

インメモリデータグリッド

# Hitachi Elastic Application Data Store ユーザー ズガイド

解説・手引・文法・操作書

3020-3-V22-60

---

## 前書き

### ■ 対象製品

●適用 OS : Red Hat Enterprise Linux 7 (64-bit x86\_64), Red Hat Enterprise Linux 8 (64-bit x86\_64)

P-9W43-8A61 Hitachi Elastic Application Data Store 06-10

P-9W43-8B61 Hitachi Elastic Application Data Store Client for Java 06-10

P-9W43-8C61 Hitachi Elastic Application Data Store Client for C 06-10

### ■ 輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

### ■ 商標類

HITACHI, Cosminexus, JP1, uCosminexus は、株式会社 日立製作所の商標または登録商標です。  
AMD は、Advanced Micro Devices, Inc.の商標です。

Intel は、Intel Corporation またはその子会社の商標です。

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。

Oracle および Java は、オラクルおよびその関連会社の登録商標です。

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat は、米国およびその他の国における Red Hat, Inc.の登録商標です。

Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat Enterprise Linux は、米国およびその他の国における Red Hat, Inc.の登録商標です。

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.

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 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.

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



1. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)
2. This product includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com))
3. This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com))
4. 本製品には OpenSSL Toolkit ソフトウェアを OpenSSL License および Original SSLeay License に従い使用しています。OpenSSL License および Original SSLeay License は以下のとおりです。

#### LICENSE ISSUES

=====

The OpenSSL toolkit stays under a double license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit.

See below for the actual license texts.

#### OpenSSL License

-----

/\*

=====  
=====

\* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\*

\* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\*

\* 3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

\* "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

\*

\* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [openssl-core@openssl.org](mailto:openssl-core@openssl.org).

\*

\* 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

\*

\* 6. Redistributions of any form whatsoever must retain the following acknowledgment:

\* "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

\*

\* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT



\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
\* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
\* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
\* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
\* OF THE POSSIBILITY OF SUCH DAMAGE.

\*  
=====  
=====

\*  
\* This product includes cryptographic software written by Eric Young  
\* (eay@cryptsoft.com). This product includes software written by Tim  
\* Hudson (tjh@cryptsoft.com).

\*/  
Original SSLeay License

-----  
/\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)  
\* All rights reserved.

\*  
\* This package is an SSL implementation written  
\* by Eric Young (eay@cryptsoft.com).  
\* The implementation was written so as to conform with Netscapes SSL.

\*  
\* This library is free for commercial and non-commercial use as long as  
\* the following conditions are aheared to. The following conditions  
\* apply to all code found in this distribution, be it the RC4, RSA,  
\* lhash, DES, etc., code; not just the SSL code. The SSL documentation  
\* included with this distribution is covered by the same copyright terms  
\* except that the holder is Tim Hudson (tjh@cryptsoft.com).

\*  
\* Copyright remains Eric Young's, and as such any Copyright notices in  
\* the code are not to be removed.  
\* If this package is used in a product, Eric Young should be given attribution  
\* as the author of the parts of the library used.

\* This can be in the form of a textual message at program startup or  
\* in documentation (online or textual) provided with the package.  
\*  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\* 1. Redistributions of source code must retain the copyright  
\* notice, this list of conditions and the following disclaimer.  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the  
\* documentation and/or other materials provided with the distribution.  
\* 3. All advertising materials mentioning features or use of this software  
\* must display the following acknowledgement:  
\* "This product includes cryptographic software written by  
\* Eric Young (eay@cryptsoft.com)"  
\* The word 'cryptographic' can be left out if the routines from the library  
\* being used are not cryptographic related :-).  
\* 4. If you include any Windows specific code (or a derivative thereof) from  
\* the apps directory (application code) you must include an acknowledgement:  
\* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"  
\*  
\* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND  
\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
\* PURPOSE  
\* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
\* CONSEQUENTIAL  
\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE  
\* GOODS  
\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
\* STRICT  
\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

\*

\* The licence and distribution terms for any publically available version or  
\* derivative of this code cannot be changed. i.e. this code cannot simply be  
\* copied and put under another distribution licence  
\* [including the GNU Public Licence.]

\*/

-----  
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://itdoc.hitachi.co.jp/Pages/document\\_list/manuals/cosmiv11.html](http://itdoc.hitachi.co.jp/Pages/document_list/manuals/cosmiv11.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。

## ■ 発行

2023年1月 3020-3-V22-60

## ■ 著作権

All Rights Reserved. Copyright (C) 2015, 2023, Hitachi, Ltd.

## 変更内容

### 変更内容 (3020-3-V22-60) Hitachi Elastic Application Data Store 06-10

追加・変更内容	変更箇所
Java ヒープサイズの見積もり式を変更しました。	4.1.2(1), 4.1.2(2)
計算式の数値の切り上げに関連する注意事項を追加しました。	4.1.2(1)(b), 4.1.3(2), 4.3.2(1), 4.4.1(3), 4.4.1(4), 4.4.1(6), 4.4.1(7), 4.4.2(1), 4.4.2(2), 4.4.3(1)
1EADS サーバ当たりのログファイルの容量を算出する方法の記述と計算式を変更しました。	4.2.1(1)
統計情報ファイルの容量の計算方法を変更しました。	4.2.1(1)(h)
eads.admin.backup.exportCommand.generation.maxNum パラメタが 0 である場合の説明を追加しました。	4.2.1(2)
1EADS サーバプロセス当たりのファイルディスクリプタ数の計算方法を変更しました。	4.3.2(1)
レコード 1 件分のサイズを見積もる計算式の注意事項を変更しました。	4.4.1(2)
EADS サーバに同梱されていた Java クライアントは、Java クライアントインストーラでインストールするように変更しました。	5.1.3(1), 6.1, 6.1.1(1), 6.1.1(2), 6.3
定義ファイルへのパラメタの指定方法から、シェルスクリプトの場合の説明を削除しました。	7.2
次の適用 OS を追加しました。 <ul style="list-style-type: none"><li>Red Hat Enterprise Linux 8 (64-bit x86_64)</li></ul> また、次の OS を削除しました。 <ul style="list-style-type: none"><li>Red Hat Enterprise Linux 6 (64-bit x86_64)</li></ul>	7.7.2(2)
複数のキャッシュを使用する場合の注意点に、参考情報を追加しました。	10.2.3
次に示すコマンドの-rd または--remotedebug オプションに、IP アドレスまたはホスト名を指定できるようにしました。 <ul style="list-style-type: none"><li>ezstart コマンド</li><li>ezserver コマンド</li></ul> また、これに伴い、それぞれのコマンドの注意事項も追加しました。	14.2.1, 14.2.2
コマンド実行に失敗した場合の対処について、注意事項の記述を変更、または追加しました。	14.2.1(6), 14.2.2(6), 14.3.2(5), 14.3.3(5), 14.3.13(6), 14.3.14(6), 14.3.15(6), 14.3.16(6), 14.3.17(6), 14.3.24(6), 14.3.27(9), 14.3.28(6), 14.3.29(6), 14.3.32(6), 14.3.33(6), 14.3.35(6)

追加・変更内容	変更箇所
スレッドダンプのファイル容量が大きくなる場合の注意事項を追加しました。	14.3.31(5)
ソースプログラムのコンパイル方法の説明を変更しました。	16.3, 19.3

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、Hitachi 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

Hitachi 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	Hitachi Elastic Application Data Store	
	EADS クライアント	EADS クライアント (Java)	Hitachi Elastic Application Data Store Client for Java
		EADS クライアント (C)	Hitachi Elastic Application Data Store Client for C
JavaVM		Java Virtual Machine	
JDK		Java Developer's Kit	

### ■ 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
Hard disk drive	HDD
Key-Value store	KVS

英字での表記	英略語
Non-Volatile Memory Express	NVMe
Solid state drive	SSD

## ■ 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>
...	この記号の直前の項目を繰り返し指定できます。	<pre>eztool status --columns &lt;列名&gt; [, &lt;列名&gt;] ...</pre> <p>このコマンドの例の場合、&lt;列名&gt; を繰り返し指定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	8
はじめに	10

## 第1編 解説

<b>1</b>	<b>Hitachi 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	37
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)	46



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)	50
2.4.6	データの削除 (remove)	51
2.4.7	キャッシュに対する一括操作	51
2.4.8	EADS サーバの位置を意識しないデータアクセス	52
2.4.9	データアクセスの流れ	53
2.4.10	データアクセス時の排他制御	56
2.4.11	データ自動削除機能	58
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	59
2.5.1	データの分散配置の概要	59
2.5.2	データの分散配置の詳細	60
2.5.3	EADS サーバの追加とデータの再配置 (スケールアウトとリバランス)	61
2.6	特定の EADS サーバへのデータ配置 (key のグルーピング)	66
2.6.1	key のグルーピング	66
2.6.2	key をグルーピングした場合のデータの分散配置	68
2.7	ユーザファンクションによる効率的なデータ処理	69
2.7.1	ユーザファンクションの仕組み	69
2.8	イベントリスナによるイベントの発生を契機にしたデータ処理	72
2.8.1	イベントリスナの仕組み	72
2.9	データの多重化	75
2.9.1	データの多重化の概要	75
2.9.2	データの多重化による、障害時の処理の継続	75
2.10	クラスタ監視	77
2.10.1	ハートビートの送信によるクラスタ監視の概要	77
2.10.2	一定数の同意によって確定する EADS サーバダウン	78
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	81
2.12	クラスタおよび EADS サーバの状態遷移	82
2.12.1	クラスタの状態遷移	82
2.12.2	EADS サーバのクラスタへの参加状況	83
2.12.3	EADS サーバの状態遷移	84
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	89
2.13.1	スレッドプール	89
2.13.2	コネクションプール	90
2.14	ノンブロッキング I/O 通信制御	91

## 第2編 設計・構築

- 3 システム設計・構築の流れ 92**
  - 3.1 システム設計・構築の流れ 93
    - 3.1.1 必要なリソースを確認する 93
    - 3.1.2 インストールおよびセットアップを行う 93
    - 3.1.3 環境依存パラメタを設計する 93
    - 3.1.4 チューニングパラメタを設計する 94
  
- 4 必要なリソースを確認する 95**
  - 4.1 メモリ使用量を見積もる 96
    - 4.1.1 メモリ構成 96
    - 4.1.2 Java ヒープサイズを見積もる 97
    - 4.1.3 Explicit ヒープサイズを見積もる 110
    - 4.1.4 EADS サーバのメモリ使用量を見積もる 117
  - 4.2 ディスク使用量を見積もる 123
    - 4.2.1 EADS サーバのディスク使用量を見積もる 123
    - 4.2.2 EADS クライアントのディスク使用量を見積もる 129
  - 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 131
    - 4.3.1 スレッド数を見積もる 131
    - 4.3.2 ファイルディスクリプタ数を見積もる 132
  - 4.4 キャッシュファイルの容量を見積もる 135
    - 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 135
    - 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 141
    - 4.4.3 ディスクキャッシュの格納先として不揮発性メモリを使用する場合のリソースを見積もる 142
  
- 5 インストールおよびセットアップを行う (EADS サーバ) 144**
  - 5.1 EADS サーバをインストールする 145
    - 5.1.1 インストール前にすること 145
    - 5.1.2 インストール手順 145
    - 5.1.3 インストール後にすること 146
  - 5.2 EADS サーバをセットアップする 149
    - 5.2.1 運用ディレクトリを作成する 149
    - 5.2.2 定義ファイルを編集する 150
    - 5.2.3 AP を配置する 151
  - 5.3 テストを行う 152
    - 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) 152
    - 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 152
    - 5.3.3 EADS サーバを終了する 154
  - 5.4 EADS サーバをアンセットアップする 155

5.5	EADS サーバをアンインストールする	156
<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>157</b>
6.1	EADS クライアントをインストールする	158
6.1.1	インストール後にすること	158
6.2	EADS クライアントをセットアップする	161
6.2.1	AP を配置する	161
6.2.2	クライアント定義ファイルを編集する	161
6.3	EADS クライアントをアンインストールする	162
<b>7</b>	<b>環境依存パラメタを設計する (EADS サーバ)</b>	<b>163</b>
7.1	定義ファイルの種類 (EADS サーバ)	164
7.2	定義ファイルの形式	166
7.3	通信に依存するパラメタの設計	167
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	167
7.3.2	通信に依存するパラメタ	168
7.4	ログファイルに依存するパラメタの設計	171
7.4.1	ログファイルの種類	171
7.4.2	ファイルの出力先の設定	171
7.4.3	ファイルサイズおよびファイル数の設定	174
7.4.4	統計情報ファイルのローテーションの設定	174
7.4.5	ログファイルに依存するパラメタ	175
7.5	クラスタ構成に依存するパラメタの設計	184
7.5.1	EADS サーバの位置の設定	184
7.5.2	データの多重度の設定	185
7.5.3	クラスタ構成に依存するパラメタ	186
7.6	バックアップファイルに依存するパラメタの設計	188
7.6.1	ファイルの出力先の設定	188
7.6.2	ストアデータファイルの世代数の設定	189
7.6.3	バックアップファイルに依存するパラメタ	191
7.7	キャッシュの運用に依存するパラメタの設計	193
7.7.1	キャッシュタイプごとのパラメタの設定	193
7.7.2	キャッシュファイルの種類と格納先の設定	194
7.7.3	キャッシュファイルのサイズの設定	201
7.7.4	キャッシュの運用に依存するパラメタ	201
7.8	ノンブロッキング I/O 通信制御に依存するパラメタの設計	212
7.8.1	ノンブロッキング I/O 通信制御の設定	212
7.8.2	滞留中 I/O データ流量制御	214
7.8.3	ノンブロッキング I/O 通信制御に依存するパラメタ	214

- 7.9 スケールアウト処理・リバランス処理に依存するパラメタの設計 217
- 7.9.1 スケールアウト処理・リバランス処理に依存するパラメタ 217
  
- 8 環境依存パラメタを設計する (EADS クライアント) 221**
- 8.1 定義ファイルの種類 (EADS クライアント) 222
- 8.2 定義ファイルの形式 223
- 8.3 通信に依存するパラメタの設計 224
- 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 224
- 8.3.2 通信に依存するパラメタ 224
- 8.4 ログファイルに依存するパラメタの設計 227
- 8.4.1 ログファイルの種類 227
- 8.4.2 ファイルの出力先の設定 227
- 8.4.3 ファイルサイズおよびファイル数の設定 228
- 8.4.4 ログファイルに依存するパラメタ 228
  
- 9 チューニングパラメタを設計する 231**
- 9.1 メモリ・バッファに関連するパラメタの設計 232
- 9.1.1 メモリサイズの設定 232
- 9.1.2 バッファサイズの設定 233
- 9.1.3 メモリ・バッファに関連するパラメタ 234
- 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 243
- 9.2.1 最大同時接続数の設定 243
- 9.2.2 最大同時実行スレッド数の設定 243
- 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 244
- 9.3 タイムアウトに関連するパラメタの設計 248
- 9.3.1 通信を監視するタイマの設定 248
- 9.3.2 クラスタを監視するタイマの設定 256
- 9.3.3 タイムアウトに関連するパラメタ 266
- 9.4 コマンドの運用に関連するパラメタの設計 276
- 9.4.1 コマンドの運用に関連するパラメタ 276
- 9.5 AP の運用に関連するパラメタの設計 277
- 9.5.1 AP の運用に関連するパラメタ 277
- 9.6 コンパクションに関連するパラメタの設計 279
- 9.6.1 コンパクションの効果のしきい値の設定 279
- 9.6.2 コンパクションに関連するパラメタ 279

## 第3編 運用

- 10 通常運用 281**
- 10.1 システム運用管理者が行うこと 282

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)	283
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)	283
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)	285
10.2.3	複数のキャッシュを使用する場合の注意点	289
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)	291
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)	291
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)	294
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)	298
10.4.1	EADS サーバを終了する手順	298
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)	300
10.5.1	EADS サーバを終了する手順	300
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)	302
10.6.1	EADS サーバを終了する手順	302
10.7	クラスタおよび EADS サーバの状態を確認する	304
10.8	キャッシュの一覧を表示する	305
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)	306
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	307
10.9.2	コンパクションを中断する手順	311
<b>11</b>	<b>保守運用 312</b>	
11.1	EADS サーバをクラスタに追加したい	313
11.1.1	クラスタを停止させないで、EADS サーバをクラスタに追加する手順 (スケールアウト処理・リバランス処理)	313
11.1.2	クラスタを停止してから、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)	319
11.1.3	クラスタを停止してから、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)	320
11.2	EADS サーバをクラスタから削除したい	328
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)	328
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)	329
11.3	各レンジに格納できる key の数を均等にしたい	336
11.3.1	クラスタを停止させないで、各レンジに格納できる key の数を均等にする手順 (リバランス処理)	336
11.3.2	クラスタを停止してから、各レンジに格納できる key の数を均等にする手順	336
11.4	定義内容を変更したい	338
11.4.1	定義内容を変更する手順	338
11.4.2	定義内容を変更する場合の注意点	339
11.5	キャッシュを追加または削除したい	341
11.5.1	メモリキャッシュを追加または削除する手順	341
11.5.2	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	343

- 11.6 バックアップを取得したい 349
- 11.6.1 データのバックアップを取得する手順 (メモリ上のキャッシュだけを使用する場合) 349
- 11.6.2 データのバックアップを取得する手順 (ディスク上のキャッシュを使用する場合) 351
- 11.7 ストアデータファイルを管理したい 355
- 11.7.1 ストアデータファイルの確認および削除手順 355
- 11.8 データの格納先を確認したい 357
- 11.9 グループ名の一覧を確認したい 358
- 11.10 key の一覧を確認したい 360
- 11.11 ユーザファンクションが各 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 361
- 11.12 クラスタを停止させないで、EADS サーバの修正パッチを適用したい 362
- 11.12.1 EADS サーバの修正パッチを適用する手順 362
- 11.13 統計情報を取得したい 364
- 11.13.1 統計情報の格納先 364
- 11.13.2 統計情報 (eads\_stats.csv) 364
- 11.13.3 キャッシュの統計情報 (eads\_cache\_stats.csv) 369
- 11.13.4 ユーザファンクションの統計情報 (eads\_function\_stats.csv) 376
- 11.13.5 レンジごとの統計情報 (eads\_store\_stats.csv) 380
- 11.13.6 ノンブロッキング I/O 通信制御の統計情報 (eads\_nonblocking\_stats.csv) 382
- 11.14 データの格納先の空き容量を管理したい 384
- 11.14.1 データの格納先の空き容量を管理する手順 (メモリ上のキャッシュだけを使用する場合) 385
- 11.14.2 データの格納先の空き容量を管理する手順 (ディスク上のキャッシュを使用する場合) 387
- 11.15 優先度が低いデータを自動的に削除したい (データ自動削除機能の運用) 389
- 11.15.1 優先度が低いデータを自動的に削除する手順 389
- 11.16 キャッシュファイルを管理したい 395
- 11.16.1 キャッシュファイルの確認および削除手順 395

## 12 障害運用 397

- 12.1 障害に備えた運用 (障害監視と障害検知) 398
- 12.1.1 メッセージの監視 398
- 12.1.2 EADS サーバプロセスの監視 399
- 12.2 障害発生時のシステム運用管理者の作業 400
- 12.2.1 縮退状態が発生した場合 400
- 12.2.2 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 405
- 12.2.3 レスポンス低下の報告があった場合 409
- 12.3 障害情報の取得方法 411
- 12.3.1 原因調査に必要な障害情報 411
- 12.3.2 統計情報の取得方法 411
- 12.3.3 スレッドダンプの取得方法 411

<b>13</b>	<b>障害原因の調査 413</b>
13.1	レスポンス低下の原因調査 414
13.1.1	原因調査の流れ 414
13.1.2	原因の調査方法 415
<b>14</b>	<b>コマンドリファレンス 421</b>
14.1	コマンドの格納先 422
14.2	EADS のコマンド 423
14.2.1	ezstart (EADS サーバの開始) 423
14.2.2	ezserver (EADS サーバのフォアグラウンドでの開始) 426
14.2.3	eztool (クラスタの運用操作) 430
14.3	eztool コマンドのサブコマンド 432
14.3.1	コマンド間の排他制御 433
14.3.2	close (クラスタの閉塞) 435
14.3.3	open (クラスタの閉塞解除) 437
14.3.4	status (クラスタ状態の確認) 438
14.3.5	listconf (最新パラメタの一覧表示) 449
14.3.6	listcache (キャッシュの一覧表示) 454
14.3.7	listesd (ストアデータファイルの一覧表示) 458
14.3.8	listgroup (グループ名の一覧表示) 463
14.3.9	listkey (key の一覧表示) 468
14.3.10	getposition (データの格納先の表示) 473
14.3.11	storeusage (レンジおよびキャッシュの使用状況の確認) 476
14.3.12	unlock (排他解除) 482
14.3.13	createcache (キャッシュの作成) 483
14.3.14	deletecache (キャッシュの削除) 486
14.3.15	export (データの書き出し) 487
14.3.16	import (データの読み込み) 491
14.3.17	deleteesd (ストアデータファイルの削除) 494
14.3.18	put (指定したデータの格納) 496
14.3.19	get (指定したデータの取得) 498
14.3.20	remove (指定したデータの削除) 500
14.3.21	removeall (全データの削除) 502
14.3.22	listfunc (ユーザファンクションの実行可否の表示) 504
14.3.23	execfunc (ユーザファンクションの実行) 508
14.3.24	rebalance (レンジに格納できる key の数の均等化) 511
14.3.25	listecf (永続データに関する情報の一覧表示) 516
14.3.26	resume (キャッシュの再開) 525
14.3.27	expotecf (キャッシュファイルの書き出し) 527



- 14.3.28 importecf (永続データの再配置) 530
- 14.3.29 deleteecf (キャッシュファイルの削除) 534
- 14.3.30 compaction (キャッシュデータファイルのコンパクション) 536
- 14.3.31 threaddump (スレッドダンプの出力) 540
- 14.3.32 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) 541
- 14.3.33 stop (クラスタの終了) 545
- 14.3.34 forcestop (EADS サーバの強制終了) 547
- 14.3.35 isolate (EADS サーバの縮退) 548
- 14.4 実行結果を表示する eztool コマンドのサブコマンドの表示内容 550
  - 14.4.1 表示内容の構成要素 550
  - 14.4.2 表示形式の指定方法 551
  - 14.4.3 列のフィルタリングの指定方法 552
  - 14.4.4 行のフィルタリングの指定方法 553
  - 14.4.5 条件一致の指定方法 556

## 第4編 AP 開発

### 15 AP 開発の流れ 561

- 15.1 AP 開発の流れ 562
  - 15.1.1 開発環境を構築する 562
  - 15.1.2 AP を作成する 562
  - 15.1.3 AP をテストする 563
  - 15.1.4 作成した AP を実行環境に移行する 563
- 15.2 AP を開発する前に知っておくべき前提条件 564
  - 15.2.1 AP の作成言語と EADS クライアント 564
  - 15.2.2 key, グループ名, value, キャッシュ名, EADS クライアント名に指定できるデータ 565
  - 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) 569

### 16 クライアント AP を作成する (Java 言語) 571

- 16.1 ソースプログラムを作成する (Java 言語) 572
  - 16.1.1 キャッシュへのアクセスとデータ操作の流れ 572
- 16.2 クライアント AP 作成上の留意事項 (Java 言語) 576
  - 16.2.1 EADS クライアントの初期設定に関する留意事項 576
  - 16.2.2 キャッシュへのアクセス開始に関する留意事項 576
  - 16.2.3 データ操作に関する留意事項 577
  - 16.2.4 イベントリスナに関する留意事項 578
  - 16.2.5 キャッシュへのアクセス終了に関する留意事項 579
  - 16.2.6 EADS クライアントの使用終了に関する留意事項 580
- 16.3 ソースプログラムをコンパイルする (Java 言語) 581



<b>17</b>	<b>ユーザファンクションを作成する</b>	<b>582</b>
17.1	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	583
17.1.1	ユーザファンクションの作成言語	583
17.1.2	ユーザファンクションの実行方法の種類	583
17.1.3	EADS サーバが使用する Java クラスローダ	583
17.1.4	ユーザファンクションの作成の流れ	584
17.2	ソースプログラムを作成する (ユーザファンクション)	586
17.2.1	ユーザファンクションの流れ	586
17.3	ユーザファンクション作成上の留意事項	588
17.3.1	jar ファイル名に関する留意事項	588
17.3.2	パッケージ名に関する留意事項	588
17.3.3	ユーザファンクションの実装に関する留意事項	588
17.3.4	効率的に key の一覧を取得する方法	589
17.4	ファンクション定義ファイルを作成する (任意)	591
17.5	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)	593
17.5.1	ユーザファンクションをコンパイルする手順	593
17.6	ユーザファンクションをパッケージングする	594
17.6.1	ユーザファンクションをパッケージングする手順	594
17.7	ユーザファンクションをデプロイする	595
17.7.1	ユーザファンクションをデプロイする手順	595
17.8	ユーザファンクションを実行する	597
17.8.1	ユーザファンクションの呼び出し	597
17.8.2	ユーザログへの出力	597
17.8.3	ユーザファンクション運用上の留意事項	597
17.9	実行環境に配布する	599
<b>18</b>	<b>API リファレンス (Java 言語)</b>	<b>600</b>
18.1	Java 言語のクライアントライブラリが提供する API	601
18.1.1	Cache クラス	602
18.1.2	CacheManager クラス	630
18.1.3	Node クラス	648
18.1.4	FailureOperationInfo クラス	651
18.1.5	CacheException クラス	653
18.1.6	InitializeException クラス	670
18.1.7	InternalClientException クラス	671
18.1.8	InternalServerException クラス	671
18.1.9	ServerCommunicationException クラス	672
18.1.10	UserOperationException クラス	672
18.1.11	BatchOperationException クラス	673

18.1.12	AllFailureException クラス	674
18.1.13	PartFailureException クラス	674
18.1.14	ClusterEventListener インタフェース	675
18.1.15	NodeEventListener インタフェース	676
18.1.16	ClusterEventContext インタフェース	677
18.1.17	NodeEventContext インタフェース	678
18.1.18	列挙型 ClusterEventType	680
18.1.19	列挙型 NodeEventType	681
18.2	ユーザファンクションで使用できる API	682
18.2.1	Function インタフェース	683
18.2.2	FunctionContext インタフェース	685
18.2.3	InitConfig インタフェース	691
18.2.4	ClientInfo インタフェース	693
18.2.5	ServerInfo インタフェース	695
18.2.6	ClusterInfo インタフェース	704
18.2.7	CacheInfo インタフェース	710
18.2.8	Store インタフェース	714
18.2.9	Group インタフェース	733
18.2.10	Key インタフェース	759
18.2.11	Value インタフェース	761
18.2.12	UserLogger インタフェース	764
18.2.13	EADsStoreException クラス	766
18.2.14	InternalServerErrorException クラス	771
18.2.15	UserOperationException クラス	771
18.2.16	列挙型 CacheType	772
<b>19</b>	<b>クライアント AP を作成する (C 言語)</b>	<b>773</b>
19.1	ソースプログラムを作成する (C 言語)	774
19.1.1	キャッシュへのアクセスとデータ操作の流れ	774
19.2	クライアント AP 作成上の留意事項 (C 言語)	780
19.2.1	EADS クライアントの初期設定に関する留意事項	780
19.2.2	キャッシュへのアクセス開始に関する留意事項	780
19.2.3	データ操作に関する留意事項	780
19.2.4	イベントリスナに関する留意事項	782
19.2.5	キャッシュへのアクセス終了に関する留意事項	783
19.2.6	EADS クライアントの終了に関する留意事項	784
19.2.7	その他の留意事項	784
19.3	ソースプログラムをコンパイルする (C 言語)	785

<b>20</b>	<b>API リファレンス (C 言語) 786</b>
20.1	C 言語のクライアントライブラリが提供する関数 787
20.1.1	ead_init_client() (EADS クライアントの初期設定) 791
20.1.2	ead_init_client_n() (EADS クライアントの初期設定) 792
20.1.3	ead_start_cache() (キャッシュへのアクセス開始) 793
20.1.4	ead_stop_cache() (キャッシュへのアクセス終了) 794
20.1.5	ead_get_cache_name() (キャッシュ名の取得) 795
20.1.6	ead_terminate_client() (EADS クライアントの終了) 796
20.1.7	ead_put() (key と value の格納) 797
20.1.8	ead_put_array_value() (複数の value の連結格納) 798
20.1.9	ead_put_all() (key と value の一括格納) 799
20.1.10	ead_create() (key と value の新規格納) 801
20.1.11	ead_update() (value の更新) 802
20.1.12	ead_replace() (value の置換) 803
20.1.13	ead_get() (value の取得) 804
20.1.14	ead_get_all() (value の一括取得) 805
20.1.15	ead_get_group() (グループ指定による value の一括取得) 807
20.1.16	ead_remove() (value の削除) 808
20.1.17	ead_remove_all() (value の一括削除) 809
20.1.18	ead_remove_group() (グループ指定による value の一括削除) 811
20.1.19	ead_remove_node() (EADS サーバ指定による value の一括削除) 812
20.1.20	ead_get_group_names() (最上位階層のグループ名の一覧の取得) 813
20.1.21	ead_get_group_keys() (グループ指定による key の一覧の取得) 815
20.1.22	ead_get_node_keys() (EADS サーバ指定による key の一覧の取得) 816
20.1.23	ead_get_group_count() (最上位階層グループ数の取得) 817
20.1.24	ead_get_group_key_count() (グループ指定による key の数の取得) 818
20.1.25	ead_get_node_key_count() (EADS サーバ指定による key の数の取得) 819
20.1.26	ead_get_group_first_key() (グループ指定による先頭の key の取得) 820
20.1.27	ead_get_node_first_key() (EADS サーバ指定による先頭の key の取得) 821
20.1.28	ead_get_group_next_key() (グループ指定による次の key の取得) 823
20.1.29	ead_get_node_next_key() (EADS サーバ指定による次の key の取得) 824
20.1.30	ead_execute_function() (key またはグループ指定によるユーザ関数の実行) 826
20.1.31	ead_execute_function_rt() (key またはグループ、および受信タイムアウト指定によるユーザ関数の実行) 827
20.1.32	ead_execute_node_function() (EADS サーバ指定によるユーザ関数の実行) 829
20.1.33	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) 831
20.1.34	ead_get_nodelist() (接続先 EADS サーバの情報取得) 833
20.1.35	<b>【非推奨】</b> ead_get_node() (指定した key の本来のコピー元 EADS サーバの情報取得) 834

20.1.36	ead_get_slave_nodelist() (データの本来のコピー先 EADS サーバの情報取得)	835
20.1.37	ead_get_current_master_node() (現時点のコピー元 EADS サーバの情報取得)	837
20.1.38	ead_get_original_master_node() (本来のコピー元 EADS サーバの情報取得)	838
20.1.39	ead_add_cluster_eventlistener() (クラスタイベントリスナの追加)	839
20.1.40	ead_add_node_eventlistener() (ノードイベントリスナ追加)	841
20.1.41	ead_remove_cluster_eventlistener() (クラスタイベントリスナ削除)	843
20.1.42	ead_remove_node_eventlistener() (ノードイベントリスナ削除)	844
20.1.43	ead_value_element 構造体 (value 情報)	845
20.1.44	ead_key_value_pair 構造体 (key と value の組み合わせ)	845
20.1.45	ead_keys 構造体 (複数の key)	846
20.1.46	ead_group_names 構造体 (複数のグループ名)	846
20.1.47	ead_put_all_results 構造体 (ead_put_all()の実行結果)	847
20.1.48	ead_get_all_results 構造体 (ead_get_all()の実行結果)	848
20.1.49	ead_get_group_results 構造体 (ead_get_group()の実行結果)	848
20.1.50	ead_remove_all_results 構造体 (ead_remove_all()の実行結果)	849
20.1.51	ead_failure_operation_info 構造体 (一括操作の実行で失敗した操作の情報)	850
20.1.52	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)	850
20.1.53	ead_nodelist 構造体 (EADS サーバ情報)	851
20.1.54	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)	851
20.1.55	ead_address 構造体 (EADS サーバのアドレス情報)	852
20.1.56	ead_cluster_eventlistener 構造体 (クラスタイベントリスナの関数)	853
20.1.57	ead_node_eventlistener 構造体 (ノードイベントリスナの関数)	854
20.1.58	ead_cluster_eventcontext 構造体 (クラスタイベントリスナの情報)	855
20.1.59	ead_node_eventcontext 構造体 (ノードイベントリスナの情報)	855
20.1.60	ead_node_eventtype 構造体 (ノードイベントリスナの実行契機)	856
20.2	クライアントライブラリのエラーコード (C 言語)	858

## 第5編 よく使用する一覧集・メッセージ

### 21 よく使用する一覧集 870

21.1	パラメター一覧	871
21.1.1	EADS サーバで使用する定義ファイル	871
21.1.2	EADS クライアントで使用する定義ファイル	879
21.2	コマンド一覧	881
21.3	API 一覧	883
21.3.1	Java 言語のライブラリが提供する API	883
21.3.2	C 言語のクライアントライブラリが提供する関数	886
21.4	クライアントライブラリが出力するエラーコード	890
21.4.1	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	890

21.4.2 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 890

21.5 最小値・最大値一覧 891

## 22 メッセージ 893

22.1 メッセージの記述形式 894

22.2 KDEA00001～KDEA01999 896

22.3 KDEA02000～KDEA02999 916

22.4 KDEA03000～KDEA03999 920

22.5 KDEA04000～KDEA05999 925

22.6 KDEA06000～KDEA07999 952

22.7 KDEA08000～KDEA09999 968

22.8 KDEA10000～KDEA11999 1015

## 付録 1018

付録 A サンプルプログラム 1019

付録 A.1 ExportCsvFunction (データの出力) 1019

付録 A.2 RemoveAllFunction (データの削除) 1021

付録 B 03-60 以前のバージョンからの移行手順 1023

付録 B.1 パラメタの移動 (EADS サーバ) 1023

付録 B.2 パラメタの追加 1024

付録 B.3 パラメタの再設計 1025

付録 B.4 名称が変更になったパラメタの一覧 1027

付録 B.5 無効となったパラメタの一覧 1031

付録 C 用語解説 1033

## 索引 1036

## 1

## Hitachi Elastic Application Data Store とは

Hitachi 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、Hitachi 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

---

Hitachi 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。

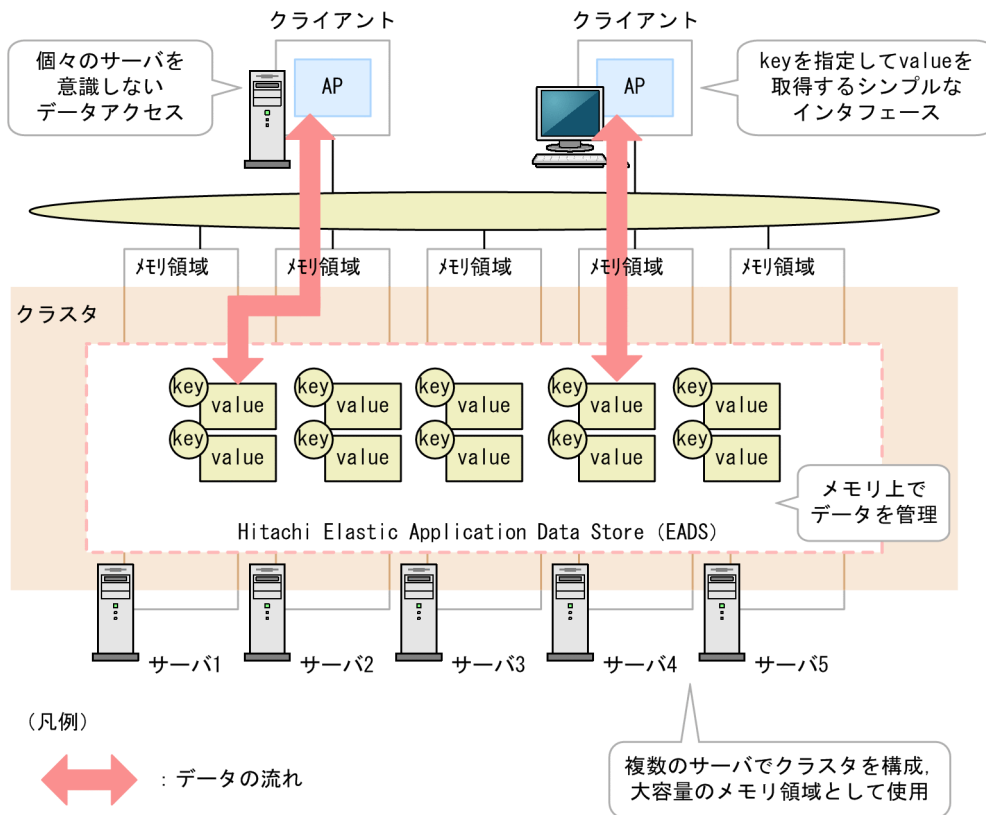
NoSQL の登場によって、RDBMS だけにすべてのデータ管理を任せるのではなく、RDBMS とそれを補完するソフトウェアを組み合わせることで、多様な要件に応えることができるようになってきました。

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 1 つの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、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。

### 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。

#### 参考

データの格納先として SSD を使用することもできます。高速かつ低レイテンシな NVMe インタフェースを採用した高性能 SSD などを使用する場合には、高性能 SSD の特性を考慮したファイルアクセスができます。

また、従来のフラッシュメモリよりも高速に読み書きができ、かつ電源が失われても書き込んだデータが保持される不揮発性のメモリ、いわゆるストレージクラスメモリもデータの格納先として使用できます。

## 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

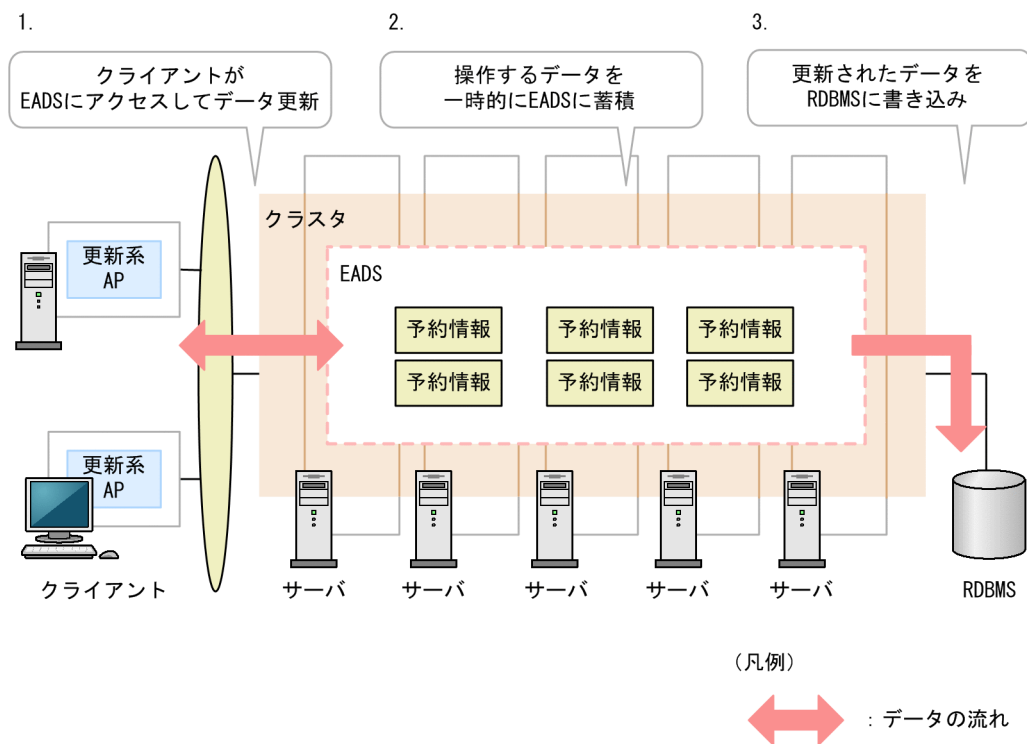
key と value というシンプルな形式でデータを管理するため、SQL のような複雑な問い合わせを必要とする非定型業務には向きません。

### 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. 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。

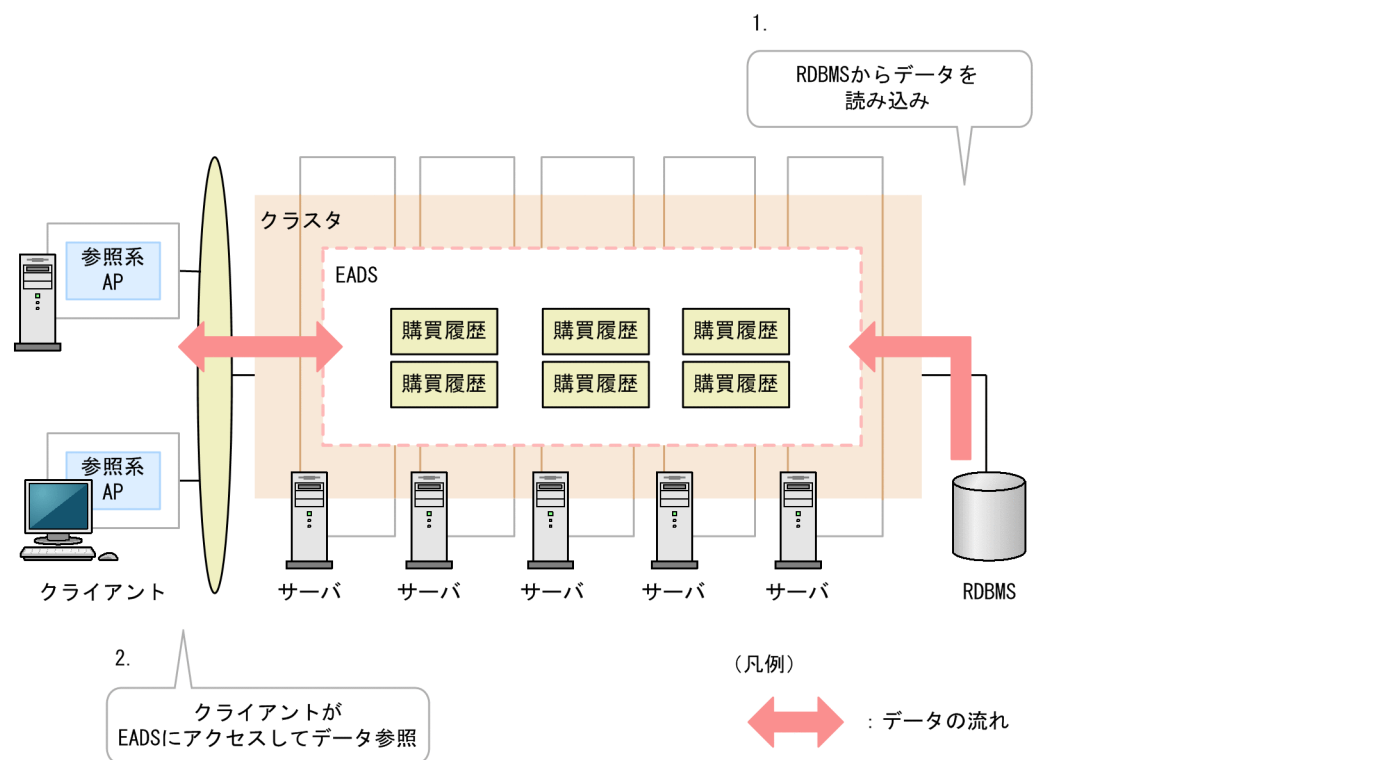
データベースにデータを直接書き込むのではなく、EADS をライトバッファとして利用することで、ディスクアクセスに掛かるオーバヘッドをなくすことができ、高速にデータを処理できます。

## 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

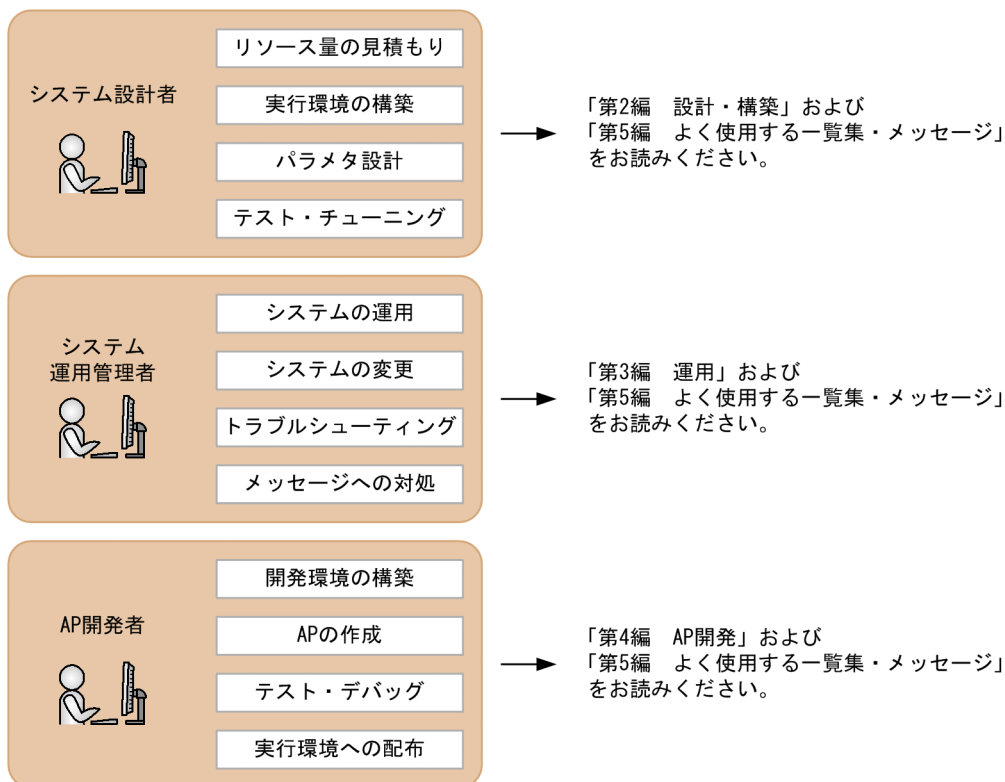
このマニュアルでは、次の対象読者を想定しています。

- 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。

- 第 1 編 解説
- 第 2 編 設計・構築
- 第 3 編 運用
- 第 4 編 AP 開発
- 第 5 編 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

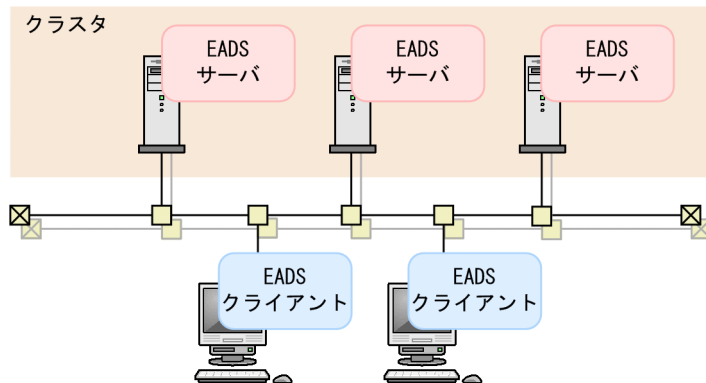
EADS を導入したシステムの構成、およびプロセス構成について説明します。

### 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

実行環境



#### (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。

- Hitachi 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）  
Hitachi Elastic Application Data Store Client for Java
- EADS クライアント（C）  
Hitachi Elastic Application Data Store Client for C

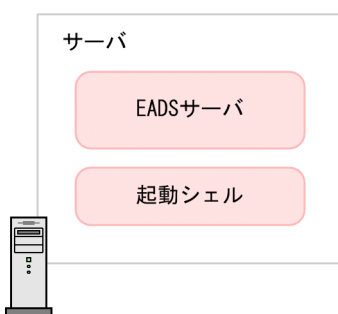
## 注意事項

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。

## 2.1.2 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) 起動シェル

EADS サーバを起動します。

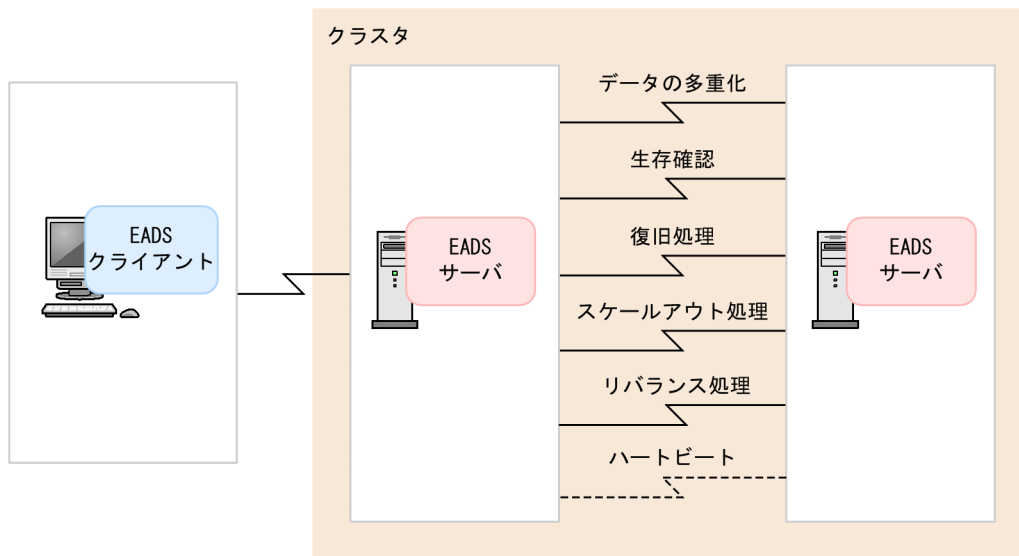
## 2.2 EADS の通信処理の仕組み

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
- スケールアウト処理 (EADS サーバの追加)

- リバランス処理

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

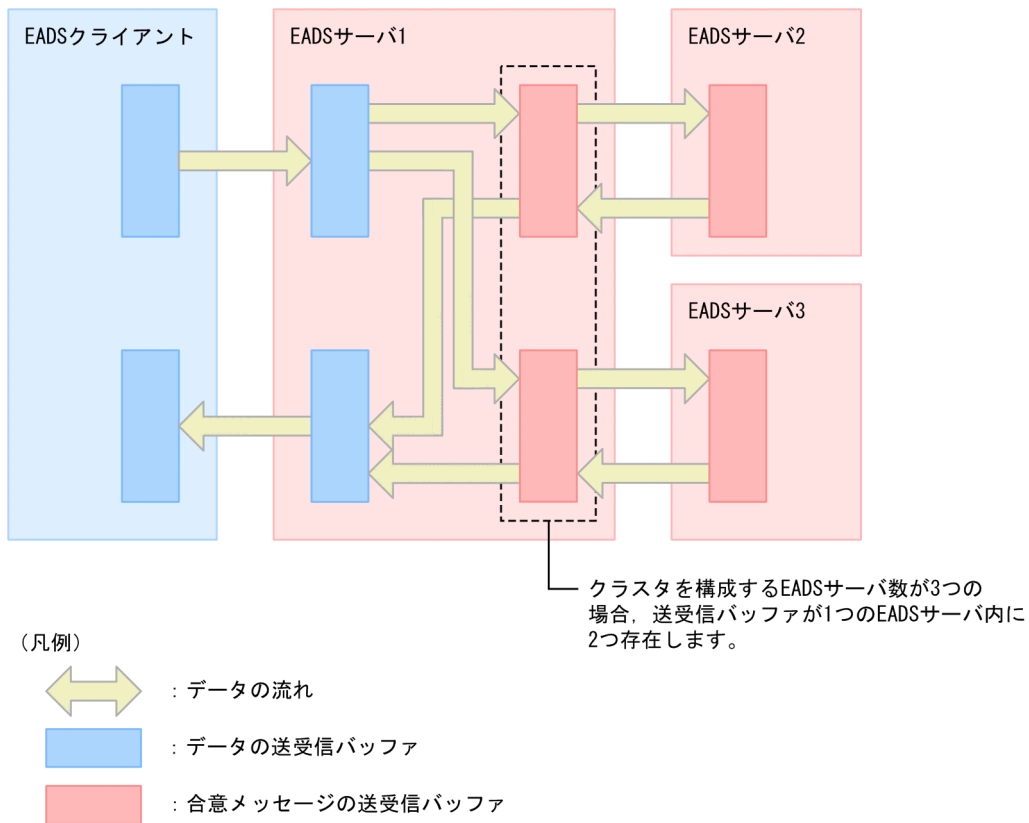
TCP プロトコルを使用する EADS クライアント・EADS サーバ間、および EADS サーバ間の通信では、次のバッファを使用します。

- 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



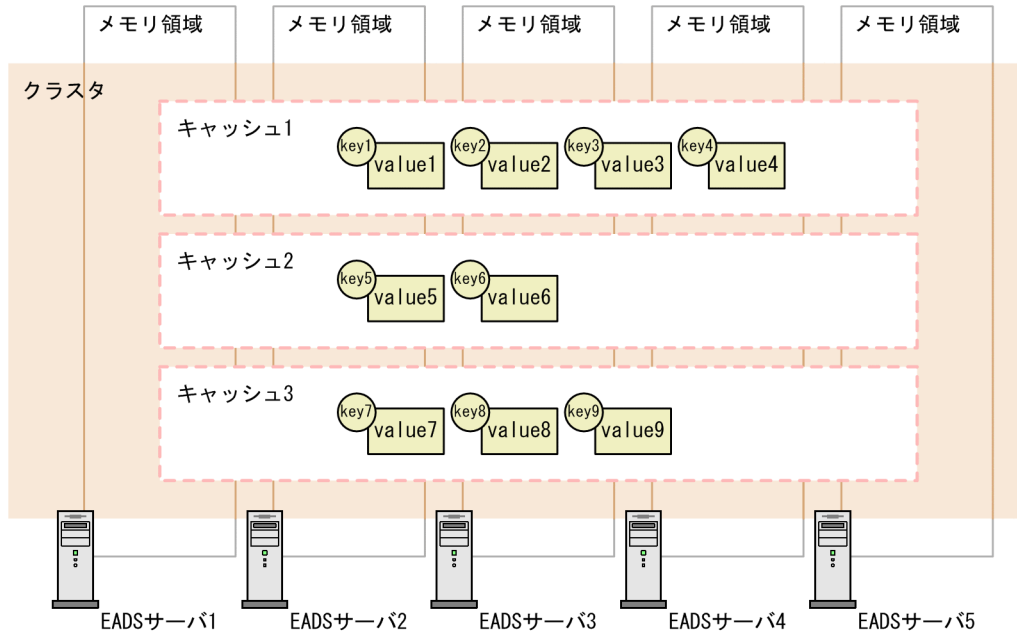
合意メッセージについては、「[2.4.9 データアクセスの流れ](#)」を参照してください。

## 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 16 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「[11.5 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)

操作の種類によっては、キャッシュ内のデータを一括で操作することもできます。

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>
データの更新操作 (一括操作)	<ul style="list-style-type: none"><li>• putAll</li><li>• removeAll</li></ul>	<code>eztool removeall</code> コマンド
データの参照操作	get	<code>eztool get</code> コマンド
データの参照操作 (一括操作)	getAll	該当するコマンドはありません。

### 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, putAll) 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>put All</li> <li>eztool put コマンド</li> </ul>	○	○
create	○	×
update	×	○
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)

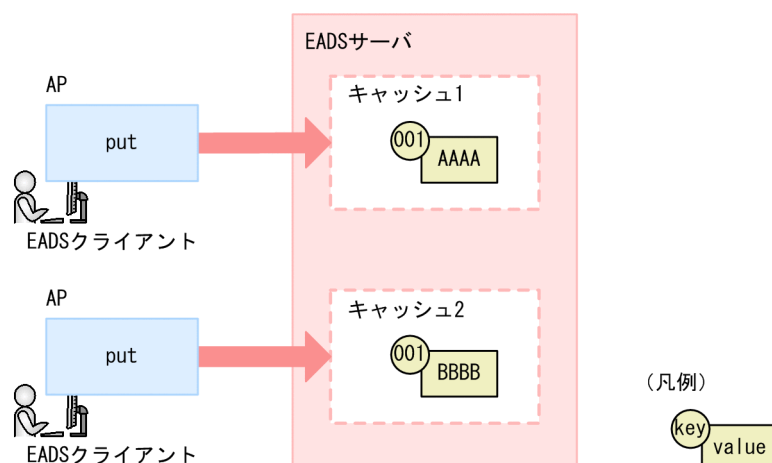
キャッシュにデータを格納します。

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。格納先のキャッシュ名が異なれば、key が同じ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



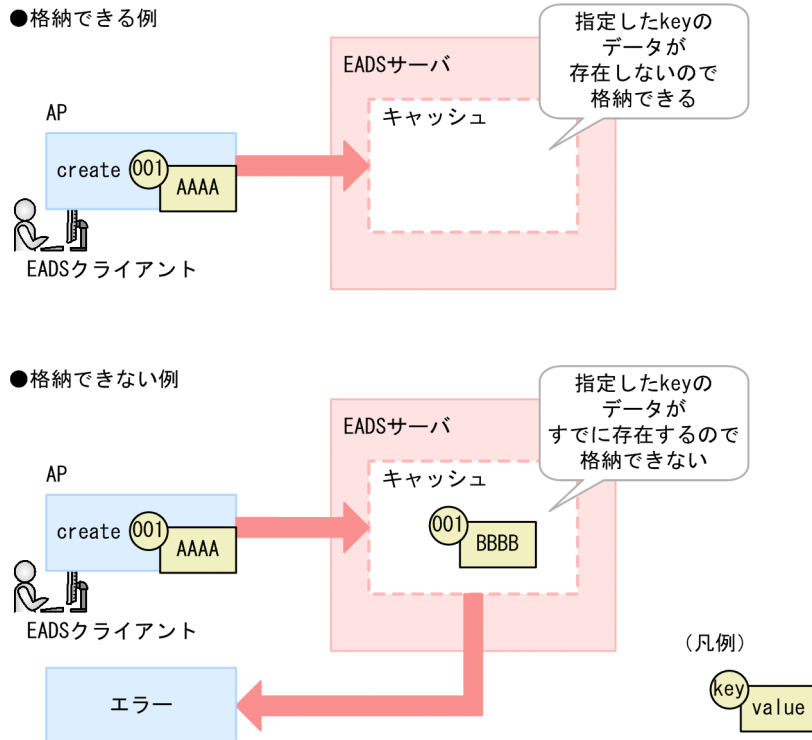
## 2.4.2 データの新規格納 (create)

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



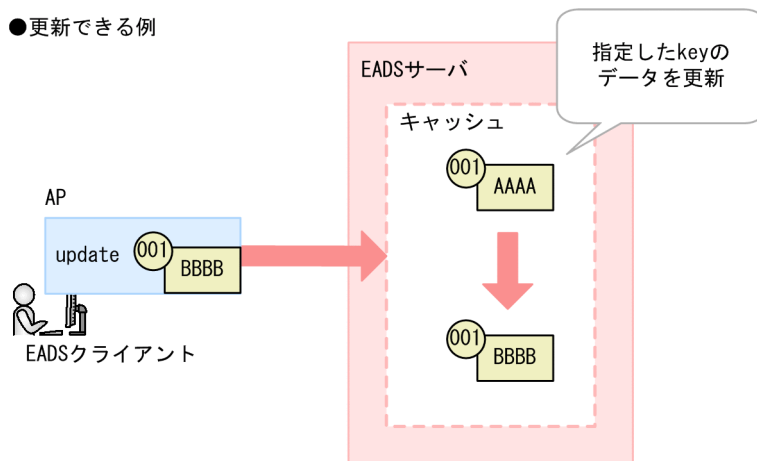
## 2.4.3 データの更新 (update)

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。

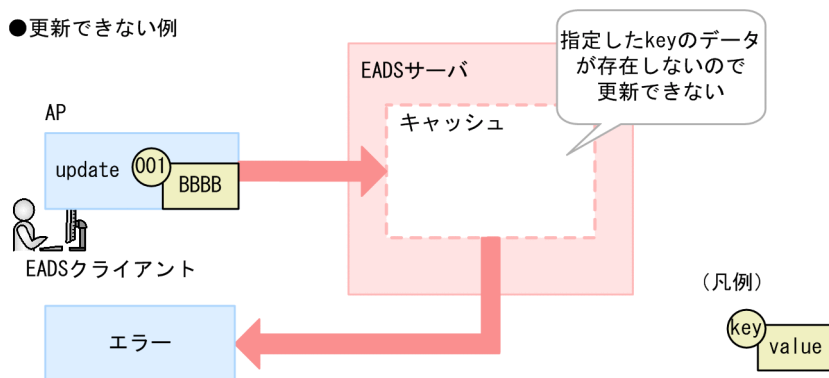
update は、指定した key がすでにキャッシュに格納されている場合だけ、value を key に関連づけて更新 (update) します。そのため、指定した key がキャッシュに格納されていない場合はエラーとなります。

## 図 2-8 データの更新 (update) の概要

### ●更新できる例



### ●更新できない例



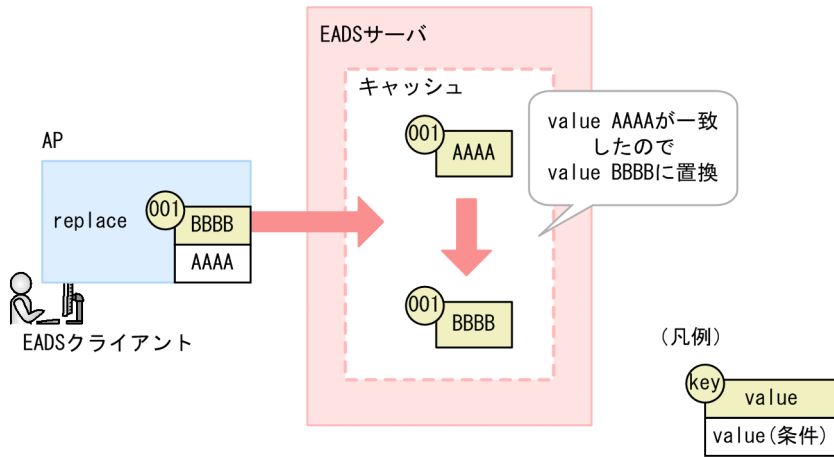
## 2.4.4 データの置換 (replace)

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

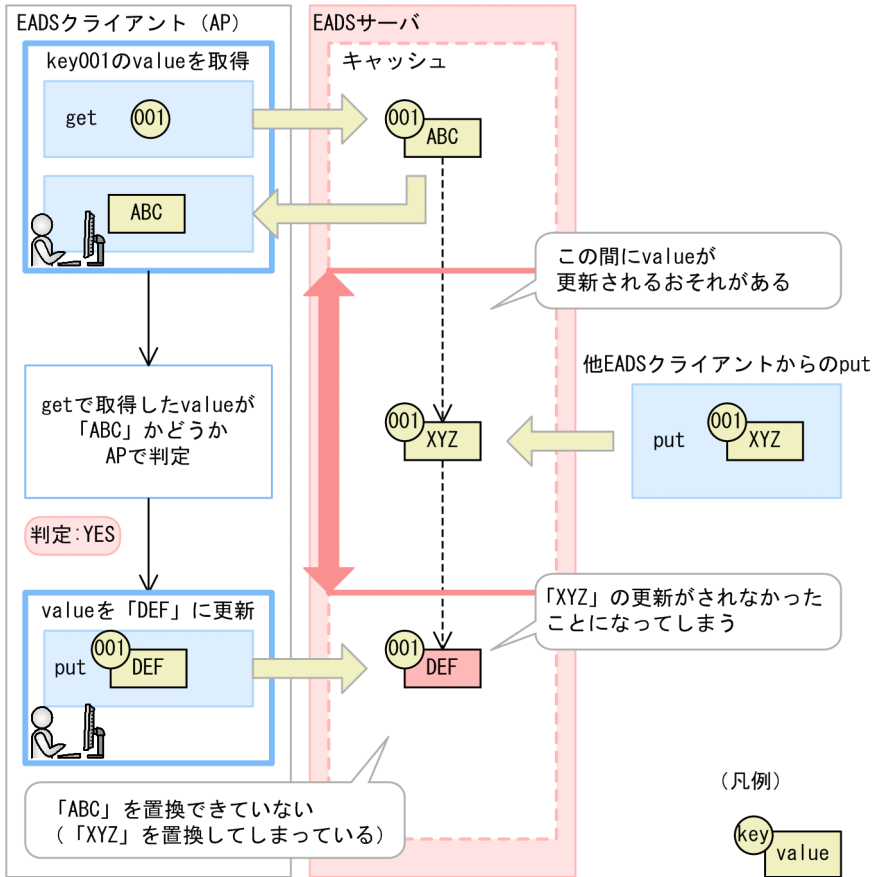
### put と replace の違い

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

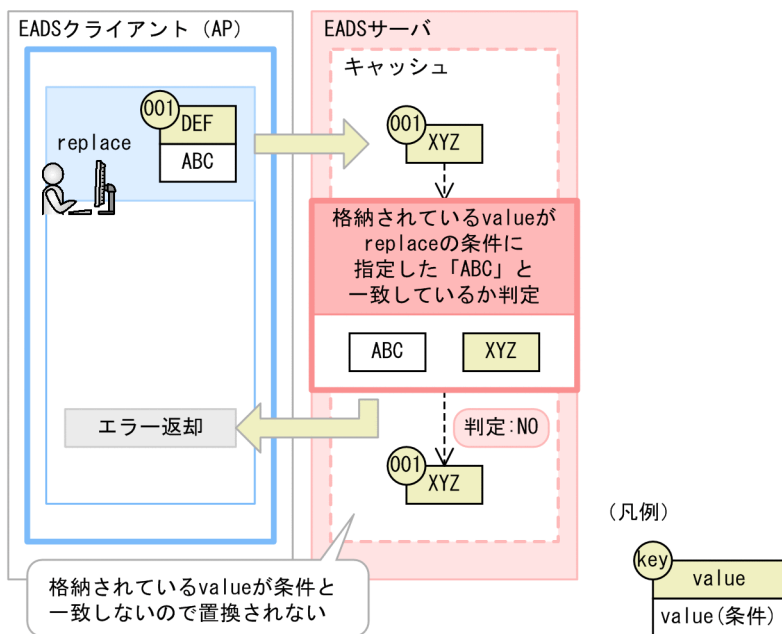
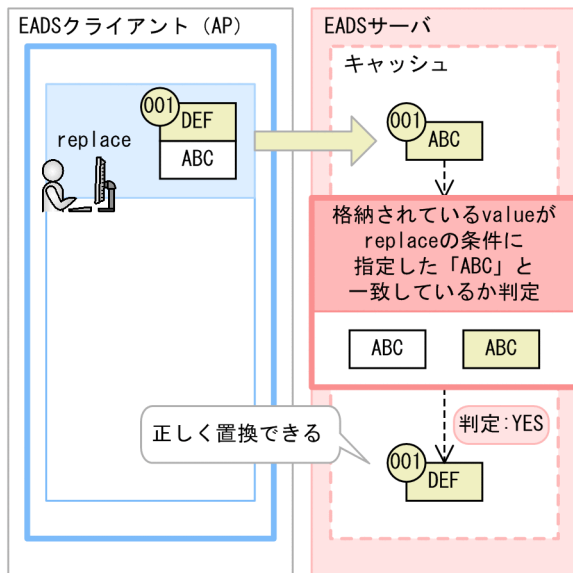
●getとputによる置換



(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。

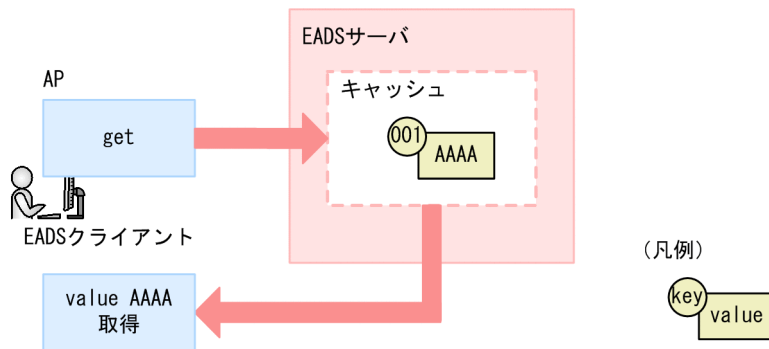
## 2.4.5 データの取得 (get)

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



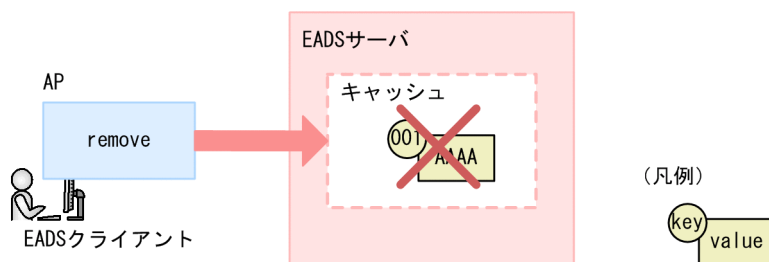
## 2.4.6 データの削除 (remove)

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

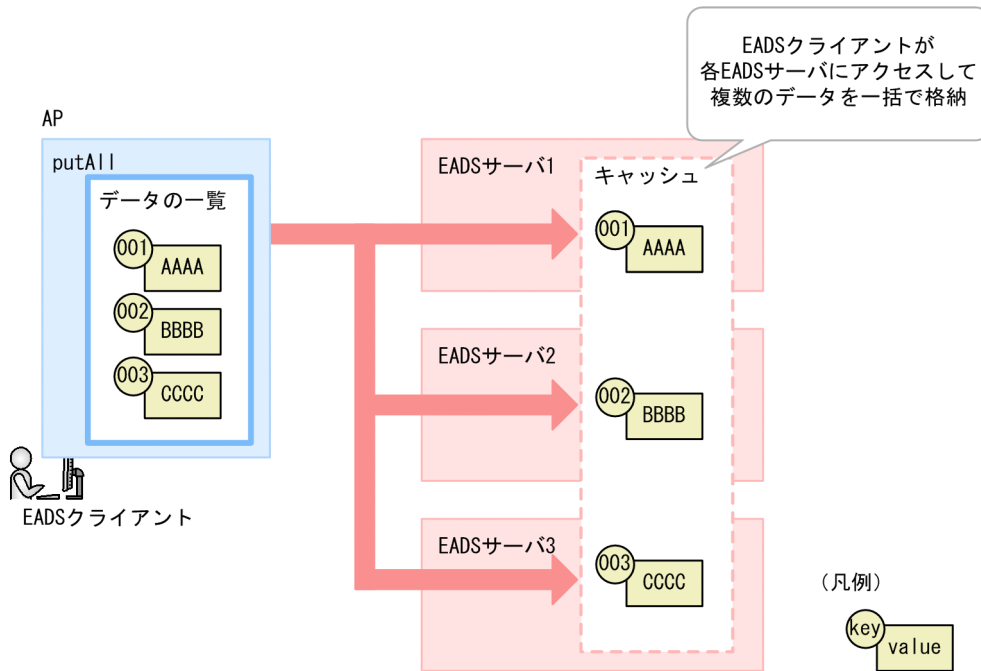


## 2.4.7 キャッシュに対する一括操作

キャッシュ内の複数のデータを一括で操作します。一括操作には次の種類があります。

- データの一括格納 (putAll)
- データの一括取得 (getAll)
- データの一括削除 (removeAll)

図 2-13 キャッシュに対する一括操作の概要 (putAll の例)



一括操作の対象となるデータは、次の方法で指定できます。

- 複数の key またはデータを一括で指定する
- key が属しているグループ名を指定する
- データの格納先の EADS サーバを指定する

グループ名 (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。

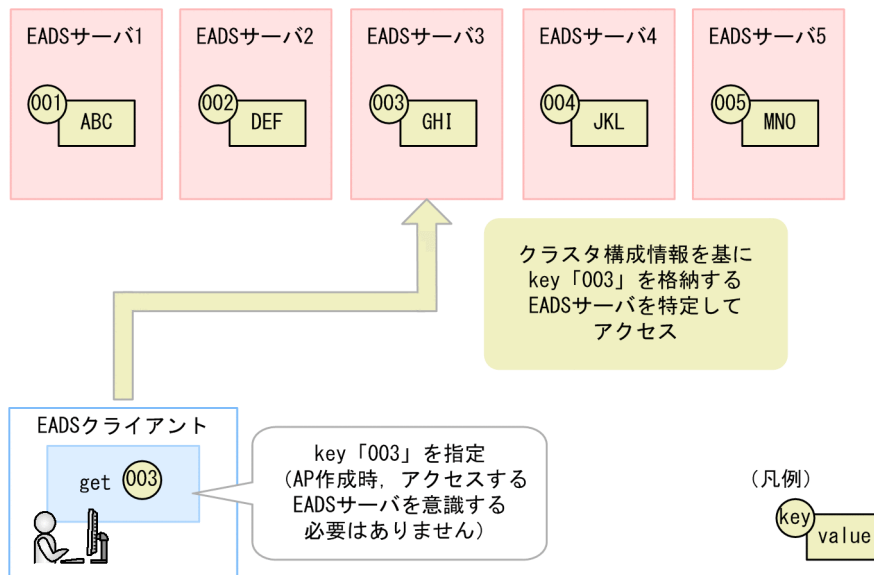
## 2.4.8 EADS サーバの位置を意識しないデータアクセス

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 2-14 図 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。

## 2.4.9 データアクセスの流れ

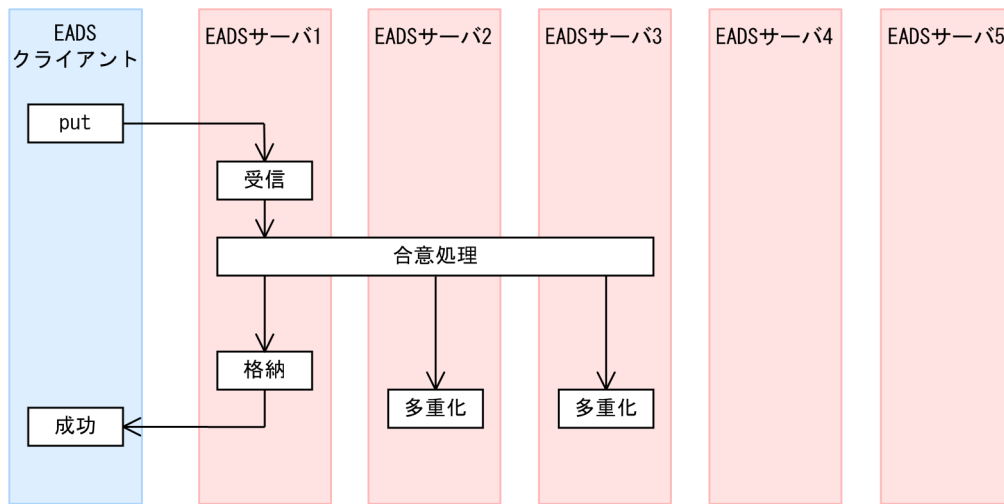
データアクセスの流れについて説明します。

### (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 2-15 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。

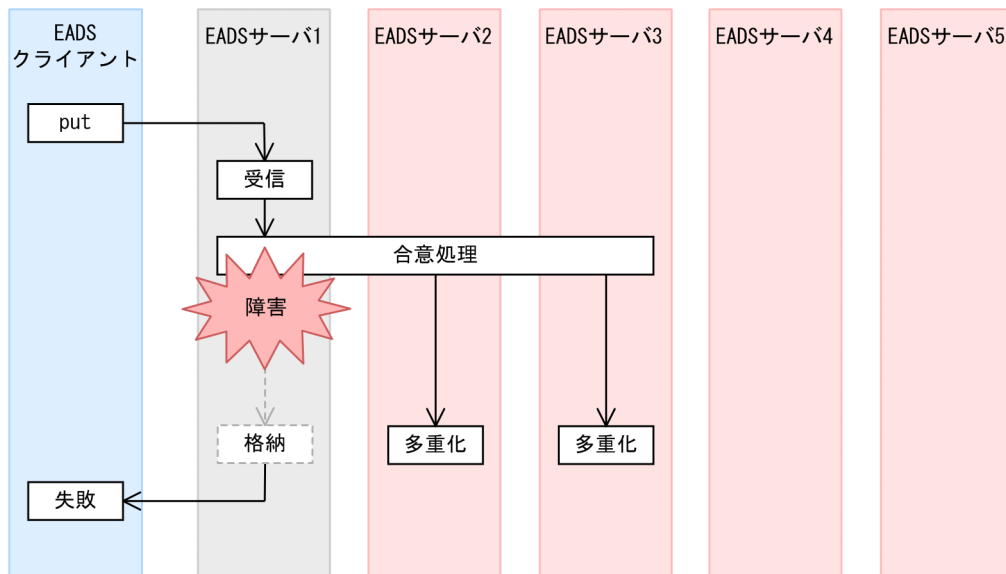
- 接続先 EADS サーバやネットワークに障害が発生している。
- key と value のうち、value 部分を格納する領域が不足している。\*
- 1EADS サーバに格納できるデータ数を超過している。\*

注※

データ総量監視機能を有効にすると、データの格納先の空き容量が不足することが予想される場合に、その処理をエラーとすることで、EADS サーバが縮退することを防ぐことができます。このため、データ総量監視機能は有効にすることを推奨します。

## (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。

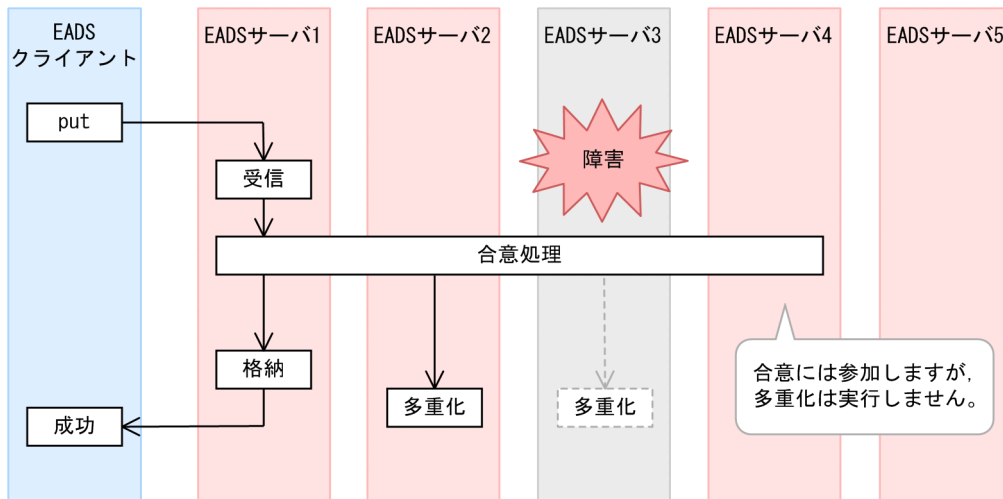
データの格納に失敗した EADS サーバは縮退状態<sup>※</sup>に遷移しますが、クラスタは継続します。この状態では、データの多重度が下がったままです。データの多重度を元に戻すには、縮退状態に遷移した EADS サーバの問題を取り除いてから、復旧を行います。

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、データの複製先である 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、データの複製先 EADS サーバにアクセスしてデータを参照します。

### 2.4.10 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, putAll, create, update, replace, remove, removeAll)
- eztool put コマンド
- eztool remove コマンド
- eztool removeall 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

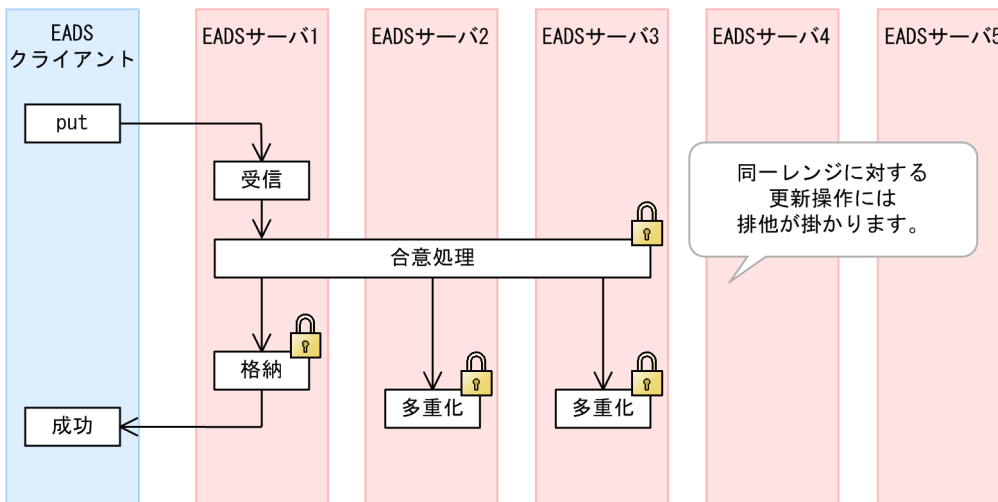
データの参照操作とは、次の API またはコマンドのことです。

- API (get, getAll)
- 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 2-16 排他範囲



### 2.4.11 データ自動削除機能

データ自動削除機能は、指定したしきい値を超えるデータがレンジに登録されないように、指定されたポリシーに従って優先度の低いデータを自動で削除する機能です。

例えば、EADS をリードキャッシュとして適用する運用などで、EADS クライアントから使用なくなったデータを削除することなく、新しいデータを登録できます。

データ総量監視機能が有効な場合に、データ自動削除機能を使用できます。

また、メモリキャッシュまたはディスクキャッシュの場合に、データ自動削除機能を使用できます。2Way キャッシュの場合には使用できません。

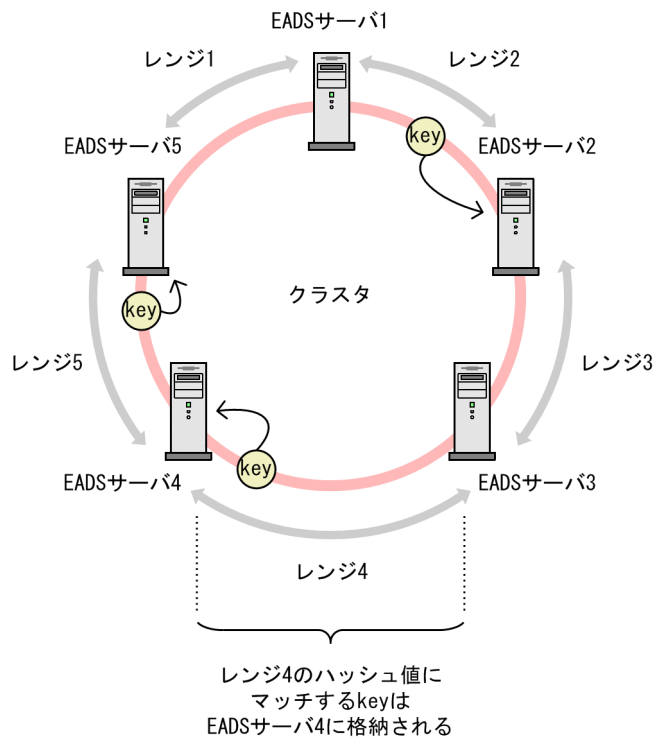
## 2.5 コンシステント・ハッシングによるデータの分散配置

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 2-17 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、連番の整数が反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。

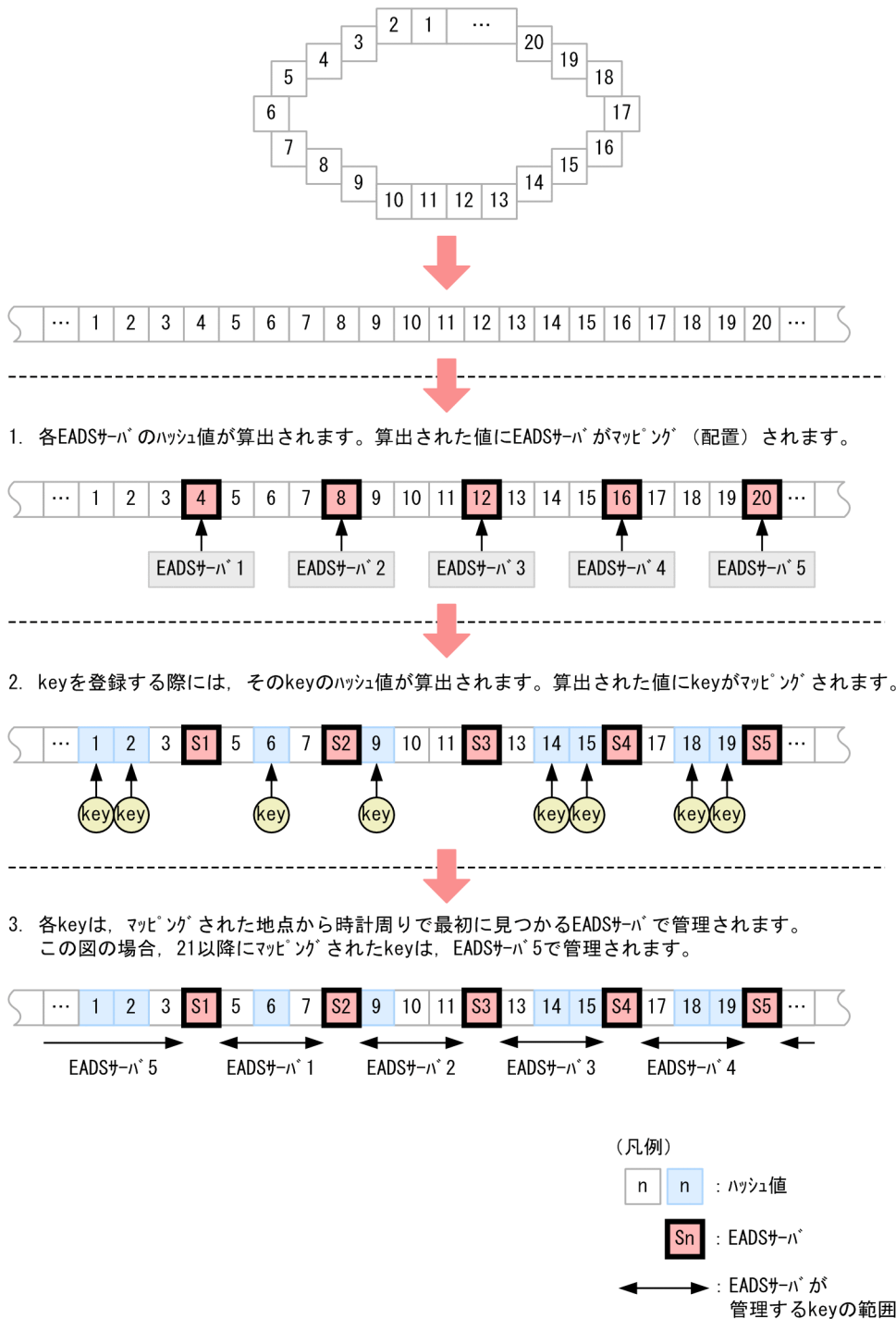
また、あるキャッシュのコンシステント・ハッシングの円周上を、EADS サーバの位置によって区切ったハッシュ値の範囲をレンジといいます。レンジはレンジ ID で管理されます。この図の場合、レンジ 4 に配置された key は、EADS サーバ 4 に格納されます。

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。





## 2.5.3 EADS サーバの追加とデータの再配置（スケールアウトとリバランス）

EADS では、クラスタを停止させないで、新しい EADS サーバを追加し、クラスタ内のデータを再配置できます。

クラスタを停止させないで、新しい EADS サーバをクラスタに追加する処理を、スケールアウト処理といいます。追加する EADS サーバは、コンシステント・ハッシングの円周上の任意の位置に追加できます。

スケールアウト処理を実行して EADS サーバを追加すると、新しい EADS サーバを追加したレンジが分割されます。そのため、各レンジの大きさ（各レンジに格納できる key の数）に偏りが生じるおそれがあります。このような場合に、EADS サーバの位置を変更して、各レンジに格納できる key の数を均等にすることで、クラスタ内のデータを配置し直すことができます。

クラスタを停止させないで、クラスタ内のデータを再配置する処理を、リバランス処理といいます。

各 EADS サーバの容量を均等に近づけるために、スケールアウト処理と併せてリバランス処理を実行することを推奨します。

### (1) EADS サーバの追加（スケールアウト）

スケールアウト処理では、クラスタを停止させないで、EADS サーバを追加できます。

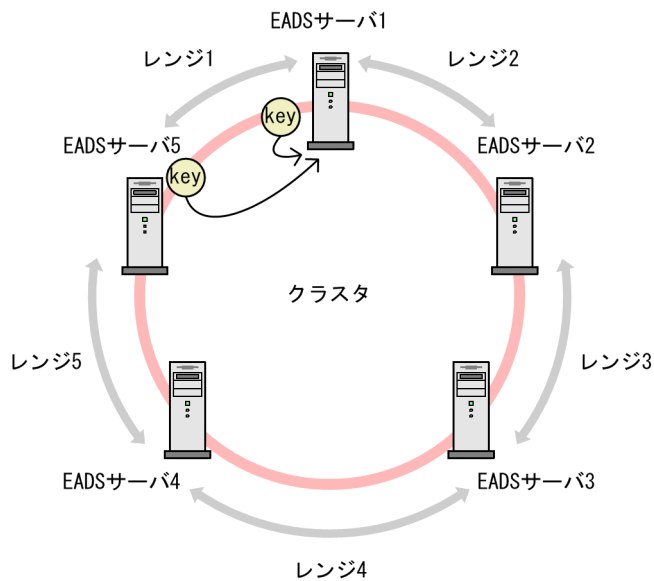
スケールアウト処理を実行すると、新しい EADS サーバを追加したレンジが分割されます。追加された EADS サーバの位置（ハッシュ値）よりも大きな値の範囲が、新しいレンジとなります。また、分割前のレンジを管理していた EADS サーバから、追加されたレンジの範囲のデータが、追加された EADS サーバに転送されます。

新しい EADS サーバをクラスタに追加することで、クラスタ全体の物理的なメモリの総容量を増加させたり、クラスタ全体の処理性能を向上させたりすることができます。また、レンジを分割することで、分割前のレンジを管理していた EADS サーバのメモリ使用量を低減したり、負荷を軽減したりすることができます。

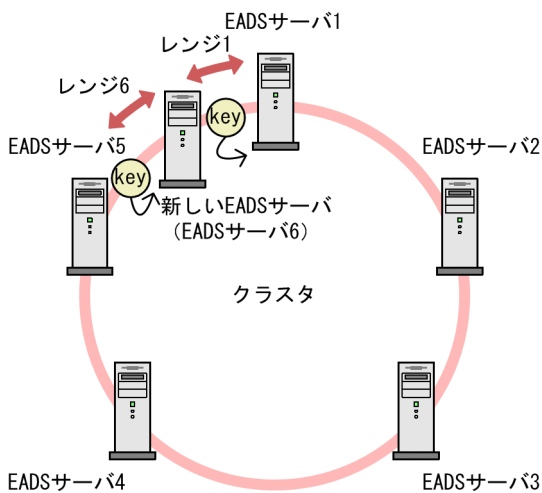
クラスタに新しい EADS サーバを 1 台追加したときの概要を次の図に示します。

図 2-18 スケールアウト処理 (EADS サーバの追加) の概要

■新しいEADSサーバを追加する前



■新しいEADSサーバを追加したあと (スケールアウト処理後)



なお、スケールアウト処理を実行すると、各レンジに格納できる key の数が均等に配分されなくなるおそれがあります。スケールアウト処理後は、リバランス処理を実行して、各レンジに格納できる key の数を均等に配分し直すことを推奨します。

## (2) クラスタ内のデータの再配置 (リバランス)

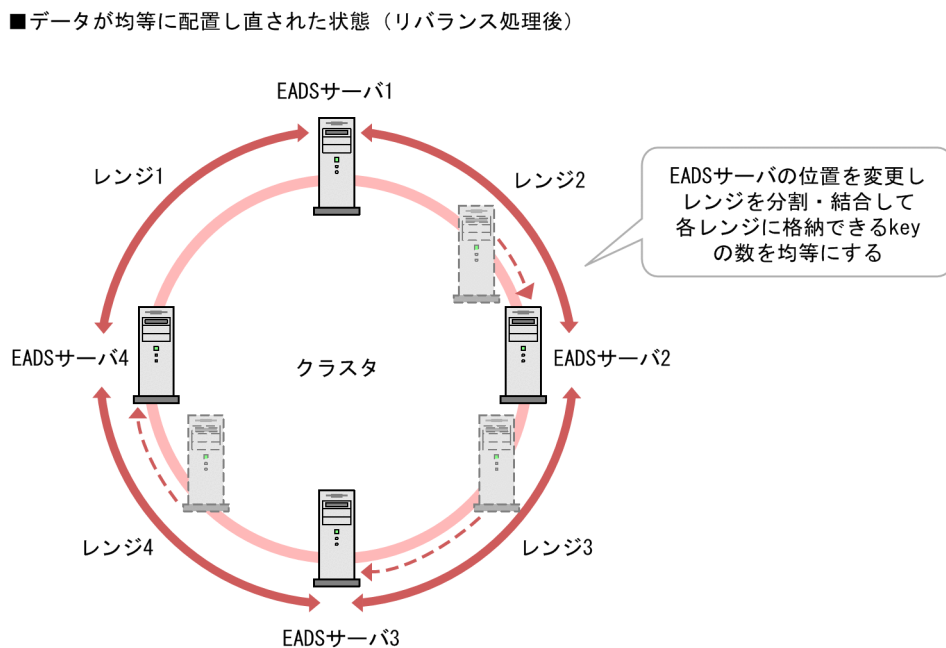
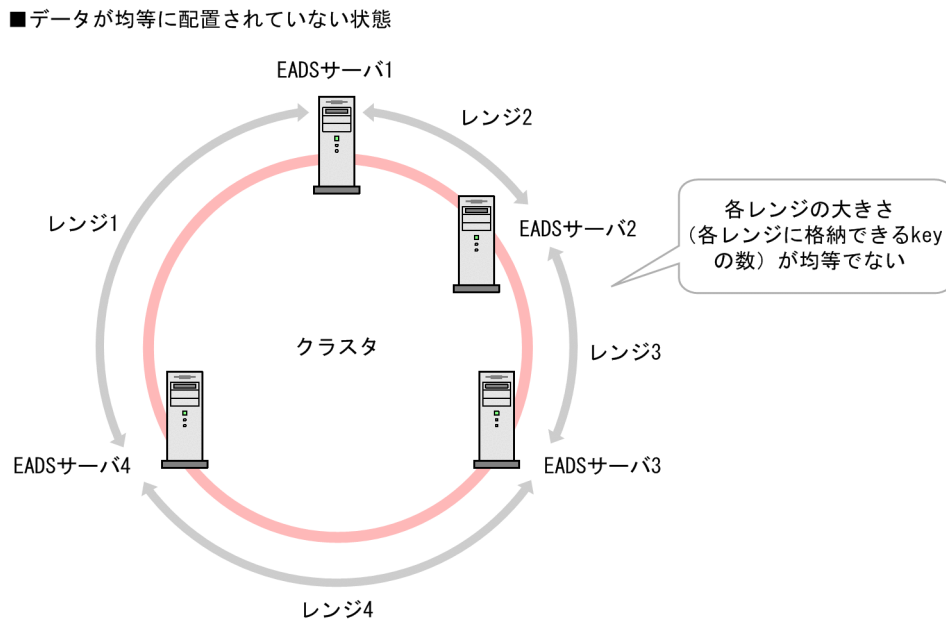
リバランス処理では、クラスタを停止させないで、クラスタ内の EADS サーバの位置を変更します。EADS サーバの位置を変更することで、クラスタ内のデータを配置し直します。

リバランス処理をすることによって、各 EADS サーバが保持するデータの容量やデータの数、および各レンジに格納できる key の数を均等に近づけることができます。

リバランス処理は、スケールアウト処理後など、各レンジに格納できる key の数が均等に配分されていない場合に実行します。

リバランス処理の概要を次の図に示します。

図 2-19 リバランス処理（クラスタ内のデータの再配置）の概要



リバランス処理を実行するには、任意の EADS サーバ上で `eztool rebalance` コマンドを実行します。

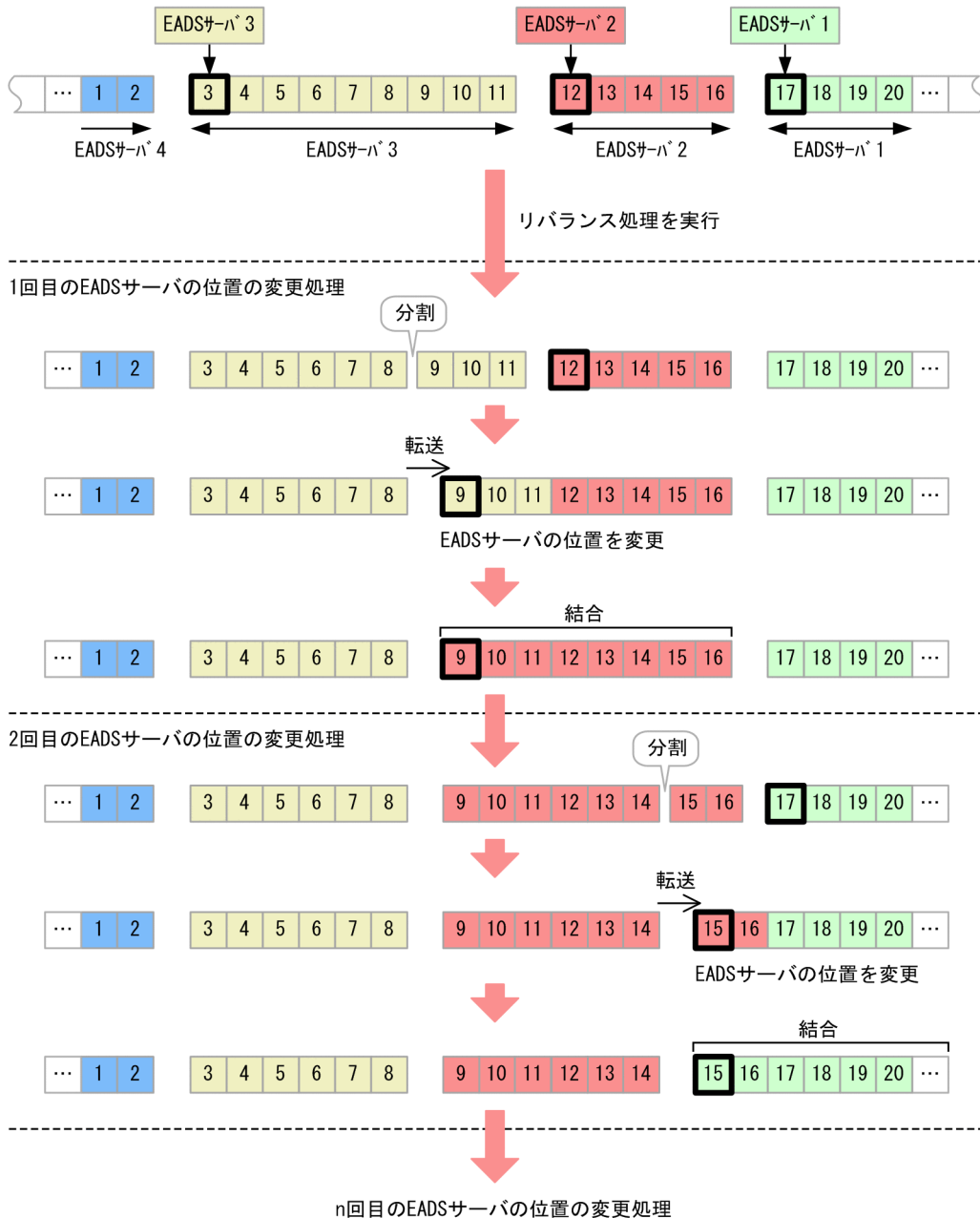
リバランス処理を実行すると、まず起点となる EADS サーバが 1 つ決められます。その後、その EADS サーバを起点に、ほかの EADS サーバをコンシステント・ハッシングの円周上に均等に配置し直します (各レンジに格納できる key の数が均等になるように EADS サーバの位置を変更します)。

各レンジの大きさ（各レンジに格納できる key の数）は、各 EADS サーバが管理するレンジを分割および結合して均等にされます。EADS サーバが新しく管理するレンジのデータは、元々そのレンジを管理していた EADS サーバ（分割前のレンジを管理していた EADS サーバ）から転送されます。

リバランス処理の流れを次の図に示します。なお、コンシステント・ハッシングでは、EADS 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。

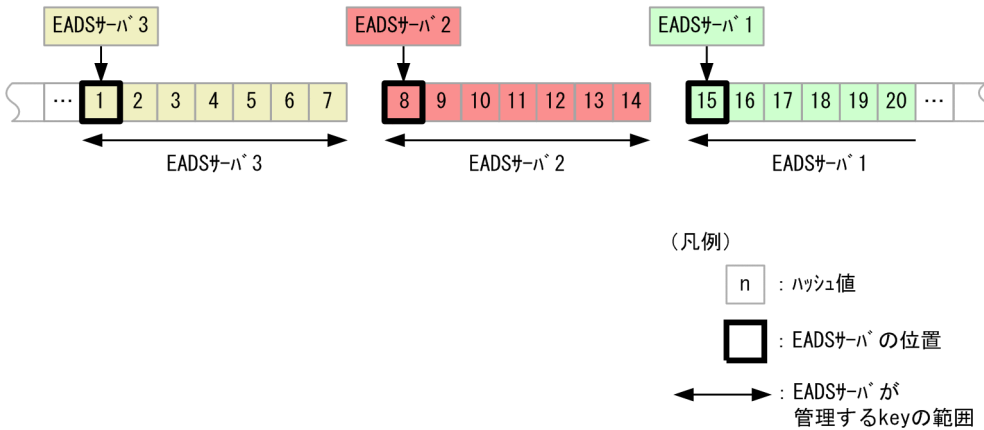
図 2-20 リバランス処理の流れ

各レンジの大きさ（各レンジに格納できるkeyの数）に偏りがある状態





リバランス処理が完了して、各レンジの大きさの偏りが解消された状態



## 参考

### EADS サーバの位置の変更処理

- EADS サーバの位置の変更処理を実行することで、各レンジの大きさ（各レンジに格納できるkeyの数）を均等にします。
- EADS サーバの位置の変更が大きい場合、1回で位置を変更しないで、複数回に分けて位置を変更します。
- 各 EADS サーバは、コンシステント・ハッシングの円周上をハッシュ値が小さくなる方に移動します。
- 1回の処理では、1つの EADS サーバの位置が変更されます。EADS サーバの位置の変更処理は、複数の EADS サーバに対して同時に実行されません。

### レンジの分割および結合処理

- 位置を変更する EADS サーバが管理しているレンジに隣接するレンジが分割されます。分割されたレンジは、位置を変更する EADS サーバが管理しているレンジに結合されます。

## ポイント

スケールアウト処理で増えたクラスタのメモリ容量を効果的に使用するため、リバランス処理が完了してからクラスタに格納するデータ量を増やすことを推奨します。

## 2.6 特定の EADS サーバへのデータ配置 (key のグルーピング)

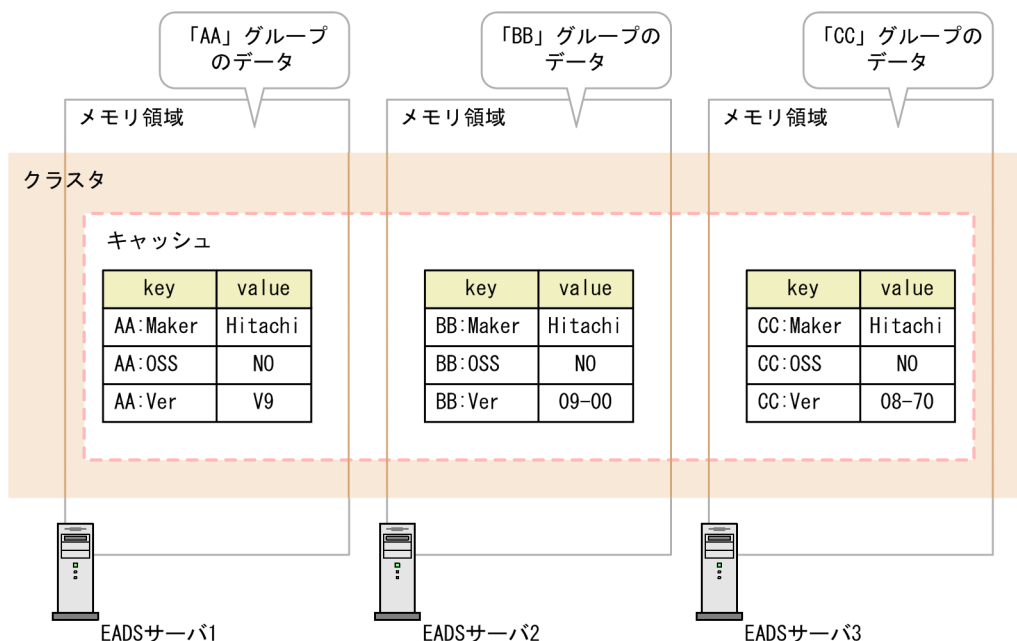
EADS では、関連するデータを同じ EADS サーバに配置するために、あらかじめ、複数の key を 1 つのグループとしてグルーピングしておくことができます。これを key のグルーピングといいます。

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でグループ単位の処理を実行できるようになり、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 2-21 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

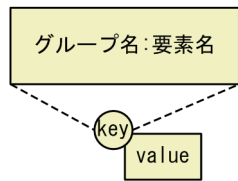
key にグループ名を指定することで、key をグルーピングできます。

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 2-22 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

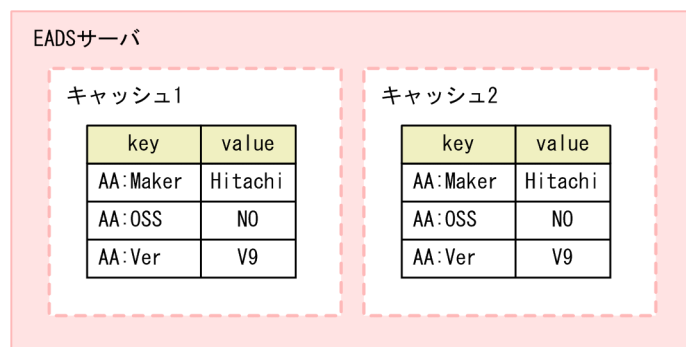
value を一意に識別するための名称です。

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、格納先のキャッシュが異なれば、グループ名と要素名の組み合わせが同じでも、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、複数のグループの名称を定義することで、グループを階層化できます。このとき、1つ1つのグループの名称をグループ階層名といいます。

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。

グループを階層化している場合、最上位階層のグループ階層名から、任意の階層のグループ階層名までをまとめたものがグループ名になります。例えば、「Group1:Group2:Group3:ElementA」の場合、グルー

階層名は「Group1」, 「Group2」 および 「Group3」, グループ名は 「Group1」, 「Group1:Group2」 および 「Group1:Group2:Group3」 となります。

## 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、先頭のグループ階層名のハッシュ値から格納先の EADS サーバが決まります。先頭のグループ階層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。

### ポイント

格納先の EADS サーバの EADS サーバ ID を指定して、key をグルーピングすることもできます。このようなグループを、EADS サーバ ID 指定グループといいます。EADS サーバ ID 指定グループを使用して key をグルーピングする場合は、key の先頭（最上位階層のグループ階層名の先頭）に、EADS サーバ ID を、左角括弧 ( [ ) および右角括弧 ( ] ) で囲んで指定します。

例えば、次に示す key のデータは、グループ 「[1]Group1」 に属し、EADS サーバ ID が 1 の EADS サーバに格納されます（「Group1」などの EADS サーバ ID のあとに続く文字列は考慮されなくて、指定した EADS サーバ ID だけで格納先の EADS サーバが決定されます）。

- 「[1]Group1:Group2:ElementA」
- 「[1]Group1:Group3:ElementB」
- 「[1]Group1:ElementC」



## 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。

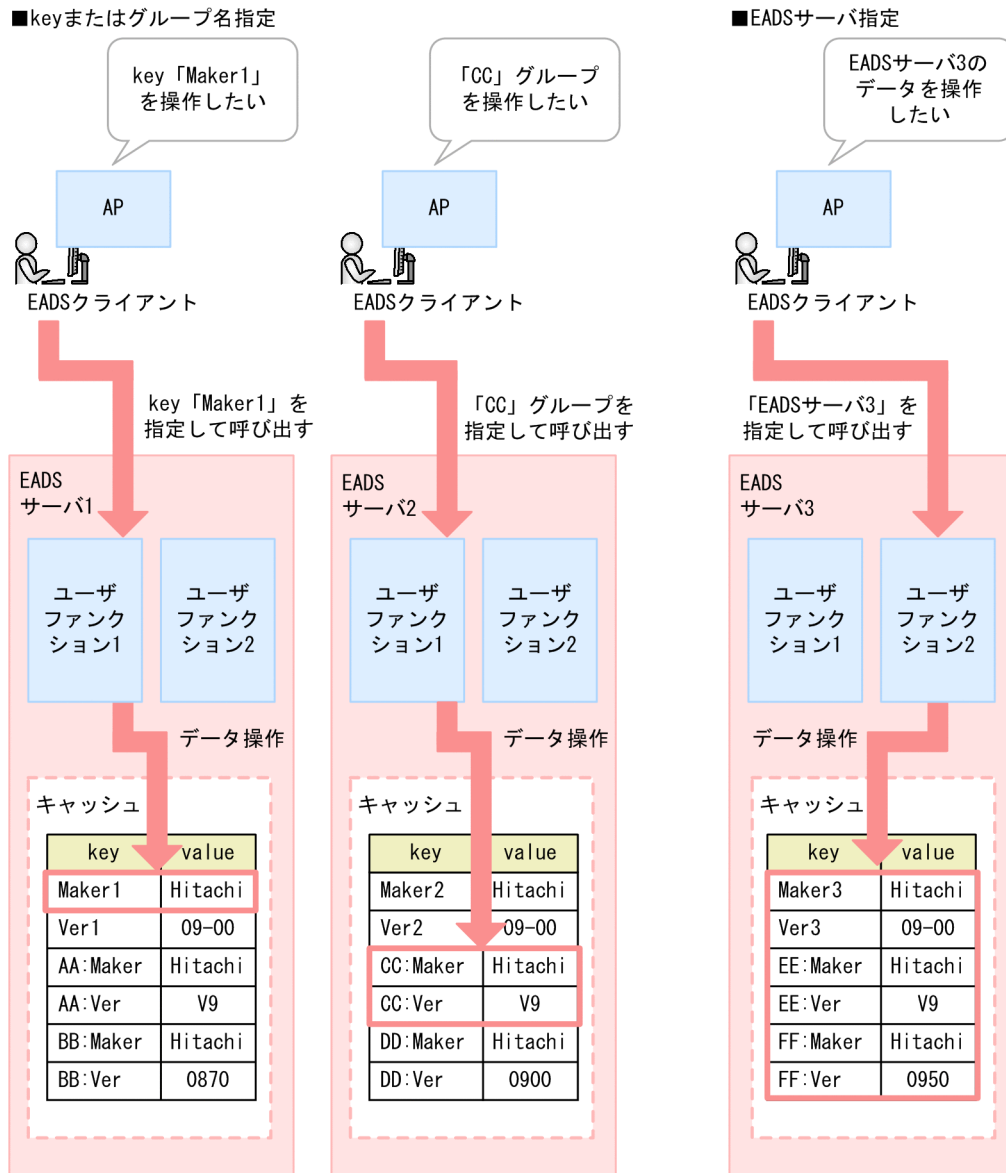
ユーザファンクションは、Java 言語で作成します。

### 2.7.1 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。

- key または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 2-23 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) key またはグループ指定による実行方法

EADSクライアントからkeyまたはグループ名、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keyまたは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。

## (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。

## 2.8 イベントリスナによるイベントの発生を契機にしたデータ処理

---

イベントリスナとは、EADS サーバ上でのイベントの発生（クラスタ構成情報の更新）を契機に、ユーザが定義した処理を、EADS クライアント上で実行させるプログラムです。

イベントリスナは、ユーザが作成します。作成したイベントリスナを、あらかじめ EADS クライアント上に配置しておきます。EADS サーバの追加などのイベントが発生すると、EADS サーバから EADS クライアントに対してクラスタ構成情報の更新が通知されて、EADS クライアント上でイベントリスナが実行されます。

イベントリスナを利用することで、クラスタ構成情報の更新と同期したユーザ処理が実行しやすくなります。

イベントリスナは、Java 言語または C 言語で作成します。

### 2.8.1 イベントリスナの仕組み

イベントリスナには、次の 2 種類があります。

#### クラスタイベントリスナ

EADS サーバがクラスタに追加された場合に実行させるイベントリスナです。

#### ノードイベントリスナ

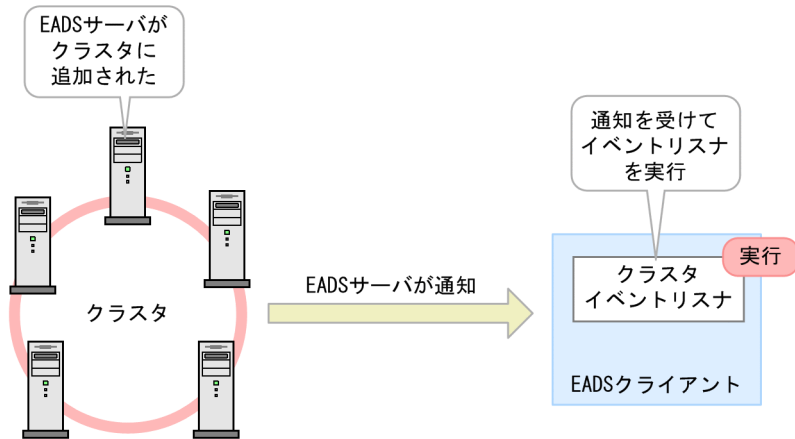
EADS サーバの縮退や復旧など、クラスタを構成する EADS サーバの状態が変わった場合に実行させるイベントリスナです。

ノードイベントリスナの実行の契機は、ノードイベントリスナを追加するときに指定できます。

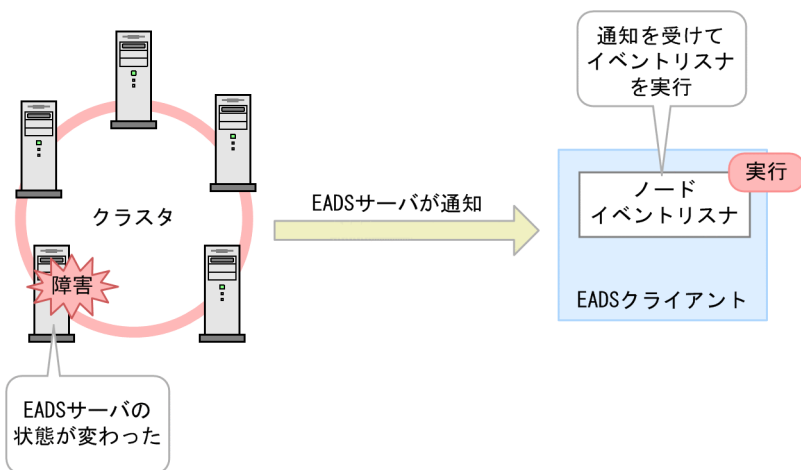
イベントリスナの仕組みを次の図に示します。

## 図 2-24 イベントリスナの仕組み

### ■クラスタイイベントリスナ



### ■ノードイベントリスナ



クラスタイイベントリスナを使用する場合、EADSサーバがクラスタに追加されると、各EADSサーバからEADSクライアントに対してクラスタ構成情報の更新が通知されます。EADSクライアントは、EADSサーバからの通知を受けて、クラスタイイベントリスナを実行します。

ノードイベントリスナを使用する場合、EADSサーバの縮退や復旧など、EADSサーバの状態が変わると、各EADSサーバからEADSクライアントに対してクラスタ構成情報の更新が通知されます。EADSクライアントは、EADSサーバからの通知を受けて、ノードイベントリスナを実行します。

### 参考

- EADSサーバとEADSクライアントで通信するため、イベントリスナ用の接続が1つ用意されます。この接続をクラスタイイベントリスナ、およびノードイベントリスナで共用します。

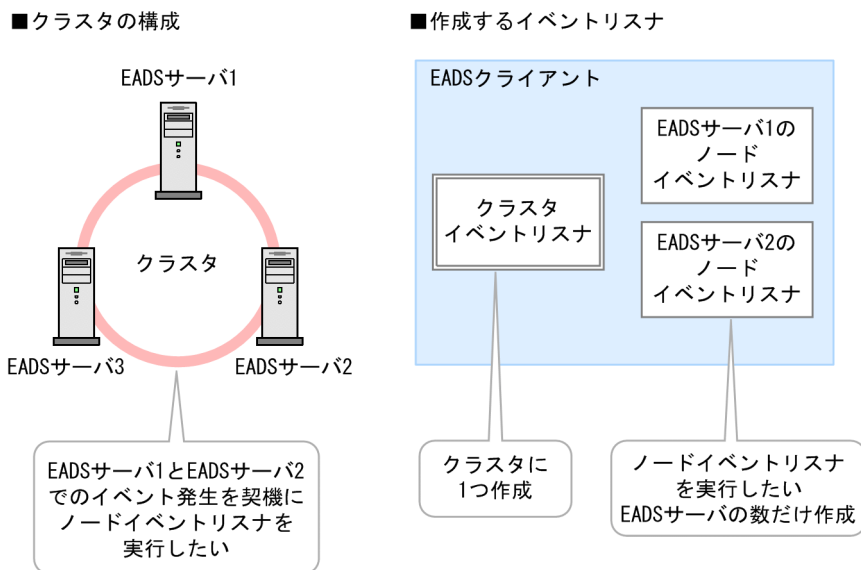
- クラスタ内の EADS サーバのうち、通信できるすべての EADS サーバからクラスタ構成情報の変更が通知されます。なお、複数の EADS サーバから同じクラスタ構成情報の変更が通知されても、ノードイベントリスナが実行されるのは 1 回だけです。

イベントリスナを使用する例を次に示します。

- クラスタに 1 つ、クラスタイベントリスナを作成する。
- ノードイベントリスナを実行したい EADS サーバごとに、ノードイベントリスナを作成する。

作成したイベントリスナは、API を使用して EADS クライアント上に配置します。

## 図 2-25 イベントリスナの配置の例



## ポイント

イベントリスナは次のように動作させることを推奨します。

- クラスタイベントリスナ、およびノードイベントリスナの両方を使用する。
- クラスタイベントリスナでは、EADS サーバの追加が通知されたら、追加された EADS サーバ用のノードイベントリスナを生成する。
- ノードイベントリスナでは、ノードイベントリスナの実行の契機となるすべてのイベントを処理する。

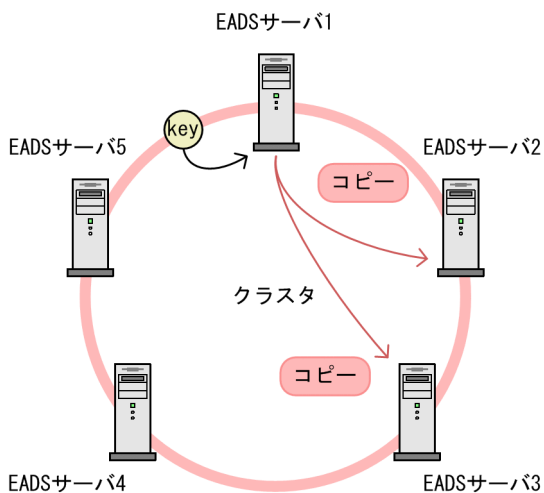
## 2.9 データの多重化

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 2-26 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、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 から時計回りの方向にある EADS サーバ2 および EADS サーバ3 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。







## 2.10.2 一定数の同意によって確定する EADS サーバダウン

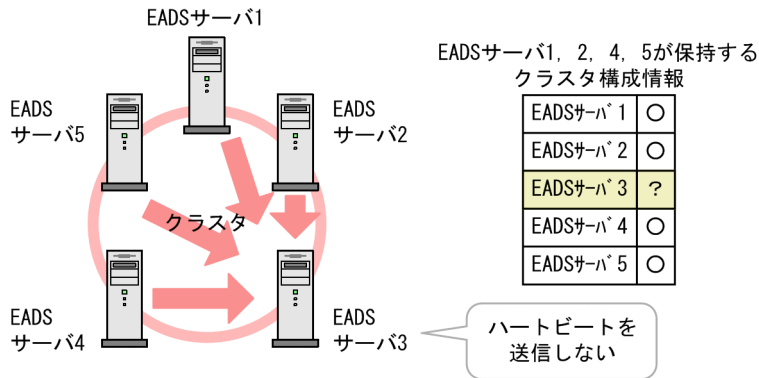
あらかじめ設定したタイムアウト時間（デフォルトは 2 秒）が経過してもハートビートを送信しない EADS サーバがある場合、クラスタ内の EADS サーバはその EADS サーバに対して生存確認を行います。生存確認がタイムアウトすると、一定数（デフォルトは 1）の同意によって EADS サーバダウンを判定します。ダウンが確定すると、その EADS サーバはクラスタから削除されます。

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。

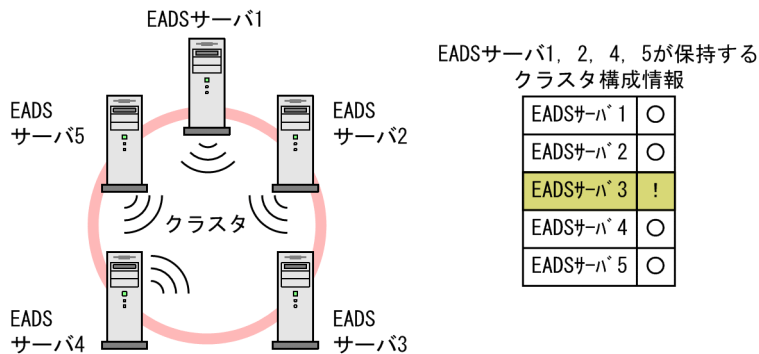
EADS サーバダウンを検知するまでの流れを、次の図に示します。

## 図 2-29 EADS サーバダウンを検知するまでの流れ

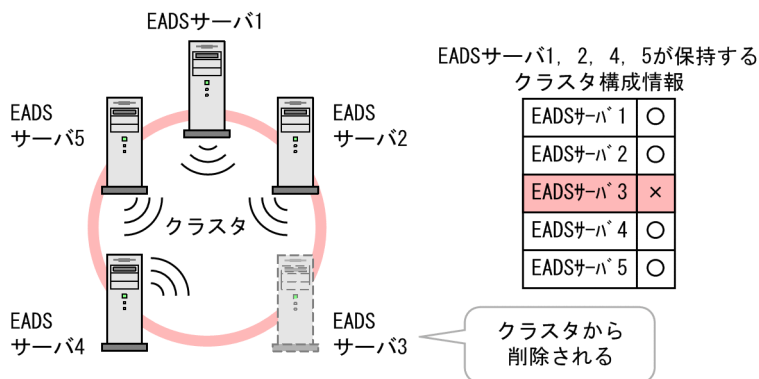
1. ハートビートを送信しないEADSサーバ3に対して生存確認を行う



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。

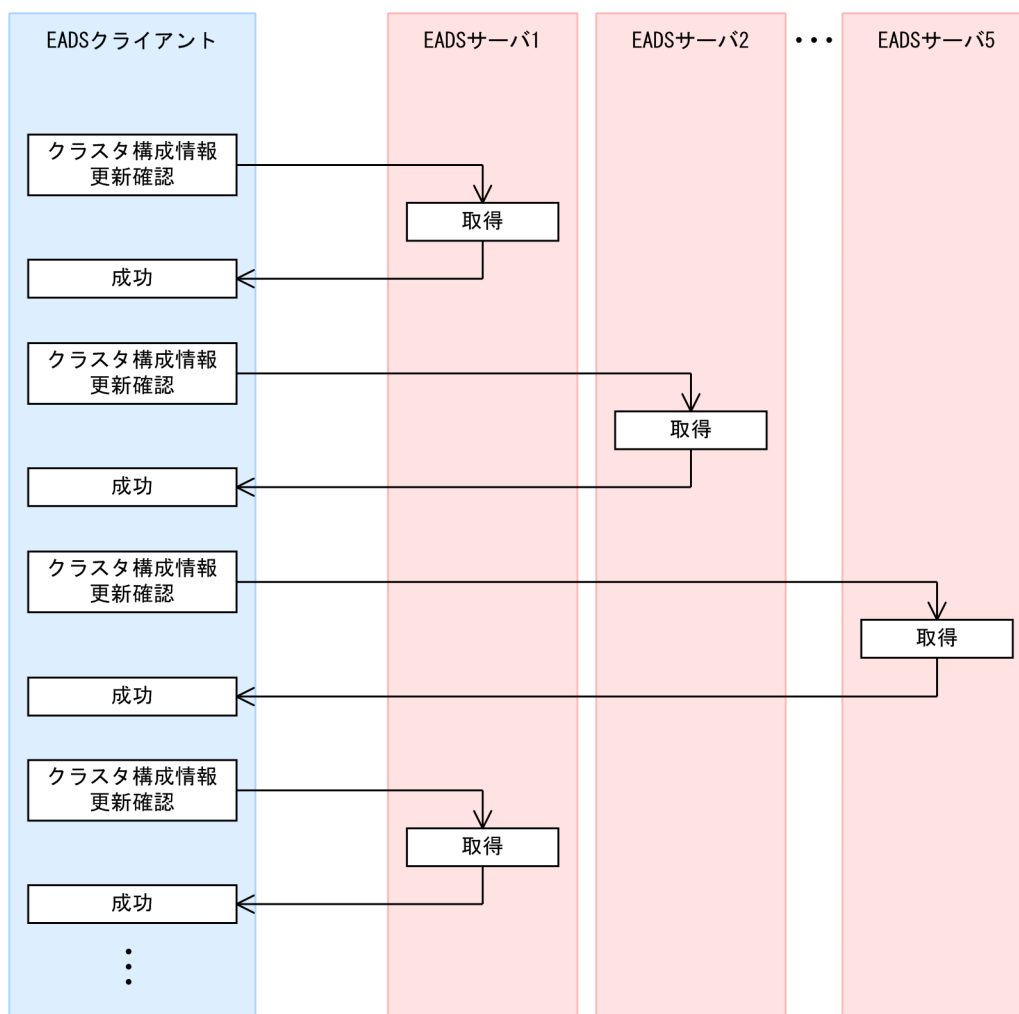
## 2.11 EADS クライアントによるクラスタ構成情報の更新確認

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 2-30 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。

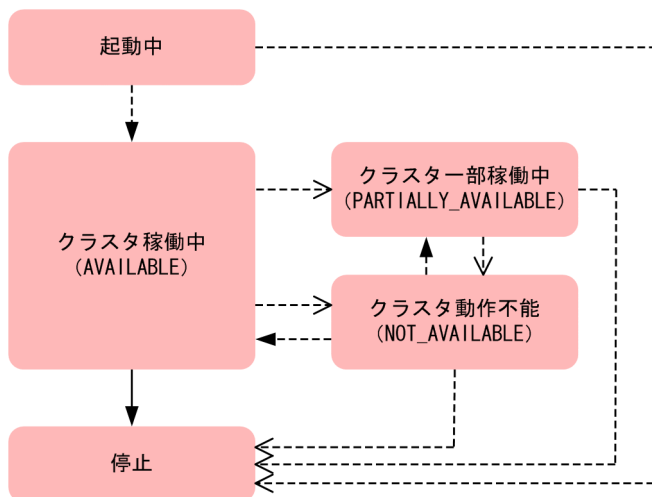
そのため、次の点に注意してください。

- 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 2-31 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

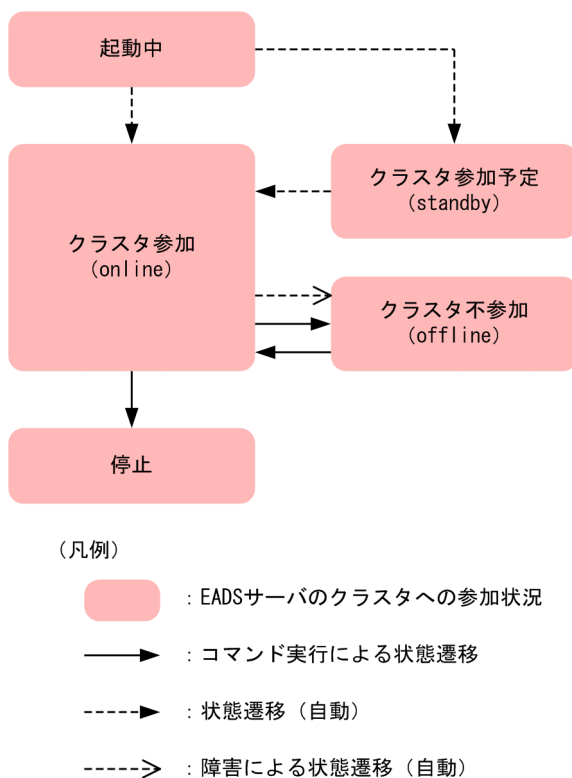
クラスタの状態		説明
状態名	eztool status コマンドでの表示内容	
クラスタ稼働中	AVAILABLE	クラスタが正常に稼働している状態です。

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 サーバのクラスタへの参加状況

EADS サーバのクラスタへの参加状況の遷移を次の図に示します。

図 2-32 クラスタへの参加状況の遷移



図中の各状態の説明を、次の表に示します。

表 2-5 EADS サーバのクラスタへの参加状況

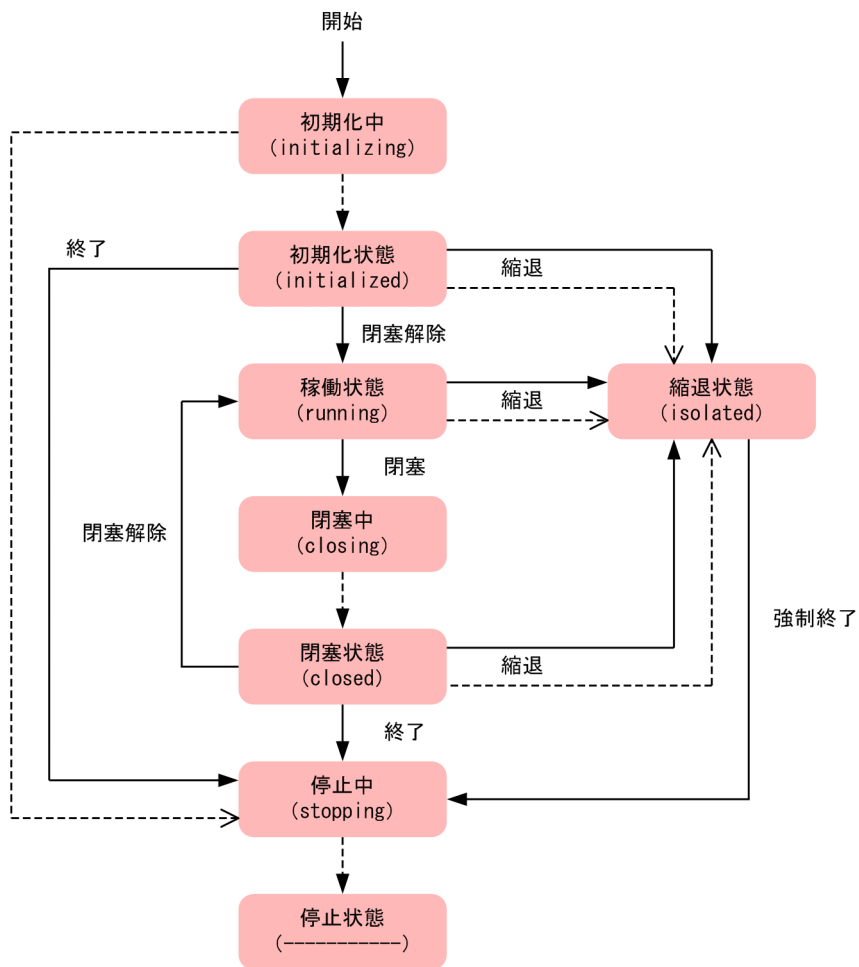
クラスタへの参加状況		説明
参加状況	eztool status コマンドでの表示内容	
クラスタ参加	online	EADS サーバが、稼働中のクラスタに参加しています。
クラスタ不参加	offline	EADS サーバが、稼働中のクラスタに参加していません。 通信障害などによって、クラスタから削除された EADS サーバはこの状況になります。 この状況の EADS サーバは、データアクセスおよびコマンドの実行対象にならない場合があります。 この状況になっている EADS サーバの状態を確認し、復旧させることで、EADS サーバがクラスタに参加します。
クラスタ参加予定	standby	EADS サーバが、稼働中のクラスタに一度も参加していません。 クラスタ定義に定義した EADS サーバを開始していない場合に、この状況になります。 この状況になっている EADS サーバを開始すると、EADS サーバがクラスタに参加します。

### 2.12.3 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。



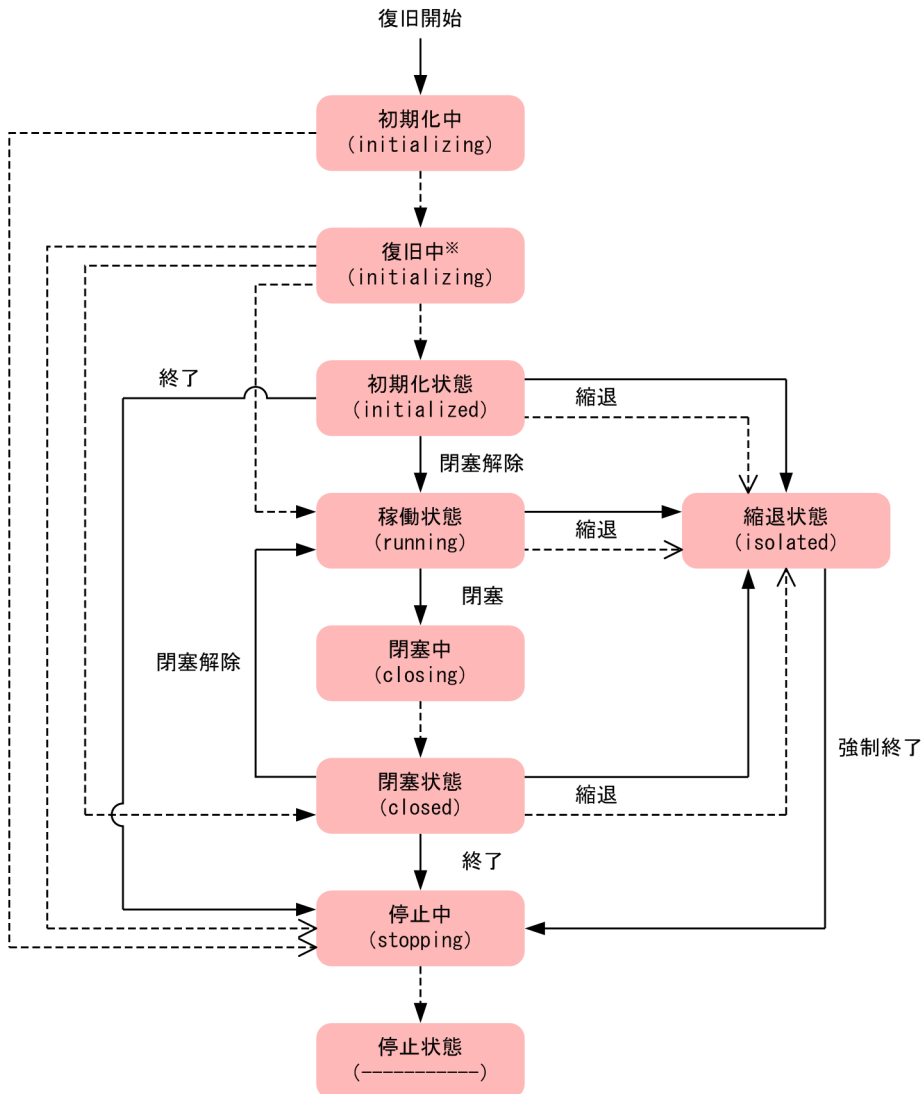
図 2-33 EADS サーバの状態遷移 (通常開始の場合)



(凡例)

- : EADSサーバの状態
- ▶ : 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 2-34 EADS サーバの状態遷移 (復旧処理の場合)



- (凡例)
- : EADSサーバの状態
  - : 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)

注※  
 復旧に成功すると、復旧開始時のクラスタ内の、すでに起動しているほかの EADS サーバの状態と同じ状態に遷移します。  
 例えば、復旧開始時に、すでに起動しているほかの EADS サーバの状態が running であれば、復旧後の EADS サーバの状態も running になります。



表 2-6 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。  ezstart -r コマンドなどによる EADS サーバの復旧中や、ezstart -ai コマンドなどによる EADS サーバの追加中にも表示されます。
復旧中		
追加中		
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。
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、eztool isolate --stop 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。

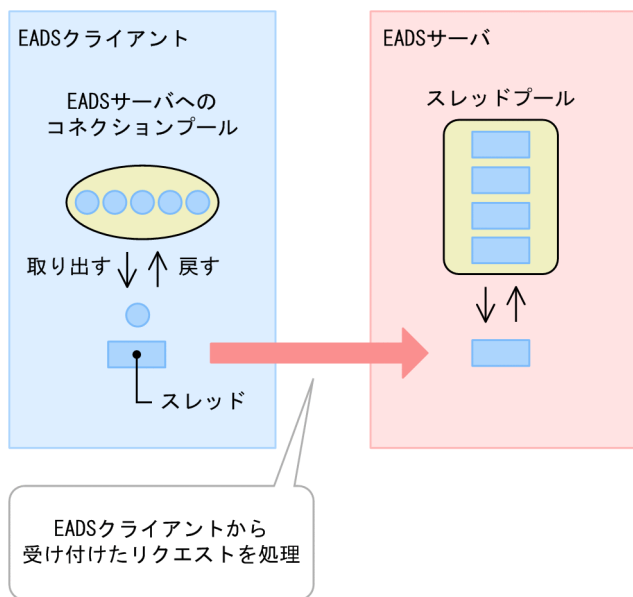
## 2.13 スレッドプールおよびコネクションプールを使用したスループットの向上

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 2-36 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、通信時のオーバヘッドを軽減できます。

## 2.14 ノンブロッキング I/O 通信制御

---

ノンブロッキング I/O 通信制御は、今までの通信制御（ブロッキング I/O 通信制御）と比べて、次に示すとおり接続数を拡張する機能です。

- 最大同時接続数を 1,024 から 32,768 に拡張できる

そのため、最大同時接続数を増やしたい場合に、ノンブロッキング I/O 通信制御を使用することを検討してください。

ただし、ノンブロッキング I/O 通信制御を使用することでリクエスト処理に関する処理数が増加します。そのため、最大同時接続数が 1,024 以下の場合にノンブロッキング I/O 通信制御を使用すると、処理速度が低下するおそれがあります。

最大同時接続数を 1,025 以上に拡張する必要がある場合に、ノンブロッキング I/O 通信制御を使用してください。

# 3

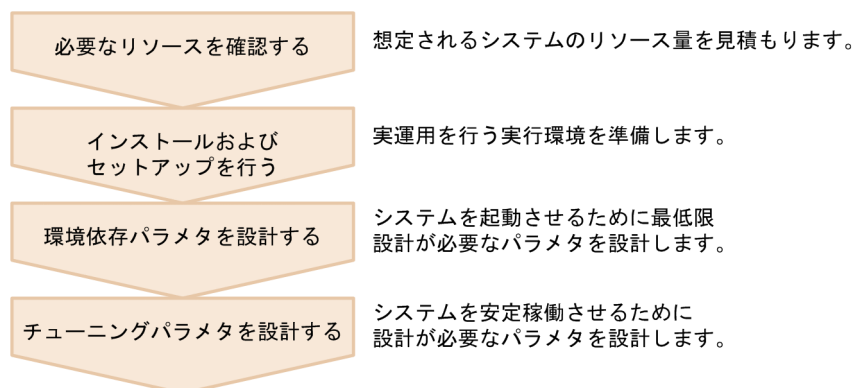
## システム設計・構築の流れ

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。

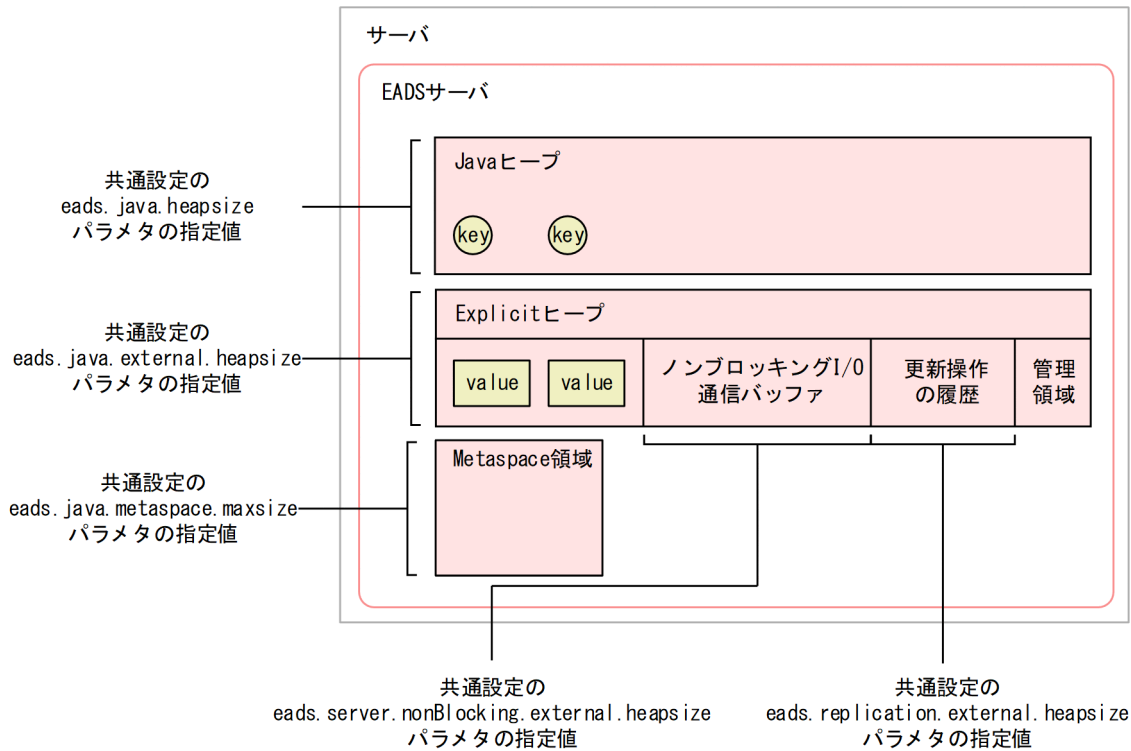
## 4.1 メモリ使用量を見積もる

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

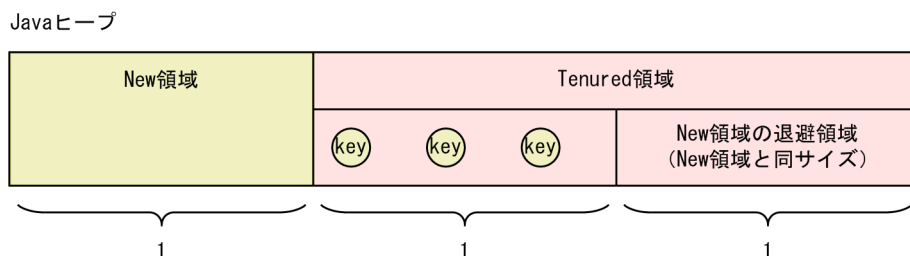
EADS サーバが使用するメモリの構成を次の図に示します。

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。



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 領域の使用量を基に 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、更新操作の履歴とノンブロッキング I/O 通信バッファ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 領域の使用量を 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、共通設定の `eads.java.heapsize` パラメタに指定します。デフォルトは 3 ギガバイトです。

- ノンブロッキング I/O 通信制御を使用しない場合は、ノンブロッキング I/O 通信制御を使用しない場合の計算式で見積もってください。
- ノンブロッキング I/O 通信制御を使用する場合は、ノンブロッキング I/O 通信制御を使用する場合の計算式で見積もってください。

#### 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 Java ヒープサイズの合計を、1EADS サーバ当たりの 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。

#### ■ノンブロッキング I/O 通信制御を使用しない場合

Java ヒープサイズ (単位: メガバイト) (ノンブロッキング I/O 通信制御を使用しない場合) =  
{(クラスタに格納できる key の最大サイズ (単位: バイト) + 746 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<sup>※1</sup>

+ EADS サーバへの最大同時接続数  
 × (14 + データの送受信バッファのサイズ (単位: バイト) × 3)  
 + EADS サーバ数<sup>\*2</sup> × 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・スケールアウト処理・リバランス処理で使用する Java ヒープ領域 (単位: バイト)  
 + データ自動削除機能で使用する Java ヒープ領域 (単位: バイト)}  
 × 3 ÷ 1,024<sup>2\*</sup><sup>3</sup>

## ■ノンブロッキング I/O 通信制御を使用する場合

Java ヒープサイズ (単位: メガバイト) (ノンブロッキング I/O 通信制御を使用する場合) =  
 {(クラスタに格納できる key の最大サイズ (単位: バイト) + 746 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<sup>\*1</sup>  
 + EADS サーバへの最大同時接続数  
 × (14 + データの送受信バッファのサイズ (単位: バイト) × 2)  
 + ノンブロッキング I/O 通信バッファのサイズ (単位: バイト)  
 + EADS サーバ数<sup>\*2</sup> × 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・スケールアウト処理・リバランス処理で使用する Java ヒープ領域 (単位: バイト)  
 + データ自動削除機能で使用する Java ヒープ領域 (単位: バイト)}  
 × 3 ÷ 1,024<sup>2\*</sup><sup>3</sup>

### 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

クラスタを停止させないで、EADS サーバをクラスタに追加する (スケールアウト処理を実行する) 予定がある場合は、EADS サーバ数に、追加する EADS サーバの数を含めてください。追加する EADS サーバの数が予測できない場合は、EADS サーバ数に、96 を代入してください。

### 注※3

小数点以下の値は切り上げてください。

クラスタに格納できる key の最大サイズ (単位: バイト):

共通設定の `eads.cache.key.maxsize` パラメタの指定値

1EADS サーバに格納するデータ件数

「4.1.2(1)(a) 1EADS サーバに格納するデータ件数」を参照してください。

EADS サーバへの最大同時接続数:

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

ノンブロッキング I/O 通信バッファのサイズ (単位: バイト):

「4.1.3(3) ノンブロッキング I/O 通信バッファのサイズ」を参照してください。

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):

「4.1.2(1)(b) 更新操作の履歴の管理で使用する Java ヒープ領域」を参照してください。

縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域 (単位: バイト):

「4.1.2(1)(c) 縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域」を参照してください。

データ自動削除機能で使用する Java ヒープ領域 (単位: バイト):

次の計算式で見積もった値を代入してください。

データ自動削除機能で使用する Java ヒープ領域 (単位: バイト) =  
 $a + \sum b$  (キャッシュ数)

a:

データ自動削除機能の使用有無に関係なく加算が必要なヒープサイズです。次の計算式で見積もった値を代入してください。

$1,232 \text{ バイト} + (\text{データの多重度} \times 2) \times \text{キャッシュ数} \times 376 \text{ バイト}$

b:

キャッシュごとに加算が必要なヒープサイズです。1EADS サーバに格納するデータ件数分のデータのアクセス順を管理するリストのヒープサイズを追加します。

次の計算式で見積もった値を代入してください。もし、データ自動削除機能を使用しない場合は、次の計算式の中で「`eads.cache.eviction.keyCount` パラメタの指定値」に 0 バイトを代入してください。

$(\text{データの多重度} \times 2 - 1) \times 37,296 \text{ バイト}$   
 $+ \text{データの多重度} \times \text{eads.cache.eviction.keyCount パラメタの指定値} \times 72 \text{ バイト}$   
 $+ 19,520 \text{ バイト}$

## (a) 1EADS サーバに格納するデータ件数

1EADS サーバに格納するデータ件数を見積もる計算式を次に示します。

1EADS サーバに格納するデータ件数 =  
1 レンジ当たりのデータ件数  $\times$  データの多重度

1 レンジ当たりのデータ件数:

1 レンジ当たりのデータ件数を見積もる計算式を次に示します。

データ総量監視機能を使用する場合は、見積もった値を共通設定の `eads.cache.keyCount` パラメタに指定してください。

1 レンジ当たりのデータ件数 =  
(1 レンジ当たりのメモリキャッシュに格納するデータ件数)

+ 1 レンジ当たりの 2Way キャッシュに格納するデータ件数  
+ 1 レンジ当たりのディスクキャッシュに格納するデータ件数)

■ 1 レンジ当たりのメモリキャッシュに格納するデータ件数：

1 レンジ当たりのメモリキャッシュに格納するデータ件数を見積もる計算式を次に示します。

1 レンジ当たりのメモリキャッシュに格納するデータ件数 =  
クラスタのメモリキャッシュに格納するデータ件数<sup>※1</sup>  
× 余裕値係数  
÷ クラスタ内の EADS サーバの台数<sup>※2</sup>  
+ データ総量監視機能用にメモリキャッシュのデータとして加算する件数

• 注※1

データ自動削除機能を使用する場合は、「4.1.2(1)(d) データ自動削除機能を使用する場合のキャッシュに格納するデータ件数」を参照してください。

• 注※2

小数点以下の値は切り上げてください。

余裕値係数：

「4.1.2(1)(e) 余裕値係数」を参照してください。

なお、メモリキャッシュが複数ある場合は、メモリキャッシュごとに余裕値係数を掛ける必要があるため、次の計算式で求めてください。

1 レンジ当たりのメモリキャッシュに格納するデータ件数 =  
(クラスタのメモリキャッシュ [1] に格納するデータ件数  
× メモリキャッシュ [1] の余裕値係数  
+ ...  
クラスタのメモリキャッシュ [n] に格納するデータ件数  
× メモリキャッシュ [n] の余裕値係数) <sup>※1</sup>  
÷ クラスタ内の EADS サーバの台数<sup>※2</sup>  
+ データ総量監視機能用にメモリキャッシュのデータとして加算する件数

• 注※1

[n] は n 番目のメモリキャッシュを示します。

• 注※2

小数点以下の値は切り上げてください。

データ総量監視機能用にメモリキャッシュのデータとして加算する件数：

データ総量監視機能用にメモリキャッシュのデータとして加算する件数を見積もる計算式を次に示します。

データ総量監視機能を使用しない場合は、0 を代入してください。

メモリキャッシュだけを使用する場合に、この件数を加算してください。2Way キャッシュまたはディスクキャッシュを使用する場合は、0 を代入してください。



データ総量監視機能用にメモリキャッシュのデータとして加算する件数=  
EADS サーバへの最大同時接続数×最大同時更新データ件数

EADS サーバへの最大同時接続数：

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

最大同時更新データ件数：

データの一括操作を実行する場合は、10 を代入してください。一括操作を実行しない場合は、1 を代入してください。

■1 レンジ当たりの 2Way キャッシュに格納するデータ件数：

1 レンジ当たりの 2Way キャッシュに格納するデータ件数を見積もる計算式を次に示します。

1 レンジ当たりの 2Way キャッシュに格納するデータ件数=  
クラスタの 2Way キャッシュに格納するデータ件数  
×余裕値係数  
÷クラスタ内の EADS サーバの台数※  
+データ総量監視機能用に 2Way キャッシュのデータとして加算する件数

• 注※

小数点以下の値は切り上げてください。

余裕値係数：

「4.1.2(1)(e) 余裕値係数」を参照してください。

なお、2Way キャッシュが複数ある場合は、2Way キャッシュごとに余裕値係数を掛ける必要があるため、次の計算式で求めてください。

1 レンジ当たりの 2Way キャッシュに格納するデータ件数=  
(クラスタの 2Way キャッシュ [1] に格納するデータ件数  
×2Way キャッシュ [1] の余裕値係数  
+...  
クラスタの 2Way キャッシュ [n] に格納するデータ件数  
×2Way キャッシュ [n] の余裕値係数) ※1  
÷クラスタ内の EADS サーバの台数※2  
+データ総量監視機能用に 2Way キャッシュのデータとして加算する件数

• 注※1

[n] は n 番目の 2Way キャッシュを示します。

• 注※2

小数点以下の値は切り上げてください。

データ総量監視機能用に 2Way キャッシュのデータとして加算する件数：

データ総量監視機能用に 2Way キャッシュのデータとして加算する件数を見積もる計算式を次に示します。

データ総量監視機能を使用しない場合は、0 を代入してください。

2Way キャッシュだけを使用する場合、またはメモリキャッシュ、ディスクキャッシュ、2Way キャッシュを混在させて使用する場合に、この件数を加算してください。メモリキャッシュだけ、またはディスクキャッシュだけを使用する場合は、0を代入してください。

データ総量監視機能用に2Way キャッシュのデータとして加算する件数=  
EADS サーバへの最大同時接続数×最大同時更新データ件数

EADS サーバへの最大同時接続数：

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

最大同時更新データ件数：

メモリキャッシュを使用して、データの一括操作を実行する場合は、10を代入してください。それ以外の場合は、1を代入してください。

#### ■1 レンジ当たりのディスクキャッシュに格納するデータ件数：

1 レンジ当たりのディスクキャッシュに格納するデータ件数を見積もる計算式を次に示します。

1 レンジ当たりのディスクキャッシュに格納するデータ件数=  
クラスタのディスクキャッシュに格納するデータ件数<sup>※1</sup>  
×余裕値係数  
÷クラスタ内の EADS サーバの台数<sup>※2</sup>  
+データ総量監視機能用にディスクキャッシュのデータとして加算する件数

##### • 注※1

データ自動削除機能を使用する場合は、「4.1.2(1)(d) データ自動削除機能を使用する場合のキャッシュに格納するデータ件数」を参照してください。

##### • 注※2

小数点以下の値は切り上げてください。

余裕値係数：

「4.1.2(1)(e) 余裕値係数」を参照してください。

なお、ディスクキャッシュが複数ある場合は、ディスクキャッシュごとに余裕値係数を掛ける必要があるため、次の計算式で求めてください。

1 レンジ当たりのディスクキャッシュに格納するデータ件数=  
(クラスタのディスクキャッシュ [1] に格納するデータ件数  
×ディスクキャッシュ [1] の余裕値係数  
+...  
クラスタのディスクキャッシュ [n] に格納するデータ件数  
×ディスクキャッシュ [n] の余裕値係数) <sup>※1</sup>  
÷クラスタ内の EADS サーバの台数<sup>※2</sup>  
+データ総量監視機能用にディスクキャッシュのデータとして加算する件数

##### • 注※1

[n] は n 番目のディスクキャッシュを示します。

##### • 注※2

小数点以下の値は切り上げてください。

データ総量監視機能用にディスクキャッシュのデータとして加算する件数：

データ総量監視機能用にディスクキャッシュのデータとして加算する件数を見積もる計算式を次に示します。

データ総量監視機能を使用しない場合は、0 を代入してください。

ディスクキャッシュだけを使用する場合に、この件数を加算してください。メモリキャッシュまたは 2Way キャッシュを使用する場合は、0 を代入してください。

データ総量監視機能用にディスクキャッシュのデータとして加算する件数 =  
EADS サーバへの最大同時接続数 × 最大同時更新データ件数

EADS サーバへの最大同時接続数：

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

最大同時更新データ件数：

1 を代入してください。

データの多重度：

共通設定の `eads.replication.factor` パラメタの指定値

## (b) 更新操作の履歴の管理で使用する Java ヒープ領域

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<sup>※1</sup> - 1)

注※

クラスタを停止させないで、EADS サーバをクラスタに追加する (スケールアウト処理を実行する) 予定がある場合は、EADS サーバ数に、追加する EADS サーバの数を含めてください。追加する EADS サーバの数が予測できない場合は、EADS サーバ数に、96 を代入してください。

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  
+ MAX (生存確認の接続タイムアウト時間, 生存確認の受信タイムアウト時間)  
× (生存確認タイムアウト時のリトライ回数 + 1)  
+ ハートビートの送信間隔時間 × 疑惑しきい値係数 ) ÷ 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) <sup>※1, ※2, ※3</sup>

#### 注※1

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。

#### 注※2

データ自動削除機能を使用する場合、create, put, putAll で更新するデータ件数分のデータを削除するため、create, put, putAll でデータを更新する件数を倍にして 1EADS サーバへのデータの更新操作のスループットを見積もってください。

#### 注※3

小数点以下の値は切り上げてください。

ハートビートのタイムアウト時間：

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

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 (2, 10) の計算結果は 10 となります。

生存確認の接続タイムアウト時間：

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

生存確認の受信タイムアウト時間：

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

生存確認タイムアウト時のリトライ回数：

サーバ定義の `eads.failureDetector.retry` パラメタの指定値

ハートビートの送信間隔時間：

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

疑惑しきい値係数：

サーバ定義の `eads.failureDetector.assertive.threshold` パラメタの指定値が 1 の場合は 1 を代入してください。2 以上の場合は 2 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：

共通設定の `eads.replication.preparations` パラメタの指定値

データの多重度：

共通設定の `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.replication.sendQueue.length` パラメタの指定値

## (c) 縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域

縮退処理および復旧処理・スケールアウト処理・リバランス処理で使用する 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。

$$\begin{aligned} & \text{縮退処理で使用する Java ヒープ領域 (単位: バイト)} = \\ & \quad (\text{合意処理で一度に合意できる処理の最大数} \times 2 \\ & \quad \times \text{更新操作の履歴の最大サイズ (単位: バイト)} \\ & \quad \times (\text{キャッシュ単位に行われる処理の最大同時実行スレッド数} \\ & \quad + \text{キャッシュ数}) + (\text{データの多重度} - 1) \times \text{キャッシュ数} \\ & \quad \times \text{更新操作の履歴の補完処理で送信するデータサイズ (単位: バイト)}) \\ & \quad \times \text{MIN} (4 \times (\text{データの多重度} - 1), \text{EADS サーバ数}^{*1} - 1) \\ & \text{復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域 (単位: バイト)} = \\ & \quad \text{更新操作の履歴が格納される領域のサイズ (単位: メガバイト)} \times 1,024^2 \\ & \quad \div (\text{データの多重度} \times 2 - 1)^{*2} \\ & \quad + \text{復旧処理・スケールアウト処理・リバランス処理で送信するデータサイズ (単位: バイト)} \end{aligned}$$

### 注※1

クラスタを停止させないで、EADS サーバをクラスタに追加する（スケールアウト処理を実行する）予定がある場合は、EADS サーバ数に、追加する EADS サーバの数を含めてください。追加する EADS サーバの数が予測できない場合は、EADS サーバ数に、96 を代入してください。

### 注※2

小数点以下の値は切り上げてください。

合意処理で一度に合意できる処理の最大数：

共通設定の `eads.replication.preparations` 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。

$$\begin{aligned} & \text{更新操作の履歴の最大サイズ (単位: バイト)} = \\ & \quad \text{MAX} (\text{クラスタに格納できる key の最大サイズ} + \text{value の最大サイズ} \times 2, (\text{クラスタに格納できる key の最大サイズ} + \\ & \quad \text{value の最大サイズ}) \times \text{最大同時更新データ件数}) \end{aligned}$$

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 (2, 10) の計算結果は 10 となります。

クラスタに格納できる key の最大サイズ：

共通設定の `eads.cache.key.maxsize` 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

最大同時更新データ件数：

メモリキャッシュを使用して、データの一括操作を実行する場合は 10 を代入してください。それ以外の場合は、1 を代入してください。

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：

データの多重度-1（データの多重度が 1 の場合は 1）

データの多重度：

共通設定の `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.replication.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（ $3 \times 6$ ,  $4 + 7$ ）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：

共通設定の `eads.replication.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・スケールアウト処理・リバランス処理で送信するデータサイズ：

復旧処理・スケールアウト処理・リバランス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.transfer.datasize` パラメタの指定値（復旧処理・スケールアウト処理）またはサーバ定義の `eads.rebalance.transfer.datasize` パラメタの指定値（リバランス処理）

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.disk.transfer.datasize` パラメタの指定値

#### (d) データ自動削除機能を使用する場合のキャッシュに格納するデータ件数

データ自動削除機能を使用する場合、次の計算式を求める必要があります。

- クラスタのメモリキャッシュに格納するデータ件数（データ自動削除機能の使用時）
- クラスタのディスクキャッシュに格納するデータ件数（データ自動削除機能の使用時）

求めた値を、「4.1.2(1)(a) 1EADS サーバに格納するデータ件数」にある次の変数に代入してください。

##### ■代入する変数

- クラスタのメモリキャッシュに格納するデータ件数
- クラスタのディスクキャッシュに格納するデータ件数

データ自動削除機能を使用する場合に求める計算式を次に示します。



クラスタのメモリキャッシュに格納するデータ件数（データ自動削除機能の使用時）：

クラスタのメモリキャッシュに格納するデータ件数（データ自動削除機能の使用時）を見積もる計算式を次に示します。

クラスタのメモリキャッシュに格納するデータ件数（データ自動削除機能の使用時） =  
各メモリキャッシュに格納するデータ件数の総数

各メモリキャッシュに格納するデータ件数の総数：

各メモリキャッシュに格納するデータ件数をすべて合計した値を代入します。

各メモリキャッシュに格納するデータ件数：

各メモリキャッシュに格納するデータ件数を見積もる計算式を次に示します。メモリキャッシュごとに見積もる必要があります。

各メモリキャッシュに格納するデータ件数 =  
(各メモリキャッシュの 1 レンジあたりに格納するデータ件数 + 1)  
×クラスタ内の EADS サーバの台数

各メモリキャッシュの 1 レンジあたりに格納するデータ件数：

データ自動削除機能のデータ削除のしきい値で使用する 1 レンジ当たりのデータ件数 (`eads.cache.eviction.keyCount` パラメタの指定値) を代入してください。もし、各メモリキャッシュに格納するデータ件数を決めてから、各メモリキャッシュの 1 レンジあたりに格納するデータ件数を決める場合は、逆算してください。

クラスタのディスクキャッシュに格納するデータ件数（データ自動削除機能の使用時）：

クラスタのディスクキャッシュに格納するデータ件数（データ自動削除機能の使用時）を見積もる計算式を次に示します。

クラスタのディスクキャッシュに格納するデータ件数（データ自動削除機能の使用時） =  
各ディスクキャッシュに格納するデータ件数の総数

各ディスクキャッシュに格納するデータ件数の総数：

各ディスクキャッシュに格納するデータ件数をすべて合計した値を代入します。

各ディスクキャッシュに格納するデータ件数：

各ディスクキャッシュに格納するデータ件数を見積もる計算式を次に示します。ディスクキャッシュごとに見積もる必要があります。

各ディスクキャッシュに格納するデータ件数 =  
(各ディスクキャッシュの 1 レンジあたりに格納するデータ件数 + 1)  
×クラスタ内の EADS サーバの台数

各ディスクキャッシュの 1 レンジあたりに格納するデータ件数：

データ自動削除機能のデータ削除のしきい値で使用する 1 レンジ当たりのデータ件数 (`eads.cache.eviction.keyCount` パラメタの指定値) を代入してください。もし、各ディスクキャッシュに格納するデータ件数を決めてから、各ディスクキャッシュの 1 レンジあたりに格納するデータ件数を決める場合は、逆算してください。

## (e) 余裕値係数

データの格納先は、key のハッシュ値によって決定されるため、レンジ間のデータ件数には差異が生じます。この差異により、平均データ件数を上回ることがあっても全データを格納できるようにするため、次の余裕値係数を掛けてください。また、余裕値係数を使用した計算式で小数点以下の値は切り上げてください。

余裕値係数は、キャッシュに格納するデータ件数と、クラスタの EADS サーバの数によって異なります。キャッシュを複数扱う場合は、キャッシュごとに余裕値係数を設定してください。

キャッシュ[n]の余裕値係数＝

「クラスタのキャッシュ[n]に格納するデータ件数÷クラスタの EADS サーバ台数」の数値を a としたとき、

- ・  $1,000,000 \leq a$  の場合は、1.01
- ・  $100,000 \leq a < 1,000,000$  の場合は、 $a \times -2.223 \times 10^{-8} + 1.033$
- ・  $10,000 \leq a < 100,000$  の場合は、 $a \times -4.445 \times 10^{-7} + 1.075$
- ・  $1,000 \leq a < 10,000$  の場合は、 $a \times -1.445 \times 10^{-5} + 1.215$
- ・ 1,000 未満の場合、レンジ間のデータ件数が極端に偏る場合があります。

実際にデータを格納して偏りの有無を検証の上で、十分な余裕値を設定することを推奨します。

なお、key のグルーピングを行った場合、データ格納先の偏りは、グループの最上位階層の種類数に依存します。上記の「クラスタのキャッシュ[n]に格納するデータ件数」は、キャッシュ[n]に格納するグループの最上位階層の種類数に置き換えて計算してください。

(例)

キー一覧が、"AAA:key1", "AAA:aaa:key2", "BBB:key3", "CCC:key4", "CCC:ccc:key5" のような場合、グループの最上位階層の種類数は、"AAA", "BBB", "CCC" の 3 種類となります。

また、グループの最上位階層の文字列で同一のキーが存在し、その件数がグループごとに異なる場合は、グループに属するデータ件数が多いものが特定のレンジに偏った場合を考慮し、安全係数を設定してください。

上記の例では、"AAA" は 2 件、"BBB" は 1 件、"CCC" は 2 件と、グループによって件数が異なります。

### 注意事項

データ件数の EADS サーバ間の差異は、キーの設計によって大きく変化します。上記の余裕率は十分に大きな値を設定していますが、キーの設計によっては超過するおそれがあります。特にデータ件数が少ない場合や、key のグルーピングによって、特定のグループにデータが偏る場合などは、実際に EADS サーバに全データを格納するような方法で、レンジ間のデータ件数に大きな差異がないか検証することを推奨します。

レンジ間でデータ件数に大きな差異が生じた場合は、実際の差異に合わせて再見積もりを行って差異を許容できるようにしてください。または、キー設計を変更したり、EADS サーバのポジションを設定したりすることでデータ配置を調整してください。



なお、ポジションの変更については「[7.5.1 EADS サーバの位置の設定](#)」を参照してください。

## (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\(1\) EADS 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。

### • 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: メガバイト) =  
(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  
+ 0.8×1 レンジ当たりの 2Way キャッシュに格納するデータ件数) ×データの多重度×3÷1,024<sup>※</sup>

#### 注※

小数点以下の値は切り上げてください。

### • ディスクキャッシュの場合<sup>※1</sup>

Java ヒープサイズの見積もりの値に加算する値 (単位: メガバイト) =  
(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  
- 0.315×1 レンジ当たりのディスクキャッシュに格納するデータ件数) ×データの多重度×3÷1,024<sup>※2</sup>

#### 注※1

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。

#### 注※2

小数点以下の値は切り上げてください。

#### 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.fileenum` 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。

#### データの多重度:

共通設定の `eads.replication.factor` パラメタの指定値

さらに、`eads.cache.disk.highSpecDrive.enable` パラメタに `true` を指定する場合は、次に示す計算式で、キャッシュごとに Java ヒープサイズに加算する値を見積もります。そのあとで算出した値を合計して、上述の計算式で求めた Java ヒープサイズに加算します。

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 ヒープサイズの見積もり値に加算する値を見積もる計算式を次に示します。

Java ヒープサイズの見積もり値に加算する値 (単位: メガバイト) =  
 $384 \times \text{データの多重度} \div 1,024^{**}$

注※

小数点以下の値は切り上げてください。

### 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。

- ノンブロッキング I/O 通信制御を使用しない場合は、ノンブロッキング I/O 通信制御を使用しない場合の計算式で見積もってください。
- ノンブロッキング I/O 通信制御を使用する場合は、ノンブロッキング I/O 通信制御を使用する場合の計算式で見積もってください。

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、共通設定の `eads.java.external.heapsize` パラメタに指定します。

#### ■ ノンブロッキング I/O 通信制御を使用しない場合

Explicit ヒープサイズ (単位: メガバイト) (ノンブロッキング I/O 通信制御を使用しない場合) =  
 $(a \div 1,024^{**1} + b \div 1,024^2) \div 0.97^{**2}$

#### ■ ノンブロッキング I/O 通信制御を使用する場合

Explicit ヒープサイズ (単位: メガバイト) (ノンブロッキング I/O 通信制御を使用する場合) =  
 $(a \div 1,024^{**1} + b \div 1,024^2 + c \div 1,024^2) \div 0.97^{**2}$

#### 変数の説明

a: value 部分が格納される領域のサイズ (単位: バイト)

b: 更新操作の履歴が格納される領域のサイズ (単位: バイト)

c: ノンブロッキング I/O 通信バッファのサイズ (単位: バイト)

注※1

データ総量監視機能を使用する場合、「 $a \div 1,024^2$ 」は、データの多重度の倍数 (単位: メガバイト) になるよう切り上げてください。

注※2

「 $a \div 1,024^2$ 」, 「 $b \div 1,024^2$ 」, 「 $c \div 1,024^2$ 」のそれぞれの計算式では小数点以下の値は切り上げてください。そのあとで、最終的な計算結果も小数点以下の値は切り上げてください。

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。

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

注※1

シリアライズ後のサイズです。

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:

共通設定の `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、共通設定の `eads.replication.external.heapsize` パラメタに指定します。

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

注

小数点以下の値は切り上げます。

データの多重度:

共通設定の `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。

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

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  
+ MAX (生存確認の接続タイムアウト時間, 生存確認の受信タイムアウト時間)  
× (生存確認タイムアウト時のリトライ回数 + 1)  
+ ハートビートの送信間隔時間 × 疑惑しきい値係数) ÷ 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) ※1, ※2, ※3

#### 注※1

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。

#### 注※2

データ自動削除機能を使用する場合、create, put, putAll で更新するデータ件数分のデータを削除するため、create, put, putAll でデータを更新する件数を倍にして 1EADS サーバへのデータの更新操作のスループットを見積もってください。

#### 注※3

小数点以下の値は切り上げてください。

ハートビートのタイムアウト時間:

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

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 (2, 10) の計算結果は 10 となります。

生存確認の接続タイムアウト時間:

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

生存確認の受信タイムアウト時間:

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

生存確認タイムアウト時のリトライ回数:

サーバ定義の `eads.failureDetector.retry` パラメタの指定値

ハートビートの送信間隔時間:

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

疑惑しきい値係数:

サーバ定義の `eads.failureDetector.assertive.threshold` パラメタの指定値が 1 の場合は 1 を代入してください。2 以上の場合は 2 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:

共通設定の `eads.replication.preparations` 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) =

MAX (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 + value の最大サイズ) × 最大同時更新データ件数)

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 (2, 10) の計算結果は 10 となります。

クラスタに格納できる key の最大サイズ:

共通設定の `eads.cache.key.maxsize` 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

最大同時更新データ件数:

メモリキャッシュを使用して、データの一括操作を実行する場合は、10 を代入してください。それ以外の場合は、1 を代入してください。

### (3) ノンブロッキング I/O 通信バッファのサイズ

ノンブロッキング I/O 通信制御用の通信バッファサイズを見積もる計算式を次に示します。求めた値は、メガバイト単位に切り上げてから、共通設定の `eads.server.nonBlocking.external.heapsize` パラメタに指定してください。

ノンブロッキング I/O 通信バッファサイズ (単位: バイト) =

MAX (最大通信データサイズ (単位: バイト) × 最大同時割当数,  $1 \times 1,024^2$ )

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 (2, 10) の計算結果は 10 となります。

最大通信データサイズ (単位: バイト):

次の表から使用するクライアント API の中で最も大きい通信データサイズを確認してください。その際、ノンブロッキング I/O 通信バッファのサイズは、クラスタを再起動して変更するため、Key 長などの変数は、現在値ではなく将来を見越した値を代入してください。

表 4-1 クライアント API と通信データサイズの一覧

項番	クライアント API	通信方向	通信データサイズ (単位: バイト) ※
1	get	リクエスト	150 + CacheName 長 + Key 長
2	get	レスポンス	100 + Value 長
3	remove	リクエスト	150 + CacheName 長 + Key 長
4	remove	レスポンス	50
5	put	リクエスト	150 + CacheName 長 + Key 長 + Value 長

項番	クライアント API	通信方向	通信データサイズ (単位: バイト) ※
6	put	レスポンス	50
7	create	リクエスト	150 + CacheName 長 + Key 長 + Value 長
8	create	レスポンス	50
9	update	リクエスト	150 + CacheName 長 + Key 長 + Value 長
10	update	レスポンス	50
11	replace	リクエスト	150 + CacheName 長 + Key 長 + Value 長 + 比較 Value 長
12	replace	レスポンス	50
13	putAll	リクエスト	150 + CacheName 長 + $\Sigma$ [一括操作数] (各要素の Key 長 + 25) + $\Sigma$ [一括操作数] (各要素の Value 長 + 25)
14	putAll	レスポンス	50
15	putAll : 失敗	レスポンス	100 + 一括操作数 $\times$ 25
16	getAll (キー指定)	リクエスト	150 + CacheName 長 + $\Sigma$ [一括操作数] (各要素の Key 長 + 25)
17	getAll (キー指定)	レスポンス	100 + $\Sigma$ [一括操作数] (各要素の Value 長 + 25)
18	getAll (グループ指定)	リクエスト	150 + CacheName 長 + Group 長 + Key 長
19	getAll (グループ指定)	レスポンス	100 + $\Sigma$ [一括操作数] (各要素の Key 長 + 25) + $\Sigma$ [一括操作数] (各要素の Value 長 + 25)
20	removeAll (キー指定)	リクエスト	150 + CacheName 長 + $\Sigma$ [一括削除数] (各要素の Key 長 + 25)
21	removeAll (キー指定)	レスポンス	50
22	removeAll (キー指定) : 失敗	レスポンス	100 + 一括削除数 $\times$ 25
23	removeAll (グループ指定)	リクエスト	150 + CacheName 長 + Group 長
24	removeAll (グループ指定)	レスポンス	50
25	removeAll (ノード指定)	リクエスト	150 + CacheName 長
26	removeAll (ノード指定)	レスポンス	50
27	getGroupNameSet	リクエスト	150 + CacheName 長
28	getGroupNameSet	レスポンス	100 + $\Sigma$ [一括取得 Group 数] (各要素の Group 長 + 25)
29	getKeySet (グループ指定)	リクエスト	150 + CacheName 長 + Group 長
30	getKeySet (グループ指定)	レスポンス	100 + $\Sigma$ [一括取得 Key 数] (各要素の Key 長 + 25)
31	getKeySet (ノード指定)	リクエスト	150 + CacheName 長
32	getKeySet (ノード指定)	レスポンス	100 + $\Sigma$ [一括取得 Key 数] (各要素の Key 長 + 25)
33	getGroupCount	リクエスト	150 + CacheName 長
34	getGroupCount	レスポンス	100

#### 4. 必要なリソースを確認する

項番	クライアント API	通信方向	通信データサイズ (単位: バイト) ※
35	getKeyCount (グループ指定)	リクエスト	150 + CacheName 長 + Group 長
36	getKeyCount (グループ指定)	レスポンス	100
37	getKeyCount (ノード指定)	リクエスト	150 + CacheName 長
38	getKeyCount (ノード指定)	レスポンス	100
39	getFirstKey (グループ指定)	リクエスト	150 + CacheName 長 + Group 長
40	getFirstKey (グループ指定)	レスポンス	100 + Key 長
41	getFirstKey (ノード指定)	リクエスト	150 + CacheName 長
42	getFirstKey (ノード指定)	レスポンス	100 + Key 長
43	getNextKey (グループ指定)	リクエスト	150 + CacheName 長 + Group 長 + Key 長
44	getNextKey (グループ指定)	レスポンス	100 + Key 長
45	getNextKey (ノード指定)	リクエスト	150 + CacheName 長 + Key 長
46	getNextKey (ノード指定)	レスポンス	100 + Key 長
47	executeFunction (キーまたはグループ指定)	リクエスト	150 + CacheName 長 + FunctionName 長 + Key 長または Group 長 + Function 引数長
48	executeFunction (キーまたはグループ指定)	レスポンス	100 + Function 戻り値長
49	executeFunction (ノード指定)	リクエスト	150 + CacheName 長 + FunctionName 長 + Function 引数長
50	executeFunction (ノード指定)	レスポンス	100 + Function 戻り値長

注※

通信データサイズは、計算した結果に対して 16 バイト単位で切り上げて見積もってください。

CacheName 長：

キャッシュ名の文字数を代入してください。不明な場合は最大値 (32) を代入してください。

Key 長：

キー名の文字数を代入してください。不明な場合は共通設定の `eads.cache.key.maxsize` パラメタの指定値を代入してください。

Group 長：

グループ名の文字数を代入してください。不明な場合は {共通設定の `eads.cache.key.maxsize` パラメタの指定値 - 2} を代入してください。



Value 長, 比較 Value 長:

Value, 比較 Value が null である場合は 0 を代入してください。バイト配列である場合はそのバイト配列の長さを代入してください。オブジェクトである場合はそのオブジェクトを `java.io.ObjectOutputStream#writeObject` メソッドでシリアライズした際のバイト配列の長さを代入してください。不明な場合は最大値 (262,144) を代入してください。

FunctionName 長:

ファンクション名の文字数を代入してください。不明な場合は EADS サーバにデプロイした全ファンクション中、最も長いファンクション名の文字数を代入してください。

Function 引数長, Function 戻り値長:

null である場合は 0 を代入してください。バイト配列である場合はそのバイト配列の長さを代入してください。オブジェクトである場合はそのオブジェクトを `java.io.ObjectOutputStream#writeObject` メソッドでシリアライズした際のバイト配列の長さを代入してください。

一括操作数:

API の引数に渡した Key 数とクライアント設定の `eads.client.batchOperation.unit` パラメタの指定値のどちらか小さい方を代入してください。

一括削除数:

API の引数に渡した Key 数と 1,024 のどちらか小さい方を代入してください。

一括取得 Group 数:

Group の格納状況によって 1EADS サーバで取得できると想定される Group 数を代入してください。

一括取得 Key 数:

Key の格納状況および引数によって取得できると想定される Key 数を代入してください。

最大同時割当数

サーバ定義の `eads.server.nonBlocking.maxAllocateConnections` パラメタの指定値  
最大同時割当数を見積もる式を次に示します。

最大同時割当数 = 「最大通信データサイズ > データの送受信バッファのサイズ」を満たしており、かつ同時に実行されるキャッシュ操作およびファンクション操作リクエスト数の最大値

- 最大通信データサイズ (単位: バイト)  
この「[4.1.3\(3\) ノンブロッキング I/O 通信バッファのサイズ](#)」の冒頭で見積もった最大通信データサイズの値
- 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.server.connection.bufferSize` パラメタの指定値

## (4) Explicit ヒープサイズを見積もる際の注意事項

Explicit ヒープサイズを見積もる際の注意事項を次に示します。



1. 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、更新操作の履歴およびノンブロッキング I/O 通信バッファ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、データ総量監視機能を使用していない場合、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。
2. 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(8) 更新操作の履歴の補完処理」を参照してください。
3. ノンブロッキング I/O 通信バッファが不足してノンブロッキング I/O 通信バッファを割り当てられない場合、または 1 ギガバイトより大きいサイズのノンブロッキング I/O 通信バッファを要求された場合、EADS クライアントが次のエラーを返す場合があります。
  - クライアント API 実行時：EAD\_ERROR\_SERVER\_LIMIT\_RECEIVE\_BUFFER, または EAD\_ERROR\_SERVER\_LIMIT\_SEND\_BUFFER
  - EADS クライアント初期化時：EAD\_ERROR\_INIT\_CLUSTERINFO

#### 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※1 (単位：メガバイト) =
Java ヒープサイズ + Explicit ヒープサイズ + Metaspace 領域サイズ + 658
+ 1 × EADS サーバプロセス当たりのスレッド数
+ {MIN (MAX (データの送受信バッファのサイズ, 8,192), OS が許可する最大 TCP 受信バッファサイズ)
+ MIN (MAX (データの送受信バッファのサイズ, 8,192), OS が許可する最大 TCP 送信バッファサイズ)}
× EADS サーバへの最大同時接続数 ÷ 1,024※2
+ ダイレクトバッファ領域のサイズ
```

##### 注※1

ユーザファンクションを使用する場合は、EADS サーバのメモリ使用量にユーザファンクションで使用するメモリ量を加算してください。

##### 注※2

小数点以下の値は切り上げてください。

Metaspace 領域サイズ：

共通設定の `eads.java.metaspace.maxsize` 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 (2, 10) の計算結果は 2 となります。

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 (2, 10) の計算結果は 10 となります。

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.server.connection.buffersize` パラメタの指定値

OS が許可する最大 TCP 受信バッファサイズ (単位: バイト)

OS が提供する TCP 受信バッファサイズの制限方法を参照してください。

OS が許可する最大 TCP 送信バッファサイズ (単位: バイト)

OS が提供する TCP 送信バッファサイズの制限方法を参照してください。

1EADS サーバプロセス当たりのスレッド数 :

「4.3.1(1) 1EADS サーバプロセス当たりのスレッド数」を参照してください。

ダイレクトバッファ領域のサイズ :

「4.1.4(1) ダイレクトバッファ領域のサイズ」を参照してください。

## (1) ダイレクトバッファ領域のサイズ

EADS サーバプロセスと TCP ソケットの間で、データのやり取りをするためのダイレクトバッファ領域のサイズを見積もる計算式を次に示します。

- ノンブロッキング I/O 通信制御を使用しない場合は、ノンブロッキング I/O 通信制御を使用しない場合の計算式で見積もってください。
- ノンブロッキング I/O 通信制御を使用する場合は、ノンブロッキング I/O 通信制御を使用する場合の計算式で見積もってください。

### ■ノンブロッキング I/O 通信制御を使用しない場合

```
ダイレクトバッファ領域のサイズ (単位: メガバイト) =  
{EADS サーバへの最大同時接続数×EADS クライアントとの通信での使用量  
+ 2× (EADS サーバ台数-1) ×合意メッセージ送受信での使用量  
+ 3×データ転送での使用量  
+ EADS クライアントとの通信での一時的使用量  
+ EADS サーバ間の通信での一時的使用量  
+ 1,048,576  
} ÷ 1,048,576*
```

注※

小数点以下の値は切り上げてください。

EADS サーバへの最大同時接続数 :

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

EADS クライアントとの通信での使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
次に示す値のうち、最も大きな値を代入してください。
  - 8192
  - key サイズ + value サイズ + 128
  - replace を使用する場合、引数に指定する key サイズ + 2 × value サイズ + 128
  - 一括操作または参照操作を実行する場合、`eads.client.batchOperation.unit` パラメタの指定値 × (key サイズ + value サイズ) + 128
  - ユーザファンクションを呼び出す場合、ユーザファンクション名のサイズ + 実行時に指定する key サイズ + 引数長と戻り値長のどちらか大きい方 + 128
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値

合意メッセージ送受信での使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
「4.1.2(1)(c) 縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域」で見積もった「更新操作の履歴の最大サイズ」と `eads.replication.connection.bufferSize` パラメタの指定値のどちらか大きい方
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値を  
`eads.replication.connection.bufferSize` パラメタの指定値の倍数に切り上げた値

データ転送での使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
key サイズの最大値 + value サイズの最大値 + 128
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値

EADS クライアントとの通信での一時的な使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
0 を代入してください。
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値（EADS サーバでキャッシュされるダイレクトバッファ 1 つ当たりのサイズ）よりも大きいサイズのデータが EADS クライアントとの通信で扱われた場合、データのサイズに応じたダイレクトバッファが一時的に使用されます。

EADS クライアントとの通信での一時的な使用量を見積もる計算式を次に示します。

$$\text{EADS クライアントとの通信での一時的な使用量 (単位: バイト)} = \text{EADS サーバへの同時リクエスト数} \times \text{EADS クライアントとの通信データサイズ}$$

EADS サーバへの同時リクエスト数：

EADS サーバに同時に送信されるリクエストの数

EADS クライアントとの通信データサイズ：

`eads.java.nio.maxCachedBufferSize` パラメタの指定値と次に示す値のうち、最も大きな値を比較してください。比較した値が等しい場合、または `eads.java.nio.maxCachedBufferSize` パラメタの指定値の方が大きい場合は 0 を代入してください。それ以外の場合は、比較した最も大きな値を代入してください。

- 8192
- key サイズ + value サイズ + 128
- replace を使用する場合、引数に指定する key サイズ + 2 × value サイズ + 128
- 一括操作または参照操作を実行する場合、`eads.client.batchOperation.unit` パラメタの指定値 × (key サイズ + value サイズ) + 128
- ユーザファンクションを呼び出す場合、ユーザファンクション名のサイズ + 実行時に指定する key サイズ + 引数長と戻り値長のどちらか大きい方 + 128

なお、特定のリクエストだけ EADS クライアントとの通信データサイズが極端に大きくなる場合、次のように特定のリクエストの同時リクエスト数と通信データサイズから特定のリクエストの使用量を個別に計算して、一時的使用量に合算してください。このとき、特定のリクエスト以外の EADS サーバへの同時リクエスト数には、全体の同時リクエスト数から特定のリクエストの同時リクエスト数を引いた値を代入してください。

EADS クライアントとの通信での一時的使用量 (単位：バイト) =  
特定のリクエスト A の同時リクエスト数 × 特定のリクエスト A の通信データサイズ  
+ 特定のリクエスト B の同時リクエスト数 × 特定のリクエスト B の通信データサイズ  
:  
+ 特定のリクエスト以外の EADS サーバへの同時リクエスト数 × EADS クライアントとの通信データサイズ

EADS サーバ間の通信での一時的使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
0 を代入してください。
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値 (EADS サーバでキャッシュされるダイレクトバッファ 1 つ当たりのサイズ) よりも大きいサイズのデータが EADS サーバ間の通信で扱われた場合、データのサイズに応じたダイレクトバッファが一時的に使用されます。  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値と「更新操作の履歴の最大サイズ」を比較して、比較した値が等しい場合、または `eads.java.nio.maxCachedBufferSize` パラメタの指定値の方が大きい場合は 0 を代入してください。それ以外の場合は、次に示す EADS サーバ間の通信での一時的使用量の計算式の計算結果を代入してください。

EADS サーバ間の通信での一時的使用量 (単位：バイト) =  
4 × (データの多重度 - 1) × 更新操作の履歴の最大サイズ

データの多重度：

共通設定の `eads.replication.factor` パラメタの指定値

更新操作の履歴の最大サイズ：

「4.1.2(1)(c) 縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域」で見積もった「更新操作の履歴の最大サイズ」の値

## ■ノンブロッキング I/O 通信制御を使用する場合

```
ダイレクトバッファ領域のサイズ (単位：メガバイト) =  
{リクエスト処理スレッド数×EADS クライアントとの通信での使用量  
+ 2× (EADS サーバ台数-1) ×合意メッセージ送受信での使用量  
+ 3×データ転送での使用量  
+ EADS サーバ間の通信での一時的使用量  
+ 1,048,576  
} ÷ 1,048,576*
```

注※

小数点以下の値は切り上げてください。

リクエスト処理スレッド数：

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

EADS クライアントとの通信での使用量：

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

合意メッセージ送受信での使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
「4.1.2(1)(c) 縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域」で見積もった「更新操作の履歴の最大サイズ」と `eads.replication.connection.bufferSize` パラメタの指定値のどちらか大きい方
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値を  
`eads.replication.connection.bufferSize` パラメタの指定値の倍数に切り上げた値

データ転送での使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
key サイズの最大値 + value サイズの最大値 + 128
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：  
`eads.java.nio.maxCachedBufferSize` パラメタの指定値

EADS サーバ間の通信での一時的使用量：

- `eads.java.nio.maxCachedBufferSize` パラメタを指定していない場合：  
0 を代入してください。
- `eads.java.nio.maxCachedBufferSize` パラメタを指定している場合：

`eads.java.nio.maxCachedBufferSize` パラメタの指定値（EADS サーバでキャッシュされるダイレクトバッファ 1 つ当たりのサイズ）よりも大きいサイズのデータが EADS サーバ間の通信で扱われた場合、データのサイズに応じたダイレクトバッファが一時的に使用されます。

`eads.java.nio.maxCachedBufferSize` パラメタの指定値と「更新操作の履歴の最大サイズ」を比較して、比較した値が等しい場合、または `eads.java.nio.maxCachedBufferSize` パラメタの指定値の方が大きい場合は 0 を代入してください。それ以外の場合は、次に示す EADS サーバ間の通信での一時的使用量の計算式の計算結果を代入してください。

$$\text{EADS サーバ間の通信での一時的使用量 (単位: バイト)} = 4 \times (\text{データの多重度} - 1) \times \text{更新操作の履歴の最大サイズ}$$

データの多重度：

共通設定の `eads.replication.factor` パラメタの指定値

更新操作の履歴の最大サイズ：

「4.1.2(1)(c) 縮退処理・復旧処理・スケールアウト処理・リバランス処理で使用する Java ヒープ領域」で見積もった「更新操作の履歴の最大サイズ」の値



## 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、スレッドダンプを除いたログファイルの容量を見積もる計算式を次に示します。プロパティファイルのパラメタの指定によっては、スレッドダンプのファイル容量が大きくなる場合があります。事前にテストを実施するなどして、次に示す 1EADS サーバ当たりのログファイルの容量とは別に、保存先のディスクに十分な空き容量があることを確認してください。

1EADS サーバ当たりのログファイルの容量 (単位: メガバイト) =  
 $a + b + c + d + e + f + g + h + i + 321$

#### 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: EADS サーバによって出力される例外ログファイルの容量
- c: ユーザログファイルの容量
- d: ユーザ例外ログファイルの容量
- e: キャッシュファイル操作ログファイルの容量
- f: 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
- g: コマンド実行時に出力される例外ログファイルの容量
- h: 統計情報ファイルの容量
- i: Java 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。

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

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

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) EADS サーバによって出力される例外ログファイルの容量

EADS サーバによって出力される例外ログファイルの容量を見積もる計算式を次に示します。

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

EADS サーバによって出力される例外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.exception.filesize` パラメタの指定値

EADS サーバによって出力される例外ファイルのファイル数:

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

サーバ定義の `eads.logger.exception.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) 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。

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

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.user.logger.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.user.logger.filenum` パラメタの指定値



サーバ定義の `eads.user.logger.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。

#### (d) ユーザ例外ログファイルの容量

ユーザ例外ログファイルの容量を見積もる計算式を次に示します。

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

ユーザ例外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.user.logger.exception.filesize` パラメタの指定値

ユーザ例外ログファイルのファイル数:

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

サーバ定義の `eads.user.logger.exception.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。

#### (e) キャッシュファイル操作ログファイルの容量

キャッシュファイル操作ログファイルの容量を見積もる計算式を次に示します。

$$\begin{aligned} & \text{キャッシュファイル操作ログファイルの容量 (単位: メガバイト)} = \\ & \quad \text{キャッシュファイル操作ログファイルのファイルサイズ (単位: バイト)} \div 1,024^2 \\ & \quad \times \text{キャッシュファイル操作ログファイルのファイル数} \end{aligned}$$

キャッシュファイル操作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.cache.logger.diskCache.filesize` パラメタの指定値

キャッシュファイル操作ログファイルのファイル数:

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

サーバ定義の `eads.cache.logger.diskCache.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。

#### (f) 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。

$$\begin{aligned} & \text{コマンド実行時に出力されるメッセージログファイルの容量 (単位: メガバイト)} = \\ & \quad \text{コマンド実行時に出力されるメッセージログファイルのファイルサイズ (単位: バイト)} \div 1,024^2 \\ & \quad \times \text{コマンド実行時に出力されるメッセージログファイルのファイル数} \end{aligned}$$

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ファイルサイズ (単位: バイト):

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

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ファイル数:

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

コマンド定義の `eads.command.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。

## (g) コマンド実行時に出力される例外ログファイルの容量

コマンド実行時に出力される例外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例外ログファイルの容量 (単位: メガバイト) =  
コマンド実行時に出力される例外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>  
 $\times$  コマンド実行時に出力される例外ログファイルのファイル数

コマンド実行時に出力される例外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.command.logger.exception.filesize` パラメタの指定値

コマンド実行時に出力される例外ログファイルのファイル数:

コマンド定義の `eads.command.logger.exception.fileenum` パラメタの指定値

コマンド定義の `eads.command.logger.exception.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。

## (h) 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。

各統計情報ファイルの容量はバイト単位に見積もるので、統計情報ファイルの容量を見積もる計算式ではメガバイト単位に切り上げてください。

- ノンブロッキング I/O 通信制御を使用しない場合は、ノンブロッキング I/O 通信制御を使用しない場合の計算式で見積もってください。
- ノンブロッキング I/O 通信制御を使用する場合は、ノンブロッキング I/O 通信制御を使用する場合の計算式で見積もってください。

### ■ ノンブロッキング I/O 通信制御を使用しない場合

統計情報ファイルの容量 (単位: メガバイト) (ノンブロッキング I/O 通信制御を使用しない場合) =  
(統計情報ファイル (`eads_stats.csv`) の容量 (単位: バイト))  
+ キャッシュの統計情報ファイル (`eads_cache_stats.csv`) の容量 (単位: バイト)  
+ レンジごとの統計情報ファイル (`eads_store_stats.csv`) の容量 (単位: バイト)  
+ 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)  
+ 保守情報の統計情報ファイル (`eads_maintenance_stats.csv`) の容量 (単位: バイト))

$\div 1,024^2$

## ■ ノンブロッキング I/O 通信制御を使用する場合

統計情報ファイルの容量 (単位:メガバイト) (ノンブロッキング I/O 通信制御を使用する場合) =  
(統計情報ファイル (eads\_stats.csv) の容量 (単位:バイト)  
+ キャッシュの統計情報ファイル (eads\_cache\_stats.csv) の容量 (単位:バイト)  
+ レンジごとの統計情報ファイル (eads\_store\_stats.csv) の容量 (単位:バイト)  
+ ユーザ関数ファイル (eads\_function\_stats.csv) の容量 (単位:バイト)  
+ ノンブロッキング I/O 通信制御の統計情報ファイル (eads\_nonblocking\_stats.csv) の容量 (単位:バイト)  
+ 保守情報の統計情報ファイル (eads\_maintenance\_stats.csv) の容量 (単位:バイト))  
 $\div 1,024^2$

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)

## ■ キャッシュの統計情報ファイル (eads\_cache\_stats.csv) の容量

キャッシュの統計情報ファイル (eads\_cache\_stats.csv) の容量 (単位:バイト) =  
{1,024 + (2,048  
× (メモリキャッシュの数 + (ディスクキャッシュの数 + 2Way キャッシュの数) × データの多重度))  
× (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)

## ■ レンジごとの統計情報ファイル (eads\_store\_stats.csv) の容量

### ■ データ総量監視機能を使用する場合

レンジごとの統計情報ファイル (eads\_store\_stats.csv) の容量 (単位:バイト) =  
{1,024 + (2,048  
× (データの多重度 + ディスクキャッシュの数 + 2Way キャッシュの数))  
× (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)

### ■ データ総量監視機能を使用しない場合

レンジごとの統計情報ファイル (eads\_store\_stats.csv) の容量 (単位:バイト) =  
0

## ■ ユーザ関数ファイル (eads\_function\_stats.csv) の容量

ユーザ関数ファイル (eads\_function\_stats.csv) の容量 (単位:バイト) =  
{1,024 + (2,048 × ユーザ関数数)  
× (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)

## ■ ノンブロッキング I/O 通信制御の統計情報ファイル (eads\_nonblocking\_stats.csv) の容量

### ■ ノンブロッキング I/O 通信制御の統計情報出力が有効な場合

統計情報ファイル (eads\_nonblocking\_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)

### ■ ノンブロッキング I/O 通信制御の統計情報出力が無効な場合

統計情報ファイル (eads\_nonblocking\_stats.csv) の容量 (単位: バイト) =  
0

## ■ 保守情報の統計情報ファイル (eads\_maintenance\_stats.csv) の容量

統計情報ファイル (eads\_maintenance\_stats.csv) の容量 (単位: バイト) =  
{1,024 + 2,048 × (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)

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.filenum` パラメタの指定値

### (i) 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 × 2

Java ログファイルのファイルサイズ (単位: メガバイト):

サーバ定義の `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:

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

## (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。

ストアデータファイルの容量 (単位: メガバイト) =  
(1 件当たりの key サイズ + 1 件当たりの value サイズ<sup>\*1</sup> + 100)  
× 1EADS サーバに格納するデータ件数<sup>\*2</sup>  
× (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  
+ 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)  
÷ 1,024<sup>2</sup>

#### 注※1

シリアライズ後のサイズです。

#### 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：

共通設定の `eads.admin.backup.exportCommand.generation.maxNum` パラメタの指定値

ただし、`eads.admin.backup.exportCommand.generation.maxNum` パラメタに 0 を指定した場合は、`eztool export` コマンドで世代数の上限管理が行われなため、計算式には実際に出力する世代数を当てはめます。

`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：

共通設定の `eads.admin.backup.stopCommand.generation.maxNum` 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

「[4.2.1\(1\) 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「[4.2.1\(2\) 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。

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

## 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保守ログファイルの容量

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.client.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.client.logger.message.filenum` パラメタの指定値

クライアント定義の `eads.client.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保守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保守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保守ログファイルの容量 (単位: メガバイト) =  
保守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>  
 $\times$  保守ログファイルのファイル数

保守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.client.logger.maintenance.filesize` パラメタの指定値

保守ログファイルのファイル数:

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

クライアント定義の `eads.client.logger.maintenance.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。

- ノンブロッキング I/O 通信制御を使用しない場合は、ノンブロッキング I/O 通信制御を使用しない場合の計算式で見積もってください。
- ノンブロッキング I/O 通信制御を使用する場合は、ノンブロッキング I/O 通信制御を使用する場合の計算式で見積もってください。

##### ■ノンブロッキング I/O 通信制御を使用しない場合

$$\begin{aligned} & \text{1EADS サーバプロセス当たりのスレッド数 (ノンブロッキング I/O 通信制御を使用しない場合) =} \\ & \{ \text{キャッシュ数} \times (\text{データの多重度} \times 2 - 1) \times 2 + \text{EADS サーバ数}^{\ast 1} \times 6 \\ & + \text{キャッシュ単位に行われる処理の最大同時実行スレッド数} \\ & + \text{キャッシュ数} \times 3 \\ & + \text{EADS サーバへの最大同時接続数} + 73 \} \div 0.8^{\ast 2} \end{aligned}$$

##### ■ノンブロッキング I/O 通信制御を使用する場合

$$\begin{aligned} & \text{1EADS サーバプロセス当たりのスレッド数 (ノンブロッキング I/O 通信制御を使用する場合) =} \\ & \{ \text{キャッシュ数} \times (\text{データの多重度} \times 2 - 1) \times 2 + \text{EADS サーバ数}^{\ast 1} \times 6 \\ & + \text{キャッシュ単位に行われる処理の最大同時実行スレッド数} \\ & + \text{キャッシュ数} \times 3 \\ & + \text{リクエスト処理スレッド数} + \text{セレクトラスレッド数} + 73 \} \div 0.8^{\ast 2} \end{aligned}$$

##### 注※1

クラスタを停止させないで、EADS サーバをクラスタに追加する（スケールアウト処理を実行する）予定がある場合は、EADS サーバ数に、追加する EADS サーバの数を含めてください。追加する EADS サーバの数が予測できない場合は、EADS サーバ数に、96 を代入してください。

##### 注※2

小数点以下の値は切り上げてください。

データの多重度：

共通設定の `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：

データの多重度 - 1（データの多重度が 1 の場合は 1）



EADS サーバへの最大同時接続数：

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

リクエスト処理スレッド数：

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

リクエスト処理スレッド数を見積もる計算式を次に示します。なお、この見積もりによる計算値は、決定値ではなくチューニングの際の参考値であるため、実際の環境や性能に応じて変更が必要な可能性があります。また、リクエスト処理スレッド数は「同時に実行されるキャッシュ操作およびファンクション操作リクエスト数の最大値」まで増やすと、EADS サーバの処理性能が改善することがあります。

リクエスト処理スレッド数 =  
MAX (1EADS サーバに対して割り当て可能な論理コア数 - セレクタスレッド数 - 1, 1)

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 (2, 10) の計算結果は 10 となります。

セレクタスレッド数：

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

セレクタスレッド数には 1 を指定することを推奨します。なお、1EADS サーバに対して割り当て可能な論理コア数が多い場合には、指定値を大きくすると、EADS サーバの処理性能が改善することがあります。

## (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。

- ノンブロッキング I/O 通信制御を使用しない場合は、ノンブロッキング I/O 通信制御を使用しない場合の計算式で見積もってください。
- ノンブロッキング I/O 通信制御を使用する場合は、ノンブロッキング I/O 通信制御を使用する場合の計算式で見積もってください。

#### ■ ノンブロッキング I/O 通信制御を使用しない場合

1EADS サーバプロセス当たりのファイルディスクリプタ数 (ノンブロッキング I/O 通信制御を使用しない場合) =



$$\begin{aligned}
 & (\text{EADS サーバ数}^{\ast 1} \times 15 + \text{EADS サーバへの最大同時接続数} \\
 & + \text{ユーザが作成するライブラリ数}^{\ast 2} \\
 & + 361 \\
 & + \text{ディスク上のキャッシュが使用するファイルディスクリプタ数} \\
 & + \text{ユーザプログラムに必要なファイルディスクリプタ数} \\
 & ) \div 0.8^{\ast 3}
 \end{aligned}$$

## ■ノンブロッキング I/O 通信制御を使用する場合

$$\begin{aligned}
 & 1\text{EADS サーバプロセス当たりのファイルディスクリプタ数 (ノンブロッキング I/O 通信制御を使用する場合)} = \\
 & (\text{EADS サーバ数}^{\ast 1} \times 15 + \text{EADS サーバへの最大同時接続数} \\
 & + \text{セレクトスレッド数} \times 3 \\
 & + \text{ユーザが作成するライブラリ数}^{\ast 2} \\
 & + 363 \\
 & + \text{ディスク上のキャッシュが使用するファイルディスクリプタ数} \\
 & + \text{ユーザプログラムに必要なファイルディスクリプタ数} \\
 & ) \div 0.8^{\ast 3}
 \end{aligned}$$

### 注※1

クラスタを停止させないで、EADS サーバをクラスタに追加する (スケールアウト処理を実行する) 予定がある場合は、EADS サーバ数に、追加する EADS サーバの数を含めてください。追加する EADS サーバの数が予測できない場合は、EADS サーバ数に、96 を代入してください。

### 注※2

ユーザが作成するユーザ関数 (jar ファイル数) と、ユーザ関数で使用するライブラリ数の合計です。

### 注※3

小数点以下の値は切り上げてください。

EADS サーバへの最大同時接続数：

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

セレクトスレッド数：

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

「4.3.1(1) 1EADS サーバプロセス当たりのスレッド数」で求めるセレクトスレッド数を確認して、値を代入してください。

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。

$$\begin{aligned}
 & \text{ディスク上のキャッシュが使用するファイルディスクリプタ数} = \\
 & 5 \times \text{データの多重度} \times \text{ディスクを使用するキャッシュ数}
 \end{aligned}$$

データの多重度：

共通設定の `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，`eztool importecf` コマンド，または `eztool resume` コマンド実行時に EADS サーバが使用する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。

これらのコマンドを実行する場合は，上記の 1EADS サーバプロセス当たりのファイルディスクリプタ数（ノンブロッキング I/O 通信制御を使用しない場合），または 1EADS サーバプロセス当たりのファイルディスクリプタ数（ノンブロッキング I/O 通信制御を使用する場合）の値に，次の計算式で見積もった値を加算してください。なお，`eztool import`，`eztool importecf`，`eztool resume` は同時実行されないため，運用の中でそれぞれを実行する場合でも見積もりが増加することはありません。

```
eztool import コマンド， eztool importecf コマンド， eztool resume コマンド実行時のファイルディスクリプタ数=  
{1 + (EADS サーバ数-1) × 2  
} ÷ 0.8※
```

注※

小数点以下の値は切り上げてください。

EADS サーバへの最大同時接続数：

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

データの多重度：

共通設定の `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。

ディスクキャッシュの格納先として不揮発性メモリを使用する場合は、「4.4.3 ディスクキャッシュの格納先として不揮発性メモリを使用する場合のリソースを見積もる」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、1レンジ当たりの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.fileenum` 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。
10. 8.から、全レンジのキャッシュデータファイルの容量を見積もります。

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 ÷ 1,024 ※2

### 注※1

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) をバイト単位に変換した値の整数倍に切り上げて見積もってください。

### 注※2

小数点以下の値は切り上げてください。

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。

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) =  
(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) × 1,024)  
- (書き込みブロックサイズ (単位: キロバイト) × 2) \*

#### 注※

管理情報の「(書き込みブロックサイズ (単位: キロバイト) × 2)」の算出値の、小数点以下の値は切り上げてください。

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。

有効データの最大量※（単位：キロバイト）＝  
1 レンジ当たりのディスクキャッシュに格納するデータ件数×レコード 1 件分のサイズ（単位：キロバイト）

注※

小数点以下の値は切り上げてください。

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。

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

注※

小数点以下の値は切り上げてください。

1 EADS 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) 1 レンジ当たりのキャッシュデータファイルに必要な容量を見積もる

1 レンジ当たりの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。

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

注※

小数点以下の値は切り上げてください。

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) 1 レンジ当たりの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、1 レンジ当たりの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、キャッシュデータファイル 1 面のサイズ、および 1 レンジ当たりのキャッシュデータファイルの面数を設計します。

$$\begin{aligned} &1 \text{ レンジ当たりのキャッシュデータファイルに必要な容量 (単位: メガバイト)} \leq \\ &(\text{キャッシュデータファイル 1 面のデータ格納領域 (単位: キロバイト)} \\ &\times 1 \text{ レンジ当たりのキャッシュデータファイルの面数}^{**}) \div 1,024 \end{aligned}$$

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。

上記の式を満たすためにキャッシュデータファイル 1 面のデータ格納領域を増やす場合は、キャッシュ定義の `eads.cache.disk.filesize` パラメタに新しい値を指定し、「4.4.1(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」を参照して、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再度算出してください。



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。

$$\begin{aligned} &1 \text{ レンジ当たりのキャッシュデータファイルの面数} = \\ &\text{キャッシュデータファイルの面数} + 2^{**} \end{aligned}$$

注※

データ総量監視機能を使用しない場合は、1 としてください。

算出した値は、キャッシュ定義の `eads.cache.disk.filenum` 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。

## (10) 全レンジのキャッシュデータファイルの容量を見積もる

データの多重度を考量した、全レンジのキャッシュ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。この計算式で算出した値を「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加算します。

$$\begin{aligned} &\text{全レンジのキャッシュデータファイルの容量 (単位: メガバイト)} = \\ &\text{キャッシュデータファイル 1 面のサイズ (単位: メガバイト)} \\ &\times 1 \text{ レンジ当たりのキャッシュデータファイルの面数} \\ &\times \text{データの多重度} \end{aligned}$$

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

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

データの多重度:

共通設定の `eads.replication.factor` パラメタの指定値



## 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

ここでは、キャッシュインデックスファイルのサイズの見積もりについて説明します。

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

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

注※

小数点以下の値は切り上げてください。

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)

### (2) 全レンジのキャッシュインデックスファイルの容量を見積もる

データの多重度を考量した、全レンジのキャッシュインデックスファイルの容量を見積もる計算式を次に示します。この計算式で算出した値を「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加算します。

$$\begin{aligned} & \text{全レンジのキャッシュインデックスファイルの容量 (単位: メガバイト)} = \\ & \text{キャッシュインデックスファイル 1 面のサイズ (単位: バイト)} \\ & \times 1 \text{ レンジ当たりのキャッシュデータファイルの面数} \\ & \times \text{データの多重度} \div 1,048,576^* \end{aligned}$$

注※

小数点以下の値は切り上げてください。

キャッシュインデックスファイル 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

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.fileenum` パラメタの指定値

データの多重度:

共通設定の `eads.replication.factor` パラメタの指定値

### 4.4.3 ディスクキャッシュの格納先として不揮発性メモリを使用する場合のリソースを見積もる

ディスクキャッシュの格納先として不揮発性メモリを使用する場合のリソースの見積もりについて説明します。

#### (1) 不揮発性メモリに格納するキャッシュファイルの容量を見積もる

ディスクキャッシュの格納先として不揮発性メモリを使用する場合、EADS サーバは、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を不揮発性メモリのデバイスファイル内に格納します。格納対象の各ファイルのサイズは、「4.4 キャッシュファイルの容量を見積もる」の内容を参照して見積もってください。

ただし、「4.4 キャッシュファイルの容量を見積もる」に記載された、キャッシュ定義の `eads.cache.disk.blocksize` パラメタの指定値の値は、キャッシュ定義の `eads.cache.disk.persistentMemory.blocksize` パラメタに読み替えてください。

不揮発性メモリのブロックサイズの単位はバイトです。計算式の書き込みブロックサイズの単位はキロバイトとなっているため、256 バイトは 0.25 キロバイト、512 バイトは 0.5 キロバイトとして計算してください。また、レコード 1 件分のサイズは小数点以下の値もそのまま計算してください。その他の計算式については、小数点以下の値を切り上げて計算すると記載されている場合は、記載に従ってください。

#### (2) 不揮発性メモリのデバイスファイルに必要な容量を見積もる

EADS サーバが 1 キャッシュで必要とする不揮発性メモリの容量を次の計算式に示します。キャッシュが複数あり、キャッシュごとに容量が異なる場合は、それぞれ見積もる必要があります。

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を格納するのに必要な不揮発性メモリのデバイスファイルの容量 (単位: メガバイト)  $\geq$

$$\begin{aligned} & ((\text{キャッシュデータファイルのサイズ (単位: メガバイト)} \times 1,048,576 \\ & + \text{キャッシュインデックスファイルのサイズ (単位: バイト)}) \\ & \times 1 \text{ レンジ当たりのキャッシュデータファイルの面数} \times \text{データの多重度} \\ & + \text{不揮発性メモリのデバイスファイル管理領域のサイズ (単位: バイト)}) \\ & \div 1,048,576 \end{aligned}$$

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

「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」で算出した値を指定します。

キャッシュインデックスファイルのサイズ:

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

「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」で算出した値を、キャッシュ定義の `eads.cache.disk.persistentMemory.blocksize` パラメタの整数倍になるように切り上げた値を指定します。

1 レンジ当たりのキャッシュデータファイルの面数：

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

「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」で算出した値を指定します。

データの多重度：

共通設定の `eads.replication.factor` パラメタの指定値

不揮発性メモリのデバイスファイル管理領域のサイズ：

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

### (3) ディスク使用量を見積もる

不揮発性メモリのデバイスファイルから `eztool exportecf` コマンドでキャッシュファイルを出力する場合、「4.4.3(2) 不揮発性メモリのデバイスファイルに必要な容量を見積もる」で算出した容量と同一の容量を「4.2 ディスク使用量を見積もる」の算出結果に加算してください。

`eztool exportecf` コマンドをバックアップのために使用して複数の世代を保存する場合には、その世代数分の容量を加算してください。

### (4) 不揮発性メモリ使用上の注意事項

ディスクキャッシュの格納先として不揮発性メモリを使用する場合、EADS サーバで使用するリソースのほかに、OS やサーバでリソースが必要になります。また、OS やサーバによっては不揮発性メモリを使用する際に条件があります。そのため、不揮発性メモリを使用する場合には OS やサーバなどのマニュアルを参照し、それぞれのリソースに問題がないか、使用条件を満たしているかを確認してください。

# 5

## インストールおよびセットアップを行う (EADS サーバ)

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せん (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。

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

- 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。
- 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。
- 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)==>
```

- 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。
- 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。
- 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。

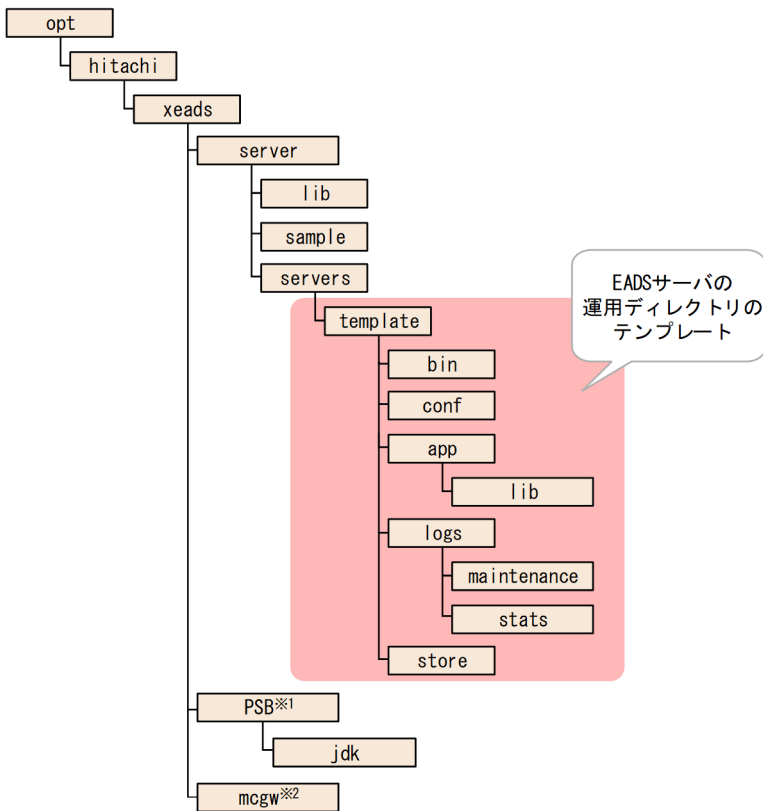
### 5.1.3 インストール後にすること

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  
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。
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>共通設定ファイル</li> <li>コマンド定義ファイル</li> </ul>

ディレクトリ	説明
	<ul style="list-style-type: none"> <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）の格納先です。
lib	ユーザファンクション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。
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/PSB	EADS で使用する JDK の格納用ディレクトリです。
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。

## 注意事項

- EADS のインストールによって配置されるディレクトリとファイルの所有者、グループ、権限を変更しないでください。また、運用ディレクトリの配下以外に、ユーザ固有のファイルを格納しないでください。これらを行うと、EADS が正しく動作しないおそれがあります。
- /opt/hitachi/xeads/javaclient は EADS V5 までは EADS サーバをインストールすることでインストールされましたが、EADS V6 以降は EADS クライアントの Hitachi Elastic Application Data Store Client for Java をインストールする必要があります。



## 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.command.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)
- 共通設定ファイル (eads\_shared.properties)
- コマンド定義ファイル (eads\_command.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、「21.1 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、「17.4 ファンクション定義ファイルを作成する (任意)」および「17.9 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および共通設定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.eviction.policy
  - eads.cache.eviction.keyCount (データ自動削除機能を使用する場合)
  - eads.cache.disk.filesize
  - eads.cache.disk.filenum
  - eads.cache.disk.blocksize
  - eads.cache.disk.persistentMemory.enable
  - eads.cache.disk.persistentMemory.blocksize (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)

## 5.2.3 AP を配置する

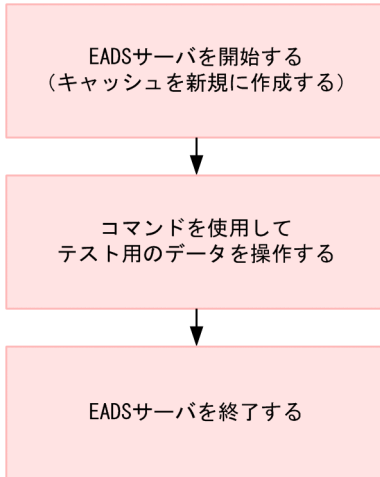
開発環境で作成した AP を、実行環境に配布します。

ユーザ機能の実行環境への配布については、「[17.9 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  Port  Position  Cluster  State  Operation  Lock  KeyCount  UsedMemoryRatio  UsedMemory  MaxMemory  Version
1  XX.XXX.XXX.168  server01    24600  1288490189  online  running  none     unlock  1         0                0           92         04-00-00

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.
```

指定したキャッシュに格納されている key, および key に関連づけられた value をすべて削除したい場合は、`eztool removeall` コマンドを実行します。

```
eztool removeall [-g <グループ名>] <キャッシュ名>
```

### コマンド実行例

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

また、`eztool removeall` コマンドに `-g` オプションを指定すると、指定したキャッシュに格納されている key のうち、指定したグループに属する key, およびその key に関連づけられた value をすべて削除できます。

## 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

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。

- Hitachi Elastic Application Data Store Client for Java

#### 参考

複数のマシンに EADS クライアントをインストールする場合、EADS クライアントインストール済みマシンの次のディレクトリをコピーしてインストールすることもできます。  
javaclient ディレクトリのコピー先は/opt/hitachi/xeads 以外でもかまいません。

/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。

- Hitachi 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。

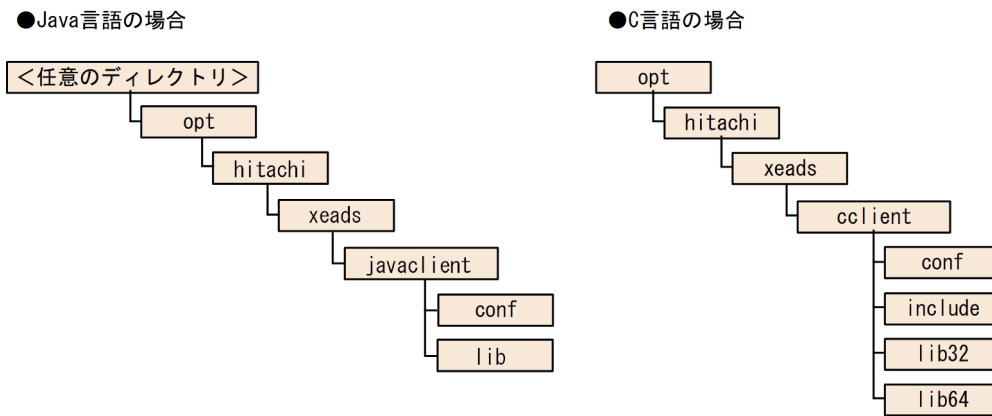
### 6.1.1 インストール後にすること

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) のインストールディレクトリです。 なお、インストーラでインストールされるディレクトリは/opt/hitachi/xeads/javaclient 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。

## 注意事項

EADS のインストールによって配置されるディレクトリとファイルの所有者、グループ、権限を変更しないでください。また、運用ディレクトリの配下以外に、ユーザ固有のファイルを格納しないでください。これらを行うと、EADS が正しく動作しないおそれがあります。

## (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。

javaclient ディレクトリを/opt/hitachi/xeads ディレクトリから別のディレクトリにコピーしている場合は、コピー先のディレクトリのファイルを指定してください。

- /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、Java EE サーバ (uCosminexus Application Server) 上で実行する場合は、上記のライブラリを 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。Java EE サーバでは、Servlet または JSP から実行できます。

## (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

Java EE 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、SecurityManager が解除された状態で実行してください。

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、「[21.1.2\(1\) 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

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。

- Hitachi Elastic Application Data Store Client for Java

### 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。

- Hitachi 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、「[21.1 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	クラスタの構成を定義します。
3	共通設定ファイル	eads_shared.properties	各 EADS サーバで共通の設定を定義します。
4	コマンド定義ファイル	eads_command.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="#">「17.4 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および共通設定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.eviction.policy`
  - `eads.cache.eviction.keyCount` (データ自動削除機能を使用する場合)
  - `eads.cache.disk.filesize`
  - `eads.cache.disk.filenum`
  - `eads.cache.disk.blocksize`
  - `eads.cache.disk.persistentMemory.enable`



・ `eads.cache.disk.persistentMemory.blocksize` (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)

- 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.logger.message.console.enable=true  
eads.logger.message.console.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。

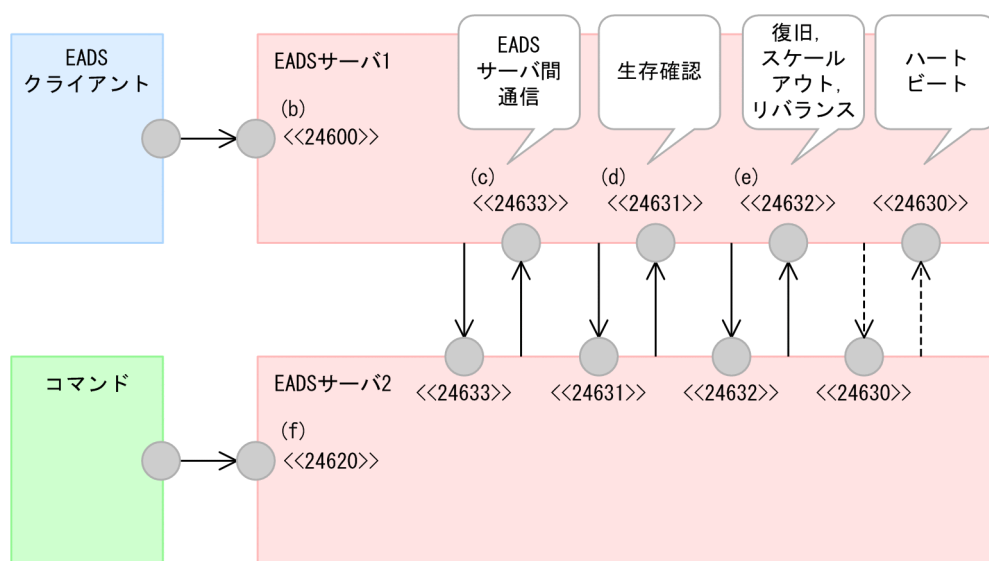
### 7.3.1 IP アドレスまたはホスト名、およびポート番号の設定

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.replication.port」

(d) : 「7.3.2(1)(d) eads.failureDetector.port」

(e) : 「7.3.2(1)(e) eads.transfer.port」

(f) : 「7.3.2(1)(f) eads.admin.operation.port」

なお、コマンド定義に eztool コマンドの接続先 EADS サーバを指定しておくことで、eztool コマンドの初期接続時にサーバ定義で指定した EADS サーバに接続できない場合に、別の EADS サーバに接続を試みることができます。

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(2)(b) eads.failureDetector.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。共通設定の eads.failureDetector.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		eads.server.port	ポート番号	24600
3		eads.replication.port	ポート番号	24633
4		eads.failureDetector.port	ポート番号	24631
5		eads.transfer.port	ポート番号	24632
6		eads.admin.operation.port	ポート番号	24620
7	コマンド定義ファイル	<b>eads.command.connect.sub.servers</b>	接続先 EADS サーバを識別するための名称 (任意の名称)	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
8		<code>eads.command.&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
9		<code>eads.command.&lt;接続先 EADS サーバ&gt;.admin.operation.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.replication.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.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.transfer.port`

EADS サーバの復旧処理、スケールアウト処理、またはリバランス処理に使用するポート番号を指定します。

### (f) `eads.admin.operation.port`

コマンドが使用するポート番号を指定します。

## (2) コマンド定義ファイル

### (a) `eads.command.connect.sub.servers`

eztool コマンドの接続先 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。

コマンドの実行時、コマンドを実行した EADS サーバ（サーバ定義の `eads.server.address` パラメタとサーバ定義の `eads.admin.operation.port` パラメタで指定した接続先）にコマンドが初期接続します。

クラスタを実行対象とする eztool コマンドの実行時、コマンドを実行した EADS サーバへの初期接続に失敗した場合は、このパラメタに指定した順に接続先 EADS サーバへの接続が試行されます。このパラメタの指定を省略した場合、およびこのパラメタに空文字を指定した場合は、コマンドを実行した EADS サーバにだけ初期接続します。

また、次に示すコマンドの実行時は、このパラメタの指定値は無視されます。

- eztool listgroup
- eztool listkey
- eztool removeall
- eztool execfunc

なお、このパラメタとコマンド定義の `eads.command.<接続先 EADS サーバ>.admin.operation.port` パラメタで指定した接続先のうち、サーバ定義の `eads.server.address` パラメタとサーバ定義の `eads.admin.operation.port` パラメタで指定した接続先と同じ接続先は無視されます。

## (b) eads.command.<接続先 EADS サーバ>.address

eztool コマンドの接続先 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.command.connect.sub.servers` パラメタと対で指定します。接続先 EADS サーバには、`eads.command.connect.sub.servers` 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.command.<接続先 EADS サーバ>.admin.operation.port

eztool コマンドの接続先 EADS サーバのポート番号を指定します。

`eads.command.connect.sub.servers` パラメタと対で指定します。接続先 EADS サーバには、`eads.command.connect.sub.servers` 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.command.connect.sub.servers パラメタの指定例

`eads.command.connect.sub.servers` パラメタの指定例を次に示します。

```
eads.command.connect.sub.servers=sv1,sv2,sv3
eads.command.sv1.address=XXX.XXX.X.138
eads.command.sv1.admin.operation.port=24600
eads.command.sv2.address=XXX.XXX.X.139
eads.command.sv2.admin.operation.port=24601
eads.command.sv3.address=XXX.XXX.X.140
eads.command.sv3.admin.operation.port=24602
```

## 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	ユーザ関数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	ユーザ関数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	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。
9	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。
10	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。
11	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.command.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_command_message[n].log</code>
			Shift	<code>eads_command_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.command.logger.dir</code> パラメータに指定したディレクトリ	Wrap	<code>eads_command_exception[n].log</code>
			Shift	<code>eads_command_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.command.logger.dir</code> パラメータに指定したディレクトリ>/maintenance	<code>eads_command_maintenance[n].log</code>	



項番	ログファイル種別	出力先	ファイル名
11	統計情報ファイル (EADS サーバ)	<サーバ定義の <a href="#">eads.logger.dir</a> 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>
12	統計情報ファイル (キャッシュ)	<サーバ定義の <a href="#">eads.logger.dir</a> 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>
13	統計情報ファイル (ユーザファンクション)	<サーバ定義の <a href="#">eads.logger.dir</a> 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>
14	統計情報ファイル (レンジ)	<サーバ定義の <a href="#">eads.logger.dir</a> パラメ タに指定したディレクト リ>/stats	<ul style="list-style-type: none"> <li>• eads_store_stats.csv</li> <li>• eads_store_stats_[n].csv<sup>*1</sup></li> </ul>
15	統計情報ファイル (ノンブロッキング I/O 通信制御)	<サーバ定義の <a href="#">eads.logger.dir</a> パラメ タに指定したディレクト リ>/stats	<ul style="list-style-type: none"> <li>• eads_nonblocking_stats.csv</li> <li>• eads_nonblocking_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 <a href="#">eads.logger.dir</a> 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 <a href="#">eads.logger.dir</a> 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 <a href="#">eads.logger.dir</a> 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 <a href="#">eads.logger.dir</a> 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.filenum` 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

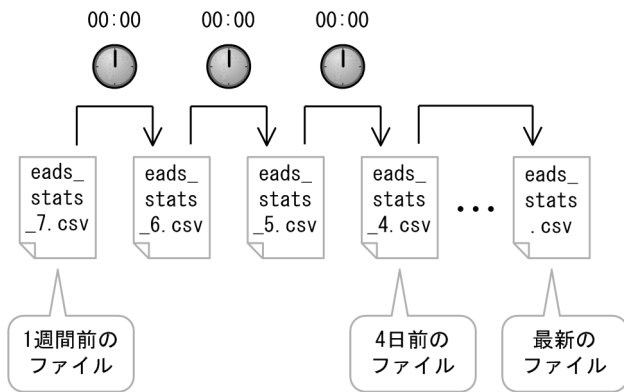
統計情報は、統計情報が更新されるタイミングでファイルに出力されます。

取得する統計情報ファイル数をサーバ定義の `eads.statistics.filenum` 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.user.logger.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.user.logger.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイト)	1048576 バイト
11		eads.user.logger.filenum	ファイル数 (1～ 16)	2
12		eads.user.logger.exception.rotationStyle	<ul style="list-style-type: none"> <li>Wrap</li> <li>Shift</li> </ul>	Wrap
13		eads.user.logger.exception.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイト)	1048576 バイト
14		eads.user.logger.exception.filenum	ファイル数 (1～ 16)	2
15		eads.cache.logger.diskCache.rotationStyle	<ul style="list-style-type: none"> <li>Wrap</li> <li>Shift</li> </ul>	Wrap
16		eads.cache.logger.diskCache.filesize	ファイルサイズ (4096 バイト～ 2147483647 バイト)	1048576 バイト
17		eads.cache.logger.diskCache.filenum	ファイル数 (1～ 16)	2
18		eads.statistics.interval	1～3600	1
19		eads.statistics.filenum	1～366	7
20		eads.statistics.compaction.effect.division	1～10	5
21		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
22		eads.java.log.filenum	ファイル数 (1～ 99)	4
23	コマンド定義ファイル	eads.command.logger.dir	パス名	<運用ディレクトリ >/logs
24		eads.command.logger.message.rotationStyle	<ul style="list-style-type: none"> <li>Wrap</li> <li>Shift</li> </ul>	Wrap
25		eads.command.logger.message.filesize	ファイルサイズ (4096 バイト～	1048576 バイト

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
			2147483647 バイト)	
26		<code>eads.command.logger.message.filenum</code>	ファイル数 (1~64)	2
27		<code>eads.command.logger.exception.rotationStyle</code>	<ul style="list-style-type: none"> <li>• Wrap</li> <li>• Shift</li> </ul>	Wrap
28		<code>eads.command.logger.exception.filesize</code>	ファイルサイズ (4096 バイト~2147483647 バイト)	1048576 バイト
29		<code>eads.command.logger.exception.filenum</code>	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 (スラッシュ (/)) 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.user.logger.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.user.logger.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.user.logger.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.user.logger.exception.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.user.logger.exception.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.user.logger.exception.fileenum

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.cache.logger.diskCache.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.cache.logger.diskCache.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.cache.logger.diskCache.fileenum

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.statistics.interval

統計情報を出力する間隔（単位：秒）を指定します。

## (s) eads.statistics.fileenum

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

#### (t) `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。

#### (u) `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。

#### (v) `eads.java.log.filenum`

Java ログのファイル数を指定します。

### (2) コマンド定義ファイル

#### (a) `eads.command.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 (スラッシュ (`/`)) 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.command.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.command.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.command.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.command.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.command.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.command.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.command.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.command.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.command.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.command.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.command.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.command.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。

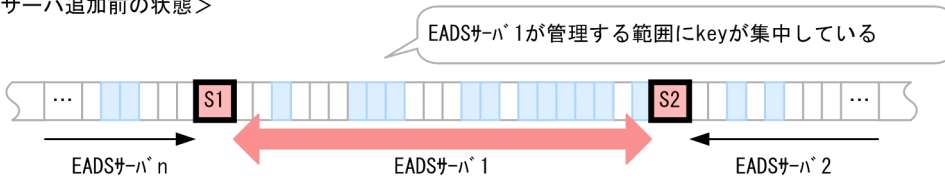
位置を指定しなければ、EADS サーバはクラスタ上の論理的な位置に均等に分散配置されます。

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。

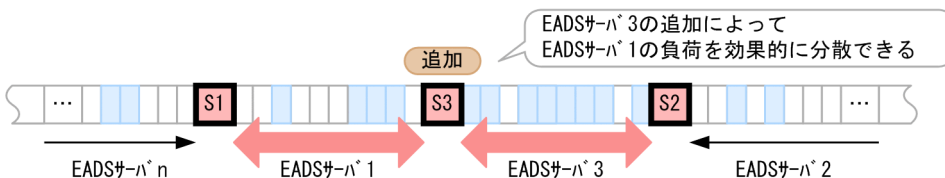
EADS サーバの位置を指定する場合は、統計情報（KeyCount, UsedMemorySize, および RequestCount）を確認し、位置を分析します。

図 7-2 EADS サーバの位置の設定

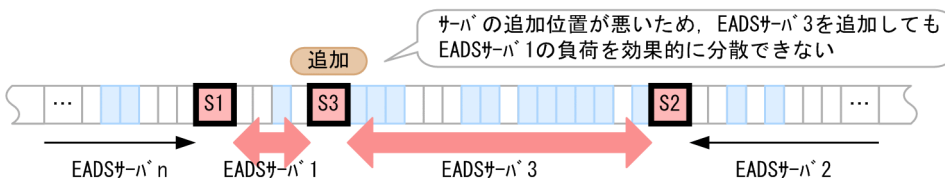
<サーバ追加前の状態>



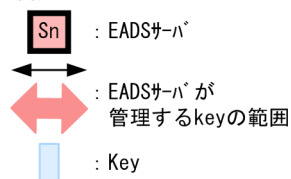
<サーバの追加位置が良い例>



<サーバの追加位置が悪い例>



(凡例)



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。

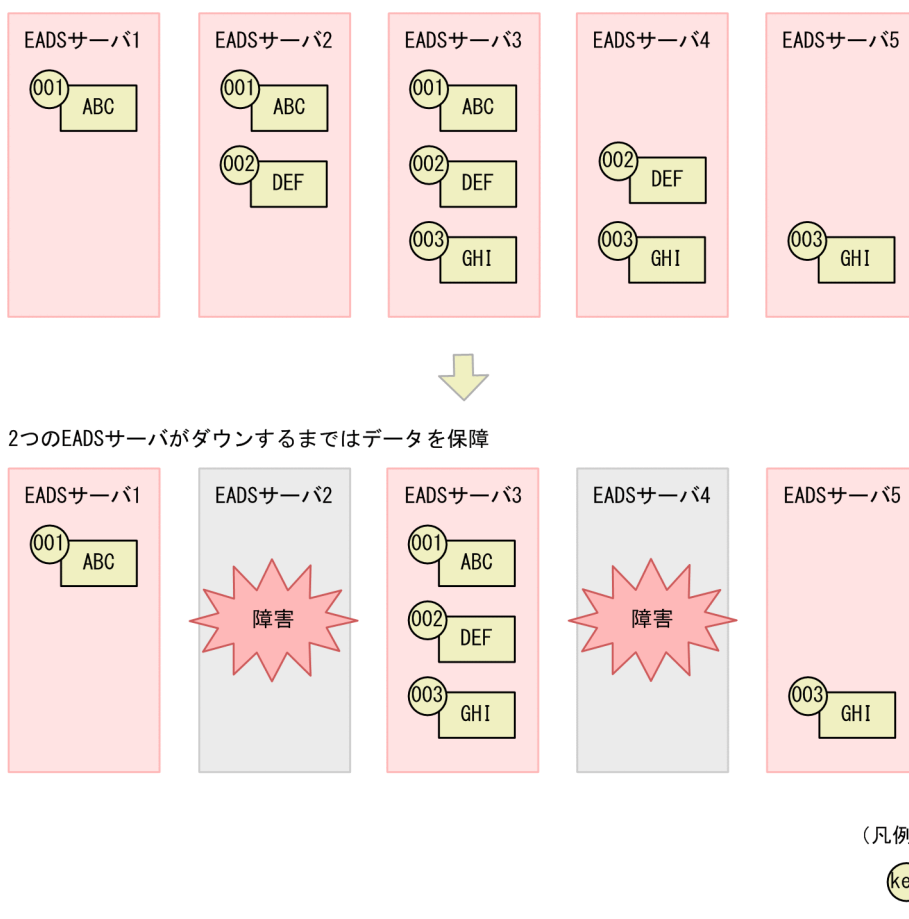
クラスタを構成する EADS サーバ数は、「データの多重度×2-1」以上であることが前提です。障害発生時にデータの多重度未満の数の EADS サーバがダウンしても、データの整合性は確保されます。そのため、多重度を上げると信頼性と耐障害性が向上します。

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	共通設定ファイル	<b>eads.failureDetector.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.failureDetector.heartbeat.port</b>	ポート番号	24630
6		<b>eads.replication.factor</b>	1～5	2

#### 注意事項

クラスタ定義ファイルおよび共通設定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。同一でない場合、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。

## (2) 共通設定ファイル

### (a) eads.failureDetector.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。

### (b) eads.failureDetector.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。

### (c) 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.admin.backup.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.admin.backup.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。

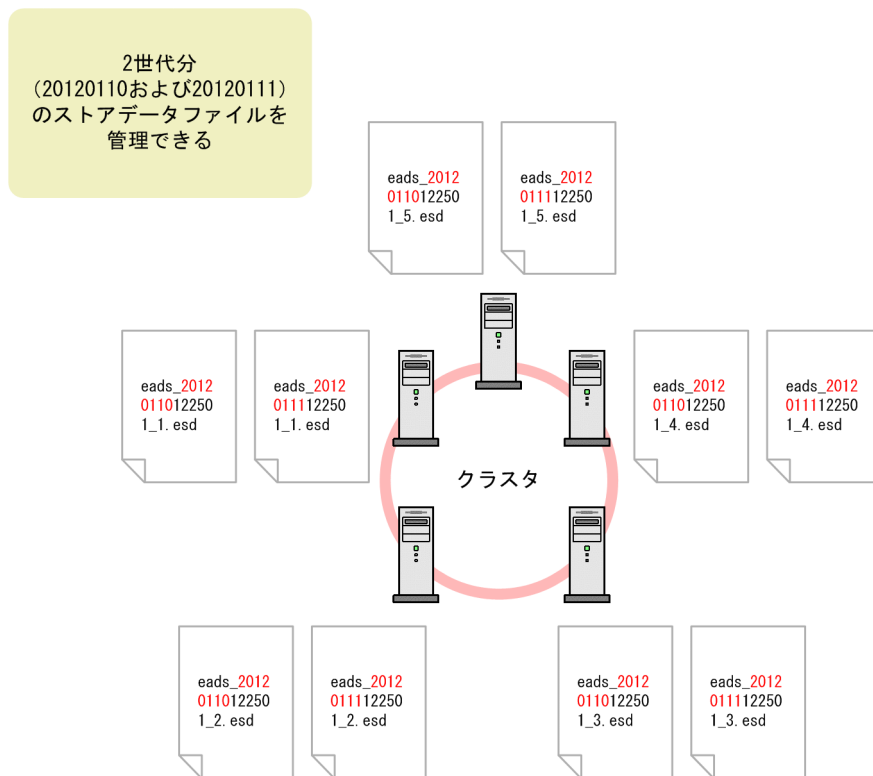
なお、出力先にすでに同じ名称のストアデータファイルが存在する場合は、上書きされないで `eztool export` コマンドがエラーとなります。

### (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、共通設定の `eads.admin.backup.exportCommand.generation.maxNum` 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、「11.7 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（共通設定の `eads.admin.backup.exportCommand.generation.maxNum` 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，共通設定の `eads.admin.backup.stopCommand.generation.maxNum` 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.admin.backup.dir</code>	パス名	<運用ディレクトリ>/store
2	共通設定ファイル	<code>eads.admin.backup.exportCommand.generation.maxNum</code>	0～32	2
3		<code>eads.admin.backup.stopCommand.generation.maxNum</code>	1～32	1

### (1) サーバ定義ファイル

#### (a) `eads.admin.backup.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) 共通設定ファイル

#### (a) `eads.admin.backup.exportCommand.generation.maxNum`

`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.admin.backup.stopCommand.generation.maxNum**

`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。
- `eads.cache.disk.persistentMemory.device`  
ディスクキャッシュの格納先として不揮発性メモリを使用する場合  
(`eads.cache.disk.persistentMemory.enable` パラメタに true を指定する場合) に必ず指定します。  
「[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。

## 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。

ディスクキャッシュの格納先には、不揮発性メモリも使用できます (eads.cache.disk.persistentMemory.enable パラメタで指定します)。詳細については、「7.7.2(4) ディスクキャッシュの格納先として不揮発性メモリを使用する」を参照してください。

### (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	eads_data_[EADS サーバ ID]_[キャッシュ名]_[レンジ ID]_[nnnnn].ecf

項番	キャッシュファイル種別	格納先	ファイル名
		ディレクトリ>/<キャッシュ名>	
		<キャッシュ定義の eads.cache.disk.persistentMemory.device パラメタで指定した不揮発性メモリのデバイスファイル>※1	—※2
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
		<キャッシュ定義の eads.cache.disk.persistentMemory.device パラメタで指定した不揮発性メモリのデバイスファイル>※1	—※2
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。データの格納先 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。

—：該当しません。

#### 注※1

ディスクキャッシュの格納先として不揮発性メモリを使用する場合の格納先です。

#### 注※2

ディスクキャッシュの格納先として不揮発性メモリを使用する場合、指定したデバイスファイルにキャッシュが作成されるため、キャッシュファイルは作成されません。

なお、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配置できるファイルシステムは、EADS がサポートしているファイルシステム（EADS が動作する OS がデフォルトで提供しているファイルシステム）です。

EADS が動作する OS と、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を配置できるファイルシステムを次の表に示します。

表 7-10 EADS が動作する OS と、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を配置できるファイルシステム

項番	EADS が動作する OS	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を配置できるファイルシステム
1	Red Hat Enterprise Linux 7 (64-bit x86_64)	xfs
2	Red Hat Enterprise Linux 8 (64-bit x86_64)	xfs

## ポイント

ディスクキャッシュの格納先として不揮発性メモリを使用する場合 (`eads.cache.disk.persistentMemory.enable` パラメタに `true` を指定する場合) は、不揮発性メモリに対応するデバイスファイルの領域に各ファイルが格納されます。そのため、OS が提供するファイルシステムは使用しません。

その代わりに、不揮発性メモリのデバイスファイルを使用する前に、不揮発性メモリを初期化する必要があります。不揮発性メモリのデバイスファイルの作成や初期化の手順については、リリースノートを参照してください。

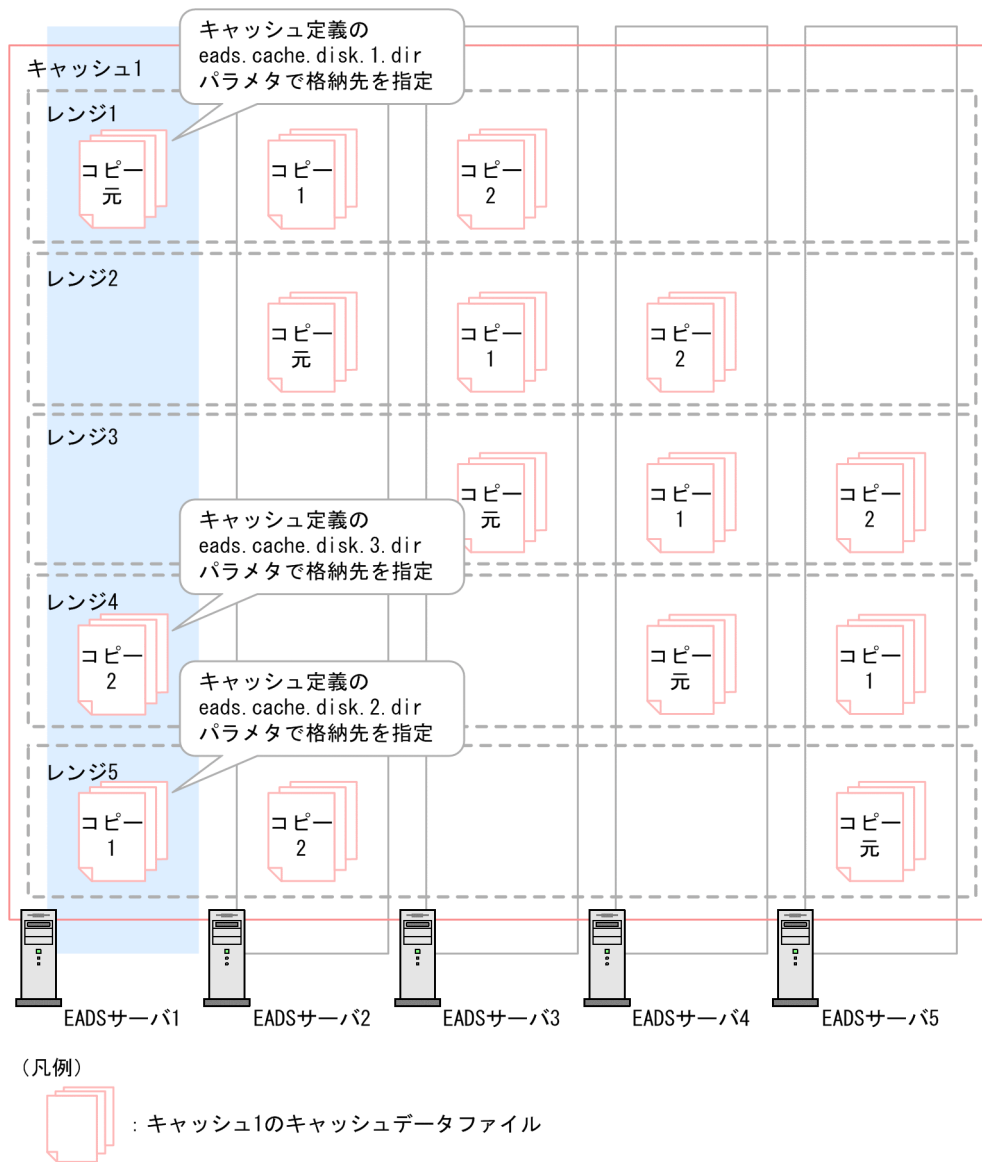
### (3) データを多重化している場合の、キャッシュデータファイルの格納先と EADS サーバの関係

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



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` パラメタで指定します。

`eads.cache.disk.<n>.dir` パラメタの指定例（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
```

## (4) ディスクキャッシュの格納先として不揮発性メモリを使用する

ディスクキャッシュの格納先として不揮発性メモリを使用することができます。不揮発性メモリを使用する場合は、キャッシュ定義の `eads.cache.disk.persistentMemory.enable` パラメタに `true` を指定します。

ディスクキャッシュの格納先として不揮発性メモリを使用すると、キャッシュファイル（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）は不揮発性メモリのデバイスファイルに格納されます。また、データアクセス処理も不揮発性メモリに最適化された方式になります。

キャッシュファイルの格納先となるデバイスファイルはキャッシュ定義の

`eads.cache.disk.persistentMemory.device` パラメタで指定します。1つのデバイスファイルに対して、対象となるキャッシュのコピー元のデータとコピーされたデータが格納されます。パラメタの指定例を次に示します。

`eads.cache.disk.persistentMemory.device` パラメタの指定例（キャッシュデータファイルを不揮発性メモリのデバイスファイルで管理する場合）

```
eads.cache.disk.persistentMemory.device=/dev/dax0.0
```

説明

図 7-5 の例の場合、`dax0.0` はコピー元のデータの格納先、レンジ5のデータ（コピー1）の格納先、レンジ4のデータ（コピー2）の格納先となります。

格納先のデバイスファイルに指定できるのは Device DAX モードで生成された character device だけです。Device DAX モードで character device を生成する方法については、リリースノートを参照してください。

### ポイント

不揮発性メモリのデバイスファイルを使用する前に、不揮発性メモリを初期化する必要があります。不揮発性メモリを初期化する手順については、リリースノートを参照してください。

## (a) 不揮発性メモリ内のファイルへのアクセス

ディスクキャッシュの格納先として不揮発性メモリを使用する場合、EADS が不揮発性メモリ内に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相当する領域を作成してアクセスします。そのため、OS から不揮発性メモリ内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は参照できません。キャッシュファイルの確認やバックアップなどを行う場合には、`eztool` コマンドを使用してください。

なお、不揮発性メモリのデバイスファイル内の各ファイルがログなどに出力される場合は、次の形式で出力されます。

形式：

<不揮発性メモリのデバイスファイルの絶対パス>/<キャッシュファイル名>

例：

/dev/dax0.0/eads\_data\_01\_cachename\_01\_00001.ecf

### 参考

不揮発性メモリのデバイスファイルに対する操作には、PMDK (Persistent Memory Development Kit) に含まれるツールとライブラリを使用します。詳細については、リリースノートを参照してください。

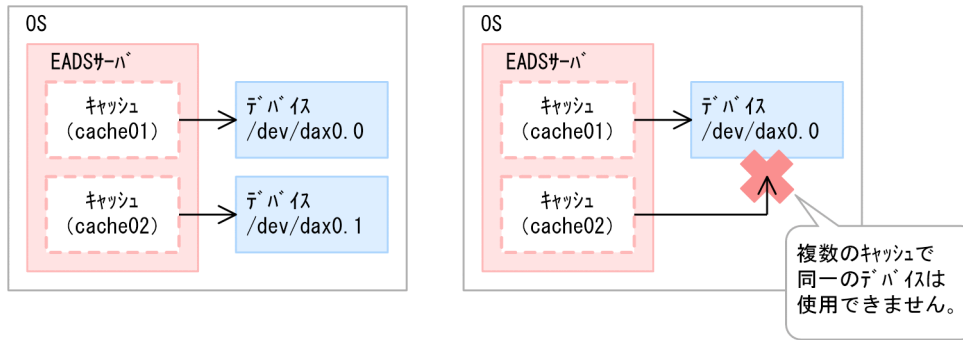
## (b) 不揮発性メモリのデバイスファイルの構成

キャッシュ定義の `eads.cache.disk.persistentMemory.device` パラメタで不揮発性メモリのデバイスファイルを指定する際は、1つのキャッシュにつき1つの不揮発性メモリのデバイスファイルを割り当ててください。また、対応するキャッシュの見積もり結果に従って、各デバイスファイルに容量を割り当ててください。

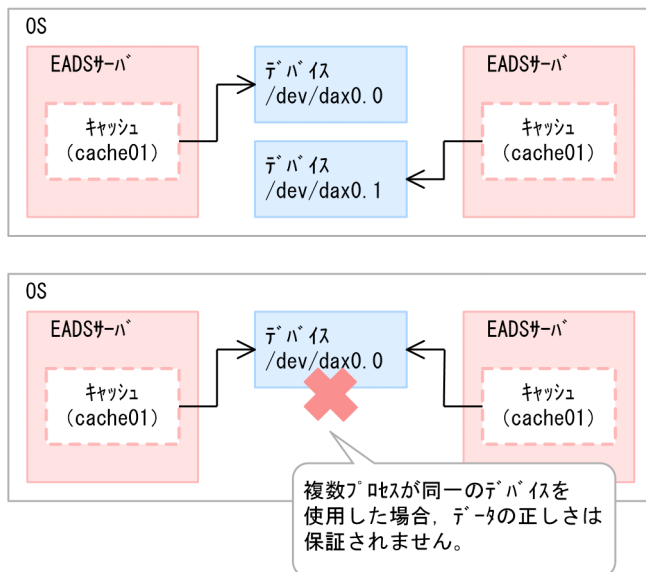
不揮発性メモリのデバイスファイルの構成を次の図に示します。

## 図 7-6 不揮発性メモリのデバイスファイルの構成

### ●複数キャッシュを作成する場合の不揮発性メモリのデバイスファイルの構成例



### ●同一OS上に複数EADSサーバを起動する場合の不揮発性メモリのデバイスファイルの構成例



EADS サーバ内の複数のキャッシュが同一の不揮発性メモリのデバイスファイルを使用することはできません。

同一 OS 上に複数の EADS サーバを配置する場合、EADS サーバ間で同一の不揮発性メモリのデバイスファイルを指定したときは、EADS サーバの動作およびデータは保証されません。同じように EADS サーバ以外の他プロセスが、EADS サーバが使用している不揮発性メモリのデバイスファイルを使用してデータを更新した場合も EADS サーバの動作およびデータは保証されません。

### 注意事項

EADS と他プロセスとの間で、デバイスファイルの排他は行われません。他 EADS サーバと他プログラムが同一のデバイスファイルを指定した場合、データが破壊されるおそれがあるので注意してください。

## (c) 不揮発性メモリを使用する場合の注意事項

ディスクキャッシュの格納先として不揮発性メモリを使用する場合の注意事項を次に示します。

- ディスクキャッシュの格納先として、高性能 SSD と不揮発性メモリを同時に使用することはできません。また、2Way キャッシュの格納先として不揮発性メモリは使用できません。
- 次の場合はキャッシュが作成できません。
  - Device DAX 以外のモードで生成されたデバイスファイルが指定された場合
  - 指定したデバイスファイルに割り当てられているサイズが、格納す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総容量よりも小さい場合
  - キャッシュ作成時に指定したデバイスファイルが、同一 EADS サーバ内の作成済みキャッシュを格納しているデバイスファイルと重複した場合
- 不揮発性メモリにミラーリングなどの冗長化機能がない場合は、不揮発性メモリが故障した際のデータの消失を防ぐため、次のような運用にすることを推奨します。
  - 共通設定の `eads.replication.factor` パラメタの値を 2 以上にして、複数の不揮発性メモリ上にデータを複製する構成にする。
  - 定期的に `eztool exportecf` コマンドを実行してバックアップを取得する。
- OS のマシントラップ機能の `ToleranceLevel` には 0 を設定してください。この設定をすると、不揮発性メモリに発生した不良ブロックにアクセスしたときに、破損データによって処理が継続することを OS が再起動することで防止できます。

ただし、OS 起動時に EADS サーバを自動で復旧する設定にしている場合、OS の起動時に毎回不良ブロックにアクセスして、再起動を繰り返すおそれがあります。そのため、EADS サーバを自動で復旧する設定にしている場合は、復旧を試みる回数に上限を設けたり、復旧処理前に不良ブロックが発生していないかを確認してから EADS サーバを起動したりするように設計してください。

### 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タ

キャッシュの運用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 7-11 キャッシュの運用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	<b><code>eads.cache.disk.getError.isolate.enable</code></b>	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	true

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
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> <li>Disk</li> <li>2Way</li> </ul>	なし
3		<code>eads.cache.eviction.policy</code>	<ul style="list-style-type: none"> <li>FIFO</li> <li>LRU</li> <li>None</li> </ul>	None
4		<code>eads.cache.eviction.keyCount</code>	1024~ 1073741823	なし
5		<code>eads.cache.disk.info.dir</code>	パス名	<運用ディレクトリ>/store
6		<code>eads.cache.disk.&lt;n&gt;.dir*</code>	パス名	なし
7		<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
8		<code>eads.cache.disk.filenum</code>	ファイル数 (8~ 32768)	なし
9		<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
10		<code>eads.cache.disk.transfer.interval</code>	0~60000 (ミリ秒)	1000 ミリ秒
11		<code>eads.cache.disk.transfer.datasize</code>	10240 バイト~ 2147483647 バイト	102400 バイト
12		<code>eads.cache.disk.highSpecDrive.enable</code>	<ul style="list-style-type: none"> <li>true</li> <li>false</li> </ul>	false
13		<code>eads.cache.disk.persistentMemory.enable</code>	<ul style="list-style-type: none"> <li>true</li> <li>false</li> </ul>	false
14		<code>eads.cache.disk.persistentMemory.device*</code>	パス名	なし
15		<code>eads.cache.disk.persistentMemory.blocksize</code>	<ul style="list-style-type: none"> <li>256 バイト</li> <li>512 バイト</li> <li>1024 バイト</li> </ul>	256 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.eviction.policy`
- `eads.cache.eviction.keyCount` (データ自動削除機能を使用する場合)
- `eads.cache.disk.filesize`
- `eads.cache.disk.filenum`
- `eads.cache.disk.blocksize`
- `eads.cache.disk.persistentMemory.enable`
- `eads.cache.disk.persistentMemory.blocksize` (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)

### 注※

ディスクキャッシュの格納先として不揮発性メモリを使用しない場合は `eads.cache.disk.<n>.dir` パラメタが、使用する場合は `eads.cache.disk.persistentMemory.device` パラメタが必ず指定するパラメタになります。

## (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`, `getAll`)
- Group インタフェースの次に示すメソッド
  - `getLastUpdateTime()`
  - `getValueUsageSize()`
- Store インタフェースの次に示すメソッド
  - `getLastUpdateTime()`

- getEHeapUsageSize()
- getDiskUsageSize()
- 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.eviction.policy

データの自動削除ポリシーを指定します。データ自動削除機能を使用して、しきい値を超えたデータを削除する場合の動作を指定します。

FIFO :

データの更新順が古いデータから自動的に削除します。

LRU :

データの更新順または参照順が古いデータから自動的に削除します。

None :

データを自動で削除しません (データ自動削除機能を使用しません)。



## 注意事項

- データ総量監視機能が無効（共通設定ファイルの `eads.cache.limiter.enable` パラメタに `false` が指定されている）な場合、データ自動削除機能を使用できません。そのため、`None` 以外を指定できません。
- `eads.cache.type` パラメタに `2Way` が指定されている場合、データ自動削除機能を使用できません。そのため、`None` 以外を指定できません。

### (c) `eads.cache.eviction.keyCount`

データ自動削除機能で、データを削除するためのしきい値として使用する 1 レンジ当たりのデータ件数を指定します。

このパラメタの指定値を超えるデータは格納されません（自動的に削除されます）。

## 注意事項

`eads.cache.eviction.policy` パラメタに `None` が指定された場合、このパラメタに指定された値は無視されます。

### (d) `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（スラッシュ（/））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。

なお、ディスクキャッシュの格納先として不揮発性メモリを使用する場合（`eads.cache.disk.persistentMemory.enable` パラメタに `true` を指定している場合）、キャッシュインデックスファイルは、`eads.cache.disk.persistentMemory.device` パラメタで指定したデバイスファイルに作成されます。

## 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。

### (e) `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（スラッシュ（`/`））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。
- ディスクキャッシュの格納先として不揮発性メモリを使用する場合（`eads.cache.disk.persistentMemory.enable` パラメタに `true` を指定する場合）、このパラメタの指定値は無視されます。

## (f) `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。

## (g) `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。

## (h) `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。

## 注意事項

ディスクキャッシュの格納先として不揮発性メモリを使用する場合（`eads.cache.disk.persistentMemory.enable` パラメタに `true` を指定する場合）、1 回で書き込む

データサイズはこのパラメタではなく、`eads.cache.disk.persistentMemory.blocksize` パラメタで指定した値が有効になります。

## 参考

ディスクキャッシュの格納先として不揮発性メモリを使用する場合でも、このパラメタに不正な値が指定されているとエラーになります。ディスクキャッシュの格納先として不揮発性メモリを使用する場合には、このパラメタをキャッシュ定義ファイル上でコメントアウトしておくことを推奨します。

### (i) `eads.cache.disk.transfer.interval`

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。

復旧処理では、復旧対象の EADS サーバでこのパラメタに指定した値が適用されます。

このパラメタの指定値とキャッシュ定義の `eads.cache.disk.transfer.datasize` 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.transfer.datasize` 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.transfer.interval` 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.transfer.datasize` 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

$$\begin{aligned} & \text{復旧処理のデータ送信に使用できる帯域 (単位: bps)} \geq \\ & (\text{復旧処理で送信するデータサイズ (単位: バイト)} \times 8) \\ & \div \{ (\text{復旧処理でのデータ送信間隔 (単位: ミリ秒)} \\ & + \text{データ送信に掛かる時間 (単位: ミリ秒)}) \div 1,000 \} \end{aligned}$$

復旧処理で送信するデータサイズ (単位: バイト):

キャッシュ定義の `eads.cache.disk.transfer.datasize` 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):

キャッシュ定義の `eads.cache.disk.transfer.interval` 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.disk.transfer.datasize` 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.transfer.interval` 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。

### (j) `eads.cache.disk.transfer.datasize`

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、復旧対象の EADS サーバでこのパラメタに指定した値が適用されます。

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。復旧対象の EADS サーバの `eads.cache.disk.transfer.interval` 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.transfer.datasize` 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。

### (k) `eads.cache.disk.highSpecDrive.enable`

高性能 SSD に対応したファイルアクセスをするかどうかを指定します。

このパラメタに `true` を指定した場合、高性能 SSD に対応した次のファイルアクセスを行います。

1. キャッシュデータファイルを DirectIO モードで開き、当該キャッシュデータファイルのページキャッシュを介さないで、ファイルにアクセスします。
2. キャッシュデータファイルの作成直後に全データを 0 で埋め尽くす（ゼロフィル）処理を実行し、データ更新時の書き込み回数と書き込みデータ量を削減します。

## 注意事項

- 高性能 SSD 以外（HDD などの記憶装置）では、`true` を指定しないでください。
- このパラメタに `true` を指定した場合は、追加でゼロフィル処理を行うため、`false` を指定した場合に比べて、次に示すコマンドの処理時間が長くなるおそれがあります。
  - `eztool createcache`（キャッシュの作成）
  - `eztool resume`（キャッシュの再開）

そのため、このパラメタに `true` を指定する場合は、`eztool` コマンドの `-t` オプション（コマンドのタイムアウト時間）を当該コマンドの処理時間よりも長くなるように設定することを推奨します。

- このパラメタに `true` を指定した場合は、追加でゼロフィル処理を行うため、`false` を指定した場合に比べて、次に示すコマンドの処理時間が長くなるおそれがあります。
  - `eztool compaction`（キャッシュデータファイルのコンパクション）
  - `ezstart -r`, `ezserver -r`（EADS サーバのクラスタへの復旧）

そのため、このパラメタに `true` を指定する場合は、当該コマンドの処理が想定時間内に終わることを事前に確認することを推奨します。

- このパラメタに `true` を指定した場合、キャッシュデータファイルへの書き込み回数が、`false` を指定した場合（1 回）に比べて、1 回増える（2 回になる）ため、高性能 SSD の寿命を計算する際には考慮してください。
- このパラメタと `eads.cache.disk.persistentMemory.enable` パラメタに、同時に `true` は指定できません。

## (l) eads.cache.disk.persistentMemory.enable

ディスクキャッシュの格納先として、不揮発性メモリを使用するかどうかを指定します。

不揮発性メモリはディスクキャッシュでだけ使用できます。また、高性能 SSD に対応したファイルアクセスとは同時に使用できません。このため、このパラメタに true を指定する場合には次の条件を満たす必要があります。

- `eads.cache.type` パラメタに Disk を指定します。
- `eads.cache.disk.highSpecDrive.enable` パラメタに false を指定します。

## (m) eads.cache.disk.persistentMemory.device

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格納先となる不揮発性メモリのデバイスファイルを指定します。

不揮発性メモリのデバイスファイルには、最大 200 文字の半角英数字 (0~9, A~Z, a~z), アンダースコア (\_), ピリオド (.), およびセパレータ (スラッシュ (/)) が指定できます。

このパラメタに指定したデバイスファイル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

- `eads.cache.disk.persistentMemory.enable` パラメタに false を指定している場合、このパラメタに指定された値は無視されます。
- 同一 EADS サーバ内で、このパラメタの指定値の絶対パスと、他キャッシュ定義ファイルの、この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。
- 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、他の EADS サーバが使用している不揮発性メモリのデバイスファイルは、このパラメタに指定できません。

## (n) eads.cache.disk.persistentMemory.blocksize

不揮発性メモリ上の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: バイト) を指定します。

このパラメタには、2 の累乗数 (256, 512, 1024) が指定できます。

`eads.cache.disk.persistentMemory.enable` パラメタに true を指定している場合、1 回で書き込むデータサイズは `eads.cache.disk.blocksize` パラメタではなく、このパラメタで指定した値が有効になります。

## 注意事項

- `eads.cache.disk.persistentMemory.enable` パラメタに `false` を指定している場合、このパラメタに指定された値は無視されます。
- 使用する不揮発性メモリに合わせて値を指定してください。詳細については、リリースノートを参照してください。使用する不揮発性メモリに関する記載がリリースノートにない場合は、キャッシュ定義ファイル上でこのパラメタをコメントアウトしてください。



## 7.8 ノンブロッキング I/O 通信制御に依存するパラメタの設計

ノンブロッキング I/O 通信制御に依存するパラメタを設計します。

### 7.8.1 ノンブロッキング I/O 通信制御の設定

今までの通信制御（ブロッキング I/O 通信制御）の場合、EADS クライアント・EADS サーバ間の最大同時接続数は 1,024 以下です。そのため、EADS クライアント・EADS サーバ間の最大同時接続数が 1,025 以上の場合は、ノンブロッキング I/O 通信制御を使用することを検討してください。ノンブロッキング I/O 通信制御を使用する場合は、`eads.server.nonBlocking.enable` パラメタに `true` を指定する必要があります。

ただし、EADS クライアント・EADS サーバ間の最大同時接続数が 1,024 以下の場合は、ノンブロッキング I/O 通信制御を使用しないでください。処理性能が低下するため、ノンブロッキング I/O 通信制御の効果が期待できません。

#### 注意事項

- 一部の EADS サーバだけでノンブロッキング I/O 通信制御を使用するような運用はできません。クラスタ内のすべての EADS サーバで、必ず `eads.server.nonBlocking.enable` パラメタの指定を一致させてください。
- ノンブロッキング I/O 通信制御とイベントリスナは同時に使用しないでください。同時に使用した場合、動作は保証されません。
- ノンブロッキング I/O 通信制御と `eztool isolate` コマンドをあわせて使用する際、同時に実行されるキャッシュ操作およびファンクション操作リクエスト数の最大値が 1,024 を超える場合は、EADS クライアントでコネクションエラーが発生するおそれがあります。そのため、`eads.admin.operation.isolate.gracefulstop.waitTime` パラメタの指定値には、十分大きな値を指定してください。EADS サーバのクラスタ構成情報の更新が完了してから縮退処理が完了するまでの時間が長くなっても問題がないようにしてください。

また、それに伴って `eztool isolate` コマンドの実行時間も延びます。そのため、プロパティやコマンドオプションを使用して、`eztool isolate` コマンドのタイムアウト時間が `eads.admin.operation.isolate.gracefulstop.waitTime` パラメタの指定値よりも大きな値になるように指定してください。

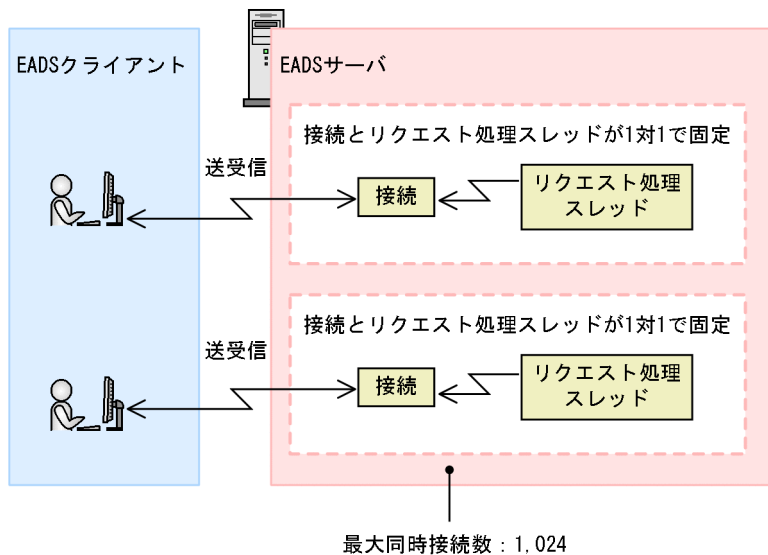
ブロッキング I/O 通信制御とノンブロッキング I/O 通信制御の違いについて、次で説明します。

#### (1) ブロッキング I/O 通信制御

今までの通信制御である、ブロッキング I/O 通信制御を使用した EADS クライアント・EADS サーバ間の通信制御を次の図に示します。



図 7-7 ブロッキング I/O 通信制御の概要

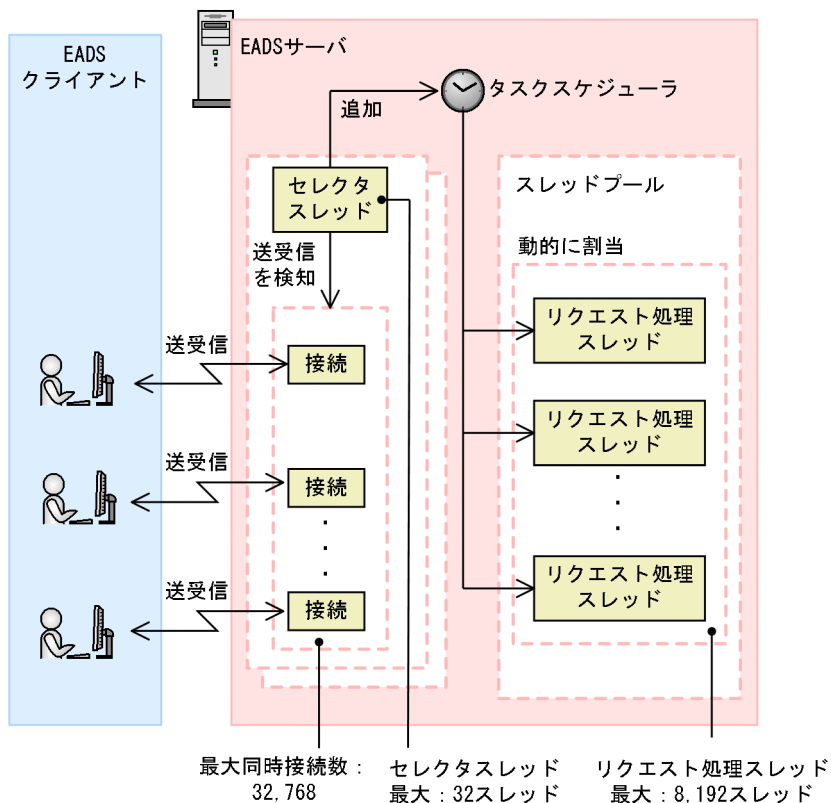


ブロッキング I/O 通信制御では、リクエストを処理する間はそのリクエストを処理するスレッドが占有されるため、EADS サーバは最大同時接続数と同じ数のリクエスト処理スレッドを生成します。

## (2) ノンブロッキング I/O 通信制御

ノンブロッキング I/O 通信制御を使用した EADS クライアント・EADS サーバ間の通信制御を次の図に示します。

図 7-8 ノンブロッキング I/O 通信制御の概要



ノンブロッキング I/O 通信制御は、リクエストを処理するスレッドが占有されないため、EADS サーバは最大同時接続数とリクエスト処理スレッド数が同数である必要はありません。これを実現するため、EADS サーバは複数の接続を一括監視して送受信可能になった接続を検知するセクタスレッドを 1 個以上生成します。

また、キャッシュおよびファンクションの同時実行スレッド数の制御はタスクスケジューラで行われます。

## 7.8.2 滞留中 I/O データ流量制御

滞留中 I/O データ流量制御とは、ノンブロッキング I/O 通信制御を使用してサーバ定義の `eads.server.connection.bufferSize` パラメタの指定値よりも大きなデータサイズの通信を行う場合に、EADS サーバはそのデータの送受信が完了するまでの間はそのリクエストを滞留させて、ほかのリクエストの処理を実行する制御のことです。

リクエストを滞留させる際、そのデータ（滞留中 I/O データ）を FullGC の対象外にするため、EADS サーバは Explicit ヒープ中に確保したノンブロッキング I/O 通信バッファをそのリクエストに一時的に割り当てて、滞留中 I/O データを格納します。

しかし、大量のリクエストが同時に滞留することによって、ノンブロッキング I/O 通信バッファが枯渇するおそれがあります。

そこで、滞留中 I/O データ流量制御によって、ノンブロッキング I/O 通信バッファを割り当てるリクエストの最大同時割当数を制御することで、滞留中 I/O データの総量が大きくなることを防ぎます。

## 7.8.3 ノンブロッキング I/O 通信制御に依存するパラメタ

ノンブロッキング I/O 通信制御に依存するパラメタの一覧を次の表に示します。

表 7-12 ノンブロッキング I/O 通信制御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.server.nonBlocking.enable</code>	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	false
2		<code>eads.server.nonBlocking.processorThreads</code>	1~8192	10
3		<code>eads.server.nonBlocking.selectorThreads</code>	1~32	1
4		<code>eads.server.nonBlocking.maxAllocateConnections</code>	1~32768	10
5		<code>eads.statistics.nonBlocking.enable</code>	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	true
6	共通設定ファイル	<code>eads.server.nonBlocking.external.heapsize</code>	1~268435456	30

## (1) サーバ定義ファイル

### (a) `eads.server.nonBlocking.enable`

ノンブロッキング I/O 通信制御を有効にするかどうかを指定します。

### (b) `eads.server.nonBlocking.processorThreads`

ノンブロッキング I/O 通信制御を使用する際の、リクエスト処理スレッド数を指定します。

#### 注意事項

- `eads.server.nonBlocking.enable` パラメタに `false` を指定している場合、このパラメタに指定された値は無視されます。
- `eads.server.nonBlocking.enable` パラメタの指定値が `true` の場合、かつ `eads.server.maxConnections` パラメタの指定値よりもこの `eads.server.nonBlocking.processorThreads` パラメタの指定値が大きい場合は、このパラメタにはデフォルト値が設定されます。

### (c) `eads.server.nonBlocking.selectorThreads`

ノンブロッキング I/O 通信制御を使用する際の、セレクタスレッド数を指定します。

#### 注意事項

`eads.server.nonBlocking.enable` パラメタに `false` を指定している場合、このパラメタに指定された値は無視されます。

### (d) `eads.server.nonBlocking.maxAllocateConnections`

ノンブロッキング I/O 通信制御を使用する際の、最大同時割当数を指定します。滞留中 I/O データ流量制御では、ノンブロッキング I/O 通信バッファを同時に割り当てるリクエスト数の上限です。

#### 注意事項

`eads.server.nonBlocking.enable` パラメタに `false` を指定している場合、このパラメタに指定された値は無視されます。

### (e) `eads.statistics.nonBlocking.enable`

ノンブロッキング I/O 通信制御に関する統計情報を出力する場合に指定します。

## 注意事項

`eads.server.nonBlocking.enable` パラメタに `false` を指定している場合、このパラメタに指定された値は無視され、ノンブロッキング I/O 通信制御に関する統計情報は出力されません。

## (2) 共通設定ファイル

### (a) `eads.server.nonBlocking.external.heapsize`

ノンブロッキング I/O 通信バッファサイズ (単位: メガバイト) を指定します。

## 注意事項

`eads.server.nonBlocking.enable` パラメタに `false` を指定している場合、このパラメタに指定された値は無視されます。

## 7.9 スケールアウト処理・リバランス処理に依存するパラメタの設計

スケールアウト処理・リバランス処理に依存するパラメタを設計します。

### 7.9.1 スケールアウト処理・リバランス処理に依存するパラメタ

スケールアウト処理・リバランス処理に依存するパラメタの一覧を次の表に示します。

表 7-13 スケールアウト処理・リバランス処理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.scaling.moveDataWithUpdateRequest.enable</code>	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	true
2		<code>eads.scaling.moveData.interval</code>	0~60000	0
3	共通設定ファイル	<code>eads.scaling.requestQueuing.enable</code>	<ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>	true
4		<code>eads.scaling.eviction.targetKeyCount</code>	1024~1073741824	8192

#### (1) サーバ定義ファイル

##### (a) `eads.scaling.moveDataWithUpdateRequest.enable`

スケールアウト処理中またはリバランス処理中の更新リクエスト実行時に、データを移動させるかどうかを指定します。

true :

データを移動させます。

false :

データを移動させません。

#### ポイント

- 05-30 より前のバージョンと同じ動作にしたい場合は、true を指定してください。
- スケールアウト処理の開始直後、またはリバランス処理中のリクエストの応答に一時的に時間が掛かる場合、それを回避したいときは false を指定してください。
- スケールアウト処理中、またはリバランス処理中にリクエストが実行されない場合は、false を指定してください。

## (b) eads.scaling.moveData.interval

スケールアウト処理中、またはリバランス処理中のデータの移動間隔（単位：ミリ秒）を指定します。

このパラメタに0以外を指定すると、スケールアウト処理、またはデータ自動削除機能を使用するキャッシュが存在しないときのリバランス処理の実行完了までの時間が、最大で次に示す時間分だけ従来よりも増加する場合があります。

- スケールアウト処理の場合

$$\begin{aligned} & \text{分割元レンジでのデータ移動回数} \times \text{このパラメタの指定値 (単位: ミリ秒)} + \text{リクエスト増加数分の処理時間}^{*1} \\ & \text{分割元レンジでのデータ移動回数} \div \\ & \quad \Sigma (\text{キャッシュ単位でのデータ移動件数} \div 1,000) \text{ }^{*2} \end{aligned}$$

### 注※1

スケールアウト処理の実行完了までの時間が長くなることで、スケールアウト処理中のリクエスト数が従来よりも増加する場合は、増加数分のリクエストの処理時間を指します。

### 注※2

小数点以下の値は切り上げてください。

- リバランス処理の場合

$$\begin{aligned} & (\text{移動元レンジでのデータ移動回数} + \text{移動先レンジでのデータ移動回数}) \\ & \times \text{このパラメタの指定値 (単位: ミリ秒)} + \text{リクエスト増加数分の処理時間}^{*1} \\ & \text{移動元レンジでのデータ移動回数} \div \\ & \quad \Sigma (\text{キャッシュ単位でのデータ移動件数} \div 1,000) \text{ }^{*2} \\ & \text{移動先レンジでのデータ移動回数} \div \\ & \quad \Sigma (\text{キャッシュ単位でのデータ移動件数} \div 1,000) \text{ }^{*2} \end{aligned}$$

### 注※1

リバランス処理の実行完了までの時間が長くなることで、リバランス処理中のリクエスト数が従来よりも増加する場合は、増加数分のリクエストの処理時間を指します。

### 注※2

小数点以下の値は切り上げてください。

スケールアウト処理、またはリバランス処理の実行完了までの時間の増加を回避するためには、次のように値を指定してください。

- 運用によって、スケールアウト処理、またはデータ自動削除機能を使用しない場合のリバランス処理の実行時間の上限があるときは、その時間内に収まるように指定値を小さくする
- スケールアウト処理中、またはデータ自動削除機能を使用しない場合のリバランス処理中のリクエストの応答時間の上限があるときは、その時間内に収まるように指定値を調整する

## ポイント

- 05-30 より前のバージョンと同じ動作にしたい場合は、0 を指定してください。
- スケールアウト処理の開始直後のリクエストの応答、またはデータ自動削除機能を使用するキャッシュが存在しないときのリバランス処理中のリクエストの応答に一時的に時間が掛かる場合、それを回避したいときは0 以外を指定してください。
- スケールアウト処理中、またはデータ自動削除機能を使用するキャッシュが存在しないときのリバランス処理中にリクエストが実行されない場合は、0 を指定してください。
- データ自動削除機能を使用するキャッシュが存在するときのリバランス処理の場合は、指定した値に関係なく、0 が指定されたときと同じ動作になります。

## (2) 共通設定ファイル

### (a) `eads.scaling.requestQueuing.enable`

スケールアウト処理において、分割前のレンジを管理していた EADS サーバから、追加された EADS サーバに追加されたレンジの範囲のデータを転送する間に、分割前のレンジを管理していた EADS サーバで受け付けたリクエストを一時的に待機させることをリクエスト・キューイングといいます。

このパラメタでは、スケールアウト処理中に、受付済みで未実行のリクエストをキューイングするかエラーにするかを指定します。

true :

受付済みで未実行のリクエストをキューイングします。

false :

受付済みで未実行のリクエストをエラーにします。

EADS クライアントにはエラーコード 4120 が返却されます。

このパラメタを指定する場合は、次の説明も参照してください。

- 「14.2.1 `ezstart` (EADS サーバの開始)」の「(6) 注意事項」
- 「14.2.2 `ezserver` (EADS サーバのフォアグラウンドでの開始)」の「(6) 注意事項」

## ポイント

- 05-30 より前のバージョンと同じ動作にしたい場合は、true を指定してください。

### (b) `eads.scaling.eviction.targetKeyCount`

データ自動削除機能を使用するキャッシュが存在する場合の、リバランス処理の移動レンジのキー数を指定します。

## ポイント

- 次のどちらかに該当する場合は、1073741824 を指定してください。
  - ・05-30 より前のバージョンと同じ動作にしたい場合
  - ・データ自動削除機能を使用するキャッシュが存在するときのリバランス処理中にリクエストが実行されない場合
- データ自動削除機能を使用するキャッシュが存在しない場合は、指定値は使用されません。
- データ自動削除機能を使用するキャッシュが存在するときのリバランス処理中のリクエストの応答に一時的に時間が掛かる場合は、1073741824 未満の値を指定してください。
- 運用上、リバランス処理の実行時間に上限がある場合、実行時間の上限を超えないようにこのパラメタの指定値を大きくしてください。



# 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、「21.1.2(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.client.logger.message.console.enable=true  
eads.client.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。

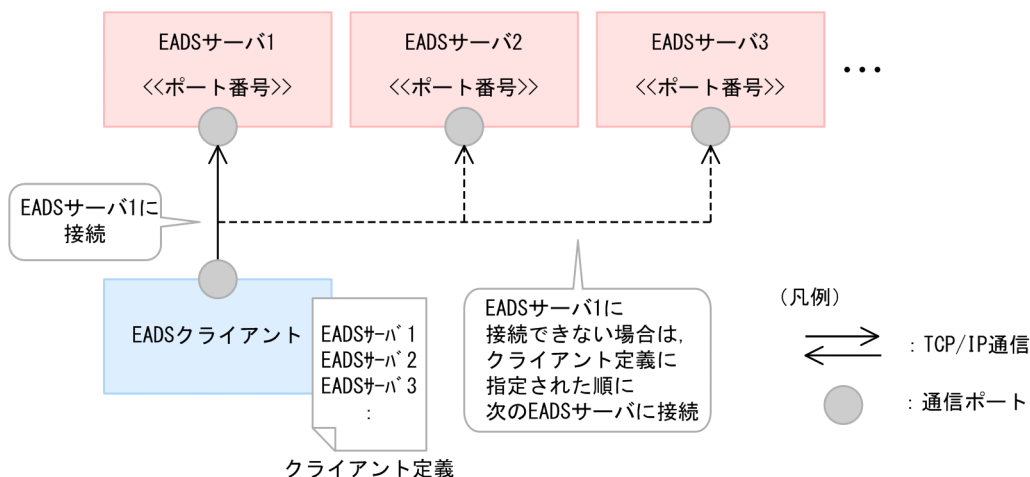
### 8.3.1 接続先 EADS サーバ, IP アドレスまたはホスト名, およびポート番号の設定

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.connect.servers` 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.connect.servers</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.connect.servers`

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。

#### 注意事項

コンマ区切りで複数の名称を指定した場合、空文字列の名称が存在するときは AP で使用する言語によって処理が異なります。

- EADS クライアント（Java）では、空文字列の名称は無視して処理が続行されます。
- EADS クライアント（C）では、不正な定義としてエラーになります。

### (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.connect.servers` パラメタと対で指定します。接続先 EADS サーバには、`eads.client.connect.servers` 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.connect.servers` パラメタと対で指定します。接続先 EADS サーバには、  
`eads.client.connect.servers` 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.connect.servers` パラメタの指定例

`eads.client.connect.servers` パラメタの指定例を次に示します。

```
eads.client.connect.servers=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	保守ログファイル	システムが使用するファイルです。

### 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.client.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
保守ログファイル	< eads.client.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しないでください。ネットワークで接続された別マシンのディレクトリを指定した場合、動作は保証されません。

### 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。

パラメタの指定値を変更する場合の注意事項を次に示します。

- ログファイルのファイル数またはファイルサイズにデフォルト値より小さい値を設定すると、当然蓄積されるログが少なくなります。そのため、障害発生時の調査に影響することがあります。ログファイルのパラメタの変更は、そのことに留意して実施してください。
- ログファイルのファイル数は1または3以上を設定してください。
- ディスク容量を節約したい場合は、ログファイルのファイルサイズを小さく変更することで対応してください。
- ログファイルのファイル数は3以上、かつすべてのログファイルの合計サイズが16メガバイト以上になるように設定してください。

### 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.client.logger.dir</b>	パス名	なし
2		<b>eads.client.logger.message.filesize</b>	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	1048576 バイト
3		<b>eads.client.logger.message.filenum</b>	ファイル数 (1～ 16)	2
4		<b>eads.client.logger.message.console.enable</b>	• true • false	false
5		<b>eads.client.logger.initErrorOut</b>	• true • false	false
6		<b>eads.client.logger.maintenance.filesize</b>	ファイルサイズ (4096 バイト～ 2147483647 バイ ト)	16777216 バイト



項番	定義ファイル	パラメタ名	指定する値	デフォルト値
7		<code>eads.client.logger.maintenance.filenum</code>	ファイル数 (1~16)	16

## (1) クライアント定義ファイル

### (a) `eads.client.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.client.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.client.logger.message.filenum`

メッセージログのファイル数を指定します。

### (d) `eads.client.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.client.logger.initErrorOut`

ロガーの初期化失敗時に、標準エラー出力にエラーメッセージを出力するかどうかを指定します。

true :

標準出力にエラーメッセージを出力します。

false :

標準出力にエラーメッセージを出力しません。

### (f) `eads.client.logger.maintenance.filesize`

保守ログの 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。

## (g) eads.client.logger.maintenance.filenum

保守ログのファイル数を指定します。

# 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。

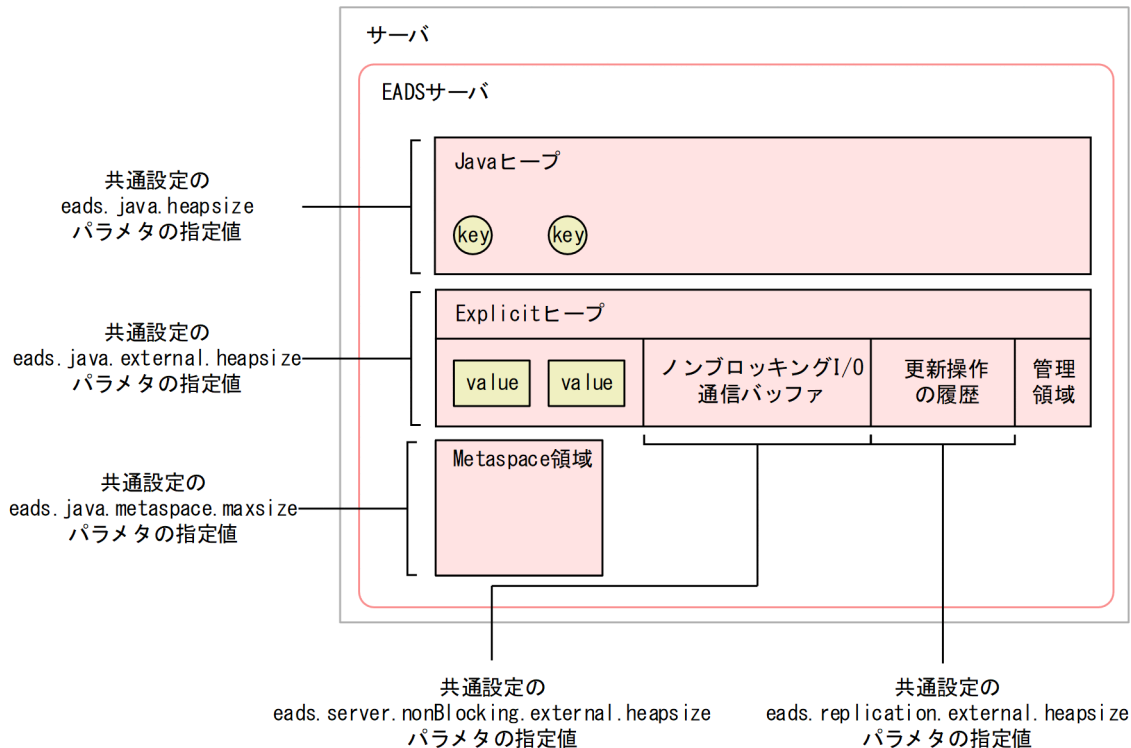
## 9.1 メモリ・バッファに関連するパラメタの設計

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。

図 9-1 メモリ構成



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
- ノンブロッキング I/O 通信バッファが格納される領域

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する際、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	...

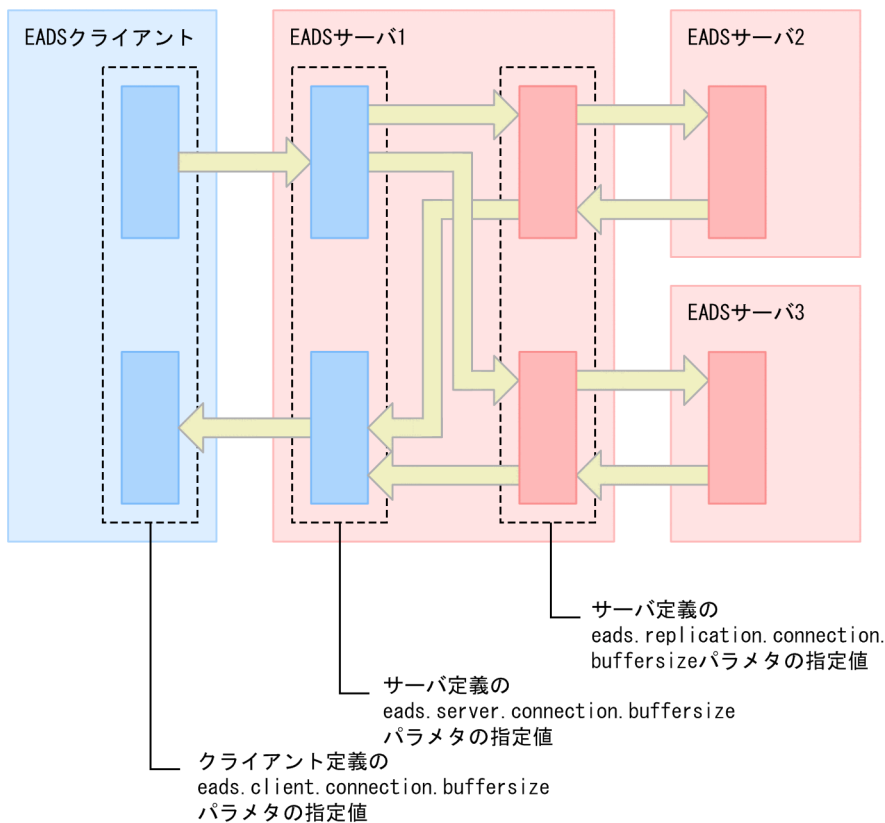
2番目の更新操作が合意され、  
書き込みが完了されるまで保持

### 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

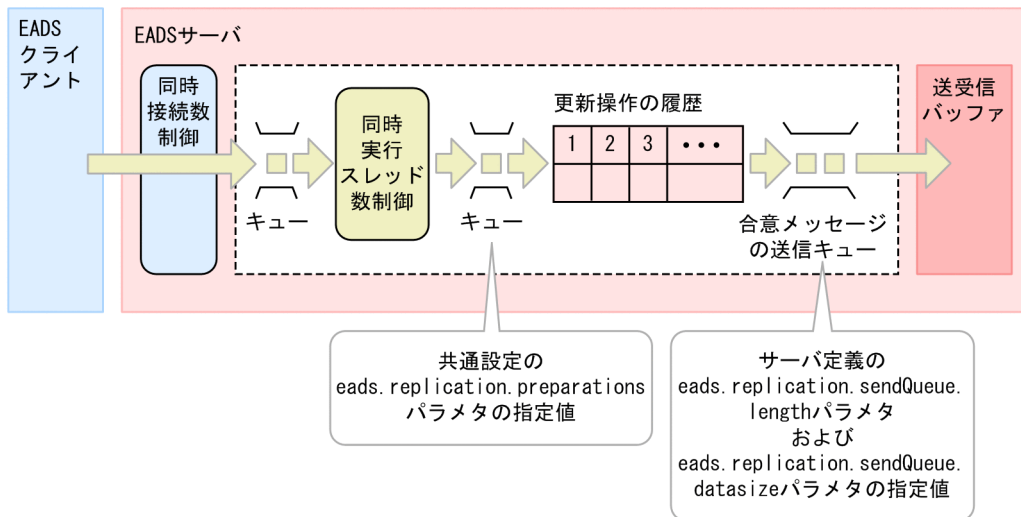
EADS サーバが通信で使用するバッファの概要を次の図に示します。



(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.replication.sendQueue.length` 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.replication.sendQueue.datasize` パラメタで設定できます。

なお、共通設定の `eads.replication.preparations` 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.server.connection.buffersize</code>	バッファサイズ (1024 バイト～ 16777216 バイト)	4096 バイト
2		<code>eads.replication.connection.buffersize</code>	バッファサイズ (1024 バイト～	131071 バイト

項番	定義ファイル	パラメタ名	指定する値	デフォルト値
			16777216 バイト)	
3		eads.replication.sendQueue.length	キューの長さ (10000~ 1000000)	100000
4		eads.replication.sendQueue.datasize	1048576 バイト~2147483647 バイト	16777216 バイト
5		eads.transfer.connection.bufferSize	バッファサイズ (1024 バイト~ 16777216 バイト)	131071 バイト
6		eads.transfer.datasize	10240 バイト~ 2147483647 バイト	1048576 バイト
7		eads.rebalance.transfer.datasize	10240 バイト~ 2147483647 バイト	1048576 バイト
8		eads.replication.fillgap.copy.datasize	1024 バイト~ 16777216 バイト	10240 バイト
9		eads.admin.operation.resume.send.datasize	0 バイト~ 2147483647 バイト	1048576 バイト
10		eads.java.nio.maxCachedBufferSize	0~ 9223372036854 775807 バイト	なし
11	共通設定ファイル	eads.cache.key.maxsize	1 バイト~1024 バイト	1024 バイト
12		eads.replication.preparations	1~100	20
13		eads.replication.external.heapsize	1 メガバイト~ 268435456 メガ バイト	450 メガバイト
14		eads.java.heapsize	ヒープサイズ※1	3072 メガバイト
15		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
16		eads.java.metaspacesize	メモリサイズ※2	83 メガバイト
17		eads.cache.limiter.enable	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	true

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		<code>eads.cache.keyCount</code>	1024～ 1073741824	1048576
19		<code>eads.server.nonBlocking.external.heapsize</code>	1 メガバイト～ 268435456 メガ バイト	30 メガバイト
20	クライアント定義 ファイル	<code>eads.client.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:MetaspaceSize および-XX:MaxMetaspaceSize) に適用されます。

## (1) サーバ定義ファイル

### (a) `eads.server.connection.bufferSize`

コネクション単位で確保する、データの送受信で使用する作業領域のバッファサイズ (単位: バイト) を指定します。

### (b) `eads.replication.connection.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。

### (c) `eads.replication.sendQueue.length`

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: 共通設定の `eads.replication.preparations` の指定値
- b: 共通設定の `eads.replication.factor` - 1 (データの多重度が 1 の場合は 1)



- c : 共通設定の `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。

### (d) `eads.replication.sendQueue.datasize`

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 : 共通設定の `eads.replication.preparations` 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。  
MAX (key の最大サイズ + value の最大サイズ × 2, (key の最大サイズ + value の最大サイズ) × 最大同時更新データ件数)

(凡例)

- 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 (2, 10) の計算結果は 10 となります。
- key の最大サイズ :  
共通設定の `eads.cache.key.maxsize` 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 の最大サイズです。
- 最大同時更新データ件数 :  
メモリキャッシュを使用して、データの一括操作を実行する場合は 10 を代入します。それ以外の場合は、1 を代入します。
- c : 共通設定の `eads.replication.factor - 1` (データの多重度が 1 の場合は 1)
- d : サーバ定義の `eads.replication.fillgap.copy.datasize` パラメタの指定値
- e : 共通設定の `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。

## (e) eads.transfer.connection.buffersize

復旧処理、スケールアウト処理、およびリバランス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。

## (f) eads.transfer.datasize

復旧処理およびスケールアウト処理で送信するデータサイズ（単位：バイト）を指定します。

復旧処理およびスケールアウト処理では、復旧対象の EADS サーバ、およびスケールアウト処理で追加した EADS サーバでこのパラメタに指定した値が適用されます。

復旧処理およびスケールアウト処理では、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、およびスケールアウト処理で追加した EADS サーバにデータを送信します。

`eads.transfer.interval` 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。スケールアウト処理については、「[9.3.2\(6\) クラスタのスケールアウト処理（EADS サーバの追加）](#)」を参照してください。

ディスクキャッシュ、および 2Way キャッシュを復旧する場合、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.disk.transfer.datasize` パラメタの指定値が有効となります。

### 注意事項

復旧処理およびスケールアウト処理では、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。

## (g) eads.rebalance.transfer.datasize

リバランス処理のデータ転送で送信するデータサイズ（単位：バイト）を指定します。

リバランス処理では、EADS サーバが新しく管理するレンジのデータは、元々そのレンジを管理していた EADS サーバ（分割前のレンジを管理していた EADS サーバ）から転送されます。その際、このパラメタに指定したデータサイズで、データを転送します。

分割前のレンジを管理していた EADS サーバと、分割後（結合後）のレンジを管理する EADS サーバで、このパラメタの指定値が異なる場合には、分割後（結合後）のレンジを管理する EADS サーバのパラメタの指定値が適用されます。

なお、`eztool rebalance` コマンドの `--datasize` オプションを指定した場合には、このパラメタの指定値よりも、`--datasize` オプションの指定値が優先されます。

`eads.rebalance.transfer.interval` 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。

## 注意事項

リバランス処理では、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。

## (h) `eads.replication.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\(8\) 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。

$\text{MAX}(\text{key の最大サイズ} + \text{value の最大サイズ} \times 2, (\text{key の最大サイズ} + \text{value の最大サイズ}) \times \text{最大同時更新データ件数})$

(凡例)

- 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 (2, 10) の計算結果は 10 となります。
- key の最大サイズ：  
共通設定の `eads.cache.key.maxsize` 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 の最大サイズです。
- 最大同時更新データ件数：  
メモリキャッシュを使用して、データの一括操作を実行する場合は 10 を代入します。それ以外の場合は、1 を代入します。

合意処理で一度に合意できる処理の最大数：

共通設定の `eads.replication.preparations` 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。

### (i) `eads.admin.operation.resume.send.datasize`

`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。

### (j) `eads.java.nio.maxCachedBufferSize`

EADS サーバでキャッシュされる 1 ダイレクトバッファ当たりのサイズの上限を指定します。上限を設定することでダイレクトバッファのメモリ使用量を制限できます。

このパラメタに指定した値は、システムプロパティ `jdk.nio.maxCachedBufferSize` に設定されます。

このパラメタに値を指定しなかった場合、または  $0 \sim 2^{63} - 1$  の範囲外の値もしくは数値以外を指定した場合は、 $2^{63} - 1$  として扱われ、無制限となります。

このパラメタには `eads.server.connection.buffersize` パラメタと同じ値を指定することを推奨します。ただし、`eads.server.nonBlocking.enable` パラメタの指定値が `false`、かつ `eads.server.connection.buffersize` パラメタの指定値が 8192 未満の場合は、8192 を指定することを推奨します。

EADS サーバの処理性能を重視する場合は、パラメタに値を指定しないか、または EADS クライアントと EADS サーバ間で通信するデータの最大サイズを設定してください。

## 注意事項

- このパラメタに 1024 未満の値を指定すると、性能が大きく劣化することがあります。
- このパラメタに指定した値以上のデータサイズを扱うリクエストでは、このパラメタに値を指定しない場合に比べて、性能が劣化することがあります。

## (2) 共通設定ファイル

### (a) `eads.cache.key.maxsize`

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.replication.preparations

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。

同時に実行される更新操作が多い場合、大きな値を設定することでスループットが向上する場合がありますが、デフォルト値のままにしておくことを推奨します。

### (c) eads.replication.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、次のどちらかを満たす場合、EADS サーバの起動に失敗するので注意してください。

- 共通設定の `eads.java.external.heapsize` パラメタの指定値 $\times 0.97$ （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
- サーバ定義の `eads.server.nonBlocking.enable` パラメタが true、かつ共通設定の `eads.server.nonBlocking.external.heapsize` パラメタの指定値+このパラメタの指定値が、共通設定の `eads.java.external.heapsize` パラメタの指定値 $\times 0.97$ （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

### (d) 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。

### (e) 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。

このパラメタの指定値が、次のどちらかを満たす場合、EADS サーバの起動に失敗するので注意してください。

- 共通設定の `eads.replication.external.heapsize` パラメタの指定値が、このパラメタの指定値 $\times 0.97$ （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

- サーバ定義の `eads.server.nonBlocking.enable` パラメタが true, かつ共通設定の `eads.server.nonBlocking.external.heapsize` パラメタの指定値 + `eads.replication.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

#### (f) `eads.java.metaspace.maxsize`

metaspace 領域のサイズ (単位: メガバイト) を指定します。

metaspace 領域は, ロードされた EADS サーバ, ユーザ関クションの class などの情報が格納される領域です。

#### (g) `eads.cache.limiter.enable`

データ総量監視機能を有効にするかどうかを指定します。

データ総量監視機能を有効にすると, データの格納先の容量が不足することが予想される場合に, その処理をエラーとすることで, EADS サーバがダウンすることを防ぎます。

true :

データ総量監視機能を有効にします。

false :

データ総量監視機能を無効にします。

#### (h) `eads.cache.keyCount`

データ総量監視機能で監視する, 1 レンジ当たりのデータ件数を指定します。

`eads.cache.limiter.enable` パラメタに false を指定している場合 (データ総量監視機能が無効になっている場合), このパラメタの指定値は無効となります。

#### (i) `eads.server.nonBlocking.external.heapsize`

ノンブロッキング I/O 通信バッファサイズ (単位: メガバイト) を指定します。詳細は, 「7.8.3 ノンブロッキング I/O 通信制御に依存するパラメタ」の `eads.server.nonBlocking.external.heapsize` パラメタを参照してください。

### (3) クライアント定義ファイル

#### (a) `eads.client.connection.bufferSize`

EADS クライアントがデータの送受信で使用するバッファのサイズ (単位: バイト) を指定します。

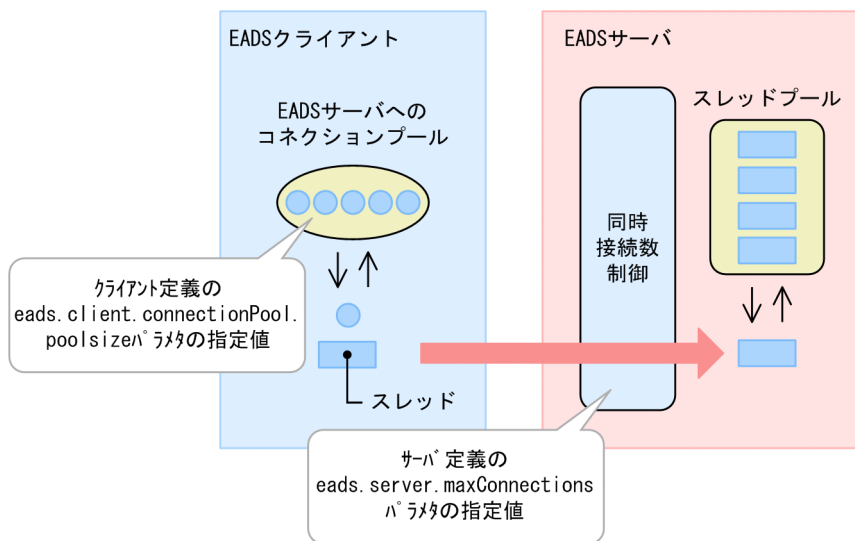


## 9.2 スレッドプールおよびコネクションプールに関連するパラメタの設計

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。



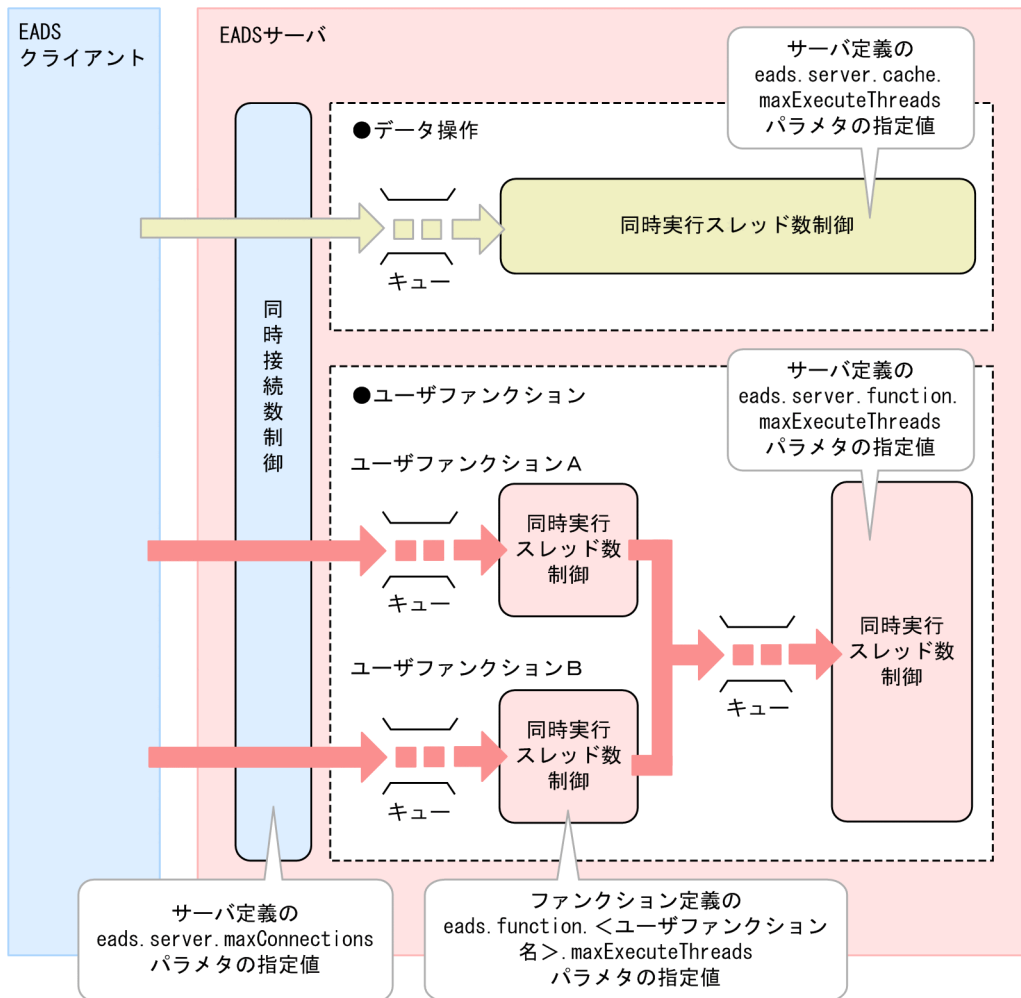
#### 考え方

最大同時接続数（サーバ定義の `eads.server.maxConnections` 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.maxExecuteThreads` 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.connection.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。

- ユーザファンクションの同時実行スレッド数

ユーザファンクションごとの同時実行スレッド数（ファンクション定義の `eads.function.<ユーザーファンクション名>.maxExecuteThreads` パラメタ）を制御し、その上でユーザファンクション全体の同時実行スレッド数（サーバ定義の `eads.server.function.maxExecuteThreads` 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.maxConnections*</code>	1~1024	10
2		<code>eads.server.cache.maxExecuteThreads</code>	1~ <code>eads.server.maxConnections</code> パラメータの指定値	<code>eads.server.maxConnections</code> パラメータの指定値
3		<code>eads.server.function.maxExecuteThreads</code>	1~ <code>eads.server.maxConnections</code> パラメータの指定値	<code>eads.server.maxConnections</code> パラメータの指定値
4		<code>eads.server.nonBlocking.processorThreads</code>	1~8192	10
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;.maxExecuteThreads</code>	0~ <code>eads.server.function.maxExecuteThreads</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.client.connectionPool.poolsize</code>	1~1024	10
7		<code>eads.client.connectionPool.exceededMaxSizeError.enable</code>	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	false

注※

サーバ定義の `eads.server.nonBlocking.enable` パラメータに true を指定した場合の最小値およびデフォルト値は 1024、最大値は 32768 です。

## (1) サーバ定義ファイル

### (a) `eads.server.maxConnections`

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。

次に示す値の合計以上を設定してください。

- 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.client.connectionPool.poolsize` パラメータの値の総和
- `eads.client.eventListener.enable` パラメータに true を指定した EADS クライアントの総和
- 同時実行する次のコマンドの数
  - `eztool put` コマンド
  - `eztool get` コマンド
  - `eztool remove` コマンド

- `eztool removeall` コマンド
- `eztool execfunc` コマンド

ただし、`eads.server.nonBlocking.enable` パラメタに `true` を指定した場合の最小値およびデフォルト値は 1024、最大値は 32768 です。

なお、`eads.client.connectionPool.poolsize` パラメタに設定した値は 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.client.connectionPool.poolsize` パラメ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.maxExecuteThreads`

データ操作の最大同時実行スレッド数を指定します。

ただし、`eads.server.nonBlocking.enable` パラメタに `true` を指定した場合のデフォルト値および最大値は、両方とも `eads.server.nonBlocking.processorThreads` パラメタの指定値になります。

## (c) `eads.server.function.maxExecuteThreads`

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。

ただし、`eads.server.nonBlocking.enable` パラメタに `true` を指定した場合のデフォルト値および最大値は、両方とも `eads.server.nonBlocking.processorThreads` パラメタの指定値になります。

## (d) `eads.server.nonBlocking.processorThreads`

ノンブロッキング I/O 通信制御を使用する際の、リクエスト処理スレッド数を指定します。詳細は、「7.8.3 ノンブロッキング I/O 通信制御に依存するパラメタ」の `eads.server.nonBlocking.processorThreads` パラメタを参照してください。

## (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>.maxExecuteThreads`

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.client.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、コネクションを確立できます。

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.client.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。

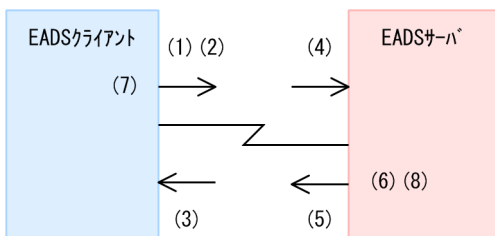
#### 考え方

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 9-2 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 9-2 の(1)～(8)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	eads.client.connection.send.timeout
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		eads.client.connection.receive.timeout

項番	設定するタイムアウト	定義ファイル	パラメタ名
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	eads.server.connection.timeout
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		eads.server.connection.keepAlive.timeout
7	イベントリスナ用の接続の再接続	クライアント定義ファイル	eads.client.eventListener.reconnect.interval
8		サーバ定義ファイル	eads.server.eventListener.notify.retry.timeout

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 9-3 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箇所（データ操作の場合）

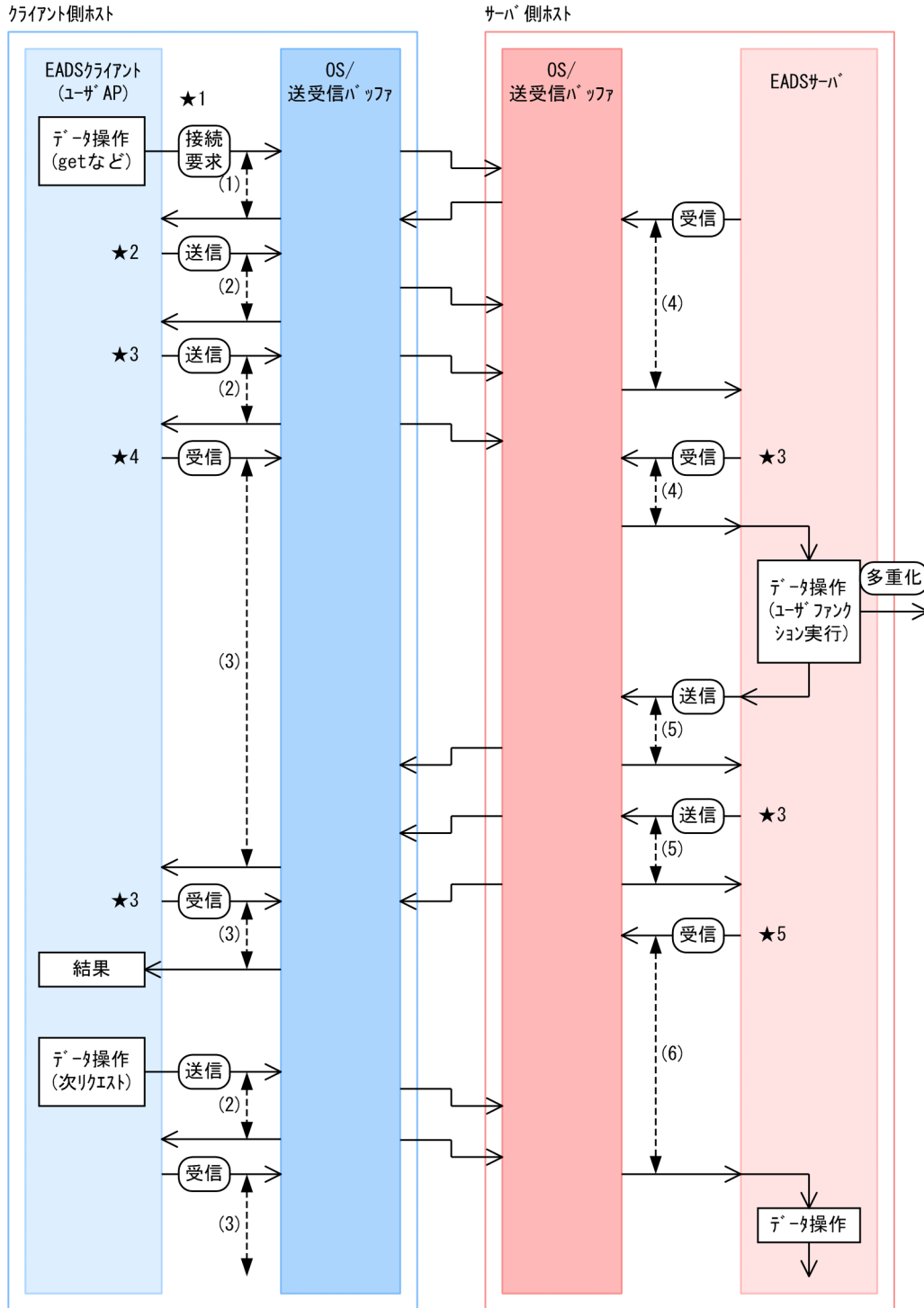
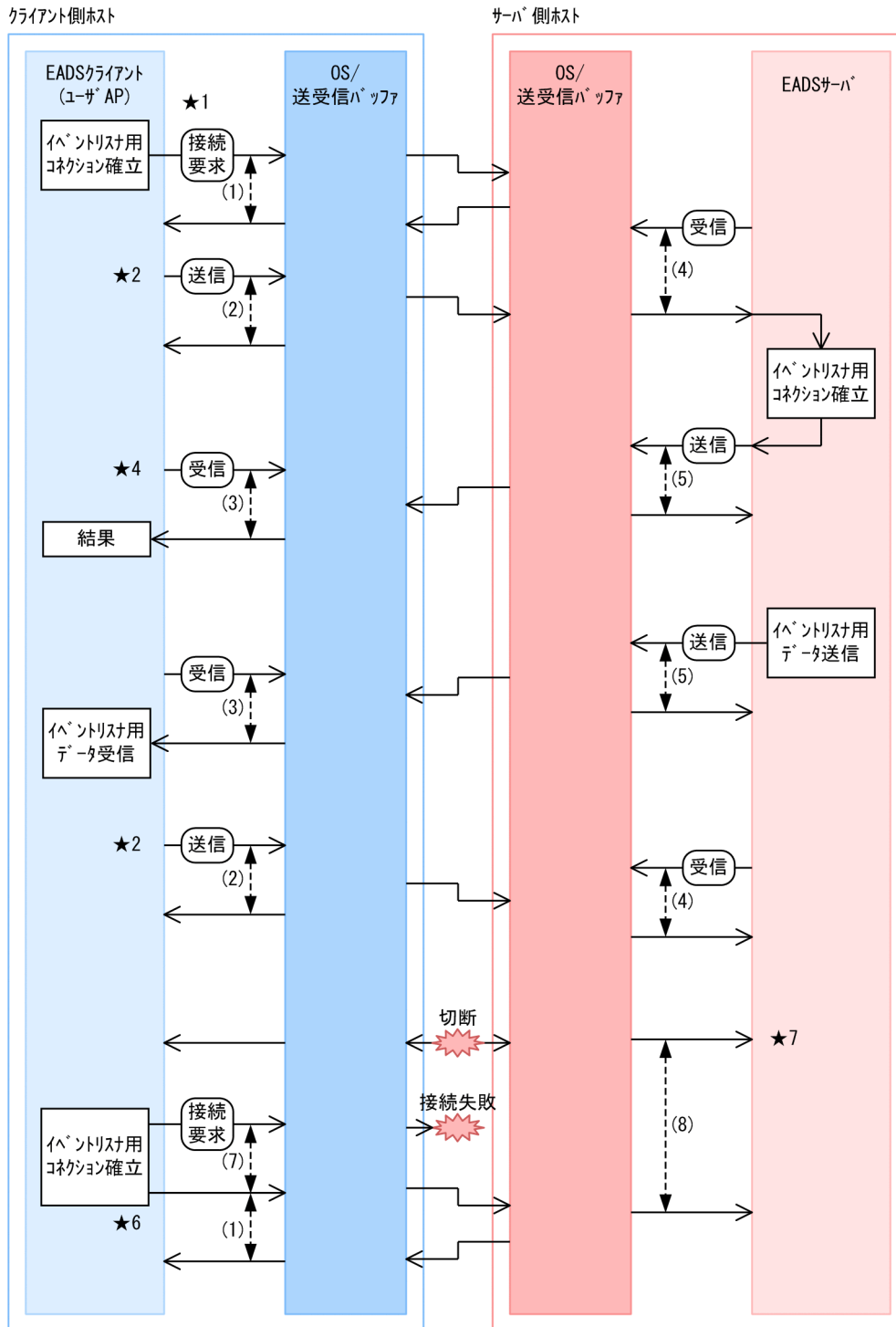


図 9-4 EADS クライアント・EADS サーバ間の通信で、タイムアウトが設定できる箇所（イベントリスナを使用する場合）



各タイムアウト（図中の(1)～(8)）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) リクエスト処理用の常設コネクション切断までのタイムアウト値の設定」
- 「9.3.1(4) イベントリスナ用のコネクションの再接続のタイムアウト値の設定」

また、ポイントとなる項目 (★1~★7) については、「9.3.1(5) 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 9-3 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) リクエスト処理用の常設コネクション切断までのタイムアウト値の設定

リクエスト処理用の常設コネクション切断までのタイムアウト値の設定 (図 9-3 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ときに、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。

むだな通信の切断やエラーを回避するために、タイムアウト値には大きめの値を設定することを推奨します。ただし、ほかの機能に影響を与えないように、次の点も考慮してください。

### • 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リクエスト処理用の常設コネクションを保持します。リクエスト処理用の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 サーバに対するリクエスト処理用の常設コネクション数は最大で、クライアント定義の `eads.client.connectionPool.poolsize` パラメー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` パラメー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.client.connection.send.timeout` パラメー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、リクエスト処理用の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.connection.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.connection.keepAlive.timeout` パラメタの指定値

> (クライアント定義の `eads.client.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.client.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.client.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.maxConnections` 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.connection.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ます。この結果、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.connection.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.maxConnections` パラメタにあらかじめ加算し、`eads.server.connection.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) イベントリスナ用のコネクションの再接続のタイムアウト値の設定

イベントリスナ用のコネクションの再接続のタイムアウト (図 9-4 の(7)および(8)) は、次の目的のために設定します。

- EADS サーバが動作する OS、またはホストがダウンした場合、EADS クライアントがコネクションの再接続を繰り返すときに、CPU やネットワークの帯域などのリソースが占有されることを防ぐ。
- 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、EADS サーバが気づくことができないときに、イベントの再送用のメモリが占有されることを防ぐ。

むだな通信の切断やエラーを回避するために、タイムアウト値には大きめの値を設定することを推奨します。ただし、ほかの機能に影響を与えないように、次の点も考慮してください。

- EADS クライアント側の再接続間隔との関係

EADS クライアントはイベントリスナ用のコネクションを保持します。1 つの EADS サーバに対するイベントリスナ用の常設コネクション数は 1 です。

コネクションが切断された場合、1 回目の再接続では、即時に再接続することで、一時的なネットワーク障害が発生してもイベントの通知が遅延しないようにします。2 回目以降の再接続では、クライアント定義の `eads.client.eventListener.reconnect.interval` パラメタに指定した間隔で、各 EADS サーバに対してコネクションを再接続します。

ただし、1 回目の再接続でも、接続先 EADS サーバの最大同時接続数を超過して接続しようとした場合は、即時に再接続しません。この場合は、クライアント定義の `eads.client.eventListener.reconnect.interval` パラメタに指定した間隔で、コネクションを再接続します。

なお、コネクションの再接続は 1 スレッドで実行されますが、複数の EADS サーバとの通信は並列に処理されます。このため、OS が正常に動作していないホストのアドレスに対して再接続しても、その間、ほかの EADS サーバからの再接続やイベントの通知が滞ることはありません。

- EADS サーバ側の再接続間隔との関係

イベントリスナ用のコネクションが切断された場合、EADS サーバは、EADS クライアントからの再接続待ちとなります。

サーバ定義の `eads.server.eventListener.notify.retry.timeout` パラメタには、EADS クライアントに再送するデータを破棄するまでの待ち時間を指定します。再接続する前にメモリを解放すると、その EADS サーバからのイベントの通知が欠落するおそれがあります。そのため、サーバ定義の `eads.server.eventListener.notify.retry.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.eventListener.notify.retry.timeout` パラメタの指定値  
> クライアント定義の `eads.client.connection.send.timeout` パラメタの指定値  
+ クライアント定義の `eads.client.eventListener.reconnect.interval` パラメタの指定値

なお、コネクションの再接続待ちは 1 スレッドで実行されますが、複数の EADS クライアントとの通信は並列に処理されます。このため、OS が正常に動作していないホストのアドレスに対して再接続しても、その間、ほかの EADS クライアントへの再接続やイベントの通知が滞ることはありません。

- EADS サーバ側の接続最大数との関係

サーバ定義の `eads.server.maxConnections` 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.connection.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ます。この結果、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、メモリに余裕がある場合は、再接続に使用するコネクション数を `eads.server.maxConnections` パラメタにあらかじめ加算することを推奨します。

## (5) 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 9-3 および図 9-4 の★1～★7 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.client.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見なされます。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.server.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.connection.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) リクエスト処理用の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。

### ポイント 6 (★6)

EADS クライアントは、イベントリスナ用のコネクションを保持し続けます。コネクションが切断された場合には、EADS サーバに対して再接続を繰り返します。

イベントリスナ用の接続の再接続については、「9.3.1(4) イベントリスナ用の接続の再接続のタイムアウト値の設定」を参照してください。

#### ポイント7 (★7)

EADS サーバは、イベントリスナ用の接続を保持し続けます。接続が切断された場合には、EADS クライアントからの再接続を待ちます。

イベントリスナ用の接続の再接続については、「9.3.1(4) イベントリスナ用の接続の再接続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 EADS サーバとの通信や、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。

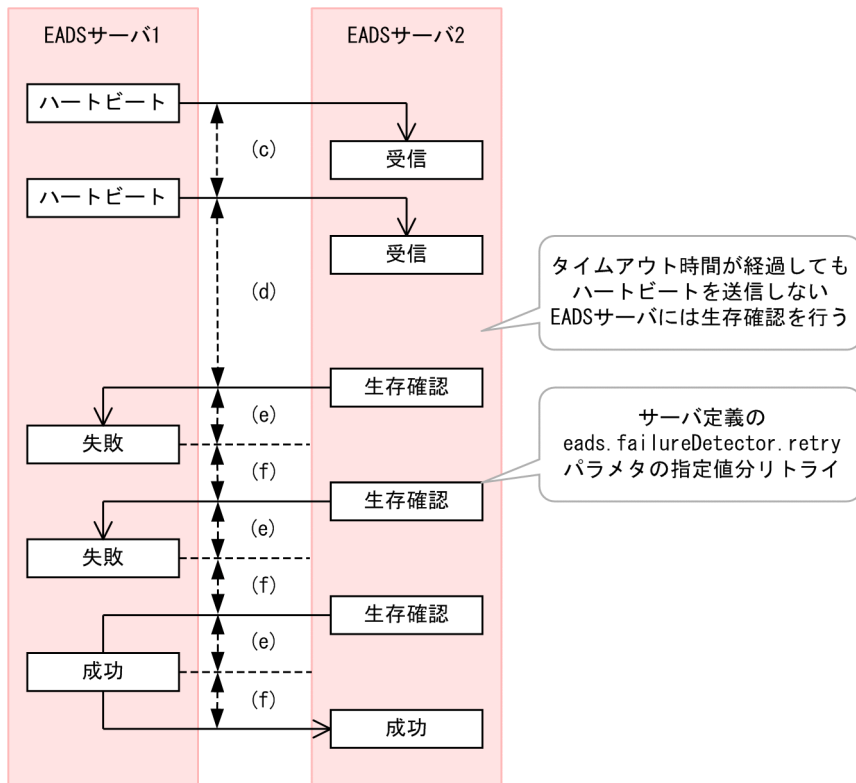
#### 考え方

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、長くすることで頻繁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。

(c) : 「9.3.3(1)(c) `eads.failureDetector.heartbeat.interval`」

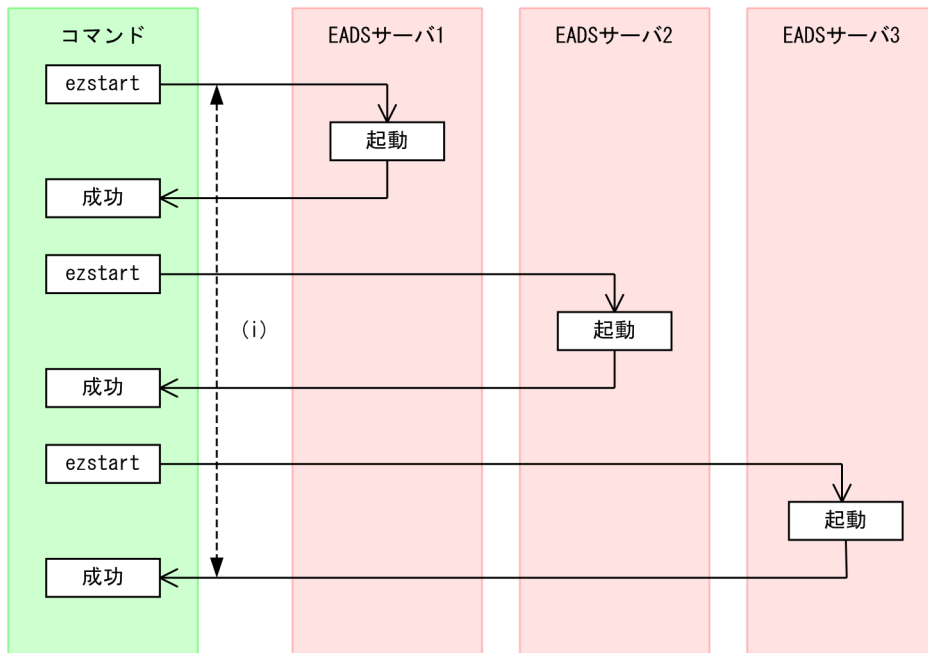
(d) : 「9.3.3(1)(d) `eads.failureDetector.heartbeat.timeout`」

(e) : 「9.3.3(1)(e) `eads.failureDetector.connection.timeout`」

(f) : 「9.3.3(1)(f) `eads.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。

(i) : 「9.3.3(1)(i) eads.admin.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。

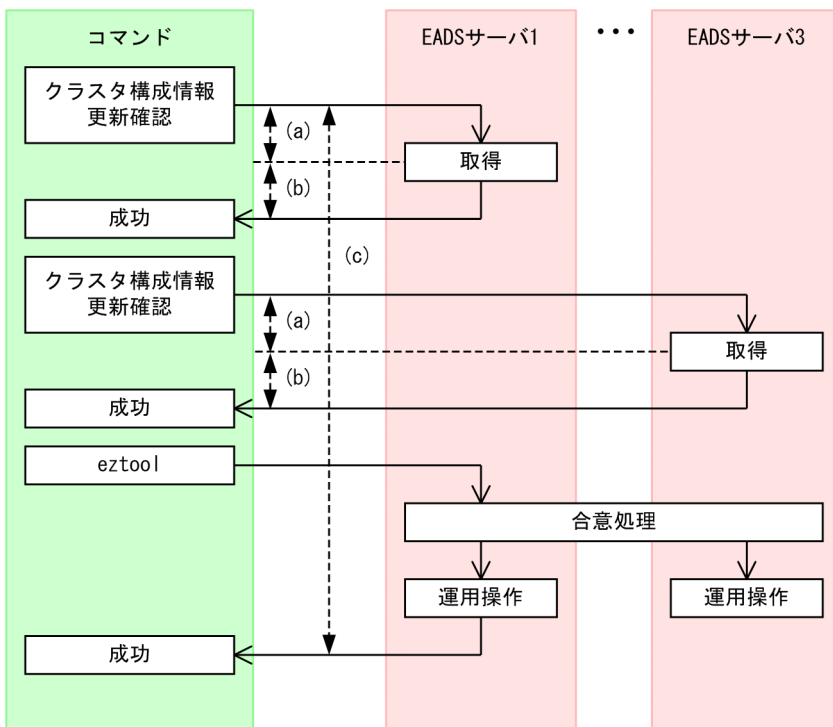
### (c) クラスタ起動中に、まだクラスタに参加していない EADS サーバがダウンした場合

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) クラスターの運用操作

eztool コマンドを実行して、クラスター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.command.connection.timeout」

(b) : 「9.3.3(2)(b) eads.command.common.read.timeout」 ※1

(c) : 「9.3.3(2)(c) eads.command.common.execution.timeout」 ※2

注※1

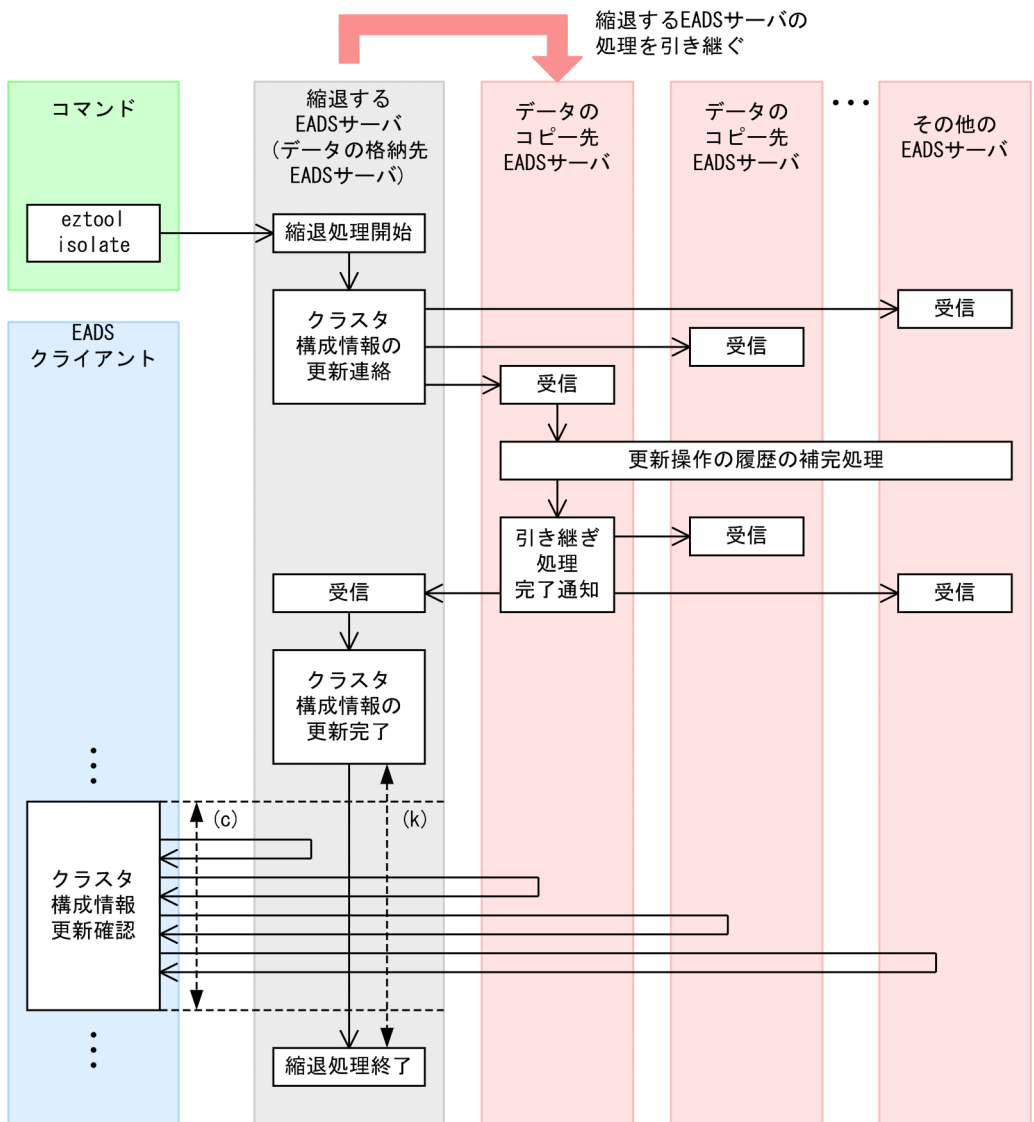
コマンド定義の eads.command.<サブコマンド名>.read.timeout パラメタを指定した場合は、eads.command.<サブコマンド名>.read.timeout パラメタの値が適用されます。

注※2

コマンド定義の eads.command.<サブコマンド名>.execution.timeout パラメタを指定した場合は、eads.command.<サブコマンド名>.execution.timeout パラメタの値が適用されます。

### (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ムの関係を示します。



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」

(k) : 「9.3.3(1)(k) eads.admin.operation.isolate.gracefulstop.waitTime」

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.admin.operation.isolate.gracefulstop.waitTime パラメタで、EADS サーバのクラスタ構成情報の更新が完了してから縮退処理が完了するまでの時間を指定できます。この値よりも小さい値をクライアント定義の eads.client.clusterInfo.update.interval パラメタに指定することで、EADS クライアントのクラスタ構成情報の更新完了後に、EADS サーバを縮退できます。

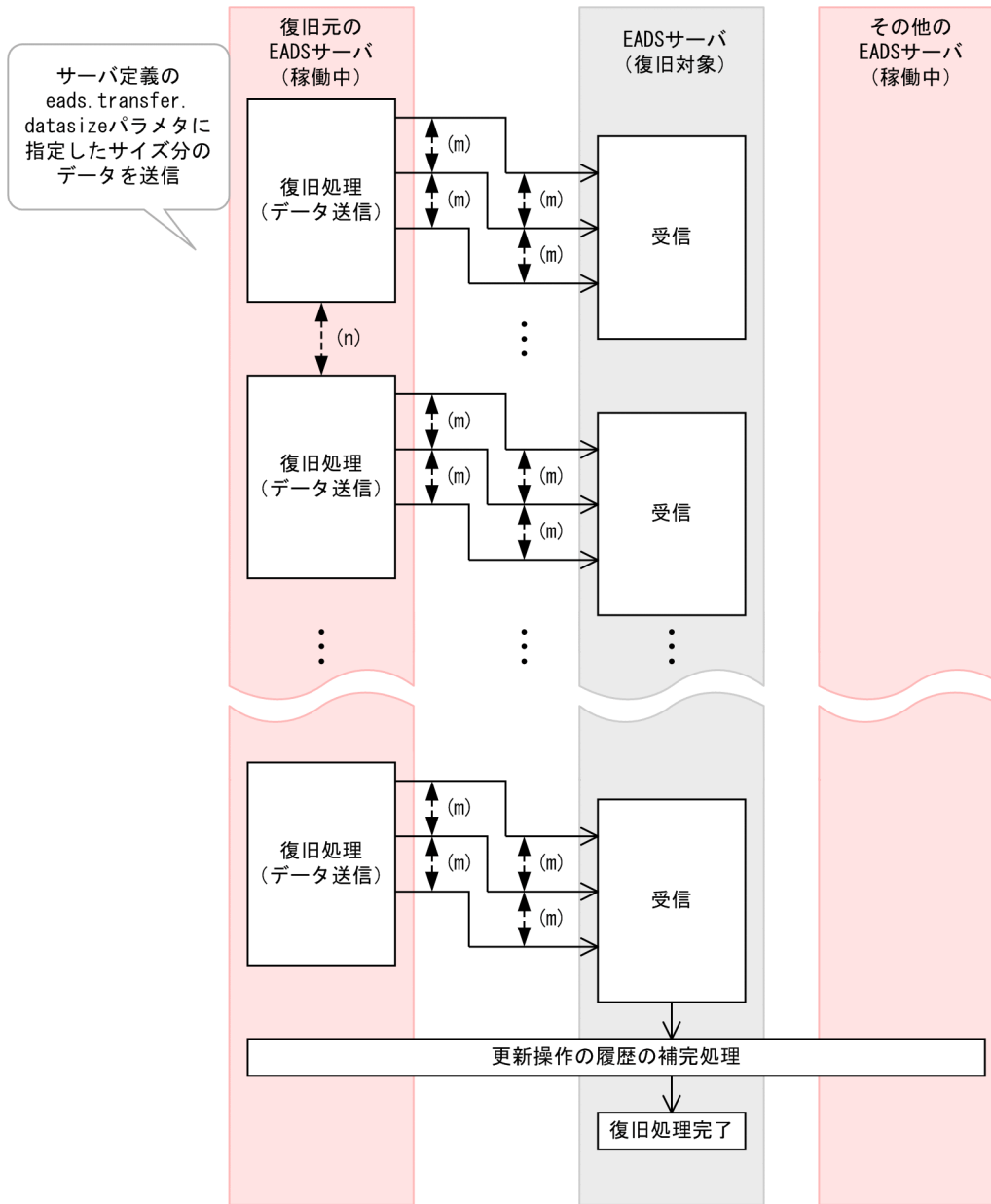
なお、クラスタ監視によって EADS サーバが縮退する場合、eads.admin.operation.isolate.gracefulstop.waitTime パラメタの指定は無効となります。

更新操作の履歴の補完処理については、「9.3.2(8) 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。

(m) : 「9.3.3(1)(m) eads.transfer.timeout」

(n) : 「9.3.3(1)(n) eads.transfer.interval」

復旧対象の EADS サーバのサーバ定義の `eads.transfer.datasize` 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、CPU のリソースやネットワークの帯域を使用するといった影響があります。
- データ操作と復旧処理が重なって EADS サーバ内の処理が追いつかなくなった場合は、メモリが枯渇しないように、データ操作を待たせることがあります。

## 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.disk.transfer.datasize` 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.disk.transfer.interval` 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\(8\) 更新操作の履歴の補完処理](#)」を参照してください。

## (6) クラスタのスケールアウト処理 (EADS サーバの追加)

クラスタをスケールアウトする (クラスタに EADS サーバを追加する) 際のタイムは、クラスタを復旧する際のタイムと同じです。

クラスタを復旧する際のタイムについては、「[9.3.2\(5\) クラスタの復旧処理](#)」を参照してください。その際には、説明の「復旧」を「スケールアウト」に読み替えてください。

## (7) クラスタのリバランス処理 (レンジに格納できる key の数の均等化)

クラスタをリバランスする (レンジに格納できる key の数を均等にする) 際のタイムは、クラスタを復旧する際のタイムと同じです。

クラスタを復旧する際のタイムについては、「[9.3.2\(5\) クラスタの復旧処理](#)」を参照してください。その際には、説明の「復旧」を「リバランス」に読み替えてください。

また、リバランスの場合は、稼働中の EADS サーバから稼働中の EADS サーバに対してデータを送信します。このため、説明のパラメタ名を次のように読み替えてください。

- 「`eads.transfer.datasize` パラメタ」を「`eads.rebalance.transfer.datasize` パラメタ」に読み替える
- 「`eads.transfer.interval` パラメタ」を「`eads.rebalance.transfer.interval` パラメタ」に読み替える

## (8) 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、スケールアウト処理、リバランス処理、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。

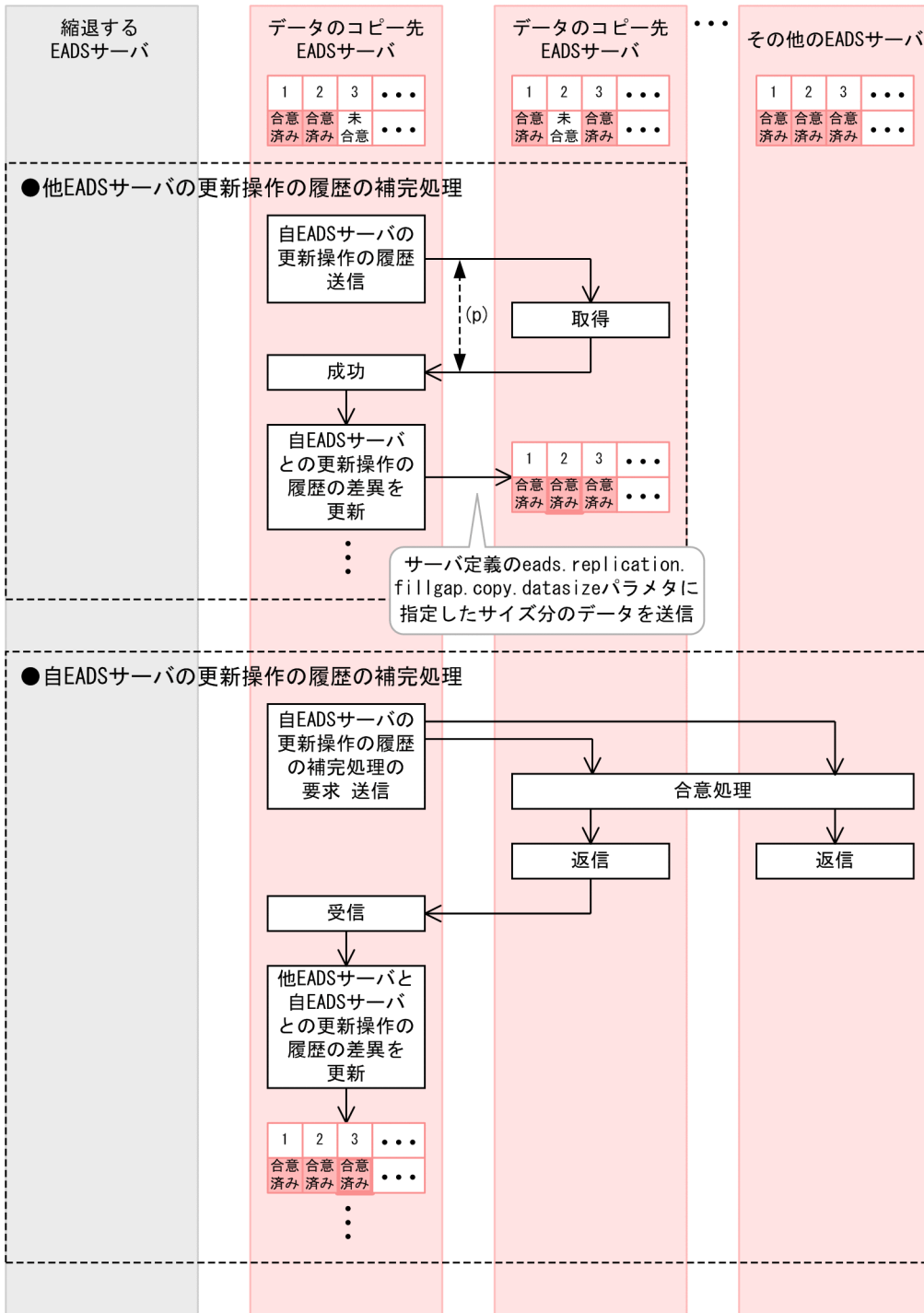
縮退処理の場合：

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、スケールアウト処理、およびリバランス処理の場合：

復旧対象の EADS サーバ、スケールアウト処理で追加する EADS サーバ、およびリバランス処理で位置を変更する 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	...
合意済み	合意済み	未合意	...

: 更新操作の履歴

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。

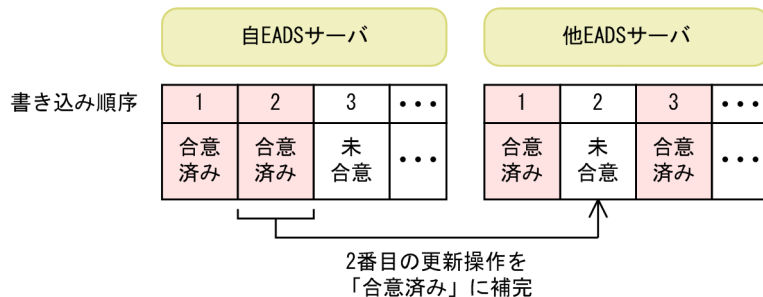
(p) : 「9.3.3(1)(p) eads.replication.fillgap.copy.timeout」

### 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。

2. 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.replication.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。
3. 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。

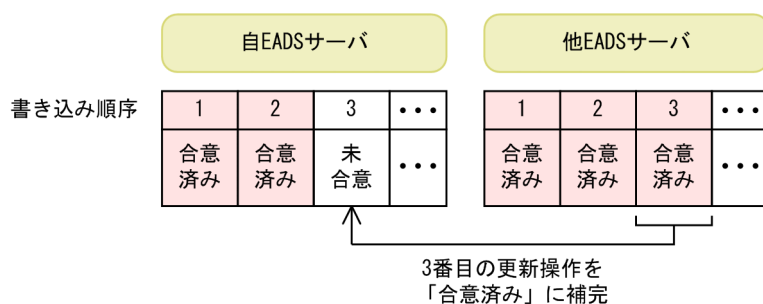


他 EADS サーバの更新操作の履歴の補完処理は、データのコピー先 EADS サーバに対して、更新操作の履歴の差異の数だけ実行されます。

### 自 EADS サーバの更新操作の履歴の補完処理

1. 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。
2. 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.replication.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。
3. 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。
4. 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 回の縮退処理、復旧処理、スケールアウト処理、またはリバランス処理に対して、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 (データの多重度-1) × (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、データの多重度-1（データの多重度が1の場合は1）になります。

### 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.server.connection.timeout</code>	100~3600000 (ミリ秒)	60000
2		<code>eads.admin.operation.connection.timeout</code>	100~3600000 (ミリ秒)	10000
3		<code>eads.failureDetector.heartbeat.interval</code>	10~60000 (ミリ秒)	400
4		<code>eads.failureDetector.heartbeat.timeout</code>	10~86400000 (ミリ秒)	2000
5		<code>eads.failureDetector.connection.timeout</code>	1~60000 (ミリ秒)	500
6		<code>eads.failureDetector.read.timeout</code>	1~60000 (ミリ秒)	500
7		<code>eads.failureDetector.retry</code>	0~100	0
8		<code>eads.failureDetector.assertive.threshold</code>	1~49	1
9		<code>eads.admin.boot.timeout</code>	1~86400 (秒)	60
10		<code>eads.server.connection.keepAlive.timeout</code>	0~3600 (秒)	3600
11		<code>eads.admin.operation.isolate.gracefulstop.waitTime</code>	0~3600000 (ミリ秒)	3000
12		<code>eads.replication.consensus.timeout</code>	10~3600000 (ミリ秒)	800
13		<code>eads.transfer.timeout</code>	100~3600000 (ミリ秒)	60000
14		<code>eads.transfer.interval</code>	0~60000 (ミリ秒)	1000
15		<code>eads.rebalance.transfer.interval</code>	1~60000 (ミリ秒)	1000

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.replication.fillgap.copy.timeout</code>	100~3600000 (ミリ秒)	2000
17		<code>eads.admin.operation.resume.send.interval</code>	0~2147483647 (ミリ秒)	0
18		<code>eads.server.eventListener.notify.retry.timeout</code>	10~3600000 (ミリ秒)	120000
19	コマンド定義ファイル	<code>eads.command.connection.timeout</code>	0~2147483647 (ミリ秒)	3000
20		<code>eads.command.common.read.timeout</code>	0~2147483647 (ミリ秒)	600000
21		<code>eads.command.common.execution.timeout</code>	0~2147483647 (秒)	600
22		<code>eads.command.&lt;サブコマンド名&gt;.read.timeout</code>	0~2147483647 (ミリ秒)	600000
23		<code>eads.command.&lt;サブコマンド名&gt;.execution.timeout</code>	0~2147483647 (秒)	600
24	クライアント定義ファイル	<code>eads.client.connection.send.timeout</code>	100~3600000 (ミリ秒)	60000
25		<code>eads.client.connection.receive.timeout</code>	100~3600000 (ミリ秒)	60000
26		<code>eads.client.clusterInfo.update.interval</code>	10~60000 (ミリ秒)	1000
27		<code>eads.client.eventListener.enable</code>	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	false
28		<code>eads.client.eventListener.reconnect.interval</code>	1000~60000 (ミリ秒)	1000

## (1) サーバ定義ファイル

### (a) `eads.server.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.admin.operation.connection.timeout`

次に示すコマンド実行時の、EADS サーバ間の接続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。

- `eztool import`

- `eztool resume`
- `eztool importecf`

### (c) `eads.failureDetector.heartbeat.interval`

ハートビートの送信間隔（単位：ミリ秒）を指定します。

このパラメタの指定値が大きいほど、復旧処理、スケールアウト処理、およびリバランス処理の性能に影響が出やすくなります。また、クラスタの起動時にタイムアウトが発生しやすくなります。

このパラメタの指定値が小さいほど、処理の通信負荷が高くなります。

このパラメタには、次に示すパラメタの指定値より小さい値を指定してください。

- サーバ定義の `eads.admin.boot.timeout` パラメタ
- サーバ定義の `eads.transfer.timeout` パラメタ
- コマンド定義の `eads.command.common.read.timeout` パラメタ
- コマンド定義の `eads.command.common.execution.timeout` パラメタ
- コマンド定義の `eads.command.<サブコマンド名>.read.timeout` パラメタ
- コマンド定義の `eads.command.<サブコマンド名>.execution.timeout` パラメタ

### (d) `eads.failureDetector.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。

このパラメタの指定値が大きいほど、障害を検知するまでに時間が掛かります。一方で、このパラメタの指定値が小さすぎると、障害を誤検知するおそれがあります。

また、このパラメタの指定値が大きいほど、更新操作の履歴を管理および格納するために、より多くの Java ヒープサイズ（共通設定の `eads.java.heapsize` パラメタの指定値）および Explicit ヒープサイズ（共通設定の `eads.java.external.heapsize` パラメタの指定値）が必要になります。

### (e) `eads.failureDetector.connection.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。

このパラメタの指定値が大きいほど、コネクションを確立できない場合に、障害を検知するまでの時間が掛かります。一方で、このパラメタの指定値が小さすぎると、コネクションを正常に確立する前にタイムアウトするおそれがあります。

### (f) `eads.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。

このパラメタの指定値が大きいほど、障害を検知するまでの時間が掛かります。一方で、このパラメタの指定値が小さすぎると、障害を誤検知するおそれがあります。



### (g) `eads.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。

### (h) `eads.failureDetector.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。

このパラメタの指定値を大きくすると、ダウンした EADS サーバをいつまでも縮退できなくなるおそれがあります。その代わりに、スプリットブレイン発生時などネットワークの一時的な障害が起きた場合に、EADS サーバがすぐに縮退されることを防ぐことができます。

### (i) `eads.admin.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。

### (j) `eads.server.connection.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。

### (k) `eads.admin.operation.isolate.gracefulstop.waitTime`

`eztool isolate` コマンドを実行した EADS サーバのクラスタ構成情報の更新が完了してから、縮退処理が完了するまでの時間 (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コネクション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 サーバを縮退できます。

### (l) `eads.replication.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し直します。

### (m) `eads.transfer.timeout`

復旧処理、スケールアウト処理、およびリバランス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。

このパラメタは、復旧処理、スケールアウト処理、およびリバランス処理で、次のようなケースが発生した場合に備えるために設定します。

- データの転送元の EADS サーバがダウンした際に、データの転送先の EADS サーバが待ち状態になる
- データの転送先の EADS サーバがコネクションを保持したままハングアップした際に、データの転送元の EADS サーバが待ち状態になる

## (n) eads.transfer.interval

復旧処理およびスケールアウト処理でのデータ送信間隔（単位：ミリ秒）を指定します。

復旧処理およびスケールアウト処理では、復旧対象の EADS サーバ、またはスケールアウト処理で追加した EADS サーバで、このパラメタに指定した値が適用されます。

このパラメタの指定値とサーバ定義の `eads.transfer.datasize` 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。

$$\begin{aligned} & \text{復旧処理およびスケールアウト処理のデータ送信に使用できる帯域 (単位: bps)} \geq \\ & \left( \text{復旧処理およびスケールアウト処理で送信するデータサイズ (単位: バイト)} \times 8 \right) \\ & \div \{ (\text{復旧処理およびスケールアウト処理でのデータ送信間隔 (単位: ミリ秒)} \\ & + \text{データ送信に掛かる時間 (単位: ミリ秒)}) \div 1,000 \} \end{aligned}$$

復旧処理およびスケールアウト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.transfer.datasize` パラメタの指定値

復旧処理およびスケールアウト処理でのデータ送信間隔（単位：ミリ秒）：

サーバ定義の `eads.transfer.interval` 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.transfer.datasize` 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（スケールアウト処理の場合は、「復旧」を「スケールアウト」に読み替えてください）。この時間は環境によって異なります。

復旧処理およびスケールアウト処理に掛かる時間を短くすると、復旧処理およびスケールアウト処理の通信負荷が高くなります。通信負荷が高くなると、キャッシュを更新する AP の処理速度が低下するおそれがあります。逆に、復旧処理およびスケールアウト処理の通信負荷を低くすると、復旧処理およびスケールアウト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 キャッシュを復旧する場合、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.disk.transfer.interval` パラメタの指定値が有効となります。

## (o) eads.rebalance.transfer.interval

リバランス処理でのデータ転送の送信間隔（単位：ミリ秒）を指定します。

リバランス処理では、EADS サーバが新しく管理するレンジのデータは、元々そのレンジを管理していた EADS サーバ（分割前のレンジを管理していた EADS サーバ）から転送されます。その際、このパラメタに指定した送信間隔で、データを転送します。

分割前のレンジを管理していた EADS サーバと、分割後（結合後）のレンジを管理する EADS サーバで、このパラメタの指定値が異なる場合には、分割後（結合後）のレンジを管理する EADS サーバのパラメタの指定値が優先されます。

また、`eztool rebalance` コマンドの `-i` または `--interval` オプションを指定した場合には、このパラメタの指定値よりも、`-i` または `--interval` オプションの指定値が優先されます。

## (p) `eads.replication.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。

次に示す値のうち、最も大きな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.replication.fillgap.copy.datasize` パラメタの値）
- key の最大サイズ + value の最大サイズ × 2
- (key の最大サイズ + value の最大サイズ) × 最大同時更新データ件数

(凡例)

- key の最大サイズ：  
共通設定の `eads.cache.key.maxsize` 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 の最大サイズです。
- 最大同時更新データ件数：  
メモリキャッシュを使用して、データの一括操作を実行する場合は 10 を代入します。それ以外の場合は、1 を代入します。

更新操作の履歴の補完処理については、「[9.3.2\(8\) 更新操作の履歴の補完処理](#)」を参照してください。

## (q) `eads.admin.operation.resume.send.interval`

`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。

## (r) `eads.server.eventListener.notify.retry.timeout`

イベントリスナ用のコネクションが切断された場合に、EADS サーバが EADS クライアントに対して、イベントの再送をリトライする時間（単位：ミリ秒）を指定します。

## 注意事項

このパラメタに指定した時間の間は、EADS クライアントに再送するデータを EADS サーバが Java ヒープに保持します。時間を過ぎると EADS クライアントに再送するデータは破棄されます。

## (2) コマンド定義ファイル

### (a) `eads.command.connection.timeout`

コマンド実行時の EADS サーバとの通信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。

なお、次のコマンドではこのパラメタの指定は無効になります。

- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

### (b) `eads.command.common.read.timeout`

コマンド実行時の EADS サーバとの通信の受信タイムアウト時間（単位：ミリ秒）を指定します。

このパラメタの指定値には、サーバ定義の `eads.admin.operation.isolate.gracefulstop.waitTime` パラメタの指定値より大きな値を指定してください。

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。

なお、次のコマンドではこのパラメタの指定は無効になります。

- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

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.client.connection.receive.timeout` パラメタに適用します。

- `eztool put`
- `eztool get`
- `eztool remove`
- `eztool listgroup`
- `eztool listkey`
- `eztool removeall`

- `eztool execfunc`

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.client.connection.receive.timeout` パラメタの最大値以上の値を指定した場合 `eads.client.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.client.connection.receive.timeout` パラメタの最小値以下の値を指定した場合 `eads.client.connection.receive.timeout` パラメタの最小値を適用します。

### (c) `eads.command.common.execution.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。

なお、次のコマンドではこのパラメタの指定は無効になります。

- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

### (d) `eads.command.<サブコマンド名>.read.timeout`

特定のサブコマンド実行時の、EADS サーバとの通信の受信タイムアウト時間（単位：ミリ秒）を指定します。

サブコマンド名には、このパラメタの指定値を適用したいサブコマンド名を指定します。サブコマンド名を指定したサブコマンドの実行時には、コマンド定義の `eads.command.common.read.timeout` パラメタの指定値よりも、このパラメタの指定値が優先されます。

このパラメタの指定値には、サーバ定義の `eads.admin.operation.isolate.gracefulstop.waitTime` パラメタの指定値より大きな値を指定してください。

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。

なお、次のコマンドではこのパラメタの指定は無効になります。

- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

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.client.connection.receive.timeout` パラメタに適用します。

- eztool put
- eztool get
- eztool remove
- eztool listgroup
- eztool listkey
- eztool removeall
- eztool execfunc

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.client.connection.receive.timeout` パラメタの最大値以上の値を指定した場合 `eads.client.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.client.connection.receive.timeout` パラメタの最小値以下の値を指定した場合 `eads.client.connection.receive.timeout` パラメタの最小値を適用します。

### (e) `eads.command.<サブコマンド名>.execution.timeout`

特定のサブ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。

サブコマンド名には、このパラメタの指定値を適用したいサブコマンド名を指定します。サブコマンド名を指定したサブコマンドの実行時には、コマンド定義の `eads.command.common.execution.timeout` パラメタの指定値よりも、このパラメタの指定値が優先されます。

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。

なお、次のコマンドではこのパラメタの指定は無効になります。

- eztool threaddump
- eztool snapshot
- eztool forcestop

## (3) クライアント定義ファイル

### (a) `eads.client.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.client.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機能の実行時間

### (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。

このパラメタの指定値には、サーバ定義の `eads.admin.operation.isolate.gracefulstop.waitTime` パラメタの指定値より小さい値を指定してください。

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。

### (d) `eads.client.eventListener.enable`

イベントリスナを使用するかどうかを指定します。

true :

イベントリスナを使用します。

false :

イベントリスナを使用しません。

true を指定すると、クラスタイベントリスナおよびノードイベントリスナのイベントが発生した場合に、EADS サーバが EADS クライアントに対してクラスタ構成情報の更新を通知します。

### (e) `eads.client.eventListener.reconnect.interval`

イベントリスナ用の接続が切断された場合に、接続を再接続する間隔（単位：ミリ秒）を指定します。



## 9.4 コマンドの運用に関連するパラメタの設計

コマンドの運用に関連するパラメタを設計します。

### 9.4.1 コマンドの運用に関連するパラメタ

コマンドの運用に関連するパラメタの一覧を次の表に示します。

表 9-5 コマンドの運用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	コマンド定義ファイル	<code>eads.command.compat</code>	0300	なし

#### (1) コマンド定義ファイル

##### (a) `eads.command.compat`

コマンドが互換動作するバージョンを指定します。

0300 を指定すると、コマンドの表示結果およびリターンコードが 03-60 以前のコマンドと同じになります。

0300 以外の値を設定した場合、またはこのパラメタの指定を省略した場合、表示結果およびリターンコードは、使用しているコマンドのバージョンに従います。



## 9.5 AP の運用に関連するパラメタの設計

AP の運用に関連するパラメタを設計します。

### 9.5.1 AP の運用に関連するパラメタ

AP の運用に関連するパラメタの一覧を次の表に示します。

表 9-6 AP の運用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	クライアント定義 ファイル	<code>eads.client.batchOperation.unit</code>	1~1024	10
2		<code>eads.client.compat</code>	0300	なし

#### (1) クライアント定義ファイル

##### (a) `eads.client.batchOperation.unit`

キャッシュ内のデータに対して一括操作を実行する際に、一度の EADS サーバとの通信で、何回データ操作を実行するかを指定します。このパラメタを指定することで、一括操作時の各リソースの使用量や使用率を調整できます。

このパラメタの指定値を大きくすることによる影響を次に示します。

- EADS サーバとの通信回数の減少による、性能向上が期待できます。
- 大量のデータ送信に伴い、ネットワークやメモリなどのリソースを占有することによって、性能に影響が出るおそれがあります。

#### 注意事項

環境によって適切な値が異なります。動作検証を行った上で、適切な値を指定してください。

このパラメタの指定値は、次に示す API で使用されます。

- Java 言語のクライアントライブラリで使用できるメソッド
  - Cache クラスの `putAll()`
  - Cache クラスの `getAll()` (Set 指定)
  - Cache クラスの `getAll()` (グループ指定)
- C 言語のクライアントライブラリで使用できる関数
  - `ead_put_all()`

- `ead_get_all()`
- `ead_get_group()`

次に示す API は、通信に value を含みません。このため、このパラメタの指定値に関係なく、一度の EADS サーバとの通信で、最大 1,024 回データ操作を実行します。

- Java 言語のクライアントライブラリで使用できるメソッド  
Cache クラスの `removeAll()` (Set 指定)
- C 言語のクライアントライブラリで使用できる関数  
`ead_remove_all()`

また、次に示す API は、このパラメタの指定値に関係なく、一度の EADS サーバとの通信で、対象の操作をすべて実行します。

- Java 言語のクライアントライブラリで使用できるメソッド
  - Cache クラスの `removeAll()` (グループ指定)
  - Cache クラスの `removeAll()` (EADS サーバ指定)
- C 言語のクライアントライブラリで使用できる関数
  - `ead_remove_group()`
  - `ead_remove_node()`

## (b) `eads.client.compat`

EADS クライアントが互換動作するバージョンを指定します。

0300 を指定すると、03-60 以前の EADS クライアントと同じエラーコードが返却されます。

0300 以外の値を設定した場合、またはこのパラメタの指定を省略した場合は、使用している EADS クライアントのバージョンのエラーコードが返却されます。

## 9.6 コンパクションに関連するパラメタの設計

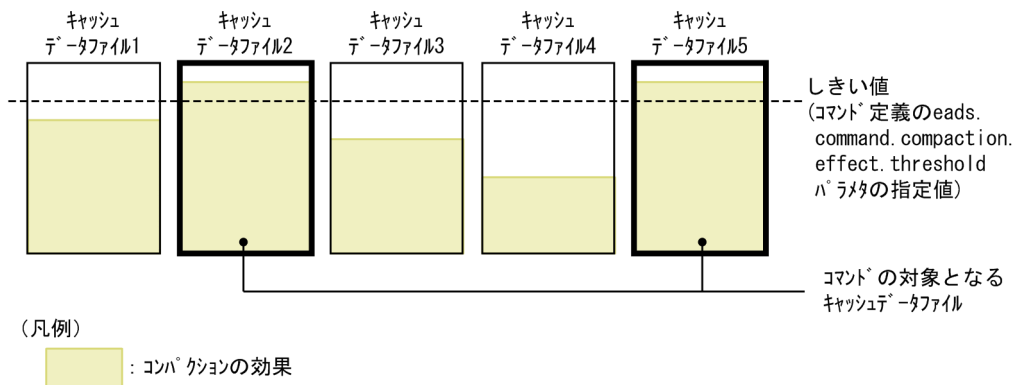
この節の内容は、ディスクキャッシュ、および2Way キャッシュを使用する場合にお読みください。

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。

### 9.6.1 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 9-5 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.command.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。

### 9.6.2 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 9-7 コンパクション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	コマンド定義ファイル	<code>eads.command.compaction.effect.threshold</code>	1~100 (%)	50
2		<code>eads.command.compaction.effect.division</code>	1~10	5

## (1) コマンド定義ファイル

### (a) `eads.command.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.command.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

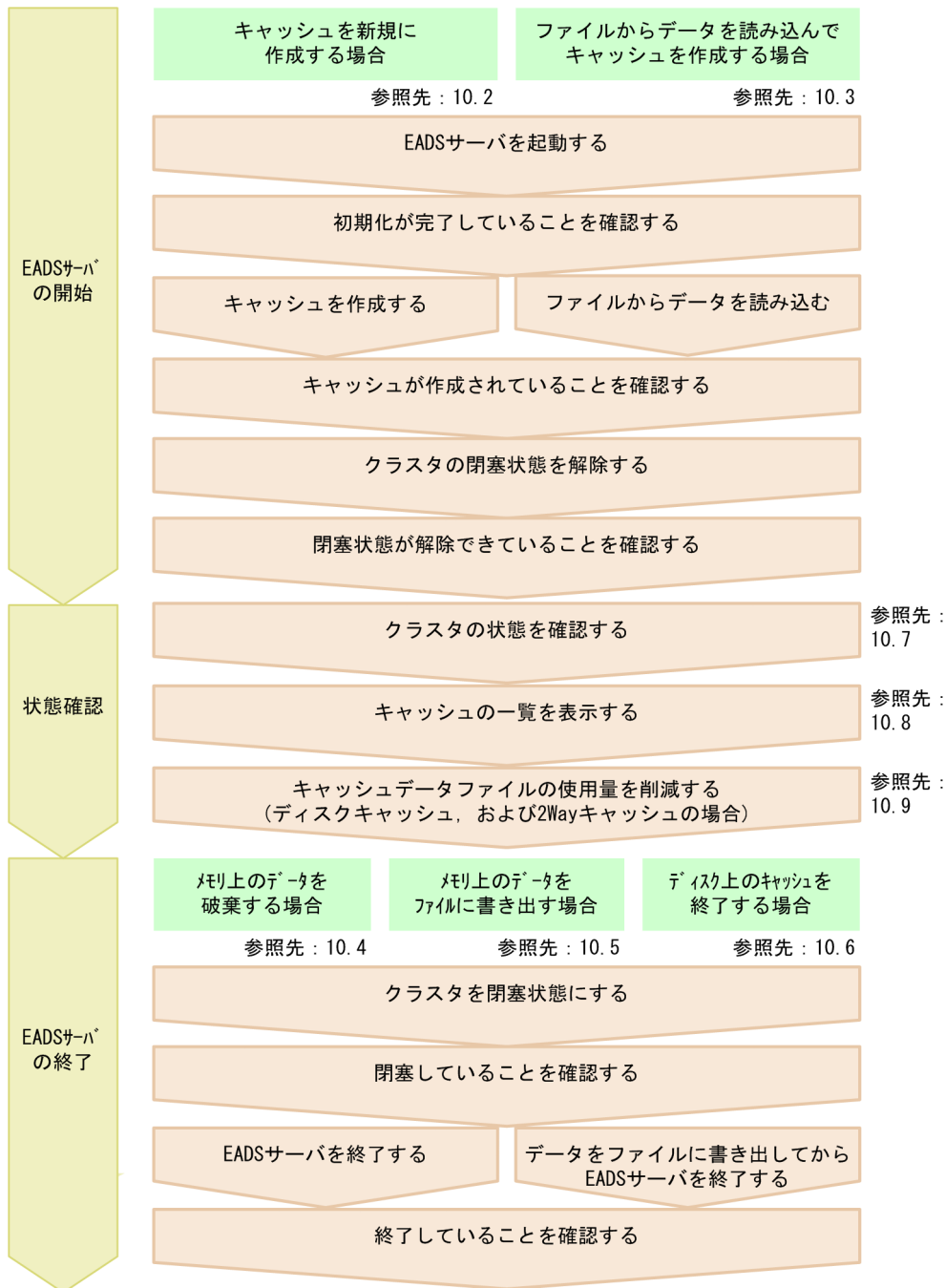
## 通常運用

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      Port  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 / 16

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      Port  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.eviction.policy
- eads.cache.eviction.keyCount (データ自動削除機能を使用する場合)
- eads.cache.disk.filesize
- eads.cache.disk.filenum
- eads.cache.disk.blocksize
- eads.cache.disk.persistentMemory.enable
- eads.cache.disk.persistentMemory.blocksize (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)

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      Port  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。

### 通常の場合（ディスクキャッシュの格納先として不揮発性メモリを使用しない場合）

- 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>）

### ディスクキャッシュの格納先として不揮発性メモリを使用する場合

- 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※  
 <キャッシュ定義の eads.cache.disk.persistentMemory.device パラメタで指定した不揮発性メモリのデバイスファイル>
- 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>)

#### 注※

指定したデバイスファイルに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格納されます。

## (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 / 16

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      Port  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。

#### 参考

きめ細かな運用ができない例を次に示します。

EADS では、キャッシュ単位に操作できるコマンドは、データ操作 (eztool put など) やユーザ機能 (eztool listfunc など) に限られています。

例えば、キャッシュ単位で閉塞状態にする (`eztool close`) / 閉塞状態を解除する (`eztool open`) 操作や、データを書き出す (`eztool export`) / データを読み込む (`eztool import`) 操作はできません。そのため、2つのキャッシュ (キャッシュ 1, キャッシュ 2) がある場合、次のようにキャッシュ単位で運用を分けられません。

- `eztool export` コマンドを実行して、キャッシュ 1 だけ、データのバックアップを取得する。
- `eztool close` コマンドを実行して、キャッシュ 2 だけを閉塞状態にすることで、クライアントからのリクエストを止める。

また、キャッシュ単位の設定ができるパラメタは、キャッシュ定義ファイルに指定できるパラメタだけです。そのため、次のような設定はできません。

- データの多重度 (`eads.replication.factor`) をキャッシュ単位で変える。
- 格納できる key の最大サイズ (`eads.cache.key.maxsize`) をキャッシュ単位で変える。

## 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
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      Port  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) 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、共通設定の `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。

また、`eztool import` コマンドを実行したときにメモリ容量が不足すると、`eztool import` コマンドがハングアップすることがあります。`eztool import` コマンドがハングアップしたときは、EADS サーバのメッセージログに、メモリ容量不足のエラーが出力されていないかを確認してください。

### (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 / 16

CacheName  CacheType  TotalMasterKeyCount
cache1     Memory     99764
cache2     Memory     99910
cache3     Memory     100326
-----
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      Port  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      Port  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 / 16

CacheName  CacheType  TotalMasterKeyCount
cache1     Disk       99764
cache2     Disk       99910
cache3     Disk       100326
-----
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      Port  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      Port  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      Port  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      Port  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      Port  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 Port  Position    Cluster  State  Operation  Lock  KeyCount  UsedMemoryRatio  UsedMemory  MaxMemory  Version
 1 XX.XXX.XXX.168  server01  24600  1288490189 online  running  none  unlock   582        13           3           23  04-00-00
 2 XX.XXX.XXX.168  server02  24700  429496730  online  running  none  unlock   618        13           3           23  04-00-00
 3 XX.XXX.XXX.168  server03  24800  -429496729 online  running  none  unlock   636        13           3           23  04-00-00
 4 XX.XXX.XXX.168  server04  24900  -1288490188 online  running  none  unlock   591        13           3           23  04-00-00
 5 XX.XXX.XXX.168  server05  25000  -2147483648 online  running  none  unlock   573        13           3           23  04-00-00
-----

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 / 16

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 / 16

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.
$
```

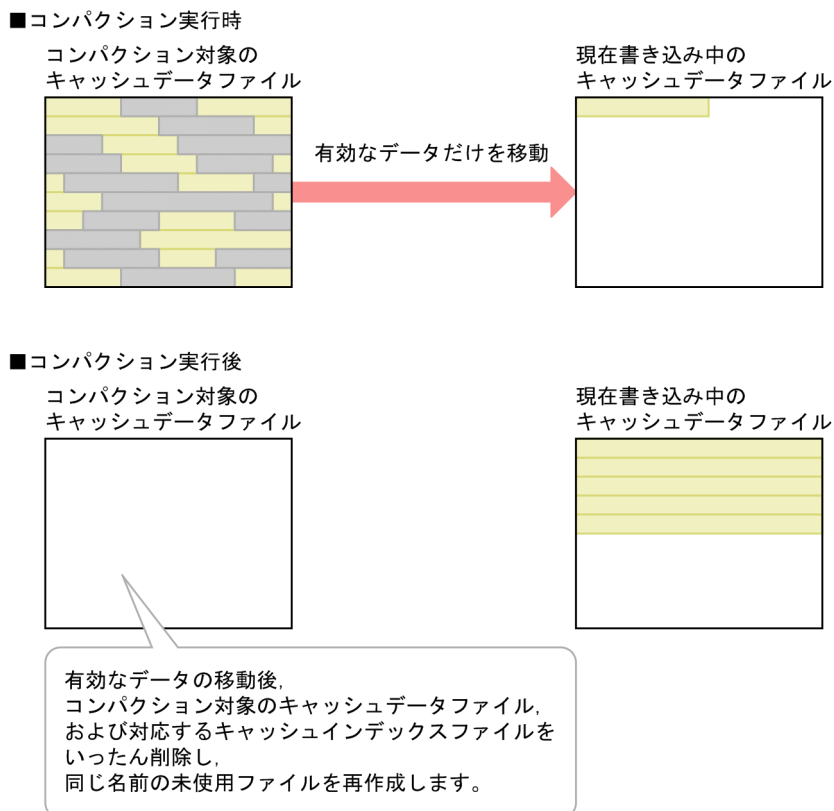
## 10.9 キャッシュデータファイルの使用量を削減する（キャッシュデータファイルをコンパクションする）

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



(凡例)

■ : 有効なデータ    ■ : 無効なデータ

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      Port  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保しておく必要があります。このような運用の場合は、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。確保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      Port  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 クラスタを停止させないで、EADS サーバをクラスタに追加する手順（スケールアウト処理・リバランス処理）

クラスタを停止させないで、次の処理を実行するときの手順について説明します。

- EADS サーバをクラスタに追加する（スケールアウト処理）
- EADS サーバの位置を変更することで、各レンジに格納できる key の数を均等に配分し直す（リバランス処理）

スケールアウト処理を実行すると、各レンジに格納できる key の数が均等に配分されなくなるおそれがあります。スケールアウト処理後は、リバランス処理を実行して、各レンジに格納できる key の数を均等に配分し直すことを推奨します。

#### (1) 前提条件

スケールアウト処理およびリバランス処理を実行するための前提条件を次に示します。

##### スケールアウト処理を実行するための前提条件

スケールアウト処理を実行するには、次に示す条件を満たしている必要があります。

- クラスタ内のキャッシュがすべてメモリキャッシュである
- データの多重度が 2 以上である
- クラスタ内に縮退状態の EADS サーバがない

これらの条件を満たさない場合には、クラスタを停止してから、EADS サーバを追加してください。なお、縮退した EADS サーバがある場合は、縮退した EADS サーバを復旧してから、スケールアウト処理を実行してください。

クラスタを停止してから EADS サーバを追加する手順については、「[11.1.2 クラスタを停止してから、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）](#)」および「[11.1.3 クラスタを停止してから、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。

#### ポイント

各レンジに key が均等に分散している場合、各レンジに、1 台以上の同じ台数の EADS サーバを追加すると、リバランス処理を実行しなくても、各レンジに格納できる key の数を均等にできます。

## リバランス処理を実行するための前提条件

リバランス処理を実行するためには、次に示す条件を満たしている必要があります。

- クラスタ内のキャッシュがすべてメモリキャッシュである
- データの多重度が 2 以上である
- クラスタ内に縮退状態の EADS サーバがない
- クラスタを構成する EADS サーバ数が「データの多重度×2-1」より多い
- クラスタ内のすべての EADS サーバで、データを格納するメモリ容量に余裕がある

これらの条件を満たさない場合には、クラスタを停止してから、各レンジに格納できる key の数を均等にしてください。なお、縮退した EADS サーバがある場合は、縮退した EADS サーバを復旧してから、リバランス処理を実行してください。

クラスタを停止してから各レンジに格納できる key の数を均等にする手順については、「[11.3.2 クラスタを停止してから、各レンジに格納できる key の数を均等にする手順](#)」を参照してください。

### ポイント

各レンジに key が均等に分散している状態で、かつ各 EADS サーバが担当しているレンジに差がある場合に、リバランス処理の効果が得られます。

例えば、各レンジが均等なクラスタに対して、スケールアウト処理後にリバランス処理を実行すると効果があります。

なお、キャッシュが複数ある場合は、任意のキャッシュが次の条件に該当するときは、リバランス処理の効果が得られないことがあります。

- 各レンジに key が均等に分散していない。
- key の数やグループの数が極端に少ない。

### ポイント

リバランス処理中は EADS サーバ間でのデータ転送が発生します。そのため、データあふれによる縮退を防止するために、データ総量監視機能を有効にすることを推奨します。

### 参考

データ総量監視機能を有効にしている場合、リバランス処理の実行前に、スケールアウト処理前のクラスタのメモリ容量を超えたデータを格納すると、リバランス処理が失敗することがあります。

また、スケールアウト処理前のクラスタのメモリ容量を超えない場合でも、クラスタ内の各レンジに格納されている key の数に偏りがあると、リバランス処理が中断されることがあります。

そのため、スケールアウト処理後には速やかにリバランス処理を実行してください。スケールアウト処理とリバランス処理を組み合わせると、クラスタ全体でメモリ容量を均等に増加させてから、クラスタに格納するデータの容量を増やすことを推奨します。

リバランス処理が中断された場合は、クラスタに格納するデータの容量を減らすか、またはスケールアウト処理でさらに EADS サーバを追加することを検討してください。

## 注意事項

データ自動削除機能を使用しているキャッシュとデータ自動削除機能を使用していないキャッシュが混在している場合、スケールアウト処理およびリバランス処理は実行できません。対象のキャッシュを削除できる場合は、データ自動削除機能を使用しているキャッシュと、データ自動削除機能を使用していないキャッシュのどちらかを削除して、キャッシュが混在しないようにしてください。そのあとで、再度スケールアウト処理およびリバランス処理を実行してください。

## (2) 各レンジのデータの分散状況を確認する

`eztool storeusage --replica` コマンドを実行して、各レンジのデータの分散状況を確認します。

```
eztool storeusage --replica
```

### コマンド実行例

```
$ eztool storeusage --replica
KDEA08001-1 The command will now start. (subcommand = storeusage, parameter = [storeusage, --replica])
RangeID  StartPosition  EndPosition  Server  StoredExternalHeapSize  KeyCount (Server ID:1 Position:1288490189)  KeyCount (Server ID:2 Position:429496730)
1  1288490189  2147483647  XX.XXX.XXX.168:24600  1  218  218
2  429496730  1288490188  XX.XXX.XXX.168:24700  1  209  209
3  -429496729  429496729  XX.XXX.XXX.168:24800  1  173  173
4  -1288490188  -429496730  XX.XXX.XXX.168:24900  1  191  191
5  -2147483648  -1288490189  XX.XXX.XXX.168:25000  1  191  191
KDEA08002-1 The command will now end.
$
```

`eztool storeusage --replica` コマンドの実行結果を基に、どのレンジに EADS サーバを追加するかを決めてください。

## (3) クラスタ定義ファイルのバックアップを取得する (任意)

必要に応じて、クラスタ定義ファイルのバックアップを取得します。

稼働中のクラスタに EADS サーバを追加する場合、クラスタ定義ファイルは自動的に EADS サーバの追加後の情報に書き換えられます。また、リバランス処理で EADS サーバの位置を変更する際にも、変更後の EADS サーバの情報に書き換えられます。

書き換えられる前のクラスタ定義ファイルのバックアップファイルは、次に示すパスに自動的に作成されます。ただし、同名のファイルがある場合は出力されません。

```
<運用ディレクトリ>/conf/eads_cluster.properties.ebf
```

クラスタ定義ファイルのバックアップファイルが存在すると、スケールアウト処理やリバランス処理の実行時に、クラスタ定義ファイルはEADS サーバ追加・変更後の情報に上書きされます。そのため、スケールアウト処理やリバランス処理の実行後に、ユーザがクラスタ定義ファイルを編集し、再度スケールアウト処理やリバランス処理を実行すると、編集した内容が消えてしまいます。編集したクラスタ定義ファイルのバックアップが必要な場合は、スケールアウト処理やリバランス処理の実行前に、手動でバックアップを取得してください。

## (4) 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` コマンドを実行して、クラスタ内に縮退状態 (isolated) または停止状態 (-----) の EADS サーバがないことを確認します。

```
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      Port  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.
$
```

クラスタ内に縮退状態または停止状態の EADS サーバが存在する場合は、EADS サーバを復旧してください。

復旧の手順については、「[12.2.1 縮退状態が発生した場合](#)」を参照してください。

## (5) 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。



ただし、クラスタ定義ファイルは EADS サーバ追加時に自動で作成されるため、作成する必要はありません（追加する EADS サーバにクラスタ定義ファイルがあっても読み込まれません）。

## (6) EADS サーバをクラスタに追加する

追加する EADS サーバで `ezstart` コマンドまたは `ezserver` コマンドを実行して、EADS サーバをクラスタに追加します。

次のどれかの方法で、EADS サーバをクラスタに追加します。

- EADS サーバ ID を指定して EADS サーバをクラスタに追加する場合
  - 追加する EADS サーバで、`ezstart -ai` コマンドを実行する
  - 追加する EADS サーバで、`ezserver -ai` コマンドを実行する
- EADS サーバの位置（ハッシュ値）を指定して EADS サーバをクラスタに追加する場合
  - 追加する EADS サーバで、`ezstart -ap` コマンドを実行する
  - 追加する EADS サーバで、`ezserver -ap` コマンドを実行する

EADS サーバ ID を指定して EADS サーバをクラスタに追加する場合（`ezstart -ai` コマンドを実行する）

```
ezstart -ai <EADSサーバID>
```

EADS サーバ ID を指定して EADS サーバをクラスタに追加する場合（`ezserver -ai` コマンドを実行する）

```
ezserver -ai <EADSサーバID>
```

EADS サーバの位置を指定して EADS サーバをクラスタに追加する場合（`ezstart -ap` コマンドを実行する）

```
ezstart -ap <EADSサーバの位置（ハッシュ値）>
```

EADS サーバの位置を指定して EADS サーバをクラスタに追加する場合（`ezserver -ap` コマンドを実行する）

```
ezserver -ap <EADSサーバの位置（ハッシュ値）>
```

EADS サーバの追加後に、何らかの要因でクラスタ定義ファイルが書き換えられなかった場合、エラーメッセージが表示されます。この場合は、`eztool listconf` コマンドを実行して最新のパラメタの値を確認し、手動でクラスタ定義ファイルに反映してください。

## (7) 追加した 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` コマンドで、追加した 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: 6
OnlineCount: 6
OfflineCount: 0
StandbyCount: 0

ID  IP Address      ServerName  Port  Position  Cluster  State  Operation  Lock  KeyCount  UsedMemoryRatio  UsedMemory  MaxMemory  Version
1  XX.XXX.XXX.168  server01   24600  1288490189 online  running  none  unlock  1317      27              6           23         04-50-00
2  XX.XXX.XXX.168  server02   24700  429496730  online  running  none  unlock  1352      28              6           23         04-50-00
6  XX.XXX.XXX.168  server06   25100  0          online  running  none  unlock  1083      22              5           23         04-50-00
3  XX.XXX.XXX.168  server03   24800  -429496729 online  running  none  unlock  883       18              4           23         04-50-00
4  XX.XXX.XXX.168  server04   24900  -1288490188 online  running  none  unlock  848       18              4           23         04-50-00
5  XX.XXX.XXX.168  server05   25000  -2147483648 online  running  none  unlock  1117     23              5           23         04-50-00

-----
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。

Position 欄、および KeyCount 欄の値を参照すると、各 EADS サーバの位置、および各レンジに格納できる key の数に偏りがあるかどうかを確認できます。レンジに格納できる key の数に偏りがある場合は、リバランス処理を実行して、偏りを解消してください。

複数台の EADS サーバを追加する場合は、追加する各 EADS サーバに対して、「[11.1.1\(5\) 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](#)」以降の手順を実行してください。

## (8) EADS サーバの位置を変更して、各レンジに格納できる key の数を均等にする (リバランス処理)

任意の EADS サーバで `eztool rebalance` コマンドを実行して、各レンジに格納できる key の数を均等にします。

```
eztool rebalance
```

## コマンド実行例

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

コマンドを実行すると、各 EADS サーバの位置がコンシステント・ハッシングの円周上に均等に配置し直されて、各レンジに格納できる key の数が均等になります。

## (9) リバランス処理の結果を確認する

`eztool status -v` コマンドを実行して、リバランス処理の結果を確認します。

```
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: 6
OnlineCount: 6
OfflineCount: 0
StandbyCount: 0

ID IP_Address ServerName Port Position Cluster State Operation Lock KeyCount UsedMemoryRatio UsedMemory MaxMemory Version
5 XX.XXX.XXX.168 server05 25000 1861152496 online running none unlock 1114 23 5 23 04-50-00
1 XX.XXX.XXX.168 server01 24600 1145324613 online running none unlock 1120 23 5 23 04-50-00
2 XX.XXX.XXX.168 server02 24700 429496730 online running none unlock 1118 23 5 23 04-50-00
6 XX.XXX.XXX.168 server06 25100 -286331152 online running none unlock 1086 23 5 23 04-50-00
3 XX.XXX.XXX.168 server03 24800 -1002159034 online running none unlock 1080 23 5 23 04-50-00
4 XX.XXX.XXX.168 server04 24900 -1717986917 online running none unlock 1082 23 5 23 04-50-00

KDEA08002-I The command will now end.
$

```

Position 欄、および KeyCount 欄の値を参照すると、各 EADS サーバの位置の偏り、および各レンジに格納できる key の数の偏りが解消されていることがわかります。

### 11.1.2 クラスタを停止してから、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、クラスタを停止してから 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、「[11.4 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。

### 11.1.3 クラスタを停止してから、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、クラスタを停止してから 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      Port  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。

キャッシュの種類がディスクキャッシュまたは 2Way キャッシュのキャッシュを確認します。

```
eztool listcache
```

#### コマンド実行例

```

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

CacheCount: 3 / 16

CacheName  CacheType  TotalMasterKeyCount
cache1     Disk       0
cache2     Disk       0
cache3     Disk       0
-----

KDEA08002-I          The command will now end.
$

```

#### ポイント

メモリキャッシュと、2Way キャッシュまたはディスクキャッシュを混在させて使用している場合、メモリキャッシュのデータを保持するため、クラスタ内の全 EADS サーバを終了する前に `eztool export` コマンド、または `eztool stop` コマンドを実行してメモリキャッシュのデータを書き出し、クラスタ内の全 EADS サーバを開始したあとで、`eztool import` コマンドを実行してデータを読み込む必要があります。

## (4) EADS サーバからキャッシュファイルを書き出す

eztool exporcecf コマンドを実行して、EADS サーバから「(3) キャッシュを確認する」で確認したすべてのキャッシュの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、任意のディレクトリに書き出します。

### 参考

eztool exporcecf コマンドを実行する代わりに、「(16) 【参考】 OS のコマンドなどを使用して EADS サーバのキャッシュファイルを移動する」の手順で EADS サーバのキャッシュファイルを移動・退避させてもかまいません。

## (5) 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確認する

キャッシュファイルの出力先を確認して、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が取得できているかを確認します。

## (6) 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 deletecache --with\_deleteecf コマンドを実行する
- eztool deletecache コマンドを実行したあとに eztool deleteecf コマンドを実行する

eztool deletecache --with\_deleteecf コマンドを実行する場合

```
eztool deletecache <キャッシュ名> --with_deleteecf
```

### コマンド実行例

```
$ eztool deletecache cache1 --with_deleteecf
KDEA08001-I      The command will now start. (subcommand = deletecache, parameter = [deletecache, cache1,
--with_deleteecf])
KDEA08064-I      The cache files were deleted.
KDEA08002-I      The command will now end.
$
```

eztool deletecache 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 deletecache <キャッシュ名>
eztool deleteecf <キャッシュ名>
```

### コマンド実行例

```

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

```

## (7) キャッシュが削除されていることを確認する

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-I 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 5 100 1 7 0 0 0 1
  3 XX.XXX.XXX 168:24800 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
  XX.XXX.XXX 168:25000 true false false 6 0 0 8 0 0 0 0
  4 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
  5 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 5 100 1 7 0 0 0 1
  3 XX.XXX.XXX 168:24800 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
  4 XX.XXX.XXX 168:24900 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
  5 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 168:24700 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 6 0 0 8 0 0 0 0
  XX.XXX.XXX 168:24800 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
  3 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
  4 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 168:24600 true true true 6 0 0 8 0 0 0 0
  5 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-I The command will now end.
$

```

- eztool deletecache --with\_deletectf コマンド、または eztool deletecache コマンドのあとに eztool deletectf 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。

## (8) 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。

## (9) 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。

## (10) 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、「[11.4 定義内容を変更したい](#)」を参照してください。

## (11) クラスタ内の全 EADS サーバを開始する

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
```

## (12) 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      Port  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。

## (13) 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、「11.1.3(4) EADS サーバからキャッシュファイルを書き出す」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格納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.
$

```

## (14) 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.
$
```

## (15) 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      Port  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。

## (16) 【参考】 OS のコマンドなどを使用して EADS サーバのキャッシュファイルを移動する

「(4) EADS サーバからキャッシュファイルを書き出す」で、`eztool exportecf` コマンドを実行して EADS サーバからキャッシュファイルを書き出す代わりに、OS のコマンドなどを使用して EADS サーバのキャッシュファイルを移動させてもかまいません。その場合の 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）を次に示します。

1. クラスタを閉塞状態にする
2. 閉塞していることを確認する
3. キャッシュを確認する
4. クラスタ内の全 EADS サーバを終了する
5. 全 EADS サーバのキャッシュファイルを移動する
6. 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
7. クラスタ定義を変更する
8. クラスタ内の全 EADS サーバを開始する
9. 初期化が完了していることを確認する
10. データを再配置する
11. クラスタの閉塞状態を解除する
12. 閉塞状態が解除できていることを確認する

## 注意事項

この手順はディスクキャッシュの格納先として不揮発性メモリを使用している場合は使用できません。

「5. 全 EADS サーバのキャッシュファイルを移動する」の手順を次に示します。

### (a) 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。

「(13) データを再配置する」で `eztool importecf` コマンドを実行して、データを再配置する際には、上記でキャッシュファイルを移動させたディレクトリを指定してください。

## 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) 各レンジのデータの分散状況を確認する

`eztool storeusage --replica` コマンドを実行して、各レンジのデータの分散状況を確認します。

```
eztool storeusage --replica
```

#### コマンド実行例

```
$ eztool storeusage --replica
KDEA08001-I The command will now start. (subcommand = storeusage, parameter = [storeusage, --replica])
RangeID StartPosition EndPosition Server StoredExternalHeapSize KeyCount (ServerID:1 Position:1288490189) KeyCount (ServerID:2 Position:429496730)
1 1288490189 2147483647 XX.XXX.XXX.168:24600 1 218
2 429496730 1288490188 XX.XXX.XXX.168:24700 1 209
3 -429496729 429496729 XX.XXX.XXX.168:24800 1
4 -1288490188 -429496730 XX.XXX.XXX.168:24900 1 173
5 -2147483648 -1288490189 XX.XXX.XXX.168:25000 1 191
KDEA08002-I The command will now end.
$
```

`eztool storeusage --replica` コマンドの実行結果を基に、どのレンジから EADS サーバを削除するかを決めてください。

#### (2) 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。

#### (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。

全 EADS サーバのクラスタ定義ファイルから、削除する EADS サーバの次に示すパラメタを削除します。

- `eads.node.< EADS サーバ ID >.address`
- `eads.node.< EADS サーバ ID >.port`

- eads.node.< EADS サーバ ID >.position

#### (4) 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。

#### (5) 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) 各レンジのデータの分散状況を確認する

eztool storeusage --replica コマンドを実行して、各レンジのデータの分散状況を確認します。

```
eztool storeusage --replica
```

#### コマンド実行例

```
$ eztool storeusage --replica
KDEA08001-1 The command will now start. (subcommand = storeusage, parameter = [storeusage, --replica])
RangeID StartPosition EndPosition Server StoredExternalHeapSize KeyCount (Server ID:1 Position:1288490189) KeyCount (Server ID:2 Position:429496730)
1 1288490189 2147483647 XX.XXX.XXX.168:24600 1 218 218
2 429496730 1288490188 XX.XXX.XXX.168:24700 1 209 209
3 429496729 429496729 XX.XXX.XXX.168:24800 1 173 173
4 -1288490188 -429496730 XX.XXX.XXX.168:24900 1 191 191
5 -2147483648 -1288490189 XX.XXX.XXX.168:25000 1 191 191
-----
KDEA08002-1 The command will now end.
$
```

eztool storeusage --replica コマンドの実行結果を基に、どのレンジから EADS サーバを削除するかを決めてください。

## (2) 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.
$
```

## (3) 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      Port  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。

キャッシュの種類がディスクキャッシュまたは 2Way キャッシュのキャッシュを確認します。

```
eztool listcache
```

### コマンド実行例

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

CacheCount: 3 / 16

CacheName  CacheType  TotalMasterKeyCount
cache1     Disk       0
cache2     Disk       0
cache3     Disk       0
-----
KDEA08002-I          The command will now end.
$
```

## ポイント

メモリキャッシュと、2Way キャッシュまたはディスクキャッシュを混在させて使用している場合、メモリキャッシュのデータを保持するため、クラスタ内の全 EADS サーバを終了する前に `eztool export` コマンド、または `eztool stop` コマンドを実行してメモリキャッシュのデータを書き出し、クラスタ内の全 EADS サーバを開始したあとで、`eztool import` コマンドを実行してデータを読み込む必要があります。

また、削除する EADS サーバのストアデータファイルは、削除しない EADS サーバのストアデータファイルの格納先ディレクトリに退避しておく必要があります。

## (5) EADS サーバからキャッシュファイルを書き出す

`eztool exportecf` コマンドを実行して、EADS サーバから「(4) キャッシュを確認する」で確認したすべてのキャッシュの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、任意のディレクトリに書き出します。

## 参考

`eztool exportecf` コマンドを実行する代わりに、「(13) 【参考】 OS のコマンドなどを使用して EADS サーバのキャッシュファイルを移動・退避する」の手順で EADS サーバのキャッシュファイルを移動・退避させてもかまいません。

## (6) 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確認する

キャッシュファイルの出力先を確認して、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が取得できているかを確認します。

## (7) 削除する EADS サーバのキャッシュファイルを退避する

「(5) EADS サーバからキャッシュファイルを書き出す」で書き出したキャッシュファイルのうち、削除する EADS サーバで書き出したすべて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、「(5) EADS サーバからキャッシュファイルを書き出す」でキャッシュファイルを書き出したディレクトリに退避します。

## (8) 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 deletecache --with_deleteecf` コマンドを実行する
- `eztool deletecache` コマンドを実行したあとに `eztool deleteecf` コマンドを実行する

`eztool deletecache --with_deleteecf` コマンドを実行する場合

```
eztool deletecache <キャッシュ名> --with_deleteecf
```

コマンド実行例

```
$ eztool deletecache cache1 --with_deleteecf
KDEA08001-I The command will now start. (subcommand = deletecache, parameter = [deletecache, cache1,
--with_deleteecf])
KDEA08064-I The cache files were deleted.
KDEA08002-I The command will now end.
$
```

`eztool deletecache` 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 deletecache <キャッシュ名>
eztool deleteecf <キャッシュ名>
```

コマンド実行例

```
$ eztool deletecache cache1
KDEA08001-I The command will now start. (subcommand = deletecache, parameter = [deletecache, cache1])
KDEA08002-I The command will now end.
$ eztool deleteecf cache1
KDEA08001-I The command will now start. (subcommand = deleteecf, parameter = [deleteecf, cache1])
KDEA08064-I The cache files were deleted.
KDEA08002-I The command will now end.
$
```

## (9) キャッシュが削除されていることを確認する

`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
  XX.XXX.XXX 168:24700 true false false
  XX.XXX.XXX 168:24800 true false false
  2 XX.XXX.XXX 168:24700 true false false
  XX.XXX.XXX 168:24800 true false false
  XX.XXX.XXX 168:24900 true false false
  3 XX.XXX.XXX 168:24800 true false false
  XX.XXX.XXX 168:24900 true false false
  XX.XXX.XXX 168:25000 true false false
  4 XX.XXX.XXX 168:24900 true false false
  XX.XXX.XXX 168:25000 true false false
  XX.XXX.XXX 168:24600 true false false
  5 XX.XXX.XXX 168:25000 true false false
  XX.XXX.XXX 168:24600 true false false
  XX.XXX.XXX 168:24700 true false false
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 5 100 1 7 0 0 0 1
  3 XX.XXX.XXX 168:24800 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
  4 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 168:24600 true true true 6 0 0 8 0 0 0 0
  5 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
  XX.XXX.XXX 168:24700 true true true
  XX.XXX.XXX 168:24800 true true true
  2 XX.XXX.XXX 168:24700 true true true
  XX.XXX.XXX 168:24800 true true true
  XX.XXX.XXX 168:24900 true true true
  3 XX.XXX.XXX 168:24800 true true true
  XX.XXX.XXX 168:24900 true true true
  XX.XXX.XXX 168:25000 true true true
  4 XX.XXX.XXX 168:24900 true true true
  XX.XXX.XXX 168:25000 true true true
  XX.XXX.XXX 168:24600 true true true
  5 XX.XXX.XXX 168:25000 true true true
  XX.XXX.XXX 168:24600 true true true
  XX.XXX.XXX 168:24700 true true true
-----
KDEA08002-1 The command will now end.
$

```

- `eztool deletecache --with_deleteecf` コマンド, または `eztool deletecache` 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 deletecache` 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。

## (10) 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。

## (11) 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。

全 EADS サーバのクラスタ定義ファイルから、削除する EADS サーバの次に示すパラメタを削除します。

- eads.node.< EADS サーバ ID >.address
- eads.node.< EADS サーバ ID >.port
- eads.node.< EADS サーバ ID >.position

## (12) 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。

eztool importecf コマンドを実行して、データを再配置する際には、「(5) EADS サーバからキャッシュファイルを書き出す」でキャッシュファイルを書き出したディレクトリを指定してください。

手順については、「11.1.3(11) クラスタ内の全 EADS サーバを開始する」～「11.1.3(15) 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。

## (13) 【参考】 OS のコマンドなどを使用して EADS サーバのキャッシュファイルを移動・退避する

「(5) EADS サーバからキャッシュファイルを書き出す」で、eztool exportecf コマンドを実行して EADS サーバからキャッシュファイルを書き出す代わりに、OS のコマンドなどを使用して EADS サーバのキャッシュファイルを移動・退避させてもかまいません。その場合の 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）を次に示します。

1. 各レンジのデータの分散状況を確認する
2. クラスタを閉塞状態にする
3. 閉塞していることを確認する
4. キャッシュを確認する
5. クラスタ内の全 EADS サーバを終了する
6. 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移動する
7. 削除する EADS サーバのキャッシュファイルを退避する
8. クラスタ定義を変更する
9. 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）

## 注意事項

この手順はディスクキャッシュの格納先として不揮発性メモリを使用している場合は使用できません。

「6. 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移動する」、および「7. 削除する EADS サーバのキャッシュファイルを退避する」の手順を次に示します。

### (a) 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。

「(12) 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）」で、`eztool importecf` コマンドを実行して、データを再配置する際には、上記でキャッシュファイルを移動させたディレクトリを指定してください。

### (b) 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、「(a) 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退避してください。

- ・ <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>)

## 11.3 各レンジに格納できる key の数を均等にしたい

スケールアウト処理などによって、各レンジに格納できる key の数が偏っている場合に、クラスタ内の EADS サーバをコンシステント・ハッシングの円周上に均等に配置し直すことで、各レンジに格納できる key の数を均等にします。

### 11.3.1 クラスタを停止させないで、各レンジに格納できる key の数を均等に する手順（リバランス処理）

クラスタを停止させないで、EADS サーバの位置を変更して、各レンジに格納できる key の数を均等に  
する手順（リバランス処理）について説明します。

リバランス処理は、スケールアウト処理後など、各レンジに格納できる key の数が均等に配分されてい  
ない場合に実行します。

リバランス処理の実行条件については、「[11.1.1\(1\) 前提条件](#)」を参照してください。

リバランス処理を実行する前に次の手順を実行してください。

- 「[11.1.1\(2\) 各レンジのデータの分散状況を確認する](#)」
- 「[11.1.1\(3\) クラスタ定義ファイルのバックアップを取得する（任意）](#)」
- 「[11.1.1\(4\) クラスタ内の全 EADS サーバに異常がないことを確認する](#)」

また、リバランス処理の手順は、次の手順を参照してください。

- 「[11.1.1\(8\) EADS サーバの位置を変更して、各レンジに格納できる key の数を均等に  
する（リバランス処理）](#)」
- 「[11.1.1\(9\) リバランス処理の結果を確認する](#)」

### 11.3.2 クラスタを停止してから、各レンジに格納できる key の数を均等に する手順

メモリキャッシュだけを使用している場合に、クラスタを停止してから、EADS サーバの位置を変更して、  
各レンジに格納できる key の数を均等に  
する手順について説明します。

メモリキャッシュだけを使用している場合には、「[11.3.1 クラスタを停止させないで、各レンジに格納で  
きる key の数を均等に  
する手順（リバランス処理）](#)」を実行することを推奨します。

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、「[11.1.3 クラスタを停止してか  
ら、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）](#)」の手順を実行し  
てください。

## (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) クラスタ定義を変更する

EADS サーバの位置 (ハッシュ値) を自動で調整するために、各 EADS サーバのクラスタ定義を変更します。

各 EADS サーバのクラスタ定義ファイルから、`eads.node.< EADS サーバ ID >.position` パラメータをすべて削除します。

## (3) 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。

## 11.4 定義内容を変更したい

定義内容を変更します。

### 11.4.1 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）](#)」、または「[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。

#### 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。

クラスタ定義または共通設定を変更する場合は、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.eviction.policy`
- `eads.cache.eviction.keyCount`（データ自動削除機能を使用する場合）
- `eads.cache.disk.filesize`
- `eads.cache.disk.filenum`
- `eads.cache.disk.blocksize`
- `eads.cache.disk.persistentMemory.enable`

- `eads.cache.disk.persistentMemory.blocksize` (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)

「`eads.command.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.command.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。

## 11.4.2 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、「11.1.3 クラスタを停止してから、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タ
  - `eads.node.< EADS サーバ ID >.position`
- 共通設定の次のパラメタ
  - `eads.replication.factor`
  - `eads.cache.key.maxsize`

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タ
  - `eads.cache.type`
  - `eads.cache.eviction.policy`
  - `eads.cache.disk.filesize`
  - `eads.cache.disk.persistentMemory.enable`
  - `eads.cache.disk.persistentMemory.device` (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)
  - `eads.cache.disk.persistentMemory.blocksize` (ディスクキャッシュの格納先として不揮発性メモリを使用する場合)

注 キャッシュ定義の `eads.cache.eviction.keyCount` パラメタおよび `eads.cache.disk.filenum` パラメタについては、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。



## 11.5 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。

### 11.5.1 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      Port  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 deletetocache` コマンドを実行し、キャッシュを削除します。

```
eztool deletetocache <キャッシュ名>
```

#### コマンド実行例

```
$ eztool deletetocache cache1
KDEA08001-I      The command will now start. (subcommand = deletetocache, parameter = [deletetocache, 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 / 16

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      Port  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。

## 11.5.2 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      Port  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。

### 通常の場合（ディスクキャッシュの格納先として不揮発性メモリを使用しない場合）

- 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>）

### ディスクキャッシュの格納先として不揮発性メモリを使用する場合

- 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※  
<キャッシュ定義の eads.cache.disk.persistentMemory.device パラメタで指定した不揮発性メモリのデバイスファイル>
- 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>）

#### 注※

指定したデバイスファイルに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格納されます。

## (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 deletecache --with_deleteecf` コマンドを実行する
- `eztool deletecache` コマンドを実行したあとに `eztool deleteecf` コマンドを実行する

### eztool deletecache --with\_deleteecf コマンドを実行する場合

```
eztool deletecache <キャッシュ名> --with_deleteecf
```

## コマンド実行例

```
$ eztool deletetecache cache1 --with_deletetecf
KDEA08001-I The command will now start. (subcommand = deletetecache, parameter = [deletetecache, cache1,
--with_deletetecf])
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 deletetecf コマンドを実行する場合

```
eztool deletetecache <キャッシュ名>
```

```
eztool deletetecf <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 / 16

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
XX.XXX.XXX 168:24700 true false false
XX.XXX.XXX 168:24800 true false false
2 XX.XXX.XXX 168:24700 true false false
XX.XXX.XXX 168:24800 true false false
XX.XXX.XXX 168:24900 true false false
3 XX.XXX.XXX 168:24800 true false false
XX.XXX.XXX 168:24900 true false false
XX.XXX.XXX 168:25000 true false false
4 XX.XXX.XXX 168:24900 true false false
XX.XXX.XXX 168:25000 true false false
XX.XXX.XXX 168:24600 true false false
5 XX.XXX.XXX 168:25000 true false false
XX.XXX.XXX 168:24600 true false false
XX.XXX.XXX 168:24700 true false false
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 5 100 1 7 0 0 0 1
3 XX.XXX.XXX 168:24800 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
4 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 168:24600 true true true 6 0 0 8 0 0 0 0
5 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
XX.XXX.XXX 168:24700 true true true
XX.XXX.XXX 168:24800 true true true
2 XX.XXX.XXX 168:24700 true true true
XX.XXX.XXX 168:24800 true true true
XX.XXX.XXX 168:24900 true true true
3 XX.XXX.XXX 168:24800 true true true
XX.XXX.XXX 168:24900 true true true
XX.XXX.XXX 168:25000 true true true
4 XX.XXX.XXX 168:24900 true true true
XX.XXX.XXX 168:25000 true true true
XX.XXX.XXX 168:24600 true true true
5 XX.XXX.XXX 168:25000 true true true
XX.XXX.XXX 168:24600 true true true
XX.XXX.XXX 168:24700 true true true

KDEA08002-1 The command will now end.
$

```

- `eztool deletecache --with_deleteecf` コマンド, または `eztool deletecache` 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 deletecache` 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      Port  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 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。

### 11.6.1 データのバックアップを取得する手順（メモリ上のキャッシュだけを使用する場合）

メモリ上のキャッシュだけを使用する場合に、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      Port  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確認する

`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  Port  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。

## 11.6.2 データのバックアップを取得する手順（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。

ディスク上のキャッシュを使用する場合に、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。

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は、eztool exportecf コマンドを使用してバックアップを取得します。eztool exportecf コマンドでバックアップを取得した場合は、eztool importecf コマンドを使用してデータを再配置します。

## 参考

`eztool exportecf` コマンドを実行する代わりに、OS のコマンドなどを使用して EADS サーバのキャッシュファイルのバックアップを取得することもできます。

OS のコマンドなどを使用してバックアップを取得する場合は、必ず全 EADS サーバを終了してからバックアップを取得してください。なお、ディスクキャッシュの格納先として不揮発性メモリを使用している場合は、この手順ではバックアップは取得できません。

OS のコマンドなどを使用して EADS サーバのキャッシュファイルのバックアップを取得する

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任意のディレクトリにコピーします。

- <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>)

## (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      Port  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) キャッシュデータファイルとキャッシュインデックスファイルを書き出す

`eztool exportecf` コマンドを実行して、EADS サーバからすべてのキャッシュの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任意のディレクトリに書き出します。

```
eztool exportecf <キャッシュファイルの出力先パス名> <キャッシュ名>
```

#### コマンド実行例

```
$ eztool exportecf bkup/201912 cache
KDEA08001-I          The command will now start. (subcommand = exportecf, parameter = [exportecf, bkup/201912, cache])
KDEA08002-I          The command will now end.
$
```

### (4) 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確認する

キャッシュファイルの出力先を確認して、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が取得できているかを確認します。

### (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  Port  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。

## 11.7 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。

### 11.7.1 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.
$
```



## 11.8 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。

## 11.9 グループ名の一覧を確認したい

`eztool listgroup` コマンドを実行して、指定したキャッシュの、最上位階層のグループの一覧を表示します。

```
eztool listgroup [-v|--verbose] <キャッシュ名>
```

### コマンド実行例

```
$ eztool listgroup cache1
KDEA08001-1      The command will now start. (subcommand = listgroup, parameter = [listgroup, cache1])

GroupCount: 20

Server          Position      ID  GroupName
XX.XXX.XXX.168:24600 1288490189  1  group02
                  1288490189  1  group04
XX.XXX.XXX.168:24700 429496730   2  group01
                  429496730   2  group09
                  429496730   2  group11
                  429496730   2  group17
                  429496730   2  group18
                  429496730   2  group19
XX.XXX.XXX.168:24800 -429496729  3  group05
                  -429496729  3  group10
                  -429496729  3  group12
                  -429496729  3  group13
                  -429496729  3  group15
XX.XXX.XXX.168:24900 -1288490188 4  group06
                  -1288490188 4  group07
                  -1288490188 4  group08
                  -1288490188 4  group14
XX.XXX.XXX.168:25000 -2147483648 5  group03
                  -2147483648 5  group16
                  -2147483648 5  group20
-----
KDEA08002-1      The command will now end.
$
```

GroupName 欄には、EADS サーバごとに、最上位階層のグループ階層名が表示されます。

また、`-v` または `--verbose` オプションを指定すると、最上位階層のグループの位置や、グループに属している key の数を確認できます。

```

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

GroupCount: 20

Server          Position      ID  GroupName  HashValue  KeyCount
XX.XXX.XXX.168:24600 1288490189  1  group02    2037266030 100
                1288490189  1  group04    1388111144 100
XX.XXX.XXX.168:24700 429496730   2  group09    1269679203 100
                429496730   2  group11    1229725181 100
                429496730   2  group17    1172163628 100
                429496730   2  group19    1006380133 100
                429496730   2  group18    703078836  100
                429496730   2  group01    648506385  100
XX.XXX.XXX.168:24800 -429496729  3  group10    260938713  100
                -429496729  3  group15    232103928  100
                -429496729  3  group12    -56466136  100
                -429496729  3  group05    -73411842  100
                -429496729  3  group13    -163284496 100
XX.XXX.XXX.168:24900 -1288490188 4  group14    -503630070 100
                -1288490188 4  group07    -699511542 100
                -1288490188 4  group08    -992621419 100
                -1288490188 4  group06    -1203276328 100
XX.XXX.XXX.168:25000 -2147483648 5  group16    -1445454454 100
                -2147483648 5  group20    -1917239959 100
                -2147483648 5  group03    -1965919439 100
-----
KDEA08002-I The command will now end.
$

```

## 11.10 key の一覧を確認したい

eztool listkey コマンドを実行して、指定したキャッシュ内の key の一覧を表示します。

```
eztool listkey [-g <グループ名>|--group <グループ名>] <キャッシュ名>
```

### コマンド実行例

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

KeyCount: 25

Server          Position      ID  Key
XX.XXX.XXX.168:24600 1288490189 1  [1]groupA:element01
                  1288490189 1  [1]groupA:element02
                  1288490189 1  [1]groupA:element03
                  1288490189 1  [1]groupA:element04
                  1288490189 1  [1]groupA:element05
XX.XXX.XXX.168:24700 429496730 2  [2]groupB:element01
                  429496730 2  [2]groupB:element02
                  429496730 2  [2]groupB:element03
                  429496730 2  [2]groupB:element04
                  429496730 2  [2]groupB:element05
XX.XXX.XXX.168:24800 -429496729 3  [3]groupC:element01
                  -429496729 3  [3]groupC:element02
                  -429496729 3  [3]groupC:element03
                  -429496729 3  [3]groupC:element04
                  -429496729 3  [3]groupC:element05
XX.XXX.XXX.168:24900 -1288490188 4  [4]groupD:element01
                  -1288490188 4  [4]groupD:element02
                  -1288490188 4  [4]groupD:element03
                  -1288490188 4  [4]groupD:element04
                  -1288490188 4  [4]groupD:element05
XX.XXX.XXX.168:25000 -2147483648 5  [5]groupE:element01
                  -2147483648 5  [5]groupE:element02
                  -2147483648 5  [5]groupE:element03
                  -2147483648 5  [5]groupE:element04
                  -2147483648 5  [5]groupE:element05

-----

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

Key 欄には、EADS サーバごとに、key が表示されます。

また、-g または--group オプションを指定すると、指定したグループに属する key の一覧を確認できます。

```
$ eztool listkey -g [1]groupA cache1
KDEA08001-I The command will now start. (subcommand = listkey, parameter = [listkey, -g, [1]groupA, cache1])

KeyCount: 5

Server          Position      ID  Key
XX.XXX.XXX.168:24600 1288490189 1  [1]groupA:element01
                  1288490189 1  [1]groupA:element02
                  1288490189 1  [1]groupA:element03
                  1288490189 1  [1]groupA:element04
                  1288490189 1  [1]groupA:element05

-----

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

## 11.11 ユーザファンクションが各 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

`eztool listfunc` コマンドを実行して、ユーザファンクションが各 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を表示します。

```
eztool listfunc [-v|--verbose]
```

### コマンド実行例

```
$ eztool listfunc
KDEA08001-l The command will now start. (subcommand = listfunc, parameter = [listfunc])

FunctionCount: 3

FunctionName Enable Disable
FunctionA     3       0
FunctionB     2       1
FunctionC     0       1
-----
KDEA08002-l 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-l 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-l 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。

## 11.12 クラスタを停止させないで、EADS サーバの修正パッチを適用したい

クラスタを停止させないで、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

### 11.12.1 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 Port Position Cluster State Operation Lock KeyCount UsedMemoryRatio UsedMemory MaxMemory Version
1 XX.XXX.XXX.168 server01 24600 1288490189 online running none unlock 582 13 3 23 04-00-01
2 XX.XXX.XXX.168 server02 24700 429496730 online running none unlock 618 13 3 23 04-00-00
3 XX.XXX.XXX.168 server03 24800 -429496729 online running none unlock 636 13 3 23 04-00-00
4 XX.XXX.XXX.168 server04 24900 -1288490188 online running none unlock 591 13 3 23 04-00-00
5 XX.XXX.XXX.168 server05 25000 -2147483648 online running none unlock 573 13 3 23 04-00-00

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。

## 11.13 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)
- レンジごとの統計情報 (eads\_store\_stats.csv)
- ノンブロッキング I/O 通信制御の統計情報 (eads\_nonblocking\_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。

### 11.13.1 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。

### 11.13.2 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	



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	<a href="#">eads.server.connection.buffersize</a>
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.server.connection.buffersize</a></li> <li><a href="#">eads.replication.connection.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			復旧した EADS サーバ, スケールアウト処理で追加した EADS サーバ, またはリバランス処理で位置を変更した EADS サーバにデータを	TransferCopyDataWriteTotalSize	統計値	<ul style="list-style-type: none"> <li><a href="#">eads.transfer.datasize</a></li> <li><a href="#">eads.rebalance.transfer.datasize</a></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
			コピーするために、データの転送元の EADS サーバから送信されるデータの合計サイズ (単位: バイト)			
13			復旧した EADS サーバ、スケールアウト処理で追加した EADS サーバ、またはリバランス処理で位置を変更した EADS サーバに、データの転送元の EADS サーバから送信される全データ <sup>※3</sup> の合計サイズ (単位: バイト)	TransferWriteTotalSize	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	KeyCount	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) <sup>※4</sup>	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) <sup>※4</sup>	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) <sup>※4</sup>	HistoriesMemorySize	現在値	eads.replication.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) <sup>※4</sup>	MaxHistoriesMemorySize	設定値	
19		実行回数	リクエスト受付数 (put, create, update,	RequestCount	統計値	eads.node.< EADS サーバ ID >.position

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
			replace, get, remove) の合計値			
20			全ユーザファンクションの合計実行回数	AllFunctionExecuteCount	統計値	
21	EADS サーバの性能測定	性能	内部処理時間の平均値※1, ※5 (単位: マイクロ秒)	InternalProcessingAverageTime	統計値	
22			リクエスト処理時間の最大値※2, ※6 (単位: マイクロ秒)	RequestProcessingMaxTime	統計値	
23			リクエスト処理時間の平均値※2, ※6 (単位: マイクロ秒)	RequestProcessingAverageTime	統計値	
24			リクエスト処理時間の最小値※2, ※6 (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

復旧した EADS サーバ, スケールアウト処理で追加した EADS サーバ, またはリバランス処理で位置を変更した EADS サーバに, 次のデータが送信されます。

- コピー用のデータ
- 復旧処理, スケールアウト処理, またはリバランス処理の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

小数点以下の値は切り捨てて表示されます。

#### 注※5

データ自動削除機能を使用している場合、put、create の処理時に行われる remove の内部処理時間は put、create の内部処理時間に含まれます。

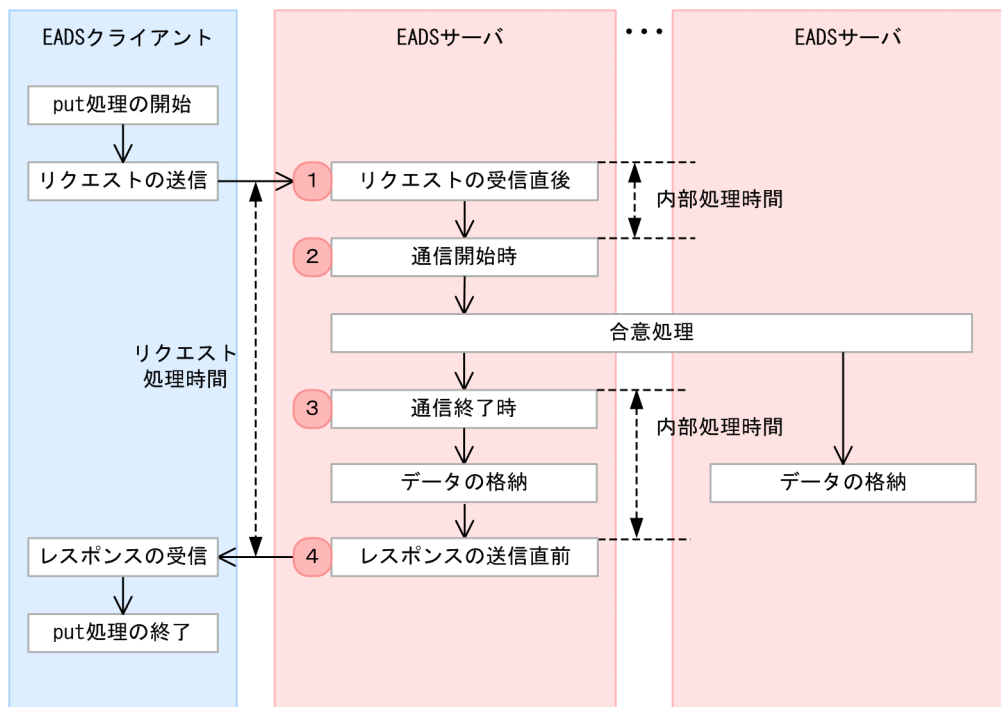
#### 注※6

データ自動削除機能を使用している場合、put、create の処理時に行われる remove の処理時間は put、create の処理時間に含まれます。

## (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個所 (put 処理の例)



### 説明

- 1 と 4 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。

- リクエスト処理時間から、合意処理に掛かった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)

### 11.13.3 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	
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	eads.cache.type
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	eads.replication.factor
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	CacheKeyCount	現在値	eads.node.< EADS サーバ ID >.position
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	

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
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	
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	<a href="#">eads.cache.disk.filesize</a>
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	<a href="#">eads.cache.disk.filenum</a>
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><a href="#">eads.cache.disk.filesize</a></li> <li><a href="#">eads.cache.disk.filenum</a></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><a href="#">eads.cache.disk.filesize</a></li> <li><a href="#">eads.cache.disk.filenum</a></li> <li><a href="#">eads.statistics.compaction.effect.division</a></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, ※18 (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, ※10 (単位:マイクロ秒)	CacheRequestProcessingMaxTime	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
17			キャッシュごとのリクエスト処理時間の平均値※5、※10 (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の最小値※5、※10 (単位: マイクロ秒)	CacheRequestProcessingMinTime	統計値	
19			キャッシュごとのputの合計回数	PutCount	統計値	
20			キャッシュごとのputの合計成功回数	PutSuccessCount	統計値	
21			キャッシュごとのgetの合計回数	GetCount	統計値	
22			キャッシュごとのgetの合計成功回数	GetSuccessCount	統計値	
23			キャッシュごとのremoveの合計回数※11	RemoveCount	統計値	
24			キャッシュごとのremoveの合計成功回数※12	RemoveSuccessCount	統計値	
25			キャッシュごとのcreateの合計回数	CreateCount	統計値	
26			キャッシュごとのcreateの合計成功回数	CreateSuccessCount	統計値	
27			キャッシュごとのupdateの合計回数	UpdateCount	統計値	
28			キャッシュごとのupdateの合計成功回数	UpdateSuccessCount	統計値	
29			キャッシュごとのreplaceの合計回数	ReplaceCount	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
30			キャッシュごとのreplaceの合計成功回数	ReplaceSuccessCount	統計値	
31			キャッシュごとのputAllの内部処理時間 <sup>※6, ※19</sup> の平均値(単位:マイクロ秒)	PutAllInternalProcessingAverageTime	統計値	
32			キャッシュごとのputAllの処理全体に掛かった時間 <sup>※7, ※13</sup> の最大値(単位:マイクロ秒)	PutAllProcessingMaxTime	統計値	
33			キャッシュごとのputAllの処理全体に掛かった時間 <sup>※7, ※13</sup> の平均値(単位:マイクロ秒)	PutAllProcessingAverageTime	統計値	
34			キャッシュごとのputAllの処理全体に掛かった時間 <sup>※7, ※13</sup> の最小値(単位:マイクロ秒)	PutAllProcessingMinTime	統計値	
35			キャッシュごとのputAllの合計回数 <sup>※8</sup>	PutAllCount	統計値	
36			キャッシュごとのputAllの合計成功回数 <sup>※8</sup>	PutAllSuccessCount	統計値	
37			キャッシュごとのputAllで一括操作したデータ数 <sup>※9</sup> の最大値	PutAllBatchDataMaxCount	統計値	
38			キャッシュごとのputAllで一括操作したデータ数 <sup>※9</sup> の平均値	PutAllBatchDataAverageCount	統計値	
39			キャッシュごとのputAllで一括操作	PutAllBatchDataMinCount	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
			したデータ数 <sup>※9</sup> の 最小値			
40			キャッシュごとの getAll の内部処理 時間 <sup>※6</sup> の平均値 (単位: マイクロ 秒)	GetAllInternalPr ocessingAverage Time	統計値	
41			キャッシュごとの getAll の処理全体 に掛かった時間 <sup>※7</sup> の最大値 (単位: マイクロ秒)	GetAllProcessin gMaxTime	統計値	
42			キャッシュごとの getAll の処理全体 に掛かった時間 <sup>※7</sup> の平均値 (単位: マイクロ秒)	GetAllProcessin gAverageTime	統計値	
43			キャッシュごとの getAll の処理全体 に掛かった時間 <sup>※7</sup> の最小値 (単位: マイクロ秒)	GetAllProcessin gMinTime	統計値	
44			キャッシュごとの getAll の合計回数 <sup>※8</sup>	GetAllCount	統計値	
45			キャッシュごとの getAll の合計成功 回数 <sup>※8</sup>	GetAllSuccessC ount	統計値	
46			キャッシュごとの getAll で一括操作 したデータ数 <sup>※9</sup> の 最大値	GetAllBatchDat aMaxCount	統計値	
47			キャッシュごとの getAll で一括操作 したデータ数 <sup>※9</sup> の 平均値	GetAllBatchDat aAverageCount	統計値	
48			キャッシュごとの getAll で一括操作 したデータ数 <sup>※9</sup> の 最小値	GetAllBatchDat aMinCount	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
49			キャッシュごとの removeAll の内部処理時間※6, ※14 の平均値 (単位: マイクロ秒)	RemoveAllInternalProcessingAverageTime	統計値	
50			キャッシュごとの removeAll の処理全体に掛かった時間※7, ※14 の最大値 (単位: マイクロ秒)	RemoveAllProcessingMaxTime	統計値	
51			キャッシュごとの removeAll の処理全体に掛かった時間※7, ※14 の平均値 (単位: マイクロ秒)	RemoveAllProcessingAverageTime	統計値	
52			キャッシュごとの removeAll の処理全体に掛かった時間※7, ※14 の最小値 (単位: マイクロ秒)	RemoveAllProcessingMinTime	統計値	
53			キャッシュごとの removeAll の合計回数※8, ※15	RemoveAllCount	統計値	
54			キャッシュごとの removeAll の成功回数の合計値※8, ※16	RemoveAllSuccessCount	統計値	
55			キャッシュごとの removeAll で一括操作したデータ数※9, ※17 の最大値	RemoveAllBatchDataMaxCount	統計値	
56			キャッシュごとの removeAll で一括操作したデータ数※9, ※17 の平均値	RemoveAllBatchDataAverageCount	統計値	
57			キャッシュごとの removeAll で一括操作したデータ数※9, ※17 の最小値	RemoveAllBatchDataMinCount	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

小数点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。

### 注※6

処理全体に掛かった時間から、EADS サーバ間の通信時間を除いた時間が対象です。

### 注※7

一括操作が一部でも失敗した場合の処理時間は含まれません。

### 注※8

EADS サーバの実行回数が表示されるため、EADS クライアントの実行回数とは一致しません。

### 注※9

一括操作が一部でも失敗した場合のデータ数は含まれません。

### 注※10

データ自動削除機能を使用している場合、put, create の処理時に行われる remove の処理時間は put, create の処理時間に含まれます。

### 注※11

データ自動削除機能を使用している場合、put, create の処理時に行われる remove の実行回数は対象になりません。

### 注※12

データ自動削除機能を使用している場合、put, create の処理時に行われる remove の成功回数は対象になりません。

#### 注※13

データ自動削除機能を使用している場合、putAll の処理時に行われる removeAll の処理時間は putAll の処理時間に含まれます。

#### 注※14

データ自動削除機能を使用している場合、putAll の処理時に行われる removeAll の処理時間は含まれません。

#### 注※15

データ自動削除機能を使用している場合、putAll の処理時に行われる removeAll の実行回数は対象になりません。

#### 注※16

データ自動削除機能を使用している場合、putAll の処理時に行われる removeAll の成功回数は対象になりません。

#### 注※17

データ自動削除機能を使用している場合、putAll の処理時に行われる removeAll によるデータ操作回数は対象になりません。

#### 注※18

データ自動削除機能を使用している場合、put、create の処理時に行われる remove の内部処理時間は put、create の内部処理時間に含まれます。

#### 注※19

データ自動削除機能を使用している場合、putAll の処理時に行われる removeAll の内部処理時間は putAll の内部処理時間に含まれます。

## 11.13.4 ユーザファクションの統計情報 (eads\_function\_stats.csv)

ユーザファ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、ユーザファクションの性能測定に使用できます。

表 11-3 ユーザファ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

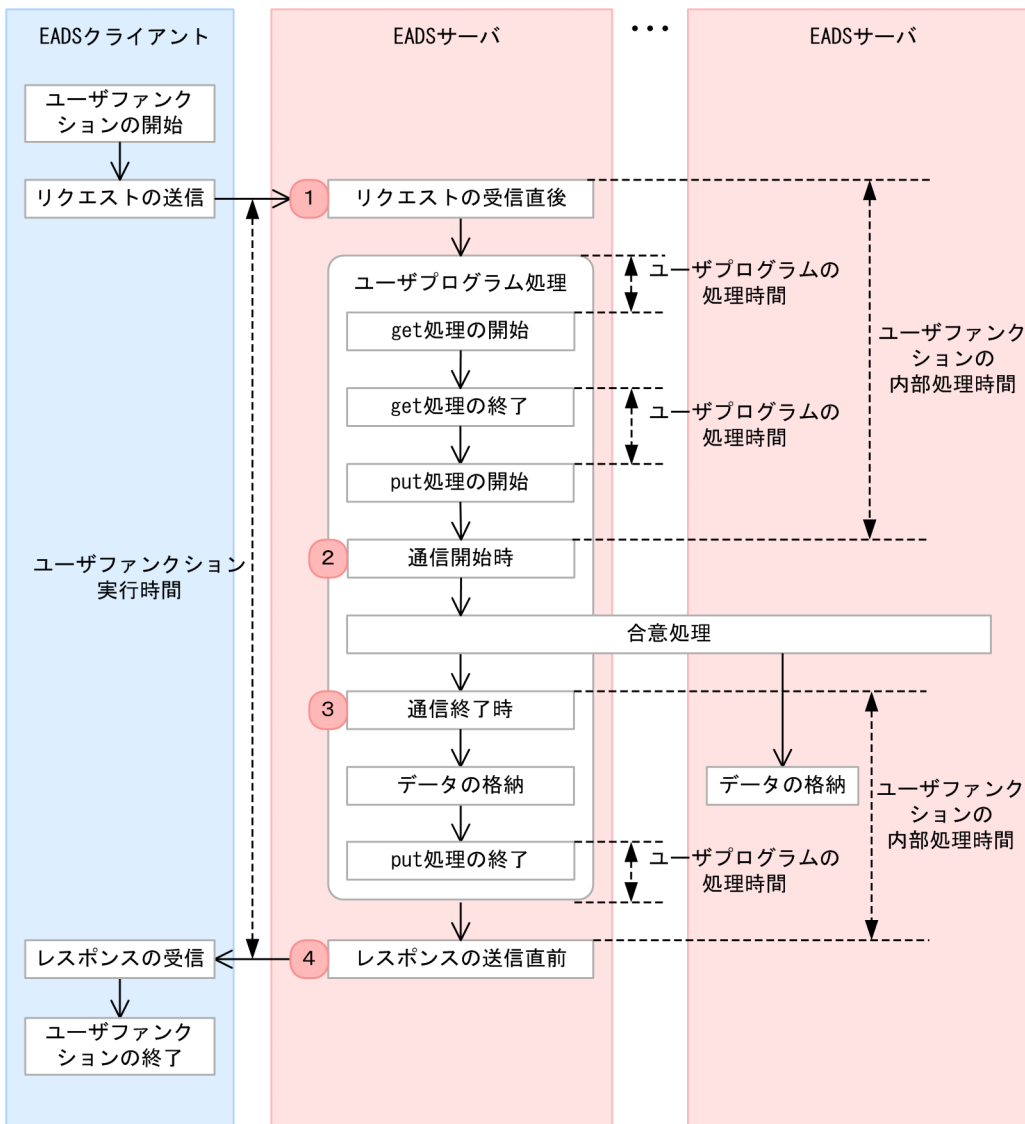
項番	項目	列名称	出力される値
2	統計情報を出力した時間	Time	現在値
3	統計情報を出力した EADS サーバの IP アドレスと、EADS クライアントとの通信に使用する EADS サーバのポート番号	ThisNode	設定値
4	ユーザファンクション名	FunctionName	設定値
5	ユーザファンクションの合計実行回数	FunctionExecuteCount	統計値
6	ユーザファンクション実行時間の最大値 (単位: マイクロ秒)	FunctionExecuteMaxTime	統計値
7	ユーザファンクション実行時間の平均値 (単位: マイクロ秒)	FunctionExecuteAverageTime	統計値
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 の合計回数	PutCount	統計値
16	ユーザファンクションから呼ばれた put の合計成功回数	PutSuccessCount	統計値
17	ユーザファンクションから呼ばれた get の合計回数	GetCount	統計値
18	ユーザファンクションから呼ばれた get の合計成功回数	GetSuccessCount	統計値
19	ユーザファンクションから呼ばれた remove の合計回数	RemoveCount	統計値

項番	項目	列名称	出力される値
20	ユーザファンクションから呼ばれた remove の合計成功回数	RemoveSuccessCount	統計値
21	ユーザファンクションから呼ばれた create の合計回数	CreateCount	統計値
22	ユーザファンクションから呼ばれた create の合計成功回数	CreateSuccessCount	統計値
23	ユーザファンクションから呼ばれた update の合計回数	UpdateCount	統計値
24	ユーザファンクションから呼ばれた update の合計成功回数	UpdateSuccessCount	統計値
25	ユーザファンクションから呼ばれた replace の合計回数	ReplaceCount	統計値
26	ユーザファンクションから呼ばれた replace の合計成功回数	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箇所（ユーザファクションの統計情報）



説明

- ユーザファクション実行時間とは、ユーザファ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4のタイミングで取得した時刻の差分から、ユーザファ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。
  - ユーザファクション実行時間の最大値 (FunctionExecuteMaxTime)
  - ユーザファクション実行時間の平均値 (FunctionExecuteAverageTime)
  - ユーザファ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。
- ユーザファクション実行時間から、合意処理に掛かった時間を除いた時間が、ユーザファ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。
  - ユーザファクションの内部処理時間の最大値 (FunctionInternalProcessingMaxTime)
  - ユーザファクションの内部処理時間の平均値 (FunctionInternalProcessingAverageTime)

- ユーザファ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、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)
  - ユーザファクションのユーザプログラムの処理時間の最小値 (UserProgramProcessingMinTime)

### 11.13.5 レンジごとの統計情報 (eads\_store\_stats.csv)

レンジごとの統計情報 (eads\_store\_stats.csv) に出力される情報について説明します。

なお、レンジごとの統計情報 (eads\_store\_stats.csv) は、データ総量監視機能が有効になっている場合に出力されます。

#### (1) レンジごとの統計情報 (eads\_store\_stats.csv) に出力される情報

レンジごとの統計情報 (eads\_store\_stats.csv) に出力される情報を、次の表に示します。

表 11-4 レンジごとの統計情報 (eads\_store\_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	
4	データ総量監視	レンジの情報	データのレンジ ID*1	RangeID	設定値	eads.replication.factor
5			キャッシュ名*2	CacheName	設定値	-
6			レンジに格納できる key 数の上限値*3	KeyCountLimit	設定値	eads.cache.keyCount
7			レンジに格納されている key 数	StoredKeyMaxCount	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
8			レンジにまだ格納されていない key 数	ReservedKeyMaxCount	統計値	<ul style="list-style-type: none"> <li>eads.server.cache.maxExecuteThreads</li> <li>eads.server.function.maxExecuteThreads</li> </ul>
9		メモリの使用量	レンジに格納できる value が使用するメモリ容量の上限値 (単位:メガバイト) ※3	ExternalHeapSizeLimit	設定値	eads.java.external.heapsize
10	レンジに格納されている value が使用しているメモリ容量 (単位:メガバイト) ※4		UsedExternalHeapMaxSize	統計値	—	
11	まだレンジに格納されていない value が使用するメモリ容量 (単位:バイト)		ReservedExternalHeapMaxSize	統計値	<ul style="list-style-type: none"> <li>eads.server.cache.maxExecuteThreads</li> <li>eads.server.function.maxExecuteThreads</li> </ul>	
12		ディスクの使用量	レンジに格納できるレコードが使用するディスク容量の上限値 (単位:メガバイト)	DiskSizeLimit	設定値	<ul style="list-style-type: none"> <li>eads.cache.disk.filesize</li> <li>eads.cache.disk.fileenum</li> </ul>
13	レンジに格納されているレコードが使用しているディスク容量 (単位:メガバイト) ※4		UsedDiskMaxSize	統計値	—	
14	まだレンジに格納されていないレコードが使用するディスク容量 (単位:バイト)		ReservedDiskMaxSize	統計値	<ul style="list-style-type: none"> <li>eads.server.cache.maxExecuteThreads</li> <li>eads.server.function.maxExecuteThreads</li> </ul>	

#### (凡例)

— : 関連するパラメタはありません。

#### 注※1

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※2

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※3

リバランス処理の実行中は、実際の値よりも少ない値が出力されることがあります。

### 注※4

小数点以下の値は切り捨てて表示されます。

## 11.13.6 ノンブロッキング I/O 通信制御の統計情報 (eads\_nonblocking\_stats.csv)

ノンブロッキング I/O 通信制御の統計情報について説明します。

### (1) ノンブロッキング I/O 通信制御の統計情報 (eads\_nonblocking\_stats.csv) に出力される情報

ノンブロッキング I/O 通信制御の統計情報 (eads\_nonblocking\_stats.csv) に出力される情報を、次の表に示します。

表 11-5 ノンブロッキング I/O 通信制御の統計情報 (eads\_nonblocking\_stats.csv) に出力される情報

項番	項目	列名称	出力される値	関連パラメタ
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	<ul style="list-style-type: none"><li>eads.server.address</li><li>eads.server.port</li></ul>
4	接続されたコネクション数	Connections	統計値	eads.server.maxConnections
5	イベントリスナで使用しているコネクション数	EventListenerExecutingConnections	統計値	
6	リクエスト処理スレッドで実行しているリクエスト数	ProcessingRequests	統計値	eads.server.nonBlocking.processorThreads
7	リクエスト処理スレッドでの実行を待っているリクエスト数	WaitingRequests	統計値	
8	現在使用中のノンブロッキング I/O 通信バッファサイズ (単位: メガバイト)	StayingIODataSize(MB)	統計値	<ul style="list-style-type: none"><li>eads.server.nonBlocking.external.heapsize</li><li>eads.server.connection.bufferSize</li></ul>
9	滞留中 I/O データ同時割当数	StayingIODataAllocatingConnections	統計値	eads.server.nonBlocking.maxAllocateConnections

項番	項目	列名称	出力される値	関連パラメタ
10	滞留中 I/O データ流量制御による退避中リクエスト数	EvacuatingRequestsByStayingDataFlow	統計値	
11	キャッシュ操作同時実行数	CacheExecutingThreads	統計値	eads.server.cache.maxExecuteThreads
12	キャッシュ操作同時実行制御退避中リクエスト数	EvacuatingRequestsByCacheConcurrency	統計値	
13	ファンクション操作同時実行数	FunctionExecutingThreads	統計値	eads.server.function.maxExecuteThreads
14	ファンクション同時実行数制御およびユーザファンクション単位ファンクション同時実行数制御による退避中リクエスト数	EvacuatingRequestsByFunctionConcurrency	統計値	<ul style="list-style-type: none"> <li>eads.server.function.maxExecuteThreads</li> <li>eads.function.&lt;ユーザファンクション名&gt;.maxExecuteThreads</li> </ul>

## 11.14 データの格納先の空き容量を管理したい

`eads.cache.limiter.enable` パラメタに `true` を指定すると、EADS サーバに格納できるデータ量（格納済みの key 数、Explicit ヒープの空き容量、キャッシュファイルの空き容量）が監視されます。格納先の空き容量が不足することが予想される場合は、処理をエラーとすることで、EADS サーバがダウンすることを防ぐことができます。これを、データ総量監視機能といいます。

データ総量監視機能を有効にすると、key の数、Explicit ヒープの容量、およびキャッシュファイルの容量について、次に示す上限値を超えないように監視されます。

- key の数の上限値  
共通設定の `eads.cache.keyCount` パラメタの指定値
- Explicit ヒープの容量の上限値（単位：メガバイト）

Explicit ヒープの容量の上限値（単位：メガバイト） =  
(共通設定の `eads.java.external.heapsize` パラメタの指定値×0.97  
−共通設定の `eads.replication.external.heapsize` パラメタの指定値  
−共通設定の `eads.server.nonBlocking.external.heapsize` パラメタの指定値\*)  
÷共通設定の `eads.replication.factor` パラメタの指定値

注

メガバイト未満の値は切り捨てられます。

注※

サーバ定義の `eads.server.nonBlocking.enable` パラメタの指定値が `false` の場合は 0 を代入してください。

- キャッシュファイルの容量の上限値（単位：メガバイト）

キャッシュファイルの容量の上限値（単位：メガバイト） =  
(キャッシュ定義の `eads.cache.disk.filenum` パラメタの指定値−2)  
× (キャッシュ定義の `eads.cache.disk.filesize` パラメタの指定値×1,024  
−キャッシュ定義の `eads.cache.disk.blocksize` パラメタの指定値×2) ÷1,024

注

メガバイト未満の値は切り捨てられます。

### ポイント

データ総量監視機能を有効にした場合、データを格納することで上限値を超える（格納先の空き容量が不足する）ことが予想されるときに、処理がエラーになります。このため、格納先の空き容量が少ないときに、データを格納する処理を同時に多数実行しようとする、処理がエラーになることがあります。

したがって、データ総量監視機能を使用する場合は、key の数、Explicit ヒープの容量、およびキャッシュファイルの容量について、実際に格納したい値より多めに見積もる必要があります。

データ総量監視機能を使用する場合の見積もりの方法については、「[4. 必要なリソースを確認する](#)」を参照してください。

なお、次に示す処理の間、データ総量監視機能は一時的に無効となります。

- スケールアウト処理
- データの読み込み (`eztool import` コマンド)
- 永続データの再配置 (`eztool importecf` コマンド)
- キャッシュの再開 (`eztool resume` コマンド)

### 11.14.1 データの格納先の空き容量を管理する手順（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、データの格納先の空き容量を管理する手順、および空き容量を増やす手順について説明します。

#### (1) 統計情報から空き容量を監視する

レンジごとの統計情報 (`eads_store_stats.csv`) に出力される情報から、データの格納先の空き容量を算出します。この値を監視します。

レンジごとの統計情報 (`eads_store_stats.csv`) は、データ総量監視機能を使用する場合にだけ出力されます。

レンジごとの統計情報 (`eads_store_stats.csv`) については、「[11.13.5 レンジごとの統計情報 \(eads\\_store\\_stats.csv\)](#)」を参照してください。

データの格納先の空き容量が少なくなった場合は、次の手順で対処してください。

#### (a) 不要なデータを削除する

不要なデータを削除することで、データの格納先の空き容量を増やします。

#### (b) クラスタに EADS サーバを追加する

(a)の手順を実行しても空き容量を確保できない場合は、クラスタに EADS サーバを追加します。クラスタに EADS サーバを追加することで、1EADS サーバに格納するデータの件数を減らすことができます。

EADS サーバをクラスタに追加する手順については、「[11.1.1 クラスタを停止させないで、EADS サーバをクラスタに追加する手順（スケールアウト処理・リバランス処理）](#)」または「[11.1.2 クラスタを停止](#)」

してから、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）」を参照してください。

ただし、スケールアウト処理中はデータ総量監視機能が無効となります。このため、十分な空き容量（スケールアウト処理中に増加する件数および容量の約 2 倍以上）が必要になります。

## (2) 空き容量不足で処理がエラーになった場合

空き容量不足で処理がエラーになった場合の対処について説明します。

### (a) EADS サーバに格納できるデータ量の上限に達した個所を確認する

`eztool storeusage` コマンドを実行して、key の数およびメモリ使用量から、EADS サーバに格納できるデータ量の上限に達した個所を確認してください。

### (b) 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。

### (c) 定義内容を変更する

Java ヒープサイズおよび Explicit ヒープサイズの再見積もりを実施し、見積もり結果に応じて定義内容を変更します。また、必要に応じて、サーバマシンおよびメモリを追加します。

見積もりについては、次の内容を参照してください。

- Java ヒープサイズの見積もり  
「[4.1.2 Java ヒープサイズを見積もる](#)」
- Explicit ヒープサイズの見積もり  
「[4.1.3 Explicit ヒープサイズを見積もる](#)」

定義内容を変更する手順については、「[11.4 定義内容を変更したい](#)」を参照してください。

### (d) 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。

## 11.14.2 データの格納先の空き容量を管理する手順（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。

### 参考

キャッシュファイルに対するデータ総量監視の有効範囲は、データのコピー元のEADSサーバだけとなります。そのため、データのコピー先のEADSサーバで空き容量不足となった場合は、データのコピー先のEADSサーバが縮退します。キャッシュファイルをコンパクションする際には、EADSサーバごとの空き容量に差が出ないように運用してください。

また、復旧処理中のEADSサーバでは、キャッシュファイルに対するデータ総量監視は無効となります。

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に、データの格納先の空き容量を監視する手順、および空き容量を増やす手順について説明します。

### (1) 統計情報から空き容量を監視する

レンジごとの統計情報 (eads\_store\_stats.csv) に出力される情報から、データの格納先の空き容量を算出します。この値を監視します。

レンジごとの統計情報 (eads\_store\_stats.csv) は、データ総量監視機能を使用する場合にだけ出力されます。

レンジごとの統計情報 (eads\_store\_stats.csv) については、「[11.13.5 レンジごとの統計情報 \(eads\\_store\\_stats.csv\)](#)」を参照してください。

データの格納先の空き容量が少なくなった場合は、次の手順で対処してください。

#### (a) 不要なデータを削除する

不要なデータを削除することで、データの格納先の空き容量を増やします。

#### (b) コンパクションを実行する

コンパクションを実行して、キャッシュデータファイルの使用量を削減します。

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は、「[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](#)」を参照してください。

#### (c) クラスタにEADSサーバを追加する

(a)(b)の手順を実行しても空き容量を確保できない場合は、クラスタにEADSサーバを追加します。クラスタにEADSサーバを追加することで、1EADSサーバに格納するデータの件数を減らすことができます。



EADS サーバをクラスタに追加する手順については、「11.1.3 クラスタを停止してから、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。

## (2) 空き容量不足で処理がエラーになった場合

空き容量不足で処理がエラーになった場合の対処について説明します。

### (a) EADS サーバに格納できるデータ量の上限に達した個所を確認する

`eztool storeusage --cache` コマンドを実行して、key の数、メモリ使用量、およびキャッシュファイルの使用量から、EADS サーバに格納できるデータ量の上限に達した個所を確認してください。

### (b) 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。

### (c) 定義内容を変更する

Java ヒープサイズ、Explicit ヒープサイズ、およびキャッシュファイルの容量の再見積もりを実施し、見積もり結果に応じて定義内容を変更します。また、必要に応じて、サーバマシン、メモリおよびディスクを追加します。

見積もりについては、次の内容を参照してください。

- Java ヒープサイズの見積もり  
「4.1.2 Java ヒープサイズを見積もる」
- Explicit ヒープサイズの見積もり  
「4.1.3 Explicit ヒープサイズを見積もる」
- キャッシュファイルの容量の見積もり  
「4.4 キャッシュファイルの容量を見積もる」

定義内容を変更する手順については、「11.4 定義内容を変更したい」を参照してください。

ディスクキャッシュの格納先として不揮発性メモリを追加する場合の手順については、リリースノートを参照してください。

### (d) 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、「11.1.3(11) クラスタ内の全 EADS サーバを開始する」～「11.1.3(15) 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。



## 11.15 優先度が低いデータを自動的に削除したい（データ自動削除機能の運用）

この節の内容は、データ自動削除機能を使用する場合にお読みください。

データ自動削除機能は、データ総量監視機能が有効な場合に使用できます。

また、メモリキャッシュまたはディスクキャッシュの場合に、データ自動削除機能を使用できます。2Wayキャッシュの場合には使用できません。

### 11.15.1 優先度が低いデータを自動的に削除する手順

この項の内容は、メモリキャッシュまたはディスクキャッシュを使用する場合にお読みください。

データ自動削除機能を使用して、優先度が低いデータを自動的に削除する手順について説明します。

#### ポイント

データ自動削除機能を使用する場合は、Java ヒープの容量、および Explicit ヒープの容量について見積もる必要があります。データ自動削除機能を使用する場合の見積もりの方法については、「[4. 必要なリソースを確認する](#)」を参照してください。

#### (1) データ総量監視機能を有効にする

データ自動削除機能の前提となるデータ総量監視機能を有効にしてください。データ総量監視機能については、「[11.14 データの格納先の空き容量を管理したい](#)」を参照してください。

#### (2) データの自動削除ポリシーを決める

`eads.cache.eviction.policy` パラメタで、データの自動削除ポリシーを決めます。データ自動削除機能を使用して、しきい値を超えたデータを削除する場合の動作を指定する必要があります。

`eads.cache.eviction.policy` パラメタに、次のどちらかを指定してください。

- FIFO  
データの更新順が古いデータから自動的に削除したい場合に指定します。
- LRU  
データの更新順または参照順が古いデータから自動的に削除したい場合に指定します。

なお、データの更新順およびデータの参照順は、次の操作が実行されるたびに更新されます（データの更新順およびデータの参照順が新しくなります）。

表 11-6 データの更新順およびデータの参照順が更新される操作一覧

データの自動削除ポリシー	データの更新順およびデータの参照順が更新される操作	
	API	コマンド
FIFO	データの更新操作 <ul style="list-style-type: none"> <li>• put</li> <li>• putAll</li> <li>• create</li> <li>• update</li> <li>• replace</li> </ul>	データの更新操作 <ul style="list-style-type: none"> <li>• <code>eztool put</code> コマンド</li> </ul>
LRU	データの更新操作 <ul style="list-style-type: none"> <li>• put</li> <li>• putAll</li> <li>• create</li> <li>• update</li> <li>• replace</li> </ul> データの参照操作 <ul style="list-style-type: none"> <li>• get</li> <li>• getAll</li> </ul>	データの更新操作 <ul style="list-style-type: none"> <li>• <code>eztool put</code> コマンド</li> </ul> データの参照操作 <ul style="list-style-type: none"> <li>• <code>eztool get</code> コマンド</li> </ul>

## 注意事項

データの更新順およびデータの参照順の情報は、EADS サーバが起動している間だけ保持されま  
す。EADS サーバを再起動した場合は初期化されます。

## 注意事項

データを多重化している場合に、`eads.cache.eviction.policy` パラメタに LRU を指定したときは、  
データが使用された順番の管理方法が EADS サーバによって異なります。

- データの格納先 EADS サーバは、データの更新操作およびデータの参照操作の順番で管理しま  
す。
- データのコピー先 EADS サーバは、データの更新操作の順番で管理します。

そのため、障害などが発生し、アクセス先が自動的にデータのコピー先 EADS サーバに切り替わっ  
た直後は、データの更新操作およびデータの参照操作の順ではなく、データの更新操作の順番が古  
いデータから削除されます。そのあとは、データの更新操作およびデータの参照操作の順番で管理  
します。

## 注意事項

スケールアウト処理またはリバランス処理で格納先の EADS サーバが変更されたデータは、変更前のデータの更新順およびデータの参照順の情報に関係なく、新たに更新されたデータとして管理します。

### (3) データを削除するためのしきい値を決める

`eads.cache.eviction.keyCount` パラメタで、データを削除するためのしきい値として使用する 1 レンジ当たりのデータ件数を決めます。

データ自動削除機能の前提となるデータ総量監視機能による 1EADS サーバに格納する 1 レンジ当たりのデータ件数 (`eads.cache.keyCount` パラメタ) は、次の条件を満たすような値を指定する必要があります。

なお、`eads.cache.keyCount` パラメタの指定値は、「4.1.2(1)(a) 1EADS サーバに格納するデータ件数」を参照して見積もってください。

#### ■条件

`eads.cache.keyCount` パラメタの指定値 >  
データ自動削除機能を使用するキャッシュの  
キャッシュ定義の `eads.cache.eviction.keyCount` パラメタの指定値の総和

### (4) データ自動削除機能で優先度が低いデータを削除する

データ総量監視機能を有効にして、`eads.cache.eviction.policy` パラメタおよび `eads.cache.eviction.keyCount` パラメタを指定したら、データ自動削除機能を使用できます。

データ自動削除機能は、API (`create`, `put`, `putAll`) および `eztool put` コマンドによって格納される新規データが `eads.cache.eviction.keyCount` パラメタで指定したしきい値を超えた場合に、`eads.cache.eviction.policy` パラメタで指定したデータの自動削除ポリシーに従って、しきい値を超えたデータが自動的に削除されます。

データの自動削除は、`create` および `put` の処理の場合は `remove` が使用されます。`putAll` の処理の場合は削除件数に関係なく、`removeAll` が使用されます。

## 注意事項

- API (`put`, `putAll`) および `eztool put` コマンドの場合、新規データを格納するかどうかに関係なく、しきい値を超えるデータは削除されます。そのため、既存のデータを更新する場合には、一時的にデータ件数がしきい値を下回る場合があります。
- データの削除に失敗した場合、データの更新操作は実行されません。エラーコードが返却されません。

- API (putAll) の場合、データの削除に失敗したキーはデータの更新が実行されません。データの削除に成功したキーはデータの更新が実行されます。

## (5) データ自動削除機能を使用する場合の注意事項

1. EADS サーバのデータが自動で削除されると問題になる場合は、データ自動削除機能を使用しないでください。
2. クラスタの停止時のデータの読み込みが不要な場合、EADS サーバを起動したときに、データを読み込まないでキャッシュを再作成する次の運用を推奨します。
  - `eztool close` コマンドで、クラスを閉塞状態にする。
  - `eztool stop` コマンドに `--no_export` オプションを指定してクラスタを終了する。
  - `ezstart` コマンド、または `ezserver` コマンドで EADS サーバを開始する。
  - ディスクキャッシュを使用している場合、すべてのディスクキャッシュに対して `eztool deletectf` コマンドでキャッシュファイルを削除する。
  - `eztool createcache` コマンドで、使用するすべてのキャッシュを再作成する。
  - `eztool open` コマンドで、クラスタの閉塞状態を解除する。
3. 次のコマンドを使用してデータを読み込んだ場合は、データの読み込み順で、データの更新操作およびデータの参照操作の順番が初期化されます。
  - メモリキャッシュおよびディスクキャッシュを使用している場合  
`ezstart -r` コマンド (復旧した EADS サーバだけが対象となります)  
`ezserver -r` コマンド (復旧した EADS サーバだけが対象となります)
  - メモリキャッシュを使用している場合  
`eztool import` コマンド
  - ディスクキャッシュを使用している場合  
`eztool resume` コマンド  
`eztool importectf` コマンド
4. メモリキャッシュでデータの読み込みを行う場合の注意点を次に示します。
  - `eztool import` コマンドでストアデータファイルからデータを読み込んだ場合に、しきい値として指定したレンジのデータ件数を超えるレンジからは、それ以上データの読み込みを行いません。しきい値として指定したデータ件数を超えるレンジのデータが格納されていた場合、コマンドのリターンコードに 10 が返却されます。また、すべてのデータを読み込む場合は、レンジに格納されたデータ件数以上の値をしきい値に指定する必要があります。
  - `eztool import` コマンドで、`eztool export` コマンドに `-s` または `--single` オプションを指定して出力したストアデータファイルは読み込まないで、`eztool createcache` コマンドでキャッシュを再作成し、必要に応じてデータを再投入することを推奨します。`eztool export` コマンドに `-s` または `--single` オプションを指定して出力したストアデータファイルを読み込む場合は、データ自動削除機能のし

きい値 (`eads.cache.eviction.keyCount` パラメタの指定値) をストアデータファイル出力時に設定してあったしきい値よりも小さい値に変更しないでください。

5. ディスクキャッシュでデータの読み込みを行う場合の注意点を次に示します。

- `eztool importecf` コマンドで永続データを再配置する場合、しきい値として指定したレンジのデータ件数を超えるレンジからは、それ以上データの読み込みを行いません。しきい値として指定したデータ件数を超えるレンジのデータが格納されていた場合、コマンドのリターンコードに 10 が返却されます。また、すべてのデータを読み込む場合は、レンジに格納されたデータ件数以上の値をしきい値に指定する必要があります。
- `eztool importecf` コマンドで、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読み込まないで `eztool createcache` コマンドでキャッシュを再作成し、必要に応じてデータを再投入することを推奨します。また、読み込み対象の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に読み込むときは、データ自動削除機能のしきい値 (`eads.cache.eviction.keyCount` パラメタの指定値) をキャッシュ使用時に設定してあったしきい値よりも小さい値に変更しないでください。

6. ディスクキャッシュを使用する場合、データ自動削除機能を使用するときでも、API で `key` を削除するときと同様に、コンパクションを実行するまでキャッシュデータファイルの空き容量は増加しません。そのため、空き容量が不足した場合は、API (`create`, `put`, `putAll`) がデータ総量監視機能のキャッシュファイルの空き容量不足でエラーとなります。

7. API (`create`, `put`, `putAll`) がデータ総量監視機能の登録済みキー数の上限でエラーとなった場合は、次のパラメタの指定値を見直してください。

- `eads.cache.keyCount` パラメタ
- `eads.cache.eviction.keyCount` パラメタ

8. API (`create`, `put`, `putAll`) がデータ総量監視機能による `value` の格納先である Explicit ヒープの空き容量不足でエラーとなった場合は、`value` のサイズが容量見積もり時より大きくなっていないかを見直してください。または、見積もり結果が過少でないかについても確認してください。

9. API (`create`, `put`, `putAll`) がデータ総量監視機能のキャッシュファイルの空き容量不足 (ディスクキャッシュ使用時) でエラーとなった場合は、レコード 1 件分のサイズが容量見積もり時より大きくなっていないかを見直してください。または、見積もり結果が過少でないか、コンパクションで空き容量を確保できているかについても確認してください。

10. スケールアウト処理およびリバランス処理を実行したときに別レンジへ移動したデータは、移動元レンジでの更新順および参照順は無視され、最新データとして再配置されます。

11. リバランス処理を実行している間、データ自動削除機能を使用しているキャッシュでは、指定したデータの自動削除ポリシーに加えて、次に示すデータの自動削除ポリシーにも従います。そのため、次に示すデータの自動削除ポリシーが適用された場合に問題となる用途では、リバランス処理を実行しないでください。

#### ■ リバランス処理を実行中に適用されるデータの自動削除ポリシー

結合先のレンジで格納できるデータの空き (キー数および容量) が少ない場合、結合元のレンジ (移動するレンジ) に格納されたデータは、キー数がしきい値以下であっても、結合先のレンジで格納

できるデータの空きに収まるまで一部または全部削除されます。また、通信タイミングによって、最も古いデータではなく、最も新しいデータが削除されることがあります。

12. リバランス処理を実行している間、結合元のレンジ（移動するレンジ）に格納されたデータに対する API (create, put, putAll) は、次に示す一時的なエラーを返却することがあります。

なお、API (create, put) の場合は、EADS クライアント (Java と C) およびユーザファンクションが該当します。API (putAll) の場合は、EADS クライアント (Java と C) が該当します。

■EADS サーバの位置変更開始時点での結合先のレンジに格納できるデータの空き（キー数）がない場合

- EADS クライアント (Java) とユーザファンクションのとき  
CacheException.EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT\_TEMPORARILY
- EADS クライアント (C) のとき  
EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT\_TEMPORARILY

■EADS サーバの位置変更開始時点での結合先のレンジに格納できるデータの空き（容量）が少ない場合

- EADS クライアント (Java) とユーザファンクションのとき  
CacheException.EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY\_TEMPORARILY
- EADS クライアント (C) のとき  
EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY\_TEMPORARILY



## 11.16 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。

### 11.16.1 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 true 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 true 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 deleteecf 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 deleteecf <キャッシュ名>
```

### コマンド実行例

```
$ eztool deleteecf cache1
KDEA08001-I      The command will now start. (subcommand = deleteecf, parameter = [deleteecf, 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.command.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_command_message[n].log</code>		
			Shift	<code>eads_command_message.log</code>		
EADS クライアント		<クライアント定義の <code>eads.client.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。

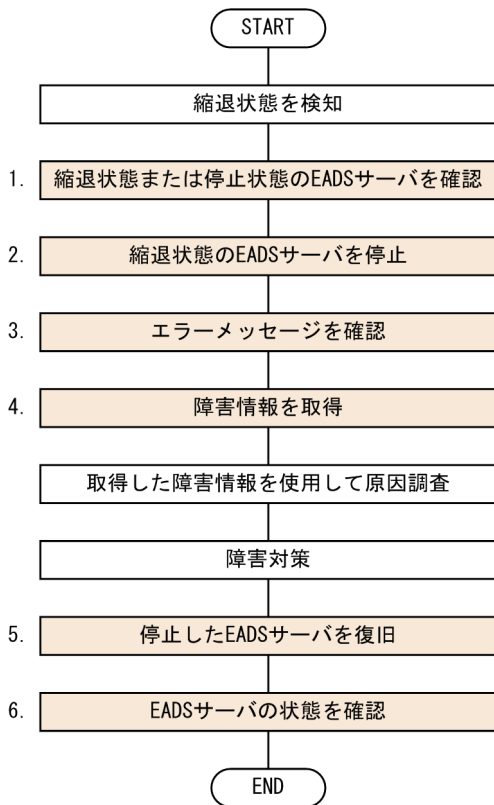
## 12.2 障害発生時のシステム運用管理者の作業

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

障害によって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 障害によって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      Port  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 2015/04/03 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、2015年04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 2015/04/21 11:55:46.116 EADS 6EF0EED6 45E67E6A KDEA04799-E Processing to isolate a  
server will now start. (server ID = 3)  
⋮
```

この例では、2015年04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、CPU のリソースやネットワークの帯域を使用するといった影響があります。
- データ操作と復旧処理が重なって EADS サーバ内の処理が追いつかなくなった場合は、メモリが枯渇しないように、データ操作を待たせることがあります。

### 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>eztool deleteecf -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。 復旧に失敗した場合は、上記の手順を実行し直してください。	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。

容量不足が解消されないときは、「11.4.1 定義内容を変更する手順」に従って、キャッシュ定義の `eads.cache.disk.fileenum` パラメタの指定値を大きくしてください。

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。
- 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      Port  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。

### 注意事項

EADS の合意処理では、クラスタ内で過半数の合意を得る必要があるため、クラスタ動作不能 (NOT\_AVAILABLE) またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) になると、一部の機能が停止します。一部の機能が停止することに伴って、クラスタ動作不能またはクラスタ一部稼働中になった直後に、次のような現象が起きることがあります。

- 予期しない例外のメッセージがログファイルに出力される。
- 実行していたコマンドが実行中のままになる。

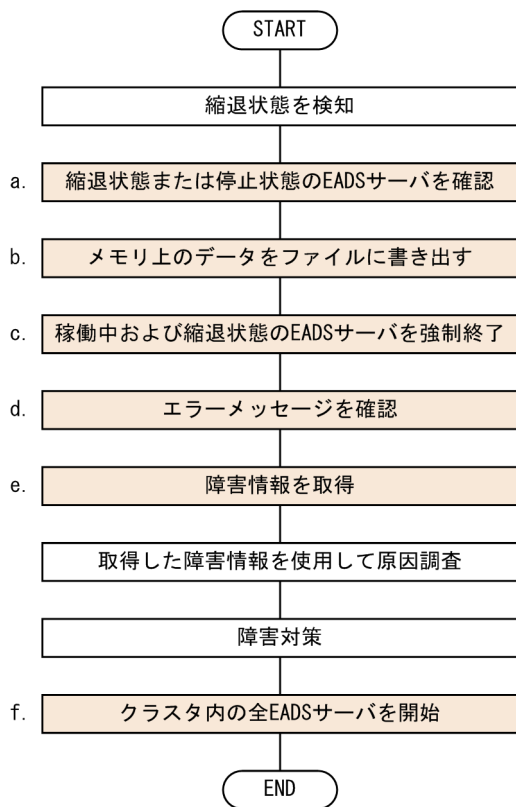
ただし、これらの現象が起きても、データが破壊されることはありません。

後述する手順で、データを復旧できます。

### (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）

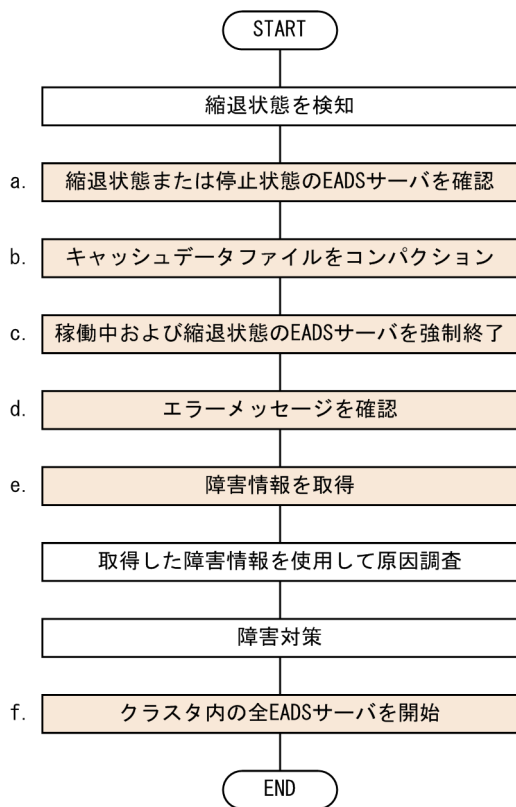
障害対策後、クラスタ内の全 EADS サーバを起動して、(a)でファイルに書き出したデータを再度メモリ上に読み込ませます。

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

`eztool 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 `eztool deletectf -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pons低下の報告があった場合、またはレスpons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.client.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]」※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

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、「[11.13 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。



# 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。

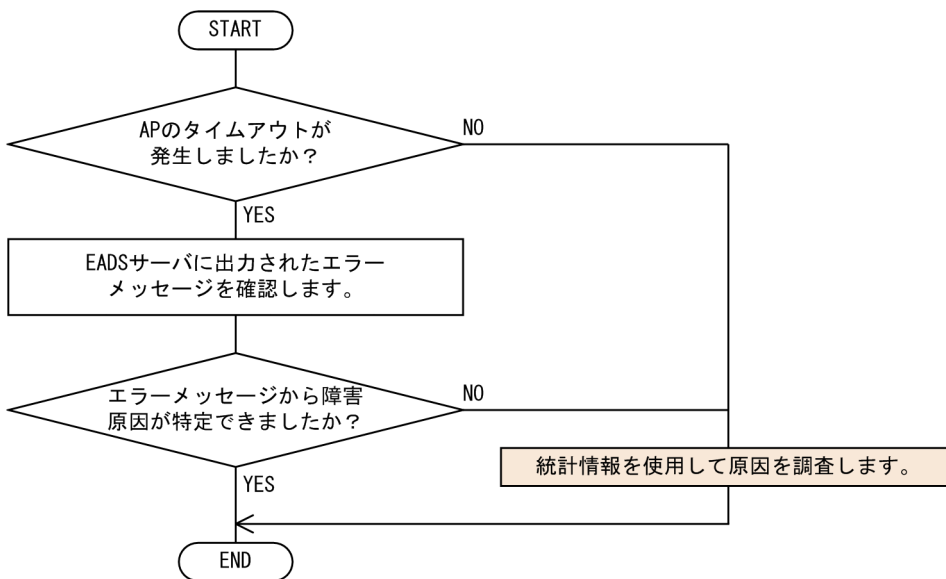
## 13.1 レスpons低下の原因調査

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、「11.13 統計情報を取得したい」を参照してください。

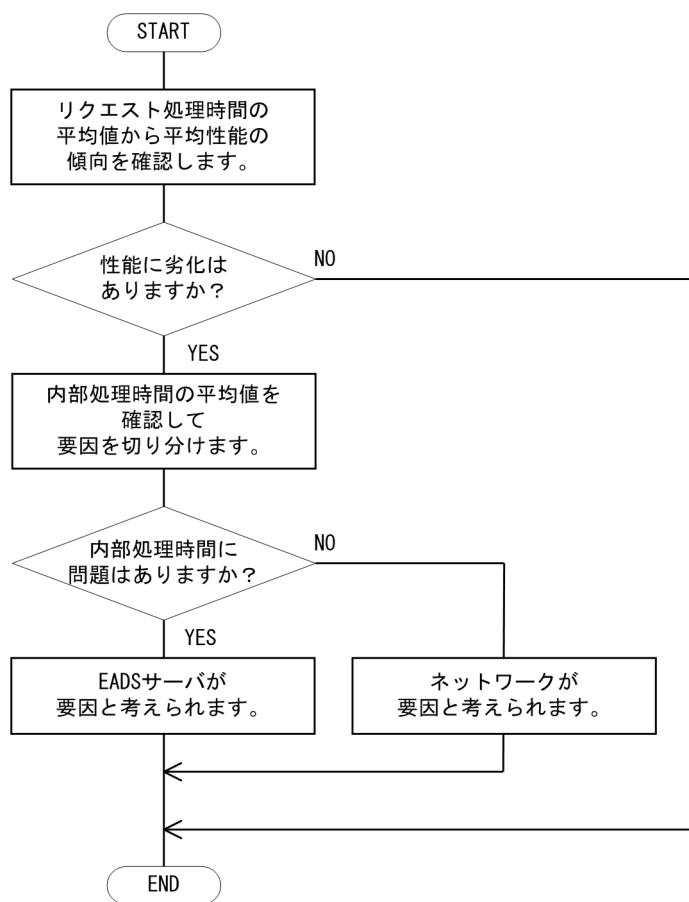
### 13.1.2 原因の調査方法

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確認して、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

## (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。
- ノンブロッキング I/O 通信制御を使用している場合、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。または、問題があった項目の関連パラメタが適切な指定値になっているか確認してください。

統計情報の種類	項目	列名称
ノンブロッキング I/O 通信制御の 統計情報 (eads_nonblocking_stats.csv)	リクエスト処理スレッドでの実行を待っているリクエスト数	WaitingRequests
	リクエスト処理スレッドで実行しているリクエスト数	ProcessingRequests
	滞留中 I/O データ流量制御による退避中リクエスト数	EvacuatingRequestsByStayingDataFlow
	キャッシュ操作同時実行制御退避中リクエスト数	EvacuatingRequestsByCacheConcurrency

- 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。

### ■ 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。

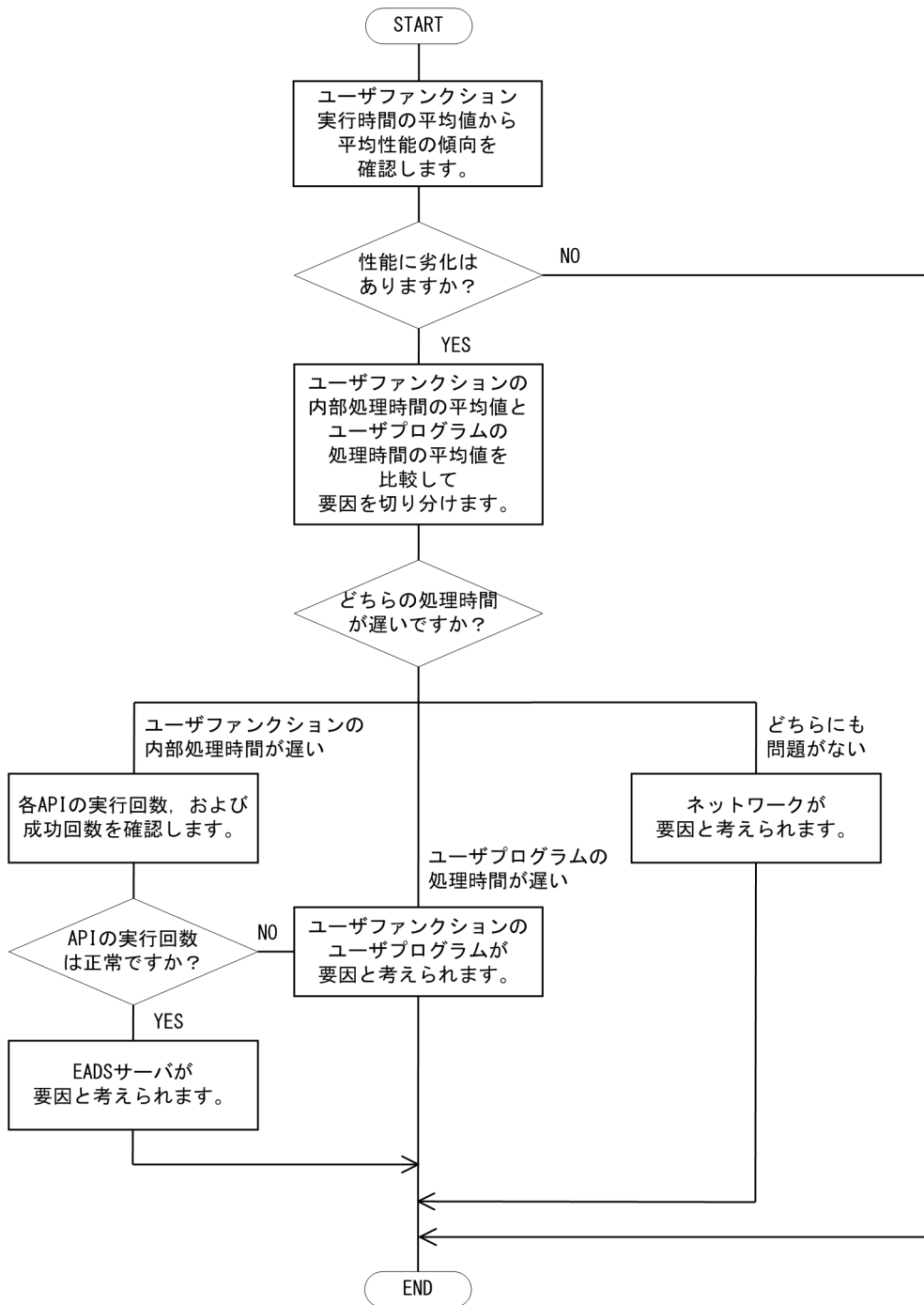
統計情報の種類	項目	列名称
統計情報 (eads_stats.csv)	EADS クライアント・EADS サーバ間の受信データの最大サイズ	CSReadMaxSize
	EADS クライアント・EADS サーバ間の受信データの合計サイズ	CSReadTotalSize
	EADS クライアント・EADS サーバ間の送信データの最大サイズ	CSWriteMaxSize
	EADS クライアント・EADS サーバ間の送信データの合計サイズ	CSWriteTotalSize
	EADS サーバ間の受信データの最大サイズ	SSReadMaxSize
	EADS サーバ間の受信データの合計サイズ	SSReadTotalSize
	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。
- ノンブロッキング I/O 通信制御を使用している場合、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。または、問題があった項目の関連パラメタが適切な指定値になっているか確認してください。

統計情報の種類	項目	列名称
ノンブロッキング I/O 通信制御の統計情報 (eads_nonblocking_stats.csv)	リクエスト処理スレッドでの実行を待っているリクエスト数	WaitingRequests
	リクエスト処理スレッドで実行しているリクエスト数	ProcessingRequests
	滞留中 I/O データ流量制御による退避中リクエスト数	EvacuatingRequestsByStayingDataFlow
	ファンクション同時実行数制御およびユーザ機能単位ファンクション同時実行数制御による退避中リクエスト数	EvacuatingRequestsByFunctionConcurency

- 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。

## ■ ユーザファクションのユーザプログラムが要因の場合

ユーザファクションのユーザプログラムが要因と考えられる場合は、各ユーザファクションの統計情報を確認し、要因となっているユーザファクション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
	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。



# 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	EADS サーバ
2		EADS サーバをフォアグラウンドで開始する	EADS サーバ
3	運用操作	運用操作を行う	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。

### 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 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 [-h]
         [-r|-ai <EADSサーバID>|-ap <EADSサーバの位置 (ハッシュ値) >]
         [-rd [<IPアドレスまたはホスト名>:] <ポート番号> <suspend方式>]
```

## (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。

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) `-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。

### (c) `-ai` または `--add_id < EADS サーバ ID >`

既存のクラスタ構成にない新しい EADS サーバを、クラスタの特定のレンジに追加したいときに指定します。このオプションには、新しい EADS サーバを追加したいレンジを担当する EADS サーバの EADS サーバ ID を指定します。

EADS サーバ ID には、1~96 の整数が指定できます。

新しい EADS サーバは、コンシステント・ハッシングの円周上で、新しい EADS サーバと隣接している 2 つの EADS サーバの位置 (ハッシュ値) の中間に追加されます。また、新しい EADS サーバの EADS サーバ ID には、1~96 の整数のうち、既存の EADS サーバに割り当てられていない最小の値が割り当てられます。

クラスタに存在しない EADS サーバの EADS サーバ ID を指定した場合は、エラーになります。また、このオプションを指定してコマンドを実行した際に、クラスタに縮退状態 (isolated) または停止状態 (-----) の EADS サーバが存在する場合は、エラーになります。

### (d) `-ap` または `--add_position < EADS サーバの位置 (ハッシュ値) >`

既存のクラスタ構成にない新しい EADS サーバを、クラスタの特定の位置 (ハッシュ値) に追加したいときに指定します。このオプションには、新しい EADS サーバを追加したい位置 (ハッシュ値) を指定します。

位置 (ハッシュ値) には、-2147483648~2147483647 の整数が指定できます。

追加する EADS サーバの EADS サーバ ID には、1~96 の整数のうち、既存の EADS サーバに割り当てられていない最小の値が割り当てられます。

既存の EADS サーバと同じ位置 (ハッシュ値) を指定した場合は、エラーになります。また、このオプションを指定してコマンドを実行した際に、クラスタに縮退状態 (isolated) または停止状態 (-----) の EADS サーバが存在する場合は、エラーになります。

## (e) `-rd` または `--remotedebug` [`<IP アドレスまたはホスト名>:`] `<ポート番号>` `<suspend 方式>`

ユーザファンクションをデバッガアプリケーションでリモートデバッグするためのオプションです。本番環境では使用しないでください。

次に示す java オプションを指定して、EADS サーバを起動したい場合に指定します。

```
-agentlib:jwp=transport=dt_socket,server=y,suspend=<suspend方式>,address=[<IPアドレスまたはホスト名>:]<ポート番号>
```

IP アドレスまたはホスト名には、EADS サーバを実行するマシンの IP アドレスまたはホスト名を指定してください。指定を省略した場合は、ローカルホスト以外のデバッガアプリケーションから接続できません。ポート番号 (EADS サーバがデバッガアプリケーションに接続するためのポート番号) には、1024~65535 の整数が指定できます。

デバッガアプリケーションの接続を待ってから EADS サーバを起動する場合は、suspend 方式に y を指定します。デバッガアプリケーションの接続を待たないで EADS サーバを起動する場合は、suspend 方式に n を指定します。

## (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。
- 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.admin.boot.timeout` パラメタの指定値を調整してください。
- `-ai` または `--add_id` オプション、`-ap` または `--add_position` オプションを指定する場合、追加する EADS サーバのクラスタ定義ファイルは不要です。追加する EADS サーバにクラスタ定義ファイルがあっても読み込まれません。このため、追加する EADS サーバに不正な内容のクラスタ定義ファイルがあっても、エラーになりません。
- `-ai` または `--add_id` オプション、`-ap` または `--add_position` オプションの指定時に、EADS サーバの追加でエラーが発生したときは、`eztool status -v` 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。コマンドの排他が解除されていない場合、`eztool unlock` コマンドを実行して排他を解除してください。

EADS サーバが追加されていない場合は、再度 EADS サーバを追加してください。

追加された EADS サーバが縮退またはプロセスダウンした場合は、EADS サーバを復旧させてください。

復旧に失敗した EADS サーバのクラスタ定義ファイルが、ほかの EADS サーバのクラスタ定義ファイルと同じかどうかを確認してください。クラスタ定義ファイルが、ほかの EADS サーバのクラスタ定義ファイルと異なる場合は、復旧に失敗した EADS サーバに、正常に稼働している EADS サーバのクラスタ定義ファイルをコピーしてから、復旧処理を再度実行してください。

6. `-r` または `--recovery` オプション指定時に EADS サーバの復旧処理でエラーが発生した場合、`eztool status -v` 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。コマンドの排他が解除されていない場合、`eztool unlock` コマンドを実行して排他を解除してください。
7. スケールアウト処理の終了直前のリクエストの応答は、一時的に時間が掛かることがあります。EADS サーバからの応答に対して EADS クライアントを長時間待機させるのではなく、EADS サーバからエラーを返却させ、EADS クライアントでそのエラーに対処させる方がよい場合は、次のパラメタの指定を検討してください。
  - `eads.scaling.requestQueuing.enable` パラメタ
8. スケールアウト処理の開始直後のリクエストの応答に一時的に時間が掛かる場合は、`eads.scaling.requestQueuing.enable` パラメタとあわせて次のパラメタの指定も検討してください。
  - `eads.scaling.moveDataWithUpdateRequest.enable` パラメタ
  - `eads.scaling.moveData.interval` パラメタ
9. 次のどちらかの指定をしている場合、リクエストのスループットの向上が期待できますが、スケールアウト処理の実行完了までの時間が長くなります。運用および環境に合わせたチューニングを実施してください。
  - `eads.scaling.moveDataWithUpdateRequest.enable` パラメタに `false` を指定している場合、かつ `ezstart` コマンドに `-ai` または `-ap` オプションを指定している場合
  - `eads.scaling.moveData.interval` パラメタに 0 以外を指定している場合、かつ `ezstart` コマンドに `-ai` または `-ap` オプションを指定している場合
10. `eads.scaling.requestQueuing.enable` パラメタに `false` を指定している場合、かつ `ezstart` コマンドに `-ai` または `-ap` オプションを指定している場合、追加する EADS サーバの移動元となる EADS サーバで実行しているクライアント API でエラーが発生することがあります。
11. `-rd` または `--remotedebug` オプションに指定するホスト名には、「`:`」を含むホスト名は指定できません。

## 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 [-h]
          [-r|-ai <EADSサーバID>|-ap <EADSサーバの位置 (ハッシュ値) >]
          [-rd [<IPアドレスまたはホスト名>:] <ポート番号> <suspend方式>]
```

## (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。

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) -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。

### (c) -ai または --add\_id < EADS サーバ ID >

既存のクラスタ構成にない新しい EADS サーバを、クラスタの特定のレンジに追加したいときに指定します。このオプションには、新しい EADS サーバを追加したいレンジを担当する EADS サーバの EADS サーバ ID を指定します。

EADS サーバ ID には、1~96 の整数が指定できます。

新しい EADS サーバは、コンシステント・ハッシングの円周上で、新しい EADS サーバと隣接している 2 つの EADS サーバの位置 (ハッシュ値) の中間に追加されます。また、新しい EADS サーバの EADS サーバ ID には、1~96 の整数のうち、既存の EADS サーバに割り当てられていない最小の値が割り当てられます。

クラスタに存在しない EADS サーバの EADS サーバ ID を指定した場合は、エラーになります。また、このオプションを指定してコマンドを実行した際に、クラスタに縮退状態 (isolated) または停止状態 (-----) の EADS サーバが存在する場合は、エラーになります。

## (d) `-ap` または `--add_position` < EADS サーバの位置 (ハッシュ値) >

既存のクラスタ構成にない新しい EADS サーバを、クラスタの特定の位置 (ハッシュ値) に追加したいときに指定します。このオプションには、新しい EADS サーバを追加したい位置 (ハッシュ値) を指定します。

位置 (ハッシュ値) には、-2147483648~2147483647 の整数が指定できます。

追加する EADS サーバの EADS サーバ ID には、1~96 の整数のうち、既存の EADS サーバに割り当てられていない最小の値が割り当てられます。

既存の EADS サーバと同じ位置 (ハッシュ値) を指定した場合は、エラーになります。また、このオプションを指定してコマンドを実行した際に、クラスタに縮退状態 (isolated) または停止状態 (-----) の EADS サーバが存在する場合は、エラーになります。

## (e) `-rd` または `--remotedebug` [`< IP アドレスまたはホスト名 >:`] `<ポート番号>` `< suspend 方式 >`

ユーザ機能をデバッガアプリケーションでリモートデバッグするためのオプションです。本番環境では使用しないでください。

次に示す java オプションを指定して、EADS サーバを起動したい場合に指定します。

```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=<suspend方式>,address=[<IPアドレスまたはホスト名>:]<ポート番号>
```

IP アドレスまたはホスト名には、EADS サーバを実行するマシンの IP アドレスまたはホスト名を指定してください。指定を省略した場合は、ローカルホスト以外のデバッガアプリケーションから接続できません。ポート番号 (EADS サーバがデバッガアプリケーションに接続するためのポート番号) には、1024~65535 の整数が指定できます。

デバッガアプリケーションの接続を待ってから EADS サーバを起動する場合は、suspend 方式に y を指定します。デバッガアプリケーションの接続を待たないで EADS サーバを起動する場合は、suspend 方式に n を指定します。

## (5) リターンコード

0: 正常終了

0 以外: エラー

## (6) 注意事項

1. 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。



2. 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.admin.boot.timeout` パラメタの指定値を調整してください。
3. 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。
4. `-ai` または `--add_id` オプション、`-ap` または `--add_position` オプションを指定する場合、追加する EADS サーバのクラスタ定義ファイルは不要です。追加する EADS サーバにクラスタ定義ファイルがあっても読み込まれません。このため、追加する EADS サーバに不正な内容のクラスタ定義ファイルがあっても、エラーになりません。

5. `-ai` または `--add_id` オプション、`-ap` または `--add_position` オプションの指定時に、EADS サーバの追加でエラーが発生したときは、`eztool status -v` 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。コマンドの排他が解除されていない場合、`eztool unlock` コマンドを実行して排他を解除してください。

EADS サーバが追加されていない場合は、再度 EADS サーバを追加してください。

追加された EADS サーバが縮退またはプロセスダウンした場合は、EADS サーバを復旧させてください。

復旧に失敗した EADS サーバのクラスタ定義ファイルが、ほかの EADS サーバのクラスタ定義ファイルと同じかどうかを確認してください。クラスタ定義ファイルが、ほかの EADS サーバのクラスタ定義ファイルと異なる場合は、復旧に失敗した EADS サーバに、正常に稼働している EADS サーバのクラスタ定義ファイルをコピーしてから、復旧を再度実行してください。

6. `-r` または `--recovery` オプション指定時に EADS サーバの復旧処理でエラーが発生した場合、`eztool status -v` 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。コマンドの排他が解除されていない場合、`eztool unlock` コマンドを実行して排他を解除してください。
7. スケールアウト処理の終了直前のリクエストの応答は、一時的に時間が掛かることがあります。EADS サーバからの応答に対して EADS クライアントを長時間待機させるのではなく、EADS サーバからエラーを返却させ、EADS クライアントでそのエラーに対処させる場合は、次のパラメタの指定を検討してください。

- `eads.scaling.requestQueuing.enable` パラメタ

8. スケールアウト処理の開始直後のリクエストの応答に一時的に時間が掛かる場合は、`eads.scaling.requestQueuing.enable` パラメタとあわせて次のパラメタの指定も検討してください。

- `eads.scaling.moveDataWithUpdateRequest.enable` パラメタ
- `eads.scaling.moveData.interval` パラメタ

9. 次のどちらかの指定をしている場合、リクエストのスループットの向上が期待できますが、スケールアウト処理の実行完了までの時間が長くなります。運用および環境に合わせたチューニングを実施してください。

- `eads.scaling.moveDataWithUpdateRequest.enable` パラメタに `false` を指定している場合で、かつ `ezserver` コマンドに `-ai` または `-ap` オプションを指定している場合

- `eads.scaling.moveData.interval` パラメタに 0 以外を指定している場合で、かつ `ezserver` コマンドに `-ai` または `-ap` オプションを指定している場合
10. `eads.scaling.requestQueuing.enable` パラメタに `false` を指定している場合で、かつ `ezserver` コマンドに `-ai` または `-ap` オプションを指定している場合、追加する EADS サーバの移動元となる EADS サーバで実行しているクライアント API でエラーが発生することがあります。
  11. `-rd` または `--remotedebug` オプションに指定するホスト名には、`[:]` を含むホスト名は指定できません。

## 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.command.logger.dir` パラメタに指定したディレクトリ下の `eads_command_message[n].log` (`[n]`はファイルの通番です) 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>]
        [--messageoff]
        <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。

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。

サブコマンドを指定しなかった場合は、すべてのサブコマンドの概要が表示されます。サブコマンドを指定した場合は、指定したサブコマンドの詳細が表示されます。

#### (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。

ただし、次のコマンドでは指定できません。

- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

### (c) `--messageoff`

標準出力にメッセージを出力したくない場合に指定します。

ただし、コマンドが構文エラーとなった場合、このオプションは無効になります。

また、次のコマンドでは指定できません。

- `eztool threaddump`
- `eztool snapshot`
- `eztool forcestop`

### (d) <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

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		最新パラメタの一覧を表示する	listconf	クラスタ※1
5		キャッシュの一覧を表示する	listcache	クラスタ
6		ストアデータファイルの一覧を表示する	listesd	クラスタ
7		グループ名の一覧を表示する	listgroup	クラスタ※1
8		key の一覧を表示する	listkey	クラスタ※1※2
9		データの格納先を表示する	getposition	クラスタ
10		レンジおよびキャッシュの使用状況を確認する	storeusage	クラスタ
11		排他制御	排他を解除する	unlock
12	メモリ・ディスク管理	キャッシュを作成する	createcache	クラスタ
13		キャッシュを削除する	deletecache	クラスタ
14	データ移行	データを書き出す	export	クラスタ※1
15		データを読み込む	import	クラスタ
16		ストアデータファイルを削除する	deleteesd	クラスタ
17	データ操作	指定したデータを格納する	put	EADS サーバ
18		指定したデータを取得する	get	EADS サーバ
19		指定したデータを削除する	remove	EADS サーバ
20		指定した範囲のデータをすべて削除する	removeall	クラスタ※1※2
21	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
22		ユーザファンクションを実行する	execfunc	クラスタ※1※2※3
23	負荷分散	レンジに格納できる key の数を均等にする	rebalance	クラスタ
24	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
25		キャッシュを再開する	resume	クラスタ
26		キャッシュファイルを書き出す	exportecf	クラスタ※5

項番	利用シーン		サブコマンド名	実行対象
27		キャッシュファイルを読み込む	importecf	クラスタ
28		キャッシュファイルを削除する	deleteecf	クラスタ※5
29		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※4
30	トラブルシュート	スレッドダンプを出力する	threaddump	EADS サーバ
31		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
32	終了	クラスタ内の全 EADS サーバを終了する	stop	クラスタ
33		EADS サーバを強制終了する	forcestop	EADS サーバ
34		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

-k または --key オプションを指定すると、指定した key が格納されている EADS サーバが対象となります。

#### 注※4

--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。

#### 注※5

-l または --local 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。

## 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・スケールアウト
更新系 • eztool close	×	○	○	×

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・スケールアウト
<ul style="list-style-type: none"> <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 rebalance</li> <li>• eztool resume</li> <li>• eztool exportecf</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>				
参照系 <ul style="list-style-type: none"> <li>• eztool status</li> <li>• eztool listconf</li> <li>• eztool listcache</li> <li>• eztool listesd</li> <li>• eztool listecf</li> <li>• eztool getposition</li> <li>• eztool storeusage</li> <li>• eztool listfunc</li> </ul>	○	○	○	○
データ操作 <ul style="list-style-type: none"> <li>• eztool listgroup</li> <li>• eztool listkey</li> <li>• eztool put</li> <li>• eztool get</li> <li>• eztool remove</li> <li>• eztool removeall</li> <li>• eztool execfunc</li> </ul>	○	○	○	○
復旧・スケールアウト <ul style="list-style-type: none"> <li>• ezstart -r または ezstart --recovery</li> <li>• ezserver -r または ezserver --recovery</li> <li>• ezstart -ai</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・スケールアウト
または <code>ezstart --add_id</code> • <code>ezstart -ap</code> または <code>ezstart --add_position</code> • <code>ezserver -ai</code> または <code>ezserver --add_id</code> • <code>ezserver -ap</code> または <code>ezserver --add_position</code>				

(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 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)

### (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

次の表に示すリターンコードが返却されます。

表 14-4 eztool close コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

### (5) 注意事項

クラスタの閉塞に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、クラスタを閉塞状態にしてください。

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。

### 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

次の表に示すリターンコードが返却されます。

表 14-5 eztool open コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (5) 注意事項

クラスタの閉塞解除に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、クラスタの閉塞状態を解除してください。

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。

### 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>]
              [--format <フォーマット名>]
              [--columns <列名> [, <列名>] ...]
              [--filter <フィルタの条件>]
              [--match <一致条件>]
```

### (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>

このオプションは非推奨です。予告なく削除することがあります。

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ハイフン (-) が表示された項目については、カウントされません。

次に示すオプションとは同時に指定できません。

- -s オプション
- --format オプション
- --columns オプション
- --filter オプション
- --match オプション

#### ■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	○	○

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
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)	×	○
Port <sup>※1</sup>	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 <sup>※2</sup>	EADS サーバ間のデータの多重化に使用するポート番号	×	○
ManagePort <sup>※2</sup>	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	○	○
Lock	EADS サーバの排他状態	×	○
KeyCount	EADS サーバ内の key の総数	×	○
UsedCache <sup>※2</sup>	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)	×	○
UsedMemoryRatio <sup>※3</sup>	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	×	○
UsedMemory <sup>※3</sup>	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) (単位: メガバイト)	×	○
MaxMemory <sup>※3</sup>	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。

#### 注※1

コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合、項目名は ClientPort になります。

#### 注※2

コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合にだけ、指定できます。

#### 注※3

コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合は、指定できません。

### ■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ハイフン (-) が表示されるため、ハイフン (-) だけを指定した場合はエラーとなります。ただし、コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合は、情報が取得できない項目にはアスタリスク (\*) が表示されます。その場合、アスタリスク (\*) だけを指定するとエラーになります。

#### (d) `--format` <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

#### (e) `--columns` <列名> [,<列名>] …

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

#### (f) `--filter` <フィルタの条件>

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

#### (g) `--match` <一致条件>

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

### (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。

表示の構成要素については、「[14.4.1 表示内容の構成要素](#)」を参照してください。

### ■`-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      Port  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  Port  Position      Cluster  State  Operation  Lock  KeyCount  UsedMemoryRatio  UsedMemory  MaxMemory  Version
1   XX.XXX.XXX.168  server01   24600  1288490189   online  running none  unlock  582       13           3             23  04-00-00
2   XX.XXX.XXX.168  server02   24700  429496730    online  running none  unlock  618       13           3             23  04-00-00
3   XX.XXX.XXX.168  server03   24800  -429496729   online  running none  unlock  636       13           3             23  04-00-00
4   XX.XXX.XXX.168  server04   24900  -1288490188  online  running none  unlock  591       13           3             23  04-00-00
5   XX.XXX.XXX.168  server05   25000  -2147483648  online  running none  unlock  573       13           3             23  04-00-00
-----

KDEA08002-I          The command will now end.
$

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-6 eztool status コマンドのサマリの表示内容

項番	サマリ名	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	○	○

項番	サマリ名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
		-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。		
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。	○	○

(凡例)

○：表示されます。

表 14-7 eztool status コマンドのメインの表示内容

項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
1	ID	EADS サーバ ID ただし、クラスタに追加する EADS サーバがまだクラスタに参加していない場合、その EADS サーバの ID にはハイフン (-) が表示されます。	○	○
2	IP_Address	EADS サーバの IP アドレス	○	○
3	ServerName	EADS サーバ名 (運用ディレクトリ名) *1*2	×	○
4	Port*3	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	○	○
5	ServerPort*4	EADS サーバ間のデータの多重化に使用するポート番号*2	×	○

項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
6	ManagePort <sup>※4</sup>	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 <sup>※2</sup>	×	○
7	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。 ただし、クラスタに追加する EADS サーバがまだクラスタに参加していない場合、その EADS サーバの位置にはハイフン（-）が表示されます。	○	○
8	Cluster	クラスタへの参加状況 <sup>※5</sup> 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>	○	○
9	State	EADS サーバの状態 <sup>※1※2</sup>	○	○
10	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 <sup>※1※2</sup> 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>• rebalance</li> <li>• resume</li> <li>• exportecf</li> <li>• importecf</li> <li>• deleteecf</li> <li>• compaction</li> <li>• stop</li> <li>• isolate</li> <li>• recovery</li> <li>• add</li> </ul>	○	○



項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
		<p>export, import, rebalance, exportecf, importecf および stop については、処理の進捗率も表示されます。</p> <p>(例) export(88%)</p> <p>ただし、rebalance の場合は、リバランス処理全体の進捗率が、位置を変更中の EADS サーバの所に表示されます。</p> <p>実行中でない場合は none が表示されます。</p>		
11	Lock	<p>EADS サーバの排他状態<sup>※5</sup></p> <p>次のどちらかが表示されます。</p> <ul style="list-style-type: none"> <li>lock 排他している</li> <li>unlock 排他していない</li> </ul>	×	○
12	KeyCount	<p>EADS サーバ内の key の総数<sup>※1※2</sup></p> <p>多重化によってコピーされる key の数も含まれます。</p>	×	○
13	UsedCache <sup>※4</sup>	<p>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) <sup>※1※2</sup></p> <ul style="list-style-type: none"> <li>小数点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> </ul>	×	○
14	UsedMemoryRatio <sup>※6</sup>	<p>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<sup>※1※2</sup></p> <ul style="list-style-type: none"> <li>メモリ使用量の割合は、(メモリ使用量÷メモリ総容量)×100 で算出されます。%単位で表示されるため、小数点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> </ul>	×	○
15	UsedMemory <sup>※6</sup>	<p>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) <sup>※1※2</sup></p> <p>(単位: メガバイト)</p> <ul style="list-style-type: none"> <li>小数点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> </ul>	×	○

項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
16	MaxMemory <sup>※6</sup>	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総容量 <sup>※1※2</sup> (単位：メガバイト) <ul style="list-style-type: none"> <li>• 小数点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 も含みます。</li> </ul>	×	○
17	Version	バージョン情報 <sup>※1※2</sup> 03-00 の場合、03-00-00 と表示されます。	×	○

(凡例)

- ：表示されます。
- ×

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、ハイフン (-) が表示されます。コマンド定義の `eads.command.compat` パラメタに 0300 を指定したときはアスタリスク (\*) になります。

注※2

EADS サーバのクラスタへの参加状況が standby の場合、ハイフン (-) が表示されます。コマンド定義の `eads.command.compat` パラメタに 0300 を指定したときはアスタリスク (\*) になります。

注※3

コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合、列名は ClientPort になります。

注※4

コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合にだけ、表示される列です。

注※5

クラスタの状態が NOT\_AVAILABLE の場合、ハイフン (-) が表示されます。コマンド定義の `eads.command.compat` パラメタに 0300 を指定したときはアスタリスク (\*) になります。

注※6

コマンド定義の `eads.command.compat` パラメタに 0300 を指定した場合、この列は表示されません。

## (6) リターンコード

次に示すオプションを指定したかどうかによって、リターンコードが異なります。

- -s または--single オプション
- -c または--count オプション

- --match オプション

-s オプション, -c オプションおよび--match オプションをすべて指定しなかった場合は、次の表に示すリターンコードが返却されます。

表 14-8 eztool status コマンドのリターンコード (-s, -c および--match オプションの指定なし)

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	150		コマンド実行中に失敗しました。
7	200		コマンドタイムアウトで失敗しました。

-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。

--match オプションと同時に指定し、かつコマンドの実行に成功したときは、--match オプションの結果が優先されます。

表 14-9 eztool status コマンドのリターンコード (-s または --single オプションの指定あり)

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	EADS サーバの状態は稼働状態 (running) です。
2	1	1	EADS サーバの状態は初期化中 (initializing) です。
3	2	2	EADS サーバの状態は初期化状態 (initialized) です。
4	3	3	EADS サーバの状態は閉塞中 (closing) です。
5	4	4	EADS サーバの状態は閉塞状態 (closed) です。
6	5	5	EADS サーバの状態は停止中 (stopping) です。
7	10	10	EADS サーバの状態は縮退状態 (isolated) です。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
8	101	101	コマンドの初期化に失敗しました。
9	110	110	接続に失敗しました。
10	111	111	通信タイムアウトで失敗しました。
11	120	101	構文不正で失敗しました。
12	150		コマンド実行中に失敗しました。
13	200		コマンドタイムアウトで失敗しました。

#### -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。ただし、コマンドの実行に失敗したときは、-c または --count オプションを指定しなかった場合と同じになります。

#### --match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。
- スケールアウト処理中の EADS サーバでこのコマンドを実行した場合、クラスタに参加するまでは、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 listconf (最新パラメタの一覧表示)

### (1) 機能

EADS サーバのシステムプロパティに設定されている、「eads.」で始まるパラメタの一覧を表示します。ただし、03-60 以前の古いパラメタ名で設定されている場合は、新しいパラメタ名で表示されます。

### (2) 規則

- このサブコマンドは、クラスタの状態に関係なく、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 listconf [-v]
                 [-s]
                 [--file <定義ファイルの種別>]
                 [--non_default]
                 [--format <フォーマット名>]
                 [--columns <列名> [, <列名>] ...]
                 [--filter <フィルタの条件>]
                 [--match <一致条件>]
```

### (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 サーバ上の定義ファイルのパラメタだけを確認したい場合に指定します。

#### (c) --file <定義ファイルの種別>

特定の定義ファイルに指定するパラメタだけを確認したい場合に指定します。

次に示す定義ファイルの種別が指定できます。

- server：サーバ定義ファイル
- cluster：クラスタ定義ファイル
- shared：共通設定ファイル

このオプションを指定しない場合、次に示す定義ファイルに指定するパラメタすべてが表示されます。

- サーバ定義ファイル
- クラスタ定義ファイル
- 共通設定ファイル

#### (d) `--non_default`

設定値がデフォルト値以外の EADS サーバが存在するパラメタだけを表示します。ただし、デフォルト値が存在しないパラメタは対象に含まれません。

#### (e) `--format <フォーマット名>`

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

#### (f) `--columns <列名> [,<列名>] ...`

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

#### (g) `--filter <フィルタの条件>`

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

#### (h) `--match <一致条件>`

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

### (5) 出力例

eztool listconf コマンドの実行結果の出力例を次に示します。

表示の構成要素については、「[14.4.1 表示内容の構成要素](#)」を参照してください。

■`-v` または `--verbose` オプションを指定しない場合

```

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

ParameterCount: 79
NonDefaultCount: 10

Parameter                DefaultValue  ServerID:1  ServerID:2  ServerID:3  ServerID:4  ServerID:5
eads.admin.backup.dir    store        store        store        store        store        store
eads.admin.backup.exportCommand.generation.maxNum  2            2            2            2            2            2
eads.admin.backup.stopCommand.generation.maxNum    1            1            1            1            1            1
eads.admin.boot.timeout  60          60          60          60          60          60
eads.admin.operation.connection.timeout            10000       10000       10000       10000       10000       10000
eads.admin.operation.isolate.gracefulstop.waitTime 3000        3000        3000        3000        3000        3000
eads.admin.operation.port 24620       24620       24720       24820       24920       25020
eads.admin.operation.resume.send.datasize         1048576     1048576     1048576     1048576     1048576     1048576
eads.admin.operation.resume.send.interval         0           0           0           0           0           0
eads.cache.disk.getError.isolate.enable           true        true        true        true        true        true
eads.cache.key.maxsize  1024        1024        1024        1024        1024        1024
eads.cache.keyCount      1048576     1048576     1048576     1048576     1048576     1048576
eads.cache.limiter.enable true         true        true        true        true        true
eads.cache.logger.diskCache.filenum              2           2           2           2           2           2
eads.cache.logger.diskCache.filesize             1048576     1048576     1048576     1048576     1048576     1048576
eads.cache.logger.diskCache.rotationStyle        Wrap        Wrap        Wrap        Wrap        Wrap        Wrap

```

< 中略 >

```

-----
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 listconf -v
KDEA08001-I      The command will now start. (subcommand = listconf, parameter = [listconf, -v])

ParameterCount: 79
NonDefaultCount: 10

Parameter                DefaultValue  MinValue  MaxValue  MinLength  MaxLength  ServerID:1  ServerID:2
eads.admin.backup.dir    store        store        store        store        store        store        store
eads.admin.backup.exportCommand.generation.maxNum  2            0          32         1           200         2            2
eads.admin.backup.stopCommand.generation.maxNum    1            1          32         1           1           1            1
eads.admin.boot.timeout  60          60         86400      60          60         60           60
eads.admin.operation.connection.timeout            10000      100         3600000    10000       10000      10000       10000
eads.admin.operation.isolate.gracefulstop.waitTime 3000        0          60000     3000        3000       3000        3000
eads.admin.operation.port 24620       1024       65535     24620       24720     24820       24920       25020
eads.admin.operation.resume.send.datasize         1048576     0          2147483647 1048576     1048576    1048576     1048576
eads.admin.operation.resume.send.interval         0           0          2147483647 0           0           0            0
eads.cache.disk.getError.isolate.enable           true        true        true        true        true        true        true
eads.cache.key.maxsize  1024        1          1024      1024        1024       1024        1024
eads.cache.keyCount      1048576     1024       1073741824 1048576     1048576    1048576     1048576
eads.cache.limiter.enable true         true        true        true        true        true        true
eads.cache.logger.diskCache.filenum              2           1          16         2           2           2            2
eads.cache.logger.diskCache.filesize             1048576     4096       2147483647 1048576     1048576    1048576     1048576
eads.cache.logger.diskCache.rotationStyle        Wrap        Wrap        Wrap        Wrap        Wrap        Wrap        Wrap
eads.failureDetector.assertive.threshold         1           1          49         1           1           1            1
eads.failureDetector.connection.timeout          500        500        60000     500         500       500         500

```

< 中略 >

```

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

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-10 eztool listconf コマンドのサマリの表示内容

項番	サマリ名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
1	ParameterCount	パラメタの数	○	○
2	NonDefaultCount	設定値がデフォルト値以外の EADS サーバが存在するパラメタの数	○	○

(凡例)

○：表示されます。

表 14-11 eztool listconf コマンドのメインの表示内容

項番	列名	説明	表示の有無	
			-v または --verbose オプション指定	
			なし	あり
1	Parameter	パラメタ名 自然順序順に表示されます。	○	○
2	DefaultValue	パラメタのデフォルト値 デフォルト値がないパラメタの場合は、何も表示されません。 EADS サーバごとにデフォルト値が異なる場合は、UNMATCH が表示されます。	○	○
3	MinValue	パラメタの最小値 (数値) 数値以外を指定できるパラメタの場合は、何も表示されません。 EADS サーバごとに最小値が異なる場合は、UNMATCH が表示されます。	×	○
4	MaxValue	パラメタの最大値 (数値) 数値以外を指定できるパラメタの場合は、何も表示されません。 EADS サーバごとに最大値が異なる場合は、UNMATCH が表示されます。	×	○
5	MinLength	パラメタの文字列の長さの最小値 長さの最小値がないパラメタの場合、何も表示されません。 EADS サーバごとに最小値が異なる場合は、UNMATCH が表示されます。	×	○
6	MaxLength	パラメタの文字列の長さの最大値 長さの最大値がないパラメタの場合、何も表示されません。 EADS サーバごとに最大値が異なる場合は、UNMATCH が表示されます。	×	○
7	ServerID:< EADS サーバ ID >	各 EADS サーバでのパラメタの設定値 情報が取得できた EADS サーバの中で、EADS サーバ ID が小さい EADS サーバから順に、各 EADS サーバの情報が表示されます。 -s または --single オプションを指定した場合は、コマンドを実行した EADS サーバの情報だけが表示されます。 クラスタに追加する EADS サーバの EADS サーバ ID が、クラスタにまだ参加していないために未決定の場合、< EADS サーバ ID >にはハイフン (-) が表示されます。	○	○



(凡例)

○：表示されます。

×：表示されません。

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-12 eztool listconf コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。
- システムプロパティに設定されている値は、指定値そのものではなく、ベリファイされた値です。また、機能としてパラメータの値を適用するときは、さらに別の値が使用されることがあるパラメータも存在します。

## 14.3.6 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]
                  [--format <フォーマット名>]
                  [--columns <列名> [, <列名>] ...]
                  [--filter <フィルタの条件>]
                  [--match <一致条件>]
```

### (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) --format <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

#### (c) --columns <列名> [, <列名>] ...

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

## (d) --filter <フィルタの条件>

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

## (e) --match <一致条件>

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

## (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。

表示の構成要素については、「[14.4.1 表示内容の構成要素](#)」を参照してください。

### ■-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 / 16

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 / 16

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.
$
```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-13 eztool listcache コマンドのサマリの表示内容

項番	サマリ名	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	○	○

(凡例)

○：表示されます。

表 14-14 eztool listcache コマンドのメインの表示内容

項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
1	CacheName	キャッシュ名※	○	○
2	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>	○	○
3	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 の数は含みません。	○	○
4	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。	×	○
5	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 の数は含みません。 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。
- ×：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. 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。

注※

実行結果に、同じ値のセルが続く場合、2 行目以降のセルの値は表示が省略されます。

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-15 eztool listcache コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。
- スケールアウト処理、またはリバランス処理の実行中にこのサブコマンドを実行すると、一時的に key 数が少なく表示されることがあります。

## 14.3.7 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>]
                [--format <フォーマット名>]
                [--columns <列名> [, <列名>] ...]
                [--filter <フィルタの条件>]
                [--match <一致条件>]
```

## (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。

### (c) `--format` <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

### (d) `--columns` <列名> [`,<列名>`] …

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

### (e) `--filter` <フィルタの条件>

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

### (f) `--match` <一致条件>

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

## (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。

表示の構成要素については、「[14.4.1 表示内容の構成要素](#)」を参照してください。

■`-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/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
stop  stop_20130402183258 eads_stop_20130402183258_1.esd                0 XX.XXX.XXX.168:24600 /opt/hitachi/xeads/server/servers/test1/store/eads_stop_20130402183258_1.esd
      eads_stop_20130402183258_2.esd                0 XX.XXX.XXX.169:24700 /opt/hitachi/xeads/server/servers/test2/store/eads_stop_20130402183258_2.esd
      eads_stop_20130402183258_3.esd                0 XX.XXX.XXX.170:24800 /opt/hitachi/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/hitachi/xeads/server/servers/test1/store/eads_single_20130402183258_1.esd
      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.
$

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-16 eztool listesd コマンドのサマリの表示内容

項番	サマリ名	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	○	○



項番	サマリ名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
		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。		
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。	○	○

(凡例)

○：表示されます。

表 14-17 eztool listesd コマンドのメインの表示内容

項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
1	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>	○	○
2	StoreDataFileKey	ストアデータファイルキー※	○	○
3	FileName	ストアデータファイル名	×	○
4	FileSize(MB)	ストアデータファイルサイズ (単位：メガバイト) 小数点以下の値は切り捨てて表示されます。	×	○

項番	列名	説明	表示の有無	
			-v または--verbose オプション指定	
			なし	あり
5	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	×	○
6	AbsolutePath	ストアデータファイル名 (絶対パス)	×	○

(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. 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。

注※

実行結果に、同じ値のセルが続く場合、2 行目以降のセルの値は表示が省略されます。

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-18 eztool listesd コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の eads.command.compat パラメータを指定していない場合	コマンド定義の eads.command.compat パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。

## 14.3.8 listgroup (グループ名の一覧表示)

### (1) 機能

キャッシュに格納されている最上位階層のグループ階層名の一覧を表示します。

### (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。
- EADS サーバの位置 (Position) の降順に実行します。
- サブコマンド実行前のチェックの時点で、対象範囲に存在する最上位階層のグループ数が 1,000 個以下の場合に実行できます。

### (3) 形式

```
eztool listgroup [-s] [-v] [-f <強制実行時のグループ数の上限>]
                 <キャッシュ名>
                 [--format <フォーマット名>]
                 [--columns <列名> [, <列名>] ...]
                 [--filter <フィルタの条件>]
                 [--match <一致条件>]
```

## (4) オプションおよび引数

### (a) -s または--single

サブコマンドを実行する EADS サーバが格納先 EADS サーバとなる、最上位階層のグループ階層名だけを表示したい場合に指定します。

### (b) -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。

### (c) -f または--force <強制実行時のグループ数の上限>

対象範囲に存在するグループの数が 1,000 個より多い場合に、サブコマンドを実行できるグループ数の上限値を引き上げて、強制的にサブコマンドを実行したいときに指定します。

ただし、このオプションで上限値を引き上げて実行すると、サブコマンドが最後まで適切に実行できなくなったり、EADS サーバのリソースを大量に消費したりするおそれがあります。

グループ数の上限には、1001~10000 の整数が指定できます。

#### ■ 注意事項

対象範囲に存在するグループの数が、1,000 個より僅かに多い場合などに、このオプションを指定することを検討してください。

各 EADS サーバに格納されている最上位階層グループの合計数が 1,000 個以内であれば、このオプションではなく -s または --single オプションを指定して、各 EADS サーバで順番に `eztool listgroup` コマンドを実行することを推奨します。

### (d) <キャッシュ名>

グループ名の一覧を表示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。

### (e) --format <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

## (f) --columns <列名> [, <列名>] ...

このオプションの詳細については、「14.4.3 列のフィルタリングの指定方法」を参照してください。

## (g) --filter <フィルタの条件>

このオプションの詳細については、「14.4.4 行のフィルタリングの指定方法」を参照してください。

## (h) --match <一致条件>

このオプションの詳細については、「14.4.5 条件一致の指定方法」を参照してください。

## (5) 出力例

eztool listgroup コマンドの実行結果の出力例を次に示します。

表示の構成要素については、「14.4.1 表示内容の構成要素」を参照してください。

### ■-v または--verbose オプションを指定しない場合

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

GroupCount: 20

Server              Position      ID  GroupName
XX.XXX.XXX.168:24600 1288490189   1  group02
                    group04
XX.XXX.XXX.168:24700 429496730    2  group01
                    group09
                    group11
                    group17
                    group18
                    group19
XX.XXX.XXX.168:24800 -429496729   3  group05
                    group10
                    group12
                    group13
                    group15
XX.XXX.XXX.168:24900 -1288490188  4  group06
                    group07
                    group08
                    group14
XX.XXX.XXX.168:25000 -2147483648  5  group03
                    group16
                    group20
-----
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 listgroup -v cache1
KDEA08001-I The command will now start. (subcommand = listgroup, parameter = [listgroup, -v, cache1])

GroupCount: 20

Server          Position      ID  GroupName HashValue  KeyCount
XX.XXX.XXX.168:24600 1288490189 1  group02 2037266030 100
                1288490189 1  group04 1388111144 100
XX.XXX.XXX.168:24700 429496730 2  group09 1269679203 100
                429496730 2  group11 1229725181 100
                429496730 2  group17 1172163628 100
                429496730 2  group19 1006380133 100
                429496730 2  group18 703078836 100
                429496730 2  group01 648506385 100
XX.XXX.XXX.168:24800 -429496729 3  group10 260938713 100
                -429496729 3  group15 232103928 100
                -429496729 3  group12 -56466136 100
                -429496729 3  group05 -73411842 100
                -429496729 3  group13 -163284496 100
XX.XXX.XXX.168:24900 -1288490188 4  group14 -503630070 100
                -1288490188 4  group07 -699511542 100
                -1288490188 4  group08 -992621419 100
                -1288490188 4  group06 -1203276328 100
XX.XXX.XXX.168:25000 -2147483648 5  group16 -1445454454 100
                -2147483648 5  group20 -1917239959 100
                -2147483648 5  group03 -1965919439 100
-----
KDEA08002-I The command will now end.
$

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-19 eztool listgroup コマンドのサマリの表示内容

項番	サマリ名	説明	表示の有無	
			-v または --verbose オプション指定	
			なし	あり
1	GroupCount	EADS サーバに格納されている最上位階層のグループ数	○	○

(凡例)

○：表示されます。

表 14-20 eztool listgroup コマンドのメインの表示内容

項番	列名	説明	表示の有無	
			-v または --verbose オプション指定	
			なし	あり
1	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	○	○
2	Position	EADS サーバの位置 (ハッシュ値) ※	○	○
3	ID	EADS サーバ ID※	○	○

項番	列名	説明	表示の有無	
			-v または --verbose オプション指定	
			なし	あり
4	GroupName	最上位階層のグループ階層名	○	○
5	HashValue	最上位階層のグループの位置 (ハッシュ値)	×	○
6	KeyCount	最上位階層のグループに属している key の数	×	○

(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. Position の値の降順で表示されます。
2. Position の値が同じ行は、GroupName の値の自然順序順に表示されます。-v または --verbose オプションを指定する場合、Position の値が同じ行は、HashValue の値の降順に表示されます。

注※

実行結果に、同じ値のセルが続く場合、2 行目以降のセルの値は表示が省略されます。

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-21 eztool listgroup コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。
5	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (7) 注意事項

- このサブコマンドを実行すると、AP を作成しなくても、最上位階層のグループ階層名の一覧を確認できます。ただし、対象範囲に存在する最上位階層のグループの数が 1,000 個を超える場合は、AP を作成する必要があります。
- このサブコマンドの実行中に、最上位階層のグループ階層名が追加された場合、グループ階層名が格納された領域によって、次のように動作します。
  - グループ名を取得済みの領域にグループ階層名が格納されたとき追加されたグループ階層名が一覧に表示されません。
  - グループ名を取得されていない領域にグループ階層名が格納されたとき追加されたグループ階層名が一覧に表示されます。

なお、最上位階層のグループ階層名の数の確認は実際にグループ名の一覧を取得する前に行うため、確認後に新規グループ名を含む大量の key を追加すると、上限数を超えるグループ名が表示されます。

- 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。
- 対象範囲に大量のグループ名が存在する場合、リソースの使用量が増加するため、注意してください。
- グループに属していない key の数は確認できません。
- このサブコマンドの実行中に、EADS サーバがクラスタに追加または復旧された場合、追加または復旧された EADS サーバが実行対象から外れて、一部のグループ名が表示されないことがあります。その場合は、サブコマンドを再実行してください。

## 14.3.9 listkey (key の一覧表示)

### (1) 機能

キャッシュに格納されている key の一覧を表示します。

### (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。
- EADS サーバの位置 (Position) の降順に実行します。
- サブコマンド実行前のチェックの時点で、対象範囲に存在する key の数が 1,000 個以下の場合に実行できます。



### (3) 形式

```
eztool listkey [-g <グループ名>|-s] [-f <強制実行時のkeyの数の上限>]
               <キャッシュ名>
               [--format <フォーマット名>]
               [--columns <列名> [, <列名>] ...]
               [--filter <フィルタの条件>]
               [--match <一致条件>]
```

### (4) オプションおよび引数

#### (a) -g または--group <グループ名>

グループに属している key だけを表示したい場合に指定します。

グループ名として指定できるデータについては、「15.2.2(2) グループ名として指定できるデータ」を参照してください。

#### (b) -s または--single

サブコマンドを実行する EADS サーバに格納されている key だけを表示したい場合に指定します。

#### (c) -f または--force <強制実行時の key の数の上限>

対象範囲に存在する key の数が 1,000 個より多い場合に、サブコマンドを実行できる key の数の上限値を引き上げて、強制的にサブコマンドを実行したいときに指定します。

ただし、このオプションで上限値を引き上げて実行すると、サブコマンドが最後まで適切に実行できなくなったり、EADS サーバのリソースを大量に消費したりするおそれがあります。

key の数の上限には、1001~10000 の整数を指定します。

#### 注意事項

対象範囲に存在する key の数が、1,000 個より僅かに多い場合などに、このオプションを指定することを検討してください。

- 各 EADS サーバに格納されている key の総数が 1,000 個以内の場合  
このオプションではなく -s または --single オプションを指定して、各 EADS サーバで順番に `eztool listkey` コマンドを実行してください。
- 各最上位階層のグループに属する key の総数が 1,000 個以内の場合  
`eztool listgroup` コマンドで確認した最上位階層のグループ階層名を、`eztool listkey` コマンドの -g または --group オプションの引数に指定して、各グループで順番に実行してください（ただし、グループに属さない key は、この方法では確認できません）。

## (d) <キャッシュ名>

key の一覧を表示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。

## (e) --format <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

## (f) --columns <列名> [,<列名>] ...

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

## (g) --filter <フィルタの条件>

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

## (h) --match <一致条件>

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

## (5) 出力例

eztool listkey コマンドの実行結果の出力例を次に示します。

表示の構成要素については、「[14.4.1 表示内容の構成要素](#)」を参照してください。

```

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

KeyCount: 25

Server          Position      ID  Key
XX.XXX.XXX.168:24600 1288490189  1  [1]groupA:element01
                [1]groupA:element02
                [1]groupA:element03
                [1]groupA:element04
                [1]groupA:element05
XX.XXX.XXX.168:24700 429496730  2  [2]groupB:element01
                [2]groupB:element02
                [2]groupB:element03
                [2]groupB:element04
                [2]groupB:element05
XX.XXX.XXX.168:24800 -429496729  3  [3]groupC:element01
                [3]groupC:element02
                [3]groupC:element03
                [3]groupC:element04
                [3]groupC:element05
XX.XXX.XXX.168:24900 -1288490188 4  [4]groupD:element01
                [4]groupD:element02
                [4]groupD:element03
                [4]groupD:element04
                [4]groupD:element05
XX.XXX.XXX.168:25000 -2147483648 5  [5]groupE:element01
                [5]groupE:element02
                [5]groupE:element03
                [5]groupE:element04
                [5]groupE:element05
-----
KDEA08002-I      The command will now end.
$

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-22 eztool listkey コマンドのサマリの表示内容

項番	サマリ名	説明
1	KeyCount	指定した範囲に存在する key の数

表 14-23 eztool listkey コマンドのメインの表示内容

項番	列名	説明
1	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
2	Position	EADS サーバの位置 (ハッシュ値) ※
3	ID	EADS サーバ ID※
4	Key	指定した範囲に存在する key

注

実行結果は、次の優先順位に従って表示されます。

1. Position の値の降順で表示されます。
2. Position の値が同じ行は、Key の値の自然順序順に表示されます。

注※

実行結果に、同じ値のセルが続く場合、2行目以降のセルの値は表示が省略されます。

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-24 eztool listkey コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。
5	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (7) 注意事項

- このサブコマンドを実行すると、AP を作成しなくても、key の一覧を確認できます。ただし、対象範囲に存在する key の数が 1,000 個を超える場合は、AP を作成する必要があります。
- このサブコマンドの実行中に key が追加された場合、追加された key が格納された領域によって、次のように動作します。
  - key の一覧を取得済みの領域に key が格納されたとき追加された key は一覧に表示されません。
  - key の一覧を取得していない領域に key が格納されたとき追加された key は一覧に表示されます。

なお、key の数の確認は実際に key の一覧を取得する前に行うため、確認後に大量の key を追加すると、上限数を超える key が表示されます。

- 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。
- 大量のデータを格納している場合、リソースの使用量が増加するため、注意してください。

- このサブコマンドの実行中に、EADS サーバがクラスタに追加または復旧された場合、追加または復旧された EADS サーバが実行対象から外れて、一部の key が表示されないことがあります。その場合は、このサブコマンドを再実行してください。

## 14.3.10 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。
- クラスタに追加する EADS サーバがクラスタにまだ参加していない場合、その EADS サーバでは実行できません。

### (3) 形式

```
eztool getposition <keyまたはグループ名> [-l]
                [--format <フォーマット名>]
                [--columns <列名> [, <列名>] ...]
                [--filter <フィルタの条件>]
                [--match <一致条件>]
```

## (4) オプションおよび引数

### (a) <key またはグループ名>

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, または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」, または「15.2.2(2) グループ名として指定できるデータ」を参照してください。

### (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。

### (c) --format <フォーマット名>

このオプションの詳細については、「14.4.2 表示形式の指定方法」を参照してください。

### (d) --columns <列名> [,<列名>] …

このオプションの詳細については、「14.4.3 列のフィルタリングの指定方法」を参照してください。

### (e) --filter <フィルタの条件>

このオプションの詳細については、「14.4.4 行のフィルタリングの指定方法」を参照してください。

### (f) --match <一致条件>

このオプションの詳細については、「14.4.5 条件一致の指定方法」を参照してください。

## (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。

表示の構成要素については、「14.4.1 表示内容の構成要素」を参照してください。

```
$ 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.
$
```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-25 eztool getposition コマンドのサマリの表示内容

項番	サマリ名	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

表 14-26 eztool getposition コマンドのメインの表示内容

項番	列名	説明
1	No.	データの優先度* 昇順で表示されます。
2	IP_Address	EADS サーバの IP アドレス
3	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
4	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

次の表に示すリターンコードが返却されます。

表 14-27 eztool getposition コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。

### 14.3.11 storeusage (レンジおよびキャッシュの使用状況の確認)

#### (1) 機能

格納されている key の数、およびメモリ使用量をレンジ単位で表示します。また、レンジ単位、キャッシュ単位に設定されている、データ総量監視の上限値を表示します。

#### (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、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。
  - 初期化中 (initializing)
  - 初期化状態 (initialized)
  - 稼働状態 (running)
  - 閉塞中 (closing)
  - 閉塞状態 (closed)

### (3) 形式

```
eztool storeusage [--replica|--cache <キャッシュ名>]
                  [--format <フォーマット名>]
                  [--columns <列名> [, <列名>] ...]
                  [--filter <フィルタの条件>]
                  [--match <一致条件>]
```

### (4) オプションおよび引数

#### (a) --replica

多重化によってコピーされた key の数も確認したい場合に指定します。

このオプションを指定しない場合、多重化によってコピーされた key の数は含まれません。

#### (b) --cache <キャッシュ名>

特定のキャッシュの key の数およびメモリ使用量を、レンジ単位で表示したい場合に指定します。ディスクキャッシュまたは 2Way キャッシュの場合は、ディスクの使用量とその上限値が表示されます。

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。

このオプションを指定しない場合は、各キャッシュの key の数をレンジごとに合計した値が、key の総数として表示されます。

#### (c) --format <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

## (d) --columns <列名> [,<列名>] ...

このオプションの詳細については、「14.4.3 列のフィルタリングの指定方法」を参照してください。

## (e) --filter <フィルタの条件>

このオプションの詳細については、「14.4.4 行のフィルタリングの指定方法」を参照してください。

## (f) --match <一致条件>

このオプションの詳細については、「14.4.5 条件一致の指定方法」を参照してください。

## (5) 出力例

eztool storeusage コマンドの実行結果の出力例を次に示します。

表示の構成要素については、「14.4.1 表示内容の構成要素」を参照してください。

### ■--replica オプションおよび--cache オプションを指定しない場合

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

RangeID StartPosition EndPosition Server StoredExternalHeapSize ExternalHeapSizeLimit RangeKeyCount KeyCountLimit
1 1288490189 2147483647 XX.XXX.XXX.168:24600 1 7 218 1048576
2 429496730 1288490188 XX.XXX.XXX.168:24700 1 7 209 1048576
3 -429496729 429496729 XX.XXX.XXX.168:24800 1 7 209 1048576
4 -1288490188 -429496730 XX.XXX.XXX.168:24900 1 7 173 1048576
5 -2147483648 -1288490189 XX.XXX.XXX.168:25000 1 7 191 1048576

-----

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

### ■--replica オプションを指定する場合

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

RangeID StartPosition EndPosition Server StoredExternalHeapSize KeyCount (Server ID:1 Position:1288490189) KeyCount (Server ID:2 Position:429496730)
1 1288490189 2147483647 XX.XXX.XXX.168:24600 1 ----- 218
2 429496730 1288490188 XX.XXX.XXX.168:24700 1 ----- 209
3 -429496729 429496729 XX.XXX.XXX.168:24800 1 -----
4 -1288490188 -429496730 XX.XXX.XXX.168:24900 1 ----- 173
5 -2147483648 -1288490189 XX.XXX.XXX.168:25000 1 ----- 191

-----

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

### ■--cache オプションを指定する場合

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

RangeID StartPosition EndPosition Server StoredExternalHeapSize StoredDiskSize DiskSizeLimit RangeKeyCount
1 1288490189 2147483647 XX.XXX.XXX.168:24600 1 ----- 218
2 429496730 1288490188 XX.XXX.XXX.168:24700 1 ----- 209
3 -429496729 429496729 XX.XXX.XXX.168:24800 1 ----- 209
4 -1288490188 -429496730 XX.XXX.XXX.168:24900 1 ----- 173
5 -2147483648 -1288490189 XX.XXX.XXX.168:25000 1 ----- 191

-----

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

メインの表示内容を次の表に示します。

表 14-28 eztool storeusage コマンドのメインの表示内容

項番	列名	説明	表示の有無		
			--replica オプションまたは--cache オプションの指定		
			どちらも指定なし	--replica オプション指定あり	---cache オプション指定あり
1	RangeID	レンジ ID	○	○	○
2	StartPosition	各レンジの開始位置 (ハッシュ値) ※1 位置 (ハッシュ値) は降順で表示されます。	○	○	○
3	EndPosition	各レンジの終了位置 (ハッシュ値) ※1	○	○	○
4	Server	各レンジのデータの格納先 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※2 次の形式で表示されます。 IP アドレス:ポート番号	○	○	○
5	StoredExternalHeapSize	メモリキャッシュおよび 2Way キャッシュのメモリ使用量の合計※3 (単位: メガバイト) • 小数点以下の値は切り捨てて表示されます。 • ディスクキャッシュのメモリ使用量は 0 として扱われます。	○	○	○
6	ExternalHeapSizeLimit	メモリ使用量の上限値※3※4 (単位: メガバイト) ディスクキャッシュの場合, 0 が表示されます。	○	×	×
7	StoredDiskSize	ディスクキャッシュまたは 2Way キャッシュのディスク使用量※3※5 (単位: メガバイト) 小数点以下の値は切り捨てて表示されます。	×	×	○
8	DiskSizeLimit	ディスクキャッシュまたは 2Way キャッシュのディスク使用量の上限値※3※4※5 (単位: メガバイト)	×	×	○
9	RangeKeyCount	各レンジの key の総数※3 多重化によってコピーされた key の数は含みません。	○	×	○
10	KeyCountLimit	各レンジの key の総数の上限値※3※4 多重化によってコピーされる key の数は含みません。	○	×	×

項番	列名	説明	表示の有無		
			--replica オプションまたは--cache オプションの指定		
			どちらも指定なし	--replica オプション指定あり	---cache オプション指定あり
11	KeyCount(ServerID:x Position:y) <sup>※6</sup>	EADS サーバごとの各レンジの key の総数 <sup>※3</sup> <ul style="list-style-type: none"> <li>多重化によってコピーされた key の数も含まれます。</li> <li>縮退状態 (isolated), または起動していない EADS サーバの key の数は含みません。</li> <li>データを多重化している場合, EADS サーバの位置 (ハッシュ値) の降順に表示されます。</li> <li>コマンド実行時点での値を取得するため, データの格納先 EADS サーバとコピー先 EADS サーバで値が異なる場合があります。</li> <li>データの格納先でもコピー先でもない EADS サーバの場合, 何も表示されません。</li> </ul>	×	○	×

(凡例)

- ：表示されます。
- ×：表示されません。

注※1

各レンジの範囲を次に示します。

- StartPosition の値が EndPosition の値よりも小さい場合  
StartPosition の値から EndPosition の値までの範囲
- StartPosition の値が EndPosition の値よりも大きい場合  
次に示す範囲を合わせた範囲となります。
  - StartPosition の値から 2,147,483,647 までの範囲
  - -2,147,483,648 から EndPosition の値までの範囲

注※2

該当するレンジのデータの格納先 EADS サーバ, およびデータのコピー先 EADS サーバが, すべて縮退状態 (isolated) またはプロセスダウンしている場合, ハイフン (-) が表示されます。

注※3

該当するレンジの情報が取得できない場合, ハイフン (-) が表示されます。

注※4

データ総量監視機能が無効になっている場合, ハイフン (-) が表示されます。

## 注※5

メモリキャッシュの場合、ハイフン (-) が表示されます。

## 注※6

x には、EADS サーバのサーバ ID が表示されます。y には、EADS サーバの位置 (ハッシュ値) が表示されます。

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-29 eztool storeusage コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

### --match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。
- スケールアウト処理の実行中、またはリバランス処理の実行中は、クラスタの構成が変更されることが原因で、このサブコマンドの実行に失敗することがあります。
- リバランス処理の実行中にこのサブコマンドを実行すると、一時的に次の値が少なく表示されることがあります。
  - StoredExternalHeapSize

- RangeKeyCount
- KeyCount(ServerID:x Position:y)
- リバランス処理の実行中にこのサブコマンドを実行すると、一時的に次の値が小さくなることがあります。
  - ExternalHeapSizeLimit
  - KeyCountLimit

## 14.3.12 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

次の表に示すリターンコードが返却されます。

表 14-30 eztool unlock コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (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。

## 14.3.13 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 16 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)
- 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 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

次の表に示すリターンコードが返却されます。

表 14-31 eztool createcache コマンドのリターンコード

項 番	リターンコード		リターンコードの説明
	コマンド定義の eads.command.compat パラメ タを指定していない場合	コマンド定義の eads.command.compat パラ メタに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しまし た。



項 番	リターンコード		リターンコードの説明
	コマンド定義の eads.command.compat パラメ タを指定していない場合	コマンド定義の eads.command.compat パラ メタに 0300 を指定した場合	
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (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、次に示す条件に該当するとエラーになります。
  - データ総量監視機能が有効な場合  
共通設定の `eads.java.external.heapsize` パラメタの value 部分が格納される領域のサイズを多重度で割った結果が 1 メガバイト未満のとき
  - データ総量監視機能が無効な場合  
共通設定の `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
- 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 status -v` コマンドでクラスタの状態を確認してください。コマンドの排他が解除されていない場合、`eztool unlock` コマンドで排他を解除してください。
  - `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 deletetecache` 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 deletetecf` 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。
  - 共通設定の `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。

## 14.3.14 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 deletecache <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

次の表に示すリターンコードが返却されます。

表 14-32 eztool deletecache コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (6) 注意事項

キャッシュの削除に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、キャッシュを削除してください。

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。

### 14.3.15 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`

障害運用で使用するためのオプションです。通常運用および保守運用では使用しないでください。

このコマンドは、クラスタ動作不能 (NOT\_AVAILABLE)、またはクラスタ一部稼働中 (PARTIALLY\_AVAILABLE) となり、`eztool export` コマンドによるデータのバックアップができない場合に指定します。

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ないため、データの書き出しと次に示す処理を同時に行った場合、動作は保証されません。

- キャッシュの作成および削除
- データの更新操作、またはデータの更新を伴う操作（復旧処理、スケールアウト処理、またはリバランス処理など）
- ストアデータファイルの削除

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

次の表に示すリターンコードが返却されます。

表 14-33 eztool export コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (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、違う形式で指定してください。

- 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。

- データの書き出しに失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、データを書き出してください。

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。

## 14.3.16 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>]
               [--convertid <EADSサーバID変換ルール>]
               [<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) --convertid < EADS サーバ ID 変換ルール >

格納先の EADS サーバ ID を指定して key をグルーピングしている場合 (EADS サーバ ID 指定グループを使用している場合) に、指定した EADS サーバ ID を別の EADS サーバ ID に変換して、データを読み込みたいときに指定します。

EADS サーバ ID 変換ルールは、次の形式で指定します。

```
<変換元EADSサーバID>><変換先EADSサーバID>
```

変換元 EADS サーバ ID および変換先 EADS サーバ ID には、1~96 の整数が指定できます (01 や 02 など、10 の位が 0 の整数は指定できません)。

変換元 EADS サーバ ID と変換先 EADS サーバ ID に、同じ値は指定できません。

EADS サーバ ID 変換ルールを複数指定する場合は、コンマ区切りで指定します。EADS サーバ ID 変換ルールを複数指定した場合に、指定順による優先度の差はありません。ただし、同じ変換元 EADS サーバ ID は、複数指定できません。

次に示す文字または文字列は無視されます。

- 先頭または末尾のコンマ  
(例) 「--convertid ,1>2,」
- コンマで区切られた空文字列、またはコンマで区切られたスペースだけの文字列  
(例) 「--convertid 1>2,,」

## (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。

`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

次の表に示すリターンコードが返却されます。

表 14-34 `eztool import` コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	10	10	データ自動削除機能を使用している場合，レンジのデータ件数がデータ自動削除機能のしきい値（キャッシュ定義の <code>eads.cache.eviction.keyCount</code> パラメータの指定値）を超えたため，一部のデータはインポートされませんでした。
3	101	101	コマンドの初期化に失敗しました。
4	110		接続に失敗しました。
5	111		通信タイムアウトで失敗しました。
6	120		構文不正で失敗しました。
7	130		コマンドを実行できない状態のため失敗しました。
8	131		ほかのコマンドが実行中のため失敗しました。
9	150		コマンド実行中に失敗しました。
10	200		コマンドタイムアウトで失敗しました。

## (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。
- データ自動削除機能を使用する場合は、レンジごとのしきい値を超えるデータは読み捨てられます。ただし、全レンジでしきい値の上限を超えた場合でも、put する key のデータがすでに存在する場合に更新時刻が新しいデータがあれば上書きできるように、読み込み処理は続きます。
- EADS サーバ ID 指定グループを含む key のデータを登録する際に、指定された EADS サーバ ID のレンジが存在しない場合、そのデータは登録しません。このとき、1 回だけ警告メッセージが出力されます。
- このサブコマンドの実行中、データ総量監視機能は動作しません。
- データ自動削除機能を使用している場合、`eztool export` コマンドに `-s` または `--single` オプションを指定して出力したストアデータファイルは読み込まないで、`eztool createcache` コマンドでキャッシュを再作成し、必要に応じてデータを再投入することを推奨します。また、`eztool export` コマンドに `-s` または `--single` オプションを指定して出力したストアデータファイルを読み込む場合は、データ自動削除機能のしきい値 (`eads.cache.eviction.keyCount` パラメタの指定値) をストアデータファイル出力時に設定してあったしきい値よりも小さい値に変更しないでください。
- データの読み込みに失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、データを読み込んでください。
  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。

## 14.3.17 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

次の表に示すリターンコードが返却されます。

表 14-35 eztool deleteesd コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (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。
- ストアデータファイルの削除に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、ストアデータファイルを削除してください。
  - `eztool status -v` コマンドでクラスタの状態を確認する。
  - `eztool unlock` コマンドで排他を解除する。

### 14.3.18 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。

このサブコマンドは、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 として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。

### (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

次の表に示すリターンコードが返却されます。

表 14-36 eztool put コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに <code>0300</code> を指定した場合	
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。
5	200		コマンドタイムアウトで失敗しました。

## (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。

### 14.3.19 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。

このサブコマンドは、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>
          [--format <フォーマット名>]
          [--columns <列名> [, <列名>] ...]
          [--filter <フィルタの条件>]
          [--match <一致条件>]
```

#### (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 として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。

## (c) --format <フォーマット名>

このオプションの詳細については、「14.4.2 表示形式の指定方法」を参照してください。

## (d) --columns <列名> [,<列名>] …

このオプションの詳細については、「14.4.3 列のフィルタリングの指定方法」を参照してください。

## (e) --filter <フィルタの条件>

このオプションの詳細については、「14.4.4 行のフィルタリングの指定方法」を参照してください。

## (f) --match <一致条件>

このオプションの詳細については、「14.4.5 条件一致の指定方法」を参照してください。

## (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。

表示の構成要素については、「14.4.1 表示内容の構成要素」を参照してください。

```
$ 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.
```

サマリの表示内容を次の表に示します。

表 14-37 eztool get コマンドのサマリの表示内容

項番	サマリ名	説明
1	Value	取得した value

項番	サマリ名	説明
2	ValueSize	取得した value のサイズ (単位: バイト)

## (6) リターンコード

次の表に示すリターンコードが返却されます。

表 14-38 eztool get コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。
5	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0, 一致しなかった場合は 1 を, リターンコードとして返します。ただし, コマンドの実行に失敗したときは, --match オプションを指定しなかった場合と同じになります。

## (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。
- format オプション, --columns オプション, および--filter オプションは, 指定しても効果がありません。今後の機能拡張を考慮して, あらかじめ用意しています。
- 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。

### 14.3.20 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。



このサブコマンドは、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 として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。

## (5) リターンコード

次の表に示すリターンコードが返却されます。

表 14-39 eztool remove コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに <code>0300</code> を指定した場合	
5	200		コマンドタイムアウトで失敗しました。

## (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。

### 14.3.21 removeall (全データの削除)

#### (1) 機能

指定した範囲の key，および key に関連づけられた value をすべて削除します。

#### (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。
- EADS サーバの位置 (Position) の降順に実行します。

#### (3) 形式

```
eztool removeall [-g <グループ名>|-s] <キャッシュ名>
```

#### (4) オプションおよび引数

##### (a) -g または --group <グループ名>

特定のグループに属している value を削除したい場合に指定します。

グループ名として指定できるデータについては，「15.2.2(2) グループ名として指定できるデータ」を参照してください。

##### (b) -s または --single

サブコマンドを実行する EADS サーバがマスタ (コピー元) となる value を削除したい場合に指定します。

## (c) <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。

## (5) リターンコード

表 14-40 eztool removeall コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。
5	200		コマンドタイムアウトで失敗しました。

## (6) 注意事項

- このサブコマンドを実行すると、AP を作成しなくても、指定範囲の value をすべて削除できます。ただし、次に示す場合など、このサブコマンドのオプションや引数で目的を達成できない場合は、AP を作成する必要があります。
  - key を複数指定して、各 key に関連づけられた value を削除したい場合
  - 一部のデータの削除に失敗した際の処理を実装したい場合
- このサブコマンドの実行中にデータが追加された場合、追加されたデータが格納された領域によって、次のように動作します。
  - データが削除済みの領域に、追加されたデータが格納されたとき  
追加されたデータは削除されません。
  - データが削除されていない領域に、追加されたデータが格納されたとき  
追加されたデータは削除されます。

- このサブコマンドの実行中に、EADS サーバがクラスタに追加または復旧された場合、追加または復旧された EADS サーバが実行対象から外れて、一部のデータが削除されないことがあります。その場合は、このサブコマンドを再実行してください。

## 14.3.22 listfunc (ユーザファンクションの実行可否の表示)

### (1) 機能

ユーザファン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] [<ユーザファンクション名>]
                [--format <フォーマット名>]
                [--columns <列名> [, <列名>] ...]
                [--filter <フィルタの条件>]
                [--match <一致条件>]
```

### (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。

指定したユーザ機能の情報のみを表示したい場合に指定します。

ユーザ機能名には、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。

### (c) --format <フォーマット名>

このオプションの詳細については、「14.4.2 表示形式の指定方法」を参照してください。

### (d) --columns <列名> [, <列名>] ...

このオプションの詳細については、「14.4.3 列のフィルタリングの指定方法」を参照してください。

### (e) --filter <フィルタの条件>

このオプションの詳細については、「14.4.4 行のフィルタリングの指定方法」を参照してください。

### (f) --match <一致条件>

このオプションの詳細については、「14.4.5 条件一致の指定方法」を参照してください。

## (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。

表示の構成要素については、「14.4.1 表示内容の構成要素」を参照してください。

### (a) -v または--verbose オプションを指定しない場合

```
$ eztool listfunc
KDEA08001-l The command will now start. (subcommand = listfunc, parameter = [listfunc])

FunctionCount: 3

FunctionName Enable Disable
FunctionA     3         0
FunctionB     2         1
FunctionC     0         1
-----
KDEA08002-l The command will now end.
$
```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-41 eztool listfunc コマンドのサマリの表示内容 (-v または--verbose オプションの指定なし)

項番	サマリ名	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。

表 14-42 eztool listfunc コマンドのメインの表示内容 (-v または--verbose オプションの指定なし)

項番	列名	説明
1	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。
2	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
3	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.
$

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-43 eztool listfunc コマンドのサマリの表示内容 (-v または--verbose オプションの指定あり)

項番	サマリ名	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。

表 14-44 eztool listfunc コマンドのメインの表示内容 (-v または--verbose オプションの指定あり)

項番	列名	説明
1	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

項番	列名	説明
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。
3	Status	ユーザファンクションの実行可否 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> ユーザファンクションが存在しない場合、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

次の表に示すリターンコードが返却されます。

表 14-45 eztool listfunc コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

### --match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。

### 14.3.23 execfunc (ユーザ関数の実行)

#### (1) 機能

指定したユーザ関数を実行します。

ユーザ関数を実行する 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関数を実行できます。

#### (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) の降順にユーザ関数を実行します。実行途中で、一部の EADS サーバでユーザ関数の実行に失敗しても、処理を継続します。

#### (3) 形式

```
eztool execfunc [-k <key>|-g <グループ名>|-s]
                 <キャッシュ名> <ユーザ関数名> [<ユーザ関数引数>]
                 [--format <フォーマット名>]
                 [--columns <列名> [, <列名>] ...]
                 [--filter <フィルタの条件>]
                 [--match <一致条件>]
```

#### (4) オプションおよび引数

##### (a) -k または --key <key >

指定した key が格納されている EADS サーバでユーザ関数を実行したい場合に指定します。

key として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。

##### (b) -g または --group <グループ名>

指定したグループが格納されている EADS サーバでユーザ関数を実行したい場合に指定します。

グループ名として指定できるデータについては、「[15.2.2\(2\) グループ名として指定できるデータ](#)」を参照してください。



### (c) **-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。

### (d) **<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。

### (e) **<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。

### (f) **<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。

### (g) **--format <フォーマット名>**

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

### (h) **--columns <列名> [,<列名>] ...**

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

### (i) **--filter <フィルタの条件>**

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

### (j) **--match <一致条件>**

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

## (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。

表示の構成要素については、「14.4.1 表示内容の構成要素」を参照してください。

```
$ 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.
```

メインの表示内容を次の表に示します。

表 14-46 eztool execfunc コマンドのメインの表示内容

項番	列名	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 自然順序順で表示されます。
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

次の表に示すリターンコードが返却されます。

表 14-47 eztool execfunc コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の eads.command.compat パラメータを指定していない場合	コマンド定義の eads.command.compat パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	120		構文不正で失敗しました。
4	150		コマンド実行中に失敗しました。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
			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。
5	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。
- 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。

## 14.3.24 rebalance (レンジに格納できる key の数の均等化)

### (1) 機能

クラスタ内の各 EADS サーバのレンジに格納できる key の数が均等になるように、各 EADS サーバの位置を変更して、レンジを分割および結合します (リバランス処理)。リバランス処理では、EADS サーバの位置を変更するので、データの格納先の EADS サーバが変わります。そのため、EADS サーバ間でデータの転送処理を行います。

EADS サーバを変更する位置や、変更する順序は、EADS が決定します。

EADS サーバの位置の変更が大きい場合 (転送が必要なデータの量が多い場合) には、位置の変更処理を複数回に分けて、目標の位置に到達するまで変更処理を繰り返します。

## (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、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)
- このサブコマンドの処理対象は、クラスタへの参加状況が online の EADS サーバです。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、このサブコマンドは実行できません。クラスタへの参加状況は、`eztool status` コマンドで確認できます。
- このサブコマンドは、次の場合には実行できません。
  - キャッシュタイプが、ディスクキャッシュまたは 2Way キャッシュであるキャッシュが存在する場合
  - データ自動削除機能を使用するキャッシュと、データ自動削除機能を使用しないキャッシュが混在する場合
  - データの多重度が 1 の場合
  - クラスタを構成する EADS サーバ数が「データの多重度×2-1」の場合

## (3) 形式

```
eztool rebalance [--datasize <送信データサイズ>] [-i <送信間隔>]
```

```
eztool rebalance [--break]
```

## (4) オプションおよび引数

### (a) --datasize <送信データサイズ>

データ転送で送信するデータサイズ (単位: バイト) を設定したい場合に、このオプションを指定します。

送信データサイズには、10240~2147483647 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、サーバ定義の `eads.rebalance.transfer.datasize` パラメタに指定したデータサイズで、データを転送します。

### (b) -i または --interval <送信間隔>

データを転送する間隔 (単位: ミリ秒) を設定したい場合に、このオプションを指定します。

送信間隔には、1~60000 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、サーバ定義の `eads.rebalance.transfer.interval` パラメタに指定した間隔で、データが転送されます。

### (c) --break

リバランス処理を中断したい場合に、このオプションを指定します。

リバランス処理の中断は、EADS サーバの位置の変更処理が完了するタイミングで行います。EADS サーバの位置を変更する際に、複数回、位置の変更処理をする場合、その 1 回の変更処理の完了時が中断の区切りとなります。EADS サーバの位置の変更処理の途中では中断できません。

## (5) リターンコード

次の表に示すリターンコードが返却されます。

表 14-48 eztool rebalance コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメタを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメタに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	1	1	すでに格納できる key の数が均等になっているため、リバランス処理を実行しませんでした。
3	10	10	--break オプションが指定されましたが、リバランス処理は実行されていません。
4	11	11	--break オプションが指定されましたが、すでにリバランス処理の中断が予約されています。
5	101	101	コマンドの初期化に失敗しました。
6	110		接続に失敗しました。
7	111		通信タイムアウトのため失敗しました。
8	120		構文不正のため失敗しました。
9	130		コマンドを実行できない状態のため失敗しました。
10	131		ほかのコマンドが実行中のため失敗しました。
11	150		コマンド実行中に失敗しました。
12	200		コマンドタイムアウトのため失敗しました。

## (6) 注意事項

1. 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。
2. `eztool rebalance --break` コマンドの実行中に、`eztool rebalance` コマンドがタイムアウトまたは停止した場合は、`eztool rebalance --break` コマンドは正常終了しません。`eztool rebalance --break` コマンドがタイムアウトするまで待つか、または `eztool rebalance --break` コマンドを停止してください。
3. ある EADS サーバの位置の変更処理が終わってから、次の位置の変更処理が始まるまでの時間は、位置の変更処理の実行中とは見なされません。このため、`eztool rebalance --break` コマンドを実行しても、EADS サーバの位置の変更処理が実行されていないというメッセージが出力され、中断できないことがあります。その場合は、`eztool rebalance --break` コマンドが実行できるまで、コマンドを繰り返し実行してください。
4. クラスタ内のメモリの空き容量が少ない状態で、リバランス処理を実行することは推奨しません。スケールアウト処理で EADS サーバを追加して、クラスタ内のメモリ容量を増加させてから、リバランス処理を実行することを検討してください。
5. EADS サーバ ID 指定グループに属する key のデータは、EADS サーバの位置が変更されても、ほかの EADS サーバに移動されないで、指定された EADS サーバに格納されたままになります。このようなデータが大量に格納されている場合は、リバランス処理を実行しても、効果的にデータが分散されないため、期待した結果が得られないおそれがあります。
6. データ総量監視機能を有効にしている場合、EADS サーバ間でのデータ転送のために、一時的にデータ数の上限、および External メモリ使用量の上限を縮小します。その結果、クライアント API、およびユーザファンクションの API でエラーが発生することがあります。  
また、レンジ統計情報ファイル (`eads_store_stats.csv`) の次の値に、一時的に実際の値よりも小さい値が出力されることがあります。
  - KeyCountLimit
  - ExternalHeapSizeLimitまた、リバランス処理の実行中に `eztool storeusage` コマンドを実行すると、上記の値に、一時的に実際の値よりも小さい値が出力されることがあります。
7. データ総量監視機能を有効にしている場合、`eztool import` コマンドで、データ総量監視機能が一時的に無効になります。そのため、`eztool import` コマンドが成功しても、クラスタ内の Java ヒープまたは Explicit ヒープの空き容量が少なくなった場合、リバランス処理で失敗することがあります。この場合は、Java ヒープまたは Explicit ヒープの容量を見直してください。
8. データ総量監視機能を無効にしている場合、クラスタ内の Java ヒープまたは Explicit ヒープの空き容量が少ない状態でリバランス処理を実行すると、EADS サーバがダウンしたり、EADS サーバが縮退したりするおそれがあります。
9. このサブコマンドでエラーが発生した場合は、`eztool status -v` コマンドを実行して、問題が発生していないかを確認してください。



- コマンドの排他が解除されていない場合は、`eztool unlock` コマンドで排他を解除してください。
- クラスタの空き容量が不足している場合は、不要なデータを削除するなどして空き容量を増やすか、またはスケールアウト処理で EADS サーバを追加してください。
- EADS サーバが縮退またはプロセスダウンした場合は、EADS サーバを復旧させてから、リバランス処理を再度実行してください。

EADS サーバの復旧処理でエラーが発生した場合は、復旧に失敗した EADS サーバのクラスタ定義ファイルに、リバランス処理中のクラスタ構成情報が反映されていない可能性があります。復旧に失敗した EADS サーバのクラスタ定義ファイルが、ほかの EADS サーバのクラスタ定義ファイルと同じかどうかを確認してください。クラスタ定義ファイルが、ほかの EADS サーバのクラスタ定義ファイルと異なる場合は、復旧に失敗した EADS サーバに、正常に稼働している EADS サーバのクラスタ定義ファイルをコピーしてから、復旧処理を再度実行してください。

10. データ自動削除機能を有効にしている状態でリバランスを実行すると、EADS サーバ間でのデータ転送のために、一時的にデータ数の上限、および External メモリ使用量の上限を縮小します。その結果、クライアント API、およびユーザファクションの API でエラーが発生することがあります。
11. データ自動削除機能を有効にしている場合、EADS サーバはデータ総量監視機能とは独立して、キャッシュ単位のデータの上限を算出し、一時的にしきい値を縮小します。
12. データ自動削除機能を有効にしている場合、結合元のレンジ（移動するレンジ）に格納されたデータに対する API (`create`, `put`, `putAll`) は、EADS クライアントおよびユーザファクションに、一時的なエラー (`EAD_ERROR_SERVER_LIMIT_KV_COUNT_TEMPORARILY`) を返却することがあります。
13. データ自動削除機能を有効にしているキャッシュでは、キャッシュ定義で指定したデータの自動削除ポリシーに加えて、次に示すデータの自動削除ポリシーにも従います。そのため、次に示すデータの自動削除ポリシーが適用された場合に問題となる用途では、リバランス処理を実行しないでください。

#### ■リバランス処理を実行中に適用されるデータの自動削除ポリシー

結合先のレンジで格納できるデータの空き（キー数および容量）が少ない場合、格納される EADS サーバが変更となるために分割されたレンジのデータは、結合先のレンジで格納できるデータの空きに収まるまで一部または全部が削除されます。また、通信タイミングによっては、優先度が低いデータではなく、優先度が高いデータが削除されることがあります。

14. リバランス処理中のリクエストの応答に一時的に時間が掛かる場合は、次のパラメタの指定を検討してください。
  - `eads.scaling.moveDataWithUpdateRequest.enable` パラメタ
  - `eads.scaling.moveData.interval` パラメタ
  - `eads.scaling.eviction.targetKeyCount` パラメタ（データ自動削除機能を有効にしているキャッシュが存在する場合）
15. データ自動削除機能を有効にしているキャッシュが存在しない場合、次のどちらかの指定をするとリクエストのスループットの向上が期待できますが、リバランス処理の実行完了までの時間が長くなります。運用および環境に合わせたチューニングを実施してください。
  - `eads.scaling.moveDataWithUpdateRequest.enable` パラメタに `false` を指定している

- `eads.scaling.moveData.interval` パラメタに 0 以外を指定している
16. データ自動削除機能を有効にしているキャッシュが存在する場合、  
`eads.scaling.eviction.targetKeyCount` パラメタを指定するとリクエストのスループットの向上が期待できますが、指定値が小さいとリバランス処理の実行完了までの時間が長くなります。運用および環境に合わせたチューニングを実施してください。

## 14.3.25 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>]
               [--format <フォーマット名>]
               [--columns <列名> [, <列名>] ...]
               [--filter <フィルタの条件>]
               [--match <一致条件>]
```

### (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せん。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.command.compaction.effect.division` パラメータに指定した値（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.command.compaction.effect.threshold` パラメータに指定した値（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。

#### (f) --format <フォーマット名>

このオプションの詳細については、「[14.4.2 表示形式の指定方法](#)」を参照してください。

**(g) --columns <列名> [,<列名>] …**

このオプションの詳細については、「[14.4.3 列のフィルタリングの指定方法](#)」を参照してください。

**(h) --filter <フィルタの条件>**

このオプションの詳細については、「[14.4.4 行のフィルタリングの指定方法](#)」を参照してください。

**(i) --match <一致条件>**

このオプションの詳細については、「[14.4.5 条件一致の指定方法](#)」を参照してください。

## **(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。

表示の構成要素については、「[14.4.1 表示内容の構成要素](#)」を参照してください。

■-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.168:24700 true true true 6 0 0 8 0 0 0 0 0
KDEA08002-1 The command will now end.
$

```

サマリとメインの表示内容を、それぞれ次の表に示します。

表 14-49 eztool listecf コマンドのサマリの表示内容

項番	サマリ名	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	サマリ名	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。	○	○	×

(凡例)

○：表示されます。

×：表示されません。

表 14-50 eztool listecf コマンドのメインの表示内容

項番	列名	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	Cache	キャッシュ名※1 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>	○	○	○
2	ExCache	キャッシュが EADS サーバに存在しているかどうか※1 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>	○	○	×
3	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※1 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</li> </ul>	×	○	×

項番	列名	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	
		一致している • false 一致していない			
4	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*1 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。 • true 一致している • false 一致していない	×	○	×
5	Range	レンジ ID*1 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。データの格納先 EADS サーバのサーバ ID と一致します。	○	○	○
6	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	○	○	×
7	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。 • true 存在する • false 存在しない 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。	×	○	×
8	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。 • true 存在する • false 存在しない 次の場合は何も表示されません。 • EADS サーバが起動していない場合	×	○	×

項番	列名	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>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>			
9	ExCD	<p>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。</p> <ul style="list-style-type: none"> <li>true 存在する</li> <li>false 存在しない</li> </ul> <p>次の場合は何も表示されません。</p> <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>	×	○	×
10	UnusedFC	<p>使用していないファイル数</p> <p>次の場合は何も表示されません。</p> <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>	○	○	○
11	MaxCE	<p>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>※2</sup></p> <p>次の場合は何も表示されません。</p> <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>	○	○	○
12	FilterCE(x%) <sup>※3</sup>	<p>コンパクションの効果がい値以上のファイル数<sup>※4</sup></p> <p>次の場合は何も表示されません。</p> <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> </ul>	○	○	○

項番	列名	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>EADS サーバに存在しないキャッシュの場合</li> </ul>			
13	CE(x-y%) <sup>*5</sup>	コンパクションの効果のファイル数の分布 <sup>*4</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。

- Cache の値の自然順序順に表示されます。
- Cache の値が同じ行は、Range の値の昇順に表示されます。
- 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

実行結果に、同じ値のセルが続く場合、2 行目以降のセルの値は表示が省略されます。

注※2

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※3

x には、しきい値として設定した値が表示されます。

注※4

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※5

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

次の表に示すリターンコードが返却されます。



表 14-51 eztool listecf コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	150		コマンド実行中に失敗しました。
8	200		コマンドタイムアウトで失敗しました。

--match オプションを指定した場合

条件に一致した場合は 0、一致しなかった場合は 1 を、リターンコードとして返します。ただし、コマンドの実行に失敗したときは、--match オプションを指定しなかった場合と同じになります。

## (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。

### 14.3.26 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

次の表に示すリターンコードが返却されます。

表 14-52 eztool resume コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (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` パラメータおよび `eads.cache.disk.<n>.dir` パラメータは除きます)。
- 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異常終了やコマンドタイムアウトによって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。そのまま運用を継続するとデータが破壊されたり、消失したりするおそれがあるため、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. キャッシュの再開が失敗したキャッシュを `eztool deletecache` コマンドで削除する。
  5. EADS サーバが縮退しているときは、EADS サーバを復旧する。
  6. キャッシュの再開を、再度実行する。
- 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または共通設定ファイルに指定したパラメー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ー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
- このサブコマンドの実行中、データ総量監視機能は動作しません。

### 14.3.27 exportecf (キャッシュファイルの書き出し)

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のキャッシュファイル（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）のデータを指定したディレクトリに書き出します。

書き出した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は、`eztool importecf` コマンドでキャッシュに読み込むことができます。

## (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 サーバが閉塞状態（closed）のときに実行できます。  
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。
- 書き出し対象として指定したキャッシュが EADS サーバにある必要があります。キャッシュが EADS サーバにあるかどうかは、`eztool listcache` コマンドで確認できます。

## (3) 形式

```
eztool exportecf <キャッシュファイルの出力先パス名>  
                 <キャッシュ名>  
                 [-l]
```

## (4) オプションおよび引数

### (a) <キャッシュファイルの出力先パス名>

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出力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。

## (5) <キャッシュ名>

キャッシュファイルを書き出したいキャッシュ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。

## (6) -l または--local

EADS サーバが停止状態の場合に、キャッシュファイルを書き出すときに指定します。

コマンドを実行する EADS サーバだけが、キャッシュファイルを書き出す対象になります。

### 注意事項

縮退した EADS サーバや正常終了できなかった EADS サーバが出力したキャッシュファイルは、不完全な状態になっているおそれがあります。

## (7) リターンコード

次の表に示すリターンコードが返却されます。

表 14-53 eztool exportecf コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (8) 出力対象

- このサブコマンドを実行すると、各 EADS サーバにある次のキャッシュファイルが<キャッシュファイルの出力先パス名>で指定したディレクトリに出力されます。キャッシュ定義ファイルに指定されたパラメータに基づいて、キャッシュファイルが出力されます。サブディレクトリは作成されません。
- キャッシュデータファイル  
<キャッシュファイルの出力先パス名>/eads\_data\_[EADS サーバ ID\*1]\_[キャッシュ名]\_[レンジ ID\*1]\_[ファイル連番\*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。

- キャッシュインデックスファイル

<キャッシュファイルの出力先パス名>/eads\_index\_[EADS サーバ ID<sup>※1</sup>]<sub>1</sub>\_[キャッシュ名]<sub>1</sub>\_[レンジ ID<sup>※1</sup>]<sub>1</sub>\_[ファイル連番<sup>※2</sup>].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。

## (9) 注意事項

- <キャッシュファイルの出力先パス名>に存在しないディレクトリを指定した場合、ディレクトリが作成されます。
- <キャッシュファイルの出力先パス名>に出力ファイルと同名のファイルがある場合、キャッシュファイルは出力できません。
- 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両方または片方が1ファイルもない場合、コマンドの実行が失敗します。また、キャッシュファイルも出力されません。
- キャッシュの作成、再開、または復旧をした時からキャッシュ定義ファイルの設定を変更しないでください。また、-I オプションを指定してこのサブコマンドを実行する場合は、クラスタ定義ファイルおよび共通設定ファイルも変更しないでください。これらの定義ファイルが変更された場合、不正な内容のキャッシュファイルが出力されるおそれがあります。
- キャッシュファイルの書き出しに失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、キャッシュファイルを書き出してください。
  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。

## 14.3.28 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や、データの多重度やデータサイズを変更する場合は、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 [--convertid <EADSサーバID変換ルール>]  
                <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) --convertid < EADS サーバ ID 変換ルール >

格納先の EADS サーバ ID を指定して key をグルーピングしている場合 (EADS サーバ ID 指定グループを使用している場合) に、指定した EADS サーバ ID を別の EADS サーバ ID に変換して、データを読み込みたいときに指定します。

EADS サーバ ID 変換ルールは、次の形式で指定します。

```
<変換元EADSサーバID>><変換先EADSサーバID>
```

変換元 EADS サーバ ID および変換先 EADS サーバ ID には、1~96 の整数が指定できます (01 や 02 など、10 の位が 0 の整数は指定できません)。

変換元 EADS サーバ ID と変換先 EADS サーバ ID に、同じ値は指定できません。

EADS サーバ ID 変換ルールを複数指定する場合は、コンマ区切りで指定します。EADS サーバ ID 変換ルールを複数指定した場合に、指定順による優先度の差はありません。ただし、同じ変換元 EADS サーバ ID は、複数指定できません。

次に示す文字または文字列は無視されます。



- 先頭または末尾のコンマ  
(例) 「--convertid ,1>2,」
- コンマで区切られた空文字列, またはコンマで区切られたスペースだけの文字列  
(例) 「--convertid 1>2,,」

## (b) <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

次の表に示すリターンコードが返却されます。

表 14-54 eztool importecf コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	10	10	データ自動削除機能を使用している場合、レンジのデータ件数がデータ自動削除機能のしきい値 (キャッシュ定義の <code>eads.cache.eviction.keyCount</code> パラメータの指定値) を超えたため、一部のデータはインポートされませんでした。
3	101	101	コマンドの初期化に失敗しました。
4	110		接続に失敗しました。
5	111		通信タイムアウトで失敗しました。
6	120		構文不正で失敗しました。
7	130		コマンドを実行できない状態のため失敗しました。
8	131		ほかのコマンドが実行中のため失敗しました。
9	150		コマンド実行中に失敗しました。
10	200		コマンドタイムアウトで失敗しました。



## (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。
- データ自動削除機能を使用している場合、読み込み対象の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読み込まないで、`eztool createcache` コマンドでキャッシュを再作成し、必要に応じてデータを再投入することを推奨します。また、読み込み対象の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に読み込むときは、データ自動削除機能のしきい値 (`eads.cache.eviction.keyCount` パラメタの指定値) をキャッシュ使用時に設定してあったしきい値よりも小さい値に変更しないでください。
- 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。
- データ自動削除機能を使用する場合は、レンジごとのしきい値を超えるデータは読み捨てられます。ただし、全レンジでしきい値の上限を超えた場合でも、`put` する `key` のデータがすでに存在する場合に更新時刻が新しいデータがあれば上書きできるように、読み込み処理は継続します。
- EADS サーバ ID 指定グループを含む `key` のデータを登録する際に、指定された EADS サーバ ID のレンジが存在しない場合、そのデータは登録しません。このとき、1 回だけ警告メッセージが出力されません。
- このサブコマンドの実行中、データ総量監視機能は動作しません。
- 永続データの再配置に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順を実施してください。
  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. 永続データの再配置が失敗したキャッシュを `eztool deletetecache --with_deletetecf` コマンドで削除する。

- 5. EADS サーバが縮退しているときは、EADS サーバを復旧する。
- 6. 永続データの再配置を、再度実行する。
- 永続データの再配置に失敗した場合は、次の項目を確認して、永続データの再配置を、再度実行してください。
  - クラスタ定義ファイルまたは共通設定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

## 14.3.29 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*]_[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]_[nnnnn*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]_[nnnnn*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>

注

ディスクキャッシュの格納先として不揮発性メモリを使用している場合  
(`eads.cache.disk.persistentMemory.enable` パラメタに `true` を指定している場合)、キャッシュファ

イルの削除ではなく、`eads.cache.disk.persistentMemory.device` パラメタで指定した不揮発性メモリのデバイスファイルの領域を初期化します。

## (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)

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 サーバが停止状態のときにだけ、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

次の表に示すリターンコードが返却されます。

表 14-55 eztool deleteecf コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (6) 注意事項

- EADS サーバが使用中の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。
- キャッシュファイルの削除に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、キャッシュファイルを削除してください。
  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。

### 14.3.30 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。データの格納先 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データが、サブコマンドを実行した EADS サーバに存在しない場合は、コンパクションできません。

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.command.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

次の表に示すリターンコードが返却されます。

表 14-56 `eztool compaction` コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメタを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメタに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	1	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。

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
3	2	2	未使用ファイル数が <code>--unused_fc</code> 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。
4	3	3	未使用ファイル数が <code>--unused_fc</code> 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。
5	10	10	<code>--break</code> 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。
6	11	11	<code>--break</code> 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。
7	101	101	コマンドの初期化に失敗しました。
8	110		接続に失敗しました。
9	111		通信タイムアウトで失敗しました。
10	120		構文不正で失敗しました。
11	130		コマンドを実行できない状態のため失敗しました。
12	131		ほかのコマンドが実行中のため失敗しました。
13	150		コマンド実行中に失敗しました。
14	200		コマンドタイムアウトで失敗しました。

## (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。
- `eztool compaction --break` コマンドの実行中に、`eztool compaction` コマンドがタイムアウトまたは停止した場合は、`eztool compaction --break` コマンドは正常終了しません。`eztool compaction --break` コマンドがタイムアウトするまで待つか、または `eztool compaction --break` コマンド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、`eztool compaction --break` コマンドを実行しても、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、`eztool compaction --break` コマンドが実行できるまで、繰り返し実行してください。



## 14.3.31 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。

### (5) 注意事項

プロパティファイルのパラメタの指定によっては、スレッドダンプのファイル容量が大きくなる場合があります。事前にテストを実施するなどして、保存先のディスクに十分な空き容量があることを確認してください。



## 14.3.32 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_command_*`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.command.logger.dir` パラメタに指定したディレクトリ直下の `eads_command_*`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.command.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_command_*`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) ※1
  - 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) ※1
  - 不揮発性メモリのデバイスファイル管理領域ダンプ (<運用ディレクトリ>/logs/snapshot\_info/cache\_file\_info/eads\_device\_<キャッシュ名>.txt) ※2

注※1

ディスクキャッシュの格納先として不揮発性メモリを使用しているキャッシュでは収集されません。

## 注※2

ディスクキャッシュの格納先として不揮発性メモリを使用していないキャッシュでは収集されません。また、ディスクキャッシュの格納先として不揮発性メモリを使用しているキャッシュでも、daxio がインストールされていない場合は収集されません。

管理領域にはキャッシュ名が含まれます。EADS サーバに格納されたユーザのデータは含まれません。

- 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\_netinfo.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\_ipinfo.txt)
  - システムリソースの制限情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_ulimit.txt)
  - 不揮発性メモリの構成情報 (<運用ディレクトリ>/logs/snapshot\_info/eads\_info/eads\_info\_pmem.txt) ※

## 注※

ディスクキャッシュの格納先として不揮発性メモリを使用していない場合など、ndctl がインストールされていないときは出力されません。

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  
snapshot --listconf, -s, -t, 60
  - -fd オプションを指定した場合  
snapshot, -v, -t, 60  
snapshot, -s, -v, -t, 60  
snapshot --listconf, -s, -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\*  
snapshot --listconf, -s, -t, 60注※ EADS サーバが縮退状態 (isolated) のときは出力されません。

- このサブコマンドを実行すると、次に示すディレクトリおよびファイルが一時的に作成されます。これらのディレクトリおよびファイルは、書庫ファイルが出力されたあとに削除されます。削除に失敗した場合は、必要に応じて手動で削除してください。
  - <運用ディレクトリ>/logs/直下の EADS サーバのスレッドダンプ  
書庫ファイルを出力したあと、サブコマンドを実行する前から存在するスレッドダンプも削除されます。-sd または--safedump オプション、-fd または--forcedump オプションを指定しない場合も削除されます。書庫ファイルには過去のスレッドダンプも収集されているため、サブコマンド実行後にスレッドダンプが必要になった場合は、書庫ファイルを展開して取得してください。
  - <運用ディレクトリ>/logs/直下に作成される snapshot\_info ディレクトリおよびその直下のファイル
- -sd または--safedump オプション、もしくは-fd または--forcedump オプションを指定した場合、スレッドダンプを 3 回出力して書庫ファイルとしてアーカイブします。プロパティファイルのパラメタの指定によっては、スレッドダンプのファイルサイズが大きくなる場合があります。事前にテストを実施するなどして、保存先のディスクに十分な空き容量があることを確認してください。
- ログと設定ファイルの収集に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、ログと設定ファイルを収集してください。
  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。

### 14.3.33 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

次の表に示すリターンコードが返却されます。

表 14-57 eztool stop コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

### (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。

- EADS サーバの終了に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、EADS サーバを終了させてください。
  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。

クラスタ内の一部の EADS サーバが終了しなかった場合、コマンドの排他が解除できないことがあります。終了していない EADS サーバは `eztool forcestop` コマンドで強制終了してください。

## 14.3.34 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 : kill コマンドの終了コードを返します。

## (5) 注意事項

このサブコマンドは、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ないで 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。

### 14.3.35 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、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

次の表に示すリターンコードが返却されます。



表 14-58 eztool isolate コマンドのリターンコード

項番	リターンコード		リターンコードの説明
	コマンド定義の <code>eads.command.compat</code> パラメータを指定していない場合	コマンド定義の <code>eads.command.compat</code> パラメータに 0300 を指定した場合	
1	0	0	コマンドの実行に成功しました。
2	101	101	コマンドの初期化に失敗しました。
3	110		接続に失敗しました。
4	111		通信タイムアウトで失敗しました。
5	120		構文不正で失敗しました。
6	130		コマンドを実行できない状態のため失敗しました。
7	131		ほかのコマンドが実行中のため失敗しました。
8	150		コマンド実行中に失敗しました。
9	200		コマンドタイムアウトで失敗しました。

## (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。
- EADS サーバの縮退に失敗した場合、コマンドの排他が解除されていないことがあります。次の手順でコマンドの排他を解除してから、EADS サーバを縮退させてください。
  - `eztool status -v` コマンドでクラスタの状態を確認する。
  - `eztool unlock` コマンドで排他を解除する。

## 14.4 実行結果を表示する eztool コマンドのサブコマンドの表示内容

実行結果を表示する eztool のサブコマンドでは、次に示すオプションを指定することで、表示する内容を変更できます。

- --format
- --columns
- --filter
- --match

### 14.4.1 表示内容の構成要素

実行結果を表示する eztool コマンドのサブコマンドの表示内容は、次の要素で構成されています。

- サマリ  
表示される内容全体に共通する情報を表示する個所です。サマリ名とサマリの値から成ります。
- メイン  
実行結果を出力する個所です。ヘッダとボディから成ります。  
また、メインの縦方向に並ぶデータの集合を列、横方向に並ぶデータの集合を行と呼びます。
  - ヘッダ  
実行結果のデータの種類（列名）を表すための行です。
  - ボディ  
実行結果です。
- 空行  
データを区切るための空の行です。
- けい線  
情報を見やすくするために項目を区切る線です。

表示内容中の各データをセルと呼びます。同じ列に同じ内容のセルが続く場合、見やすさを考慮して、2行目以降のセルの値は表示が省略されることがあります。また、データが何もないセルは、ハイフン (-) や空文字列で出力されます。

各要素の構成について、`eztool status -v` コマンドを例に、次の図に示します。

サマリ	Cluster Health: AVAILABLE							
サマリ	TotalCount: 5							
サマリ	OnlineCount: 5							
サマリ	OfflineCount: 0							
サマリ	StandbyCount: 0							
空行								
メイン	ID	IP_Address	ServerName	Port	Position	Cluster	State	...
メイン	1	XX.XXX.XXX.168	server01	24600	1288490189	online	running	...
メイン	2	XX.XXX.XXX.168	server02	24700	429496730	online	running	...
メイン	3	XX.XXX.XXX.168	server03	24800	-429496729	online	running	...
メイン	4	XX.XXX.XXX.168	server04	24900	-1288490188	online	running	...
メイン	5	XX.XXX.XXX.168	server05	25000	-2147483648	online	running	...
けい線								

## 14.4.2 表示形式の指定方法

実行結果を表示する eztool コマンドのサブコマンドに --format オプションを指定することで、表示内容の表示形式を変更できます。

--format オプションを指定しなかった場合、デフォルトの表示形式 (ALIGN フォーマット) が適用されます。ALIGN フォーマットについては、「14.4.2(2)(a) <フォーマット名>」を参照してください。

### (1) オプションの指定形式

```
--format <フォーマット名>
```

### (2) 引数

#### (a) <フォーマット名>

次に示すフォーマット名を指定できます。大文字と小文字は区別されません。

- ALIGN

デフォルトの表示形式です。サマリ、メイン、空行およびけい線が表示されます。同じ列の1つ上のセルと値が同じ場合、2行目以降のセルの値は表示が省略されることがあります。また、データがないセルは、空文字列やハイフン (-) で出力されます。

- CSV

ボディ内のセルだけが、コンマ区切りで表示されます。サマリ、空行およびけい線は表示されません。ALIGN フォーマットと異なり、同じ列の1つ上のセルと値が同じでも、セルの値は省略されません。また、データがないセルは、空文字列で出力されます。

eztool コマンドのサブコマンドの運用を自動化する場合に、このフォーマット名の指定を検討してください。--messageoff オプションと同時に指定することを推奨します。

### (3) 実行例

eztool status -v コマンドの--format オプションに CSV を指定した場合の実行例を次に示します。

```
$ eztool status -v --format csv --messageoff
1,XXX.X.X.1,server01,50101,1288490189,online,initialized,none,unlock,0,0,0,108,04-00-00
2,XXX.X.X.1,server02,50102,429496730,online,initialized,none,unlock,0,0,0,108,04-00-00
3,XXX.X.X.1,server03,50103,-429496729,online,initialized,none,unlock,0,0,0,108,04-00-00
4,XXX.X.X.1,server04,50104,-1288490188,online,initialized,none,unlock,0,0,0,108,04-00-00
5,XXX.X.X.1,server05,50105,-2147483648,online,initialized,none,unlock,0,0,0,108,04-00-00
```

#### 14.4.3 列のフィルタリングの指定方法

実行結果を表示する eztool コマンドのサブコマンドに---columns オプションを指定することで、表示内容の一部の列を選択して表示できます。また、列を表示する順序を変更できます。

--columns オプションを指定しなかった場合、各サブコマンドで決まった表示項目（列）が、決まった順序で表示されます。

次に示す場合は、コマンドの実行に失敗します。

- 同じ列名を 2 回以上指定した場合
- 存在しない列名を指定した場合

#### (1) オプションの指定形式

```
--columns <列名> [, <列名>] ...
```

#### (2) 引数

##### (a) <列名> [, <列名>] ...

実行結果に表示させたい順に、表示内容の列名を指定します。大文字と小文字は区別されます。

--columns オプションは表示のフィルタリングのため、表示されない列の列名は指定できません。例えば、-v オプションを指定した場合にしか表示されない列の列名は、-v オプションを指定していないときは指定できません。

#### (3) 実行例

eztool status -v コマンドで--columns オプションを指定する場合の実行例を次に示します。

```
$ eztool status -v --format csv --messageoff --columns "ID,State"
1,initialized
2,initialized
3,initialized
4,initialized
5,initialized
```

## 14.4.4 行のフィルタリングの指定方法

--filter オプションで条件を指定することで、条件に合う行だけを表示できます。

条件を判定する際、値の表示が省略されるセルは、省略された内容が表示されているものとして扱われ  
ます。また、データが何もないセルは、空文字列として扱われます。

条件の指定方法が誤っている場合は、コマンドの実行に失敗します。

### (1) オプションの指定形式 (BNF 表記)

```
--filter <フィルタの条件>
<フィルタの条件> ::= <行条件>

<行条件> ::= <列条件> | <列条件> <論理演算子> <列条件>
<列条件> ::= <列名> <比較演算子 (文字列)> <文字列>
           | <列名> <比較演算子 (数値)> <数値>

<論理演算子> ::= "&&" | "||"
<比較演算子 (文字列)> ::= "==" | "!="
<比較演算子 (数値)> ::= ">" | "<" | ">=" | "<="

<文字列> ::= Stringの文字列
<数値> ::= Long.parseLong()で変換できる数値
```

#### 注意事項

次に示す場合は、条件全体を二重引用符 (") で囲んでください。

- <列名>または<文字列>に空白が含まれる場合
- <比較演算子 (数値)>を指定する場合

二重引用符 (") で囲まないと、別の引数として扱われたり、不当にリダイレクトされたりするお  
それがあります。

#### 参考

BNF 表記の読み方

： :=の左の項目は、右の項目で示す形式で記述することを意味しています。〈フィルタの条件〉の指定形式を例に説明します。

- 1.「〈フィルタの条件〉 := 〈行条件〉」は、〈フィルタの条件〉を「〈行条件〉」の形式で指定することを示しています。
- 2.「〈行条件〉 := 〈列条件〉 | 〈列条件〉 〈論理演算子〉 〈列条件〉」は、〈行条件〉を「〈列条件〉」または「〈列条件〉 〈論理演算子〉 〈列条件〉」の形式で指定することを示しています。
3. 1.および 2.から、〈フィルタの条件〉は、「〈列条件〉」または「〈列条件〉 〈論理演算子〉 〈列条件〉」の形式で指定するという意味になります。

## (2) 引数

### (a) 〈フィルタの条件〉

〈行条件〉を指定します。

### (b) 〈行条件〉

〈列条件〉、または〈列条件〉 〈論理演算子〉 〈列条件〉を指定します。

### (c) 〈列条件〉

〈列名〉 〈比較演算子 (文字列)〉 〈文字列〉、または〈列名〉 〈比較演算子 (数値)〉 〈数値〉を指定します。

### (d) 〈列名〉

表示内容の列名を指定します。

列名は、--columns オプションで指定しているかどうかに関係なく指定できます。ただし、--columns オプションに関係なく、表示されない列の列名は指定できません。例えば、-v オプションを指定した場合にしか表示されない列の列名は、-v オプションを指定していないときは指定できません。

### (e) 〈論理演算子〉

次の表に示す論理演算子を指定できます。

複数の論理演算子を指定した場合、左から順番に評価されます。

表 14-59 --filter オプションで指定できる論理演算子

項番	論理演算子	指定例	説明
1	&&	A && B	A かつ B (A と B 両方が真のときに真になります。それ以外は偽になります)

項番	論理演算子	指定例	説明
2		A    B	A または B (A と B の両方が偽のときに偽になります。それ以外は真になります)

### (f) <比較演算子 (文字列)>

次の表に示す比較演算子を指定できます。

表 14-60 --filter オプションで指定できる比較演算子 (文字列)

項番	比較演算子	指定例	意味
1	==	A == B	文字列 A が文字列 B のパターンに一致しているかどうか (大文字と小文字は区別されます)
2	!=	A != B	文字列 A が文字列 B のパターンに一致していないかどうか (大文字と小文字は区別されます)

### (g) <比較演算子 (数値)>

次の表に示す比較演算子を指定できます。

表 14-61 --filter オプションで指定できる比較演算子 (数値)

項番	比較演算子	指定例	意味
1	>	A > B	long 値に変換した A が, long 値に変換した B より大きいかどうか (A にデータが何もない場合は, 偽になります)
2	<	A < B	long 値に変換した A が, long 値に変換した B より小さいかどうか (A にデータが何もない場合は偽になります)
3	>=	A >= B	long 値に変換した A が, long 値に変換した B 以上かどうか (A にデータが何もない場合は偽になります)
4	<=	A <= B	long 値に変換した A が, long 値に変換した B 以下かどうか (A にデータが何もない場合は偽になります)

### (h) <文字列>

String の文字列を指定します。ワイルドカード (「\*」 および 「?」) が使用できます。

### (i) <数値>

Long.parseLong() で変換できる数値を指定します。

## (3) 実行例

eztool status -v コマンドで--filter オプションを指定する場合の実行例を次に示します。

■key の総数が 3,000 個以上の EADS サーバの EADS サーバ ID と, その key の総数を表示したいとき

```
$ eztool status -v --format csv --messageoff --filter "KeyCount>=3000" --columns "ID,KeyCount"
5,3034
```

## ■ログに関するパラメタだけを表示したいとき（出力例は省略）

```
$ eztool listconf --messageoff --filter "Parameter==eads.logger.*"
```

## 14.4.5 条件一致の指定方法

--match オプションで条件を指定することで、実行結果が条件に合うかどうかをリターンコードとして返すことができます。指定した条件に合う場合は 0、合わない場合は 1 が返却されます。詳細は、各サブコマンドのリターンコードの説明を参照してください。なお、サブコマンドの結果の表示内容は、このオプションを指定しなかった場合と同じになります。

--filter オプションと同時に指定している場合は、フィルタリング後の結果が条件に合うかどうかリターンコードとして返却されます。ただし、警告の表示などによって表示結果が出力されないときは、1 が返却されます。

条件を判定する際、値の表示が省略されるセルは、省略された内容が表示されているものとして扱われます。また、データが何もないセルは、空文字列として扱われます。

条件の指定方法が誤っている場合は、コマンドの実行に失敗します。

### (1) オプションの指定形式 (BNF 表記)

```
--match <一致条件>
<一致条件> ::= <Boolean関数条件> | <Value関数条件> | <サマリ条件>

<Boolean関数条件> ::= <Boolean関数名> "(" <行条件> ")"
    <Boolean関数名> ::= ALL | EXIST
<Value関数条件> ::= <Value関数名> "(" <行条件> ")" <比較演算子 (文字列)> <文字列>
    | <Value関数名> "(" <行条件> ")" <比較演算子 (数値)> <数値>
    <Value関数名> ::= COUNT
<サマリ条件> ::= <サマリ名> <比較演算子 (文字列)> <文字列>
    | <サマリ名> <比較演算子 (数値)> <数値>

<行条件> ::= <列条件> | <列条件> <論理演算子> <列条件>
<列条件> ::= <列名> <比較演算子 (文字列)> <文字列>
    | <列名> <比較演算子 (数値)> <数値>

<比較演算子 (文字列)> ::= "=" | "!="
<比較演算子 (数値)> ::= ">" | "<" | ">=" | "<="
<論理演算子> ::= "&&" | "||"

<文字列> ::= Stringの文字列
<数値> ::= Long.parseLong()で変換できる数値
```



## 注意事項

次に示す場合は、条件全体を二重引用符 (") で囲んでください。

- ・ <サマリ名>、<列名>、または<文字列>に空白が含まれる場合
- ・ <比較演算子 (数値)>を指定する場合

二重引用符 (") で囲まないと、別の引数として扱われたり、不当にリダイレクトされたりするおそれがあります。

## 参考

BNF 表記の読み方については、「14.4.4(1) オプションの指定形式 (BNF 表記)」を参照してください。

## (2) 引数

### (a) <一致条件>

< Boolean 関数条件>、< Value 関数条件>、または<サマリ条件>を指定します。

### (b) < Boolean 関数条件>

< Boolean 関数名>"(<行条件>)"を指定します。

### (c) < Boolean 関数名>

次の表に示す関数を指定できます。

表 14-62 --match オプションで指定できる Boolean 関数

項番	関数	意味
1	ALL(<行条件>)	すべての行が行条件に合うかどうか
2	EXIST(<行条件>)	行条件に合う行が 1 つ以上あるかどうか

### (d) < Value 関数条件>

< Value 関数名>"(<行条件>)"<比較演算子 (文字列)><文字列>、または< Value 関数名>"(<行条件>)"<比較演算子 (数値)><数値>を指定します。

### (e) < Value 関数名>

次の表に示す Value 関数を指定できます。

表 14-63 --match オプションで指定できる Value 関数

項番	関数	意味
1	COUNT(<行条件>)	行条件に合う行の数を返す

### (f) <サマリ条件>

<サマリ名><比較演算子 (文字列)><文字列>, または<サマリ名><比較演算子 (数値)><数値>を指定します。

### (g) <サマリ名>

表示内容のサマリ名を指定します。

サマリ名は、--format オプションおよび--column オプションの指定に関係なく指定できます。

--format オプションおよび--column オプション以外のオプションの指定有無によって表示されないサマリ名は指定できません。例えば、-v オプションを指定した場合にしか表示されないサマリ名は、-v オプションを指定していないときは指定できません。

### (h) <行条件>

<列条件>, または<列条件><論理演算子><列条件>を指定します。

### (i) <列条件>

<列名><比較演算子 (文字列)><文字列>, または<列名><比較演算子 (数値)><数値>を指定します。

### (j) <列名>

表示内容の列名を指定します。

列名は、--format オプション、--columns オプションの指定に関係なく指定できます。ただし、--format オプションおよび--column オプション以外のオプションの指定有無によって表示されない列の列名は指定できません。例えば、-v オプションを指定した場合にしか表示されない列の列名は、-v オプションを指定していないときは指定できません。

### (k) <比較演算子 (文字列) >

次の表に示す比較演算子を指定できます。

表 14-64 --match オプションで指定できる比較演算子 (文字列)

項番	比較演算子	指定例	意味
1	==	A == B	文字列 A が文字列 B のパターンに一致しているかどうか (大文字と小文字は区別されます)

項番	比較演算子	指定例	意味
2	!=	A != B	文字列 A が文字列 B のパターンに一致していないかどうか (大文字と小文字は区別されます)

### (l) <比較演算子 (数値)>

次の表に示す比較演算子を指定できます。

表 14-65 --match オプションで指定できる比較演算子 (数値)

項番	比較演算子	指定例	意味
1	>	A > B	long 値に変換した A が, long 値に変換した B より大きいかどうか (A にデータが何もない場合は, 偽になります)
2	<	A < B	long 値に変換した A が, long 値に変換した B より小さいかどうか (A にデータが何もない場合は, 偽になります)
3	>=	A >= B	long 値に変換した A が, long 値に変換した B 以上かどうか (A にデータが何もない場合は, 偽になります)
4	<=	A <= B	long 値に変換した A が, long 値に変換した B 以下かどうか (A にデータが何もない場合は, 偽になります)

### (m) <論理演算子>

次の表に示す論理演算子を指定できます。

複数の論理演算子を指定した場合, 左から順番に評価されます。

表 14-66 --match オプションで指定できる論理演算子

項番	論理演算子	指定例	説明
1	&&	A && B	A かつ B (A と B 両方が真のときに真になります。それ以外は偽になります)
2		A    B	A または B (A と B の両方が偽のときに偽になります。それ以外は真になります)

### (n) <文字列>

String の文字列を指定します。ワイルドカード (「\*」および「?」) が使用できます。

### (o) <数値>

Long.parseLong() で変換できる数値を指定します。

## (3) 実行例

eztool status -v コマンドで--match オプションを指定する場合の実行例を次に示します (出力例は省略します)。

■クラスタの起動が完了したかどうかを確認したいとき

```
$ eztool status -v --match "ALL(State==initialized)"
```

■Explicit ヒープの使用率が 70%以上の EADS サーバが存在するかどうかを確認したいとき

```
$ eztool status -v --match "EXIST(UsedMemoryRatio>=70)"
```

■指定したキャッシュ名のキャッシュが存在するかどうかを確認したいとき

```
$ eztool listcache --match "EXIST(CacheName==cache)"
```

■EADS サーバの復旧が完了したかどうかを確認したいとき

```
$ eztool status -s -v --match "ALL(State==initialized || State==running || State==closed)"
```

■キャッシュファイルの数を監視したいとき

```
$ eztool listecf -s --match "EXIST(UnusedFC<=3)"
```

# 15

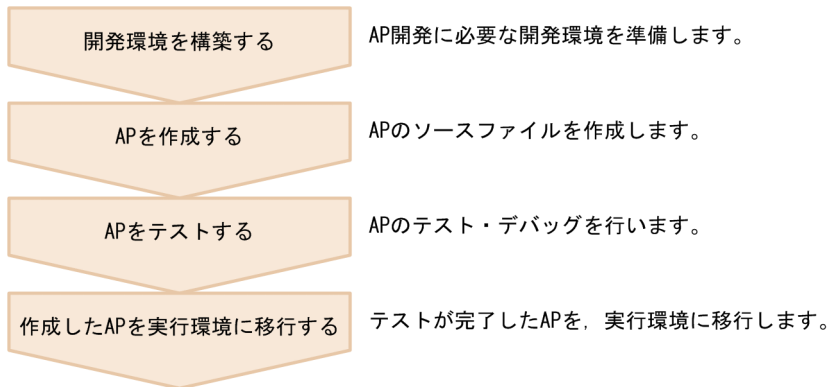
## AP 開発の流れ

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
  - Hitachi Elastic Application Data Store
- C 言語の場合
  - Hitachi Elastic Application Data Store
  - Hitachi 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）」](#)

### 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、「[16. 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、「[17. ユーザファンクションを作成する](#)」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、「[19. 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、グループ名、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  
Hitachi Elastic Application Data Store Client for Java
- C 言語の場合  
Hitachi 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, グループ名, 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, グループ名, 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 の形式と指定できるデータについて説明します。

#### (a) key の形式

```
key=  
[<グループ名>:] <要素名>
```

key をグルーピングしない場合、<要素名>が key になります。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>を合わせたものが key になります。このため、<要素名>が同じでも<グループ名>が異なれば別の key となります。

<グループ名>の形式については、「15.2.2(2) グループ名として指定できるデータ」を参照してください。

key の指定例を次に示します。

- グルーピングしていない key

```
key1
```

- グルーピングした key

```
group1:key1
```

- グループを階層化した key

```
group1:group2:key1
```

#### (b) key として指定できるデータ型と文字数

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。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。

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	null または空文字列は不正となります。
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>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。

## (c) key を指定する際の規則および注意事項

key を指定する際の規則および注意事項を次に示します。

- key には、ASCII コード「0x20」～「0x7E」の文字が指定できます。  
ただし、次の表に示す文字は EADS の予約文字のため、一部の例外を除いて key には使用できません。

表 15-2 key に使用できない文字

ASCII コード	文字	例外
0x22	二重引用符 (")	なし
0x28	左丸括弧 (()	なし
0x29	右丸括弧 ())	なし
0x3a	コロン (:)	グループ階層名および要素名の区切りにだけ使用できます。グループ階層名および要素名には指定できません。
0x3c	小なり演算子 (<)	なし
0x3e	大なり演算子 (>)	なし
0x5b	左角括弧 ([)	EADS サーバ ID 指定グループで、格納先の EADS サーバの EADS サーバ ID を指定するときだけに使用できます。
0x5d	右角括弧 (])	
0x7c	縦線 ( )	なし

## (2) グループ名として指定できるデータ

グループ名の形式と指定できるデータについて説明します。

## (a) グループ名の形式

```
<グループ名>=  
<グループ階層名>[:<グループ階層名>]…
```

key の最上位階層のグループ階層名から、任意の階層のグループ階層名までをグループ名として指定できます。中間のグループ階層名から始まる名称は指定できません。

API などグループ名を指定する場合、中間のグループ階層名から始まる名称は指定できません。例えば、key が「groupA:groupB:groupC:key」の場合に、グループ名として「groupB:groupC」は指定できません。

key とグループ名の関係の例を次に示します。

- key が「groupA:groupB:groupC:key」の場合に指定できるグループ名

```
groupA
```

```
groupA:groupB
```

```
groupA:groupB:groupC
```

- key が「[10]group1:group2:key」の場合に指定できるグループ名

```
[10]group1
```

```
[10]group1:group2
```

## (b) EADS サーバ ID 指定グループ

key の格納先の EADS サーバを指定して key をグルーピングする場合は、格納先の EADS サーバの EADS サーバ ID を指定します。このようなグループを、EADS サーバ ID 指定グループといいます。

EADS サーバ ID 指定グループを含む key の形式を次に示します。

```
[<EADSサーバID>]<グループ階層名>[:<グループ階層名>:]…<要素名>
```

EADS サーバ ID 指定グループを使用して key をグルーピングする場合、key の先頭のグループ階層名に、左角括弧 ( [ ) および右角括弧 ( ] ) で囲んだ EADS サーバ ID を指定します（形式中の「[< EADS サーバ ID >]」の左角括弧 ( [ ) および右角括弧 ( ] ) は、実際に指定することを表しています。省略の意味ではありません）。EADS サーバ ID には、先頭が 0 でない整数を指定します。

なお、この場合、左角括弧 ( [ )、EADS サーバ ID、および右角括弧 ( ] ) も含めて、グループ名として扱われます。

EADS サーバ ID 指定グループを含む key の指定例を次に示します。

```
[1]group1:key1
```

### (c) グループ名として指定できるデータ型と文字数

グループ名として指定できるデータ型については、「15.2.2(1)(b) keyとして指定できるデータ型と文字数」を参照してください。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。そのため、グループ名として指定できる文字数は1,022文字までとなります。

### (d) グループ名を指定する際の規則および注意事項

グループ名を指定する際の規則および注意事項を次に示します。これ以外の規則および注意事項については、「15.2.2(1)(c) keyを指定する際の規則および注意事項」を参照してください。

- 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。ただし、グループを階層化した場合、操作する階層が深くなるにつれて処理に時間が掛かります。そのため、適切な階層を設定してください。
- EADS サーバ ID 指定グループを使用する場合、クラスタに存在しない EADS サーバの EADS サーバ ID は指定できません。
- EADS サーバ ID 指定グループを含む key のハッシュ値は、EADS サーバのレンジ上の値です。リバランス処理によって EADS サーバのレンジが変わると、ハッシュ値も変わることがあります。

## (3) 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 15-3 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。

## (4) 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 15-4 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><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>

## (5) 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 15-5 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

## 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。

## 16.1 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。

### 16.1.1 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);
```



```

// キャッシュへのアクセスを開始する
cache = cacheManager.getCache(CACHENAME);
System.out.println("cache start succeeded. (cache name = " + CACHENAME + ")");

// keyとvalueを格納する
final String KEY = "key1";
final String VALUE = "value1";

cache.put(KEY, VALUE);
System.out.println("PUT succeeded. (key = " + KEY + ", value = " + VALUE + ")");

String value = (String)cache.get(KEY);
System.out.println("GET succeeded. (key = " + KEY + ", value = " + value + ")");

} catch (CacheException e) {
    System.out.println("cache operation failed. (cache name = " + CACHENAME + ", error code = " + e.getErrorCode() + ")");
}finally{
    if(cacheManager != null){
        // キャッシュへのアクセスを終了する
        if(cache != null){
            try{
                cacheManager.removeCache(CACHENAME);
                System.out.println("cache stop succeeded. (cache name = " + CACHENAME + ")");
            }catch(CacheException e){
                System.out.println("CacheManager.removeCache() failed. (error code = " + e.getErrorCode() + ")");
            }
        }
        // EADSクライアントの使用を終了する
        try{
            cacheManager.destroy();
        }catch(CacheException e){
            System.out.println("CacheManager.destroy() failed. (error code = " + e.getErrorCode() + ")");
        }
    }
}
}
}
}

```

## (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.client.api.*;
```

## (3) 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、CacheManager クラスの create() を使用して、CacheManager クラスのインスタンスを生成します。

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 `key` またはグループ名、もしくは `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。

## 16.2 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。

### 16.2.1 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。
- Java EE 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。
- Java EE 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。

### 16.2.2 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。

## 16.2.3 データ操作に関する留意事項

### (1) 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 が返却されません。
- 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 が返却されません。
- 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 が返却されません。

### (2) データの一括操作に関する留意事項

データの一括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、putAll()は無条件に value を更新します。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

- 複数の key を指定する一括操作の実行時に、ある key に対する操作に失敗した場合やクラスタ構成が変更された場合は、一括操作の処理が終了し、その時点で未実行の操作がキャンセルされます。一括操作中に、ユーザの操作（例：スケールアウト処理、リバランス処理、または復旧処理）によってクラスタ構成が変更された場合も、未実行の操作がキャンセルされます。

操作に失敗した key および失敗した要因については、各 API の戻り値から判断してください。また、操作がキャンセルされた key については、

CacheException.EAD\_ERROR\_CLIENT\_BATCH\_CANCEL のエラーコードで確認できます。

- 複数の key を指定する一括操作の実行時に、key に対する操作がすべて失敗した場合は、CacheException.EAD\_ERROR\_BATCH\_FAILED\_ALL が返却されます。
- 複数の key を指定する一括操作の実行時に、key に対する操作が部分的に失敗した場合は、CacheException.EAD\_ERROR\_BATCH\_FAILED\_PART が返却されます。

## 16.2.4 イベントリスナに関する留意事項

イベントリスナに関する留意事項を次に示します。

### (1) イベントリスナの実装に関する留意事項

- イベントが発生した場合、クラスタイベントリスナおよびノードイベントリスナは、EADS クライアントに追加した順序（addClusterEventListener()および addNodeEventListener()で追加した順序）で実行されます。
- イベントリスナ内では、CacheManager クラスの次のメソッドを呼び出さないでください。
  - create()
  - destroy()
- イベントリスナの toString()はメッセージログに使用します。オーバーライドする場合は、次の点に注意してください。
  - インスタンスを識別できる文字列を返却すること。
  - 改行を含まない文字列を返却すること。
- ClusterEventContext と、そこから取得できるオブジェクトの有効範囲は、ClusterEventListener 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、動作は保証されません。
- NodeEventContext と、そこから取得できるオブジェクトの有効範囲は、NodeEventListener 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、動作は保証されません。
- イベントリスナ内には、EADS クライアントの正常な処理に影響を及ぼす処理（例えば、リソースを大量に消費する処理）を実装しないでください。



- 同一 EADS クライアント内のすべてのイベントリスナは、1つのスレッドから呼び出されて動作します。イベントリスナの動作中は、EADS サーバから受信したデータの保持用にメモリを占有します。そのため、イベントリスナ内には、長時間動作する処理を実装しないでください。

## (2) イベントの通知に関する留意事項

- EADS サーバが初期化状態 (initialized) 以降の状態の場合に、EADS サーバから EADS クライアントにイベントが通知されます。ただし、EADS サーバとの接続の再接続に成功した場合は、EADS サーバの状態に関係なく、その EADS サーバがまだ通知していないイベントが通知されます。
- EADS クライアントがクラスタ内のどの EADS サーバとも通信できなかった場合は、イベントが通知されません。ただし、EADS クライアントは EADS サーバとの接続の再接続を繰り返すため、再接続に成功すると、そのあとでイベントが通知されます。
- 半数以上の EADS サーバが縮退している場合、イベントが通知されないことがあります。
- EADS クライアントと EADS サーバ間の接続で、切断と再接続を繰り返した場合や、通信が遅延している場合には、イベントの通知が遅延します。
- EADS クライアントと EADS サーバ間の接続が切断している間に、複数の EADS サーバが縮退した場合、EADS サーバのイベント再送時間 (サーバ定義の `eads.server.eventListener.notify.retry.timeout` パラメタの指定値) を過ぎて再送用のデータが破棄されると、イベントを通知する順序が逆転するおそれがあります。
- EADS クライアントと EADS サーバ間の接続が切断している間に、ある EADS サーバが縮退後に復旧した場合、EADS サーバのイベント再送時間 (サーバ定義の `eads.server.eventListener.notify.retry.timeout` パラメタの指定値) を過ぎて再送用のデータが破棄されると、何もイベントが通知されないおそれがあります。
- 同じプロセス内で、EADS クライアントを終了 (CacheManager クラスの `destroy()`) したあとに EADS クライアントを開始 (CacheManager クラスの `create()`) した場合は、すでに終了した EADS クライアントの再送用のデータは破棄されています。そのため、EADS クライアントを開始したあとのイベントだけが通知されます。
- 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。そのため、EADS クライアントへのイベントの通知が完了していなくても、クラスタの終了 (`eztool stop`) や EADS サーバの縮退などの処理は完了します。

## 16.2.5 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。

## 16.2.6 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。
- Java EE 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。



## 16.3 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。

javaclient ディレクトリを/opt/hitachi/xeads ディレクトリから別のディレクトリにコピーしている場合は、コピー先のディレクトリのファイルを指定してください。

- /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、Java EE サーバ (uCosminexus Application Server) 上で実行する場合は、上記のクラスライブラリを 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。javac コマンドは、Servlet または JSP から実行できません (Filter, Listener を含みます)。

# 17

## 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。

## 17.1 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。

### 17.1.1 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。

### 17.1.2 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。

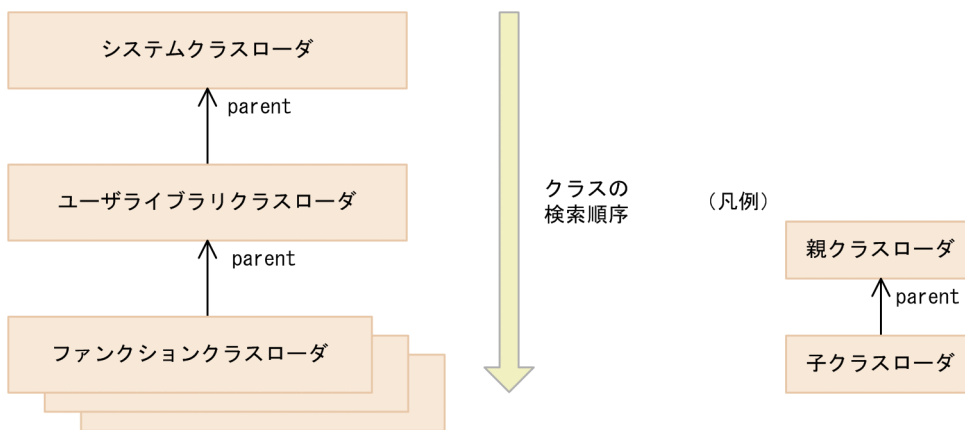
- key またはグループ指定による実行方法
- EADS サーバ指定による実行方法

それぞれのユーザファンクションの仕組みについては、「[2.7.1 ユーザファンクションの仕組み](#)」を参照してください。

### 17.1.3 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ファンクション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  
ユーザファンクションの 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（ユーザファンクション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、ユーザファンクションの 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。

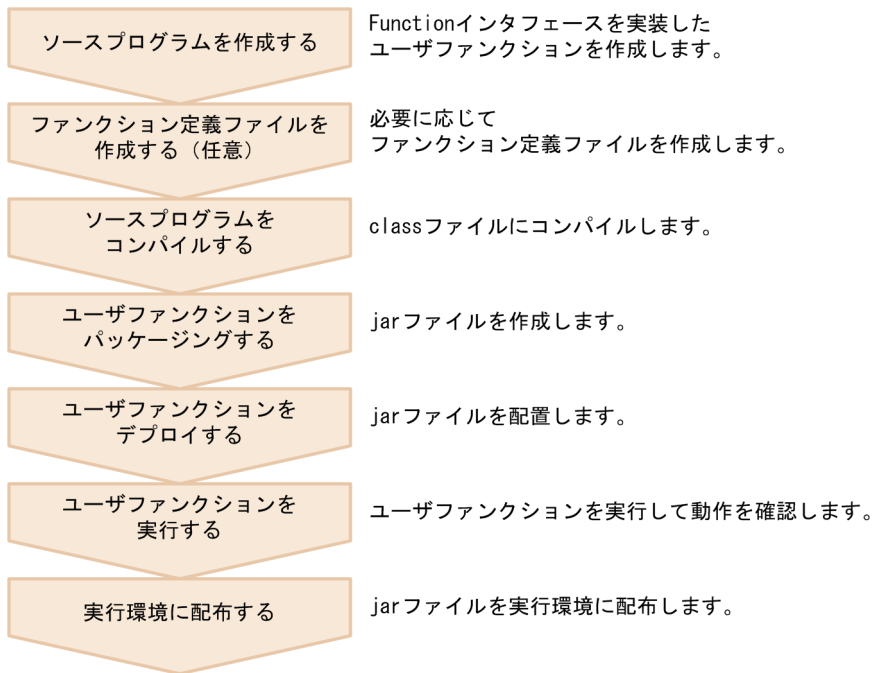
各クラスローダのクラス検索は、親クラスローダであるシステムクラスローダから行います。

ユーザファンクションのロード順序は、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 インタフェースの `init()` は実行されません。

## 17.1.4 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。



## 17.2 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。

### 17.2.1 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) 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;
```

#### (2) ユーザファン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、ユーザファン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、このユーザファンクションは使用できません。

#### (3) ユーザファン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ファン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。

ユーザファンクションで実行できる操作については、「[18.2 ユーザファンクションで使用できる API](#)」を参照してください。

#### (4) ユーザファン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、ユーザファン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、ユーザファン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。

## 17.3 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。

### 17.3.1 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。

### 17.3.2 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。

### 17.3.3 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のリクエスト 1 回当たりの実行時間が短くなるよう設計してください。

次のコマンドを実行した際に、ユーザファンクションの実行時間の分だけ、「各処理が完了するまでの時間」および「各処理中に実行したリクエストが完了するまでの時間」が長くなるおそれがあります。

- 次のオプションを指定した `ezstart` コマンド
  - r または `--recovery`
  - ai または `--add_id`
  - ap または `--add_position`
- 次のオプションを指定した `ezserver` コマンド
  - r または `--recovery`
  - ai または `--add_id`
  - ap または `--add_position`
- `eztool close` コマンド
- `eztool isolate` コマンド
- `eztool rebalance` コマンド
- 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。
- 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。
- C 言語のクライアント AP にユーザファンクションの引数を指定する場合、ユーザファンクションで扱うことができるオブジェクトは、バイト配列だけです。

### 17.3.4 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 `keyIterator()` (Key インタフェース指定) 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 keyIterator() (Keyインタフェース指定) 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、「18.2.9 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 descendingKeyIterator() (Key インタフェース指定) 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。

## 17.4 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>.maxExecuteThreads` 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。

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

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。

## 17.5 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。

### 17.5.1 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
```

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
```

## 17.6 ユーザファンクションをパッケージングする

---

[17.5 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。

### 17.6.1 ユーザファンクションをパッケージングする手順

[17.5 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。

#### (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
```

## 17.7 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。

### 17.7.1 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。

#### (2) 作成した jar ファイルをコピーする

「17.6.1(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

#### (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

#### (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

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。

#### (6) デプロイされたかどうかを確認する (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を確認します。

手順については、「11.11 ユーザ機能が各 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。



## 17.8 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。

### 17.8.1 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) key または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()` (key またはグループ指定)

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()`

### 17.8.2 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。

### 17.8.3 ユーザファンクション運用上の留意事項

- ユーザファンクションは稼働状態の EADS サーバでだけ実行できます。ユーザファンクション実行中に EADS サーバの状態が変更された場合、それ以降のユーザファンクションによるデータ操作はエラーとなります。
- `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機能の整合性は、ユーザが確保する必要があります。特定の EADS サーバにユーザ機能が存在しない状態で、ユーザ機能を実行しようとすると、EADS クライアントにはエラーが返されます。
- ユーザ機能の実装が EADS サーバごとに異なっても、EADS サーバはユーザ機能を実行します。

## 17.9 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。

# 18

## 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。

## 18.1 Java 言語のクライアントライブラリが提供する API

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 API を提供します。これらの API 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 API はすべてスレッドセーフです。

表 18-1 Java 言語のクライアントライブラリで使用できる 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	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	FailureOperationInfo	一括操作を試みた結果、一部またはすべての操作に失敗した場合に、失敗した操作の情報を格納するためのクラスです。	
5	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。	
6	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。	
7	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。	
8	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。	
9	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。	
10	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	

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
		CacheException のサブクラスです。	
11	BatchOperationException	一括操作を試みた結果, 一部またはすべての操作に失敗した場合に返却される CacheException のサブクラスです。	
12	AllFailureException	一括操作を試みた結果, すべての操作に失敗した場合に返却される BatchOperationException のサブクラスです。	
13	PartFailureException	一括操作を試みた結果, 一部の操作に失敗した場合に返却される BatchOperationException のサブクラスです。	
14	ClusterEventListener	クラスタイベントリスナが実装する必要があるインタフェースです。	
15	NodeEventListener	ノードイベントリスナが実装する必要があるインタフェースです。	
16	ClusterEventContext	クラスタイベントリスナの実行に必要な情報を ClusterEventListener インタフェースに渡すインタフェースです。	
17	NodeEventContext	ノードイベントリスナの実行に必要な情報を NodeEventListener インタフェースに渡すインタフェースです。	
18	列挙型 ClusterEventType	クラスタイベントリスナの実行の契機を表す列挙型です。	
19	列挙型 NodeEventType	ノードイベントリスナの実行の契機を表す列挙型です。	

## 18.1.1 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
<code>getName()</code>	Cache クラスのインスタンスに関連づけられたキャッシュ名を取得します。
<code>put()</code>	value を key に関連づけて格納します。
<code>putAll()</code>	指定した Map 内で関連づけられた key と value を、キャッシュに一括格納します。
<code>create()</code>	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。
<code>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。
<code>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 (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。
<code>get()</code>	指定した key に関連づけられた value を取得します。
<code>getAll()</code> (Set 指定)	指定した key の一覧に関連づけられている value を一括取得します。取得した value は、key と関連づけられて、引数に指定した Map に格納されます。
<code>getAll()</code> (グループ指定)	指定したグループ、およびその下位階層のグループに属する key に関連づけられている value を、一括取得します。取得した value は、key と関連づけられて、引数に指定した Map に格納されます。
<code>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。
<code>removeAll()</code> (Set 指定)	指定した key の一覧に関連づけられている value を一括削除します。
<code>removeAll()</code> (グループ指定)	指定したグループに属する key および value を、下位階層のグループに属する key および value も含めて一括削除します。
<code>removeAll()</code> (EADS サーバ指定)	指定した EADS サーバをコピー元 EADS サーバとする key および value を一括削除します。
<code>getGroupNameSet()</code>	指定した EADS サーバに格納されている最上位階層のグループ名の一覧を取得します。グループ名の一覧は、ASCII コードの昇順になります。
<code>getKeySet()</code> (グループ指定)	指定したグループに属する key の一覧を取得します。key の一覧には、指定したグループの下位階層のグループに属する key も含まれます。

メソッド名	説明
	key の一覧は、ASCII コードの昇順になります。
getKeySet() (EADS サーバ指定)	指定した EADS サーバに格納されている key の一覧を取得します。 key の一覧は、ASCII コードの昇順になります。
getGroupCount()	指定した EADS サーバに格納されている最上位階層のグループ数を取得します。
getKeyCount() (グループ指定)	指定したグループに属する key の数を取得します。指定したグループの下位階層のグループに属する key も、取得される key の数に含まれます。
getKeyCount() (EADS サーバ指定)	指定した EADS サーバに格納されている key の数を取得します。
getFirstKey() (グループ指定)	指定したグループに属する key のうち、ASCII コードの昇順で先頭の key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。
getFirstKey() (EADS サーバ指定)	指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で先頭となる key を取得します。
getNextKey() (グループおよび key 指定)	指定したグループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。
getNextKey() (EADS サーバおよび key 指定)	指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で指定した key の次に位置する key を取得します。
executeFunction() (key またはグループ指定)	指定した key またはグループから実行する EADS サーバを決定して、ユーザファンクションを実行します。
executeFunction() (key またはグループ、および受信タイムアウト指定)	指定した key またはグループから実行する EADS サーバを決定して、ユーザファンクションを実行します。また、受信タイムアウト時間を指定します。
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) getName()

### (a) 機能

Cache クラスのインスタンスに関連づけられたキャッシュ名を取得します。

CacheManager クラスの removeCache() による、キャッシュへのアクセス終了後にも使用できます。

### (b) 形式

```
public String getName()
```



## (c) 戻り値

Cache クラスのインスタンスに関連づけられたキャッシュ名が返却されます。

## (6) 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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの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\(3\) 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) putAll()

### (a) 機能

指定した Map 内で関連づけられた key と value を、キャッシュに一括格納します。

value の格納で問題が発生した場合は、例外が返却されます。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってしきい値を超える件数分のデータを削除してから、複数の key と value を一括格納します。既存のデータを更新する場合もデータは削除されます。データの削除に失敗した場合、失敗した件数分は value の格納を実行しません。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しなかった key は、データの削除に失敗したときの例外情報が設定されます。

### (b) 形式

```
public void putAll(java.util.Map<String,? extends Object> map)
                throws CacheException
```

### (c) パラメータ

map

一括格納する key および value を関連づけた Map を指定します。

null または要素数が 0 の Map を指定した場合はエラーとなります。

キャッシュに格納できる key 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。

キャッシュに格納できる value については、「[15.2.2\(3\) 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)
- AllFailureException (一括操作全体の失敗)
- PartFailureException (一括操作の部分的な失敗)
- InternalClientException (EADS クライアントの内部エラー)

### (e) 注意事項

- value の格納にすべて失敗した場合、例外として AllFailureException、およびエラーコードとして CacheException.EAD\_ERROR\_BATCH\_FAILED\_ALL が返却されます。  
value の格納が部分的に失敗した場合、例外として PartFailureException、およびエラーコードとして CacheException.EAD\_ERROR\_BATCH\_FAILED\_PART が返却されます。

それぞれの例外から情報を取得することで、どの key の操作が何の要因で失敗したかを判別できます。

- キャッシュの操作が部分的に失敗した場合、例外からどの操作が失敗したかを判別して、必要に応じてこのメソッドを再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## (8) 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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。データの削除は、指定した key がキャッシュに格納されている場合も実行します。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの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\(3\) 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)

## (9) 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\(3\) 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)

## (10) 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\(3\) 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\(3\) 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)

## (11) 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)
- `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

`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。

## (12) `getAll()` (Set 指定)

#### (a) 機能

指定した key の一覧に関連づけられている value を一括取得します。取得した value は、key と関連づけられて、引数に指定した Map に格納されます。

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 void getAll(java.util.Set<String> keys,  
                  java.util.Map<String, Object> returnMap)  
    throws CacheException
```

#### (c) パラメータ

keys

取得する value に関連づけられている key の一覧を指定します。

null または要素数が 0 の key の一覧を指定した場合はエラーとなります。

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。

returnMap

取得した value を格納する Map を指定します。

null を指定した場合は不正となります。

## (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- AllFailureException (一括操作全体の失敗)
- PartFailureException (一括操作の部分的な失敗)
- InternalClientException (EADS クライアントの内部エラー)

## (e) 注意事項

- value の取得がすべて失敗した場合、例外として AllFailureException, およびエラーコードとして CacheException.EAD\_ERROR\_BATCH\_FAILED\_ALL が返却されます。  
value の取得が部分的に失敗した場合、例外として PartFailureException, およびエラーコードとして CacheException.EAD\_ERROR\_BATCH\_FAILED\_PART が返却されます。  
それぞれの例外から情報を取得することで、どの key の操作が何の要因で失敗したかを判別できます。
- 引数に指定した Map に、取得対象の key と value がすでに格納されている場合は、key と value は取得した値で上書きされます。
- キャッシュに value が格納されていない key を指定した場合、引数に指定した Map には何も格納されません。
- キャッシュの操作が部分的に失敗した場合、例外からどの操作が失敗したかを判別して、必要に応じてこのメソッドを再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## (13) getAll() (グループ指定)

### (a) 機能

指定したグループ、およびその下位階層のグループに属する key に関連づけられている value を、一括取得します。取得した value は、key と関連づけられて、引数に指定した Map に格納されます。

value の一括取得中に何らかの要因で value の取得に失敗した場合は、成功した分の value だけが Map に格納されます。また、失敗の要因を示す例外が返却されます。

## (b) 形式

```
public void getAll(String groupName,  
                  java.util.Map<String, Object> returnMap)  
    throws CacheException
```

## (c) パラメータ

groupName

value を取得する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\(2\) グループ名として指定できるデータ](#)」を参照してください。

returnMap

取得した value を格納する Map を指定します。

null を指定した場合はエラーとなります。

## (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)

## (e) 注意事項

- 引数に指定した Map に、取得対象の key と value がすでに格納されている場合は、key と value は取得した値で上書きされます。
- キャッシュに value が格納されていない key を指定した場合、引数に指定した Map には何も格納されません。
- キャッシュの操作が部分的に失敗した場合、キャッシュの操作の実行結果を確認して、必要に応じてこのメソッドを再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。



## (14) 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)
- 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)

## (15) removeAll() (Set 指定)

### (a) 機能

指定した key の一覧に関連づけられている value を一括削除します。

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 removeAll(java.util.Set<String> keys)
           throws CacheException
```

### (c) パラメータ

keys

削除する value に関連づけられている key の一覧を指定します。

null または要素数が 0 の key の一覧を指定した場合はエラーとなります。

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)
- `AllFailureException` (一括操作全体の失敗)
- `PartFailureException` (一括操作の部分的な失敗)
- `InternalClientException` (EADS クライアントの内部エラー)

## (e) 注意事項

- `value` の削除がすべて失敗した場合、例外として `AllFailureException`、およびエラーコードとして `CacheException.EAD_ERROR_BATCH_FAILED_ALL` が返却されます。  
`value` の削除が部分的に失敗した場合、例外として `PartFailureException`、およびエラーコードとして `CacheException.EAD_ERROR_BATCH_FAILED_PART` が返却されます。  
それぞれの例外から情報を取得することで、どの `key` の操作が何の要因で失敗したかを判別できます。
- 例外が返却された場合、削除対象のデータが残っているおそれがあります。そのため、例外からどの操作が失敗したかを判別して対処してください。必要に応じて、`removeAll()` (Set 指定) を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の `value` が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## (16) removeAll() (グループ指定)

### (a) 機能

指定したグループに属する `key` および `value` を、下位階層のグループに属する `key` および `value` も含めて一括削除します。

`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 removeAll(String groupName)
    throws CacheException
```

## (c) パラメータ

groupName

削除する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(2) グループ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)
- 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)

## (e) 注意事項

- 例外が返却された場合、削除対象のデータが残っているおそれがあります。そのため、実行結果を確認して対処してください。必要に応じて、removeAll() (グループ指定) を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## (17) removeAll() (EADS サーバ指定)

### (a) 機能

指定した EADS サーバをコピー元 EADS サーバとする key および value を一括削除します。

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 removeAll(Node targetNode)
    throws CacheException
```

### (c) パラメータ

targetNode

一括削除を実行する EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

## (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)

## (e) 注意事項

- 例外が返却された場合、削除対象のデータが残っているおそれがあります。そのため、実行結果を確認して対処してください。必要に応じて、removeAll() (EADS サーバ指定) を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## (18) getGroupNameSet()

### (a) 機能

指定した EADS サーバに格納されている最上位階層のグループ名の一覧を取得します。

グループ名の一覧は、ASCII コードの昇順になります。

### (b) 形式

```
public java.util.Set<String> getGroupNameSet(Node targetNode)
                               throws CacheException
```

### (c) パラメータ

targetNode

グループ名の一覧を取得したい EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

### (d) 戻り値

指定した EADS サーバに格納されている最上位階層のグループ名の一覧が返却されます。

指定した EADS サーバに、グループに属する key が存在しない場合は、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)
- 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)

## (f) 注意事項

指定した EADS サーバに存在するグループ数が多いほど、取得に時間が掛かるおそれがあります。また、取得する際のリソースの使用量も増加するため、注意してください。

## (19) getKeySet() (グループ指定)

### (a) 機能

指定したグループに属する key の一覧を取得します。key の一覧には、指定したグループの下位階層のグループに属する key も含まれます。

key の一覧は、ASCII コードの昇順になります。

### (b) 形式

```
public java.util.Set<String> getKeySet(String groupName)
    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\(2\) グループ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

指定したグループに属する key の一覧が返却されます。

指定したグループに属する key が存在しない場合、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)
- 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)

## (f) 注意事項

指定したグループに存在する key の数が多いほど、取得に時間が掛かるおそれがあります。また、取得する際のリソースの使用量が増加します。

## (20) getKeySet() (EADS サーバ指定)

### (a) 機能

指定した EADS サーバに格納されている key の一覧を取得します。

key の一覧は、ASCII コードの昇順になります。

### (b) 形式

```
public java.util.Set<String> getKeySet(Node targetNode)
                               throws CacheException
```

### (c) パラメータ

targetNode

key の一覧を取得したい EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

### (d) 戻り値

指定した EADS サーバに格納されている key の一覧が返却されます。

### (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

指定した EADS サーバに存在する key の数が多いほど、取得に時間が掛かるおそれがあります。また、取得する際のリソースの使用量が増加します。

## (21) getGroupCount()

### (a) 機能

指定した EADS サーバに格納されている最上位階層のグループの数を取得します。

### (b) 形式

```
public int getGroupCount(Node targetNode)
    throws CacheException
```

### (c) パラメータ

targetNode

グループの数を取得したい EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

### (d) 戻り値

指定した EADS サーバに格納されている最上位階層のグループの数が返却されます。

### (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)

## (22) getKeyCount() (グループ指定)

### (a) 機能

指定したグループに属する key の数を取得します。指定したグループの下位階層のグループに属する key も、取得される key の数に含まれます。

### (b) 形式

```
public int getKeyCount(String groupName)
    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(2) グループ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

指定したグループに属する key の数が返却されます。

## (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)

## (23) getKeyCount() (EADS サーバ指定)

### (a) 機能

指定した EADS サーバに格納されている key の数を取得します。

### (b) 形式

```
public int getKeyCount(Node targetNode)
    throws CacheException
```

## (c) パラメータ

targetNode

key の数を取得したい EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

## (d) 戻り値

指定した EADS サーバに属する key の数が返却されます。

## (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)

## (24) `getFirstKey()` (グループ指定)

### (a) 機能

指定したグループに属する key のうち、ASCII コードの昇順で先頭となる key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。

### (b) 形式

```
public String getFirstKey(String groupName)
                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\(2\) グループ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

指定したグループに属する key のうち、ASCII コードの昇順で先頭となる key が返却されます。

指定したグループに属する key が存在しない場合、`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)
- `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)

## (25) `getFirstKey()` (EADS サーバ指定)

### (a) 機能

指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で先頭となる key を取得します。

### (b) 形式

```
public String getFirstKey(Node targetNode)
                throws CacheException
```

## (c) パラメータ

targetNode

key を取得したい EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

## (d) 戻り値

指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で先頭となる key が返却されます。

指定した EADS サーバに key が存在しない場合、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)
- 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)

## (26) getNextKey() (グループおよび key 指定)

### (a) 機能

指定したグループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。

なお、接続先の EADS サーバに存在しない key を指定した場合も同様に、指定した key の次に位置する key を取得します。

### (b) 形式

```
public String getNextKey(String groupName, String key)
    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(2) グループ名として指定できるデータ」を参照してください。

key

基準とする 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。

#### (d) 戻り値

指定したグループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key が返却されます。

指定した key の次に位置する key が存在しない場合、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)
- 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

- このメソッドでは、指定されたグループを基に接続先の EADS サーバを決定します。
- 接続先以外の EADS サーバの縮退、復旧、追加 (スケールアウト処理)、および位置の変更 (リバランス処理) によって、指定したグループに属する key を格納する EADS サーバが変更されることがあります。そのため、getFirstKey() (グループ指定) を実行して先頭となる key を取得したときとは、異なる EADS サーバに接続する可能性があります。
- EADS サーバ上でグループに対して排他処理をしないため、getFirstKey() (グループ指定) または getNextKey() (グループおよび key 指定) の実行後、他プロセスからの操作によって、グループに属する key が挿入、または削除されることがあります。

### (27) getNextKey() (EADS サーバおよび key 指定)

#### (a) 機能

指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で指定した key の次に位置する key を取得します。

なお、接続先の EADS サーバに存在しない key を指定した場合も同様に、指定した key の次に位置する key を取得します。

#### (b) 形式

```
public String getNextKey(Node targetNode, String key)
    throws CacheException
```

## (c) パラメータ

targetNode

key を取得したい EADS サーバを指定します。

CacheManager クラスから取得した Node クラスのインスタンスだけ指定できます。それ以外のインスタンスを指定した場合、動作は保証されません。

key

基準とする 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。

## (d) 戻り値

指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で、指定した key の次に位置する key が返却されます。

指定した key の次に位置する key が存在しない場合、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)
- 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

- 接続先以外の EADS サーバの縮退、復旧、追加 (スケールアウト処理)、および位置の変更 (リバランス処理) によって、指定した key を格納する EADS サーバが変更される場合があります。
- EADS サーバ上でグループに対して排他処理をしないため、getFirstKey() (EADS サーバ指定) または getNextKey() (EADS サーバおよび key 指定) の実行後、他プロセスからの操作によって、グループに属する key が挿入、または削除されることがあります。

## (28) executeFunction() (key またはグループ指定)

### (a) 機能

指定した key またはグループから実行する EADS サーバを決定して、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 keyOrGroupName,  
                             String funcName,
```

Object arg)  
throws CacheException

## (c) パラメータ

keyOrGroupName

key または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」、または「15.2.2(2) グループ名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。

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。

## (29) executeFunction() (key またはグループ, および受信タイムアウト指定)

### (a) 機能

指定した key またはグループから実行する EADS サーバを決定して、ユーザファンクションを実行します。また、受信タイムアウト時間を指定します。

受信タイムアウト時間には、クライアント定義の `eads.client.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 keyOrGroupName,
                             String funcName,
                             Object arg,
                             int recvTimeout)
    throws CacheException
```

### (c) パラメータ

`keyOrGroupName`

key または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](#)」, または「[15.2.2\(2\) グループ名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。

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.client.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。

### (30) 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。

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。

# (31) 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.client.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。

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.client.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)

- 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)

## (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。

## 18.1.2 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。

メソッド名	説明
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。
<b>【非推奨】</b> getNode()	<b>【非推奨】</b> 指定した key (またはグループ) を格納する、本来の EADS サーバ (コピー元 EADS サーバ) の情報を取得します。
getSlaveNodeList()	指定した EADS サーバに格納されているデータの、本来のコピー元 EADS サーバの情報を取得します。
getCurrentMasterNode()	指定した key (またはグループ) を現時点で格納する、コピー元 EADS サーバの情報を取得します。
getOriginalMasterNode()	指定した key (またはグループ) を格納する、本来のコピー元 EADS サーバの情報を取得します。
addClusterEventListener()	クラスタイベントリスナを追加します。
addNodeEventListener()	ノードイベントリスナを追加します。
removeClusterEventListener()	クラスタイベントリスナを削除します。
removeNodeEventListener()	ノードイベントリスナを削除します。

## (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。

クライアント定義の `eads.client.eventListener.enable` パラメタに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、destroy()実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが1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。

## (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。
- Java EE 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。

クライアント定義の `eads.client.eventListener.enable` パラメータに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、destroy()実行時に削除されます。
- クラスイベントリスナおよびノードイベントリスナを実行するためのスレッドが1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。

## (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\(5\) 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。
- Java EE 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。

クライアント定義の `eads.client.eventListener.enable` パラメタに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、destroy()実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが1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。

### (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。
- Java EE 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。

クライアント定義の `eads.client.eventListener.enable` パラメータに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、`destroy()`実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが1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。

### (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\(5\) 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。
- Java EE 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。

クライアント定義の `eads.client.eventListener.enable` パラメタに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、destroy() 実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが1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。

### (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。
- Java EE 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。

クライアント定義の `eads.client.eventListener.enable` パラメータに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、destroy() 実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが1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。

## (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\(5\) 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。
- Java EE 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\(4\) 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)
- `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)

## (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\(4\) 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。

クライアント定義の `eads.client.eventListener.enable` パラメタに `true` を指定していた場合は、次のように動作します。

- クラスタイベントリスナおよびノードイベントリスナが存在する場合は、すべてのクラスタイベントリスナおよびノードイベントリスナを削除します。
- すべての EADS サーバ間で、イベントリスナ用のコネクションを切断します。

#### (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。
- Java EE 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。
- このメソッドの実行時には、EADS サーバの接続可否は考慮されません。そのため、取得した情報には、縮退状態などで接続できない EADS サーバが含まれていることがあります。取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうか、Node クラスの isEnabled() で確認してください。

Node クラスの isEnabled() については、「[18.1.3\(6\) isEnabled\(\)](#)」を参照してください。

## (15) 【非推奨】getNode()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、CacheManager クラスの [getOriginalMasterNode\(\)](#) を使用してください。

### (a) 機能

指定した key (またはグループ) を格納する、本来のコピー元 EADS サーバの情報を取得します。

本来のコピー元 EADS サーバとは、クラスタを構成するすべての EADS サーバが正常に接続できる場合に、指定した key（またはグループ）のコピー元のデータを格納する EADS サーバのことです。

## (b) 形式

```
public Node getNode(String key)
    throws CacheException
```

## (c) パラメータ

key

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）の本来のコピー元 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。
- クラスタ構成に変更がなければ、EADS サーバの接続可否に関係なく、常に同じ EADS サーバの情報が取得されます。
- 取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうか、Node クラスの `isEnabled()` で確認してください。  
Node クラスの `isEnabled()` については、「[18.1.3\(6\) isEnabled\(\)](#)」を参照してください。

# (16) `getSlaveNodeList()`

## (a) 機能

指定した EADS サーバに格納されているデータの、本来のコピー先 EADS サーバの情報を取得します。

本来のコピー先 EADS サーバとは、クラスタを構成するすべての EADS サーバが正常に接続できる場合に、指定した EADS サーバ（コピー元 EADS サーバ）に格納されているデータの、コピー先となる EADS サーバのことです。

## (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

本来のコピー先 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。
- このメソッドの実行時には、指定した EADS サーバ、および情報の取得対象となる EADS サーバの接続可否は考慮されません。そのため、取得した情報には、縮退状態などで接続できない EADS サーバの情報が含まれていることがあります。

## (17) getCurrentMasterNode()

### (a) 機能

指定した key (またはグループ) を現時点で格納する、コピー元 EADS サーバの情報を取得します。

## (b) 形式

```
public Node getCurrentMasterNode(String key)
    throws CacheException
```

## (c) パラメータ

key

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) を現時点で格納するコピー元 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。

## (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)
- `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。

# (18) `getOriginalMasterNode()`

## (a) 機能

指定した key (またはグループ) を格納する、本来のコピー元 EADS サーバの情報を取得します。

本来のコピー元 EADS サーバとは、クラスタを構成するすべての EADS サーバが正常に接続できる場合に、指定した key (またはグループ) のマスタとなるデータ (コピー元となるデータ) を格納する EADS サーバのことです。現時点のコピー元 EADS サーバとは異なることがあります。

クラスタ構成に変更がなければ、EADS サーバの接続可否に関係なく、常に同じ EADS サーバの情報が取得されます。

## (b) 形式

```
public Node getOriginalMasterNode(String key)
    throws CacheException
```



## (c) パラメータ

key

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) を格納する本来のコピー元 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。
- 取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうか、Node クラスの `isEnabled()` で確認してください。  
Node クラスの `isEnabled()` については、「[18.1.3\(6\) isEnabled\(\)](#)」を参照してください。

## (19) addClusterEventListener()

### (a) 機能

クラスタイベントリスナを追加します。

### (b) 形式

```
public Node[] addClusterEventListener(  
    ClusterEventListener clusterEventListener)  
    throws CacheException
```

### (c) パラメータ

clusterEventListener

ユーザが作成した `ClusterEventListener` インタフェース実装クラスを指定します。

## (d) 戻り値

クラスタイベントリスナの追加が終了した時点での接続先 EADS サーバの情報一覧が、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。

## (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

- `clusterEventListener` には、同一クライアント内のほかのクラスタイベントリスナまたはノードイベントリスナと同じインスタンスを指定できません。
- このメソッドの実行と同時にクラスタ構成情報が変わった場合、縮退している EADS サーバ、または追加された EADS サーバに対しては、`ClusterEventListener` インタフェースのメソッドを呼び出しません。そのため、メソッド実行後に、返却された EADS サーバの情報を確認してください。
- このメソッドの実行時には、EADS サーバの接続可否は考慮されません。そのため、取得した情報には、接続できない EADS サーバ (例: EADS サーバの縮退) が含まれていることがあります。取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうかを、Node クラスの `isEnabled()` で確認してください。

Node クラスの `isEnabled()` については、「[18.1.3\(6\) isEnabled\(\)](#)」を参照してください。

## (20) addNodeEventListener()

### (a) 機能

ノードイベントリスナを追加します。

### (b) 形式

```
public Node[] addNodeEventListener(
    NodeEventListener nodeEventListener,
    int nodeId,
    NodeEventType[] nodeEventTypeList)
    throws CacheException
```

### (c) パラメータ

`nodeEventListener`

ユーザが作成した `NodeEventListener` インタフェース実装クラスを指定します。

`nodeId`

ノードイベントリスナを実行する EADS サーバ ID を指定します。

## nodeEventListener

ノードイベントリスナを実行する条件の一覧を、NodeEventType クラスの配列として指定します。

### (d) 戻り値

ノードイベントリスナの追加が終了した時点での接続先 EADS サーバの情報一覧が、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。

### (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

- nodeEventListener には、同一クライアント内のほかのクラスイベントリスナまたはノードイベントリスナと同じインスタンスを指定できません。
- このメソッドの実行と同時にクラスタ構成情報が変わった場合、縮退している EADS サーバ、または追加された EADS サーバに対しては、NodeEventListener インタフェースのメソッドを呼び出しません。そのため、メソッド実行後に、返却された EADS サーバの情報を確認してください。
- このメソッドの実行時には、EADS サーバの接続可否は考慮されません。そのため、取得した情報には、接続できない EADS サーバ (例: EADS サーバの縮退) が含まれていることがあります。取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうかを、Node クラスの isEnabled() で確認してください。

Node クラスの isEnabled() については、「18.1.3(6) isEnabled()」を参照してください。

## (21) removeClusterEventListener()

### (a) 機能

クラスタイベントリスナを削除します。

未通知のイベントの情報を破棄します。

このメソッドは、addClusterEventListener() と対になるように呼び出してください。

### (b) 形式

```
public void removeClusterEventListener(  
    ClusterEventListener clusterEventListener)  
    throws CacheException
```

## (c) パラメータ

clusterEventListener

ユーザが作成した ClusterEventListener インタフェース実装クラスを指定します。

## (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)

## (22) removeNodeEventListener()

### (a) 機能

ノードイベントリスナを削除します。

未通知のイベントの情報を破棄します。

このメソッドは、addNodeEventListener()と対になるように、呼び出してください。

### (b) 形式

```
public void removeNodeEventListener(  
    NodeEventListener nodeEventListener)  
    throws CacheException
```

## (c) パラメータ

nodeEventListener

ユーザが作成した NodeEventListener インタフェース実装クラスを指定します。

## (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)

## 18.1.3 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>”
```

## 18.1.4 FailureOperationInfo クラス

### (1) 説明

一括操作を試みた結果、一部またはすべての操作に失敗した場合に、失敗した操作の情報を格納するためのクラスです。

### (2) 継承関係

```
java.lang.Object  
└─com.hitachi.software.xeads.client.api.FailureOperationInfo
```

### (3) 形式

```
public class FailureOperationInfo  
extends java.lang.Object
```

### (4) メソッド一覧

FailureOperationInfo 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>getKey()</code>	失敗した操作で使用した key を取得します。
<code>getErrorCode()</code>	失敗した操作のエラーの要因を示すエラーコードを取得します。

メソッド名	説明
<code>getException()</code>	失敗した操作で発生した例外を取得します。

## (5) `getKey()`

### (a) 機能

失敗した操作で使った `key` を取得します。

### (b) 形式

```
public String getKey()
```

### (c) 戻り値

失敗した操作で使った `key` が返却されます。

## (6) `getErrorCode()`

### (a) 機能

失敗した操作のエラーの要因を示すエラーコードを取得します。

### (b) 形式

```
public int getErrorCode()
```

### (c) 戻り値

失敗した操作のエラーの要因を示すエラーコードが返却されます。

## (7) `getException()`

### (a) 機能

失敗した操作で発生した例外を取得します。

### (b) 形式

```
public CacheException getException()
```

### (c) 戻り値

失敗した操作で発生した例外が返却されます。



## 18.1.5 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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
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。	△	1000
1010	EAD_ERROR_INVALID_PARAMETER	UserOperationException	指定されたパラメータが不正です。	API の引数に不正なパラメータを指定しています。	×	1010
1030	EAD_ERROR_CLIENT_FINALIZED	UserOperationException	EADS クライアントの使用を終了しているため、処理を実行できませんでした。	CacheManager クラスの <code>destroy()</code> 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。	×	1030
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 <code>removeCache()</code> 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終了したあと (CacheManager クラスの <code>destroy()</code> 実行後) に、データを操作した。</li> </ul>	×	1040
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。	×	1050
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。	—	1060

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
				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_NEEDED_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 <code>removeCache()</code> の実行後) に、キャッシュを終了した。</li> <li>CacheManager クラスの <code>getCache()</code> を実行していない状態で、CacheManager クラスの <code>removeCache()</code> を実行した。</li> </ul>	—	1100
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 EADS サーバのアドレス情報とも一致しません。	—	1110
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。	×	1120
1200	EAD_ERROR_EVENTLISTENER	UserOperationException	クラスイベントリスナとノードイベントリスナ	イベントリスナが無効になっている状態 (クライアント定義の	—	—

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
	R_NOT_INITIALIZED		トリスナを追加できません。	<p><code>eads.client.eventListener.enable</code> パラメータに <code>true</code> が指定されていない) で、次の処理を実行しました。</p> <ul style="list-style-type: none"> <li>クラスタイベントリスナの追加 (CacheManager クラスの <code>addClusterEventListener()</code> の実行)</li> <li>ノードイベントリスナの追加 (CacheManager クラスの <code>addNodeEventListener()</code> の実行)</li> </ul>		
1210	EAD_ERROR_EVENTLISTENER_EXIST	UserOperation Exception	すでに追加されているクラスタイベントリスナまたはノードイベントリスナを、追加しようとした。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>クラスタイベントリスナの追加後 (CacheManager クラスの <code>addClusterEventListener()</code> の実行後) に、クラスタイベントリスナの追加をした。</li> <li>ノードイベントリスナの追加後 (CacheManager クラスの <code>addNodeEventListener()</code> の実行後) に、ノードイベントリスナの追加をした。</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	データを更新するAPIの処理状態※	クライアント定義の eads.client.compat パラメータに 0300 を指定した場合のエラーコード
1220	EAD_ERROR_EVENTLISTENER_NOT_NEED_REMOVE	UserOperation Exception	削除する必要がないクラスイベントリスナまたはノードイベントリスナを、削除しようとしてしました。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>CacheManager クラスの destroy() の実行後に、イベントリスナの削除をした。</li> <li>クラスイベントリスナの削除後 (CacheManager クラスの removeClusterEventListener() の実行後) に、クラスイベントリスナの削除をした。</li> <li>ノードイベントリスナの削除後 (CacheManager クラスの removeNodeEventListener() の実行後) に、ノードイベントリスナの削除をした。</li> <li>クラスイベントリスナの追加 (CacheManager クラスの addClusterEventListener() の実行) をしていない状態で、クラスイベントリスナの削除 (CacheManager クラスの removeClusterEve</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	データを更新するAPIの処理状態※	クライアント定義の eads.client.compat パラメータに 0300 を指定した場合のエラーコード
				<p>ntListener()の実行)をした。</p> <ul style="list-style-type: none"> <li>ノードイベントリスナの追加 (CacheManager クラスの addNodeEventListener()の実行)をしていない状態で、ノードイベントリスナの削除 (CacheManager クラスの removeNodeEventListener()の実行)をした。</li> </ul>		
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。	—	2000
2010	EAD_ERROR_INIT_PROPERTIES	InitializeException	クライアント定義ファイルが読み込みませんでした。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <li>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> <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>	—	2010

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	データを更新するAPIの処理状態※	クライアント定義の <a href="#">eads.client.compat</a> パラメータに 0300 を指定した場合のエラーコード
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。	—	2020
2030	EAD_ERROR_INIT_LOGGER	InitializeException	ログの初期化に失敗しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>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>	—	2030
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。	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> </ul>	—	2040

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
				<ul style="list-style-type: none"> <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> <li>接続先 EADS サーバでノンブロッキング I/O 通信バッファが不足した。</li> <li>接続先 EADS サーバとの通信中にタイムアウトが発生した。</li> </ul>		
3000	EAD_ERROR_N ET	ServerCommunicationException	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> </ul>	△	3000
3001	EAD_ERROR_N ET_SEND_REQUEST	ServerCommunicationException	EADS サーバへのリクエスト送信処理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> </ul>	△	3000
3002	EAD_ERROR_N ET_RECEIVE_RESPONSE	ServerCommunicationException	EADS サーバへのレスポンス受信処理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> </ul>	△	3000



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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
				<ul style="list-style-type: none"> <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> </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。	<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>	△	3010
3011	EAD_ERROR_NET_SEND_TIMEOUT	ServerCommunicationException	EADS サーバへのリクエスト送信処理でタイムアウトが発生しました。	<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>	△	3010
3012	EAD_ERROR_NET_RECEIVE_TIMEOUT	ServerCommunicationException	EADS サーバへのレスポンス受信処理でタイムアウトが発生しました。	<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> </ul>	△	3010

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
				<ul style="list-style-type: none"> <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_NETWORK_CONNECTION	ServerCommunicationException	EADS サーバと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>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>	×	3020
3030	EAD_ERROR_NETWORK_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。	△	3030
3040	EAD_ERROR_NETWORK_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。	<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> </ul>	—	3040

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
				<ul style="list-style-type: none"> <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> <li>• EADS がサポートしていない JDK (例: Oracle の JDK 1.7) を使用している。</li> <li>• 接続先 EADS サーバとの通信中にタイムアウトが発生した。</li> </ul>		
4000	EAD_ERROR_SERVER	InternalServerException	予期しない内部エラーが EADS サーバで発生しました。	予期しない問題が接続先 EADS サーバで発生しました。	△	4000
4010	EAD_ERROR_SERVER_UNSUPPORTED_REQUEST	InternalServerException	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>• 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> <li>• 接続先 EADS サーバがサポートしていない API を使用した。</li> </ul>	×	4010

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
4030	EAD_ERROR_SERVER_UNAVAILABLE	InternalServerException	EADS サーバ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。	×	4030
4040	EAD_ERROR_SERVER_INCOMPATIBLE_CLUSTERINFO	InternalServerException	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。	×	4040
4060	EAD_ERROR_SERVER_REPLACE_METHOD_NOT_MATCHED	InternalServerException	<code>replace()</code> 実行時、すでに格納されている value が <code>comparativeValue</code> 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 <code>replace()</code> 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。	×	4060
4070	EAD_ERROR_SERVER_REPLACE_METHOD_KEY_NOT_EXIST	InternalServerException	<code>replace()</code> 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 <code>replace()</code> 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。	×	4070
4080	EAD_ERROR_SERVER_CREATE_METHOD_KEY_EXIST	InternalServerException	<code>create()</code> 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 <code>create()</code> 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。	×	4080
4090	EAD_ERROR_SERVER_UPDATE_METHOD_KEY_NOT_EXIST	InternalServerException	<code>update()</code> 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 <code>update()</code> 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。	×	4090

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
4100	EAD_ERROR_SERVER_NOT_RUNNING	InternalServerException	リクエストを処理できる EADS サーバがありません。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>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になっている。</li> <li>クラスタが稼働していない状態になっている。</li> </ul>	×	4100
4110	EAD_ERROR_SERVER_STATUS	InternalServerException	EADS サーバの状態が、リクエストを実行できる状態ではありません。	接続先 EADS サーバの状態がリクエストを実行できる状態でないため、リクエストを処理できませんでした。	×	4000
4120	EAD_ERROR_SERVER_NOT_EXECUTE_REQUEST_TEMPORARILY	InternalServerException	EADS サーバは、一時的にリクエストを実行できません。	スケールアウト処理中のため、受付済みのリクエストを一時的に処理できませんでした。スケールアウト処理が完了するまで待機しないでエラーとなる設定になっています。そのため、スケールアウト処理が完了したあとに、リクエストを再実行してください。	×	4000
4200	EAD_ERROR_SERVER_CACHE	InternalServerException	キャッシュに対する操作を実行できませんでした。	接続先 EADS サーバで何らかの問題が発生しているため、キャッシュを操作できない状態になっています。操作を中止して EADS	×	4000

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
				サーバの状態を確認してください。		
4210	EAD_ERROR_SERVER_CACHE_NOT_FOUND	InternalServerException	指定したキャッシュが存在しないため、キャッシュに対する操作を実行できませんでした。	指定したキャッシュが存在しないため、キャッシュを操作できませんでした。指定したキャッシュに対する操作を中止して、EADS サーバの状態を確認してください。	×	4000
4230	EAD_ERROR_SERVER_CACHE_CLUSTER_UPDATE	InternalServerException	リクエスト処理中にクラスタ構成変更が発生したため、キャッシュに対する操作を実行できませんでした。	リクエスト処理中にクラスタ構成変更が発生したため、キャッシュを操作できませんでした。クラスタ構成の変更が完了したあとに、キャッシュに対する操作を再度実行してください。	×	4000
4300	EAD_ERROR_SERVER_CACHE_BEFORE_REPLICATION	InternalServerException	キャッシュに対する操作中に内部エラーが発生しましたが、データの多重化は発生しませんでした。	接続先 EADS サーバでキャッシュに対する操作中、内部エラーが発生しました。データの多重化は発生していないため、ほかの正常な EADS サーバへの影響はありません。EADS サーバが縮退したあと、正常な EADS サーバに接続先が変更されると、同じ操作から処理を再開できます。	×	4000
4310	EAD_ERROR_SERVER_CACHE_AFTER_REPLICATION	InternalServerException	キャッシュに対する操作中に EADS サーバで内部エラーが発生しました。データ	キャッシュに対する操作中、接続先 EADS サーバで内部エラーが発生しました。データ	△	4000

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
			生し、データ更新処理が失敗しました。	の多重化はすでに実行されているため、問題が発生した接続先 EADS サーバが縮退したあとに正常なサーバに接続先が変更されると、データが更新された状態から操作を再開できます。		
4400	EAD_ERROR_SERVER_LIMIT_RECEIVE_BUFFER	InternalServerException	EADS サーバのリクエスト受信処理でノンブロッキング I/O 通信バッファが不足しました。	接続先 EADS サーバのリクエスト受信処理でノンブロッキング I/O 通信バッファが不足しているため、リクエストを処理できませんでした。	×	4000
4410	EAD_ERROR_SERVER_LIMIT_SEND_BUFFER	InternalServerException	EADS サーバのレスポンス送信処理でノンブロッキング I/O 通信バッファが不足しました。	接続先 EADS サーバのレスポンス送信処理でノンブロッキング I/O 通信バッファが不足しているため、レスポンスを送信できませんでした。	△	4000
4700	EAD_ERROR_SERVER_FUNCTION_EXECUTE	InternalServerException	EADS サーバのユーザ機能内でエラーが発生しました。	接続先 EADS サーバのユーザ機能内でエラーが発生しました。ユーザ機能内の処理を確認してください。	—	4000
4710	EAD_ERROR_SERVER_FUNCTION_RETURN_SERIALIZE	InternalServerException	EADS サーバでユーザ機能の戻り値のシリアライズに失敗しました。	接続先 EADS サーバで実行したユーザ機能の戻り値に、シリアライズできないオブジェクトを指定しています。	—	4000

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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
4720	EAD_ERROR_SERVER_FUNCTION_ARG_DESERIALIZE	InternalServerException	ユーザ関クションの引数に、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ています。	ユーザ関クションの引数に、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ています。	—	4100
4730	EAD_ERROR_SERVER_FUNCTION_NOT_FOUND	InternalServerException	指定したユーザ関クション名のユーザ関クションが、EADS サーバに存在しません。	指定したユーザ関クション名のユーザ関クションが、接続先 EADS サーバに存在しません。	—	4000
4800	EAD_ERROR_SERVER_LIMIT_EXTERNAL_MEMORY	InternalServerException	データを格納するメモリの容量が不足しています。	接続先 EADS サーバでデータを格納するメモリ (Explicit ヒープ) の容量が不足しているため、リクエストを処理できませんでした。	×	4000
4810	EAD_ERROR_SERVER_LIMIT_CACHE_FILE	InternalServerException	データを格納するキャッシュファイルの容量が不足しています。	接続先 EADS サーバでデータを格納するキャッシュファイルの容量が不足しているため、リクエストを処理できませんでした。	×	4000
4820	EAD_ERROR_SERVER_LIMIT_KV_COUNT	InternalServerException	EADS サーバに格納できる key の数が上限に達しています。	接続先 EADS サーバに格納できる key の数が上限に達しているため、リクエストを処理できませんでした。	×	4000
4830	EAD_ERROR_SERVER_LIMIT_KEY_VALUE_LENGTH	InternalServerException	指定した key、グループ名、または value のサイズがクラスタに格納できる最大サイズを超えています。	指定した key、グループ名、または value のサイズがクラスタに格納できる最大サイズを超えているため、リクエストを処理できませんでした。	×	4000



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	データを更新するAPIの処理状態※	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合のエラーコード
4840	EAD_ERROR_SERVER_LIMIT_EXTERNAL_MEMORY_TEMPORARILY	InternalServerException	接続先 EADS サーバで、データを格納するメモリの容量が一時的に不足しています。	リバランス処理中は、EADS サーバでデータを格納するメモリ (Explicit ヒープ) の容量が一時的に減少します。そのため、接続先 EADS サーバでリクエストを処理できませんでした。	×	4000
4850	EAD_ERROR_SERVER_LIMIT_KV_COUNT_TEMPORARILY	InternalServerException	接続先 EADS サーバで、格納できる key の数が一時的に上限に達しています。	リバランス処理中は、EADS サーバに格納できる key の数の上限が一時的に減少します。そのため、接続先 EADS サーバでリクエストを処理できませんでした。	×	4000
4999	EAD_ERROR_SERVER_UNKNOWN	InternalServerException	解析できない内部エラーが EADS サーバで発生しました。	接続先 EADS サーバで内部エラーが発生しましたが、接続先 EADS サーバのバージョンが EADS クライアントライブラリのバージョンより新しいため、解析できませんでした。	△	4000
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。	△	5000
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。	△	5010
5020	EAD_ERROR_CLIENT_BATCH_CANCEL	InternalClientException	一括操作の実行をキャンセルしました。	一括操作を継続できないため、未実行の操作をキャンセルしました。	×	5020

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	データを更新するAPIの処理状態※	クライアント定義の eads.client.compat パラメータに 0300 を指定した場合のエラーコード
6000	EAD_ERROR_BATCH_FAILED_ALL	BatchOperationException	一括操作のすべての操作に失敗しました。	データを一括で操作する API で実行しようとした操作は、すべて失敗しました。	—	6000
6010	EAD_ERROR_BATCH_FAILED_PART	BatchOperationException	一括操作の一部の操作に失敗しました。	データを一括で操作する API で実行しようとした操作のうち、一部が失敗しました。	—	6010

#### 注※

put(), remove()などのデータを更新する API の実行時にエラーコードが出力された場合に、データが更新されているかどうかを表しています。

記号の意味を次に示します。

△：データが更新されているかどうかは不明です。処理が完了したかどうか、確認してください。

×：データは更新されていません。

—：put(), remove()などのデータを更新する API の実行時に、このエラーコードは出力されません。

## 18.1.6 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
```

## 18.1.7 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
```

## 18.1.8 InternalServerException 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.InternalServerException
```

### (3) 形式

```
public class InternalServerException
extends CacheException
```

## 18.1.9 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
```

## 18.1.10 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
```

## 18.1.11 BatchOperationException クラス

### (1) 説明

一括操作を試みた結果、一部またはすべての操作に失敗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.BatchOperationException
```

### (3) 形式

```
public class BatchOperationException
    extends CacheException
```

### (4) メソッド一覧

BatchOperationException 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
<a href="#">getSuccessOperationNumber()</a>	一括操作を試みた結果、操作に成功した key の数を取得します。
<a href="#">getFailureOperationInfo()</a>	一括操作を試みた結果、失敗した操作の情報の一覧を取得します。

### (5) getSuccessOperationNumber()

#### (a) 機能

一括操作を試みた結果、操作に成功した key の数を取得します。

#### (b) 形式

```
public int getSuccessOperationNumber()
```

#### (c) 戻り値

操作に成功した key の数が返却されます。

## (6) getFailureOperationInfo()

### (a) 機能

一括操作を試みた結果、失敗した操作の情報の一覧を取得します。

### (b) 形式

```
public java.util.Set<FailureOperationInfo> getFailureOperationInfo()
```

### (c) 戻り値

失敗した操作の情報を格納した FailureOperationInfo クラスの一覧が返却されます。

## 18.1.12 AllFailureException クラス

### (1) 説明

一括操作を試みた結果、すべての操作に失敗した場合に返却される BatchOperationException 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.BatchOperationException
│               └─com.hitachi.software.xeads.client.api.AllFailureException
```

### (3) 形式

```
public class AllFailureException
extends BatchOperationException
```

## 18.1.13 PartFailureException クラス

### (1) 説明

一括操作を試みた結果、一部の操作に失敗した場合に返却される BatchOperationException 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.BatchOperationException
│               └─com.hitachi.software.xeads.client.api.PartFailureException
```

## (3) 形式

```
public class PartFailureException
extends BatchOperationException
```

### 18.1.14 ClusterEventListener インタフェース

#### (1) 説明

クラスタイベントリスナが実装する必要のあるインタフェースです。

#### (2) インタフェース名

```
com.hitachi.software.xeads.client.api.ClusterEventListener
```

#### (3) メソッド一覧

ClusterEventListener 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>onEvent()</code>	クラスタイベントリスナの処理を実装します。イベントの発生時に呼び出されます。

#### (4) onEvent()

##### (a) 機能

クラスタイベントリスナの処理を実装します。イベントの発生時に呼び出されます。

##### (b) 形式

```
public void onEvent(ClusterEventContext context)
```

## (c) パラメータ

context

発生したイベントに関する情報を取得します。

詳細は「[18.1.16 ClusterEventContext インタフェース](#)」を参照してください。

## (d) 注意事項

クラスイベントリスナで例外やエラーを catch しない場合、クラスイベントリスナ内で予期しないエラーが発生しても、EADS クライアントの処理を継続します。

# 18.1.15 NodeEventListener インタフェース

## (1) 説明

ノードイベントリスナが実装する必要があるインタフェースです。

## (2) インタフェース名

```
com.hitachi.software.xeads.client.api.NodeEventListener
```

## (3) メソッド一覧

NodeEventListener 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
onEvent()	ノードイベントリスナの処理を実装します。イベントの発生時に呼び出されます。

## (4) onEvent()

### (a) 機能

ノードイベントリスナの処理を実装します。イベントの発生時に呼び出されます。

### (b) 形式

```
public void onEvent(NodeEventContext context)
```

## (c) パラメータ

context

発生したイベントに関する情報を取得します。



詳細は「18.1.17 NodeEventContext インタフェース」を参照してください。

## (d) 注意事項

- ノードイベントリスナで例外やエラーを catch しない場合、ノードイベントリスナ内で予期しないエラーが発生しても、EADS クライアントの処理を継続します。NodeEventContext インタフェースの `getRangeId()` で発生した例外を catch しない場合も、EADS クライアントの処理を継続します。
- NodeEventContext インタフェースの `getNodeEventType()` で RECOVERED を取得した直後に、Cache クラスのメソッドを呼び出すと、イベントが発生した EADS サーバの処理で例外が発生するおそれがあります。例外が発生した場合は、しばらくしてから再実行してください。  
イベントが発生した EADS サーバとは、NodeEventContext インタフェースの `getNodeId()` が示す EADS サーバです。

## 18.1.16 ClusterEventContext インタフェース

### (1) 説明

クラスタイベントリスナの実行に必要な情報を ClusterEventListener インタフェースに渡すインタフェースです。

ClusterEventContext インタフェースのインスタンスは、クラスタイベントリスナの呼び出し直前に生成され、クラスタイベントリスナの処理完了後に解放されます。

### (2) インタフェース名

```
com.hitachi.software.xeads.client.api.ClusterEventContext
```

### (3) メソッド一覧

ClusterEventContext 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>getClusterEventType()</code>	イベントが発生した契機を取得します。
<code>getNode()</code>	イベントが発生した EADS サーバの情報を取得します。

### (4) `getClusterEventType()`

#### (a) 機能

イベントが発生した契機を取得します。

## (b) 形式

```
public ClusterEventType getClusterEventType()
```

## (c) 戻り値

イベントが発生した契機が返却されます。

## (5) getNode()

### (a) 機能

EADS サーバの情報を取得します。

### (b) 形式

```
public Node getNode()
```

### (c) 戻り値

イベントが発生した EADS サーバの情報が返却されます。

## 18.1.17 NodeEventContext インタフェース

### (1) 説明

ノードイベントリスナーの実行に必要な情報を NodeEventListener インタフェースに渡すインタフェースです。

NodeEventContext インタフェースのインスタンスは、ノードイベントリスナーの呼び出し直前に生成され、ノードイベントリスナーの処理完了後に解放されます。

### (2) インタフェース名

```
com.hitachi.software.xeads.client.api.NodeEventContext
```

### (3) メソッド一覧

NodeEventContext 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>getNodeEventType()</code>	イベントが発生した契機を取得します。
<code>getNodeId()</code>	イベントが発生した EADS サーバ ID を取得します。

メソッド名	説明
<code>getRangeId()</code>	イベントが発生したレンジ ID を取得します。

`getNodeEventType()`で返却される値 (NodeEventType) によって、NodeEventListener インタフェースの `onEvent()`内で実行できるメソッドが変わります。

NodeEventListener インタフェースの `onEvent()`内でのメソッドの実行可否を次の表に示します。

表 18-2 NodeEventListener インタフェースの `onEvent()`内でのメソッドの実行可否

メソッド名	<code>getNodeEventType()</code> で返却される値 (NodeEventType)	
	NodeEventType. ISOLATED, NodeEventType. RECOVERED	NodeEventType.PROMOTED_TO_MASTER, NodeEventType.PROMOTED_TO_SLAVE, NodeEventType.DEMOTED_TO_SLAVE, NodeEventType.NO_ROLE
<code>getNodeId()</code>	○	○
<code>getRangeId()</code>	×	○

(凡例) ○：実行できます。 ×：実行できません。

注

`onEvent()`内で実行できないメソッドの場合は、例外として `IllegalStateException` が返却されます。

## (4) `getNodeEventType()`

### (a) 機能

イベントが発生した契機を取得します。

### (b) 形式

```
public NodeEventType getNodeEventType()
```

### (c) 戻り値

イベントが発生した契機が返却されます。

## (5) `getNodeId()`

### (a) 機能

イベントが発生した EADS サーバ ID を取得します。

## (b) 形式

```
public int getNodeId()
```

## (c) 戻り値

イベントが発生した EADS サーバ ID が返却されます。

## (6) getRangeId()

### (a) 機能

イベントが発生したレンジ ID を取得します。

### (b) 形式

```
public int getRangeId()
```

### (c) 戻り値

イベントが発生したレンジ ID が返却されます。

### (d) 発生する例外

- `IllegalStateException` (実行できないメソッドの呼び出し)

## 18.1.18 列挙型 ClusterEventType

### (1) 説明

クラスタイベントリスナーの実行の契機を表す列挙型です。

### (2) 列挙型名

```
com.hitachi.software.xeads.client.api.ClusterEventType
```

### (3) 列挙定数

列挙型 `ClusterEventType` の定数を次に示します。

定数	説明
ADDED	EADS サーバが追加されました。

## 18.1.19 列挙型 NodeEventType

### (1) 説明

ノードイベントリスナーの実行の契機を表す列挙型です。

### (2) 列挙型名

```
com.hitachi.software.xeads.client.api.NodeEventType
```

### (3) 列挙定数

列挙型 NodeEventType の定数を次に示します。

定数	説明
ISOLATED	<ul style="list-style-type: none"><li>• 障害の発生によって、EADS サーバが縮退しました。</li><li>• EADS サーバが停止しました。</li><li>• <code>eztool isolate</code> コマンドの実行によって、EADS サーバが縮退しました。</li></ul>
RECOVERED	EADS サーバが復旧しました。
PROMOTED_TO_MASTER	データを多重化している場合に、マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバに変わりました。
PROMOTED_TO_SLAVE	データを多重化している場合に、データのコピー先の EADS サーバに変わりました。
DEMOTED_TO_SLAVE	データを多重化している場合に、マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバから、データのコピー先の EADS サーバに変わりました。
NO_ROLE	EADS サーバが、次のどちらでもなくなりました。 <ul style="list-style-type: none"><li>• マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバ</li><li>• データのコピー先の EADS サーバ</li></ul>

## 18.2 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 18-3 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	ClusterInfo	クラスタの情報を取得するインタフェースです。	com.hitachi.software.xeads.common.ClusterInfo
7	CacheInfo	キャッシュの情報を取得するインタフェースです。	com.hitachi.software.xeads.common.CacheInfo
8	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
9	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
10	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
11	Value	ユーザファンクションで使用できる API 内で, key に関連づけて格納される value を表現するインタフェースです。	com.hitachi.software.xeads.func.store.Value
12	UserLogger	ユーザログのインタフェースです。	com.hitachi.software.xeads.common.UserLogger

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
13	<a href="#">EADsStoreException</a>	データ操作に関連する処理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
14	<a href="#">InternalServerException</a>	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
15	<a href="#">UserOperationException</a>	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
16	列挙型 <a href="#">CacheType</a>	キャッシュタイプを表す列挙型です。	com.hitachi.software.xeads.common.CacheType

## 18.2.1 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が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
<a href="#">init()</a>	ユーザファンクションの初期化処理を実装します。
<a href="#">execute()</a>	ユーザファンクションの処理を実装します。
<a href="#">destroy()</a>	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

ユーザ関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関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

ユーザ関クションの実行に必要な情報を取得します。

ユーザ関クションの実行に必要な情報については、「[18.2.2 FunctionContext インタフェース](#)」を参照してください。

## (5) execute()

### (a) 機能

ユーザ関クションの処理を実装します。

EADS クライアントからの、ユーザ関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

ユーザ関クションの実行に必要な情報を取得します。

ユーザ関クションの実行に必要な情報については、「[18.2.2 FunctionContext インタフェース](#)」を参照してください。

### (d) 戻り値

ユーザ関クションの処理結果が返却されます。

Value インスタンスを戻り値に指定した場合だけ、Value インタフェースの getObject() の内容が呼び出し元に返却されます。



## (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、「[18.2.2 FunctionContext インタフェース](#)」を参照してください。

## 18.2.2 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
<b>【非推奨】</b> <code>getServerName()</code>	<b>【非推奨】</b> 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。

メソッド名	説明
【非推奨】 <code>getCacheName()</code>	【非推奨】 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。
【非推奨】 <code>getGroupName()</code>	【非推奨】 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。
【非推奨】 <code>getArgument()</code>	【非推奨】 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を取得します。
<code>getInitConfig()</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情報を取得します。
<code>getServerInfo()</code>	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。
<code>getClusterInfo()</code>	クラスタに関する情報を取得します。

## (4) 【非推奨】 `getServerName()`

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、`ServerInfo` インタフェースの `getName()` を使用してください。

### (a) 機能

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。

### (b) 形式

```
public String getServerName()
```

### (c) 戻り値

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) が返却されます。

## (5) 【非推奨】 getCacheName()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、ClientInfo インタフェースの `getCacheName()` を使用してください。

### (a) 機能

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。

### (b) 形式

```
public String getCacheName()
```

### (c) 戻り値

ユーザファン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) 【非推奨】 getGroupName()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、ClientInfo インタフェースの `getKeyOrGroupName()` を使用してください。

### (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名を取得します。

### (b) 形式

```
public String getGroupName()
```

### (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) 【非推奨】 getArgument()

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、ClientInfo インタフェースの `getFunctionArgument()` を使用してください。

### (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 getArgument()
```

### (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。

データを操作するためのインスタンスについては、「[18.2.8 Store 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。

## (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\(4\) 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。

データを操作するためのインスタンスについては、「[18.2.8 Store 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。

### (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。

ユーザログを出力するためのインタフェースについては、「[18.2.12 UserLogger インタフェース](#)」を参照してください。

## (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、「[18.2.4 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、「[18.2.3 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。

EADS サーバの情報については、「[18.2.5 ServerInfo インタフェース](#)」を参照してください。

## (14) getClusterInfo()

### (a) 機能

クラスタに関する情報を取得します。

### (b) 形式

```
public ClusterInfo getClusterInfo()
```

### (c) 戻り値

クラスタに関する情報が返却されます。

クラスタに関する情報については、「[18.2.6 ClusterInfo 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。

## 18.2.3 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>	ユーザ関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>	ユーザ関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関クションの名称を取得します。

### (4) `getFunctionProperty()`

#### (a) 機能

ユーザ関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タ名を指定します。

#### (d) 戻り値

パラメタ名に対応する指定値が返却されます。

パラメ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` (パラメタ名が `null`)

### (5) `getFunctionPropertyNames()`

#### (a) 機能

ユーザ関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名の一覧を取得します。



## (b) 形式

```
public java.util.Set<String> getFunctionPropertyNames()
```

## (c) 戻り値

ユーザ関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関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関クションの名称が返却されます。

## 18.2.4 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 PID を取得します。
<code>getCacheName()</code>	EADS クライアントで指定したキャッシュ名を取得します。
<code>getKeyOrGroupName()</code>	EADS クライアントで指定した key, またはグループ名を取得します。

メソッド名	説明
<code>getFunctionArgument()</code>	EADS クライアントで指定したユーザ関数の引数を取得します。

## (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 PID を取得します。

取得する PID は、実行した EADS クライアントが出力するログの pid 列と同じ値となります。

### (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 PID が返却されます。

## (6) `getCacheName()`

### (a) 機能

EADS クライアントで指定したキャッシュ名を取得します。

### (b) 形式

```
public String getCacheName()
```

### (c) 戻り値

EADS クライアントで指定したキャッシュ名が返却されます。

## (7) getKeyOrGroupName()

### (a) 機能

EADS クライアントで指定した key, またはグループ名を取得します。

### (b) 形式

```
public String getKeyOrGroupName()
```

### (c) 戻り値

EADS クライアントで指定した key, またはグループ名が返却されます。

EADS クライアントで key, またはグループを指定しなかった場合は, null が返却されます。

## (8) getFunctionArgument()

### (a) 機能

EADS クライアントで指定したユーザ関クションの引数を取得します。

### (b) 形式

```
public Object getFunctionArgument()
```

### (c) 戻り値

EADS クライアントで指定したユーザ関クションの引数が返却されます。

EADS クライアントでユーザ関クションの引数を指定しなかった場合, または null を指定した場合は, null が返却されます。

## 18.2.5 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
getName()	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。
getAddress()	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。
getId()	ユーザファンクションを実行している EADS サーバの、EADS サーバ ID を取得します。
【非推奨】 getCacheNames()	【非推奨】 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。 キャッシュ名の一覧は、ASCII コードの昇順となります。
【非推奨】 getCacheType()	【非推奨】 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。
getEHeapSize()	ユーザファンクションを実行している EADS サーバが、キャッシュ用に確保している Explicit ヒープサイズを取得します。
getEHeapUsageSize()	ユーザファンクションを実行している EADS サーバが、キャッシュ用に確保している Explicit ヒープの使用量を取得します。
【非推奨】 getCacheDataFileSpecifiedSize()	【非推奨】 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。
【非推奨】 getCacheDataFileRemainingSize()	【非推奨】 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。
【非推奨】 getCacheDataFileSpecifiedNumber()	【非推奨】 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。
【非推奨】 getCacheDataFileUnusedNumber()	【非推奨】 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) getId()

### (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、EADS サーバ ID を取得します。

### (b) 形式

```
public int getId()
```

### (c) 戻り値

ユーザファンクションを実行している EADS サーバの、EADS サーバ ID が返却されます。

## (7) 【非推奨】 getCacheNames()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、`ClusterInfo` インタフェースの `getCacheNames()` を使用してください。

### (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。

キャッシュ名の一覧は昇順となります。

### (b) 形式

```
public java.util.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。

## (8) 【非推奨】 `getCacheType()`

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、`ClusterInfo` インタフェースの `getType()` を使用してください。

## (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\(4\) 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)

## (9) 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、キャッシュ用に確保している 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 サーバがキャッシュ用に確保している 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

- `InternalServerError`（EADS サーバの内部エラー）
- `EADsStoreException`（予期しないエラー）

## (10) 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、キャッシュ用に確保している Explicit ヒープの使用量を取得します。

### (b) 形式

```
public long getEHeapUsageSize()  
            throws EADsStoreException
```

### (c) 戻り値

EADS サーバがキャッシュ用に確保している Explicit ヒープの使用量（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

- `InternalServerError`（EADS サーバの内部エラー）

- EADsStoreException (予期しないエラー)

## (11) 【非推奨】 getCacheDataFileSpecifiedSize()

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに CacheInfo インタフェースの `getCacheDataFileSize()` を使用してください。

### (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\(4\) 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)



## (12) 【非推奨】 getCacheDataFileRemainingSize()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、CacheInfo インタフェースの `getRemainingAreaSizeOfWritingCacheDataFile()` を使用してください。

### (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\(4\) 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。

## (13) 【非推奨】 getCacheDataFileSpecifiedNumber()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、CacheInfo インタフェースの `getCacheDataFilesNumber()` を使用してください。

### (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\(4\) 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)

## (14) 【非推奨】 getCacheDataFileUnusedNumber()

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、CacheInfo インタフェースの `getRemainingCacheDataFilesNumber()` を使用してください。

## (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\(4\) 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)
- `InternalServerError` (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。

## 18.2.6 ClusterInfo インタフェース

### (1) 説明

クラスタの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.ClusterInfo
```

### (3) メソッド一覧

`ClusterInfo` 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>getReplicationFactor()</code>	データの多重度を取得します。
<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。 キャッシュ名の一覧は、ASCII コードの昇順となります。
<code>getCacheInfo()</code>	指定したキャッシュの情報を取得します。

メソッド名	説明
<code>getPosition()</code>	指定した key またはグループ名に対応するポジション（ハッシュ値）を取得します。
<code>getRangeId()</code> (ポジション指定)	指定したポジションに対応するレンジのレンジ ID を取得します。
<code>getRangeId()</code> (key またはグループ名指定)	指定した key またはグループ名に対応するレンジのレンジ ID を取得します。
<code>getLocalRangeId()</code>	ユーザファンクションを実行している EADS サーバが、コピー元、またはコピー先として処理するすべてのレンジのレンジ ID の一覧を取得します。
<code>isLocalRange()</code>	指定したレンジが、ユーザファンクションを実行している EADS サーバで、コピー元、またはコピー先としての役割を持つかどうかを取得します。
<code>isLocalMasterRange()</code>	指定したレンジが、ユーザファンクションを実行している EADS サーバで、コピー元としての役割を持つかどうかを取得します。

## (4) `getReplicationFactor()`

### (a) 機能

データの多重度を取得します。

### (b) 形式

```
public int getReplicationFactor()
```

### (c) 戻り値

データの多重度が返却されます。

## (5) `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。

キャッシュ名の一覧は、ASCII コードの昇順となります。

### (b) 形式

```
public 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のキャッシュ名の一覧が返却されます。

## (6) getCacheInfo()

### (a) 機能

指定したキャッシュの情報を取得します。

### (b) 形式

```
public CacheInfo getCacheInfo(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(4) 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のキャッシュに対応する情報を持つ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)

## (7) getPosition()

### (a) 機能

指定した key またはグループ名に対応するポジション (ハッシュ値) を取得します。

### (b) 形式

```
public int getPosition(String keyOrGroupName)
    throws EADsStoreException
```

### (c) パラメータ

keyOrGroupName

key または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」、または「15.2.2(2) グループ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 key またはグループ名に対応するポジション（ハッシュ値）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）

### (8) getRangeId()（ポジション指定）

#### (a) 機能

指定したポジションに対応するレンジのレンジ ID を取得します。

#### (b) 形式

```
public int getRangeId(int position)
    throws EADsStoreException
```

#### (c) パラメータ

position

ポジションを指定します。

#### (d) 戻り値

指定したポジションに対応するレンジのレンジ ID が返却されます。

#### (e) 発生する例外

- EADsStoreException（予期しないエラー）

### (9) getRangeId()（key またはグループ名指定）

#### (a) 機能

指定した key またはグループ名に対応するレンジのレンジ ID を取得します。

## (b) 形式

```
public int getRangeId(String keyOrGroupName)
    throws EADsStoreException
```

## (c) パラメータ

keyOrGroupName

key または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」、または「15.2.2(2) グループ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 key またはグループ名に対応するレンジのレンジ ID 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)

# (10) getLocalRangeId()

## (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、コピー元、またはコピー先として処理するすべてのレンジのレンジ ID の一覧を取得します。

## (b) 形式

```
public int[] getLocalRangeId()
    throws EADsStoreException
```

## (c) 戻り値

ユーザファンクションを実行している EADS サーバが、コピー元、またはコピー先として処理するすべてのレンジのレンジ ID の一覧が返却されます。

## (d) 発生する例外

- EADsStoreException (予期しないエラー)



## (11) isLocalRange()

### (a) 機能

指定したレンジが、ユーザファンクションを実行している EADS サーバで、コピー元、またはコピー先としての役割を持つかどうかを取得します。

### (b) 形式

```
public boolean isLocalRange(int rangeId)
    throws EADsStoreException
```

### (c) パラメータ

rangeId

レンジのレンジ ID を指定します。

### (d) 戻り値

true

指定したレンジは、ユーザファンクションを実行している EADS サーバで、コピー元、またはコピー先としての役割を持ちます。

false

指定したレンジは、ユーザファンクションを実行している EADS サーバで、コピー元、またはコピー先としての役割を持ちません。

### (e) 発生する例外

- EADsStoreException (予期しないエラー)

## (12) isLocalMasterRange()

### (a) 機能

指定したレンジが、ユーザファンクションを実行している EADS サーバで、コピー元としての役割を持つかどうかを取得します。

### (b) 形式

```
public boolean isLocalMasterRange(int rangeId)
    throws EADsStoreException
```

## (c) パラメータ

rangeId

レンジのレンジ ID を指定します。

## (d) 戻り値

true

指定したレンジは、ユーザファンクションを実行している EADS サーバで、コピー元としての役割を持ちます。

false

指定したレンジは、ユーザファンクションを実行している EADS サーバで、コピー元としての役割を持ちません。

## (e) 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)

## 18.2.7 CacheInfo インタフェース

### (1) 説明

キャッシュの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.CacheInfo
```

### (3) メソッド一覧

CacheInfo 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>getType()</code>	キャッシュタイプを取得します。
<code>getCacheDataFileSize()</code>	キャッシュごとに定義したキャッシュデータファイルの、1 ファイル当たりのサイズ (単位: バイト) を取得します。
<code>getCacheDataFilesNumber()</code>	キャッシュごとに定義したキャッシュデータファイルのファイル数を取得します。

メソッド名	説明
<code>getRemainingAreaSizeOfWritingCacheDataFile()</code>	ユーザファンクションを実行している EADS サーバで現在書き込み中であるキャッシュデータファイルの、残りの書き込み可能領域サイズ（単位：バイト）を取得します。
<code>getRemainingCacheDataFilesNumber()</code>	ユーザファンクションを実行している EADS サーバで現在使用できる、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) `getType()`

### (a) 機能

キャッシュタイプを取得します。

### (b) 形式

```
public CacheType getType()
```

### (c) 戻り値

キャッシュタイプが返却されます。

列挙型 `CacheType` については、「[18.2.16 列挙型 CacheType](#)」を参照してください。

## (5) `getCacheDataFileSize()`

### (a) 機能

キャッシュごとに定義したキャッシュデータファイルの、1 ファイル当たりのサイズ（単位：バイト）を取得します。

### (b) 形式

```
public long getCacheDataFileSize()
            throws EADsStoreException
```

### (c) 戻り値

キャッシュごとに定義したキャッシュデータファイルの、1 ファイル当たりのサイズが返却されます。

## (6) `getCacheDataFilesNumber()`

### (a) 機能

キャッシュごとに定義したキャッシュデータファイルの、ファイル数を取得します。

## (b) 形式

```
public int getCacheDataFilesNumber()  
           throws EADsStoreException
```

## (c) 戻り値

キャッシュごとに定義したキャッシュデータファイルの、ファイル数が返却されます。

## (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- EADsStoreException (予期しないエラー)

## (7) getRemainingAreaSizeOfWritingCacheDataFile()

### (a) 機能

ユーザファンクションを実行している EADS サーバで現在書き込み中のキャッシュデータファイルの、残りの書き込み可能領域サイズ (単位: バイト) を取得します。

### (b) 形式

```
public long getRemainingAreaSizeOfWritingCacheDataFile(int rangeId)  
           throws EADsStoreException
```

### (c) パラメータ

rangeId

ユーザファンクションを実行している EADS サーバに属するレンジのレンジ ID を指定します。

### (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

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が確認できた場合でも、EADS サーバ間でデータの配置状態に違いがあるため、同じレンジを扱うほかの EADS サーバ上では空き領域が異なります。

コンパクションを十分に実行することで、キャッシュデータファイルの空き領域の差異が小さくなることは期待できます。ただし、ほかの EADS サーバの空き領域と完全に一致することは保証できません。

## (8) getRemainingCacheDataFilesNumber()

### (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 getRemainingCacheDataFilesNumber(int rangeId)
        throws EADsStoreException
```

### (c) パラメータ

rangeId

ユーザファンクションを実行している EADS サーバに属するレンジのレンジ ID を指定します。

### (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。

### (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

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未使用ファイル数を確認できた場合でも、EADS サーバ間でデータの配置状態の違いがあるため、同じレンジを扱うほかの EADS サーバ上では未使用ファイル数が異なることがあります。

コンパクションを十分に実行することで、ほかの EADS サーバの未使用ファイル数とほぼ一致させることが期待できます。

## 18.2.8 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>createKey()</code>	キャッシュ内で操作可能な key を示すインスタンスを生成します。
<code>createValue()</code>	指定した value をシリアライズして、キャッシュ内で操作可能な value インスタンスを生成します。
<code>createGroup()</code>	キャッシュ内で操作可能なグループを示すインスタンスを生成します。
<b>【非推奨】</b> <code>getGroup()</code>	<b>【非推奨】</b> 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。
<b>【非推奨】</b> <code>getGroup()</code> (グループ名指定)	<b>【非推奨】</b> 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>containsKey()</code> (Key インタフェース指定)	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>【非推奨】</b> <code>containsKey()</code> (文字列指定)	<b>【非推奨】</b> 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。
<code>put()</code>	value を key に関連づけて格納します。
<code>create()</code>	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。
<code>update()</code>	格納済みの value を指定した value に更新します。
<code>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 (comparativeValue) と比較します。value が一致している場合だけ、value を更新します。
<code>get()</code>	指定した key に関連づけられた value を取得します。
<code>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。

メソッド名	説明
getLastUpdateTime()	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。
getKeyCount()	キャッシュ内に格納されている、すべての key の数を取得します。
getGroupCount()	キャッシュ内に格納された key が属するグループのうち、最上位階層のグループ数を取得します。
getGroupNames()	キャッシュ内に格納された key が属するグループのうち、最上位階層のグループ名の一覧を取得します。 グループ名の一覧は、ASCII コードの昇順になります。
<b>【非推奨】</b> getGroupNameSet()	<b>【非推奨】</b> 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。
getEHeapUsageSize() (Key インタフェース指定)	value を格納したことで発生した Explicit ヒープの使用量 (単位: バイト) を取得します。
getEHeapUsageSize() (Group インタフェース指定)	グループに属する key が value を格納するために使用している Explicit ヒープの使用量 (単位: バイト) を取得します。
getDiskUsageSize() (Key インタフェース指定)	value を格納したことで発生したディスクの使用量 (単位: バイト) を取得します。
getDiskUsageSize() (Group インタフェース指定)	グループに属する key が value を格納するために使用しているディスクの使用量 (単位: バイト) を取得します。
calcEHeapUsageSize()	value を格納した場合に発生する Explicit ヒープの使用量 (単位: バイト) を計算します。
calcDiskUsageSize()	value を格納した場合に発生するディスクの使用量 (単位: バイト) を計算します。

## (4) createKey()

### (a) 機能

キャッシュ内で操作可能な key を示すインスタンスを生成します。

### (b) 形式

```
public Key createKey(String key)
    throws EADsStoreException
```

### (c) パラメータ

key

生成するインスタンスで示された 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。

なお、Store インタフェースによる操作の対象外となる key を指定した場合は不正となります。

## (d) 戻り値

Key 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)
- InternalServerErrorException (EADS サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (5) createValue()

### (a) 機能

指定した value をシリアライズして、キャッシュ内で操作可能な value インスタンスを生成します。

### (b) 形式

```
public Value createValue(Object value)
    throws EADsStoreException
```

### (c) パラメータ

value

生成するインスタンスで示された value を指定します。

指定できるデータについては、「[15.2.2\(3\) value として指定できるデータ](#)」を参照してください。

次の場合は不正となります。

- null の場合
- シリアライズできないオブジェクトの場合

## (d) 戻り値

Value インタフェースを実装している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)
- EADsStoreException (予期しないエラー)

## (6) createGroup()

### (a) 機能

キャッシュ内で操作可能なグループを示すインスタンスを生成します。



## (b) 形式

```
public Group createGroup(String groupName)
    throws EADsStoreException
```

## (c) パラメータ

groupName

生成するインスタンスで示されたグループの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\(2\) グループ名として指定できるデータ](#)」を参照してください。

なお、Store インタフェースによる操作の対象外となるグループ名を指定した場合は不正となります。

## (d) 戻り値

指定したグループにアクセス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)

## (7) 【非推奨】 getGroup()

### 参考

このメソッドは非推奨です。このメソッドの代わりに ClientInfo インタフェースの [getKeyOrGroupName\(\)](#)、および Store インタフェースの [createGroup\(\)](#) を使用してください。

## (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。

## (8) 【非推奨】 getGroup() (グループ名指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `createGroup()` を使用してください。

### (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\(2\) グループ名として指定できるデータ](#)」を参照してください。

次の場合は不正となります。

- Store インタフェースによる操作の対象外となるグループ名を指定した場合

### (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)
- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (9) containsKey() (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 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(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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

## (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)

## (10) 【非推奨】 `containsKey()` (文字列指定)

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `containsKey()` (Key インタフェース指定) を使用してください。

## (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、Store インタフェースによる操作の対象外となる 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)
- InternalServerErrorException (EADS サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (11) put()

### (a) 機能

value を key に関連づけて格納します。

すでに value が格納されている場合でも、指定した value で更新されます。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの例外が返却されます。

### (b) 形式

```
public void put(Key key, Value 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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

value

格納する Value インタフェースの value を指定します。

次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

## (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) 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。

すでに value が格納されている場合は、EADsStoreException が返却されます。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。データの削除はすでに value がある場合も実行されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの例外が返却されます。

### (b) 形式

```
public void create(Key key, Value 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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

value

格納する value を指定します。

次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

## (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)

## (13) 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 を更新します)。

value が格納されていない場合は、EADsStoreException が返却されます。

### (b) 形式

```
public void update(Key key, Value value)
    throws EADsStoreException
```

### (c) パラメータ

key

格納する 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。

次の場合は不正となります。

- null の場合

- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

value

格納する Value インタフェースの value を指定します。

次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

#### (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)

## (14) 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 を更新します。

次の場合は、EADsStoreException が返却されます。

- value が格納されているが、条件として指定した value (comparativeValue) と一致しない場合
- value が格納されていない場合

### (b) 形式

```
public void replace(Key key, Value value, Value comparativeValue)
    throws EADsStoreException
```

### (c) パラメータ

key

格納する 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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

value

格納する Value インタフェースの value を指定します。  
次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

comparativeValue

比較する Value インタフェースの value を指定します。  
次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

#### (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)

## (15) get()

### (a) 機能

指定した key に関連づけられた value を取得します。

### (b) 形式

```
public Value 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。  
次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

### (d) 戻り値

指定した 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)

## (16) 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。

### (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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

### (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)

## (17) getLastUpdateTime()

### (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。  
次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

## (d) 戻り値

データの最終更新時刻（1970/1/1 00:00:00 (UTC) 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）

# (18) getKeyCount()

## (a) 機能

キャッシュ内に格納されている、すべての key の数を取得します。

key と value は 1 対 1 で対応しているため、取得した key の数は、キャッシュ内に格納されている value の数に相当します。

## (b) 形式

```
public int getKeyCount()
    throws EADsStoreException
```

## (c) 戻り値

キャッシュ内に格納されている、すべての key の数が返却されます。

## (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)

## (19) getGroupCount()

### (a) 機能

キャッシュ内に格納された key が属するグループのうち、最上位階層のグループ数を取得します。

### (b) 形式

```
public int getGroupCount()  
           throws EADsStoreException
```

### (c) 戻り値

キャッシュ内に格納された key が属するグループのうち、最上位階層のグループ数が返却されます。

### (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)

## (20) getGroupNames()

### (a) 機能

キャッシュ内に格納された key が属するグループのうち、最上位階層のグループ名の一覧を取得します。

グループ名の一覧は、ASCII コードの昇順になります。

### (b) 形式

```
public String[] getGroupNames()  
                throws EADsStoreException
```

### (c) 戻り値

キャッシュ内に格納された key が属するグループのうち、最上位階層のグループ名の一覧が返却されます。

### (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)

## (21) 【非推奨】 getGroupNameSet()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `getGroupNames()` を使用してください。

### (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 java.util.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)

## (22) getEHeapUsageSize() (Key インタフェース指定)

### (a) 機能

value を格納したことで発生した Explicit ヒープの使用量 (単位: バイト) を取得します。

### (b) 形式

```
public long getEHeapUsageSize(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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

## (d) 戻り値

value を格納したことで発生した Explicit ヒープの使用量が返却されます。

## (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)

## (23) getEHeapUsageSize() (Group インタフェース指定)

### (a) 機能

グループに属する key が value を格納するために使用している Explicit ヒープの使用量 (単位: バイト) を取得します。

### (b) 形式

```
public long getEHeapUsageSize(Group group)
    throws EADsStoreException
```

### (c) パラメータ

group

key が属する Group インタフェースのグループ名を指定します。  
次の場合は不正となります。

- null の場合
- 不正な Group インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

### (d) 戻り値

グループに属する key が value を格納するために使用している Explicit ヒープの使用量が返却されます。

### (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)

## (24) getDiskUsageSize() (Key インタフェース指定)

### (a) 機能

value を格納したことで発生したディスクの使用量 (単位: バイト) を取得します。

### (b) 形式

```
public long getDiskUsageSize(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。  
次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

### (d) 戻り値

value を格納したことで発生したディスクの使用量が返却されます。

### (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) getEHeapUsageSize() (Group インタフェース指定)

### (a) 機能

グループに属する key が value を格納するために使用しているディスクの使用量 (単位: バイト) を取得します。

### (b) 形式

```
public long getEHeapUsageSize(Group group)
    throws EADsStoreException
```

## (c) パラメータ

group

key が属する Group インタフェースのグループ名を指定します。

次の場合は不正となります。

- null の場合
- 不正な Group インタフェースのオブジェクトの場合
- Store インタフェースによる操作の対象外となる key の場合

## (d) 戻り値

グループに属する key が value を格納するために使用しているディスクの使用量が返却されます。

## (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)

## (26) calcEHeapUsageSize()

### (a) 機能

value を格納した場合に発生する Explicit ヒープの使用量 (単位: バイト) を計算します。

### (b) 形式

```
public long calcEHeapUsageSize(Value value)
    throws EADsStoreException
```

## (c) パラメータ

value

格納した Value インタフェースの value を指定します。

次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

## (d) 戻り値

value を格納した場合に発生する Explicit ヒープの使用量が返却されます。

## (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)

## (27) calcDiskUsageSize()

### (a) 機能

value を格納した場合に発生するディスクの使用量 (単位: バイト) を計算します。

### (b) 形式

```
public long calcDiskUsageSize(Key key, Value value)
    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。

次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

value

格納した Value インタフェースの value を指定します。

次の場合は不正となります。

- null の場合
- 不正な Value インタフェースのオブジェクトの場合

### (d) 戻り値

value を格納した場合に発生するディスクの使用量が返却されます。

## (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)



## 18.2.9 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>equals()</code>	指定したオブジェクトが同一のグループを示す Group インスタンスかどうかを判定します。
<code>getPosition()</code>	グループに対応するポジション（ハッシュ値）を取得します。
<code>createKey()</code>	指定したグループに属する key を生成します。
<code>createGroup()</code>	指定したグループに属する直下の階層のグループを生成します。
<code>firstKey()</code>	指定したグループに属する格納済みの key のうち、ASCII コード順で先頭となる key を取得します。
<code>lastKey()</code>	指定したグループに属する格納済みの key のうち、ASCII コード順で末尾となる key 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>keyIterator()</code> (Key インタフェース指定)	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、グループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する 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>descendingKeyIterator()</code> (Key インタフェース指定)	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、グループに属する key のうち、ASCII コードの降順で、指定した key の次に位置する key となります。

メソッド名	説明
【非推奨】 higherKeyIterator()	【非推奨】 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。
【非推奨】 lowerDescendingKeyIterator()	【非推奨】 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。
【非推奨】 put() (文字列指定)	【非推奨】 value を key に関連づけて格納します。
【非推奨】 put() (Key インタフェース指定)	【非推奨】 value を key に関連づけて格納します。
【非推奨】 create()	【非推奨】 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。
【非推奨】 update() (文字列指定)	【非推奨】 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)。
【非推奨】 update() (Key インタフェース指定)	【非推奨】 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)。
【非推奨】 replace() (文字列指定)	【非推奨】 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)。
【非推奨】 replace() (Key インタフェース指定)	【非推奨】 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)。
【非推奨】 get() (文字列指定)	【非推奨】 指定した key に関連づけられた value を取得します。
【非推奨】 get() (Key インタフェース指定)	【非推奨】 指定した key に関連づけられた value を取得します。
【非推奨】 remove() (文字列指定)	【非推奨】 指定した key、および key に関連づけられた value を削除します。
【非推奨】 remove() (Key インタフェース指定)	【非推奨】 指定した key、および key に関連づけられた value を削除します。
【非推奨】 getLastUpdateTime() (文字列指定)	【非推奨】 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。
【非推奨】 getLastUpdateTime() (Key インタフェース指定)	【非推奨】 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。
getKeyCount()	指定したグループに属する key の数を取得します。
getGroupCount()	直下の階層のグループ数を取得します。
getGroupNames()	直下の階層のグループ名の一覧を取得します。 グループ名の一覧は、ASCII コードの昇順になります。
getGroupLayerNames()	直下の階層のグループ階層名の一覧を取得します。 グループ階層名の一覧は、ASCII コードの昇順になります。
【非推奨】 getGroupNameSet()	【非推奨】 グループ名の一覧を ASCII コードの昇順で取得します。

メソッド名	説明
【非推奨】 <code>getValueUsageSize()</code> (文字列指定)	【非推奨】 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。
【非推奨】 <code>getValueUsageSize()</code> (Key インタフェース指定)	【非推奨】 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。
【非推奨】 <code>getValueUsageSize()</code>	【非推奨】 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) equals()

### (a) 機能

指定したオブジェクトが同一のグループを示す Group インスタンスかどうかを判定します。

### (b) 形式

```
public boolean equals(Object obj)
```

### (c) パラメータ

obj

比較対象のオブジェクトを指定します。

### (d) 戻り値

true

指定したオブジェクトは、同一のグループを示す Group インスタンスです。

false

指定したオブジェクトは、同一のグループを示す Group インスタンスではありません。

## (6) getPosition()

### (a) 機能

グループに対応するポジション（ハッシュ値）を取得します。

### (b) 形式

```
public int getPosition()  
    throws EADsStoreException
```

### (c) 戻り値

グループに対応するポジション（ハッシュ値）が返却されます。

### (d) 発生する例外

- UserOperationException（ユーザの操作不正）
- EADsStoreException（予期しないエラー）

## (7) createKey()

### (a) 機能

指定したグループに属する key を生成します。

### (b) 形式

```
public Key createKey(String element)  
    throws EADsStoreException
```

### (c) パラメータ

element

グループ名に連結する要素名を指定します。

次の場合は不正となります。

- 要素名を連結したあとの key が、「15.2.2(1) key として指定できるデータ」に示す条件を満たさない場合
- コロン (:) を含む場合

### (d) 戻り値

Key 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)
- EADsStoreException (予期しないエラー)

## (8) createGroup()

### (a) 機能

指定したグループの直下の階層のグループを生成します。

### (b) 形式

```
public Group createGroup(String groupLayerName)
    throws EADsStoreException
```

### (c) パラメータ

groupLayerName

グループの階層名を指定します。

次の場合は不正となります。

- グループの階層名を連結したあとのグループ名が、「15.2.2(1) key として指定できるデータ」に示す条件を満たさない場合
- コロン (:) を含む場合

### (d) 戻り値

指定したグループにアクセス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)
- EADsStoreException (予期しないエラー)

## (9) firstKey()

### (a) 機能

指定したグループに属する格納済みの key のうち、ASCII コード順で先頭となる key を取得します。

### (b) 形式

```
public Key firstKey()
    throws EADsStoreException
```

## (c) 戻り値

指定したグループに属する格納済みの key のうち、ASCII コード順で先頭となる key が返却されます。

指定したグループに属する key が存在しない場合は、null 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)

## (10) lastKey()

### (a) 機能

指定したグループに属する格納済みの key のうち、ASCII コード順で末尾となる key を取得します。

### (b) 形式

```
public Key lastKey()  
    throws EADsStoreException
```

### (c) 戻り値

指定したグループに属する格納済みの key のうち、ASCII コード順で末尾となる key が返却されます。

指定したグループに属する key が存在しない場合は、null 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)

## (11) 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。

## (12) keyIterator() (Key インタフェース指定)

### (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、グループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key となります。

### (b) 形式

```
public java.util.Iterator<Key> keyIterator(Key previousKey)
    throws EADsStoreException
```

### (c) パラメータ

previousKey

反復子の開始位置を示す直前の key を指定します。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

### (d) 戻り値

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、削除されていることがあります。

## (13) 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。

## (14) descendingKeyIterator() (Key インタフェース指定)

### (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、グループに属する key のうち、ASCII コードの降順で、指定した key の次に位置する key となります。

### (b) 形式

```
public java.util.Iterator<Key> descendingKeyIterator(Key previousKey)  
    throws EADsStoreException
```



## (c) パラメータ

previousKey

反復子の開始位置を示す直前の key を指定します。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

## (d) 戻り値

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。

## (15) 【非推奨】 higherKeyIterator()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、[keyIterator\(\)](#) (Key インタフェース指定) を使用してください。

## (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。

### (16) 【非推奨】 lowerDescendingKeyIterator()

#### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、[descendingKeyIterator\(\)](#) (Key インタフェース指定) を使用してください。

#### (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。

## (17) 【非推奨】 put() (文字列指定)

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `put()` を使用してください。

## (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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの例外が返却されます。

## (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\(3\) 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)

## (18) 【非推奨】 put() (Key インタフェース指定)

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `put()` を使用してください。

### (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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの例外が返却されます。

### (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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

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\(3\) 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)

## (19) 【非推奨】 create()

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `create()` を使用してください。

## (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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。データの削除はすでに value がある場合も実行されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときの例外が返却されます。

## (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\(3\) 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)

## (20) 【非推奨】 `update()` (文字列指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `update()` を使用してください。

### (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\(3\) 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)

## (21) 【非推奨】 update() (Key インタフェース指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `update()` を使用してください。

### (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。

次の場合は不正となります。

- `null` の場合
- 不正な Key インタフェースのオブジェクトの場合

`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\(3\) 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)

## (22) 【非推奨】 replace() (文字列指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `replace()` を使用してください。

### (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\(3\) 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\(3\) 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)



## (23) 【非推奨】 replace() (Key インタフェース指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `replace()` を使用してください。

### (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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

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\(3\) 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\(3\) 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)

## (24) 【非推奨】 get() (文字列指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `get()` を使用してください。

### (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)
- `InternalServerException` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (25) 【非推奨】 get() (Key インタフェース指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `get()` を使用してください。

## (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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

## (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)
- InternalServerException (EADS サーバの内部エラー)
- EADsStoreException (予期しないエラー)

## (26) 【非推奨】 remove() (文字列指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの [remove\(\)](#) を使用してください。

## (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)
- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (27) 【非推奨】 `remove()` (Key インタフェース指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `remove()` を使用してください。

## (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。

次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

## (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)

## (28) 【非推奨】 getLastUpdateTime() (文字列指定)

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `getLastUpdateTime()` を使用してください。

## (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)

## (29) 【非推奨】 getLastUpdateTime() (Key インタフェース指定)

### 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `getLastUpdateTime()` を使用してください。

### (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。  
次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

### (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)
- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (30) getKeyCount()

### (a) 機能

このグループに属する key の数を取得します。

### (b) 形式

```
public int getKeyCount()
    throws EADsStoreException
```

### (c) 戻り値

このグループに属する key の数が返却されます。

### (d) 発生する例外

- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (31) `getGroupCount()`

### (a) 機能

このグループの直下の階層のグループ数を取得します。

### (b) 形式

```
public int getGroupCount()  
           throws EADsStoreException
```

### (c) 戻り値

このグループの直下の階層のグループ数が返却されます。

### (d) 発生する例外

- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (32) `getGroupNames()`

### (a) 機能

直下の階層のグループ名の一覧を取得します。

グループ名の一覧は、ASCII コードの昇順になります。

### (b) 形式

```
public String[] getGroupNames()  
               throws EADsStoreException
```

### (c) 戻り値

直下の階層のグループ名の一覧が返却されます。

## (d) 発生する例外

- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (33) `getGroupLayerNames()`

### (a) 機能

直下の階層のグループ階層名の一覧を取得します。

グループ階層名の一覧は、ASCII コードの昇順になります。

### (b) 形式

```
public String[] getGroupLayerNames()  
                throws EADsStoreException
```

### (c) 戻り値

直下の階層のグループ階層名の一覧が返却されます。

## (d) 発生する例外

- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (34) 【非推奨】 `getGroupNameSet()`

### 参考

このメソッドは非推奨です。このメソッドの代わりに、`Group` インタフェースの `getGroupNames()` を使用してください。

### (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 java.util.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

- `InternalServerError` (EADS サーバの内部エラー)
- `EADsStoreException` (予期しないエラー)

## (35) 【非推奨】 `getValueUsageSize()` (文字列指定)

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、`Store` インタフェースの `getEHeapUsageSize()` (`Key` インタフェース指定) を使用してください。

## (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)

## (36) 【非推奨】 getValueUsageSize() (Key インタフェース指定)

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `getEHeapUsageSize()` (Key インタフェース指定) を使用してください。

### (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。  
次の場合は不正となります。

- null の場合
- 不正な Key インタフェースのオブジェクトの場合

### (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)

## (37) 【非推奨】 getValueUsageSize()

### ■ 参考

このメソッドは非推奨です。このメソッドの代わりに、Store インタフェースの `getEHeapUsageSize()` (Group インタフェース指定) を使用してください。

## (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

- InternalServerErrorException（EADS サーバの内部エラー）
- EADsStoreException（予期しないエラー）

## 18.2.10 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
<code>toString()</code>	key を取得します。
<code>equals()</code>	指定したオブジェクトが同一の key を示す Key インスタンスかどうかを判定します。
<code>getPosition()</code>	key に対応するポジション（ハッシュ値）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。

## (5) equals()

### (a) 機能

指定したオブジェクトが同一の key を示す Key インスタンスかどうかを判定します。

### (b) 形式

```
public boolean equals(Object obj)
```

### (c) パラメータ

obj

比較対象のオブジェクトを指定します。

### (d) 戻り値

true

指定したオブジェクトは、同一の key を示す Key インスタンスです。

false

指定したオブジェクトは、同一の key を示す Key インスタンスではありません。

## (6) getPosition()

### (a) 機能

key に対応するポジション (ハッシュ値) を取得します。

## (b) 形式

```
public int getPosition()  
    throws EADsStoreException
```

## (c) 戻り値

key に対応するポジション（ハッシュ値）が返却されます。

## (d) 発生する例外

- UserOperationException（ユーザの操作不正）
- EADsStoreException（予期しないエラー）

## 18.2.11 Value 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 に関連づけて格納される value 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.Value
```

### (3) メソッド一覧

Value 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()	String 型の value を取得します。
equals()	指定したオブジェクトが同一の value を示す Value インスタンスかどうかを判定します。
getObject()	Object 型の value を取得します。
size()	キャッシュに格納する際の内部形式である、シリアライズ後の value のサイズを取得します。

### (4) toString()

#### (a) 機能

String 型の value を取得します。

Value インタフェースの getObject() で取得する Object 型の value の、Object クラスの toString() と同じ文字列を取得します。

## (b) 形式

```
public String toString()
```

## (c) 戻り値

String 型の value が返却されます。

## (d) 注意事項

Value インタフェースの getObject() がデシリアライズできないで失敗する場合は、Value インスタンスの Object クラスの toString() と同じ戻り値が返却されます。

# (5) equals()

## (a) 機能

指定したオブジェクトが同一の value を示す Value インスタンスかどうかを判定します。

## (b) 形式

```
public boolean equals(Object obj)
```

## (c) パラメータ

obj

比較対象のオブジェクトを指定します。

## (d) 戻り値

true

指定したオブジェクトは、同一の value を示す Value インスタンスです。

false

指定したオブジェクトは、同一の value を示す Value インスタンスではありません。

## (e) 注意事項

同一のオブジェクトかどうかは、シリアライズ後のバイト配列が等しいかどうかで判定されます。

## (6) getObject()

### (a) 機能

Object 型の value を取得します。

### (b) 形式

```
public Object getObject()  
    throws EADsStoreException
```

### (c) 戻り値

Object 型の value が返却されます。

### (d) 発生する例外

- UserOperationException (ユーザの操作不正)
- EADsStoreException (予期しないエラー)

## (7) size()

### (a) 機能

キャッシュに格納する際の内部形式である、シリアライズ後の value のサイズを取得します。

### (b) 形式

```
public int size()
```

### (c) 戻り値

キャッシュに格納する際の内部形式である、シリアライズ後の value のサイズ (単位: バイト) が返却されます。

### (d) 注意事項

このメソッドで返却されるサイズは、value を格納する際に value だけが使用する領域のサイズです。

value を格納する際は、value 以外のデータも領域を使用する場合があります。そのため、value の格納で使用される領域サイズを計算したい場合は、次の API を使用してください。

- Store インタフェースの [calcEHeapUsageSize\(\)](#)
- Store インタフェースの [calcDiskUsageSize\(\)](#)

## 18.2.12 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
<code>log()</code> (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。
<code>log()</code> (形式 2)	メッセージテキストだけを出力します。
<code>putStackTrace()</code> (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。
<code>putStackTrace()</code> (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。

## 18.2.13 EADsStoreException クラス

### (1) 説明

データの操作に関連する処理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.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
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	データを更新する API の処理状態※
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。	<p>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。</p> <p>次の要因が考えられます。</p> <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>	—

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	データを更新する API の処理状態※
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	InternalServerErrorException	EADS サーバで、分類できない内部エラーが発生しました。	EADS サーバで内部エラーが発生しました。 通常、このエラーコードが返却されることはありません。次の原因でエラー要因を詳細化できなかった場合に、このエラーコードが返却されます。 <ul style="list-style-type: none"> <li>非推奨となった古いバージョンの API を使用した。</li> <li>予期しない問題によって、EADS サーバの内部エラーを分類できなかった。</li> </ul>	△
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。	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。	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。	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	InternalServerErrorException	update()実行時、格納されている key が存在しなかったため、	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。	×

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	データを更新する API の処理状態※
	ETHOD_KEY_NOT_EXIST		value を格納できませんでした。		
4200	EAD_ERROR_SERVER_CACHE	InternalServerException	キャッシュに対する操作を実行できませんでした。	ユーザファンクションを実行中の EADS サーバで何らかの問題が発生しているため、キャッシュを操作できない状態になっています。操作を中止して、EADS サーバの状態を確認してください。	×
4220	EAD_ERROR_SERVER_CACHE_CLUSTER_NOT_AVAILABLE	InternalServerException	クラスタの状態がクラスタ動作不能のため、キャッシュに対する操作を実行できませんでした。	ほかの EADS サーバまたはネットワーク障害などの要因で、クラスタの状態がクラスタ動作不能になっているため、キャッシュを操作できませんでした。操作を中止して、クラスタを構成するすべての EADS サーバの状態を確認してください。	×
4300	EAD_ERROR_SERVER_CACHE_BEFORE_REPLICATION	InternalServerException	キャッシュに対する操作中に内部エラーが発生しましたが、データの多重化は発生しませんでした。	キャッシュに対する操作中に、ユーザファンクションを実行中の EADS サーバで内部エラーが発生しました。データの多重化は発生していないため、ほかの正常な EADS サーバへの影響はありません。EADS サーバが縮退したあと、正常な EADS サーバ上で同じ操作から処理を再開できます。	×
4310	EAD_ERROR_SERVER_CACHE_AFTER_REPLICATION	InternalServerException	キャッシュに対する操作中に EADS サーバで内部エラーが発生し、データ更新処理が失敗しました。	キャッシュに対する操作中に、ユーザファンクションを実行中の EADS サーバで内部エラーが発生しました。データの多重化はすでに実行されているため、問題が発生した EADS サーバが縮退したあとに正常なサーバ上でデータが更新された状態から操作を再開できます。	△
4800	EAD_ERROR_SERVER_LIMIT_EXTERNAL_MEMORY	InternalServerException	データを格納するメモリの容量が不足しています。	データを格納するメモリ (Explicit ヒープ) の容量が不足しているため、処理を実行できませんでした。	×
4810	EAD_ERROR_SERVER_LIMIT_CACHE_FILE	InternalServerException	データを格納するキャッシュファイル	データを格納するキャッシュファイルの容量が不足している	×

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	データを更新する API の処理状態※
			の容量が不足しています。	ため、処理を実行できませんでした。	
4820	EAD_ERROR_SERVER_LIMIT_KV_COUNT	InternalServerException	格納できる key の数が上限に達しています。	格納できる key の数が上限に達しているため、処理を実行できませんでした。	×
4830	EAD_ERROR_SERVER_LIMIT_KEY_VALUE_LENGTH	InternalServerException	指定した key, グループ名, または value のサイズがクラスタに格納できる最大サイズを超えています。	指定した key, グループ名, または value のサイズがクラスタに格納できる最大サイズを超えているため、処理を実行できませんでした。	×
4840	EAD_ERROR_SERVER_LIMIT_EXTERNAL_MEMORY_TEMPORARILY	InternalServerException	データを格納するメモリの容量が一時的に不足しています。	リバランス処理中は、EADS サーバでデータを格納するメモリ (Explicit ヒープ) の容量が一時的に減少します。そのため、処理を実行できませんでした。 リトライする場合は、1 度ユーザ関数の実行を終了して、EADS クライアントからユーザ関数を再度実行してください。	×
4850	EAD_ERROR_SERVER_LIMIT_KV_COUNT_TEMPORARILY	InternalServerException	格納できる key の数が一時的に上限に達しています。	リバランス処理中は、EADS サーバに格納できる key の数の上限が一時的に減少します。そのため、処理を実行できませんでした。 リトライする場合は、1 度ユーザ関数の実行を終了して、EADS クライアントからユーザ関数を再度実行してください。	×

#### 注※

put(), remove()などのデータを更新する API の実行時にエラーコードが出力された場合に、データが更新されているかどうかを表しています。

記号の意味を次に示します。

△：データが更新されているかどうかは不明です。処理が完了したかどうか、確認してください。

×：データは更新されていません。

－：put(), remove()などのデータを更新する API の実行時に、このエラーコードは出力されません。

## (d) 注意事項

非推奨の API を使用した場合、エラーコードが詳細に分類されないことがあります。

## 18.2.14 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
```

## 18.2.15 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
```

## 18.2.16 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



# 19

## 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。

## 19.1 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。

### 19.1.1 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、「[19.1.1\(10\) 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。

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) key またはグループ指定による実行方法

key または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 key またはグループ名と, 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;
}
```

戻り値として返却されたメモリ領域を AP で解放する必要がある関数を次の表に示します。次の表に示す各関数の解放対象（戻り値）が NULL でない場合は、戻り値を使用しなくなったタイミングで、メモリ領域を解放してください。

表 19-1 メモリ領域を解放する必要がある関数および解放対象となる戻り値

項番	関数名	解放対象となる戻り値
1	ead_get_cache_name()	キャッシュ名として返却される char 型のポインタ
2	ead_put_all()	ead_put_all_results 構造体の failure_info メンバ
3	ead_get()	ead_value_element 構造体の value メンバ
4	ead_get_group()	ead_get_group_results 構造体の次に示すメンバ

項番	関数名	解放対象となる戻り値
		<ul style="list-style-type: none"> <li>key_value_array メンバ</li> <li>key_value_array メンバ内の ead_key_value_pair 構造体の key メンバ</li> <li>key_value_array メンバ内の ead_key_value_pair 構造体内の ead_value_element 構造体の value メンバ</li> </ul>
5	ead_get_all()	ead_get_all_results 構造体の次に示すメンバ <ul style="list-style-type: none"> <li>values メンバ</li> <li>values メンバ内の ead_value_element 構造体の value メンバ</li> <li>failure_info メンバ</li> </ul>
6	ead_remove_all()	ead_remove_all_results 構造体の failure_info メンバ
7	ead_get_group_names()	ead_group_names 構造体の group_names メンバ
8	ead_get_group_keys()	ead_keys 構造体の keys メンバ
9	ead_get_node_keys()	
10	ead_get_group_first_key()	key として返却された char 型のポインタ
11	ead_get_node_first_key()	
12	ead_get_group_next_key()	
13	ead_get_node_next_key()	
14	ead_execute_function()	
15	ead_execute_function_rt()	
16	ead_execute_node_function()	
17	ead_execute_node_function_rt()	
18	ead_get_nodelist()	ead_nodelist 構造体の nodes メンバ
19	ead_get_slave_nodelist()	
20	ead_add_cluster_eventlistener()	
21	ead_add_node_eventlistener()	

## 19.2 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。

### 19.2.1 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。

### 19.2.2 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。

### 19.2.3 データ操作に関する留意事項

#### (1) 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  
ます。
  - 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()などの実行時に、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、「[19.1.1\(10\) 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。

## (2) データの一括操作に関する留意事項

データの一括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\_all()は無条件に value を更新しま  
す。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によっ  
て操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費さ  
れることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイム  
アウト時間を適切に設計してください。

- `ead_put_all()`などの実行時に、戻り値として返却された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、「[19.1.1\(10\) 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。
- 複数の key を指定する一括操作の実行時に、ある key に対する操作に失敗した場合やクラスタ構成が変更された場合は、一括操作の処理が終了し、その時点で未実行の操作がキャンセルされます。一括操作中に、ユーザの操作（例：スケールアウト処理、リバランス処理、または復旧処理）によってクラスタ構成が変更された場合も、未実行の操作がキャンセルされます。  
操作に失敗した key および失敗した要因については、各関数の戻り値から判断してください。また、操作がキャンセルされた key については、`EAD_ERROR_CLIENT_BATCH_CANCEL` のエラーコードで確認できます。
- 複数の key を指定する一括操作の実行時に、key に対する操作がすべて失敗した場合は、`EAD_ERROR_BATCH_FAILED_ALL` が返却されます。
- 複数の key を指定する一括操作の実行時に、key に対する操作が部分的に失敗した場合は、`EAD_ERROR_BATCH_FAILED_PART` が返却されます。

## 19.2.4 イベントリスナに関する留意事項

イベントリスナに関する留意事項を次に示します。

### (1) イベントリスナの実装に関する留意事項

- イベントが発生した場合、クラスタイベントリスナおよびノードイベントリスナは、EADS クライアントに追加した順序 (`ead_add_cluster_eventlistener()` および `ead_add_node_eventlistener()` で追加した順序) で実行されます。
- イベントリスナ内では、次の関数を呼び出さないでください。
  - `ead_init_client()`
  - `ead_init_client_n()`
  - `ead_terminate_client()`
- クラスタイベントリスナの引数に指定された `ead_cluster_eventcontext` 構造体を指すポインタは、クラスタイベントリスナの実行後に自動で解放されます。そのため、クラスタイベントリスナの終了後に、このポインタを参照しないでください。クラスタイベントリスナの終了後に、このポインタを参照した場合、動作は保証されません。
- ノードイベントリスナの引数に指定された `ead_node_eventcontext` 構造体を指すポインタは、ノードイベントリスナの実行後に自動で解放されます。そのため、ノードイベントリスナの終了後に、このポインタを参照しないでください。ノードイベントリスナの終了後に、このポインタを参照した場合、動作は保証されません。
- イベントリスナ内には、EADS クライアントの正常な処理に影響を及ぼす処理（例えば、リソースを大量に消費する処理）を実装しないでください。

- 同一 EADS クライアント内のすべてのイベントリスナは、1つのスレッドから呼び出されて動作します。イベントリスナの動作中は、EADS サーバから受信したデータの保持用にメモリを占有します。そのため、イベントリスナ内には、長時間動作する処理を実装しないでください。

## (2) イベントの通知に関する留意事項

- EADS サーバが初期化状態 (initialized) 以降の状態の場合に、EADS サーバから EADS クライアントにイベントが通知されます。ただし、EADS サーバとの接続の再接続に成功した場合は、EADS サーバの状態に関係なく、その EADS サーバがまだ通知していないイベントが通知されます。
- EADS クライアントがクラスタ内のどの EADS サーバとも通信できなかった場合は、イベントが通知されません。ただし、EADS クライアントは EADS サーバとの接続の再接続を繰り返すため、再接続に成功すると、そのあとでイベントが通知されます。
- 半数以上の EADS サーバが縮退している場合、イベントが通知されないことがあります。
- EADS クライアントと EADS サーバ間の接続で、切断と再接続を繰り返した場合や、通信が遅延している場合には、イベントの通知が遅延します。
- EADS クライアントと EADS サーバ間の接続が切断している間に、複数の EADS サーバが縮退した場合、EADS サーバのイベント再送時間 (サーバ定義の `eads.server.eventListener.notify.retry.timeout` パラメタの指定値) を過ぎて再送用のデータが破棄されると、イベントを通知する順序が逆転するおそれがあります。
- EADS クライアントと EADS サーバ間の接続が切断している間に、ある EADS サーバが縮退後に復旧した場合、EADS サーバのイベント再送時間 (サーバ定義の `eads.server.eventListener.notify.retry.timeout` パラメタの指定値) を過ぎて再送用のデータが破棄されると、何もイベントが通知されないおそれがあります。
- 同じプロセス内で、EADS クライアントを終了 (`ead_terminate_client()`) したあとに EADS クライアントを開始 (`ead_init_client()` または `ead_init_client_n()`) した場合は、すでに終了した EADS クライアントの再送用のデータは破棄されています。そのため、EADS クライアントを開始したあとのイベントだけが通知されます。
- 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。そのため、EADS クライアントへのイベントの通知が完了していなくても、クラスタの終了 (`eztool stop`) や EADS サーバの縮退などの処理は完了します。

## 19.2.5 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。

## 19.2.6 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。

## 19.2.7 その他の留意事項

EADS が提供する C 言語のクライアントライブラリは、SIGPIPE (シグナル) の動作を無視 (SIG\_IGN) に設定します。クライアント AP のプログラム内で SIGPIPE の動作を無視 (SIG\_IGN) 以外に再設定しないでください。

## 19.3 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-eads<sup>※</sup>  
hntr2-eads-t
- 64 ビット版  
hcc-eads-x64<sup>※</sup>  
hntr2-eads-te64
- 32 ビット版・64 ビット版共通  
eads

### 注※

既存のクライアント AP の互換用として、32 ビット版は hcc-4.1.1、64 ビット版は hccx64-4.1.1 も指定できます。

# 20

## 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。

## 20.1 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 20-1 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。	eads.h
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_get_cache_name()</code>	指定したキャッシュのハンドルに関連づけられたキャッシュ名を取得します。	
6	<code>ead_terminate_client()</code>	EADS クライアントを終了します。	
7	<code>ead_put()</code>	value を key に関連づけて格納します。	
8	<code>ead_put_array_value()</code>	複数の value を連結して、key に関連づけて格納します。	
9	<code>ead_put_all()</code>	複数の key と value をキャッシュに一括格納します。	
10	<code>ead_create()</code>	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。	
11	<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)。	

項番	関数名	機能	ヘッダファイル
12	<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)。	
13	<code>ead_get()</code>	指定した key に関連づけられた value を取得します。	
14	<code>ead_get_all()</code>	指定した key の一覧に関連づけられた value を一括取得します。	
15	<code>ead_get_group()</code>	指定したグループおよびその下位階層のグループに属する key に関連づけられた value を一括取得します。	
16	<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。	
17	<code>ead_remove_all()</code>	指定した key の一覧に関連づけられた value を一括削除します。	
18	<code>ead_remove_group()</code>	指定したグループに属する key および value を、下位階層のグループに属する key および value も含めて一括削除します。	
19	<code>ead_remove_node()</code>	指定した EADS サーバをコピー元 EADS サーバとする key および value を一括削除します。	
20	<code>ead_get_group_names()</code>	指定した EADS サーバに格納されている最上位階層のグループ名の一覧を取得します。 グループ名の一覧は、ASCII コードの昇順になります。	
21	<code>ead_get_group_keys()</code>	指定したグループに属する key の一覧を、下位階層のグループに属する key も含めて取得します。 key の一覧は、ASCII コードの昇順になります。	



項番	関数名	機能	ヘッダファイル
22	<code>ead_get_node_keys()</code>	指定した EADS サーバに格納されている key の一覧を取得します。 key の一覧は、ASCII コードの昇順になります。	
23	<code>ead_get_group_count()</code>	指定した EADS サーバに格納されている最上位階層のグループの数を取得します。	
24	<code>ead_get_group_key_count()</code>	指定したグループに属する key の数を取得します。指定したグループの下位階層のグループに属する key も、取得される key の数に含まれます。	
25	<code>ead_get_node_key_count()</code>	指定した EADS サーバに格納されている key の数を取得します。	
26	<code>ead_get_group_first_key()</code>	指定したグループに属する key のうち、ASCII コードの昇順で先頭となる key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。	
27	<code>ead_get_node_first_key()</code>	指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で先頭となる key を取得します。	
28	<code>ead_get_group_next_key()</code>	指定したグループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。	
29	<code>ead_get_node_next_key()</code>	指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。	
30	<code>ead_execute_function()</code>	指定した key またはグループから実行する EADS サーバを決定して、ユーザファンクションを実行します。	
31	<code>ead_execute_function_rt()</code>	指定した key またはグループから実行する EADS サーバを	

項番	関数名	機能	ヘッダファイル
		決定して、ユーザファンクションを実行します。また、受信タイムアウト時間を指定します。	
32	<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。	
33	<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。	
34	<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。	
35	【非推奨】 <code>ead_get_node()</code>	【非推奨】 指定した key またはグループを格納する、本来のコピー元 EADS サーバの情報を取得します。	
36	<code>ead_get_slave_nodelist()</code>	指定した EADS サーバに格納されているデータの、本来のコピー先 EADS サーバの情報を取得します。	
37	<code>ead_get_current_master_node()</code>	指定した key (またはグループ) を現時点で格納する、コピー元 EADS サーバの情報を取得します。	
38	<code>ead_get_original_master_node()</code>	指定した key (またはグループ) を格納する、本来のコピー元 EADS サーバの情報を取得します。	
39	<code>ead_add_cluster_eventlistener()</code>	クラスイベントリスナを追加します。	
40	<code>ead_add_node_eventlistener()</code>	ノードイベントリスナを追加します。	
41	<code>ead_remove_cluster_eventlistener()</code>	クラスイベントリスナを削除します。	
42	<code>ead_remove_node_eventlistener()</code>	ノードイベントリスナを削除します。	

## 20.1.1 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。

クライアント定義の `eads.client.eventListener.enable` パラメタに `true` を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、ead\_terminate\_client() 実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが 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、「[20.2 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。

## 20.1.2 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。

クライアント定義の eads.client.eventListener.enable パラメータに true を指定していた場合は、次のように動作します。

- 新たにイベントリスナ用のコネクションが生成されて、EADS サーバとの接続に成功すると、クラスタ構成情報の通知を受け取れる状態になります。生成されたコネクションは、ead\_terminate\_client()実行時に削除されます。
- クラスタイベントリスナおよびノードイベントリスナを実行するためのスレッドが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(5) 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、「20.2 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。

## 20.1.3 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\(4\) 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、「[20.2 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。

## 20.1.4 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、「[20.2 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。

## 20.1.5 ead\_get\_cache\_name() (キャッシュ名の取得)

### (1) 機能

指定したキャッシュのハンドルに関連づけられたキャッシュ名を取得します。

ead\_stop\_cache()でアクセスを終了したキャッシュのキャッシュ名も取得できます。

### (2) 形式

```
#include <eads.h>
char* ead_get_cache_name
(
    const EAD_CACHE    *cp,          /* In */
    int                *error_code   /* Out */
);
```

### (3) 引数

cp

キャッシュ名を取得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。

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、「[20.2 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

キャッシュのハンドルに関連づけられたキャッシュ名が返却されます。

キャッシュ名の取得で問題が発生した場合、NULL が返却されます。

## (5) 注意事項

戻り値として返却されたキャッシュ名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。詳細については、「19.1.1(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。

## 20.1.6 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。

クライアント定義の `eads.client.eventListener.enable` パラメータに `true` を指定していた場合は、次のように動作します。

- クラスタイベントリスナおよびノードイベントリスナが存在する場合は、すべてのクラスタイベントリスナおよびノードイベントリスナを削除します。
- すべての EADS サーバ間で、イベントリスナ用の接続を切断します。
- 実行中のイベントリスナが存在する場合は、そのイベントリスナの終了を待ちます。

### (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、「[20.2 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。

## 20.1.7 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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときのエラーの内容を示す value を error\_code に設定します。

### (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、「[20.1.43 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、「20.2 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。

## 20.1.8 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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときのエラーの内容を示す value を error\_code に設定します。

### (2) 形式

```
#include <eads.h>
void ead_put_array_value
(
    const EAD_CACHE           *cp,           /* In */
    const char                *key,         /* In */
    size_t                    array_length, /* In */
    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、「[20.2 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を超えない値を設定してください。

## 20.1.9 ead\_put\_all() (key と value の一括格納)

### (1) 機能

複数の key と value をキャッシュに一括格納します。

引数には ead\_key\_value\_pair 構造体の配列を指定し、それぞれの key と value を関連づけてキャッシュに格納します。

同じ key を複数指定した場合は、指定した順にそれぞれ実行されます。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってしきい値を超える件数分のデータを削除してから、複数の key と value を一括格納します。既存のデータを更新する場合もデータは削除されます。

データ自動削除機能でデータの削除に失敗した場合、失敗した件数分は value の格納を実行しません。

### (2) 形式

```
#include <eads.h>
ead_put_all_results ead_put_all
(
    const EAD_CACHE           *cp,           /* In */
    size_t                    array_length,  /* In */
    const ead_key_value_pair  *key_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。

array\_length

格納する ead\_key\_value\_pair 構造体の配列の要素数を指定します。

ead\_key\_value\_pair 構造体の形式と説明については、「[20.1.44 ead\\_key\\_value\\_pair 構造体 \(key と value の組み合わせ\)](#)」を参照してください。

key\_value\_array

格納する ead\_key\_value\_pair 構造体の配列の先頭を指すポインタを指定します。

ead\_key\_value\_pair 構造体の形式と説明については、「[20.1.44 ead\\_key\\_value\\_pair 構造体 \(key と value の組み合わせ\)](#)」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、「[20.2 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 と value の一括格納の結果 (ead\_put\_all\_results 構造体) が返却されます。

- 一括格納に成功している場合  
success\_operation\_number メンバが、引数に指定した key の一覧中の key の数と同じ値になり、failure\_info メンバが NULL になります。
- 一括格納に一部またはすべて失敗した場合  
failure\_info メンバに、failure\_operation\_number メンバが示す個数分、失敗した操作の情報が返却されます。データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しなかった key は、データの削除に失敗したときの情報が返却されます。
- key ごとの処理以外でエラーが発生した場合（引数の不正やメモリ領域の不足などが発生した場合）  
ead\_put\_all\_results 構造体の数値型のメンバは 0、ポインタ型のメンバは NULL になります。

ead\_put\_all\_results 構造体の形式と説明については、「[20.1.47 ead\\_put\\_all\\_results 構造体 \(ead\\_put\\_all\(\) の実行結果\)](#)」を参照してください。

### (5) 注意事項

- value の一括格納に失敗した場合、次のエラーコードが返却されます。
  - 一括格納にすべて失敗したとき  
EAD\_ERROR\_BATCH\_FAILED\_ALL

- 一括格納に一部失敗したとき

EAD\_ERROR\_BATCH\_FAILED\_PART

- 戻り値として返却された、key と value の一括格納の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。詳細については、「[19.1.1\(10\) 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。
- 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\_key\_value\_pair 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。
- キャッシュの操作が部分的に失敗した場合、戻り値からどの操作が失敗したか判別して、必要に応じてこの関数を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## 20.1.10 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。

データ自動削除機能を使用している場合、key 数がしきい値の上限に達しているときは、データの削除ポリシーに従ってデータを削除してから、value を key に関連付けて格納します。データの削除は、指定した key がキャッシュに格納されている場合も実行します。

データ自動削除機能でデータの削除に失敗した場合、value の格納を実行しないで remove が失敗したときのエラーの内容を示す value を error\_code に設定します。

### (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、「[20.1.43 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、「[20.2 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。

## 20.1.11 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、「[20.1.43 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、「[20.2 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。

## 20.1.12 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、「[20.1.43 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、「[20.1.43 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、「[20.2 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。

## 20.1.13 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、「[20.2 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、「[20.1.43 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。詳細については、「[19.1.1\(10\) 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。

## 20.1.14 ead\_get\_all() (value の一括取得)

### (1) 機能

指定した key の一覧に関連づけられた value を一括取得します。

同じ key を複数指定した場合は、key の一覧に指定した順にそれぞれ実行します。

### (2) 形式

```
#include <eads.h>
ead_get_all_results ead_get_all
(
    const EAD_CACHE      *cp,          /* In */
    const ead_keys       *keys,       /* 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。

#### keys

取得する value に関連づけられた key の一覧 (`ead_keys` 構造体) 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。

`ead_keys` 構造体の形式と説明については、「[20.1.45 ead\\_keys 構造体 \(複数の key\)](#)」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、「[20.2 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 の一覧に関連づけられた value の一括取得の結果 (`ead_get_all_results` 構造体) が返却されます。

- 一括取得に成功している場合  
values メンバに、引数に指定した key の一覧に対応する value が返却されます。
- 一括取得に一部失敗した場合  
failure\_info メンバに、failure\_operation\_number メンバに示す個数分、失敗した操作の情報が返却されます。  
取得に失敗した key に対応する values メンバの位置には、value が NULL である `ead_value_element` 構造体が返却されます。
- 一括取得にすべて失敗した場合  
values\_length メンバに 0、values メンバには NULL が格納されます。
- key ごとの処理以外でエラーが発生した場合 (引数の不正やメモリ領域の不足などが発生した場合)  
`ead_get_all_results` 構造体の数値型のメンバは 0、ポインタ型のメンバは NULL になります。

`ead_get_all_results` 構造体の形式と説明については、「[20.1.48 ead\\_get\\_all\\_results 構造体 \(ead\\_get\\_all\(\)の実行結果\)](#)」を参照してください。

## (5) 注意事項

- value の一括取得に失敗した場合、次のエラーコードが返却されます。
  - 一括取得にすべて失敗したとき  
EAD\_ERROR\_BATCH\_FAILED\_ALL
  - 一括取得に一部失敗したとき  
EAD\_ERROR\_BATCH\_FAILED\_PART

- 戻り値として返却された、value の一括取得の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。詳細については、「[19.1.1\(10\) 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。
- キャッシュの操作が部分的に失敗した場合、戻り値からどの操作が失敗したか判別して、必要に応じてこの関数を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## 20.1.15 ead\_get\_group() (グループ指定による value の一括取得)

### (1) 機能

指定したグループおよびその下位階層のグループに属する key に関連づけられた value を一括取得します。

### (2) 形式

```
#include <eads.h>
ead_get_group_results ead_get_group
(
    const EAD_CACHE    *cp,           /* In */
    const char         *group_name,   /* 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。

**group\_name**  
value を取得する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\(2\) グループ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、「[20.2 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

指定したグループおよびその下位階層のグループに属する key に関連づけられた value の一括取得の結果 (ead\_get\_group\_results 構造体) が返却されます。

- 一括取得に一部またはすべて成功した場合  
value の取得が成功した分の key, および引数に指定した key の一覧に対応する value が, key\_value\_array メンバに返却されます。
- 一括取得にすべて失敗した場合  
ead\_get\_group\_results 構造体の数値型のメンバは 0, ポインタ型のメンバは NULL になります。

ead\_get\_group\_results 構造体の形式と説明については、「[20.1.49 ead\\_get\\_group\\_results 構造体 \(ead\\_get\\_group\(\)の実行結果\)](#)」を参照してください。

## (5) 注意事項

- 戻り値として返却された, value の一括取得の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。詳細については、「[19.1.1\(10\) 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。
- キャッシュの操作が部分的に失敗した場合, キャッシュの操作の実行結果を確認して, 必要に応じて再実行してください。
- キャッシュの操作中に排他処理はされません。そのため, 一括操作中に, ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合, EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合, 操作の完了までに時間が掛かることがあります。そのため, タイムアウト時間を適切に設計してください。

## 20.1.16 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、「20.2 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。

## 20.1.17 ead\_remove\_all() (value の一括削除)

### (1) 機能

指定した key の一覧に関連づけられた value を一括削除します。

同じ key を複数指定した場合は、key の一覧に指定した順にそれぞれ実行されます。

### (2) 形式

```
#include <eads.h>
ead_remove_all_results ead_remove_all
(
    const EAD_CACHE      *cp,          /* In */
    const ead_keys       *keys,       /* In */
    int                  *error_code  /* Out */
);
```

### (3) 引数

cp

削除する 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。

keys

削除する key の一覧（ead\_keys 構造体）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。

ead\_keys 構造体の形式と説明については、「[20.1.45 ead\\_keys 構造体 \(複数の key\)](#)」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、「[20.2 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 の一覧に関連づけられた value の一括削除の結果 (ead\_remove\_all\_results 構造体) が返却されます。

- 一括削除にすべて成功した場合  
success\_operation\_number メンバが、引数に指定した key の数と同じ値になり、failure\_info メンバが NULL になります。
- 一括削除に一部またはすべて失敗した場合  
failure\_info メンバに、failure\_operation\_number メンバに示す個数分、失敗した操作の情報が返却されます。
- key ごとの処理以外でエラーが発生した場合 (引数の不正やメモリ領域の不足など)  
ead\_remove\_all\_results 構造体の数値型のメンバは 0、ポインタ型のメンバは NULL になります。

ead\_remove\_all\_results 構造体の形式と説明については、「[20.1.50 ead\\_remove\\_all\\_results 構造体 \(ead\\_remove\\_all\(\)の実行結果\)](#)」を参照してください。

## (5) 注意事項

- value の一括削除に失敗した場合、次のエラーコードが返却されます。
  - 一括削除にすべて失敗したとき  
EAD\_ERROR\_BATCH\_FAILED\_ALL
  - 一括削除に一部失敗したとき  
EAD\_ERROR\_BATCH\_FAILED\_PART
- 戻り値として返却された、value の一括削除の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。詳細については、「[19.1.1\(10\) 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。
- 削除に失敗した場合、削除対象のデータが残っているおそれがあります。そのため、エラーコードからエラーの要因を調査して、どの操作が失敗したか判別して対処してください。必要に応じて、ead\_remove\_all() (value の一括削除) を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。

- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## 20.1.18 ead\_remove\_group() (グループ指定による value の一括削除)

### (1) 機能

指定したグループに属する key および value を、下位階層のグループに属する key および value も含めて一括削除します。

key および value の一括削除中に何らかの要因で削除に失敗した場合は、失敗の要因を示すエラーコードが返却されます。

### (2) 形式

```
#include <eads.h>
void ead_remove_group
(
    const EAD_CACHE    *cp,           /* In */
    const char         *group_name,   /* In */
    int                *error_code    /* Out */
);
```

### (3) 引数

**cp**  
削除する 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。

**group\_name**  
削除する key および value が属する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\(2\) グループ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、「[20.2 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

- 削除に失敗した場合、削除対象のデータが残っているおそれがあります。そのため、実行結果を確認して対処してください。必要に応じて、`ead_remove_group()`（グループ指定による value の一括削除）を再度実行してください。
- キャッシュの操作が部分的に失敗した場合、戻り値からどの操作が失敗したか判別して、必要に応じて再実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## 20.1.19 ead\_remove\_node() (EADS サーバ指定による value の一括削除)

### (1) 機能

指定した EADS サーバをコピー元 EADS サーバとする key および value を一括削除します。

key および value の一括削除中に何らかの要因で削除に失敗した場合は、失敗の要因を示すエラーコードが返却されます。

### (2) 形式

```
#include <eads.h>
void ead_remove_node
(
    const EAD_CACHE      *cp,           /* In */
    const ead_node       *target_node,  /* In */
    int                  *error_code    /* Out */
);
```

### (3) 引数

`cp`

削除する 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。

`target_node`

一括削除を実行する 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が格納するアドレス情報 (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。

#### 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、「[20.2 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

- 削除に失敗した場合、削除対象のデータが残っているおそれがあります。そのため、実行結果を確認して対処してください。必要に応じて、ead\_remove\_node() (EADS サーバ指定による value の一括削除) を再度実行してください。
- キャッシュの操作中に排他処理はされません。そのため、一括操作中に、ほかのキャッシュ操作によって操作対象の value が変更されることがあります。
- 大量のデータを一括操作する場合、EADS クライアントおよび EADS サーバでメモリが大量に消費されることがあります。
- 大量のデータを一括操作する場合、操作の完了までに時間が掛かることがあります。そのため、タイムアウト時間を適切に設計してください。

## 20.1.20 ead\_get\_group\_names() (最上位階層のグループ名の一覧の取得)

### (1) 機能

指定した EADS サーバに格納されている最上位階層のグループ名の一覧を取得します。

グループ名の一覧は、ASCII コードの昇順になります。

### (2) 形式

```
#include <eads.h>
ead_group_names ead_get_group_names
(
    const EAD_CACHE      *cp,           /* In */
    const ead_node       *target_node,  /* In */
    int                  *error_code    /* Out */
);
```

### (3) 引数

`cp`

グループ名の一覧を取得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`

取得先の 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、「[20.1.54 ead\\_node 構造体（EADS サーバ指定のユーザ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` 構造体が格納するアドレス情報（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、「[20.2 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 EADS サーバに格納されている最上位階層のグループ名の一覧（`ead_group_names` 構造体）が返却されます。

次の場合、`ead_group_names` 構造体の `group_names` メンバに `NULL` が返却されます。

- 指定した EADS サーバに、グループに属する `key` が存在しない場合
- エラーによってグループ名の一覧の取得に失敗した場合

`ead_group_names` 構造体の形式と説明については、「[20.1.46 ead\\_group\\_names 構造体（複数のグループ名）](#)」を参照してください。

### (5) 注意事項

- 戻り値として返却された `ead_group_names` 構造体の `group_names` 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。詳細については、「[19.1.1\(10\) 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。
- 指定した EADS サーバに存在するグループの数が多いほど、取得に時間が掛かったり、リソースの使用量が増加したりするおそれがあります。

## 20.1.21 ead\_get\_group\_keys() (グループ指定による key の一覧の取得)

### (1) 機能

指定したグループに属する key の一覧を、下位階層のグループに属する key も含めて取得します。

key の一覧は、ASCII コードの昇順になります。

### (2) 形式

```
#include <eads.h>
ead_keys ead_get_group_keys
(
    const EAD_CACHE      *cp,           /* In */
    const char           *group_name,   /* In */
    int                  *error_code    /* Out */
);
```

### (3) 引数

**cp**

key の一覧を取得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\(2\) グループ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、「[20.2 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

指定したグループに属する key の一覧 (ead\_keys 構造体) が ASCII コードの昇順で返却されます。

次の場合、ead\_keys 構造体の keys メンバに NULL が返却されます。

- 指定したグループに属する key が存在しない場合
- エラーによって key の一覧の取得に失敗した場合

ead\_keys 構造体の形式と説明については、「[20.1.45 ead\\_keys 構造体 \(複数の key\)](#)」を参照してください。

## (5) 注意事項

- 戻り値として返却された ead\_keys 構造体の keys 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。詳細については、「[19.1.1\(10\) 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。
- 指定した EADS サーバに存在する key の数が多いほど、取得に時間が掛かったり、リソースの使用量が増加したりするおそれがあります。

## 20.1.22 ead\_get\_node\_keys() (EADS サーバ指定による key の一覧の取得)

### (1) 機能

指定した EADS サーバに格納されている key の一覧を取得します。

key の一覧は、ASCII コードの昇順になります。

### (2) 形式

```
#include <eads.h>
ead_keys ead_get_node_keys
(
    const EAD_CACHE      *cp,           /* In */
    const ead_node       *target_node, /* In */
    int                  *error_code   /* Out */
);
```

### (3) 引数

cp

key の一覧を取得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

取得先の 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が格納するアドレス情報 (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。

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、「[20.2 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 EADS サーバに格納されている key の一覧 (ead\_keys 構造体) が ASCII コードの昇順で返却されます。

次の場合、ead\_keys 構造体の keys メンバは NULL が返却されます。

- 指定した EADS サーバにコピー元の key が存在しない場合
- エラーによって key の一覧の取得に失敗した場合

ead\_keys 構造体の形式と説明については、「[20.1.45 ead\\_keys 構造体 \(複数の key\)](#)」を参照してください。

## (5) 注意事項

- 戻り値として返却された ead\_keys 構造体の keys 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。詳細については、「[19.1.1\(10\) 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。
- 指定した EADS サーバに存在する key の数が多いほど、取得に時間が掛かったり、リソースの使用量が増加したりするおそれがあります。

## 20.1.23 ead\_get\_group\_count() (最上位階層グループ数の取得)

### (1) 機能

指定した EADS サーバに格納されている最上位階層のグループの数を取得します。

### (2) 形式

```
#include <eads.h>
size_t ead_get_group_count
(
    const EAD_CACHE      *cp,           /* In */
    const ead_node       *target_node, /* In */
    int                  *error_code    /* Out */
);
```

### (3) 引数

`cp`

グループ数を取得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`

取得先の 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザ関数で使用されるオブジェクト\)](#)」を参照してください。

指定した `ead_node` 構造体が格納するアドレス情報（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、「[20.2 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 EADS サーバに格納されている最上位階層のグループの数が返却されます。

エラーによってグループ数の取得に失敗した場合、0 が返却されます。

## 20.1.24 ead\_get\_group\_key\_count() (グループ指定による key の数の取得)

### (1) 機能

指定したグループに属する key の数を取得します。指定したグループの下位階層のグループに属する key も、取得される key の数に含まれます。

### (2) 形式

```
#include <eads.h>
size_t ead_get_group_key_count
(
    const EAD_CACHE    *cp,           /* In */
    const char         *group_name,   /* In */
    int                *error_code    /* Out */
);
```

### (3) 引数

cp

key の数を取得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\(2\) グループ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、「[20.2 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

指定したグループに属する key の数が返却されます。

エラーによって key の数の取得に失敗した場合、0 が返却されます。

## 20.1.25 ead\_get\_node\_key\_count() (EADS サーバ指定による key の数の取得)

### (1) 機能

指定した EADS サーバに格納されている key の数を取得します。

### (2) 形式

```
#include <eads.h>
size_t ead_get_node_key_count
(
    const EAD_CACHE    *cp,           /* In */
    const ead_node     *target_node, /* In */
    int                 *error_code   /* Out */
);
```

### (3) 引数

cp

key の数を取得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

取得先の 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザ関数で使用されるオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が格納するアドレス情報 (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。

## 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、「[20.2 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 EADS サーバに格納されている key の数が返却されます。

エラーによって key の数の取得に失敗した場合、0 が返却されます。

## 20.1.26 ead\_get\_group\_first\_key() (グループ指定による先頭の key の取得)

### (1) 機能

指定したグループに属する key のうち、ASCII コードの昇順で先頭となる key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。

### (2) 形式

```
#include <eads.h>
char* ead_get_group_first_key
(
    const EAD_CACHE    *cp,           /* In */
    const char         *group_name,   /* In */
    int                *error_code    /* Out */
);
```

### (3) 引数

cp

key を取得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\(2\) グループ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、「[20.2 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

指定したグループに属する key のうち、ASCII コードの昇順で先頭となる key が返却されます。

次の場合、NULL が返却されます。

- 指定したグループに属する key が存在しない場合
- エラーによって key の数の取得に失敗した場合

## (5) 注意事項

key の取得に成功した場合（戻り値が 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。詳細については、「[19.1.1\(10\) 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。

## 20.1.27 ead\_get\_node\_first\_key() (EADS サーバ指定による先頭の key の取得)

### (1) 機能

指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で先頭となる key を取得します。

### (2) 形式

```
#include <eads.h>
char* ead_get_node_first_key
(
    const EAD_CACHE    *cp,           /* In */
    const ead_node     *target_node, /* In */
    int                 *error_code  /* Out */
);
```

### (3) 引数

`cp`

`key` を取得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`

取得先の 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザ関数で使用されるオブジェクト\)](#)」を参照してください。

指定した `ead_node` 構造体が格納するアドレス情報 (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、「[20.2 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 EADS サーバに格納されている `key` のうち、ASCII コードの昇順で先頭となる `key` が返却されます。

次の場合、NULL が返却されます。

- 指定した EADS サーバにコピー元の `key` が存在しない場合
- エラーによって `key` の数の取得に失敗した場合

### (5) 注意事項

`key` の取得に成功した場合 (戻り値が 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。詳細については、「[19.1.1\(10\) 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。

## 20.1.28 ead\_get\_group\_next\_key() (グループ指定による次の key の取得)

### (1) 機能

指定したグループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。

なお、接続先の EADS サーバに存在しない key を指定した場合も同様に、指定した key の次に位置する key を取得します。

### (2) 形式

```
#include <eads.h>
char* ead_get_group_next_key
(
    const EAD_CACHE    *cp,           /* In */
    const char         *group_name,   /* In */
    const char         *key,         /* In */
    int                *error_code    /* Out */
);
```

### (3) 引数

`cp`

key を取得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\(2\) グループ名として指定できるデータ](#)」を参照してください。

`key`

基準とする 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、「[20.2 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 EADS サーバに格納されている key のうち、ASCII コードの昇順で、指定した key の次に位置する key が返却されます。

次の場合、NULL が返却されます。

- 指定した key の次に位置する key が存在しない場合
- エラーによって key の取得に失敗した場合

## (5) 注意事項

- key の取得に成功した場合（戻り値が 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。詳細については、「[19.1.1\(10\) 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。
- この関数では、指定されたグループを基に接続先の EADS サーバが決定されます。
- 接続先以外の EADS サーバの縮退、復旧、追加（スケールアウト処理）、および位置の変更（リバランス処理）によって、指定したグループに属する key を格納する EADS サーバが変更されることがあります。そのため、`ead_get_group_first_key()`を実行して先頭となる key を取得したときとは、異なる EADS サーバに接続する可能性があります。
- EADS サーバ上でグループに対して排他処理をしないため、`ead_get_group_first_key()`または `ead_get_group_next_key()`の実行後、他プロセスからの操作によって、グループに属する key が挿入、または削除されることがあります。

## 20.1.29 ead\_get\_node\_next\_key() (EADS サーバ指定による次の key の取得)

### (1) 機能

指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。

なお、接続先の EADS サーバに存在しない key を指定した場合も同様に、指定した key の次に位置する key を取得します。

### (2) 形式

```
#include <eads.h>
char* ead_get_node_next_key
(
    const EAD_CACHE    *cp,           /* In */
    const ead_node     *target_node,  /* In */
    const char         *key,          /* In */
    int                *error_code    /* Out */
);
```

### (3) 引数

`cp`

`key` を取得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`

取得先の 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザ関数で使用されるオブジェクト\)](#)」を参照してください。

指定した `ead_node` 構造体が格納するアドレス情報 (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。

`key`

基準とする `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、「[20.2 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 EADS サーバに格納されている `key` のうち、ASCII コードの昇順で、指定した `key` の次に位置する `key` が返却されます。

次の場合、NULL が返却されます。

- 指定した `key` の次に位置する `key` が存在しない場合
- エラーによって `key` の取得に失敗した場合

### (5) 注意事項

- `key` の取得に成功した場合 (戻り値が 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。詳細については、「[19.1.1\(10\) 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。
- 接続先以外の EADS サーバの縮退、復旧、追加 (スケールアウト処理)、および位置の変更 (リバランス処理) によって、指定したグループに属する `key` を格納する EADS サーバが変更される場合があります。

- EADS サーバ上でグループに対して排他処理をしないため、`ead_get_node_first_key()`または`ead_get_node_next_key()`の実行後、他プロセスからの操作によって、グループに属する key が挿入、または削除されることがあります。

## 20.1.30 `ead_execute_function()` (key またはグループ指定によるユーザ関数の実行)

### (1) 機能

指定した key またはグループから実行する EADS サーバを決定して、ユーザ関数を実行します。

### (2) 形式

```
#include <eads.h>
ead_object ead_execute_function
(
    const EAD_CACHE      *cp,                /* In */
    const char           *key_or_group_name, /* 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。

**key\_or\_group\_name**  
key または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」、または「15.2.2(2) グループ名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**  
ユーザ関数に渡す引数 (`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、「[20.1.52 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、「[20.2 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、「[20.1.52 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。詳細については、「[19.1.1\(10\) 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。

## 20.1.31 ead\_execute\_function\_rt() (key またはグループ, および受信タイムアウト指定によるユーザファンクションの実行)

### (1) 機能

指定した key またはグループから実行する EADS サーバを決定して、ユーザファンクションを実行します。また、受信タイムアウト時間を指定します。

ead\_execute\_function\_rt()の実行中は、クライアント定義の `eads.client.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           *key_or_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。

**key\_or\_group\_name**  
key または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](#)」, または「[15.2.2\(2\) グループ名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、「[20.1.52 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.client.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、「[20.2 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、ユーザ関シヨンの実行結果 (`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、「[20.1.52 ead\\_object 構造体 \(ユーザ関シヨンで使用するオブジェクト\)](#)」を参照してください。

`ead_object` 構造体の `object_size` メンバは、ユーザ関シヨ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。

- ユーザ関シヨ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、ユーザ関シヨンの実行結果を取得できなかった場合
- ユーザ関シヨンの実行結果として `null` が返却された場合
- ユーザ関シヨ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、ユーザ関シヨ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。詳細については、「[19.1.1\(10\) 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。

## 20.1.32 ead\_execute\_node\_function() (EADS サーバ指定によるユーザ関シヨンの実行)

### (1) 機能

EADS サーバを指定して、ユーザ関シヨンを実行します。

### (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。

**target\_node**  
ユーザファンクションを実行する 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が格納するアドレス情報 (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、「[20.1.52 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、「[20.2 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、「[20.1.52 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。詳細については、「[19.1.1\(10\) 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。

## 20.1.33 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クションの実行)

### (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、ユーザ関クションを実行します。

ead\_execute\_node\_function\_rt()の実行中は、クライアント定義の `eads.client.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。

#### target\_node

ユーザファンクションを実行する 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が格納するアドレス情報 (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。

#### 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、「[20.1.52 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.client.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、「[20.2 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、「[20.1.52 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。詳細については、「[19.1.1\(10\) 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。

## 20.1.34 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、「[20.2 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、「20.1.53 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。詳細については、「19.1.1(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。
- 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。
- この関数の実行時には、EADS サーバの接続可否は考慮されません。そのため、取得した情報には、縮退状態などで接続できない EADS サーバが含まれていることがあります。取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうか、ead\_node 構造体の is\_enable メンバで接続可否を確認してください。

ead\_node 構造体の形式と説明については、「20.1.54 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)」を参照してください。

## 20.1.35 【非推奨】 ead\_get\_node() (指定した key の本来のコピー元 EADS サーバの情報取得)

### 注意事項

この関数は非推奨です。この関数の代わりに、[ead\\_get\\_original\\_master\\_node\(\)](#) を使用してください。

## (1) 機能

指定した key またはグループを格納する、本来のコピー元 EADS サーバの情報を取得します。

本来のコピー元 EADS サーバとは、クラスタを構成するすべての EADS サーバが正常に接続できる場合に、指定した key またはグループのコピー元のデータを格納する EADS サーバのことです。

## (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、「[20.2 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 またはグループを格納する、本来のコピー元 EADS サーバの情報（ead\_node 構造体）が返却されます。

異常終了時には、ead\_node 構造体の各メンバに 0 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。
- クラスタ構成に変更がなければ、EADS サーバの接続可否に関係なく、常に同じ EADS サーバの情報が取得されます。
- 取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうか、ead\_node 構造体の is\_enable メンバで接続可否を確認してください。  
ead\_node 構造体の形式と説明については、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

## 20.1.36 ead\_get\_slave\_nodelist() (データの本来のコピー先 EADS サーバの情報取得)

### (1) 機能

指定した EADS サーバに格納されているデータの、本来のコピー先 EADS サーバの情報を取得します。



本来のコピー先 EADS サーバとは、クラスタを構成するすべての EADS サーバが正常に接続できる場合に、指定した EADS サーバ（コピー元 EADS サーバ）に格納されているデータの、コピー先となる EADS サーバのことです。

## (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

コピー元のデータを格納する 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、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

指定した ead\_node 構造体が格納するアドレス情報（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、「[20.2 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

本来のコピー先 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、「[20.1.53 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。詳細については、「[19.1.1\(10\) 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。
- 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。
- この関数の実行時には、指定した EADS サーバ、および情報の取得対象となる EADS サーバの接続可否は考慮されません。そのため、取得した情報には、縮退状態などで接続できない EADS サーバの情報が含まれていることがあります。

## 20.1.37 ead\_get\_current\_master\_node() (現時点のコピー元 EADS サーバの情報取得)

### (1) 機能

指定した key (またはグループ) を現時点で格納する、コピー元 EADS サーバの情報を取得します。

### (2) 形式

```
#include <eads.h>
ead_node ead_get_current_master_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

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。

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、「[20.2 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) を現時点で格納する, コピー元 EADS サーバの情報が返却されます。

異常終了時, およびコピー元 EADS サーバが縮退または停止している場合, ead\_node 構造体の各メンバに 0 が返却されます。

## (5) 注意事項

この関数の実行時には, EADS サーバへの通信は行われません。そのため, 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。

## 20.1.38 ead\_get\_original\_master\_node() (本来のコピー元 EADS サーバの情報取得)

### (1) 機能

指定した key (またはグループ) を格納する, 本来のコピー元 EADS サーバの情報を取得します。

本来のコピー元 EADS サーバとは, クラスタを構成するすべての EADS サーバが正常に接続できる場合に, 指定した key (またはグループ) のマスタとなるデータ (コピー元となるデータ) を格納する EADS サーバのことです。現時点のコピー元 EADS サーバとは異なることがあります。

クラスタ構成に変更がなければ, EADS サーバの接続可否に関係なく, 常に同じ EADS サーバの情報が取得されます。

### (2) 形式

```
#include <eads.h>
ead_node ead_get_original_master_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

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。

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、「20.2 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) を格納する本来のコピー元 EADS サーバの情報が返却されます。

異常終了時には、ead\_node 構造体の各メンバに 0 が返却されます。

## (5) 注意事項

- この関数の実行時には、EADS サーバへの通信は行われません。そのため、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。
- 取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうか、ead\_node 構造体の is\_enable メンバを使用して確認してください。ead\_node 構造体の is\_enable メンバについては、「20.1.54(2) メンバの説明」を参照してください。

## 20.1.39 ead\_add\_cluster\_eventlistener() (クラスタイベントリスナの追加)

### (1) 機能

クラスタイベントリスナを追加します。

### (2) 形式

```
#include <eads.h>
ead_nodelist ead_add_cluster_eventlistener
(
    const EAD_CACHE_MANAGER      *cmp,                /* In */
    const char                    *cluster_eventlistener_name, /* In */
    const ead_cluster_eventlistener *cluster_eventlistener, /* In */
    void                          *userdata,          /* 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。

#### cluster\_eventlistener\_name

追加するクラスタイベントリスナの名前を指定します。

指定できるデータ型	文字数	説明
'\0'を終端記号とする文字列 (char *)	1~1,024	<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です。</li><li>長さ0の文字列 ('\0'だけ) を指定できません。</li></ul>

同一クライアント内のほかのクラスタイベントリスナまたはノードイベントリスナと同じ名称を指定した場合は, エラーとなります。

#### cluster\_eventlistener

追加するクラスタイベントリスナを指すポインタを指定します。

#### userdata

追加するクラスタイベントリスナとのデータの伝達に使用するポインタを指定します。

任意のアドレスを指定できます。

追加するクラスタイベントリスナとのデータの伝達にポインタを使用しない場合は, 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, 「[20.2 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

クラスタイベントリスナの追加が終了した時点での, 接続先 EADS サーバの情報一覧 (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, 「[20.1.53 ead\\_nodelist 構造体 \(EADS サーバ情報\)](#)」を参照してください。

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。

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。詳細については、「[19.1.1\(10\) 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。
- 関数実行と同時にクラスタ構成情報が変更された場合、縮退済みの EADS サーバや、追加済みの EADS サーバに対しては、cluster\_eventlistener に設定した関数を呼び出せません。関数実行後に返却された EADS サーバの情報を確認してください。
- この関数の実行時には、EADS サーバの接続可否は考慮されません。そのため、取得した情報には、接続できない EADS サーバ（例：EADS サーバの縮退）が含まれていることがあります。取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうかを、ead\_node 構造体の is\_enable メンバで確認してください。

ead\_node 構造体の形式と説明については、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

## 20.1.40 ead\_add\_node\_eventlistener() (ノードイベントリスナ追加)

### (1) 機能

ノードイベントリスナを追加します。

### (2) 形式

```
#include <eads.h>
ead_nodelist ead_add_node_eventlistener
(
    const EAD_CACHE_MANAGER      *cmp,                /* In */
    const char                    *node_eventlistener_name, /* In */
    const int                     *nodeid,             /* In */
    const ead_node_eventtype      *node_eventtype,     /* In */
    const ead_node_eventlistener  *node_eventlistener, /* In */
    void                          *userdata,          /* 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。

node\_eventlistener\_name

追加するノードイベントリスナの名称を指定します。

指定できるデータ型	文字数	説明
'\0'を終端記号とする文字列 (char *)	1~1,024	<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です。</li> <li>長さ 0 の文字列 ('\0'だけ) を指定できません。</li> </ul>

同一クライアント内のほかのクラスタイベントリスナまたはノードイベントリスナと同じ名称を指定した場合は、エラーとなります。

#### nodeid

追加するノードイベントリスナに対応する EADS サーバ ID を指定します。

#### node\_eventtype

追加するノードイベントリスナの実行契機の情報を目指すポインタを指定します。

#### node\_eventlistener

追加するノードイベントリスナを目指すポインタを指定します。

#### userdata

追加するノードイベントリスナとのデータの伝達に使用するポインタを指定します。

任意のアドレスを指定できます。

追加するノードイベントリスナとのデータの伝達にポインタを使用しない場合は、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、「[20.2 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

ノードイベントリスナの追加が終了した時点での、接続先 EADS サーバの情報一覧 (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、「[20.1.53 ead\\_nodelist 構造体 \(EADS サーバ情報\)](#)」を参照してください。

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。

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、AP でメモリ領域を解放してください。詳細については、「[19.1.1\(10\) 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。

- 関数実行と同時にクラスタ構成情報が変更された場合、縮退済みの EADS サーバや追加済みの EADS サーバに対しては、node\_eventlistener に設定した関数を呼び出せません。関数実行後に返却された EADS サーバの情報を確認してください。
- この関数の実行時には、EADS サーバの接続可否は考慮されません。そのため、取得した情報には、接続できない EADS サーバ（例：EADS サーバの縮退）が含まれていることがあります。取得した EADS サーバを接続先として使用する場合は、その EADS サーバに接続できるかどうかを、ead\_node 構造体の is\_enable メンバで確認してください。  
ead\_node 構造体の形式と説明については、「[20.1.54 ead\\_node 構造体 \(EADS サーバ指定のユーザファンクションで使用するオブジェクト\)](#)」を参照してください。

## 20.1.41 ead\_remove\_cluster\_eventlistener() (クラスタイベントリスナ削除)

### (1) 機能

クラスタイベントリスナを削除します。

未通知のイベントの情報を破棄します。

この関数は、ead\_add\_cluster\_eventlistener() と対になるように呼び出してください。

### (2) 形式

```
#include <eads.h>
void ead_remove_cluster_eventlistener
(
  const EAD_CACHE_MANAGER *cmp,           /* In */
  const char *cluster_eventlistener_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。

cluster\_eventlistener\_name

削除するクラスタイベントリスナの名称を指定します。

指定できるデータ型	文字数	説明
'%0'を終端記号とする文字列 (char *)	1~1,024	<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> </ul>

指定できるデータ型	文字数	説明
		<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です。</li> <li>長さ0の文字列('¥0'だけ)を指定できません。</li> </ul>

`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、「[20.2 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。

## 20.1.42 ead\_remove\_node\_eventlistener() (ノードイベントリスナ削除)

### (1) 機能

ノードイベントリスナを削除します。

未通知のイベントの情報を破棄します。

この関数は、`ead_add_node_eventlistener()`と対になるように呼び出してください。

### (2) 形式

```
#include <eads.h>
void ead_remove_node_eventlistener
(
    const EAD_CACHE_MANAGER    *cmp,           /* In */
    const char                 *node_eventlistener_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。

`node_eventlistener_name`

削除するノードイベントリスナの名称を指定します。

指定できるデータ型	文字数	説明
'¥0'を終端記号とする文字列 (char *)	1~1,024	<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です。</li> <li>長さ0の文字列('¥0'だけ)を指定できません。</li> </ul>



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、「[20.2 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。

## 20.1.43 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を超えない値を設定してください。

## 20.1.44 ead\_key\_value\_pair 構造体 (key と value の組み合わせ)

ead\_key\_value\_pair 構造体には、key と value の組み合わせの情報を格納します。

### (1) 形式

```
struct ead_key_value_pair {
    char      *key;
```

```
    ead_value_element  value;
};
```

## (2) メンバの説明

### key

value に関連づける key を格納します。

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) を格納します。

ead\_value\_element 構造体の形式と説明については、「20.1.43 ead\_value\_element 構造体 (value 情報)」を参照してください。

## 20.1.45 ead\_keys 構造体 (複数の key)

ead\_keys 構造体には、複数の key を格納します。

### (1) 形式

```
struct ead_keys {
    size_t  size;
    char    **keys;
};
```

## (2) メンバの説明

### size

格納する key の個数を格納します。

### keys

key の一覧を格納した配列の先頭アドレスを格納します。

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。

## 20.1.46 ead\_group\_names 構造体 (複数のグループ名)

ead\_group\_names 構造体には、複数のグループ名を格納します。

### (1) 形式

```
struct ead_group_names {
    size_t  size;
```

```
char    **group_names;
};
```

## (2) メンバの説明

size

格納するグループ名の個数を格納します。

group\_names

グループ名の一覧を格納した配列の先頭アドレスを格納します。

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。

## 20.1.47 ead\_put\_all\_results 構造体 (ead\_put\_all()の実行結果)

ead\_put\_all\_results 構造体には、ead\_put\_all()の実行結果を格納します。

### (1) 形式

```
struct ead_put_all_results {
    size_t          success_operation_number;
    size_t          failure_operation_number;
    ead_failure_operation_info *failure_info;
};
```

## (2) メンバの説明

success\_operation\_number

一括操作を試みた結果、一部またはすべての操作が失敗した場合に、操作が成功した key の数を格納します。

failure\_operation\_number

一括操作を試みた結果、一部またはすべての操作が失敗した場合に、failure\_info メンバの要素数を格納します。

failure\_info

失敗した一括操作の情報の一覧 (ead\_failure\_operation\_info 構造体) の先頭を示すポインタを格納します。

ead\_failure\_operation\_info 構造体の形式と説明については、「[20.1.51 ead\\_failure\\_operation\\_info 構造体](#) (一括操作の実行で失敗した操作の情報)」を参照してください。

## 20.1.48 ead\_get\_all\_results 構造体 (ead\_get\_all()の実行結果)

ead\_get\_all\_results 構造体には、ead\_get\_all()の実行結果を格納します。

### (1) 形式

```
struct ead_get_all_results {
    size_t          values_length;
    ead_value_element *values;
    size_t          success_operation_number;
    size_t          failure_operation_number;
    ead_failure_operation_info *failure_info;
};
```

### (2) メンバの説明

values\_length

取得した value の個数を格納します。

values

取得した value 情報を格納した、ead\_value\_element 構造体の配列の先頭アドレスを指定します。

ead\_value\_element 構造体の形式と説明については、「[20.1.43 ead\\_value\\_element 構造体 \(value 情報\)](#)」を参照してください。

success\_operation\_number

一括操作を試みた結果、一部またはすべての操作が失敗した場合に、操作が成功した key の数を格納します。

failure\_operation\_number

一括操作を試みた結果、一部またはすべての操作が失敗した場合に、failure\_info の要素数を格納します。

failure\_info

失敗した一括操作の情報の一覧 (ead\_failure\_operation\_info 構造体) の先頭を示すポインタを格納します。

ead\_failure\_operation\_info 構造体の形式と説明については、「[20.1.51 ead\\_failure\\_operation\\_info 構造体 \(一括操作の実行で失敗した操作の情報\)](#)」を参照してください。

## 20.1.49 ead\_get\_group\_results 構造体 (ead\_get\_group()の実行結果)

ead\_get\_group\_results 構造体には、ead\_get\_group()の実行結果を格納します。

## (1) 形式

```
struct ead_get_group_results {
    size_t          key_value_length;
    ead_key_value_pair *key_value_array;
};
```

## (2) メンバの説明

### key\_value\_length

取得した key と value を格納した ead\_key\_value\_pair 構造体の配列の要素数を格納します。

### key\_value\_array

取得した key と value を格納した ead\_key\_value\_pair 構造体の配列の先頭アドレスを指定します。  
ead\_key\_value\_pair 構造体の形式と説明については、「[20.1.44 ead\\_key\\_value\\_pair 構造体 \(key と value の組み合わせ\)](#)」を参照してください。

## 20.1.50 ead\_remove\_all\_results 構造体 (ead\_remove\_all()の実行結果)

ead\_remove\_all\_results 構造体には、ead\_remove\_all()の実行結果を格納します。

## (1) 形式

```
struct ead_remove_all_results {
    size_t          success_operation_number;
    size_t          failure_operation_number;
    ead_failure_operation_info *failure_info;
};
```

## (2) メンバの説明

### success\_operation\_number

一括操作を試みた結果、一部またはすべての操作が失敗した場合に、操作が成功した key の数を格納します。

### failure\_operation\_number

一括操作を試みた結果、一部またはすべての操作が失敗した場合に、failure\_info の要素数を格納します。

### failure\_info

失敗した一括操作の情報の一覧 (ead\_failure\_operation\_info 構造体) の先頭を示すポインタを格納します。

ead\_failure\_operation\_info 構造体の形式と説明については、「[20.1.51 ead\\_failure\\_operation\\_info 構造体 \(一括操作の実行で失敗した操作の情報\)](#)」を参照してください。

## 20.1.51 ead\_failure\_operation\_info 構造体（一括操作の実行で失敗した操作の情報）

ead\_failure\_operation\_info 構造体には、一括操作を試みた結果、一部またはすべての操作が失敗した場合に、失敗した操作の情報を格納します。

### (1) 形式

```
struct ead_failure_operation_info {
    int index;
    int error_code;
};
```

### (2) メンバの説明

index

一括操作の実行時に指定した key の一覧中での位置（添え字）を格納します。

error\_code

失敗の要因を示すエラーコードを格納します。

## 20.1.52 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を超えない値を設定してください。

## 20.1.53 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 (ead\_node 構造体) 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。

## 20.1.54 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、「[20.1.55 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。

## 20.1.55 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。



## 20.1.56 ead\_cluster\_eventlistener 構造体 (クラスタイベントリスナの関数)

ead\_cluster\_eventlistener 構造体には、クラスタイベントリスナ用にユーザが定義する関数のポインタを格納します。

### (1) 形式

```
struct ead_cluster_eventlistener {
    void (*on_event)(EAD_CACHE_MANAGER*,
                    char*,
                    ead_cluster_eventcontext*,
                    void*);
};
```

### (2) メンバの説明

on\_event

ead\_cluster\_eventcontext 構造体の event\_type メンバが示すイベントの発生時に、呼び出される関数を格納します。

### (3) 引数

EAD\_CACHE\_MANAGER

キャッシュマネージャを指すポインタを指定します。

char

実行されるクラスタイベントリスナの名称を指定します。

ead\_cluster\_eventcontext

クラスタイベントリスナ実行時に使用する情報を指すポインタを指定します。

void

クラスタイベントリスナ追加時に指定した、クラスタイベントリスナとのデータの伝達に使用するポインタを指定します。

### (4) 注意事項

- ポインタに NULL を設定していた場合は、クラスタイベントリスナの登録に失敗します。
- ポインタに設定した関数が異常終了した場合、EADS クライアントがダウンするおそれがあります。
- ポインタに設定した関数の第 3 引数 (ead\_cluster\_eventcontext) に指定されたポインタは、その関数が終了すると自動で解放されます。

## 20.1.57 ead\_node\_eventlistener 構造体 (ノードイベントリスナの関数)

ead\_node\_eventlistener 構造体には、ノードイベントリスナ用にユーザが定義する関数のポインタを格納します。

### (1) 形式

```
struct ead_node_eventlistener {
    void (*on_event)(char*,
                    ead_node_eventcontext*,
                    void*);
};
```

### (2) メンバの説明

on\_event

ead\_node\_eventcontext 構造体の event\_type メンバが示すイベントの発生時に、呼び出される関数を格納します。

なお、event\_type メンバに EAD\_EVENT\_TYPE\_RECOVERED(2)が設定されている場合、関数の呼び出し直後にキャッシュを操作する API を実行すると、イベントが発生した EADS サーバの処理でエラーが発生するおそれがあります。エラーが発生した場合は、しばらくしてから再実行してください。

イベントが発生した EADS サーバとは、ead\_node\_eventcontext 構造体の nodeid メンバが示す EADS サーバです。

### (3) 引数

char

実行されるノードイベントリスナの名称を指定します。

ead\_node\_eventcontext

ノードイベントリスナ実行時に使用する情報を指すポインタを指定します。

void

ノードイベントリスナ追加時に指定した、ノードイベントリスナとのデータの伝達に使用するポインタを指定します。

### (4) 注意事項

- ポインタに NULL を設定した場合は、ノードイベントリスナの登録に失敗します。
- ポインタに設定した関数が異常終了した場合、EADS クライアントがダウンするおそれがあります。
- ポインタに設定した関数の第 2 引数 (ead\_node\_eventcontext) に指定されたポインタは、その関数が終了すると自動で解放されます。

## 20.1.58 ead\_cluster\_eventcontext 構造体 (クラスタイベントリスナの情報)

ead\_cluster\_eventcontext 構造体には、クラスタイベントリスナ実行時に使用する情報を格納します。

### (1) 形式

```
struct ead_cluster_eventcontext {
    int      event_type;
    ead_node node;
};
```

### (2) メンバの説明

event\_type

発生したイベントの種別を格納します。

記号定数(値)	説明
EAD_EVENT_TYPE_ADDED(1)	EADS サーバがクラスタに追加されました。

node

イベントが発生した EADS サーバの情報を格納します。

## 20.1.59 ead\_node\_eventcontext 構造体 (ノードイベントリスナの情報)

ead\_node\_eventcontext 構造体は、ノードイベントリスナ実行時に使用する情報を格納します。

### (1) 形式

```
struct ead_node_eventcontext {
    int      event_type;
    int      nodeid;
    int      rangeid;
};
```

### (2) メンバの説明

event\_type

発生したイベントの種別を格納します。

記号定数(値)	説明
EAD_EVENT_TYPE_ISOLATED(1)	<ul style="list-style-type: none"><li>障害の発生によって、EADS サーバが縮退しました。</li></ul>

記号定数(値)	説明
	<ul style="list-style-type: none"> <li>EADS サーバが停止しました。</li> <li><code>eztool isolate</code> コマンドの実行によって、EADS サーバが縮退しました。</li> </ul>
<code>EAD_EVENT_TYPE_RECOVERED(2)</code>	EADS サーバが復旧しました。
<code>EAD_EVENT_TYPE_PROMOTED_TO_MASTER(3)</code>	データを多重化している場合に、マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバに変わりました。
<code>EAD_EVENT_TYPE_PROMOTED_TO_SLAVE(4)</code>	データを多重化している場合に、データのコピー先の EADS サーバに変わりました。
<code>EAD_EVENT_TYPE_DEMOTED_TO_SLAVE(5)</code>	データを多重化している場合に、マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバから、データのコピー先の EADS サーバに変わりました。
<code>EAD_EVENT_TYPE_NO_ROLE(6)</code>	<p>EADS サーバが、次のどちらでもなくなりました。</p> <ul style="list-style-type: none"> <li>マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバ</li> <li>データのコピー先の EADS サーバ</li> </ul>

`nodeid`

イベントが発生した EADS サーバ ID を格納します。

`rangeid`

イベントが発生したレンジ ID を格納します。

なお、`event_type` に `ISOLATED(1)` または `RECOVERED(2)` が設定されていた場合、`rangeid` には無効な値(-1)が設定されます。

## 20.1.60 `ead_node_eventtype` 構造体（ノードイベントリスナの実行契機）

`ead_node_eventtype` 構造体は、ノードイベントリスナの実行契機を示す情報を格納します。

### (1) 形式

```
struct ead_node_eventtype {
    EAD_BOOL    isolated;
    EAD_BOOL    recovered;
    EAD_BOOL    promoted_to_master;
    EAD_BOOL    promoted_to_slave;
    EAD_BOOL    demoted_to_slave;
    EAD_BOOL    no_role;
};
```

## (2) メンバの説明

メンバ	イベント	指定値
isolated	<ul style="list-style-type: none"><li>障害の発生によって、EADS サーバが縮退しました。</li><li>EADS サーバが停止しました。</li><li><code>eztool isolate</code> コマンドの実行によって、EADS サーバが縮退しました。</li></ul>	該当するイベントの発生時にノードイベントリスナを実行する場合は、 <code>EAD_TRUE</code> を指定します。ノードイベントリスナを実行しない場合は、 <code>EAD_FALSE</code> を指定します。
recovered	EADS サーバが復旧しました。	
promoted_to_master	データを多重化している場合に、マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバに変わりました。	
promoted_to_slave	データを多重化している場合に、データの複製先の EADS サーバに変わりました。	
demoted_to_slave	データを多重化している場合に、マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバから、データの複製先の EADS サーバに変わりました。	
no_role	EADS サーバが、次のどちらでもなくなりました。 <ul style="list-style-type: none"><li>マスタとなるデータ（コピー元となるデータ）を格納する EADS サーバ</li><li>データの複製先の EADS サーバ</li></ul>	

## (3) 注意事項

すべてのメンバに `EAD_FALSE` を指定した場合は、ノードイベントリスナの登録に失敗します。

## 20.2 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 20-2 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	データを更新する API の処理状態*	クライアント定義の <code>eads.client.compat</code> パラメータに 0300 を指定した場合の <code>error_code</code> の値
0	EAD_OK	処理は正常に終了しました。	該当しません。	○	0
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。	△	1000
1010	EAD_ERROR_ILLEGAL_ARGUMENT	指定されたパラメータが不正です。	API の引数に不正なパラメータが指定されています。	×	1010
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。	×	1040
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。	—	1100
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。	—	1110
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。	×	1120

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
1200	EAD_ERROR_EVENTLISTENER_NOT_INITIALIZED	クラスタイベントリスナとノードイベントリスナを追加できません。	イベントリスナが無効になっている状態（クライアント定義の <code>eads.client.eventListener.enable</code> パラメタに <code>true</code> が指定されていない）で、次の処理を実行しました。 <ul style="list-style-type: none"> <li>クラスタイベントリスナの追加 (<code>ead_add_cluster_event_listener()</code>の実行)</li> <li>ノードイベントリスナの追加 (<code>ead_add_node_eventlistener()</code>の実行)</li> </ul>	—	—
1210	EAD_ERROR_EVENTLISTENER_EXIST	すでに追加されているクラスタイベントリスナまたはノードイベントリスナを、追加しようとした。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>クラスタイベントリスナの追加後 (<code>ead_add_cluster_event_listener()</code>の実行後)に、クラスタイベントリスナの追加をした。</li> <li>ノードイベントリスナの追加後 (<code>ead_add_node_eventlistener()</code>の実行後)に、ノードイベントリスナの追加をした。</li> </ul>	—	—
1220	EAD_ERROR_EVENTLISTENER_NOT_REMOVED	削除する必要がないクラスタイベントリスナまたはノードイベントリスナを、削除しようとした。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>クラスタイベントリスナの削除後 (<code>ead_remove_cluster_eventlistener()</code>の実行後)に、クラスタイベントリスナの削除をした。</li> <li>ノードイベントリスナの削除後 (<code>ead_remove_node_eventlistener()</code>の実行後)に、</li> </ul>	—	—

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
			<p>ノードイベントリスナの削除をした。</p> <ul style="list-style-type: none"> <li>クラスタイベントリスナの追加 (ead_add_cluster_event_listener() の実行) をしていない状態で、クラスタイベントリスナの削除 (ead_remove_cluster_eventlistener() の実行) をした。</li> <li>ノードイベントリスナの追加 (ead_add_node_eventlistener() の実行) をしていない状態で、ノードイベントリスナの削除 (ead_remove_node_eventlistener() の実行) をした。</li> </ul>		
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。	ead_init_client() または ead_init_client_n() 実行中に予期しないエラーが発生しました。	—	2000
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> <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>	—	2010
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。	—	2020
2030	EAD_ERROR_INIT_LOGGER	ログの初期化に失敗しました。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>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> </ul>	—	2030



error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
			<ul style="list-style-type: none"> <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> <li>接続先 EADS サーバでノンブロッキング I/O 通信バッファが不足した。</li> <li>接続先 EADS サーバとの通信中にタイムアウトが発生した。</li> </ul>	—	2040
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> </ul>	△	3000
3001	EAD_ERROR_NET_SEND_REQUEST	EADS サーバへのリクエスト送信処理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> </ul>	△	3000

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
			<ul style="list-style-type: none"> <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> </ul>		
3002	EAD_ERROR_NET_RECEIVE_RESPONSE	EADS サーバへのレスポンス受信処理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> </ul>	△	3000
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。	<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>	△	3010
3011	EAD_ERROR_NET_SEND_TIMEOUT	EADS サーバへのリクエスト送信処理でタイムアウトが発生しました。	<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>	△	3010
3012	EAD_ERROR_NET_RECEIVE_TIMEOUT	EADS サーバへのレスポンス受信処理でタイムアウトが発生しました。	<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> </ul>	△	3010

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
			<ul style="list-style-type: none"> <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。	<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>	×	3020
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。	<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>	△	3030
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。	<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、再起動した接続先 EADS サーバが保持する</li> </ul>	—	3040

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
			<p>クラスタ構成情報が一致しない。</p> <ul style="list-style-type: none"> <li>接続先 EADS サーバの最大同時接続数を越えた。</li> <li>接続先 EADS サーバとの通信中にタイムアウトが発生した。</li> </ul>		
4000	EAD_ERROR_SERVER	予期しない内部エラーが EADS サーバで発生しました。	予期しない内部エラーが接続先 EADS サーバで発生しました。	△	4000
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。	<p>次の要因が考えられます。</p> <ul style="list-style-type: none"> <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> <li>接続先 EADS サーバがサポートしていない API を使用した。</li> </ul>	×	4010
4030	EAD_ERROR_SERVER_UNAVAILABLE	EADS サーバ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。	×	4030
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。	×	4040
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。	×	4060
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)	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。	×	4070

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
		め), 処理を続行できませんでした。			
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。	×	4080
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。	×	4090
4100	EAD_ERROR_SERVER_NOT_RUNNING	リクエストを処理できる EADS サーバがありません。	次の要因が考えられます。 <ul style="list-style-type: none"> <li>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になっている。</li> <li>クラスタが稼働していない状態になっている。</li> </ul>	×	4100
4110	EAD_ERROR_SERVER_STATUS	EADS サーバの状態が, リクエストを実行できる状態ではありません。	接続先 EADS サーバの状態がリクエストを実行できる状態でないため, リクエストを処理できませんでした。	×	4000
4120	EAD_ERROR_SERVER_NOT_EXECUTE_REQUEST_TEMPORARILY	EADS サーバは, 一時的にリクエストを実行できません。	スケールアウト処理中のため, 受付済みのリクエストを一時的に処理できませんでした。スケールアウト処理が完了するまで待機しないでエラーとなる設定になっています。そのため, スケールアウト処理が完了したあとに, リクエストを再実行してください。	×	4000
4200	EAD_ERROR_SERVER_CACHE	キャッシュに対する操作を実行できませんでした。	接続先 EADS サーバで何らかの問題が発生しているため, キャッシュを操作できない状態になっています。操作を中	×	4000

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
			止して EADS サーバの状態を確認してください。		
4210	EAD_ERROR_SERVER_CACHE_NOT_FOUND	指定したキャッシュが存在しないため、キャッシュに対する操作を実行できませんでした。	指定したキャッシュが存在しないため、キャッシュを操作できませんでした。指定したキャッシュに対する操作を中止して、EADS サーバの状態を確認してください。	×	4000
4230	EAD_ERROR_SERVER_CACHE_CLUSTER_UPDATE	リクエスト処理中にクラスタ構成変更が発生したため、キャッシュに対する操作を実行できませんでした。	リクエスト処理中にクラスタ構成変更が発生したため、キャッシュを操作できませんでした。クラスタ構成の変更が完了したあとに、キャッシュに対する操作を再度実行してください。	×	4000
4300	EAD_ERROR_SERVER_CACHE_BEFORE_REPLICATION	キャッシュに対する操作中に内部エラーが発生しましたが、データの多重化は発生しませんでした。	接続先 EADS サーバでキャッシュに対する操作中、内部エラーが発生しました。データの多重化は発生していないため、ほかの正常な EADS サーバへの影響はありません。EADS サーバが縮退したあと、正常な EADS サーバに接続先が変更されると、同じ操作から処理を再開できます。	×	4000
4310	EAD_ERROR_SERVER_CACHE_AFTER_REPLICATION	キャッシュに対する操作中に EADS サーバで内部エラーが発生し、データ更新処理が失敗しました。	接続先 EADS サーバでキャッシュに対する操作中、内部エラーが発生しました。データの多重化はすでに実行されているため、問題が発生した接続先 EADS サーバが縮退したあとに正常なサーバに接続先が変更されると、データが更新された状態から操作を再開できます。	△	4000
4400	EAD_ERROR_SERVER_LIMIT_RECEIVE_BUFFER	EADS サーバのリクエスト受信処理でノンブロッキング I/O 通信	接続先 EADS サーバのリクエスト受信処理でノンブロッキング I/O 通信バッファが不足	×	4000

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
		バッファが不足しました。	しているため、リクエストを処理できませんでした。		
4410	EAD_ERROR_SERVER_LIMIT_SEND_BUFFER	EADS サーバのレスポンス送信処理でノンブロッキング I/O 通信バッファが不足しました。	接続先 EADS サーバのレスポンス送信処理でノンブロッキング I/O 通信バッファが不足しているため、レスポンスを送信できませんでした。	△	4000
4700	EAD_ERROR_SERVER_FUNCTION_EXECUTE	EADS サーバのユーザファンクション内でエラーが発生しました。	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ました。ユーザファンクション内の処理を確認してください。	—	4000
4710	EAD_ERROR_SERVER_FUNCTION_RETURN_SERIALIZE	EADS サーバでユーザファンクションの戻り値のシリアライズに失敗しました。	接続先 EADS サーバで実行したユーザファンクションの戻り値に、シリアライズできないオブジェクトを指定しています。	—	4000
4730	EAD_ERROR_SERVER_FUNCTION_NOT_FOUND	指定したユーザファンクション名のユーザファンクションが、EADS サーバに存在しません。	指定したユーザファンクション名のユーザファンクションが、接続先 EADS サーバに存在しません。	—	4000
4800	EAD_ERROR_SERVER_LIMIT_EXTERNAL_MEMORY	データを格納するメモリの容量が不足しています。	接続先 EADS サーバでデータを格納するメモリ (Explicit ヒープ) の容量が不足しているため、リクエストを処理できませんでした。	×	4000
4810	EAD_ERROR_SERVER_LIMIT_CACHE_FILE	データを格納するキャッシュファイルの容量が不足しています。	接続先 EADS サーバでデータを格納するキャッシュファイルの容量が不足しているため、リクエストを処理できませんでした。	×	4000
4820	EAD_ERROR_SERVER_LIMIT_KV_COUNT	EADS サーバに格納できる key の数が上限に達しています。	接続先 EADS サーバに格納できる key の数が上限に達しているため、リクエストを処理できませんでした。	×	4000
4830	EAD_ERROR_SERVER_LIMIT	指定した key, グループ名, または value の	指定した key, グループ名, または value のサイズがクラ	×	4000

error_code の値	記号定数	エラーの内容	発生要因	データを更新する API の処理状態※	クライアント定義の eads.client.compat パラメタに 0300 を指定した場合の error_code の値
	T_KEY_VALUE_LENGTH	サイズがクラスタに格納できる最大サイズを超えています。	スタに格納できる最大サイズを超えているため、リクエストを処理できませんでした。		
4840	EAD_ERROR_SERVER_LIMIT_EXTERNAL_MEMORY_TEMPORARILY	接続先 EADS サーバで、データを格納するメモリの容量が一時的に不足しています。	リバランス処理中は、EADS サーバでデータを格納するメモリ (Explicit ヒープ) の容量が一時的に減少します。そのため、接続先 EADS サーバでリクエストを処理できませんでした。	×	4000
4850	EAD_ERROR_SERVER_LIMIT_KV_COUNT_TEMPORARILY	接続先 EADS サーバで、格納できる key の数が一時的に上限に達しています。	リバランス処理中は、EADS サーバに格納できる key の数の上限が一時的に減少します。そのため、接続先 EADS サーバでリクエストを処理できませんでした。	×	4000
4999	EAD_ERROR_SERVER_UNKNOWN	解析できない内部エラーが EADS サーバで発生しました。	接続先 EADS サーバで内部エラーが発生しましたが、接続先 EADS サーバのバージョンが EADS クライアントライブラリのバージョンより新しいため、解析できませんでした。	△	4000
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。	△	5000
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。	△	5010
5020	EAD_ERROR_CLIENT_BATCH_CANCEL	一括操作の実行をキャンセルしました。	一括操作を継続できないため、未実行の操作をキャンセルしました。	×	5020
6000	EAD_ERROR_BATCH_FAILED_ALL	一括操作のすべての操作に失敗しました。	データを一括で操作する API で実行しようとした操作は、すべて失敗しました。	—	6000
6010	EAD_ERROR_BATCH_FAILED_PART	一括操作の一部の操作に失敗しました。	データを一括で操作する API で実行しようとした操作のうち、一部が失敗しました。	—	6010



## 注※

ead\_put(), ead\_remove()などのデータを更新する API の実行時にエラーコードが出力された場合に、データが更新されているかどうかを表しています。

記号の意味を次に示します。

○：データは更新されています。

△：データが更新されているかどうかは不明です。処理が完了したかどうか、確認してください。

×：データは更新されていません。

－：ead\_put(), ead\_remove()などのデータを更新する API の実行時に、このエラーコードは出力されません。

# 21

## 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。

## 21.1 パラメーター一覧

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ーターの一覧について説明します。

### 21.1.1 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ー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ーターを太字で表記しています。

環境依存パラメーターの設計については, 「7. 環境依存パラメーターを設計する (EADS サーバ)」を参照してください。

チューニングパラメーターの設計については, 「9. チューニングパラメーターを設計する」を参照してください。

表 21-1 サーバ定義ファイルに指定できるパラメーター一覧

項番	分類	パラメーター名	指定する内容
1	通信 (「7.3 通信に依存するパラメーターの設計」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.replication.port</b>	EADS サーバ間の通信に使用するポート番号
4		<b>eads.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.transfer.port</b>	EADS サーバの復旧処理, スケールアウト処理, またはリバランス処理に使用するポート番号
6		<b>eads.admin.operation.port</b>	コマンドが使用するポート番号
7	ログファイル (「7.4 ログファイルに依存するパラメーターの設計」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
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

項番	分類	パラメタ名	指定する内容
13		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
14		eads.logger.exception.filenum	例外ログのファイル数
15		eads.user.logger.rotationStyle	ユーザログのローテーション方式
16		eads.user.logger.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.user.logger.filenum	ユーザログのファイル数
18		eads.user.logger.exception.rotationStyle	ユーザ例外ログのローテーション方式
19		eads.user.logger.exception.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.user.logger.exception.filenum	ユーザ例外ログのファイル数
21		eads.cache.logger.diskCache.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.cache.logger.diskCache.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.cache.logger.diskCache.filenum	キャッシュファイル操作ログのファイル数
24		eads.statistics.interval	統計情報を出力する間隔
25		eads.statistics.filenum	統計情報の取得日数
26		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
27		eads.java.log.filesize	Java ログ (JavaVM の保守情報) のファイルサイズ
28		eads.java.log.filenum	Java ログ (JavaVM の保守情報) のファイル数
29	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.admin.backup.dir	ストアデータファイルの出力先パス名
30	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.server.connection.buffersize	コネクション単位で確保する、データの送受信で使用する作業領域のバッファサイズ
31		eads.replication.connection.buffersize	合意メッセージの送受信バッファサイズ
32		eads.replication.sendQueue.length	合意メッセージの送信キューの長さ
33		eads.replication.sendQueue.datasize	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
34		eads.transfer.connection.bufferSize	復旧処理、スケールアウト処理、およびリバランス処理で使用するデータの送受信バッファサイズ
35		eads.transfer.dataSize	復旧処理およびスケールアウト処理で送信するデータサイズ
36		eads.rebalance.transfer.dataSize	リバランス処理のデータ転送で送信するデータサイズ
37		eads.replication.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.admin.operation.resume.send.dataSize	eztool resume コマンド実行時に転送する差分のデータサイズ
39		eads.java.nio.maxCachedBufferSize	EADS サーバでキャッシュされる 1 ダイレクトバッファ当たりのサイズの上限
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.maxConnections
41	eads.server.cache.maxExecuteThreads		データ操作の最大同時実行スレッド数
42	eads.server.function.maxExecuteThreads		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	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.server.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
44		eads.admin.operation.connection.timeout	特定のコマンド実行時の、EADS サーバ間の接続やデータ送受信のタイムアウト時間
45		eads.failureDetector.heartbeat.interval	ハートビートの送信間隔
46		eads.failureDetector.heartbeat.timeout	ハートビートのタイムアウト時間
47		eads.failureDetector.connection.timeout	生存確認の接続タイムアウト時間
48		eads.failureDetector.read.timeout	生存確認の受信タイムアウト時間
49		eads.failureDetector.retry	生存確認タイムアウト時のリトライ回数
50		eads.failureDetector.assertive.threshold	EADS サーバダウンに同意する EADS サーバ数
51		eads.admin.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.connection.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

項番	分類	パラメタ名	指定する内容
53		eads.admin.operation.isolate.gracefulstop.wait Time	eztool isolate コマンドを実行した EADS サーバのクラスタ構成情報の更新が完了してから、縮退処理が完了するまでの時間
54		eads.replication.consensus.timeout	合意処理のタイムアウト時間
55		eads.transfer.timeout	復旧処理、スケールアウト処理、およびリバランス処理でのデータ送信タイムアウト時間
56		eads.transfer.interval	復旧処理およびスケールアウト処理でのデータ送信間隔
57		eads.rebalance.transfer.interval	リバランス処理のデータ転送での送信間隔
58		eads.replication.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
59		eads.admin.operation.resume.send.interval	eztool resume コマンド実行時の差分転送間隔
60		eads.server.eventListener.notify.retry.timeout	イベントリスナのイベント再送のタイムアウト時間
61	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.getError.isolate.enable	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
62	ノンブロッキング I/O 通信制御 (「7.8 ノンブロッキング I/O 通信制御に依存するパラメタの設計」参照)	eads.server.nonBlocking.enable	ノンブロッキング I/O 通信制御が有効かどうか
63		eads.server.nonBlocking.processorThreads	リクエスト処理スレッド数
64		eads.server.nonBlocking.selectorThreads	セレクタスレッド数
65		eads.server.nonBlocking.maxAllocateConnections	最大同時割当数
66		eads.statistics.nonBlocking.enable	ノンブロッキング I/O 通信制御の統計情報出力が有効かどうか
67	スケールアウト処理・リバランス処理 (「7.9 スケールアウト処理・リバランス処理に依存するパラメタの設計」参照)	eads.scaling.moveDataWithUpdateRequest.enable	スケールアウト処理中またはリバランス処理中の更新リクエスト実行時に、データを移動させるかどうか
68		eads.scaling.moveData.interval	スケールアウト処理中、またはリバランス処理中のデータの移動間隔

## (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 21-2 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	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)	<b>eads.node.&lt; EADS サーバ ID &gt;.address</b>	クラスタを構成する EADS サーバの IP アドレス
2		<b>eads.node.&lt; EADS サーバ ID &gt;.port</b>	クラスタを構成する EADS サーバのポート番号
3		<b>eads.node.&lt; EADS サーバ ID &gt;.position</b>	EADS サーバの位置 (ハッシュ値)

### (3) 共通設定ファイル

共通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 21-3 共通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	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)	<b>eads.failureDetector.heartbeat.address</b>	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)
2		<b>eads.failureDetector.heartbeat.port</b>	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
3		<b>eads.replication.factor</b>	データの多重度
4	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)	<b>eads.admin.backup.exportCommand.generation.maxNum</b>	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
5		<b>eads.admin.backup.stopCommand.generation.maxNum</b>	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
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)	<b>eads.cache.key.maxsize</b>	クラスタに格納できる key の最大サイズ
7		<b>eads.replication.preparations</b>	一度に合意できる処理の最大数
8		<b>eads.replication.external.heapsize</b>	更新操作の履歴が格納される領域のサイズ
9		<b>eads.java.heapsize</b>	Java ヒープサイズ

項番	分類	パラメタ名	指定する内容
10		eads.java.external.heapsize	Explicit ヒープサイズ
11		eads.java.metaspace.maxsize	metaspace 領域のサイズ
12		eads.cache.limiter.enable	データ総量監視機能を有効にするかどうか
13		eads.cache.keyCount	データ総量監視機能で監視する、1 レンジ当たりのデータ件数
14	ノンブロッキング I/O 通信制御 (「7.8 ノンブロッキング I/O 通信制御に依存するパラメタの設計」参照)	eads.server.nonBlocking.external.heapsize	ノンブロッキング I/O 通信バッファサイズ (単位: メガバイト)
15	スケールアウト処理・リバランス処理 (「7.9 スケールアウト処理・リバランス処理に依存するパラメタの設計」参照)	eads.scaling.requestQueuing.enable	スケールアウト処理中に、受付済みで未実行のリクエストをキューイングするかエラーにするか
16	スケールアウト処理・リバランス処理に依存するパラメタの設計」参照)	eads.scaling.eviction.targetKeyCount	データ自動削除機能を使用するキャッシュが存在する場合のリバランスの移動レンジのキー数

## (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 21-4 コマンド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	通信 (「7.3 通信に依存するパラメタの設計」参照)	eads.command.connect.sub.servers	eztool コマンドの接続先 EADS サーバを識別するための名称 (任意の名称)
2		eads.command.<接続先 EADS サーバ>.address	eztool コマンドの接続先 EADS サーバの IP アドレスまたはホスト名
3		eads.command.<接続先 EADS サーバ>.admin.operation.port	eztool コマンドの接続先 EADS サーバのポート番号
4	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.command.logger.dir	ログファイルの出力先ディレクトリのパス
5		eads.command.logger.message.rotationStyle	メッセージログのローテーション方式
6		eads.command.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



項番	分類	パラメタ名	指定する内容
7		<code>eads.command.logger.message.filenum</code>	メッセージログのファイル数
8		<code>eads.command.logger.exception.rotationStyle</code>	例外ログのローテーション方式
9		<code>eads.command.logger.exception.filesize</code>	例外ログの1ファイル当たりのファイルサイズ
10		<code>eads.command.logger.exception.filenum</code>	例外ログのファイル数
11	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.command.connection.timeout</code>	コマンド実行時のEADSサーバとの通信の接続タイムアウト時間
12		<code>eads.command.common.read.timeout</code>	コマンド実行時のEADSサーバとの通信の受信タイムアウト時間
13		<code>eads.command.common.execution.timeout</code>	コマンド実行開始からのタイムアウト時間
14		<code>eads.command.&lt;サブコマンド名&gt;.read.timeout</code>	特定のサブコマンド実行時の、EADSサーバとの通信の受信タイムアウト時間
15		<code>eads.command.&lt;サブコマンド名&gt;.execution.timeout</code>	特定のサブコマンドの実行開始からのタイムアウト時間
16	コマンドの運用 (「9.4 コマンドの運用に関連するパラメタの設計」参照)	<code>eads.command.compat</code>	コマンドの表示結果のバージョン、およびコマンドが返却するリターンコードのバージョン
17	コンパクション (「9.6 コンパクションに関連するパラメタの設計」参照)	<code>eads.command.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>
18		<code>eads.command.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 21-5 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	<b><code>eads.cache.type</code></b>	キャッシュタイプ

項番	分類	パラメタ名	指定する内容
2	([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.eviction.policy</code>	データの自動削除ポリシー
3		<code>eads.cache.eviction.keyCount</code>	データ自動削除機能でデータを削除するためのしきい値として使用する 1 レンジ当たりのデータ件数
4		<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
5		<code>eads.cache.disk.&lt;n&gt;.dir</code> <sup>※</sup>	キャッシュデータファイルの格納先
6		<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
7		<code>eads.cache.disk.filenum</code>	キャッシュデータファイルの面数
8		<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
9		<code>eads.cache.disk.transfer.interval</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
10		<code>eads.cache.disk.transfer.datasize</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
11		<code>eads.cache.disk.highSpecDrive.enable</code>	高性能 SSD に対応したファイルアクセスをするかどうか
12		<code>eads.cache.disk.persistentMemory.enable</code>	ディスクキャッシュの格納先として, 不揮発性メモリを使用するかどうか
13		<code>eads.cache.disk.persistentMemory.device</code> <sup>※</sup>	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格納先となる不揮発性メモリのデバイスファイル
14		<code>eads.cache.disk.persistentMemory.blocksize</code>	不揮発性メモリ上のキャッシュデータファイルに 1 回に書き込むデータサイズ

#### 注※

ディスクキャッシュの格納先として不揮発性メモリを使用しない場合は `eads.cache.disk.<n>.dir` パラメタが, 使用する場合は `eads.cache.disk.persistentMemory.device` パラメタが必ず指定するパラメタになります。

## (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 21-6 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)	<code>eads.function.&lt;ユーザファンクション名&gt;.maxExecuteThreads</code>	ユーザファンクション単位の最大同時実行スレッド数

## 21.1.2 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 21-7 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><code>eads.client.connect.servers</code></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><code>eads.client.&lt;接続先 EADS サーバ&gt;.address</code></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><code>eads.client.&lt;接続先 EADS サーバ&gt;.port</code></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><code>eads.client.logger.dir</code></b>	ログファイルの出力先ディレクトリのパス
5		<b><code>eads.client.logger.message.filesize</code></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><code>eads.client.logger.message.filenum</code></b>	メッセージログのファイル数
7		<b><code>eads.client.logger.message.console.enable</code></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.client.logger.initErrorOut</code>	ログの初期化失敗時に、標準エラー出力にエラーメッセージを出力するかどうか
9		<code>eads.client.logger.maintenance.filesize</code>	保守ログの1ファイル当たりのファイルサイズ
10		<code>eads.client.logger.maintenance.filenum</code>	保守ログのファイル数
11	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.client.connection.bufferSize</code>	EADS クライアントがデータの送受信で使用するバッファサイズ
12	スレッドプールおよび接続プール	<code>eads.client.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
13	(「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.client.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
14	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.client.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
15		<code>eads.client.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
16		<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
17		<code>eads.client.eventListener.enable</code>	イベントリスナを使用するかどうか
18		<code>eads.client.eventListener.reconnect.interval</code>	イベントリスナ用の接続を再接続する間隔
19	AP の運用 (「9.5 AP の運用に関連するパラメタの設計」参照)	<code>eads.client.batchOperation.unit</code>	一括操作を実行する際に、一度の EADS サーバとの通信で実行するデータ操作の回数
20		<code>eads.client.compat</code>	EADS クライアントが出力するエラーコードのバージョン

## 21.2 コマンド一覧

EADS のコマンドの一覧を次の表に示します。

表 21-8 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

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 21-9 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		最新パラメタの一覧を表示する	listconf	クラスタ※1
5		キャッシュの一覧を表示する	listcache	クラスタ
6		ストアデータファイルの一覧を表示する	listesd	クラスタ
7		グループ名の一覧を表示する	listgroup	クラスタ※1
8		key の一覧を表示する	listkey	クラスタ※1※2
9		データの格納先を表示する	getposition	クラスタ
10		レンジおよびキャッシュの使用状況を確認する	storeusage	クラスタ
11	排他制御	排他を解除する	unlock	クラスタ
12	メモリ・ディスク管理	キャッシュを作成する	createcache	クラスタ
13		キャッシュを削除する	deletecache	クラスタ
14	データ移行	データを書き出す	export	クラスタ※1
15		データを読み込む	import	クラスタ
16		ストアデータファイルを削除する	deleteesd	クラスタ
17	データ操作	指定したデータを格納する	put	EADS サーバ
18		指定したデータを取得する	get	EADS サーバ

項番	利用シーン		サブコマンド名	実行対象
19		指定したデータを削除する	remove	EADS サーバ
20		指定した範囲のデータをすべて削除する	removeall	クラスタ※1※2
21	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
22		ユーザファンクションを実行する	execfunc	クラスタ※1※2※3
23	負荷分散	レンジに格納できる key の数を均等にする	rebalance	クラスタ
24	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
25		キャッシュを再開する	resume	クラスタ
26		キャッシュファイルを書き出す	exportecf	クラスタ※5
27		キャッシュファイルを読み込む	importecf	クラスタ
28		キャッシュファイルを削除する	deleteecf	クラスタ※5
29		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※4
30	トラブルシュート	スレッドダンプを出力する	threaddump	EADS サーバ
31		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
32	終了	クラスタ内の全 EADS サーバを終了する	stop	クラスタ
33		EADS サーバを強制終了する	forcestop	EADS サーバ
34		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

-k または --key オプションを指定すると、指定した key が格納されている EADS サーバが対象となります。

#### 注※4

--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。

#### 注※5

-l または --local 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。

## 21.3 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。

### 21.3.1 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 API

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 API を提供します。

表 21-10 Java 言語のクライアントライブラリで使用できる 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	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	FailureOperationInfo	一括操作を試みた結果, 一部またはすべての操作に失敗した場合に, 失敗した操作の情報を格納するためのクラスです。	
5	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。	
6	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。	
7	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。	
8	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。	

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
9	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。	
10	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。	
11	BatchOperationException	一括操作を試みた結果, 一部またはすべての操作に失敗した場合に返却される CacheException のサブクラスです。	
12	AllFailureException	一括操作を試みた結果, すべての操作に失敗した場合に返却される BatchOperationException のサブクラスです。	
13	PartFailureException	一括操作を試みた結果, 一部の操作に失敗した場合に返却される BatchOperationException のサブクラスです。	
14	ClusterEventListener	クラスタイベントリスナが実装する必要のあるインタフェースです。	
15	NodeEventListener	ノードイベントリスナが実装する必要のあるインタフェースです。	
16	ClusterEventContext	クラスタイベントリスナの実行に必要な情報を ClusterEventListener インタフェースに渡すインタフェースです。	
17	NodeEventContext	ノードイベントリスナの実行に必要な情報を NodeEventListener インタフェースに渡すインタフェースです。	
18	列挙型 ClusterEventType	クラスタイベントリスナの実行の契機を表す列挙型です。	
19	列挙型 NodeEventType	ノードイベントリスナの実行の契機を表す列挙型です。	

## (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 21-11 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の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	ClusterInfo	クラスタの情報を取得するインタフェースです。	com.hitachi.software.xeads.common.ClusterInfo
7	CacheInfo	キャッシュの情報を取得するインタフェースです。	com.hitachi.software.xeads.common.CacheInfo
8	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
9	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
10	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
11	Value	ユーザファンクションで使用できる API 内で, key に関連づけて格納される value を表現するインタフェースです。	com.hitachi.software.xeads.func.store.Value
12	UserLogger	ユーザログのインタフェースです。	com.hitachi.software.xeads.common.UserLogger
13	EADsStoreException	データ操作に関連する処理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
14	InternalServerException	EADS サーバで内部エラーが発生した場合に返却される	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
		EADsStoreException のサブクラスです。	
15	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
16	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

## 21.3.2 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 21-12 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_get_cache_name()	指定したキャッシュのハンドルに関連づけられたキャッシュ名を取得します。	
6	ead_terminate_client()	EADS クライアントを終了します。	
7	ead_put()	value を key に関連づけて格納します。	
8	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。	

項番	関数名	機能	ヘッダファイル
9	ead_put_all()	複数の key と value をキャッシュに一括格納します。	
10	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。	
11	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 (value を更新します)。	
12	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 (value を置換します)。	
13	ead_get()	指定した key に関連づけられた value を取得します。	
14	ead_get_all()	指定した key の一覧に関連づけられた value を一括取得します。	
15	ead_get_group()	指定したグループおよびその下位階層のグループに属する key に関連づけられた value を一括取得します。	
16	ead_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。	
17	ead_remove_all()	指定した key の一覧に関連づけられた value を一括削除します。	
18	ead_remove_group()	指定したグループに属する key および value を、下位階層のグループに属する key および value も含めて一括削除します。	
19	ead_remove_node()	指定した EADS サーバをコピー元 EADS サーバとする key および value を一括削除します。	
20	ead_get_group_names()	指定した EADS サーバに格納されている最上位階層のグループ名の一覧を取得します。 グループ名の一覧は、ASCII コードの昇順になります。	
21	ead_get_group_keys()	指定したグループに属する key の一覧を、下位階層のグループ	

項番	関数名	機能	ヘッダファイル
		に属する key も含めて取得します。 key の一覧は、ASCII コードの昇順になります。	
22	<code>ead_get_node_keys()</code>	指定した EADS サーバに格納されている key の一覧を取得します。 key の一覧は、ASCII コードの昇順になります。	
23	<code>ead_get_group_count()</code>	指定した EADS サーバに格納されている最上位階層のグループの数を取得します。	
24	<code>ead_get_group_key_count()</code>	指定したグループに属する key の数を取得します。指定したグループの下位階層のグループに属する key も、取得される key の数に含まれます。	
25	<code>ead_get_node_key_count()</code>	指定した EADS サーバに格納されている key の数を取得します。	
26	<code>ead_get_group_first_key()</code>	指定したグループに属する key のうち、ASCII コードの昇順で先頭となる key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。	
27	<code>ead_get_node_first_key()</code>	指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で先頭となる key を取得します。	
28	<code>ead_get_group_next_key()</code>	指定したグループに属する key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。指定したグループの下位階層のグループに属する key も取得対象となります。	
29	<code>ead_get_node_next_key()</code>	指定した EADS サーバに格納されている key のうち、ASCII コードの昇順で、指定した key の次に位置する key を取得します。	
30	<code>ead_execute_function()</code>	指定した key またはグループから実行する EADS サーバを決定	

項番	関数名	機能	ヘッダファイル
		して、ユーザファンクションを実行します。	
31	<code>ead_execute_function_rt()</code>	指定した key またはグループから実行する EADS サーバを決定して、ユーザファンクションを実行します。また、受信タイムアウト時間を指定します。	
32	<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。	
33	<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。	
34	<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。	
35	<b>【非推奨】</b> <code>ead_get_node()</code>	<b>【非推奨】</b> 指定した key またはグループの、本来のコピー元 EADS サーバの情報を取得します。	
36	<code>ead_get_slave_nodelist()</code>	指定した EADS サーバに格納されているデータの、本来のコピー先 EADS サーバの情報を取得します。	
37	<code>ead_get_current_master_node()</code>	指定した key (またはグループ) を現時点で格納する、コピー元 EADS サーバの情報を取得します。	
38	<code>ead_get_original_master_node()</code>	指定した key (またはグループ) を格納する、本来のコピー元 EADS サーバの情報を取得します。	
39	<code>ead_add_cluster_eventlistener()</code>	クラスタイベントリスナを追加します。	
40	<code>ead_add_node_eventlistener()</code>	ノードイベントリスナを追加します。	
41	<code>ead_remove_cluster_eventlistener()</code>	クラスタイベントリスナを削除します。	
42	<code>ead_remove_node_eventlistener()</code>	ノードイベントリスナを削除します。	

## 21.4 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。

### 21.4.1 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、「[18.1.5 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、「[18.2.13 EADsStoreException クラス](#)」の「(5)(c)戻り値」を参照してください。

### 21.4.2 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、「[20.2 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。

## 21.5 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 21-13 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（ノンブロッキング I/O 通信制御を使用しない場合）	1	1,024
2	1EADS サーバ当たりの、EADS クライアントからの接続数（ノンブロッキング I/O 通信制御を使用する場合）	1,024	32,768
3	1 クラスタを構成する EADS サーバ数	1※1	96
4	1 クラスタに作成できるキャッシュ数	0	16
5	1EADS サーバに格納できるデータ数※2	0	2,147,483,647
6	データの多重度	1	5
7	key の長さ (単位：文字数)	1	1,024
8	value のサイズ※3 (単位：バイト)	1	262,144
9	グループ名の長さ (単位：文字数)	1	1,022
10	キャッシュ名の長さ (単位：文字数)	1	32
11	グループ数	1	制限なし
12	グループの階層数	1	511
13	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)	16	128
14	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
15	1 キャッシュデータファイル 1 面に格納できるデータ数（ディスクキャッシュの格納先として不揮発性メモリを使用しない場合）	—	131,070
16	1 キャッシュデータファイル 1 面に格納できるデータ数（ディスクキャッシュの格納先として不揮発性メモリを使用する場合）	—	524,286

(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\(3\) value として指定できるデータ](#)」を参照してください。



# 22

## 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。

## 22.1 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。 XXXX 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。 nnnnn 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。

## 22.2 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 サーバが縮退状態 (isolated) になりました。

#### 対処

次の要因が考えられます。

- `eztool isolate` コマンドを実行した。
- EADS サーバ内で運用を続行できない問題が発生した。

「[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.admin.operation.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。

## 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.maxConnections` 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。

## 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。

### 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。
nonBlocking	通信データを格納するノンブロッキング I/O 通信バッファの空き容量が不足しています。

## 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	共通設定の <code>eads.java.external.heapsize</code> パラメタの指定値を大きくしてください。
nonBlocking	共通設定の <code>eads.server.nonBlocking.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、共通設定の <code>eads.java.external.heapsize</code> パラメタの指定値も大きくしてください。

## 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、共通設定の <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、共通設定の <code>eads.replication.external.heapsize</code> パラメタの指定値が小さいです。
nonBlocking	通信データに対して、共通設定の <code>eads.server.nonBlocking.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	共通設定の <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	共通設定の <code>eads.replication.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、共通設定の <code>eads.java.external.heapsize</code> パラメタの指定値も大きくしてください。
nonBlocking	共通設定の <code>eads.server.nonBlocking.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、共通設定の <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	共通設定の <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	共通設定の <code>eads.replication.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、共通設定の <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。

共通設定の `eads.cache.key.maxsize` パラメタの指定値を見直してください。

## KDEA00062-E

The size of group name exceeds the maximum. (maximum size (bytes) = aa....aa, group name size (bytes) = bb....bb)

aa....aa : 最大サイズ (単位: バイト)

bb....bb : グループ名サイズ (単位: バイト)

### 説明

指定したグループ名のサイズが、クラスタに格納できるグループ名の最大サイズ (共通設定の `eads.cache.key.maxsize` パラメタの指定値-2) を超えています。

### 対処

指定したグループ名のサイズを見直してください。

共通設定の `eads.cache.key.maxsize` パラメタの指定値を見直してください。

## KDEA00063-E

An error occurred in the processing to send cluster configuration information. (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

クラスタ構成情報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。

例外ログを参照して要因を特定し、問題を取り除いてください。

## 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

ユーザ関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 : ユーザ関クション名

## 説明

ユーザ関クションの 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 : ユーザ関クション名

## 説明

ユーザ関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 : ユーザ関クション名

## 説明

ユーザ関クションの終了処理に失敗しました (ユーザ関クションの 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 : ユーザ関クション名

## 説明

ユーザ関クション内で例外 (またはエラー) が発生しました (ユーザ関クションの 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

ユーザファンクションの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、ユーザファンクションの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ファンクション名

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

ユーザファンクション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ファンクションの数

cc....cc：初期化に失敗したユーザファンクションの数

#### 説明

ユーザファンクション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。

#### KDEA01904-I

```
CORRECTION (cache = aa....aa, range = bb....bb, ecf = cc....cc)
```

aa....aa : キャッシュ名

bb....bb : レンジ ID

cc....cc : キャッシュファイルのパス

## 説明

キャッシュファイルを修正しました。

## 22.3 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 was 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

「[20.2 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

```
Preparations for using the cache 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。

## KDEA02024-E

Initialization of the logger failed. (details = aa....aa)

aa....aa：エラー詳細情報

## 説明

ロガー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、問題を取り除いてください。

### KDEA02025-E

The connection for the event listener failed.

## 説明

イベントリスナ用の接続の確立に失敗しました。

## 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。

### KDEA02026-I

The connection for the event listener was restored.

## 説明

イベントリスナ用の接続が回復しました。

### KDEA02028-I

Preparations for using the event listener finished. (event listener name = aa....aa)

aa....aa：イベントリスナ名

## 説明

イベントリスナを使用する準備ができました。

### KDEA02029-I

Use of the event listener will now end. (event listener name = aa....aa)

aa....aa：イベントリスナ名

## 説明

イベントリスナの使用を終了します。

## 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。

## 22.4 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

```
Preparations for using the cache 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

「18.1.5 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 was 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。

### KDEA03023-E

Initialization of the logger failed. (details = aa....aa)

aa....aa：エラー詳細情報

## 説明

ロガーの初期化に失敗しました。

## 対処

ロガーの初期化処理中に問題が発生しました。  
エラー詳細情報を確認して、問題を取り除いてください。

### KDEA03101-I

```
Preparations for using the event listener finished. (event listener name = aa....aa, toString = bb....bb)
```

aa....aa：クラスタイベントリスナ名，またはノードイベントリスナ名

bb....bb：オブジェクトの文字列表現

## 説明

クラスタイベントリスナ，またはノードイベントリスナを使用する準備ができました。

### KDEA03102-I

```
Use of the event listener will now end. (event listener name = aa....aa, toString = bb....bb)
```

aa....aa：クラスタイベントリスナ名，またはノードイベントリスナ名

bb....bb：オブジェクトの文字列表現

## 説明

クラスタイベントリスナ，またはノードイベントリスナの使用を終了します。

### KDEA03103-E

```
An exception or error occurred in the event listener. (event listener name = aa....aa, toString = bb....bb)
```

aa....aa：クラスタイベントリスナ名，またはノードイベントリスナ名

bb....bb：オブジェクトの文字列表現

## 説明

クラスタイベントリスナ，またはノードイベントリスナのメソッドの実行中に，例外またはエラーが発生しました。

## 対処

クラスタイベントリスナ，またはノードイベントリスナ内で問題が発生しました。  
メッセージログを参照して要因を特定し，問題を取り除いてください。

## KDEA03104-E

The connection for the event listener failed.

### 説明

イベントリスナ用のコネクションの確立に失敗しました。

### 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。

## KDEA03105-I

The connection for the event listener was restored.

### 説明

イベントリスナ用のコネクションが回復しました。

## KDEA03106-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 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。

要因を特定し、問題を取り除いてください。

## 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。



## 22.5 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 invalid property was detected. (definition 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ました。

#### 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。

### KDEA04506-E

```
A data discrepancy in the definition file was detected. (node ID = aa....aa, definition file name = bb....bb)
```

aa....aa：他 EADS サーバの定義ファイルと不一致がある EADS サーバの EADS サーバ ID

bb....bb：他 EADS サーバの定義ファイルと不一致がある定義ファイル名

#### 説明

他 EADS サーバの定義ファイルと不一致がありました。

#### 対処

他 EADS サーバの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。

スケールアウト処理またはリバランス処理中に縮退した EADS サーバを復旧する場合、縮退した EADS サーバのクラスタ定義ファイルに、スケールアウト処理またはリバランス処理後のクラスタ構成情報が反映されていない可能性があります。縮退した EADS サーバに、正常に稼働している EADS サーバのクラスタ定義ファイルをコピーしてから、復旧処理を再実行してください。

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（共通設定の `eads.failureDetector.heartbeat.address` パラメタおよび `eads.failureDetector.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

A boot timeout 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。

クラスタを構成する EADS サーバの台数が多い場合に、このエラーが出力されることがあります。この場合は、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.admin.boot.timeout` パラメタの指定）を、「クラスタを構成する EADS サーバ台数×5 秒」に設定し直してください。

また、EADS サーバ ID が小さい EADS サーバから順番に起動してください。

## KDEA04528-E

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ります。

### 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。

< EADS サーバ ID >に、1~96 の整数が指定されているかどうかを確認してください。

## KDEA04531-E

A server position was duplicated. (duplicated node ID1 = aa....aa, duplicated node ID2 = bb....bb)

aa....aa：位置が重複した EADS サーバの EADS サーバ ID1

bb....bb：位置が重複した EADS サーバの EADS サーバ ID2

### 説明

クラスタを構成する EADS サーバの位置が重複しました。

### 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。

## KDEA04540-E

A client connection address was duplicated. (duplicated node ID1 = aa....aa, duplicated node ID2 = 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 サーバの EADS サーバ ID1

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 サーバの EADS サーバ ID2

### 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ます。

### 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

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ていません。

### 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

Participation in the cluster failed. (error code = aa....aa)

aa....aa : エラーコード

### 説明

クラスタ参加処理に失敗しました。

### 対処

次の手順で対処してください。

1. 必要に応じて、直前に出力されたメッセージを確認して要因を特定し、原因を取り除いてください。
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` コマンドを実行して、起動処理、復旧処理、およびスケールアウト処理を再実行してください。

ただし、キャッシュの再開に失敗したあと、再開に失敗したキャッシュを `eztool deletecache` コマンドで削除していないときは、EADS サーバを復旧できません。キャッシュの再開が失敗したキャッシュを `eztool deletecache` コマンドで削除する必要があります。

## KDEA04664-E

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」以上になっていません。

### 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（共通設定の `eads.replication.factor` パラメタの指定値）を下げてください。

## KDEA04665-E

A server cannot be added when the number of replications is 1.

### 説明

データの多重度が 1 のときは、EADS サーバを追加できません。

### 対処

データの多重度（共通設定の `eads.replication.factor` パラメタの指定値）を見直してください。

データの多重度 1 でクラスタを起動している場合は、必要なデータを書き出してから全 EADS サーバを停止してください。そのあとで、最新のパラメタの設定にあわせてクラスタ定義を書き換え、全 EADS サーバを再起動してください。

## KDEA04666-E

The number of servers in the cluster has already reached the maximum.

### 説明

クラスタ内の EADS サーバの数が、すでに上限に達しています。

### 対処

クラスタを構成する EADS サーバの上限である 96 台がクラスタに参加済みのため、これ以上 EADS サーバを追加できません。

別のクラスタを使用することを検討してください。

## KDEA04667-E

In the current state of the cluster, the server cannot participate in the cluster in the specified start type. (start type = aa....aa, cluster state = bb....bb)

aa....aa : 起動方法

bb....bb : クラスタの状態

### 説明

現在のクラスタの状態では、EADS サーバは指定した起動方法でクラスタに参加できません。

### 対処

次の要因が考えられます。

- `ezstart` コマンドまたは `ezserver` コマンドのオプションの指定が誤っている。
- 縮退状態への遷移中に、EADS サーバを再起動、または復旧させた。
- クラスタの状態がクラスタ稼働中 (AVAILABLE) ではない。
- EADS サーバ追加時に、縮退状態の EADS サーバが存在する。

EADS サーバ起動時のオプションの指定誤りや、指定忘れがないかどうかを確認してください。また、クラスタの状態が、指定した起動方法に必要な条件を満たしているかどうかを確認してから、再度 EADS サーバを起動してください。

## KDEA04668-E

The server could not participate in the cluster with the conditions specified in the options. (specified options = aa....aa, details = bb....bb)

aa....aa : EADS サーバ起動時に指定したオプション

bb....bb : 詳細情報

### 説明

オプションで指定した条件では、EADS サーバがクラスタに参加できませんでした。

### 対処

次の要因が考えられます。

- 指定した EADS サーバ ID に該当する EADS サーバが存在しない。
- 指定した EADS サーバの位置 (ハッシュ値) に、すでに EADS サーバが存在する。
- 指定した EADS サーバ ID の EADS サーバが担当するレンジの範囲が 1 である。

オプションに指定する値を見直して、EADS サーバを再度追加してください。

## KDEA04669-E

An attempt to lock the server failed.

### 説明

排他の取得に失敗しました。

### 対処

`eztool status -v` コマンドを実行して実行中の処理を確認し、処理が終了するまで待ってから EADS サーバを再起動してください。もし、処理が終了しているのに排他が解除されていない場合は、`eztool unlock` コマンドを実行し、排他を解除してから EADS サーバを再起動してください。

## KDEA04691-I

The transmission 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転送を開始します。

## KDEA04692-I

The cache data was successfully transferred. (cache name = aa....aa)

aa....aa : キャッシュ名

### 説明

キャッシュのデータの転送に成功しました。

## KDEA04693-E

The transmission 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転送に失敗しました。

### 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、問題を取り除いてください。

## KDEA04697-E

Transmission of the data failed because a required server was not found.

### 説明

データの転送に必要な起動中の EADS サーバが存在しないため、転送に失敗しました。  
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 on the server that sent data. (error code = aa....aa)

aa....aa : エラーコード

### 説明

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。

- 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。
- 復旧処理、スケールアウト処理、およびリバランス処理が実行できない状態だった。
- `ezstart` コマンドまたは `ezserver` コマンドのオプション指定に誤りがあった。

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。

## 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、データのコピー元となる EADS サーバの切り替えに失敗しました。

## 対処

直前に出力されたメッセージを確認してください。

### KDEA04722-E

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ます。

## 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。

### 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.failureDetector.connection.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.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。

### 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。

### 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。

### 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。

### 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。

## 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。

## 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。

## 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。

## 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

KDEA04752-I メッセージが出力されて、縮退処理が完了したことを確認してください。そのあとで、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

スケールアウト処理またはリバランス処理中に縮退した EADS サーバを復旧する場合、縮退した EADS サーバのクラスタ定義ファイルに、スケールアウト処理またはリバランス処理後のクラスタ構成情報が反映されていない可能性があります。縮退した EADS サーバに、正常に稼働している EADS サーバのクラスタ定義ファイルをコピーしてから、復旧処理を実行してください。

## 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

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、共通設定の `eads.replication.external.heapsize` パラメタの値を変更してください。

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 ID = bb....bb)

aa....aa : キャッシュ名

bb....bb : レンジ ID

## 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

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、共通設定の `eads.replication.external.heapsize` パラメタの値を変更してください。

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。

## 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。

## KDEA04807-E

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りました。

### 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. (isolated 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 code = aa....aa)

aa....aa：エラーコード

### 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 timed out. (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.replication.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、共通設定の `eads.replication.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、共通設定の `eads.replication.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、共通設定の `eads.replication.external.heapsize` パラメタの値を変更してください。

## KDEA04824-W

During processing to delete part of the history in order to resolve a memory shortage in the history storage area, processing to build a consensus stopped because the memory shortage could not be resolved. (requested size = aa....aa, cache name = bb....bb, range ID = cc....cc)

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 ID

### 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、共通設定の `eads.replication.external.heapsize` パラメタの値を変更してください。

## KDEA04825-E

During processing to delete part of the history in order to resolve a memory shortage in the history storage area, all processing ended abnormally because the memory shortage could not be resolved. (requested size = aa....aa, cache name = bb....bb, range ID = cc....cc)

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 ID

### 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、共通設定の `eads.replication.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、コネクションを切断しました。

### 対処

サーバ定義の `eads.replication.sendQueue.length` パラメタ、または `eads.replication.sendQueue.datasize` 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。

### 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 name = aa....aa, property value = bb....bb, ReceiveBufferSize = cc....cc)

aa....aa : パラメタ名

bb....bb : パラメタの指定値

cc....cc : 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.replication.connection.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 name = aa....aa, property value = bb....bb, SendBufferSize = cc....cc)

aa....aa : パラメタ名

bb....bb : パラメタの指定値

cc....cc : 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.replication.connection.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。

## 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.transfer.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.transfer.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。

## 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。

## KDEA04971-W

An attempt to change the server position stopped.

### 説明

EADS サーバの縮退以外の問題（例：データ総量監視の上限値の縮小に失敗した）が発生したため、EADS サーバの位置の移動を中止しました。

### 対処

直前に出力されたメッセージを確認してください。

## KDEA04972-E

An attempt to change the server position failed because some servers were isolated.

### 説明

複数の EADS サーバが縮退したため、EADS サーバの位置の移動に失敗しました。

### 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。縮退している EADS サーバを復旧させてから、`eztool rebalance` コマンドを再実行してください。

## 22.6 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。
- ストアデータファイル以外のパスを指定した。
- 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。

- ストアデータファイルではない。
- 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。
- 共通設定の `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、共通設定の `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。
- キャッシュファイルを配置するファイルシステムとして、サポートしていないファイルシステムを使用している。

## 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。または、キャッシュファイルが配置しているファイルシステムが、EADS がサポートしているファイルシステムであることを確認してください。EADS がサポートしているファイルシステムについては、「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。

## 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.eviction.policy](#)
- [eads.cache.disk.filesize](#)

- eads.cache.disk.blocksize
- eads.cache.disk.persistentMemory.enable

ディスクキャッシュの格納先として不揮発性メモリを使用する場合は、次に示すキャッシュ定義のパラメタも変更することはできません。

- eads.cache.disk.persistentMemory.device
- eads.cache.disk.persistentMemory.blocksize

eztool resume コマンド実行時に、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
- eads.cache.eviction.keyCount (データ自動削除機能を使用する場合)

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.disk.filenum
- eads.cache.eviction.keyCount (データ自動削除機能を使用する場合)

## 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.eviction.policy
- eads.cache.eviction.keyCount (データ自動削除機能を使用する場合)
- eads.cache.disk.filesize
- eads.cache.disk.blocksize
- eads.cache.disk.filenum
- eads.cache.disk.persistentMemory.enable
- eads.cache.disk.persistentMemory.device
- eads.cache.disk.persistentMemory.blocksize

## 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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合は、キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルの容量が不足していないか確認してください。

## 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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、詳細情報に「The device cannot be used.」が出力されているときは次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - 不揮発性メモリのデバイスファイルが Device DAX モードであること

- ・不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと
- ・不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

## KDEA07117-E

A data discrepancy between the shared property file and the cache file was detected. (cache name = aa....aa, parameter = bb....bb, shared 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：共通設定ファイルの値

dd....dd：キャッシュファイルの値

### 説明

共通設定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、次に示す共通設定のパラメタを変更することはできません。

- [eads.cache.key.maxsize](#)
- [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。

共通設定ファイルの値が誤っている場合は、次に示す共通設定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.key.maxsize](#)
- [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.eviction.policy`
- `eads.cache.disk.filesize`
- `eads.cache.disk.filenum`
- `eads.cache.disk.blocksize`
- `eads.cache.disk.persistentMemory.enable`

ディスクキャッシュの格納先として不揮発性メモリを使用する場合は、次に示すキャッシュ定義のパラメタも変更することはできません。

- `eads.cache.disk.persistentMemory.blocksize`

データ自動削除機能を使用する場合は、次に示すキャッシュ定義のパラメタも変更することはできません。

- `eads.cache.eviction.keyCount`

## 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.eviction.policy
- eads.cache.eviction.keyCount
- eads.cache.disk.filesize
- eads.cache.disk.filenum
- eads.cache.disk.blocksize
- eads.cache.disk.persistentMemory.enable
- eads.cache.disk.persistentMemory.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

「[11.4.1 定義内容を変更する手順](#)」に従って、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 deletectf` 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

## KDEA07201-E

Some data of cache might be inconsistent. (details = aa....aa)

aa....aa : 詳細情報

### 説明

キャッシュの一部のデータは、矛盾しているおそれがあります。

キャッシュの再開が失敗しているため、EADS サーバの復旧に失敗したと考えられます。

### 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、EADS サーバを復旧してください。

## KDEA07202-E

The operation and cache type combination is invalid. (operation = aa....aa, cache name = bb....bb, cache type = cc....cc)

aa....aa : 操作

bb....bb : キャッシュ名

cc....cc : キャッシュタイプ

## 説明

操作とキャッシュタイプの組み合わせが不正です。

メモリキャッシュ以外のキャッシュが存在しているため、EADS サーバの追加に失敗したと考えられます。

## 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、EADS サーバを追加してください。

## KDEA07211-E

```
The number of keys in range and move range exceeds the limit. (range ID = aa....aa, move range ID = bb....bb, move first position = cc....cc, move last position = dd....dd, key count = ee....ee, key count limit = ff....ff)
```

aa....aa：結合先のレンジ ID

bb....bb：結合元のレンジ（移動するレンジ） ID

cc....cc：結合元のレンジ（移動するレンジ）の開始位置（ハッシュ値）

dd....dd：結合元のレンジ（移動するレンジ）の終了位置（ハッシュ値）

ee....ee：結合先のレンジと結合元のレンジ（移動するレンジ）に格納されている key の数

ff....ff：レンジに格納できる key の数の上限値

## 説明

次に示す key の数の合計が、レンジに格納できる key の数の上限値を超えているため、EADS サーバの位置を変更できません。

- 結合先のレンジに格納されている key の数
- 結合元のレンジ（移動するレンジ）に格納されている key の数

## 対処

結合先のレンジまたは結合元のレンジ（移動するレンジ）に格納されている key を削除してから、`eztool rebalance` コマンドを再実行してください。

## KDEA07212-E

```
Memory capacity in range and move range exceeds the limit. (range ID = aa....aa, move range ID = bb....bb, move first position = cc....cc, move last position = dd....dd, memory capacity = ee....ee, memory capacity limit = ff....ff)
```

aa....aa：結合先のレンジ ID

bb....bb：結合元のレンジ（移動するレンジ） ID

cc....cc：結合元のレンジ（移動するレンジ）の開始位置（ハッシュ値）

dd....dd：結合元のレンジ（移動するレンジ）の終了位置（ハッシュ値）

ee....ee：結合先のレンジと結合元のレンジ（移動するレンジ）に格納されているデータのメモリ使用量

ff....ff：レンジに格納できるデータのメモリ使用量の上限値

## 説明

次に示すメモリ使用量の合計が、レンジに格納できるデータのメモリ使用量の上限値を超えているため、EADS サーバの位置を変更できません。

- 結合先のレンジに格納されているデータのメモリ使用量
- 結合元のレンジ（移動するレンジ）に格納されているデータのメモリ使用量

## 対処

結合先のレンジまたは結合元のレンジ（移動するレンジ）に格納されているデータを削除して、メモリ使用量を減らしてから、`eztool rebalance` コマンドを再実行してください。

## KDEA07302-E

The cluster contains a mixture of caches with automatic data deletion and caches without automatic data deletion.

## 説明

クラスタにデータ自動削除機能を使用しているキャッシュと、データ自動削除機能を使用していないキャッシュが混在するため、EADS サーバの追加に失敗しました。

## 対処

データ自動削除機能を使用しているキャッシュと、データ自動削除機能を使用していないキャッシュのどちらかを削除してから、EADS サーバをクラスタに追加してください。

## 22.7 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.command.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。

ログ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。

## 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。

次に示す値を確認して、必要に応じて値を変更してから、再実行してください。

- コマンドの引数で指定したタイムアウト値
- コマンド定義の `eads.command.common.execution.timeout` パラメタ
- `eads.command.<サブコマンド名>.execution.timeout` パラメタ

`eztool resume` コマンドの場合、データが不整合になっているおそれがあります。

そのまま運用を継続するとデータが破壊されたり、消失したりするおそれがあるため、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. キャッシュの再開が失敗したキャッシュを `eztool deletecache` コマンドで削除する。
5. EADS サーバが縮退している場合、EADS サーバを復旧する。
6. キャッシュの再開を、再度実行する。

## 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（16 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。

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。

## 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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、エラー詳細情報に「The device cannot be used.」が出力されているときは、エラーが発生した EADS サーバで次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - ・ 不揮発性メモリのデバイスファイルが Device DAX モードであること
  - ・ 不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと
- 不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

不揮発性メモリのデバイスファイルの指定に問題がない場合は、デバイスファイルを初期化してください。

## 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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、エラー詳細情報に「The device cannot be used.」が出力されているときは、エラーが発生した EADS サーバで次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - 不揮発性メモリのデバイスファイルが Device DAX モードであること
  - 不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと
- 不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

## 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. キャッシュの再開に失敗したキャッシュを `eztool deletecache` コマンドで削除する。
5. EADS サーバが縮退している場合、EADS サーバを復旧する。
6. キャッシュの再開を、再度実行する。

## 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。

### KDEA08081-E

```
Data addition 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

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。

EADS サーバの状態を確認してください。

### KDEA08082-E

```
Data acquisition 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

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。

EADS サーバの状態を確認してください。

## KDEA08083-E

Data deletion 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

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。

EADS サーバの状態を確認してください。

## KDEA08084-E

The command cannot be executed because the command version and server version are different. (command version = aa....aa, server version = bb....bb)

aa....aa : コマンドのバージョン

bb....bb : EADS サーバのバージョン

### 説明

コマンドのバージョンが、EADS サーバのバージョンと不一致であるため、コマンドは実行できません。

### 対処

EADS サーバのバージョンと同じバージョンのコマンドを実行してください。

## KDEA08085-W

No range for the server ID specified in the server ID specification group was found. (server = aa....aa, server ID = bb....bb)

aa....aa : EADS サーバ名

bb....bb : EADS サーバ ID

### 説明

EADS サーバ ID 指定グループで指定された EADS サーバ ID のレンジが存在しません。

### 対処

`eztool import` コマンド、または `eztool importecf` コマンドで指定した EADS サーバ ID 変換ルールが正しいかどうかを確認してください。

クラスタの構成と、各 EADS サーバの EADS サーバ ID を確認してください。

## KDEA08086-E

The execution of the command failed because the cluster configuration was changed while the command was being executed.

### 説明

コマンド実行中にクラスタの構成が変更されたため、コマンドの実行に失敗しました。

### 対処

クラスタの構成が変更される処理（スケールアウト処理など）が終了してから、コマンドを再実行してください。

## KDEA08092-I

Rebalancing was not executed, because the servers in the cluster are already in the appropriate positions.

### 説明

クラスタ内の各 EADS サーバはすでに適切な位置に配置されているため、リバランス処理は実行されませんでした。

## KDEA08093-E

Rebalancing failed. (error details = aa.....aa)

aa.....aa：エラー詳細情報

### 説明

リバランス処理の実行に失敗しました。

### 対処

エラー詳細情報を確認してください。

すべての EADS サーバの key の数や、データ容量の空きが不足しているおそれがあります。必要に応じて、不要なデータを削除して、空き容量を増やしてください。

## KDEA08094-I

Stopping the rebalancing processing is already reserved.

### 説明

リバランス処理の中断はすでに予約されています。

## KDEA08095-I

Rebalancing was stopped.



## 説明

リバランス処理が中断されました。

### KDEA08096-I

```
The rebalancing processing is not running.
```

## 説明

リバランス処理が実行されていません。

### KDEA08098-E

```
The exporting 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エクス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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、エラー詳細情報に「The device cannot be used.」が出力されているときは、エラーが発生した EADS サーバで次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - ・ 不揮発性メモリのデバイスファイルが Device DAX モードであること
  - ・ 不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと
- 不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

### KDEA08099-W

```
Some data was not imported because the number of data in the range has exceeded the limit.  
(cache name = aa....aa)
```

aa....aa：キャッシュ名

## 説明

`eztool import` コマンドでデータをインポートしましたが、レンジのデータ件数がデータ自動削除機能のしきい値（キャッシュ定義の `eads.cache.eviction.keyCount` パラメタ）で指定した上限を超えたため、一部のデータはインポートされませんでした。

## 対処

- 意図しない状態でこのメッセージが出力された場合は、データ自動削除機能のデータ削除のしきい値、およびレンジのデータ件数の上限が想定通りであることを確認してください。
- レンジのすべてのデータの読み込みが必要な場合は、キャッシュを削除したあとで、キャッシュ定義ファイルの `eads.cache.eviction.keyCount` パラメタにレンジに登録されたデータ件数以上の値を指定して、再実行してください。

## KDEA08100-W

```
Some data was not imported because the number of data in the range has exceeded the limit.
(cache name = aa....aa)
```

aa....aa：キャッシュ名

## 説明

`eztool importecf` コマンドでデータをインポートしましたが、レンジのデータ件数がデータ自動削除機能のしきい値（キャッシュ定義の `eads.cache.eviction.keyCount` パラメタ）で指定した上限を超えたため、一部のデータはインポートされませんでした。

## 対処

- 意図しない状態でこのメッセージが出力された場合は、データ自動削除機能のデータ削除のしきい値、およびレンジのデータ件数の上限が想定通りであることを確認してください。
- レンジのすべてのデータの読み込みが必要な場合は、キャッシュを削除したあとで、キャッシュ定義ファイルの `eads.cache.eviction.keyCount` パラメタにレンジに登録されたデータ件数以上の値を指定して、再実行してください。

## 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。

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。

## 対処

サーバ定義の `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。

## 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。

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。

#### 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。

### KDEA08403-E

```
Startup of the server failed.
```

#### 説明

EADS サーバの起動に失敗しました。

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。

#### 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。

### 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。

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。

#### 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。

### 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。

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。

## 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。

## 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。

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。

## 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。

## KDEA08407-E

```
The specified option is invalid. (error details = aa....aa)
```

aa....aa : エラー詳細情報

## 説明

指定されたオプションが不正です。

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。

## 対処

オプションを確認してください。

## 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。

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。

## 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。

## KDEA08409-E

```
An attempt to lock the server failed.
```

## 説明

排他に失敗しました。

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。

## 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。

## 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。

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。

## 対処

オプションを確認してください。

## 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。

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。

## 対処

エラー詳細情報を確認してください。

## KDEA08412-W

```
An attempt to unlock the server failed.
```

## 説明

排他の解除に失敗しました。

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。

## 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。

## 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。

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。

## 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。

## KDEA08414-W

```
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。

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。

## 対処

パラメタに不正な値を指定しています。

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。

## KDEA08415-E

```
The specified option is duplicated. (option = aa....aa)
```

aa....aa : オプション名

#### 説明

指定したオプションが重複しています。

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。

#### 対処

オプションが重複していないか確認してください。

### KDEA08416-E

```
The specified parameter is invalid. (parameter = aa....aa)
```

aa....aa : 引数

#### 説明

指定した引数は不正です。

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。

#### 対処

不正な引数を指定していないか確認してください。

### KDEA08417-W

```
Deletion of a file failed. (warning details = aa....aa)
```

aa....aa : 警告詳細情報

#### 説明

ファイルの削除に失敗しました。

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。

#### 対処

警告詳細情報を確認してください。

必要に応じて、削除できなかったファイルを手動で削除してください。

### KDEA08418-E

```
An attempt to read the file 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

ファイルの読み込みに失敗しました。

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。

## 対処

ファイルが存在するかどうかを確認してください。または、ファイルの権限を確認してください。

## KDEA08420-W

```
The device 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バイスが不正です。

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。

## 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で指定しているデバイスが正しい値かを確認してください。

## 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.admin.operation.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。

`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。

## 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 deletocache` 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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、エラー詳細情報に「The device cannot be used.」が出力されているときは、次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - 不揮発性メモリのデバイスファイルが Device DAX モードであること
  - 不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと

- 不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

不揮発性メモリのデバイスファイルの指定に問題がない場合は、デバイスファイルを初期化してください。

#### 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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、エラー詳細情報に「The device cannot be used.」が出力されているときは、次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - 不揮発性メモリのデバイスファイルが Device DAX モードであること
  - 不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと

- 不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

## 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。

## KDEA08599-E

The command cannot be executed because the command version and server version are different. (command version = aa....aa, server version = bb....bb)

aa....aa : コマンドのバージョン

bb....bb : EADS サーバのバージョン

### 説明

コマンドのバージョンが、EADS サーバのバージョンと不一致であるため、コマンドは実行できません。

### 対処

EADS サーバのバージョンと同じバージョンのコマンドを実行してください。

## KDEA08600-W

No range for the server ID specified in the server ID specification group was found. (server ID = aa....aa)

aa....aa : EADS サーバ ID

### 説明

EADS サーバ ID 指定グループで指定された EADS サーバ ID のレンジが存在しません。

### 対処

`eztool import` コマンド、または `eztool importecf` コマンドで指定した EADS サーバ ID 変換ルールが正しいかどうかを確認してください。

クラスタの構成と、各 EADS サーバの EADS サーバ ID を確認してください。

## KDEA08601-E

The user does not have the write permission for the file. (file path = aa....aa)

aa....aa：ファイルパス

### 説明

ファイルに書き込み権限がありません。

### 対処

ファイルに書き込み権限を追加してください。

## KDEA08602-E

The user does not have the execute permission for the file. (file path = aa....aa)

aa....aa：ファイルパス

### 説明

ファイルに実行権限がありません。

### 対処

ファイルに実行権限を追加してください。

## KDEA08603-E

An attempt to output the configuration after scaling failed. (error details = aa....aa)

aa....aa：エラー詳細情報

### 説明

スケールアウト後、またはリバランス後のクラスタ構成の出力に失敗しました。

### 対処

エラー詳細情報を確認してください。

出力に失敗した場合、現在のクラスタ構成と定義ファイルに記載されているクラスタ構成が異なるおそれがあります。その場合は、正常に出力されたほかの EADS サーバの定義ファイルをコピーするか、スケールアウト後、またはリバランス後のクラスタ構成を定義ファイルに手動で記載してください。

## KDEA08604-E

An attempt to change server positions 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 サーバの位置の変更に失敗しました。

## 対処

エラーの内容を確認してください。

## KDEA08605-I

```
Stopped the processing to change server positions was reserved. (management client = aa....aa)
```

aa....aa : コマンド実行元の IP アドレス

## 説明

EADS サーバの位置の変更の中断が予約されました。

## KDEA08606-I

```
The processing to change server positions was stopped. (management client = aa....aa)
```

aa....aa : コマンド実行元の IP アドレス

## 説明

EADS サーバの位置の変更が中断されました。

## KDEA08607-I

```
The exporting of cache files started. (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のエクスポートを開始しました。

## KDEA08608-E

```
The exporting 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エクス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。

ディスクキャッシュの格納先として不揮発性メモリを使用している場合で、エラー詳細情報に「The device cannot be used.」が出力されているときは、次の点を確認してください。

- キャッシュ定義ファイルの `eads.cache.disk.persistentMemory.device` パラメタに指定した不揮発性メモリのデバイスファイルについて、次の点を確認してください。
  - 不揮発性メモリのデバイスファイルが Device DAX モードであること
  - 不揮発性メモリのデバイスファイルが他のキャッシュで使用されていないこと
- 不揮発性メモリを使用するための前提パッケージがインストールされていることを確認してください。不揮発性メモリを使用するための前提パッケージについては、リリースノートを参照してください。

## KDEA08609-I

```
The exporting of cache files finished. (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のエクスポートが完了しました。

## KDEA08610-W

```
Some data was not imported because the number of data in the range has exceeded the limit.  
(cache name = aa....aa, range ID = bb....bb)
```

aa....aa：キャッシュ名

bb....bb：レンジ ID

## 説明

`eztool import` コマンドでデータをインポートしましたが、レンジのデータ件数がデータ自動削除機能のしきい値（キャッシュ定義の `eads.cache.eviction.keyCount` パラメタ）で指定した上限を超えたため、一部のデータはインポートされませんでした。

## 対処

- 意図しない状態でこのメッセージが出力された場合は、データ自動削除機能のデータ削除のしきい値、およびレンジのデータ件数の上限が想定通りであることを確認してください。

- レンジのすべてのデータの読み込みが必要な場合は、キャッシュを削除したあとで、キャッシュ定義ファイルの `eads.cache.eviction.keyCount` パラメタにレンジに登録されたデータ件数以上の値を指定して、再実行してください。

## KDEA08611-W

Some data was not imported because the number of data in the range has exceeded the limit.  
(cache name = aa....aa, range ID = bb....bb)

aa....aa：キャッシュ名

bb....bb：レンジ ID

### 説明

`eztool importecf` コマンドでデータをインポートしましたが、レンジのデータ件数がデータ自動削除機能のしきい値（キャッシュ定義の `eads.cache.eviction.keyCount` パラメタ）で指定した上限を超えたため、一部のデータはインポートされませんでした。

### 対処

- 意図しない状態でこのメッセージが出力された場合は、データ自動削除機能のデータ削除のしきい値、およびレンジのデータ件数の上限が想定通りであることを確認してください。
- レンジのすべてのデータの読み込みが必要な場合は、キャッシュを削除したあとで、キャッシュ定義ファイルの `eads.cache.eviction.keyCount` パラメタにレンジに登録されたデータ件数以上の値を指定して、再実行してください。

## 22.8 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、「[18.2 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>.maxExecuteThreads` 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, ユーザ関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>.maxExecuteThreads` 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 03-60 以前のバージョンからの移行手順

EADS を 03-60 以前のバージョンからアップデートして使用する場合、パラメタの設定に変更があります。03-60 以前のバージョンからの移行に必要な手順について説明します。

### 付録 B.1 パラメタの移動 (EADS サーバ)

EADS サーバで指定する一部のパラメタについて、指定する定義ファイルが変更になっています。

EADS を 03-60 以前のバージョンからアップデートする場合、次に示すパラメタを、それぞれの定義ファイルに移動してください。

#### ポイント

- パラメタを移動しなかった場合、そのパラメタの設定は無効となり、デフォルトの値で EADS サーバが動作します。
- 名称が変更になったパラメタは、ほかの定義ファイルに移動する際、新しいパラメタ名で指定できます。新しい名称との対応については「付録 B.4 名称が変更になったパラメタの一覧」を参照してください。
- 03-60 以前に使用できたパラメタのうち、無効となったパラメタがあります。これらのパラメタを誤って定義した場合、EADS サーバの正常な動作は保証できません。無効となったパラメタについては「付録 B.5 無効となったパラメタの一覧」を参照してください。

#### (1) サーバ定義ファイルから移動するパラメタ

サーバ定義ファイルから移動するパラメタおよび移動先の定義ファイルについて、次の表に示します。

表 B-1 サーバ定義ファイルから移動するパラメタ

項番	パラメタ名	移動先の定義ファイル
1	eads.consensus.external.heapsize	共通設定ファイル (eads_shared.properties)

#### (2) クラスタ定義ファイルから移動するパラメタ

クラスタ定義ファイルから移動するパラメタおよび移動先の定義ファイルについて、次の表に示します。

表 B-2 クラスタ定義ファイルから移動するパラメタ

項番	パラメタ名	移動先の定義ファイル
1	eads.server.max_key_size	共通設定ファイル (eads_shared.properties)

項番	パラメタ名	移動先の定義ファイル
2	eads.cluster.heartbeat.address	
3	eads.cluster.heartbeat.port	
4	eads.replication.factor	
5	eads.storeData.generation.count	
6	eads.storeData.stop_command.generation.count	

### (3) コマンド定義ファイルの変更

コマンド定義ファイルは、ファイルの名称が変更になっています。

03-60 以前のコマンド定義ファイル (eads\_management.properties) の名称を、新しい名称 (eads\_command.properties) に変更して使用してください。

### (4) 起動設定ファイルから移動するパラメタ

起動設定ファイルから移動するパラメタおよび移動先の定義ファイルについて、次の表に示します。

表 B-3 起動設定ファイルから移動するパラメタ

項番	パラメタ名	移動先の定義ファイル
1	eads.java.log.filesize	サーバ定義ファイル (eads_server.properties)
2	eads.java.log.filenum	
3	eads.java.heapsize	共通設定ファイル (eads_shared.properties)
4	eads.java.external.heapsize	
5	eads.java.permanent.size*	

注※

バージョン 04-50 以降では、eads.java.metaspace.maxsize パラメタとなります。

## 付録 B.2 パラメタの追加

EADS を 03-60 以前のバージョンからアップデートして運用する場合、EADS サーバ、および EADS クライアントの動作に互換性を保つため、次に示すパラメタを各定義ファイルに追加してください。

### (1) サーバ定義ファイルに追加するパラメタ

サーバ定義ファイル (eads\_server.properties) に追加するパラメタと指定値について、次の表に示します。

表 B-4 サーバ定義ファイルに追加するパラメタ

項番	パラメタ名	指定値
1	<code>eads.admin.operation.connection.timeout</code>	サーバ定義の <code>eads.connection.timeout</code> パラメタに設定していた値

## (2) 共通設定ファイルに追加するパラメタ

共通設定ファイル (`eads_shared.properties`) に追加するパラメタと指定値について、次の表に示します。

表 B-5 共通設定ファイルに追加するパラメタ

項番	パラメタ名	指定値
1	<code>eads.cache.limiter.enable</code>	false

## (3) コマンド定義ファイルに追加するパラメタ

コマンド定義ファイル (`eads_command.properties`) に追加するパラメタと指定値について、次の表に示します。

表 B-6 コマンド定義ファイルに追加するパラメタ

項番	パラメタ名	指定値
1	<code>eads.command.compat</code>	0300

## (4) クライアント定義ファイルに追加するパラメタ

クライアント定義ファイル (`eads_client.properties`) に追加するパラメタと指定値について、次の表に示します。

表 B-7 クライアント定義ファイルに追加するパラメタ

項番	パラメタ名	指定値
1	<code>eads.client.compat</code>	0300

## 付録 B.3 パラメタの再設計

次に示すパラメタのデフォルト値が変更になっています。必要に応じて、パラメタに指定する値を再設計してください。

## (1) サーバ定義ファイルに指定していたパラメタ

サーバ定義ファイル (eads\_server.properties) に指定していた次に示すパラメタは、デフォルト値が変更になっています。

### (a) eads.connection.timeout

デフォルト値が 60000 (ミリ秒) に変更になっています。

03-60 で 60000 未満の値を設定していた場合、デフォルト値での使用を推奨します。

### (b) eads.restore.dataSender.send.timeout

デフォルト値が 60000 (ミリ秒) に変更になっています。

03-60 で 60000 未満の値を設定していた場合、デフォルト値での使用を推奨します。

### (c) eads.statistics.interval

デフォルト値が 1 (秒) に変更になっています。

03-60 以前でデフォルト値のまま使用していた場合で、アップデート後も同様の設定で運用するときは、サーバ定義ファイル (eads\_server.properties) に eads.statistics.interval=60 を指定してください。

## (2) クラスタ定義ファイルに指定していたパラメタ

クラスタ定義ファイル (eads\_cluster.properties) に指定していた次に示すパラメタは、デフォルト値が変更になっています。

### (a) eads.cache.prepare.num

デフォルト値を 20 に変更しています。

03-60 以前にデフォルト値のまま使用していた場合で、アップデート後も同様の設定で運用するときは、共通設定ファイル (eads\_shared.properties) に eads.cache.prepare.num=10 を指定してください。

## (3) クライアント定義ファイルに指定していたパラメタ

クライアント定義ファイル (eads\_client.properties) に指定していた次に示すパラメタは、デフォルト値が変更になっています。

### (a) eads.connection.send.timeout

デフォルトの値が 60000 (ミリ秒) に変更になっています。

03-60 で 60000 未満の値を設定していた場合、デフォルト値での使用を推奨します。

## (b) eads.connection.receive.timeout

デフォルト値が 60000（ミリ秒）に変更になっています。

03-60 で 60000 未満の値を設定していた場合、デフォルト値での使用を推奨します。

## 付録 B.4 名称が変更になったパラメタの一覧

次に示すパラメタは、名称が変更になっていますが、03-60 以前のパラメタ名を指定した場合も、正常に動作します。

### 注意事項

名称が変更になったパラメタについては、次の点に注意してください。

- 変更前と変更後の両方のパラメタを指定した場合、変更後のパラメタの指定値が有効となります。
- 変更前の名称で指定したパラメタに誤りがあった場合、出力されるメッセージ（KDEA10001-W メッセージ、KDEA10002-E メッセージ）には、変更後のパラメタ名が出力されます。
- 一部のパラメタは、指定する定義ファイルが変更になっています。「付録 B.1 パラメタの移動（EADS サーバ）」を参照して、パラメタを移動してください。

### (1) サーバ定義ファイルに指定していたパラメタ

サーバ定義ファイル（eads\_server.properties）に指定していたパラメタのうち、名称が変更になったパラメタを次の表に示します。

表 B-8 名称が変更になったサーバ定義のパラメタ

項番	03-60 以前のパラメタ名	変更後のパラメタ名
1	eads.server.max_connections	eads.server.maxConnections
2	eads.server.session.keepAlive.timeout	eads.server.connection.keepAlive.timeout
3	eads.server.cache.max_execute_threads	eads.server.cache.maxExecuteThreads
4	eads.server.function_container.max_execute_threads	eads.server.function.maxExecuteThreads
5	eads.management.port	eads.admin.operation.port
6	eads.logger.user.rotationStyle	eads.user.logger.rotationStyle
7	eads.logger.user.filesize	eads.user.logger.filesize
8	eads.logger.user.filenum	eads.user.logger.filenum
9	eads.logger.userException.rotationStyle	eads.user.logger.exception.rotationStyle

項番	03-60 以前のパラメタ名	変更後のパラメタ名
10	eads.logger.userException.filesize	eads.user.logger.exception.filesize
11	eads.logger.userException.filenum	eads.user.logger.exception.filenum
12	eads.logger.cache.rotationStyle	eads.cache.logger.diskCache.rotationStyle
13	eads.logger.cache.filesize	eads.cache.logger.diskCache.filesize
14	eads.logger.cache.filenum	eads.cache.logger.diskCache.filenum
15	eads.connection.bufferSize	eads.server.connection.bufferSize
16	eads.connection.timeout	eads.server.connection.timeout
17	eads.cluster.boot.timeout	eads.admin.boot.timeout
18	eads.gracefulstop.wait_time	eads.admin.operation.isolate.gracefulstop.waitTime
19	eads.cluster.failureDetector.connect.timeout	eads.failureDetector.connection.timeout
20	eads.cluster.failureDetector.read.timeout	eads.failureDetector.read.timeout
21	eads.cluster.failureDetector.retry	eads.failureDetector.retry
22	eads.cluster.failureDetector.port	eads.failureDetector.port
23	eads.cluster.assertive.threshold	eads.failureDetector.assertive.threshold
24	eads.cluster.heartbeat.interval	eads.failureDetector.heartbeat.interval
25	eads.cluster.heartbeat.timeout	eads.failureDetector.heartbeat.timeout
26	eads.cache.consensus.timeout	eads.replication.consensus.timeout
27	eads.consensus.external.heapsize	eads.replication.external.heapsize*
28	eads.consensus.message.bufferSize	eads.replication.connection.bufferSize
29	eads.consensus.message.send.queueSize	eads.replication.sendQueue.length
30	eads.consensus.message.send.max_queuedatasize	eads.replication.sendQueue.datasize
31	eads.consensus.message.port	eads.replication.port
32	eads.consensus.fillgap.copy.dataSize	eads.replication.fillgap.copy.datasize
33	eads.consensus.fillgap.copy.timeout	eads.replication.fillgap.copy.timeout
34	eads.restore.connection.bufferSize	eads.transfer.connection.bufferSize
35	eads.restore.dataSender.send.timeout	eads.transfer.timeout
36	eads.restore.dataSender.sendInterval	eads.transfer.interval
37	eads.restore.dataSender.sendSize	eads.transfer.datasize
38	eads.restore.port	eads.transfer.port
39	eads.storeData.dir	eads.admin.backup.dir



項番	03-60 以前のパラメタ名	変更後のパラメタ名
40	eads.management.resume.sendInterval	eads.admin.operation.resume.send.interval
41	eads.management.resume.sendSize	eads.admin.operation.resume.send.datasize
42	eads.statistics.backupnum	eads.statistics.filenum

注※

このパラメタは、共通設定ファイル (eads\_shared.properties) に移動になっています。

## (2) クラスタ定義ファイルに指定していたパラメタ

クラスタ定義ファイル (eads\_cluster.properties) に指定していたパラメタのうち、名称が変更になったパラメタを次の表に示します。

表 B-9 名称が変更になったクラスタ定義のパラメタ

項番	03-60 以前のパラメタ名	変更後のパラメタ名
1	eads.server.max_key_size	eads.cache.key.maxsize※
2	eads.cache.prepare.num	eads.replication.preparations※
3	eads.cluster.heartbeat.address	eads.failureDetector.heartbeat.address※
4	eads.cluster.heartbeat.port	eads.failureDetector.heartbeat.port※
5	eads.storeData.generation.count	eads.admin.backup.exportCommand.generation.maxNum※
6	eads.storeData.stop_command.generation.count	eads.admin.backup.stopCommand.generation.maxNum※

注※

これらのパラメタは、共通設定ファイル (eads\_shared.properties) に移動になっています。

## (3) 起動設定ファイルに指定していたパラメタ

起動設定ファイル (eads\_start.properties) に指定していたパラメタのうち、名称が変更になったパラメタを次の表に示します。

表 B-10 名称が変更された起動設定のパラメタ

項番	03-60 以前のパラメタ名	変更後のパラメタ名
1	eads.java.permanent.size	eads.java.permanent.maxsize※

注※

このパラメータは、共通設定ファイル (eads\_shared.properties) に移動になっています。なお、バージョン 04-50 以降では、変更後のパラメータ名は、eads.java.metaspace.maxsize パラメータとなります。

## (4) コマンド定義ファイルに指定していたパラメータ

コマンド定義ファイル (eads\_management.properties) に指定していたパラメータのうち、名称が変更されたパラメータを次の表に示します。

なお、コマンド定義ファイルは、ファイル名を eads\_management.properties から eads\_command.properties に変更しています。

表 B-11 名称が変更されたコマンド定義のパラメータ

項番	03-60 以前のパラメータ名	変更後のパラメータ名
1	eads.logger.dir	eads.command.logger.dir
2	eads.logger.message.rotationStyle	eads.command.logger.message.rotationStyle
3	eads.logger.message.filesize	eads.command.logger.message.filesize
4	eads.logger.message.filenum	eads.command.logger.message.filenum
5	eads.logger.exception.rotationStyle	eads.command.logger.exception.rotationStyle
6	eads.logger.exception.filesize	eads.command.logger.exception.filesize
7	eads.logger.exception.filenum	eads.command.logger.exception.filenum
8	eads.management.compaction.effect.threshold	eads.command.compaction.effect.threshold
9	eads.management.compaction.effect.division	eads.command.compaction.effect.division
10	eads.management.connect.timeout	eads.command.connection.timeout
11	eads.management.long_running.read.timeout	eads.command.common.read.timeout
12	eads.management.long_running.command.timeout	eads.command.common.execution.timeout

## (5) キャッシュ定義ファイルに指定していたパラメータ

キャッシュ定義ファイル(eads\_cache.<キャッシュ名>.properties)に指定していたパラメータのうち、名称が変更されたパラメータを次の表に示します。

表 B-12 名称が変更されたキャッシュ定義のパラメータ

項番	03-60 以前のパラメータ名	変更後のパラメータ名
1	eads.cache.restore.dataSender.sendInterval	eads.cache.disk.transfer.interval
2	eads.cache.restore.dataSender.sendSize	eads.cache.disk.transfer.datasize

## (6) ファンクション定義ファイルに指定していたパラメタ

ファンクション定義ファイル (eads\_function.properties または <任意の文字列>\_function.properties) に指定していたパラメタのうち、名称が変更されたパラメタを次の表に示します。

表 B-13 名称が変更されたファンクション定義のパラメタ

項番	03-60 以前のパラメタ名	変更後のパラメタ名
1	eads.function.<ユーザファンクション名>.max_execute_threads	eads.function.<ユーザファンクション名>.maxExecuteThreads

## (7) クライアント定義ファイルに指定していたパラメタ

クライアント定義ファイル (eads\_client.properties) に指定していたパラメタのうち、名称が変更されたパラメタを次の表に示します。

表 B-14 名称が変更されたクライアント定義のパラメタ

項番	03-60 以前のパラメタ名	変更後のパラメタ名
1	eads.client.node.list	eads.client.connect.servers
2	eads.logger.dir	eads.client.logger.dir
3	eads.logger.message.filesize	eads.client.logger.message.filesize
4	eads.logger.message.filenum	eads.client.logger.message.filenum
5	eads.logger.message.console.enable	eads.client.logger.message.console.enable
6	eads.connection.bufferSize	eads.client.connection.bufferSize
7	eads.connectionPool.poolSize	eads.client.connectionPool.poolSize
8	eads.connectionPool.exceedMaxSizeError.enable	eads.client.connectionPool.exceedMaxSizeError.enable
9	eads.connection.send.timeout	eads.client.connection.send.timeout
10	eads.connection.receive.timeout	eads.client.connection.receive.timeout

## 付録 B.5 無効となったパラメタの一覧

03-60 以前に使用できたパラメタのうち、次に示すパラメタは無効となっています。

これらのパラメタを誤って設定した場合は、EADS サーバおよび EADS クライアントの正常な動作は保証されません。

表 B-15 無効となったパラメタの一覧

項番	定義ファイル名	無効となったパラメタ名
1	サーバ定義ファイル	eads.server.max.datasize.enable
2		eads.server.max.datasize
3		eads.prf.enable
4		eads.cluster.controller.cache.max_execute_threads
5		eads.restore.dataReceiver.read.timeout
6	起動設定ファイル	eads.prf.filesize
7		eads.prf.filenum
8		eads.prf.level
9		eads.prf.bufferSize
10	コマンド定義ファイル	eads.management.read.timeout
11		eads.management.command.timeout
12	クライアント定義ファイル	eads.logger.commTrace.enable
13		eads.logger.commTrace.filesize
14		eads.logger.commTrace.filenum
15		eads.logger.commTrace.commNo.prefix

### (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。

### (ア行)

#### イベントリスナ

EADS サーバでのイベントの発生（クラスタ構成情報の更新）を契機に、ユーザが定義した処理を、EADS クライアント上で実行させるプログラムのことです。

イベントリスナには、次の 2 種類があります。

- クラスタイベントリスナ  
EADS サーバがクラスタに追加された場合に実行させるイベントリスナです。
- ノードイベントリスナ  
EADS サーバの縮退や復旧など、クラスタを構成する EADS サーバの状態が変わった場合に実行させるイベントリスナです。

### (カ行)

#### 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 16 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)

### スケールアウト処理

クラスタを停止させないで、新しい EADS サーバをクラスタに追加する処理のことです。

### 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。

### 不揮発性メモリ

従来のフラッシュメモリよりも高速に読み書きができ、かつ電源が失われても書き込んだデータが保持される不揮発性のメモリ、いわゆるストレージクラスメモリのことです。

EADS では、不揮発性メモリをストレージデバイスとして使用することで、データの格納先として使用できます。EADS が対応する製品については、リリースノートを参照してください。

## (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。

- key または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

## (ラ行)

### リバランス処理

クラスタを停止させないで、クラスタ内の EADS サーバを均等に配置し直すことで、各レンジに格納できる key の数を均等にする処理のことです。

# 索引

## 数字

- 2Way キャッシュ 43
- 2Way キャッシュ [用語解説] 1033

## A

- addClusterEventListener() [CacheManager クラス] 645
- addNodeEventListener() [CacheManager クラス] 646
- AllFailureException クラス 674

## B

- BatchOperationException クラス 673
- BNF 表記の読み方 553

## C

- CacheException クラス 653
- CacheInfo インタフェース 710
- CacheManager クラス 630
- Cache クラス 602
- calcDiskUsageSize() [Store インタフェース] 732
- calcEHeapUsageSize() [Store インタフェース] 731
- ClientInfo インタフェース 693
- close 435
- ClusterEventContext インタフェース 677
- ClusterEventListener インタフェース 675
- ClusterInfo インタフェース 704
- compaction 536
- containsKey() (Key インタフェース指定) [Store インタフェース] 718
- containsKey() (文字列指定) [Store インタフェース] 719
- create 46
- 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] 637
- 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] 635

- create() (EADS クライアント名およびパス指定) [CacheManager クラス] 632
- create() (クライアント定義指定) [CacheManager クラス] 636
- create() (入力ストリーム指定) [CacheManager クラス] 634
- create() (パス指定) [CacheManager クラス] 631
- create() [Cache クラス] 607
- create() [Group インタフェース] 745
- create() [Store インタフェース] 721
- createcache 483
- createGroup() [Group インタフェース] 737
- createGroup() [Store インタフェース] 716
- createKey() [Group インタフェース] 736
- createKey() [Store インタフェース] 715
- createValue() [Store インタフェース] 716

## D

- deletcache 486
- deleteecf 534
- deleteesd 494
- descendingKeylterator() (Key インタフェース指定) [Group インタフェース] 740
- descendingKeylterator() [Group インタフェース] 740
- destroy() [CacheManager クラス] 640
- destroy() [Function インタフェース] 685

## E

- ead\_add\_cluster\_eventlistener() 839
- ead\_add\_node\_eventlistener() 841
- ead\_address 構造体 852
- ead\_cluster\_eventcontext 構造体 855
- ead\_cluster\_eventlistener 構造体 853
- ead\_create() 801
- EAD\_ERROR\_BATCH\_FAILED\_ALL [C 言語] 868
- EAD\_ERROR\_BATCH\_FAILED\_ALL [Java 言語] 670



EAD\_ERROR\_BATCH\_FAILED\_PART [C 言語] 868

EAD\_ERROR\_BATCH\_FAILED\_PART [Java 言語] 670

EAD\_ERROR\_CACHE\_NOT\_CREATED [ユーザファンクション] 768

EAD\_ERROR\_CACHE\_NOT\_NEED\_STOP [C 言語] 858

EAD\_ERROR\_CACHE\_NOT\_NEED\_STOP [Java 言語] 655

EAD\_ERROR\_CACHE\_NOT\_STARTED [C 言語] 858

EAD\_ERROR\_CACHE\_NOT\_STARTED [Java 言語] 654

EAD\_ERROR\_CACHE\_SETTING [ユーザファンクション] 768

EAD\_ERROR\_CLIENT\_BATCH\_CANCEL [C 言語] 868

EAD\_ERROR\_CLIENT\_BATCH\_CANCEL [Java 言語] 669

EAD\_ERROR\_CLIENT\_FINALIZED [Java 言語] 654

EAD\_ERROR\_CLIENT\_OUT\_OF\_MEMORY [C 言語] 868

EAD\_ERROR\_CLIENT\_OUT\_OF\_MEMORY [Java 言語] 669

EAD\_ERROR\_CLIENT [C 言語] 868

EAD\_ERROR\_CLIENT [Java 言語] 669

EAD\_ERROR\_EVENTLISTENER\_EXIST [C 言語] 859

EAD\_ERROR\_EVENTLISTENER\_EXIST [Java 言語] 656

EAD\_ERROR\_EVENTLISTENER\_NOT\_INITIALIZE D [C 言語] 859

EAD\_ERROR\_EVENTLISTENER\_NOT\_INITIALIZE D [Java 言語] 655

EAD\_ERROR\_EVENTLISTENER\_NOT\_NEED\_REMOVE [C 言語] 859

EAD\_ERROR\_EVENTLISTENER\_NOT\_NEED\_REMOVE [Java 言語] 657

EAD\_ERROR\_EXCEED\_MAX\_CONNECTION\_POOL\_SIZE [C 言語] 858

EAD\_ERROR\_EXCEED\_MAX\_CONNECTION\_POOL\_SIZE [Java 言語] 655

EAD\_ERROR\_ILLEGAL\_ARGUMENT [C 言語] 858

EAD\_ERROR\_INIT\_CLUSTERINFO [C 言語] 861

EAD\_ERROR\_INIT\_CLUSTERINFO [Java 言語] 659

EAD\_ERROR\_INIT\_INVALID\_PROPERTY [C 言語] 860

EAD\_ERROR\_INIT\_INVALID\_PROPERTY [Java 言語] 659

EAD\_ERROR\_INIT\_LOGGER [C 言語] 860

EAD\_ERROR\_INIT\_LOGGER [Java 言語] 659

EAD\_ERROR\_INIT\_PROPERTIES [C 言語] 860

EAD\_ERROR\_INIT\_PROPERTIES [Java 言語] 658

EAD\_ERROR\_INIT [C 言語] 860

EAD\_ERROR\_INIT [Java 言語] 658

EAD\_ERROR\_INVALID\_NODE\_ADDRESS [C 言語] 858

EAD\_ERROR\_INVALID\_NODE\_ADDRESS [Java 言語] 655

EAD\_ERROR\_INVALID\_PARAMETER [Java 言語] 654

EAD\_ERROR\_INVALID\_PARAMETER [ユーザファンクション] 767

EAD\_ERROR\_NET\_CLUSTERINFO [C 言語] 863

EAD\_ERROR\_NET\_CLUSTERINFO [Java 言語] 662

EAD\_ERROR\_NET\_CONNECTION [C 言語] 863

EAD\_ERROR\_NET\_CONNECTION [Java 言語] 662

EAD\_ERROR\_NET\_PROTOCOL [C 言語] 863

EAD\_ERROR\_NET\_PROTOCOL [Java 言語] 662

EAD\_ERROR\_NET\_RECEIVE\_RESPONSE [C 言語] 862

EAD\_ERROR\_NET\_RECEIVE\_RESPONSE [Java 言語] 660

EAD\_ERROR\_NET\_RECEIVE\_TIMEOUT [C 言語] 862

EAD\_ERROR\_NET\_RECEIVE\_TIMEOUT [Java 言語] 661

EAD\_ERROR\_NET\_SEND\_REQUEST [C 言語] 861

EAD\_ERROR\_NET\_SEND\_REQUEST [Java 言語] 660

EAD\_ERROR\_NET\_SEND\_TIMEOUT [C 言語] 862

EAD\_ERROR\_NET\_SEND\_TIMEOUT [Java 言語] 661

EAD\_ERROR\_NET\_TIMEOUT [C 言語] 862

EAD\_ERROR\_NET\_TIMEOUT [Java 言語] 661

EAD\_ERROR\_NET [C 言語] 861

EAD\_ERROR\_NET [Java 言語] 660

EAD\_ERROR\_NOT\_DESERIALIZABLE [Java 言語] 654

EAD\_ERROR\_NOT\_DESERIALIZABLE [ユーザファンクション] 767

EAD\_ERROR\_NOT\_SERIALIZABLE [Java 言語] 654

EAD\_ERROR\_NOT\_SERIALIZABLE [ユーザファンクション] 767

EAD\_ERROR\_SERVER\_CACHE\_AFTER\_REPLICATION [C 言語] 866

EAD\_ERROR\_SERVER\_CACHE\_AFTER\_REPLICATION [Java 言語] 666

EAD\_ERROR\_SERVER\_CACHE\_AFTER\_REPLICATION [ユーザファンクション] 769

EAD\_ERROR\_SERVER\_CACHE\_BEFORE\_REPLICATION [C 言語] 866

EAD\_ERROR\_SERVER\_CACHE\_BEFORE\_REPLICATION [Java 言語] 666

EAD\_ERROR\_SERVER\_CACHE\_BEFORE\_REPLICATION [ユーザファンクション] 769

EAD\_ERROR\_SERVER\_CACHE\_CLUSTER\_NOT\_AVAILABLE [ユーザファンクション] 769

EAD\_ERROR\_SERVER\_CACHE\_CLUSTER\_UPDATE [C 言語] 866

EAD\_ERROR\_SERVER\_CACHE\_CLUSTER\_UPDATE [Java 言語] 666

EAD\_ERROR\_SERVER\_CACHE\_NOT\_FOUND [C 言語] 866

EAD\_ERROR\_SERVER\_CACHE\_NOT\_FOUND [java 言語] 666

EAD\_ERROR\_SERVER\_CACHE [C 言語] 865

EAD\_ERROR\_SERVER\_CACHE [Java 言語] 665

EAD\_ERROR\_SERVER\_CACHE [ユーザファンクション] 769

EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST [C 言語] 865

EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST [Java 言語] 664

EAD\_ERROR\_SERVER\_CREATE\_METHOD\_KEY\_EXIST [ユーザファンクション] 768

EAD\_ERROR\_SERVER\_FUNCTION\_ARG\_DESERIALIZE [Java 言語] 668

EAD\_ERROR\_SERVER\_FUNCTION\_EXECUTE [C 言語] 867

EAD\_ERROR\_SERVER\_FUNCTION\_EXECUTE [Java 言語] 667

EAD\_ERROR\_SERVER\_FUNCTION\_NOT\_FOUND [C 言語] 867

EAD\_ERROR\_SERVER\_FUNCTION\_NOT\_FOUND [java 言語] 668

EAD\_ERROR\_SERVER\_FUNCTION\_RETURN\_SERIALIZE [C 言語] 867

EAD\_ERROR\_SERVER\_FUNCTION\_RETURN\_SERIALIZE [Java 言語] 667

EAD\_ERROR\_SERVER\_INCOMPATIBLE\_CLUSTER\_INFO [C 言語] 864

EAD\_ERROR\_SERVER\_INCOMPATIBLE\_CLUSTER\_INFO [Java 言語] 664

EAD\_ERROR\_SERVER\_LIMIT\_CACHE\_FILE [C 言語] 867

EAD\_ERROR\_SERVER\_LIMIT\_CACHE\_FILE [Java 言語] 668

EAD\_ERROR\_SERVER\_LIMIT\_CACHE\_FILE [ユーザファンクション] 769

EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY\_TEMPORARILY [C 言語] 868

EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY\_TEMPORARILY [Java 言語] 669

EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY\_TEMPORARILY [ユーザファンクション] 770

EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY [C 言語] 867

EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY [Java 言語] 668

EAD\_ERROR\_SERVER\_LIMIT\_EXTERNAL\_MEMORY [ユーザファンクション] 769

EAD\_ERROR\_SERVER\_LIMIT\_KEY\_VALUE\_LENGTH [C 言語] 867

EAD\_ERROR\_SERVER\_LIMIT\_KEY\_VALUE\_LENGTH [Java 言語] 668

EAD\_ERROR\_SERVER\_LIMIT\_KEY\_VALUE\_LENGTH [ユーザファンクション] 770

EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT\_TEMPORARILY [C 言語] 868

EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT\_TEMPORARILY [Java 言語] 669

EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT\_TEMPORARILY [ユーザファンクション] 770

EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT [C 言語] 867

EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT [Java 言語] 668

EAD\_ERROR\_SERVER\_LIMIT\_KV\_COUNT [ユーザファンクション] 770

EAD\_ERROR\_SERVER\_LIMIT\_RECEIVE\_BUFFER [C 言語] 866

EAD\_ERROR\_SERVER\_LIMIT\_RECEIVE\_BUFFER [Java 言語] 667

EAD\_ERROR\_SERVER\_LIMIT\_SEND\_BUFFER [C 言語] 867

EAD\_ERROR\_SERVER\_LIMIT\_SEND\_BUFFER [Java 言語] 667

EAD\_ERROR\_SERVER\_NOT\_EXECUTE\_REQUEST\_TEMPORARILY [C 言語] 865

EAD\_ERROR\_SERVER\_NOT\_EXECUTE\_REQUEST\_TEMPORARILY [Java 言語] 665

EAD\_ERROR\_SERVER\_NOT\_RUNNING [C 言語] 865

EAD\_ERROR\_SERVER\_NOT\_RUNNING [Java 言語] 665

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST [C 言語] 864

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST [Java 言語] 664

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_KEY\_NOT\_EXIST [ユーザファンクション] 768

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED [C 言語] 864

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED [Java 言語] 664

EAD\_ERROR\_SERVER\_REPLACE\_METHOD\_NOT\_MATCHED [ユーザファンクション] 768

EAD\_ERROR\_SERVER\_STATUS [C 言語] 865

EAD\_ERROR\_SERVER\_STATUS [Java 言語] 665

EAD\_ERROR\_SERVER\_UNAVAILABLE [C 言語] 864

EAD\_ERROR\_SERVER\_UNAVAILABLE [Java 言語] 664

EAD\_ERROR\_SERVER\_UNKNOWN [C 言語] 868

EAD\_ERROR\_SERVER\_UNKNOWN [Java 言語] 669

EAD\_ERROR\_SERVER\_UNSUPPORTED\_REQUEST [C 言語] 864

EAD\_ERROR\_SERVER\_UNSUPPORTED\_REQUEST [Java 言語] 663

EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST [C 言語] 865

EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST [Java 言語] 664

EAD\_ERROR\_SERVER\_UPDATE\_METHOD\_KEY\_NOT\_EXIST [ユーザファンクション] 768

EAD\_ERROR\_SERVER [C 言語] 864

EAD\_ERROR\_SERVER [Java 言語] 663

EAD\_ERROR\_SERVER [ユーザファンクション] 768

EAD\_ERROR\_UNEXPECTED [C 言語] 858

EAD\_ERROR\_UNEXPECTED [Java 言語] 654

EAD\_ERROR\_UNEXPECTED [ユーザファンクション] 767

ead\_execute\_function\_rt() 827

ead\_execute\_function() 826

ead\_execute\_node\_function\_rt() 831

ead\_execute\_node\_function() 829

ead\_failure\_operation\_info 構造体 850

ead\_get\_all\_results 構造体 848

ead\_get\_all() 805

ead\_get\_cache\_name() 795  
 ead\_get\_current\_master\_node() 837  
 ead\_get\_group\_count() 817  
 ead\_get\_group\_first\_key() 820  
 ead\_get\_group\_key\_count() 818  
 ead\_get\_group\_keys() 815  
 ead\_get\_group\_names() 813  
 ead\_get\_group\_next\_key() 823  
 ead\_get\_group\_results 構造体 848  
 ead\_get\_group() 807  
 ead\_get\_node\_first\_key() 821  
 ead\_get\_node\_key\_count() 819  
 ead\_get\_node\_keys() 816  
 ead\_get\_node\_next\_key() 824  
 ead\_get\_node() 834  
 ead\_get\_nodelist() 833  
 ead\_get\_original\_master\_node() 838  
 ead\_get\_slave\_nodelist() 835  
 ead\_get() 804  
 ead\_group\_names 構造体 846  
 ead\_init\_client\_n() 792  
 ead\_init\_client() 791  
 ead\_key\_value\_pair 構造体 845  
 ead\_keys 構造体 846  
 ead\_node\_eventcontext 構造体 855  
 ead\_node\_eventlistener 構造体 854  
 ead\_node\_eventtype 構造体 856  
 ead\_nodelist 構造体 851  
 ead\_node 構造体 851  
 ead\_object 構造体 850  
 EAD\_OK [C 言語] 858  
 ead\_put\_all\_results 構造体 847  
 ead\_put\_all() 799  
 ead\_put\_array\_value() 798  
 ead\_put() 797  
 ead\_remove\_all\_results 構造体 849  
 ead\_remove\_all() 809  
 ead\_remove\_cluster\_eventlistener() 843  
 ead\_remove\_group() 811  
 ead\_remove\_node\_eventlistener() 844  
 ead\_remove\_node() 812  
 ead\_remove() 808  
 ead\_replace() 803  
 ead\_start\_cache() 793  
 ead\_stop\_cache() 794  
 ead\_terminate\_client() 796  
 ead\_update() 802  
 ead\_value\_element 構造体 845  
 eads\_[xxx]\_[EADS サーバ ID].esd [ファイル名] 188  
 eads\_cache\_stats\_[n].csv [ファイル名] 173  
 eads\_cache\_stats.csv [ファイル名] 173  
 eads\_cache.<キャッシュ名>.properties [ファイル名] 164  
 eads\_client\_maintenance[n].log [ファイル名] 227  
 eads\_client\_message[n].log [ファイル名] 227  
 eads\_cluster.properties [ファイル名] 164  
 eads\_command\_exception.log [ファイル名] 172  
 eads\_command\_exception[n].log [ファイル名] 172  
 eads\_command\_maintenance[n].log [ファイル名] 172  
 eads\_command\_message.log [ファイル名] 172  
 eads\_command\_message[n].log [ファイル名] 172  
 eads\_command.properties [ファイル名] 164  
 eads\_data\_[EADS サーバ ID]\_[キャッシュ名]\_[レンジ ID]\_[nnnnn].ecf 194  
 eads\_dist\_maintenance[n].log [ファイル名] 172  
 eads\_function\_stats\_[n].csv [ファイル名] 173  
 eads\_function\_stats.csv [ファイル名] 173  
 eads\_index\_[EADS サーバ ID]\_[キャッシュ名]\_[レンジ ID]\_[nnnnn].ecf 195  
 eads\_info\_[EADS サーバ ID]\_[キャッシュ名].ecf [ファイル名] 195  
 eads\_maintenance\_stats\_[n].csv [ファイル名] 173  
 eads\_maintenance\_stats.csv [ファイル名] 173

eads\_nonblocking\_stats\_[n].csv [ファイル名] 173  
eads\_nonblocking\_stats.csv [ファイル名] 173  
eads\_server\_cache.log [ファイル名] 172  
eads\_server\_cache[n].log [ファイル名] 172  
eads\_server\_exception.log [ファイル名] 172  
eads\_server\_exception[n].log [ファイル名] 172  
eads\_server\_maintenance[n].log [ファイル名] 172  
eads\_server\_message.log [ファイル名] 172  
eads\_server\_message[n].log [ファイル名] 172  
eads\_server.properties [ファイル名] 164  
eads\_shared.properties [ファイル名] 164  
eads\_single\_[xxx]\_[EADS サーバID].esd [ファイル名] 188  
eads\_stats\_[n].csv [ファイル名] 173  
eads\_stats.csv [ファイル名] 173  
eads\_stop\_[YYYYMMDDhhmmss]\_[EADS サーバID].esd [ファイル名] 188  
eads\_store\_stats\_[n].csv [ファイル名] 173  
eads\_store\_stats.csv [ファイル名] 173  
eads\_user\_exception.log [ファイル名] 172  
eads\_user\_exception[n].log [ファイル名] 172  
eads\_user\_message.log [ファイル名] 172  
eads\_user\_message[n].log [ファイル名] 172  
eads.admin.backup.dir 191  
eads.admin.backup.exportCommand.generation.maxNum 191  
eads.admin.backup.stopCommand.generation.maxNum 192  
eads.admin.boot.timeout 269  
eads.admin.operation.connection.timeout 267  
eads.admin.operation.isolate.gracefulstop.waitTime 269  
eads.admin.operation.port 169  
eads.admin.operation.resume.send.datasize 240  
eads.admin.operation.resume.send.interval 271  
eads.cache.disk.<n>.dir 205  
eads.cache.disk.blocksize 206  
eads.cache.disk.filenum 206  
eads.cache.disk.filesize 206  
eads.cache.disk.getError.isolate.enable 203  
eads.cache.disk.highSpecDrive.enable 209  
eads.cache.disk.info.dir 205  
eads.cache.disk.persistentMemory.blocksize 210  
eads.cache.disk.persistentMemory.device 210  
eads.cache.disk.persistentMemory.enable 210  
eads.cache.disk.transfer.datasize 208  
eads.cache.disk.transfer.interval 207  
eads.cache.eviction.keyCount 205  
eads.cache.eviction.policy 204  
eads.cache.key.maxsize 240  
eads.cache.keyCount 242  
eads.cache.limiter.enable 242  
eads.cache.logger.diskCache.filenum 180  
eads.cache.logger.diskCache.filesize 180  
eads.cache.logger.diskCache.rotationStyle 180  
eads.cache.type 204  
eads.client.<接続先 EADS サーバ>.address 225  
eads.client.<接続先 EADS サーバ>.port 225  
eads.client.batchOperation.unit 277  
eads.client.clusterInfo.update.interval 275  
eads.client.compat 278  
eads.client.connect.servers 225  
eads.client.connection.bufferSize 242  
eads.client.connection.receive.timeout 274  
eads.client.connection.send.timeout 274  
eads.client.connectionPool.exceedMaxSizeError.enable 247  
eads.client.connectionPool.poolSize 246  
eads.client.eventListener.enable 275  
eads.client.eventListener.reconnect.interval 275  
eads.client.logger.dir 229  
eads.client.logger.initErrorOut 229  
eads.client.logger.maintenance.filenum 230  
eads.client.logger.maintenance.filesize 229  
eads.client.logger.message.console.enable 229



eads.client.logger.message.filenum 229  
eads.client.logger.message.filesize 229  
eads.command.<サブコマンド名>.execution.timeout 274  
eads.command.<サブコマンド名>.read.timeout 273  
eads.command.<接続先 EADS サーバ>.address 170  
eads.command.<接続先 EADS サーバ>.admin.operation.port 170  
eads.command.common.execution.timeout 273  
eads.command.common.read.timeout 272  
eads.command.compaction.effect.division 280  
eads.command.compaction.effect.threshold 280  
eads.command.compat 276  
eads.command.connect.sub.servers 169  
eads.command.connection.timeout 272  
eads.command.logger.dir 181  
eads.command.logger.exception.filenum 183  
eads.command.logger.exception.filesize 182  
eads.command.logger.exception.rotationStyle 182  
eads.command.logger.message.filenum 182  
eads.command.logger.message.filesize 182  
eads.command.logger.message.rotationStyle 181  
eads.failureDetector.assertive.threshold 269  
eads.failureDetector.connection.timeout 268  
eads.failureDetector.heartbeat.address 187  
eads.failureDetector.heartbeat.interval 268  
eads.failureDetector.heartbeat.port 187  
eads.failureDetector.heartbeat.timeout 268  
eads.failureDetector.port 169  
eads.failureDetector.read.timeout 268  
eads.failureDetector.retry 269  
eads.function.<ユーザファンクション名>.maxExecuteThreads 246  
eads.java.external.heapsize 241  
eads.java.heapsize 241  
eads.java.log.filenum 181  
eads.java.log.filesize 181  
eads.java.metaspace.maxsize 242  
eads.java.nio.maxCachedBufferSize 240  
eads.logger.dir 177  
eads.logger.exception.filenum 179  
eads.logger.exception.filesize 178  
eads.logger.exception.rotationStyle 178  
eads.logger.message.console.enable 178  
eads.logger.message.filenum 178  
eads.logger.message.filesize 178  
eads.logger.message.rotationStyle 177  
eads.node.< EADS サーバ ID >.address 186  
eads.node.< EADS サーバ ID >.port 187  
eads.node.< EADS サーバ ID >.position 187  
eads.rebalance.transfer.datasize 238  
eads.rebalance.transfer.interval 270  
eads.replication.connection.bufferSize 236  
eads.replication.consensus.timeout 269  
eads.replication.external.heapsize 241  
eads.replication.factor 187  
eads.replication.fillgap.copy.datasize 239  
eads.replication.fillgap.copy.timeout 271  
eads.replication.port 169  
eads.replication.preparations 241  
eads.replication.sendQueue.datasize 237  
eads.replication.sendQueue.length 236  
eads.scaling.eviction.targetKeyCount 219  
eads.scaling.moveData.interval 218  
eads.scaling.moveDataWithUpdateRequest.enable 217  
eads.scaling.requestQueueing.enable 219  
eads.server.address 169  
eads.server.cache.maxExecuteThreads 246  
eads.server.connection.bufferSize 236  
eads.server.connection.keepAlive.timeout 269  
eads.server.connection.timeout 267  
eads.server.eventListener.notify.retry.timeout 271

- eads.server.function.maxExecuteThreads 246
- eads.server.maxConnections 245
- eads.server.nonBlocking.enable 215
- eads.server.nonBlocking.external.heapsize 216, 242
- eads.server.nonBlocking.maxAllocateConnections 215
- eads.server.nonBlocking.processorThreads 215, 246
- eads.server.nonBlocking.selectorThreads 215
- eads.server.port 169
- eads.statistics.compaction.effect.division 181
- eads.statistics.filenum 180
- eads.statistics.interval 180
- eads.statistics.nonBlocking.enable 215
- eads.transfer.connection.bufferSize 238
- eads.transfer.datasize 238
- eads.transfer.interval 270
- eads.transfer.port 169
- eads.transfer.timeout 269
- eads.user.logger.exception.filenum 180
- eads.user.logger.exception.filesize 180
- eads.user.logger.exception.rotationStyle 179
- eads.user.logger.filenum 179
- eads.user.logger.filesize 179
- eads.user.logger.rotationStyle 179
- EADsStoreException クラス 766
- EADS クライアント 37
- EADS クライアント [用語解説] 1033
- EADS サーバ 36
- EADS サーバ ID 指定グループ 68
- EADS サーバの開始 423
- EADS サーバの強制終了 547
- EADS サーバの縮退 548
- EADS サーバの縮退処理 259
- EADS サーバのフォアグラウンドでの開始 426
- EADS サーバ [用語解説] 1033
- EADS サーバを開始する
  - キャッシュを新規に作成する 283
  - 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 291
- EADS サーバを終了する
  - メモリ上のデータを破棄する 298
  - メモリ上のデータをファイルに書き出す 300
- ehjavalog[nn].log [ファイル名] 173
- equals() [Group インタフェース] 735
- equals() [Key インタフェース] 760
- equals() [Value インタフェース] 762
- execfunc 508
- execute() [Function インタフェース] 684
- executeFunction() (EADS サーバおよび受信タイムアウト指定) [Cache クラス] 628
- executeFunction() (EADS サーバ指定) [Cache クラス] 627
- executeFunction() (key またはグループ, および受信タイムアウト指定) [Cache クラス] 626
- executeFunction() (key またはグループ指定) [Cache クラス] 624
- Explicit ヒープ 97
- export 487
- exportecf (キャッシュファイルの書き出し) 527
- ezserver 426
- ezstart 423
- eztool 430
- eztool コマンドのサブコマンド 432

## F

- FailureOperationInfo クラス 651
- firstKey() [Group インタフェース] 737
- FunctionContext インタフェース 685
- Function インタフェース 683

## G

- get 50
- get() (Key インタフェース指定) [Group インタフェース] 750
- get() (文字列指定) [Group インタフェース] 750
- get() [Cache クラス] 609
- get() [Store インタフェース] 724

[getAddress\(\) \[Node クラス\]](#) 650  
[getAddress\(\) \[ServerInfo インタフェース\]](#) 697  
[getAll\(\) \(Set 指定\) \[Cache クラス\]](#) 610  
[getAll\(\) \(グループ指定\) \[Cache クラス\]](#) 611  
[getArgument\(\) \[FunctionContext インタフェース\]](#) 688  
[getCache\(\) \[CacheManager クラス\]](#) 638  
[getCacheDataFileRemainingSize\(\) \[ServerInfo インタフェース\]](#) 701  
[getCacheDataFileSize\(\) \[CacheInfo インタフェース\]](#) 711  
[getCacheDataFilesNumber\(\) \[CacheInfo インタフェース\]](#) 711  
[getCacheDataFileSpecifiedNumber\(\) \[ServerInfo インタフェース\]](#) 702  
[getCacheDataFileSpecifiedSize\(\) \[ServerInfo インタフェース\]](#) 700  
[getCacheDataFileUnusedNumber\(\) \[ServerInfo インタフェース\]](#) 703  
[getCacheInfo\(\) \[ClusterInfo インタフェース\]](#) 706  
[getCacheName\(\) \[ClientInfo インタフェース\]](#) 694  
[getCacheName\(\) \[FunctionContext インタフェース\]](#) 687  
[getCacheNames\(\) \[ClusterInfo インタフェース\]](#) 705  
[getCacheNames\(\) \[ServerInfo インタフェース\]](#) 697  
[getCacheType\(\) \[ServerInfo インタフェース\]](#) 698  
[getClientInfo\(\) \[FunctionContext インタフェース\]](#) 690  
[getClusterEventType\(\) \[ClusterEventContext インタフェース\]](#) 677  
[getClusterInfo\(\) \[FunctionContext インタフェース\]](#) 691  
[getCurrentMasterNode\(\) \[CacheManager クラス\]](#) 643  
[getDiskUsageSize\(\) \(Group インタフェース指定\) \[Store インタフェース\]](#) 730  
[getDiskUsageSize\(\) \(Key インタフェース指定\) \[Store インタフェース\]](#) 730  
[getEHeapSize\(\) \[ServerInfo インタフェース\]](#) 699  
[getEHeapUsageSize\(\) \(Group インタフェース指定\) \[Store インタフェース\]](#) 729  
[getEHeapUsageSize\(\) \(Key インタフェース指定\) \[Store インタフェース\]](#) 728  
[getEHeapUsageSize\(\) \[ServerInfo インタフェース\]](#) 699  
[getErrorCode\(\) \[CacheException クラス\]](#) 653  
[getErrorCode\(\) \[EADsStoreException クラス\]](#) 767  
[getErrorCode\(\) \[FailureOperationInfo クラス\]](#) 652  
[getException\(\) \[FailureOperationInfo クラス\]](#) 652  
[getFailureOperationInfo\(\) \[BatchOperationException クラス\]](#) 674  
[getFirstKey\(\) \(EADS サーバ指定\) \[Cache クラス\]](#) 621  
[getFirstKey\(\) \(グループ指定\) \[Cache クラス\]](#) 621  
[getFunctionArgument\(\) \[ClientInfo インタフェース\]](#) 695  
[getFunctionName\(\) \[InitConfig インタフェース\]](#) 693  
[getFunctionProperty\(\) \[InitConfig インタフェース\]](#) 692  
[getFunctionPropertyNames\(\) \[InitConfig インタフェース\]](#) 692  
[getGroup\(\) \(グループ名指定\) \[Store インタフェース\]](#) 718  
[getGroup\(\) \[Store インタフェース\]](#) 717  
[getGroupCount\(\) \[Cache クラス\]](#) 619  
[getGroupCount\(\) \[Group インタフェース\]](#) 755  
[getGroupCount\(\) \[Store インタフェース\]](#) 727  
[getGroupLayerNames\(\) \[Group インタフェース\]](#) 756  
[getGroupName\(\) \[FunctionContext インタフェース\]](#) 687  
[getGroupNames\(\) \[Group インタフェース\]](#) 755  
[getGroupNames\(\) \[Store インタフェース\]](#) 727  
[getGroupNameSet\(\) \[Cache クラス\]](#) 616  
[getGroupNameSet\(\) \[Group インタフェース\]](#) 756  
[getGroupNameSet\(\) \[Store インタフェース\]](#) 728



getId() [ServerInfo インタフェース] 697  
getInitConfig() [FunctionContext インタフェース]  
690  
getIp() [ClientInfo インタフェース] 694  
getKey() [FailureOperationInfo クラス] 652  
getKeyCount() (EADS サーバ指定) [Cache クラ  
ス] 620  
getKeyCount() (グループ指定) [Cache クラス]  
619  
getKeyCount() [Group インタフェース] 754  
getKeyCount() [Store インタフェース] 726  
getKeyOrGroupName() [ClientInfo インタフェー  
ス] 695  
getKeySet() (EADS サーバ指定) [Cache クラス]  
618  
getKeySet() (グループ指定) [Cache クラス] 617  
getLastUpdateTime() (Key インタフェース指定)  
[Group インタフェース] 754  
getLastUpdateTime() (文字列指定) [Group イン  
タフェース] 753  
getLastUpdateTime() [Store インタフェース]  
725  
getLocalRangeld() [ClusterInfo インタフェース]  
708  
getLogger() [FunctionContext インタフェース]  
689  
getName() [Cache クラス] 604  
getName() [ServerInfo インタフェース] 696  
getNextKey() (EADS サーバおよび key 指定)  
[Cache クラス] 623  
getNextKey() (グループおよび key 指定) [Cache  
クラス] 622  
getNode() [CacheManager クラス] 641  
getNode() [ClusterEventContext インタフェース]  
678  
getNodeEventType() [NodeEventContext イン  
タフェース] 679  
getNodeId() [Node クラス] 649  
getNodeId() [NodeEventContext インタフェース]  
679  
getNodeList() [CacheManager クラス] 641  
getObject() [Value インタフェース] 763  
getOriginalMasterNode() [CacheManager クラ  
ス] 644  
getPid() [ClientInfo インタフェース] 694  
getPosition 473  
getPosition() [ClusterInfo インタフェース] 706  
getPosition() [Group インタフェース] 736  
getPosition() [Key インタフェース] 760  
getPosition() [Node クラス] 650  
getRangeld() (key またはグループ名指定)  
[ClusterInfo インタフェース] 707  
getRangeld() (ポジション指定) [ClusterInfo イン  
タフェース] 707  
getRangeld() [NodeEventContext インタフェー  
ス] 680  
getRemainingAreaSizeOfWritingCacheDataFile(  
) [CacheInfo インタフェース] 712  
getRemainingCacheDataFilesNumber()  
[CacheInfo インタフェース] 713  
getReplicationFactor() [ClusterInfo インタフェー  
ス] 705  
getServerInfo() [FunctionContext インタフェー  
ス] 690  
getServerName() [FunctionContext インタフェー  
ス] 686  
getSlaveNodeList() [CacheManager クラス]  
642  
getStore() (キャッシュ名指定) [FunctionContext  
インタフェース] 689  
getStore() [FunctionContext インタフェース]  
688  
getSuccessOperationNumber()  
[BatchOperationException クラス] 673  
getType() [CacheInfo インタフェース] 711  
getValueUsageSize() (Key インタフェース指定)  
[Group インタフェース] 758  
getValueUsageSize() (文字列指定) [Group インタ  
フェース] 757  
getValueUsageSize() [Group インタフェース]  
758  
get [サブコマンド] 498  
Group インタフェース 733

## H

higherKeyIterator() [Group インタフェース] 741

## I

import 491

importecf 530

init() [Function インタフェース] 684

InitConfig インタフェース 691

InitializeException クラス 670

InternalClientException クラス 671

InternalServerErrorException クラス 671, 771

isEnabled() [Node クラス] 649

isLocalMasterRange() [ClusterInfo インタフェース] 709

isLocalRange() [ClusterInfo インタフェース] 709

isolate 548

## J

javacore[PID].[YYMMDDhhmmss].txt [ファイル名] 173

javalog[nn].log [ファイル名] 173

Java ヒープ 96

Java ログファイル 171

## K

keyIterator() (Key インタフェース指定) [Group インタフェース] 739

keyIterator() [Group インタフェース] 738

Key インタフェース 759

key の一覧表示 468

## L

lastKey() [Group インタフェース] 738

listcache 454

listconf 449

listecf 516

listesd 458

listfunc 504

listgroup 463

listkey 468

log() (形式 1) [UserLogger インタフェース] 764

log() (形式 2) [UserLogger インタフェース] 765

lowerDescendingKeyIterator() [Group インタフェース] 742

## N

NodeEventContext インタフェース 678

NodeEventListener インタフェース 676

Node クラス 648

## O

onEvent() [ClusterEventListener インタフェース] 675

onEvent() [NodeEventListener インタフェース] 676

open 437

## P

PartFailureException クラス 674

put 45

put() (Key インタフェース指定) [Group インタフェース] 744

put() (文字列指定) [Group インタフェース] 743

put() [Cache クラス] 605

put() [Store インタフェース] 720

putAll() [Cache クラス] 606

putStackTrace() (形式 1) [UserLogger インタフェース] 765

putStackTrace() (形式 2) [UserLogger インタフェース] 766

put [サブコマンド] 496

put と replace の違い 48

## R

rebalance 511

remove 51

remove() (Key インタフェース指定) [Group インタフェース] 752

remove() (文字列指定) [Group インタフェース] 751

remove() [Cache クラス] 613

remove() [Store インタフェース] 725  
removeAll() (EADS サーバ指定) [Cache クラス] 615  
removeAll() (Set 指定) [Cache クラス] 613  
removeAll() (グループ指定) [Cache クラス] 614  
removeall [サブコマンド] 502  
removeCache() [CacheManager クラス] 639  
removeClusterEventListener() [CacheManager クラス] 647  
removeNodeEventListener() [CacheManager クラス] 648  
remove [サブコマンド] 500  
replace 47  
replace() (Key インタフェース指定) [Group インタフェース] 749  
replace() (文字列指定) [Group インタフェース] 748  
replace() [Cache クラス] 608  
replace() [Store インタフェース] 723  
resume 525

## S

ServerCommunicationException クラス 672  
ServerInfo インタフェース 695  
size() [Value インタフェース] 763  
snapshot 541  
status 438  
stop 545  
storeusage 476  
Store インタフェース 714

## T

threaddump 540  
toString() [Group インタフェース] 735  
toString() [Key インタフェース] 760  
toString() [Node クラス] 651  
toString() [Value インタフェース] 761

## U

unlock 482

update 46  
update() (Key インタフェース指定) [Group インタフェース] 747  
update() (文字列指定) [Group インタフェース] 746  
update() [Cache クラス] 608  
update() [Store インタフェース] 722  
UserLogger インタフェース 764  
UserOperationException クラス 672, 771

## V

Value インタフェース 761

## い

イベントリスナ 72  
イベントリスナ [用語解説] 1033

## う

運用ディレクトリ 149

## え

永続データに関する情報の一覧表示 516  
永続データの再配置 530

## か

稼働状態 88  
環境依存パラメタ 93

## き

起動シェル 38  
起動ログファイル 171  
キャッシュ 42  
キャッシュインデックスファイル 194  
キャッシュ情報ファイル 194  
キャッシュ定義ファイル 164  
キャッシュデータファイル 194  
キャッシュデータファイルのコンパクション 536  
キャッシュの一覧表示 454  
キャッシュの一覧を表示する 305  
キャッシュの再開 525

キャッシュの削除 486  
キャッシュの作成 483  
キャッシュファイル 194  
キャッシュファイル操作ログファイル 171  
キャッシュファイルの削除 534  
キャッシュ [用語解説] 1033  
共通設定ファイル 164

## <

クライアント定義ファイル 222  
クラスタ 36  
クラスタ一部稼働中 83  
クラスタイベントリスナ 72  
クラスタおよび EADS サーバの状態を確認する 304  
クラスタ稼働中 82  
クラスタ構成情報 77  
クラスタ参加 84  
クラスタ参加予定 84  
クラスタ状態の確認 438  
クラスタ定義ファイル 164  
クラスタ動作不能 83  
クラスタの運用操作 430  
クラスタの終了 545  
クラスタのスケールアウト処理 262  
クラスタの復旧処理 261  
クラスタの閉塞 435  
クラスタの閉塞解除 437  
クラスタのリバランス処理 262  
クラスタ不参加 84  
クラスタへの参加状況 83  
クラスタ [用語解説] 1034  
グループ階層名 67  
グループ名 66  
グループ名の一覧表示 463

## こ

合意メッセージ 54  
更新操作の履歴 97  
更新操作の履歴の補完処理 263

コネクションプール 90  
コマンド定義ファイル 164  
コンパクション 306

## さ

サーバ定義ファイル 164  
最新パラメタの一覧表示 449  
サマリ [実行結果を表示する eztool コマンドのサブコマンド] 550

## し

指定したデータの格納 496  
指定したデータの削除 500  
指定したデータの取得 498  
シフト方式 174  
縮退状態 88  
    障害発生時の対処 400  
障害情報の取得方法 411  
初期化状態 88  
初期化中 88

## す

スケールアウト処理 61  
スケールアウト処理 [用語解説] 1034  
ストアデータファイル 188  
ストアデータファイルキー 188  
ストアデータファイルの一覧表示 458  
ストアデータファイルの削除 494  
スレッドダンプ 171  
スレッドダンプの出力 540  
スレッドダンプの取得方法 411  
スレッドプール 89

## せ

生存確認 78  
生存確認 [用語解説] 1034  
全データの削除 502

## た

滞留中 I/O データ流量制御 214

## ち

チューニングパラメタ 94

## つ

追加中 88

## て

定義ファイル (EADS クライアント) 222

定義ファイル (EADS サーバ) 164

停止状態 88

停止中 88

ディスクキャッシュ 42

ディスクキャッシュ [用語解説] 1034

ディレクトリ構成 [EADS クライアント] 158

ディレクトリ構成 [EADS サーバ] 146

データ自動削除機能 58

データ自動削除機能の運用 389

データ総量監視機能 384

データの書き出し 487

データの格納先の表示 473

データの多重化 75

データの読み込み 491

## と

統計情報ファイル 171

## の

ノードイベントリスナ 72

ノンブロッキング I/O 通信制御 91

ノンブロッキング I/O 通信制御の設定 212

ノンブロッキング I/O 通信バッファ 97

## は

ハートビート 77

ハートビート [用語解説] 1034

排他解除 482

排他制御 56

## ふ

ファンクション定義ファイル 164

不揮発性メモリ [用語解説] 1034

復旧中 88

ブロッキング I/O 通信制御 91, 212

分散保守ログファイル 171

## へ

閉塞状態 88

閉塞中 88

ヘッダ [実行結果を表示する eztool コマンドのサブコマンド] 550

## ほ

保守ログファイル 171

ボディ [実行結果を表示する eztool コマンドのサブコマンド] 550

## め

メイン [実行結果を表示する eztool コマンドのサブコマンド] 550

メッセージログファイル 171

メモリキャッシュ 42

メモリキャッシュ [用語解説] 1035

メモリ構成 96

## ゆ

ユーザファンクション 69

ユーザファンクションの実行 508

ユーザファンクションの実行可否の表示 504

ユーザファンクション [用語解説] 1035

ユーザ例外ログファイル 171

ユーザログファイル 171

## よ

要素名 66

## ら

ラップアラウンド方式 174

## り

- リクエスト・キューイング 219
- リバランス処理 61
- リバランス処理 [用語解説] 1035

## れ

- 例外ログファイル 171
- レスポンス低下
  - 原因の調査方法 414
  - 障害情報の取得 409
- 列挙型 CacheType 772
- 列挙型 ClusterEventType 680
- 列挙型 NodeEventType 681
- レンジ 59
- レンジおよびキャッシュの使用状況の確認 476

## 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 541