
   XX.XXX.XXX.168:24800  true  true  true  5  100  1  7  0  0  0  1
   XX.XXX.XXX.168:24900  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24900  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24600  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24600  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24700  true  true  true  6  0  0  8  0  0  0  0
cache3  false  true  true  1  XX.XXX.XXX.168:24600  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24700  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24800  true  true  true  6  0  0  8  0  0  0  0
  2  XX.XXX.XXX.168:24700  true  true  true  5  100  1  7  0  0  0  1
   XX.XXX.XXX.168:24800  true  true  true  5  100  1  7  0  0  0  1
   XX.XXX.XXX.168:24900  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24900  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24600  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:25000  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24600  true  true  true  6  0  0  8  0  0  0  0
   XX.XXX.XXX.168:24700  true  true  true  6  0  0  8  0  0  0  0
KDEA08002-1 The command will now end.
$

```

- `eztool deletocache --with_deleteecf` コマンド, または `eztool deletocache` コマンドのあとに `eztool deleteecf` コマンドを実行した場合

キャッシュおよびキャッシュファイルが正しく削除されていれば, 次の欄に `false` が表示されます。

- ExCache 欄
- ExCI 欄
- ExCD 欄

図中の `cache1` はキャッシュおよびキャッシュファイルが正しく削除されている状態です。

- `eztool deletocache` コマンドだけを実行した場合

キャッシュが正しく削除されていれば, ExCache 欄に `false`, ExCI 欄および ExCD 欄に `true` が表示されます。また, UnusedFC 欄以降に値は表示されません。

図中の `cache3` はキャッシュが正しく削除されている状態です。

## (5) クラスタの閉塞状態を解除する

`eztool open` コマンドを実行し, クラスタの閉塞状態を解除します。

```
eztool open
```

## コマンド実行例

```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する

eztool status コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

## コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1   XX.XXX.XXX.168  24600      1288490189  online  running none
2   XX.XXX.XXX.168  24700      429496730   online  running none
3   XX.XXX.XXX.168  24800      -429496729  online  running none
4   XX.XXX.XXX.168  24900      -1288490188 online  running none
5   XX.XXX.XXX.168  25000      -2147483648 online  running none
-----
KDEA08002-I      The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 11.5 バックアップを取得したい

データのバックアップを取得します。

### 11.5.1 データのバックアップを取得する手順

データのバックアップを取得する手順について説明します。

#### (1) クラスタを閉塞状態にする

`eztool close` コマンドを実行して、クラスタを閉塞状態にします。

```
eztool close
```

#### コマンド実行例

```
$ eztool close
KDEA08001-I      The command will now start. (subcommand = close, parameter = [close])
KDEA08002-I      The command will now end.
$
```

#### (2) 閉塞していることを確認する

EADs サーバを閉塞後、`eztool status` コマンドを実行して、クラスタの状態を確認します。

```
eztool status
```

#### コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
 1  XX.XXX.XXX.168  24600      1288490189 online   closed none
 2  XX.XXX.XXX.168  24700      429496730  online   closed none
 3  XX.XXX.XXX.168  24800      -429496729 online   closed none
 4  XX.XXX.XXX.168  24900      -1288490188 online   closed none
 5  XX.XXX.XXX.168  25000      -2147483648 online   closed none
-----
KDEA08002-I      The command will now end.
$
```

閉塞している場合、State 欄に `closed` が表示されます。

### (3) ファイルにデータを書き出す

`eztool export` コマンドを実行して、ストアデータファイルにデータを書き出します。

```
eztool export
```

#### コマンド実行例

```
$ eztool export
KDEA08001-I The command will now start. (subcommand = export, parameter = [export])
KDEA08055-I The store data file was exported. (store data file key = 20130402183257)
KDEA08002-I The command will now end.
$
```

この例のように、ストアデータファイルキーを省略すると、コマンド実行日時がストアデータファイルキーとなります。

省略しない場合は、任意のストアデータファイルキーを指定します。

### (4) ストアデータファイルを退避する

`eztool listesd -v` コマンドを実行し、取得したストアデータファイルの格納先を確認します。

```
eztool listesd -v
```

#### コマンド実行例

```
$ eztool listesd -v
KDEA08001-I The command will now start. (subcommand = listesd, parameter = [listesd, -v])

TotalCount: 2
export: 1 / 3
stop : 0 / 2
other : 1
latest: 20130402183257

Type StoreDataFileKey FileName FileSize(MB) Server AbsolutePath
export 20130402183257 eads_20130402183257_1.esd 0 XX.XXX.XXX.168:24600 /opt/hitachi/xeads/server/servers/test1/store/eads_20130402183257_1.esd
eads_20130402183257_2.esd 0 XX.XXX.XXX.169:24700 /opt/hitachi/xeads/server/servers/test2/store/eads_20130402183257_2.esd
eads_20130402183257_3.esd 0 XX.XXX.XXX.170:24800 /opt/hitachi/xeads/server/servers/test3/store/eads_20130402183257_3.esd
other test eads_test_1.esd 0 XX.XXX.XXX.168:24600 /opt/hitachi/xeads/server/servers/test1/store/eads_test_1.esd
eads_test_2.esd 0 XX.XXX.XXX.169:24700 /opt/hitachi/xeads/server/servers/test2/store/eads_test_2.esd
eads_test_3.esd 0 XX.XXX.XXX.170:24800 /opt/hitachi/xeads/server/servers/test3/store/eads_test_3.esd

-----
KDEA08002-I The command will now end.
$
```

### (5) クラスタの閉塞状態を解除する

`eztool open` コマンドを実行し、クラスタの閉塞状態を解除します。

```
eztool open
```

#### コマンド実行例



```
$ eztool open
KDEA08001-I      The command will now start. (subcommand = open, parameter = [open])
KDEA08002-I      The command will now end.
$
```

## (6) 閉塞状態が解除できていることを確認する

eztool status コマンドを実行して、全 EADs サーバの閉塞状態が解除できていることを確認します。

```
eztool status
```

### コマンド実行例

```
$ eztool status
KDEA08001-I      The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address  ClientPort  Position  Cluster  State  Operation
1   XX.XXX.XXX.168  24600  1288490189  online  running  none
2   XX.XXX.XXX.168  24700  429496730  online  running  none
3   XX.XXX.XXX.168  24800  -429496729  online  running  none
4   XX.XXX.XXX.168  24900  -1288490188  online  running  none
5   XX.XXX.XXX.168  25000  -2147483648  online  running  none
-----
KDEA08002-I      The command will now end.
$
```

閉塞状態が解除できている場合、State 欄に running が表示されます。

## 11.6 ストアデータファイルを管理したい

ストアデータファイルの世代数には上限があるため、上限に達する前に削除する必要があります。

ストアデータファイルの世代数については、「7.6.2 ストアデータファイルの世代数の設定」を参照してください。

### 11.6.1 ストアデータファイルの確認および削除手順

ストアデータファイルの確認および削除手順について説明します。

#### (1) ストアデータファイルの一覧を表示する

`eztool listesd` コマンドを実行して、クラスタ内のストアデータファイルの一覧を表示します。

```
eztool listesd
```

#### コマンド実行例

```
$ eztool listesd
KDEA08001-I      The command will now start. (subcommand = listesd, parameter = [listesd])

TotalCount: 4
export: 1 / 3
stop  : 1 / 2
other  : 2
latest: stop_20130402183258

Type          StoreDataFileKey
export        20130402183257
stop          stop_20130402183258
other         single_20130402183258
              test
-----
KDEA08002-I      The command will now end.
$
```

StoreDataFileKey 欄に、ストアデータファイルのストアデータファイルキーの一覧が表示されます。

#### (2) 不要なストアデータファイルを削除する

`eztool deleteesd` コマンドを実行して、クラスタ内の不要なストアデータファイルを削除します。

```
eztool deleteesd <ストアデータファイルキー>
```

#### コマンド実行例

```
$ eztool deleteesd test
KDEA08001-I      The command will now start. (subcommand = deleteesd, parameter = [deleteesd, test])
KDEA08002-I      The command will now end.
$
```

## 11.7 データの格納先を確認したい

`eztool getposition` コマンドを実行して、指定した key やグループの格納先 EADs サーバを表示します。

なお、`-l` または `--local` オプションを指定すると、EADs サーバの状態に関係なく実行できます。この場合、稼働中のクラスタ構成情報ではなく、コマンドを実行する EADs サーバのクラスタ定義ファイルの情報を読み込みます。

```
eztool getposition <key>
```

### コマンド実行例

```
$ eztool getposition group:element
KDEA08001-l           The command will now start. (subcommand = getposition, parameter = [getposition, group:element])

ReplicationCount: 3 / 3
HashValue: -1039521297

No.  IP_Address      ClientPort  Position
 1  XX.XXX.XXX.171    24900      -1288490188
 2  XX.XXX.XXX.172    25000      -2147483648
 3  XX.XXX.XXX.168    24600      1288490189
-----

KDEA08002-l           The command will now end.
$
```

データを多重化している場合、No.2 以降にデータのコピー先 EADs サーバが表示されます。

## 11.8 ユーザファンクションまたはユーザイベントハンドラが、各 EADs サーバに正しく配置されているかどうか、また、実行できるかどうか確認したい

ユーザファンクションやユーザイベントハンドラが各 EADs サーバに正しく配置されているかどうか、また、実行できるかどうか確認する手順について説明します。

### 11.8.1 ユーザファンクションの実行可否を表示する

`eztool listfunc` コマンドを実行して、ユーザファンクションの実行可否を表示します。

```
eztool listfunc [-v|--verbose]
```

#### コマンド実行例

```
$ eztool listfunc
KDEA08001-I The command will now start. (subcommand = listfunc, parameter = [listfunc])

FunctionCount: 3

FunctionName  Enable  Disable
FunctionA     3       0
FunctionB     2       1
FunctionC     0       1
-----
KDEA08002-I The command will now end.
$
```

Enable 欄には、ユーザファンクションごとに、実行可能な EADs サーバ数が表示されます。

また、`-v` または `--verbose` オプションを指定すると、各 EADs サーバに配置されているユーザファンクションが確認できます。また、それぞれ実行可能かどうかも確認できます。

```
$ eztool listfunc -v
KDEA08001-I The command will now start. (subcommand = listfunc, parameter = [listfunc, -v])

FunctionCount: 3

Server          FunctionName  Status
-----
XX.XXX.XXX.168:24600 FunctionA     enable
XX.XXX.XXX.168:24600 FunctionB     enable
XX.XXX.XXX.168:24600 FunctionC     none

XX.XXX.XXX.169:24700 FunctionA     enable
XX.XXX.XXX.169:24700 FunctionB     enable
XX.XXX.XXX.169:24700 FunctionC     none

XX.XXX.XXX.170:24800 FunctionA     enable
XX.XXX.XXX.170:24800 FunctionB     disable
XX.XXX.XXX.170:24800 FunctionC     disable
-----
KDEA08002-I The command will now end.
$
```

なお、disable が表示されたユーザファンクションを実行可能な状態にするには、全 EADs サーバを終了したあと、メッセージログから要因を特定し、問題を取り除いてください。

## 11.8.2 ユーザイベントハンドラの実行可否を表示する

eztool listevent コマンドを実行して、ユーザイベントハンドラの実行可否を表示します。

```
eztool listevent [-v|--verbose]
```

### コマンド実行例

```
$ eztool listevent
KDEA08001-I      The command will now start. (subcommand = listevent, parameter = [listevent])

EventCount: 7

EventName  Enable  Disable  Priority  OperationFilter  DataFilter
EventA     3       0        highest  [put, remove]   none
EventB     3       0        UNMATCH  [put]            [cache]
EventC     3       0        high     [remove]        [cache, group]
EventD     3       0        normal   UNMATCH         [cache, group, element]
EventE     2       1        low      [put, remove]   UNMATCH
EventF     1       1        lowest   [put]            none
EventG     0       2

-----
KDEA08002-I      The command will now end.
```

Enable 欄には、ユーザイベントハンドラごとに、実行可能な EADs サーバ数が表示されます。

また、-v または--verbose オプションを指定すると、各 EADs サーバに配置されているユーザイベントハンドラが確認できます。また、それぞれ実行可能かどうかも確認できます。

```
$ eztool listevent -v
KDEA08001-I      The command will now start. (subcommand = listevent, parameter = [listevent, -v])

EventCount: 7

Server      EventName  Status  Priority  OperationFilter  DataFilter
-----
XX.XXX.XXX.168:24600 EventA     enable  highest  [put, remove]   none
XX.XXX.XXX.168:24600 EventB     enable  highest  [put]            [cache = [hoge, foo]]
XX.XXX.XXX.168:24600 EventC     enable  high     [remove]        [cache = [hoge], group = [bar]]
XX.XXX.XXX.168:24600 EventD     enable  normal   [put, remove]   [cache = [foo], group = [boo], element=[bar]]
XX.XXX.XXX.168:24600 EventE     enable  low      [put, remove]   [group = [bar]]
XX.XXX.XXX.168:24600 EventF     enable  lowest   [put]            none
XX.XXX.XXX.168:24600 EventG     disable

XX.XXX.XXX.169:24700 EventA     enable  highest  [put, remove]   none
XX.XXX.XXX.169:24700 EventB     enable  high     [put]            [cache = [hoge, foo]]
XX.XXX.XXX.169:24700 EventC     enable  high     [remove]        [cache = [hoge], group = [bar]]
XX.XXX.XXX.169:24700 EventD     enable  normal   [remove]        [cache = [foo], group = [boo], element=[bar]]
XX.XXX.XXX.169:24700 EventE     enable  low      [put, remove]   [element = [bar]]
XX.XXX.XXX.169:24700 EventF     disable
XX.XXX.XXX.169:24700 EventG     disable

XX.XXX.XXX.170:24800 EventA     enable  highest  [put, remove]   none
XX.XXX.XXX.170:24800 EventB     enable  low      [put]            [cache = [hoge, foo]]
XX.XXX.XXX.170:24800 EventC     enable  normal   [remove]        [cache = [hoge], group = [bar]]
XX.XXX.XXX.170:24800 EventD     enable  low      [put]            [cache = [foo], group = [boo], element=[bar]]
XX.XXX.XXX.170:24800 EventE     disable
XX.XXX.XXX.170:24800 EventF     none
XX.XXX.XXX.170:24800 EventG     none

-----
KDEA08002-I      The command will now end.
```

なお、disable が表示されたユーザイベントハンドラを実行可能な状態にするには、全 EADs サーバを終了したあと、メッセージログから要因を特定し、問題を取り除いてください。

## 11.9 クラスタを稼働させたまま、EADs サーバの修正パッチを適用したい

クラスタを稼働させたまま、EADs サーバの修正パッチを適用します。

通常は、全 EADs サーバを終了してから修正パッチを適用することを推奨します。

修正パッチの適用はマシン 1 台ずつ行います。

なお、1 マシン上に 1EADs サーバのシステム構成ではない場合、次の条件を満たしている必要があります。

- 1 マシン上の EADs サーバ数がデータの多重度未満であること
- 1 マシン上の EADs サーバ数がクラスタを構成する EADs サーバ数の半数未満であること

### 11.9.1 EADs サーバの修正パッチを適用する手順

EADs サーバの修正パッチを適用する手順について説明します。

#### (1) クラスタ内の全 EADs サーバに異常がないことを確認する

`eztool status -v` コマンドを実行して、クラスタ内の全 EADs サーバに異常がないことを確認します。

```
eztool status -v
```

#### (2) 修正パッチを適用する EADs サーバを縮退および終了する

`eztool isolate --stop` コマンドを実行して、修正パッチを適用するマシン上の EADs サーバを縮退および終了します。

```
eztool isolate --stop
```

同じマシン上にほかの EADs サーバがあれば、(2)を繰り返し実行します。

#### (3) マシンに修正パッチを適用する

停止状態の EADs サーバのマシンに修正パッチを適用します。

#### (4) 停止状態の EADs サーバを再起動する

`ezstart -r` または `ezserver -r` コマンドを実行して、修正パッチを適用した EADs サーバを再起動します。

`ezstart -r` コマンドを実行する場合

```
ezstart -r
```



ezserver -r コマンドを実行する場合

```
ezserver -r
```

## (5) 再起動した EADs サーバがクラスタに参加していることを確認する

eztool status -v コマンドを実行して、再起動した EADs サーバがクラスタに参加していることを確認します。

```
eztool status -v
```

### コマンド実行例

```
$ eztool status -v
KDEA08001-I The command will now start. (subcommand = status, parameter = [status, -v])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID IP_Address ServerName ClientPort ServerPort ManagePort Position Cluster State Operation Lock KeyCount UsedCache Version
1 XX.XXX.XXX.168 server01 24600 24633 24620 1288490189 online running none unlock 0 0% ( 0 / 92 MB ) 03-60-01
2 XX.XXX.XXX.168 server02 24700 24733 24720 429496730 online running none unlock 0 0% ( 0 / 92 MB ) 03-60-00
3 XX.XXX.XXX.168 server03 24800 24833 24820 -429496729 online running none unlock 0 0% ( 0 / 92 MB ) 03-60-00
4 XX.XXX.XXX.168 server04 24900 24933 24920 -1288490188 online running none unlock 0 0% ( 0 / 92 MB ) 03-60-00
5 XX.XXX.XXX.168 server05 25000 25033 25020 -2147483648 online running none unlock 0 0% ( 0 / 92 MB ) 03-60-00

KDEA08002-I The command will now end.
$
```

クラスタに参加している場合、Cluster 欄に online が表示されます。

ほかにも停止状態の EADs サーバがあれば、(4)と(5)を繰り返し実行します。

また、Version 欄にバージョン情報が表示されます。正しいバージョンとなっているかどうかを確認します。

## (6) クラスタ内の全 EADs サーバがクラスタに参加していることを確認する

eztool status -v コマンドを実行して、クラスタ内の全 EADs サーバがクラスタに参加していることを確認します。

```
eztool status -v
```

すべてのマシンに対して(1)~(6)の手順を繰り返し実行します。

## 11.10 統計情報を取得したい

EADs では次に示す統計情報を取得できます。

- 統計情報 (eads\_stats.csv)
- キャッシュの統計情報 (eads\_cache\_stats.csv)
- ユーザ関クションの統計情報 (eads\_function\_stats.csv)
- ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv)

ここでは、統計情報の格納先とそれぞれの統計情報に出力される情報、統計情報が取得される個所について説明します。

### 11.10.1 統計情報の格納先

統計情報は<サーバ定義の `eads.logger.dir` パラメタに指定されたディレクトリ>/stats 下に格納されています。

### 11.10.2 統計情報 (eads\_stats.csv)

統計情報 (eads\_stats.csv) に出力される情報と、統計情報が取得される個所について説明します。

#### (1) 統計情報 (eads\_stats.csv) に出力される情報

統計情報 (eads\_stats.csv) に出力される情報を、次の表に示します。

表 11-1 統計情報 (eads\_stats.csv) に出力される情報

項番	目的	分類	項目	列名称	出力される値	関連するパラメタ
1	状況の確認	共通項目	統計情報を出力した日付	Date	現在値	-
2			統計情報を出力した時間	Time	現在値	
3			統計情報を出力した EADs サーバの IP アドレスと、EADs クライアントとの通信に使用する EADs サーバのポート番号	ThisNode	設定値	<ul style="list-style-type: none"><li>• <code>eads.server.address</code></li><li>• <code>eads.server.port</code></li></ul>

項番	目的	分類	項目	列名称	出力される値	関連するパラメタ
4	チューニングの見直し	メモリ・バッファ	EADs クライアント・EADs サーバ間の受信データの最大サイズ (単位: バイト)	CSReadMaxSize	統計値	<ul style="list-style-type: none"> <li>• <a href="#">eads.connection.buffersize</a></li> <li>• <a href="#">eads.server.max.datasize</a></li> </ul>
5			EADs クライアント・EADs サーバ間の受信データの合計サイズ (単位: バイト)	CSReadTotalSize	統計値	<ul style="list-style-type: none"> <li>• <a href="#">eads.connection.buffersize</a></li> <li>• <a href="#">eads.consensus.message.buffersize</a></li> </ul>
6			EADs クライアント・EADs サーバ間の送信データの最大サイズ (単位: バイト)	CSWriteMaxSize	統計値	
7			EADs クライアント・EADs サーバ間の送信データの合計サイズ (単位: バイト)	CSWriteTotalSize	統計値	
8			EADs サーバ間の受信データの最大サイズ (単位: バイト)	SSReadMaxSize	統計値	
9			EADs サーバ間の受信データの合計サイズ (単位: バイト)	SSReadTotalSize	統計値	
10			EADs サーバ間の送信データの最大サイズ (単位: バイト)	SSWriteMaxSize	統計値	
11			EADs サーバ間の送信データの合計サイズ (単位: バイト)	SSWriteTotalSize	統計値	
12			復旧時に復旧先 EADs サーバにデータをコピーするために送信されるデータの合計サイズ (単位: バイト)	RestoreCopyDataWriteTotalSize	統計値	<a href="#">eads.restore.dataSender.sendSize</a>

項番	目的	分類	項目	列名称	出力される値	関連するパラメタ
13			復旧時に復旧先 EADs サーバに送信される全データ ※3 の合計サイズ (単位: バイト)	RestoreWriteTotalSize	統計値	
14	リソース見積もりの見直し	データ数	キャッシュに格納されている key 数	KvCount	現在値	eads.node.< EADs サーバ ID >.position
15		メモリ使用量	現在のメモリ使用量 (メモリを使用するキャッシュに格納されている value の合計) (単位: メガバイト) ※4	UsedMemorySize	現在値	
16			value を格納できるメモリの最大サイズ (単位: メガバイト) ※4	MaxMemorySize	設定値	<ul style="list-style-type: none"> <li>eads.java.heapsize</li> <li>eads.java.external.heapsize</li> </ul>
17			更新操作の履歴が格納される領域の使用量 (単位: メガバイト) ※4	ConsensusMemorySize	現在値	eads.consensus.external.heapsize
18			更新操作の履歴が格納される領域の最大サイズ (単位: メガバイト) ※4	MaxConsensusMemorySize	設定値	
19		実行回数	リクエスト受付数 (put, create, update, replace, get, remove)	RequestCount	統計値	eads.node.< EADs サーバ ID >.position
20			全ユーザファンクションの実行回数	AllFunctionExecuteCount	統計値	
21		全ユーザイベントハンドラの実行回数	AllEventHandlerExecuteCount	統計値	—	
22	EADs サーバの性能測定	性能	内部処理時間の平均値※1 (単位: マイクロ秒)	InternalProcessingAverageTime	統計値	

項番	目的	分類	項目	列名称	出力される値	関連するパラメタ
23			リクエスト処理時間の最大値※2 (単位：マイクロ秒)	RequestProcessingMaxTime	統計値	
24			リクエスト処理時間の平均値※2 (単位：マイクロ秒)	RequestProcessingAverageTime	統計値	
25			リクエスト処理時間の最小値※2 (単位：マイクロ秒)	RequestProcessingMinTime	統計値	

(凡例)

－：関連するパラメタはありません。

注※1

データの多重化およびユーザイベントハンドラを除く、次に示すリクエスト処理に掛かった時間の平均値です。

- put, remove の処理
- create, update, replace の処理で、データ更新が行われた場合

注※2

最初にデータが操作される EADs サーバに対するリクエスト処理 (put, remove の処理, または create, update, replace の処理でデータ更新が行われた場合) 全体に掛かった時間が対象です。

注※3

復旧時には、次のデータが復旧先 EADs サーバに送信されます。

- コピー用のデータ
- 復旧開始後にデータが更新された場合、更新を通知するデータ

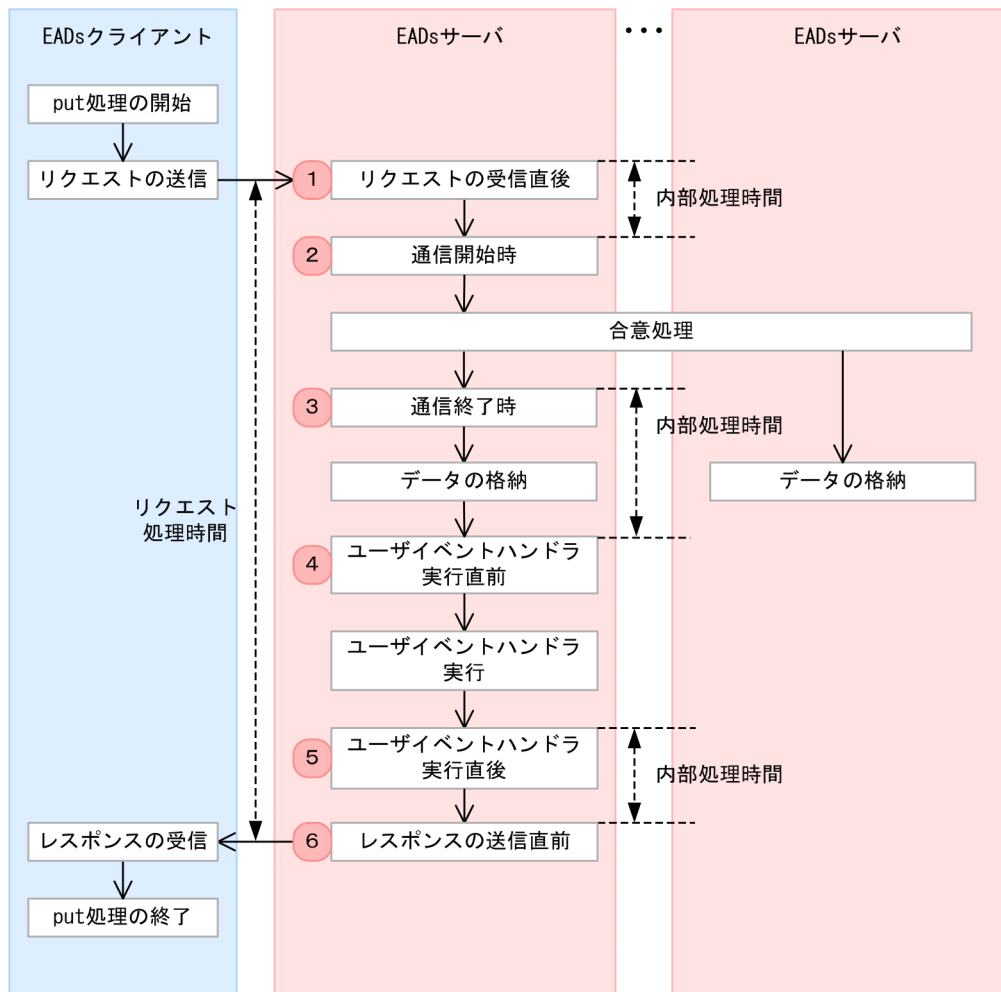
注※4

メガバイト以下の値は切り捨てて表示されます。

## (2) EADs サーバの性能測定に使用する統計情報が取得される個所

EADs サーバの性能測定に使用する統計情報が取得される個所を、put 処理を例にして次の図に示します。

図 11-1 EADs サーバの性能測定に使用する統計情報が取得される箇所 (put 処理の例)



## 説明

- 1 と 6 のタイミングで取得した時刻の差分から、リクエスト処理時間を算出します。また、次に示す統計情報を算出します。
  - リクエスト処理時間の最大値 (RequestProcessingMaxTime)
  - リクエスト処理時間の平均値 (RequestProcessingAverageTime)
  - リクエスト処理時間の最小値 (RequestProcessingMinTime)
- 2 と 3 のタイミングで取得した時刻の差分から、合意処理に掛かった時間を算出します。
- 4 と 5 のタイミングで取得した時刻の差分から、ユーザイベントハンドラ実行時間を算出します。
- リクエスト処理時間から、合意処理に掛かった時間、およびユーザイベントハンドラ実行時間を除いた時間が、内部処理時間に当たります。また、次に示す統計情報を算出します。
  - 内部処理時間の平均値 (InternalProcessingAverageTime)

### 11.10.3 キャッシュの統計情報 (eads\_cache\_stats.csv)

キャッシュの統計情報 (eads\_cache\_stats.csv) に出力される情報と、統計情報が取得される個所について説明します。

#### (1) キャッシュの統計情報 (eads\_cache\_stats.csv) に出力される情報

キャッシュの統計情報 (eads\_cache\_stats.csv) に出力される情報を、次の表に示します。

表 11-2 キャッシュの統計情報 (eads\_cache\_stats.csv) に出力される情報

項番	目的	分類	項目	列名称	出力される値	関連するパラメタ	
1	状況の確認	共通項目	統計情報を出力した日付	Date	現在値	-	
2			統計情報を出力した時間	Time	現在値		
3			統計情報を出力した EADs サーバの IP アドレスと、EADs クライアントとの通信に使用する EADs サーバのポート番号	ThisNode	設定値		<ul style="list-style-type: none"> <li>eads.server.address</li> <li>eads.server.port</li> </ul>
4	チューニングの見直し	キャッシュの設定	キャッシュ名*1	CacheName	設定値	-	
5			キャッシュタイプ	CacheType	設定値		<ul style="list-style-type: none"> <li>eads.cache.type</li> </ul>
6			キャッシュデータファイルの出力先ディレクトリに格納されているデータのレンジ ID*2	RangeID	設定値		<ul style="list-style-type: none"> <li>eads.replication.factor</li> </ul>
7		キャッシュの使用量	キャッシュに格納されている key 数	CacheKvCount	現在値	<ul style="list-style-type: none"> <li>eads.node.&lt; EADs サーバ ID &gt;.position</li> </ul>	
8			現在キャッシュが使用しているメモリ容量 (value の合計) (単位: メガバイト) *1	CacheUsedMemorySize	現在値		
9			現在キャッシュが使用しているディスク容量 (value の合計) (単位: メガバイト) *3	CacheUsedDiskSize	現在値		

項番	目的	分類	項目	列名称	出力される値	関連するパラメタ
10			キャッシュごとのキャッシュデータファイル1つ当たりの最大サイズ (単位:メガバイト) ※3	CacheMaxFileSize	設定値	<ul style="list-style-type: none"> <li>• eads.cache.disk.filesize</li> </ul>
11			キャッシュごとの最大キャッシュデータファイル数	CacheMaxFileCount	設定値	<ul style="list-style-type: none"> <li>• eads.cache.disk.filenum</li> </ul>
12			レンジごとの使用していないキャッシュデータファイル数	RangeUnusedFileCount	現在値	<ul style="list-style-type: none"> <li>• eads.cache.disk.filesize</li> <li>• eads.cache.disk.filenum</li> </ul>
13		コンパクション	レンジごとの使用しているキャッシュデータファイルの中で、最もコンパクションの効果があるファイルの圧縮率 (単位: %)	RangeMaxFileCompactionEffect	現在値	
14			レンジごとの使用しているキャッシュデータファイルの中で、コンパクションの圧縮率が各範囲に当てはまるファイル数	RangeFileCompactionEffectCount	現在値	<ul style="list-style-type: none"> <li>• eads.cache.disk.filesize</li> <li>• eads.cache.disk.filenum</li> <li>• eads.statistics.compaction.effect.division</li> </ul>
15	EADs サーバの性能測定	性能	キャッシュごとの内部処理時間の平均値※4 (単位: マイクロ秒)	CacheInternalProcessingAverageTime	統計値	—
16			キャッシュごとのリクエスト処理時間の最大値※5 (単位: マイクロ秒)	CacheRequestProcessingMaxTime	統計値	
17			キャッシュごとのリクエスト処理時間の平均値※5 (単位: マイクロ秒)	CacheRequestProcessingAverageTime	統計値	
18			キャッシュごとのリクエスト処理時間	CacheRequestProcessingMinTime	統計値	



項番	目的	分類	項目	列名称	出力される値	関連するパラメタ
			間の最小値※5 (単位：マイクロ秒)			
19			キャッシュごとのputの回数	PutCount	統計値	
20			キャッシュごとのputの成功回数	PutSuccessCount	統計値	
21			キャッシュごとのgetの回数	GetCount	統計値	
22			キャッシュごとのgetの成功回数	GetSuccessCount	統計値	
23			キャッシュごとのremoveの回数	RemoveCount	統計値	
24			キャッシュごとのremoveの成功回数	RemoveSuccessCount	統計値	
25			キャッシュごとのcreateの回数	CreateCount	統計値	
26			キャッシュごとのcreateの成功回数	CreateSuccessCount	統計値	
27			キャッシュごとのupdateの回数	UpdateCount	統計値	
28			キャッシュごとのupdateの成功回数	UpdateSuccessCount	統計値	
29			キャッシュごとのreplaceの回数	ReplaceCount	統計値	
30			キャッシュごとのreplaceの成功回数	ReplaceSuccessCount	統計値	

(凡例)

－：関連するパラメタはありません。

注※1

キャッシュが複数ある場合は、上からキャッシュ名の自然順序順に表示されます。

注※2

データの多重度が2以上の場合は、上からデータの格納先 EADs サーバ、データのコピー先 EADs サーバの順に表示されます。

### 注※3

メガバイト以下の値は切り捨てて表示されます。

### 注※4

データの多重化およびユーザイベントハンドラを除く、次に示すリクエスト処理に掛かった時間の平均値です。

- put, remove の処理
- create, update, replace の処理で、データ更新が行われた場合

### 注※5

最初にデータが操作される EADs サーバに対するリクエスト処理 (put, remove の処理, または create, update, replace の処理でデータ更新が行われた場合) 全体に掛かった時間が対象です。

## 11.10.4 ユーザファンクションの統計情報 (eads\_function\_stats.csv)

ユーザファンクションの統計情報 (eads\_function\_stats.csv) に出力される情報と、統計情報が取得される個所について説明します。

### (1) ユーザファンクションの統計情報 (eads\_function\_stats.csv) に出力される情報

ユーザファンクションの統計情報 (eads\_function\_stats.csv) に出力される情報を、次の表に示します。

これらの情報は、ユーザファンクションの性能測定に使用できます。

表 11-3 ユーザファンクションの統計情報 (eads\_function\_stats.csv) に出力される情報

項番	項目	列名称	出力される値
1	統計情報を出力した日付	Date	現在値
2	統計情報を出力した時間	Time	現在値
3	統計情報を出力した EADs サーバの IP アドレスと、EADs クライアントとの通信に使用する EADs サーバのポート番号	ThisNode	設定値
4	ユーザファンクション名	FunctionName	設定値
5	ユーザファンクションの実行回数	FunctionExecuteCount	統計値
6	ユーザファンクション実行時間の最大値 (単位: マイクロ秒)	FunctionExecuteMaxTime	統計値
7	ユーザファンクション実行時間の平均値 (単位: マイクロ秒)	FunctionExecuteAverageTime	統計値

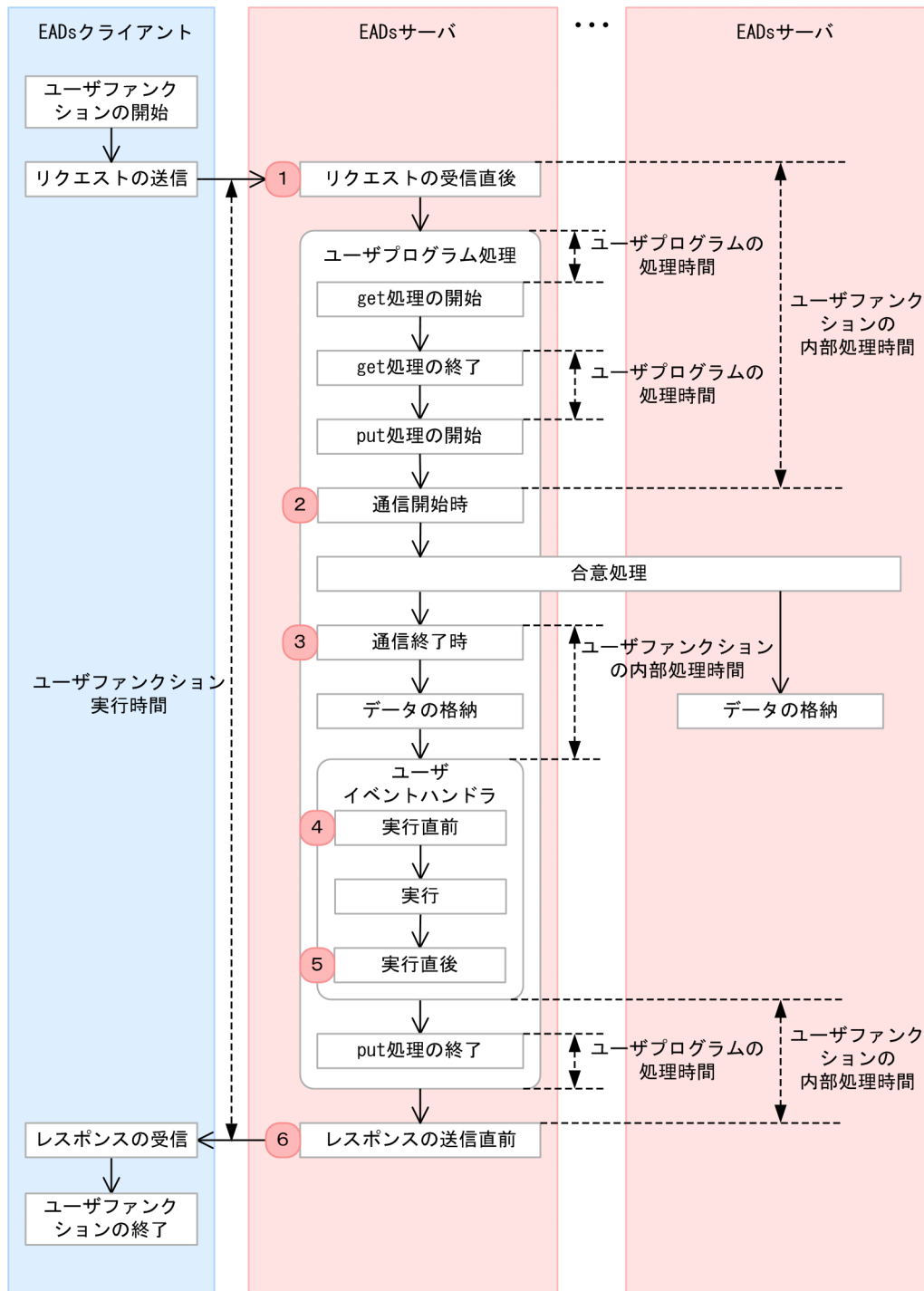
項番	項目	列名称	出力される値
8	ユーザファンクション実行時間の最小値 (単位: マイクロ秒)	FunctionExecuteMinTime	統計値
9	ユーザファンクションの内部処理時間の最大値 (単位: マイクロ秒)	FunctionInternalProcessingMaxTime	統計値
10	ユーザファンクションの内部処理時間の平均値 (単位: マイクロ秒)	FunctionInternalProcessingAverageTime	統計値
11	ユーザファンクションの内部処理時間の最小値 (単位: マイクロ秒)	FunctionInternalProcessingMinTime	統計値
12	ユーザファンクションのユーザプログラムの処理時間の最大値 (単位: マイクロ秒)	UserProgramProcessingMaxTime	統計値
13	ユーザファンクションのユーザプログラムの処理時間の平均値 (単位: マイクロ秒)	UserProgramProcessingAverageTime	統計値
14	ユーザファンクションのユーザプログラムの処理時間の最小値 (単位: マイクロ秒)	UserProgramProcessingMinTime	統計値
15	ユーザファンクションから呼ばれた put の回数	PutCount	統計値
16	ユーザファンクションから呼ばれた put の成功回数	PutSuccessCount	統計値
17	ユーザファンクションから呼ばれた get の回数	GetCount	統計値
18	ユーザファンクションから呼ばれた get の成功回数	GetSuccessCount	統計値
19	ユーザファンクションから呼ばれた remove の回数	RemoveCount	統計値
20	ユーザファンクションから呼ばれた remove の成功回数	RemoveSuccessCount	統計値
21	ユーザファンクションから呼ばれた create の回数	CreateCount	統計値
22	ユーザファンクションから呼ばれた create の成功回数	CreateSuccessCount	統計値
23	ユーザファンクションから呼ばれた update の回数	UpdateCount	統計値
24	ユーザファンクションから呼ばれた update の成功回数	UpdateSuccessCount	統計値
25	ユーザファンクションから呼ばれた replace の回数	ReplaceCount	統計値

項番	項目	列名称	出力される値
26	ユーザファンクションから呼ばれた replace の成功回数	ReplaceSuccessCount	統計値

## (2) 統計情報が取得される個所

統計情報が取得される個所を次の図に示します。

図 11-2 統計情報が取得される個所 (ユーザファンクションの統計情報)



## 説明

- ユーザファンクション実行時間とは、ユーザファンクション実行のリクエストを受け付けてから、レスポンスを返すまでの処理時間のことです。1と6のタイミングで取得した時刻の差分から、ユーザファンクション実行時間を算出します。また、次に示す統計情報を算出します。
  - ユーザファンクション実行時間の最大値 (FunctionExecuteMaxTime)
  - ユーザファンクション実行時間の平均値 (FunctionExecuteAverageTime)
  - ユーザファンクション実行時間の最小値 (FunctionExecuteMinTime)
- 2と3のタイミングで取得した時刻の差分から、合意処理に掛かった時間を算出します。
- 4と5のタイミングで取得した時刻の差分から、ユーザイベントハンドラ実行時間を算出します。
- ユーザファンクション実行時間から、合意処理に掛かった時間、およびユーザイベントハンドラ実行時間を除いた時間が、ユーザファンクションの内部処理時間に当たります。また、次に示す統計情報を算出します。
  - ユーザファンクションの内部処理時間の最大値 (FunctionInternalProcessingMaxTime)
  - ユーザファンクションの内部処理時間の平均値 (FunctionInternalProcessingAverageTime)
  - ユーザファンクションの内部処理時間の最小値 (FunctionInternalProcessingMinTime)
- ユーザプログラム処理のうち、合意処理に掛かった時間、ユーザイベントハンドラ実行時間、およびAPIの処理時間を除いた時間が、ユーザプログラムの処理時間に当たります。また、次に示す統計情報を算出します。
  - ユーザファンクションのユーザプログラムの処理時間の最大値 (UserProgramProcessingMaxTime)
  - ユーザファンクションのユーザプログラムの処理時間の平均値 (UserProgramProcessingAverageTime)
  - ユーザファンクションのユーザプログラムの処理時間の最小値 (UserProgramProcessingMinTime)

### 11.10.5 ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv)

ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv) に出力される情報と、統計情報が取得される個所について説明します。

#### (1) ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv) に出力される情報

ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv) に出力される情報を、次の表に示します。

これらの情報は、ユーザイベントハンドラの性能測定に使用できます。

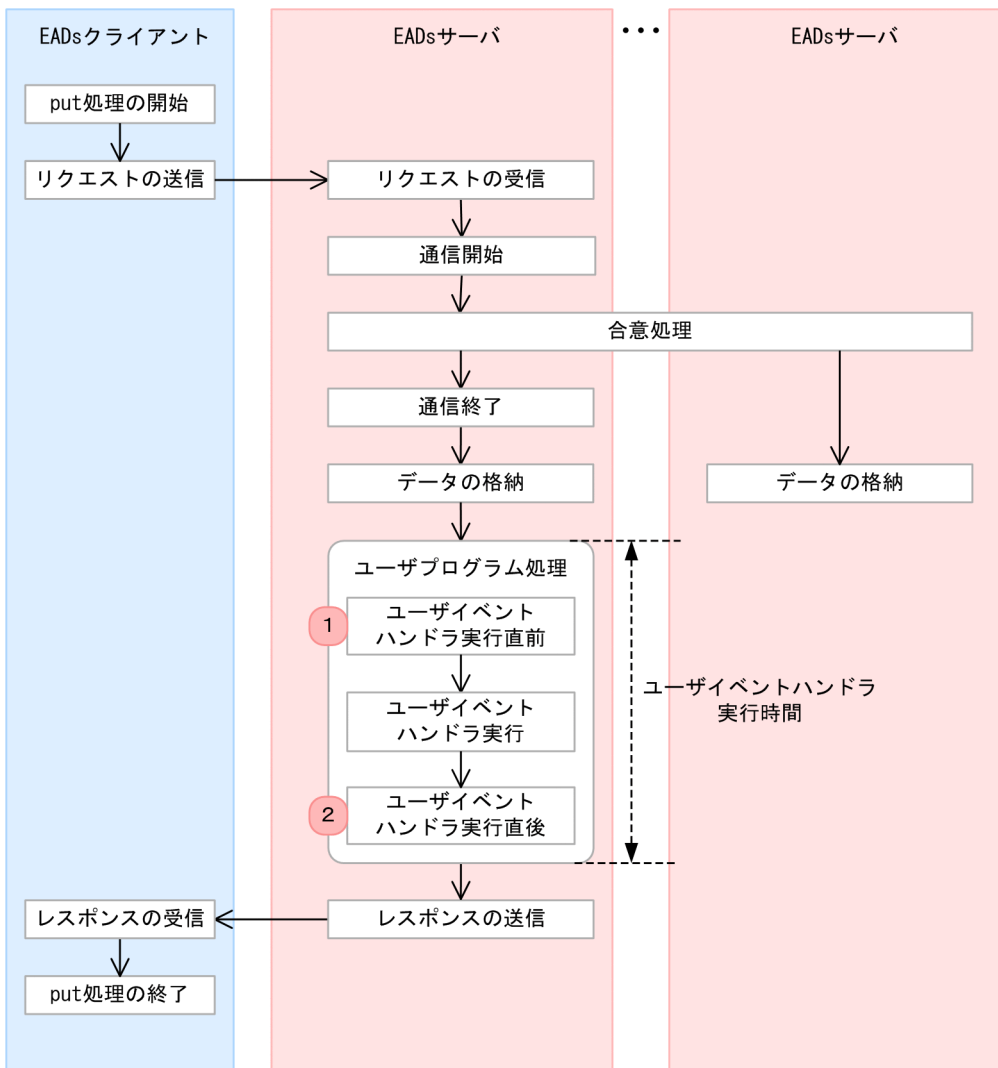
表 11-4 ユーザイベントハンドラの統計情報 (eads\_event\_stats.csv) に出力される情報

項番	項目	列名称	出力される値
1	統計情報を出力した日付	Date	現在値
2	統計情報を出力した時間	Time	現在値
3	統計情報を出力した EADs サーバの IP アドレスと、EADs クライアントとの通信に使用する EADs サーバのポート番号	ThisNode	設定値
4	ユーザイベントハンドラ名	EventHandlerName	設定値
5	ユーザイベントハンドラの実行回数	EventHandlerExecuteCount	統計値
6	ユーザイベントハンドラ実行時間の最大値 (単位：マイクロ秒)	EventHandlerExecuteMaxTime	統計値
7	ユーザイベントハンドラ実行時間の平均値 (単位：マイクロ秒)	EventHandlerExecuteAverageTime	統計値
8	ユーザイベントハンドラ実行時間の最小値 (単位：マイクロ秒)	EventHandlerExecuteMinTime	統計値

## (2) 統計情報が取得される個所

統計情報が取得される個所を次の図に示します。

図 11-3 統計情報が取得される箇所（ユーザイベントハンドラの統計情報）



説明

- 1 と 2 のタイミングで時刻を取得し、次に示す統計情報を算出します。
  - ユーザイベントハンドラ実行時間の最大値 (EventHandlerExecuteMaxTime)
  - ユーザイベントハンドラ実行時間の平均値 (EventHandlerExecuteAverageTime)
  - ユーザイベントハンドラ実行時間の最小値 (EventHandlerExecuteMinTime)

## 11.11 キャッシュファイルを管理したい

この節の内容は、ディスクキャッシュ、および2Wayキャッシュを使用する場合にお読みください。

定義を誤った状態でキャッシュを作成してしまったときなどに、不要なキャッシュファイルを削除します。

### 11.11.1 キャッシュファイルの確認および削除手順

キャッシュファイルの確認および削除手順について説明します。

#### (1) キャッシュファイルが存在するキャッシュを確認する

eztool listecf -v コマンドを実行して、キャッシュファイルが存在するキャッシュを確認します。

```
eztool listecf -v
```

#### コマンド実行例

```
$ eztool listecf -v
KDEA08001-1 The command will now start. (subcommand = listecf, parameter = [listecf, -v])

OP: CacheProperties
CI: CacheInfoFile
CD: CacheDataFiles
FC: FileCount
CE: CompactionEffect
Ac: Accord
Ex: Exist

Cache  ExCache  AcCP  AcCI  Range  Server  ExCP  ExCI  ExCD  UnusedFC  MaxCE  FilterCE (50%)  CE (0-20%)  (21-40%)  (41-60%)  (61-80%)  (81-100%)
cache1 true     true  true  1  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0
  2  XX.XXX.XXX.168:24700 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24800 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24900 true true true 5 100 1 7 0 0 0 0 1
  3  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  4  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  5  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
cache2 true     true  true  1  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  2  XX.XXX.XXX.168:24700 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24800 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24900 true true true 5 100 1 7 0 0 0 0 1
  3  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  4  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  5  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
cache3 true     true  true  1  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  2  XX.XXX.XXX.168:24700 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24800 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24900 true true true 5 100 1 7 0 0 0 0 1
  3  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  4  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  5  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0

KDEA08002-1 The command will now end.
$
```

- キャッシュのキャッシュ情報ファイルが存在する場合は、ExCI 欄に ture が表示されます。



- キャッシュのキャッシュデータファイルが存在する場合は、ExCD 欄に ture が表示されます。

## (2) 不要なキャッシュファイルを削除する

eztool deletectf コマンドを実行して、クラスタ内の不要なキャッシュファイルを削除します。

```
eztool deletectf <キャッシュ名>
```

### コマンド実行例

```
$ eztool deletectf cache1
KDEA08001-I The command will now start. (subcommand = deletectf, parameter = [deletectf, cache1])
KDEA08064-I The cache files were deleted.
KDEA08002-I The command will now end.
$
```

# 12

## 障害運用

この章では、障害発生時にシステム運用管理者が行うことについて説明します。

## 12.1 障害に備えた運用（障害監視と障害検知）

JP1などのシステム運用管理ソフトウェアを使用して、EADsが出力するメッセージやEADsサーバプロセスを監視することで、システム運用管理者に障害の発生やEADsサーバプロセスの状態を通知することができます。

### 12.1.1 メッセージの監視

メッセージの監視を行い、障害の発生をシステム運用管理者に通知します。

メッセージを監視する場合は、次のログを監視します。

監視対象		出力先	ローテーション方式	ログファイル名	形式	改行コード
プロセス	ログ					
EADsサーバ	メッセージログ	サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ	Wrap (デフォルト)	<code>eads_server_message[n].log</code>	ラップアラウンドファイル ※1	CR + LF
			Shift	<code>eads_server_message.log</code>		
コマンド		コマンド定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ	Wrap (デフォルト)	<code>eads_management_message[n].log</code>		
			Shift	<code>eads_management_message.log</code>		
EADsクライアント		<クライアント定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/<EADsクライアント名※2>	Wrap	<code>eads_client_message[n].log</code>		

(凡例)

[n]：ファイル数の通番です。

Wrap：ラップアラウンド方式

Shift：シフト方式

注※1

ラップアラウンド時、データを削除して再び先頭からデータを書き込む形式のファイルです。

注※2

クライアントAPIで指定したEADsクライアント名です。EADsクライアント名が空文字列の場合、EADsクライアント名のサブディレクトリを省略して出力します。

## 12.1.2 EADs サーバプロセスの監視

EADs サーバプロセスの監視を行い、プロセスの状態や障害の発生をシステム運用管理者に通知します。

EADs サーバプロセスを監視する場合は、EADs サーバプロセスごとに、定期的に `eztool status -s` コマンドを実行し、リターンコードから EADs サーバの状態を確認します。

`eztool status -s` コマンドのリターンコードについては、「[14.3.4\(6\) リターンコード](#)」を参照してください。

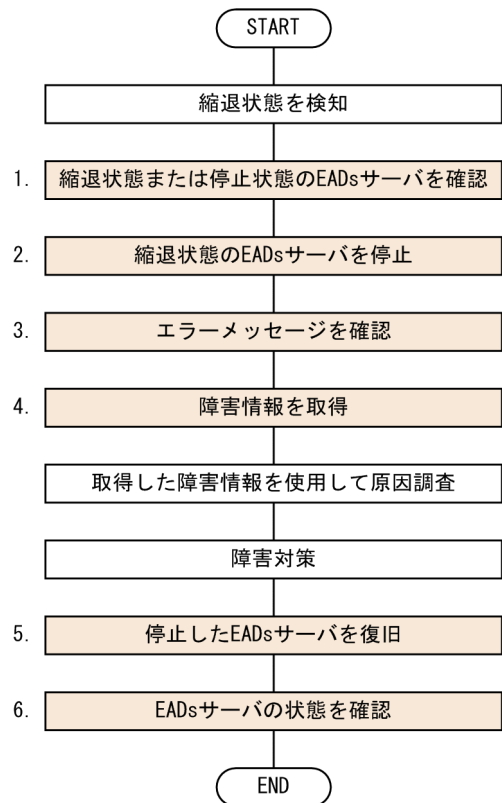
## 12.2 障害発生時のシステム運用管理者の作業

障害が発生した場合、システム運用管理者はエラーメッセージの内容を確認し、障害情報を取得してください。ここでは、障害発生時のシステム運用管理者の作業を障害別に説明します。

### 12.2.1 縮退状態が発生した場合

縮退状態が発生した場合の復旧までの流れを次の図に示します。

図 12-1 縮退状態が発生した場合の復旧までの流れ



(凡例)  : システム運用管理者が行う作業

#### 注意事項

次の場合、ここで説明する手順では復旧できません。

- クラスタの状態が、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスター一部稼働中 (PARTIALLY\_AVAILABLE) の場合  
なお、クラスタの状態が AVAILABLE でも、半数以上の EADs サーバが縮退している場合は、クラスタ動作不能 (NOT\_AVAILABLE) の場合と同様の対処が必要です。
- オンライン性能の劣化が許容できない場合

- 復旧する EADs サーバが、クラスタ定義に定義されていない場合
- EADs サーバダウン時のクラスタ定義と、復旧時のクラスタ定義が異なる場合

クラスタの状態が、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の場合の復旧手順は、「[12.2.2 クラスタ動作不能 \(NOT\\_AVAILABLE\) またはクラスタ一部稼働中 \(PARTIALLY\\_AVAILABLE\) の場合](#)」を参照してください。

システム運用管理者が行う各作業について説明します。

## (1) 縮退状態または停止状態の EADs サーバを確認する

`eztool status` コマンドで、縮退状態または停止状態の EADs サーバを確認します。

### コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 4
OfflineCount: 1
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State   Operation
 1  XX.XXX.XXX.168   24600      1288490189 online   running none
 2  XX.XXX.XXX.168   24700      429496730  online   running none
 3  XX.XXX.XXX.168   24800      -429496729 online   running none
 4  XX.XXX.XXX.168   24900      -1288490188 online   running none
 5  XX.XXX.XXX.168   25000      -2147483648 offline  isolated none
-----
KDEA08002-I          The command will now end.
$
```

この例の場合、State 欄に `isolated` と表示されている EADs サーバが縮退状態になっています。この EADs サーバを停止する必要があります。

なお、この例の場合、停止状態の EADs サーバはありません。停止状態の EADs サーバがある場合は、State 欄に `*****` と表示されます。

## (2) 縮退状態の EADs サーバを停止する

縮退状態の EADs サーバがある場合、`eztool isolate --stop` コマンドで、縮退状態の EADs サーバを停止します。

縮退状態の EADs サーバがない場合は実行する必要はありません。

## 注意事項

eztool isolate --stop コマンドは、縮退状態の EADs サーバで実行してください。

### (3) エラーメッセージを確認する

(2)で停止した EADs サーバのメッセージログに出力されているエラーメッセージを確認します。

### (4) 障害情報を取得する

障害の原因調査の際に次に示す情報が必要になります。全 EADs サーバで次に示す情報を取得してください。

- サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイル
- <運用ディレクトリ>/conf 下の全定義ファイル
- スレッドダンプ

なお、`eztool snapshot` コマンドを使用すると、ログと設定ファイルを一括して収集できます。

障害情報の取得方法については、「[12.3 障害情報の取得方法](#)」を参照してください。

#### 縮退状態になった時刻を調べる方法

縮退状態になった EADs サーバのメッセージログに出力されている KDEA04783-I または KDEA04799-E メッセージを参照してください。このメッセージが出力された時刻が縮退状態になった時刻です。

#### メッセージの出力例 (KDEA04783-I)

```
0385 2013/09/03 11:59:25.354 EADs 4A913FE2 276A38B5 KDEA04783-I Processing to isolate a
server will now start. (server ID = 1)
```

この例では、2013年09月03日の11時59分25秒に EADs サーバ ID が 1 である EADs サーバが縮退状態になっています。

#### メッセージの出力例 (KDEA04799-E)

```
0953 2013/08/21 11:55:46.116 EADs 6EFOEED6 45E67E6A KDEA04799-E Processing to isolate a
server will now start. (server ID = 3)
```

この例では、2013年08月21日の11時55分46秒に EADs サーバ ID が 3 である EADs サーバが縮退状態になっています。

## (5) 停止した EADs サーバを復旧する

障害対策後、次のどちらかのコマンドで、停止した EADs サーバを復旧します。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

復旧処理では、データの整合性を回復するために、稼働中の EADs サーバが復旧対象の EADs サーバにデータを送信します。

そのため、次の点に留意してください。

- EADs サーバが復旧するまでに、少なくともデータ取得分の時間が掛かります。
- データ送信元の EADs サーバでは、データを送信する分、オンライン性能に影響があります。

### ポイント

ディスクキャッシュ、および 2Way キャッシュを使用している場合は、次のどちらかの方法で EADs サーバを復旧してください。

復旧方法	復旧の手順	復旧時の動作	適用基準
キャッシュファイルを使用して復旧する	停止した EADs サーバのキャッシュファイルを削除しないで、 <code>ezstart -r</code> コマンド、または <code>ezserver -r</code> コマンドで停止した EADs サーバを復旧します。	キャッシュファイルからデータを読み込んだあと、稼働中の EADs サーバとの間でデータの過不足を補うことで、復旧します。	データの更新頻度や削除頻度が低いキャッシュで、キャッシュファイルに有効なデータが多く残っている場合、キャッシュファイルを使用しないときに比べて、復旧に掛かる時間を短縮できる可能性があります。
キャッシュファイルを使用しないで復旧する	停止した EADs サーバを指定して <code>deletecf -l</code> コマンドを実行し、該当するキャッシュのキャッシュデータファイルを削除します。 そのあとで、 <code>ezstart -r</code> コマンド、または <code>ezserver -r</code> コマンドを実行して、停止した EADs サーバを復旧します。	稼働中の EADs サーバからすべてのデータを取得することで、復旧します。	データの更新頻度や削除頻度が高いキャッシュで、キャッシュファイルに無効なデータが多く残っている場合、キャッシュファイルを使用するときに比べて、復旧に掛かる時間を短縮できる可能性があります。

データの復旧中にキャッシュデータファイルの容量不足が発生した場合、内部的にコンパクションが実行されます。その結果、容量不足が解消されると、復旧処理が継続されます。

なお、次のような理由で復旧に失敗したときは、問題のあるファイルが存在するキャッシュのキャッシュファイルを削除してから、`ezstart -r` コマンド、または `ezserver -r` コマンドを実行して、停止した EADs サーバを復旧してください。



- キャッシュファイルが破損している。
- キャッシュデータファイルの容量不足が発生した（内部的にコンパクションを実行したが解消できない）。
- Java ヒープあふれが発生した。
- 内部的に実行するコンパクションが失敗した。

## (6) 再起動した EADs サーバがクラスタに参加していることを確認する

eztool status コマンドで、再起動した EADs サーバがクラスタに復帰していることを確認します。

### コマンド実行例

```
$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1   XX.XXX.XXX.168  24600      1288490189 online   running none
2   XX.XXX.XXX.168  24700      429496730  online   running none
3   XX.XXX.XXX.168  24800      -429496729 online   running none
4   XX.XXX.XXX.168  24900      -1288490188 online   running none
5   XX.XXX.XXX.168  25000      -2147483648 online   running none
-----
KDEA08002-I          The command will now end.
$
```

クラスタに参加している場合、Cluster 欄に online が表示されます。

ほかにも縮退状態または停止状態の EADs サーバがあれば、「12.2.1(1) 縮退状態または停止状態の EADs サーバを確認する」から再実行してください。

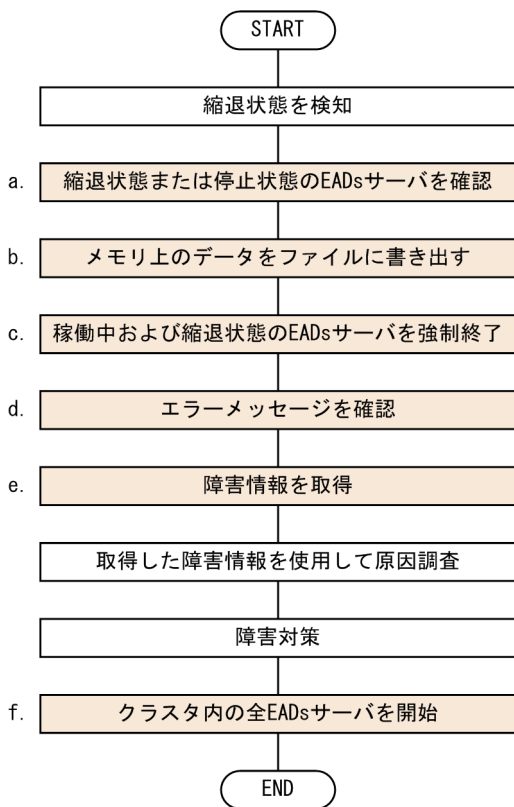
## 12.2.2 クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の場合

クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の場合の復旧までの流れについて説明します。

### (1) メモリ上のキャッシュだけを使用しているとき

メモリキャッシュを使用している場合に、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) となったときの復旧までの流れを次の図に示します。

図 12-2 クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) となったときの復旧までの流れ (メモリ上のキャッシュだけを使用している場合)



(凡例)  : システム運用管理者が行う作業

システム運用管理者が行う各作業について説明します。

### (a) 縮退状態または停止状態の EADs サーバを確認する

縮退状態または停止状態の EADs サーバを確認します。

手順については、「[12.2.1\(1\) 縮退状態または停止状態の EADs サーバを確認する](#)」を参照してください。

### (b) メモリ上のデータをファイルに書き出す

`eztool export -s` コマンドで、メモリ上のデータを個別に書き出します。このコマンドは EADs サーバごとに実行する必要があります。

```
eztool export -s
```

#### 注意事項

レンジが動作不能になる直前に受信したリクエストについては、合意できている保証がありません。そのため、書き出したデータが不整合になっているおそれがありますので、注意してください。

### (c) 稼働中および縮退状態の EADs サーバを強制終了する

`eztool forcestop` コマンドで、稼働中および縮退状態の EADs サーバを強制終了します。

すべての EADs サーバが停止状態の場合は、実行する必要はありません。

`eztool forcestop` コマンドは、稼働中および縮退状態の EADs サーバで実行してください。

```
eztool forcestop
```

### (d) エラーメッセージを確認する

(a)で確認した EADs サーバのメッセージログに出力されているエラーメッセージを確認します。

### (e) 障害情報を取得する

全 EADs サーバで障害情報を取得します。

手順については、「[12.2.1\(4\) 障害情報を取得する](#)」を参照してください。

### (f) クラスタ内の全 EADs サーバを開始する（ファイルからデータを読み込む）

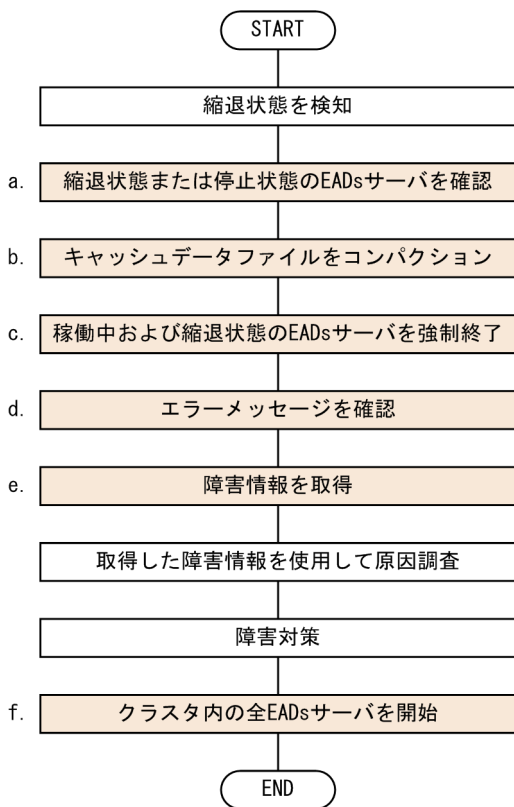
障害対策後、クラスタ内の全 EADs サーバを起動して、(a)でファイルに書き出したデータを再度メモリ上に読み込ませます。

手順については、「[10.3 EADs サーバを開始する（ファイルからデータを読み込んでキャッシュを作成する場合）](#)」を参照してください。

## (2) ディスク上のキャッシュを使用しているとき

ディスク上のキャッシュを使用している場合に、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) となったときの復旧までの流れを次の図に示します。

図 12-3 クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) となったときの復旧までの流れ (ディスク上のキャッシュを使用している場合)



(凡例)  : システム運用管理者が行う作業

システム運用管理者が行う各作業について説明します。

### (a) 縮退状態または停止状態の EADs サーバを確認する

縮退状態または停止状態の EADs サーバを確認します。

手順については、「[12.2.1\(1\) 縮退状態または停止状態の EADs サーバを確認する](#)」を参照してください。

### (b) キャッシュデータファイルをコンパクションする

`compaction` コマンドで、キャッシュデータファイルをコンパクションします。このコマンドは EADs サーバごとに実行する必要があります。

```
eztool compaction
```

### (c) 稼働中および縮退状態の EADs サーバを強制終了する

`eztool forcestop` コマンドで、稼働中および縮退状態の EADs サーバを強制終了します。

すべての EADs サーバが停止状態の場合は、実行する必要はありません。

eztool forcestop コマンドは、稼働中および縮退状態の EADs サーバで実行してください。

```
eztool forcestop
```

#### (d) エラーメッセージを確認する

(a)で確認した EADs サーバのメッセージログに出力されているエラーメッセージを確認します。

#### (e) 障害情報を取得する

全 EADs サーバで障害情報を取得します。

手順については、「[12.2.1\(4\) 障害情報を取得する](#)」を参照してください。

#### (f) クラスタ内の全 EADs サーバを開始する（ディスク上のキャッシュを再開する）

障害対策後、クラスタ内の全 EADs サーバを起動して、ディスク上のキャッシュを再開します。

手順については、「[10.3.2 EADs サーバを開始する手順（ディスク上のキャッシュを再開する）](#)」を参照してください。

### 参考

キャッシュファイルの破損が原因でキャッシュの再開に失敗する場合は、キャッシュファイルが破損している EADs サーバの数を確認してください。

- キャッシュファイルが破損している EADs サーバの数が、データの多重度より少ないとき  
ファイルが破損した EADs サーバに対して `deleteecf -l` コマンドを実行し、破損したファイルが存在するキャッシュのキャッシュファイルをすべて削除してください。そのあとで、再度 `eztool resume` コマンドを実行してください。
- キャッシュファイルが破損している EADs サーバの数が、データの多重度以上のとき  
EADs サーバを復旧できません。

## 12.2.3 レスpons低下の報告があった場合

業務 AP の管理者からレスポンス低下の報告があった場合、またはレスポンスの低下によって AP の処理がタイムアウトした場合、原因調査のために EADs サーバと EADs クライアントで障害情報を取得してください。

### (1) EADs サーバで取得する障害情報

全 EADs サーバで次に示す情報を取得してください。

- 統計情報

- スレッドダンプ
- サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイル
- <運用ディレクトリ>/conf 下の全定義ファイル

障害情報の取得方法については、「[12.3 障害情報の取得方法](#)」を参照してください。

## (2) EADs クライアントで取得する障害情報

全 EADs クライアントで次に示す情報を取得してください。

- クライアント定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイル
- クライアント定義ファイル

## (3) 業務 AP の管理者に確認すること

業務 AP の管理者に次に示すことを確認してください。

- レスポンスが低下する時間帯に規則性があるか。
- レスポンスが低下する EADs クライアントが決まっているか。
- レスポンスが低下する操作に規則性があるか。

## 12.3 障害情報の取得方法

ここでは、原因調査に必要な障害情報と、その取得方法について説明します。

### 12.3.1 原因調査に必要な障害情報

障害の原因調査には、次に示す障害情報が必要になります。障害が発生した場合、システム運用管理者はこれらの障害情報を取得してください。

なお、`eztool snapshot` コマンドを使用すると、ログと設定ファイルを一括して収集することができます。

- **メッセージログ**

メッセージログファイルには、動作確認やエラー監視用のメッセージログが出力されます。

- **統計情報**

統計情報ファイルには、チューニングの見直し、性能測定、リソース見積もりなどに使用する統計情報が出力されます。

- **スレッドダンプ**

スレッドダンプには、Java プロセス内で動作しているスレッドの情報が出力されます。必要に応じて取得してください。

- **コアダンプ**

EADs サーバがダウンした場合、運用ディレクトリ直下にコアダンプが出力されます。ファイル名は「`core.[PID].[YYMMDDhhmmss]`」※です。

なお、EADs サーバによって出力されるコアダンプは上限値が設定されていないため、ディスクの空き容量に注意してください。

注※

[PID] : EADs サーバのプロセス ID

[YYMMDDhhmmss] : YY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

- **PRF トレース**

PRF トレースには、EADs クライアントから送信されたリクエストが、EADs サーバで処理されて、処理結果が EADs クライアントに返却されるまでの通信のトレース情報が出力されます。また、EADs サーバ間の通信処理に関する情報も出力されます。必要に応じて取得してください。

PRF トレースに出力される情報については、「[13.2.2 PRF トレースに出力される情報](#)」を参照してください。

#### ■ 注意事項

PRF トレースを取得する場合は、サーバ定義の `eads.prf.enable` パラメタの指定を `true` にする必要があります。

また、次に示す障害発生時の定義ファイルも取得してください。

- <運用ディレクトリ>/conf 下の全定義ファイル

## 12.3.2 統計情報の取得方法

統計情報の取得方法については、「[11.10 統計情報を取得したい](#)」を参照してください。

## 12.3.3 スレッドダンプの取得方法

`eztool threaddump` コマンドでスレッドダンプを取得してください。

コマンド実行例

```
eztool threaddump
```

スレッドダンプはサーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下に出力されます。

サーバ定義の `eads.logger.dir` パラメタを指定していない場合は、<運用ディレクトリ>/logs 下に出力されます。

## 12.3.4 PRF トレースの取得方法

### 注意事項

PRF トレースを取得する場合は、サーバ定義の `eads.prf.enable` パラメタの指定を `true` にする必要があります。

`eztool cprfed` コマンドで PRF トレースを取得してください。この例では、コマンド実行日の 10 時 23 分 45 秒から 10 時 23 分 50 秒までの PRF トレースを編集対象とします。

コマンド実行例

```
eztool cprfed -Time "102345,102350"
```

PRF トレースファイルは<サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ>/spool/utt/prf/REQ\_PRF\_ID/dcopltrc 下に出力されます。

サーバ定義の `eads.logger.dir` パラメタを指定していない場合は、<運用ディレクトリ>/logs/spool/utt/prf/REQ\_PRF\_ID/dcopltrc 下に出力されます。



コマンドのオプションについて説明します。

- -Time "102345,102350"  
PRF トレースの編集対象範囲を時刻で指定します。

# 13

## 障害原因の調査

この章では、障害原因（原因元の切り分け）の調査方法について説明します。

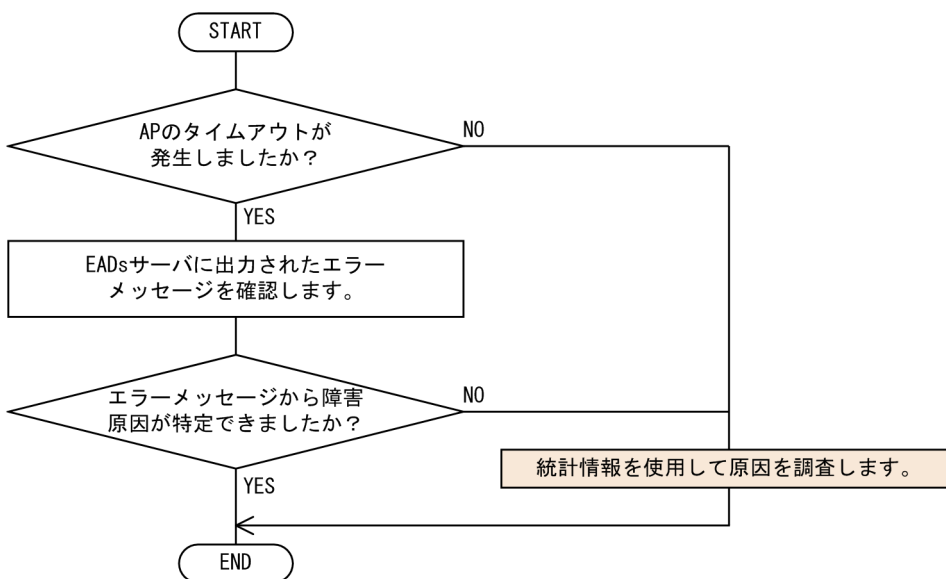
## 13.1 レスpons低下の原因調査

ここでは、レスポンス低下の原因を調査する方法について説明します。

### 13.1.1 原因調査の流れ

レスポンスの低下、およびレスポンスの低下による AP のタイムアウトの原因調査の流れを次の図に示します。

図 13-1 レスポンスの低下、およびレスポンスの低下による AP のタイムアウトの原因調査の流れ



注 統計情報を使用して原因を調査します。 の調査方法については以降で説明します。

#### 発生した現象の例

- レスポンス低下の報告があった。
- レスポンスの低下によって AP の処理がタイムアウトした。

#### 考えられる主な原因

- 特定の EADs サーバにアクセスが集中している。
- 格納データのサイズなどが適切でない。
- EADs 以外のプロセスが重い処理を実行している。
- リクエストが集中または増加している。

#### 統計情報を使用した調査の観点

- EADs サーバ、ユーザイベントハンドラ、ネットワークのどれが要因か。

内部処理時間の平均値とユーザイベントハンドラ実行時間の平均値から、レスポンス低下の要因を切り分けます。

- EADs サーバ、ユーザファンクション、ネットワークのどれが要因か。

ユーザファンクションの内部処理時間の平均値とユーザファンクションのユーザプログラムの処理時間の平均値から、レスポンス低下の要因を切り分けます。

#### 調査に必要な障害情報

- 全 EADs サーバの統計情報

統計情報の取得方法については、「[11.10 統計情報を取得したい](#)」を参照してください。

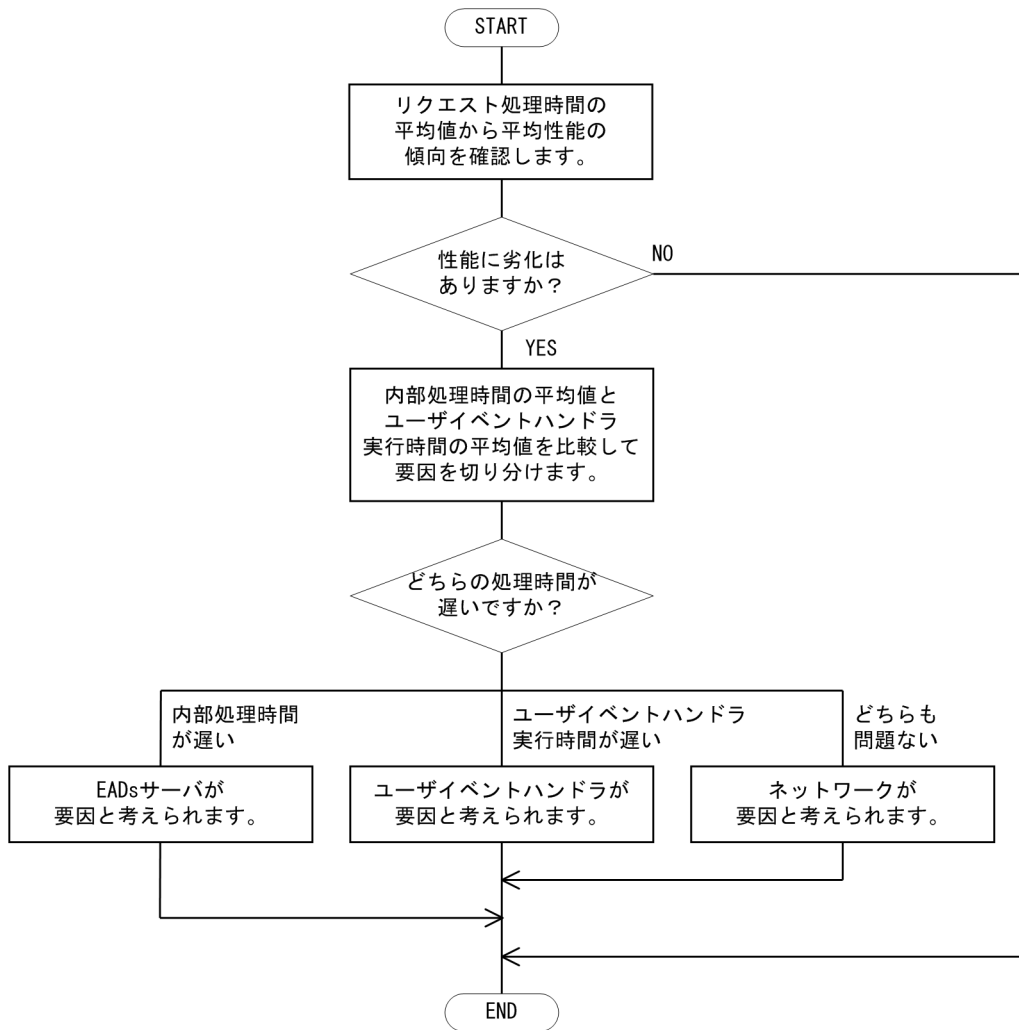
## 13.1.2 原因の調査方法

統計情報を使用した性能の確認方法について説明します。

### (1) AP の性能を確認する

統計情報を使用して AP の性能を確認する手順を次の図に示します。また、詳細の手順を以降に示します。

図 13-2 AP の性能を確認する手順



### (a) 平均性能の傾向を確認する

次の表に示す統計情報の項目から、平均性能の傾向を確認します。

統計情報の種類	項目	列名称
統計情報 (eads_stats.csv)	リクエスト処理時間の平均値	RequestProcessingAverageTime

### (b) 性能劣化の要因を切り分ける

次の表に示す統計情報の項目を比較して、性能劣化の要因を切り分けます。

統計情報の種類	項目	列名称
統計情報 (eads_stats.csv)	内部処理時間の平均値	InternalProcessingAverageTime
ユーザイベントハンドラの統計情報	ユーザイベントハンドラ実行時間の平均値	EventHandlerExecuteAverageTime

統計情報の種類	項目	列名称
(eads_event_stats.csv)		

### (c) 性能劣化の要因を分析する

それぞれの要因に応じて、性能劣化の要因を分析します。

#### ■ EADs サーバが要因の場合

EADs サーバが要因と考えられる場合は、次のことを確認してください。

- 次の表に示す統計情報の項目を確認し、EADs サーバに対するアクセスが集中していないか確認してください。

統計情報の種類	項目	列名称
統計情報 (eads_stats.csv)	リクエスト受付数 (put, create, update, replace, get, remove)	RequestCount

- Java ログから GC が発生していないか確認してください。
- OS の情報 (CPU の利用率やメモリの使用量など) を確認してください。

#### ■ ユーザイベントハンドラが要因の場合

ユーザイベントハンドラが要因と考えられる場合は、各ユーザイベントハンドラの統計情報を確認し、要因となっているユーザイベントハンドラを特定してください。

#### ■ ネットワークが要因の場合

ネットワークが要因と考えられる場合は、次のことを確認してください。

- 次の表に示す統計情報の項目を確認し、通信量を確認してください。

統計情報の種類	項目	列名称
統計情報 (eads_stats.csv)	EADs クライアント・EADs サーバ間の受信データの最大サイズ	CSReadMaxSize
	EADs クライアント・EADs サーバ間の受信データの合計サイズ	CSReadTotalSize
	EADs クライアント・EADs サーバ間の送信データの最大サイズ	CSWriteMaxSize
	EADs クライアント・EADs サーバ間の送信データの合計サイズ	CSWriteTotalSize
	EADs サーバ間の受信データの最大サイズ	SSReadMaxSize
	EADs サーバ間の受信データの合計サイズ	SSReadTotalSize

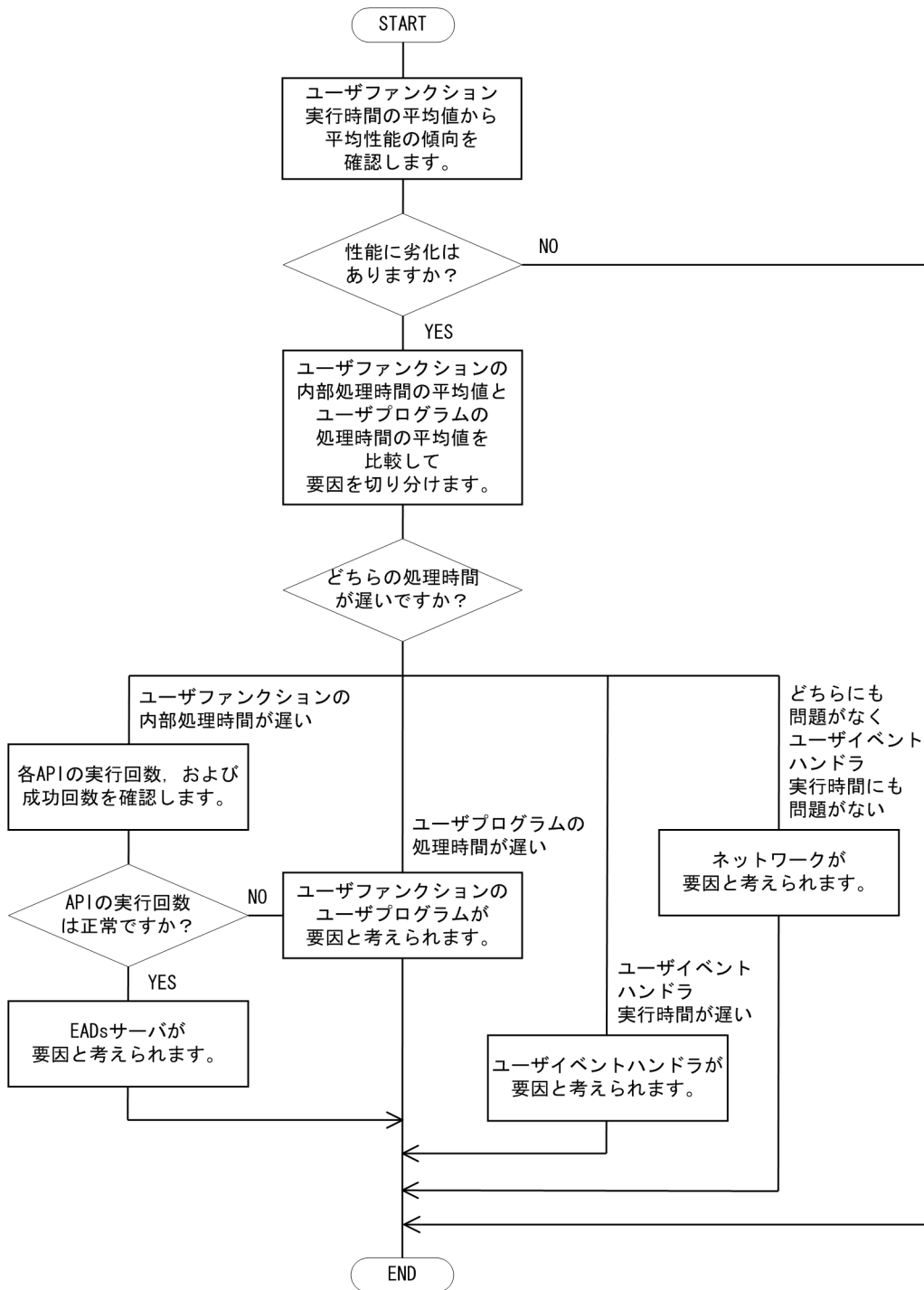
統計情報の種類	項目	列名称
	EADs サーバ間の送信データの最大サイズ	SSWriteMaxSize
	EADs サーバ間の送信データの合計サイズ	SSWriteTotalSize

- ネットワークの状態を確認してください。

## (2) ユーザファンクションの性能を確認する

統計情報を使用してユーザファンクションの性能を確認する手順を次の図に示します。また、詳細の手順を以降に示します。

図 13-3 ユーザファンクションの性能を確認する手順



### (a) 平均性能の傾向を確認する

次の表に示す統計情報の項目から、平均性能の傾向を確認します。

統計情報の種類	項目	列名称
ユーザーファンクションの統計情報 (eads_function_stats.csv)	ユーザーファンクション実行時間の平均値	FunctionExecuteAverageTime



## (b) 性能劣化の要因を切り分ける

次の表に示す統計情報の項目を比較して、性能劣化の要因を切り分けます。

統計情報の種類	項目	列名称
ユーザ機能の統計情報 (eads_function_stats.csv)	ユーザ機能の内部処理時間の平均値	FunctionInternalProcessingAverageTime
	ユーザ機能のユーザプログラムの処理時間の平均値	UserProgramProcessingAverageTime

## (c) 性能劣化の要因を分析する

それぞれの要因に応じて、性能劣化の要因を分析します。

### ■ EADs サーバが要因の場合

EADs サーバが要因と考えられる場合は、次のことを確認してください。

- 次の表に示す統計情報の項目を確認し、EADs サーバに対するアクセスが集中していないか確認してください。

統計情報の種類	項目	列名称
ユーザ機能の統計情報 (eads_function_stats.csv)	ユーザ機能の実行回数	FunctionExecuteCount

- Java ログから GC が発生していないか確認してください。
- OS の情報（CPU の利用率やメモリの使用量など）を確認してください。

### ■ ユーザ機能のユーザプログラムが要因の場合

ユーザ機能のユーザプログラムが要因と考えられる場合は、各ユーザ機能の統計情報を確認し、要因となっているユーザ機能を特定してください。

### ■ ユーザイベントハンドラが要因の場合

ユーザイベントハンドラが要因と考えられる場合は、各ユーザイベントハンドラの統計情報を確認し、要因となっているユーザイベントハンドラを特定してください。

### ■ ネットワークが要因の場合

ネットワークが要因と考えられる場合は、次のことを確認してください。

- 次の表に示す統計情報の項目を確認し、通信量を確認してください。

統計情報の種類	項目	列名称
統計情報 (eads_stats.csv)	EADs クライアント・EADs サーバ間の受信データの最大サイズ	CSReadMaxSize

統計情報の種類	項目	列名称
	EADs クライアント・EADs サーバ間の受信データの合計サイズ	CSReadTotalSize
	EADs クライアント・EADs サーバ間の送信データの最大サイズ	CSWriteMaxSize
	EADs クライアント・EADs サーバ間の送信データの合計サイズ	CSWriteTotalSize
	EADs サーバ間の受信データの最大サイズ	SSReadMaxSize
	EADs サーバ間の受信データの合計サイズ	SSReadTotalSize
	EADs サーバ間の送信データの最大サイズ	SSWriteMaxSize
	EADs サーバ間の送信データの合計サイズ	SSWriteTotalSize

- ネットワークの状態を確認してください。

## 13.2 PRF トレースの仕組みと出力される情報

ここでは、PRF トレースの仕組みと、PRF トレースに出力される情報について説明します。

### 13.2.1 PRF トレースの仕組み

EADs ではデータを多重化しているため、EADs サーバ間でデータの送受信が発生しています。通信障害などでデータを送受信できなかった場合、どのサーバのどの処理で、またいつエラーが発生したのか、原因を特定する必要があります。

EADs では、EADs サーバ間の通信処理ごとにトレース情報を PRF トレースとして取得しています。この PRF トレースの情報を追いつけると、どの処理でエラーになったのか、またどの処理で時間が掛かっているのかが特定できます。

PRF トレースのレコード (1 データ) は処理ごとに出力され、決められた ID (イベント ID) が付けられます。

PRF トレースを取得する処理とイベント ID の対応を次の図に示します。

図 13-4 PRF トレースを取得する処理とイベント ID の対応

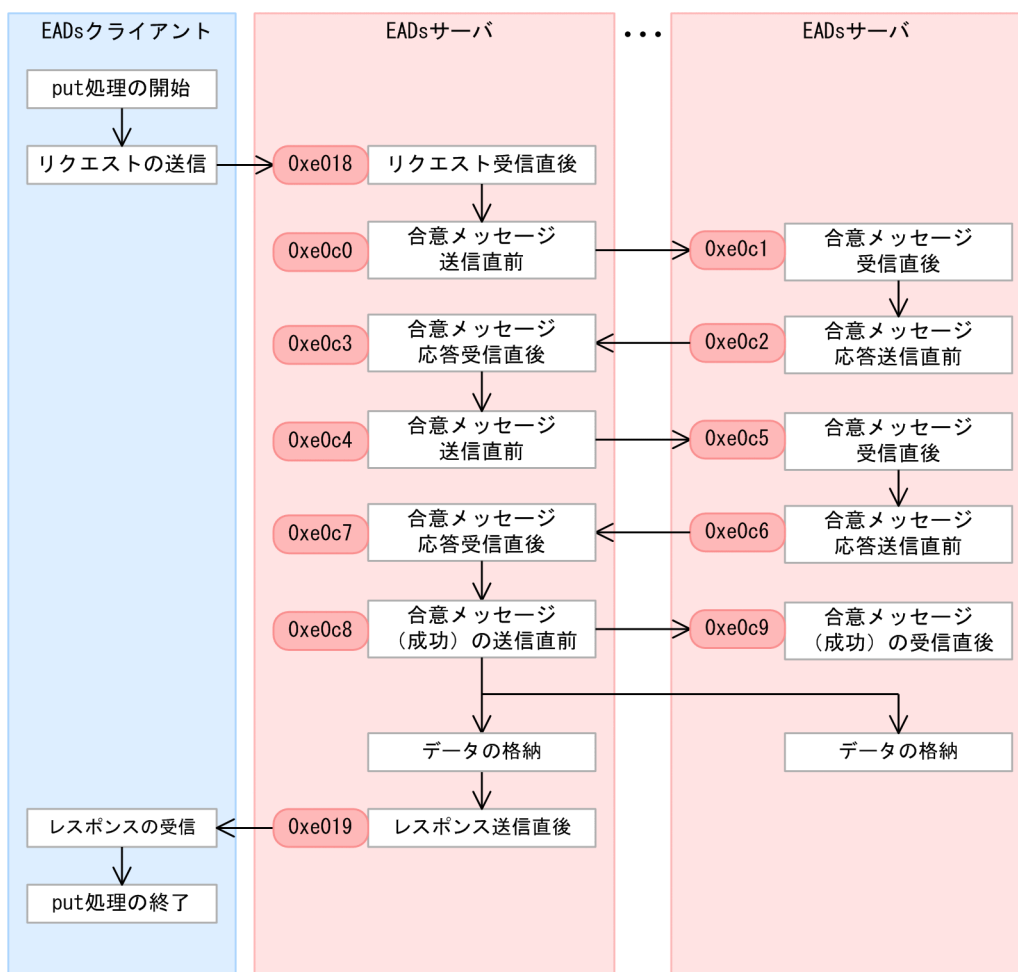


図 13-5 PRF トレースを取得する処理とイベント ID の対応 (ユーザファンクションの場合)

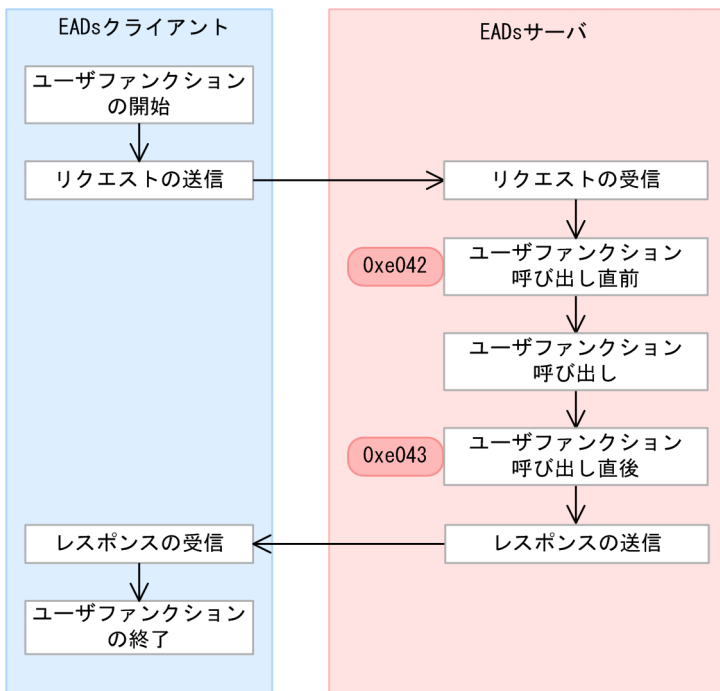
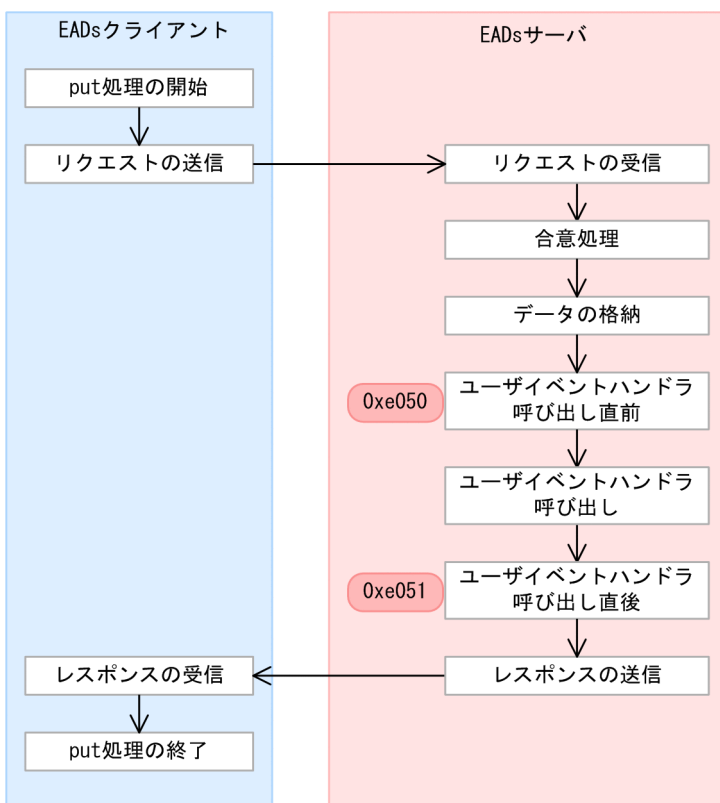


図 13-6 PRF トレースを取得する処理とイベント ID の対応 (ユーザイベントハンドラの場合)



## 13.2.2 PRF トレースに出力される情報

PRF トレースを出力した場合、次の表に示す情報が出力されます。

表 13-1 PRF トレースに出力される情報

項番	ヘッダ名 (列名)	説明
1	PRF	PRF トレースのレコードの状態が出力されます。次のどちらかが出力されます。 <ul style="list-style-type: none"> <li>• Rec：正常</li> <li>• ErrRec：異常</li> </ul>
2	Process	PRF トレースを取得したプロセスのプロセス ID が出力されます。
3	Thread(hashcode)	PRF トレースを取得したプロセスのスレッド ID が出力されます。( ) 内にはスレッドのハッシュ値が出力されます。
4	Trace	保守情報です。
5	ProcessName	保守情報です。
6	Event	イベント ID (16 進数) が出力されます。出力されるイベント ID の意味については、「表 13-2 出力されるイベント ID とその意味」を参照してください。
7	Date	PRF トレースを取得した年月日が出力されます。
8	Time	PRF トレースを取得した時刻が秒単位で出力されます。
9	Time(msec/usec/nsec)	PRF トレースを取得した時刻がミリ秒/マイクロ秒/ナノ秒単位で出力されます。
10	Rc	リターンコード (16 進数) が出力されます。次のどちらかが出力されます。 <ul style="list-style-type: none"> <li>• 0：処理成功</li> <li>• 1：処理失敗</li> </ul>
11	ClientAP IP	保守情報です。
12	ClientAP PID	保守情報です。
13	ClientAP CommNo.	保守情報です。
14	RootAP IP*	EADs クライアントまたは EADs サーバの IP アドレスが出力されます。 一連の通信処理は、RootAP IP 列、RootAP PID 列、および RootAP CommNo 列の値が同じになります。
15	RootAP PID*	EADs クライアントが付与する番号が出力されます。
16	RootAP CommNo.*	通信番号 (16 進数) が出力されます。
17	SendSCD IP	保守情報です。
18	SendSCD PID	保守情報です。
19	ReceiveSCD IP	保守情報です。
20	ReceiveSCD PID	保守情報です。
21	INT	イベント ID ごとに出力される情報が異なります。出力される情報については、「表 13-2 出力されるイベント ID とその意味」を参照してください。 なお、情報が出力されないこともあります。
22	OPR	イベント ID ごとに出力される情報が異なります。出力される情報については、「表 13-2 出力されるイベント ID とその意味」を参照してください。

項番	ヘッダ名 (列名)	説明
		なお、情報が出力されないこともあります。
23	LookupName	保守情報です。
24	OPT	保守情報です。
25	ASCII	イベント ID ごとに出力される情報が異なります。出力される情報については、「表 13-2 出力されるイベント ID とその意味」を参照してください。 なお、情報が出力されないこともあります。

#### 注※

合意メッセージは内部的に送受信される場合もあります。その場合、RootAP IP 列、RootAP PID 列、RootAp CommNo.列には、次に示すダミーの情報が格納されます。

RootAP IP : 0.0.0.0

RootAP PID : 0

RootAp CommNo. : 0x00000000xxxxxxxx (xxxxxxxx はスレッド ID を int 型に変換した値)

出力されるイベント ID とその意味を次の表に示します。

表 13-2 出力されるイベント ID とその意味

分類	イベント ID	PRF トレースの取得ポイント	INT	OPR	ASCII
EADs クライアント・EADs サーバ間の送受信	0xe018	リクエストの受信直後	保守情報	EADs クライアントの IP アドレス, ポート番号, 通番※1	—
	0xe019	レスポンスの送信直後	保守情報	—	例外情報※2
ユーザファンクション	0xe042	ユーザファンクション (Function インタフェースの execute()) 呼び出し直前	ユーザファンクション名	—	—
	0xe043	ユーザファンクション (Function インタフェースの execute()) 呼び出し直後	—	—	例外情報※2
ユーザイベントハンドラ	0xe050	ユーザイベントハンドラ (EventHandler インタフェースの execute()) 呼び出し直前	ユーザイベントハンドラ名	—	—
	0xe051	ユーザイベントハンドラ (EventHandler インタフェースの execute()) 呼び出し直後	—	—	例外情報※2
EADs サーバ間の送受信	0xe0c0	合意メッセージ (1 相目) の送信直前	保守情報	合意メッセージ送信先 EADs サーバの IP ア	保守情報

分類	イベント ID	PRF トレースの取得ポイント	INT	OPR	ASCII
				ドレス, ポート 番号*3	
	0xe0c1	合意メッセージ (1 相目) の 受信直後	保守情報	合意メッセージ 送信元 EADs サーバの IP ア ドレス, ポート 番号*3	保守情報
	0xe0c2	合意メッセージ (1 相目) の 応答送信直前	保守情報	合意メッセージ 送信先 EADs サーバの IP ア ドレス, ポート 番号*3*4	保守情報
	0xe0c3	合意メッセージ (1 相目) の 応答受信直後	保守情報	合意メッセージ 送信元 EADs サーバの IP ア ドレス, ポート 番号*3*5	保守情報または 例外情報*2
	0xe0c4	合意メッセージ (2 相目) の 送信直前	保守情報	合意メッセージ 送信先 EADs サーバの IP ア ドレス, ポート 番号*3	保守情報
	0xe0c5	合意メッセージ (2 相目) の 受信直後	保守情報	合意メッセージ 送信元 EADs サーバの IP ア ドレス, ポート 番号*3	保守情報
	0xe0c6	合意メッセージ (2 相目) の 応答送信直前	保守情報	合意メッセージ 送信先 EADs サーバの IP ア ドレス, ポート 番号*3*4	保守情報
	0xe0c7	合意メッセージ (2 相目) の 応答受信直後	保守情報	合意メッセージ 送信元 EADs サーバの IP ア ドレス, ポート 番号*3*5	例外情報*2
	0xe0c8	合意メッセージ (成功) の送 信直前	保守情報	合意メッセージ 送信先 EADs サーバの IP ア ドレス, ポート 番号*3	—

分類	イベント ID	PRF トレースの取得ポイント	INT	OPR	ASCII
	0xe0c9	合意メッセージ (成功) の受信直後	保守情報	合意メッセージ送信元 EADs サーバの IP アドレス, ポート番号 <sup>※3</sup>	—

(凡例)

— : 出力される情報はありません。

注

- 情報が取得できなかった場合、またはリクエスト中に情報が設定されていない場合、情報は出力されません。
- 上記の表で説明している以外のイベント ID (保守情報のイベント ID) も出力されます。

注※1

次に示す形式で出力されます。

IP アドレス:ポート番号/通番

注※2

例外が発生した場合に出力されます。

注※3

次に示す形式で出力されます。

IP アドレス:ポート番号

注※4

否定応答の場合は出力されません。

注※5

タイムアウトが発生した場合は出力されません。

ただし、次に示す例外が発生した場合は、合意メッセージ送信先 EADs サーバの IP アドレス、およびポート番号が出力されます。

- コネクション確立に失敗した。
- コネクションがクローズした。
- 送信キューがあふれた。



# 14

## コマンドリファレンス

この章では、EADs のコマンドの文法について説明します。

## 14.1 コマンドの格納先

---

EADs で使用するコマンドは、次のディレクトリに格納されています。

<運用ディレクトリ>/bin

### 参考

コマンドの実行を簡略化したい場合、運用ディレクトリのシンボリックリンクを作成することで、リンク側でコマンドを実行できます。

## 14.2 EADs のコマンド

EADs のコマンドの一覧を次の表に示します。

表 14-1 EADs のコマンド一覧

項番	利用シーン	コマンド名	実行対象	
1	開始	EADs サーバを開始する	ezstart	EADs サーバ
2		EADs サーバをフォアグラウンドで開始する	ezserver	EADs サーバ
3	運用操作	運用操作を行う	eztool	クラスタ または EADs サーバ

### 参考

このマニュアルでは、コマンドのオプション指定について、オプションが 1 文字の場合はハイフンを 1 つ (-), 2 文字以上の場合にはハイフンを 2 つ (--) で記載していますが、ハイフンは 1 つでも 2 つでも指定できます。例えば、-h または --help オプションは、[-h], [--h], [-help] または [--help] のどれでも指定できます。

ただし、eztool cprfed コマンドの-Time オプションは除きます。

### 14.2.1 ezstart (EADs サーバの開始)

#### (1) 機能

EADs サーバをバックグラウンドで開始します。

#### (2) 規則

- このコマンドは Developer では実行できません。
- このコマンドは EADs サーバが停止状態のときに実行できます。
- このコマンドは EADs サーバを開始するホストにログインして実行します。クラスタ内の全 EADs サーバを開始する場合は、このコマンドを EADs サーバごとに実行する必要があります。
- コマンド実行時に出力されるメッセージログは、コンソールには出力されません。<運用ディレクトリ>/logs 下の eads\_start.log に出力されます。

### (3) 形式

```
ezstart [-r]
```

### (4) オプションおよび引数

#### (a) -r または--recovery

停止状態の EADs サーバをクラスタに復旧させて、クラスタへの参加状況を online にしたい場合に指定します。

### (5) リターンコード

0：正常終了

0 以外：エラー

### (6) 注意事項

- このコマンドを実行するには、環境変数 PATH に /bin および /usr/bin が設定されている必要があります。
- このコマンドを複数同時に実行した場合、eads\_start.log は後から実行したプロセスで上書きされます。PRF トレース情報を確認したい場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の `eads_server_message[n].log`、および `spool/log/REQ_PRF_ID/ctmlog[n]` を参照してください ([n] はファイル数の通番です)。
- 一度 EADs サーバを起動したあと、別のユーザが同じ EADs サーバを起動した場合、PRF デーモンの起動に失敗することがあります。この場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリを削除したあと、再実行してください。
- PRF デーモンが起動した状態でこのコマンドを実行し、EADs サーバの起動に失敗した場合、PRF デーモンを停止しないで処理を終了することがあります。この場合は、EADs サーバの起動に失敗した要因を取り除いたあと、EADs サーバを再起動してください。
- クラスタを構成する EADs サーバ数が多い場合、EADs サーバを同時に起動すると、処理に時間が掛かり、タイムアウトして起動に失敗することがあります。このような場合は、間隔を空けながら EADs サーバを起動してください。または、サーバ定義の `eads.cluster.boot.timeout` パラメタの指定値を調整してください。

## 14.2.2 ezserver (EADs サーバのフォアグラウンドでの開始)

### (1) 機能

EADs サーバをフォアグラウンドで開始します。

## (2) 規則

- このコマンドは EADs サーバが停止状態のときに実行できます。
- このコマンドは EADs サーバを開始するホストにログインして実行します。クラスタ内の全 EADs サーバを開始する場合は、このコマンドを EADs サーバごとに実行する必要があります。
- コマンド実行時に出力されるメッセージログは、コンソール、およびサーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の `eads_server_message[n].log` (`[n]`はファイル数の通番です) に出力されます。

## (3) 形式

```
ezserver [-r]
```

## (4) オプションおよび引数

### (a) `-r` または `--recovery`

停止状態の EADs サーバをクラスタに復旧させて、クラスタへの参加状況を `online` にしたい場合に指定します。

このオプションを指定した場合、Developer では実行できません。

## (5) リターンコード

0: 正常終了

0 以外: エラー

## (6) 注意事項

- Linux 上でこのコマンドを実行するには、環境変数 `PATH` に `/bin` および `/usr/bin` が設定されている必要があります。
- 一度 EADs サーバを起動したあと、別のユーザが同じ EADs サーバを起動した場合、PRF デーモンの起動に失敗することがあります。この場合は、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリを削除したあと、再実行してください。
- PRF デーモンが起動した状態でこのコマンドを実行し、EADs サーバの起動に失敗した場合、PRF デーモンを停止しないで処理を終了することがあります。この場合は、EADs サーバの起動に失敗した要因を取り除いたあと、EADs サーバを再起動してください。
- クラスタを構成する EADs サーバ数が多い場合、EADs サーバを同時に起動すると、処理に時間が掛かり、タイムアウトして起動に失敗することがあります。このような場合は、間隔を空けながら EADs サーバを起動してください。または、サーバ定義の `eads.cluster.boot.timeout` パラメタの指定値を調整してください。

- すでに起動している EADs サーバに対してこのコマンドを実行した場合、KDEA08408-E メッセージが出力されて処理を中止します。ただし、ほぼ同時に EADs サーバを二重に起動した場合、メッセージが出力されない場合があります。この場合は間隔を開けてから、EADs サーバを起動してください。

## 14.2.3 eztool (クラスタの運用操作)

### (1) 機能

クラスタに対して運用操作を行います。

### (2) 規則

- このコマンドはクラスタ内の任意の EADs サーバで実行します。
- コマンド実行時に出力されるメッセージログは、コンソール、およびコマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の `eads_management_message[n].log` (`[n]`はファイル数の通番です) に出力されます。

### (3) 形式

```
eztool [-h]
        [-t <コマンドのタイムアウト時間>]
        <サブコマンド> <オプションおよび引数>
```

### (4) オプションおよび引数

#### (a) -h または --help

コマンドの使用方法を表示したい場合に指定します。

ただし、次のコマンドでは指定できません。

- `eztool cprfed`
- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

このオプションを指定した場合、その他のオプションを指定しても無視されます。

#### (b) -t または --timeout <コマンドのタイムアウト時間>

コマンドのタイムアウト時間 (単位: 秒) を設定したい場合に指定します。

0~2,147,483,647 の整数が指定できます。

0 を指定した場合はタイムアウトしません。

省略すると、コマンド定義の `eads.management.command.timeout` パラメタの指定値が設定されます。

次のコマンドを実行する場合は、コマンド定義の `eads.management.long_running.command.timeout` パラメタの指定値が設定されます。

- `eztool export`
- `eztool import`
- `eztool createcache`
- `eztool deletetecache`
- `eztool execfunc`
- `eztool importecf`
- `eztool resume`
- `eztool compaction`
- `eztool stop`

ただし、次のコマンドでは指定できません。

- `eztool cprfed`
- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

### (c) <サブコマンド> <オプションおよび引数>

このコマンドに指定できるサブコマンドについては、「14.3 `eztool` コマンドのサブコマンド」を参照してください。

サブコマンドに指定できるオプションおよび引数については、各サブコマンドの説明を参照してください。

#### ■ 注意事項

サブコマンドの引数にハイフン (-) で始まる文字列を指定したい場合は、ハイフン 2 つ (--) を前方に付与して指定してください。ただし、ハイフン 2 つ (--) を引数として指定することはできません。

## (5) リターンコード

各サブコマンドの説明を参照してください。

## (6) 注意事項

Linux 上でこのコマンドを実行するには、環境変数 PATH に /bin および /usr/bin が設定されている必要があります。



## 14.3 eztool コマンドのサブコマンド

eztool コマンドに指定できるサブコマンドの一覧を次の表に示します。

表 14-2 eztool コマンドに指定できるサブコマンド一覧

項番	利用シーン		サブコマンド名	実行対象
1	閉塞化と閉塞解除	クラスタを閉塞状態にする	close	クラスタ
2		クラスタの閉塞状態を解除する	open	クラスタ
3	状態確認	クラスタの状態を確認する	status	クラスタ※1
4		キャッシュの一覧を表示する	listcache	クラスタ
5		ストアデータファイルの一覧を表示する	listesd	クラスタ
6		データの格納先を表示する	getposition	クラスタ
7	排他制御	排他を解除する	unlock	クラスタ
8	メモリ・ディスク管理	キャッシュを作成する	createcache	クラスタ
9		キャッシュを削除する	deletecache	クラスタ
10	データ移行	データを書き出す	export	クラスタ※1
11		データを読み込む	import	クラスタ
12		ストアデータファイルを削除する	deleteesd	クラスタ
13	データ操作	指定したデータを格納する	put	EADs サーバ
14		指定したデータを取得する	get	EADs サーバ
15		指定したデータを削除する	remove	EADs サーバ
16	ユーザファンクション	ユーザファンクションの実行可否を表示する	listfunc	クラスタ
17		ユーザファンクションを実行する	execfunc	クラスタ※1※2
18	ユーザイベントハンドラ	ユーザイベントハンドラの実行可否を表示する	listevent	クラスタ
19	永続化	永続データに関する情報の一覧を表示する	listecf	クラスタ※1
20		キャッシュを再開する	resume	クラスタ
21		キャッシュファイルを読み込む	importecf	クラスタ
22		キャッシュファイルを削除する	deleteecf	クラスタ
23		キャッシュデータファイルをコンパクションする	compaction	EADs サーバ※3
24	トラブルシュート	PRF トレース情報を編集出力する	cprfed	EADs サーバ
25		スレッドダンプを出力する	threaddump	EADs サーバ

項番	利用シーン		サブコマンド名	実行対象
26		ログ, 設定情報, ハードウェア情報, およびネットワーク情報を収集する	snapshot	EADs サーバ
27	終了	クラスタ内の全 EADs サーバを終了する	stop	クラスタ
28		EADs サーバを強制終了する	forcestop	EADs サーバ
29		EADs サーバを縮退させる	isolate	EADs サーバ

#### 注※1

-s または--single オプションを指定すると, コマンドを実行した EADs サーバが対象となります。

#### 注※2

-g または--group オプションを指定すると, 指定したグループ名のハッシュ値から決定した EADs サーバが対象となります。

#### 注※3

--cache オプションを指定すると, コマンドを実行したキャッシュが対象となります。

--range オプションを指定すると, コマンドを実行したレンジが対象となります。

## 14.3.1 コマンド間の排他制御

コマンド間の同時実行可否を次の表に示します。

表 14-3 コマンド間の同時実行可否

コマンドの種類	同時実行可否			
	更新系	参照系	データ操作	復旧
更新系 <ul style="list-style-type: none"> <li>• eztool close</li> <li>• eztool open</li> <li>• eztool createcache</li> <li>• eztool deletetecache</li> <li>• eztool export</li> <li>• eztool import</li> <li>• eztool deleteesd</li> <li>• eztool resume</li> <li>• eztool importecf</li> <li>• eztool deleteecf</li> <li>• eztool snapshot -sd または eztool snapshot --safedump</li> <li>• eztool stop</li> <li>• eztool isolate</li> </ul>	×	○	○	×

コマンドの種類	同時実行可否			
	更新系	参照系	データ操作	復旧
参照系 • eztool status • eztool listcache • eztool listesd • eztool listecf • eztool getposition • eztool listfunc • eztool listevent	○	○	○	○
データ操作 • eztool put • eztool get • eztool remove • eztool exefunc	○	○	○	○
復旧 • ezstart -r または ezstart --recovery • ezserver -r または ezserver --recovery	×	○	○	×

(凡例)

- ：同時に実行できます。
- ×：同時に実行できません。

注

eztool unlock コマンドは、排他を解除できるときだけ実行できます。  
 次のコマンドは常に、同時に実行できます。

- eztool compaction
- eztool cprfed
- eztool threaddump
- eztool snapshot (-sd または --safedump オプション指定は除く)
- eztool forcestop

更新系のコマンド実行中は、EADs サーバから排他を取得します。これによって、ほかのホストから更新系のコマンドを同時に実行できないようにしています。

通常、コマンドの処理終了時に自動的に排他は解除されます。しかし、何らかの理由で排他の解除に失敗した場合（エラーが発生して排他が解除されないままコマンドが終了してしまった場合など）は、eztool unlock コマンドを実行して、排他を解除する必要があります。

## 14.3.2 close (クラスタの閉塞化)

### (1) 機能

クラスタを閉塞状態にします。

### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 稼働状態 (running)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool close
```

### (4) リターンコード

0 : 成功

101 : 失敗

## 14.3.3 open (クラスタの閉塞解除)

### (1) 機能

クラスタの閉塞状態を解除します。

### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)

- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool open
```

### (4) リターンコード

0 : 成功

101 : 失敗

## 14.3.4 status (クラスタ状態の確認)

### (1) 機能

クラスタの状態を確認します。

### (2) 規則

- このサブコマンドはクラスタの状態に関係なく実行できます。
- このサブコマンドは、EADs サーバが次の状態のときに実行できます。
  - 初期化中 (initializing)
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
  - 停止中 (stopping)

### (3) 形式

```
eztool status [-v]
               [-s|-c <表示内容の項目名>==<状態>]
```

### (4) オプションおよび引数

#### (a) -v または--verbose

コマンド実行結果の詳細を表示したい場合に指定します。

#### (b) -s または--single

クラスタ全体ではなく、コマンドを実行する EADs サーバの状態だけを確認したい場合に指定します。

#### (c) -c または--count <表示内容の項目名>==<状態>

クラスタに参加する各 EADs サーバの状態がそろっていることを確認したい場合に指定します。

指定した項目が、指定した状態と完全一致する EADs サーバ数を、リターンコードとして返します。ただし、情報が取得できないためにアスタリスク (\*) が表示された項目については、カウントされません。

#### ■表示内容の項目名

指定できる表示内容の項目名を次の表に示します。

表示内容の項目名	説明	表示の有無	
		-v または--verbose オプション指定	
		なし	あり
ID	EADs サーバ ID	○	○
IP_Address	EADs サーバの IP アドレス	○	○
ServerName	EADs サーバ名 (運用ディレクトリ名)	×	○
ClientPort	EADs クライアントとの通信に使用する EADs サーバのポート番号	○	○
ServerPort	EADs サーバ間のデータの多重化に使用するポート番号	×	○
ManagePort	コマンドが使用する通信ポートを確認するためのポート番号	×	○
Position	EADs サーバの位置	○	○
Cluster	クラスタへの参加状況	○	○
State	EADs サーバの状態	○	○
Operation	現在実行中のサブコマンド名, またはコマンドのオプション名	○	○

表示内容の項目名	説明	表示の有無	
		-v または --verbose オプション指定	
		なし	あり
Lock	EADs サーバの排他状態	×	○
KeyCount	EADs サーバ内の key の総数	×	○
UsedCache	EADs サーバ内の、メモリキャッシュまたは 2Way キャッシュの value 部分が格納される領域のメモリ使用量 (Explicit ヒープ) の割合 (メモリ使用量/メモリ総容量) (単位: メガバイト)	×	○
Version	バージョン情報	×	○

(凡例)

- : 表示されます。
- ×: 表示されません。

### ■状態

指定した表示内容の項目名で表示される文字列を指定します。

ASCII コード「0x20」～「0x7E」の文字が指定できます。

半角スペースは二重引用符 (") で囲むことで使用できますが、表示内容の項目名、および状態の前後の半角スペースは無視されます。

なお、情報が取得できない項目にはアスタリスク (\*) が表示されるため、アスタリスク (\*) だけを指定した場合はエラーとなります。

## (5) 出力例

eztool status コマンドの実行結果の出力例を次に示します。

### ■-v または --verbose オプションを指定しない場合

```

$ eztool status
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ClientPort  Position    Cluster  State  Operation
1   XX.XXX.XXX.168  24600      1288490189 online  running none
2   XX.XXX.XXX.168  24700      429496730  online  running none
3   XX.XXX.XXX.168  24800      -429496729 online  running none
4   XX.XXX.XXX.168  24900      -1288490188 online  running none
5   XX.XXX.XXX.168  25000      -2147483648 online  running none
-----

KDEA08002-I          The command will now end.
$

```

## ■-v または--verbose オプションを指定する場合

```

$ eztool status -v
KDEA08001-I          The command will now start. (subcommand = status, parameter = [status, -v])

Cluster Health: AVAILABLE
TotalCount: 5
OnlineCount: 5
OfflineCount: 0
StandbyCount: 0

ID  IP_Address      ServerName  ClientPort  ServerPort  ManagePort  Position    Cluster  State  Operation  Lock  KeyCount  UsedCache  Version
1   XX.XXX.XXX.168  server01   24600      24633      24620      1288490189 online  running  none     unlock  0  0% ( 0 / 92 MB ) 03-60-00
2   XX.XXX.XXX.168  server02   24700      24733      24720      429496730  online  running  none     unlock  0  0% ( 0 / 92 MB ) 03-60-00
3   XX.XXX.XXX.168  server03   24800      24833      24820      -429496729 online  running  none     unlock  0  0% ( 0 / 92 MB ) 03-60-00
4   XX.XXX.XXX.168  server04   24900      24933      24920      -1288490188 online  running  none     unlock  0  0% ( 0 / 92 MB ) 03-60-00
5   XX.XXX.XXX.168  server05   25000      25033      25020      -2147483648 online  running  none     unlock  0  0% ( 0 / 92 MB ) 03-60-00
-----

KDEA08002-I          The command will now end.
$

```

表示内容を次の表に示します。

項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
1	Cluster Health	クラスタの状態 次のどれかが表示されます。 <ul style="list-style-type: none"> <li>• AVAILABLE クラスタが正常に稼働している状態です。</li> <li>• PARTIALLY_AVAILABLE クラスタの一部が稼働している状態です。 keyによっては、アクセスできない場合があります。</li> <li>• NOT_AVAILABLE クラスタが稼働していない状態です。</li> </ul>	○	○
2	TotalCount	EADs サーバ数 -s または--single オプションを指定した場合は表示されません。	○	○



項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
3	OnlineCount	クラスタに参加している EADs サーバ数 -s または--single オプションを指定した場合は表示されません。 また、クラスタの状態が NOT_AVAILABLE の場合、カウントされません。	○	○
4	OfflineCount	クラスタに参加していない EADs サーバ数 -s または--single オプションを指定した場合は表示されません。 また、クラスタの状態が NOT_AVAILABLE の場合、カウントされません。	○	○
5	StandbyCount	クラスタに参加する予定の EADs サーバ数 -s または--single オプションを指定した場合は表示されません。 また、クラスタの状態が NOT_AVAILABLE の場合、カウントされません。	○	○
6	ID	EADs サーバ ID	○	○
7	IP_Address	EADs サーバの IP アドレス	○	○
8	ServerName	EADs サーバ名 (運用ディレクトリ名)	×	○※1※2
9	ClientPort	EADs クライアントとの通信に使用する EADs サーバのポート番号	○	○
10	ServerPort	EADs サーバ間のデータの多重化に使用するポート番号	×	○※2
11	ManagePort	コマンドが使用する通信ポートを確認するためのポート番号	×	○※2
12	Position	EADs サーバの位置 位置 (ハッシュ値) は降順で表示されます。	○	○
13	Cluster	クラスタへの参加状況 次のどれかが表示されます。 <ul style="list-style-type: none"> <li>• online クラスタに参加している</li> <li>• offline クラスタに参加していない</li> <li>• standby クラスタに参加する予定</li> </ul>	○	○※3
14	State	EADs サーバの状態 次のどれかが表示されます。 <ul style="list-style-type: none"> <li>• initializing</li> </ul>	○	○※1※2

項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
		初期化中 <ul style="list-style-type: none"> <li>• initialized 初期化状態</li> <li>• running 稼働状態</li> <li>• closing 閉塞中</li> <li>• closed 閉塞状態</li> <li>• isolated 縮退状態</li> <li>• stopping 停止中</li> </ul>		
15	Operation	現在実行中のサブコマンド名, またはコマンドのオプション名 次のどれかが表示されます。 <ul style="list-style-type: none"> <li>• close</li> <li>• open</li> <li>• unlock</li> <li>• createcache</li> <li>• deletetecache</li> <li>• export</li> <li>• import</li> <li>• deleteesd</li> <li>• resume</li> <li>• importecf</li> <li>• deleteecf</li> <li>• compaction</li> <li>• stop</li> <li>• isolate</li> <li>• recovery</li> </ul> export, import, importecf および stop については, 処理の進捗率も表示されます。 (例) export(88%) 実行中でない場合は none が表示されます。	○	○※1※2
16	Lock	EADs サーバの排他状態 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• lock 排他している</li> </ul>	×	○※3

項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
		<ul style="list-style-type: none"> <li>unlock 排除していない</li> </ul>		
17	KeyCount	EADs サーバ内の key の総数 多重化によってコピーされる key 数も含まれます。	×	○※1※2
18	UsedCache	EADs サーバ内の、メモリキャッシュまたは 2Way キャッシュの value 部分が格納される領域 のメモリ使用量 (Explicit ヒープ) の割合 (メモ リ使用量/メモリ総容量) (単位: メガバイト) <ul style="list-style-type: none"> <li>メガバイト以下の値は切り捨てて表示されます。</li> <li>メモリ使用量には、多重化によってコピーされ る value も含まれます。</li> <li>ここで表示されるメモリ使用量およびメモリ総 容量については、key と value のうち、value 部分が格納される領域の容量だけが対象となり ます。管理領域として使用される分 (Explicit ヒープサイズの 3% (メガバイト単位で小数点 以下は切り上げ)), および更新操作の履歴が格 納される領域の分は含まれません。</li> </ul>	×	○※1※2
19	Version	バージョン情報 03-00 の場合、03-00-00 と表示されます。	×	○※1※2

(凡例)

- : 表示されます。
- ×: 表示されません。

注※1

接続失敗, 通信エラー, または通信タイムアウトの場合, アスタリスク (\*) が表示されます。

注※2

EADs サーバのクラスタへの参加状況が standby の場合, アスタリスク (\*) が表示されます。

注※3

クラスタの状態が NOT\_AVAILABLE の場合, アスタリスク (\*) が表示されます。

## (6) リターンコード

0: 成功

0 以外: エラー

-s または --single オプションを指定した場合

EADs サーバの状態をリターンコードとして返します。

- 0：稼働状態 (running)
- 1：初期化中 (initializing)
- 2：初期化状態 (initialized)
- 3：閉塞中 (closing)
- 4：閉塞状態 (closed)
- 5：停止中 (stopping)
- 10：縮退状態 (isolated)
- 101：失敗
- 110：接続失敗
- 111：通信タイムアウト

-c または --count オプションを指定した場合

指定した項目が、指定した状態になっている EADs サーバ数を、リターンコードとして返します。

コマンドの実行に失敗した場合は 101 を返します。

## (7) 注意事項

- EADs サーバ ID がいちばん小さい EADs サーバが起動していない場合、ほかの EADs サーバはクラスタに参加できません。また、クラスタに参加していない EADs サーバでこのコマンドを実行した場合、ほかの EADs サーバが複数起動していても、コマンドを実行した EADs サーバの情報しか表示されません。
- 復旧中の EADs サーバでこのコマンドを実行した場合、復旧が完了するまではほかの EADs サーバの情報は表示されません。
- 縮退状態の EADs サーバしか存在しない場合、クラスタ構成情報が更新されないために古い情報が表示されることがあります。
- 出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。

## 14.3.5 listcache (キャッシュの一覧表示)

### (1) 機能

キャッシュの一覧を表示します。

## (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)

## (3) 形式

```
eztool listcache [-v]
```

## (4) オプションおよび引数

### (a) -v または--verbose

コマンド実行結果の詳細を表示したい場合に指定します。

## (5) 出力例

eztool listcache コマンドの実行結果の出力例を次に示します。

### ■-v または--verbose オプションを指定しない場合

```
$ eztool listcache
KDEA08001-I          The command will now start. (subcommand = listcache, parameter = [listcache])

CacheCount: 3 / 64

CacheName  CacheType  TotalMasterKeyCount
cache1     Memory     0
cache2     Memory     0
cache3     Memory     0
-----
KDEA08002-I          The command will now end.
$
```

### ■-v または--verbose オプションを指定する場合

```

$ eztool listcache -v
KDEA08001-I          The command will now start. (subcommand = listcache, parameter = [listcache, -v])

CacheCount: 3 / 64

CacheName CacheType TotalMasterKeyCount Server MasterKeyCount
cache1    Memory      0 XX.XXX.XXX.168:24600 0
           XX.XXX.XXX.168:24700 0
           XX.XXX.XXX.168:24800 0
           XX.XXX.XXX.168:24900 0
           XX.XXX.XXX.168:25000 0
cache2    Memory      0 XX.XXX.XXX.168:24600 0
           XX.XXX.XXX.168:24700 0
           XX.XXX.XXX.168:24800 0
           XX.XXX.XXX.168:24900 0
           XX.XXX.XXX.168:25000 0
cache3    Memory      0 XX.XXX.XXX.168:24600 0
           XX.XXX.XXX.168:24700 0
           XX.XXX.XXX.168:24800 0
           XX.XXX.XXX.168:24900 0
           XX.XXX.XXX.168:25000 0
-----
KDEA08002-I          The command will now end.
$

```

表示内容を次の表に示します。

項番	表示内容	説明	表示の有無	
			-v または --verbose オプション指定	
			なし	あり
1	CacheCount	キャッシュ数の現在値と最大値	○	○
2	CacheName	キャッシュ名	○	○
3	CacheType	キャッシュタイプ 次のどれかが表示されます。 <ul style="list-style-type: none"> <li>• Memory メモリキャッシュ</li> <li>• Disk ディスクキャッシュ</li> <li>• 2Way 2Way キャッシュ</li> </ul>	○	○
4	TotalMasterKey Count	各キャッシュの key の総数 多重化によってコピーされた key 数は含みません。	○	○
5	Server	EADs サーバの IP アドレス、および EADs クライアントとの通信に使用する EADs サーバのポート番号 次の形式で表示されます。 IP アドレス:ポート番号 縮退状態 (isolated)、または起動していない EADs サーバの場合は、何も表示されません。	×	○

項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
6	MasterKeyCount	EADs サーバごとの、各キャッシュの key の数 多重化によってコピーされた key 数は含みません。 縮退状態 (isolated)、または起動していない EADs サーバの場合は、何も表示されません。	×	○

(凡例)

- ：表示されます。
- ×

注

実行結果は、次の優先順位に従って表示されます。

1. CacheName の値の自然順序順に表示されます。
2. -v または--verbose オプションを指定する場合、CacheName の値が同じ行は、Server の値の自然順序順に表示されます。

## (6) リターンコード

0：成功

101：失敗

## (7) 注意事項

出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。

### 14.3.6 listesd (ストアデータファイルの一覧表示)

#### (1) 機能

クラスタ内のストアデータファイルの一覧を表示します。

#### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。

- クラスタ稼働中 (AVAILABLE)
- クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool listesd [-v]
               [-d <ストアデータファイルの格納先パス名>]
```

### (4) オプションおよび引数

#### (a) `-v` または `--verbose`

コマンド実行結果の詳細を表示したい場合に指定します。

#### (b) `-d` または `--directory` <ストアデータファイルの格納先パス名>

ストアデータファイルの格納先パス名を指定します。

指定したディレクトリ下にあるストアデータファイルの一覧を表示したい場合に指定します。

パス名には、アスタリスク (\*), 二重引用符 ("), 疑問符 (?), 縦線 (|), 小なり演算子 (<), 大なり演算子 (>) を含むディレクトリは指定できません。

ストアデータファイルの格納先パスに相対パスを指定した場合、運用ディレクトリが起点となります。

### (5) 出力例

`eztool listesd` コマンドの実行結果の出力例を次に示します。

#### ■ `-v` または `--verbose` オプションを指定しない場合



```

$ eztool listesd
KDEA08001-I          The command will now start. (subcommand = listesd, parameter = [listesd])

TotalCount: 4
export: 1 / 3
stop  : 1 / 2
other  : 2
latest: stop_20130402183258

Type           StoreDataFileKey
export         20130402183257
stop          stop_20130402183258
other         single_20130402183258
              test
-----
KDEA08002-I          The command will now end.
$

```

## ■-v または--verbose オプションを指定する場合

```

$ eztool listesd -v
KDEA08001-I          The command will now start. (subcommand = listesd, parameter = [listesd, -v])

TotalCount: 4
export: 1 / 3
stop  : 1 / 3
other  : 2
latest: stop_20130402183258

Type  StoreDataFileKey  FileName                                     FileSize(MB)  Server                                     AbsolutePath
-----
export 20130402183257    eads_20130402183257_1.esd                   0 XX.XXX.XXX.168:24600 /opt/hi tachi/xeads/server/servers/test1/store/eads_20130402183257_1.esd
      eads_20130402183257_2.esd                   0 XX.XXX.XXX.169:24700 /opt/hi tachi/xeads/server/servers/test2/store/eads_20130402183257_2.esd
      eads_20130402183257_3.esd                   0 XX.XXX.XXX.170:24800 /opt/hi tachi/xeads/server/servers/test3/store/eads_20130402183257_3.esd
stop   stop_20130402183258 eads_stop_20130402183258_1.esd                0 XX.XXX.XXX.168:24600 /opt/hi tachi/xeads/server/servers/test1/store/eads_stop_20130402183258_1.esd
      eads_stop_20130402183258_2.esd                0 XX.XXX.XXX.169:24700 /opt/hi tachi/xeads/server/servers/test2/store/eads_stop_20130402183258_2.esd
      eads_stop_20130402183258_3.esd                0 XX.XXX.XXX.170:24800 /opt/hi tachi/xeads/server/servers/test3/store/eads_stop_20130402183258_3.esd
other  single_20130402183258 eads_single_20130402183258_1.esd              0 XX.XXX.XXX.168:24600 /opt/hi tachi/xeads/server/servers/test1/store/eads_single_20130402183258_1.esd
      test                                           eads_test_1.esd                               0 XX.XXX.XXX.168:24600 /opt/hi tachi/xeads/server/servers/test1/store/eads_test_1.esd
      eads_test_2.esd                               0 XX.XXX.XXX.169:24700 /opt/hi tachi/xeads/server/servers/test2/store/eads_test_2.esd
      eads_test_3.esd                               0 XX.XXX.XXX.170:24800 /opt/hi tachi/xeads/server/servers/test3/store/eads_test_3.esd
-----
KDEA08002-I          The command will now end.
$

```

表示内容を次の表に示します。

項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
1	TotalCount	全ストアデータファイルキー数	○	○
2	export	eztool export コマンド (引数指定なし) 実行時に出力されるストアデータファイルのストアデータファイルキー数と世代数の上限値 次の形式で表示されます。 ストアデータファイルキー数 / 世代数の上限値 世代数の上限値に 0 を指定している場合は表示されません。	○	○
3	stop	eztool stop コマンド実行時に出力されるストアデータファイルのストアデータファイルキー数と世代数の上限値 次の形式で表示されます。	○	○

項番	表示内容	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
		ストアデータファイルキー数 / 世代数の上限値 世代数の上限値に 0 を指定している場合は表示されません。		
4	other	eztool export コマンド (引数や-s または--single オプション指定あり) 実行時に出力されるストアデータファイルのストアデータファイルキー数	○	○
5	latest	eztool import コマンド (引数指定なし) 実行時に読み込み対象となるストアデータファイルのストアデータファイルキー 存在しない場合は none が表示されます。	○	○
6	Type	ストアデータファイルの種別 次のどれかが表示されます。 <ul style="list-style-type: none"> <li>• export eztool export コマンド (引数指定なし) 実行時に出力されるストアデータファイル</li> <li>• stop eztool stop コマンド実行時に出力されるストアデータファイル</li> <li>• other eztool export コマンド (引数や-s または--single オプション指定あり) 実行時に出力されるストアデータファイル</li> </ul>	○	○
7	StoreDataFileKey	ストアデータファイルキー	○	○
8	FileName	ストアデータファイル名	×	○
9	FileSize(MB)	ストアデータファイルサイズ (単位: メガバイト) メガバイト以下の値は切り捨てて表示されます。	×	○
10	Server	ストアデータファイルがある EADs サーバの IP アドレス, および EADs クライアントとの通信に使用する EADs サーバのポート番号 次の形式で表示されます。 IP アドレス:ポート番号	×	○
11	AbsolutePath	ストアデータファイル名 (絶対パス)	×	○

(凡例)

- : 表示されます。
- ×: 表示されません。

## 注

実行結果は、次の優先順位に従って表示されます。

1. Type の値が export, stop, other の順に表示されます。
2. Type の値が同じ行は、StoreDataFileKey の値の自然順序順に表示されます。
3. -v または--verbose オプションを指定する場合、Type の値および StoreDataFileKey の値が同じ行は、Server の値の自然順序順に表示されます。

## (6) リターンコード

0：成功

101：失敗

## (7) 注意事項

出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。

## 14.3.7 getposition (データの格納先の表示)

### (1) 機能

指定した key またはグループの、格納先 EADs サーバを表示します。

### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスター一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、EADs サーバが次の状態のときに実行できます。
  - 初期化中 (initializing)
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
  - 停止中 (stopping)

- `-l` オプションまたは `--local` オプションを指定した場合は、クラスタおよび EADs サーバの状態に関係なく実行できます。

### (3) 形式

```
eztool getposition <key> [-l]
```

### (4) オプションおよび引数

#### (a) <key>

格納先 EADs サーバを表示したいデータに関連づけられた key を指定します。

key には、半角 1,024 文字までの、ASCII コード「0x20」～「0x7E」の文字が指定できます。

ただし、次の場合はエラーとなります。

- 区切り文字 (:) が key の先頭または後尾にある場合  
(例) 「:element」 「group:」
- 区切り文字 (:) が連続している場合  
(例) 「group::element」

#### (b) `-l` または `--local`

クラスタ定義で設定したクラスタ構成で、指定した key またはグループが格納される EADs サーバを知りたい場合に指定します。

このオプションを指定すると、コマンドを実行する EADs サーバのクラスタ定義ファイルの内容を読み込みます。

次の場合は、エラーとなります。

- コマンドを実行する EADs サーバにクラスタ定義ファイルが存在しない場合
- クラスタ定義の `eads.node.< EADs サーバ ID >.address` パラメタおよび `eads.node.< EADs サーバ ID >.port` パラメタが定義されていない場合

#### 注意事項

稼働中のクラスタ構成情報で、指定した key またはグループが格納される EADs サーバを知りたい場合は、このオプションを指定しないで実行してください。

なお、このオプションを指定しない場合、ダウンしている EADs サーバの情報は表示されません。表示される EADs サーバ数と ReplicationCount (データの多重度) を比較することで、データのコピー状況を確認します。

## (5) 出力例

eztool getposition コマンドの実行結果の出力例を次に示します。

```
$ eztool getposition group:element
KDEA08001-I          The command will now start. (subcommand = getposition, parameter = [getposition, group:element])

ReplicationCount: 3 / 3
HashValue: -1039521297

No.  IP_Address  ClientPort  Position
 1  XX.XXX.XXX.171  24900      -1288490188
 2  XX.XXX.XXX.172  25000      -2147483648
 3  XX.XXX.XXX.168  24600      1288490189

KDEA08002-I          The command will now end.
$
```

表示内容を次の表に示します。

項番	表示内容	説明
1	ReplicationCount	データの多重度
2	HashValue	指定された key またはグループ名のハッシュ値
3	No.	データの優先度* 昇順で表示されます。
4	IP_Address	EADs サーバの IP アドレス
5	ClientPort	EADs クライアントとの通信に使用する EADs サーバのポート番号
6	Position	EADs サーバの位置 (ハッシュ値)

### 注※

データを多重化している場合、データの格納先 EADs サーバ、データのコピー先 EADs サーバの順に表示されます。

この例の場合、No.1 に表示されている EADs サーバがデータの格納先、No.2 以降に表示されている EADs サーバが、データのコピー先となります。

## (6) リターンコード

0 : 成功

101 : 失敗

## (7) 注意事項

- 出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。
- すべての EADs サーバが縮退状態 (isolated) の場合はエラーとなります。

## 14.3.8 unlock (排他解除)

### (1) 機能

コマンドの排他を解除します。

詳細については、「[14.3.1 コマンド間の排他制御](#)」を参照してください。

### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、次の場合は実行できません。
  - 実行中の操作がある場合
  - クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合クラスタへの参加状況については、[eztool status](#) コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool unlock
```

### (4) リターンコード

0：成功

101：失敗

### (5) 注意事項

すべての EADs サーバが縮退状態 (isolated) の場合はエラーとなります。

## 14.3.9 createcache (キャッシュの作成)

### (1) 機能

キャッシュを作成します。

キャッシュはメモリキャッシュ、ディスクキャッシュ、および2Wayキャッシュの合計で、クラスタ内に最大64個作成できます。

### (2) 規則

- このサブコマンドは、クラスタの状態がクラスタ稼働中 (AVAILABLE) のときに実行できます。
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool createcache <キャッシュ名>
```

### (4) オプションおよび引数

#### (a) <キャッシュ名>

新しく作成するキャッシュの名称を指定します。

キャッシュ名に指定できる文字を次に示します。

- キャッシュ定義ファイルを使用しない場合  
半角32文字までの、ASCIIコード「0x20」～「0x7E」の文字が指定できます。
- キャッシュ定義ファイルを使用する場合  
半角32文字までの、半角英数字 (0~9, A~Z, および a~z) が指定できます。

### (5) リターンコード

0: 成功

101: 失敗

## (6) 注意事項

- キャッシュ定義ファイルを作成していない場合は、メモリキャッシュが作成されます。
- キャッシュタイプがメモリキャッシュまたは 2Way キャッシュの場合、起動設定の `eads.java.external.heapsize` パラメタの value 部分が格納される領域のサイズが 0 のときはエラーになります。
- キャッシュを作成すると、キャッシュ 1 個につき、(データの多重度×2-1) × 2 個のスレッドが作成されます。スレッド数が多くなると、メモリ使用量が増加するため、注意してください。
- キャッシュの作成に失敗した場合、次のどれかの手順で再度キャッシュを作成してください。
  - `eztool listcache` コマンドでキャッシュの有無を確認してください。キャッシュが存在しているときは、`eztool deletecache` コマンドでキャッシュを削除して、再度キャッシュを作成してください。
  - キャッシュタイプがディスクキャッシュまたは 2Way キャッシュのときは、`eztool listecf` コマンドでキャッシュファイルの有無を確認してください。キャッシュファイルが存在しているときは、`eztool deleteecf` コマンドでキャッシュファイルを削除して、再度キャッシュを作成してください。
  - キャッシュ定義ファイルに指定したパラメタの指定値を確認してください。誤っているときは、正しい値に変更して、再度キャッシュを作成してください。
  - 起動設定の `eads.java.external.heapsize` パラメタの指定値を確認してください。誤っているときは、正しい値に変更して、再度キャッシュを作成してください。

### 14.3.10 deletecache (キャッシュの削除)

#### (1) 機能

キャッシュを削除します。

このサブコマンドを実行すると、キャッシュと、そのキャッシュが持つデータが削除されます。

#### (2) 規則

- このサブコマンドは、クラスタの状態がクラスタ稼働中 (AVAILABLE) のときに実行できます。
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 閉塞状態 (closed)
- 稼働中の FullGC の発生を抑えるため、このサブコマンドが終了するタイミングで、各 EADs サーバが FullGC を実施します。



- このサブコマンドが EADs サーバから排他を取得している間、EADs サーバは縮退状態 (isolated) になりません。なお、EADs サーバから排他を取得している間にプロセスがダウンしたり、EADs サーバが停止したりした場合は、排他を解除したあとに縮退が実行されます。

### (3) 形式

```
eztool deletocache <キャッシュ名>  
                    [--with_deleteecf]
```

### (4) オプションおよび引数

#### (a) <キャッシュ名>

削除するキャッシュの名称を指定します。

キャッシュ名に指定できる文字を次に示します。

- キャッシュ作成時にキャッシュ定義ファイルを使用しなかった場合  
半角 32 文字までの、ASCII コード「0x20」～「0x7E」の文字が指定できます。
- キャッシュ作成時にキャッシュ定義ファイルを使用した場合  
半角 32 文字までの、半角英数字 (0~9, A~Z, および a~z) が指定できます。

#### (b) --with\_deleteecf

ディスクキャッシュまたは 2Way キャッシュを削除する場合で、キャッシュファイルもあわせて削除したいときに指定します。

### (5) リターンコード

0：成功

101：失敗

## 14.3.11 export (データの書き出し)

### (1) 機能

メモリキャッシュのデータをストアデータファイルに書き出します。

メモリキャッシュが存在しない場合は、実行されません。

## (2) 規則

- このサブコマンドは、クラスタの状態がクラスタ稼働中 (AVAILABLE) のときに実行できます。
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが閉塞状態 (closed) のときに実行できます。
- `-s` または `--single` オプションを指定した場合は、クラスタの状態に関係なく、EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
- 稼働中の FullGC の発生を抑えるため、このサブコマンドが終了するタイミングで、各 EADs サーバが FullGC を実施します。ただし、`-s` または `--single` オプションを指定した場合は、FullGC を実施しません。
- このサブコマンドが EADs サーバから排他を取得している間、EADs サーバは縮退状態 (isolated) になりません。なお、EADs サーバから排他を取得している間にプロセスがダウンしたり、EADs サーバが停止したりした場合は、排他を解除したあとに縮退が実行されます。

## (3) 形式

```
eztool export [-s]
               [-d <ストアデータファイルの出力先パス名>]
               [<ストアデータファイルキー>]
```

## (4) オプションおよび引数

### (a) `-s` または `--single`

クラスタ全体で保持するデータではなく、コマンドを実行する EADs サーバが保持するデータだけを書き出したい場合に指定します。

例えば、縮退状態になった EADs サーバのデータをバックアップしておきたい場合などに指定します。

このオプションを指定した場合、EADs サーバから排他を取得しません。データの書き出し中にキャッシュの削除、キャッシュに格納されているデータの削除、またはストアデータファイルの削除を行った場合、動作は保証されません。また、`eztool status` コマンドで、実行中の Operation として表示されません。

## (b) `-d` または `--directory` <ストアデータファイルの出力先パス名>

ストアデータファイルの出力先パス名を指定します。

指定したディレクトリ下にストアデータファイルを出力したい場合に指定します。

パス名には、アスタリスク (\*), 二重引用符 ("), 疑問符 (?), 縦線 (|), 小なり演算子 (<), 大なり演算子 (>) を含むディレクトリは指定できません。

ストアデータファイルの出力先パスに相対パスを指定した場合、各 EADs サーバの運用ディレクトリが起点となります。

## (c) <ストアデータファイルキー>

ストアデータファイルのストアデータファイルキーを指定します。

ストアデータファイルキーは、半角 32 文字までです。半角英数字 (0~9, A~Z, a~z), アンダースコア (\_) またはハイフン (-) が指定できます。

また、ストアデータファイルキーには、自動的に接頭辞 (eads\_ または eads\_single\_) と接尾辞 (\_EADs サーバ ID.拡張子) が付きます。拡張子は「.esd」です。

ストアデータファイルキーを省略すると、次の表に示すように、コマンド実行日時がストアデータファイルキーとなります。

<code>-s</code> または <code>--single</code> オプション指定	ストアデータファイル名	世代管理の対象
なし	eads_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd	対象
あり	eads_single_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd	対象外

(凡例)

[YYYYMMDDhhmmss] : コマンド実行日時

YYYY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

## (5) リターンコード

0 : 成功

101 : 失敗

## (6) 注意事項

次に示す形式のストアデータファイルキーを指定すると、世代管理の対象になる場合があります。ストアデータファイルを世代管理の対象にたくない場合は、同じ形式にならないように指定してください。

- スストアデータファイルキーを省略した場合と同じ形式

eads\_[YYYYMMDDhhmmss]\_[EADs サーバ ID].esd

- `eztool stop` コマンド実行時に出力されるストアデータファイル名と同じ形式  
eads\_stop\_[YYYYMMDDhhmmss]\_[EADs サーバ ID].esd

(凡例)

[YYYYMMDDhhmmss] : コマンド実行日時

YYYY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

また, `eztool stop` コマンド実行時に出力されるストアデータファイル名と同じ形式の場合, 削除されるおそれがあるため, 注意してください。詳細については, 「[7.6.2 ストアデータファイルの世代数の設定](#)」を参照してください。

## 14.3.12 import (データの読み込み)

### (1) 機能

メモリキャッシュのデータを書き出したストアデータファイルから, データを読み込みます。

ストアデータファイルのデータを再度 put することで, データを再配置します。

### (2) 規則

- このサブコマンドは, クラスタの状態がクラスタ稼働中 (AVAILABLE) のときに実行できます。
- このサブコマンドは, クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし, クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については, `eztool status` コマンドで確認できます。
- このサブコマンドは, 処理対象の EADs サーバが初期化状態 (initialized) のときに実行できます。
- 稼働中の FullGC の発生を抑えるため, このサブコマンドが終了するタイミングで, 各 EADs サーバが FullGC を実施します。
- このサブコマンドが EADs サーバから排他を取得している間, EADs サーバは縮退状態 (isolated) になりません。なお, EADs サーバから排他を取得している間にプロセスがダウンしたり, EADs サーバが停止したりした場合は, 排他を解除したあとに縮退が実行されます。

### (3) 形式

```
eztool import [-d <ストアデータファイルの格納先パス名>]
               [<ストアデータファイルキー>]
```

## (4) オプションおよび引数

### (a) `-d` または `--directory` <ストアデータファイルの格納先パス名>

ストアデータファイルの格納先パス名を指定します。

指定したディレクトリ下にあるストアデータファイルだけを読み込みたい場合に指定します。

パス名には、アスタリスク (\*), 二重引用符 ("), 疑問符 (?), 縦線 (|), 小なり演算子 (<), 大なり演算子 (>) を含むディレクトリは指定できません。

ストアデータファイルの格納先パスに相対パスを指定した場合、各 EADs サーバの運用ディレクトリが起点となります。

### (b) <ストアデータファイルキー>

ストアデータファイルのストアデータファイルキーを指定します。

`eztool export` コマンドまたは `eztool stop` コマンド実行時に出力されたストアデータファイルのストアデータファイルキーを指定します。

ストアデータファイルキーを省略すると、`eztool listesd` コマンドで、latest に表示される最新のストアデータファイルキーのストアデータファイルを自動的に読み込みます。

読み込み対象となるストアデータファイル名を次に示します。

ストアデータファイルを出力したコマンド	ストアデータファイル名
<code>eztool export</code>	<code>eads_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd</code>
<code>eztool stop</code>	<code>eads_stop_[YYYYMMDDhhmmss]_[EADs サーバ ID].esd</code>

(凡例)

[YYYYMMDDhhmmss] : コマンド実行日時

YYYY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

ストアデータファイルが各 EADs サーバ、またはストアデータファイルの格納先ディレクトリ下に複数ある場合、コマンド実行日時が最新のストアデータファイルを自動的に読み込みます。

コマンド実行日時が同じ (EADs サーバ ID だけ異なる) ストアデータファイルが複数ある場合、該当するすべてのストアデータファイルを読み込みます。

また、`eztool export` コマンド実行時に出力されたストアデータファイルと、`eztool stop` コマンド実行時に出力されたストアデータファイルのコマンド実行日時が同じ場合、`eztool export` コマンド実行時に出力されたストアデータファイルを優先して読み込みます。

## (5) リターンコード

0：成功

101：失敗

## (6) 注意事項

- 壊れている、または不正なストアデータファイルは読み込みません。
- 03-00 より古いバージョンの EADs サーバで出力したストアデータファイルを読み込んだ場合、動作は保証されません。
- データを一つ一つ登録し直すため、データ数やデータ量によっては、かなりの時間が掛かります。
- ストアデータファイルのデータを put する際、キャッシュに同じ名称の key が存在する場合は、key の更新日時をチェックして、ストアデータファイルの更新日時の方が新しいときにだけデータを上書きします。

### 14.3.13 deleteesd (ストアデータファイルの削除)

#### (1) 機能

指定されたストアデータファイルをクラスタから削除します。

#### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、[eztool status](#) コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞状態 (closed)

#### (3) 形式

```
eztool deleteesd [-d <ストアデータファイルの格納先パス名>]
                  <ストアデータファイルキー>
```

## (4) オプションおよび引数

### (a) `-d` または `--directory` <ストアデータファイルの格納先パス名>

ストアデータファイルの格納先パス名を指定します。

指定したディレクトリ下にあるストアデータファイルだけを削除したい場合に指定します。

パス名には、アスタリスク (\*), 二重引用符 ("), 疑問符 (?), 縦線 (|), 小なり演算子 (<), 大なり演算子 (>) を含むディレクトリは指定できません。

ストアデータファイルの格納先パスに相対パスを指定した場合、各 EADs サーバの運用ディレクトリが起点となります。

### (b) <ストアデータファイルキー>

クラスタから削除するストアデータファイルのストアデータファイルキーを指定します。

`eztool export` コマンドまたは `eztool stop` コマンド実行時に出力されたストアデータファイルのストアデータファイルキーを指定します。

ストアデータファイルキーは、半角 32 文字までです。半角英数字 (0~9, A~Z, a~z), アンダースコア (\_) またはハイフン (-) が指定できます。

## (5) リターンコード

0 : 成功

101 : 失敗

## (6) 注意事項

指定したストアデータファイルキーのストアデータファイルはすべて削除されます。

例えば、ストアデータファイルキーに「20111101130203」を指定してこのコマンドを実行した場合、各 EADs サーバにある `eads_20111101130203_[EADs サーバ ID].esd` ファイルはすべて削除されます。

## 14.3.14 put (指定したデータの格納)

### (1) 機能

指定した value を key に関連づけて格納します。

このコマンドは、構築後の実行環境が正常に動作するかどうかテストするために使用します。

## (2) 規則

このサブコマンドは EADs サーバが稼働状態 (running) のときに実行できます。

## (3) 形式

```
eztool put <キャッシュ名> <key> <value>
```

## (4) オプションおよび引数

### (a) <キャッシュ名>

value を格納するキャッシュの名称を指定します。

キャッシュ名に指定できる文字を次に示します。

- キャッシュ作成時にキャッシュ定義ファイルを使用しなかった場合  
半角 32 文字までの、ASCII コード「0x20」～「0x7E」の文字が指定できます。
- キャッシュ作成時にキャッシュ定義ファイルを使用した場合  
半角 32 文字までの、半角英数字 (0~9, A~Z, および a~z) が指定できます。

### (b) <key>

格納する value に関連づけられた key を指定します。

key には、半角 1,024 文字までの、ASCII コード「0x20」～「0x7E」の文字列 (java.lang.String) が指定できます。

ただし、次の場合はエラーとなります。

- 区切り文字 (:) が key の先頭または後尾にある場合  
(例) 「:element」「group:」
- 区切り文字 (:) が連続している場合  
(例) 「group::element」

### (c) <value>

格納する value を指定します。

value には、半角 1,024 文字までの文字列 (java.lang.String) が指定できます。

## (5) リターンコード

0: 成功



## (6) 注意事項

API で操作したデータをコマンドで操作したり、コマンドで操作したデータを API で操作したりすることができます。

ただし、key や value に指定できるデータ型とサイズについて、API との完全な互換性はありません。例えば、このコマンドで文字列 (java.lang.String) 以外のオブジェクトや、半角 1,024 文字以上の value を格納することはできません。

### 14.3.15 get (指定したデータの取得)

#### (1) 機能

指定した value を取得します。

このコマンドは、構築後の実行環境が正常に動作するかどうかテストするために使用します。

#### (2) 規則

このサブコマンドは EADs サーバが稼働状態 (running) のときに実行できます。

#### (3) 形式

```
eztool get <キャッシュ名> <key>
```

#### (4) オプションおよび引数

##### (a) <キャッシュ名>

取得する value が格納されているキャッシュの名称を指定します。

キャッシュ名に指定できる文字を次に示します。

- キャッシュ作成時にキャッシュ定義ファイルを使用しなかった場合  
半角 32 文字までの、ASCII コード [0x20] ~ [0x7E] の文字が指定できます。
- キャッシュ作成時にキャッシュ定義ファイルを使用した場合  
半角 32 文字までの、半角英数字 (0~9, A~Z, および a~z) が指定できます。

##### (b) <key>

取得する value に関連づけられた key を指定します。

key には、半角 1,024 文字までの、ASCII コード「0x20」～「0x7E」の文字列 (java.lang.String) が指定できます。

ただし、次の場合はエラーとなります。

- 区切り文字 (:) が key の先頭または後尾にある場合  
(例) 「:element」 「group:」
- 区切り文字 (:) が連続している場合  
(例) 「group::element」

## (5) 出力例

eztool get コマンドの実行結果の出力例を次に示します。

```
$ eztool get cache1 key1
KDEA08001-I          The command will now start. (subcommand = get, parameter = [get, cache1, key1])

Value: value1
ValueSize: 12

KDEA08002-I          The command will now end.
```

表示内容を次の表に示します。

項番	表示内容	説明
1	Value	取得した value
2	ValueSize	取得した value のサイズ (単位: バイト)

## (6) リターンコード

0: 成功

101: 失敗

## (7) 注意事項

API で操作したデータをコマンドで操作したり、コマンドで操作したデータを API で操作したりすることができます。

ただし、key や value に指定できるデータ型とサイズについて、API との完全な互換性はありません。

## 14.3.16 remove (指定したデータの削除)

### (1) 機能

指定した key, および key に関連づけられた value を削除します。

このコマンドは、構築後の実行環境が正常に動作するかどうかテストするために使用します。

### (2) 規則

このサブコマンドは EADs サーバが稼働状態 (running) のときに実行できます。

### (3) 形式

```
eztool remove <キャッシュ名> <key>
```

### (4) オプションおよび引数

#### (a) <キャッシュ名>

削除する value が格納されているキャッシュの名称を指定します。

キャッシュ名に指定できる文字を次に示します。

- キャッシュ作成時にキャッシュ定義ファイルを使用しなかった場合  
半角 32 文字までの、ASCII コード「0x20」～「0x7E」の文字が指定できます。
- キャッシュ作成時にキャッシュ定義ファイルを使用した場合  
半角 32 文字までの、半角英数字 (0~9, A~Z, および a~z) が指定できます。

#### (b) <key >

削除する value に関連づけられた key を指定します。

key には、半角 1,024 文字までの、ASCII コード「0x20」～「0x7E」の文字列 (java.lang.String) が指定できます。

ただし、次の場合はエラーとなります。

- 区切り文字 (:) が key の先頭または後尾にある場合  
(例) 「:element」 「group:」
- 区切り文字 (:) が連続している場合  
(例) 「group::element」

## (5) リターンコード

0：成功

101：失敗

## (6) 注意事項

API で操作したデータをコマンドで操作したり，コマンドで操作したデータを API で操作したりすることができます。

ただし，key や value に指定できるデータ型とサイズについて，API との完全な互換性はありません。

## 14.3.17 listfunc (ユーザ関クションの実行可否の表示)

### (1) 機能

ユーザ関クションの実行可否を表示します。

### (2) 規則

- このサブコマンドは，クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは，クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし，クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については，[eztool status](#) コマンドで確認できます。
- このサブコマンドは，処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool listfunc [-v] [<ユーザ関クション名>]
```

## (4) オプションおよび引数

### (a) -v または--verbose

コマンド実行結果の詳細を表示したい場合に指定します。

### (b) <ユーザファンクション名>

実行可否を表示したいユーザファンクション名を指定します。

指定したユーザファンクションの情報だけを表示したい場合に指定します。

ユーザファンクション名には、半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

## (5) 出力例

eztool listfunc コマンドの実行結果の出力例を次に示します。

### (a) -v または--verbose オプションを指定しない場合

```
$ eztool listfunc
KDEA08001-I          The command will now start. (subcommand = listfunc, parameter = [listfunc])

FunctionCount: 3

FunctionName  Enable  Disable
FunctionA     3       0
FunctionB     2       1
FunctionC     0       1
-----
KDEA08002-I          The command will now end.
$
```

表示内容を次の表に示します。

項番	表示内容	説明
1	FunctionCount	ユーザファンクションの総数 ユーザファンクション名を指定した場合は表示されません。
2	FunctionName	ユーザファンクション名 ユーザファンクション名を指定した場合は、指定したユーザファンクション名だけが表示されます。 自然順序順で表示されます。
3	Enable	ユーザファンクションを実行できる状態の EADs サーバ数
4	Disable	ユーザファンクションを実行できない状態の EADs サーバ数

## (b) -v または--verbose オプションを指定する場合

```
$ eztool listfunc -v
KDEA08001-I The command will now start. (subcommand = listfunc, parameter = [listfunc, -v])

FunctionCount: 3

Server          FunctionName  Status
-----
XX.XXX.XXX.168:24600 FunctionA     enable
XX.XXX.XXX.168:24600 FunctionB     enable
XX.XXX.XXX.168:24600 FunctionC     none

XX.XXX.XXX.169:24700 FunctionA     enable
XX.XXX.XXX.169:24700 FunctionB     enable
XX.XXX.XXX.169:24700 FunctionC     none

XX.XXX.XXX.170:24800 FunctionA     enable
XX.XXX.XXX.170:24800 FunctionB     disable
XX.XXX.XXX.170:24800 FunctionC     disable
-----
KDEA08002-I The command will now end.
$
```

表示内容を次の表に示します。

項番	表示内容	説明
1	FunctionCount	ユーザ機能の総数 ユーザ機能名を指定した場合は表示されません。
2	Server	EADs サーバの IP アドレス、および EADs クライアントとの通信に使用する EADs サーバのポート番号 次の形式で表示されます。 IP アドレス:ポート番号
3	FunctionName	ユーザ機能名 ユーザ機能名を指定した場合は、指定したユーザ機能名だけが表示されます。
4	Status	ユーザ機能の実行可否 次のどちらかが表示されます。 <ul style="list-style-type: none"><li>• enable 実行できる</li><li>• disable 実行できない</li></ul> ユーザ機能が存在しない場合、none と表示されます。

### 注

実行結果は、次の優先順位に従って表示されます。

1. Server の値の自然順序順に表示されます。
2. Server の値が同じ行は、FunctionName の値の自然順序順に表示されます。

## (6) リターンコード

0：成功

101：失敗

## (7) 注意事項

出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。

## 14.3.18 execfunc (ユーザファンクションの実行)

### (1) 機能

指定したユーザファンクションを実行します。

ユーザファンクションを実行する EADs クライアントを構築しなくても、このコマンドでユーザファンクションを実行できます。

### (2) 規則

- このサブコマンドは EADs サーバが稼働状態 (running) のときに実行できます。
- EADs サーバの位置 (Position) の降順にユーザファンクションを実行します。実行途中で、一部の EADs サーバでユーザファンクションの実行に失敗しても、処理を継続します。

### (3) 形式

```
eztool execfunc [-g <グループ名>|-s]
                 <キャッシュ名> <ユーザファンクション名> [<ユーザファンクション引数>]
```

### (4) オプションおよび引数

#### (a) -g または --group <グループ名>

グループ名を指定してユーザファンクションを実行したい場合に指定します。

グループ名には、半角 1,024 文字までの、ASCII コード「0x20」～「0x7E」の文字が指定できます。

ただし、次の場合はエラーとなります。

- 区切り文字 (:) が key の先頭または後尾にある場合

(例) 「:element」 「group:」

- 区切り文字 (:) が連続している場合

(例) 「group::element」

## (b) -s または--single

コマンドを実行する EADs サーバだけでユーザファンクションを実行したい場合に指定します。

## (c) <キャッシュ名>

ユーザファンクションを実行するキャッシュの名称を指定します。

キャッシュ名に指定できる文字を次に示します。

- キャッシュ作成時にキャッシュ定義ファイルを使用しなかった場合  
半角 32 文字までの、ASCII コード 「0x20」 ~ 「0x7E」 の文字が指定できます。
- キャッシュ作成時にキャッシュ定義ファイルを使用した場合  
半角 32 文字までの、半角英数字 (0~9, A~Z, および a~z) が指定できます。

## (d) <ユーザファンクション名>

実行したいユーザファンクション名を指定します。

ユーザファンクション名には、半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

## (e) <ユーザファンクション引数>

ユーザファンクションに特定の引数を渡したい場合に指定します。

省略した場合は null が引数として渡されます。

ユーザファンクション引数には、半角 1,024 文字までの、ASCII コード 「0x20」 ~ 「0x7E」 の文字が指定できます。

## (5) 出力例

eztool execfunc コマンドの実行結果の出力例を次に示します。

```
$ eztool execfunc cache1 samplefunc.ExportCsvFunction export_test.csv
KDEA08001-I The command will now start. (subcommand = execfunc, parameter = [execfunc, cache1, samplefunc.ExportCsvFunction, export_test.csv])

Server      Result
XX.XXX.XXX.168:24600 Success
XX.XXX.XXX.168:24600 Success
XX.XXX.XXX.168:24600 Success
XX.XXX.XXX.168:24600 Success
XX.XXX.XXX.168:24600 Success
KDEA08002-I The command will now end.
```



表示内容を次の表に示します。

項番	表示内容	説明
1	Server	ユーザファンクションを実行した EADs サーバの IP アドレス, および EADs クライアントとの通信に使用する EADs サーバのポート番号 次の形式で表示されます。 IP アドレス:ポート番号
2	Result	成功時 取得したユーザファンクションの実行結果オブジェクトの toString() 値 ただし, null の場合は文字列の "null" が出力されます。 失敗時 発生した例外の情報

## (6) リターンコード

0 : 成功

101 : 失敗

失敗時のリターンコードは, コマンドの実行に失敗した場合に返します。ユーザファンクション実行時に例外が発生した場合, コマンドは成功として扱います。例外の内容については, Result の表示結果を参照してください。

## (7) 注意事項

通常のユーザファンクションとは異なり, ユーザファンクション引数と取得データの型は, 文字列 (java.lang.String) だけとなります。このコマンドを使用する場合は, この点を考慮してユーザファンクションを作成してください。

ただし, サイズの大きい文字列を使用すると, EADs サーバで GC が発生して性能に影響が出るおそれがあるので注意してください。

### 14.3.19 listevent (ユーザイベントハンドラの実行可否の表示)

#### (1) 機能

ユーザイベントハンドラの実行可否を表示します。

#### (2) 規則

- このサブコマンドは, クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)

- クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool listevent [-v] [<ユーザイベントハンドラ名>]
```

### (4) オプションおよび引数

#### (a) -v または --verbose

コマンド実行結果の詳細を表示したい場合に指定します。

#### (b) <ユーザイベントハンドラ名>

実行可否を表示したいユーザイベントハンドラ名を指定します。

指定したユーザイベントハンドラの情報だけを表示したい場合に指定します。

ユーザイベントハンドラ名には、半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

### (5) 出力例

`eztool listevent` コマンドの実行結果の出力例を次に示します。

## (a) -v または--verbose オプションを指定しない場合

```
$ eztool listevent
KDEA08001-I      The command will now start. (subcommand = listevent, parameter = [listevent])

EventCount: 7

EventName  Enable  Disable  Priority  OperationFilter  DataFilter
EventA     3       0  highest  [put, remove]   none
EventB     3       0 UNMATCH  [put]            [cache]
EventC     3       0  high    [remove]        [cache, group]
EventD     3       0  normal  UNMATCH         [cache, group, element]
EventE     2       1  low     [put, remove]   UNMATCH
EventF     1       1  lowest  [put]           none
EventG     0       2

-----
KDEA08002-I      The command will now end.
```

表示内容を次の表に示します。

項番	表示内容	説明
1	EventCount	ユーザイベントハンドラの総数 ユーザイベントハンドラ名を指定した場合は表示されません。
2	EventName	ユーザイベントハンドラ名 ユーザイベントハンドラ名を指定した場合は、指定したユーザイベントハンドラ名だけが表示されます。 自然順序順で表示されます。
3	Enable	ユーザイベントハンドラを実行できる状態の EADs サーバ数
4	Disable	ユーザイベントハンドラを実行できない状態の EADs サーバ数
5	Priority	ユーザイベントハンドラの優先度 次のどれかが表示されます。 <ul style="list-style-type: none"><li>• highest</li><li>• high</li><li>• normal</li><li>• low</li><li>• lowest</li></ul> 初期化に成功しているユーザイベントハンドラ間の優先度がそろっていない場合、UNMATCH と表示されます。 全ユーザイベントハンドラが初期化に成功していない場合は、何も表示されません。
6	OperationFilter	オペレーションタイプフィルタの内容 put, remove のうち、設定されているものが表示されます。 初期化に成功しているユーザイベントハンドラ間のオペレーションタイプフィルタの内容がそろっていない場合は、UNMATCH と表示されます。 全ユーザイベントハンドラが初期化に成功していない場合は、何も表示されません。
7	DataFilter	データフィルタの内容 cache, group, element のうち、設定されているものが表示されます。何も設定されていない場合は none が表示されます。

項番	表示内容	説明
		初期化に成功しているユーザイベントハンドラ間のデータフィルタの内容がそろっていない場合は、UNMATCH と表示されます。 全ユーザイベントハンドラが初期化に成功していない場合は、何も表示されません。

## (b) -v または--verbose オプションを指定する場合

```

$ eztool listevent -v
KDEA08001-I The command will now start. (subcommand = listevent, parameter = [listevent, -v])

EventCount: 7

Server          EventName  Status  Priority  OperationFilter  DataFilter
-----
XX.XXX.XXX.168:24600 EventA    enable  highest  [put, remove]    none
XX.XXX.XXX.168:24600 EventB    enable  highest  [put]             [cache = [hoge, foo]]
XX.XXX.XXX.168:24600 EventC    enable  high     [remove]          [cache = [hoge], group = [bar]]
XX.XXX.XXX.168:24600 EventD    enable  normal   [put, remove]     [cache = [foo], group = [boo], element=[bar]]
XX.XXX.XXX.168:24600 EventE    enable  low      [put, remove]     [group = [bar]]
XX.XXX.XXX.168:24600 EventF    enable  lowest   [put]             none
XX.XXX.XXX.168:24600 EventG    disable

XX.XXX.XXX.169:24700 EventA    enable  highest  [put, remove]    none
XX.XXX.XXX.169:24700 EventB    enable  high     [put]             [cache = [hoge, foo]]
XX.XXX.XXX.169:24700 EventC    enable  high     [remove]          [cache = [hoge], group = [bar]]
XX.XXX.XXX.169:24700 EventD    enable  normal   [remove]          [cache = [foo], group = [boo], element=[bar]]
XX.XXX.XXX.169:24700 EventE    enable  low      [put, remove]     [element = [bar]]
XX.XXX.XXX.169:24700 EventF    disable
XX.XXX.XXX.169:24700 EventG    disable

XX.XXX.XXX.170:24800 EventA    enable  highest  [put, remove]    none
XX.XXX.XXX.170:24800 EventB    enable  low      [put]             [cache = [hoge, foo]]
XX.XXX.XXX.170:24800 EventC    enable  normal   [remove]          [cache = [hoge], group = [bar]]
XX.XXX.XXX.170:24800 EventD    enable  low      [put]             [cache = [foo], group = [boo], element=[bar]]
XX.XXX.XXX.170:24800 EventE    disable
XX.XXX.XXX.170:24800 EventF    none
XX.XXX.XXX.170:24800 EventG    none

KDEA08002-I The command will now end.

```

表示内容を次の表に示します。

項番	表示内容	説明
1	EventCount	ユーザイベントハンドラの総数 ユーザイベントハンドラ名を指定した場合は表示されません。
2	Server	EADs サーバの IP アドレス、および EADs クライアントとの通信に使用する EADs サーバのポート番号 次の形式で表示されます。 IP アドレス:ポート番号
3	EventName	ユーザイベントハンドラ名 ユーザイベントハンドラ名を指定した場合は、指定したユーザイベントハンドラ名だけが表示されます。
4	Status	ユーザイベントハンドラの実行可否 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• enable 実行できる</li> <li>• disable 実行できない</li> </ul>

項番	表示内容	説明
		ユーザイベントハンドラが存在しない場合、none と表示されます。
5	Priority	<p>ユーザイベントハンドラの優先度 次のどれかが表示されます。</p> <ul style="list-style-type: none"> <li>• highest</li> <li>• high</li> <li>• normal</li> <li>• low</li> <li>• lowest</li> </ul> <p>Status に disable または none と表示されるユーザイベントハンドラについては、何も表示されません。</p>
6	OperationFilter	<p>オペレーションタイプフィルタの内容 put, remove のうち、設定されているものが表示されます。</p> <p>Status に disable または none と表示されるユーザイベントハンドラについては、何も表示されません。</p>
7	DataFilter	<p>データフィルタの内容 cache, group, element のうち、設定されているものが表示されます。何も設定されていない場合は none が表示されます。</p> <p>Status に disable または none と表示されるユーザイベントハンドラについては、何も表示されません。</p>

## 注

実行結果は、次の優先順位に従って表示されます。

1. Server の値の自然順序順に表示されます。
2. Server の値が同じ行は、EventName の値の自然順序順に表示されます。

## (6) リターンコード

0：成功

101：失敗

## (7) 注意事項

出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。

### 14.3.20 listecf (永続データに関する情報の一覧表示)

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

## (1) 機能

クラスタ内の永続データに関する情報の一覧を表示します。

## (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online および offline の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
- `-s` または `--single` オプションを指定した場合は、クラスタの状態に関係なく、EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)

## (3) 形式

```
eztool listecf [-v [--divide <分割する個数>] ] [-s] [--threshold <しきい値>]  
               [<キャッシュ名>]
```

## (4) オプションおよび引数

### (a) `-v` または `--verbose`

コマンド実行結果の詳細を表示したい場合に指定します。

## (b) `--divide` <分割する個数>

`-v` または `--verbose` オプションを指定した場合、コンパクションの効果のファイル数の分布が表示されます。コンパクションの効果のファイル数の分布を、任意の個数で分割したい場合に指定します。

分割する個数には、1 から 10 までの整数が指定できます。

このオプションを指定しない場合、コマンド定義の `eads.management.compaction.effect.division` パラメタに指定した値（デフォルトは 5）が分割する個数になります。

## (c) `-s` または `--single`

コマンドを実行する EADs サーバの実行結果だけを表示したい場合に指定します。

## (d) `--threshold` <しきい値>

しきい値（単位：%）以上のコンパクションの効果があるキャッシュデータファイルのファイル数を表示したい場合に指定します。

しきい値には、1 から 100 までの整数が指定できます。

このオプションを指定しない場合、コマンド定義の `eads.management.compaction.effect.threshold` パラメタに指定した値（デフォルトは 50）がしきい値になります。

## (e) <キャッシュ名>

指定したキャッシュ名のキャッシュの情報だけを表示したい場合に指定します。

キャッシュ名には、半角 32 文字までの半角英数字（0~9, A~Z, および a~z）が指定できます。

## (5) 出力例

`eztool listecf` コマンドの実行結果の出力例を次に示します。

### ■ `-v` または `--verbose` オプションを指定しない場合

```

$ eztool listecf
KDEA08001-I          The command will now start. (subcommand = listecf, parameter = [listecf])

FC: FileCount
CE: CompactionEffect
Ex: Exist

Cache  ExCache  Range  Server                UnusedFC  MaxCE  FilterCE(50%)
cache1  true      1      XX.XXX.XXX.168:24600  6         0      0
        XX.XXX.XXX.168:24700  6         0      0
        XX.XXX.XXX.168:24800  6         0      0
        2      XX.XXX.XXX.168:24700  6         0      0
        XX.XXX.XXX.168:24800  6         0      0
        XX.XXX.XXX.168:24900  6         0      0
        3      XX.XXX.XXX.168:24800  6         0      0
        XX.XXX.XXX.168:24900  6         0      0
        XX.XXX.XXX.168:25000  6         0      0
        4      XX.XXX.XXX.168:24900  6         0      0
        XX.XXX.XXX.168:25000  6         0      0
        XX.XXX.XXX.168:24600  6         0      0
        5      XX.XXX.XXX.168:25000  6         0      0
        XX.XXX.XXX.168:24600  6         0      0
        XX.XXX.XXX.168:24700  6         0      0
cache2  true      1      XX.XXX.XXX.168:24600  6         0      0
        XX.XXX.XXX.168:24700  6         0      0
        XX.XXX.XXX.168:24800  6         0      0
        2      XX.XXX.XXX.168:24700  6         0      0
        XX.XXX.XXX.168:24800  6         0      0
        XX.XXX.XXX.168:24900  6         0      0
        3      XX.XXX.XXX.168:24800  6         0      0
        XX.XXX.XXX.168:24900  6         0      0
        XX.XXX.XXX.168:25000  6         0      0
        4      XX.XXX.XXX.168:24900  6         0      0
        XX.XXX.XXX.168:25000  6         0      0
        XX.XXX.XXX.168:24600  6         0      0
        5      XX.XXX.XXX.168:25000  6         0      0
        XX.XXX.XXX.168:24600  6         0      0
        XX.XXX.XXX.168:24700  6         0      0
cache3  true      1      XX.XXX.XXX.168:24600  6         0      0
        XX.XXX.XXX.168:24700  6         0      0
        XX.XXX.XXX.168:24800  6         0      0
        2      XX.XXX.XXX.168:24700  6         0      0
        XX.XXX.XXX.168:24800  6         0      0
        XX.XXX.XXX.168:24900  6         0      0
        3      XX.XXX.XXX.168:24800  6         0      0
        XX.XXX.XXX.168:24900  6         0      0
        XX.XXX.XXX.168:25000  6         0      0
        4      XX.XXX.XXX.168:24900  6         0      0
        XX.XXX.XXX.168:25000  6         0      0
        XX.XXX.XXX.168:24600  6         0      0
        5      XX.XXX.XXX.168:25000  6         0      0
        XX.XXX.XXX.168:24600  6         0      0
        XX.XXX.XXX.168:24700  6         0      0
-----
KDEA08002-I          The command will now end.
$

```

■-v または--verbose オプションを指定する場合



```

$ eztool listecf -v
KDEA08001-1 The command will now start. (subcommand = listecf, parameter = [listecf, -v])

CP: CacheProperties
CI: CacheInfoFile
CD: CacheDataFiles
FC: FileCount
CE: CompactionEffect
Ac: Accord
Ex: Exist

Cache  ExCache  AcCP  AcCI  Range  Server  ExCP  ExCI  ExCD  UnusedFC  MaxCE  FilterCE (50%)  CE (0-20%)  (21-40%)  (41-60%)  (61-80%)  (81-100%)
cache1 true     true  true  1  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0
  2  XX.XXX.XXX.168:24700 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24800 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24900 true true true 5 100 1 7 0 0 0 0 1
  3  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  4  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
cache2 true     true  true  1  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  2  XX.XXX.XXX.168:24700 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24800 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24900 true true true 5 100 1 7 0 0 0 0 1
  3  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  4  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
cache3 true     true  true  1  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  2  XX.XXX.XXX.168:24700 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24800 true true true 5 100 1 7 0 0 0 0 1
  XX.XXX.XXX.168:24900 true true true 5 100 1 7 0 0 0 0 1
  3  XX.XXX.XXX.168:24800 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  4  XX.XXX.XXX.168:24900 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:25000 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24600 true true true 6 0 0 8 0 0 0 0 0
  XX.XXX.XXX.168:24700 true true true 6 0 0 8 0 0 0 0 0

-----
KDEA08002-1 The command will now end.
$

```

表示内容を次の表に示します。

項番	表示内容	説明	表示の有無		
			-v または --verbose オプション指定		-s または --single オプション指定
			なし	あり	
1	CP: CacheProperties	「CP」が「キャッシュ定義ファイル (CacheProperties)」の略称であることを表しています。	×	○	×
2	CI: CacheInfoFile	「CI」が「キャッシュ情報ファイル (CacheInfoFile)」の略称であることを表しています。	×	○	×
3	CD: CacheDataFiles	「CD」が「キャッシュデータファイル (CacheDataFiles)」の略称であることを表しています。	×	○	×
4	FC: FileCount	「FC」が「ファイル数 (FileCount)」の略称であることを表しています。	○	○	○

項番	表示内容	説明	表示の有無		
			-v または--verbose オプション指定		-s または--single オプション指定
			なし	あり	
5	CE: CompactionEffect	「CE」が「コンパクションの効果 (CompactionEffect)」の略称であることを表しています。	○	○	○
6	Ac: Accord	「Ac」が「一致する (Accord)」の略称であることを表しています。	×	○	×
7	Ex: Exist	「Ex」が「存在する (Exist)」の略称であることを表しています。	○	○	×
8	Cache	キャッシュ名 ディスクキャッシュおよび 2Way キャッシュのキャッシュ名が、次のキャッシュの情報から抽出されて表示されます。 <ul style="list-style-type: none"> <li>• EADs サーバに存在するキャッシュ</li> <li>• キャッシュ定義ファイルに指定したキャッシュ</li> </ul>	○	○	○
9	ExCache	キャッシュが EADs サーバに存在しているかどうか 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• true EADs サーバに存在する</li> <li>• false EADs サーバに存在しない</li> </ul>	○	○	×
10	AcCP	EADs サーバ間でキャッシュ定義ファイルが一致しているかどうか キャッシュ定義ファイルの有無も含めて、すべての EADs サーバ間で一致しているかどうかが表示されます。 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• true 一致している</li> <li>• false 一致していない</li> </ul>	×	○	×
11	AcCI	EADs サーバ間でキャッシュ情報ファイルが一致しているかどうか キャッシュ情報ファイルが存在しない EADs サーバは無視されて、一致しているかどうかが表示されます。 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• true 一致している</li> </ul>	×	○	×

項番	表示内容	説明	表示の有無		
			-v または--verbose オプション指定		-s または--single オプション指定
			なし	あり	
		<ul style="list-style-type: none"> <li>• false 一致していない</li> </ul>			
12	Range	レンジ ID キャッシュ内のレンジを識別するための番号（1～96 の整数）です。データのコピー元 EADs サーバのサーバ ID と一致します。	○	○	○
13	Server	Range の範囲に含まれる EADs サーバの IP アドレス、および EADs クライアントとの通信に使用する EADs サーバのポート番号 次の形式で表示されます。 IP アドレス:ポート番号	○	○	×
14	ExCP	キャッシュ定義ファイルが存在するかどうか 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• true 存在する</li> <li>• false 存在しない</li> </ul> EADs サーバが起動していない場合、または通信障害が発生した場合は、何も表示されません。	×	○	×
15	ExCI	キャッシュ情報ファイルが存在するかどうか 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• true 存在する</li> <li>• false 存在しない</li> </ul> 次の場合は何も表示されません。 <ul style="list-style-type: none"> <li>• EADs サーバが起動していない場合</li> <li>• 通信障害が発生した場合</li> <li>• EADs サーバにキャッシュが読み込まれていない状態で、ExCP が false の場合</li> <li>• EADs サーバにキャッシュが読み込まれていない状態で、ExCI の情報取得に必要なパラメタがキャッシュ定義ファイルに指定されていない場合</li> </ul>	×	○	×
16	ExCD	キャッシュデータファイルが存在するかどうか 次のどちらかが表示されます。 <ul style="list-style-type: none"> <li>• true 存在する</li> </ul>	×	○	×

項番	表示内容	説明	表示の有無		
			-v または--verbose オプション指定		-s または--single オプション指定
			なし	あり	
		<ul style="list-style-type: none"> <li>• false 存在しない</li> </ul> 次の場合は何も表示されません。 <ul style="list-style-type: none"> <li>• EADs サーバが起動していない場合</li> <li>• 通信障害が発生した場合</li> <li>• EADs サーバにキャッシュが読み込まれていない状態で、ExCP が false の場合</li> <li>• EADs サーバにキャッシュが読み込まれていない状態で、ExCD の情報取得に必要なパラメタがキャッシュ定義ファイルに指定されていない場合</li> </ul>			
17	UnusedFC	使用していないファイル数 次の場合は何も表示されません。 <ul style="list-style-type: none"> <li>• EADs サーバが起動していない場合</li> <li>• 通信障害が発生した場合</li> <li>• EADs サーバに存在しないキャッシュの場合</li> </ul>	○	○	○
18	MaxCE	レンジ内のキャッシュデータファイルが持つコンパクションの効果の最大値 <sup>※1</sup> 次の場合は何も表示されません。 <ul style="list-style-type: none"> <li>• EADs サーバが起動していない場合</li> <li>• 通信障害が発生した場合</li> <li>• EADs サーバに存在しないキャッシュの場合</li> </ul>	○	○	○
19	FilterCE(x%) <sup>※2</sup>	コンパクションの効果がいずれの値以上のファイル数 <sup>※3</sup> 次の場合は何も表示されません。 <ul style="list-style-type: none"> <li>• EADs サーバが起動していない場合</li> <li>• 通信障害が発生した場合</li> <li>• EADs サーバに存在しないキャッシュの場合</li> </ul>	○	○	○
20	CE(x-y%) <sup>※4</sup>	コンパクションの効果のファイル数の分布 <sup>※3</sup> 次の場合は何も表示されません。 <ul style="list-style-type: none"> <li>• EADs サーバが起動していない場合</li> <li>• 通信障害が発生した場合</li> <li>• EADs サーバに存在しないキャッシュの場合</li> </ul>	×	○	○ (-v または--verbose オプションを指定した場合だけ、表示されます)

(凡例)

○：表示されます。

×：表示されません。

## 注

実行結果は、次の優先順位に従って表示されます。

1. Cache の値の自然順序順に表示されます。
2. Cache の値が同じ行は、Range の値の昇順に表示されます。
3. Cache の値、および Range の値が同じ行は、Sever の値の順（データの格納先 EADs サーバ、データのコピー先 EADs サーバの順）に表示されます。-s オプションまたは--single オプションを指定した場合は該当しません。

## 注※1

コンパクションの効果の小数点以下は切り捨てて表示されます。

## 注※2

x には、しきい値として設定した値が表示されます。

## 注※3

コンパクションの効果の小数点以下を切り捨てた値で、ファイル数が集計されます。

## 注※4

分割する個数から算出した範囲が表示されます。2 列目以降のヘッダには、範囲として「(x-y%)」だけが表示されます。

## (6) リターンコード

0：成功

101：失敗

## (7) 注意事項

出力処理中にコマンドがタイムアウトした場合、出力結果が欠落することがあります。また、出力処理が完了していても、コマンドがタイムアウトになることがあります。このような場合には、コマンドのタイムアウト時間を大きくしてから再実行してください。

## 14.3.21 resume (キャッシュの再開)

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

### (1) 機能

ディスクキャッシュ、および 2Way キャッシュを再開します。

このコマンドを実行すると、キャッシュ定義ファイルとキャッシュ情報ファイルが存在するキャッシュが再開されます。

次の場合は、このサブコマンドを使用して、キャッシュを再開します。

- 正常にクラスタを停止したあとで、EADs サーバを再起動する場合
- クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の状態から、クラスタを復旧する場合

## (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が offline および standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
- 稼働中の FullGC の発生を抑えるため、このサブコマンドが終了するタイミングで、各 EADs サーバが FullGC を実施します。
- このサブコマンドが EADs サーバから排他を取得している間、EADs サーバは縮退状態 (isolated) になりません。なお、EADs サーバから排他を取得している間にプロセスがダウンしたり、EADs サーバが停止したりした場合は、排他を解除したあとに縮退が実行されます。

## (3) 形式

```
eztool resume
```

## (4) リターンコード

0 : 成功

101 : 失敗

## (5) 注意事項

- キャッシュの再開の対象とならないキャッシュのキャッシュ定義の定義内容も、クラスタを構成する全 EADs サーバで同じになるようにしてください (`eads.cache.disk.info.dir` パラメタおよび `eads.cache.disk.<n>.dir` パラメタは除きます)。
- 正常にクラスタを停止していない場合、削除したデータが復活することがあります。
- キャッシュの再開が中途半端に終了した場合、排他は解除されません。キャッシュの再開を実行後、排他が解除されていない場合は、データが不整合になっているおそれがあります。そのまま運用を続けるとデータが破壊されたり、消失したりする危険があるため、次の手順で対処してください。

1. `eztool status -v` コマンドでクラスタの状態を確認する。
  2. `eztool unlock` コマンドで排他を解除する。
  3. `eztool listcache` コマンドでキャッシュの一覧を確認する。
  4. 必要に応じて `eztool deletetecache --with_deleteecf` コマンドでキャッシュを削除する。
  5. 再度キャッシュを再開する。
- キャッシュの再開に失敗した場合は、次の項目を確認して、再度キャッシュを再開してください。
    - クラスタ定義ファイルに指定したパラメタの指定値が誤っていないかどうか
    - キャッシュ定義ファイルに指定したパラメタの指定値が誤っていないかどうか
    - キャッシュファイルが不正な状態になっていないかどうか

## 14.3.22 importecf (永続データの再配置)

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

### (1) 機能

指定したディレクトリのキャッシュデータファイル、およびキャッシュインデックスファイルのデータを再度 put することで、永続データを再配置します。

EADs サーバをクラスタに追加したり、EADs サーバをクラスタから削除したりする場合は、このサブコマンドを使用して、キャッシュデータファイルからデータを再配置します。

### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
- 稼働中の FullGC の発生を抑えるため、このサブコマンドが終了するタイミングで、各 EADs サーバが FullGC を実施します。
- このサブコマンドが EADs サーバから排他を取得している間、EADs サーバは縮退状態 (isolated) になりません。なお、EADs サーバから排他を取得している間にプロセスがダウンしたり、EADs サーバが停止したりした場合は、排他を解除したあとに縮退が実行されます。

### (3) 形式

```
eztool importecf <キャッシュデータファイルとキャッシュインデックスファイルの格納先パス名>
```

### (4) オプションおよび引数

#### (a) <キャッシュデータファイルとキャッシュインデックスファイルの格納先パス名>

読み込みたいキャッシュデータファイルとキャッシュインデックスファイルの格納先パス名を指定します。

パス名には、アスタリスク (\*), 二重引用符 ("), 疑問符 (?), 縦線 (|), 小なり演算子 (<), 大なり演算子 (>) を含むディレクトリは指定できません。

キャッシュデータファイルとキャッシュインデックスファイルの格納先パスに相対パスを指定した場合、運用ディレクトリが起点となります。

### (5) リターンコード

0: 成功

101: 失敗

### (6) 注意事項

- キャッシュ情報ファイル, キャッシュインデックスファイル, およびキャッシュデータファイルの出力先ディレクトリは指定できません。
- このサブコマンドを実行すると, 読み込み対象のディレクトリ下にあるキャッシュ永続データファイル, およびキャッシュインデックスファイルはすべて読み込まれます。
- 読み込み対象となるキャッシュデータファイル, およびキャッシュインデックスファイルが破損していたり, 障害時のファイルだったりした場合は, 削除したデータが復活することがあります。
- 読み込み対象となるキャッシュデータファイル, およびキャッシュインデックスファイルの処理中にエラーが発生した場合, 次のキャッシュデータファイル, およびキャッシュインデックスファイルの処理を続けます。ただし, サブコマンドの結果はエラーとなります。このため, サブコマンドの結果がエラーとなった場合は, 次の内容に問題がないかどうかを確認してください。
  - `eztool status` コマンドでクラスタの状態を確認する。
  - `eztool listcache` コマンドでキャッシュの一覧を確認する。
- データを一つ一つ登録し直すため, データ数やデータ量によっては, かなりの時間が掛かります。
- put する key のデータがすでに存在する場合は, key の更新日時をチェックして, put する key のデータの更新日時が新しいときだけデータを上書きします。



## 14.3.23 deleteecf (キャッシュファイルの削除)

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

### (1) 機能

指定したキャッシュ名のキャッシュファイルを削除します。

このサブコマンドを実行すると、各 EADs サーバにあるキャッシュ定義ファイルに指定したパラメタに従って、次のキャッシュファイルが削除されます。

- <キャッシュ定義の `eads.cache.disk.info.dir` パラメタに指定したディレクトリ>/<キャッシュ名>/  
`eads_info_[EADs サーバ ID*1]_[キャッシュ名].ecf`  
注※ 01~96 の整数です。
- <キャッシュ定義の `eads.cache.disk.info.dir` パラメタに指定したディレクトリ>/<キャッシュ名>/  
`eads_index_[EADs サーバ ID*1]_[キャッシュ名]_[レンジ ID*1]_[nnnn*2].ecf`  
注※1 01~96 の整数です。  
注※2 ファイル数の連番 (5桁の整数) です。
- <キャッシュ定義の `eads.cache.disk.<n>.dir` パラメタに指定したディレクトリ>/<キャッシュ名>/  
`eads_data_[EADs サーバ ID*1]_[キャッシュ名]_[レンジ ID*1]_[nnnn*2].ecf`  
注※1 01~96 の整数です。  
注※2 ファイル数の連番 (5桁の整数) です。

キャッシュファイルの削除後、次のディレクトリ内が空になっていれば、ディレクトリも削除します。

- <キャッシュ定義の `eads.cache.disk.info.dir` パラメタに指定したディレクトリ>/<キャッシュ名>
- <キャッシュ定義の `eads.cache.disk.<n>.dir` パラメタに指定したディレクトリ>/<キャッシュ名>

### (2) 規則

- このサブコマンドは、クラスタが次の状態のときに実行できます。
  - クラスタ稼働中 (AVAILABLE)
  - クラスタ一部稼働中 (PARTIALLY\_AVAILABLE)
- このサブコマンドは、クラスタへの参加状況が `online` の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が `standby` の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞状態 (closed)

なお、-l オプションまたは--local オプションを指定した場合は、EADs サーバが停止状態のときにだけ実行できます。

### (3) 形式

```
eztool deleteecf <キャッシュ名> [-l]
```

### (4) オプションおよび引数

#### (a) <キャッシュ名>

削除したいキャッシュファイルのキャッシュ名を指定します。

キャッシュ名には、半角 32 文字までの半角英数字 (0~9, A~Z, および a~z) が指定できます。

#### (b) -l または--local

コマンドを実行する EADs サーバのキャッシュファイルだけを削除したい場合に指定します。

例えば、データの更新頻度や削除頻度が高いシステムで縮退状態が発生した場合、復旧に不要なキャッシュファイルをあらかじめ削除して、データの読み込み時間を削減したいときに指定します。

このオプションを指定した場合、処理対象の EADs サーバを停止するときに、キャッシュファイルが削除されます。処理対象の EADs サーバが起動中のときは削除されません。

### (5) リターンコード

0：成功

101：失敗

### (6) 注意事項

- EADs サーバが使用しているキャッシュのキャッシュファイルは削除できません。また、指定したキャッシュ名のキャッシュ定義ファイルが存在しない場合は、キャッシュファイルを削除できません。

## 14.3.24 compaction (キャッシュデータファイルのコンパクション)

この項の内容は、ディスクキャッシュ、および 2Way キャッシュを使用する場合にお読みください。

### (1) 機能

サブコマンドを実行した EADs サーバのキャッシュデータファイルをコンパクションします。

## (2) 規則

- このサブコマンドは、クラスタの状態に関係なく、EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
- このサブコマンドを実行した EADs サーバの中で、コンパクションの効果が最も大きいキャッシュデータファイルが実行対象に選ばれます。対象が複数ファイルになる場合は、コンパクションの効果が大きい順に処理されます。

## (3) 形式

```
eztool compaction [--cache <キャッシュ名> [--range <レンジID>] ]  
                  [--limit <実行回数> | --unused_fc <未使用ファイル数>]  
                  [--threshold <しきい値>]
```

```
eztool compaction [--break]
```

## (4) オプションおよび引数

### (a) --cache <キャッシュ名>

コンパクションを実行するキャッシュの名称を指定します。

キャッシュ名には、半角 32 文字までの半角英数字 (0~9, A~Z, および a~z) が指定できます。

このオプションを指定しない場合は、すべてのキャッシュが対象となります。

### (b) --range <レンジ ID >

レンジ ID は、キャッシュ内のレンジを識別するための番号 (1~96 の整数) です。データのコピー元 EADs サーバのサーバ ID と一致します。

コンパクションを実行するレンジを指定します。指定されたレンジ ID のレンジに所属している各 EADs サーバで、それぞれコンパクションします。

このオプションを指定しない場合は、指定されたキャッシュのすべてのレンジが対象となります。

### (c) --limit <実行回数>

コンパクションの実行回数の上限 (1 度のコマンド実行でコンパクションするキャッシュデータファイル数の上限) を指定します。実行対象のファイルがない場合は、指定した値より少ない回数であっても終了します。

実行回数には、1 から 20971520 までの整数が指定できます。

このオプションを指定しない場合、デフォルトは 20971520 です。

#### (d) `--unused_fc` <未使用ファイル数>

未使用のキャッシュデータファイルが、指定した数になるまでコンパクションを実行したい場合に指定します。すべてのレンジの未使用ファイル数が指定値以上の場合は、コンパクションは実行されません。実行対象のファイルがない場合や、コンパクションの実行回数が 20,971,520 回に達した場合は、指定した値よりも未使用ファイル数が少ない状態でも、警告を出して終了します。

未使用ファイル数には、1 から 32766 までの整数が指定できます。

#### (e) `--threshold` <しきい値>

しきい値（単位：%）以上の効果があるキャッシュデータファイルを対象に、コンパクションを実行したい場合に指定します。しきい値以上の効果があるキャッシュデータファイルがない場合は、コンパクションは実行されません。

しきい値には、1 から 100 までの整数が指定できます。

このオプションを指定しない場合は、コマンド定義の `eads.management.compaction.effect.threshold` パラメタに指定した値（デフォルトは 50）がしきい値になります。

#### (f) `--break`

コンパクションを中断したいときに指定します。

コンパクションの中断は、ファイル単位で実行します。コンパクションを実行しているファイルの途中で中断することはできません。

### (5) リターンコード

0：成功

1：実行対象のファイルが存在しなかったため、コンパクションを実行しませんでした。

2：未使用ファイル数が`--unused_fc` オプションに指定した値以上に存在するため、コンパクションを実行しませんでした。

3：未使用ファイル数が`--unused_fc` オプションに指定した値に達する前に、実行対象のファイルがなくなりました。コンパクションは実行済みです。

10：`--break` オプションが指定されましたが、コンパクションは実行されていません。

11：`--break` オプションが指定されましたが、すでにコンパクションの中断が予約されています。

101：失敗

## (6) 注意事項

- このサブコマンドは、EADs サーバが稼働状態 (running) でも実行できますが、性能に影響が出るおそれがあるため、注意してください。
- コンパクションの中断を予約している場合に、このサブコマンドがタイムアウトしたり、プロセスが停止したりしたときは、中断待ちプロセスは完了しません。プロセスがタイムアウトするまで待つか、プロセスを停止してください。
- あるファイルのコンパクションが終わってから、次のファイルのコンパクションが始まるまでの時間は、コンパクションの実行中ではありません。このため、`--break` オプションを指定してコンパクションを中断しようとしても、コンパクションが実行されていないというメッセージが出力され、中断できない場合があります。その場合は、コンパクションが中断できるまで、`eztool compaction --break` コマンドを繰り返し実行してください。

## 14.3.25 cprfed (PRF トレース情報の編集出力)

### (1) 機能

バッファ上に格納されている PRF トレース情報を PRF トレースファイルに出力します。PRF トレース情報は CSV 形式で編集出力されます。

なお、PRF トレース情報は <サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ>/`spool/utt/prf/REQ_PRF_ID/dcopltrc` 下に出力されます。

サーバ定義の `eads.logger.dir` パラメタを指定していない場合は、<運用ディレクトリ>/`logs/spool/utt/prf/REQ_PRF_ID/dcopltrc` 下に出力されます。

### (2) 規則

- このサブコマンドはクラスタおよび EADs サーバの状態に関係なく実行できます。
- Developer でこのサブコマンドを実行する場合、管理者特権で EADs サーバを起動してください。管理者特権がないと、エラーとなり実行できません。

### (3) 形式

```
eztool cprfed [-Time <開始時刻>, <終了時刻>]
```

### (4) オプションおよび引数

#### (a) -Time <開始時刻>, <終了時刻>

指定された時刻内に取得された PRF トレース情報だけを出力したい場合に指定します。

開始時刻、終了時刻は、hhmmss[MMdd[yyyy]]形式で指定します。MMdd, yyyy は省略できます。開始時刻と終了時刻は二重引用符 (") で囲みます。

hhmmss[MMdd[yyyy]] :

時 (hh), 分 (mm), 秒 (ss), 月 (MM), 日 (dd), 年 (yyyy)

MMdd, yyyy を省略すると、コマンド実行月日年が設定されます。

また、どちらかの時刻を省略できます。開始時刻を省略した場合は、PRF トレースファイルの最初から、終了時刻を省略した場合は、PRF トレースファイルの最後まで出力されます。

このオプションを省略すると、前回ファイル出力した PRF トレース情報の、次の PRF トレース情報から、コマンドが入力された時点までの PRF トレース情報が出力されます。

2012 年 4 月 1 日にコマンドを実行した場合の引数の指定例とその編集範囲の例を次に示します。

引数の指定	編集範囲
-Time "102345,"	2012 年 4 月 1 日 10 時 23 分 45 秒から PRF トレースファイルの最後まで
-Time ",102345"	PRF トレースファイルの最初から 2012 年 4 月 1 日 10 時 23 分 45 秒まで
-Time "102345,102350"	2012 年 4 月 1 日 10 時 23 分 45 秒から 2012 年 4 月 1 日 10 時 23 分 50 秒まで
-Time "1023450401,1023500414"	2012 年 4 月 1 日 10 時 23 分 45 秒から 2012 年 4 月 14 日 10 時 23 分 50 秒まで
-Time "10234510102011,10235001102012"	2011 年 10 月 10 日 10 時 23 分 45 秒から 2012 年 1 月 10 日 10 時 23 分 50 秒まで

## (5) 出力例

eztool cprfed コマンドの実行結果の出力例を次に示します。

```
PRF, Process, Thread(hashcode), Trace, ProcessName, Event, Date, Time, Time(msec/usec/nsec), Rc, ClientAP IP, ClientAP PID, ClientAP CommNo., RootAP IP, RootAP PID, RootAP CommNo., SendSCD IP, SendSCD PID, ReceiveSCD IP, ReceiveSCD PID, INT, OPR, LookupName, OPT, ASCII
Rec, 516, 2204, 10, ctmd.exe, 0x1406, 2004/02/05, 16:46:55, 576/000/000, 0, XX. XXX. XX. XXX,
2148, 0x00000000000000107, XX. XXX. XX. XXX, 2200, 0x00000000000000104, XX. XXX. XX. XXX,
516, ****, ****, com/abcd/BookShop, orderBook, BookShop,
```

表示内容を次の表に示します。

項番	表示内容	説明
1	PRF	PRF トレースのレコードの状態が出力されます。次のどちらかが出力されます。 <ul style="list-style-type: none"> <li>Rec : 正常</li> <li>ErrRec : 異常</li> </ul>

項番	表示内容	説明
2	Process	PRF トレースを取得したプロセスのプロセス ID が出力されます。
3	Thread(hashcode)	PRF トレースを取得したプロセスのスレッド ID が出力されます。( ) 内にはスレッドのハッシュ値が出力されます。
4	Trace	保守情報です。
5	ProcessName	保守情報です。
6	Event	イベント ID (16 進数) が出力されます。出力されるイベント ID の意味については、「表 13-2 出力されるイベント ID とその意味」を参照してください。
7	Date	PRF トレースを取得した年月日が出力されます。
8	Time	PRF トレースを取得した時刻が秒単位で出力されます。
9	Time(msec/usec/nsec)	PRF トレースを取得した時刻がミリ秒/マイクロ秒/ナノ秒単位で出力されます。
10	Rc	リターンコード (16 進数) が出力されます。次のどちらかが出力されます。 <ul style="list-style-type: none"> <li>• 0: 処理成功</li> <li>• 1: 処理失敗</li> </ul>
11	ClientAP IP	保守情報です。
12	ClientAP PID	保守情報です。
13	ClientAP CommNo.	保守情報です。
14	RootAP IP	EADs クライアントまたは EADs サーバの IP アドレスが出力されます。 一連の通信処理は、RootAP IP 列, RootAP PID 列, および RootAP CommNo 列の値が同じになります。
15	RootAP PID	EADs クライアントが付与する番号が出力されます。
16	RootAP CommNo.	通信番号 (16 進数) が出力されます。
17	SendSCD IP	保守情報です。
18	SendSCD PID	保守情報です。
19	ReceiveSCD IP	保守情報です。
20	ReceiveSCD PID	保守情報です。
21	INT	イベント ID ごとに出力される情報が異なります。出力される情報については、「表 13-2 出力されるイベント ID とその意味」を参照してください。 なお、情報が出力されないこともあります。
22	OPR	イベント ID ごとに出力される情報が異なります。出力される情報については、「表 13-2 出力されるイベント ID とその意味」を参照してください。 なお、情報が出力されないこともあります。
23	LookupName	保守情報です。
24	OPT	保守情報です。

項番	表示内容	説明
25	ASCII	イベント ID ごとに出力される情報が異なります。出力される情報については、「表 13-2 出力されるイベント ID とその意味」を参照してください。 なお、情報が出力されないこともあります。

## (6) リターンコード

0 : eztool cprfed コマンドが正常に終了しました。

0 以外 : エラー

## (7) 注意事項

高負荷時に排他によるボトルネックが生じないように、PRF トレース情報の出力時に排他制御を行っていません。そのため、一部の PRF トレース情報が正しく出力されない場合があります。

### 14.3.26 threaddump (スレッドダンプの出力)

#### (1) 機能

EADs サーバのスレッドダンプを出力します。

なお、スレッドダンプはサーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下に出力されます。

サーバ定義の `eads.logger.dir` パラメタを指定していない場合は、<運用ディレクトリ>/logs 下に出力されます。

#### (2) 規則

- このサブコマンドはクラスタの状態に関係なく実行できます。
- このサブコマンドは、EADs サーバが次の状態のときに実行できます。
  - 初期化中 (initializing)
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
  - 停止中 (stopping)



### (3) 形式

```
eztool threaddump
```

### (4) リターンコード

1 : EADs サーバが見つかりませんでした。

1 以外 : JavaSE の jheapprof コマンドの終了コードを返します。

## 14.3.27 snapshot (ログ, 設定情報, ハードウェア情報, およびネットワーク情報の収集)

### (1) 機能

次のファイルを収集し、書庫ファイルとしてアーカイブします。

ただし、拡張子が .tar, .tar.gz, または .tgz のファイルは除きます。

- EADs ログ情報
  - 稼働中の EADs サーバのスレッドダンプ
  - サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下のファイル  
ただし、`eads_management_*`ファイル (\*は 0 文字以上の任意の文字列) は除きます。
  - コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ直下の `eads_management_*`ファイル (\*は 0 文字以上の任意の文字列)
  - コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリの `maintenance` ディレクトリ直下の `eads_management_*`ファイル (\*は 0 文字以上の任意の文字列)
  - `<運用ディレクトリ>/hs_err_pid*.log` (\*は 0 文字以上の任意の文字列)  
注 JavaVM のエラーレポートファイルの出力先ファイル名です。
- EADs 設定情報
  - ファンクション定義ファイル (`<運用ディレクトリ>/app` 下の定義ファイル)
  - 定義ファイル (`<運用ディレクトリ>/conf` 下のファイル)
  - 運用ディレクトリの構成情報 (`<運用ディレクトリ>/logs/snapshot_info/eads_info/eads_info_ls.txt`)
  - キャッシュデータファイルの一覧 (`<運用ディレクトリ>/logs/snapshot_info/cache_file_info/eads_data_<キャッシュ名>_<キャッシュ定義ファイルの通し番号>.txt`)
  - キャッシュ情報ファイルの一覧 (`<運用ディレクトリ>/logs/snapshot_info/cache_file_info/eads_info <キャッシュ名>.txt`)

- キャッシュインデックスファイルの一覧 (<運用ディレクトリ>/logs/snapshot\_info/cache\_file\_info/eads\_index\_<キャッシュ名>.txt)
- ハードウェア・ネットワーク情報
  - OS のバージョン情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_version.txt)
  - CPU 情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_cpu.txt)
  - メモリ情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_memory.txt)
  - ディスク情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_df.txt)
  - ネットワーク接続情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_netstat.txt)
  - カーネルパラメタ net.core 情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_netcore.txt)
  - ホストと IP アドレスの対応情報 (/etc/hosts ファイル)
  - ネットワークインタフェース情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_ifconfig.txt)
  - コマンドに割り当てる資源の制限情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_ulimit.txt)

なお、gz ファイルはサーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下に出力されます。

サーバ定義の `eads.logger.dir` パラメタを指定していない場合は、<運用ディレクトリ>/logs 下に出力されます。

ファイル名は、次に示す形式となります。

snapshot\_[EADs サーバ名]\_[YYYYMMDDhhmmss].tar.gz

(凡例)

[YYYYMMDDhhmmss] : コマンド実行日時

YYYY : 年, MM : 月, DD : 日, hh : 時 (00~23), mm : 分, ss : 秒

## (2) 規則

- このサブコマンドは Developer では実行できません。
- このサブコマンドはクラスタおよび EADs サーバの状態に関係なく実行できます。
- `-sd` または `--safedump` オプションを指定した場合、クラスタの状態がクラスタ稼働中 (AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) のときは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)

- 稼働状態 (running)
- 閉塞状態 (closed)
- 縮退状態 (isolated)
- -sd または --safedump オプションを指定した場合、クラスタの状態がクラスタ動作不能 (NOT\_AVAILABLE) のときは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 縮退状態 (isolated)
- -fd または --forcedump オプションを指定した場合は、クラスタの状態に関係なく、EADs サーバが次の状態のときに実行できます。
  - 初期化中 (initializing)
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
  - 停止中 (stopping)

### (3) 形式

```
eztool snapshot [-sd|-fd]
```

### (4) オプションおよび引数

#### (a) -sd または --safedump

`eztool threaddump` コマンドを実行してから、ログと設定ファイルを収集したい場合に指定します。

次の場合はエラーとなります。

- 排他の取得に失敗した場合
- EADs サーバが起動していない場合

#### (b) -fd または --forcedump

`eztool threaddump` コマンドを強制的に実行してから、ログと設定ファイルを収集したい場合に指定します。

EADs サーバが起動していない場合はエラーとなります。

`eztool threaddump` コマンドを強制的に実行することで、EADs サーバが縮退するおそれがあります。

## (5) リターンコード

0：成功

0 以外：失敗

## (6) 注意事項

- このサブコマンドで作成された gz ファイルは自動的に削除されません。また、作成されるファイル数の上限もありません。定期的に行う場合は、ディスク容量に注意してください。
- このサブコマンドの実行が終了しない場合は、ファイルシステムが壊れているおそれがあります。その場合はこのコマンドを強制終了して、ファイルシステムが壊れていないか確認してください。
- 収集対象のファイルに読み取り権限がないと、アーカイブされません。
- このサブコマンドの実行中、コマンド実行時に出力されるメッセージログファイルに、次のようなログが出力されます。これらは情報収集のために EADs が内部的に実行する処理であり、ユーザは使用できません。
  - オプションを指定していない場合  
snapshot, -v, -t, 60
  - -fd オプションを指定した場合  
snapshot, -v, -t, 60  
snapshot, -s, -v, -t, 60
  - -sd オプションを指定した場合  
snapshot, -v, -t, 60  
snapshot, -s, -v, -t, 60  
snapshot, --lock, -t, 60\*  
snapshot, --unlock, -t, 60\*  
注※ EADs サーバが縮退状態 (isolated) のときは出力されません。

## 14.3.28 stop (クラスタの終了)

### (1) 機能

クラスタ内の全 EADs サーバを終了します。

オプションを指定しない場合、メモリキャッシュのデータをストアデータファイルに書き出してから EADs サーバを終了します。

## (2) 規則

- このサブコマンドは、クラスタの状態がクラスタ稼働中 (AVAILABLE) のときに実行できます。
- このサブコマンドは、クラスタへの参加状況が online の EADs サーバを処理対象とします。ただし、クラスタへの参加状況が standby の EADs サーバがクラスタ内にある場合は実行できません。クラスタへの参加状況については、`eztool status` コマンドで確認できます。
- このサブコマンドは、処理対象の EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 閉塞状態 (closed)
- このサブコマンドが EADs サーバから排他を取得している間、EADs サーバは縮退状態 (isolated) になりません。なお、EADs サーバから排他を取得している間にプロセスがダウンしたり、EADs サーバが停止したりした場合は、排他を解除したあとに縮退が実行されます。

## (3) 形式

```
eztool stop [--no_export]
```

## (4) オプションおよび引数

### (a) --no\_export

EADs サーバ終了時に、メモリキャッシュのデータをストアデータファイルに出力しない場合に指定します。

## (5) リターンコード

0：成功

101：失敗

## (6) 注意事項

このコマンドの実行時に出力されるストアデータファイルの詳細については、「[7.6.2\(3\) eztool stop コマンド実行時に出力されるストアデータファイル](#)」を参照してください。

ストアデータファイルの出力先については、「[7.4.2 ファイルの出力先の設定](#)」を参照してください。

## 14.3.29 forcestop (EADs サーバの強制終了)

### (1) 機能

EADs サーバを強制終了します。

## (2) 規則

- このサブコマンドはクラスタの状態に関係なく実行できます。
- このサブコマンドは、EADs サーバが次の状態のときに実行できます。
  - 初期化中 (initializing)
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
  - 停止中 (stopping)

## (3) 形式

```
eztool forcestop
```

## (4) リターンコード

1 : EADs サーバが見つかりませんでした。

1 以外 : Linux では kill コマンドの終了コードを、Windows では taskkill コマンドの終了コードを返します。

## (5) 注意事項

このコマンドは、メモリキャッシュのデータをストアデータファイルに書き出さずに EADs サーバを強制終了します。このため、ほかの EADs サーバに同一レンジのデータが存在しない場合、メモリキャッシュのデータは失われます。

### 14.3.30 isolate (EADs サーバの縮退)

#### (1) 機能

EADs サーバを縮退させます。

#### (2) 規則

- このサブコマンドは Developer では実行できません。
- このサブコマンドは、クラスタの状態がクラスタ稼働中 (AVAILABLE) のときに実行できます。

- このサブコマンドは、EADs サーバが次の状態のときに実行できます。
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞状態 (closed)
  - 縮退状態 (isolated)
- 稼働状態 (running) でこのサブコマンドを実行した場合、実行中のユーザファンクションやユーザイベントハンドラが終了するのを待ってから縮退します。

### (3) 形式

```
eztool isolate [--stop]
```

### (4) オプションおよび引数

#### (a) --stop

EADs サーバの縮退後、終了まで行いたい場合に指定します。

すでに縮退している EADs サーバに対して実行した場合は、EADs サーバの終了だけを行います。

### (5) リターンコード

0：成功

101：失敗

### (6) 注意事項

- このサブコマンドを実行して EADs サーバを縮退したあと、クラスタ構成情報が更新されていない EADs クライアントから API を実行した場合、EADs サーバとの接続でエラーが発生することがあります。
- このサブコマンドを実行することでレンジが動作不能となり、データの継続操作ができなくなる場合は、エラーとなります。

# 15

## AP 開発の流れ

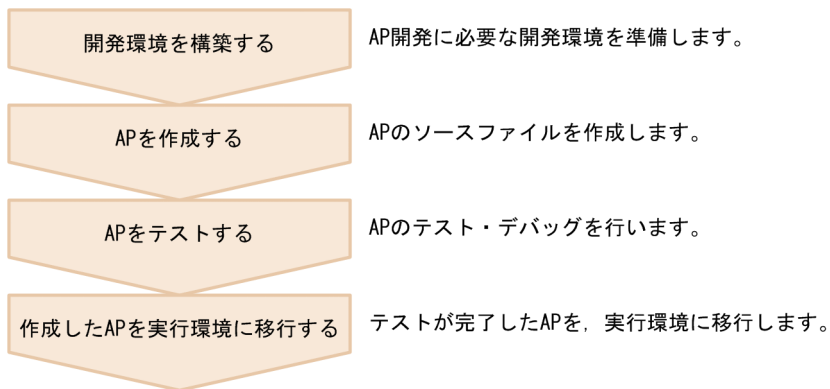
この章では、AP 開発の流れと、AP を開発する前に知っておくべき前提条件について説明します。



## 15.1 AP 開発の流れ

---

AP 開発の流れを次の図に示します。



### 15.1.1 開発環境を構築する

AP を作成するための開発環境を準備します。

AP の開発環境は、AP を作成する言語（Java 言語または C 言語）によって、必要なプログラムプロダクトが異なります。それぞれ必要なプログラムプロダクトをインストールし、AP の開発環境をセットアップします。

- Java 言語の場合
  - uCosminexus Elastic Application Data store
- C 言語の場合
  - uCosminexus Elastic Application Data store
  - uCosminexus Elastic Application Data store Client for C

次の項目を参照して、EADs サーバと EADs クライアントを実行環境と同じように構築して、開発環境として使用してください。

- [「5. インストールおよびセットアップを行う \(EADs サーバ\)」](#)
- [「6. インストールおよびセットアップを行う \(EADs クライアント\)」](#)

なお、ユーザファンクションまたはユーザイベントハンドラを作成する場合、次のプログラムプロダクトを提供しています。

- uCosminexus Elastic Application Data store Developer

Developer を使用してユーザファンクションまたはユーザイベントハンドラを作成する方は、「[16. Developer を構築する](#)」を参照してください。

## 15.1.2 AP を作成する

開発環境が構築できたら、AP を作成します。

Java 言語を使用してクライアント AP を作成する方は、「[17. クライアント AP を作成する \(Java 言語\)](#)」を参照してください。

ユーザファンクションを作成する方は、「[18. ユーザファンクションを作成する](#)」を参照してください。

ユーザイベントハンドラを作成する方は、「[19. ユーザイベントハンドラを作成する](#)」を参照してください。

C 言語を使用してクライアント AP を作成する方は、「[21. クライアント AP を作成する \(C 言語\)](#)」を参照してください。

## 15.1.3 AP をテストする

AP が完成したら、AP のテスト・デバッグを行います。

## 15.1.4 作成した AP を実行環境に移行する

AP のテストが完了したら、作成した AP を実行環境に移行します。

## 15.2 APを開発する前に知っておくべき前提条件

EADsのAPを開発する前に知っておくべき前提条件として、APの作成言語とEADsクライアントの種類や、key、value、キャッシュ名に指定できるデータなどについて説明します。

### 15.2.1 APの作成言語とEADsクライアント

EADsのAPを作成する際に使用できる言語と、EADsクライアントの種類について説明します。

#### (1) APの作成言語

EADsのAPは次の言語で作成できます。

- Java 言語
- C 言語

#### (2) EADsクライアントの種類

EADsクライアントには、APを作成する言語によって、次の2種類のプログラムプロダクト（クライアントライブラリ）があります。

- Java 言語の場合  
uCosminexus Elastic Application Data store Client for Java
- C 言語の場合  
uCosminexus Elastic Application Data store Client for C

実行環境で上記のプログラムプロダクトを使用する場合は、実行環境ごとにこれらのプログラムプロダクトをインストールしてください。

#### ■ 注意事項

- EADsクライアントより新しいバージョンのクライアントライブラリで作成したAPを使用した場合、動作は保証されません。この場合、ソースプログラムの変更や再コンパイルが必要です。なお、03-00以降のバージョンのクライアントライブラリで作成したAPは、新しいバージョンのEADsクライアントでも使用できます。
- 03-00より前のバージョンのEADsクライアント・EADsサーバと、03-00以降のバージョンのEADsクライアント・EADsサーバ間には互換性はありません。例えば、03-50のEADsクライアントおよびEADsサーバは、02-00以前のバージョンとの互換性はありません。また、互換性のないバージョン間で接続を試みた場合、動作は保証されません。

- ユーザファンクション内で EADs クライアントを使用する場合、EADs サーバと同じバージョンの EADs クライアントを使用してください。EADs サーバと異なるバージョンの EADs クライアントを使用した場合、動作は保証されません。

## 15.2.2 key, value, キャッシュ名, EADs クライアント名に指定できるデータ

key, value, キャッシュ名, EADs クライアント名に指定できるデータについて説明します。

### (1) key として指定できるデータ

key として指定できるデータ型と文字数を、AP を作成する言語ごとに次の表に示します。

表 15-1 key として指定できるデータ

AP を作成する言語	key として指定できるデータ型	文字数	備考
Java 言語	文字列 (java.lang.String)	1~1,024	<ul style="list-style-type: none"> <li>• ASCII コード「0x20」～「0x7E」の文字が指定できます。</li> <li>• key をグルーピングする場合、作成できるグループ数、およびグループの階層数に上限はありません。</li> <li>• key をグルーピングする場合、グループ名、区切り文字 (:), 要素名を合わせて 1,024 文字までとなります。</li> <li>• key をグルーピングする場合、グループ名と要素名にコロン (:) は指定できません。</li> <li>• null または空文字列は不正となります。</li> </ul>
C 言語	'%0'を終端記号とする文字列 (char *)	1~1,024	<ul style="list-style-type: none"> <li>• ASCII コード「0x20」～「0x7E」の文字が指定できます。</li> <li>• key をグルーピングする場合、作成できるグループ数、およびグループの階層数に上限はありません。</li> <li>• key をグルーピングする場合、グループ名、区切り文字 (:), 要素名を合わせて 1,024 文字までとなります。</li> <li>• key をグルーピングする場合、グループ名と要素名にコロン (:) は指定できません。</li> <li>• 文字列の最大サイズは、'%0'を含まないバイト数となります。</li> <li>• 長さ 0 の文字列 ('%0'だけ) は指定できません。</li> </ul>

### 注意事項

eztool put, eztool get, eztool remove の各コマンドは、構築後の実行環境が正常に動作するかどうかテストするために使用します。そのため、コマンドで指定できるデータ型とサイズと、API (put, get, remove) で指定できるデータ型とサイズの間には完全な互換性はありません。

## (2) valueとして指定できるデータ

valueとして指定できるデータ型とサイズを、APを作成する言語ごとに次の表に示します。

表 15-2 valueとして指定できるデータ

APを作成する言語	valueとして指定できるデータ型	サイズ (単位:バイト)	備考
Java 言語	シリアライズ可能な任意のオブジェクト (java.lang.Object)	1~262,144	<ul style="list-style-type: none"><li>シリアライズ後のバイト配列の長さとなります。</li><li>nullの場合は不正となります。</li></ul>
C 言語	任意のバイト配列 (void *)	1~262,144	任意のバイト配列の長さとなります。

### 注意事項

- 指定できるデータ型とサイズについて、API (put, get, remove)・コマンド (eztool put, eztool get, eztool remove) 間に完全な互換性はありません。
- Java 言語のクライアント AP またはユーザファンクションと、C 言語のクライアント AP で同一の key を扱う場合、value はバイト配列にしてください。Java 言語のクライアント AP またはユーザファンクションから格納したバイト配列の value を、C 言語のクライアント AP で取得できます。バイト配列でない value を C 言語のクライアント AP で取得した場合、エラーとなります。  
C 言語のクライアント AP から格納した value は、Java 言語のクライアント AP またはユーザファンクションでバイト配列として取得できます。
- C 言語のクライアント AP からユーザファンクションを実行する場合、ユーザファンクションの戻り値として扱うことができるオブジェクトは、バイト配列と null だけです。バイト配列および null 以外のオブジェクトを戻り値とした場合、エラーとなります。

## (3) キャッシュ名として指定できるデータ

キャッシュ名として指定できるデータ型と文字数を、APを作成する言語ごとに次の表に示します。

表 15-3 キャッシュ名として指定できるデータ

APを作成する言語	キャッシュ名として指定できるデータ型	文字数	備考
Java 言語	文字列 (java.lang.String)	1~32	<ul style="list-style-type: none"><li>ASCII コード「0x20」~「0x7E」の文字が指定できます。</li><li>null または空文字列は不正となります。</li></ul>
C 言語	'%0'を終端記号とする文字列 (char *)	1~32	<ul style="list-style-type: none"><li>ASCII コード「0x20」~「0x7E」の文字が指定できます。</li></ul>

APを作成する言語	キャッシュ名として指定できるデータ型	文字数	備考
			<ul style="list-style-type: none"> <li>文字列の最大サイズは、'¥0'を含まないバイト数となります。</li> <li>長さ0の文字列（'¥0'だけ）は指定できません。</li> </ul>

#### (4) EADs クライアント名として指定できるデータ

EADs クライアント名として指定できるデータ型と文字数を、AP を作成する言語ごとに次の表に示します。

表 15-4 EADs クライアント名として指定できるデータ

APを作成する言語	EADs クライアント名として指定できるデータ型	文字数	備考
Java 言語	文字列 (java.lang.String)	0~16	<ul style="list-style-type: none"> <li>半角英数字 (0~9, A~Z, a~z), アンダースコア ( _ ) が指定できます。</li> <li>空文字列を指定できます。</li> <li>null は不正となります。</li> </ul>
C 言語	'¥0'を終端記号とする文字列 (char *)	0~16	<ul style="list-style-type: none"> <li>半角英数字 (0~9, A~Z, a~z), アンダースコア ( _ ) が指定できます。</li> <li>文字列の最大サイズは、'¥0'を含まないバイト数となります。</li> <li>長さ0の文字列（'¥0'だけ）を指定できます。</li> </ul>

### 15.2.3 予約パッケージ名および予約システムプロパティ名 (Java 言語の場合)

次に示す予約パッケージ名および予約システムプロパティ名を、AP で使用しないでください。

- 予約パッケージ名  
com.hitachi.software.xeads.で始まるパッケージ名
- 予約システムプロパティ名  
eads.で始まるプロパティ名

# 16

## Developer を構築する

EADs では、ユーザファンクションおよびユーザイベントハンドラの開発をサポートするための製品として、Developer を提供しています。Developer は Windows 上で動作します。

この章では、ユーザファンクションおよびユーザイベントハンドラの開発環境である Developer の構築方法について説明します。

## 16.1 Developerに必要なリソースを確認する

---

Developerに必要なマシンスペックを次に示します。

リソース	説明
メモリ容量	1 ギガバイト以上必要です。
ディスク容量	1 ギガバイト必要です (すべてのパラメタをデフォルト値に設定した場合)。



## 16.2 Developer のインストールおよびセットアップを行う

ここでは、Developer のインストール方法について説明します。

インストールは管理者特権を持ったユーザが行います。

### 16.2.1 インストール手順

#### 注意事項

すでに別バージョンの Developer 用 EADs サーバがインストールされている場合、上書きインストールされます（ただし、03-00 以降のバージョンに限ります）。また、Developer 用 EADs サーバプロセスが起動していないことを確認してからインストールしてください。

Windows Installer を使用して Developer をインストールします。

画面の指示に従ってインストールを完了させます。

デフォルトのインストールディレクトリは、< OS のインストールドライブ >¥xeads です。

(例) OS が C ドライブにインストールされている場合

C:¥xeads¥

インストールディレクトリ名には、最大 43 文字の半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ハイフン (-), 丸括弧 (()), コロン (:), 円符号 (¥), および半角スペースが指定できます。

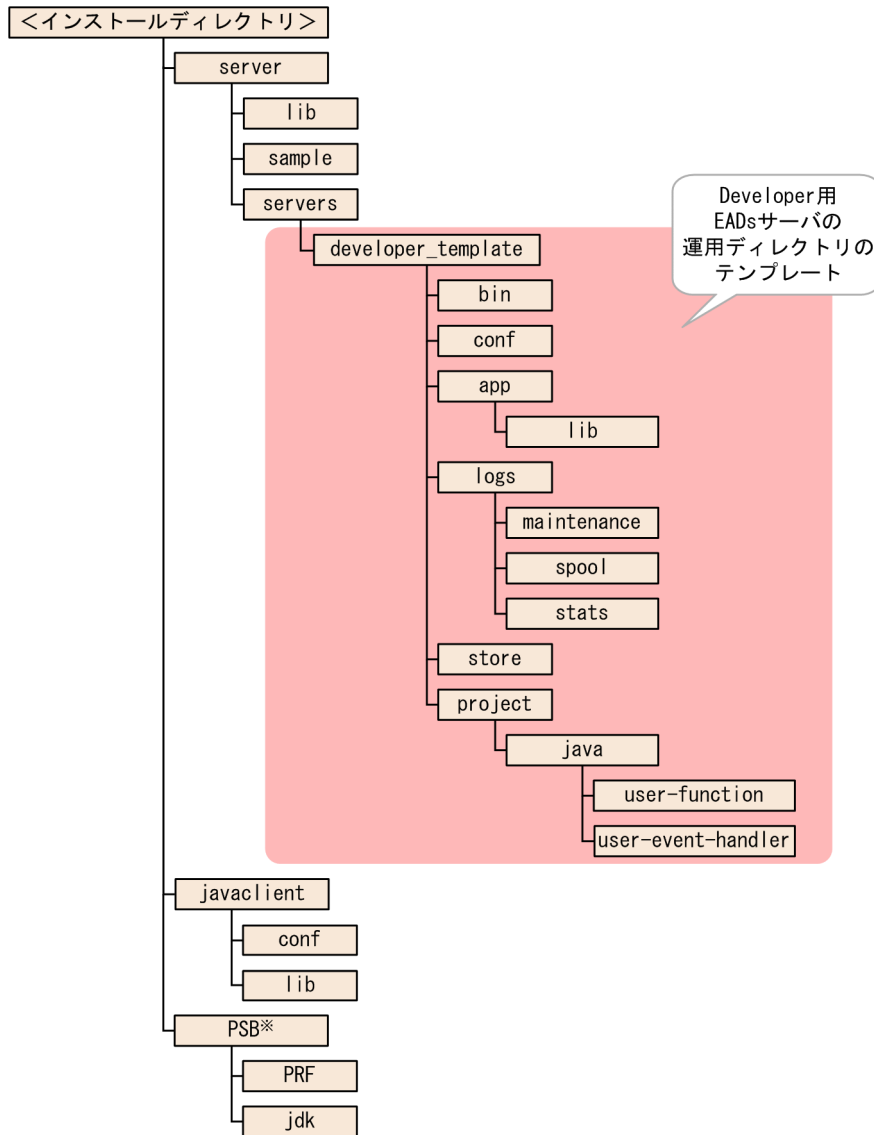
### 16.2.2 インストール後にすること

インストールしたあと、管理者特権を持ったユーザで次のことを行います。

#### (1) ディレクトリ構成を確認する

インストール後、ディレクトリ構成を確認します。

図 16-1 Developer のディレクトリ構成



注※  
PRFおよびjdk以外のディレクトリは、EADsでは使用しません。

図に示した Developer のディレクトリ構成について説明します。

ディレクトリ	説明
<インストールディレクトリ>*server *	Developer 用 EADs サーバのインストールディレクトリです。
lib	Developer 用 EADs サーバを実行するためのライブラリファイルが格納されています。 ライブラリファイルは編集できません。
sample	<code>eztool execfunc</code> コマンドで実行できるユーザファンクションのサンプルプログラムが格納されています。
servers	Developer 用 EADs サーバの運用ディレクトリを配置します。

ディレクトリ	説明
developer_template	Developer 用 EADs サーバの運用ディレクトリのテンプレートです。
bin	Developer 用 EADs サーバを運用操作するコマンドが格納されています。 コマンドは編集できません。
conf	次に示す定義ファイルが格納されています。 <ul style="list-style-type: none"> <li>• サーバ定義ファイル</li> <li>• クラスタ定義ファイル</li> <li>• 起動設定ファイル</li> <li>• コマンド定義ファイル</li> <li>• キャッシュ定義ファイル</li> </ul>
app	ファンクション定義ファイル, ユーザファンクション, およびユーザイベントハンドラの格納先です。
lib	ユーザファンクション, およびユーザイベントハンドラを実行するためのライブラリファイルの格納先です。
logs	ログファイルの出力先 (デフォルト) です。
maintenance	保守ログファイル (システムが使用するファイル) の出力先 (デフォルト) です。
spool	PRF トレースファイルの出力先 (デフォルト) です。
stats	統計情報ファイルの出力先 (デフォルト) です。
store	ストアデータファイル, キャッシュ情報ファイルおよびキャッシュインデックスファイルの出力先 (デフォルト) です。
project	Apache Ant を実行するためのビルドファイルが格納されています。
java	ユーザファンクション, およびユーザイベントハンドラを実行するためのディレクトリです。
user-function	ユーザファンクションのサンプルプログラムが格納されています。
user-event-handler	ユーザイベントハンドラのサンプルプログラムが格納されています。
<インストールディレクトリ> ¥javaclient¥	Developer 用 EADs クライアント (Java) のインストールディレクトリです。
conf	クライアント定義ファイルが格納されています。
lib	ライブラリファイルが格納されています。
<インストールディレクトリ>¥PSB¥	EADs で使用する機能の格納用ディレクトリです。
PRF	パフォーマンストレーサのインストールディレクトリです。
jdk	JDK のインストールディレクトリです。

## 16.3 Developer 用 EADs サーバをセットアップする

インストール以降の作業は管理者特権がなくても実施できます。

### 16.3.1 Developer 用 EADs サーバの運用ディレクトリを作成する

運用ディレクトリのテンプレートである<インストールディレクトリ>¥server¥servers ¥developer\_template ディレクトリを，<インストールディレクトリ>¥server¥servers ディレクトリ下に別名でコピーします。

運用ディレクトリ名には，最大 32 文字の半角英数字 (0~9, A~Z, a~z)，およびアンダースコア ( \_ ) が指定できます。

#### 注意事項

すでに存在する運用ディレクトリを使用する場合，次に示すファイルとディレクトリを別のディレクトリに移動するか，または削除してください。

- サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリ
- コマンド定義の `eads.logger.dir` パラメタに指定したディレクトリ下の全ファイルと全ディレクトリ

### 16.3.2 Developer 用 EADs サーバで使用する定義ファイルを確認する

次のディレクトリに格納されている各種定義ファイルを確認します。

<運用ディレクトリ>¥conf

特に確認が必要な点を次に示します。

- ポート番号
- ログファイルおよびストアデータファイルの出力先
- ログファイルのファイルサイズおよびファイル数

上記の項目については，デフォルト値で問題なければ定義内容を変更する必要はありません。

なお，実行環境の EADs サーバで使用する各種定義ファイルとはデフォルト値が異なります。Developer で使用する定義ファイルで設定される値については，「付録 B Developer で使用する定義ファイルで設定される値」を参照してください。

## 16.4 Developer 用 EADs クライアントをセットアップする

---

クライアント定義ファイルの編集方法について説明します。

### 16.4.1 Developer 用クライアント定義ファイルを編集する

クライアント定義ファイルを編集する手順を説明します。

1. 次を示すディレクトリ下のクライアント定義ファイルを、AP で指定したディレクトリにコピーします (ファイル名も AP で指定したファイル名に合わせます)。

<インストールディレクトリ>%javaclient%conf%eads\_sample\_client.properties

クライアント定義ファイル名は最大 32 文字で、半角英数字 (0~9, A~Z, a~z), およびアンダースコア ( \_ ) が指定できます。

2. クライアント定義ファイルを編集します。

次を示す点を確認し、定義内容を変更します。

- 通信に依存するパラメタ  
サーバ定義の設定を確認して設定します。
- ログファイルの出力先

その他の定義内容については、デフォルト値で問題なければ変更する必要はありません。

### 16.4.2 サンプルプログラムを使用する

サンプルプログラムは次のディレクトリに格納されています。次のディレクトリをコピーして使用してください。

ユーザファクションのサンプルプログラム

<インストールディレクトリ>%server%servers%developer\_template%project%java%user-function

ユーザイベントハンドラのサンプルプログラム

<インストールディレクトリ>%server%servers%developer\_template%project%java%user-event-handler

## 16.5 Developer をアンインストールする

---

### ■ 注意事項

Developer 用 EADs サーバプロセスが起動していないことを確認してからアンインストールしてください。

Developer は、次の手順でアンインストールしてください。

1. 「Windows マーク (スタート)」 - 「コントロール パネル」 を選択します。
2. 表示された画面で 「プログラムのアンインストール」 を選択します。
3. プログラムの一覧から uCosminexus Elastic Application Data store Developer を選択して、アンインストールします。

## 16.6 Developer で実行できるコマンドおよびタスク

---

セットアップが完了したら、Developer 用 EADs サーバを開始して、AP（ユーザファンクションおよびユーザイベントハンドラ）の開発やテストを行います。

Developer では、AP の開発やテストを支援するためのコマンドや、Apache Ant のタスクを用意しています。ここでは、Developer で実行できる EADs のコマンドと Developer が提供する Apache Ant のタスクについて説明します。

### 16.6.1 EADs のコマンド

Developer では、次に示すコマンドは実行できません。

- バックグラウンドでの EADs サーバの開始 (`ezstart`)
- EADs サーバの復旧 (`ezserver -r`)
- ログ、設定情報、ハードウェア情報、およびネットワーク情報の収集 (`eztool snapshot`)
- EADs サーバの縮退 (`eztool isolate`)

EADs のコマンドについては、「[14. コマンドリファレンス](#)」を参照してください。

### 16.6.2 Apache Ant のタスク

Developer では、Apache Ant を使用することによって、コマンドラインから複数の運用操作を一括して実行できます。

#### (1) Apache Ant をインストールする

Apache Ant のインストール手順について説明します。

1. Apache Ant version 1.7.1 以降をインストールします。
2. システム環境変数 `ANT_HOME` を設定します。  
システム環境変数 `ANT_HOME` に Apache Ant を展開したディレクトリ名を指定します。
3. システム環境変数 `Path` を設定します。  
システム環境変数 `Path` に「`%ANT_HOME%\bin;`」を追加します。
4. システム環境変数 `JAVA_HOME` を設定します。  
システム環境変数 `JAVA_HOME` に「`<インストールディレクトリ>%PSB%\jdk`」を指定します。

## (2) Developer が提供する Apache Ant のタスク

Developer が提供する Apache Ant のタスクを次の表に示します。

表 16-1 Developer が提供する Apache Ant のタスク一覧

項番	利用シーン	実行形式
1	Developer 用 EADs サーバを開始する	ファイルからデータを読み込む場合 ant.bat start
2	Developer 用 EADs サーバを開始する	ファイルからデータを読み込まない場合 ant.bat start_nocache
3	Developer 用 EADs サーバの状態を確認する	ant.bat status
4	キャッシュを作成する	ant.bat createcache -DcacheName=<キャッシュ名>
5	キャッシュを削除する	ant.bat deletetocache -DcacheName=<キャッシュ名>
6	ソースプログラムをコンパイルする	ユーザファンクションをコンパイルする場合 ant.bat compilefunc
7		ユーザイベントハンドラをコンパイルする場合 ant.bat compileevent
8		ユーザファンクションとユーザイベントハンドラをコンパイルする場合 ant.bat compile
9	デプロイする	ユーザファンクションをデプロイする場合 ant.bat deployfunc
10		ユーザイベントハンドラをデプロイする場合 ant.bat deployevent
11		ユーザファンクションとユーザイベントハンドラをデプロイする場合 ant.bat deploy
12	Developer 用 EADs サーバを終了する	ant.bat stop
13	コマンドの使用方法を表示する	ant.bat -p

### 16.6.3 Apache Ant のタスクの格納先

Apache Ant のタスクは、次のディレクトリに格納されています。

<運用ディレクトリ>%project

### 16.6.4 Developer 用 EADs サーバを開始する

Developer 用 EADs サーバを開始します。



Developer 用 EADs サーバを開始したあと、前回稼働時にファイルに書き出したデータを再度メモリ上に読み込ませるかどうかで、実行するタスクが異なります。

それぞれのタスクについて説明します。

## (1) ファイルからデータを読み込む場合

次のタスクを実行します。

```
ant.bat start
```

次の処理（コマンド）が順番に実行されます。

1. EADs サーバの開始 ([ezserver](#))
2. データの読み込み※ ([eztool import](#))
3. クラスタの閉塞状態解除 ([eztool open](#))
4. クラスタの状態表示（詳細） ([eztool status -v](#))
5. キャッシュの一覧表示 ([eztool listcache](#))
6. ユーザファンクションの実行可否の表示 ([eztool listfunc](#))
7. ユーザイベントハンドラの実行可否の表示 ([eztool listevent](#))

注※

存在するストアデータファイル (`eads_developer_[EADs サーバ ID].esd`) をすべて読み込みます。  
存在しない場合、エラーが発生しますが、処理は中断されません。

## (2) ファイルからデータを読み込まない場合

次のタスクを実行します。

```
ant.bat start_nocache
```

次の処理（コマンド）が順番に実行されます。

1. EADs サーバの開始 ([ezserver](#))
2. クラスタの閉塞状態解除 ([eztool open](#))
3. クラスタの状態表示（詳細） ([eztool status -v](#))
4. キャッシュの一覧の表示 ([eztool listcache](#))
5. ユーザファンクションの実行可否の表示 ([eztool listfunc](#))
6. ユーザイベントハンドラの実行可否の表示 ([eztool listevent](#))

## 16.6.5 Developer 用 EADs サーバの状態を確認する

Developer 用 EADs サーバの状態を確認する場合は、次のタスクを実行します。

```
ant.bat status
```

次の処理（コマンド）が実行されます。

- クラスタの状態表示（詳細）（[eztool status -v](#)）

## 16.6.6 キャッシュを作成する

キャッシュを作成する場合は、次のタスクを実行します。

```
ant.bat createcache -DcacheName=<キャッシュ名>
```

### 参考

キャッシュ名には、半角英数字（0～9, A～Z, a～z）、アンダースコア（\_）またはハイフン（-）が指定できます。

次の処理（コマンド）が順番に実行されます。

1. クラスタの閉塞（[eztool close](#)）
2. キャッシュの作成（[eztool createcache](#)）
3. クラスタの閉塞状態解除（[eztool open](#)）
4. キャッシュの一覧表示（[eztool listcache](#)）

## 16.6.7 キャッシュを削除する

キャッシュを削除する場合は、次のタスクを実行します。

```
ant.bat deletocache -DcacheName=<キャッシュ名>
```

### 参考

キャッシュ名には、半角英数字（0～9, A～Z, a～z）、アンダースコア（\_）またはハイフン（-）が指定できます。

次の処理（コマンド）が順番に実行されます。

1. クラスタの閉塞 (eztool close)
2. キャッシュの削除 (eztool deletocache)
3. クラスタの閉塞状態解除 (eztool open)
4. キャッシュの一覧表示 (eztool listcache)

## 16.6.8 ソースプログラムをコンパイルする

ソースプログラムをコンパイルする手順について説明します。

### (1) ソースプログラムを配置する

作成したソースプログラムを次に示すディレクトリに配置します。

#### 注意事項

Apache Ant のタスクを使用してコンパイルする場合は、ソースプログラムの文字コードは UTF-8 にしてください。

ユーザファンクション

<運用ディレクトリ>%project%java%user-function

ユーザイベントハンドラ

<運用ディレクトリ>%project%java%user-event-handler

### (2) ライブラリを設定する (任意)

コンパイルに必要なライブラリのうち、EADs が提供するライブラリは自動的に読み込まれるため、設定は不要です。EADs が提供するライブラリ以外に必要なライブラリがある場合は、次に示すディレクトリに配置します。なお、コンパイル時と実行時で同じライブラリを使用してください。

<運用ディレクトリ>%app%lib

### (3) マニフェストを設定する

マニフェストを設定する場合は、ソースプログラムを配置したディレクトリにマニフェストファイル (MANIFEST.MF) を配置します。

### (4) コンパイルする

ユーザファンクションおよびユーザイベントハンドラをコンパイルするためのタスクを実行すると、class ファイルおよび jar ファイルの作成が行われます。

同名のファイルがすでに存在する場合は上書きされます。

ソースプログラムをコンパイルするためのタスクは、コンパイルする対象ごとに異なります。

それぞれのタスクについて説明します。

### (a) ユーザファンクションをコンパイルする場合

ユーザファンクションをコンパイルする場合は、次のタスクを実行します。

```
ant.bat compilefunc
```

コンパイルされた class ファイルは、一時的に作成される次のディレクトリに出力されます。

<運用ディレクトリ>%project%temp%classes%user-function

jar ファイル (user-function.jar) は、<運用ディレクトリ>%project%temp 下に出力されます。

### (b) ユーザイベントハンドラをコンパイルする場合

ユーザイベントハンドラをコンパイルする場合は、次のタスクを実行します。

```
ant.bat compileevent
```

コンパイルされた class ファイルは、一時的に作成される次のディレクトリに出力されます。

<運用ディレクトリ>%project%temp%classes%user-event-handler

jar ファイル (user-event-handler.jar) は、<運用ディレクトリ>%project%temp 下に出力されます。

### (c) ユーザファンクションとユーザイベントハンドラをコンパイルする場合

ユーザファンクションとユーザイベントハンドラをコンパイルする場合は、次のタスクを実行します。

```
ant.bat compile
```

コンパイルされた class ファイルは、一時的に作成される次のディレクトリにそれぞれ出力されます。

ユーザファンクション

<運用ディレクトリ>%project%temp%classes%user-function

ユーザイベントハンドラ

<運用ディレクトリ>%project%temp%classes%user-event-handler

jar ファイル (user-function.jar および user-event-handler.jar) は、<運用ディレクトリ>%project%temp 下に出力されます。

## 16.6.9 ユーザファンクションまたはユーザイベントハンドラをデプロイする

ユーザファンクションおよびユーザイベントハンドラをデプロイするためのタスクを実行すると、ソースプログラムのコンパイル、jar ファイルの作成（パッケージング）および配置（デプロイ）のあと、EADs サーバの再起動まで行われます。

jar ファイルがすでに存在する場合は上書きされます。

### 参考

ソースプログラムをコンパイルするためのタスク実行時に作成されるディレクトリ（<運用ディレクトリ>%project%temp 以下のディレクトリ）は、ユーザファンクションおよびユーザイベントハンドラをデプロイするためのタスク実行時に削除されます。

ユーザファンクションおよびユーザイベントハンドラをデプロイするためのタスクは、デプロイする対象ごとに異なります。

それぞれのタスクについて説明します。

### (1) ユーザファンクションをデプロイする場合

ユーザファンクションをデプロイする場合は、次のタスクを実行します。

```
ant.bat deployfunc
```

jar ファイル (user-function.jar) は、次のディレクトリに出力されます。

<運用ディレクトリ>%app

### (2) ユーザイベントハンドラをデプロイする場合

ユーザイベントハンドラをデプロイする場合は、次のタスクを実行します。

```
ant.bat deployevent
```

jar ファイル (user-event-handler.jar) は、次のディレクトリに出力されます。

<運用ディレクトリ>%app

### (3) ユーザファンクションとユーザイベントハンドラをデプロイする場合

ユーザファンクションとユーザイベントハンドラをデプロイする場合は、次のタスクを実行します。

```
ant.bat deploy
```

jar ファイル (user-function.jar, および user-event-handler.jar) は、次のディレクトリに出力されます。

<運用ディレクトリ>%app

## 16.6.10 Developer 用 EADs サーバを終了する

Developer 用 EADs サーバを終了する場合は、次のタスクを実行します。

```
ant.bat stop
```

次の処理 (コマンド) が順番に実行されます。

1. クラスタの閉塞 (`eztool close`)
2. データの書き出し※ (`eztool export -s`)
3. EADs サーバの終了 (`eztool stop --no_export`)

注※

ストアデータファイルが存在する場合、ストアデータファイル名は次のようにリネームされます。

eads\_developer\_1.esd → backup\_eads\_developer\_1.esd

なお、backup\_eads\_developer\_1.esd がすでに存在する場合は上書きされます。

# 17

## クライアント AP を作成する (Java 言語)

この章では、Java 言語を使用したクライアント AP の作成方法について説明します。

## 17.1 ソースプログラムを作成する (Java 言語)

ここでは、キャッシュへのアクセスとデータ操作の流れと、ソースプログラムの作成例について説明します。

### 17.1.1 キャッシュへのアクセスとデータ操作の流れ

キャッシュへのアクセスとデータ操作の流れについて説明します。



#### (1) Java 言語を使用する場合のソースプログラムの例

Java 言語を使用する場合のソースプログラムの例 (key と value を格納する場合) を次に示します。

```
// EADsが提供するパッケージをインポートする
import com.hitachi.software.xeads.client.api.*;

public class PutSample {

public static void main(String[] args) {

    // EADsクライアントの初期設定をする

    final String CONFPATH = "./conf/eads_sample_client.properties";
    final String CACHENAME = "cache1";
    CacheManager cacheManager = null;
    Cache cache = null;

    try {
        cacheManager = CacheManager.create(CONFPATH);
```





複数のクラスと同時に接続するなど、設定の異なる複数の CacheManager クラスのインスタンスを同時に使用したい場合は、クライアント定義を変更して CacheManager クラスの create() を複数回実行します。複数回実行した場合は、取得した各 CacheManager クラスのインスタンスに対して CacheManager クラスの destroy() を実行して、EADs クライアントの使用を終了してください。

## (4) キャッシュへのアクセスを開始する

EADs クライアントの初期設定が完了したら、キャッシュへのアクセスを開始します。

キャッシュへのアクセスを開始するには、CacheManager クラスの getCache() を使用して、データを操作するためのインスタンス (Cache クラスのインスタンス) を生成します。

## (5) key と value を格納する

キャッシュに key と value を格納するには、Cache クラスの put() を使用します。

put() には、キャッシュに格納する key と value を指定します。

## (6) value を取得する

キャッシュから value を取得するには、Cache クラスの get() を使用します。

get() には、取得する value に関連づけられた key を指定します。

get() によって value の取得が成功すると、戻り値として key に関連づけられた value が返却されます。

value を取得する場合のソースプログラムの例を次に示します。

ソースプログラムの例 (value を取得する場合)

```
// valueを取得する
final String KEY = "key1";
try {
    String value = (String) cache.get(KEY);
    System.out.println("GET succeeded. (key = " + KEY + ", value = " + value + ")");
} catch (CacheException e) {
    int errcode = e.getErrorCode();
    System.out.println("GET failed. (key = " + KEY + ", error code = " + errcode + ")");
}
```

## (7) key と value を削除する

キャッシュから指定した key、および key に関連づけられた value を削除するには、Cache クラスの remove() を使用します。

remove() には、削除する value に関連づけられた key を指定します。

## (8) ユーザファンクションを実行する

ユーザファンクションを実行するには、`executeFunction()`を使用します。

`executeFunction()`には、ユーザファンクションを実行するグループ名または Node クラスのインスタンス、実行するユーザファンクション名、およびユーザファンクションに渡す引数を指定します。

`executeFunction()`によってユーザファンクションが実行されると、ユーザファンクションの実行結果が返却されます。

## (9) キャッシュへのアクセスを終了する

キャッシュへのアクセスを終了するには、`CacheManager` クラスの `removeCache()`を使用します。

`removeCache()`には、アクセスを終了するキャッシュ名を指定します。

## (10) EADs クライアントの使用を終了する

EADs クライアントの使用を終了するには、`CacheManager` クラスの `destroy()`を使用します。

## 17.2 クライアント AP 作成上の留意事項 (Java 言語)

---

クライアント AP 作成上の留意事項について説明します。

### 17.2.1 EADs クライアントの初期設定に関する留意事項

EADs クライアントの初期設定に関する留意事項を次に示します。

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- EADs クライアント名の指定を省略して初期設定をした場合は、EADs クライアント名に空文字列を指定して初期設定をした場合と同じ動作をします。EADs クライアント名とログ出力先の関係については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、次に示す方法などで、AP 開始時に EADs クライアントの初期設定をしてください。
  - ServletContextListener を使用する。
  - Servlet の init メソッドを使用する。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、CacheManager クラスの create() の実行後、CacheManager クラスの destroy() を実行しないとメモリリークとなるので必ず実行してください。
- クライアント定義を変更する場合は、次の順に行います。
  1. CacheManager クラスの destroy() で、EADs クライアントの使用を終了します。
  2. クライアント定義ファイルを更新します。
  3. CacheManager クラスの create() で、再度 EADs クライアントの初期設定をします。

### 17.2.2 キャッシュへのアクセス開始に関する留意事項

キャッシュ名には、あらかじめ `eztool createcache` コマンドで作成した名称を指定します。指定したキャッシュ名が存在しない場合は、`CacheException.EAD_ERROR_NET_CLUSTERINFO` が返却されます。

### 17.2.3 データ操作に関する留意事項

データ操作に関する留意事項を次に示します。

- 指定した key がすでにキャッシュに格納されている場合、put()は無条件に value を更新します。無条件に value を更新したくない場合は、次のメソッドを使用してください。
  - create()
 

新規に key を格納する場合だけ、value を key に関連づけて格納します。
  - update()
 

指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します。
  - replace()
 

指定した key に関連づけられている value を、条件として指定した value (comparativeValue) と比較します。value が一致している場合だけ、value を key に関連づけて格納します。
- get()の実行時に指定した key がキャッシュに存在しない場合は、null が返却されます。
- replace()の実行時に指定した key が存在しない場合は、CacheException.EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST が返却されます。
- replace()の実行時に比較結果が一致しなかった場合は CacheException.EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED が返却されます。
- create()の実行時に指定した key がすでに存在している場合は、CacheException.EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST が返却されます。
- update()の実行時に指定した key が存在しない場合は、CacheException.EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST が返却されます。

## 17.2.4 キャッシュへのアクセス終了に関する留意事項

キャッシュへのアクセス終了に関する留意事項を次に示します。

- 一度、CacheManager クラスの removeCache()でアクセスを終了したキャッシュに再度アクセスする場合は、CacheManager クラスの getCache()を発行してください。
- CacheManager クラスの removeCache()は同じキャッシュ名を指定して取得したすべてのキャッシュ (Cache クラスのインスタンス) へのアクセスを終了するため、複数スレッドで実行している場合などは注意してください。

## 17.2.5 EADs クライアントの使用終了に関する留意事項

EADs クライアントの使用終了に関する留意事項を次に示します。

- CacheManager クラスの destroy() は、CacheManager クラスの create() と対になるように実行してください。CacheManager クラスの create() で取得したインスタンスにつき 1 度だけ実行してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、CacheManager クラスの destroy() を実行しないとメモリリークとなるので必ず実行してください。
- CacheManager クラスの destroy() を実行して EADs クライアントがすでに終了している場合は、CacheException が返却されます。

## 17.3 ソースプログラムをコンパイルする (Java 言語)

---

作成したソースプログラムは、JDK の javac コマンドを使用してコンパイルします。

javac コマンドの詳細については、Java 言語のコンパイラに関するドキュメントを参照してください。

コンパイルに必要なとなるクラスライブラリを次に示します。

- /opt/hitachi/xeads/javaclient/lib/eads-client.jar
- /opt/hitachi/xeads/javaclient/lib/eads-common.jar
- /opt/hitachi/xeads/javaclient/lib/hntrlib2-eads-j.jar

なお、J2EE サーバ (uCosminexus Application Server) 上で実行する場合は、次に示すライブラリを AP (WAR ファイル (WEB-INF/lib ディレクトリ)) に取り込む必要があります。\*

- /opt/hitachi/xeads/javaclient/lib/eads-client.jar
- /opt/hitachi/xeads/javaclient/lib/eads-common.jar
- /opt/hitachi/xeads/javaclient/lib/hntrlib2-eads-j.jar

注※

Servlet または JSP から実行できます (Filter, Listener を含みます)。

# 18

## ユーザファンクションを作成する

この章では、ユーザファンクションの作成から実行までの方法について説明します。



## 18.1 ユーザファンクションを作成する前に知っておくべき前提条件

ユーザファンクションを使用する場合は、key をグルーピングすることが前提となります。

あらかじめ key をグルーピングして、処理対象とするデータが特定の EADs サーバに配置されるようにしてください。

key のグルーピングについては、「[2.6 特定の EADs サーバへのデータ配置 \(key のグルーピング\)](#)」を参照してください。

そのほかに、ユーザファンクションを作成する前に知っておくべき前提条件について説明します。

ユーザファンクションの概要については、「[2.7 ユーザファンクションによる効率的なデータ処理](#)」を参照してください。

### 18.1.1 ユーザファンクションの作成言語

ユーザファンクションは Java 言語で作成できます。

ユーザファンクションは、EADs が提供する Function インタフェースを実装し、jar ファイルにアーカイブする必要があります。

#### 注意事項

EADs サーバより新しいバージョンの Function インタフェースで実装されたユーザファンクションを使用した場合、動作は保証されません。この場合、ソースプログラムの変更や再コンパイルが必要です。

なお、03-00 以降のバージョンの Function インタフェースで実装されたユーザファンクションは、新しいバージョンの EADs サーバでも使用できます。

なお、1 つの EADs サーバに複数の jar ファイルを配置することができます。そのため、1 つの EADs サーバで複数のユーザファンクションを実行したい場合、用途に合わせて jar ファイルを分割することができます。

### 18.1.2 ユーザファンクションの実行方法の種類

ユーザファンクションには次の 2 種類の実行方法があります。

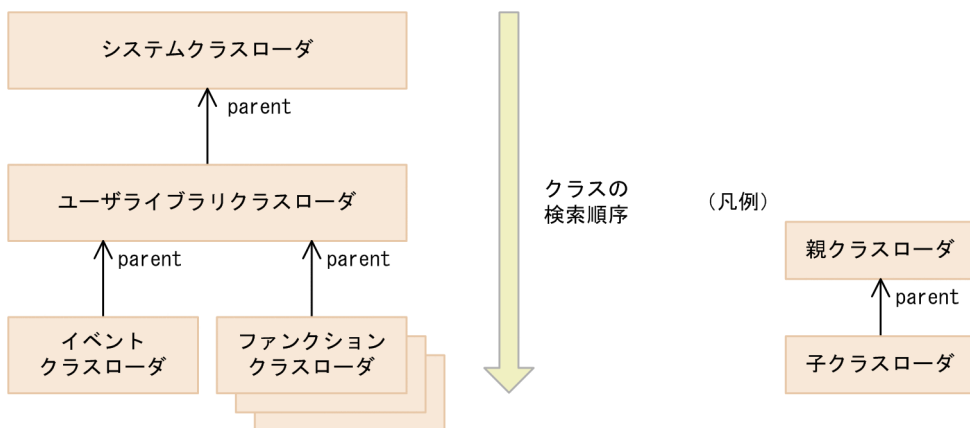
- グループ指定による実行方法
- EADs サーバ指定による実行方法

それぞれのユーザ機能の仕組みについては、「2.7.2 ユーザ機能の仕組み」を参照してください。

### 18.1.3 EADs サーバが使用する Java クラスローダ

EADs サーバで使用する Java クラスローダについて説明します。

Java クラスローダとは、Java クラスを Java 仮想マシンに動的にロードする役割を持ちます。EADs サーバが使用する Java クラスローダの構成を次の図に示します。



- システムクラスローダ  
jdk ライブラリと EADs サーバで使用するクラスをロードします。
- ユーザライブラリクラスローダ  
＜運用ディレクトリ＞/app/lib 下に配置したユーザ機能およびユーザイベントハンドラ共通で使用するライブラリのクラスをロードします。  
このクラスローダは、EADs サーバで 1 つ作成されます。
- ファンクションクラスローダ  
ユーザ機能の jar ファイルに含まれるクラスと、jar ファイルのマニフェストファイルに記載されたライブラリに含まれるクラス（ユーザ機能だけで使用するクラス）をロードします。  
このクラスローダは、jar ファイルごとに作成されます。そのため、ユーザ機能の jar ファイルを複数配置した場合、クラスローダも複数作成されます。
- イベントクラスローダ  
ユーザイベントハンドラの jar ファイルに含まれるクラスと、jar ファイルのマニフェストファイルに記載されたライブラリに含まれるクラス（ユーザイベントハンドラだけで使用するクラス）をロードします。  
このクラスローダは、jar ファイルごとに作成されます。

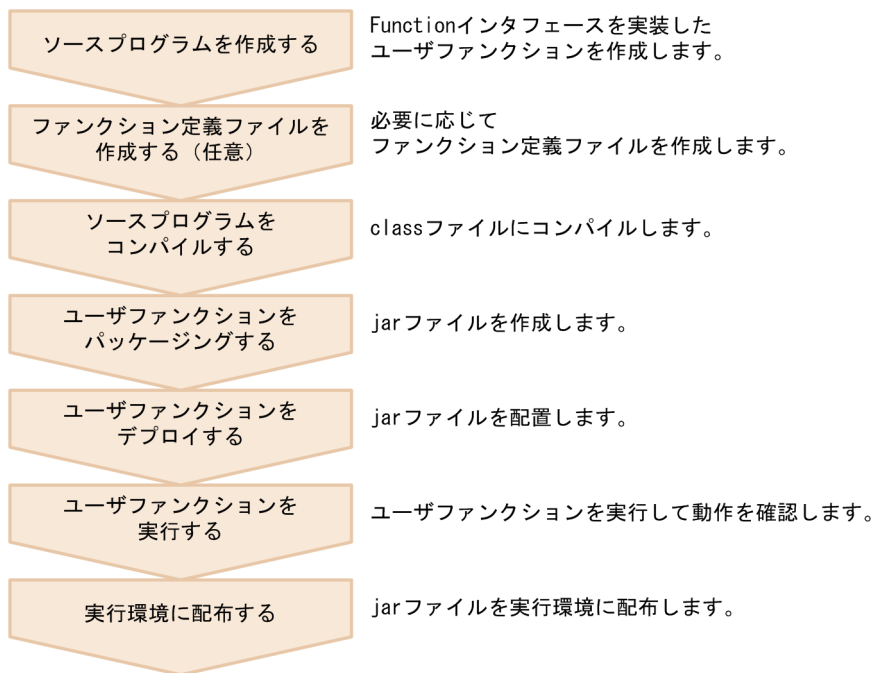
各クラスローダのクラス検索は、親クラスローダであるシステムクラスローダから行います。

ユーザ機能のロード順序は、jar ファイルの ASCII コード順です。

異なるファンクションクラスローダに同じ完全修飾クラス名のクラスをロードできますが、ユーザファンクションとして実行できるのは先にロードされたユーザファンクションとなります。あとからロードされたユーザファンクションについては、インスタンスの生成および Function インターフェースの `init()` は実行されません。

## 18.1.4 ユーザファンクションの作成の流れ

ユーザファンクションの作成の流れを次の図に示します。



## 18.2 ソースプログラムを作成する（ユーザファンクション）

ユーザファンクションの作成方法と、作成上の留意事項について説明します。

### 18.2.1 ユーザファンクションの流れ

ユーザファンクションの流れを次の図に示します。



#### (1) ユーザファンクションのサンプルプログラム

Developer を使用している場合、ユーザファンクションのサンプルプログラムは次のディレクトリに格納されています。

<運用ディレクトリ>%project%java%user-function

ユーザファンクションのサンプルプログラムを次に示します。

```
// EADsが提供するパッケージをインポートする
import com.hitachi.software.xeads.func.Function;
import com.hitachi.software.xeads.func.FunctionContext;

public class FunctionSample implements Function {

    // ユーザファンクションの初期化処理を実装する
    public void init(FunctionContext context) {
        context.getLogger().log("init function at " + context.getServerName());
    }

    // ユーザファンクションの処理を実装する
    public Object execute(FunctionContext context) {
        context.getLogger().log("execute function at " + context.getServerName() + ", group
name=" + context.getGroupName());
        return null;
    }

    // ユーザファンクションの終了処理を実装する
    public void destroy(FunctionContext context) {
        context.getLogger().log("destroy function at " + context.getServerName());
    }
}
```

```
}  
  
}
```

## (2) EADs が提供するパッケージをインポートする

EADs が提供する次のパッケージをインポートします。

```
import com.hitachi.software.xeads.func.Function;  
  
import com.hitachi.software.xeads.func.FunctionContext;
```

## (3) ユーザ関クションの初期化処理を実装する

Function インタフェースの `init()` を使用して、EADs サーバの開始時に呼び出される、ユーザ関クションの初期化処理を実装します。

`init()` の引数 `FunctionContext` を通じて、EADs サーバの情報を取得できます。ただし、`init()` は EADs サーバ起動時に呼び出されるため、キャッシュのデータは取得できません。

なお、`init()` の実行時に例外が発生した場合、このユーザ関クションは使用できません。

## (4) ユーザ関クションの処理を実装する

Function インタフェースの `execute()` を使用して、EADs クライアントからの要求時に EADs サーバで実行するユーザ関クションの処理を実装します。

`execute()` の引数 `FunctionContext` を通じて、EADs サーバの情報を取得したり、キャッシュのデータを操作したりできます。

実行できる操作を次に示します。

- ユーザ関クションの実行に必要な情報の取得 (`FunctionContext` インタフェース)
- ファンクション定義の情報の取得 (`InitConfig` インタフェース)
- EADs クライアントの情報の取得 (`ClientInfo` インタフェース)
- EADs サーバの情報の取得 (`ServerInfo` インタフェース)
- データ操作 (`Store` インタフェース)
- ユーザログの出力 (`UserLogger` インタフェース)

## (5) ユーザ関クションの終了処理を実装する

Function インタフェースの `destroy()` を使用して、EADs サーバの終了時に呼び出される、ユーザ関クションの終了処理を実装します。

`destroy()` の引数 `FunctionContext` を通じて、EADs サーバの情報を取得できます。

なお、ユーザ関クションの初期化処理に失敗した場合、destroy()は呼び出されません。

## 18.3 ユーザファンクション作成上の留意事項

---

ユーザファンクション作成上の留意事項について説明します。

### 18.3.1 jar ファイル名に関する留意事項

jar ファイル名に関する留意事項を次に示します。

- eads から始まるファイル名は使用しないでください。
- ファイル名は次の形式にしてください。  
<任意の文字列>\_function.jar  
任意の文字列には、半角英数字 (0~9, A~Z, a~z), およびアンダースコア ( \_ ) が指定できます。

### 18.3.2 パッケージ名に関する留意事項

パッケージ名に関する留意事項を次に示します。

- com.hitachi.software.xeads から始まるパッケージ名は使用しないでください。
- ユーザファンクションのパッケージ名およびクラス名には、半角英数字 (0~9, A~Z, a~z), およびアンダースコア ( \_ ) が指定できます。

### 18.3.3 ユーザファンクションの実装に関する留意事項

ユーザファンクションの実装に関する留意事項を次に示します。

- ユーザファンクションのデフォルトコンストラクタのアクセス修飾子は public にして、EADs サーバから使用できるようにしてください。
- EADs クライアントからの引数と戻り値は、シリアライズ可能なオブジェクトにしてください。
- Group インタフェースを使用してデータを取得する場合、次のどちらかの方法で、オブジェクトをデシリアライズするために十分な内容をクラスパスに追加してください。
  - jar ファイルのマニフェストの Class-Path 属性に jar ファイルのパスを記載する。
  - <運用ディレクトリ>/app/lib ディレクトリ下に jar ファイルを配置する。
- ユーザファンクションのインスタンスは単一であり、複数の EADs クライアントで同時に実行すると、EADs サーバではマルチスレッドで同一のインスタンスを使用します。そのため、ユーザファンクションには次のような処理を実装しないでください。
  - 排他制御をしないで、インスタンス変数、static 変数を更新する。
  - 排他制御をしないで、スレッドセーフでない API を使用する。

- FunctionContext と、そこから取得できるオブジェクトの有効範囲は、Function インタフェースのメソッドの中だけです。これらのオブジェクトを、メソッドの範囲外で参照した場合の動作は保証されません。
- ユーザファンクション内でスレッドを生成しないでください。スレッドを生成・実行した場合の動作は保証されません。
- ユーザファンクションは、EADs サーバプロセスで動作するため、カレントディレクトリは運用ディレクトリになります。

### 18.3.4 効率的に key の一覧を取得する方法

あるグループの key を ASCII コードの昇順に格納し、ユーザファンクションを使用して非同期に key の一覧を取得する場合、Group インタフェースの `keyIterator()` と `higherKeyIterator()` を組み合わせると、1 度取得した key を、2 度目以降に取得する際に対象外にできます（2 度目以降は、未取得の key の一覧だけを取得できます）。

これによって、データが追加されるたびに、すべての key の一覧を取得し直すことなく、効率的に key の一覧を取得できます。



1. Groupインタフェースの `keyIterator()` で keyの一覧を取得します。

keyの一覧取得

key	value
time:1201	value1
time:1202	value2
time:1203	value3

2. 1. で最後に処理したkey (time:1203) を覚えておくために valueとして挿入 (put) します。

key	value
time:!	time:1203
time:1201	value1
time:1202	value2
time:1203	value3

3. 新規にデータが追加 (put) されます。

key	value
time:!	time:1203
time:1201	value1
time:1202	value2
time:1203	value3
time:1204	value4
time:1205	value5
time:1206	value6

4. 「time:!」に関連づけられたvalue (1. で最後に処理したkey) を取得 (get) します。

key	value
time:!	time:1203
time:1201	value1
time:1202	value2
time:1203	value3
time:1204	value4
time:1205	value5
time:1206	value6

5. Groupインタフェースの `higherKeyIterator()` の引数にtime:1203を指定して それよりも大きい値 (time:1204) のkey からkeyの一覧を取得します。

keyの一覧取得

key	value
time:!	time:1203
time:1201	value1
time:1202	value2
time:1203	value3
time:1204	value4
time:1205	value5
time:1206	value6

Group インタフェースについては、「20.2.7 Group インタフェース」を参照してください。

key の一覧を ASCII コードの降順で取得する場合は、Group インタフェースの `descendingKeyIterator()` と `lowerDescendingKeyIterator()` を組み合わせて、同様のことができます。

## ポイント

key の格納順序に対応するシーケンス (日付や時刻など) を key として設定すると、効率的に一覧を取得できます。

## 18.4 ファンクション定義ファイルを作成する（任意）

必要に応じて、ファンクション定義ファイルを作成します。

ファンクション定義ファイルには 2 種類あります。

### 全ユーザファンクションに適用するファンクション定義ファイルを作成する場合

ファイル名は `eads_function.properties` です。

ファンクション定義ファイルを配置しなかった場合、実行に必要なパラメタにはデフォルト値が使用されます。

ファンクション定義のパラメタについては、「[9.2.3 スレッドプールおよびコネクションプールに関連するパラメタ](#)」を参照してください。

### ユーザファンクションの jar ファイルごとにファンクション定義ファイルを作成する場合

ファイル名は次の形式です。

<任意の文字列>\_function.properties

任意の文字列には、適用したい jar ファイルの「\_function.jar」を除いた文字列を指定してください。

(例) jar ファイル名が `sample_function.jar` の場合

ファンクション定義ファイル名は `sample_function.properties` となります。

`eads_function.properties` と <任意の文字列>\_function.properties は同時に使用できますが、両定義ファイルに同じパラメタが定義されていた場合、<任意の文字列>\_function.properties の定義内容が適用されます。

ファンクション定義ファイルを配置しなかった場合、`eads_function.properties` の定義内容が適用されます。

#### 注意事項

03-00 以前のユーザファンクションの jar ファイル名 (`user-function.jar`) の場合、ファンクション定義ファイル名は `user_function.properties` となります。

ファンクション定義には、`eads.function.<ユーザファンクション名>.max_execute_threads` パラメタを指定して、ユーザファンクション単位の最大同時実行スレッド数を設定できます。

また、ユーザ独自のパラメタを定義することもできます。

ユーザファンクション単位の最大同時実行スレッド数の設計については、「[9.2.2 最大同時実行スレッド数の設定](#)」を参照してください。

ファンクション定義に指定したパラメタは、InitConfig インタフェースを使用することで、ユーザファンクション内で取得できます。

ファンクション定義ファイルを作成する手順を説明します。

#### 1. ファンクション定義ファイルを編集します。

## 参考

Developer を使用している場合、ファンクション定義ファイル (eads\_function.properties) は次のディレクトリに格納されています。

```
<運用ディレクトリ>¥project¥java¥user-function
```

2. 次のディレクトリにファンクション定義ファイルを格納します。

```
<運用ディレクトリ>/app
```

## 参考

Developer を使用している場合は、次のディレクトリになります。

```
<運用ディレクトリ>¥app
```

ファンクション定義ファイルの例を次に示します。

```
eads.function.com.abc.def.outputGroupKeys.max_execute_threads=10
eads.function.com.abc.def.getSpectialKey.max_execute_threads=5
eads.function.com.abc.def.updateInstance.max_execute_threads=2
```

```
# ユーザ独自のパラメタ
com.FunctionSample.def.version=0100
com.FunctionSample.def.isUpdateEnabled=true
```

## 注意事項

ユーザ独自のパラメタを定義する場合、eads.で始まるパラメタ名は使用できません。使用した場合の動作は保証されません。

## 18.5 ソースプログラムをコンパイルする (ユーザファンクション)

ユーザファンクションをコンパイルします。

なお、Developer では、EADs が提供する Apache Ant のタスクを使用して、コンパイルすることもできます。手順については、「[16.6.8 ソースプログラムをコンパイルする](#)」を参照してください。

### 18.5.1 ユーザファンクションをコンパイルする手順

ユーザファンクションをコンパイルする手順について説明します。

#### (1) ソースプログラムを配置する

作成したユーザファンクションを任意の場所に配置します。

#### (2) コンパイルする

作成したユーザファンクションのソースプログラムは、次に示すディレクトリに格納されている JDK の javac コマンドを使用してコンパイルします。

javac コマンドの詳細については、Java 言語のコンパイラに関するドキュメントを参照してください。

```
/opt/hitachi/xeads/PSB/jdk
```

#### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<インストールディレクトリ>%PSB%jdk
```

次に示すライブラリをクラスパスに設定します。

```
/opt/hitachi/xeads/server/lib/eads-function.jar
```

#### 参考

Developer を使用している場合は、次のライブラリを設定します。

```
<インストールディレクトリ>%server%lib%eads-function.jar
```

## 18.6 ユーザファンクションをパッケージングする

「18.5 ソースプログラムをコンパイルする (ユーザファンクション)」でコンパイルした class ファイルを格納する jar ファイルを作成します。

なお、Developer では、EADs が提供する Apache Ant のタスクを使用して、jar ファイルを作成することもできます。手順については、「16.6.9 ユーザファンクションまたはユーザイベントハンドラをデプロイする」を参照してください。

### 18.6.1 ユーザファンクションをパッケージングする手順

「18.5 ソースプログラムをコンパイルする (ユーザファンクション)」でコンパイルした class ファイルを格納する jar ファイルを作成する手順について説明します。

#### (1) マニフェストを設定する (任意)

ユーザファンクションだけで使用するクラスは、jar ファイルのマニフェストファイル (MANIFEST.MF) の Class-Path 属性に記載します。

##### 参考

Developer を使用している場合、マニフェストファイルは次のディレクトリに格納されています。

```
<運用ディレクトリ>%project%java%user-function
```

#### (2) jar ファイルを作成する

次に示すディレクトリに格納されている JDK の jar コマンドを使用して jar ファイルを作成します。

jar コマンドの詳細については、Java 言語に関するドキュメントを参照してください。

```
/opt/hitachi/xeads/PSB/jdk
```

##### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<インストールディレクトリ>%PSB%jdk
```

## 18.7 ユーザファンクションをデプロイする

ユーザファンクションをデプロイします。

ユーザファンクションのデプロイは EADs サーバが停止した状態で行います。

なお、Developer では、EADs が提供する Apache Ant のタスクを使用して、デプロイすることもできます。手順については、「[16.6.9 ユーザファンクションまたはユーザイベントハンドラをデプロイする](#)」を参照してください。

### 18.7.1 ユーザファンクションをデプロイする手順

ユーザファンクションをデプロイする手順について説明します。

#### (1) EADs サーバを終了する

`eztool stop` コマンドを実行して、EADs サーバを終了します。

Developer を使用している場合は、Apache Ant のタスク (`ant.bat stop`) を実行して、Developer 用 EADs サーバを終了することもできます。

#### (2) 作成した jar ファイルをコピーする

「[18.6.1\(2\) jar ファイルを作成する](#)」で作成した jar ファイルを、次に示すディレクトリにコピーします。

<運用ディレクトリ>/app

##### 参考

Developer を使用している場合は、次のディレクトリになります。

<運用ディレクトリ>%app

#### (3) ファンクション定義ファイルをコピーする (任意)

ファンクション定義ファイルを作成した場合、次に示すディレクトリにコピーします。

<運用ディレクトリ>/app

##### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<運用ディレクトリ>¥app
```

## (4) ライブラリを配置する (任意)

必要に応じて、ユーザファンクションおよびユーザイベントハンドラ共通で使用するライブラリを次に示すディレクトリに配置します。拡張子が jar であるライブラリだけが有効です。

```
<運用ディレクトリ>/app/lib
```

### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<運用ディレクトリ>¥app¥lib
```

jar ファイルのマニフェストファイル (MANIFEST.MF) の Class-Path 属性に jar ファイルのパスを記載している場合、そのパスに jar ファイルをコピーしてください。

## (5) EADs サーバを開始する

`ezserver` コマンドを実行し、EADs サーバを開始します。

Developer を使用している場合は、Apache Ant のタスク (`ant.bat start` または `ant.bat start_nocache`) を実行し、Developer 用 EADs サーバを開始することもできます。

## (6) デプロイされたかどうか確認する (任意)

`eztool listfunc` コマンドを使用して、正しくデプロイされたかどうか確認します。

手順については、「11.8 ユーザファンクションまたはユーザイベントハンドラが、各 EADs サーバに正しく配置されているかどうか、また、実行できるかどうか確認したい」を参照してください。

## 18.8 ユーザファンクションを実行する

---

ユーザファンクションを実行して動作を確認します。

### 18.8.1 ユーザファンクションの呼び出し

ユーザファンクションは、EADs クライアントの次のメソッドを使用して呼び出します。

#### (1) グループを指定して実行するユーザファンクションの場合

Java 言語の場合

Cache クラスの `executeFunction()` (グループ指定)

C 言語の場合

`ead_execute_function()`

#### (2) EADs サーバを指定して実行するユーザファンクションの場合

Java 言語の場合

Cache クラスの `executeFunction()` (EADs サーバ指定)

C 言語の場合

`ead_execute_node_function()`

### 18.8.2 ユーザログへの出力

UserLogger インタフェースを使用して、ユーザファンクションの実行情報など (ユーザログ) を出力できます。ユーザログは、サーバ定義の `eads.logger.dir` パラメータに指定したディレクトリに、`eads_user_message[n].log` ([n]はファイル数の通番) というファイル名で出力されます。

### 18.8.3 ユーザファンクション運用上の留意事項

- `eztool forcestop` コマンドや OS の機能などによって EADs サーバプロセスが強制終了された場合、Function インタフェースの `destroy` メソッドは呼び出されません。
- EADs サーバ間でのユーザファンクションの整合性は、ユーザが確保する必要があります。特定の EADs サーバにユーザファンクションが存在しない状態で、ユーザファンクションを実行しようとする、EADs クライアントにはエラーが返されます。
- ユーザファンクションの実装が EADs サーバごとに異なっても、EADs サーバはユーザファンクションを実行します。



- ユーザ機能は EADs クライアントで指定されたキャッシュにアクセスするため、異なるキャッシュにアクセスするには、EADs クライアント側でキャッシュを切り替える必要があります。

## 18.9 実行環境に配布する

---

<運用ディレクトリ>/app ディレクトリを、実行環境の EADs サーバの<運用ディレクトリ>/app ディレクトリにコピーします。

### 参考

Developer を使用している場合は、次のディレクトリになります。

<運用ディレクトリ>¥app

# 19

## ユーザイベントハンドラを作成する

この章では、ユーザイベントハンドラの作成から実行までの方法について説明します。

## 19.1 ユーザイベントハンドラを作成する前に知っておくべき前提条件

---

ユーザイベントハンドラを作成する前に知っておくべき前提条件について説明します。

ユーザイベントハンドラの概要については、「[2.8 データ操作を契機とするユーザ処理の実行（ユーザイベントハンドラ）](#)」を参照してください。

なお、EADs サーバが使用する Java クラスローダについては、「[18.1.3 EADs サーバが使用する Java クラスローダ](#)」を参照してください。

### 19.1.1 ユーザイベントハンドラの作成言語

ユーザイベントハンドラは Java 言語で作成できます。

ユーザイベントハンドラは、EADs が提供する EventHandler インタフェースを実装し、jar ファイル (user-event-handler.jar) にアーカイブする必要があります。

### 19.1.2 ユーザイベントハンドラが動作するケース、動作しないケース

ユーザイベントハンドラが動作するケース、動作しないケースについて説明します。

#### (1) ユーザイベントハンドラが動作するケース

ユーザイベントハンドラで設定した条件を満たし、かつデータの更新操作を行った場合に、ユーザイベントハンドラが動作します。

データの更新操作とは、次の API またはコマンドのことです。

- API (put, create, update, replace, remove) ※
- `eztool put` コマンド
- `eztool remove` コマンド

注※ ユーザファンクション内で実行される API を含みます。

#### (2) ユーザイベントハンドラが動作しないケース

次の場合はユーザイベントハンドラが動作しません。

- データの参照操作を行った場合  
データの参照操作とは、次の API またはコマンドのことです。
  - API (get) ※

- `eztool get` コマンド

注※ ユーザファンクション内で実行される API を含みます。

- データ操作で例外が発生した場合
- `create`, `update`, `replace` の処理で、データが更新されなかった場合

### 19.1.3 ユーザイベントハンドラが動作する条件の限定

ユーザイベントハンドラが動作する条件を、フィルタを掛けることで限定できます。

フィルタには次の 2 種類があります。なお、2 種類のフィルタを同時に指定できます。

- オペレーションタイプフィルタ
- データフィルタ

#### (1) オペレーションタイプフィルタ

データ操作の種類に応じてユーザイベントハンドラが動作するかどうかを決定します。

- データの格納処理  
該当するデータの格納処理を次に示します。
  - Cache クラス、または Group インタフェースのメソッド (`put()`, `replace()`, `create()`, `update()`)
  - `eztool put` コマンド
- データの削除処理  
該当するデータの削除処理を次に示します。
  - Cache クラス、または Group インタフェースのメソッド (`remove()`)
  - `eztool remove` コマンド

オペレーションタイプフィルタは、`EventHandler` インタフェースを実装したクラスに `OperationFilter` アノテーションを付けることで設定します。

#### (2) データフィルタ

操作が行われたデータのキャッシュ名、グループ名、および要素名に応じてユーザイベントハンドラが動作するかどうかを決定します。

データフィルタは、`EventHandler` インタフェースを実装したクラスに `DataFilter` アノテーションを付けることで設定します。

データフィルタが無効の場合、ユーザイベントハンドラは動作しません。

- キャッシュ名のフィルタリング

指定したキャッシュ名と完全一致した場合にユーザイベントハンドラが動作します。

- **グループ名のフィルタリング**

指定したグループ名が階層単位で完全一致した場合にユーザイベントハンドラが動作します。

- **要素名のフィルタリング**

指定した要素名と前方一致した場合にユーザイベントハンドラが動作します。

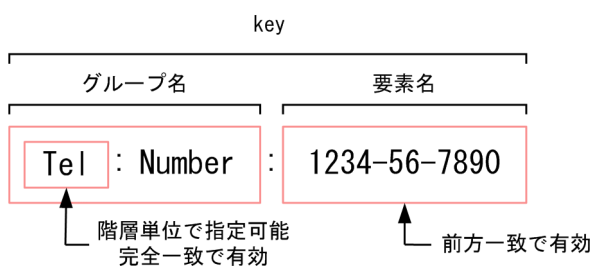
キャッシュ名、グループ名、および要素名のデータフィルタを組み合わせ、フィルタを掛けることができます。

## ポイント

例えば、キャッシュ名のフィルタリングとグループ名のフィルタリングの両方でフィルタを掛ける場合、それぞれのデータフィルタが有効になるときだけ、ユーザイベントハンドラが動作します。

### (3) グループ名および要素名のデータフィルタが有効となる指定範囲の例

グループ名および要素名のデータフィルタが有効となる指定範囲の例を、次の図に示します。



グループ名のデータフィルタは、指定したグループ名が階層単位で完全一致した場合に有効となります。この図の場合、グループ名のデータフィルタは「Tel:Number」または「Tel」と指定すると有効となります。なお、「Tel:Number:1234-56-7890」のように、グループ名以外（要素名など）を指定すると無効となります。

要素名のデータフィルタは、指定した要素名と前方一致した場合に有効となります。この図の場合、「12」や「1234-56-7890」のように指定すると有効となります。なお、「Tel」や「7890」のように、要素名以外（グループ名など）を指定したり、前方一致していなかったりすると無効となります。

## 19.1.4 ユーザイベントハンドラの優先度の設定

EADs サーバ上で複数のユーザイベントハンドラが同じ条件で動作する場合、ユーザイベントハンドラの実行順序について、ユーザイベントハンドラ間で優先度を設定できます。優先度は高いものから順に Highest, High, Normal, Low, Lowest の 5 段階で設定できます（デフォルトは Normal）。

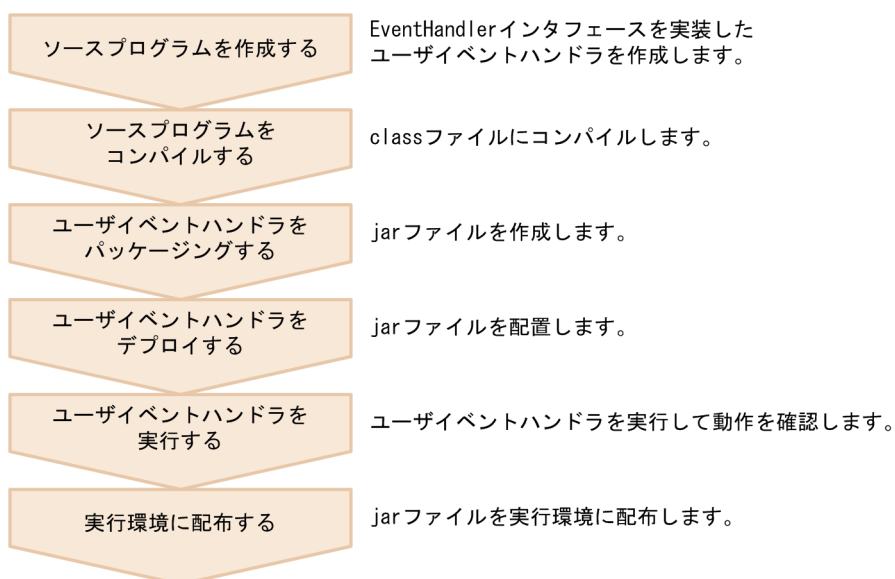
ユーザイベントハンドラの優先度は、EventHandler インタフェースを実装したクラスに Priority アノテーションを付けることで設定します。

## ■ 注意事項

同じ優先度のユーザイベントハンドラが複数存在する場合、同一優先度内での実行順序は保証されません。

## 19.1.5 ユーザイベントハンドラの作成の流れ

ユーザイベントハンドラの作成の流れを次の図に示します。



## 19.2 ソースプログラムを作成する（ユーザイベントハンドラ）

ユーザイベントハンドラの作成方法と、作成上の留意事項について説明します。

### 19.2.1 ユーザイベントハンドラの流れ

ユーザイベントハンドラの流れを次の図に示します。



#### (1) ユーザイベントハンドラのサンプルプログラム

Developer を使用している場合、ユーザイベントハンドラのサンプルプログラムは次のディレクトリに格納されています。

<運用ディレクトリ>%project%java%user-event-handler

ユーザイベントハンドラのサンプルプログラムを次に示します。

```
// EADsが提供するパッケージをインポートする
import com.hitachi.software.xeads.event.EventHandler;
import com.hitachi.software.xeads.event.EventContext;

public class EventHandlerSample implements EventHandler {

    // ユーザイベントハンドラの初期化処理を実装する
    public void init(EventContext context) {
        context.getLogger().log("init event at " + context.getServerName());
    }

    // ユーザイベントハンドラの処理を実装する
    public void execute(EventContext context) {
        context.getLogger().log("execute event at " + context.getServerName() + ",key=" +
context.getKey());
    }

    // ユーザイベントハンドラの終了処理を実装する
    public void destroy(EventContext context) {
        context.getLogger().log("destroy event at " + context.getServerName());
    }
}
```



```
}
```

## (2) EADs が提供するパッケージをインポートする

EADs が提供する次のパッケージをインポートします。

```
import com.hitachi.software.xeads.event.EventHandler;  
  
import com.hitachi.software.xeads.event.EventContext;
```

## (3) ユーザイベントハンドラが動作する条件を指定する

ユーザイベントハンドラが動作する条件を指定する場合は、アノテーションを付けます。

オペレーションタイプフィルタは、EventHandler インタフェースを実装したクラスに OperationFilter アノテーションを付けることで設定します。

データフィルタは、EventHandler インタフェースを実装したクラスに DataFilter アノテーションを付けることで設定します。

## (4) ユーザイベントハンドラの初期化処理を実装する

EventHandler インタフェースの `init()` を使用して、EADs サーバの開始時に呼び出される、ユーザイベントハンドラの初期化処理を実装します。

`init()` の引数 `EventContext` を通じて、EADs サーバの情報を取得できます。

なお、`init()` の実行時に例外が発生した場合、このユーザイベントハンドラは使用できません。

## (5) ユーザイベントハンドラの処理を実装する

EventHandler インタフェースの `execute()` を使用して、ユーザイベントハンドラの処理を実装します。

`execute()` の引数 `EventContext` を通じて、EADs サーバの情報を取得できます。

実行できる操作を次に示します。

- EADs サーバの情報の取得 (EventContext インタフェース)
- EADs クライアントの情報の取得 (ClientInfo インタフェース)
- ユーザログへの出力 (UserLogger インタフェース)

## (6) ユーザイベントハンドラの終了処理を実装する

EventHandler インタフェースの `destroy()` を使用して、EADs サーバの終了時に呼び出される、ユーザイベントハンドラの終了処理を実装します。

destroy()の引数 EventContext を通じて、EADs サーバの情報を取得できます。

なお、ユーザイベントハンドラの初期化処理に失敗した場合、destroy()は呼び出されません。

## 19.3 ユーザイベントハンドラ作成上の留意事項

---

ユーザイベントハンドラ作成上の留意事項について説明します。

### 19.3.1 パッケージ名に関する留意事項

パッケージ名に関する留意事項を次に示します。

- com.hitachi.software.xeads から始まるパッケージ名は使用しないでください。
- ユーザイベントハンドラのパッケージ名およびクラス名には、半角英数字 (0~9, A~Z, a~z), およびアンダースコア ( \_ ) が指定できます。

### 19.3.2 ユーザイベントハンドラの実装に関する留意事項

ユーザイベントハンドラの実装に関する留意事項を次に示します。

- ユーザイベントハンドラのデフォルトコンストラクタのアクセス修飾子は public にして、EADs サーバから使用できるようにしてください。
- EventContext からデータを取得する場合、次のどちらかの方法で、オブジェクトをデシリアライズするために十分な内容をクラスパスに追加してください。
  - jar ファイルのマニフェストの Class-Path 属性に jar ファイルのパスを記載する。
  - <運用ディレクトリ>/app/lib ディレクトリ下に jar ファイルを配置する。
- ユーザイベントハンドラのインスタンスは単一であり、複数の EADs クライアントから 1 つのユーザイベントハンドラを同時に実行すると、EADs サーバではマルチスレッドで同一のインスタンスを使用します。そのため、ユーザイベントハンドラには次のような処理を実装しないでください。
  - 排他制御をしないで、インスタンス変数, static 変数を更新する。
  - 排他制御をしないで、スレッドセーフでない API を使用する。
- EventContext と、そこから取得できるオブジェクトの有効範囲は、EventHandler インタフェースのメソッドの中だけです。これらのオブジェクトを、メソッドの範囲外で参照した場合の動作は保証されません。
- ユーザイベントハンドラは、EADs サーバプロセスで動作するため、カレントディレクトリは運用ディレクトリになります。
- ユーザイベントハンドラでクライアントライブラリを使用して、自クラスタのデータを操作しないでください。デッドロックや無限ループが発生するおそれがあります。
- ユーザイベントハンドラでクライアントライブラリを使用して他クラスタのデータを操作する場合、複数のクラスタが互いにデータを操作し合うことのないようにしてください。デッドロックや無限ループが発生するおそれがあります。

## 19.4 ソースプログラムをコンパイルする (ユーザイベントハンドラ)

ユーザイベントハンドラをコンパイルします。

なお、Developer では、EADs が提供する Apache Ant のタスクを使用して、コンパイルすることもできます。手順については、「[16.6.8 ソースプログラムをコンパイルする](#)」を参照してください。

### 19.4.1 ユーザイベントハンドラをコンパイルする手順

ユーザイベントハンドラをコンパイルする手順について説明します。

#### (1) ソースプログラムを配置する

作成したユーザイベントハンドラを任意の場所に配置します。

#### (2) コンパイルする

作成したユーザイベントハンドラのソースプログラムは、次に示すディレクトリに格納されている JDK の javac コマンドを使用してコンパイルします。

javac コマンドの詳細については、Java 言語のコンパイラに関するドキュメントを参照してください。

```
/opt/hitachi/xeads/PSB/jdk
```

#### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<インストールディレクトリ>%PSB%jdk
```

次に示すライブラリをクラスパスに設定します。

```
/opt/hitachi/xeads/server/lib/eads-function.jar
```

#### 参考

Developer を使用している場合は、次のライブラリを設定します。

```
<インストールディレクトリ>%server%lib%eads-function.jar
```

## 19.5 ユーザイベントハンドラをパッケージングする

「19.4 ソースプログラムをコンパイルする (ユーザイベントハンドラ)」でコンパイルした class ファイルを格納する jar ファイルを作成します。

なお、Developer では、EADs が提供する Apache Ant のタスクを使用して、jar ファイルを作成することもできます。手順については、「16.6.9 ユーザファンクションまたはユーザイベントハンドラをデプロイする」を参照してください。

### 19.5.1 ユーザイベントハンドラをパッケージングする手順

「19.4 ソースプログラムをコンパイルする (ユーザイベントハンドラ)」でコンパイルした class ファイルを格納する jar ファイルを作成する手順について説明します。

#### (1) マニフェストを設定する (任意)

ユーザイベントハンドラだけで使用するクラスは、jar ファイルのマニフェストファイル (MANIFEST.MF) の Class-Path 属性に記載します。

##### 参考

Developer を使用している場合、マニフェストファイルは次のディレクトリに格納されています。

```
<運用ディレクトリ>%project%java%user-event-handler
```

#### (2) jar ファイルを作成する

次に示すディレクトリに格納されている JDK の jar コマンドを使用して jar ファイルを作成します。

jar コマンドの詳細については、Java 言語に関するドキュメントを参照してください。

```
/opt/hitachi/xeads/PSB/jdk
```

##### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<インストールディレクトリ>%PSB%jdk
```

複数のユーザイベントハンドラを 1 つの EADs サーバで使用する場合、1 つの jar ファイルにまとめてアーカイブします。

(例)

example.SampleEventHandler および example.SampleHandler2 の、2つのユーザイベントハンドラを1つのEADsサーバで使用する場合

user-event-handler.jar

└example

├SampleEventHandler.class

└SampleEventHandler2.class

## 19.6 ユーザイベントハンドラをデプロイする

ユーザイベントハンドラをデプロイします。

ユーザイベントハンドラのデプロイは EADs サーバが停止した状態で行います。

なお、Developer では、EADs が提供する Apache Ant のタスクを使用して、デプロイすることもできます。手順については、「16.6.9 ユーザファンクションまたはユーザイベントハンドラをデプロイする」を参照してください。

### 19.6.1 ユーザイベントハンドラをデプロイする手順

ユーザイベントハンドラをデプロイする手順について説明します。

#### (1) EADs サーバを終了する

`eztool stop` コマンドを実行して、EADs サーバを終了します。

Developer を使用している場合は、Apache Ant のタスク (`ant.bat stop`) を実行して、Developer 用 EADs サーバを終了することもできます。

#### (2) 作成した jar ファイルをコピーする

「19.5.1(2) jar ファイルを作成する」で作成した jar ファイルを、次に示すディレクトリにコピーします。

<運用ディレクトリ>/app

##### 参考

Developer を使用している場合は、次のディレクトリになります。

<運用ディレクトリ>%app

#### (3) ライブラリを配置する (任意)

必要に応じて、ユーザファンクションおよびユーザイベントハンドラ共通で使用するライブラリを次に示すディレクトリに配置します。拡張子が jar であるライブラリだけが有効です。

<運用ディレクトリ>/app/lib

##### 参考

Developer を使用している場合は、次のディレクトリになります。

```
<運用ディレクトリ>¥app¥lib
```

jar ファイルのマニフェストファイル (MANIFEST.MF) の Class-Path 属性に jar ファイルのパスを記載している場合、そのパスに jar ファイルをコピーしてください。

## (4) EADs サーバを開始する

`ezserver` コマンドを実行し、EADs サーバを開始します。

Developer を使用している場合は、Apache Ant のタスク (`ant.bat start` または `ant.bat start_nocache`) を実行し、Developer 用 EADs サーバを開始することもできます。

## (5) デプロイされたかどうか確認する (任意)

`eztool listevent` コマンドを使用して、正しくデプロイされたかどうか確認します。

手順については、「[11.8 ユーザファンクションまたはユーザイベントハンドラ](#)が、各 EADs サーバに正しく配置されているかどうか、また、実行できるかどうか確認したい」を参照してください。



## 19.7 ユーザイベントハンドラを実行する

---

ユーザイベントハンドラを実行して動作を確認します。

### 19.7.1 ユーザイベントハンドラの実行

ユーザイベントハンドラを実行します。動作確認の結果、ソースプログラムを修正する場合は、「19.2 ソースプログラムを作成する (ユーザイベントハンドラ)」に戻ります。

### 19.7.2 ユーザログへの出力

UserLogger インタフェースを使用して、ユーザイベントハンドラの実行情報など (ユーザログ) を出力できます。ユーザログは、サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリに、`eads_user_message[n].log` ([n]はファイル数の通番) というファイル名で出力されます。

### 19.7.3 ユーザイベントハンドラ運用上の留意事項

- `eztool forcestop` コマンドや OS の機能などによって EADs サーバプロセスが強制終了された場合、EventHandler インタフェースの `destroy` メソッドは呼び出されません。
- EADs サーバ間でのユーザイベントハンドラの整合性は、ユーザが確保する必要があります。特定の EADs サーバにユーザイベントハンドラが存在しない場合や、ユーザイベントハンドラの内部実装が EADs サーバ間で異なっている場合でも、チェックされません。

## 19.8 実行環境に配布する

---

<運用ディレクトリ>/app ディレクトリを、実行環境の EADs サーバの<運用ディレクトリ>/app ディレクトリにコピーします。

### 参考

Developer を使用している場合は、次のディレクトリになります。

<運用ディレクトリ>¥app

# 20

## API リファレンス (Java 言語)

この章では、EADs で使用できる API (Java 言語) について説明します。

## 20.1 Java 言語のクライアントライブラリが提供するクラス

Java 言語のクライアントライブラリでは、EADs の API として次の表に示すクラスを提供します。これらのクラスは、Java 言語で記述したソースプログラムに、EADs が提供するパッケージ名を指定することで使用できます。

なお、Java 言語のクライアントライブラリで使用できるクラスはすべてスレッドセーフです。

表 20-1 Java 言語のクライアントライブラリで使用できるクラスとパッケージ名

項番	クラス名	機能	パッケージ名
1	Cache	データを操作するためのクラスです。	com.hitachi.software.xeads.client.api
2	CacheManager	キャッシュを管理するためのクラスです。	
3	Node	EADs サーバの情報を取得するためのクラスです。	
4	CacheException	Cache クラス、および CacheManager クラスに対する操作が失敗した場合に返却する例外クラスです。	
5	InitializeException	CacheManager クラスの初期化でエラーが発生した場合に返却される CacheException のサブクラスです。	
6	InternalClientException	EADs クライアントで内部エラーが発生した場合に返却される CacheException のサブクラスです。	
7	InternalServerException	EADs サーバで内部エラーが発生した場合に返却される CacheException のサブクラスです。	
8	ServerCommunicationException	通信障害が発生した場合に返却される CacheException のサブクラスです。	
9	UserOperationException	不正なユーザ操作によってエラーが発生した場合に返却される CacheException のサブクラスです。	

## 20.1.1 Cache クラス

### (1) 説明

データを操作するためのクラスです。

### (2) 継承関係

```
java.lang.Object
└─com.hitachi.software.xeads.client.api.Cache
```

### (3) 形式

```
public class Cache
extends java.lang.Object
```

### (4) メソッド一覧

Cache クラスが提供するメソッド一覧を次の表に示します。

メソッド名	説明
put()	value を key に関連づけて格納します。
create()	新規に key を格納する場合だけ、value を key に関連づけて格納します。
update()	指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します。
replace()	指定した key に関連づけられている value を、条件として指定した value (comparativeValue) と比較します。value が一致している場合だけ、value を key に関連づけて格納します。
get()	指定した key に関連づけられた value を取得します。
remove()	指定した key、および key に関連づけられた value を削除します。
executeFunction() (グループ指定)	グループを指定して、ユーザファンクションを実行します。
executeFunction() (グループおよび受信タイムアウト指定)	グループおよび受信タイムアウト時間を指定して、ユーザファンクションを実行します。
executeFunction() (EADs サーバ指定)	EADs サーバを指定して、ユーザファンクションを実行します。
executeFunction() (EADs サーバおよび受信タイムアウト指定)	EADs サーバおよび受信タイムアウト時間を指定して、ユーザファンクションを実行します。

## (5) put()

### (a) 機能

value を key に関連づけて格納します。

value の格納で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public void put(String key,  
                Object value)  
                throws CacheException
```

### (c) パラメータ

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

### (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- ServerCommunicationException (通信障害)
- InternalServerErrorException (EADs サーバの内部エラー)
- InternalClientException (EADs クライアントの内部エラー)

## (6) create()

### (a) 機能

新規に key を格納する場合だけ、value を key に関連づけて格納します。

value の格納で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public void create(String key,  
                  Object value)  
                  throws CacheException
```

## (c) パラメータ

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerException` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (7) update()

### (a) 機能

指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します。

value の格納で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public void update(String key,
                  Object value)
                  throws CacheException
```

## (c) パラメータ

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)

- `ServerCommunicationException` (通信障害)
- `InternalServerException` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (8) `replace()`

### (a) 機能

指定した `key` に関連づけられている `value` を、条件として指定した `value` (`comparativeValue`) と比較します。`value` が一致している場合だけ、`value` を `key` に関連づけて格納します。

`value` の置き換えで問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public void replace(String key,
                   Object value,
                   Object comparativeValue)
    throws CacheException
```

### (c) パラメータ

`key`

置き換える `value` に関連づけられた `key` を指定します。

指定できるデータについては、「[15.2.2\(1\) keyとして指定できるデータ](#)」を参照してください。

`value`

格納する `value` を指定します。

指定できるデータについては、「[15.2.2\(2\) valueとして指定できるデータ](#)」を参照してください。

`comparativeValue`

比較する `value` を指定します。

指定できるデータについては、「[15.2.2\(2\) valueとして指定できるデータ](#)」を参照してください。

### (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerException` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)



## (9) get()

### (a) 機能

指定した key に関連づけられた value を取得します。

value の取得で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public java.lang.Object get(String key)
    throws CacheException
```

### (c) パラメータ

key

取得する value に関連づけられた key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

### (d) 戻り値

key に関連づけられた value が返却されます。

指定した key に何も関連づけられていなかった場合は、null が返却されます。

### (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerError` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

### (f) 注意事項

`get()` を実行するスレッドのコンテキストクラスローダを変更しないでください。コンテキストクラスローダの設定が不正な場合、戻り値のオブジェクトのデシリアライズに失敗します。

## (10) remove()

### (a) 機能

指定した key, および key に関連づけられた value を削除します。

value の削除で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public void remove(String key)
    throws CacheException
```

## (c) パラメータ

key

削除する value に関連づけられた key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerError` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (11) executeFunction() (グループ指定)

### (a) 機能

グループを指定して、ユーザファンクションを実行します。

ユーザファンクションの実行で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public Object executeFunction(String groupName,
    String funcName,
    Object arg)
    throws CacheException
```

### (c) パラメータ

groupName

グループ名を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

funcName

ユーザファンクション名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数の制限はありません。

null または空文字列は不正となります。

arg

ユーザファンクションに渡す引数を指定します。

シリアライズできない場合、エラーとなります。

オブジェクトのサイズに上限はありません。ただし、EADs サーバが受け付けることができるデータの最大サイズ（サーバ定義の `eads.server.max.datasize` パラメタの指定値）を超えない値を設定してください。

引数を渡さない場合は null を指定します。

## (d) 戻り値

ユーザファンクションで設定した実行結果が返却されます。

## (e) 発生する例外

- `UserOperationException`（ユーザの操作不正）
- `ServerCommunicationException`（通信障害）
- `InternalServerException`（EADs サーバの内部エラー）
- `InternalClientException`（EADs クライアントの内部エラー）

## (f) 注意事項

`executeFunction()` を実行するスレッドのコンテキストクラスローダを変更しないでください。コンテキストクラスローダの設定が不正な場合、戻り値のオブジェクトのデシリアライズに失敗します。

# (12) `executeFunction()`（グループおよび受信タイムアウト指定）

## (a) 機能

グループおよび受信タイムアウト時間を指定して、ユーザファンクションを実行します。

受信タイムアウト時間には、クライアント定義の `eads.connection.receive.timeout` パラメタの値ではなく、引数（`recvTimeout`）で指定した値を適用します。

ユーザファンクションの実行で問題が発生した場合は、例外が返却されます。

タイムアウトが発生した場合は、`CacheException.EAD_ERROR_NET_TIMEOUT` が返却されます。

## (b) 形式

```
public Object executeFunction(String groupName,
                             String funcName,
                             Object arg,
                             int recvTimeout)
    throws CacheException
```

## (c) パラメータ

### groupName

グループ名を指定します。

指定できるデータについては、「[15.2.2\(1\) key](#)として指定できるデータ」を参照してください。

### funcName

ユーザファンクション名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数の制限はありません。

null または空文字列は不正となります。

### arg

ユーザファンクションに渡す引数を指定します。

シリアルライズできない場合、エラーとなります。

オブジェクトのサイズに上限はありません。ただし、EADs サーバが受け付けることができるデータの最大サイズ (サーバ定義の `eads.server.max.datasize` パラメタの指定値) を超えない値を設定してください。

引数を渡さない場合は null を指定します。

### recvTimeout

データ受信のタイムアウト時間 (単位: ミリ秒) を指定します。

指定できるデータについては、「[9.3.3\(3\)\(b\) eads.connection.receive.timeout](#)」を参照してください。

## (d) 戻り値

ユーザファンクションで設定した実行結果が返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerException` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (f) 注意事項

`executeFunction()` を実行するスレッドのコンテキストクラスローダを変更しないでください。コンテキストクラスローダの設定が不正な場合、戻り値のオブジェクトのデシリアルライズに失敗します。

## (13) executeFunction() (EADs サーバ指定)

### (a) 機能

EADs サーバを指定して、ユーザファンクションを実行します。

ユーザファンクションの実行で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public Object executeFunction(Node targetNode,  
                             String funcName,  
                             Object arg)  
    throws CacheException
```

### (c) パラメータ

targetNode

EADs サーバ (CacheManager クラスから取得した Node クラスのインスタンス) を指定します。それ以外のインスタンスを指定した場合の動作は保証しません。

次の場合は不正となります。

- null の場合
- 指定した Node クラスが保持するアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持する EADs サーバのアドレス情報とも一致しない場合

funcName

ユーザファンクション名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

null または空文字列は不正となります。

arg

ユーザファンクションに渡す引数を指定します。

シリアライズできない場合、エラーとなります。

オブジェクトのサイズに上限はありません。ただし、EADs サーバが受け付けることができるデータの最大サイズ (サーバ定義の `eads.server.max.datasize` パラメタの指定値) を超えない値を設定してください。

引数を渡さない場合は null を指定します。

### (d) 戻り値

ユーザファンクションで設定した実行結果が返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerException` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (f) 注意事項

`executeFunction()` を実行するスレッドのコンテキストクラスローダを変更しないでください。コンテキストクラスローダの設定が不正な場合、戻り値のオブジェクトのデシリアライズに失敗します。

# (14) `executeFunction()` (EADs サーバおよび受信タイムアウト指定)

## (a) 機能

EADs サーバおよび受信タイムアウト時間を指定して、ユーザファンクションを実行します。

受信タイムアウト時間には、クライアント定義の `eads.connection.receive.timeout` パラメタの値ではなく、引数 (`recvTimeout`) で指定した値を適用します。

ユーザファンクションの実行で問題が発生した場合は、例外が返却されます。

タイムアウトが発生した場合は、`CacheException.EAD_ERROR_NET_TIMEOUT` が返却されます。

## (b) 形式

```
public Object executeFunction(Node targetNode,
                             String funcName,
                             Object arg,
                             int recvTimeout)
    throws CacheException
```

## (c) パラメータ

`targetNode`

EADs サーバ (`CacheManager` クラスから取得した `Node` クラスのインスタンス) を指定します。それ以外のインスタンスを指定した場合の動作は保証しません。

次の場合は不正となります。

- `null` の場合
- 指定した `Node` クラスが保持するアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持する EADs サーバのアドレス情報とも一致しない場合

`funcName`

ユーザファンクション名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

null または空文字列は不正となります。

arg

ユーザファンクションに渡す引数を指定します。

シリアル化できない場合、エラーとなります。

オブジェクトのサイズに上限はありません。ただし、EADs サーバが受け付けることができるデータの最大サイズ (サーバ定義の `eads.server.max.datasize` パラメタの指定値) を超えない値を設定してください。

引数を渡さない場合は null を指定します。

recvTimeout

データ受信のタイムアウト時間 (単位: ミリ秒) を指定します。

指定できるデータについては、[「9.3.3\(3\)\(b\) eads.connection.receive.timeout」](#) を参照してください。

## (d) 戻り値

ユーザファンクションで設定した実行結果が返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerException` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (f) 注意事項

`executeFunction()` を実行するスレッドのコンテキストクラスローダを変更しないでください。コンテキストクラスローダの設定が不正な場合、戻り値のオブジェクトのデシリアライズに失敗します。

## 20.1.2 CacheManager クラス

### (1) 機能

EADs のキャッシュを管理するためのクラスです。

`create()` を実行して、EADs クライアントの初期設定をします。

EADs クライアントの使用を終了する場合は、必ず `destroy()` を実行してください。

## (2) 継承関係

```
java.lang.Object
└─com.hitachi.software.xeads.client.api.CacheManager
```

## (3) 形式

```
public class CacheManager
extends java.lang.Object
```

## (4) メソッド一覧

CacheManager クラスが提供するメソッド一覧を次の表に示します。

メソッド名	説明
create() (パス指定)	クライアント定義に従って、EADs クライアントの初期設定をします。
create() (EADs クライアント名およびパス指定)	クライアント定義に従って、EADs クライアントの初期設定をします。
create() (入力ストリーム指定)	クライアント定義に従って、EADs クライアントの初期設定をします。
create() (EADs クライアント名および入力ストリーム指定)	クライアント定義に従って、EADs クライアントの初期設定をします。
create() (クライアント定義指定)	クライアント定義に従って、EADs クライアントの初期設定をします。
create() (EADs クライアント名およびクライアント定義指定)	クライアント定義に従って、EADs クライアントの初期設定をします。
getCache()	Cache クラスのインスタンスを取得します。
removeCache()	キャッシュへのアクセスを終了します。
destroy()	EADs クライアントの使用を終了します。
getNodeList()	EADs クライアントが保持する接続先 EADs サーバの情報を取得します。
getNode()	指定した key (またはグループ) が格納されている接続先 EADs サーバの情報を取得します。
getSlaveNodeList()	指定した EADs サーバにあるデータの、コピー先となっている EADs サーバの情報を取得します。

## (5) create() (パス指定)

### (a) 機能

クライアント定義に従って、EADs クライアントの初期設定をします。



このメソッドを実行するたびに、通信タイムアウトを監視するスレッドと、クラスタを監視するスレッドが1つずつ生成されます。生成されたスレッドは、destroy()実行時に終了します。

EADs クライアントの初期設定で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public static CacheManager create(String fileName)
    throws CacheException
```

## (c) パラメータ

fileName

EADs クライアントのクライアント定義ファイルの格納先パス名を指定します。

null または空文字列は不正となります。

## (d) 戻り値

CacheManager クラスのインスタンスが返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InitializeException (CacheManager クラスの初期化失敗)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドの実行後、destroy()を実行しないとメモリリークとなるので必ず実行してください。

## (6) create() (EADs クライアント名およびパス指定)

### (a) 機能

クライアント定義に従って、EADs クライアントの初期設定をします。

このメソッドを実行するたびに、通信タイムアウトを監視するスレッドと、クラスタを監視するスレッドが1つずつ生成されます。生成されたスレッドは、destroy()実行時に終了します。

EADs クライアントの初期設定で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public static CacheManager create(String clientName, String fileName)
    throws CacheException
```

## (c) パラメータ

clientName

EADs クライアント名を指定します。

指定できるデータについては、「[15.2.2\(4\) EADs クライアント名として指定できるデータ](#)」を参照してください。

EADs クライアント名とログファイルの出力先の関係については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。

fileName

EADs クライアントのクライアント定義ファイルの格納先パス名を指定します。

null または空文字列は不正となります。

## (d) 戻り値

CacheManager クラスのインスタンスが返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InitializeException (CacheManager クラスの初期化失敗)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドの実行後、destroy()を実行しないとメモリリークとなるので必ず実行してください。

## (7) create() (入カストリーム指定)

### (a) 機能

クライアント定義に従って、EADs クライアントの初期設定をします。

このメソッドを実行するたびに、通信タイムアウトを監視するスレッドと、クラスタを監視するスレッドが1つずつ生成されます。生成されたスレッドは、destroy()実行時に終了します。

EADs クライアントの初期設定で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public static CacheManager create(InputStream in)
    throws CacheException
```

## (c) パラメータ

in

EADs クライアントのクライアント定義ファイルを読み込む入力ストリームを指定します。  
null の場合は不正となります。

## (d) 戻り値

CacheManager クラスのインスタンスが返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InitializeException (CacheManager クラスの初期化失敗)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドの実行後、destroy() を実行しないとメモリリークとなるので必ず実行してください。

# (8) create() (EADs クライアント名および入力ストリーム指定)

## (a) 機能

クライアント定義に従って、EADs クライアントの初期設定をします。

このメソッドを実行するたびに、通信タイムアウトを監視するスレッドと、クラスタを監視するスレッドが1つずつ生成されます。生成されたスレッドは、destroy() 実行時に終了します。

EADs クライアントの初期設定で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public static CacheManager create(String clientName, InputStream in)
    throws CacheException
```

## (c) パラメータ

clientName

EADs クライアント名を指定します。

指定できるデータについては、「[15.2.2\(4\) EADs クライアント名として指定できるデータ](#)」を参照してください。

EADs クライアント名とログファイルの出力先の関係については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。

in

EADs クライアントのクライアント定義ファイルを読み込む入力ストリームを指定します。

null の場合は不正となります。

## (d) 戻り値

CacheManager クラスのインスタンスが返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InitializeException (CacheManager クラスの初期化失敗)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドの実行後、destroy()を実行しないとメモリリークとなるので必ず実行してください。

## (9) create() (クライアント定義指定)

### (a) 機能

クライアント定義に従って、EADs クライアントの初期設定をします。

このメソッドを実行するたびに、通信タイムアウトを監視するスレッドと、クラスタを監視するスレッドが1つずつ生成されます。生成されたスレッドは、destroy()実行時に終了します。

EADs クライアントの初期設定で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public static CacheManager create(Properties properties)
    throws CacheException
```

## (c) パラメータ

properties

EADs クライアントのクライアント定義を指定します。  
null の場合は不正となります。

## (d) 戻り値

CacheManager クラスのインスタンスが返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InitializeException (CacheManager クラスの初期化失敗)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドの実行後、destroy() を実行しないとメモリリークとなるので必ず実行してください。

# (10) create() (EADs クライアント名およびクライアント定義指定)

## (a) 機能

クライアント定義に従って、EADs クライアントの初期設定をします。

このメソッドを実行するたびに、通信タイムアウトを監視するスレッドと、クラスタを監視するスレッドが1つずつ生成されます。生成されたスレッドは、destroy() 実行時に終了します。

EADs クライアントの初期設定で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public static CacheManager create(String clientName, Properties properties)
    throws CacheException
```

## (c) パラメータ

clientName

EADs クライアント名を指定します。

指定できるデータについては、「[15.2.2\(4\) EADs クライアント名として指定できるデータ](#)」を参照してください。

EADs クライアント名とログファイルの出力先の関係については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。

properties

EADs クライアントのクライアント定義を指定します。

null の場合は不正となります。

## (d) 戻り値

CacheManager クラスのインスタンスが返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InitializeException (CacheManager クラスの初期化失敗)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドの実行後、destroy()を実行しないとメモリリークとなるので必ず実行してください。

## (11) getCache()

### (a) 機能

キャッシュへのアクセスを開始し、Cache クラスのインスタンスを取得します。

また、removeCache()でアクセスを終了したキャッシュのインスタンスを再度開始状態にします。

キャッシュへのアクセス開始で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public Cache getCache(String name)
    throws CacheException
```

## (c) パラメータ

name

取得する Cache クラスのキャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

## (d) 戻り値

指定したキャッシュ名に関連づけられた Cache クラスのインスタンスが返却されます。

すでに同じキャッシュ名を指定して `getCache()` を実行している場合は、初回実行時と同じインスタンスが返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `ServerCommunicationException` (通信障害)
- `InternalServerError` (EADs サーバの内部エラー)
- `InternalClientException` (EADs クライアントの内部エラー)

## (12) `removeCache()`

### (a) 機能

キャッシュへのアクセスを終了します。

キャッシュのアクセス終了で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public void removeCache(String name)
    throws CacheException
```

### (c) パラメータ

name

終了する Cache クラスのキャッシュ名を指定します。

指定できるデータについては、「15.2.2(3) キャッシュ名として指定できるデータ」を参照してください。

#### (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalClientException (EADs クライアントの内部エラー)

#### (e) 注意事項

同じキャッシュ名を指定して取得したすべてのキャッシュ (Cache クラスのインスタンス) へのアクセスを終了するため、複数スレッドで実行している場合などは注意してください。

### (13) destroy()

#### (a) 機能

すべてのキャッシュに対して終了処理を行い、EADs クライアントの使用を終了します。

destroy()を実行して EADs クライアントがすでに終了している場合は、CacheException.EAD\_ERROR\_CLIENT\_FINALIZED が返却されます。

#### (b) 形式

```
public void destroy()  
        throws CacheException
```

#### (c) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalClientException (EADs クライアントの内部エラー)

#### (d) 注意事項

- destroy()は create()と対になるように実行してください。create()で取得したインスタンスにつき1度だけ実行してください。
- J2EE サーバ (uCosminexus Application Server) 上で実行する場合、このメソッドを実行しないとメモリリークとなるので必ず実行してください。

### (14) getNodeList()

#### (a) 機能

EADs クライアントが保持する接続先 EADs サーバの情報を取得します。

接続先 EADs サーバの情報の取得で問題が発生した場合は、例外が返却されます。



## (b) 形式

```
public Node[] getNodeList()  
    throws CacheException
```

## (c) 戻り値

EADs クライアントが保持する接続先 EADs サーバの情報が、Node クラスの配列として返却されます。

EADs サーバの状態に関係なく、すべての EADs サーバの情報が返却されます。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalClientException` (EADs クライアントの内部エラー)

## (e) 注意事項

このメソッドの実行時には、EADs サーバへの通信は行われません。そのため、取得した情報が最新でないことがあります。

# (15) getNode()

## (a) 機能

指定した key (またはグループ) が格納されている接続先 EADs サーバの情報を取得します。

接続先 EADs サーバの取得で問題が発生した場合は、例外が返却されます。

## (b) 形式

```
public Node getNode(String key)  
    throws CacheException
```

## (c) パラメータ

key

取得したい接続先 EADs サーバに格納されている key (またはグループ) を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

## (d) 戻り値

指定した key (またはグループ) が格納されている接続先 EADs サーバを表す Node クラスのインスタンスが返却されます。

接続先 EADs サーバの接続可否については、戻り値に影響しません。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

このメソッドの実行時には、EADs サーバへの通信は行われません。そのため、取得した情報が最新でないことがあります。

## (16) getSlaveNodeList()

### (a) 機能

指定した EADs サーバにあるデータの、コピー先となっている EADs サーバの情報を取得します。

データの複製先 EADs サーバの情報の取得で問題が発生した場合は、例外が返却されます。

### (b) 形式

```
public Node[] getSlaveNodeList(Node masterNode)
                throws CacheException
```

### (c) パラメータ

masterNode

コピー元のデータを格納する EADs サーバ (CacheManager クラスから取得した Node クラスのインスタンス) を指定します。それ以外のインスタンスを指定した場合の動作は保証しません。

次の場合は不正となります。

- null の場合
- 指定した Node クラスが保持するアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持する EADs サーバのアドレス情報とも一致しない場合

### (d) 戻り値

複製先 EADs サーバの情報が、Node クラスの配列として返却されます。

接続先 EADs サーバの接続可否については、戻り値に影響しません。

データの多重度が 1 の場合、要素数 0 の配列が返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalClientException (EADs クライアントの内部エラー)

## (f) 注意事項

このメソッドの実行時には、EADs サーバへの通信は行われません。そのため、取得した情報が最新でないことがあります。

## 20.1.3 Node クラス

### (1) 説明

EADs サーバの情報を取得するためのクラスです。

### (2) 継承関係

```
java.lang.Object
└com.hitachi.software.xeads.client.api.Node
```

### (3) 形式

```
public class Node
extends java.lang.Object
```

### (4) メソッド一覧

Node クラスが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getNodeId()</code>	EADs サーバ ID を取得します。
<code>isEnabled()</code>	EADs サーバとの接続可否を取得します。
<code>getAddress()</code>	EADs サーバの IP アドレスとポート番号を取得します。
<code>getPosition()</code>	EADs サーバの位置 (ハッシュ値) を取得します。
<code>toString()</code>	EADs サーバの情報 (文字列表現) を取得します。

### (5) `getNodeId()`

#### (a) 機能

EADs サーバ ID を取得します。

#### (b) 形式

```
public int getNodeId()
```

## (c) 戻り値

EADs サーバ ID が返却されます。

## (6) isEnabled()

### (a) 機能

EADs サーバとの接続可否を取得します。

### (b) 形式

```
public boolean isEnabled()
```

## (c) 戻り値

EADs サーバとの接続可否が返却されます。

true

接続できます。

false

接続できません。

## (7) getAddress()

### (a) 機能

EADs サーバの IP アドレスとポート番号を取得します。

### (b) 形式

```
public java.net.InetSocketAddress getAddress()
```

## (c) 戻り値

EADs サーバの IP アドレスとポート番号を示す `java.net.InetSocketAddress` インスタンスが返却されます。

## (8) getPosition()

### (a) 機能

EADs サーバの位置（ハッシュ値）を取得します。

## (b) 形式

```
public int getPosition()
```

## (c) 戻り値

EADs サーバの位置（ハッシュ値）が返却されます。

## (9) toString()

### (a) 機能

EADs サーバの情報（文字列表現）を取得します。

### (b) 形式

```
public String toString()
```

### (c) 戻り値

EADs サーバの情報（文字列表現）が次の形式で返却されます。

```
” node id = <EADsサーバID>, status = <ENABLE | DISABLE>, position = <EADsサーバの位置>, address = <EADsサーバのIPアドレスとポート番号>”
```

## 20.1.4 CacheException クラス

### (1) 説明

Cache クラス、および CacheManager クラスに対する操作が失敗した場合に返却される例外クラスです。

getErrorCode()によってエラーコードを取得し、エラーの内容を判別します。

### (2) 継承関係

```
java.lang.Object
├─ java.lang.Throwable
│   └─ java.lang.Exception
│       └─ com.hitachi.software.xeads.client.api.CacheException
```

### (3) 形式

```
public class CacheException
extends Exception
```

## (4) メソッド一覧

CacheException クラスが提供するメソッド一覧を次の表に示します。

メソッド名	説明
getErrorCode()	発生した例外のエラーコードを取得します。

## (5) getErrorCode()

### (a) 機能

発生した例外のエラーコードを取得します。

### (b) 形式

```
public int getErrorCode()
```

### (c) 戻り値

戻り値としてエラーコードが返却されます。エラーコードとエラーの内容、発生要因を次の表に示します。

エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
1000	EAD_ERROR_UNEXPECTED	CacheException	予期しないエラーが発生しました。	プログラム内で予期しないエラーが発生しました。
1010	EAD_ERROR_INVALID_PARAMETER	UserOperationException	指定されたパラメタが不正です。	API の引数に不正なパラメタを指定しています。
1030	EAD_ERROR_CLIENT_FINALIZED	UserOperationException	EADs クライアントの使用を完了しているため、処理を実行できませんでした。	CacheManager クラスの destroy() を実行したあとで、CacheManager クラスのメソッドが実行されました。
1040	EAD_ERROR_CACHE_NOT_STARTED	UserOperationException	キャッシュが開始されていないため実行できません。	次の要因が考えられます。 <ul style="list-style-type: none"><li>• キャッシュの終了後 (CacheManager クラスの removeCache() の実行後) に、データを操作した。</li><li>• EADs クライアントの使用を完了したあと (CacheManager クラスの destroy() 実行後) に、データを操作した。</li></ul>

エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
1050	EAD_ERROR_NOT_SERIALIZABLE	UserOperationException	EADs クライアントでのシリアライズ処理に失敗しました。	API の引数にシリアライズできないオブジェクトを指定しています。
1060	EAD_ERROR_NOT_DESERIALIZABLE	UserOperationException	EADs クライアントでのデシリアライズ処理に失敗しました。	EADs サーバから取得したオブジェクトは EADs クライアントでデシリアライズできませんでした。 EADs クライアントのクラスパスに含まれないオブジェクトであることが考えられます。
1100	EAD_ERROR_CACHE_NOT_NEED_STOP	UserOperationException	すでに終了しているキャッシュに対して、終了処理をしています。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• キャッシュの終了後 (CacheManager クラスの removeCache() の実行後) に、キャッシュを終了した。</li> <li>• CacheManager クラスの getCache() を実行していない状態で、CacheManager クラスの removeCache() を実行した。</li> </ul>
1110	EAD_ERROR_INVALID_NODE_ADDRESS	UserOperationException	指定した EADs サーバのアドレス情報が、EADs クライアントが保持する EADs サーバのアドレス情報と一致しません。	API の引数に指定した EADs サーバのアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持するどの EADs サーバのアドレス情報とも一致しません。
1120	EAD_ERROR_EXCEED_MAX_CONNECTION_POOL_SIZE	UserOperationException	同一接続先に対してプールしておく接続数がすでに最大個数に達していて、すべて使用中です。	特定の EADs サーバに対して同時にリクエストしているスレッド数が、接続数の最大個数を超えています。
2000	EAD_ERROR_INIT	InitializeException	EADs クライアントの初期設定でエラーが発生しました。	CacheManager クラスの create() 実行中に予期しないエラーが発生しました。
2010	EAD_ERROR_INIT_PROPERTIES	InitializeException	クライアント定義ファイルが読み込めませんでした。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• クライアント定義ファイルが存在しない。</li> </ul>

エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
				<ul style="list-style-type: none"> <li>クライアント定義ファイルに読み込み権限がない。</li> <li>格納先パス名がファイルではなくディレクトリを指している。</li> <li>CacheManager クラスの create() で指定した入力ストリームに問題がある。</li> </ul>
2020	EAD_ERROR_INIT_INVALID_PROPERTY	InitializeException	クライアント定義ファイルの定義内容に不正があります。	クライアント定義ファイルの定義内容が不正です。
2030	EAD_ERROR_INIT_LOGGER	InitializeException	ログの初期化に失敗しました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>指定されたディレクトリ、または出力先のログファイルに書き込み権限がない。</li> <li>指定されたディレクトリに同名のファイルが存在する。</li> <li>指定されたパス名、またはファイル名が不正である。</li> <li>ログファイル名と同名のディレクトリが存在した。</li> <li>ログ出力を開始するためのメモリが不足している。</li> </ul>
2040	EAD_ERROR_INIT_CLUSTERINFO	InitializeException	クライアント定義ファイルで指定された EADs サーバへの接続に失敗しました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>クライアント定義に指定した接続先 EADs サーバの指定に誤りがある。</li> <li>接続先 EADs サーバとの通信に失敗した。または接続先 EADs サーバで障害が発生している。</li> <li>接続先 EADs サーバの最大同時接続数を超えた。</li> <li>接続先 EADs サーバがリクエストの受付を</li> </ul>



エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
				開始できる状態でない。
3000	EAD_ERROR_NET	ServerCommunicationException	EADs サーバとの間で通信エラーが発生しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 通信中にネットワーク障害が発生した。</li> <li>• 通信先の EADs サーバで問題が発生した。</li> <li>• 通信先のホストで問題が発生した。</li> <li>• EADs サーバが受信できるデータの最大サイズを超えた。</li> </ul>
3010	EAD_ERROR_NET_TIMEOUT	ServerCommunicationException	EADs サーバとの通信中にタイムアウトが発生しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバで問題が発生した。</li> <li>• 接続先のホストで問題が発生した。</li> <li>• ネットワークで問題が発生した。</li> <li>• タイムアウト時間の設定が誤っている。</li> </ul>
3020	EAD_ERROR_NET_CONNECTION	ServerCommunicationException	EADs サーバとの接続に失敗しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバで問題が発生した。</li> <li>• 接続先 EADs サーバに関する設定が誤っている。</li> <li>• ネットワークで問題が発生した。</li> <li>• タイムアウト時間の設定が誤っている。</li> </ul>
3030	EAD_ERROR_NET_PROTOCOL	ServerCommunicationException	EADs サーバとの通信中にプロトコル障害が発生しました。	接続先 EADs サーバが誤りです。
3040	EAD_ERROR_NET_CLUSTERINFO	ServerCommunicationException	接続できる、すべての EADs サーバへの接続に失敗しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバで問題が発生した。</li> <li>• 接続先 EADs サーバに関する設定が誤っている。</li> <li>• ネットワークで問題が発生した。</li> </ul>

エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
				<ul style="list-style-type: none"> <li>指定したキャッシュが接続先 EADs サーバに存在しない。</li> <li>接続先 EADs サーバが閉塞状態となっている。</li> <li>EADs クライアントが保持するクラスタ構成情報と、再起動した接続先 EADs サーバが保持するクラスタ構成情報が一致しない。</li> <li>接続先 EADs サーバの最大同時接続数を超えた。</li> </ul>
4000	EAD_ERROR_SERVER	InternalServerErrorException	EADs サーバで内部エラーが発生しました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>接続先 EADs サーバで内部エラーが発生した。</li> <li>接続先 EADs サーバが閉塞状態となっている。</li> <li>接続先 EADs サーバのユーザファンクション内でエラーが発生した。</li> <li>指定したキャッシュ名のキャッシュが接続先 EADs サーバに存在しない。</li> <li>指定した key のサイズがクラスタに格納できる key の最大サイズを超えている。</li> <li>データの格納先の空き容量が不足している。</li> </ul>
4010	EAD_ERROR_SERVER_UNSUPPORTED_REQUEST	InternalServerErrorException	EADs クライアントが送信したリクエストを、接続先 EADs サーバが処理できませんでした。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>ユーザファンクションの引数に、接続先 EADs サーバがデシリアライズできないオブジェクトを指定した。</li> <li>データが破損しているなどの理由で、接続先 EADs サーバがリクエストを処理できない。</li> </ul>

エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
4020	EAD_ERROR_SERVER_EVENTHANDLER	InternalServerErrorException	データの更新操作および多重化は成功したが、ユーザイベントハンドラ内の処理で例外が発生しました。	接続先 EADs サーバのユーザイベントハンドラ内の処理で例外が発生しました。
4030	EAD_ERROR_SERVER_UNAVAILABLE	InternalServerErrorException	接続先 EADs サーバプロセスが一時的に利用できません。	接続先 EADs サーバの最大同時接続数を超過しました。
4040	EAD_ERROR_SERVER_INCOMPATIBLE_CLUSTERINFO	InternalServerErrorException	接続先 EADs サーバが保持するクラスタ構成情報と、EADs クライアントが保持するクラスタ構成情報に互換性がありません。	再起動した接続先 EADs サーバのクラスタ構成情報が、EADs クライアントが保持するクラスタ構成情報と一致しません。
4060	EAD_ERROR_SERVER_REPLACE_METHOD_NOT_MATCHED	InternalServerErrorException	replace()実行時、すでに格納されている value が comparativeValue と一致しなかったため、value を格納できませんでした。	Cache クラスの replace() で条件に指定した value とキャッシュ上の value が一致しませんでした。
4070	EAD_ERROR_SERVER_REPLACE_METHOD_KEY_NOT_EXIST	InternalServerErrorException	replace()実行時、指定した key が存在しなかったため (key に関連づけられた value が存在しなかったため)、value を格納できませんでした。	Cache クラスの replace() で指定した key に関連づけられた value が存在しなかったため、value の比較ができませんでした。
4080	EAD_ERROR_SERVER_CREATE_METHOD_KEY_EXIST	InternalServerErrorException	create()実行時、すでに格納されている key が存在したため、value を格納できませんでした。	Cache クラスの create() で指定した key に関連づけられた value がすでに格納されています。
4090	EAD_ERROR_SERVER_UPDATE_METHOD_KEY_NOT_EXIST	InternalServerErrorException	update()実行時、格納されている key が存在しなかったため、value を格納できませんでした。	Cache クラスの update() で指定した key に関連づけられた value が格納されていません。
4100	EAD_ERROR_SERVER_NOT_RUNNING	InternalServerErrorException	データのコピー元、およびコピー先の EADs サーバがすべて縮退状態または停止状態です。	EADs クライアントからのリクエストを処理する EADs サーバと、データのコピー先 EADs サーバがすべて縮退状態または停止状態です。
5000	EAD_ERROR_CLIENT	InternalClientException	EADs クライアントで内部エラーが発生しました。	クライアントライブラリ内で予期しないエラーが発生しました。
5010	EAD_ERROR_CLIENT_OUT_OF_MEMORY	InternalClientException	EADs クライアント内でメモリ確保に失敗しました。	クライアントライブラリ内でメモリ確保に失敗しました。

## 20.1.5 InitializeException クラス

### (1) 説明

CacheManager クラスの初期化でエラーが発生した場合に返却される CacheException のサブクラスです。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.client.api.CacheException
│           └─com.hitachi.software.xeads.client.api.InitializeException
```

### (3) 形式

```
public class InitializeException
extends CacheException
```

## 20.1.6 InternalClientException クラス

### (1) 説明

EADs クライアントで内部エラーが発生した場合に返却される CacheException のサブクラスです。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.client.api.CacheException
│           └─com.hitachi.software.xeads.client.api.InternalClientException
```

### (3) 形式

```
public class InternalClientException
extends CacheException
```

## 20.1.7 InternalServerErrorException クラス

### (1) 説明

EADs サーバで内部エラーが発生した場合に返却される CacheException のサブクラスです。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.client.api.CacheException
│           └─com.hitachi.software.xeads.client.api.InternalServerErrorException
```

### (3) 形式

```
public class InternalServerErrorException
extends CacheException
```

## 20.1.8 ServerCommunicationException クラス

### (1) 説明

通信障害が発生した場合に返却される CacheException のサブクラスです。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.client.api.CacheException
│           └─com.hitachi.software.xeads.client.api.ServerCommunicationException
```

### (3) 形式

```
public class ServerCommunicationException
extends CacheException
```

## 20.1.9 UserOperationException クラス

### (1) 説明

不正なユーザ操作によってエラーが発生した場合に返却される CacheException のサブクラスです。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.client.api.CacheException
│           └─com.hitachi.software.xeads.client.api.UserOperationException
```

### (3) 形式

```
public class UserOperationException
extends CacheException
```

## 20.2 ユーザファンクションで使用できる API

ユーザファンクションで使用できる API を次の表に示します。

表 20-2 ユーザファンクションで使用できる API

項番	インタフェース名, クラス名, または列挙型	機能	パッケージ名
1	Function	EADs サーバで使用するユーザファンクションが実装する必要があるインタフェースです。	com.hitachi.software.xeads.func.Function
2	FunctionContext	ユーザファンクションの実行に必要な情報を Function インタフェースに渡すインタフェースです。	com.hitachi.software.xeads.func.FunctionContext
3	InitConfig	ユーザファンクションの初期化に使用したファンクション定義の情報を取得するインタフェースです。	com.hitachi.software.xeads.func.InitConfig
4	ClientInfo	実行した EADs クライアントの情報 (PRF トレースに出力される RootAP 情報に相当する情報) を取得するインタフェースです。	com.hitachi.software.xeads.common.ClientInfo
5	ServerInfo	EADs サーバの情報を取得するためのインタフェースです。	com.hitachi.software.xeads.common.ServerInfo
6	Store	ユーザファンクションの実行時にデータを操作するインタフェースです。	com.hitachi.software.xeads.func.store.Store
7	Group	ユーザファンクションを実行中の EADs サーバに属するグループを操作するためのインタフェースです。	com.hitachi.software.xeads.func.store.Group
8	Key	ユーザファンクションで使用できる API 内で, key を表現するインタフェースです。	com.hitachi.software.xeads.func.store.Key
9	UserLogger	ユーザログのインタフェースです。	com.hitachi.software.xeads.common.UserLogger
10	EADsStoreException	Group インタフェースに対する操作が失敗した場合に返却される例外クラスです。	com.hitachi.software.xeads.func.store.EADsStoreException
11	InternalServerException	EADs サーバで内部エラーが発生した場合に返却される EADsStoreException のサブクラスです。	com.hitachi.software.xeads.func.store.InternalServerException

項番	インタフェース名, クラス名, または列挙型	機能	パッケージ名
12	UserOperationException	不正なユーザ操作によってエラーが発生した場合に返却される EADsStoreException のサブクラスです。	com.hitachi.software.xeads.func.store.UserOperationException
13	列挙型 CacheType	キャッシュタイプを表す列挙型です。	com.hitachi.software.xeads.common.CacheType

## 20.2.1 Function インタフェース

### (1) 説明

EADs サーバで使用するユーザファンクションが実装する必要のあるインタフェースです。

ユーザファンクションの初期化時にインスタンスを生成します。インスタンスは単一で、EADs サーバの終了処理まで生存します。

### (2) インタフェース名

```
com.hitachi.software.xeads.func.Function
```

### (3) メソッド一覧

Function インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
init()	ユーザファンクションの初期化処理を実装します。
execute()	ユーザファンクションの処理を実装します。
destroy()	ユーザファンクションの終了処理を実装します。

### (4) init()

#### (a) 機能

ユーザファンクションの初期化処理を実装します。

EADs サーバ起動時のユーザファンクションの初期化時に呼び出されます。



## (b) 形式

```
public void init(FunctionContext context)
```

## (c) パラメータ

context

ユーザファンクションの実行に必要な情報を取得します。

ユーザファンクションの実行に必要な情報については、「[20.2.2 FunctionContext インタフェース](#)」を参照してください。

## (5) execute()

### (a) 機能

ユーザファンクションの処理を実装します。

EADs クライアントからの、ユーザファンクション実行リクエスト時に呼び出されます。

### (b) 形式

```
public Object execute(FunctionContext context)
```

### (c) パラメータ

context

ユーザファンクションの実行に必要な情報を取得します。

ユーザファンクションの実行に必要な情報については、「[20.2.2 FunctionContext インタフェース](#)」を参照してください。

### (d) 戻り値

ユーザファンクションの処理結果が返却されます。

### (e) 注意事項

マルチスレッドで実行するユーザファンクションの場合は、スレッドセーフに実装してください。

## (6) destroy()

### (a) 機能

ユーザファンクションの終了処理を実装します。

EADs サーバ終了時のユーザファンクションの終了時に呼び出されます。

## (b) 形式

```
public void destroy(FunctionContext context)
```

## (c) パラメータ

context

ユーザファンクションの実行に必要な情報を取得します。

ユーザファンクションの実行に必要な情報については、「[20.2.2 FunctionContext インタフェース](#)」を参照してください。

## 20.2.2 FunctionContext インタフェース

### (1) 説明

ユーザファンクションの実行に必要な情報を Function インタフェースに渡すインタフェースです。

FunctionContext インタフェースのインスタンスは、1 リクエストにつき 1 つ生成されます。

### (2) インタフェース名

```
com.hitachi.software.xeads.func.FunctionContext
```

### (3) メソッド一覧

FunctionContext インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getServerName()</code>	ユーザファンクションを実行する EADs サーバ名 (運用ディレクトリ名) を取得します。
<code>getCacheName()</code>	ユーザファンクションが実行されたキャッシュ名を取得します。
<code>getGroupName()</code>	ユーザファンクションの実行時に、EADs クライアントで指定したグループ名を取得します。
<code>getArgument()</code>	ユーザファンクションの実行時に、EADs クライアントで指定した引数を取得します。
<code>getStore()</code>	ユーザファンクションの呼び出しに使用したキャッシュの、データを操作するためのインスタンスを取得します。
<code>getStore()</code> (キャッシュ名指定)	指定したキャッシュ名のキャッシュの、データを操作するためのインスタンスを取得します。
<code>getLogger()</code>	ユーザログを出力するロガーを取得します。
<code>getClientInfo()</code>	ユーザファンクションを実行した EADs クライアントの情報 (PRF トレースに出力される RootAP 情報に相当する情報) を取得します。

メソッド名	説明
<code>getInitConfig()</code>	ユーザ関クションの初期化に使用した関クション定義の情報を取得します。
<code>getServerInfo()</code>	ユーザ関クションを実行した EADs サーバの情報を取得します。

## (4) `getServerName()`

### (a) 機能

ユーザ関クションを実行する EADs サーバ名 (運用ディレクトリ名) を取得します。

### (b) 形式

```
public String getServerName()
```

### (c) 戻り値

ユーザ関クションを実行する EADs サーバ名 (運用ディレクトリ名) が返却されます。

## (5) `getCacheName()`

### (a) 機能

ユーザ関クションが実行されたキャッシュ名を取得します。

### (b) 形式

```
public String getCacheName()
```

### (c) 戻り値

ユーザ関クションが実行されたキャッシュ名が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (6) `getGroupName()`

### (a) 機能

ユーザ関クションの実行時に、EADs クライアントで指定したグループ名を取得します。

### (b) 形式

```
public String getGroupName()
```

## (c) 戻り値

ユーザ関数の実行時に、EADs クライアントで指定したグループ名が返却されます。

次の場合は null が返却されます。

- EADs クライアントでグループを指定しなかった場合
- Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合

## (7) getArguments()

### (a) 機能

ユーザ関数の実行時に、EADs クライアントで指定した引数を取得します。

### (b) 形式

```
public Object getArguments()
```

## (c) 戻り値

ユーザ関数の実行時に、EADs クライアントで指定した引数が返却されます。

次の場合は null が返却されます。

- EADs クライアントで引数に null を指定した場合
- Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合

## (8) getStore()

### (a) 機能

ユーザ関数の呼び出しに使用したキャッシュの、データを操作するためのインスタンスを取得します。

### (b) 形式

```
public Store getStore()
```

## (c) 戻り値

ユーザ関数の呼び出しに使用したキャッシュの、データを操作するためのインスタンスが返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、null が返却されます。

## (9) `getStore()` (キャッシュ名指定)

### (a) 機能

指定したキャッシュ名のキャッシュの、データを操作するためのインスタンスを取得します。

### (b) 形式

```
public Store getStore(String cacheName)
                throws EADsStoreException
```

### (c) パラメータ

`cacheName`

操作対象となるキャッシュのキャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

なお、作成していないキャッシュのキャッシュ名を指定した場合は不正となります。

### (d) 戻り値

指定したキャッシュ名のキャッシュの、データを操作するためのインスタンスが返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、`null` が返却されます。

### (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `EADsStoreException` (予期しないエラー)

## (10) `getLogger()`

### (a) 機能

ユーザログを出力するロガーを取得します。

### (b) 形式

```
public UserLogger getLogger()
```

### (c) 戻り値

ユーザログを出力するロガーが返却されます。

## (11) getClientInfo()

### (a) 機能

ユーザファンクションを実行した EADs クライアントの情報（PRF トレースに出力される RootAP 情報に相当する情報）を取得します。

### (b) 形式

```
public ClientInfo getClientInfo()
```

### (c) 戻り値

ユーザファンクションを実行した EADs クライアントの情報が返却されます。EADs クライアントの情報については、「[20.2.4 ClientInfo インタフェース](#)」を参照してください。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (12) getInitConfig()

### (a) 機能

ユーザファンクションの初期化に使用したファンクション定義の情報を取得します。

### (b) 形式

```
public InitConfig getInitConfig()
```

### (c) 戻り値

ユーザファンクションの初期化に使用したファンクション定義の情報が返却されます。

ユーザファンクションの初期化に使用したファンクション定義の情報については、「[20.2.3 InitConfig インタフェース](#)」を参照してください。

## (13) getServerInfo()

### (a) 機能

ユーザファンクションを実行した EADs サーバの情報を取得します。

### (b) 形式

```
public ServerInfo getServerInfo()
```

## (c) 戻り値

ユーザファンクションを実行した EADs サーバの情報が返却されます。

## 20.2.3 InitConfig インタフェース

### (1) 説明

ユーザファンクションの初期化に使用したファンクション定義の情報を取得するインタフェースです。

InitConfig インタフェースのインスタンスは、ユーザファンクションの初期化時に、ユーザファンクションごとに 1 つ生成されます。

### (2) インタフェース名

```
com.hitachi.software.xeads.func.InitConfig
```

### (3) メソッド一覧

InitConfig インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getFunctionProperty()</code>	ユーザファンクションの初期化に使用したファンクション定義の指定値を取得します。
<code>getFunctionPropertyNames()</code>	ユーザファンクションの初期化に使用したファンクション定義のパラメタの一覧を取得します。
<code>getFunctionName()</code>	現在実行しているユーザファンクションの名称を取得します。

### (4) `getFunctionProperty()`

#### (a) 機能

ユーザファンクションの初期化に使用したファンクション定義の指定値を取得します。

ファンクション定義の指定値は、指定値そのものではなく、ベリファイされた値です。

#### (b) 形式

```
public String getFunctionProperty(java.lang.String propName)
```

## (c) パラメータ

propName

ファンクション定義のパラメータ名を指定します。

## (d) 戻り値

パラメータ名に対応する指定値が返却されます。

パラメータ名に対応する指定値が存在しない場合は、null が返却されます。

## (e) 発生する例外

- NullPointerException (パラメータ名が null)

## (5) getFunctionPropertyNames()

### (a) 機能

ユーザファンクションの初期化に使用したファンクション定義のパラメータ名の一覧を取得します。

### (b) 形式

```
public Set<String> getFunctionPropertyNames ()
```

### (c) 戻り値

ユーザファンクションの初期化に使用したファンクション定義のパラメータ名の一覧が返却されます。

## (6) getFunctionName()

### (a) 機能

現在実行しているユーザファンクションの名称を取得します。

### (b) 形式

```
public String getFunctionName()
```

### (c) 戻り値

現在実行しているユーザファンクションの名称が返却されます。



## 20.2.4 ClientInfo インタフェース

### (1) 説明

実行した EADs クライアントの情報（PRF トレースに出力される RootAP 情報に相当する情報）を取得するインタフェースです。

### (2) インタフェース名

```
com.hitachi.software.xeads.common.ClientInfo
```

### (3) メソッド一覧

ClientInfo インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getIp()</code>	実行した EADs クライアントの IP アドレス（PRF トレースに出力される RootAP IP に相当する情報）を取得します。
<code>getPid()</code>	実行した EADs クライアントが付与した番号（PRF トレースに出力される RootAP PID に相当する情報）を取得します。

### (4) `getIp()`

#### (a) 機能

実行した EADs クライアントの IP アドレス（PRF トレースに出力される RootAP IP に相当する情報）を取得します。

#### (b) 形式

```
public byte[] getIp()
```

#### (c) 戻り値

実行した EADs クライアントの IP アドレスが返却されます。

### (5) `getPid()`

#### (a) 機能

実行した EADs クライアントが付与した番号（PRF トレースに出力される RootAP PID に相当する情報）を取得します。

## (b) 形式

```
public int getPid()
```

## (c) 戻り値

実行した EADs クライアントが付与した番号が返却されます。

## 20.2.5 ServerInfo インタフェース

### (1) 説明

EADs サーバの情報を取得するためのインタフェースです。

### (2) インタフェース名

```
com.hitachi.software.xeads.common.ServerInfo
```

### (3) メソッド一覧

ServerInfo インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getName()</code>	ユーザファンクションを実行している EADs サーバ名（運用ディレクトリ名）を取得します。
<code>getAddress()</code>	ユーザファンクションを実行している EADs サーバの IP アドレスおよびポート番号を取得します。
<code>getCacheNames()</code>	ユーザファンクションを実行している EADs サーバで作成済みのキャッシュの、キャッシュ名の一覧を取得します。
<code>getCacheType()</code>	指定したキャッシュ名のキャッシュから、キャッシュタイプの情報を取得します。
<code>getEHeapSize()</code>	ユーザファンクションを実行している、EADs サーバが確保している Explicit ヒープサイズを取得します。
<code>getEHeapUsageSize()</code>	ユーザファンクションを実行している、EADs サーバが使用している Explicit ヒープサイズを取得します。
<code>getCacheDataFileSpecifiedSize()</code>	キャッシュデータファイルの 1 ファイル当たりのファイルサイズ（キャッシュ定義ファイルに定義した値）を取得します。
<code>getCacheDataFileRemainingSize()</code>	現在書き込み中のキャッシュデータファイルについて、永続データを格納できる残りのファイルサイズを取得します。

メソッド名	説明
<code>getCacheDataFileSpecifiedNumber()</code>	キャッシュデータファイルのファイル数（キャッシュ定義ファイルに定義した値）を取得します。
<code>getCacheDataFileUnusedNumber()</code>	現在使用できる、未使用のキャッシュデータファイルの数を取得します。

## (4) getName()

### (a) 機能

ユーザファンクションを実行している EADs サーバ名（運用ディレクトリ名）を取得します。

### (b) 形式

```
public String getName()
```

### (c) 戻り値

ユーザファンクションを実行している EADs サーバ名（運用ディレクトリ名）が返却されます。

## (5) getAddress()

### (a) 機能

ユーザファンクションを実行している EADs サーバの IP アドレスおよびポート番号を取得します。

### (b) 形式

```
public InetSocketAddress getAddress()
```

### (c) 戻り値

サーバの IP アドレスとポート番号を示す `java.net.InetSocketAddress` インスタンスが返却されます。

## (6) getCacheNames()

### (a) 機能

ユーザファンクションを実行している EADs サーバで作成済みのキャッシュの、キャッシュ名の一覧を取得します。

### (b) 形式

```
public Set<String> getCacheNames()
```

## (c) 戻り値

ユーザファンクションを実行している EADs サーバで作成済みのキャッシュ名の一覧が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (7) `getCacheType()`

### (a) 機能

指定したキャッシュ名のキャッシュから、キャッシュタイプの情報を取得します。

### (b) 形式

```
public CacheType getCacheType(String cacheName)
                    throws EADsStoreException
```

### (c) パラメータ

`cacheName`

キャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

なお、作成していないキャッシュのキャッシュ名を指定した場合は不正となります。

### (d) 戻り値

キャッシュタイプの情報が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、`null` が返却されます。

### (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `EADsStoreException` (予期しないエラー)

## (8) `getEHeapSize()`

### (a) 機能

ユーザファンクションを実行している、EADs サーバが確保している Explicit ヒープサイズを取得します。

## (b) 形式

```
public long getEHeapSize()  
            throws EADsStoreException
```

## (c) 戻り値

EADs サーバが確保している Explicit ヒープサイズ（単位：バイト）が返却されます。

Function インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は 0 が返却されます。

## (d) 発生する例外

- `InternalServerException`（EADs サーバの内部エラー）
- `EADsStoreException`（予期しないエラー）

## (9) getEHeapUsageSize()

### (a) 機能

ユーザファンクションを実行している、EADs サーバが使用している Explicit ヒープサイズを取得します。

### (b) 形式

```
public long getEHeapUsageSize()  
            throws EADsStoreException
```

### (c) 戻り値

EADs サーバが使用している Explicit ヒープサイズ（単位：バイト）が返却されます。

Function インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は 0 が返却されます。

### (d) 発生する例外

- `InternalServerException`（EADs サーバの内部エラー）
- `EADsStoreException`（予期しないエラー）

## (10) getCacheDataFileSpecifiedSize()

### (a) 機能

キャッシュデータファイルの 1 ファイル当たりのファイルサイズ（キャッシュ定義ファイルに定義した値）（単位：バイト）を取得します。

## (b) 形式

```
public long getCacheDataFileSpecifiedSize(String cacheName)
        throws EADsStoreException
```

## (c) パラメータ

cacheName

キャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

なお、作成していないキャッシュのキャッシュ名を指定した場合は不正となります。

## (d) 戻り値

キャッシュデータファイルの 1 ファイル当たりのファイルサイズ（キャッシュ定義ファイルに定義した値）（単位：バイト）が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (e) 発生する例外

- `UserOperationException`（ユーザの操作不正）
- `EADsStoreException`（予期しないエラー）

# (11) `getCacheDataFileRemainingSize()`

## (a) 機能

現在書き込み中のキャッシュデータファイルについて、永続データを格納できる残りのファイルサイズ（単位：バイト）を取得します。

## (b) 形式

```
public long getCacheDataFileRemainingSize(String cacheName, String key)
        throws EADsStoreException
```

## (c) パラメータ

cacheName

キャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

なお、作成していないキャッシュのキャッシュ名を指定した場合は不正となります。

key

value に関連づける key を指定します。グループ名だけの指定もできます。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

なお、ユーザファンクションを実行中の EADs サーバに属していない key を指定した場合は不正となります。

## (d) 戻り値

現在書き込み中のキャッシュデータファイルについて、永続データを格納できる残りのファイルサイズ（単位：バイト）が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、0 が返却されます。

## (e) 発生する例外

- `UserOperationException`（ユーザの操作不正）
- `InternalServerException`（EADs サーバの内部エラー）
- `EADsStoreException`（予期しないエラー）

## (f) 注意事項

- 返却される残りのファイルサイズは、メソッド実行時点の情報です。そのため、残りのファイルサイズは、キャッシュ操作やコンパクションによって変動することがあります。
- このメソッドを実行している EADs サーバの情報を取得します。ほかの EADs サーバの情報は取得できません。
- このメソッドの戻り値で、キャッシュデータファイルに十分な空き領域を確認できた場合でも、データのコピー先の EADs サーバが容量不足となっていると、データの格納時にデータのコピー先の EADs サーバが縮退してしまいます。このメソッドの戻り値に加えて、データのコピー先の EADs サーバで使用できる容量も考慮して、データを格納できるか判断してください。

## (12) `getCacheDataFileSpecifiedNumber()`

### (a) 機能

キャッシュデータファイルのファイル数（キャッシュ定義ファイルに定義した値）を取得します。

### (b) 形式

```
public int getCacheDataFileSpecifiedNumber(String cacheName)
        throws EADsStoreException
```

## (c) パラメータ

cacheName

キャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

なお、作成していないキャッシュのキャッシュ名を指定した場合は不正となります。

## (d) 戻り値

キャッシュデータファイルのファイル数（キャッシュ定義ファイルに定義した値）が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、0 が返却されます。

## (e) 発生する例外

- `UserOperationException`（ユーザの操作不正）
- `EADsStoreException`（予期しないエラー）

## (13) `getCacheDataFileUnusedNumber()`

### (a) 機能

現在使用できる、未使用のキャッシュデータファイルの数を取得します。

### (b) 形式

```
public int getCacheDataFileUnusedNumber(String cacheName, String key)
        throws EADsStoreException
```

## (c) パラメータ

cacheName

キャッシュ名を指定します。

指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

なお、作成していないキャッシュのキャッシュ名を指定した場合は不正となります。

key

value に関連づける key を指定します。グループ名だけの指定もできます。

指定できるデータについては、「[15.2.2\(1\) keyとして指定できるデータ](#)」を参照してください。

なお、ユーザファンクションを実行中の EADs サーバに属していない key を指定した場合は不正となります。



## (d) 戻り値

現在使用できる、未使用のキャッシュデータファイルの数が返却されます。ただし、EADs サーバによって使用予約済みのファイル数は含まれません。

現在書き込み中のキャッシュデータファイルが、書き込みできる最後のキャッシュデータファイルの場合は、0 が返却されます。

Function インタフェースの `init()`、または `destroy()` 内でこのメソッドを呼び出した場合は、0 が返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerException` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (f) 注意事項

- 返却される残りのファイル数は、メソッド実行時点の情報です。そのため、残りのファイル数は、キャッシュ操作やコンパクションによって変動することがあります。
- このメソッドを実行している EADs サーバの情報を取得します。ほかの EADs サーバの情報は取得できません。
- このメソッドの戻り値で、キャッシュデータファイルに十分な空き領域を確認できた場合でも、データのコピー先の EADs サーバが容量不足となっていると、データの格納時にデータのコピー先の EADs サーバが縮退してしまいます。このメソッドの戻り値に加えて、データのコピー先の EADs サーバで使用できる容量も考慮して、データを格納できるか判断してください。

## (14) ServerInfo インタフェースの使用例

キャッシュデータファイルのデータを格納できる残りのファイルサイズを確認してから、データを put する場合の例を次に示します。

```
// ファイルに書き込むサイズ
int writeFileSize;
// メモリに書き込むサイズ
int writeMemorySize;

書き込むサイズの計算 (省略)

final String cacheName = context.getCacheName();
final ServerInfo serverInfo = context.getServerInfo();

final CacheType cacheType = serverInfo.getCacheType(cacheName);

// ディスク容量のチェック
if (cacheType == CacheType.DISK || cacheType == CacheType.TWOWAY) {
    // キャッシュデータファイルの残りを確認
```

```

    if (serverInfo.getCacheDataFileUnusedNumber(cacheName, KEY) == 0) {
        // 現在書き込み中のキャッシュデータファイルが最後の1つ
        if (serverInfo.getCacheDataFileRemainingSize(cacheName, KEY) < writeFileSize) {
            // キャッシュデータファイルに空きがない
            return "Error(CacheDataFile capacity)";
        }
    }
}

// メモリ容量のチェック
if (cacheType == CacheType.MEMORY || cacheType == CacheType.TWOWAY) {
    final long remainMemorySize = serverInfo.getEHeapSize() -
serverInfo.getEHeapUsageSize();
    if (remainMemorySize < writeMemorySize) {
        return "Error(EHeap capacity)";
    }
}
// 空きがあるためputを実行
context.getStore().getGroup().put(KEY, VALUE);

```

## 20.2.6 Store インタフェース

### (1) 説明

ユーザ関クションの実行時にデータを操作するインタフェースです。

### (2) インタフェース名

```
com.hitachi.software.xeads.func.store.Store
```

### (3) メソッド一覧

Store インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getGroup()</code>	ユーザ関クションの実行時に、EADs クライアントで指定したグループを操作するためのインスタンスを取得します。
<code>getGroup()</code> (グループ名指定)	指定したグループを操作するためのインスタンスを取得します。
<code>getGroupNameSet()</code>	キャッシュ内の最上位階層のグループ名の一覧を ASCII コードの昇順で取得します。
<code>containsKey()</code>	指定した key に関連づけられた value が、キャッシュに格納されているかどうかを取得します。

## (4) getGroup()

### (a) 機能

ユーザファンクションの実行時に、EADs クライアントで指定したグループを操作するためのインスタンスを取得します。

### (b) 形式

```
public Group getGroup()
```

### (c) 戻り値

ユーザファンクションの実行時に、EADs クライアントで指定したグループを操作するためのインスタンスが返却されます。

EADs クライアントでグループを指定しなかった場合は null が返却されます。

## (5) getGroup() (グループ名指定)

### (a) 機能

指定したグループを操作するためのインスタンスを取得します。

### (b) 形式

```
public Group getGroup(String groupName)
    throws EADsStoreException
```

### (c) パラメータ

groupName

グループ名を指定します。

指定できるデータについては、「[15.2.2\(1\) keyとして指定できるデータ](#)」を参照してください。

なお、ユーザファンクションを実行中の EADs サーバに属していないグループ名を指定した場合は不正となります。

### (d) 戻り値

指定したグループを操作するためのインスタンスが返却されます。

### (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerException` (EADs サーバの内部エラー)

- EADsStoreException (予期しないエラー)

## (6) getGroupNameSet()

### (a) 機能

キャッシュ内の最上位階層のグループ名の一覧を ASCII コードの昇順で取得します。

### (b) 形式

```
public Set<String> getGroupNameSet()  
                    throws EADsStoreException
```

### (c) 戻り値

キャッシュ内の最上位階層のグループ名の一覧 (ASCII コードの昇順) が返却されます。

### (d) 発生する例外

- EADsStoreException (予期しないエラー)

## (7) containsKey()

### (a) 機能

指定した key に関連づけられた value が、キャッシュに格納されているかどうかを取得します。

### (b) 形式

```
public boolean containsKey(String key)  
                    throws EADsStoreException
```

### (c) パラメータ

key

確認対象の key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

なお、ユーザファンクションを実行中の EADs サーバに、コピー元のデータが格納されていない key を指定した場合は不正となります。

### (d) 戻り値

指定した key に関連づけられた value が、キャッシュに格納されているかどうか返却されます。

true

value がキャッシュに格納されています。

false

value がキャッシュに格納されていません。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerError` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## 20.2.7 Group インタフェース

### (1) 説明

ユーザファンクションを実行中の EADs サーバに属する、グループを操作するためのインタフェースです。指定したグループに属さない key は操作できません。

### (2) インタフェース名

```
com.hitachi.software.xeads.func.store.Group
```

### (3) メソッド一覧

Group インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>toString()</code>	グループ名を取得します。
<code>keyIterator()</code>	指定したグループに属する key に、ASCII コードの昇順でアクセスする反復子を取得します。 反復子の開始位置は、グループの先頭の key となります。
<code>descendingKeyIterator()</code>	指定したグループに属する key に、ASCII コードの降順でアクセスする反復子を取得します。 反復子の開始位置は、グループの末尾の key となります。
<code>higherKeyIterator()</code>	指定したグループに属する key に、ASCII コードの昇順でアクセスする反復子を取得します。 反復子の開始位置は、ASCII コードの昇順で、指定された key の次に位置する key となります。
<code>lowerDescendingKeyIterator()</code>	指定したグループに属する key に、ASCII コードの降順でアクセスする反復子を取得します。 反復子の開始位置は、ASCII コードの降順で、指定された key の次に位置する key となります。

メソッド名	説明
put() (文字列指定)	value を key に関連づけて格納します。
put() (Key インタフェース指定)	value を key に関連づけて格納します。
create()	新規に key を格納する場合だけ、value を key に関連づけて格納します。
update() (文字列指定)	指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します (value を更新します)。
update() (Key インタフェース指定)	指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します (value を更新します)。
replace() (文字列指定)	指定した key に関連づけられている value を、条件として指定した value (comparativeValue) と比較します。value が一致している場合だけ、value を key に関連づけて格納します (value を置換します)。
replace() (Key インタフェース指定)	指定した key に関連づけられている value を、条件として指定した value (comparativeValue) と比較します。value が一致している場合だけ、value を key に関連づけて格納します (value を置換します)。
get() (文字列指定)	指定した key に関連づけられた value を取得します。
get() (Key インタフェース指定)	指定した key に関連づけられた value を取得します。
remove() (文字列指定)	指定した key, および key に関連づけられた value を削除します。
remove() (Key インタフェース指定)	指定した key, および key に関連づけられた value を削除します。
getLastUpdateTime() (文字列指定)	指定した key に関連づけられた value の最終更新時刻を取得します。
getLastUpdateTime() (Key インタフェース指定)	指定した key に関連づけられた value の最終更新時刻を取得します。
getGroupNameSet()	グループ名の一覧を ASCII コードの昇順で取得します。
getValueUsageSize() (文字列指定)	指定した key に関連づけられた value のメモリ使用量を取得します。
getValueUsageSize() (Key インタフェース指定)	指定した key に関連づけられた value のメモリ使用量を取得します。
getValueUsageSize()	下位のグループも含めて、グループに属する全 value のメモリ使用量の合計を取得します。

## (4) toString()

### (a) 機能

グループ名を取得します。

### (b) 形式

```
public String toString()
```

## (c) 戻り値

グループ名が返却されます。

## (5) keyIterator()

### (a) 機能

指定したグループに属する key に、ASCII コードの昇順でアクセスする反復子を取得します。

反復子の開始位置は、グループの先頭の key となります。

### (b) 形式

```
public java.util.Iterator<Key> keyIterator()  
                                throws EADsStoreException
```

## (c) 戻り値

グループに属する key に、ASCII コードの昇順でアクセスする反復子が返却されます。

## (d) 発生する例外

- EADsStoreException (予期しないエラー)

## (e) 注意事項

- タイミングによっては、反復子で取得した key に関連づけられた value が、削除されている場合があります。
- key は UTF-8 でエンコーディングしたバイト配列でソートされます。

## (6) descendingKeyIterator()

### (a) 機能

指定したグループに属する key に、ASCII コードの降順でアクセスする反復子を取得します。

反復子の開始位置は、グループの末尾の key となります。

### (b) 形式

```
public java.util.Iterator<Key> descendingKeyIterator()  
                                throws EADsStoreException
```

## (c) 戻り値

グループに属する key に、ASCII コードの降順でアクセスする反復子が返却されます。

## (d) 発生する例外

- EADsStoreException (予期しないエラー)

## (e) 注意事項

- タイミングによっては、反復子で取得した key に関連づけられた value が、削除されている場合があります。
- key は UTF-8 でエンコーディングしたバイト配列でソートされます。

## (7) higherKeyIterator()

### (a) 機能

指定したグループに属する key に、ASCII コードの昇順でアクセスする反復子を取得します。

反復子の開始位置は、ASCII コードの昇順で、指定された key の次に位置する key となります。

### (b) 形式

```
public java.util.Iterator<Key> higherKeyIterator(java.lang.String key)
    throws EADsStoreException
```

### (c) パラメータ

key

key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

### (d) 戻り値

グループに属する key に、ASCII コードの昇順でアクセスする反復子が返却されます。

### (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- EADsStoreException (予期しないエラー)

### (f) 注意事項

- タイミングによっては、反復子で取得した key に関連づけられた value が、削除されている場合があります。
- key は UTF-8 でエンコーディングしたバイト配列でソートされます。



## (8) lowerDescendingKeyIterator()

### (a) 機能

指定したグループに属する key に、ASCII コードの降順でアクセスする反復子を取得します。

反復子の開始位置は、ASCII コードの降順で、指定された key の次に位置する key となります。

### (b) 形式

```
public java.util.Iterator<Key> lowerDescendingKeyIterator(java.lang.String key)
    throws EADsStoreException
```

### (c) パラメータ

key

key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

### (d) 戻り値

グループに属する key に、ASCII コードの降順でアクセスする反復子が返却されます。

### (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- EADsStoreException (予期しないエラー)

### (f) 注意事項

- タイミングによっては、反復子で取得した key に関連づけられた value が、削除されている場合があります。
- key は UTF-8 でエンコーディングしたバイト配列でソートされます。

## (9) put() (文字列指定)

### (a) 機能

value を key に関連づけて格納します。

value の格納で問題が発生した場合は、EADsStoreException が返却されます。

## (b) 形式

```
public void put(java.lang.String key,  
               java.lang.Object value)  
               throws EADsStoreException
```

## (c) パラメータ

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerError` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (10) put() (Key インタフェース指定)

### (a) 機能

value を key に関連づけて格納します。

value の格納で問題が発生した場合は、`EADsStoreException` が返却されます。

### (b) 形式

```
public void put(Key key,  
               java.lang.Object value)  
               throws EADsStoreException
```

### (c) パラメータ

key

value に関連づける Key インタフェースの key を指定します。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerErrorException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (11) create()

### (a) 機能

新規に key を格納する場合だけ、value を key に関連づけて格納します。

value の格納で問題が発生した場合は、EADsStoreException が返却されます。

### (b) 形式

```
public void create(java.lang.String key,  
                  java.lang.Object value)  
    throws EADsStoreException
```

### (c) パラメータ

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerErrorException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (12) update() (文字列指定)

### (a) 機能

指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します (value を更新します)。

value の格納で問題が発生した場合は、EADsStoreException が返却されます。

## (b) 形式

```
public void update(java.lang.String key,  
                  java.lang.Object value)  
    throws EADsStoreException
```

## (c) パラメータ

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerError` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (13) update() (Key インタフェース指定)

### (a) 機能

指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します (value を更新します)。

value の格納で問題が発生した場合は、`EADsStoreException` が返却されます。

### (b) 形式

```
public void update(Key key,  
                  java.lang.Object value)  
    throws EADsStoreException
```

### (c) パラメータ

key

value に関連づける Key インタフェースの key を指定します。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

## (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerException` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (14) `replace()` (文字列指定)

### (a) 機能

指定した `key` に関連づけられている `value` を、条件として指定した `value` (`comparativeValue`) と比較します。`value` が一致している場合だけ、`value` を `key` に関連づけて格納します (`value` を置換します)。

`value` の置き換えで問題が発生した場合は、`EADsStoreException` が返却されます。

### (b) 形式

```
public void replace(java.lang.String key,
                   java.lang.Object value,
                   java.lang.Object comparativeValue)
    throws EADsStoreException
```

### (c) パラメータ

`key`

置き換える `value` に関連づけられた `key` を指定します。

指定できるデータについては、「[15.2.2\(1\) keyとして指定できるデータ](#)」を参照してください。

`value`

格納する `value` を指定します。

指定できるデータについては、「[15.2.2\(2\) valueとして指定できるデータ](#)」を参照してください。

`comparativeValue`

比較する `value` を指定します。

指定できるデータについては、「[15.2.2\(2\) valueとして指定できるデータ](#)」を参照してください。

### (d) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerException` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (15) replace() (Key インタフェース指定)

### (a) 機能

指定した key に関連づけられている value を、条件として指定した value (comparativeValue) と比較します。value が一致している場合だけ、value を key に関連づけて格納します (value を置換します)。

value の置き換えで問題が発生した場合は、EADsStoreException が返却されます。

### (b) 形式

```
public void replace(Key key,  
                   java.lang.Object value,  
                   java.lang.Object comparativeValue)  
    throws EADsStoreException
```

### (c) パラメータ

key

置き換える value に関連づけられた Key インタフェースの key を指定します。

value

格納する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

comparativeValue

比較する value を指定します。

指定できるデータについては、「[15.2.2\(2\) value として指定できるデータ](#)」を参照してください。

### (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (16) get() (文字列指定)

### (a) 機能

指定した key に関連づけられた value を取得します。

value の取得で問題が発生した場合は、EADsStoreException が返却されます。

## (b) 形式

```
public java.lang.Object get(java.lang.String key)
    throws EADsStoreException
```

## (c) パラメータ

key

取得する value に関連づけられた key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

## (d) 戻り値

key に関連づけられた value が返却されます。

指定した key に何も関連づけられていなかった場合は、null が返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerError` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (17) get() (Key インタフェース指定)

### (a) 機能

指定した key に関連づけられた value を取得します。

value の取得で問題が発生した場合は、`EADsStoreException` が返却されます。

### (b) 形式

```
public java.lang.Object get(Key key)
    throws EADsStoreException
```

### (c) パラメータ

key

取得する value に関連づけられた Key インタフェースの key を指定します。

### (d) 戻り値

key に関連づけられた value が返却されます。

指定した key に何も関連づけられていなかった場合は、null が返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerErrorException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (18) remove() (文字列指定)

### (a) 機能

指定した key, および key に関連づけられた value を削除します。

value の削除で問題が発生した場合は, EADsStoreException が返却されます。

### (b) 形式

```
public void remove(java.lang.String key)
                throws EADsStoreException
```

### (c) パラメータ

key

削除する value に関連づけられた key を指定します。

指定できるデータについては, 「15.2.2(1) key として指定できるデータ」を参照してください。

### (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerErrorException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (19) remove() (Key インタフェース指定)

### (a) 機能

指定した key, および key に関連づけられた value を削除します。

value の削除で問題が発生した場合は, EADsStoreException が返却されます。

### (b) 形式

```
public void remove(Key key)
                throws EADsStoreException
```



## (c) パラメータ

key

削除する value に関連づけられた Key インタフェースの key を指定します。

## (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerErrorException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (20) getLastUpdateTime() (文字列指定)

### (a) 機能

指定した key に関連づけられた value の最終更新時刻を取得します。

### (b) 形式

```
public long getLastUpdateTime(String key)
    throws EADsStoreException
```

## (c) パラメータ

key

key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

## (d) 戻り値

データの最終更新時刻 (1970 年 1 月 1 日 00:00:00 からの通算ミリ秒) が返却されます。

## (e) 発生する例外

- UserOperationException (ユーザの操作不正)
- InternalServerErrorException (EADs サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (21) getLastUpdateTime() (Key インタフェース指定)

### (a) 機能

指定した key に関連づけられた value の最終更新時刻を取得します。

## (b) 形式

```
public long getLastUpdateTime(Key key)
    throws EADsStoreException
```

## (c) パラメータ

key

value に関連づけられた Key インタフェースの key を指定します。

## (d) 戻り値

データの最終更新時刻（1970年1月1日00:00:00からの通算ミリ秒）が返却されます。

## (e) 発生する例外

- UserOperationException（ユーザの操作不正）
- InternalServerErrorException（EADs サーバの内部エラー）
- EADsStoreException（予期しないエラー）

## (22) getGroupNameSet()

### (a) 機能

グループ名の一覧を ASCII コードの昇順で取得します。

### (b) 形式

```
public Set<String> getGroupNameSet()
    throws EADsStoreException
```

### (c) 戻り値

直下の階層のグループ名の一覧（ASCII コードの昇順）が返却されます。

### (d) 発生する例外

- InternalServerErrorException（EADs サーバの内部エラー）
- EADsStoreException（予期しないエラー）

## (23) getValueUsageSize()（文字列指定）

### (a) 機能

指定した key に関連づけられた value のメモリ使用量を取得します。

## (b) 形式

```
public long getValueUsageSize(String key)
    throws EADsStoreException
```

## (c) パラメータ

key

key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

## (d) 戻り値

指定した key に関連づけられた value のメモリ使用量（単位：バイト）が返却されます。

value をメモリに格納しないキャッシュの場合は、0 が返却されます。

## (e) 発生する例外

- `UserOperationException`（ユーザの操作不正）
- `InternalServerError`（EADs サーバの内部エラー）
- `EADsStoreException`（予期しないエラー）

## (24) getValueUsageSize() (Key インタフェース指定)

### (a) 機能

指定した key に関連づけられた value のメモリ使用量を取得します。

### (b) 形式

```
public long getValueUsageSize(Key key)
    throws EADsStoreException
```

### (c) パラメータ

key

value に関連づけられた Key インタフェースの key を指定します。

### (d) 戻り値

指定した key に関連づけられた value のメモリ使用量（単位：バイト）が返却されます。

value をメモリに格納しないキャッシュの場合は、0 が返却されます。

## (e) 発生する例外

- `UserOperationException` (ユーザの操作不正)
- `InternalServerException` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (25) `getValueUsageSize()`

### (a) 機能

下位のグループも含めて、グループに属する全 value のメモリ使用量の合計を取得します。

### (b) 形式

```
public long getValueUsageSize()  
        throws EADsStoreException
```

### (c) 戻り値

グループに属する全 value のメモリ使用量 (単位: バイト) の合計が返却されます。

value をメモリに格納しないキャッシュの場合は、0 が返却されます。

### (d) 発生する例外

- `InternalServerException` (EADs サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## 20.2.8 Key インタフェース

### (1) 説明

ユーザファンクションで使用できる API 内で、key を表現するインタフェースです。

### (2) インタフェース名

```
com.hitachi.software.xeads.func.store.Key
```

### (3) メソッド一覧

Key インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
toString()	key を取得します。

## (4) toString()

### (a) 機能

key を取得します。

### (b) 形式

```
public String toString()
```

### (c) 戻り値

key が返却されます。

## 20.2.9 UserLogger インタフェース

### (1) 説明

ユーザログのインタフェースです。

### (2) インタフェース名

```
com.hitachi.software.xeads.common.UserLogger
```

### (3) メソッド一覧

UserLogger インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
log() (形式 1)	メッセージ ID, およびメッセージテキストを出力します。
log() (形式 2)	メッセージテキストだけを出力します。
putStackTrace() (形式 1)	メッセージ ID, およびスタックトレースを出力します。
putStackTrace() (形式 2)	スタックトレースだけを出力します。

## (4) log() (形式 1)

### (a) 機能

メッセージ ID, およびメッセージテキストを出力します。

### (b) 形式

```
public void log(java.lang.String messageID,  
               java.lang.String message)
```

### (c) パラメータ

messageID

メッセージ ID を指定します。

半角 21 文字まで指定できます。

全角文字や制御文字を指定すると、フォーマットが乱れる原因となるため指定しないでください。

null を指定した場合、空文字列が出力されます。

message

メッセージテキストを指定します。

制御文字を指定すると、フォーマットが乱れる原因となるため指定しないでください。

null を指定した場合、空文字列が出力されます。

## (5) log() (形式 2)

### (a) 機能

メッセージテキストだけを出力します。メッセージ ID には空文字列を出力します。

### (b) 形式

```
public void log(java.lang.String message)
```

### (c) パラメータ

message

メッセージテキストを指定します。

制御文字を指定すると、フォーマットが乱れる原因となるため指定しないでください。

null を指定した場合、空文字列が出力されます。

## (6) putStackTrace() (形式 1)

### (a) 機能

メッセージ ID, およびスタックトレースを出力します。

指定した例外の詳細メッセージと原因を出力します。

### (b) 形式

```
public void putStackTrace(java.lang.String messageID,  
                          Throwable cause)
```

### (c) パラメータ

messageID

メッセージ ID を指定します。

半角 21 文字まで指定できます。

全角文字や制御文字を指定すると、フォーマットが乱れる原因となるため指定しないでください。

null を指定した場合、空文字列が出力されます。

cause

例外オブジェクトを指定します。

null を指定した場合、空文字列が出力されます。

## (7) putStackTrace() (形式 2)

### (a) 機能

スタックトレースだけを出力します。メッセージ ID には空文字列を出力します。

指定した例外の詳細メッセージと原因を出力します。

### (b) 形式

```
public void putStackTrace(Throwable cause)
```

### (c) パラメータ

cause

例外オブジェクトを指定します。

null を指定した場合、空文字列が出力されます。

## 20.2.10 EADsStoreException クラス

### (1) 説明

Group インタフェースに対する操作が失敗した場合に返却される例外クラスです。

getStatusCode()によってエラーコードを取得し、エラーの内容を判別します。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.func.store.EADsStoreException
```

### (3) 形式

```
public class EADsStoreException
extends Exception
```

### (4) メソッド一覧

EADsStoreException クラスが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getStatusCode()</code>	発生した例外のエラーコードを取得します。

### (5) getStatusCode()

#### (a) 機能

発生した例外のエラーコードを取得します。

#### (b) 形式

```
public int getStatusCode()
```

#### (c) 戻り値

戻り値としてエラーコードが返却されます。エラーコードとエラーの内容、発生要因を次の表に示します。



エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
1000	EAD_ERROR_UNEXPECTED	EADsStoreException	予期しないエラーが発生しました。	プログラム内で予期しないエラーが発生しました。
1010	EAD_ERROR_INVALID_PARAMETER	UserOperationException	指定されたパラメタが不正です。	API の引数に不正なパラメタを指定しています。
1050	EAD_ERROR_NOT_SERIALIZABLE	UserOperationException	シリアライズ処理に失敗しました。	API の引数にシリアライズできないオブジェクトを指定しています。
1060	EAD_ERROR_NOT_DESERIALIZABLE	UserOperationException	デシリアライズ処理に失敗しました。	EADs サーバから取得したオブジェクトはデシリアライズできませんでした。 次の要因が考えられます。 <ul style="list-style-type: none"> <li>デシリアライズするために必要なオブジェクトを含む jar ファイルのパスが jar ファイルのマニフェストの Class-Path 属性に記載されていない。</li> <li>デシリアライズするために必要なオブジェクトを含む jar ファイルが &lt;運用ディレクトリ&gt;/app/lib ディレクトリ下に配置されていない。</li> </ul>
1130	EAD_ERROR_CACHE_NOT_CREATED	UserOperationException	作成していないキャッシュのキャッシュ名が指定されました。	作成していないキャッシュのキャッシュ名を指定しています。
1140	EAD_ERROR_CACHE_SETTING	UserOperationException	現在のキャッシュの設定では実行できない API が実行されました。	キャッシュから取得できない情報を取得する API を実行しました。
4000	EAD_ERROR_SERVER	InternalServerError	EADs サーバで内部エラーが発生しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>EADs サーバで内部エラーが発生した。</li> <li>指定した key のサイズがクラスタに格納できる key の最大サイズを超えている。</li> <li>データの格納先の空き容量が不足している。</li> </ul>

エラーコード	エラーコード定数	例外クラス	エラーの内容	発生要因
4020	EAD_ERROR_SERVER_EVENTHANDLER	InternalServerErrorException	データの更新操作および多重化は成功したが、ユーザーイベントハンドラ内の処理で例外が発生しました。	ユーザーイベントハンドラ内の処理で例外が発生しました。
4060	EAD_ERROR_SERVER_REPLACE_METHOD_NOT_MATCHED	InternalServerErrorException	Group インタフェースの replace() 実行時、すでに格納されている value が comparativeValue と一致しなかったため、value を格納できませんでした。	Group インタフェースの replace() で条件に指定した value とキャッシュ上の value が一致しませんでした。
4070	EAD_ERROR_SERVER_REPLACE_METHOD_KEY_NOT_EXIST	InternalServerErrorException	Group インタフェースの replace() 実行時、指定した key が存在しなかったため (key に関連づけられた value が存在しなかったため)、value を格納できませんでした。	Group インタフェースの replace() で指定した key に関連づけられた value が存在しなかったため、value の比較ができませんでした。
4080	EAD_ERROR_SERVER_CREATE_METHOD_KEY_EXIST	InternalServerErrorException	Group インタフェースの create() 実行時、すでに格納されている key が存在したため、value を格納できませんでした。	Group インタフェースの create() で指定した key に関連づけられた value がすでに格納されています。
4090	EAD_ERROR_SERVER_UPDATE_METHOD_KEY_NOT_EXIST	InternalServerErrorException	Group インタフェースの update() 実行時、格納されている key が存在しなかったため、value を格納できませんでした。	Group インタフェースの update() で指定した key に関連づけられた value が格納されていません。

## 20.2.11 InternalServerErrorException クラス

### (1) 説明

EADs サーバで内部エラーが発生した場合に返却される EADsStoreException のサブクラスです。

### (2) 継承関係

```

java.lang.Object
├─ java.lang.Throwable
│   └─ java.lang.Exception
│       └─ com.hitachi.software.xeads.func.store.EADsStoreException
│           └─ com.hitachi.software.xeads.func.store.InternalServerErrorException

```

### (3) 形式

```
public class InternalServerErrorException
extends EADsStoreException
```

## 20.2.12 UserOperationException クラス

### (1) 説明

不正なユーザ操作によってエラーが発生した場合に返却される EADsStoreException のサブクラスです。

### (2) 継承関係

```
java.lang.Object
├─java.lang.Throwable
│   └─java.lang.Exception
│       └─com.hitachi.software.xeads.func.store.EADsStoreException
│           └─com.hitachi.software.xeads.func.store.UserOperationException
```

### (3) 形式

```
public class UserOperationException
extends EADsStoreException
```

## 20.2.13 列挙型 CacheType

### (1) 説明

キャッシュタイプを表す列挙型です。キャッシュタイプについては、「[2.3.1 キャッシュの種類 \(キャッシュタイプ\)](#)」を参照してください。

### (2) 列挙型名

```
com.hitachi.software.xeads.common.CacheType
```

### (3) 形式

```
public enum CacheType {
MEMORY,
DISK,
```

```
TWOWAY  
}
```

#### (4) 列挙定数

定数	説明
MEMORY	メモリキャッシュ
DISK	ディスクキャッシュ
TWOWAY	2Way キャッシュ

## 20.3 ユーザイベントハンドラで使用できる API

ユーザイベントハンドラで使用できる API を次の表に示します。

表 20-3 ユーザイベントハンドラで使用できる API

項番	インタフェース名, アノテーション名, または列挙型名	機能	パッケージ名
1	<a href="#">EventHandler</a>	EADs サーバで使用されるユーザイベントハンドラが実装する必要があるインタフェースです。	com.hitachi.software.xeads.event.EventHandler
2	<a href="#">EventContext</a>	EADs サーバ, EADs クライアント, および発生したイベントに関する情報を, ユーザイベントハンドラに渡すインタフェースです。	com.hitachi.software.xeads.event.EventContext
3	<a href="#">ClientInfo</a> *	実行した EADs クライアントの情報 (PRF トレースに出力される RootAP 情報に相当する情報) を取得するインタフェースです。	com.hitachi.software.xeads.common.ClientInfo
4	<a href="#">UserLogger</a> *	ユーザログのインタフェースです。	com.hitachi.software.xeads.common.UserLogger
5	<a href="#">OperationFilter</a>	イベントが発生したデータ操作の種類に関して, ユーザイベントハンドラが動作する条件を指定するアノテーションです。	com.hitachi.software.xeads.event.annotation.OperationFilter
6	<a href="#">DataFilter</a>	イベントが発生したデータのキャッシュ名, グループ名, または要素名に関して, ユーザイベントハンドラが動作する条件を指定するアノテーションです。	com.hitachi.software.xeads.event.annotation.DataFilter
7	<a href="#">Priority</a>	ユーザイベントハンドラを実行する優先度を指定するアノテーションです。	com.hitachi.software.xeads.event.annotation.Priority
8	<a href="#">OperationType</a>	イベントが発生したデータ操作の種類を表す列挙型です。	com.hitachi.software.xeads.event.OperationType
9	<a href="#">MethodType</a>	実行した API の種類を表す列挙型です。	com.hitachi.software.xeads.event.MethodType
10	<a href="#">PriorityLevel</a>	ユーザイベントハンドラを実行する優先度を表す列挙型です。	com.hitachi.software.xeads.event.PriorityLevel

注※

[ClientInfo](#) インタフェースについては, 「[20.2.4 ClientInfo インタフェース](#)」を参照してください。

[UserLogger](#) インタフェースについては, 「[20.2.9 UserLogger インタフェース](#)」を参照してください。

## 20.3.1 EventHandler インタフェース

### (1) 説明

EADs サーバで使用するユーザイベントハンドラが実装する必要があるインタフェースです。

ユーザイベントハンドラの初期化時にインスタンスを生成します。インスタンスは単一で、EADs サーバの終了処理まで生存します。

### (2) インタフェース名

```
com.hitachi.software.xeads.event.EventHandler
```

### (3) メソッド一覧

EventHandler インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>init()</code>	ユーザイベントハンドラの初期化処理を実装します。
<code>execute()</code>	ユーザイベントハンドラの処理を実装します。
<code>destroy()</code>	ユーザイベントハンドラの終了処理を実装します。

### (4) `init()`

#### (a) 機能

ユーザイベントハンドラの初期化処理を実装します。

EADs サーバ起動時のユーザイベントハンドラの初期化時に呼び出されます。

#### (b) 形式

```
public void init(EventContext context)
```

#### (c) パラメータ

`context`

ユーザイベントハンドラの実行に必要な情報を取得します。

ユーザイベントハンドラの実行に必要な情報については、「[20.3.2 EventContext インタフェース](#)」を参照してください。

## (5) execute()

### (a) 機能

ユーザイベントハンドラの処理を実装します。

EADs クライアント、ユーザファンクション、またはコマンド (`eztool put`, `eztool remove`) からデータを操作する際に呼び出されます。

### (b) 形式

```
public void execute(EventContext context)
    throws Exception
```

### (c) パラメータ

context

ユーザイベントハンドラの実行に必要な情報を取得します。

ユーザイベントハンドラの実行に必要な情報については、「[20.3.2 EventContext インタフェース](#)」を参照してください。

### (d) 発生する例外

- Exception (ユーザイベントハンドラの内部エラー)

### (e) 注意事項

ユーザイベントハンドラで例外やエラーを catch しない場合、ユーザイベントハンドラの内部エラーが発生すると、ユーザイベントハンドラの実行契機となったリクエストがエラーとなります。

## (6) destroy()

### (a) 機能

ユーザイベントハンドラの終了処理を実装します。

EADs サーバ終了時のユーザイベントハンドラの終了時に呼び出されます。

### (b) 形式

```
public void destroy(EventContext context)
```

### (c) パラメータ

context

ユーザイベントハンドラの実行に必要な情報を取得します。

ユーザイベントハンドラの実行に必要な情報については、「[20.3.2 EventContext インタフェース](#)」を参照してください。

## 20.3.2 EventContext インタフェース

### (1) 説明

EADs サーバ、EADs クライアント、および発生したイベントに関する情報を、ユーザイベントハンドラに渡すインタフェースです。

インスタンスは、ユーザイベントハンドラを呼び出す直前に生成され、ユーザイベントハンドラの処理完了後に解放されます。

1 回のデータ操作で複数のユーザイベントハンドラを呼び出す場合、ユーザイベントハンドラごとにインスタンスが生成されます。

### (2) インタフェース名

```
com.hitachi.software.xeads.event.EventContext
```

### (3) メソッド一覧

EventContext インタフェースが提供するメソッド一覧を次の表に示します。

メソッド名	説明
<code>getServerName()</code>	ユーザイベントハンドラを実行している EADs サーバ名（運用ディレクトリ名）を取得します。
<code>getCacheName()</code>	API を実行したキャッシュ名を取得します。
<code>getGroupName()</code>	実行した API に指定した key に含まれるグループ名を取得します。
<code>getKey()</code>	実行した API に指定した key を取得します。
<code>getValue()</code>	データ操作完了後の value を取得します。
<code>getLastUpdateTime()</code>	データの更新時刻を取得します。
<code>isNewData()</code>	新規に追加されたデータかどうかを取得します。
<code>getClientInfo()</code>	API を実行した EADs クライアントの情報（PRF トレースに出力される RootAP 情報に相当する情報）を取得します。
<code>getMethod()</code>	実行した API の種類を取得します。
<code>getLogger()</code>	ユーザログを出力するロガーを取得します。



## (4) `getServerName()`

### (a) 機能

ユーザイベントハンドラを実行している EADs サーバ名（運用ディレクトリ名）を取得します。

### (b) 形式

```
public String getServerName()
```

### (c) 戻り値

ユーザイベントハンドラを実行している EADs サーバ名（運用ディレクトリ名）が返却されます。

## (5) `getCacheName()`

### (a) 機能

API を実行したキャッシュ名を取得します。

### (b) 形式

```
public String getCacheName()
```

### (c) 戻り値

API を実行したキャッシュ名が返却されます。

EventHandler インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (6) `getGroupName()`

### (a) 機能

実行した API に指定した key に含まれるグループ名を取得します。

### (b) 形式

```
public String getGroupName()
```

### (c) 戻り値

実行した API に指定した key に含まれるグループ名が返却されます。

EventHandler インタフェースの `init()`、もしくは `destroy()`内でこのメソッドを呼び出した場合、または、実行した API に指定した `key` にグループ名が含まれていない場合は、`null` が返却されます。

## (7) getKey()

### (a) 機能

実行した API に指定した `key` を取得します。

### (b) 形式

```
public String getKey()
```

### (c) 戻り値

実行した API に指定した `key` が返却されます。

EventHandler インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (8) getValue()

### (a) 機能

データ操作完了後の `value` を取得します。

ユーザイベントハンドラを実行する契機となったリクエストで、操作が完了した時点の `value` を取得します。

このメソッドを呼び出すまでに、ほかのリクエストによって更新された `value` を取得することはできません。

### (b) 形式

```
public Object getValue()  
throws IOException, ClassNotFoundException
```

### (c) 戻り値

データ操作完了後の `value` が返却されます。

次の場合は `null` が返却されます。

- EventHandler インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合
- `remove` 操作の際にこのメソッドを呼び出した場合
- リクエストの操作が完了する前にこのメソッドを呼び出した場合

## (d) 発生する例外

- IOException (デシリアライズ処理内で問題が発生)
- ClassNotFoundException (シリアライズされたオブジェクトのクラスが見つからない)

## (9) getLastUpdateTime()

データの更新時刻を取得します。

ユーザイベントハンドラを実行する契機となったリクエストで、操作が完了した時点のデータの更新時刻を取得します。

このメソッドを呼び出すまでに、ほかのリクエストによって更新されたデータの更新時刻を取得することはできません。

### (a) 形式

```
public long getLastUpdateTime()
```

### (b) 戻り値

データの更新時刻 (1970 年 1 月 1 日 00:00:00 からの通算ミリ秒) が返却されます。

次の場合は 0 が返却されます。

- EventHandler インタフェースの init(), または destroy()内でこのメソッドを呼び出した場合
- remove 操作の際にこのメソッドを呼び出した場合
- リクエストの操作が完了する前にこのメソッドを呼び出した場合

## (10) isNewData()

新規に追加されたデータかどうかを取得します。

### (a) 形式

```
public boolean isNewData()
```

### (b) 戻り値

新規に追加されたデータかどうか返却されます。

true

新規に追加されたデータです。

false

既存のデータの更新または削除です。

EventHandler インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は、`false` が返却されます。

## (11) `getClientInfo()`

### (a) 機能

API を実行した EADs クライアントの情報 (PRF トレースに出力される RootAP 情報に相当する情報) を取得します。

ユーザファンクションから API を実行した場合は、Cache クラスの `executeFunction()` を実行した EADs クライアントの情報を取得します。

コマンド (`eztool put`, `eztool remove`) から API を実行した場合は、コマンドを実行したプロセスの情報を取得します。

### (b) 形式

```
public ClientInfo getClientInfo()
```

### (c) 戻り値

API を実行した EADs クライアントの情報が返却されます。

EADs クライアントの情報については、「[20.2.4 ClientInfo インタフェース](#)」を参照してください。

EventHandler インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (12) `getMethod()`

### (a) 機能

実行した API の種類を取得します。

### (b) 形式

```
public MethodType getMethod()
```

### (c) 戻り値

実行した API の種類が返却されます。

EventHandler インタフェースの `init()`、または `destroy()`内でこのメソッドを呼び出した場合は、`null` が返却されます。

## (13) getLogger()

### (a) 機能

ユーザログを出力するロガーを取得します。

### (b) 形式

```
public UserLogger getLogger()
```

### (c) 戻り値

ユーザログを出力するロガーが返却されます。

## 20.3.3 OperationFilter アノテーション

### (1) 説明

イベントが発生したデータ操作の種類に関して、ユーザイベントハンドラが動作する条件を指定するアノテーションです。

OperationFilter アノテーションは EventHandler インタフェースを実装したクラスにだけ付けることができます。

OperationFilter アノテーションを付けていない場合、ユーザイベントハンドラは value に {OperationType.PUT, OperationType.REMOVE} が設定された場合と同様に動作します。

### (2) アノテーション名

```
com.hitachi.software.xeads.event.annotation.OperationFilter
```

### (3) 継承関係

OperationFilter アノテーションの内容はサブクラスに継承されます。

EventHandler インタフェースを実装したクラスに OperationFilter アノテーションの宣言がない場合、そのクラスのスーパークラスの OperationFilter アノテーションの内容が継承されます。

### (4) 形式

```
public @interface OperationFilter {  
    public OperationType[] value() default {OperationType.PUT, OperationType.REMOVE};  
}
```

## (5) 要素一覧

要素名	説明
OperationType[] value	ユーザイベントハンドラが動作するデータ操作の種類のセットです。 指定値については、「 <a href="#">20.3.6 列挙型 OperationType</a> 」を参照してください。デフォルト値は{OperationType.PUT,OperationType.REMOVE}です。 条件は配列として複数指定でき、どれかの条件を満たす場合、ユーザイベントハンドラが動作します。 指定値が重複する場合、2つ目以降の重複した指定値は無視されます。

## 20.3.4 DataFilter アノテーション

### (1) 説明

イベントが発生したデータのキャッシュ名、グループ名、または要素名に関して、ユーザイベントハンドラが動作する条件を指定するアノテーションです。

DataFilter アノテーションは EventHandler インタフェースを実装したクラスにだけ付けることができます。

DataFilter アノテーションを付けていない場合、キャッシュ名、グループ名、または要素名に関係なく、ユーザイベントハンドラは動作します。

### (2) アノテーション名

```
com.hitachi.software.xeads.event.annotation.DataFilter
```

### (3) 継承関係

DataFilter アノテーションの内容はサブクラスに継承されます。

EventHandler インタフェースを実装したクラスに DataFilter アノテーションの宣言がない場合、そのクラスのスーパークラスの DataFilter アノテーションが継承されます。

### (4) 形式

```
public @interface DataFilter {  
    public String[] cache() default {};  
    public String[] group() default {};  
    public String[] element() default {};  
}
```

## (5) 要素一覧

要素名	説明
String[] cache	ユーザイベントハンドラが動作するキャッシュ名のセットです。 デフォルト値は{}です。{}を指定した場合、データのキャッシュ名に関係なく、ユーザイベントハンドラは動作します。 キャッシュ名は配列として複数指定でき、指定したどれかのキャッシュ名と操作対象のデータが一致すると、ユーザイベントハンドラが動作します。 キャッシュ名が重複する場合、2つ目以降の重複したキャッシュ名は無視されます。
String[] group	ユーザイベントハンドラが動作するグループ名のセットです。 デフォルト値は{}です。{}を指定した場合、データのグループ名に関係なく、ユーザイベントハンドラは動作します。 グループ名は配列として複数指定でき、指定したどれかのグループ名に操作対象のデータが含まれていると、ユーザイベントハンドラが動作します。 グループ名が重複する場合、2つ目以降の重複したグループ名は無視されます。
String[] element	ユーザイベントハンドラが動作する要素名のセットです。 デフォルト値は{}です。{}を指定した場合、データの要素名に関係なく、ユーザイベントハンドラは動作します。 要素名は配列として複数指定でき、指定したどれかの要素名に操作対象のデータが含まれていると、ユーザイベントハンドラが動作します。 要素名が重複する場合、2つ目以降の重複した要素名は無視されます。

## 20.3.5 Priority アノテーション

### (1) 説明

ユーザイベントハンドラを実行する優先度を指定するアノテーションです。

Priority アノテーションは EventHandler インタフェースを実装したクラスにだけ付けることができます。

Priority アノテーションを付けていない場合、ユーザイベントハンドラは PriorityLevel.NORMAL が設定された場合と同様に動作します。

### (2) アノテーション名

```
com.hitachi.software.xeads.event.annotation.Priority
```

### (3) 継承関係

Priority アノテーションの内容はサブクラスに継承されます。

EventHandler インタフェースを実装したクラスに Priority アノテーションの宣言がない場合、そのクラスのスーパークラスの Priority アノテーションが継承されます。

## (4) 形式

```
public @interface Priority {  
    public PriorityLevel value() default PriorityLevel.NORMAL;  
}
```

## (5) 要素一覧

要素名	説明
PriorityLevel value	ユーザイベントハンドラを実行する優先度です。 指定値については、「 <a href="#">20.3.8 列挙型 PriorityLevel</a> 」を参照してください。デフォルト値は PriorityLevel.NORMAL です。

## 20.3.6 列挙型 OperationType

### (1) 説明

イベントが発生したデータ操作の種類を表す列挙型です。

### (2) 列挙型名

```
com.hitachi.software.xeads.event.OperationType
```

### (3) 形式

```
public enum OperationType {  
    PUT,  
    REMOVE  
}
```

### (4) 列挙定数

定数	説明
PUT	データの格納処理 該当するデータの格納処理を次に示します。 <ul style="list-style-type: none"><li>Cache クラス, または Group インタフェースのメソッド (put(), replace(), create(), update())</li><li>eztool put コマンド</li></ul>
REMOVE	データの削除処理 該当するデータの削除処理を次に示します。 <ul style="list-style-type: none"><li>Cache クラス, または Group インタフェースのメソッド (remove())</li></ul>



定数	説明
	<ul style="list-style-type: none"> <li>• <code>eztool remove</code> コマンド</li> </ul>

## 20.3.7 列挙型 MethodType

### (1) 説明

実行した API の種類を表す列挙型です。

### (2) 列挙型名

```
com.hitachi.software.xeads.event.MethodType
```

### (3) 形式

```
public enum MethodType {
    PUT,
    REMOVE,
    REPLACE,
    CREATE,
    UPDATE
}
```

### (4) 列挙定数

定数	説明
PUT	データの格納 ( <code>put()</code> )
REMOVE	データの削除 ( <code>remove()</code> )
REPLACE	データの置換 ( <code>replace()</code> )
CREATE	データの新規格納 ( <code>create()</code> )
UPDATE	データの更新 ( <code>update()</code> )

## 20.3.8 列挙型 PriorityLevel

### (1) 説明

ユーザイベントハンドラを実行する優先度を表す列挙型です。

## (2) 列挙型名

```
com.hitachi.software.xeads.event.PriorityLevel
```

## (3) 形式

```
public enum PriorityLevel {  
    HIGHEST,  
    HIGH,  
    NORMAL,  
    LOW,  
    LOWEST  
}
```

## (4) 列挙定数

定数	優先度
HIGHEST	最も高い
HIGH	高い
NORMAL	通常
LOW	低い
LOWEST	最も低い

# 21

## クライアント AP を作成する (C 言語)

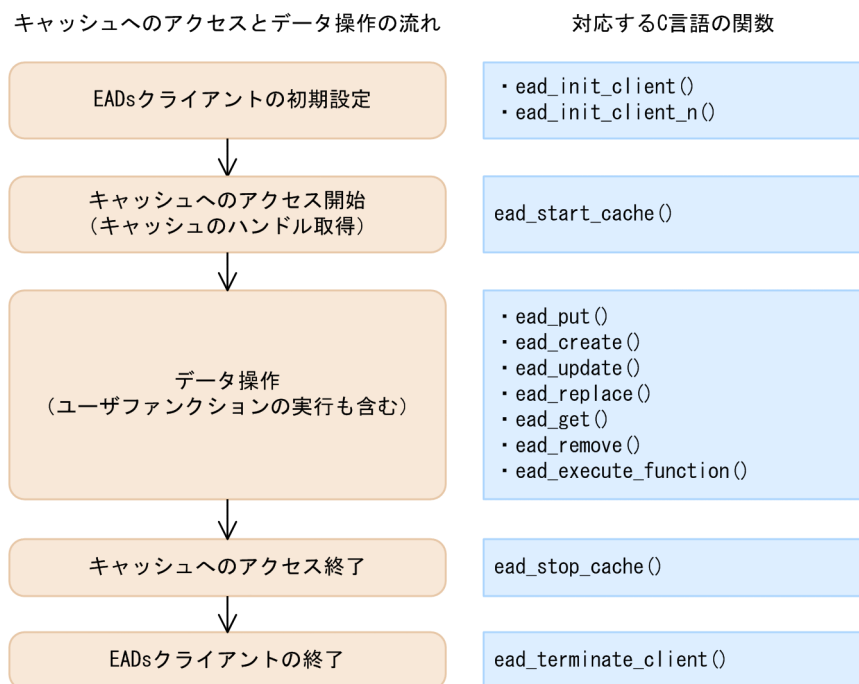
この章では、C 言語を使用したクライアント AP の作成方法について説明します。

## 21.1 ソースプログラムを作成する (C 言語)

ここでは、キャッシュへのアクセスとデータ操作の流れと、ソースプログラムの作成例について説明します。

### 21.1.1 キャッシュへのアクセスとデータ操作の流れ

キャッシュへのアクセスとデータ操作の流れについて説明します。



#### (1) C 言語を使用する場合のソースプログラムの例

C 言語を使用する場合のソースプログラムの例 (key と value を格納する場合) を次に示します。

```
#include <stdio.h>
#include <string.h>

#include <eads.h>

int main(int argc, char **argv) {
    int ret = 0;
    int error_code = 0;
    char CONFPATH[] = "./conf/eads_sample_client.properties";
    char CACHENAME[] = "cache1";
    char KEY[] = "key1";
    char VALUE[] = "value1";
    EAD_CACHE_MANAGER* cmp = NULL;
    EAD_CACHE *cp = NULL;
    ead_value_element value_element;

    value_element.value = (void *)VALUE;
```

```

value_element.value_size = strlen(VALUE) + 1;

/* EADsクライアントの初期設定をする */
cmp = ead_init_client(CONFPATH, &error_code);
printf("ead_init_client() done. (error_code = %d)\n", error_code);
if (error_code != EAD_OK) {
    return 1;
}

/* キャッシュへのアクセスを開始する */
cp = ead_start_cache(cmp, CACHENAME, &error_code);
printf("ead_start_cache() done. (error_code = %d)\n", error_code);
if (error_code != EAD_OK) {
    /* EADsクライアントを終了する場合 */
    goto ERR;
}

/* keyとvalueを格納する */
ead_put(cp, KEY, &value_element, &error_code);
printf("ead_put() done. (error_code = %d)\n", error_code);
if (error_code != EAD_OK) {
    /* EADsクライアントを終了する場合 */
    goto ERR;
}

/* キャッシュへのアクセスを終了する */
ead_stop_cache(cp, &error_code);
cp = NULL;
printf("ead_stop_cache() done. (error_code = %d)\n", error_code);
if (error_code != EAD_OK) {
    /* EADsクライアントを終了する場合 */
    goto ERR;
}

/* EADsクライアントを終了する */
ead_terminate_client(cmp, &error_code);
cmp = NULL;
printf("ead_terminate_client() done. (error_code = %d)\n", error_code);
if (error_code != EAD_OK) {
    /* EADsクライアントを終了する場合 */
    goto ERR;
}

return 0;

ERR:
if(cp != NULL) {
    /* キャッシュへのアクセスを終了する */
    ead_stop_cache(cp, &error_code);
    cp = NULL;
    printf("ead_stop_cache() done. (error_code = %d)\n", error_code);
}

if(cmp != NULL) {
    /* EADsクライアントを終了する */
    ead_terminate_client(cmp, &error_code);
    cmp = NULL;
    printf("ead_terminate_client() done. (error_code = %d)\n", error_code);
}

```

```
    }  
  
    return 1;  
}
```

## (2) EADs クライアントの初期設定をする

EADs クライアントの初期設定をするには、`ead_init_client()`または`ead_init_client_n()`を使用します。また、キャッシュを管理するキャッシュマネージャのハンドル（ポインタ）を取得します。

クライアント定義に従って、接続先 EADs サーバなどが設定されます。

複数のクラスタと同時に接続するなど、設定の異なる複数のキャッシュマネージャを同時に使用したい場合は、クライアント定義を変更して`ead_init_client()`または`ead_init_client_n()`を複数回実行します。複数回実行した場合は、取得した各キャッシュマネージャのハンドルに対して`ead_terminate_client()`を実行して、EADs クライアントを終了してください。

## (3) キャッシュへのアクセスを開始する

EADs クライアントの初期設定が完了したら、キャッシュへのアクセスを開始します。

キャッシュへのアクセスを開始するには、`ead_start_cache()`を使用して、キャッシュへのアクセスを制御するためのハンドル（ポインタ）を取得します。このハンドルを使用することによって、キャッシュにアクセスできるようになります。

## (4) key と value を格納する

キャッシュに key と value を格納するには、`ead_put()`を使用します。

`ead_put()`には、`ead_start_cache()`で取得したキャッシュのハンドルを指定します。また、キャッシュに格納する key と value 情報（value と value のサイズ）を指定します。value 情報は `ead_value_element` 構造体として用意します。

## (5) value を取得する

キャッシュから value を取得するには、`ead_get()`を使用します。

`ead_get()`には、`ead_start_cache()`で取得したキャッシュのハンドルを指定します。また、取得する value に関連づけられた key を指定します。

`ead_get()`によって value の取得が成功すると、key に関連づけられた value 情報が `ead_value_element` 構造体として返却されます。

なお、戻り値として返却された value のメモリ領域は、自動的に解放されません。メモリ領域の解放については、「(10)戻り値として返却されたメモリ領域を解放する」を参照してください。

value を取得する場合のソースプログラムの例を次に示します。

## ソースプログラムの例 (value を取得する場合)

```
{
    /* valueを取得する */
    ead_value_element ret_value;
    char KEY[] = "key1";
    ret_value = ead_get(cp, KEY, &error_code);
    printf("ead_get() done. (error_code = %d)%n", error_code);
    if (error_code != EAD_OK) {
        /* EADsクライアントを終了する場合 */
        goto ERR;
    }
    /* 取得成功, 取得したvalueは使用後に解放する */
    freeValue(&ret_value);
}
```

## (6) key と value を削除する

キャッシュから指定した key, および key に関連づけられた value を削除するには, `ead_remove()` を使用します。

`ead_remove()`には, `ead_start_cache()`で取得したキャッシュのハンドルを指定します。また, 削除する value に関連づけられた key を指定します。

## (7) ユーザファンクションを実行する

### (a) グループ指定による実行方法

グループを指定してユーザファンクションを実行するには, `ead_execute_function()`を使用します。

`ead_execute_function()`には, `ead_start_cache()`で取得したキャッシュのハンドルを指定します。

また, ユーザファンクションを実行するグループ名と, 実行するユーザファンクション名を指定します。

また, ユーザファンクションに指定する引数を `ead_object` 構造体として指定します。

`ead_execute_function()`によってユーザファンクションが実行されると, ユーザファンクションの実行結果が `ead_object` 構造体として返却されます。

### (b) EADs サーバ指定による実行方法

EADs サーバを指定してユーザファンクションを実行するには, `ead_execute_node_function()`を使用します。

`ead_execute_node_function()`には, `ead_start_cache()`で取得したキャッシュのハンドルを指定します。

また, ユーザファンクションを実行する EADs サーバを `ead_node` 構造体として, ユーザファンクションに指定する引数を `ead_object` 構造体として指定します。

また、実行するユーザファンクション名を指定します。

`ead_execute_node_function()`によってユーザファンクションが実行されると、ユーザファンクションの実行結果が `ead_object` 構造体として返却されます。

## (8) キャッシュへのアクセスを終了する

キャッシュへのアクセスを終了するには、`ead_stop_cache()`を使用します。

`ead_stop_cache()`には、`ead_start_cache()`で取得したキャッシュのハンドルを指定します。

## (9) EADs クライアントを終了する

EADs クライアントを終了するには、`ead_terminate_client()`を使用します。

`ead_terminate_client()`には、`ead_init_client()`または `ead_init_client_n()`で取得したキャッシュマネージャのハンドルを指定します。

## (10) 戻り値として返却されたメモリ領域を解放する

`ead_get()`の戻り値として返却された `value` のメモリ領域は、自動的に解放されません。次のような関数を定義して、AP で解放してください。

```
/* 戻り値として返却されたメモリ領域を解放する */
void freeValue(ead_value_element *value) {
    if (value->value != NULL) {
        free(value->value);
        value->value = NULL;
    }
    value->value_size = 0;
}
```

また、次の場合も同様に、メモリ領域を AP で解放してください。

- ユーザファンクションを実行し、ユーザファンクションの実行結果として返却された `ead_object` 構造体の `object` メンバが `NULL` でない場合
- `ead_get_nodelist()`、または `ead_get_slave_nodelist()`の戻り値として返却された `ead_nodelist` 構造体の `nodes` メンバが `NULL` でない場合



## 21.2 クライアント AP 作成上の留意事項 (C 言語)

---

クライアント AP 作成上の留意事項について説明します。

### 21.2.1 EADs クライアントの初期設定に関する留意事項

EADs クライアントの初期設定に関する留意事項を次に示します。

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- `ead_init_client_n()` で EADs クライアント名に空文字列を指定した場合、`ead_init_client()` と同じ動作をします。EADs クライアント名とログ出力先の関係については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- クライアント定義を変更する場合は、次の順に行います。
  1. `ead_terminate_client()` で、EADs クライアントを終了します。
  2. クライアント定義ファイルを更新します。
  3. `ead_init_client()` または `ead_init_client_n()` で、再度 EADs クライアントの初期設定をします。

### 21.2.2 キャッシュへのアクセス開始に関する留意事項

キャッシュへのアクセス開始に関する留意事項を次に示します。

- キャッシュへのアクセスを開始する場合は、`ead_start_cache()` を実行してください。`ead_start_cache()` は、`ead_stop_cache()` と対となるように実行してください。
- キャッシュ名には、あらかじめ `eztool createcache` コマンドで作成した名称を指定します。
- 指定したキャッシュ名が存在しない場合、`EAD_ERROR_NET_CLUSTERINFO` が返却されます。

### 21.2.3 データ操作に関する留意事項

データ操作に関する留意事項を次に示します。

- 指定した `key` がすでにキャッシュに格納されている場合、`ead_put()` は無条件に `value` を更新します。無条件に `value` を更新したくない場合は、次の関数を使用してください。
  - `ead_create()`  
新規に `key` を格納する場合だけ、`value` を `key` に関連づけて格納します。
  - `ead_update()`

指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します。

- ead\_replace()  
指定した key に関連づけられている value を、条件として指定した value 情報と比較します。value が一致している場合だけ、value を key に関連づけて格納します。
- ead\_get()の実行時、次の場合には ead\_value\_element 構造体の value メンバに NULL が返却されません。
  - key がキャッシュに存在しない場合
  - value 情報の取得に失敗した場合
  - value のデータサイズが 0 バイトの場合 (Java 言語の API で格納された、長さ 0 のバイト配列を取得した場合)
- ead\_get()の実行時に、戻り値として返却された value のメモリ領域は自動的に解放されません。AP で解放してください。詳細については、「[21.1.1\(10\)戻り値として返却されたメモリ領域を解放する](#)」を参照してください。
- ead\_replace()の実行時に、指定した key が存在しない場合は、EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST が返却されます。
- ead\_replace()の実行時に、指定した key に関連づけられた value が、比較条件に指定した value と一致しなかった場合、EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED が返却されます。
- ead\_create()の実行時に、指定した key がすでに存在する場合、EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST が返却されます。
- ead\_update()の実行時に、指定した key が存在しない場合、EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST が返却されます。

## 21.2.4 キャッシュへのアクセス終了に関する留意事項

キャッシュへのアクセス終了に関する留意事項を次に示します。

- キャッシュへのアクセスを終了する場合は、ead\_stop\_cache()を実行してください。ead\_stop\_cache() は、ead\_start\_cache()と対となるように実行してください。
- 一度、ead\_stop\_cache()でアクセスを終了したキャッシュに再度アクセスする場合は、再度、ead\_start\_cache()を発行してください。

## 21.2.5 EADs クライアントの終了に関する留意事項

EADs クライアントの終了に関する留意事項を次に示します。これらの留意事項は必ず守ってください。守らない場合、セグメンテーション違反の原因となります。

- `ead_terminate_client()`は、指定したキャッシュマネージャを使用する、すべての API の処理が終了した状態で実行してください。
- 初期設定した EADs クライアントを終了する場合は、必ず `ead_terminate_client()`を実行してください。

## 21.3 ソースプログラムをコンパイルする (C 言語)

---

作成したソースプログラムは、C 言語のコンパイラでコンパイルします。C 言語のコンパイルには、gcc コマンドを使用します。gcc コマンドの詳細については、C 言語のコンパイラに関するドキュメントを参照してください。

コンパイルに必要なヘッダおよびライブラリを次に示します。

### インクルードパス (-I)

/opt/hitachi/xeads/cclient/include

### ライブラリパス (-L)

- 32 ビット版  
/opt/hitachi/xeads/cclient/lib32
- 64 ビット版  
/opt/hitachi/xeads/cclient/lib64

### ライブラリ (-l)

- 32 ビット版  
hcc-4.1.1  
hntr2-eads-t
- 64 ビット版  
hccx64-4.1.1  
hntr2-eads-te64
- 32 ビット版・64 ビット版共通  
eads

# 22

## API リファレンス (C 言語)

この章では、EADs で使用できる API (C 言語) について説明します。

## 22.1 C 言語のクライアントライブラリが提供する関数

C 言語のクライアントライブラリでは、EADs の API として次の表に示す関数を提供します。これらの関数は、C 言語で記述したソースプログラムに、EADs が提供するヘッダファイルを取り込むことで使用できます。

なお、C 言語のクライアントライブラリで使用できる関数（`ead_terminate_client()`を除く）はスレッドセーフです。

表 22-1 C 言語のクライアントライブラリで使用できる関数とヘッダファイル

項番	関数名	機能	ヘッダファイル
1	<code>ead_init_client()</code>	クライアント定義に従って、EADs クライアントの初期設定をします。	<code>eads.h</code>
2	<code>ead_init_client_n()</code>	EADs クライアント名およびクライアント定義に従って、EADs クライアントの初期設定をします。	
3	<code>ead_start_cache()</code>	キャッシュへのアクセスを開始します。指定したキャッシュにアクセスするためのハンドル（ポインタ）を取得します。	
4	<code>ead_stop_cache()</code>	指定したキャッシュへのアクセスを終了します。	
5	<code>ead_terminate_client()</code>	EADs クライアントを終了します。	
6	<code>ead_put()</code>	<code>value</code> を <code>key</code> に関連づけて格納します。	
7	<code>ead_put_array_value()</code>	複数の <code>value</code> を連結して、 <code>key</code> に関連づけて格納します。	
8	<code>ead_create()</code>	新規に <code>key</code> を格納する場合だけ、 <code>value</code> を <code>key</code> に関連づけて格納します。	
9	<code>ead_update()</code>	指定した <code>key</code> がすでに格納されている場合だけ、 <code>value</code> を <code>key</code> に関連づけて格納します（ <code>value</code> を更新します）。	
10	<code>ead_replace()</code>	指定した <code>key</code> に関連づけられている <code>value</code> を、条件として指定した <code>value</code> 情報と比較します。 <code>value</code> が一致している場合だけ、 <code>value</code> を <code>key</code> に関連づけて格納します（ <code>value</code> を置換します）。	
11	<code>ead_get()</code>	指定した <code>key</code> に関連づけられた <code>value</code> を取得します。	

項番	関数名	機能	ヘッダファイル
12	<code>ead_remove()</code>	指定した key, および key に関連づけられた value を削除します。	
13	<code>ead_execute_function()</code>	グループを指定して, ユーザファンクションを実行します。	
14	<code>ead_execute_function_rt()</code>	グループおよび受信タイムアウト時間を指定して, ユーザファンクションを実行します。	
15	<code>ead_execute_node_function()</code>	EADs サーバを指定して, ユーザファンクションを実行します。	
16	<code>ead_execute_node_function_rt()</code>	EADs サーバおよび受信タイムアウト時間を指定して, ユーザファンクションを実行します。	
17	<code>ead_get_nodelist()</code>	EADs クライアントが保持している接続先 EADs サーバの情報を取得します。	
18	<code>ead_get_node()</code>	指定した key またはグループが格納されている接続先 EADs サーバの情報を取得します。	
19	<code>ead_get_slave_nodelist()</code>	データのコピー先 EADs サーバの情報を取得します。	

## 22.1.1 `ead_init_client()` (EADs クライアントの初期設定)

### (1) 機能

クライアント定義に従って, EADs クライアントの初期設定をします。

EADs クライアント名は空文字列として扱います。

また, キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) を取得します。

`ead_init_client()` を実行するたびに, クラスタを監視するスレッドが 1 つ生成されます。生成されたスレッドは, `ead_terminate_client()` 実行時に終了します。

### (2) 形式

```
#include <eads.h>
EAD_CACHE_MANAGER *ead_init_client
(
    const char    *filename,    /* In */
```

```
int          *error_code    /* Out */
);
```

### (3) 引数

filename

EADs クライアントのクライアント定義ファイルのパスを指定します。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

### (4) 戻り値

ead\_init\_client()が正常に終了した場合は、キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) が返却されます。

ead\_init\_client()が異常終了した場合は、NULL が返却されます。

### (5) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- 初期設定した EADs クライアントを終了する場合は、必ず ead\_terminate\_client()を実行してください。

## 22.1.2 ead\_init\_client\_n() (EADs クライアントの初期設定)

### (1) 機能

EADs クライアント名およびクライアント定義に従って、EADs クライアントの初期設定をします。

また、キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) を取得します。

ead\_init\_client\_n()を実行するたびに、クラスタを監視するスレッドが 1 つ生成されます。生成されたスレッドは、ead\_terminate\_client()実行時に終了します。

### (2) 形式

```
#include <eads.h>
EAD_CACHE_MANAGER *ead_init_client_n
(
    const char    *client_name,    /* In */
```



```
const char    *file_name,    /* In */
int          *error_code    /* Out */
);
```

### (3) 引数

#### client\_name

EADs クライアント名を指定します。

指定できるデータについては、「[15.2.2\(4\) EADs クライアント名として指定できるデータ](#)」を参照してください。

EADs クライアント名とログファイルの出力先の関係については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。

#### file\_name

EADs クライアントのクライアント定義ファイルのパスを指定します。

#### error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

### (4) 戻り値

`ead_init_client_n()`が正常に終了した場合は、キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) が返却されます。

`ead_init_client_n()`が異常終了した場合は、NULL が返却されます。

### (5) 注意事項

- EADs クライアントを同一マシンで複数同時に実行する場合は、それぞれ異なるログ出力先を指定してください。同一のログ出力先を指定した場合、動作は保証されません。ログ出力先の設定については、「[8.4.2 ファイルの出力先の設定](#)」を参照してください。
- 初期設定した EADs クライアントを終了する場合は、必ず `ead_terminate_client()`を実行してください。

## 22.1.3 ead\_start\_cache() (キャッシュへのアクセス開始)

### (1) 機能

キャッシュへのアクセスを開始します。

指定したキャッシュにアクセスするためのハンドル (ポインタ) を取得します。

## (2) 形式

```
#include <eads.h>
EAD_CACHE *ead_start_cache
(
    const EAD_CACHE_MANAGER *cmp,           /* In */
    const char *cache_name,                /* In */
    int *error_code                         /* Out */
);
```

## (3) 引数

### cmp

キャッシュを管理するキャッシュマネージャのハンドル（ポインタ）を指定します。  
ead\_init\_client()または ead\_init\_client\_n()で取得したハンドルを指定してください。

### cache\_name

アクセスするキャッシュ名を指定します。  
指定できるデータについては、「[15.2.2\(3\) キャッシュ名として指定できるデータ](#)」を参照してください。

### error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## (4) 戻り値

ead\_start\_cache()が正常に終了した場合は、指定したキャッシュのハンドル（ポインタ）が返却されます。

ead\_start\_cache()が異常終了した場合は、NULL が返却されます。

## (5) 注意事項

キャッシュへのアクセスを終了する場合は、ead\_stop\_cache()を実行してください。

## 22.1.4 ead\_stop\_cache() (キャッシュへのアクセス終了)

### (1) 機能

指定したキャッシュへのアクセスを終了します。

## (2) 形式

```
#include <eads.h>
void ead_stop_cache
(
    const EAD_CACHE      *cp,          /* In */
    int                  *error_code   /* Out */
);
```

## (3) 引数

cp

アクセスを終了するキャッシュのポインタを指定します。

ead\_start\_cache()で取得したハンドルを指定してください。

キャッシュへのアクセス開始時に ead\_start\_cache()で取得したハンドル (ポインタ) を指定します。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## 22.1.5 ead\_terminate\_client() (EADs クライアントの終了)

### (1) 機能

キャッシュを管理するキャッシュマネージャを削除し、EADs クライアントを終了します。

ead\_terminate\_client()は、指定したキャッシュマネージャ、およびそれを使用して開始したキャッシュを解放します。そのため、ead\_terminate\_client()実行以降、これらを参照する操作は実行できません。実行した場合、動作は保証されません。

### (2) 形式

```
#include <eads.h>
void ead_terminate_client
(
    const EAD_CACHE_MANAGER *cmp,      /* In */
    int                    *error_code /* Out */
);
```

## (3) 引数

cmp

キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) を指定します。  
ead\_init\_client()または ead\_init\_client\_n()で取得したハンドルを指定してください。

error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## 22.1.6 ead\_put() (key と value の格納)

### (1) 機能

value を key に関連づけて格納します。

### (2) 形式

```
#include <eads.h>
void ead_put
(
    const EAD_CACHE          *cp,          /* In */
    const char               *key,        /* In */
    const ead_value_element *value,      /* In */
    int                      *error_code /* Out */
);
```

### (3) 引数

cp

key と value を格納するキャッシュのハンドル (ポインタ) を指定します。  
ead\_start\_cache()で取得したハンドルを指定してください。

key

value に関連づける key を指定します。  
指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value 情報 (ead\_value\_element 構造体) を指定します。  
ead\_value\_element 構造体の形式と説明については、「[22.1.20 ead\\_value\\_element 構造体 \(value 情報\)](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## 22.1.7 ead\_put\_array\_value() (複数の value の連結格納)

### (1) 機能

複数の value を連結して、key に関連づけて格納します。

格納した value を取得すると、連結した value が取得できます。

### (2) 形式

```
#include <eads.h>
void ead_put_array_value
(
    const EAD_CACHE           *cp,           /* In */
    const char                *key,         /* In */
    size_t                    array_length,
    const ead_value_element   *value_array, /* In */
    int                        *error_code) /* Out */
);
```

### (3) 引数

cp

key と value を格納するキャッシュのハンドル (ポインタ) を指定します。

ead\_start\_cache() で取得したハンドルを指定してください。

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

array\_length

格納する値配列の要素数を指定します。

value\_array

格納する値配列の先頭を指すポインタを指定します。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## (4) 注意事項

- array\_length には、0 は指定できません。また、不正な値を指定した場合、動作は保証されません。
- ead\_value\_element 構造体の配列は、必ず要素数分の連続したメモリ領域を確保してください。
- 格納する値配列の要素数、サイズに上限はありません。ただし、連結後のサイズには、EADs サーバが受け付けることができるデータの最大サイズ（サーバ定義の `eads.server.max.datasize` パラメタの指定値）を超えない値を設定してください。

## 22.1.8 ead\_create() (key と value の新規格納)

### (1) 機能

新規に key を格納する場合だけ、value を key に関連づけて格納します。

### (2) 形式

```
#include <eads.h>
void ead_create
(
    const EAD_CACHE           *cp,           /* In */
    const char                *key,         /* In */
    const ead_value_element   *value,       /* In */
    int                       *error_code  /* Out */
);
```

### (3) 引数

cp

key と value を格納するキャッシュのハンドル (ポインタ) を指定します。

`ead_start_cache()` で取得したハンドルを指定してください。

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value 情報 (`ead_value_element` 構造体) を指定します。

`ead_value_element` 構造体の形式と説明については、「[22.1.20 ead\\_value\\_element 構造体 \(value 情報\)](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## 22.1.9 ead\_update() (value の更新)

### (1) 機能

指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します (value を更新します)。

### (2) 形式

```
#include <eads.h>
void ead_update
(
    const EAD_CACHE          *cp,          /* In */
    const char               *key,        /* In */
    const ead_value_element *value,      /* In */
    int                      *error_code /* Out */
);
```

### (3) 引数

cp

key と value を格納するキャッシュのハンドル (ポインタ) を指定します。

ead\_start\_cache() で取得したハンドルを指定してください。

key

value に関連づける key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value 情報 (ead\_value\_element 構造体) を指定します。

ead\_value\_element 構造体の形式と説明については、「[22.1.20 ead\\_value\\_element 構造体 \(value 情報\)](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## 22.1.10 ead\_replace() (value の置換)

### (1) 機能

指定した key に関連づけられている value を、条件として指定した value 情報と比較します。value が一致している場合だけ、value を key に関連づけて格納します (value を置換します)。

### (2) 形式

```
#include <eads.h>
void ead_replace
(
    const EAD_CACHE          *cp,           /* In */
    const char               *key,         /* In */
    const ead_value_element *value,       /* In */
    const ead_value_element *comparative_value, /* In */
    int                      *error_code  /* Out */
);
```

### (3) 引数

cp

key と value を格納するキャッシュのハンドル (ポインタ) を指定します。  
ead\_start\_cache() で取得したハンドルを指定してください。

key

置き換える value に関連づけられた key を指定します。  
指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

value

格納する value 情報 (ead\_value\_element 構造体) を指定します。  
ead\_value\_element 構造体の形式と説明については、「[22.1.20 ead\\_value\\_element 構造体 \(value 情報\)](#)」を参照してください。

comparative\_value

比較する value 情報 (ead\_value\_element 構造体) を指定します。  
ead\_value\_element 構造体の形式と説明については、「[22.1.20 ead\\_value\\_element 構造体 \(value 情報\)](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。



## 22.1.11 ead\_get() (value の取得)

### (1) 機能

指定した key に関連づけられた value を取得します。

### (2) 形式

```
#include <eads.h>
ead_value_element ead_get
(
    const EAD_CACHE      *cp,           /* In */
    const char           *key,         /* In */
    int                  *error_code  /* Out */
);
```

### (3) 引数

cp

取得する value が格納されているキャッシュのハンドル (ポインタ) を指定します。  
ead\_start\_cache() で取得したハンドルを指定してください。

key

取得する value に関連づけられた key を指定します。  
指定できるデータについては、「15.2.2(1) key として指定できるデータ」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「22.2 クライアントライブラリのエラーコード (C 言語)」を参照してください。

### (4) 戻り値

key に関連づけられた value 情報 (ead\_value\_element 構造体) が返却されます。

ead\_value\_element 構造体の形式と説明については、「22.1.20 ead\_value\_element 構造体 (value 情報)」を参照してください。

次の場合は、ead\_value\_element 構造体の value メンバに NULL が設定されます。

- key がキャッシュに存在しない場合
- value 情報の取得で問題が発生した場合
- value のデータサイズが 0 バイトの場合 (Java 言語の API で格納された、長さ 0 のバイト配列を取得した場合)

## (5) 注意事項

戻り値として返却された value 情報のメモリ領域は、自動的に解放されないため、AP で解放してください。

### 22.1.12 ead\_remove() (value の削除)

#### (1) 機能

指定した key, および key に関連づけられた value を削除します。

#### (2) 形式

```
#include <eads.h>
void ead_remove
(
    const EAD_CACHE    *cp,          /* In */
    const char         *key,        /* In */
    int                *error_code  /* Out */
);
```

#### (3) 引数

cp

削除する key と value が格納されているキャッシュのハンドル (ポインタ) を指定します。  
ead\_start\_cache() で取得したハンドルを指定してください。

key

削除する value に関連づけられた key を指定します。  
指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

### 22.1.13 ead\_execute\_function() (グループ指定によるユーザファクションの実行)

#### (1) 機能

グループを指定して、ユーザファクションを実行します。

## (2) 形式

```
#include <eads.h>
ead_object ead_execute_function
(
    const EAD_CACHE      *cp,           /* In */
    const char           *group_name,   /* In */
    const char           *func_name,    /* In */
    const ead_object     *arg,         /* In */
    int                  *error_code    /* Out */
);
```

## (3) 引数

cp

ユーザファンクションを実行するキャッシュのハンドル（ポインタ）を指定します。  
ead\_start\_cache()で取得したハンドルを指定してください。

group\_name

グループ名を指定します。  
指定できるデータについては、「15.2.2(1) keyとして指定できるデータ」を参照してください。

func\_name

ユーザファンクション名を指定します。  
半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。  
文字数に制限はありません。

arg

ユーザファンクションに渡す引数 (ead\_object 構造体) を指定します。  
ead\_object 構造体の形式と説明については、「22.1.21 ead\_object 構造体 (ユーザファンクションで使用するオブジェクト)」を参照してください。  
引数を渡さない場合は NULL を指定します。

error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「22.2 クライアントライブラリのエラーコード (C 言語)」を参照してください。

## (4) 戻り値

ead\_execute\_function()が正常に終了した場合は、ユーザファンクションの実行結果 (ead\_object 構造体) が返却されます。

ead\_object 構造体の形式と説明については、「22.1.21 ead\_object 構造体 (ユーザファンクションで使用するオブジェクト)」を参照してください。

ead\_object 構造体の object\_size メンバは、ユーザ関クションの戻り値に指定したバイト配列のサイズになります。

次の場合は、ead\_object 構造体の object メンバに NULL が設定されます。

- ユーザ関クションの実行で問題が発生した場合
- ネットワーク障害などの問題によって、ユーザ関クションの実行結果を取得できなかった場合
- ユーザ関クションの実行結果として null が返却された場合
- ユーザ関クションの実行結果としてデータサイズが 0 のバイト配列が返却された場合

## (5) 注意事項

戻り値として返却された、ユーザ関クションの実行結果を格納するメモリ領域は、自動的に解放されないため、AP で解放してください。

## 22.1.14 ead\_execute\_function\_rt() (グループおよび受信タイムアウト指定によるユーザ関クションの実行)

### (1) 機能

グループおよび受信タイムアウト時間を指定して、ユーザ関クションを実行します。

ead\_execute\_function\_rt()の実行中は、クライアント定義の eads.connection.receive.timeout パラメタの値を使用しません。

### (2) 形式

```
#include <eads.h>
ead_object ead_execute_function_rt
(
    const EAD_CACHE      *cp,           /* In */
    const char           *group_name,   /* In */
    const char           *func_name,    /* In */
    const ead_object     *arg,         /* In */
    int                  recv_timeout,  /* In */
    int                  *error_code    /* Out */
);
```

### (3) 引数

cp  
ユーザ関クションを実行するキャッシュのハンドル (ポインタ) を指定します。  
ead\_start\_cache()で取得したハンドルを指定してください。

#### group\_name

グループ名を指定します。

指定できるデータについては、「[15.2.2\(1\) key](#)として指定できるデータ」を参照してください。

#### func\_name

ユーザファンクション名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

#### arg

ユーザファンクションに渡す引数 (ead\_object 構造体) を指定します。

ead\_object 構造体の形式と説明については、「[22.1.21 ead\\_object 構造体 \(ユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

引数を渡さない場合は NULL を指定します。

#### recv\_timeout

データ受信のタイムアウト時間 (単位: ミリ秒) を指定します。

指定できるデータについては、「[9.3.3\(3\)\(b\) eads.connection.receive.timeout](#)」を参照してください。

#### error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## (4) 戻り値

ead\_execute\_function\_rt() が正常に終了した場合は、ユーザファンクションの実行結果 (ead\_object 構造体) が返却されます。

ead\_object 構造体の形式と説明については、「[22.1.21 ead\\_object 構造体 \(ユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

ead\_object 構造体の object\_size メンバは、ユーザファンクションの戻り値に指定したバイト配列のサイズになります。

次の場合は、ead\_object 構造体の object メンバに NULL が設定されます。

- ユーザファンクションの実行で問題が発生した場合
- ネットワーク障害などの問題によって、ユーザファンクションの実行結果を取得できなかった場合
- ユーザファンクションの実行結果として null が返却された場合
- ユーザファンクションの実行結果としてデータサイズが 0 のバイト配列が返却された場合

## (5) 注意事項

戻り値として返却された、ユーザ関数の実行結果を格納するメモリ領域は、自動的に解放されないため、AP で解放してください。

### 22.1.15 ead\_execute\_node\_function() (EADs サーバ指定によるユーザ関数の実行)

#### (1) 機能

EADs サーバを指定して、ユーザ関数を実行します。

#### (2) 形式

```
#include <eads.h>
ead_object ead_execute_node_function
(
    const EAD_CACHE      *cp,           /* In */
    const ead_node       *target_node, /* In */
    const char           *func_name,    /* In */
    const ead_object     *arg,         /* In */
    int                  *error_code    /* Out */
);
```

#### (3) 引数

**cp**  
ユーザ関数を実行するキャッシュのハンドル (ポインタ) を指定します。  
ead\_start\_cache() で取得したハンドルを指定してください。

**target\_node**  
ユーザ関数を実行する EADs サーバ (ead\_node 構造体) を指すポインタを指定します。

EADs が提供するクライアントライブラリを使用して取得した ead\_node 構造体のポインタだけ指定できます。それ以外のポインタを指定した場合の動作は保証しません。

ead\_node 構造体の形式と説明については、「[22.1.23 ead\\_node 構造体 \(EADs サーバ指定のユーザ関数で使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が保持するアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持する EADs サーバのアドレス情報とも一致しない場合、エラーとなります。

**func\_name**  
ユーザ関数名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。

#### arg

ユーザファンクションに渡す引数 (ead\_object 構造体) を指定します。

ead\_object 構造体の形式と説明については、「[22.1.21 ead\\_object 構造体 \(ユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

引数を渡さない場合は NULL を指定します。

#### error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## (4) 戻り値

ead\_execute\_node\_function() が正常に終了した場合は、ユーザファンクションの実行結果 (ead\_object 構造体) が返却されます。

ead\_object 構造体の形式と説明については、「[22.1.21 ead\\_object 構造体 \(ユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

ead\_object 構造体の object\_size メンバは、ユーザファンクションの戻り値に指定したバイト配列のサイズになります。

次の場合は、ead\_object 構造体の object メンバに NULL が設定されます。

- ユーザファンクションの実行で問題が発生した場合
- ネットワーク障害などの問題によって、ユーザファンクションの実行結果を取得できなかった場合
- ユーザファンクションの実行結果として null が返却された場合
- ユーザファンクションの実行結果としてデータサイズが 0 のバイト配列が返却された場合

## (5) 注意事項

戻り値として返却された、ユーザファンクションの実行結果を格納するメモリ領域は、自動的に解放されないため、AP で解放してください。

## 22.1.16 ead\_execute\_node\_function\_rt() (EADs サーバおよび受信タイムアウト指定によるユーザ関数の実行)

### (1) 機能

EADs サーバおよび受信タイムアウト時間を指定して、ユーザ関数を実行します。

ead\_execute\_node\_function\_rt()の実行中は、クライアント定義の `eads.connection.receive.timeout` パラメタの値を使用しません。

### (2) 形式

```
#include <eads.h>
ead_object ead_execute_node_function_rt
(
    const EAD_CACHE      *cp,           /* In */
    const ead_node       *target_node,  /* In */
    const char           *func_name,    /* In */
    const ead_object     *arg,         /* In */
    int                  recv_timeout,  /* In */
    int                  *error_code    /* Out */
);
```

### (3) 引数

**cp**

ユーザ関数を実行するキャッシュのハンドル (ポインタ) を指定します。

ead\_start\_cache()で取得したハンドルを指定してください。

**target\_node**

ユーザ関数を実行する EADs サーバ (ead\_node 構造体) を指すポインタを指定します。

EADs が提供するクライアントライブラリを使用して取得した ead\_node 構造体のポインタだけ指定できます。それ以外のポインタを指定した場合の動作は保証しません。

ead\_node 構造体の形式と説明については、「[22.1.23 ead\\_node 構造体 \(EADs サーバ指定のユーザ関数で使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が保持するアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持する EADs サーバのアドレス情報とも一致しない場合、エラーとなります。

**func\_name**

ユーザ関数名を指定します。

半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.) またはドルマーク (\$) が指定できます。

文字数に制限はありません。



arg

ユーザファンクションに渡す引数 (ead\_object 構造体) を指定します。

ead\_object 構造体の形式と説明については、「[22.1.21 ead\\_object 構造体 \(ユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

引数を渡さない場合は NULL を指定します。

recv\_timeout

データ受信のタイムアウト時間 (単位: ミリ秒) を指定します。

指定できるデータについては、「[9.3.3\(3\)\(b\) eads.connection.receive.timeout](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## (4) 戻り値

ead\_execute\_node\_function\_rt() が正常に終了した場合は、ユーザファンクションの実行結果 (ead\_object 構造体) が返却されます。

ead\_object 構造体の形式と説明については、「[22.1.21 ead\\_object 構造体 \(ユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

ead\_object 構造体の object\_size メンバは、ユーザファンクションの戻り値に指定したバイト配列のサイズになります。

次の場合は、ead\_object 構造体の object メンバに NULL が設定されます。

- ユーザファンクションの実行で問題が発生した場合
- ネットワーク障害などの問題によって、ユーザファンクションの実行結果を取得できなかった場合
- ユーザファンクションの実行結果として null が返却された場合
- ユーザファンクションの実行結果としてデータサイズが 0 のバイト配列が返却された場合

## (5) 注意事項

戻り値として返却された、ユーザファンクションの実行結果を格納するメモリ領域は、自動的に解放されないため、AP で解放してください。

## 22.1.17 ead\_get\_nodelist() (接続先 EADs サーバの情報取得)

### (1) 機能

EADs クライアントが保持している接続先 EADs サーバの情報を取得します。

### (2) 形式

```
#include <eads.h>
ead_nodelist ead_get_nodelist
(
    const EAD_CACHE_MANAGER *cmp,          /* In */
    int *error_code                       /* Out */
);
```

### (3) 引数

cmp

キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) を指定します。  
ead\_init\_client()または ead\_init\_client\_n()で取得したハンドルを指定してください。

error\_code

エラーコード取得先のポインタを指定します。  
エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

### (4) 戻り値

EADs クライアントが保持している接続先 EADs サーバの情報 (ead\_nodelist 構造体) が返却されます。

ead\_nodelist 構造体の形式と説明については、「[22.1.22 ead\\_nodelist 構造体 \(EADs サーバ情報\)](#)」を参照してください。

異常終了時には、ead\_nodelist 構造体の nodes メンバに NULL が返却されます。

### (5) 注意事項

- 戻り値として返却された ead\_nodelist 構造体の nodes メンバが NULL でない場合、メモリ領域は自動的に解放されないため、AP で解放してください。
- AP 実行時に EADs サーバへの通信は行われません。そのため、返却された接続先 EADs サーバの情報が最新でないことがあります。

## 22.1.18 ead\_get\_node() (指定した key が格納されている接続先 EADs サーバの情報取得)

### (1) 機能

指定した key またはグループが格納されている接続先 EADs サーバの情報を取得します。

### (2) 形式

```
#include <eads.h>
ead_node ead_get_node
(
    const EAD_CACHE_MANAGER *cmp,           /* In */
    const char *key,                       /* In */
    int *error_code                        /* Out */
);
```

### (3) 引数

cmp

キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) を指定します。ead\_init\_client() または ead\_init\_client\_n() で取得したハンドルを指定してください。

key

取得する EADs サーバに格納されている key を指定します。

指定できるデータについては、「[15.2.2\(1\) key として指定できるデータ](#)」を参照してください。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

### (4) 戻り値

指定した key またはグループが格納されている接続先 EADs サーバの情報 (ead\_node 構造体) が返却されます。

### (5) 注意事項

AP 実行時に EADs サーバへの通信は行われません。そのため、返却された接続先 EADs サーバの情報が最新でないことがあります。

## 22.1.19 ead\_get\_slave\_nodelist() (データのコピー先 EADs サーバの情報取得)

### (1) 機能

データのコピー先 EADs サーバの情報を取得します。

### (2) 形式

```
#include <eads.h>
ead_nodelist ead_get_slave_nodelist(
    const EAD_CACHE_MANAGER *cmp,          /* In */
    const ead_node          *master_node,  /* In */
    int                     *error_code    /* Out */
);
```

### (3) 引数

cmp

キャッシュを管理するキャッシュマネージャのハンドル (ポインタ) を指定します。ead\_init\_client() または ead\_init\_client\_n() で取得したハンドルを指定してください。

master\_node

データのコピー元 EADs サーバ (ead\_node 構造体) を指すポインタを指定します。

EADs が提供するクライアントライブラリを使用して取得した ead\_node 構造体のポインタだけ指定できます。それ以外のポインタを指定した場合の動作は保証しません。

ead\_node 構造体の形式と説明については、「[22.1.23 ead\\_node 構造体 \(EADs サーバ指定のユーザー関数で使用されるオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が保持するアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持するどの EADs サーバのアドレス情報とも一致しない場合、エラーとなります。

error\_code

エラーコード取得先のポインタを指定します。

エラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

### (4) 戻り値

データのコピー先 EADs サーバの情報 (ead\_nodelist 構造体) が返却されます。

ead\_nodelist 構造体の形式と説明については、「[22.1.22 ead\\_nodelist 構造体 \(EADs サーバ情報\)](#)」を参照してください。

データを多重化しない場合（多重度が 1 の場合）、ead\_nodelist 構造体の list\_size メンバには 0 が、nodes メンバには NULL が返却されます。

異常終了時は、ead\_nodelist 構造体の nodes メンバに NULL が返却されます。

## (5) 注意事項

- 戻り値として返却された ead\_nodelist 構造体の nodes メンバが NULL でない場合、メモリ領域は自動的に解放されないため、AP で解放してください。
- AP 実行時に EADs サーバへの通信は行われません。そのため、返却された接続先 EADs サーバの情報が最新でないことがあります。

## 22.1.20 ead\_value\_element 構造体 (value 情報)

ead\_value\_element 構造体には、value 情報 (value と value のサイズ) を格納します。

### (1) 形式

```
struct ead_value_element {
    size_t    value_size;
    void      *value;
};
```

### (2) メンバの説明

value\_size

value のデータサイズ (単位: バイト) を格納します。

value

value を格納します。

格納するデータの先頭アドレスを指定します。

### (3) 注意事項

- 先頭アドレス+データサイズで指定した範囲が不正な領域を参照していた場合、動作は保証されません。
- value には、NULL は指定できません。
- value\_size には、0 は指定できません。
- value\_size に上限はありません。ただし、EADs サーバが受け付けることができるデータの最大サイズ (サーバ定義の `eads.server.max.datasize` パラメタの指定値) を超えない値を設定してください。

## 22.1.21 ead\_object 構造体 (ユーザファンクションで使用するオブジェクト)

ead\_object 構造体には、ユーザファンクションで使用するオブジェクト (引数や戻り値) を格納します。

### (1) 形式

```
struct ead_object {
    size_t    object_size;
    void     *object;
};
```

### (2) メンバの説明

object\_size

オブジェクトのデータサイズ (単位: バイト) を格納します。

object

ユーザファンクションで使用するオブジェクト (引数や戻り値) を格納します。

格納するデータの先頭アドレスを指定します。

### (3) 注意事項

- 先頭アドレス+データサイズで指定した範囲が不正な領域を参照していた場合、動作は保証されません。
- object には、NULL は指定できません。
- object\_size には、0 は指定できません。
- object\_size に上限はありません。ただし、EADs サーバが受け付けることができるデータの最大サイズ (サーバ定義の `eads.server.max.datasize` パラメタの指定値) を超えない値を設定してください。

## 22.1.22 ead\_nodelist 構造体 (EADs サーバ情報)

ead\_nodelist 構造体には、EADs クライアントが保持する EADs サーバの情報を格納します。

### (1) 形式

```
struct ead_nodelist {
    size_t    list_size;
    struct ead_node *nodes;
};
```

## (2) メンバの説明

`list_size`

EADs サーバ数を格納します。

`nodes`

EADs サーバの情報を格納します。

格納するデータの先頭アドレスを指定します。

### 22.1.23 ead\_node 構造体 (EADs サーバ指定のユーザファンクションで使用するオブジェクト)

`ead_node` 構造体には、EADs サーバを指定するための情報を格納します。

#### (1) 形式

```
struct ead_node {
    int         node_id;
    EAD_BOOL    is_enable;
    int         position;
    struct ead_address address;
    char        reserved[8];
};
```

#### (2) メンバの説明

`node_id`

EADs サーバ ID を格納します。

`is_enable`

EADs サーバの状態を格納します。

EAD\_TRUE：接続可能

EAD\_FALSE：接続不可

`position`

EADs サーバの位置を格納します。

`address`

EADs サーバのアドレス情報 (`ead_address` 構造体) を格納します。

`ead_address` 構造体の形式と説明については、「[22.1.24 ead\\_address 構造体 \(EADs サーバのアドレス情報\)](#)」を参照してください。

`reserved[8]`

予備領域です。

## 22.1.24 ead\_address 構造体 (EADs サーバのアドレス情報)

ead\_address 構造体には、EADs サーバのアドレス情報を格納します。

### (1) 形式

```
struct ead_address {  
    unsigned char    ip[4];  
    unsigned short   port;  
};
```

### (2) メンバの説明

ip

EADs サーバの IP アドレスを格納します。

port

EADs サーバのポート番号を格納します。



## 22.2 クライアントライブラリのエラーコード (C 言語)

C 言語のクライアントライブラリで使用できる関数は、引数 `error_code` に指定された領域に次に示すエラーコードを返却します。

エラーコードとエラーの内容、発生要因を次の表に示します。

表 22-2 C 言語のクライアントライブラリで使用できる関数が返却するエラーコード

error_code の値	記号定数	エラーの内容	発生要因
0	EAD_OK	処理は正常に終了しました。	—
1000	EAD_ERROR_UNEXPECTED	予期しないエラーが発生しました。	プログラム内で予期しないエラーが発生しました。
1010	EAD_ERROR_ILLEGAL_ARGUMENT	指定されたパラメタが不正です。	API の引数に不正なパラメタが指定されています。
1040	EAD_ERROR_CACHE_NOT_STARTED	キャッシュが開始されていないため実行できません。	キャッシュの終了後 ( <code>ead_stop_cache()</code> の実行後) に、データを操作しました。
1100	EAD_ERROR_CACHE_NOT_NEED_STOP	すでに終了しているキャッシュに対して、終了処理をしています。	キャッシュはすでに終了しているため、 <code>ead_stop_cache()</code> を実行する必要はありません。
1110	EAD_ERROR_INVALID_NODE_ADDRESS	指定した EADs サーバのアドレス情報が、EADs クライアントが保持する EADs サーバのアドレス情報と一致しません。	API の引数に指定した EADs サーバのアドレス情報 (IP アドレスおよびポート番号) が、EADs クライアントが保持する EADs サーバのアドレス情報とも一致しません。
1120	EAD_ERROR_EXCEED_MAX_CONNECTION_POOL_SIZE	同一接続先に対してプールしておく接続がすでに最大個数に達していて、すべて使用中です。	特定の EADs サーバに対して同時にリクエストしているスレッド数が、接続の最大個数を超えています。
2000	EAD_ERROR_INIT	EADs クライアントの初期設定でエラーが発生しました。	<code>ead_init_client()</code> または <code>ead_init_client_n()</code> 実行中に予期しないエラーが発生しました。
2010	EAD_ERROR_INIT_PROPERTIES	クライアント定義ファイルが読み込めませんでした。	次の要因が考えられます。 <ul style="list-style-type: none"><li>クライアント定義ファイルが存在しない。</li><li>クライアント定義ファイルに読み込み権限がない。</li></ul>

error_code の値	記号定数	エラーの内容	発生要因
			<ul style="list-style-type: none"> <li>格納先パス名がファイルではなくディレクトリを指している。</li> </ul>
2020	EAD_ERROR_INIT_INVALID_PROPERTY	クライアント定義ファイルの定義内容に不正があります。	クライアント定義ファイルの定義内容が不正です。
2030	EAD_ERROR_INIT_LOGGER	ログの初期化に失敗しました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>指定されたディレクトリ、または出力先のログファイルに書き込み権限がない。</li> <li>指定されたディレクトリに同名のファイルが存在する。</li> <li>指定されたパス名、またはファイル名が不正である。</li> <li>ログファイル名と同名のディレクトリが存在した。</li> <li>ログ出力を開始するためのメモリが不足している。</li> </ul>
2040	EAD_ERROR_INIT_CLUSTERINFO	クライアント定義ファイルで指定された EADs サーバへの接続に失敗しました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>クライアント定義に指定した接続先 EADs サーバの指定に誤りがある。</li> <li>接続先 EADs サーバとの通信に失敗した。または接続先 EADs サーバで障害が発生している。</li> <li>接続先 EADs サーバの最大同時接続数を超えた。</li> <li>接続先 EADs サーバがリクエストの受付を開始できる状態でない。</li> </ul>
3000	EAD_ERROR_NET	EADs サーバとの間で通信エラーが発生しました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>通信中にネットワーク障害が発生した。</li> <li>通信先の EADs サーバで問題が発生した。</li> <li>通信先のホストで問題が発生した。</li> <li>EADs サーバが受信できるデータの最大サイズを超えた。</li> </ul>

error_code の値	記号定数	エラーの内容	発生要因
3010	EAD_ERROR_NET_TIMEOUT	EADs サーバとの通信中にタイムアウトが発生しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバで問題が発生した。</li> <li>• 接続先のホストで問題が発生した。</li> <li>• ネットワークで問題が発生した。</li> <li>• タイムアウト時間の設定が誤っている。</li> </ul>
3020	EAD_ERROR_NET_CONNECTION	EADs サーバとの接続に失敗しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバで問題が発生した。</li> <li>• 接続先 EADs サーバに関する設定が誤っている。</li> <li>• ネットワークで問題が発生した。</li> <li>• タイムアウト時間の設定が誤っている。</li> </ul>
3030	EAD_ERROR_NET_PROTOCOL	EADs サーバとの通信中にプロトコル障害が発生しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバが誤っている。</li> <li>• 実行したユーザファンクションがバイト配列、または NULL 以外を戻り値として返却した。</li> <li>• 接続しているクラスタのキャッシュが Java 言語のクライアントライブラリで使用されている。</li> </ul>
3040	EAD_ERROR_NET_CLUSTERINFO	接続できる、すべての EADs サーバへの接続に失敗しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>• 接続先 EADs サーバで問題が発生した。</li> <li>• 接続先 EADs サーバに関する設定が誤っている。</li> <li>• ネットワークで問題が発生した。</li> <li>• 指定したキャッシュが接続先 EADs サーバに存在しない。</li> <li>• 接続先 EADs サーバが閉塞状態となっている。</li> <li>• EADs クライアントが保持するクラスタ構成情報と、</li> </ul>

error_code の値	記号定数	エラーの内容	発生要因
			再起動した接続先 EADs サーバが保持するクラスタ構成情報が一致しない。 <ul style="list-style-type: none"> <li>接続先 EADs サーバの最大同時接続数を超えた。</li> </ul>
4000	EAD_ERROR_SERVER	EADs サーバで内部エラーが発生しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>接続先 EADs サーバで内部エラーが発生した。</li> <li>接続先 EADs サーバが閉塞状態となっている。</li> <li>接続先 EADs サーバのユーザファンクション内でエラーが発生した。</li> <li>指定したキャッシュ名のキャッシュが接続先 EADs サーバに存在しない。</li> <li>指定した key のサイズがクラスタに格納できる key の最大サイズを超えている。</li> <li>データの格納先の空き容量が不足している。</li> </ul>
4010	EAD_ERROR_SERVER_UNSUPPORTED_REQUEST	EADs クライアントが送信したリクエストを、接続先 EADs サーバが処理できませんでした。	データが破損しているなどの理由で、接続先 EADs サーバがリクエストを処理できません。
4020	EAD_ERROR_SERVER_EVENTHANDLER	データの更新操作および多重化は成功したが、ユーザイベントハンドラ内の処理で例外が発生しました。	接続先 EADs サーバのユーザイベントハンドラ内の処理で例外が発生しました。
4030	EAD_ERROR_SERVER_UNAVAILABLE	接続先 EADs サーバプロセスが一時的に利用できません。	接続先 EADs サーバの最大同時接続数を超えました。
4040	EAD_ERROR_SERVER_INCOMPATIBLE_CLUSTERINFO	接続先 EADs サーバが保持するクラスタ構成情報と、EADs クライアントが保持するクラスタ構成情報に互換性がありません。	再起動した接続先 EADs サーバのクラスタ構成情報が、EADs クライアントが保持するクラスタ構成情報と一致しません。
4060	EAD_ERROR_SERVER_REPLACE_METHOD_NOT_MATCHED	ead_replace()実行時、すでに格納されている value が条件に指定した value と一致しなかったため、value を格納できませんでした。	ead_replace()で条件に指定した value とキャッシュ上の value が一致しませんでした。

error_code の値	記号定数	エラーの内容	発生要因
4070	EAD_ERROR_SERVER_REPLACE_METHOD_KEY_NOT_EXIST	ead_replace()実行時、指定したkeyが存在しなかったため (keyに関連づけられたvalueが存在しなかったため)、処理を続行できませんでした。	ead_replace()で指定したkeyに関連づけられたvalueが存在しなかったため、valueの比較ができませんでした。
4080	EAD_ERROR_SERVER_CREATE_METHOD_KEY_EXIST	ead_create()実行時、指定したkeyがすでに存在したため、valueを格納できませんでした。	ead_create()で指定したkeyに関連づけられたvalueがすでに格納されています。
4090	EAD_ERROR_SERVER_UPDATE_METHOD_KEY_NOT_EXIST	ead_update()実行時、指定したkeyが存在しなかったため、valueを格納できませんでした。	ead_update()で指定したkeyに関連づけられたvalueが格納されていません。
4100	EAD_ERROR_SERVER_NOT_RUNNING	データのコピー元、およびコピー先EADsサーバがすべて縮退状態または停止状態です。	EADsクライアントからのリクエストを処理するEADsサーバと、データのコピー先EADsサーバがすべて縮退状態または停止状態です。
5000	EAD_ERROR_CLIENT	EADsクライアントで内部エラーが発生しました。	クライアントライブラリ内で予期しないエラーが発生しました。
5010	EAD_ERROR_CLIENT_OUT_OF_MEMORY	EADsクライアント内でメモリ確保に失敗しました。	クライアントライブラリ内でメモリ確保に失敗しました。

(凡例)

— : 該当しません。

# 23

## よく使用する一覧集

この章では、パラメタ、コマンド、API、エラーコード、および最小値・最大値の一覧について説明します。

## 23.1 パラメター一覧

EADs サーバ、EADs クライアントで使用する定義ファイルごとのパラメターの一覧について説明します。

### 23.1.1 EADs サーバで使用する定義ファイル

#### (1) サーバ定義ファイル

サーバ定義ファイルに指定できるパラメターの一覧を次の表に示します。

なお、必ず指定するパラメターを太字で表記しています。

環境依存パラメターの設計については、「[7. 環境依存パラメターを設計する \(EADs サーバ\)](#)」を参照してください。

チューニングパラメターの設計については、「[9. チューニングパラメターを設計する](#)」を参照してください。

表 23-1 サーバ定義ファイルに指定できるパラメター一覧

項番	分類	パラメター名	指定する内容
1	通信 (「 <a href="#">7.3 通信に依存するパラメターの設計</a> 」参照)	<b>eads.server.address</b>	EADs サーバの IP アドレスまたはホスト名
2		<b>eads.server.port</b>	EADs サーバのポート番号
3		<b>eads.consensus.message.port</b>	EADs サーバ間の通信に使用するポート番号
4		<b>eads.cluster.failureDetector.port</b>	EADs サーバ間の生存確認に使用するポート番号
5		<b>eads.restore.port</b>	EADs サーバの復旧に使用するポート番号
6		<b>eads.management.port</b>	コマンドが使用するポート番号
7	ログファイル (「 <a href="#">7.4 ログファイルに依存するパラメターの設計</a> 」参照)	<b>eads.logger.dir</b>	ログファイルの出力先ディレクトリのパス
8		<b>eads.logger.message.rotationStyle</b>	メッセージログのローテーション方式
9		<b>eads.logger.message.filesize</b>	メッセージログの 1 ファイル当たりのファイルサイズ
10		<b>eads.logger.message.filenum</b>	メッセージログのファイル数
11		<b>eads.logger.message.console.enable</b>	メッセージログの標準出力への出力を有効にするか、無効にするか
12		<b>eads.logger.exception.rotationStyle</b>	例外ログのローテーション方式
13		<b>eads.logger.exception.filesize</b>	例外ログの 1 ファイル当たりのファイルサイズ

項番	分類	パラメタ名	指定する内容
14		eads.logger.exception.filenum	例外ログのファイル数
15		eads.logger.user.rotationStyle	ユーザログのローテーション方式
16		eads.logger.user.filesize	ユーザログの 1 ファイル当たりのファイルサイズ
17		eads.logger.user.filenum	ユーザログのファイル数
18		eads.logger.userException.rotationStyle	ユーザ例外ログのローテーション方式
19		eads.logger.userException.filesize	ユーザ例外ログの 1 ファイル当たりのファイルサイズ
20		eads.logger.userException.filenum	ユーザ例外ログのファイル数
21		eads.logger.cache.rotationStyle	キャッシュファイル操作ログのローテーション方式
22		eads.logger.cache.filesize	キャッシュファイル操作ログの 1 ファイル当たりのファイルサイズ
23		eads.logger.cache.filenum	キャッシュファイル操作ログのファイル数
24		eads.prf.enable	PRF トレースの出力を有効にするか、無効にするか
25		eads.statistics.interval	統計情報を出力する間隔
26		eads.statistics.backupnum	統計情報の取得日数
27		eads.statistics.compaction.effect.division	統計情報で使用する、コンパクションの効果のファイル数の分布を分割する個数
28	バックアップファイル (「7.6 バックアップファイルに依存するパラメタの設計」参照)	eads.storeData.dir	ストアデータファイルの出力先パス名
29	メモリ・バッファ (「9.1 メモリ・バッファに関連するパラメタの設計」参照)	eads.connection.bufferSize	データの送受信バッファサイズ
30		eads.server.max.datasize.enable	EADs サーバが受け付けるデータの最大サイズのチェックを有効にするか、無効にするか
31		eads.server.max.datasize	EADs サーバが受け付けるデータの最大サイズ
32		eads.consensus.message.bufferSize	合意メッセージの送受信バッファサイズ
33		eads.consensus.message.send.queueSize	合意メッセージの送信キューの長さ
34		eads.consensus.message.send.max_queuedatasize	合意メッセージの送信キューに格納できるデータの最大サイズ



項番	分類	パラメタ名	指定する内容
35		eads.restore.connection.bufferSize	復旧処理で使用するデータの送受信バッファサイズ
36		eads.restore.dataSender.sendSize	復旧処理で送信するデータサイズ
37		eads.consensus.fillgap.copy.dataSize	EADs サーバに対して、更新操作の履歴の補完処理でデータをコピーする際、1 回当たりに送信するデータサイズ
38		eads.consensus.external.heapSize	更新操作の履歴が格納される領域のサイズ
39		eads.management.resume.sendSize	eztool resume コマンド実行時に転送する差分のデータサイズ
40	スレッドプール およびコネクション プール  (「9.2 スレッド プールおよび コネクション プールに関連する パラメタの 設計」参照)	eads.server.max_connections	EADs サーバへの最大同時接続数
41		eads.server.cache.max_execute_threads	データ操作の最大同時実行スレッド数
42		eads.server.function_container.max_execute_threads	ユーザファンクション全体の最大同時実行スレッド数
43		eads.cluster.controller.cache.max_execute_threads	キャッシュ単位に行われる処理の最大同時実行スレッド数
44	タイムアウト (「9.3 タイム アウトに関連する パラメタの 設計」参照)	eads.connection.timeout	クラスタ構成情報の更新確認やデータ送受信のタイムアウト時間
45		eads.cluster.heartbeat.interval	ハートビートの送信間隔
46		eads.cluster.heartbeat.timeout	ハートビートのタイムアウト時間
47		eads.cluster.failureDetector.connect.timeout	生存確認の接続タイムアウト時間
48		eads.cluster.failureDetector.read.timeout	生存確認の受信タイムアウト時間
49		eads.cluster.failureDetector.retry	生存確認タイムアウト時のリトライ回数
50		eads.cluster.assertive.threshold	EADs サーバダウンに同意する EADs サーバ数
51		eads.cluster.boot.timeout	クラスタを構成する全 EADs サーバが開始するまでの最大待ち時間
52		eads.server.session.keepAlive.timeout	EADs クライアント・EADs サーバ間の接続を切断するまでの時間
53		eads.gracefulstop.wait_time	eztool isolate コマンドを実行した EADs サーバの縮退処理が完了してから、停止するまでの時間
54		eads.cache.consensus.timeout	合意処理のタイムアウト時間
55		eads.restore.dataSender.send.timeout	復旧処理でのデータ送信タイムアウト時間
56	eads.restore.dataReceiver.read.timeout	復旧処理でのデータ受信タイムアウト時間	

項番	分類	パラメタ名	指定する内容
57		<code>eads.restore.dataSender.sendInterval</code>	復旧処理でのデータ送信間隔
58		<code>eads.consensus.fillgap.copy.timeout</code>	更新操作の履歴の補完処理で、更新操作の履歴を送信する際の送信タイムアウト時間
59		<code>eads.management.resume.sendInterval</code>	<code>eztool resume</code> コマンド実行時の差分転送間隔
60	キャッシュ (「7.7 キャッシュの運用に依存するパラメタの設計」参照)	<code>eads.cache.disk.getError.isolate.enable</code>	ディスクキャッシュおよび 2Way キャッシュで、データの整合性に影響がないディスク I/O エラーが発生した場合に、EADs サーバを縮退させるか、縮退させないか

## (2) クラスタ定義ファイル

クラスタ定義ファイルに指定できるパラメタの一覧を次の表に示します。

なお、必ず指定するパラメタを太字で表記しています。

環境依存パラメタの設計については、「7. 環境依存パラメタを設計する (EADs サーバ)」を参照してください。

表 23-2 クラスタ定義ファイルに指定できるパラメタ一覧

項番	分類	パラメタ名	指定する内容
1	バックアップ ファイル	<code>eads.storeData.generation.count</code>	<code>eztool export</code> コマンド実行時に出力されるストアデータファイルの世代数の上限値
2	(「7.6 バックアップファイルに依存するパラメタの設計」参照)	<code>eads.storeData.stop_command.generation.count</code>	<code>eztool stop</code> コマンド実行時に出力されるストアデータファイルの世代数の上限値
3	クラスタ構成 (「7.5 クラスタ構成に依存するパラメタの設計」参照)	<code>eads.node.&lt; EADs サーバ ID &gt;.address</code>	クラスタを構成する EADs サーバの IP アドレス
4		<code>eads.node.&lt; EADs サーバ ID &gt;.port</code>	クラスタを構成する EADs サーバのポート番号
5		<code>eads.node.&lt; EADs サーバ ID &gt;.position</code>	EADs サーバの位置 (ハッシュ値)
6		<code>eads.cluster.heartbeat.address</code>	EADs サーバ間のハートビートの送信に使用する IP アドレス (マルチキャストアドレス)
7		<code>eads.cluster.heartbeat.port</code>	EADs サーバ間のハートビートの送信に使用するポート番号
8		<code>eads.replication.factor</code>	データの多重度

項番	分類	パラメタ名	指定する内容
9	メモリ・バッファ	eads.server.max_key_size	クラスタに格納できる key の最大サイズ
10	(「9.1 メモリ・バッファに関連するパラメタの設計」参照)	eads.cache.prepare.num	一度に合意できる処理の最大数

### (3) 起動設定ファイル

起動設定ファイルに指定できるパラメタの一覧を次の表に示します。

環境依存パラメタの設計については、「7. 環境依存パラメタを設計する (EADs サーバ)」を参照してください。

チューニングパラメタの設計については、「9. チューニングパラメタを設計する」を参照してください。

表 23-3 起動設定ファイルに指定できるパラメタ一覧

項番	分類	パラメタ名	指定する内容
1	ログファイル (「7.4 ログファイルに依存するパラメタの設計」参照)	eads.prf.filesize	PRF トレースのファイルサイズ
2		eads.prf.filenum	PRF トレースのファイル数
3		eads.prf.level	PRF トレースの出力レベル
4		eads.java.log.filesize	Java ログ (JavaVM の保守情報) のファイルサイズ
5		eads.java.log.filenum	Java ログ (JavaVM の保守情報) のファイル数
6	メモリ・バッファ (「9.1 メモリ・バッファに関連するパラメタの設計」参照)	eads.prf.buffersize	PRF デーモンが共用メモリに確保するバッファのサイズ
7		eads.java.heapsize	Java ヒープサイズ
8		eads.java.external.heapsize	Explicit ヒープサイズ
9		eads.java.permanent.size	Permanent 領域のサイズ

### (4) コマンド定義ファイル

コマンド定義ファイルに指定できるパラメタの一覧を次の表に示します。

環境依存パラメタの設計については、「7. 環境依存パラメタを設計する (EADs サーバ)」を参照してください。

チューニングパラメタの設計については、「9. チューニングパラメタを設計する」を参照してください。

表 23-4 コマンド定義ファイルに指定できるパラメーター一覧

項番	分類	パラメータ名	指定する内容
1	ログファイル (「7.4 ログ ファイルに依存 するパラメータの 設計」参照)	eads.logger.dir	ログファイルの出力先ディレクトリのパス
2		eads.logger.message.rotationStyle	メッセージログのローテーション方式
3		eads.logger.message.filesize	メッセージログの 1 ファイル当たりのファイルサイズ
4		eads.logger.message.filenum	メッセージログのファイル数
5		eads.logger.exception.rotationStyle	例外ログのローテーション方式
6		eads.logger.exception.filesize	例外ログの 1 ファイル当たりのファイルサイズ
7		eads.logger.exception.filenum	例外ログのファイル数
8	タイムアウト (「9.3 タイム アウトに関連する パラメータの 設計」参照)	eads.management.connect.timeout	コマンド実行時のクラスタ構成情報の更新確認の接続タイムアウト時間
9		eads.management.read.timeout	コマンド実行時のクラスタ構成情報の更新確認の受信タイムアウト時間
10		eads.management.command.timeout	コマンド実行開始からのタイムアウト時間
11		eads.management.long_running.read.timeout	次に示すコマンド実行時のクラスタ構成情報の更新確認の受信タイムアウト時間 <ul style="list-style-type: none"> <li>• eztool export</li> <li>• eztool import</li> <li>• eztool createcache</li> <li>• eztool deletetecache</li> <li>• eztool execfunc</li> <li>• eztool importecf</li> <li>• eztool resume</li> <li>• eztool compaction</li> <li>• eztool stop</li> </ul>
12		eads.management.long_running.command.timeout	次に示すコマンドの実行開始からのタイムアウト時間 <ul style="list-style-type: none"> <li>• eztool export</li> <li>• eztool import</li> <li>• eztool createcache</li> <li>• eztool deletetecache</li> <li>• eztool execfunc</li> <li>• eztool importecf</li> <li>• eztool resume</li> <li>• eztool compaction</li> <li>• eztool stop</li> </ul>

項番	分類	パラメタ名	指定する内容
13	コンパクション (「9.4 コンパクションに関連するパラメタの設計」参照)	<code>eads.management.compaction.effect.threshold</code>	次に示すコマンドで使用する、コンパクションの効果のしきい値 <ul style="list-style-type: none"> <li>• <code>eztool listecf</code></li> <li>• <code>eztool compaction</code></li> </ul>
14		<code>eads.management.compaction.effect.division</code>	<code>eztool listecf</code> コマンドで使用する、コンパクションの効果のファイル数の分布を分割する個数

## (5) キャッシュ定義ファイル

キャッシュ定義ファイルに指定できるパラメタの一覧を次の表に示します。

なお、必ず指定するパラメタを太字で表記しています。

環境依存パラメタの設計については、「7. 環境依存パラメタを設計する (EADs サーバ)」を参照してください。

表 23-5 キャッシュ定義ファイルに指定できるパラメタ一覧

項番	分類	パラメタ名	指定する内容
1	キャッシュ (「7.7 キャッシュの運用に依存するパラメタの設計」参照)	<b><code>eads.cache.type</code></b>	キャッシュタイプ
2		<code>eads.cache.disk.info.dir</code>	キャッシュ情報ファイルおよびキャッシュインデックスファイルの格納先
3		<code>eads.cache.disk.&lt;n&gt;.dir</code>	キャッシュデータファイルの格納先
4		<code>eads.cache.disk.filesize</code>	キャッシュデータファイルのファイルサイズ
5		<code>eads.cache.disk.filenum</code>	キャッシュデータファイルの面数
6		<code>eads.cache.disk.blocksize</code>	キャッシュデータファイルに1回で書き込むデータサイズ
7		<code>eads.cache.restore.dataSender.sendInterval</code>	ディスクキャッシュ、および2Wayキャッシュの復旧処理でのデータ送信間隔
8		<code>eads.cache.restore.dataSender.sendSize</code>	ディスクキャッシュ、および2Wayキャッシュの復旧処理で送信するデータサイズ

## (6) ファンクション定義ファイル

ファンクション定義ファイルに指定できるパラメタの一覧を次の表に示します。

チューニングパラメタの設計については、「9. チューニングパラメタを設計する」を参照してください。

表 23-6 ファンクション定義ファイルに指定できるパラメーター一覧

項番	分類	パラメータ名	指定する内容
1	スレッドプール およびコネクションプール (「9.2 スレッドプールおよびコネクションプールに関連するパラメータの設計」参照)	eads.function.<ユーザファンクション名>.max_execute_threads	ユーザファンクション単位の最大同時実行スレッド数

## 23.1.2 EADs クライアントで使用する定義ファイル

### (1) クライアント定義ファイル

クライアント定義ファイルに指定できるパラメータの一覧を次の表に示します。

なお、必ず指定するパラメータを太字で表記しています。

環境依存パラメータの設計については、「8. 環境依存パラメータを設計する (EADs クライアント)」を参照してください。

チューニングパラメータの設計については、「9. チューニングパラメータを設計する」を参照してください。

表 23-7 クライアント定義ファイルに指定できるパラメーター一覧

項番	分類	パラメータ名	指定する内容
1	通信 (「8.3 通信に依存するパラメータの設計」参照)	<b>eads.client.node.list</b>	EADs クライアントの初期設定時に接続する、接続先 EADs サーバを識別するための名称 (任意の名称)
2		<b>eads.client.&lt;接続先 EADs サーバ&gt;.address</b>	EADs クライアントの初期設定時に接続する、接続先 EADs サーバの IP アドレスまたはホスト名
3		<b>eads.client.&lt;接続先 EADs サーバ&gt;.port</b>	EADs クライアントの初期設定時に接続する、接続先 EADs サーバのポート番号
4	ログファイル (「8.4 ログファイルに依存するパラメータの設計」参照)	<b>eads.logger.dir</b>	ログファイルの出力先ディレクトリのパス
5		<b>eads.logger.message.filesize</b>	メッセージログの 1 ファイル当たりのファイルサイズ
6		<b>eads.logger.message.filenum</b>	メッセージログのファイル数
7		<b>eads.logger.message.console.enable</b>	メッセージログの標準出力への出力を有効にするか、無効にするか

項番	分類	パラメタ名	指定する内容
8		<code>eads.logger.commTrace.enable</code>	通信トレースの出力を有効にするか、無効にするか
9		<code>eads.logger.commTrace.filesize</code>	通信トレースの1ファイル当たりのファイルサイズ
10		<code>eads.logger.commTrace.filenum</code>	通信トレースのファイル数
11		<code>eads.logger.commTrace.commNo.prefix</code>	PRFトレースに出力されるRootAP情報の通信番号(RootAP CommNo.)に含めるプレフィックス
12	メモリ・バッファ (「9.1 メモリ・バッファに関連するパラメタの設計」参照)	<code>eads.connection.bufferSize</code>	データの送受信バッファサイズ
13	スレッドプールおよび接続プール	<code>eads.connectionPool.poolSize</code>	同一接続先に対してプールしておく接続の最大個数
14	(「9.2 スレッドプールおよび接続プールに関連するパラメタの設計」参照)	<code>eads.connectionPool.exceedMaxSizeError.enable</code>	接続がすでに最大個数に達していて、すべて使用中の場合に、EADsクライアントからのリクエストをエラーにするかどうか
15	タイムアウト (「9.3 タイムアウトに関連するパラメタの設計」参照)	<code>eads.connection.send.timeout</code>	クラスタ構成情報の更新確認やデータ送信のタイムアウト時間
16		<code>eads.connection.receive.timeout</code>	クラスタ構成情報の更新確認やデータ受信のタイムアウト時間
17		<code>eads.client.clusterInfo.update.interval</code>	クラスタ構成情報の更新確認を行う間隔

## 23.2 コマンド一覧

コマンドの一覧を次の表に示します。

表 23-8 コマンド一覧

項番	利用シーン		コマンド名	実行対象
1	開始	EADs サーバを開始する	ezstart	EADs サーバ
2		EADs サーバをフォアグラウンドで開始する	ezserver	EADs サーバ
3	閉塞化と閉塞解除	クラスタを閉塞状態にする	eztool close	クラスタ
4		クラスタの閉塞状態を解除する	eztool open	クラスタ
5	状態確認	クラスタの状態を確認する	eztool status	クラスタ※1
6		キャッシュの一覧を表示する	eztool listcache	クラスタ
7		ストアデータファイルの一覧を表示する	eztool listesd	クラスタ
8		データの格納先を表示する	eztool getposition	クラスタ
9	排他制御	排他を解除する	eztool unlock	クラスタ
10	メモリ・ディスク管理	キャッシュを作成する	eztool createcache	クラスタ
11		キャッシュを削除する	eztool deletetecache	クラスタ
12	データ移行	データを書き出す	eztool export	クラスタ※1
13		データを読み込む	eztool import	クラスタ
14		ストアデータファイルを削除する	eztool deleteesd	クラスタ
15	データ操作	指定したデータを格納する	eztool put	EADs サーバ
16		指定したデータを取得する	eztool get	EADs サーバ
17		指定したデータを削除する	eztool remove	EADs サーバ
18	ユーザファンクション	ユーザファンクションの実行可否を表示する	eztool listfunc	クラスタ
19		ユーザファンクションを実行する	eztool execfunc	クラスタ※1※2
20	ユーザイベントハンドラ	ユーザイベントハンドラの実行可否を表示する	eztool listevent	クラスタ
21	永続化	永続データに関する情報の一覧を表示する	eztool listecf	クラスタ※1
22		キャッシュを再開する	eztool resume	クラスタ
23		キャッシュデータファイルからデータを再配置する	eztool importecf	クラスタ
24		キャッシュファイルを削除する	eztool deleteecf	クラスタ



項番	利用シーン		コマンド名	実行対象
25		キャッシュデータファイルをコンパクションする	eztool compaction	EADs サーバ※3
26	トラブルシュート	PRF トレース情報を編集出力する	eztool cprfed	EADs サーバ
27		スレッドダンプを出力する	eztool threaddump	EADs サーバ
28		ログ, 設定情報, ハードウェア情報, およびネットワーク情報を収集する	eztool snapshot	EADs サーバ
29	終了	クラスタ内の全 EADs サーバを終了する	eztool stop	クラスタ
30		EADs サーバを強制終了する	eztool forcestop	EADs サーバ
31		EADs サーバを縮退させる	eztool isolate	EADs サーバ

#### 注※1

-s または --single オプションを指定すると, コマンドを実行した EADs サーバが対象となります。

#### 注※2

-g または --group オプションを指定すると, 指定したグループ名のハッシュ値から決定した EADs サーバが対象となります。

#### 注※3

--cache オプションを指定すると, コマンドを実行したキャッシュが対象となります。

--range オプションを指定すると, コマンドを実行したレンジが対象となります。

## 23.3 API 一覧

AP を作成する言語ごとの API の一覧について説明します。

### 23.3.1 Java 言語のライブラリが提供する API

#### (1) Java 言語のクライアントライブラリが提供するクラス

Java 言語のクライアントライブラリでは、EADs の API として次の表に示すクラスを提供します。

表 23-9 Java 言語のクライアントライブラリで使用できるクラスとパッケージ名

項番	クラス名	機能	パッケージ名
1	Cache	データを操作するためのクラスです。	com.hitachi.software.xeads.client.api
2	CacheManager	キャッシュを管理するためのクラスです。	
3	Node	EADs サーバの情報を取得するためのクラスです。	
4	CacheException	Cache クラス、および CacheManager クラスに対する操作が失敗した場合に返却する例外クラスです。	
5	InitializeException	CacheManager クラスの初期化でエラーが発生した場合に返却される CacheException のサブクラスです。	
6	InternalClientException	EADs クライアントで内部エラーが発生した場合に返却される CacheException のサブクラスです。	
7	InternalServerException	EADs サーバで内部エラーが発生した場合に返却される CacheException のサブクラスです。	
8	ServerCommunicationException	通信障害が発生した場合に返却される CacheException のサブクラスです。	
9	UserOperationException	不正なユーザ操作によってエラーが発生した場合に返却される CacheException のサブクラスです。	

## (2) ユーザファンクションで使用できる API

ユーザファンクションで使用できる API を次の表に示します。

表 23-10 ユーザファンクションで使用できる API

項番	インタフェース名, クラス名, または列挙型	機能	パッケージ名
1	Function	EADs サーバで使用するユーザファンクションが実装する必要があるインタフェースです。	com.hitachi.software.xeads.func.Function
2	FunctionContext	ユーザファンクションの実行に必要な情報を Function インタフェースに渡すインタフェースです。	com.hitachi.software.xeads.func.FunctionContext
3	InitConfig	ユーザファンクションの初期化に使用したファンクション定義の情報を取得するインタフェースです。	com.hitachi.software.xeads.func.InitConfig
4	ClientInfo	実行した EADs クライアントの情報 (PRF トレースに出力される RootAP 情報に相当する情報) を取得するインタフェースです。	com.hitachi.software.xeads.common.ClientInfo
5	ServerInfo	EADs サーバの情報を取得するためのインタフェースです。	com.hitachi.software.xeads.common.ServerInfo
6	Store	ユーザファンクションの実行時にデータを操作するインタフェースです。	com.hitachi.software.xeads.func.store.Store
7	Group	ユーザファンクションを実行中の EADs サーバに属するグループを操作するためのインタフェースです。	com.hitachi.software.xeads.func.store.Group
8	Key	ユーザファンクションで使用できる API 内で, key を表現するインタフェースです。	com.hitachi.software.xeads.func.store.Key
9	UserLogger	ユーザログのインタフェースです。	com.hitachi.software.xeads.common.UserLogger
10	EADsStoreException	Group インタフェースに対する操作が失敗した場合に返却される例外クラスです。	com.hitachi.software.xeads.func.store.EADsStoreException
11	InternalServerException	EADs サーバで内部エラーが発生した場合に返却される EADsStoreException のサブクラスです。	com.hitachi.software.xeads.func.store.InternalServerException

項番	インタフェース名, クラス名, または列挙型	機能	パッケージ名
12	UserOperationException	不正なユーザ操作によってエラーが発生した場合に返却される EADsStoreException のサブクラスです。	com.hitachi.software.xeads.func.store.UserOperationException
13	列挙型 CacheType	キャッシュタイプを表す列挙型です。	com.hitachi.software.xeads.common.CacheType

### (3) ユーザイベントハンドラで使用できる API

ユーザイベントハンドラで使用できる API を次の表に示します。

表 23-11 ユーザイベントハンドラで使用できる API

項番	インタフェース名, アノテーション名, または列挙型名	機能	パッケージ名
1	EventHandler	EADs サーバで使用するユーザイベントハンドラが実装する必要があるインタフェースです。	com.hitachi.software.xeads.event.EventHandler
2	EventContext	EADs サーバ, EADs クライアント, および発生したイベントに関する情報を, ユーザイベントハンドラに渡すインタフェースです。	com.hitachi.software.xeads.event.EventContext
3	ClientInfo*	実行した EADs クライアントの情報 (PRF トレースに出力される RootAP 情報に相当する情報) を取得するインタフェースです。	com.hitachi.software.xeads.common.ClientInfo
4	UserLogger*	ユーザログのインタフェースです。	com.hitachi.software.xeads.common.UserLogger
5	OperationFilter	イベントが発生したデータ操作の種類に関して, ユーザイベントハンドラが動作する条件を指定するアノテーションです。	com.hitachi.software.xeads.event.annotation.OperationFilter
6	DataFilter	イベントが発生したデータのキャッシュ名, グループ名, または要素名に関して, ユーザイベントハンドラが動作する条件を指定するアノテーションです。	com.hitachi.software.xeads.event.annotation.DataFilter
7	Priority	ユーザイベントハンドラを実行する優先度を指定するアノテーションです。	com.hitachi.software.xeads.event.annotation.Priority

項番	インタフェース名, アノテーション名, または列挙型名	機能	パッケージ名
8	OperationType	イベントが発生したデータ操作の種類を表す列挙型です。	com.hitachi.software.xeads.event.OperationType
9	MethodType	実行した API の種類を表す列挙型です。	com.hitachi.software.xeads.event.MethodType
10	PriorityLevel	ユーザイベントハンドラを実行する優先度を表す列挙型です。	com.hitachi.software.xeads.event.PriorityLevel

注※

ClientInfo インタフェースについては、「[20.2.4 ClientInfo インタフェース](#)」を参照してください。  
 UserLogger インタフェースについては、「[20.2.9 UserLogger インタフェース](#)」を参照してください。

### 23.3.2 C 言語のクライアントライブラリが提供する関数

C 言語のクライアントライブラリでは、EADs の API として次の表に示す関数を提供します。

表 23-12 C 言語のクライアントライブラリで使用できる関数とヘッダファイル

項番	関数名	機能	ヘッダファイル
1	ead_init_client()	クライアント定義に従って、EADs クライアントの初期設定をします。	eads.h
2	ead_init_client_n()	EADs クライアント名およびクライアント定義に従って、EADs クライアントの初期設定をします。	
3	ead_start_cache()	キャッシュへのアクセスを開始します。指定したキャッシュにアクセスするためのハンドル（ポインタ）を取得します。	
4	ead_stop_cache()	指定したキャッシュへのアクセスを終了します。	
5	ead_terminate_client()	EADs クライアントを終了します。	
6	ead_put()	value を key に関連づけて格納します。	
7	ead_put_array_value()	複数の value を連結して、key に関連づけて格納します。	
8	ead_create()	新規に key を格納する場合だけ、value を key に関連づけて格納します。	

項番	関数名	機能	ヘッダファイル
9	<code>ead_update()</code>	指定した key がすでに格納されている場合だけ、value を key に関連づけて格納します (value を更新します)。	
10	<code>ead_replace()</code>	指定した key に関連づけられている value を、条件として指定した value 情報と比較します。value が一致している場合だけ、value を key に関連づけて格納します (value を置換します)。	
11	<code>ead_get()</code>	指定した key に関連づけられた value を取得します。	
12	<code>ead_remove()</code>	指定した key, および key に関連づけられた value を削除します。	
13	<code>ead_execute_function()</code>	グループを指定して、ユーザファンクションを実行します。	
14	<code>ead_execute_function_rt()</code>	グループおよび受信タイムアウト時間を指定して、ユーザファンクションを実行します。	
15	<code>ead_execute_node_function()</code>	EADs サーバを指定して、ユーザファンクションを実行します。	
16	<code>ead_execute_node_function_rt()</code>	EADs サーバおよび受信タイムアウト時間を指定して、ユーザファンクションを実行します。	
17	<code>ead_get_nodelist()</code>	EADs クライアントが保持している接続先 EADs サーバの情報を取得します。	
18	<code>ead_get_node()</code>	指定した key またはグループが格納されている接続先 EADs サーバの情報を取得します。	
19	<code>ead_get_slave_nodelist()</code>	データの複製先 EADs サーバの情報を取得します。	

## 23.4 クライアントライブラリが出力するエラーコード

---

クライアントライブラリが出力するエラーコードについて説明します。

### 23.4.1 Java 言語のクライアントライブラリが出力するエラーコード

Java 言語のクライアントライブラリが出力するエラーコードについては、「[20.1.4 CacheException クラス](#)」の「(5)(c)戻り値」を参照してください。

なお、ユーザ関クションのエラーコードについては、「[20.2.10 EADsStoreException クラス](#)」の「(5)(c)戻り値」を参照してください。

### 23.4.2 C 言語のクライアントライブラリが出力するエラーコード

C 言語のクライアントライブラリが出力するエラーコードについては、「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照してください。

## 23.5 最小値・最大値一覧

EADsの最小値・最大値の一覧を次の表に示します。

表 23-13 最小値・最大値一覧

項番	項目	最小値	最大値
1	1EADs サーバ当たりの、EADs クライアントからの接続数	1	1,024
2	1 クラスタを構成する EADs サーバ数	1※1	96
3	1 クラスタに作成できるキャッシュ数	0	64
4	1EADs サーバに格納できるデータ数※2	0	2,147,483,647
5	データの多重度	1	10
6	key の長さ (単位：文字数)	1	1,024
7	value のサイズ※3 (単位：バイト)	1	262,144
8	グループ名の長さ (単位：文字数)	1	1,022
9	キャッシュ名の長さ (単位：文字数)	1	32
10	グループ数	1	制限なし
11	グループの階層数	1	511
12	1 レンジ当たりのキャッシュデータファイルのサイズ (単位：メガバイト)	16	128
13	1 レンジ当たりのキャッシュデータファイルのファイル数	8	32,768
14	1 キャッシュデータファイルに格納できるデータ数	—	131,070

(凡例)

—：該当しません。

注※1

1 クラスタを構成する EADs サーバ数は、「データの多重度×2-1」以上であることが前提です。

注※2

多重化によってコピーされるデータ数も含まれます。

注※3

value のサイズについては、「15.2.2(2) value として指定できるデータ」を参照してください。



# 24

## メッセージ

この章では、出力されるメッセージを示し、その意味、および対策について説明します。

## 24.1 メッセージの記述形式

XXXXnnnnn-Y

メッセージテキスト

可変値に関する説明

説明

メッセージテキストに対する補足説明

対処

ユーザが実施する対処

「可変値に関する説明」および「対処」は、メッセージによって記述しないものもあります。各項目について、次の表に示します。

項目	説明
XXXXnnnnn	メッセージ ID を表します。次の要素で構成されます。 <b>XXXX</b> EADs のメッセージのプレフィックスである、KDEA が出力されます。 <b>nnnnn</b> EADs で管理しているメッセージ番号を表します。 それぞれのメッセージには、5桁の固有の番号が付いています。
Y	メッセージのレベルを表します。 メッセージのレベルは英字 1 文字で示します。 メッセージのレベルを示す文字とその意味を次に示します。 E (Error) エラーレベルのトラブルが発生したことを通知するメッセージです。 このメッセージが出力されたときは、処理を中断します。 W (Warning) 警告レベルのトラブルが発生したことを通知するメッセージです。 このメッセージが出力されたあとも処理を続行します。 I (Information) システムの動作を通知するメッセージです。 このメッセージが出力されたあとも処理を続行します。
メッセージテキスト	EADs が出力するメッセージテキストを表します。
可変値に関する説明	メッセージテキスト中の可変値に表示される情報を「xx....xx：表示される情報」（xx は英小文字）の形式で示します。 可変値に関する説明の記述例を次に示します。 (例) aa....aa：IP アドレス bb....bb：ポート番号

項目	説明
説明	メッセージが通知された要因やメッセージを出力した EADs の動作など、メッセージに対する補足説明を示します。
対処	ユーザが実施する対処を表します。

## 24.2 KDEA00001～KDEA01999

---

KDEA00001～KDEA01999 のメッセージとその対処方法について説明します。

### KDEA00001-I

```
The server will now start. (server name = aa....aa)
```

aa....aa : EADs サーバ名 (運用ディレクトリ名)

#### 説明

EADs サーバを開始します。

### KDEA00002-I

```
The server will now shut down. (server name = aa....aa)
```

aa....aa : EADs サーバ名 (運用ディレクトリ名)

#### 説明

EADs サーバを終了します。

### KDEA00003-W

```
The server status was set to "ISOLATED".
```

#### 説明

EADs サーバ内で障害が発生したため、EADs サーバが縮退状態 (isolated) になりました。以後、EADs クライアントからのリクエストを受け付けることができません。

#### 対処

「[12.2.1 縮退状態が発生した場合](#)」を参照して対処してください。

### KDEA00004-E

```
Log initialization failed. (directory path = aa....aa)
```

aa....aa : ログの出力先パス名

#### 説明

ログの初期化に失敗しました。

#### 対処

次の要因が考えられます。

- ログの出力先ディレクトリの指定が不正である。

- ログの出力先ディレクトリまたはログファイルの権限などに問題がある。
- 要因を特定し、問題を取り除いたあと、EADs サーバを再起動してください。

## KDEA00005-E

A property file cannot be read. (file path = aa....aa)

aa....aa：定義ファイルの格納先パス名

### 説明

定義ファイルが読み込めません。

次の要因が考えられます。

- 定義ファイルがない。
- 定義ファイルが開けない。
- 格納先パス名がファイルではなくディレクトリを指している。

### 対処

標準出力、メッセージログまたは例外ログから要因を特定し、問題を取り除いたあと、EADs サーバを再起動してください。

## KDEA00006-E

The server failed to start. Startup of the server will now end.

### 説明

開始処理に失敗しました。

EADs サーバの開始処理中に問題が発生したため、EADs サーバの開始を中止します。

### 対処

直前に出力されたメッセージを参照して要因を特定し、問題を取り除いたあと、EADs サーバを再起動してください。

## KDEA00016-E

An attempt to open a port failed. (address = aa....aa:bb....bb)

aa....aa：IP アドレス

bb....bb：ポート番号

### 説明

通信ポートが開けません。

次の要因が考えられます。

- 指定された IP アドレスが不正である。

- 指定されたポート番号がすでに使用されている。
- 異なるパラメタに同じポート番号を指定している。
- 指定されたポート番号が well-known ポートであり，ユーザに権限がない。

## 対処

次に示すサーバ定義のパラメタの指定値を見直してください。

- `eads.server.address`
- `eads.server.port`
- `eads.management.port`

例外ログから要因を特定し，問題を取り除いたあと，EADs サーバを再起動してください。

## KDEA00017-E

PRF trace failed to start. (return code = aa....aa)

aa....aa : PRF トレースの初期化処理の戻り値

## 説明

PRF トレースの初期化処理に失敗しました。

## 対処

次の表に示す PRF トレースの初期化処理の戻り値を参照して，対処してください。

PRF トレースの初期化処理の戻り値	意味
-1	環境変数 PRFSPOOL が設定されていません。 または，環境変数 PRFSPOOL に設定したディレクトリがありません。
-2	PRF デーモンのサービス開始処理でエラーが発生しました。 PRF デーモンが起動していないおそれがあります。 PRF デーモンが正常に起動しているかどうかを確認してください。EADs サーバが使用する PRF デーモンの識別子は，REQ_PRF_ID（固定値）です。
-3	PRF デーモンのサービス開始処理でエラーが発生しました（システムコールのエラーなど）。 <サーバ定義の <code>eads.logger.dir</code> パラメタに指定したディレクトリ>/spool/log/REQ_PRF_ID 下にあるファイルを参照して，パフォーマンストレーサのメッセージを調査し，要因を特定してから問題を取り除いてください。
-4	PRF デーモンで必要となるライブラリのロードに失敗しました。次の要因が考えられます。 <ul style="list-style-type: none"> <li>• パフォーマンストレーサがインストールされていない。</li> <li>• クラスパスに PRF トレース出力ライブラリが指定されていない。</li> <li>• 環境変数に誤りがある。</li> </ul>

要因を特定し，問題を取り除いたあと，EADs サーバを再起動してください。

## KDEA00020-I

Initialization of the server will now start. (version = aa....aa)

aa....aa : EADs サーバのバージョン

### 説明

EADs サーバの初期化を開始します。

## KDEA00021-I

Initialization of the server was completed.

### 説明

EADs サーバの初期化が終了しました。

## KDEA00028-E

An exception occurred on the server. (exception = aa....aa)

aa....aa : 発生した例外（またはエラー）

### 説明

EADs サーバ内で例外（またはエラー）が発生しました。

EADs サーバを終了します。

### 対処

例外ログ，および直前に出力されたメッセージを確認して要因を特定し，原因を取り除いたあと，EADs サーバを再起動してください。

## KDEA00029-E

The specified cache was not found. (cache name = aa....aa)

aa....aa : キャッシュ名

### 説明

指定されたキャッシュが見つかりません。存在しないキャッシュに対する操作が行われました。

### 対処

キャッシュを作成するか，処理を見直してください。

## KDEA00031-E

The thread stopped. (thread name = aa....aa)

aa....aa : スレッド名

## 説明

EADs サーバ内で問題が発生し、スレッドが停止しました。

## 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA00032-E

```
An exception occurred while waiting for a connection request. (local = aa....aa:bb....bb)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

## 説明

接続要求の待機中に例外が発生しました。

## 対処

次の要因が考えられます。

- 通信先の EADs クライアントで問題が発生した。
- 通信先の EADs サーバで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。

例外ログから要因を特定し、問題を取り除いてください。

## KDEA00033-W

```
The number of received connection requests exceeds the maximum number of simultaneous connections. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd, max connections = ee....ee)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス (java.net.InetAddress の文字列表現)

dd....dd : 接続元のポート番号

ee....ee : 最大同時接続数

## 説明

最大同時接続数を超える接続要求を受信しました。

最大同時接続数を超えた接続に対してはエラーを返し、通信を切断します。

なお、このメッセージは次の場合にも出力されることがあります。



- EADs クライアントを稼働させたまま、クラスタ内の全 EADs サーバを再起動した場合
- EADs サーバの一時的な負荷上昇によって、EADs クライアント・EADs サーバ間の接続がタイムアウトした場合

## 対処

最大同時接続数（サーバ定義の `eads.server.max_connections` パラメタ）を見直してください。

## KDEA00036-E

```
An exception occurred during request handling. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス (java.net.InetAddress の文字列表現)

dd....dd : 接続元のポート番号

## 説明

リクエストの処理中に例外が発生しました。

## 対処

次の要因が考えられます。

- 通信先の EADs クライアントで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。

要因を特定し、問題を取り除いてください。

## KDEA00037-I

```
The PRF trace is disabled.
```

## 説明

PRF トレースは無効です。PRF トレースは出力されません。

## KDEA00039-E

```
An exception or error occurred on the server. (details = aa....aa)
```

aa....aa : 詳細情報

## 説明

リクエスト処理中に EADs サーバ内で例外（またはエラー）が発生しました。そのため、通信を切断します。

## 対処

例外ログから要因を特定し、問題を取り除いてください。  
それでも解決しない場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA00041-I

```
The port will now close. (address = aa....aa:bb....bb)
```

aa....aa : IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : ポート番号

## 説明

ポートを閉じます。

## KDEA00042-I

```
The library was added to the class path. (file path = aa....aa)
```

aa....aa : ライブラリファイルのパス名

## 説明

ライブラリをクラスパスに追加しました。

## KDEA00045-E

```
A received request is not supported by the server. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd)
```

aa....aa : EADs サーバの IP アドレス

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス

dd....dd : 接続元のポート番号

## 説明

EADs サーバでサポートされていないリクエストを受信しました。  
リクエストは処理できません。

## 対処

次の要因が考えられます。

- 不正な EADs クライアントが接続されている。
- 異なるバージョンの EADs クライアントが接続されている。

正しい EADs クライアントが使用されているか確認してください。

## KDEA00046-E

```
The packet size of a received request exceeds the maximum. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd, maximum size (bytes) = ee....ee, packet size (bytes) = ff....ff)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス (java.net.InetAddress の文字列表現)

dd....dd : 接続元のポート番号

ee....ee : 最大サイズ (単位: バイト)

ff....ff : 受信したパケットサイズ (単位: バイト)

### 説明

受信したリクエストのパケットサイズが最大値を超えています。  
リクエストは処理されません。

### 対処

次に示すサーバ定義のパラメタの指定値を見直してください。

- [eads.server.max.datasize.enable](#)
- [eads.server.max.datasize](#)

## KDEA00047-E

```
An error occurred during reception of a request. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス (java.net.InetAddress の文字列表現)

dd....dd : 接続元のポート番号

### 説明

リクエストの受信処理中にエラーが発生しました。

## 対処

次の要因が考えられます。

- 通信先の EADs クライアントで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。

例外ログから要因を特定し、問題を取り除いてください。

## KDEA00048-E

```
An error occurred during the sending of a response. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス (java.net.InetAddress の文字列表現)

dd....dd : 接続元のポート番号

## 説明

レスポンスの送信処理中にエラーが発生しました。

## 対処

次の要因が考えられます。

- 通信先の EADs クライアントで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。

例外ログから要因を特定し、問題を取り除いてください。

## KDEA00049-I

```
A redirect notification was sent to the client. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd, redirect = ee....ee:ff....ff)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

cc....cc : 接続元の IP アドレス (java.net.InetAddress の文字列表現)

dd....dd : 接続元のポート番号

ee....ee : リダイレクト先の IP アドレス (java.net.InetAddress の文字列表現)

ff...ff：リダイレクト先のポート番号

#### 説明

EADs クライアントにリダイレクトを通知します。

#### KDEA00050-E

An unexpected initialization error occurred.

#### 説明

予期しない初期化エラーが発生しました。

#### 対処

弊社問い合わせ窓口にご連絡ください。

#### KDEA00051-E

The content of the request is invalid.

#### 説明

リクエストの内容が不正です。

#### 対処

次の要因が考えられます。

- 通信先の EADs クライアントで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。

要因を特定し、問題を取り除いてください。

#### KDEA00052-E

An unexpected exception occurred.

#### 説明

予期しない例外が発生しました。

#### 対処

弊社問い合わせ窓口にご連絡ください。

#### KDEA00053-E

Loading of the library file failed.

## 説明

ライブラリファイルの読み込みに失敗しました。

## 対処

次の要因が考えられます。

- ライブラリファイルが不正である。
- ライブラリファイルまたはディレクトリの権限などに問題がある。

要因を特定し、問題を取り除いてください。

## KDEA00054-I

```
A server port was opened. (local = aa....aa:bb....bb)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

## 説明

ポートを開きました。

## KDEA00055-I

```
The server will now start accepting requests. (local = aa....aa:bb....bb)
```

aa....aa : EADs サーバの IP アドレス (java.net.InetAddress の文字列表現)

bb....bb : EADs サーバのポート番号

## 説明

リクエスト受付を開始します。

## KDEA00056-E

```
An unexpected exception occurred. (detail = aa....aa)
```

aa....aa : 保守情報

## 説明

予期しない例外が発生しました。

## 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA00057-E

```
An attempt to allocate the data area has failed. (type = aa....aa, data size = bb....bb)
```

aa....aa：種別

bb....bb：確保しようとしたメモリ領域のサイズ

## 説明

メモリの空き領域が不足しているため、メモリの確保に失敗しました。

種別ごとの要因を次の表に示します。

種別	要因
store	value を格納するメモリの空き領域が不足しています。

## 対処

種別ごとに次の表に示す対処を検討してください。

種別	対処
store	起動設定の <code>eads.java.external.heapsize</code> パラメタの指定値を大きくしてください。

## KDEA00058-E

The input data are too long. (type = aa....aa, request = bb....bb, limit = cc....cc)

aa....aa：種別

bb....bb：要求サイズ

cc....cc：上限サイズ

## 説明

入力データの要求サイズが上限サイズを超えています。

種別ごとの要因を次の表に示します。

種別	要因
store	value のサイズに対して、起動設定の <code>eads.java.external.heapsize</code> パラメタの指定値が小さいです。 または、value のサイズが指定できる最大値を超えています。
dist	更新操作の履歴のサイズに対して、サーバ定義の <code>eads.consensus.external.heapsize</code> パラメタの指定値が小さいです。

## 対処

value のサイズが指定できる最大値を超えないようにしてください。

value のサイズが指定できる最大値を超えていない場合は、種別ごとに次の表に示す対処を検討してください。

種別	対処
store	起動設定の <code>eads.java.external.heapsize</code> パラメタの指定値を大きくしてください。

種別	対処
dist	サーバ定義の <code>eads.consensus.external.heapsize</code> パラメタの指定値を大きくしてください。 このパラメタの指定値を大きくする場合は、起動設定の <code>eads.java.external.heapsize</code> パラメタの指定値も大きくしてください。

## KDEA00060-W

The fragmentation of data occurred. (type = aa....aa, count = bb....bb)

aa....aa：種別

bb....bb：データのフラグメントの発生回数

### 説明

Explicit ヒープに格納されるデータのフラグメントが発生しました。

種別ごとの要因を次の表に示します。

種別	要因
store	value のフラグメントが発生しました。
dist	更新操作の履歴のフラグメントが発生しました。

なお、出力される発生回数は、前回このメッセージが出力されてからの回数です。

### 対処

このメッセージが頻繁に出力される場合、Explicit ヒープに十分な空き容量がないためにフラグメントが発生し、性能が低下しているおそれがあります。種別ごとに次の表に示す対処を検討してください。

種別	対処
store	起動設定の <code>eads.java.external.heapsize</code> パラメタの指定値を大きくしてください。
dist	サーバ定義の <code>eads.consensus.external.heapsize</code> パラメタの指定値を大きくしてください。 このパラメタの指定値を大きくする場合は、起動設定の <code>eads.java.external.heapsize</code> パラメタの指定値も大きくしてください。

## KDEA00061-E

The size of key exceeds the maximum. (maximum size (bytes) = aa....aa, key size (bytes) = bb....bb)

aa....aa：最大サイズ（単位：バイト）

bb....bb：key サイズ（単位：バイト）

### 説明

指定した key のサイズがクラスタに格納できる key の最大サイズを超えています。



## 対処

指定した key のサイズを見直してください。

クラスタ定義の `eads.server.max_key_size` パラメタの指定値を見直してください。

## KDEA00511-I

Initialization of the user event handler is complete. (event handler name = aa....aa)

aa....aa : ユーザイベントハンドラ名

## 説明

ユーザイベントハンドラの初期化が完了しました。

## KDEA00512-W

An exception or error occurred in the init method of the user event handler. (event handler name = aa....aa)

aa....aa : ユーザイベントハンドラ名

## 説明

ユーザイベントハンドラの `init()` で例外（またはエラー）が発生しました。

## 対処

ユーザ例外ログを確認し、問題を取り除いてください。

## KDEA00513-W

Generation of a user event handler instance failed. (event handler name = aa....aa)

aa....aa : ユーザイベントハンドラ名

## 説明

ユーザイベントハンドラのインスタンスの生成に失敗しました。

## 対処

次の要因が考えられます。

- ユーザイベントハンドラのインスタンス生成時に例外（またはエラー）が発生した。
- ユーザイベントハンドラのデフォルトコンストラクタのアクセス権がない。
- ユーザイベントハンドラがインスタンス生成できない形式である。

ユーザ例外ログを確認して問題を取り除いたあと、EADs サーバを再起動してください。

## KDEA00514-I

Processing to terminate the user event handler is complete. (event handler name = aa....aa)

aa....aa : ユーザイベントハンドラ名

#### 説明

ユーザイベントハンドラの終了処理が完了しました。

### KDEA00515-W

Processing to terminate the user event handler failed. (event handler name = aa....aa)

aa....aa : ユーザイベントハンドラ名

#### 説明

ユーザイベントハンドラの終了処理に失敗しました (ユーザイベントハンドラの destroy() で例外 (またはエラー) が発生しました)。

#### 対処

ユーザ例外ログを確認し、問題を取り除いてください。

### KDEA00516-E

An exception or error occurred on the user event handler. (event handler name = aa....aa)

aa....aa : ユーザイベントハンドラ名

#### 説明

ユーザイベントハンドラ内で例外 (またはエラー) が発生しました (ユーザイベントハンドラの execute() 実行中に例外 (またはエラー) が発生しました)。

#### 対処

ユーザ例外ログを確認し、問題を取り除いてください。

### KDEA00517-E

An exception or error occurred in the user event handler call processing. (call event handler name = aa....aa, details = bb....bb)

aa....aa : 呼び出し対象のユーザイベントハンドラ名 (特定できない場合は null)

bb....bb : 詳細情報

#### 説明

ユーザイベントハンドラの呼び出し処理で例外 (またはエラー) が発生しました。

#### 対処

詳細情報および直前に出力されたメッセージから要因を特定し、問題を取り除いてください。

## KDEA00518-W

The annotation of a user event handler is invalid. (event handler name = aa....aa, details = bb....bb)

aa....aa：ユーザイベントハンドラ名

bb....bb：詳細情報

### 説明

ユーザイベントハンドラのアノテーションが不正です。

### 対処

次の表に示す詳細情報を参照して、問題を取り除いてください。

詳細情報	意味
The length of the specified cache name is outside the valid range. (index = <インデックス>)	キャッシュ名の長さが指定できる範囲を超えています。
The specified cache name includes an invalid character. (index = <インデックス>)	キャッシュ名に指定できない文字が含まれています。
The length of the specified group name is outside the valid range. (index = <インデックス>)	グループ名の長さが指定できる範囲を超えています。
The specified group name includes an invalid character. (index = <インデックス>)	グループ名に指定できない文字が含まれています。
The length of the specified element name is outside the valid range. (index = <インデックス>)	要素名の長さが指定できる範囲を超えています。
The specified element name includes an invalid character. (index = <インデックス>)	要素名に指定できない文字が含まれています。
The specified element name includes a group separator. (index = <インデックス>)	要素名に区切り文字が含まれています。
The combined length of the specified group name and specified element name is outside the valid range.	グループ名と要素名を組み合わせた長さが、指定できる範囲を超えています。

### 注

インデックスは cache, group, element の配列で何番目に指定した値が不正かを示します。インデックスは 0 から始まります（最初に指定した値のインデックスは 0 です）。

#### キャッシュ名が不正の場合

DataFilter アノテーションの cache に指定した値を見直してください。

#### グループ名が不正の場合

DataFilter アノテーションの group に指定した値を見直してください。

#### 要素名が不正の場合

DataFilter アノテーションの element に指定した値を見直してください。

グループ名と要素名を組み合わせた長さが不正の場合

DataFilter アノテーションの group と element に指定したすべてのグループ名と要素名の長さを見直してください。

#### KDEA00519-I

```
Initialization of the user event-handler library will now start. (file path = aa....aa)
```

aa....aa : ファイルパス

#### 説明

ユーザイベントハンドラライブラリの初期化を開始します。

#### KDEA00520-I

```
Initialization of the user event-handler library is complete. (file path = aa....aa, success = bb....bb, failure = cc....cc)
```

aa....aa : ファイルパス

bb....bb : 初期化に成功したユーザイベントハンドラの数

cc....cc : 初期化に失敗したユーザイベントハンドラの数

#### 説明

ユーザイベントハンドラライブラリの初期化が完了しました。

#### KDEA00521-I

```
No user event-handler library was found.
```

#### 説明

ユーザイベントハンドラライブラリが存在しないため、ユーザイベントハンドラの初期化は行われません。

#### KDEA00522-E

```
An exception or error occurred in the initialization processing of the user event handler. (details = aa....aa)
```

aa....aa : 詳細情報

#### 説明

ユーザイベントハンドラの初期化処理で例外（またはエラー）が発生しました。

## 対処

詳細情報および例外ログから要因を特定し、問題を取り除いたあと、EADs サーバを再起動してください。

### KDEA01004-I

```
Initialization of the user function is complete. (function name = aa....aa)
```

aa....aa : ユーザファンクション名

## 説明

ユーザファンクションの初期化が完了しました。

### KDEA01005-W

```
An exception or error occurred in the init method of the user function. (function name = aa....aa)
```

aa....aa : ユーザファンクション名

## 説明

ユーザファンクションの init() で例外（またはエラー）が発生しました。

## 対処

ユーザ例外ログを確認し、問題を取り除いたあと、EADs サーバを再起動してください。

### KDEA01009-I

```
Processing to terminate the user function is complete. (function name = aa....aa)
```

aa....aa : ユーザファンクション名

## 説明

ユーザファンクションの終了処理が完了しました。

### KDEA01010-W

```
Processing to terminate the user function failed. (function name = aa....aa)
```

aa....aa : ユーザファンクション名

## 説明

ユーザファンクションの終了処理に失敗しました（ユーザファンクションの destroy() で例外（またはエラー）が発生しました）。

## 対処

ユーザ例外ログを確認し、問題を取り除いてください。

## KDEA01011-E

An exception or error occurred in the user function. (function name = aa....aa)

aa....aa：ユーザファンクション名

### 説明

ユーザファンクション内で例外（またはエラー）が発生しました（ユーザファンクションの execute() 実行中に例外（またはエラー）が発生しました）。

### 対処

ユーザ例外ログを確認し、問題を取り除いてください。

## KDEA01012-W

Generation of a user function instance failed. (function name = aa....aa)

aa....aa：ユーザファンクション名

### 説明

ユーザファンクションのインスタンスの生成に失敗しました。

### 対処

次の要因が考えられます。

- ユーザファンクションのインスタンス生成時に例外（またはエラー）が発生した。
- ユーザファンクションのデフォルトコンストラクタのアクセス権がない。
- ユーザファンクションがインスタンス生成できない形式である。

ユーザ例外ログを確認し、問題を取り除いたあと、EADs サーバを再起動してください。

## KDEA01016-E

An exception or error occurred in the user function call processing. (function name = aa....aa, details = bb....bb)

aa....aa：呼び出し対象のユーザファンクション名

bb....bb：詳細情報

### 説明

ユーザファンクションの呼び出し処理で例外（またはエラー）が発生しました。

### 対処

詳細情報から要因を特定し、問題を取り除いてください。

## KDEA01017-E

The specified user function was not found. (function name = aa....aa)

aa....aa：ユーザファンクション名

### 説明

指定されたユーザファンクションが見つかりません。

### 対処

次の要因が考えられます。

- 指定された名称のユーザファンクションが EADs サーバに存在しない。
- 無効なユーザファンクションを実行しようとした。

ユーザファンクション名の指定に誤りがないか確認してください。

ユーザファンクション名の指定に誤りがない場合、指定されたユーザファンクションが EADs サーバに有効なユーザファンクションとして登録されているか確認してください。

## KDEA01018-E

An attempt to serialize the user function execution results failed. (function name = aa....aa, details = bb....bb)

aa....aa：呼び出し対象のユーザファンクション名

bb....bb：詳細情報

### 説明

ユーザファンクションの実行結果として戻り値に指定されたオブジェクトはシリアライズできませんでした。

### 対処

指定された値がシリアライズできるオブジェクトかどうか確認してください。

シリアライズできないオブジェクトの場合は、シリアライズできるオブジェクトに変更してください。

## KDEA01019-E

An attempt to deserialize the argument failed. (function name = aa....aa, details = bb....bb)

aa....aa：呼び出し対象のユーザファンクション名

bb....bb：詳細情報

### 説明

EADs クライアントから送信された引数はデシリアライズできませんでした。

## 対処

ユーザ関数ライブラリファイルに含まれるクラスを確認し、EADs クライアントで指定された引数のオブジェクトがデシリアライズできるかどうかを確認してください。

クラスが不足している場合、ユーザ関数ライブラリまたは共通ライブラリにクラスを追加したあと、EADs サーバを再起動してください。

## KDEA01021-E

```
The specified cache was not found. (function name = aa....aa, cache name = bb....bb)
```

aa....aa：呼び出し対象のユーザ関数名

bb....bb：キャッシュ名

## 説明

存在しないキャッシュに対する操作が行われました。

## 対処

キャッシュを作成するか、または指定したキャッシュ名を見直してください。

## KDEA01022-I

```
Initialization of the user function library will now start. (file path = aa....aa)
```

aa....aa：ファイルパス

## 説明

ユーザ関数ライブラリの初期化を開始します。

## KDEA01023-I

```
Initialization of the user function library is complete. (file path = aa....aa, success = bb....bb, failure = cc....cc)
```

aa....aa：ファイルパス

bb....bb：初期化に成功したユーザ関数の数

cc....cc：初期化に失敗したユーザ関数の数

## 説明

ユーザ関数ライブラリの初期化が完了しました。

## KDEA01024-I

```
No user function library was found.
```



## 説明

ユーザファンクションライブラリが存在しないため、ユーザファンクションの初期化は行われません。

## KDEA01025-I

```
"user_function.jar" will not be initialized because a file called "user-function.jar" also exists.
```

## 説明

ファイル名が「user-function.jar」のユーザファンクションが存在するため、ファイル名が「user\_function.jar」のユーザファンクションは初期化できません。

## KDEA01026-I

```
A user function cannot be loaded because a user function with the same fully-qualified class name already exists. (function name = aa....aa)
```

aa....aa：ユーザファンクション名

## 説明

同じ完全修飾クラス名のユーザファンクションがすでに存在するため、対象のユーザファンクションは読み込みません。

## KDEA01027-E

```
An exception or error occurred in the initialization processing of the user function. (details = aa....aa)
```

aa....aa：詳細情報

## 説明

ユーザファンクションの初期化処理で例外（またはエラー）が発生しました。

## 対処

詳細情報および例外ログから要因を特定し、問題を取り除いたあと、EADs サーバを再起動してください。

## KDEA01901-I

```
WRITING (cache = aa....aa, range = bb....bb, ecf = cc....cc, remain = dd....dd)
```

aa....aa：キャッシュ名

bb....bb：レンジ ID

cc....cc：キャッシュデータファイルのパス

dd....dd：未使用のキャッシュデータファイル数

## 説明

キャッシュデータファイルへの書き込みを開始しました。

### KDEA01902-I

```
COMPACTION (cache = aa....aa, range = bb....bb, ecf = cc....cc, size (total = dd....dd,
relocated = ee....ee, valid = ff....ff -> gg....gg), count (total = hh....hh, relocated = ii....ii, valid
= kk....kk -> mm....mm), time = nn....nn, remain = pp....pp)
```

aa....aa：キャッシュ名

bb....bb：レンジ ID

cc....cc：コンパクション対象のキャッシュデータファイルのパス（絶対パス）

dd....dd：ファイルの使用済み領域サイズ（バイト）

ee....ee：移動した量（バイト）

ff....ff：コンパクション前の有効データ量（バイト）

gg....gg：コンパクション後の有効データ量（バイト）

hh....hh：無効データも含めたすべてのデータ数

ii....ii：移動した数

kk....kk：コンパクション前の有効データ数

mm....mm：コンパクション後の有効データ数

nn....nn：処理時間（ミリ秒）

pp....pp：未使用のキャッシュデータファイル数

## 説明

キャッシュデータファイルに対してコンパクションを実行しました。

### KDEA01903-I

```
FILE INFO (cache = aa....aa, range = bb....bb, ecf = cc....cc, size = dd....dd, allocated size =
ee....ee)
```

aa....aa：キャッシュ名

bb....bb：レンジ ID

cc....cc：キャッシュデータファイルのパス

dd....dd：ファイルサイズ

ee....ee：実際に割り当てられたサイズ

## 説明

キャッシュデータファイルに実際に割り当てられた領域のサイズが、ファイルサイズと異なります。

## 24.3 KDEA02000～KDEA02999

---

KDEA02000～KDEA02999 のメッセージとその対処方法について説明します。

### KDEA02001-I

```
Initialization of the client library will now start. (version = aa....aa)
```

aa....aa : EADs クライアントのバージョン

#### 説明

クライアントライブラリの初期設定を開始します。

### KDEA02002-I

```
Initialization of the client library completed.
```

#### 説明

クライアントライブラリの初期設定が完了しました。

### KDEA02003-E

```
Initialization of the client library failed.
```

#### 説明

クライアントライブラリの初期設定に失敗しました。

#### 対処

「[22.2 クライアントライブラリのエラーコード \(C 言語\)](#)」を参照して、問題を取り除いてください。

### KDEA02008-I

```
Cache preparations finished. (cache name = aa....aa)
```

aa....aa : キャッシュ名

#### 説明

キャッシュを使用する準備ができました。

### KDEA02009-I

```
The cache will no longer be used. (cache name = aa....aa)
```

aa....aa : キャッシュ名

## 説明

キャッシュの使用を終了します。

## KDEA02011-I

```
The client library will now terminate.
```

## 説明

クライアントライブラリを終了します。

## KDEA02022-E

```
The connections to all servers making up the cluster failed.
```

## 説明

クラスタを構成するすべての EADs サーバとの接続に失敗しました。

## 対処

次の要因が考えられます。

- クラスタを停止した。
- 通信先の EADs サーバで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。
- タイムアウトの設定が適切でない。

要因を特定し、問題を取り除いてください。

## KDEA02023-I

```
The connection to the cluster was restored.
```

## 説明

クラスタとの接続が回復しました。

## KDEA02911-I

```
start RootAP(address = aa....aa, pid = bb....bb, no = cc....cc)
```

aa....aa : IP アドレス

bb....bb : PID (AP ごとに EADs クライアントが付与する番号)

cc....cc : 通番 (16 進数)

## 説明

EADs サーバとの通信を開始します。

## KDEA02912-I

```
end RootAP(address = aa....aa, pid = bb....bb, no = cc....cc)
```

aa....aa : IP アドレス

bb....bb : PID (AP ごとに EADs クライアントが付与する番号)

cc....cc : 通番 (16 進数)

## 説明

EADs サーバとの通信を終了します。

## 24.4 KDEA03000～KDEA03999

---

KDEA03000～KDEA03999 のメッセージとその対処方法について説明します。

### KDEA03001-I

```
Initialization of the client library will now start. (version = aa....aa)
```

aa....aa : EADs クライアントのバージョン

#### 説明

クライアントライブラリの初期設定を開始します。

### KDEA03002-I

```
The client library process will now end.
```

#### 説明

クライアントライブラリを終了します。

### KDEA03003-I

```
Cache preparations finished. (cache name = aa....aa)
```

aa....aa : キャッシュ名

#### 説明

キャッシュを使用する準備ができました。

### KDEA03004-I

```
Use of the cache will now end. (cache name = aa....aa)
```

aa....aa : キャッシュ名

#### 説明

キャッシュの使用を終了します。

### KDEA03006-E

```
Initialization of the client library failed. (error code = aa....aa)
```

aa....aa : エラーコード

#### 説明

クライアントライブラリの初期設定に失敗しました。

## 対処

「20.1.4 CacheException クラス」の「(5)(c)戻り値」を参照して、問題を取り除いてください。

### KDEA03019-I

```
Initialization of the client library completed.
```

## 説明

クライアントライブラリの初期設定が完了しました。

### KDEA03021-E

```
The connections to all servers making up the cluster failed.
```

## 説明

クラスタを構成するすべての EADs サーバとの接続に失敗しました。

## 対処

次の要因が考えられます。

- クラスタを停止した。
- 通信先の EADs サーバで問題が発生した。
- 通信先のホストで問題が発生した。
- ネットワークで問題が発生した。
- タイムアウトの設定が適切でない。

要因を特定し、問題を取り除いてください。

### KDEA03022-I

```
The connection to the cluster was restored.
```

## 説明

クラスタとの接続が回復しました。

### KDEA03911-I

```
start RootAP(address = aa....aa, pid = bb....bb, no = cc....cc)
```

aa....aa : IP アドレス

bb....bb : PID (AP ごとに EADs クライアントが付与する番号)

cc....cc : 通番 (16 進数)



## 説明

EADs サーバとの通信を開始します。

## KDEA03912-I

```
end RootAP(address = aa....aa, pid = bb....bb, no = cc....cc)
```

aa....aa : IP アドレス

bb....bb : PID (AP ごとに EADs クライアントが付与する番号)

cc....cc : 通番 (16 進数)

## 説明

EADs サーバとの通信を終了します。

## 24.5 KDEA04000~KDEA05999

---

KDEA04000~KDEA05999 のメッセージとその対処方法について説明します。

### KDEA04504-E

```
An attempt to initialize the distribution manager failed because an illegal property was detected. (property-file-name = aa....aa, property-name = bb....bb, property-value = cc....cc)
```

aa....aa：不正な定義ファイル名

bb....bb：不正なパラメタ名

cc....cc：不正なパラメタ指定値（値がない場合は空白）

#### 説明

不正なパラメタが検出されたため、クラスタが構築できませんでした。

#### 対処

メッセージ中に記されたパラメタを見直してください。

### KDEA04505-E

```
Initialization of the distribution manager failed because there is no cluster-definition file.
```

#### 説明

クラスタ定義ファイルが存在しないため、クラスタが構築できませんでした。

#### 対処

クラスタ定義ファイルの格納先を見直してください。

### KDEA04506-E

```
Initialization of the distribution-manager failed because a data discrepancy in the cluster-definition file was detected. (node = aa....aa)
```

aa....aa：他 EADs サーバのクラスタ定義ファイルと不一致がある EADs サーバの IP アドレスとポート番号

#### 説明

他 EADs サーバのクラスタ定義ファイルと不一致があるため、クラスタが構築できませんでした。

#### 対処

他 EADs サーバのクラスタ定義ファイルと内容が一致しているかどうかを確認してください。

このエラーが再発する場合は、他クラスタからハートビートを受けている場合があります。その場合は、異なるクラスタ間で配信先 IP アドレスまたは配信先ポート番号の指定（クラスタ定義の

`eads.cluster.heartbeat.address` パラメタおよび `eads.cluster.heartbeat.port` パラメタの指定) が重複していないか確認してください。

## KDEA04507-E

```
Initialization of the distribution manager has failed because a data discrepancy in the table-  
boot id was detected. (node = aa....aa)
```

aa....aa : ほかのクラスタを構成する EADs サーバの IP アドレスとポート番号

### 説明

ほかのクラスタを構成する EADs サーバが検出されたため、クラスタが構築できませんでした。

### 対処

全 EADs サーバを終了し、終了したことを確認した上で、全 EADs サーバを再起動してください。

このエラーが再発する場合は、他クラスタからハートビートを受けている場合があります。その場合は、異なるクラスタ間で配信先 IP アドレスまたは配信先ポート番号の指定 (クラスタ定義の `eads.cluster.heartbeat.address` パラメタおよび `eads.cluster.heartbeat.port` パラメタの指定) が重複していないか確認してください。

## KDEA04508-E

```
An attempt to initialize the distribution manager failed because a time-out occurred. (node-  
list = aa....aa)
```

aa....aa : 時間内に起動が完了しなかった EADs サーバの IP アドレスとポート番号のリスト

### 説明

指定された時間内に、クラスタを構成する全 EADs サーバの起動が完了しなかったため、クラスタが構築できませんでした。

### 対処

メッセージ中に記された EADs サーバが起動していたかどうかを確認してください。

それでも発生する場合は、クラスタを構成する全 EADs サーバが開始するまでの最大待ち時間の指定 (サーバ定義の `eads.cluster.boot.timeout` パラメタの指定) を次に示す時間の合計となるように、設定し直してください。

- 全 EADs サーバを起動する際、最初の EADs サーバを起動するためのコマンドを実行してから、最後の EADs サーバを起動するためのコマンドを実行するまでに掛かった時間
- ハートビート送信間隔 (サーバ定義の `eads.cluster.heartbeat.interval` パラメタの指定) を 5 倍した時間

## KDEA04509-E

An attempt to terminate the distribution manager in advance failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、クラスタが構築できませんでした。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04510-I

Termination of the distribution manager will now start.

### 説明

クラスタの終了処理を開始します。

## KDEA04511-I

Termination of the distribution-manager succeeded.

### 説明

クラスタの終了処理が完了しました。

## KDEA04512-E

Termination of the distribution manager failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、クラスタの終了処理が完了しませんでした。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04513-I

Initialization of the distribution manager in developer mode will now start.

### 説明

Developer の初期化処理を開始します。

## KDEA04514-I

Initialize of the distribution manager in developer mode succeeded.

### 説明

Developer の初期化処理が成功しました。

## KDEA04515-E

An attempt to initialize the distribution-manager on developer mode has failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、Developer の初期化処理が失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04517-I

Termination of the distribution-manager in developer mode will now start.

### 説明

Developer の終了処理を開始します。

## KDEA04518-I

Termination of the distribution manager in developer mode was successful.

### 説明

Developer の終了処理が成功しました。

## KDEA04519-E

Termination of the distribution manager in developer mode failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、Developer の終了処理が失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04520-E

Heartbeat reception processing for the distribution manager failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa：エラーコード

### 説明

予期しない例外が発生したため、ハートビートの受信処理が失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04521-E

Heartbeat reception processing for the distribution manager was terminated because an unknown message was received. (delivery-source-address = aa....aa, delivery-source-port = bb....bb, message-length = cc....cc, message-header = dd....dd)

aa....aa：配信元の IP アドレス

bb....bb：配信元のポート番号

cc....cc：メッセージ長

dd....dd：メッセージのヘッダ（最大 6 バイト分を 16 進形式で表示します）

### 説明

不明なメッセージを受信したため、ハートビートの受信処理が中断しました。

### 対処

他システムのメッセージ配信と混線しないように、クラスタ定義の `eads.cluster.heartbeat.address` パラメタおよび `eads.cluster.heartbeat.port` パラメタを見直してください。

## KDEA04522-E

Transmission of the heartbeat failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa：エラーコード

### 説明

予期しない例外が発生したため、ハートビートの送信が失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04523-E

An attempt by the server manager to accept a client's request failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、EADs クライアントからのリクエストの受け付けに失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04524-I

A cluster node (aa....aa) determined that another node (bb....bb) has stopped.

aa....aa : EADs サーバダウンを確定した EADs サーバの IP アドレスとポート番号

bb....bb : EADs サーバダウンと確定された EADs サーバの IP アドレスとポート番号

### 説明

EADs サーバ aa....aa が、EADs サーバ bb....bb のダウンを確定しました。

## KDEA04527-E

An attempt to notify the server manager of the stoppage of a local node failed because an unexpected exception occurred. (error-code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、自 EADs サーバへの停止通知処理が失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04528-E

Initialization of the distribution manager failed because a data discrepancy between the cluster-definition file and the server-definition file was detected.

### 説明

サーバ定義ファイルとクラスタ定義ファイルの内容に不一致があるため、クラスタが構築できませんでした。

## 対処

サーバ定義の `eads.server.address` パラメタおよび `eads.server.port` パラメタに指定された EADs サーバが、クラスタ定義の `eads.node.< EADs サーバ ID >.address` パラメタおよび `eads.node.< EADs サーバ ID >.port` パラメタに指定されているかどうかを確認してください。

## KDEA04531-E

```
Initialization of the distribution manager failed because a position was duplicated.
(duplication-node-1 = aa....aa, duplication-node-2 = bb....bb)
```

aa....aa：位置が重複した EADs サーバ 1 の IP アドレスとポート番号

bb....bb：位置が重複した EADs サーバ 2 の IP アドレスとポート番号

## 説明

クラスタを構成する EADs サーバの位置が重複したため、クラスタが構築できませんでした。

## 対処

クラスタ定義の `eads.node.< EADs サーバ ID >.position` パラメタに重複がないかどうかを確認してください。

## KDEA04532-E

```
Initialization of the distribution manager failed because the cluster definition file failed to be
read.
```

## 説明

クラスタ定義ファイルの読み込みに失敗したため、クラスタが構築できませんでした。

次の要因が考えられます。

- クラスタ定義ファイルが破損している。
- クラスタ定義ファイルが開けない。
- クラスタ定義ファイル内に不正な Unicode エスケープがある。

## 対処

システム運用管理者権限でクラスタ定義ファイルを開けることを確認してください。

また、クラスタ定義ファイルの内容が正しく読み取れるかどうかを確認してください。

## KDEA04535-E

```
Heartbeat transmission processing failed due to a network failure. (error-message = aa....aa)
```

aa....aa：JavaVM からスローされた例外のエラーメッセージ



## 説明

ネットワーク障害が発生したため、ハートビートの送信に失敗しました。

## 対処

クラスタ定義の `eads.cluster.heartbeat.address` パラメタに指定したマルチキャストアドレスに対して、エラーが発生した EADs サーバでマルチキャストの送信が成功することを確認してください。それでも繰り返し発生する場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA04536-E

Initialization of the distribution manager failed because an error occurred during initialization of a socket for heartbeat transmission processing. (error-message = aa....aa)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

## 説明

ハートビート送信用のソケットの初期化中に障害が発生したため、クラスタが構築できませんでした。

## 対処

ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。クラスタ定義の `eads.cluster.heartbeat.address` パラメタに指定したマルチキャストアドレスに対して、エラーが発生した EADs サーバでマルチキャストの送信が成功することを確認してください。それでも繰り返し発生する場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA04537-E

Initialization of the distribution manager failed because an error occurred during initialization of a socket for heartbeat reception processing. (error-message = aa....aa)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

## 説明

ハートビート受信用のソケットの初期化中に障害が発生したため、クラスタが構築できませんでした。

## 対処

ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。クラスタ定義の `eads.cluster.heartbeat.address` パラメタに指定したマルチキャストアドレスで、エラーが発生した EADs サーバでマルチキャストの受信が成功することを確認してください。それでも繰り返し発生する場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA04539-E

Heartbeat reception processing failed because of a network failure. (error-message = aa....aa)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

## 説明

ネットワーク障害が発生したため、ハートビートの受信に失敗しました。

## 対処

クラスタ定義の `eads.cluster.heartbeat.address` パラメタに指定したマルチキャストアドレスで、エラーが発生した EADs サーバでマルチキャストの受信が成功することを確認してください。

それでも繰り返し発生する場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA04540-E

```
Initialization of the distribution manager failed because a client connection address was duplicated. (duplicated server 1 = aa....aa, duplicated server 2 = bb....bb)
```

aa....aa : IP アドレスとポート番号の組み合わせが重複した EADs サーバ 1

bb....bb : IP アドレスとポート番号の組み合わせが重複した EADs サーバ 2

## 説明

クラスタ定義ファイルに同一の EADs サーバが複数登録されているため、クラスタが構築できませんでした。

## 対処

クラスタ定義の `eads.node.< EADs サーバ ID >.address` パラメタと `eads.node.< EADs サーバ ID >.port` パラメタに重複がないかどうかを確認してください。

## KDEA04541-E

```
Initialization of the distribution manager failed because the cluster-definition file contains only part of the server position specification.
```

## 説明

クラスタ定義ファイル中に、EADs サーバの位置の指定が一部しか存在しないため、クラスタが構築できませんでした。

## 対処

クラスタ定義の `eads.node.< EADs サーバ ID >.position` パラメタを指定する場合は、クラスタを構成する全 EADs サーバの位置を指定してください。省略する場合は、全 EADs サーバの位置の指定を省略してください。

## KDEA04661-E

```
Initialization of the distribution manager failed. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

分散制御機能の初期化処理に失敗しました。

## 対処

EADs サーバを再起動してください。

## KDEA04663-E

```
Initialization failed because an unexpected exception occurred. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、初期化処理に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04664-E

```
Initialization of the distribution manager failed because the relation between the number of servers and the number of replications was invalid. (number of servers = aa....aa, number of replications = bb....bb)
```

aa....aa : EADs サーバ数

bb....bb : データの多重度

## 説明

クラスタを構成する EADs サーバ数が「データの多重度×2-1」以上でないため、分散制御機能の初期化処理に失敗しました。

## 対処

クラスタ定義を見直して、クラスタを構成する EADs サーバ数が「データの多重度×2-1」以上になるように EADs サーバ数を増やすか、またはデータの多重度 (`eads.replication.factor` パラメタの指定値) を下げてください。

## KDEA04691-I

```
Restoration of cache data will now start. (cache name = aa....aa)
```

aa....aa : キャッシュ名

## 説明

キャッシュのデータの復旧を開始します。

## KDEA04692-I

The cache data were successfully restored. (cache name = aa....aa)

aa....aa：キャッシュ名

### 説明

キャッシュのデータの復旧に成功しました。

## KDEA04693-E

Restoration of cache data failed. (cache name = aa....aa, error code = bb....bb)

aa....aa：キャッシュ名

bb....bb：エラーコード

### 説明

キャッシュのデータの復旧に失敗しました。

### 対処

次の要因が考えられます。

- データの送信元 EADs サーバでエラーが発生した。
- 通信エラーが発生した。
- 予期しないエラーが発生した。

要因を特定し、問題を取り除いたあと、次のどちらかのコマンドを実行して、EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

## KDEA04696-E

Restoration of the server failed. (error code = aa....aa)

aa....aa：エラーコード

### 説明

EADs サーバの復旧に失敗しました。

次の要因が考えられます。

- ほかの運用操作が実行中である。
- 復旧が実行できない状態である。
- `ezstart` コマンドまたは `ezserver` コマンドを、`-r` または `--recovery` オプションを指定しないで実行した。

- キャッシュの生成に失敗した。
- キャッシュデータの送信サーバでエラーが発生した。
- 通信エラーが発生した。
- 予期しないエラーが発生した。

## 対処

次の手順で対処してください。

1. 次のメッセージが出力されている場合は、出力されたメッセージを参照して対処してください。
  - KDEA07101-E
  - KDEA07115-E
  - KDEA07116-E
  - KDEA07121-E
  - KDEA07127-E
  - KDEA07128-E
2. `eztool status` コマンドを実行して、クラスタの状態を確認してください。  
EADs サーバの排他状態が `lock` の場合、排他が解除されていません。この場合は、`eztool unlock` コマンドを実行してください。
3. `ezstart` コマンドまたは `ezserver` コマンドを、`-r` または `--recovery` オプションを指定して再実行してください。

クラスタの状態が、クラスタ動作不能 (`NOT_AVAILABLE`) またはクラスタ一部稼働中 (`PARTIALLY_AVAILABLE`) の場合、EADs サーバは復旧できません。クラスタ内の全 EADs サーバを再起動する必要があります。

## KDEA04697-E

Restoration of the server failed because a required server is not running.

## 説明

EADs サーバの復旧に必要な起動中の EADs サーバが存在しないため、復旧できません。

データを保持している EADs サーバが存在しないか、または多重度以下の数の EADs サーバしか起動していません。

## 対処

次の手順で対処してください。

1. `eztool export` コマンドを実行して、必要なデータをファイルに書き出してください。
2. 起動中のすべての EADs サーバを終了してください。
3. すべての EADs サーバを開始してください。

## KDEA04698-E

An error occurred in the data-sending server during restoration or when referencing a cluster.  
(error code = aa....aa)

aa....aa : エラーコード

### 説明

EADs サーバのクラスタ参加処理中、または復旧処理中に、データの送信元 EADs サーバでエラーが発生しました。

### 対処

次の要因が考えられます。

- ほかの運用操作が実行中であった。
- 合意メッセージの送信キューがオーバーフローした。
- 送信タイムアウトが発生した。
- 通信エラーが発生した。
- 予期しないエラーが発生した。

要因を特定し、問題を取り除いたあと、EADs サーバを再起動してください。

または、次のどちらかのコマンドを実行して、EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

## KDEA04701-E

The master switchover failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、復旧に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04702-E

The master switchover failed because an unexpected interrupt exception occurred.

### 説明

予期しない割り込み例外が発生したため、復旧に失敗しました。

## 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04703-E

The master switchover failed because a cache operation failed.

## 説明

データ操作に失敗したため、復旧に失敗しました。

## 対処

直前に出力されたメッセージを確認してください。

## KDEA04705-E

A restoration failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、復旧に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04707-E

An initialization or online restoration failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、初期化処理または復旧処理に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04722-E

Initialization of the cluster-monitoring function failed because a specified port is already being used. (self address = aa....aa, property name = bb....bb, port number = cc....cc)

aa....aa : 自 EADs サーバの IP アドレス

bb....bb : ポート番号を表すパラメタ名

cc....cc：指定されたポート番号

#### 説明

指定されたポート番号はすでに使用されているため、クラスタが構築できませんでした。

#### 対処

他プロセスが使用していないポート番号を指定してください。

### KDEA04723-E

```
An unknown message was received. (source address = aa....aa, source port = bb....bb, header = cc....cc)
```

aa....aa：メッセージ配信元の IP アドレス

bb....bb：メッセージ配信元のポート番号

cc....cc：受信したメッセージのヘッダ（16 進数）

#### 説明

不正なメッセージを受信しました。使用中の通信ポートに対して、他システムがメッセージを送信しています。

#### 対処

他システムのメッセージ配信と混線しないように、配信先 IP アドレス、または配信先ポート番号を見直してください。

### KDEA04724-E

```
Failure detection processing stopped because an unexpected exception occurred. (error code = aa....aa)
```

aa....aa：エラーコード

#### 説明

予期しない例外によって、生存確認が中断しました。

#### 対処

弊社問い合わせ窓口にご連絡ください。

### KDEA04725-E

```
A message of an unsupported protocol version was received. (source address = aa....aa, source port = bb....bb, protocol version = cc....cc)
```

aa....aa：メッセージ配信元の IP アドレス



bb....bb：メッセージ配信元のポート番号

cc....cc：受信したメッセージのプロトコルバージョン

#### 説明

サポートされていないプロトコルバージョンのメッセージを受信しました。  
異なるバージョンのプロトコルを使用しているシステムと混在しています。

#### 対処

インストールされている EADs のバージョンと同じかどうか確認してください。

### KDEA04726-W

A connection timeout occurred during failure detection processing. (destination address = aa....aa, destination port = bb....bb)

aa....aa：接続先 EADs サーバの IP アドレス

bb....bb：接続先 EADs サーバのポート番号

#### 説明

生存確認で、タイムアウト時間までに接続を確立できませんでした。

#### 対処

頻繁にタイムアウトが発生する場合は、ネットワーク機器に故障がないか確認してください。  
または、サーバ定義の `eads.cluster.heartbeat.interval` パラメタおよび  
`eads.cluster.heartbeat.timeout` パラメタの指定値を見直してください。

### KDEA04727-W

A read timeout occurred during failure detection processing. (destination address = aa....aa, destination port = bb....bb)

aa....aa：接続先 EADs サーバの IP アドレス

bb....bb：接続先 EADs サーバのポート番号

#### 説明

接続確立後の生存確認で、タイムアウト時間までに受信できませんでした。

#### 対処

頻繁にタイムアウトが発生する場合は、ネットワーク機器に故障がないか確認してください。  
または、サーバ定義の `eads.cluster.failureDetector.read.timeout` パラメタの指定値を見直してください。

## KDEA04728-E

An error occurred during establishment of a connection for failure detection processing. (error message = aa....aa)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

### 説明

生存確認のコネクション確立中に、JavaVM から例外がスローされたため、ソケットが作成できませんでした。

### 対処

ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。  
それでも繰り返し発生する場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA04729-E

Cluster monitoring failed because an unexpected exception occurred.

### 説明

予期しない例外が発生したため、クラスタ監視に失敗しました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04730-E

Failure detection processing failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa : エラーコード

### 説明

予期しない例外が発生したため、生存確認に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04752-I

Processing to isolate a server was successful. (server ID = aa....aa)

aa....aa : 縮退した EADs サーバの EADs サーバ ID

### 説明

縮退を完了しました。

## KDEA04753-E

Processing to isolate a server failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa：エラーコード

### 説明

予期しない例外が発生したため、縮退に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04754-E

Processing to isolate a server failed because an unexpected interrupt exception occurred.

### 説明

予期しない割り込み例外が発生したため、縮退に失敗しました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04755-E

Processing to isolate a server failed because a cache operation failed.

### 説明

データ操作に失敗したため、縮退に失敗しました。

### 対処

直前に出力されたメッセージを確認してください。

## KDEA04757-E

Processing to isolate a server failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa：エラーコード

### 説明

予期しない例外が発生したため、縮退に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04781-I

A server will now be added to the cluster. (server ID = aa....aa)

aa....aa : クラスタに追加される EADs サーバの EADs サーバ ID

### 説明

クラスタに EADs サーバを追加します。

## KDEA04782-E

Addition of a server to the cluster failed because an unexpected error occurred. (error code = aa....aa)

aa....aa : エラーコード

### 説明

予期しないエラーが発生したため、クラスタへの EADs サーバの追加に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04783-I

Processing to isolate a server will now start. (server ID = aa....aa)

aa....aa : 縮退する EADs サーバの EADs サーバ ID

### 説明

`eztool isolate` コマンドが実行されました。EADs サーバが縮退します。

## KDEA04784-E

Processing to isolate a server failed because an unexpected error occurred. (error code = aa....aa)

aa....aa : エラーコード

### 説明

予期しないエラーが発生したため、EADs サーバの縮退に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04785-I

A command lock will now be acquired.

## 説明

コマンドの排他を取得します。

### KDEA04786-I

A command lock could not be acquired.

## 説明

すでにほかのコマンドによって排他が取得されていたため、コマンドの排他を取得できませんでした。

## 対処

実行中のコマンドが終了するのを確認してから、コマンドを再実行してください。

### KDEA04787-I

Acquisition of a command lock failed. (error code = aa....aa)

aa....aa：エラーコード

## 説明

コマンドの排他を取得できませんでした。

## 対処

次の要因が考えられます。

- ほかのクラスタが操作中であった。
- EADs サーバが停止した。

要因を特定し、問題を取り除いたあと、コマンドを再実行してください。

### KDEA04788-E

Acquisition of a command lock failed because an unexpected error occurred. (error code = aa....aa)

aa....aa：エラーコード

## 説明

予期しないエラーが発生したため、コマンドの排他を取得できませんでした。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

### KDEA04789-I

A command lock will now be released.

## 説明

コマンドの排他を解除します。

## KDEA04790-I

```
An attempt to release a lock failed. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

コマンドの排他を解除できませんでした。

## 対処

次の要因が考えられます。

- ほかのクラスタが操作中であった。
- EADs サーバが停止した。

要因を特定し、問題を取り除いたあと、`eztool unlock` コマンドを実行して、排他を解除してください。

## KDEA04791-E

```
An attempt to release a lock failed because an unexpected error occurred. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

予期しないエラーが発生したため、コマンドの排他を解除できませんでした。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04792-I

```
Now waiting for executing requests to complete...
```

## 説明

実行中のリクエストが完了するのを待っています。

## KDEA04793-I

```
All requests are complete.
```

## 説明

実行中であった、すべてのリクエストが完了しました。  
待機中に到着したリクエストの実行を開始します。

## KDEA04794-I

The wait for the lock to end will now start.

### 説明

排他の完了待ちを開始しました。

## KDEA04795-I

The wait for the lock to end ended successfully.

### 説明

排他の完了待ちが終了しました。

## KDEA04796-E

The wait for the lock to end failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa：エラーコード

### 説明

排他完了待ち中に予期しない例外が発生しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04797-E

The wait for the lock to end failed because an unexpected interrupt exception occurred.

### 説明

排他完了待ち中に予期しない割り込み例外が発生しました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04798-E

The cluster configuration management failed because an unexpected exception occurred. (error code = aa....aa)

aa....aa：エラーコード

### 説明

予期しない例外が発生したため、クラスタの構成管理に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

### KDEA04799-E

```
Processing to isolate a server will now start. (server ID = aa....aa)
```

aa....aa：縮退する EADs サーバの EADs サーバ ID

## 説明

クラスタ監視によって EADs サーバダウンを検知しました。EADs サーバが縮退します。

## 対処

次のどちらかのコマンドを実行して、縮退した EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

### KDEA04800-E

```
This server will be stopped because a gap in the data could not be filled.
```

## 説明

クラスタで合意処理が継続できなくなったため、この EADs サーバを停止します。

## 対処

次のどちらかのコマンドを実行して、停止した EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

または、クラスタ内の全 EADs サーバを再起動してください。

### KDEA04801-E

```
This server will be stopped because a gap in the data could not be filled. (cache name = aa....aa, range = bb....bb - cc....cc)
```

aa....aa：キャッシュ名

bb....bb：レンジの左端

cc....cc：レンジの右端

## 説明

クラスタで合意処理が継続できなくなったため、この EADs サーバを停止します。



## 対処

次のどちらかのコマンドを実行して、停止した EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

または、クラスタ内の全 EADs サーバを再起動してください。

## KDEA04802-E

```
The cluster configuration management failed because an unexpected exception occurred.  
(error code = aa....aa)
```

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、クラスタの構成管理に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04803-E

```
A lock failed because an unexpected exception occurred. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、排他に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04804-E

```
Unlock processing failed because an unexpected exception occurred. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、排他の解除に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04805-E

Only some of the data are available because a server is isolated. (server ID = aa....aa, aborted positions = bb....bb)

aa....aa：縮退した EADs サーバの EADs サーバ ID

bb....bb：アクセスできないデータ範囲のコンシステント・ハッシング上の開始位置の一覧

### 説明

EADs サーバが縮退したため、一部のデータにアクセスできません。

### 対処

データへのアクセスはできませんが、キャッシュ上にデータは残っています。次の手順で対処してください。

1. `eztool export` コマンドを実行して、必要なデータをファイルに書き出してください。
2. 起動中のすべての EADs サーバを終了してください。
3. すべての EADs サーバを開始してください。

EADs サーバの復旧によって、クラスタを通常状態に戻すことはできません。

## KDEA04806-E

Cluster configuration management failed because an unexpected interrupt exception occurred.

### 説明

予期しない割り込み例外が発生したため、クラスタの構成管理に失敗しました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04807-E

Cluster configuration management failed because the system could not continue building a consensus on distribution among servers in the range.

### 説明

合意処理が続行できなくなったため、クラスタの構成管理に失敗しました。

### 対処

クラスタ内の全 EADs サーバを再起動してください。

## KDEA04808-E

Only some of the data are available because a server is isolated. Some data were lost. (server ID = aa....aa, lost data start positions = bb....bb, aborted data start positions = cc....cc)

aa....aa：縮退した EADs サーバの EADs サーバ ID

bb....bb：欠落したデータ範囲のコンシステント・ハッシング上の開始位置の一覧

cc....cc：アクセスできないデータ範囲のコンシステント・ハッシング上の開始位置の一覧

### 説明

EADs サーバが縮退したため、一部のデータにアクセスできません。  
また、一部のデータが欠落しました。

### 対処

欠落していないデータについて、データへのアクセスはできませんが、キャッシュ上にデータは残っています。次の手順で対処してください。

1. `eztool export` コマンドを実行して、必要なデータをファイルに書き出してください。
2. 起動中のすべての EADs サーバを終了してください。
3. すべての EADs サーバを開始してください。

EADs サーバの復旧によって、クラスタを通常状態に戻すことはできません。

## KDEA04809-W

The number of running servers is the minimum required for the cluster to be available. (isolated server ID = aa....aa, start positions of data to be lost = bb....bb, start positions of data to be aborted = cc....cc)

aa....aa：縮退した EADs サーバの EADs サーバ ID

bb....bb：さらに EADs サーバが縮退した場合、欠落のおそれがあるデータ範囲のコンシステント・ハッシング上の開始位置の一覧

cc....cc：さらに EADs サーバが縮退した場合、アクセスできないおそれがあるデータ範囲のコンシステント・ハッシング上の開始位置の一覧

### 説明

稼働中の EADs サーバ数が、最低限クラスタが動作できる数になりました。

さらに EADs サーバが縮退した場合、一部のデータにアクセスできなくなるか、または一部のデータが欠落するおそれがあります。

### 対処

次のどちらかのコマンドを実行して、縮退している EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

## KDEA04810-W

The number of running servers is the minimum required for the cluster to be available.  
(isolated server ID = aa....aa, start positions = bb....bb)

aa....aa : 縮退した EADs サーバの EADs サーバ ID

bb....bb : さらに EADs サーバが縮退した場合、アクセスできないおそれがあるデータ範囲のコンシステント・ハッシング上の開始位置の一覧

### 説明

稼働中の EADs サーバ数が、最低限クラスタが動作できる数になりました。

さらに EADs サーバが縮退した場合、一部のデータにアクセスできなくなるおそれがあります。

### 対処

次のどちらかのコマンドを実行して、縮退している EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

## KDEA04812-E

An unexpected exception occurred. (error detail = aa....aa)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

### 説明

予期しない例外が発生しました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04813-E

A sequence number is too large and exceeded the upper limit.

### 説明

長期運用によって、内部的なシーケンス番号がオーバーフローしました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA04814-W

```
Server-to-server communication timeout. (timeout value = aa....aa, destination addresses = bb....bb)
```

aa....aa：タイムアウト時間

bb....bb：送信先 EADs サーバの IP アドレス一覧

### 説明

EADs サーバ間の通信でタイムアウトが発生しました。

次の要因が考えられます。

- 通信先 EADs サーバ、ホスト、またはネットワークで障害が発生している。
- タイムアウト時間が適切でない。

### 対処

タイムアウトした原因を取り除いてください。

または、サーバ定義の `eads.cache.consensus.timeout` パラメタの指定値を大きくしてください。ただし、次の場合は指定値を変更する必要はありません。

- 通信障害が発生している場合
- プロセス障害などによって EADs サーバが縮退している場合
- `eztool isolate` コマンドによって EADs サーバが縮退している場合

## KDEA04815-E

```
The cluster status has become NOT_AVAILABLE.
```

### 説明

クラスタの状態が、クラスタ動作不能 (NOT\_AVAILABLE) になりました。

スプリットブレインが発生したか、クラスタ内の半数以上の EADs サーバに障害が発生しました。

### 対処

スプリットブレインが発生している場合、ネットワークを復旧してください。

クラスタ内の半数以上の EADs サーバに障害が発生している場合、クラスタ内の全 EADs サーバを再起動してください。

## KDEA04816-I

```
The cluster has recovered from the NOT_AVAILABLE state.
```

### 説明

クラスタの状態が、クラスタ動作不能 (NOT\_AVAILABLE) から回復しました。

スプリットブレインから回復しました。

## KDEA04817-E

A gap was found that the server could not resolve.

### 説明

遅延していた EADs サーバの処理を同期しようとしたのですが、同期を取るための履歴が残っていないため、回復できませんでした。

### 対処

次の要因が考えられます。

- 合意メッセージの送信キューがオーバーフローした。
- EADs サーバが縮退した。
- 通信エラーが発生した。
- 予期しないエラーが発生した。

要因を特定し、問題を取り除いたあと、次のどちらかのコマンドを実行して、EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

## KDEA04821-W

Processing to delete part of the history in order to resolve a memory shortage in the history storage area has started.

### 説明

更新操作の履歴を格納する領域が不足したため、既存の更新操作の履歴の削除を開始しました。

### 対処

更新操作の履歴が格納される領域のサイズが見積もりどおりに設定されているか確認してください。

十分なサイズが設定されていない場合は、サーバ定義の `eads.consensus.external.heapsize` パラメタの値を変更してください。

## KDEA04822-W

During the building of a consensus on distribution among the servers in the cluster, processing to delete part of the history in order to resolve a memory shortage in the history storage area failed. (requested size = aa.....aa)

aa.....aa：更新操作の履歴を格納するために要求されたサイズ

### 説明

更新操作の履歴を格納する領域が不足し、既存の更新操作の履歴を削除しようとしたのですが、それでも要求されたサイズの領域を確保できなかったため、合意処理を中断しました。

## 対処

ネットワーク機器に故障がないか確認したあと、しばらくしてから再実行してください。

それでも繰り返し発生する場合は、更新操作の履歴が格納される領域のサイズが見積もりどおりに設定されているか確認してください。

十分なサイズが設定されていない場合は、サーバ定義の `eads.consensus.external.heapsize` パラメタの値を変更してください。

## KDEA04823-E

```
During the building of a consensus on distribution among the servers in the cluster, processing to delete part of the history in order to resolve a memory shortage in the history storage area failed. (requested size = aa....aa)
```

aa....aa：更新操作の履歴を格納するために要求されたサイズ

## 説明

更新操作の履歴を格納する領域が不足し、既存の更新操作の履歴を削除しようとしたが、それでも要求されたサイズの領域を確保できなかったため、すべての処理が異常終了しました。

## 対処

更新操作の履歴が格納される領域のサイズが見積もりどおりに設定されているか確認してください。

十分なサイズが設定されていない場合は、サーバ定義の `eads.consensus.external.heapsize` パラメタの値を変更してください。

## KDEA04824-W

```
During the building of a consensus on distribution among the servers in the range, processing to delete part of the history in order to resolve a memory shortage in the history storage area failed. (requested size = aa....aa, cache name = bb....bb, range = cc....cc - dd....dd)
```

aa....aa：更新操作の履歴を格納するために要求されたサイズ

bb....bb：キャッシュ名

cc....cc：レンジの左端

dd....dd：レンジの右端

## 説明

更新操作の履歴を格納する領域が不足し、既存の更新操作の履歴を削除しようとしたが、それでも要求されたサイズの領域を確保できなかったため、合意処理を中断しました。

## 対処

ネットワーク機器に故障がないか確認したあと、しばらくしてから再実行してください。

それでも繰り返し発生する場合は、更新操作の履歴が格納される領域のサイズが見積もりどおりに設定されているか確認してください。

十分なサイズが設定されていない場合は、サーバ定義の `eads.consensus.external.heapsize` パラメタの値を変更してください。

## KDEA04825-E

During the building of a consensus on distribution among the servers in the range, processing to delete part of the history in order to resolve a memory shortage in the history storage area failed. (requested size = aa....aa, cache name = bb....bb, range = cc....cc - dd....dd)

aa....aa：更新操作の履歴を格納するために要求されたサイズ

bb....bb：キャッシュ名

cc....cc：レンジの左端

dd....dd：レンジの右端

### 説明

更新操作の履歴を格納する領域が不足し、既存の更新操作の履歴を削除しようとしたが、それでも要求されたサイズの領域を確保できなかったため、すべての処理が異常終了しました。

### 対処

更新操作の履歴が格納される領域のサイズが見積もりどおりに設定されているか確認してください。

十分なサイズが設定されていない場合は、サーバ定義の `eads.consensus.external.heapsize` パラメタの値を変更してください。

## KDEA04841-E

A TCP connection was closed because the queue for sending distribution-consensus messages overflowed. (destination address = aa....aa, destination port = bb....bb, queuesize = cc....cc, queuedatasize = dd....dd)

aa....aa：接続先 EADs サーバの IP アドレス

bb....bb：接続先 EADs サーバのポート番号

cc....cc：送信キュー内のメッセージ数

dd....dd：送信キュー内のメッセージのデータサイズ

### 説明

合意メッセージの送信キューがあふれたため、接続を切断しました。



## 対処

サーバ定義の `eads.consensus.message.send.queue.size` パラメタ, または `eads.consensus.message.send.max_queuedatasize` パラメタの指定値を大きくしてください。

## KDEA04871-E

Initialization of UDP reception failed because an error occurred during initialization of the receiving socket. (error message = aa....aa, multicast address = bb....bb, multicast port = cc....cc)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

bb....bb : IP アドレス (マルチキャストアドレス)

cc....cc : ポート番号

## 説明

受信ソケットの初期化中に障害が発生したため, UDP 受信の初期化処理に失敗しました。

## 対処

出力された例外のエラーメッセージから要因を特定し, 問題を取り除いてください。また, ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。

## KDEA04872-E

Initialization of UDP transmission failed because an error occurred during initialization of the sending socket. (error message = aa....aa, multicast address = bb....bb, multicast port = cc....cc)

aa....aa : JavaVM からスローされた例外のエラーメッセージ

bb....bb : IP アドレス (マルチキャストアドレス)

cc....cc : ポート番号

## 説明

送信ソケットの初期化中に障害が発生したため, UDP 送信の初期化処理に失敗しました。

## 対処

出力された例外のエラーメッセージから要因を特定し, 問題を取り除いてください。また, ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。

## KDEA04875-E

UDP reception failed because an unexpected exception occurred. (error message = aa....aa)

aa....aa : エラーメッセージ

## 説明

予期しない例外が発生したため、UDP 受信処理に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

### KDEA04876-E

```
UDP transmission failed because an unexpected exception occurred. (error message = aa....aa)
```

aa....aa : エラーメッセージ

## 説明

予期しない例外が発生したため、UDP 送信処理に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

### KDEA04881-E

```
Initialization of TCP/IP reception failed because an error occurred during initialization of the receiving socket. (error message = aa....aa, local address = bb....bb, local port = cc....cc)
```

aa....aa : JavaVM からスローされた例外のエラーメッセージ

bb....bb : 自 IP アドレス

cc....cc : 自ポート番号

## 説明

受信ソケットの初期化中に障害が発生したため、TCP/IP 受信の初期化処理に失敗しました。

## 対処

エラーメッセージごとに次のことを確認してください。

- Cannot assign requested address  
自 IP アドレスが OS で使用できるアドレスかどうか確認してください。
- Address already in use  
自ポート番号がほかのプロセスで使用されていないかどうか確認してください。
- その他

出力された例外のエラーメッセージから要因を特定し、問題を取り除いてください。また、ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。

## KDEA04882-E

```
Initialization of TCP/IP transmission failed because an error occurred during initialization of the sending socket. (error message = aa....aa, destination address = bb....bb, destination port = cc....cc)
```

aa....aa : JavaVM からスローされた例外のエラーメッセージ

bb....bb : 接続先 IP アドレス

cc....cc : 接続先ポート番号

### 説明

送信ソケットの初期化中に障害が発生したため、TCP/IP 送信の初期化処理に失敗しました。

### 対処

エラーメッセージごとに次のことを確認してください。

- Connection refused  
接続先ポート番号が正しいかどうか確認してください。  
正しい場合、接続先 EADs サーバが起動済みかどうか確認してください。
- Connection Timedout  
接続先 IP アドレスが正しいかどうか確認してください。  
正しい場合、接続先の OS が起動済みかどうか確認してください。
- java.net.SocketTimeoutException  
接続先 IP アドレスが正しいかどうか確認してください。  
正しい場合、接続先の OS が起動済みかどうか確認してください。
- その他  
出力された例外のエラーメッセージから要因を特定し、問題を取り除いてください。また、ファイルディスクリプタが見積もりどおりに設定されているかどうかを確認してください。

## KDEA04883-I

```
A different ReceiveBufferSize from the property value will be used. (property value = aa....aa, ReceiveBufferSize = bb....bb)
```

aa....aa : パラメタの指定値

bb....bb : 実際に使用される合意メッセージの受信バッファサイズ

### 説明

サーバ定義の `eads.consensus.message.buffer.size` パラメタに指定した値が、OS で使用できるウィンドウサイズを超えたため、指定値と異なる受信バッファサイズが使用されました。

## KDEA04884-I

A different SendBufferSize from the property value will be used. (property value = aa....aa, SendBufferSize = bb....bb)

aa....aa：パラメタの指定値

bb....bb：実際に使用される合意メッセージの送信バッファサイズ

### 説明

サーバ定義の `eads.consensus.message.buffersize` パラメタに指定した値が、OS で使用できるウィンドウサイズを超えたため、指定値と異なる送信バッファサイズが使用されました。

## KDEA04885-E

TCP/IP reception failed because an unexpected exception occurred. (error message = aa....aa)

aa....aa：エラーメッセージ

### 説明

予期しない例外が発生したため、TCP/IP 受信処理に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04886-E

TCP/IP transmission failed because an unexpected exception occurred. (error message = aa....aa)

aa....aa：エラーメッセージ

### 説明

予期しない例外が発生したため、TCP/IP 送信処理に失敗しました。

### 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04887-W

A read timeout occurred in a TCP/IP reception. (local address = aa....aa, local port = bb....bb, destination address = cc....cc, destination port = dd....dd)

aa....aa：自 IP アドレス

bb....bb：自ポート番号

cc....cc：接続先 IP アドレス

dd....dd：接続先ポート番号

#### 説明

コネクション確立後のデータ受信で、タイムアウト時間までにデータを受信できませんでした。

#### 対処

頻繁にタイムアウトが発生する場合は、サーバ定義の `eads.restore.dataReceiver.read.timeout` パラメタの指定値を見直してください。

### KDEA04888-W

A send timeout occurred in a TCP/IP transmission. (local address = aa....aa, local port = bb....bb, destination address = cc....cc, destination port = dd....dd)

aa....aa：自 IP アドレス

bb....bb：自ポート番号

cc....cc：接続先 IP アドレス

dd....dd：接続先ポート番号

#### 説明

コネクション確立後のデータ送信で、タイムアウト時間までにデータを送信できませんでした。

#### 対処

頻繁にタイムアウトが発生する場合は、サーバ定義の `eads.restore.dataSender.send.timeout` パラメタの指定値を見直してください。

### KDEA04891-E

An unknown message was received. (source address = aa....aa, source port = bb....bb, error message = cc....cc)

aa....aa：メッセージ配信元の IP アドレス

bb....bb：メッセージ配信元のポート番号

cc....cc：エラーメッセージ

#### 説明

不正なメッセージを受信しました。使用中の通信ポートに対して、他システムがメッセージを送信しています。

## 対処

他システムのメッセージ配信と混線しないように、配信先 IP アドレス、または配信先ポート番号を見直してください。

## KDEA04892-W

```
TCP/IP transmission failed because the server is not running.
```

## 説明

EADs サーバが終了しているため、TCP/IP での送信処理に失敗しました。

## KDEA04932-E

```
An error occurred during cache processing. (exception = aa....aa, cache name = bb....bb)
```

aa....aa：例外クラス名

bb....bb：キャッシュ名（キャッシュ名がない場合は null）

## 説明

キャッシュの処理中にエラーが発生しました。

## 対処

直前に出力されたメッセージを確認してください。

## KDEA04962-W

```
Cache creation was aborted because a consensus on distribution for part of the range is not executable. (cache name = aa....aa, management client = bb....bb)
```

aa....aa：キャッシュ名

bb....bb：コマンド実行元の IP アドレス

## 説明

一部のレンジで合意処理が実行できないため、キャッシュの作成を中断しました。

## 対処

クラスタ内の全 EADs サーバを再起動してから、コマンドを再実行してください。

## KDEA04963-E

```
Cache creation failed because an unexpected exception occurred. (error code = aa....aa)
```

aa....aa：エラーコード

## 説明

予期しない例外が発生したため、キャッシュの作成に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04964-E

```
Cache deletion failed because an unexpected exception occurred. (error code = aa....aa)
```

aa....aa : エラーコード

## 説明

予期しない例外が発生したため、キャッシュの削除に失敗しました。

## 対処

メッセージ中に記されたエラーコードを添えて、弊社問い合わせ窓口にご連絡ください。

## KDEA04965-E

```
Addition of a server to the cluster failed because the system could not continue building a consensus on distribution among the servers in the cluster. (node id = aa....aa)
```

aa....aa : 追加しようとした EADs サーバの EADs サーバ ID

## 説明

クラスタで合意処理が継続できなくなったため、クラスタへの EADs サーバの追加に失敗しました。

## 対処

`eztool status` コマンドを実行して、クラスタの状態を確認してください。クラスタの状態がクラスタ稼働中 (AVAILABLE) の場合、EADs サーバを復旧できます。

クラスタの状態が、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の場合、EADs サーバは復旧できません。クラスタ内の全 EADs サーバを再起動する必要があります。

## KDEA04966-E

```
Processing to isolate a server failed because the system could not continue building a consensus on distribution among the servers in the cluster. (node id = aa....aa)
```

aa....aa : 縮退しようとした EADs サーバの EADs サーバ ID

## 説明

クラスタで合意処理が継続できなくなったため、EADs サーバの縮退に失敗しました。

## 対処

`eztool status` コマンドを実行して、クラスタの状態を確認してください。クラスタの状態がクラスタ稼働中 (AVAILABLE) の場合、EADs サーバを復旧できます。

クラスタの状態が、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) の場合、EADs サーバは復旧できません。クラスタ内の全 EADs サーバを再起動する必要があります。

## KDEA04967-E

Acquisition of a command lock failed because the system could not continue building a consensus on distribution among the servers in the cluster.

## 説明

クラスタで合意処理が継続できなくなったため、コマンドの排他的取得に失敗しました。

## 対処

`eztool status` コマンドを実行してクラスタの状態を確認し、別の EADs サーバでコマンドを再実行してください。

## KDEA04968-E

The release of a lock failed because the system could not continue building a consensus on distribution among the servers in the cluster.

## 説明

クラスタで合意処理が継続できなくなったため、コマンドの排他的解除に失敗しました。

## 対処

別の EADs サーバで `eztool unlock` コマンドを再実行してください。

## KDEA04969-E

Initialization of the distribution manager failed because this JavaVM does not support the SHA-1 message digest algorithm.

## 説明

JavaVM が SHA-1 メッセージダイジェストアルゴリズムをサポートしていないため、分散制御機能の初期化処理に失敗しました。

## 対処

EADs サーバを再インストールしてください。



## KDEA04970-E

Initialization of the distribution manager failed because this JavaVM does not support the UTF-8 character set.

### 説明

JavaVM が UTF-8 文字セットをサポートしていないため、分散制御機能の初期化処理に失敗しました。

### 対処

EADs サーバを再インストールしてください。

## 24.6 KDEA06000~KDEA07999

---

KDEA06000~KDEA07999 のメッセージとその対処方法について説明します。

### KDEA06001-E

The specified cache already exists. (cache name = aa....aa)

aa....aa : キャッシュ名

#### 説明

指定されたキャッシュ名はすでに存在します。

#### 対処

すでに存在するキャッシュを削除するか、またはキャッシュ名を変更してください。  
対処が必要なければ、このメッセージは無視してください。

### KDEA06002-E

The specified cache was not found. (cache name = aa....aa)

aa....aa : キャッシュ名

#### 説明

指定されたキャッシュが見つかりません。

#### 対処

キャッシュを新規に作成するか、または処理を見直してください。

### KDEA06004-E

An unexpected exception occurred. (detail = aa....aa)

aa....aa : 保守情報

#### 説明

予期しない例外が発生しました。

#### 対処

弊社問い合わせ窓口にご連絡ください。

### KDEA07001-E

An attempt to open a file failed. (file name = aa....aa, detail = bb....bb)

aa....aa : オープンしようとしたストアデータファイル名 (絶対パス)

bb....bb：詳細情報

## 説明

ストアデータファイルのオープンに失敗しました。  
次の要因が考えられます。

- 読み込み対象のストアデータファイルがない。
- スストアデータファイル以外のパスを指定した。
- スストアデータファイルのアクセス権がない。

## 対処

詳細情報を参照し、OS 環境に問題がないかどうかを確認してください。

## KDEA07002-E

```
An I/O error occurred. (file name = aa....aa, detail = bb....bb)
```

aa....aa：ストアデータファイル名（絶対パス）

bb....bb：詳細情報

## 説明

I/O エラーが発生しました。

## 対処

詳細情報を参照し、OS 環境に問題がないかどうかを確認してください。

## KDEA07003-E

```
The data format is invalid. (file name = aa....aa, detail = bb....bb)
```

aa....aa：ストアデータファイル名（絶対パス）

bb....bb：保守情報

## 説明

データ形式が不正です。  
次の要因が考えられます。

- スストアデータファイルではない。
- スストアデータファイルが破損している。

## 対処

正しいストアデータファイルを指定しているかどうかを見直してください。  
または、別のストアデータファイルを使用してください。

## KDEA07004-E

A checksum error occurred. (file name = aa....aa)

aa....aa：ストアデータファイル名（絶対パス）

### 説明

チェックサムエラーです。  
ストアデータファイルが破損していると考えられます。

### 対処

正しいストアデータファイルを指定しているかどうかを見直してください。  
または、別のストアデータファイルを使用してください。

## KDEA07005-E

The input data version is too new. (file name = aa....aa, input version = bb....bb)

aa....aa：ストアデータファイル名（絶対パス）

bb....bb：ストアデータファイルのフォーマットバージョン

### 説明

新しいフォーマットバージョンで取得したストアデータファイルを読み込もうとしています。

### 対処

正しいストアデータファイルを指定しているかどうかを見直してください。  
または、別のストアデータファイルを使用してください。

## KDEA07006-E

An unexpected exception occurred. (details = aa....aa)

aa....aa：保守情報

### 説明

予期しない例外が発生しました。

### 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA07101-E

The specified cache type cannot be used. (cache name = aa....aa, cache type = bb....bb)

aa....aa：キャッシュ名

bb....bb：キャッシュタイプ

## 説明

指定されたキャッシュタイプは使えません。

次の要因が考えられます。

- キャッシュ定義の `eads.cache.type` パラメタの指定値が誤っている。
- 起動設定の `eads.java.external.heapsize` パラメタの指定値が小さい。

## 対処

キャッシュ定義の `eads.cache.type` パラメタの指定値を見直してください。

キャッシュ定義の `eads.cache.type` パラメタの指定値が正しい場合は、起動設定の `eads.java.external.heapsize` パラメタの指定値を大きくしてください。

## KDEA07104-E

```
The specified cache or cache file was not found. (cache name = aa....aa)
```

aa....aa：キャッシュ名

## 説明

指定されたキャッシュ、またはキャッシュファイルが見つかりません。

## 対処

キャッシュを作成してください。

または、正しいキャッシュファイルを指定しているかどうかを見直してください。

## KDEA07106-E

```
An attempt to open a file failed. (file name = aa....aa, details = bb....bb)
```

aa....aa：オープンしようとしたキャッシュファイル名（絶対パス）

bb....bb：詳細情報

## 説明

キャッシュファイルのオープンに失敗しました。

次の要因が考えられます。

- 読み込み対象のキャッシュファイルがない。
- キャッシュファイル以外のパスを指定した。
- キャッシュファイルのアクセス権がない。

## 対処

詳細情報を参照し、OS環境に問題がないかどうかを確認してください。

## KDEA07107-E

An I/O error occurred. (file name = aa....aa, details = bb....bb)

aa....aa：キャッシュファイル名（絶対パス）（特定できない場合は null）

bb....bb：詳細情報

### 説明

I/O エラーが発生しました。

### 対処

詳細情報を参照し、OS 環境に問題がないかどうかを確認してください。

## KDEA07108-E

The data format is invalid. (file name = aa....aa, details = bb....bb)

aa....aa：キャッシュファイル名（絶対パス）

bb....bb：保守情報

### 説明

データ形式が不正です。

次の要因が考えられます。

- キャッシュファイルではない。
- キャッシュファイルが破損している。

### 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

## KDEA07109-E

A checksum error occurred. (file name = aa....aa)

aa....aa：キャッシュファイル名（絶対パス）

### 説明

チェックサムエラーです。

キャッシュファイルが破損していると考えられます。

### 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

## KDEA07110-E

The input data version is too new. (file name = aa....aa, input version = bb....bb)

aa....aa : キャッシュファイル名 (絶対パス)

bb....bb : キャッシュファイルのフォーマットバージョン

### 説明

新しいフォーマットバージョンで取得したキャッシュファイルを読み込もうとしています。

### 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

## KDEA07111-E

A data discrepancy between the cache property file and the cache file was detected. (cache property file name = aa....aa, parameter = bb....bb, cache property value = cc....cc, cache file value = dd....dd)

aa....aa : キャッシュ定義ファイル名

bb....bb : パラメタ名

cc....cc : キャッシュ定義ファイルの値

dd....dd : キャッシュファイルの値

### 説明

キャッシュ定義ファイルの値とキャッシュファイルの値に不一致があります。

キャッシュファイルを作成したあとで、次に示すキャッシュ定義のパラメタを変更することはできません。

- [eads.cache.type](#)
- [eads.cache.disk.filesize](#)
- [eads.cache.disk.blocksize](#)

また、キャッシュファイルを作成したあとで、次に示すキャッシュ定義のパラメタの指定値を小さい値に変更することはできません。

- [eads.cache.disk.filenum](#)

### 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

キャッシュ定義ファイルの値が誤っている場合は、次に示すキャッシュ定義のパラメタの指定値を見直し、正しい値に変更したあと、EADs サーバを再起動してください。

- eads.cache.type
- eads.cache.disk.filesize
- eads.cache.disk.blocksize
- eads.cache.disk.filenum

## KDEA07112-E

A data discrepancy between the cluster property file and the cache file was detected. (cache name = aa....aa, type = bb....bb, cluster property value = cc....cc, cache file value = dd....dd)

aa....aa：キャッシュ名

bb....bb：種別

cc....cc：クラスタ定義ファイルの値（パラメタが設定されていない場合は null）

dd....dd：キャッシュファイルの値

### 説明

クラスタ定義ファイルの値とキャッシュファイルの値に不一致があります。

キャッシュファイルを作成したあとで、クラスタ定義のパラメタを変更することはできません。

種別ごとの要因を次の表に示します

種別	要因
server number	EADs サーバの数が不一致です。
no server ID	指定した ID の EADs サーバがありません。
position	EADs サーバの位置が不一致です。

### 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

クラスタ定義ファイルの値が誤っている場合は、次に示すクラスタ定義のパラメタを見直し、正しい指定に変更したあと、EADs サーバを再起動してください。

- eads.node.< EADs サーバ ID >.address
- eads.node.< EADs サーバ ID >.port
- eads.node.< EADs サーバ ID >.position



## KDEA07113-E

A data discrepancy between the cache and the cache file was detected. (cache name = aa....aa, error details = bb....bb)

aa....aa：キャッシュ名

bb....bb：エラー詳細情報

### 説明

キャッシュとキャッシュファイルの値に不一致があるため、EADs サーバを復旧できませんでした。

### 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

## KDEA07115-E

A cache file cannot be written to. (cache name = aa....aa, details = bb....bb)

aa....aa：キャッシュ名

bb....bb：詳細情報

### 説明

キャッシュファイルの書き込み中に問題が発生したため、キャッシュファイルの書き込みに失敗しました。

### 対処

詳細情報および直前に出力されたメッセージから要因を特定し、問題を取り除いてください。

## KDEA07116-E

A cache file cannot be read. (cache name = aa....aa, details = bb....bb)

aa....aa：キャッシュ名

bb....bb：詳細情報

### 説明

キャッシュファイルの読み込みに失敗しました。

次の要因が考えられます。

- キャッシュ定義ファイルの読み込み中に問題が発生した。
- キャッシュファイルの読み込み中に問題が発生した。
- キャッシュファイルに過不足があります。

## 対処

詳細情報および直前に出力されたメッセージから要因を特定し、問題を取り除いてください。

なお、`ezstart -r` コマンド、または `ezserver -r` コマンド実行時にキャッシュファイルの過不足が発生した場合、`deleteecf -l` コマンドの実行後に、`ezstart -r` コマンド、または `ezserver -r` コマンドを再実行すると、復旧できることがあります。

## KDEA07117-E

```
A data discrepancy between the cluster property file and the cache file was detected. (cache name = aa....aa, parameter = bb....bb, cluster property value = cc....cc, cache file value = dd....dd)
```

aa....aa：キャッシュ名

bb....bb：パラメタ名

cc....cc：クラスタ定義ファイルの値

dd....dd：キャッシュファイルの値

## 説明

クラスタ定義ファイルの値とキャッシュファイルの値に不一致があります。

キャッシュファイルを作成したあとで、次に示すクラスタ定義のパラメタを変更することはできません。

- `eads.server.max_key_size`
- `eads.replication.factor`

## 対処

正しいキャッシュファイルを指定しているかどうかを見直してください。

または、別のキャッシュファイルを使用してください。

クラスタ定義ファイルの値が誤っている場合は、次に示すクラスタ定義のパラメタの指定値を見直し、正しい値に変更したあと、EADs サーバを再起動してください。

- `eads.server.max_key_size`
- `eads.replication.factor`

## KDEA07120-E

```
An I/O error occurred. (directory = aa....aa, details = bb....bb)
```

aa....aa：ディレクトリ名（絶対パス）

bb....bb：詳細情報

## 説明

I/O エラーが発生しました。

## 対処

詳細情報を参照し、OS 環境に問題がないかどうかを確認してください。  
特に、次に示す問題がないかどうかを確認してください。

- 必要なディレクトリが作成されていない
- ディレクトリを作成できない

## KDEA07121-E

A data discrepancy between the cache property file and the cache was detected. (cache property file name = aa....aa, parameter = bb....bb, cache property value = cc....cc, cache value = dd....dd)

aa....aa：キャッシュ定義ファイル名

bb....bb：パラメタ名

cc....cc：キャッシュ定義ファイルの値

dd....dd：キャッシュの値

## 説明

キャッシュ定義ファイルの指定値とキャッシュの値に不一致があります。  
キャッシュを作成したあとで、次に示すキャッシュ定義のパラメタを変更することはできません。

- `eads.cache.type`
- `eads.cache.disk.filesize`
- `eads.cache.disk.filenum`
- `eads.cache.disk.blocksize`

## 対処

次に示すキャッシュ定義のパラメタの指定値を見直し、正しい値に変更したあと、EADs サーバを再起動してください。

- `eads.cache.type`
- `eads.cache.disk.filesize`
- `eads.cache.disk.filenum`
- `eads.cache.disk.blocksize`

## KDEA07122-E

The area for writing the cache data files is insufficient. (cache name = aa....aa)

aa....aa：キャッシュ名

## 説明

キャッシュデータファイルの書き込み領域が不足しました。

## 対処

キャッシュ定義の `eads.cache.disk.filenum` パラメタの指定値を大きくしたあと、EADs サーバを再起動してください。

## KDEA07123-E

```
An I/O error occurred. (cache name = aa....aa, details = bb....bb)
```

aa....aa：キャッシュ名

bb....bb：詳細情報

## 説明

I/O エラーが発生しました。

## 対処

例外ログから要因を特定し、問題を取り除いてください。

## KDEA07124-E

```
An unexpected exception occurred. (cache name = aa....aa, details = bb....bb)
```

aa....aa：キャッシュ名

bb....bb：保守情報

## 説明

予期しない例外が発生しました。

## 対処

弊社問い合わせ窓口にご連絡ください。

## KDEA07125-I

```
A parameter different from the cache file value will be used. (cache property file name = aa....aa, parameter = bb....bb, cache property value = cc....cc, cache file value = dd....dd)
```

aa....aa：キャッシュ定義ファイル名

bb....bb：パラメタ名

cc....cc：キャッシュ定義ファイルの値

dd....dd：キャッシュファイルの値

## 説明

キャッシュファイルの値とは異なるパラメタが使用されました。

## KDEA07126-W

```
The loading of the cache property file failed. (cache property file name = aa....aa)
```

aa....aa：キャッシュ定義ファイル名

## 説明

キャッシュ定義ファイルの読み込み中に問題が発生したため、キャッシュ定義ファイルの読み込みに失敗しました。

## 対処

直前に出力されたメッセージを確認してください。

## KDEA07127-E

```
The loading of the cache property file failed. (cache property file name = aa....aa)
```

aa....aa：キャッシュ定義ファイル名

## 説明

キャッシュ定義ファイルの読み込み中に問題が発生したため、キャッシュ定義ファイルの読み込みに失敗しました。

## 対処

直前に出力されたメッセージを確認してください。

## KDEA07128-E

```
The compaction of cache data files failed. (cache name = aa....aa, range ID = bb....bb, error details = cc....cc)
```

aa....aa：キャッシュ名

bb....bb：レンジ ID

cc....cc：エラー詳細情報

## 説明

キャッシュデータファイルのコンパクションに失敗しました。

## 対処

キャッシュファイルが破損していると考えられます。

`eztool deleteecf` コマンドでキャッシュファイルを削除したあと、次のどちらかのコマンドを実行して、再度 EADs サーバを復旧してください。

- `ezstart -r` コマンド
- `ezserver -r` コマンド

## 24.7 KDEA08000~KDEA09999

---

KDEA08000~KDEA09999 のメッセージとその対処方法について説明します。

### KDEA08001-I

```
The command will now start. (subcommand = aa....aa, parameter = bb....bb)
```

aa....aa : サブコマンド

bb....bb : オプションおよび引数

#### 説明

コマンドを開始します。

### KDEA08002-I

```
The command will now end.
```

#### 説明

コマンドを終了します。

### KDEA08006-E

```
Log initialization failed. (log directory = aa....aa)
```

aa....aa : ログの出力先パス名

#### 説明

ログの初期化に失敗しました。

#### 対処

コマンド定義の `eads.logger.dir` パラメタに指定したログファイルの出力先パス名が適切かどうかを確認してください。

日立トレース共通ライブラリの初期化に失敗している場合、EADs サーバの負荷が低くなってから、再実行してください。

### KDEA08007-E

```
The specified option is invalid. (error details = aa....aa) For more information, use "eztool -h".
```

aa....aa : エラー詳細情報

#### 説明

指定されたコマンドのオプションが不正です。

コマンド形式の詳細を知りたい場合は、`eztool -h` コマンドを実行してください。

## 対処

コマンドのオプションを確認してください。

## KDEA08008-E

The specified option is duplicated. (option = aa....aa)

aa....aa : コマンドのオプション

## 説明

指定されたコマンドのオプションが重複しています。

## 対処

コマンドのオプションを確認してください。

## KDEA08009-E

The specified subcommand is invalid. (subcommand = aa....aa) For more information, use "eztool -h".

aa....aa : 不正なサブコマンド

## 説明

指定されたサブコマンドが不正です。

コマンド形式の詳細を知りたい場合は、`eztool -h` コマンドを実行してください。

## 対処

サブコマンドを確認してください。

## KDEA08010-E

The specified parameter is invalid. (parameter = aa....aa) For more information, use "eztool -h".

aa....aa : 不正なコマンドの引数

## 説明

指定されたコマンドの引数が不正です。

コマンド形式の詳細を知りたい場合は、`eztool -h` コマンドを実行してください。

## 対処

コマンドの引数を確認してください。

## KDEA08011-E

An attempt to connect to a server failed. (server = aa....aa)



aa....aa：接続先 EADs サーバのホスト名（または IP アドレス）とポート番号

## 説明

EADs サーバへの接続に失敗しました。

## 対処

接続先 EADs サーバのサーバ定義を確認してください。

接続先 EADs サーバが起動しているかどうかを確認してください。

クラスタ内の EADs サーバに障害が発生している場合、EADs サーバダウンが確定するまで待つてから、再実行してください。EADs サーバダウンが確定するまでに掛かる時間については、「9.3.2(1) ハートビートの送信、および生存確認」を参照してください。

## KDEA08013-E

```
The server is not in a state in which commands can be executed. (subcommand = aa....aa, error details = bb....bb)
```

aa....aa：サブコマンド

bb....bb：エラー詳細情報（コマンドが実行できない状態の EADs サーバとその状態の一覧）

## 説明

EADs サーバがコマンドを実行できる状態ではありません。

## 対処

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

コマンドが実行できる状態にしてから、再実行してください。

## KDEA08014-E

```
An attempt to lock the server failed. (server = aa....aa, error details = bb....bb)
```

aa....aa：排他に失敗した EADs サーバのホスト名（または IP アドレス）とポート番号

bb....bb：エラー詳細情報

## 説明

排他に失敗しました。

## 対処

`eztool status -v` コマンドを実行して、ほかのコマンドが実行中かどうかを確認してください。

コマンドが終了しているのに、EADs サーバの排他状態が lock（排他している）であれば、`eztool unlock` コマンドを実行して、コマンドの排他を解除してください。

エラーの要因が不明な場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA08015-E

A server error occurred during the execution of a command. (server = aa....aa, error details = bb....bb)

aa....aa：エラーが発生した EADs サーバのホスト名（または IP アドレス）とポート番号

bb....bb：エラー詳細情報

### 説明

コマンドの実行中に EADs サーバでエラーが発生しました。

### 対処

エラー詳細情報を確認してください。

EADs サーバで発生しているエラーを、EADs サーバのメッセージで確認してください。

エラーの要因が不明な場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA08016-E

An attempt to unlock the server failed. (server = aa....aa, error details = bb....bb)

aa....aa：排他解除に失敗した EADs サーバのホスト名（または IP アドレス）とポート番号

bb....bb：エラー詳細情報

### 説明

排他の解除に失敗しました。

### 対処

`eztool status -v` コマンドを実行して、ほかのコマンドが実行中かどうかを確認してください。

コマンドが終了しているのに、EADs サーバの排他状態が lock（排他している）であれば、`eztool unlock` コマンドを実行して、コマンドの排他を解除してください。

エラーの要因が不明な場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA08018-E

The server connection timed out. (server = aa....aa, timeout value = bb....bb)

aa....aa：タイムアウトした EADs サーバのホスト名（または IP アドレス）とポート番号

bb....bb：タイムアウト時間

### 説明

EADs サーバとの通信がタイムアウトしました。

### 対処

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

クラスタ内の EADs サーバに障害が発生している場合、EADs サーバダウンが確定するまで待ってから、再実行してください。EADs サーバダウンが確定するまでに掛かる時間については、「9.3.2(1) ハートビートの送信、および生存確認」を参照してください。

## KDEA08019-E

```
The command timed out. (timeout value = aa....aa)
```

aa....aa : タイムアウト時間

### 説明

コマンドがタイムアウトしました。

### 対処

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

コマンドの引数で指定したタイムアウト値か、コマンド定義の `eads.management.command.timeout` パラメタまたは `eads.management.long_running.command.timeout` パラメタを確認して、必要に応じて値を変更してから、再実行してください。

## KDEA08020-E

```
An unexpected error occurred during command execution. (error details = aa....aa)
```

aa....aa : エラー詳細情報

### 説明

コマンドの実行中に予期しないエラーが発生しました。

### 対処

エラー詳細情報を確認してください。

エラーの要因が不明な場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA08023-I

```
The status has already changed. (server = aa....aa)
```

aa....aa : EADs サーバのホスト名（または IP アドレス）とポート番号

### 説明

EADs サーバの状態がすでに変更されています。

## KDEA08024-W

```
No matching store data files were found. (server = aa....aa, warning details = bb....bb)
```

aa....aa : EADs サーバのホスト名（または IP アドレス）とポート番号

bb....bb：警告詳細情報

## 説明

該当するストアデータファイルが見つかりませんでした。

## 対処

指定したストアデータファイルキーを確認してください。

ストアデータファイルの出力先にストアデータファイルが存在するか確認してください。

ディレクトリ構成の変更などで該当するストアデータファイルを移動した場合や、EADs サーバをクラスタに追加した場合に、この警告が表示されます。警告詳細情報を確認し、問題がなければ無視してください。

## KDEA08025-E

The specified subcommand requires a value. (subcommand = aa....aa) For more information, use "eztool -h".

aa....aa：引数が指定されていないサブコマンド

## 説明

指定されたサブコマンドには引数が必要です。

コマンド形式の詳細を知りたい場合は、`eztool -h` コマンドを実行してください。

## 対処

サブコマンドの引数を確認してください。

## KDEA08026-W

The specified cache already exists. (cache name = aa....aa)

aa....aa：キャッシュ名

## 説明

指定されたキャッシュ名はすでに存在します。

## 対処

指定したキャッシュ名が適切かどうかを確認してください。

## KDEA08027-W

A connection cannot be established because the cluster is offline. (server = aa....aa)

aa....aa：EADs サーバのホスト名（または IP アドレス）とポート番号

## 説明

EADs サーバがクラスタに参加していないため、接続できません。

## 対処

`eztool status` コマンドを実行して、クラスタの状態を確認してください。

## KDEA08029-E

The number of store data file generations has exceeded the limit. (generation count = aa....aa, limit = bb....bb)

aa....aa : 世代数

bb....bb : 世代数の上限値

## 説明

ストアデータファイルの世代数が上限値を超えています。

## 対処

次のどれかの対処をしてください。

- `eztool deleteesd` コマンドを実行して、クラスタ内の不要なストアデータファイルを削除してください。
- `eztool export -d` コマンドを実行して、別のディレクトリにデータを書き出してください。
- スストアデータファイルを任意のディレクトリに退避してください。

## KDEA08030-E

No generation-managed store data files were found.

## 説明

世代管理されたストアデータファイルが見つかりませんでした。

## 対処

`eztool listesd` コマンドを実行して、ストアデータファイルを確認してください。

## KDEA08031-W

The specified cache name does not exist. (cache name = aa....aa)

aa....aa : キャッシュ名

## 説明

指定されたキャッシュが見つかりませんでした。

## 対処

指定したキャッシュ名が適切かどうかを確認してください。

## KDEA08032-E

Cache creation failed. (cache name = aa....aa, error details = bb....bb)

aa....aa : キャッシュ名

bb....bb : エラー詳細情報

### 説明

キャッシュの作成に失敗しました。

### 対処

キャッシュ数が上限（64 個）に達していないか確認してください。

指定したキャッシュ名が適切かどうかを確認してください。

エラー詳細情報を確認してください。

## KDEA08033-E

Importing failed. (store date file key = aa....aa, error details = bb....bb)

aa....aa : ストアデータファイルキー

bb....bb : エラー詳細情報

### 説明

データの読み込みに失敗しました。

### 対処

指定したストアデータファイルキーを確認してください。

エラー詳細情報を確認してください。

直前に出力された EADs サーバのメッセージを確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08034-E

Data addition failed. (cache name = aa....aa, key = bb....bb, error details = cc....cc)

aa....aa : キャッシュ名

bb....bb : key

cc....cc : エラー詳細情報

### 説明

データの追加に失敗しました。

## 対処

キャッシュ名を確認してください。

エラー詳細情報を確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08035-E

```
Data acquisition failed. (cache name = aa....aa, key = bb....bb, error details = cc....cc)
```

aa....aa : キャッシュ名

bb....bb : key

cc....cc : エラー詳細情報

## 説明

データの取得に失敗しました。

## 対処

キャッシュ名を確認してください。

エラー詳細情報を確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08036-E

```
Data deletion failed. (cache name = aa....aa, key = bb....bb, error details = cc....cc)
```

aa....aa : キャッシュ名

bb....bb : key

cc....cc : エラー詳細情報

## 説明

データの削除に失敗しました。

## 対処

キャッシュ名を確認してください。

エラー詳細情報を確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08037-E

```
The cluster property file is incorrect. (error details = aa....aa)
```

aa....aa : エラー詳細情報

## 説明

クラスタの構成に関する定義が正しくありません。

## 対処

クラスタ定義ファイルが存在するか確認してください (Developer にはクラスタ定義ファイルはありません)。

エラー詳細情報を確認してください。

クラスタ定義の次のパラメタが適切か確認してください。

- `eads.node.< EADs サーバ ID >.address`
- `eads.node.< EADs サーバ ID >.port`
- `eads.node.< EADs サーバ ID >.position`

## KDEA08046-W

The specified function name does not exist. (function name = aa....aa)

aa....aa : ユーザファンクション名

## 説明

指定されたユーザファンクション名が見つかりませんでした。

## 対処

ユーザファンクション名が適切か確認してください。

## KDEA08047-W

The specified event name does not exist. (event name = aa....aa)

aa....aa : ユーザイベントハンドラ名

## 説明

指定されたユーザイベントハンドラ名が見つかりませんでした。

## 対処

ユーザイベントハンドラ名が適切か確認してください。

## KDEA08049-E

No matching store data file was found. (store data file key = aa....aa)

aa....aa : ストアデータファイルキー

## 説明

該当するストアデータファイルが一つも見つかりませんでした。



## 対処

指定したストアデータファイルキーを確認してください。

ストアデータファイルの出力先にストアデータファイルが存在するか確認してください。

ディレクトリ構成の変更などで該当するストアデータファイルを移動した場合に、このメッセージが表示されます。エラー詳細情報を確認し、問題がなければ無視してください。

## KDEA08050-E

```
The specified store data file already exists. (store data file key = aa....aa)
```

aa....aa：ストアデータファイルキー

## 説明

指定されたストアデータファイル名はすでに存在します。

## 対処

指定したストアデータファイルキーが適切かどうか確認してください。

## KDEA08051-E

```
Processing to open the cluster failed. (error details = aa....aa)
```

aa....aa：エラー詳細情報

## 説明

クラスタの閉塞状態の解除に失敗しました。

## 対処

エラー詳細情報を確認してください。

## KDEA08052-E

```
Processing to isolate the server failed. (error details = aa....aa)
```

aa....aa：エラー詳細情報

## 説明

EADs サーバの縮退に失敗しました。

## 対処

エラー詳細情報を確認してください。

`eztool status` コマンドを実行して、コマンドを実行した EADs サーバの状態を確認してください。

- コマンドを実行した EADs サーバが縮退状態ではない場合  
クラスタの状態がクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) にならないことを確認し、コマンドを再実行してください。

- コマンドを実行した EADs サーバが縮退状態で、`eztool isolate --stop` コマンドを実行している場合  
コマンドを再実行して、EADs サーバプロセスをダウンさせてください。

## KDEA08053-E

The wait for completion of execution failed. (error details = aa....aa)

aa....aa : エラー詳細情報

### 説明

コマンド実行の完了待ちに失敗しました。

### 対処

- `eztool createcache` コマンド、または `eztool deletocache` コマンドの場合  
`eztool listcache` コマンドを実行して、キャッシュ数を確認してください。  
コマンドを再実行してください。
- `eztool stop` コマンド、または `eztool isolate --stop` コマンドの場合  
`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。  
また、プロセスの状態を確認してください。  
コマンドを再実行してください。

## KDEA08054-I

The store data file was imported. (store data file key = aa....aa)

aa....aa : ストアデータファイルキー

### 説明

ストアデータファイルを読み込みました。

## KDEA08055-I

The store data file was exported. (store data file key = aa....aa)

aa....aa : ストアデータファイルキー

### 説明

ストアデータファイルを出力しました。

## KDEA08056-E

A property file cannot be read. (file path = aa....aa)

aa....aa : 定義ファイルの格納先パス名

## 説明

定義ファイルが読み込めません。

次の要因が考えられます。

- 定義ファイルが開けない。
- 格納先パス名がファイルではなくディレクトリを指している。

## 対処

定義ファイルの格納先パス名を見直してください。

要因を特定し、問題を取り除いてください。

## KDEA08057-W

```
The obtained cluster information might be old.
```

## 説明

取得したクラスタ構成情報が古いおそれがあります。

縮退した EADs サーバだけが動作していると考えられます。

## 対処

`eztool status` コマンドを実行して、クラスタおよび EADs サーバの状態を確認してください。

## KDEA08058-E

```
Execution of a function failed. (cache name = aa....aa, function name = bb....bb, error details = cc....cc)
```

aa....aa：キャッシュ名

bb....bb：ユーザファンクション名

cc....cc：エラー詳細情報

## 説明

ユーザファンクションの実行に失敗しました。

## 対処

キャッシュ名、ユーザファンクション名、およびエラー詳細情報を確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08059-I

```
Stopping the compaction of cache data files is already reserved.
```

## 説明

キャッシュデータファイルのコンパクションの中断はすでに予約されています。

## KDEA08060-E

The resumption of cache files failed. (error details = aa....aa)

aa....aa：エラー詳細情報

### 説明

キャッシュの再開に失敗しました。

### 対処

エラー詳細情報を確認してください。

## KDEA08061-E

The compaction of cache data files failed. (error details = aa....aa)

aa....aa：エラー詳細情報

### 説明

キャッシュデータファイルのコンパクションに失敗しました。

### 対処

エラー詳細情報を確認してください。

## KDEA08062-I

The compaction of cache data files was stopped.

### 説明

キャッシュデータファイルのコンパクションが中断されました。

## KDEA08063-I

The compaction of cache data files is not executing.

### 説明

キャッシュデータファイルのコンパクションが実行されていません。

## KDEA08064-I

The cache files were deleted.

### 説明

キャッシュファイルを削除しました。

## KDEA08065-E

Deletion of cache files failed. (error details = aa....aa)

aa....aa：エラー詳細情報

### 説明

キャッシュファイルの削除に失敗しました。

### 対処

エラー詳細情報を確認してください。

## KDEA08066-W

No matching cache file was found. (server = aa....aa, warning details = bb....bb)

aa....aa：EADs サーバのホスト名（または IP アドレス）とポート番号

bb....bb：警告詳細情報

### 説明

該当するキャッシュファイルが見つかりませんでした。

### 対処

キャッシュ定義ファイルで指定したキャッシュファイルの格納先を確認してください。

ディレクトリ構成の変更などで該当するキャッシュファイルを移動したり、キャッシュファイルを削除したりした場合、またはキャッシュ定義ファイルの内容を書き換えた場合に、この警告が表示されません。警告詳細情報を確認してください。

## KDEA08067-E

No matching cache file was found.

### 説明

該当するキャッシュファイルが見つかりませんでした。

### 対処

`eztool listecf` コマンドでキャッシュファイルの有無を確認してください。

キャッシュ定義ファイルの内容を確認してください。

キャッシュ定義ファイルで指定したキャッシュファイルの格納先を確認してください。

## KDEA08068-E

The importing of cache files failed. (error details = aa....aa)

aa....aa：エラー詳細情報

## 説明

キャッシュファイルのインポートに失敗しました。

## 対処

エラー詳細情報を確認してください。

## KDEA08070-E

```
Creation of cache files failed. (cache name = aa....aa, error details = bb....bb)
```

aa....aa：キャッシュ名

bb....bb：エラー詳細情報

## 説明

キャッシュファイルの作成に失敗しました。

## 対処

エラー詳細情報を確認してください。

## KDEA08071-W

```
Cache files do not exist because the type of the specified cache is MemoryCache.
```

## 説明

指定したキャッシュタイプがメモリキャッシュであるため、キャッシュファイルは存在しません。

## KDEA08072-W

```
The cache already exists. (cache name = aa....aa)
```

aa....aa：キャッシュ名

## 説明

キャッシュは、すでに存在します。

## 対処

キャッシュを確認してください。

必要に応じて、`eztool deletecache` コマンドでキャッシュを削除してから、再度実行してください。

## KDEA08073-I

```
The resumption of cache started. (cache name = aa....aa)
```

aa....aa：キャッシュ名

## 説明

キャッシュの再開を開始しました。

## KDEA08074-I

```
The resumption of cache finished. (cache name = aa....aa)
```

aa....aa：キャッシュ名

## 説明

キャッシュの再開が完了しました。

## KDEA08075-W

```
The resumption of cache files did not run, because the cache to resume target does not exist.
```

## 説明

再開する対象のキャッシュが存在しないため、キャッシュファイルの再開は実行されませんでした。

## 対処

キャッシュを確認してください。

必要に応じて、`eztool deletecache` コマンドでキャッシュを削除してから、再度実行してください。

## KDEA08076-E

```
Some data might be inconsistent, because resumption of cache failed. Therefore a lock was not released.
```

## 説明

キャッシュの再開に失敗したため、一部のデータは矛盾しているおそれがあります。そのため、排他は解除されませんでした。

## 対処

データが不整合になっているおそれがあります。そのため、そのまま運用を継続するとデータが破壊されたり、データが失われたりするおそれがあります。次の手順で対処してください。

1. `eztool status -v` コマンドでクラスタの状態を確認する。
2. `eztool unlock` コマンドで排他を解除する。
3. `eztool listcache` コマンドでキャッシュの一覧を確認する。
4. 必要に応じて `eztool deletecache --with_deleteecf` コマンドでキャッシュを削除する。
5. `eztool resume` コマンドでキャッシュを再開する。

## KDEA08077-W

The compaction of cache data files was not executed. (warning details = aa....aa)

aa....aa：警告詳細情報

### 説明

キャッシュデータファイルのコンパクションは実行されませんでした。

### 対処

警告詳細情報を確認してください。

必要に応じて、しきい値を下げたり、未使用ファイル数の指定を増やしたりして、再度実行してください。

## KDEA08078-W

The compaction of cache data files could not reach to the specified unused file count.  
(minimum unused file count = aa....aa)

aa....aa：未使用ファイル数の最小値

### 説明

キャッシュデータファイルのコンパクションは、指定された未使用ファイル数まで到達できませんでした。

### 対処

必要に応じて、しきい値を下げて、再度実行してください。

## KDEA08079-I

Exporting was not executed because no memory cache exist on the server.

### 説明

EADs サーバにメモリキャッシュが存在しなかったため、データの書き出しは実行されませんでした。

## KDEA08401-E

Log initialization failed. (log directory = aa....aa)

aa....aa：ログの出力先パス名

### 説明

ログの初期化に失敗しました。

### 対処

コマンド定義の `eads.logger.dir` パラメタに指定したログファイルの出力先パス名が適切かどうかを確認してください。



サーバ定義の `eads.logger.dir` パラメタに指定したディレクトリ下の `spool` ディレクトリが適切かどうかを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08402-E

```
The server directory is invalid. (directory = aa....aa)
```

aa....aa : 運用ディレクトリ

### 説明

運用ディレクトリが不正です。

### 対処

運用ディレクトリが適切かどうかを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08403-E

```
Startup of the server failed.
```

### 説明

EADs サーバの起動に失敗しました。

### 対処

`ezserver` ファイルが実行可能なファイルかどうかを確認してください。

また、ログの出力先およびログファイルにアクセスできるかどうかを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08404-E

```
The directory name contains invalid characters. (directory = aa....aa)
```

aa....aa : ディレクトリ名

### 説明

ディレクトリ名が不正です。ディレクトリ名に不正な文字が含まれています。

### 対処

ディレクトリ名には、半角英数字 (0~9, A~Z, a~z)、アンダースコア (`_`)、およびスラッシュ (`/`) を使用してください。

また、ディレクトリ名にピリオドを含んだ相対パスを指定しないでください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08405-E

The server process was not found. (server = aa....aa)

aa....aa : EADs サーバ名 (運用ディレクトリ名)

### 説明

EADs サーバプロセスが見つかりませんでした。

### 対処

EADs サーバが起動しているかどうかを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08406-E

A required program was not found. (directory = aa....aa)

aa....aa : EADs サーバ名 (運用ディレクトリ名)

### 説明

必要となるプログラムが見つかりませんでした。

### 対処

製品が適切にインストールされているかどうかを確認してください。

運用ディレクトリの位置が適切かどうかを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08407-E

The specified option is invalid. (error details = Unrecognized option: aa....aa)

aa....aa : オプション

### 説明

指定されたオプションが不正です。

### 対処

オプションを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08408-E

The server process already exists. (server = aa....aa)

aa....aa : EADs サーバ名 (運用ディレクトリ名)

## 説明

EADs サーバはすでに起動しています。

## 対処

EADs サーバが起動しているかどうか確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08409-E

```
An attempt to lock the server failed.
```

## 説明

排他に失敗しました。

## 対処

`eztool status -v` コマンドを実行して、ほかのコマンドが実行中かどうかを確認してください。

コマンドが終了しているのに、EADs サーバの排他状態が `lock` (排他している) であれば、`eztool unlock` コマンドを実行して、コマンドの排他を解除してください。

エラーの要因が不明な場合は、弊社問い合わせ窓口にご連絡ください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08410-E

```
The specified options cannot be specified at the same time. (option names = aa....aa, bb....bb)
```

aa....aa : オプション

bb....bb : オプション

## 説明

指定されたオプションは同時に指定できません。

## 対処

オプションを確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08411-E

```
File creation failed. (file name = aa....aa, error details = bb....bb)
```

aa....aa : ファイル名

bb....bb : エラー詳細情報

## 説明

ファイルの作成に失敗しました。

## 対処

エラー詳細情報を確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08412-W

```
An attempt to unlock the server failed.
```

## 説明

排他の解除に失敗しました。

## 対処

`eztool status -v` コマンドを実行して、ほかのコマンドが実行中かどうかを確認してください。

コマンドが終了しているのに、EADs サーバの排他状態が `lock` (排他している) であれば、`eztool unlock` コマンドを実行して、コマンドの排他を解除してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08413-W

```
The directory of cache files is invalid. (cache properties name = aa....aa, warning details = bb....bb)
```

aa....aa：キャッシュ定義ファイル名

bb....bb：警告詳細情報

## 説明

キャッシュ定義ファイルで指定したディレクトリが不正です。

## 対処

警告詳細情報を確認してください。

必要に応じて、キャッシュ定義ファイルに指定したディレクトリが適切か確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08414-W

```
The directory of cache files is invalid. (cache properties name = aa....aa, warning details = bb....bb)
```

```
The parameter is invalid. The default value will be used. (parameter = aa....aa, value = bb....bb, default value = cc....cc)
```

aa....aa：パラメタ名

bb....bb：指定値

cc....cc：デフォルト値

## 説明

パラメタに指定した値が不正です。デフォルト値を使用します。

## 対処

パラメタに不正な値を指定してします。

必要に応じて、パラメタに指定した値が正しいか確認してください。

なお、このメッセージはスクリプトから出力されるため、ログファイルには出力されません。

## KDEA08501-I

```
The collection of statistics started.
```

## 説明

統計情報の収集を開始しました。

## KDEA08502-I

```
The collection of statistics stopped.
```

## 説明

統計情報の収集を終了しました。

## KDEA08505-I

```
Export processing started. (store data file path = aa....aa, management client = bb....bb)
```

aa....aa：ストアデータファイルの出力先パス名

bb....bb：コマンド実行元の IP アドレス

## 説明

データの書き出しを開始しました。

## KDEA08506-I

```
Import processing started. (store data directory = aa....aa, store data file key = bb....bb,  
management client = cc....cc)
```

aa....aa：ストアデータファイルの格納先パス名

bb....bb：ストアデータファイルキー

cc....cc：コマンド実行元の IP アドレス

## 説明

データの読み込みを開始しました。

### KDEA08507-I

A cache was created. (cache name = aa....aa, management client = bb....bb)

aa....aa：作成したキャッシュ名

bb....bb：コマンド実行元の IP アドレス

## 説明

キャッシュを作成しました。

### KDEA08508-I

A cache was deleted. (cache name = aa....aa, management client = bb....bb)

aa....aa：削除したキャッシュ名

bb....bb：コマンド実行元の IP アドレス

## 説明

キャッシュを削除しました。

### KDEA08510-I

Open processing finished. (management client = aa....aa)

aa....aa：コマンド実行元の IP アドレス

## 説明

EADs サーバの閉塞状態を解除しました。

### KDEA08512-E

Initialization of the management service failed. (error details = aa....aa)

aa....aa：エラー詳細情報

## 説明

運用管理機能の初期化に失敗しました。

## 対処

エラー詳細情報を確認して、EADs サーバを再起動してください。

また、サーバ定義の `eads.management.port` パラメタを確認してください。

統計情報の出力先のディレクトリやファイルを確認してください。

## KDEA08513-E

```
Output of statistics failed. (statistics file path = aa....aa, error details = bb....bb)
```

aa....aa：統計情報ファイルのパス名

bb....bb：エラー詳細情報

### 説明

統計情報ファイルの出力に失敗しました。

### 対処

統計情報ファイルの出力先を確認してください。

## KDEA08516-E

```
Exporting failed. (store data file path = aa....aa, management client = bb....bb, error details = cc....cc)
```

aa....aa：ストアデータファイルの出力先パス名

bb....bb：コマンド実行元の IP アドレス

cc....cc：エラー詳細情報

### 説明

データの書き出しに失敗しました。

### 対処

ストアデータファイルキーを確認してください。

エラー詳細情報を確認してください。

直前に出力されたメッセージを確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08517-E

```
Importing failed. (store data directory = aa....aa, store date file key = bb....bb, management client = cc....cc, error details = dd....dd)
```

aa....aa：ストアデータファイルの格納先パス名

bb....bb：ストアデータファイルキー

cc....cc：コマンド実行元の IP アドレス

dd....dd：エラー詳細情報

## 説明

データの読み込みに失敗しました。

## 対処

ストアデータファイルキーを確認してください。

エラー詳細情報を確認してください。

直前に出力されたメッセージを確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08518-E

```
Creation of a cache failed. (cache name = aa....aa, management client = bb....bb, error details = cc....cc)
```

aa....aa：作成しようとしたキャッシュ名

bb....bb：コマンド実行元の IP アドレス

cc....cc：エラー詳細情報

## 説明

キャッシュの作成に失敗しました。

## 対処

キャッシュ名を確認してください。

エラー詳細情報を確認してください。

直前に出力されたメッセージを確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08519-E

```
Deletion of a cache failed. (cache name = aa....aa, management client = bb....bb, error details = cc....cc)
```

aa....aa：削除しようとしたキャッシュ名

bb....bb：コマンド実行元の IP アドレス

cc....cc：エラー詳細情報

## 説明

キャッシュの削除に失敗しました。



## 対処

キャッシュ名を確認してください。

エラー詳細情報を確認してください。

直前に出力されたメッセージを確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

## KDEA08530-E

```
The server is not in a state in which commands can be executed. (subcommand = aa....aa,
management client = bb....bb, error details = cc....cc)
```

aa....aa：サブコマンド

bb....bb：コマンド実行元の IP アドレス

cc....cc：エラー詳細情報

## 説明

EADs サーバがコマンドを実行できる状態ではありません。

## 対処

EADs サーバ、コマンド、およびクラスタの状態を確認してください。

## KDEA08531-I

```
The status has already been changed. (subcommand = aa....aa, management client = bb....bb,
details = cc....cc)
```

aa....aa：サブコマンド

bb....bb：コマンド実行元の IP アドレス

cc....cc：詳細情報

## 説明

EADs サーバの状態がすでに変更されています。

## KDEA08532-I

```
Close processing finished. (management client = aa....aa)
```

aa....aa：コマンド実行元の IP アドレス

## 説明

EADs サーバを閉塞しました。

## KDEA08533-I

Export processing finished. (store data file path = aa....aa, management client = bb....bb)

aa....aa：ストアデータファイルの出力先パス名

bb....bb：コマンド実行元の IP アドレス

### 説明

データの書き出しを終了しました。

## KDEA08534-I

Import processing finished. (store data directory = aa....aa, store data file key = bb....bb, management client = cc....cc)

aa....aa：ストアデータファイルの格納先パス名

bb....bb：ストアデータファイルキー

cc....cc：コマンド実行元の IP アドレス

### 説明

データの読み込みを終了しました。

## KDEA08535-E

An unexpected error occurred. (error details = aa....aa)

aa....aa：エラー詳細情報

### 説明

予期しないエラーが発生しました。

### 対処

エラー詳細情報を確認してください。

エラーの要因が不明な場合は、弊社問い合わせ窓口にご連絡ください。

## KDEA08537-I

The importing of the store data file started. (store data file path = aa....aa)

aa....aa：ストアデータファイルの格納先パス名

### 説明

データの読み込みを開始しました。

## KDEA08538-I

The importing of the store data file finished. (store data file path = aa....aa)

aa....aa：ストアデータファイルの格納先パス名

### 説明

データの読み込みを終了しました。

## KDEA08539-E

The importing of the store data file failed. (store data file path = aa....aa, error details = bb....bb)

aa....aa：ストアデータファイルの格納先パス名

bb....bb：エラー詳細情報

### 説明

データの読み込みに失敗しました。

### 対処

エラー詳細情報を確認してください。

ストアデータファイルが適切かどうかを確認してください。

## KDEA08545-W

The specified cache name already exists. (cache name = aa....aa, management client = bb....bb)

aa....aa：キャッシュ名

bb....bb：コマンド実行元の IP アドレス

### 説明

指定されたキャッシュ名はすでに存在します。

### 対処

キャッシュ名が適切かどうかを確認してください。

## KDEA08547-W

The specified cache name does not exist. (cache name = aa....aa, management client = bb....bb)

aa....aa：キャッシュ名

bb....bb：コマンド実行元の IP アドレス

#### 説明

指定されたキャッシュが存在しません。

#### 対処

キャッシュ名が適切かどうかを確認してください。

### KDEA08548-I

```
The store data file was deleted. (store data file path = aa....aa)
```

aa....aa：ストアデータファイルの出力先パス名

#### 説明

ストアデータファイルを削除しました。

### KDEA08549-E

```
Deletion of the store data file failed. (store data file name = aa....aa, error details = bb....bb)
```

aa....aa：ストアデータファイル名

bb....bb：エラー詳細情報

#### 説明

ストアデータファイルの削除に失敗しました。

#### 対処

直前に出力されたメッセージを確認してください。

`eztool status` コマンドを実行して、EADs サーバの状態を確認してください。

### KDEA08569-E

```
The specified cache name is invalid. (cache name = aa....aa, management client = bb....bb)
```

aa....aa：キャッシュ名

bb....bb：コマンド実行元の IP アドレス

#### 説明

指定されたキャッシュ名が不正です。

#### 対処

指定したキャッシュ名が適切かどうかを確認してください。

## KDEA08570-E

The number of caches exceeds the limit. (cache count = aa....aa, limit = bb....bb, management client = cc....cc)

aa....aa：キャッシュ数

bb....bb：キャッシュ数の上限

cc....cc：コマンド実行元の IP アドレス

### 説明

キャッシュ数が上限値を超えています。

### 対処

`eztool deletecache` コマンドを実行して、不要なキャッシュを削除してください。

## KDEA08571-E

The specified directory name is invalid. (store data directory name = aa....aa, management client = bb....bb)

aa....aa：ストアデータファイルの格納先パス名

bb....bb：コマンド実行元の IP アドレス

### 説明

指定されたディレクトリ名が不正です。

### 対処

指定したディレクトリ名が適切かどうかを確認してください。

## KDEA08572-E

The specified store data file name is invalid. (store data file key = aa....aa, management client = bb....bb)

aa....aa：ストアデータファイルキー

bb....bb：コマンド実行元の IP アドレス

### 説明

指定されたストアデータファイル名が不正です。

### 対処

指定したストアデータファイルキーが適切かどうかを確認してください。

## KDEA08575-E

Acquisition of the store data failed. (management client = aa....aa, error details = bb....bb)

aa....aa : コマンド実行元の IP アドレス

bb....bb : エラー詳細情報

### 説明

ストアデータの取得に失敗しました。

### 対処

ストアデータファイルの出力先にストアデータファイルが存在するか確認してください。

ストアデータファイルの出力先, およびストアデータファイルのアクセス権を確認してください。

## KDEA08576-I

Processing to isolate the server finished. (management client = aa....aa)

aa....aa : コマンド実行元の IP アドレス

### 説明

EADs サーバを縮退しました。

## KDEA08577-E

Processing to isolate the server failed. (management client = aa....aa, error details = bb....bb)

aa....aa : コマンド実行元の IP アドレス

bb....bb : エラー詳細情報

### 説明

EADs サーバの縮退に失敗しました。

### 対処

エラー詳細情報を確認してください。

ほかの EADs サーバの状態を確認してください。

縮退しているほかの EADs サーバを復旧してから, コマンドを再実行してください。

## KDEA08580-I

Processing to close the server started. (management client = aa....aa)

aa....aa : コマンド実行元の IP アドレス

## 説明

EADs サーバの閉塞化を開始しました。

## KDEA08582-E

```
The wait for completion of execution failed. (management client = aa....aa, error details = bb....bb)
```

aa....aa : コマンド実行元の IP アドレス

bb....bb : エラー詳細情報

## 説明

コマンド実行の完了待ちに失敗しました。

## 対処

コマンドを再実行してください。

## KDEA08585-E

```
The compaction of cache data files failed. (management client = aa....aa, cache name = bb....bb, range ID = cc....cc, error details = dd....dd)
```

aa....aa : コマンド実行元の IP アドレス

bb....bb : キャッシュ名

cc....cc : レンジ ID

dd....dd : エラー詳細情報

## 説明

キャッシュデータファイルのコンパクションに失敗しました。

## 対処

エラー詳細情報を確認してください。

## KDEA08586-I

```
Stopping the compaction of cache data files was reserved. (management client = aa....aa)
```

aa....aa : コマンド実行元の IP アドレス

## 説明

キャッシュデータファイルのコンパクションの中断が予約されました。

## KDEA08587-I

The compaction of cache data files was stopped. (management client = aa....aa)

aa....aa：コマンド実行元の IP アドレス

### 説明

キャッシュデータファイルのコンパクションが中断されました。

## KDEA08588-I

The cache files were deleted. (management client = aa....aa, cache name = bb....bb)

aa....aa：コマンド実行元の IP アドレス

bb....bb：キャッシュ名

### 説明

キャッシュファイルを削除しました。

## KDEA08589-E

Deletion of cache files failed. (management client = aa....aa, cache name = bb....bb, error details = cc....cc)

aa....aa：コマンド実行元の IP アドレス

bb....bb：キャッシュ名

cc....cc：エラー詳細情報

### 説明

キャッシュファイルの削除に失敗しました。

### 対処

エラー詳細情報を確認してください。

## KDEA08591-I

The resumption of cache files started. (management client = aa....aa)

aa....aa：コマンド実行元の IP アドレス

### 説明

キャッシュの再開を開始しました。



## KDEA08592-I

The resumption of cache files finished. (management client = aa....aa)

aa....aa : コマンド実行元の IP アドレス

### 説明

キャッシュの再開が完了しました。

## KDEA08593-E

The resumption of cache files failed. (management client = aa....aa, error details = bb....bb)

aa....aa : コマンド実行元の IP アドレス

bb....bb : エラー詳細情報

### 説明

キャッシュの再開に失敗しました。

### 対処

エラー詳細情報を確認してください。

## KDEA08594-I

The importing of cache files started. (management client = aa....aa)

aa....aa : コマンド実行元の IP アドレス

### 説明

キャッシュファイルのインポートを開始しました。

## KDEA08595-I

The importing of cache files finished. (management client = aa....aa)

aa....aa : コマンド実行元の IP アドレス

### 説明

キャッシュファイルのインポートが完了しました。

## KDEA08596-E

The importing of cache files failed. (management client = aa....aa, error details = bb....bb)

aa....aa : コマンド実行元の IP アドレス

bb....bb : エラー詳細情報

## 説明

キャッシュファイルのインポートに失敗しました。

## 対処

エラー詳細情報を確認してください。

## KDEA08597-E

```
Creation of cache files failed. (cache name = aa....aa, management client = bb....bb, error details = cc....cc)
```

aa....aa：キャッシュ名

bb....bb：コマンド実行元の IP アドレス

cc....cc：エラー詳細情報

## 説明

キャッシュファイルの作成に失敗しました。

## 対処

エラー詳細情報を確認してください。

## KDEA08598-I

```
Exporting was not executed because no memory cache exist on the server. (management client = aa....aa)
```

aa....aa：コマンド実行元の IP アドレス

## 説明

EADs サーバにメモリキャッシュが存在しなかったため、データの書き出しは実行されませんでした。

## 対処

エラー詳細情報を確認してください。

## 24.8 KDEA10000~KDEA11999

---

KDEA10000~KDEA11999 のメッセージとその対処方法について説明します。

### KDEA10001-W

The parameter is outside the valid range. The default value will be used. (parameter = aa....aa, value = bb....bb, default value = cc....cc)

aa....aa : パラメタ名

bb....bb : パラメタの指定値

cc....cc : パラメタのデフォルト値

#### 説明

パラメタの指定値が不正です。デフォルト値を使用します。

#### 対処

パラメタの指定値を見直し、正しい値に変更してください。

### KDEA10002-E

The parameter is outside the valid range or was not specified. (parameter = aa....aa, value = bb....bb)

aa....aa : パラメタ名

bb....bb : 不正なパラメタの指定値 (指定されていない場合は null または NULL)

#### 説明

パラメタが指定されていないか、または指定値が不正です。

#### 対処

パラメタの指定値を見直し、正しい値に変更したあと、EADs サーバを再起動してください。

### KDEA10005-E

A received packet was invalid for the protocol. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd)

aa....aa : 自 IP アドレス

bb....bb : 自ポート番号

cc....cc : 通信先の IP アドレス

dd....dd：通信先のポート番号

## 説明

プロトコルとして不正なパケットを受信しました。

## 対処

次の要因が考えられます。

- 通信先で問題が発生した。
- ネットワークで問題が発生した。

要因を特定し、問題を取り除いてください。

## KDEA10006-W

```
A packet incompatible with the communications protocol was received. (local = aa....aa:bb....bb, remote = cc....cc:dd....dd)
```

aa....aa：自 IP アドレス

bb....bb：自ポート番号

cc....cc：接続元の IP アドレス

dd....dd：接続元のポート番号

## 説明

互換性のないパケットを受信しました。

## 対処

次の要因が考えられます。

- 通信先で問題が発生した。
- 通信先と互換性がない。
- ネットワークで問題が発生した。

要因を特定し、問題を取り除いてください。

## KDEA10007-E

```
Some parameter values conflict. (parameter1 = aa....aa, parameter2 = bb....bb)
```

aa....aa：値が矛盾しているパラメタ 1

bb....bb：値が矛盾しているパラメタ 2

## 説明

パラメタ 1 とパラメタ 2 の値が矛盾しています。

大小関係などに問題があります。

## 対処

パラメタの値を見直し、正しい値に変更したあと、EADs サーバを再起動してください。

# 付録

## 付録 A サンプルプログラム

---

EADs では 2 種類のユーザ関クションのサンプルプログラムを提供しています。

- ExportCsvFunction  
EADs サーバに格納されているデータを CSV 形式でファイルに出力します。
- RemoveAllFunction  
EADs サーバに格納されているデータを削除します。

サンプルプログラムは、`/opt/hitachi/xeads/server/sample` ディレクトリ下に格納されています。

サンプルプログラムでは、「[20.2 ユーザ関クションで使用できる API](#)」の API を使用して、実際にデータを操作します。

なお、サンプルプログラムは C 言語のクライアント AP での実行には対応していません。Java 言語のクライアント AP または `eztool execfunc` コマンドを使用して実行してください。

### 付録 A.1 ExportCsvFunction (データの出力)

このサンプルプログラムは、EADs サーバに格納されているデータを CSV 形式でファイルに出力します。

このユーザ関クションの引数と戻り値は文字列のため、`eztool execfunc` コマンドで実行できます。

#### (1) 規則

- key が `<グループ名>:<yyyyMMddHHmmss>` に合致する形式で、対象の EADs サーバに格納されているデータが出力されます。  
年 (yyyy), 月 (MM), 日 (dd), 時 (HH), 分 (mm), 秒 (ss) の各桁は固定なので、足りない桁は 0 で補ってください。これ以外の形式の key のデータは出力されません。
- value は文字列表現である `toString()` の戻り値が出力されます。
- 出力する CSV ファイルの文字エンコーディングは、Java のデフォルトエンコーディングになります。
- key, value にコンマなどが含まれていた場合でも、そのまま出力されます。

#### (2) ユーザ関クション名

`samplefunc.ExportCsvFunction`

#### (3) ユーザ関クション引数

ユーザ関クションに次の引数を渡します。

`< CSV ファイル名 > [, < 時間範囲 >]`

データ出力先の CSV ファイル名を指定します。CSV ファイルは、<運用ディレクトリ>/store 下に出力されます。

CSV ファイル名には、最大 32 文字の半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), およびピリオド (.) が指定できます。

特定のデータを出力したい場合は、時間範囲 (開始時刻, 終了時刻) を指定します。

<時間範囲> :

[<開始時刻>] : [<終了時刻>]

開始時刻, 終了時刻は, yyyyMMddHHmmss 形式で指定します。

yyyyMMddHHmmss :

yyyy : 年, MM : 月, dd : 日, HH : 時 (00~23), mm : 分, ss : 秒

また, どちらかの時刻を省略できます。開始時刻を省略した場合は最初のデータから, 終了時刻を省略した場合は最後のデータまで出力されます。

(例) exportdata2.csv,20131120150000:

2013 年 11 月 20 日 15 時以降の key, value を exportdata2.csv に出力します。

## (4) 実行例

eztool execfunc コマンドでサンプルプログラムを実行します。

(例 1)

キャッシュ cache1 に格納されている key, value を exportdata1.csv に出力します。

```
eztool execfunc cache1 samplefunc.ExportCsvFunction exportdata1.csv
```

(例 2)

キャッシュ cache2 に格納されている 2013 年 11 月 20 日 15 時までの key, value を exportdata2.csv に出力します。

```
eztool execfunc cache2 samplefunc.ExportCsvFunction exportdata2.csv, :20131120150000
```

## (5) CSV ファイル出力例

# key,value

group0:20131122102030,value1

group0:20131122103000,value2



## (6) 例外処理

例外が発生した場合は、例外の内容をユーザログに出力し、ユーザファンクションの実行を中断します。

## (7) 戻り値

"Success" : CSV ファイルの出力に成功した場合

"Failure" : CSV ファイルの出力に失敗した場合

## (8) 注意事項

この API は複数同時に実行できません。ファンクション定義の `eads.function.<ユーザファンクション名>.max_execute_threads` パラメタに 1 を指定し、最大同時実行スレッド数を制限してください。

## 付録 A.2 RemoveAllFunction (データの削除)

このサンプルプログラムは、EADs サーバに格納されているデータを削除します。

このユーザファンクションの引数と戻り値は文字列のため、`eztool execfunc` コマンドで実行できます。

### (1) 規則

key が <グループ名>:<yyyyMMddHHmmss > に合致する形式で、対象の EADs サーバに格納されているデータが削除されます。

年 (yyyy), 月 (MM), 日 (dd), 時 (HH), 分 (mm), 秒 (ss) の各桁は固定なので、足りない桁は 0 で補ってください。これ以外の形式の key のデータは削除されません。

### (2) ユーザファンクション名

samplefunc.RemoveAllFunction

### (3) ユーザファンクション引数

特定のデータを削除したい場合は、ユーザファンクションに次の引数を渡します。

[,<時間範囲>]

<時間範囲> :

[<開始時刻>] : [<終了時刻>]

開始時刻, 終了時刻は, yyyyMMddHHmmss 形式で指定します。

yyyyMMddHHmmss :

yyyy : 年, MM : 月, dd : 日, HH : 時 (00~23), mm : 分, ss : 秒

また, どちらかの時刻を省略できます。開始時刻を省略した場合は最初のデータから, 終了時刻を省略した場合は最後のデータまで削除されます。

(例) 20131120150000:

2013年11月20日15時以降の key, value を削除します。

## (4) 実行例

`eztool execfunc` コマンドでサンプルプログラムを実行します。

(例 1)

キャッシュ `cache1` に格納されている key, value をすべて削除します。

```
eztool execfunc cache1 samplefunc.RemoveAllFunction
```

(例 2)

キャッシュ `cache2` に格納されている 2013年11月22日16時までの key, value をすべて削除します。

```
eztool execfunc cache2 samplefunc.RemoveAllFunction :20131122160000
```

## (5) 例外処理

例外が発生した場合は, 例外の内容をユーザログに出力し, ユーザファンクションの実行を中断します。

## (6) 戻り値

"Success" : データの削除に成功した場合

"Failure" : データの削除に失敗した場合

## (7) 注意事項

この API は複数同時に実行できません。ファンクション定義の `eads.function.<ユーザファンクション名>.max_execute_threads` パラメタに 1 を指定し, 最大同時実行スレッド数を制限してください。

## 付録 B Developer で使用する定義ファイルで設定される値

次の定義ファイルの一部のパラメタについては、実行環境で使用する定義ファイルとは設定される値が異なります。

- サーバ定義ファイル
- クラスタ定義ファイル
- 起動設定ファイル
- コマンド定義ファイル

### 付録 B.1 サーバ定義ファイル

Developer で使用するサーバ定義で設定される値を、次の表に示します。

次の表に示す値を省略したり、範囲外の値に変更したりした場合は、実行環境で使用する定義ファイルと同じデフォルト値が設定されます。

なお、実行環境で使用するサーバ定義と設定される値が異なるパラメタを太字で表記しています。

表 B-1 Developer で使用するサーバ定義で設定される値

項番	分類	パラメタ名	設定される値
1	通信 (「7.3 通信に依存するパラメタの設計」参照)	<b>eads.server.address</b>	localhost
2		eads.server.port	24600
3		eads.consensus.message.port	24633
4		eads.restore.port	24632
5		eads.management.port	24620
6	ログファイル (「7.4 ログファイルに依存するパラメタの設計」参照)	<b>eads.logger.dir</b>	<運用ディレクトリ>%logs
7		eads.logger.message.rotationStyle	Wrap
8		eads.logger.message.filesize	1048576
9		eads.logger.message.filenum	2
10		<b>eads.logger.message.console.enable</b>	true
11		eads.logger.exception.rotationStyle	Wrap
12		eads.logger.exception.filesize	1048576
13		eads.logger.exception.filenum	2
14		eads.logger.user.rotationStyle	Wrap
15		eads.logger.user.filesize	1048576

項番	分類	パラメタ名	設定される値
16		eads.logger.user.filenum	2
17		eads.logger.userException.rotationStyle	Wrap
18		eads.logger.userException.filesize	1048576
19		eads.logger.userException.filenum	2
20		eads.logger.cache.rotationStyle	Wrap
21		eads.logger.cache.filesize	1048576
22		eads.logger.cache.filenum	2
23		eads.prf.enable	false
24		eads.statistics.interval	60
25		eads.statistics.backupnum	7
26		バックアップ ファイル (「7.6 バック アップファイル に依存するパラ メタの設計」 参照)	eads.storeData.dir
27	メモリ・バッ ファ (「9.1 メモ リ・バッファに 関連するパラメ タの設計」 参照)	eads.connection.bufferSize	4096
28		eads.server.max.datasize.enable	false
29		eads.server.max.datasize	67108864
30		eads.consensus.external.heapsize	30
31		eads.management.resume.sendSize	1048576
32	スレッドプール およびコネク ションプール (「9.2 スレッ ドプールおよび コネクション プールに関連す るパラメタの 設計」参照)	eads.server.max_connections	10
33		eads.server.cache.max_execute_threads	10
34		eads.server.function_container.max_execute_th reads	10
35	タイムアウト (「9.3 タイム アウトに関連す るパラメタの 設計」参照)	eads.connection.timeout	3000
36		eads.server.session.keepAlive.timeout	3600
37		eads.management.resume.sendInterval	0
38	キャッシュ	eads.cache.disk.getError.isolate.enable	true

項番	分類	パラメタ名	設定される値
	(「7.7 キャッシュの運用に依存するパラメタの設計」参照)		

## 付録 B.2 クラスタ定義ファイル

Developer で使用するクラスタ定義で設定される値を、次の表に示します。

次の表に示す値を省略したり、範囲外の値に変更したりした場合は、実行環境で使用する定義ファイルと同じデフォルト値が設定されます。

なお、実行環境で使用するクラスタ定義と設定される値が異なるパラメタを太字で表記しています。

表 B-2 Developer で使用するクラスタ定義で設定される値

項番	分類	パラメタ名	設定される値
1	バックアップファイル	<code>eads.storeData.generation.count</code>	2
2	(「7.6 バックアップファイルに依存するパラメタの設計」参照)	<code>eads.storeData.stop_command.generation.count</code>	1
3	メモリ・バッファ	<code>eads.server.max_key_size</code>	1024
4	(「9.1 メモリ・バッファに関連するパラメタの設計」参照)	<code>eads.cache.prepare.num</code>	1

## 付録 B.3 起動設定ファイル

Developer で使用する起動設定で設定される値を、次の表に示します。

次の表に示す値を省略したり、範囲外の値に変更したりした場合は、実行環境で使用する定義ファイルと同じデフォルト値が設定されます。ただし、`eads.java.heapsize` パラメタおよび `eads.java.external.heapsize` パラメタについては、次の表に示す値がデフォルト値となります。

なお、実行環境で使用する起動設定と設定される値が異なるパラメタを太字で表記しています。

表 B-3 Developer で使用する起動設定で設定される値

項番	分類	パラメタ名	設定される値
1	ログファイル (「7.4 ログ ファイルに依存 するパラメタの 設計」参照)	<code>eads.prf.filesize</code>	16384
2		<code>eads.prf.filenum</code>	16
3		<code>eads.prf.level</code>	0x40000000
4		<code>eads.java.log.filesize</code>	8
5		<code>eads.java.log.filenum</code>	4
6	メモリ・バッ ファ (「9.1 メモ リ・バッファに 関連するパラメ タの設計」 参照)	<code>eads.prf.buffersize</code>	2048
7		<code>eads.java.heapsize</code>	384
8		<code>eads.java.external.heapsize</code>	128
9		<code>eads.java.permanent.size</code>	83

## 付録 B.4 コマンド定義ファイル

Developer で使用するコマンド定義で設定される値を、次の表に示します。

次の表に示す値を省略したり、範囲外の値に変更したりした場合は、実行環境で使用する定義ファイルと同じデフォルト値が設定されます。

なお、実行環境で使用するコマンド定義と設定される値が異なるパラメタを太字で表記しています。

表 B-4 Developer で使用するコマンド定義で設定される値

項番	分類	パラメタ名	設定される値
1	ログファイル (「7.4 ログ ファイルに依存 するパラメタの 設計」参照)	<code>eads.logger.dir</code>	<運用ディレクトリ>%logs
2		<code>eads.logger.message.rotationStyle</code>	Wrap
3		<code>eads.logger.message.filesize</code>	1048576
4		<code>eads.logger.message.filenum</code>	2
5		<code>eads.logger.exception.rotationStyle</code>	Wrap
6		<code>eads.logger.exception.filesize</code>	1048576
7		<code>eads.logger.exception.filenum</code>	2
8	タイムアウト (「9.3 タイム アウトに関連す るパラメタの 設計」参照)	<code>eads.management.connect.timeout</code>	3000
9		<code>eads.management.read.timeout</code>	10000
10		<code>eads.management.command.timeout</code>	60

項番	分類	パラメタ名	設定される値
11		eads.management.long_running.read.timeout	180000
12		eads.management.long_running.command.timeout	180

## 付録 B.5 キャッシュ定義ファイル

Developer で使用するキャッシュ定義で設定される値を、次の表に示します。

次の表に示す値を省略したり、範囲外の値に変更したりした場合は、実行環境で使用する定義ファイルと同じデフォルト値が設定されます。

表 B-5 Developer で使用するキャッシュ定義で設定される値

項番	分類	パラメタ名	設定される値
1	キャッシュ (「7.7 キャッシュの運用に依存するパラメタの設計」参照)	eads.cache.disk.info.dir	<運用ディレクトリ>¥store
2		eads.cache.disk.blocksize	1
3		eads.cache.restore.dataSender.sendInterval	1000
4		eads.cache.restore.dataSender.sendSize	102400

### (数字)

#### 2Way キャッシュ

メモリ領域およびディスク領域にデータを格納するキャッシュのことです。

### (英字)

#### EADs クライアント

EADs が提供するクライアントライブラリを使用して、EADs サーバに接続するユーザプログラムのことです。

#### EADs サーバ

key と value で構成されるデータを管理するサーバプロセスのことです。

### (力行)

#### キャッシュ

EADs で扱うデータ (key と value のペア) を格納する領域のことです。

キャッシュはクラスタ内に最大 64 個作成できます。

EADs クライアントは、クラスタ内のキャッシュに対してデータ操作を行います。

#### クラスタ

複数の EADs サーバの集合です。EADs クライアントからは 1 つの格納先として認識されます。

同じセグメント内に同じマルチキャストアドレスおよびポート番号を設定した EADs サーバ群で構成されます。

### (サ行)

#### 生存確認

ハートビートを送信しない EADs サーバに対して、ダウンしていないかどうかクラスタ内で確認する機能のことです。

生存確認には TCP プロトコルを使用します。



## (タ行)

### ディスクキャッシュ

ディスク領域にデータを格納するキャッシュのことです。

## (ハ行)

### ハートビート

正常に稼働していることを知らせるために、クラスタ内にマルチキャストで配信するパケットのことです。

ハートビートの送信には UDP プロトコルを使用します。

## (マ行)

### メモリキャッシュ

メモリ領域にデータを格納するキャッシュのことです。

## (ヤ行)

### ユーザイベントハンドラ

データの更新操作を契機に、自動的に動作するユーザ処理のプログラムのことです。ユーザが作成して、あらかじめ EADs サーバ上に配置しておきます。

データの更新操作と同期して動作することで、外部システムと連携できます。

### ユーザファンクション

データの集計や分析など、キャッシュに対して実行する一連のデータ操作（ユーザ処理）を定義したプログラムのことです。

ユーザが作成して、あらかじめ EADs サーバ上に配置しておき、EADs クライアントから呼び出すことで実行できます。なお、ユーザファンクションには次の 2 種類の実行方法があります。

- グループ指定による実行方法
- EADs サーバ指定による実行方法

# 索引

## 数字

2Way キャッシュ 43

2Way キャッシュ [用語解説] 760

## C

CacheException クラス 493

CacheManager クラス 479

Cache クラス 469

ClientInfo インタフェース 513

close 332

compaction 386

containsKey() [Store インタフェース] 524

cprfed 389

create 45

create() (EADs クライアント名およびクライアント定義指定) [CacheManager クラス] 485

create() (EADs クライアント名および入力ストリーム指定) [CacheManager クラス] 483

create() (EADs クライアント名およびパス指定) [CacheManager クラス] 481

create() (クライアント定義指定) [CacheManager クラス] 484

create() (入力ストリーム指定) [CacheManager クラス] 482

create() (パス指定) [CacheManager クラス] 480

create() [Cache クラス] 470

create() [Group インタフェース] 531

createcache 351

## D

DataFilter アノテーション 558

deletecache 352

deleteecf 385

deleteesd 358

descendingKeyIterator() [Group インタフェース] 527

destroy() [CacheManager クラス] 488

destroy() [EventHandler インタフェース] 551

destroy() [Function インタフェース] 505

## E

ead\_address 構造体 600

ead\_create() 582

EAD\_ERROR\_CACHE\_NOT\_CREATED [ユーザファンクション] 545

EAD\_ERROR\_CACHE\_NOT\_NEED\_STOP [C言語] 601

EAD\_ERROR\_CACHE\_NOT\_NEED\_STOP [Java言語] 495

EAD\_ERROR\_CACHE\_NOT\_STARTED [C言語] 601

EAD\_ERROR\_CACHE\_NOT\_STARTED [Java言語] 494

EAD\_ERROR\_CACHE\_SETTING [ユーザファンクション] 545

EAD\_ERROR\_CLIENT\_FINALIZED [Java言語] 494

EAD\_ERROR\_CLIENT\_OUT\_OF\_MEMORY [C言語] 605

EAD\_ERROR\_CLIENT\_OUT\_OF\_MEMORY [Java言語] 499

EAD\_ERROR\_CLIENT [C言語] 605

EAD\_ERROR\_CLIENT [Java言語] 499

EAD\_ERROR\_EXCEED\_MAX\_CONNECTION\_POOL\_SIZE [C言語] 601

EAD\_ERROR\_EXCEED\_MAX\_CONNECTION\_POOL\_SIZE [Java言語] 495

EAD\_ERROR\_ILLEGAL\_ARGUMENT [C言語] 601

EAD\_ERROR\_INIT\_CLUSTERINFO [C言語] 602

EAD\_ERROR\_INIT\_CLUSTERINFO [Java言語] 496

EAD\_ERROR\_INIT\_INVALID\_PROPERTY [C言語] 602

EAD\_ERROR\_INIT\_INVALID\_PROPERTY [Java言語] 496

EAD\_ERROR\_INIT\_LOGGER [C言語] 602

EAD\_ERROR\_INIT\_LOGGER [Java 言語] 496

EAD\_ERROR\_INIT\_PROPERTIES [C 言語] 601

EAD\_ERROR\_INIT\_PROPERTIES [Java 言語] 495

EAD\_ERROR\_INIT [C 言語] 601

EAD\_ERROR\_INIT [Java 言語] 495

EAD\_ERROR\_INVALID\_NODE\_ADDRESS [C 言語] 601

EAD\_ERROR\_INVALID\_NODE\_ADDRESS [Java 言語] 495

EAD\_ERROR\_INVALID\_PARAMETER [Java 言語] 494

EAD\_ERROR\_INVALID\_PARAMETER [ユーザファンクション] 545

EAD\_ERROR\_NET\_CLUSTERINFO [C 言語] 603

EAD\_ERROR\_NET\_CLUSTERINFO [Java 言語] 497

EAD\_ERROR\_NET\_CONNECTION [C 言語] 603

EAD\_ERROR\_NET\_CONNECTION [Java 言語] 497

EAD\_ERROR\_NET\_PROTOCOL [C 言語] 603

EAD\_ERROR\_NET\_PROTOCOL [Java 言語] 497

EAD\_ERROR\_NET\_TIMEOUT [C 言語] 603

EAD\_ERROR\_NET\_TIMEOUT [Java 言語] 497

EAD\_ERROR\_NET [C 言語] 602

EAD\_ERROR\_NET [Java 言語] 497

EAD\_ERROR\_NOT\_DESERIALIZABLE [Java 言語] 495

EAD\_ERROR\_NOT\_DESERIALIZABLE [ユーザファンクション] 545

EAD\_ERROR\_NOT\_SERIALIZABLE [Java 言語] 495

EAD\_ERROR\_NOT\_SERIALIZABLE [ユーザファンクション] 545

EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST [C 言語] 605

EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST [Java 言語] 499

EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST [ユーザファンクション] 546

EAD\_ERROR\_SERVER\_EVENTHANDLER [C 言語] 604

EAD\_ERROR\_SERVER\_EVENTHANDLER [Java 言語] 499

EAD\_ERROR\_SERVER\_EVENTHANDLER [ユーザファンクション] 546

EAD\_ERROR\_SERVER\_INCOMPATIBLE\_CLUSTER\_INFO [C 言語] 604

EAD\_ERROR\_SERVER\_INCOMPATIBLE\_CLUSTER\_INFO [Java 言語] 499

EAD\_ERROR\_SERVER\_NOT\_RUNNING [C 言語] 605

EAD\_ERROR\_SERVER\_NOT\_RUNNING [Java 言語] 499

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST [C 言語] 605

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST [Java 言語] 499

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST [ユーザファンクション] 546

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED [C 言語] 604

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED [Java 言語] 499

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED [ユーザファンクション] 546

EAD\_ERROR\_SERVER\_UNAVAILABLE [C 言語] 604

EAD\_ERROR\_SERVER\_UNAVAILABLE [Java 言語] 499

EAD\_ERROR\_SERVER\_UNSUPPORTED\_REQUEST [C 言語] 604

EAD\_ERROR\_SERVER\_UNSUPPORTED\_REQUEST [Java 言語] 498

EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST [C 言語] 605

EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST [Java 言語] 499

EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST [ユーザファンクション] 546

EAD\_ERROR\_SERVER [C 言語] 604

EAD\_ERROR\_SERVER [Java 言語] 498

EAD\_ERROR\_SERVER [ユーザファンクション] 545

EAD\_ERROR\_UNEXPECTED [C 言語] 601

EAD\_ERROR\_UNEXPECTED [Java 言語] 494  
 EAD\_ERROR\_UNEXPECTED [ユーザファンクショ  
ン] 545  
 ead\_execute\_function() 586  
 ead\_execute\_node\_function() 590  
 ead\_get\_node() 595  
 ead\_get\_nodelist() 594  
 ead\_get\_slave\_nodelist() 596  
 ead\_get() 585  
 ead\_init\_client\_n() 576  
 ead\_init\_client() 575  
 ead\_nodelist 構造体 598  
 ead\_node 構造体 599  
 ead\_object 構造体 598  
 EAD\_OK [C 言語] 601  
 ead\_put\_array\_value() 581  
 ead\_put() 580  
 ead\_remove() 586  
 ead\_replace() 584  
 ead\_start\_cache() 577  
 ead\_stop\_cache() 578  
 ead\_terminate\_client() 579  
 ead\_update() 583  
 ead\_value\_element 構造体 597  
 eads\_[xxx]\_[EADs サーバ ID].esd [ファイル名]  
151  
 eads\_cache\_stats\_[n].csv [ファイル名] 135  
 eads\_cache\_stats.csv [ファイル名] 135  
 eads\_cache.<キャッシュ名>.properties [ファイ  
ル名] 128  
 eads\_client\_comm\_trace[n].log [ファイル名]  
172  
 eads\_client\_maintenance[n].log [ファイル名]  
172  
 eads\_client\_message[n].log [ファイル名] 172  
 eads\_cluster.properties [ファイル名] 128  
 eads\_data\_[EADs サーバ ID]\_[キャッシュ名]\_[レン  
ジ ID]\_[nnnnn].ecf 157  
 eads\_dist\_maintenance[n].log [ファイル名] 134  
 eads\_event\_stats\_[n].csv [ファイル名] 135  
 eads\_event\_stats.csv [ファイル名] 135  
 eads\_function\_stats\_[n].csv [ファイル名] 135  
 eads\_function\_stats.csv [ファイル名] 135  
 eads\_index\_[EADs サーバ ID]\_[キャッシュ名]\_[レン  
ジ ID]\_[nnnnn].ecf 157  
 eads\_info\_[EADs サーバ ID]\_[キャッシュ名].ecf  
[ファイル名] 157  
 eads\_maintenance\_stats\_[n].csv [ファイル名]  
135  
 eads\_maintenance\_stats.csv [ファイル名] 135  
 eads\_management\_exception.log [ファイル名]  
134  
 eads\_management\_exception[n].log [ファイ  
ル名] 134  
 eads\_management\_maintenance[n].log [ファイ  
ル名] 134  
 eads\_management\_message.log [ファイル名]  
134  
 eads\_management\_message[n].log [ファイル名]  
134  
 eads\_management.properties [ファイル名] 128  
 eads\_server\_cache.log [ファイル名] 134  
 eads\_server\_cache[n].log [ファイル名] 134  
 eads\_server\_exception.log [ファイル名] 134  
 eads\_server\_exception[n].log [ファイル名] 134  
 eads\_server\_maintenance[n].log [ファイル名]  
134  
 eads\_server\_message.log [ファイル名] 134  
 eads\_server\_message[n].log [ファイル名] 134  
 eads\_server.properties [ファイル名] 128  
 eads\_single\_[xxx]\_[EADs サーバ ID].esd [ファイ  
ル名] 151  
 eads\_start.properties [ファイル名] 128  
 eads\_stats\_[n].csv [ファイル名] 135  
 eads\_stats.csv [ファイル名] 135  
 eads\_stop\_[YYYYMMDDhhmmss]\_[EADs サーバ  
ID].esd [ファイル名] 151  
 eads\_user\_exception.log [ファイル名] 134  
 eads\_user\_exception[n].log [ファイル名] 134  
 eads\_user\_message.log [ファイル名] 134  
 eads\_user\_message[n].log [ファイル名] 134

eads.cache.consensus.timeout 207  
eads.cache.prepare.num 184  
eads.cache.restore.dataSender.sendInterval 163  
eads.cache.restore.dataSender.sendSize 164  
eads.client.<接続先 EADs サーバ>.address 170  
eads.client.<接続先 EADs サーバ>.port 170  
eads.client.clusterInfo.update.interval 211  
eads.client.node.list 170  
eads.cluster.assertive.threshold 207  
eads.cluster.boot.timeout 207  
eads.cluster.controller.cache.max\_execute\_threads 189  
eads.cluster.failureDetector.connect.timeout 206  
eads.cluster.failureDetector.port 132  
eads.cluster.failureDetector.read.timeout 206  
eads.cluster.failureDetector.retry 207  
eads.cluster.heartbeat.address 150  
eads.cluster.heartbeat.interval 206  
eads.cluster.heartbeat.port 150  
eads.cluster.heartbeat.timeout 206  
eads.connection.bufferSize [クライアント定義] 185  
eads.connection.bufferSize [サーバ定義] 181  
eads.connection.receive.timeout 211  
eads.connection.send.timeout 211  
eads.connection.timeout 206  
eads.connectionPool.exceedMaxSizeError.enable 190  
eads.connectionPool.poolSize 189  
eads.consensus.external.heapSize 184  
eads.consensus.fillgap.copy.dataSize 183  
eads.consensus.fillgap.copy.timeout 208  
eads.consensus.message.bufferSize 181  
eads.consensus.message.port 132  
eads.consensus.message.send.max\_queuedata\_size 182  
eads.consensus.message.send.queueSize 181  
eads.function.<ユーザファンクション名>.max\_execute\_threads 189  
eads.gracefulstop.wait\_time 207  
eads.java.external.heapSize 185  
eads.java.heapSize 185  
eads.java.log.fileNum 144  
eads.java.log.fileSize 144  
eads.java.permanent.size 185  
eads.logger.cache.fileNum 143  
eads.logger.cache.fileSize 142  
eads.logger.cache.rotationStyle 142  
eads.logger.commTrace.commNo.prefix 175  
eads.logger.commTrace.enable 174  
eads.logger.commTrace.fileNum 175  
eads.logger.commTrace.fileSize 174  
eads.logger.dir [クライアント定義] 173  
eads.logger.dir [コマンド定義] 145  
eads.logger.dir [サーバ定義] 139  
eads.logger.exception.fileNum [コマンド定義] 146  
eads.logger.exception.fileNum [サーバ定義] 141  
eads.logger.exception.fileSize [コマンド定義] 146  
eads.logger.exception.fileSize [サーバ定義] 141  
eads.logger.exception.rotationStyle [コマンド定義] 146  
eads.logger.exception.rotationStyle [サーバ定義] 140  
eads.logger.message.console.enable [クライアント定義] 174  
eads.logger.message.console.enable [サーバ定義] 140  
eads.logger.message.fileNum [クライアント定義] 174  
eads.logger.message.fileNum [コマンド定義] 145  
eads.logger.message.fileNum [サーバ定義] 140  
eads.logger.message.fileSize [クライアント定義] 174  
eads.logger.message.fileSize [コマンド定義] 145  
eads.logger.message.fileSize [サーバ定義] 140

eads.logger.message.rotationStyle [コマンド定義] 145  
eads.logger.message.rotationStyle [サーバ定義] 140  
eads.logger.user.filenum 141  
eads.logger.user.filesize 141  
eads.logger.user.rotationStyle 141  
eads.logger.userException.filenum 142  
eads.logger.userException.filesize 142  
eads.logger.userException.rotationStyle 142  
eads.management.command.timeout 209  
eads.management.long\_running.command.timeout 210  
eads.management.long\_running.read.timeout 209  
eads.management.port 132  
eads.management.resume.sendInterval 209  
eads.management.resume.sendSize 184  
eads.node.< EADs サーバ ID >.address 149  
eads.node.< EADs サーバ ID >.port 150  
eads.node.< EADs サーバ ID >.position 150  
eads.prf.bufferSize 185  
eads.prf.enable 143  
eads.prf.filenum 144  
eads.prf.filesize 144  
eads.prf.level 144  
eads.replication.factor 150  
eads.restore.connection.bufferSize 182  
eads.restore.dataReceiver.read.timeout 208  
eads.restore.dataSender.send.timeout 207  
eads.restore.dataSender.sendInterval 208  
eads.restore.dataSender.sendSize 183  
eads.restore.port 132  
eads.server.address 131  
eads.server.cache.max\_execute\_threads 188  
eads.server.function\_container.max\_execute\_threads 189  
eads.server.max\_connections 188  
eads.server.max\_key\_size 184  
eads.server.max.datasize 181  
eads.server.max.datasize.enable 181  
eads.server.port 132  
eads.server.session.keepAlive.timeout 207  
eads.statistics.backupnum 143  
eads.statistics.compaction.effect.division 143  
eads.statistics.interval 143  
eads.storeData.dir 154  
eads.storeData.generation.count 155  
eads.storeData.stop\_command.generation.count 155  
EADsStoreException クラス 544  
EADs クライアント 37  
EADs クライアント [用語解説] 760  
EADs サーバ 36  
EADs サーバの開始 323  
EADs サーバの強制終了 397  
EADs サーバの縮退 398  
EADs サーバの縮退処理 199  
EADs サーバのフォアグラウンドでの開始 324  
EADs サーバ [用語解説] 760  
EADs サーバを開始する  
    キャッシュを新規に作成する 216  
    ファイルからデータを読み込んでキャッシュを作成する 223  
EADs サーバを終了する  
    メモリ上のデータを破棄する 230  
    メモリ上のデータをファイルに書き出す 232  
ehjavalog[nn].log [ファイル名] 135  
EventContext インタフェース 552  
EventHandler インタフェース 550  
execfunc 367  
execute() [EventHandler インタフェース] 551  
execute() [Function インタフェース] 505  
executeFunction() (EADs サーバおよび受信タイムアウト指定) [Cache クラス] 478  
executeFunction() (EADs サーバ指定) [Cache クラス] 477  
executeFunction() (グループおよび受信タイムアウト指定) [Cache クラス] 475

executeFunction() (グループ指定) [Cache クラス] 474

Explicit ヒープ 85

export 353

ezserver 324

ezstart 323

eztool 326

eztool コマンドのサブコマンド 329

## F

FunctionContext インタフェース 506

Function インタフェース 504

## G

get 49

get() (Key インタフェース指定) [Group インタフェース] 535

get() (文字列指定) [Group インタフェース] 534

get() [Cache クラス] 473

getAddress() [Node クラス] 492

getAddress() [ServerInfo インタフェース] 515

getArgument() [FunctionContext インタフェース] 508

getCache() [CacheManager クラス] 486

getCacheDataFileRemainingSize() [ServerInfo インタフェース] 518

getCacheDataFileSpecifiedNumber() [ServerInfo インタフェース] 519

getCacheDataFileSpecifiedSize() [ServerInfo インタフェース] 517

getCacheDataFileUnusedNumber() [ServerInfo インタフェース] 520

getCacheName() [EventContext インタフェース] 553

getCacheName() [FunctionContext インタフェース] 507

getCacheNames() [ServerInfo インタフェース] 515

getCacheType() [ServerInfo インタフェース] 516

getClientInfo() [EventContext インタフェース] 556

getClientInfo() [FunctionContext インタフェース] 510

getEHeapSize() [ServerInfo インタフェース] 516

getEHeapUsageSize() [ServerInfo インタフェース] 517

getErrorCode() [CacheException クラス] 494

getErrorCode() [EADsStoreException クラス] 544

getFunctionName() [InitConfig インタフェース] 512

getFunctionProperty() [InitConfig インタフェース] 511

getFunctionPropertyNames() [InitConfig インタフェース] 512

getGroup() (グループ名指定) [Store インタフェース] 523

getGroup() [Store インタフェース] 523

getGroupName() [EventContext インタフェース] 553

getGroupName() [FunctionContext インタフェース] 507

getGroupNameSet() [Group インタフェース] 538

getGroupNameSet() [Store インタフェース] 524

getInitConfig() [FunctionContext インタフェース] 510

getIpl() [ClientInfo インタフェース] 513

getKey() [EventContext インタフェース] 554

getLastUpdateTime() (Key インタフェース指定) [Group インタフェース] 537

getLastUpdateTime() (文字列指定) [Group インタフェース] 537

getLastUpdateTime() [EventContext インタフェース] 555

getLogger() [EventContext インタフェース] 557

getLogger() [FunctionContext インタフェース] 509

getMethod() [EventContext インタフェース] 556

getName() [ServerInfo インタフェース] 515

getNode() [CacheManager クラス] 489

getNodeId() [Node クラス] 491

getNodeList() [CacheManager クラス] 488  
getPid() [ClientInfo インタフェース] 513  
getPosition 347  
getPosition() [Node クラス] 492  
getServerInfo() [FunctionContext インタフェース] 510  
getServerName() [EventContext インタフェース] 553  
getServerName() [FunctionContext インタフェース] 507  
getSlaveNodeList() [CacheManager クラス] 490  
getStore() (キャッシュ名指定) [FunctionContext インタフェース] 509  
getStore() [FunctionContext インタフェース] 508  
getValue() [EventContext インタフェース] 554  
getValueUsageSize() (Key インタフェース指定) [Group インタフェース] 539  
getValueUsageSize() (文字列指定) [Group インタフェース] 538  
getValueUsageSize() [Group インタフェース] 540  
get [サブコマンド] 361  
Group インタフェース 525

## H

higherKeyIterator() [Group インタフェース] 528

## I

import 356  
importecf 383  
init() [EventHandler インタフェース] 550  
init() [Function インタフェース] 504  
InitConfig インタフェース 511  
InitializeException クラス 500  
InternalClientException クラス 500  
InternalServerException クラス 501, 546  
isEnabled() [Node クラス] 492  
isNewData() [EventContext インタフェース] 555  
isolate 398

## J

javacore[PID].[YYMMDDhhmmss].txt [ファイル名] 135  
javalog[nn].log [ファイル名] 135  
Java ヒープ 84  
Java ログファイル 133

## K

keyIterator() [Group インタフェース] 527  
Key インタフェース 540

## L

listcache 340  
listecf 373  
listesd 343  
listevent 369  
listfunc 364  
log() (形式 1) [UserLogger インタフェース] 542  
log() (形式 2) [UserLogger インタフェース] 542  
lowerDescendingKeyIterator() [Group インタフェース] 529

## N

Node クラス 491

## O

open 332  
OperationFilter アノテーション 557

## P

prf\_[nnn] [ファイル名] 134  
PRF デーモン 38  
PRF トレース情報の編集出力 389  
PRF トレースに出力される情報 316  
PRF トレースの仕組み 315  
PRF トレースの取得方法 304  
PRF トレースファイル 133  
Priority アノテーション 559  
put 45



put() (Key インタフェース指定) [Group インタフェース] 530  
put() (文字列指定) [Group インタフェース] 529  
put() [Cache クラス] 470  
putStackTrace() (形式 1) [UserLogger インタフェース] 543  
putStackTrace() (形式 2) [UserLogger インタフェース] 543  
put [サブコマンド] 359  
put と replace の違い 48

## R

remove 50  
remove() (Key インタフェース指定) [Group インタフェース] 536  
remove() (文字列指定) [Group インタフェース] 536  
remove() [Cache クラス] 473  
removeCache() [CacheManager クラス] 487  
remove [サブコマンド] 363  
replace 47  
replace() (Key インタフェース指定) [Group インタフェース] 534  
replace() (文字列指定) [Group インタフェース] 533  
replace() [Cache クラス] 472  
resume 381

## S

ServerCommunicationException クラス 501  
ServerInfo インタフェース 514  
snapshot 393  
status 333  
stop 396  
Store インタフェース 522

## T

threaddump 392  
toString() [Group インタフェース] 526  
toString() [Key インタフェース] 541

toString() [Node クラス] 493

## U

unlock 350  
update 46  
update() (Key インタフェース指定) [Group インタフェース] 532  
update() (文字列指定) [Group インタフェース] 531  
update() [Cache クラス] 471  
UserLogger インタフェース 541  
UserOperationException クラス 502, 547

## い

イベント ID 315  
インストールディレクトリ [Developer] 409

## う

運用ディレクトリ 113  
運用ディレクトリ [Developer] 412

## え

永続データに関する情報の一覧表示 373  
永続データの再配置 383

## か

稼働状態 77  
環境依存パラメタ 81

## き

起動シェル 38  
起動設定ファイル 128  
起動ログファイル 133  
キャッシュ 42  
キャッシュインデックスファイル 157  
キャッシュ情報ファイル 157  
キャッシュ定義ファイル 128  
キャッシュデータファイル 157  
キャッシュデータファイルのコンパクション 386  
キャッシュの一覧表示 340

キャッシュの一覧を表示する 237  
キャッシュの再開 381  
キャッシュの削除 352  
キャッシュの作成 351  
キャッシュファイル 156  
キャッシュファイル操作ログファイル 133  
キャッシュファイルの削除 385  
キャッシュ [用語解説] 760

## く

クライアント定義ファイル 167  
クラスタ 37  
クラスタ一部稼働中 75  
クラスタおよび EADs サーバの状態を確認する 236  
クラスタ稼働中 74  
クラスタ構成情報 69  
クラスタ状態の確認 333  
クラスタ定義ファイル 128  
クラスタ動作不能 75  
クラスタの運用操作 326  
クラスタの終了 396  
クラスタの復旧処理 201  
クラスタの閉塞化 332  
クラスタの閉塞解除 332  
クラスタ [用語解説] 760  
グループ名 58

## こ

合意メッセージ 52  
更新操作の履歴 85  
更新操作の履歴の補完処理 202  
コネクションプール 79  
コマンド定義ファイル 128  
コンパクション 238

## さ

サーバ定義ファイル 128

## し

指定したデータの格納 359  
指定したデータの削除 363  
指定したデータの取得 361  
シフト方式 136  
縮退状態 77  
    障害発生時の対処 293  
障害情報の取得方法 303  
初期化状態 76  
初期化中 76

## す

ストアデータファイル 151  
ストアデータファイルキー 151  
ストアデータファイルの一覧表示 343  
ストアデータファイルの削除 358  
スレッドダンプ 133  
スレッドダンプの出力 392  
スレッドダンプの取得方法 304  
スレッドプール 78

## せ

生存確認 70  
生存確認 [用語解説] 760

## ち

チューニングパラメタ 82

## つ

通信トレースファイル 172

## て

定義ファイル (EADs クライアント) 167  
定義ファイル (EADs サーバ) 128  
停止状態 77  
停止中 77  
ディスクキャッシュ 42  
ディスクキャッシュ [用語解説] 761  
ディレクトリ構成 [Developer] 409

ディレクトリ構成 [EADs クライアント] 122  
ディレクトリ構成 [EADs サーバ] 110  
データの書き出し 353  
データの格納先の表示 347  
データの多重化 67  
データの読み込み 356

## と

統計情報ファイル 133

## は

ハートビート 69  
ハートビート [用語解説] 761  
排他解除 350  
排他制御 54

## ふ

ファンクション定義ファイル 128  
分散保守ログファイル 133

## へ

閉塞状態 77  
閉塞中 77

## ほ

保守ログファイル 133

## め

メッセージログファイル 133  
メモリキャッシュ 42  
メモリキャッシュ [用語解説] 761  
メモリ構成 84

## ゆ

ユーザイベントハンドラ 64  
ユーザイベントハンドラの実行可否の表示 369  
ユーザイベントハンドラ [用語解説] 761  
ユーザファンクション 61  
ユーザファンクションの実行 367  
ユーザファンクションの実行可否の表示 364

ユーザファンクション [用語解説] 761  
ユーザ例外ログファイル 133  
ユーザログファイル 133

## よ

要素名 58

## ら

ラップアラウンド方式 136

## れ

例外ログファイル 133  
レスポンス低下  
原因の調査方法 307  
障害情報の取得 301  
列挙型 MethodType 561  
列挙型 OperationType 560  
列挙型 PriorityLevel 561  
レンジ 56

## ろ

ログ, 設定情報, ハードウェア情報, およびネットワーク情報の収集 393